Citrix ADC 13.0

Machine translated content

Disclaimer

Contents

Citrix ADC リリースノート

August 12, 2022

リリースノートには、特定のビルドでソフトウェアがどのように変更されたか、およびビルドに存在する既知の問題が記載されています。

リリースノートドキュメントには、次のセクションのすべてまたは一部が含まれています。

- 新機能: ビルドでリリースされた機能強化とその他の変更。
- 修正された問題: ビルドで修正される問題。
- 既知の問題: ビルドに存在する問題。
- 注意点: ビルドを使用する際に留意すべき重要な点です。
- 制限事項: ビルドに存在する制限事項。

> 注
>
> - 問題の説明の下にある [# XXXXXX] ラベルは、Citrix ADC チームが使用する内部追跡 ID です。
> - これらのリリースノートには、セキュリティ関連の修正は記載されていません。セキュリティに関する修正とアドバイスの一覧については、Citrix セキュリティ情報を参照してください。

特定のビルドのリリースノートドキュメントを表示するには、次の表にある対応するリンクをクリックします。ビルドのリリースノートが更新されると、リリースノートのバージョン番号と公開日も更新されます。リリースノートの公開日は、ビルド GA の日付と同じではない場合があります。

Citrix ADC リリース 13.0 のリリースノート	リリースノートの公開日	リリースノートのバージョン
ビルド 87.9	2022 年 8 月 3 日	1.0
ビルド 86.17	2022 年 6 月 17 日	1.0
85.19 をビルドする	2022 年 5 月 25 日	3.0
84.11 をビルドする	2022 年 1 月 31 日	3.0
ビルド 83.29	2021 年 11 月 15 日	1.0
ビルド 82.45	2021 年 7 月 19 日	2.0
79.64 をビルドする	2021 年 4 月 28 日	3.0
ビルド 76.31	2021 年 4 月 06 日	3.0
ビルド 71.44	2021 年 2 月 19 日	4.0
ビルド 67.43	2020 年 11 月 26 日	2.0
ビルド 64.35	2021 年 1 月 21 日	4.0
ビルド 61.48	2020 年 9 月 4 日	3.0

Citrix ADC リリース 13.0 のリリースノート	リリースノートの公開日	リリースノートのバージョン
ビルド 58.32	2020 年 7 月 7 日	1.0
ビルド 52.24	2020 年 7 月 20 日	4.0
Build 47.24	2020 年 4 月 20 日	2.0
Build 41.28	2020 年 4 月 20 日	3.0

Citrix ADC の製品概要

July 15, 2022

このトピックでは、Citrix ADC アプライアンスの基本的な機能と構成の詳細について説明します。ネットワーク機器を設置および構成するシステムおよびネットワーク管理者は、この内容を参照してください。

Citrix ADC について

Citrix ADC アプライアンスは、アプリケーション固有のトラフィックを分析し、Web アプリケーションのレイヤー 4～レイヤー 7（L4～L7）ネットワークトラフィックを、インテリジェントに分散、最適化、および保護するアプリケーションスイッチです。たとえば、Citrix ADC アプライアンスは、長時間持続する TCP 接続の代わりに、個別の HTTP 要求に基づいて負荷分散を行います。負荷分散機能は、サーバーの障害を遅らせ、クライアントとの切断を少なくします。ADC の機能は大まかに次のように分類されます:

1. データの切り替え
2. ファイアウォールのセキュリティ
3. 最適化
4. ポリシーインフラストラクチャ
5. パケットフロー
6. システムの制限

データの切り替え

アプリケーションサーバーの前に Citrix ADC を導入すれば、クライアント要求を送信する方法によって、トラフィックの最適な分散を実現できます。管理者は、HTTP または TCP 要求の本文に含まれる情報と、URL、アプリケーションデータタイプ、または Cookie などの L4～L7 ヘッダー情報に基づいて、アプリケーショントラフィックをセグメント化できます。多数の負荷分散アルゴリズムと広範なサーバーヘルスチェックによって、クライアント要求が適切なサーバーに確実に送信されるので、アプリケーションの可用性が向上します。

ファイアウォールのセキュリティ

Citrix ADC のセキュリティおよび保護機能は、アプリケーションレイヤー攻撃から Web アプリケーションを保護します。ADC アプライアンスでは適正なクライアント要求を許可して、不正な要求をブロックできます。サービス拒否（Denial Of Service：DoS）攻撃に対する防御機能を組み込んでおり、サーバーに大きな負担をかけるアプリケーショントラフィックの適正なサージから保護する機能をサポートしています。組み込まれたファイアウォールは、バッファーオーバーフローの悪用、SQL インジェクション、クロスサイトスクリプト攻撃など、アプリケーション層の攻撃から Web アプリケーションを保護します。また、ファイアウォールは、企業の機密情報と重要な顧客データを保護する、個人情報盗難保護機能を備えています。

最適化

最適化は、SSL（Secure Sockets Layer）処理、データ圧縮、クライアントキープアライブ、TCP バッファリング、サーバーからの静的および動的コンテンツのキャッシュなど、リソースを消費する処理をオフロードします。これにより、サーバーファーム内のサーバーのパフォーマンスが向上し、アプリケーションの処理速度が上昇します。ADC アプライアンスは、複数の透過的な TCP 最適化をサポートして、長い待ち時間と混雑したネットワークリンクによって発生する問題を緩和し、クライアントまたはサーバーの設定変更を行わずにアプリケーションのデリバリーを高速化します。

ポリシーインフラストラクチャ

「ポリシー」は、Citrix ADC のトラフィックフィルタリングと管理の詳細を定義し、「式」と「アクション」の 2 つの部分で構成されます。式は、ポリシーと一致する要求の種類を定義します。アクションは、要求が式と一致した場合に、ADC アプライアンスが実行する処理を示します。たとえば、式で特定の URL パターンをセキュリティ攻撃と一致させ、接続をドロップまたはリセットするよう設定します。各ポリシーには優先度があり、優先度によってポリシーを評価する順序が決定されます。

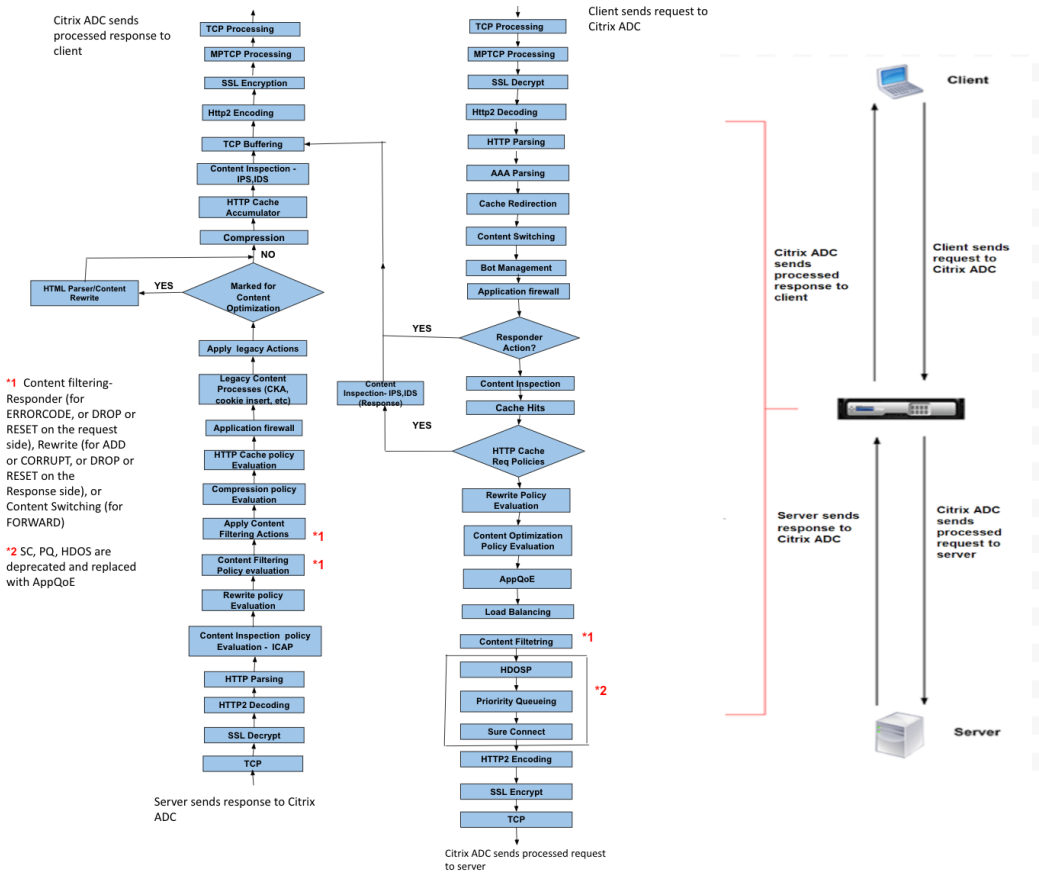
ADC アプライアンスがトラフィックを受信した場合、該当するポリシーの一覧によってトラフィックの処理方法が決定されます。一覧の各ポリシーには 1 つまたは複数の式が含まれており、それらが一緒になって、ポリシーと一致するために接続が満たす必要のある条件を定義します。

書き換えポリシーを除くすべてのポリシータイプで、ADC アプライアンスは要求と一致する最初のポリシーのみを実行します。書き換えポリシーの場合、ADC アプライアンスは順にポリシーを評価し、関連するアクションを順に実行します。必要な結果を得るには、ポリシーの優先度が重要です。

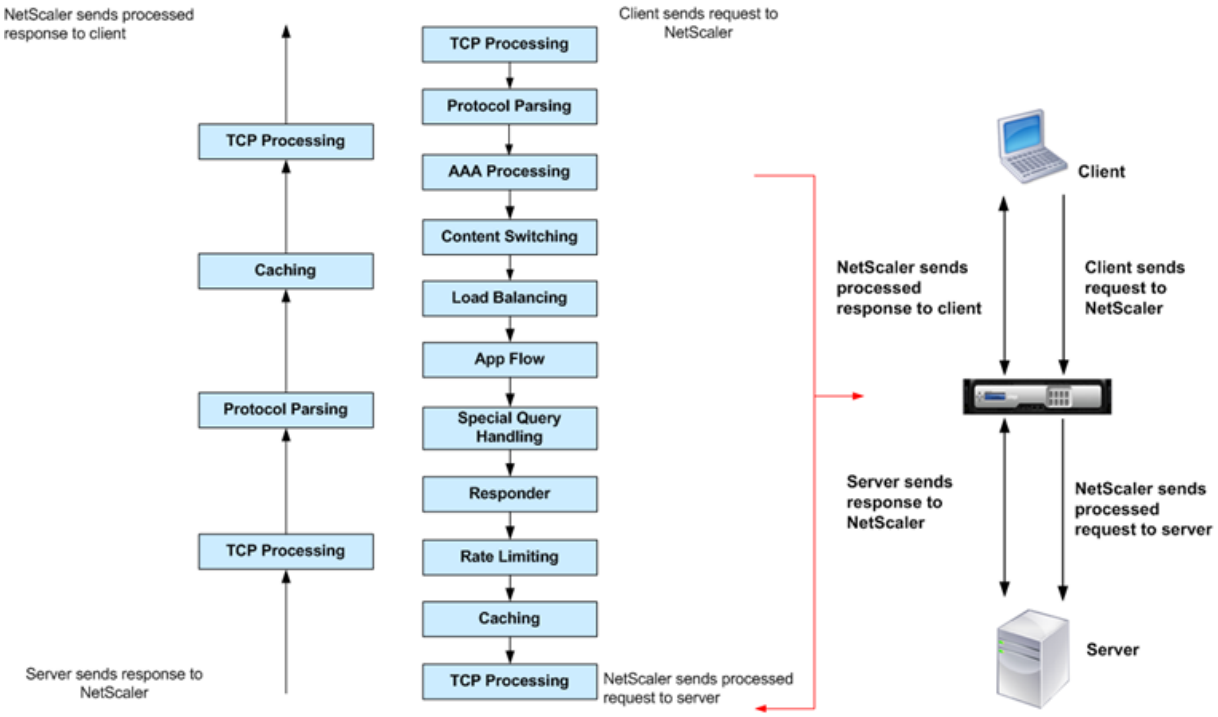
パケットフロー

要件に応じて、複数の機能を構成することを選択できます。たとえば、圧縮と SSL オフロードの両方を構成できます。この場合、発信パケットは圧縮されてから暗号化されて、クライアントに送信されます。

次の図は、Citrix ADC アプライアンスの DataStream パケットフローを示しています。DataStream は、MySQL と MS SQL のデータベースでサポートされています。



次の図は、Citrix ADC アプライアンスの DataStream パケットフローを示しています。DataStream は、MySQL と MS SQL のデータベースでサポートされています。DataStream 機能に関する情報について詳しくは、「DataStream」を参照してください



> 注：コンテンツスイッチ仮想サーバーのトラフィックの場合、アプライアンスは次の順序でポリシーを評価します：
>
> 1. グローバルオーバーライドにバインドされる。
> 2. 負荷分散仮想サーバーにバインドされる。
> 3. コンテンツスイッチ仮想サーバーにバインドされる。
> 4. グローバルデフォルトにバインドされる。
>
> このように、ポリシー規則が true で、gotopriorityexpression が END の場合、それ以上のポリシー評価を停止します。
> コンテンツスイッチでは、負荷分散仮想サーバーが選択されていないか、コンテンツスイッチ仮想サーバーにバインドされていない場合、コンテンツスイッチ仮想サーバーにのみバインドされているレスポンダーポリシーを評価します。

システムの制限

Citrix ADC ソフトウェア 9.2 以降をインストールする場合、Citrix ADC の各機能にはシステム制限があります。詳しくは、Citrix の記事CTX118716を参照してください。

Citrix ADC アプライアンスはネットワークのどこに適合しますか?

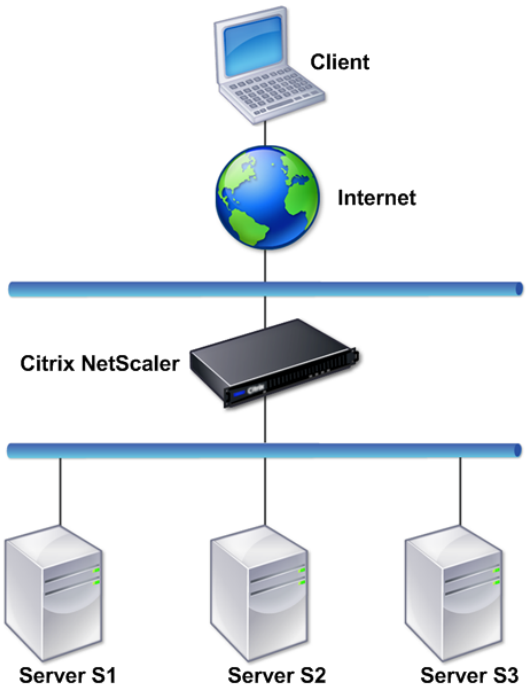
April 25, 2022

NetScaler アプライアンスはクライアントとサーバーの間に設置され、クライアント要求とサーバー応答は Citrix ADC アプライアンスを経由します。一般的な設置では、アプライアンス上で構成された仮想サーバーによって接続ポイントが提供され、クライアントはこれを使用してアプライアンスの背後にあるアプリケーションにアクセスします。この場合、アプライアンスは仮想サーバーに関連付けられたパブリック IP アドレスを所有し、実際のサーバーはプライベートネットワーク内で分離されています。また、アプライアンスを L2 ブリッジや L3 ルーターとして透過モードで動作させたり、これらのモードとそのほかのモードの特徴を組み合わせたりできます。

物理的な展開モード

クライアントとサーバーの間に論理的に設置される Citrix ADC アプライアンスは、インラインまたはワンアームのいずれかの物理モードで展開できます。インラインモードでは、複数のネットワークインターフェイスが異なる Ethernet セグメントに接続され、アプライアンスはクライアントとサーバーの間に配置されます。アプライアンスは、各クライアントネットワークに対する個別のネットワークインターフェイスと、各サーバーネットワークに対する個別のネットワークインターフェイスを持ちます。この構成では、アプライアンスとサーバーを異なるサブネット上に配置できます。アプライアンスの L4～L7 機能を透過的に利用して、サーバーをパブリックネットワーク内に配置し、クライアントがアプライアンスを介してサーバーに直接アクセスするよう構成できます。通常は、実際のサーバーを抽象化した仮想サーバー（後述）を構成します。次の図は、一般的なインライン展開の例を示しています。

図 1: インライン展開



ワンアームモードでは、アプライアンスの 1 つのネットワークインターフェイスのみが、Ethernet セグメントに接続されます。この場合のアプライアンスは、ネットワークのクライアント側とサーバー側を分離せずに、構成済みの仮想サーバーを介してアプリケーションへのアクセスを提供します。一部の環境では、ワンアームモードを使用すると、Citrix ADC の設定に必要なネットワーク変更を簡略化することができます。

インライン（ツーアーム）およびワンアーム展開の例については、「一般的なネットワークトポロジを理解する」を参照してください。

L2 デバイスとしての Citrix ADC

L2 デバイスとして機能する Citrix ADC アプライアンスは、L2 モードで動作すると言われています。L2 モードでは、以下のすべての条件が満たされている場合に、ADC アプライアンスがネットワークインターフェイス間でパケットを転送します。

- パケットの宛先が、別のデバイスの MAC（Media Access Control：メディアアクセスコントロール）アドレスである。
- 宛先 MAC アドレスが別のネットワークインターフェイス上にある。
- ネットワークインターフェイスが、同じ VLAN（Virtual LAN：仮想 LAN）のメンバーである。

デフォルトでは、すべてのネットワークインターフェイスが定義済み VLAN（VLAN 1）のメンバーになります。ARP（Address Resolution Protocol：アドレス解決プロトコル）要求および応答は、同じ VLAN のメンバーであるすべてのネットワークインターフェイスに転送されます。ブリッジループを避けるため、別の L2 デバイスが Citrix ADC アプライアンスと並行して動作している場合、L2 モードを無効にする必要があります。

L2 と L3 モードがどのように相互作用するかについて詳しくは、「パケット転送モード」を参照してください。

L2 モードの構成については、「パケット転送モード」の「レイヤー 2 モードの有効化と無効化」を参照してください。

パケット転送デバイスとしての Citrix ADC

Citrix ADC アプライアンスは、パケット転送デバイスとして機能できます。この動作モードは L3 モードと呼ばれます。L3 モードを有効にすると、アプライアンスに属してない IP アドレス宛のすべてのユニキャストパケットがその宛先に転送されます。アプライアンスは、VLAN 間でパケットをルーティングすることもできます。

通常、L2 と L3 のどちらの動作モードでも、以下に含まれるパケットはアプライアンスによりドロップされます。

- マルチキャストフレーム
- アプライアンスの MAC アドレス（非 IP かつ非 ARP）宛の不明なプロトコルフレーム
- スパニングツリープロトコル（BridgeBPDU がオンになっていない場合）

L2 と L3 モードがどのように相互作用するかについて詳しくは、「パケット転送モード」を参照してください。

L3 モードの構成については、「パケット転送モード」を参照してください。

Citrix ADC アプライアンスとクライアント/サーバーとの通信方法

October 7, 2021

Citrix ADC アプライアンスは通常、サーバーファームの前に展開され、クライアント側で構成を変更しなくても、クライアントとサーバーの透過的な TCP プロキシとして機能します。この基本的な動作モードは「Request Switching 技術」と呼ばれ、Citrix ADC 機能の中核を成しています。Request Switching により、アプライアンスは TCP 接続を多重化してオフロードし、固定接続を維持し、要求（アプリケーションレイヤー）レベルでトラフィックを管理することができます。これらの機能が実現されるのは、アプライアンスが HTTP 要求をその TCP 接続から分離できるからです。

構成によっては、アプライアンスが要求をサーバーに転送する前に、トラフィックを処理する場合があります。たとえば、クライアントがサーバー上の安全なアプリケーションにアクセスしようとする場合に、アプライアンスは必要な SSL 処理を実行してから、トラフィックをサーバーに送信することがあります。

サーバーリソースへの効率的で安全なアクセスを実現するため、アプライアンスは、Citrix ADC 所有 IP アドレスと呼ばれる IP アドレスのセットを使用します。ネットワークトラフィックを管理するには、Citrix ADC 所有 IP アドレスを、構成の構築ブロックになる仮想エンティティに割り当てます。たとえば、負荷分散を構成するには、仮想サーバーを作成し、クライアント要求を受信してサービスに配布します。これらのサービスは、サーバー上のアプリケーションとして振る舞うエンティティです。

Citrix ADC 所有 IP アドレスについて

Citrix ADC アプライアンスでは、プロキシとして機能するためにさまざまな IP アドレス（「Citrix ADC 所有 IP アドレス」）が使用されます。主な Citrix ADC 所有 IP アドレスは、次のとおりです。

- Citrix ADC IP（NSIP）アドレス

  NSIP アドレスは、アプライアンス自体に対する管理アクセスや一般的なシステムアクセス、および高可用性構成のアプライアンス間の通信用の IP アドレスです。

- 仮想サーバー IP（VIP）アドレス

  VIP アドレスは仮想サーバーに関連付けられた IP アドレスです。クライアントが接続するパブリック IP アドレスです。広範なトラフィックを管理するアプライアンスでは、多くの VIP が構成されます。

- サブネット IP（SNIP）アドレス

  SNIP アドレスは、接続の管理とサーバーの監視で使用します。各サブネットに複数の SNIP アドレスを指定できます。SNIP アドレスは VLAN にバインドできます。

- IP セット

  IP セットは、アプライアンス上で SNIP として構成される IP アドレスのセットです。IP セットには、そのセットに含まれる IP アドレスの用途を識別するためのわかりやすい名前を付けます。

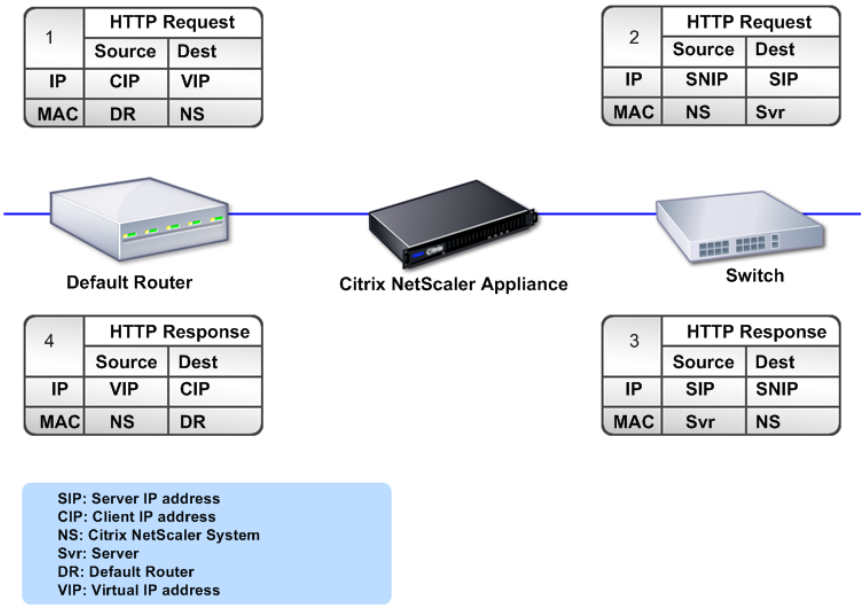
- ネットプロファイル

  ネットプロファイル（ネットワークプロファイル）には、1 つの IP アドレスまたは IP セットが含まれます。ネットプロファイルは負荷分散またはコンテンツスイッチ仮想サーバー、サービス、サービスグループ、またはモニターにバインドされます。アプライアンスが物理サーバーまたはピアと通信するときは、このプロファイルでソース IP アドレスとして指定されているアドレスが使用されます。

トラフィックフローの管理方法

Citrix ADC アプライアンスは TCP プロキシとして機能するので、IP アドレスを変換してから、パケットをサーバーに送信します。仮想サーバーを構成した場合、クライアントはサーバーに直接接続する代わりに Citrix ADC 上の VIP に接続します。仮想サーバーの設定に基づく判断として、アプライアンスは適切なサーバーを選択し、クライアントの要求をそのサーバーに送信します。デフォルトでは、次の図に示すように、アプライアンスは SNIP アドレスを使用して、サーバーとの接続を確立します。

図 1: 仮想サーバーベースの接続



仮想サーバーがない場合、アプライアンスは受信した要求をサーバーへ透過的に転送します。この動作は、透過モードと呼ばれます。透過モードで動作している場合、アプライアンスは、着信したクライアント要求のソース IP アドレスを SNIP アドレスに変換しますが、宛先 IP アドレスは変更しません。このモードが動作するには、L2 または L3 モードが適切に構成されている必要があります。

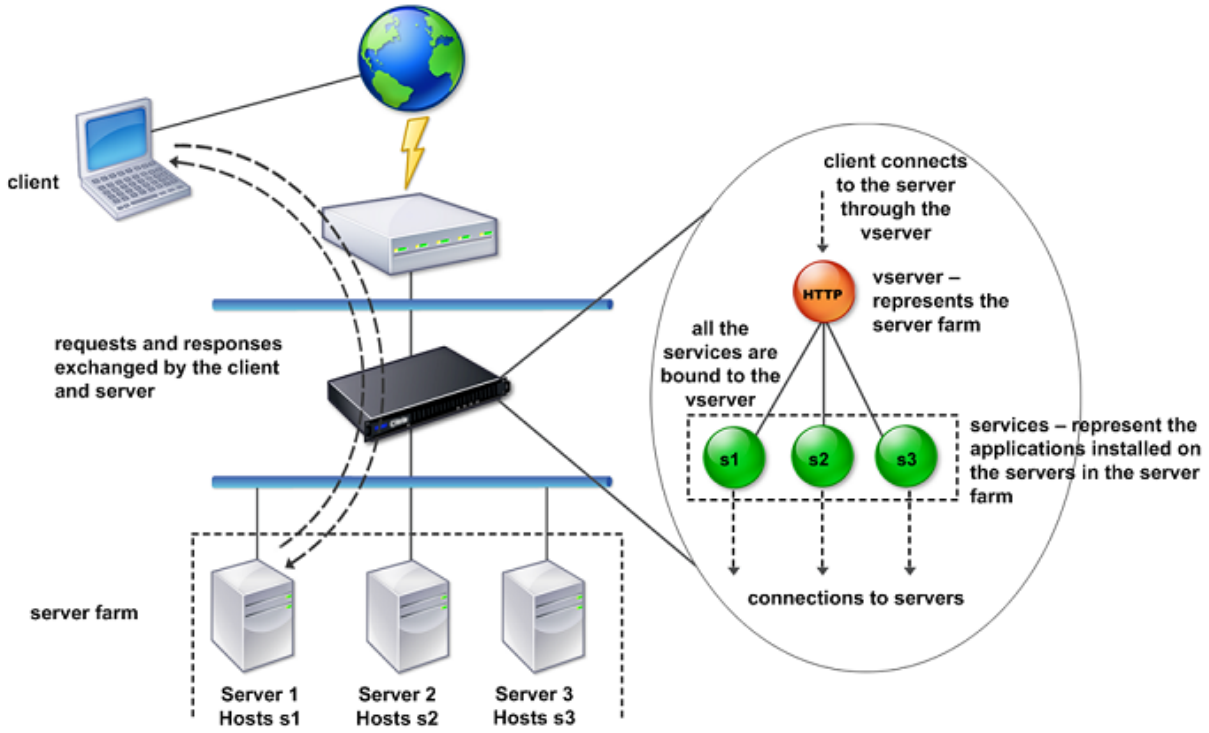
サーバーが実際のクライアント IP アドレスを必要とする場合には、アプライアンスを構成して、クライアント IP アドレスを追加フィールドとして挿入して HTTP ヘッダーを変更するか、またはサーバーとの接続に SNIP ではなくクライアント IP アドレスを使用することができます。

トラフィック管理構築ブロック

通常、Citrix ADC アプライアンスの構成は、トラフィック管理用の構築ブロックとして動作する一連の仮想エンティティで構築されます。この構築ブロックの手法により、トラフィックフローを分離できます。仮想エンティティは抽象型であり、通常、トラフィックを処理するための IP アドレス、ポート、およびプロトコルハンドラーを表しています。クライアントは、これらの仮想エンティティを介して、アプリケーションとリソースにアクセスします。最もよく使用されるエンティティは、「仮想サーバー」と「サービス」です。仮想サーバーはサーバーファームまたはリモートネットワーク内のサーバーグループとして振る舞い、サービスは各サーバー上の個々のアプリケーションとして機能します。

ほとんどの機能とトラフィックの設定値は、仮想エンティティを介して有効化されます。たとえば、特定の仮想サーバー経由でサーバーファームに接続するクライアントへのすべてのサーバー応答が、アプライアンスにより圧縮されるように構成できます。特定の環境に合わせてアプライアンスを構成するには、適切な機能を確認して仮想エンティティの正しい組み合わせを選択し、それらの機能を提供する必要があります。ほとんどの機能は、相互にバインドされた仮想エンティティをカスケードすることで提供されます。この場合の仮想エンティティは、提供されるアプリケーションの最終的な構造に組み込まれるブロックのようなものです。仮想エンティティを追加、削除、変更、バインド、有効化、および無効化して、機能を構成できます。次の図は、ここで説明されている概念を示しています。

図 2: トラフィック管理構築ブロックのしくみ

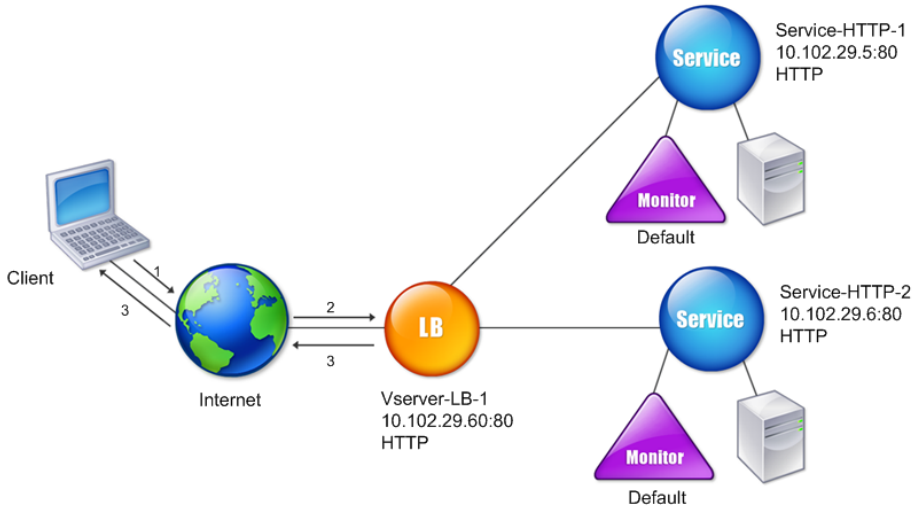


シンプルな負荷分散構成

次の図の例では、Citrix ADC アプライアンスがロードバランサーとして機能するように構成されています。この構成では、負荷分散に固有の仮想エンティティを構成し、それらを特定の順序でバインドする必要があります。ロードバランサーとして機能する場合、アプライアンスはクライアント要求を複数のサーバー間に分散して、リソース使用率を最適化します。

一般的な負荷分散構成の基本的な構築ブロックは、サービスと負荷分散仮想サーバーです。サービスはサーバー上のアプリケーションとして振る舞い、仮想サーバーはクライアントが接続する単一の IP アドレスを提供してサーバーを抽象化します。クライアント要求がサーバーに送信されるようにするため、各サービスを仮想サーバーにバインドする必要があります。つまり、各サーバーに対してサービスを作成し、サービスを仮想サーバーにバインドする必要があります。クライアントは VIP アドレスを使用して Citrix ADC アプライアンスに接続します。アプライアンスは VIP アドレスにクライアント要求を受信すると、負荷分散アルゴリズムによって決定されたサーバーに要求を送信します。負荷分散機能は、モニターと呼ばれる仮想エンティティを使用して、特定の構成済みサービス（サーバーおよびアプリケーション）が要求を受信できるかどうかを追跡します。

図 3：負荷分散仮想サーバー、サービス、およびモニター



負荷分散アルゴリズムを構成するほか、負荷分散構成の動作やパフォーマンスに関する複数のパラメーターを構成できます。たとえば、送信元の IP アドレスに基づいてパーシステンスが維持されるように仮想サーバーを構成できます。この場合、特定の IP アドレスからのすべての要求が同じサーバーに送信されます。

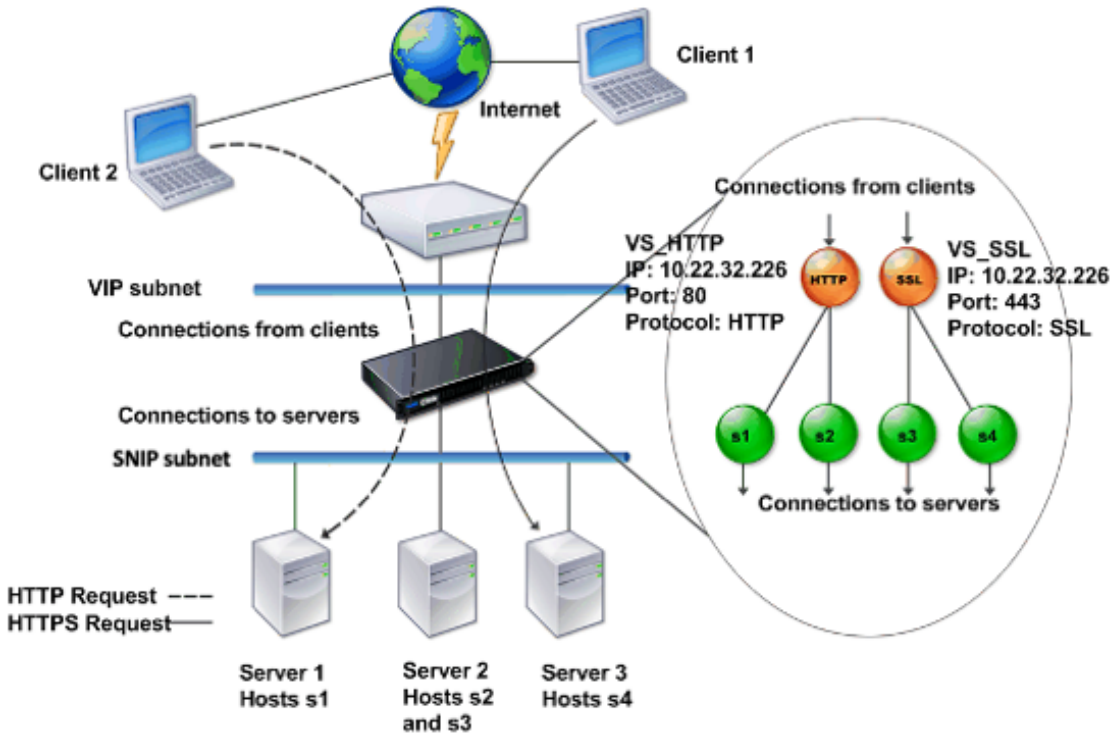
仮想サーバーについて

仮想サーバーは名前付きの Citrix ADC エンティティであり、外部クライアントはそのサーバー上でホストされたアプリケーションにアクセスします。仮想サーバーは英数字名、仮想 IP（VIP）アドレス、ポート、およびプロトコルによって表されます。仮想サーバーの名前はローカル上でのみ意味を持ち、仮想サーバーを識別しやすくするために指定されます。クライアントがサーバー上のアプリケーションにアクセスを試みる場合、クライアントは物理サーバーの IP アドレスではなく、VIP に要求を送信します。アプライアンスが VIP アドレスで要求を受信すると、仮想サーバーでの接続を終了して、クライアントに代わってサーバーとの独自の接続を使用します。仮想サーバーのポートおよびプロトコル設定値によって、その仮想サーバーが振る舞うアプリケーションが決定されます。たとえば、Web サーバーは、ポートとプロトコルがそれぞれ 80 と HTTP に設定された仮想サーバーとサービスによって構成されます。複数の仮想サーバーで同じ VIP アドレスを使用して、異なるプロトコルとポートを使用することもできます。

仮想サーバーは、さまざまな機能の配信ポイントとして動作します。圧縮、キャッシュ、SSL オフロードなどのほとんどの機能は、通常、仮想サーバーで有効になっています。アプライアンスは VIP アドレスで要求を受信すると、要求を受信したポートとそのプロトコルによって、適切な仮想サーバーを選択します。次にアプライアンスは、仮想サーバーに構成されている機能に従って要求を処理します。

ほとんどの場合、仮想サーバーはサービスと協調して動作します。複数のサービスを 1 つの仮想サーバーにバインドすることができます。これらのサービスは、サーバーファーム内の物理サーバーで動作するアプリケーションとして振る舞います。アプライアンスは、VIP アドレスで受信した要求を処理した後、仮想サーバーで設定された負荷分散アルゴリズムの決定に従って、要求をサーバーに転送します。次の図は、これらの概念を示しています。

図 4：単一の VIP アドレスを持つ複数の仮想サーバー



上の図は、VIP アドレスが同じでポートとプロトコルが異なる、2 つの仮想サーバーで構成された環境を示しています。これらの各仮想サーバーには、2 つのサービスがバインドされています。サービス s1 と s2 は VS_HTTP にバインドされており、サーバー 1 とサーバー 2 の HTTP アプリケーションとして動作しています。サービス s3 と s4 は VS_SSL にバインドされており、サーバー 2 とサーバー 3 の SSL アプリケーションとして動作しています（サーバー 2 は、HTTP アプリケーションと SSL アプリケーションの両方を提供します）。アプライアンスが VIP アドレスで HTTP 要求を受信すると、VS_HTTP の設定値により指定されたとして要求を処理し、サーバー 1 またはサーバー 2 に要求を送信します。同様に、アプライアンスが VIP アドレスで HTTPS 要求を受信すると、VS_SSL の設定値により指定されたとして要求を処理し、サーバー 2 またはサーバー 3 に要求を送信します。

仮想サーバーの IP アドレス、ポート番号、またはプロトコルに特定の値を指定せずに、ワイルドカード文字を使用して指定することもできます。このような仮想サーバーは、ワイルドカード仮想サーバーと呼ばれます。たとえば、特定の VIP の代わりにワイルドカード文字を使用し、特定のポート番号で仮想サーバーを構成した場合、アプライアンスは、そのプロトコルおよびポート宛のすべてのトラフィックをインターセプトして処理します。特定の VIP およびポート番号の代わりにワイルドカード文字を使用して仮想サーバーを構成した場合は、そのプロトコルのすべてのトラフィックをインターセプトして処理します。

仮想サーバーは、以下のカテゴリに分類できます。

- 負荷分散仮想サーバー

  要求を受信して、適切なサーバーにリダイレクトします。適切なサーバーの選択は、ユーザーが設定したさまざまな負荷分散方式に基づいて行われます。

- キャッシュリダイレクト仮想サーバー

  動的コンテンツに対するクライアント要求を配信元のサーバーにリダイレクトし、静的コンテンツに対するクライアント要求をキャッシュサーバーにリダイレクトします。キャッシュリダイレクト仮想サーバーは、通常、負荷分散仮想サーバーと一緒に動作します。

- コンテンツスイッチ仮想サーバー

  クライアントが要求したコンテンツに基づいて、トラフィックをサーバーに送信します。たとえば、画像に対するすべてのクライアント要求を、画像のみを処理するサーバーに送信するコンテンツスイッチ仮想サーバーを作成できます。コンテンツスイッチ仮想サーバーは、通常、負荷分散仮想サーバーと一緒に動作します。

- VPN（Virtual Private Network：仮想プライベートネットワーク）仮想サーバー

  トンネリングされたトラフィックを復号化して、イントラネットアプリケーションに送信します。

- SSL 仮想サーバー

  SSL トラフィックを受信して復号化し、適切なサーバーにリダイレクトします。適切なサーバーの選択は、負荷分散仮想サーバーの選択と類似しています。

サービスについて

サービスは、サーバー上のアプリケーションとして機能します。通常、サービスは仮想サーバーと組み合わされていますが、仮想サーバーがなくてもアプリケーション固有のトラフィックを管理できます。たとえば、Citrix ADC アプ

ライアンスで Web サーバーアプリケーションとして振る舞う HTTP サービスを作成できます。この Web サーバーでホストされた Web サイトへのアクセスをクライアントが試みると、アプライアンスが HTTP 要求をインターセプトして Web サーバーとの透過的な接続を作成します。

サービス専用モードでは、アプライアンスがプロキシとして機能します。NetScaler はクライアント接続を終了し、SNIP アドレスを使用してサーバーとの接続を確立し、着信したクライアント要求のソース IP アドレスを SNIP アドレスに変換します。クライアントは要求をサーバーの IP アドレスに直接送信しますが、サーバーは要求が SNIP アドレスから送られてきたものと見なします。アプライアンスは IP アドレス、ポート番号、およびシーケンス番号を変換します。

サービスは、機能を適用するポイントでもあります。SSL Acceleration の例を考えてみましょう。この機能を使用するには、SSL サービスを作成して、そのサービスに SSL 証明書をバインドする必要があります。アプライアンスは HTTPS 要求を受信すると、トラフィックを復号化し、クリアテキストとしてサーバーに送信します。サービス専用モードでは、限られたわずかな機能しか設定できません。

サービスは「モニター」と呼ばれるエンティティを使用して、アプリケーションのヘルスを追跡します。すべてのサービスには、サービスタイプに基づく「デフォルトモニター」がバインドされています。モニターで設定された値に従って、アプライアンスは定期的にアプリケーションにプローブを送信し、アプリケーションの状態を判定します。プローブが失敗した場合、アプライアンスはサービスがダウンしたものとしてマークします。このような場合、アプライアンスは、適切なエラーメッセージでクライアント要求に応答するか、設定された負荷分散ポリシーに従って要求を転送します。

Citrix ADC 製品ラインの概要

April 25, 2022

Citrix ADC の製品ラインは、アプリケーションレベルのセキュリティ、最適化、およびトラフィック管理を単一の統合アプライアンス上に集約して、インターネットおよびプライベートネットワーク経由のアプリケーション配信を最適化します。ユーザーはサーバールームに Citrix ADC アプライアンスを設置し、Citrix ADC アプライアンスを介してすべての接続を管理対象サーバーにルーティングします。次に、有効化された Citrix ADC 機能と設定されたポリシーが、着信および発信トラフィックに適用されます。

Citrix ADC アプライアンスは、既存の負荷分散装置、サーバー、キャッシュ、およびファイアウォールを補完する目的で、ネットワークに統合できます。クライアント側またはサーバー側にソフトウェアを追加する必要はなく、Citrix ADC の Web ベースの GUI および CLI 構成ユーティリティを使用して設定することができます。

このセクションでは、以下のトピックについて説明します:

- Citrix ADC ハードウェアプラットフォーム
- Citrix ADC エディション
- Citrix ADC ハードウェアでサポートされるリリース
- サポートされているブラウザー

Citrix ADC ハードウェアプラットフォーム

Citrix ADC ハードウェアは、さまざまなハードウェア仕様を持つさまざまなプラットフォームで利用できます。

Citrix ADC MPX ハードウェアプラットフォーム

Citrix ADC SDX ハードウェアプラットフォーム

Citrix ADC エディション

Citrix ADC オペレーティングシステムには、次の 3 つのエディションがあります。

- Standard
- 詳細設定
- Premium

Standard エディションと Advanced エディションでは、使用できる機能が制限されています。すべてのエディションで機能のライセンスが必要です。

Citrix ADC ソフトウェアエディションの詳細については、Citrix ADC Editions データシートを参照してください。

ライセンスを取得してインストールする方法については、「ライセンス」を参照してください。

Citrix ADC ハードウェアでサポートされているリリース

すべての Citrix ADC ハードウェアプラットフォームおよびこれらのプラットフォームでサポートされているソフトウェアリリースについては、次の互換性マトリックス表を参照してください。

Citrix ADC MPX ハードウェア-ソフトウェア互換性マトリックス

Citrix ADC SDX ハードウェア-ソフトウェア互換性マトリックス

サポートされているブラウザー

Citrix ADC GUI にアクセスするには、ワークステーションにサポートされている Web ブラウザーが必要です。

次の表に、NetScaler GUI バージョン 12.0、12.1、および 13.0 と互換性のあるブラウザーを示します。

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 7 以降	Internet Explorer	11、Edge、それ以降
Windows 7 以降	Mozilla Firefox	45 以降
Windows 7 以降	Chrome	60 以降
MAC	Mozilla Firefox	45 以降
MAC	Safari	10.1.1 以降

Citrix ADC 11.1 と互換性のあるブラウザーのバージョンは次のとおりです。

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 7 以降	Internet Explorer	8、9、10、11、Edge
Windows 7 以降	Mozilla Firefox	45 以降
Windows 7 以降	Chrome	60 以降
MAC	Mozilla Firefox	45 以降
MAC	Safari	10.1.1 以降

ハードウェアをインストールします

October 7, 2021

Citrix ADC アプライアンスをインストールする前に、インストール前のチェックリストを確認してください。

SDX アプライアンスを使用するには、表に記載されているリソースに記載されている手順に従って、以下のタスクを完了する必要があります。指定された順序でタスクを完了します。

タスク

説明

1. 安全、注意、警告、およびその他の情報を読む

   製品を設置する前に、知っておく必要のある注意と危険に関する情報をお読みください。

2. インストールの準備

   新しいアプライアンスを設置する前に、アプライアンスを開梱し、すべての部品が納品されたことを確認し、サイトとラックを準備し、基本的な電気安全上の注意事項に従ってください。

3. ハードウェアをインストールします

   アプライアンスをラックマウントし、トランシーバー（使用可能な場合）を取り付け、アプライアンスをネットワークと電源に接続します。

4. アプライアンスを構成します。

   GUI またはシリアルコンソールを使用して、Citrix ADC アプライアンスの初期設定を構成します。

   これらのタスクを完了するには、次のドキュメントに記載されている手順に従ってください。

   - Citrix ADC MPX ハードウェアのドキュメント
   - Citrix ADC SDX ハードウェアのドキュメント

Citrix ADC アプライアンスにアクセスする

April 25, 2022

Citrix ADC アプライアンスには、コマンドラインインターフェイス（CLI）と GUI の両方があります。GUI には、アプライアンスを構成するための構成ユーティリティと、ダッシュボードと呼ばれる統計ユーティリティがあります。初回のアクセス用に、すべてのアプライアンスには出荷時にデフォルトの Citrix ADC IP アドレス（NSIP）192.168.100.1 とデフォルトのサブネットマスク 255.255.0.0 が割り当てられています。初回構成時に、新しい NSIP アドレスとそのサブネットマスクを割り当てることができます。

複数の Citrix ADC 装置の展開時に IP アドレスの競合が発生した場合は、以下の点について確認してください：

- ネットワーク上の別のデバイスに既に割り当てられている IP アドレスを、NSIP として選択していないかどうか。
- 複数の Citrix ADC アプライアンスに同じ NSIP を割り当てていないかどうか。
- NSIP は、すべての物理ポートでアクセス可能です。Citrix ADC のポートはホストポートであり、スイッチポートではありません。

次の表は、使用可能なアクセス方法の一覧です。

アクセス方法	ポート	デフォルト IP アドレス（必要/不要）
CLI	Console	×
CLI と GUI	イーサネット	☒

コマンドラインインターフェイス

CLI には、ワークステーションをコンソールポートに接続してローカルでアクセスすることも、同じネットワーク上の任意のワークステーションから SSH（Secure Shell）を介して接続してリモートアクセスすることもできます。

コンソールポートを使用したコマンドラインインターフェイスへのログオン

アプライアンスには、ワークステーションに接続するためのコンソールポートがあります。アプライアンスにログオンするには、シリアルクロスオーバーケーブルと、端末エミュレーションプログラムを備えたワークステーションが必要です。

コンソールポートを使用して CLI にログオンするには、次の手順を実行します：

1. コンソールポートをワークステーションのシリアルポートに接続します。詳しくは、「コンソールケーブルの接続」を参照してください。
2. ワークステーションで、ハイパーターミナルまたはその他のターミナルエミュレーションプログラムを起動します。ログオンプロンプトが表示されない場合は、Enter キーを 1 回または数回押すことが必要な場合があり

ます。

3. [User name] にnsrootを入力します。Password にnsrootを入力し、パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。

SSH を使用したコマンドラインインターフェイスへのログオン

SSH プロトコルを使用すると、同じネットワーク上の任意のワークステーションからアプライアンスにリモートアクセスできます。SSH Version 1（SSH1）または SSH Version 2（SSH2）を使用できます。

動作している SSH クライアントがない場合は、以下の SSH クライアントプログラムをダウンロードしてインストールできます。

- PuTTY

  複数のプラットフォームでサポートされている、オープンソースソフトウェアです。次のサイトから入手できます。

  http://www.chiark.greenend.org.uk/~sgtatham/putty/

- Vandyke Software SecureCRT

  Windows プラットフォームでサポートされている、市販のソフトウェアです。次のサイトから入手できます。

  http://www.vandyke.com/products/securecrt/

これらのプログラムは Citrix ADC チームによってテストされ、Citrix ADC アプライアンスと正常に動作することが確認されています。そのほかのプログラムも正常に動作する可能性がありますが、テストは実施されていません。

SSH クライアントが正しくインストールされていることを確認するには、対象のクライアントを使用して、SSH 接続を受け付ける、ネットワーク上の任意のデバイスに接続します。

SSH クライアントを使用して Citrix ADC アプライアンスにログオンするには、次の手順を実行します:

1. ワークステーションで、SSH クライアントを起動します。
2. 初期構成には、デフォルトの IP アドレス（NSIP）である 192.168.100.1 を使用します。その後のアクセスでは、初回構成時に割り当てる NSIP を使用します。プロトコルとして SSH1 または SSH2 のいずれかを選択します。
3. [User name] にnsrootを入力します。Password にnsrootを入力し、パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。たとえば、以下のようなものです。

```
1 login as: nsroot
2
3
4 Using keyboard-interactive authentication.
5
```



```
6
7  Password:
8
9
10 Last login: Tue Jun 16 10:37:28 2009 from 10.102.29.9
11
12
13
14
15
16 Done
17
18
19 >
20
21 <!--NeedCopy-->
```

Citrix ADC GUI

> 重要:
>
> Citric ADC GUI への HTTPS アクセスには、証明書とキーのペアが必要です。ADC では、証明書とキーのペアは内部サービスに自動的にバインドされます。デフォルトのキーサイズは、MPX または SDX アプライアンスで 1024 バイト、VPX インスタンスで 512 バイトです。ただし、最新の Web ブラウザーの多くは 1024 バイト未満のキーを受け入れません。このため、VPX 構成ユーティリティへの HTTPS アクセスがブロックされてしまいます。

また、ライセンスがない状態で MPX アプライアンスを起動して、その後でライセンスを追加してからアプライアンスを再起動すると、証明書のバインドが失われることがあります。

GUI への HTTPS アクセスのために、アプライアンスに少なくとも 1024 バイトの証明書とキーのペアをインストールすることをお勧めします。また、アプライアンスを起動する前に適切なライセンスをインストールしてください。

GUI には、構成ユーティリティと「ダッシュボード」と呼ばれる統計ユーティリティが含まれています。アプライアンスのイーサネットポートに接続されたワークステーションを介して、これらのツールにアクセスします。

GUI を実行するワークステーションのシステム要件は、次のとおりです。

- Windows ベースのワークステーションの場合は、Pentium 166 MHz 以上のプロセッサが必要です。
- Linux ベースのワークステーションの場合は、Linux カーネル v2.2.12 以降と`glibc`バージョン 2.12-11 以降を実行する、Pentium プラットフォームが必要です。32MB 以上の RAM が必要であり、48MB 以上を推奨します。ワークステーションは、ディスプレイをローカルホストに設定し、16 ビットカラーモードで KDE および KWM のウィンドウマネージャーをサポートする必要があります。
- Solaris ベースのワークステーションの場合は、Solaris 2.6、Solaris 7、または Solaris 8 を実行する Sun が必要です。

構成ユーティリティとダッシュボードにアクセスするには、サポートされている Web ブラウザーがワークステーションにインストールされている必要があります。

次のブラウザーがサポートされています。

オペレーティングシステム： Windows 7

ブラウザー：Internet Explorer（バージョン 9、10、11）、Mozilla Firefox（バージョン 3.6.25 以降）、Google Chrome（最新）。

オペレーティングシステム： Windows 64 ビット

ブラウザー：Internet Explorer（バージョン 8、9、10、11）、Google Chrome（最新バージョン）

オペレーティングシステム： MAC
ブラウザー：Mozilla Firefox（バージョン 3.6.25 以降）、Safari（バージョン 5.1.3 以降）、Google Chrome（最新バージョン）

Citrix ADC GUI を使用

構成ユーティリティにログオンしたら、状況依存ヘルプが付属するグラフィックインターフェイスを介して、アプライアンスを構成できます。

GUI にログオンするには、次の手順を実行します：

1. Web ブラウザーを開いて、Citrix ADC IP（NSIP）を HTTP アドレスとして入力します。初回構成がまだ完了していない場合は、デフォルトの NSIP（http://192.168.100.1）を入力します。[Citrix Logon] ページが開きます。

   注：2 台の Citrix ADC アプライアンスが高可用性ペアとしてセットアップされている場合は、GUI にアクセスするときにセカンダリ Citrix ADC アプライアンスの IP アドレスを入力しないでください。このような方法でアクセスすると、GUI を使用してセカンダリアプライアンスを構成しても、その変更内容がプライマリ Citrix ADC アプライアンスに適用されません。

2. [User Name] ボックスに「`nsroot`」と入力します。
3. [Password] ボックスに、初回構成時に`nsroot`アカウントに割り当てた管理パスワードを入力し、[**Login**] をクリックします。パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。

   オンラインヘルプにアクセスするには、右上隅の［Help］メニューの［Help］を選択します。

統計ユーティリティの使用

ダッシュボード（統計ユーティリティ）は、Citrix ADC アプライアンスのパフォーマンスを監視できる図と表を表示する、ブラウザーベースのアプリケーションです。

ダッシュボードにログオンするには、次の手順を実行します：

1. Web ブラウザーを開いて、NSIP を HTTP アドレスとして入力します。［Citrix Logon］ページが開きます。
2. ［User Name］ボックスに「nsroot」と入力します。
3. ［Password］ボックスに、初回構成時にnsrootアカウントに割り当てた管理パスワードを入力します。パスワードが機能しない場合はアプライアンスのシリアル番号を入力します。シリアル番号のバーコードは、アプライアンスの背面にあります。

ADC の初回構成

April 21, 2022

Citrix ADC MPX アプライアンスの初期構成については、「Citrix MPX アプライアンスの初期構成」を参照してください。

Citrix ADC MPX アプライアンスの初期構成については、「Citrix MPX アプライアンスの初期構成」を参照してください。

NITRO API

NITRO API を使用して Citrix ADC アプライアンスを構成できます。NITRO では、Representational State Transfer（REST）インターフェイスを介して機能が提供されます。そのため、NITRO アプリケーションはあらゆるプログラミング言語で開発することができます。さらに、Java、.NET、または Python で開発する必要があるアプリケーションの場合、NITRO API は、個別のソフトウェア開発キット（SDK）としてパッケージ化された関連ライブラリを介して提供されます。詳しくは、「NITRO API」を参照してください。

Citrix ADC の導入を保護する

April 25, 2022

Citrix ADC アプライアンスの展開ライフサイクルを通じてセキュリティを維持するために、次のセキュリティの側面を考慮することをお勧めします。

- 物理的セキュリティ
- アプライアンスのセキュリティ
- Network Security
- 管理と管理

展開が異なれば、セキュリティに関する考慮事項も異なる場合があります。Citrix ADC の安全な導入ガイドラインは、特定のセキュリティ要件に基づいて適切な安全な導入を決定するのに役立つ一般的なセキュリティガイダンスを提供します。

Citrix ADC アプライアンスを安全に展開するためのガイドラインの詳細については、「Citrix ADC の安全な導入ガイドライン」を参照してください。

高可用性の構成

April 25, 2022

2 台の Citrix ADC アプライアンスを高可用性構成で展開できます。この構成では、1 台の装置がアクティブに接続を受け付けてサーバーを管理し、2 台目の装置は 1 台目の装置を監視します。高可用性構成では、アクティブに接続を受け付けてサーバーを管理する Citrix ADC はプライマリ装置と呼ばれ、もう 1 台はセカンダリ装置と呼ばれます。プライマリ装置が故障した場合は、セカンダリ装置がプライマリになって、アクティブに接続の受け付けを開始します。

高可用性ペアの各 Citrix ADC アプライアンスは、「ハートビートメッセージ」または「ヘルスチェック」と呼ばれる定期的なメッセージを送信してもう一方の装置を監視し、ピアノードのヘルスまたは状態を判定します。プライマリ装置のヘルスチェックが失敗した場合、セカンダリ装置は指定された時間、接続を再試行します高可用性について詳しくは、「高可用性」を参照してください。指定された時間内に再試行が成功しない場合、セカンダリ装置は「フェールオーバー」と呼ばれるプロセスによって、プライマリ装置の役割を引き継ぎます。次の図は、2 つの高可用性構成を示しています。1 つはワンアームモード構成で、もう 1 つはツーアームモード構成です。

図 1: ワンアームモードでの高可用性

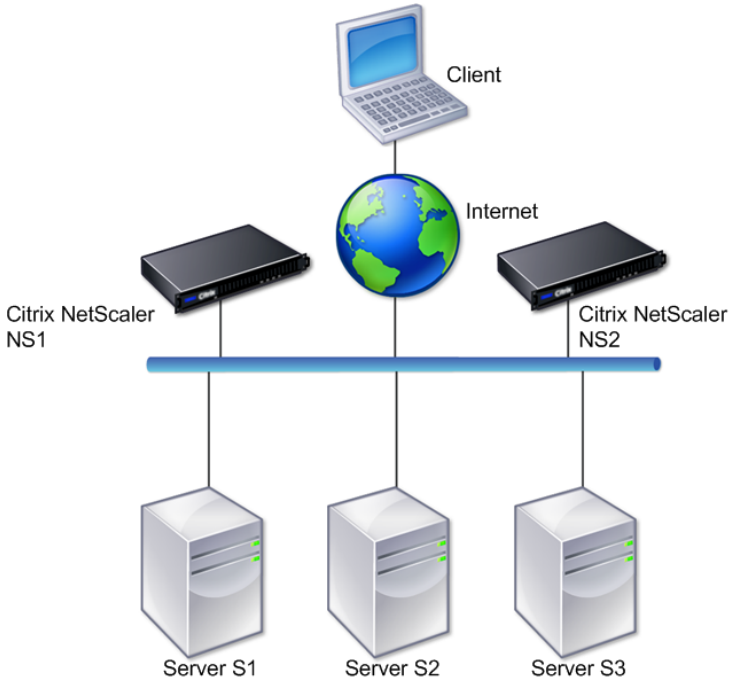
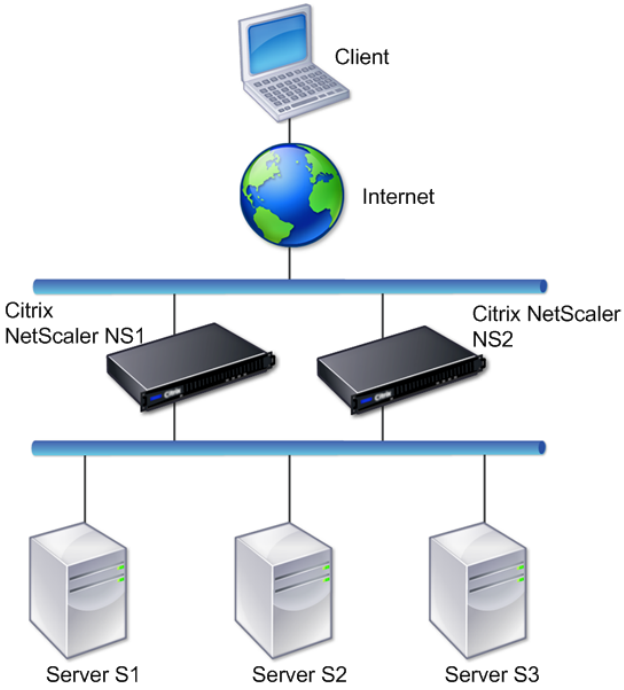


図 2：ツーアームモードでの高可用性



ワンアーム構成では、NS1 と NS2 の両方、およびサーバー S1、S2、S3 がスイッチに接続されています。

ツーアーム構成では、NS1 と NS2 の両方が 2 つのスイッチに接続されています。サーバー S1、S2、S3 は、2 番目のスイッチに接続されています。クライアントとサーバーの間のトラフィックは、NS1 または NS2 のいずれかを経由します。

高可用性環境をセットアップするには、1 台の ADC アプライアンスをプライマリとして、もう 1 台のアプライアンスをセカンダリとして設定します。各 ADC アプライアンスで次のタスクを実行します。

- ノードを追加します。
- 未使用のインターフェイスの高可用性モニターを無効にします。

ノードを追加

ノードは、ピア Citrix ADC アプライアンスの論理表現です。ID と NSIP でピア装置を識別します。アプライアンスはこれらのパラメーターを使用して、ピアと通信してその状態を追跡します。ノードを追加すると、プライマリ装置とセカンダリ装置は、非同期的にハートビートメッセージを交換します。ノード ID は 64 以下の整数です。

CLI 経由

コマンドラインインターフェイスを使用してノードを追加するには、次の手順に従います。

コマンドプロンプトで次のコマンドを入力し、ノードを追加して構成を確認します。

- add HA node <id> <IP アドレス >
- show HA node <id>

  例

```
1  add HA node 0 10.102.29.170
2   Done
3 > show HA node 0
4  1)      Node ID:      0
5          IP:   10.102.29.200 (NS200)
6          Node State: UP
7          Master State: Primary
8          SSL Card Status: UP
9          Hello Interval: 200 msecs
10         Dead Interval: 3 secs
11         Node in this Master State for: 1:0:41:50 (days:hrs:min:
               sec)
12 <!--NeedCopy-->
```

GUI 経由

GUI を使用してノードを追加するには、次の手順に従います。

1. ［**System**］ > ［**High Availability**］に移動します。
2. ［**Nodes**］タブで ［**Add**］をクリックします。
3. ［**Create HA Node**］ページの［**Remote Node IP Address**］テキストボックスに、リモートノードの NSIP アドレス（たとえば、10.102.29.170）を入力します。
4. ［**Configure remote system to participate in High Availability setup**］チェックボックスがオンになっていることを確認します。［**Remote System Login Credentials**］の下のボックスに、リモートノードのログイン情報を入力します。
5. ［**Turn off HA monitor on interfaces/channels that are down**］チェックボックスをオンにして、ダウンしているインターフェイスでの HA モニターを無効にします。

追加したノードが［Nodes］タブの一覧に表示されていることを確認します。

未使用のインターフェイスの高可用性モニターを無効にします

高可用性モニターは、インターフェイスを監視する仮想エンティティです。接続されていない、またはトラフィックに使用されていないインターフェイスのモニターを、無効にする必要があります。ステータスが DOWN になっているインターフェイスでモニターが有効になっている場合、ノードの状態は NOT UP になります。高可用性構成では、

プライマリノードが NOT UP 状態になると、高可用性フェールオーバーが行われる可能性があります。以下のような場合、インターフェイスには DOWN のマークが付けられます。

- インターフェイスが接続されていない。
- インターフェイスが正常に動作していない。
- インターフェイスを接続するケーブルが正常に機能していない。

CLI 経由

コマンドラインインターフェイスを使用して未使用のインターフェイスの高可用性モニターを無効化するには、次の手順を実行します

コマンドプロンプトで次のコマンドを入力し、未使用のインターフェイスの高可用性モニターを無効にして構成を確認します。

- set interface <id> -haMonitor OFF
- show interface <id>

  例

```
> set interface 1/8 -haMonitor OFF
  Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4000 <ENABLED, DOWN, down, autoneg, 802.1q>
        MTU=1514, native vlan=1, MAC=00:d0:68:15:fd:3d, downtime
            238h55m44s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
                throughput 0

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
            Muted(0)
        Bandwidth thresholds are not set.
 <!--NeedCopy-->
```

未使用のインターフェイスで高可用性モニターが無効になっている場合、そのインターフェイスの showinterface コマンドの出力には「HAMON」が含まれません。

GUI 経由

GUI を使用して未使用のインターフェイスの高可用性モニターを無効にするには、次の手順に従います。

1. ［System］> ［Network］> ［Interfaces］に移動します。
2. モニターを無効にする必要があるインターフェイスを選択します。
3. ［開く］をクリックします。［Modify Interface］ダイアログボックスが開きます。
4. ［HA Monitoring］で［OFF］をクリックします。
5. ［OK］をクリックします。
6. インターフェイスを選択すると、ページ下部の詳細に「HA Monitoring: OFF」が表示されることを確認してください。

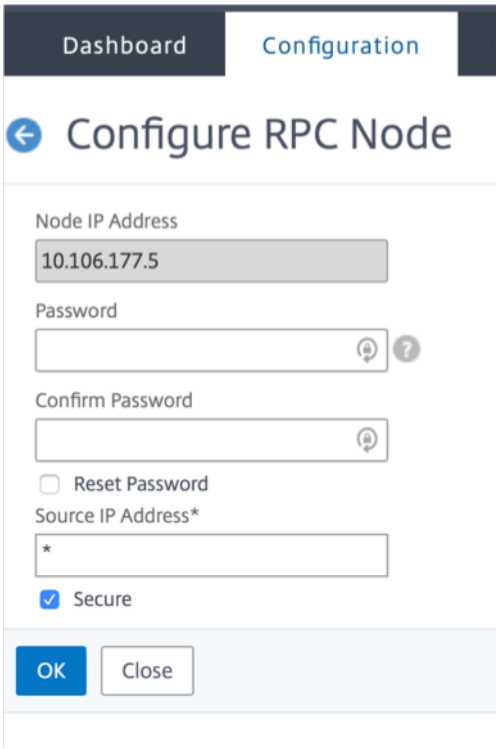
RPC ノードのパスワードを変更

April 21, 2022

各アプライアンスがほかの Citrix ADC アプライアンスと通信するには、それらのアプライアンスについての知識（Citrix ADC アプライアンスでの認証方法など）が必要です。RPC ノードは、構成およびセッション情報のシステム間通信に使用される内部システムエンティティです。1 つの RPC ノードが各 Citrix ADC アプライアンスに存在し、他の Citrix ADC アプライアンスの IP アドレスや認証に使用されるパスワードなどの情報を格納します。他の Citrix ADC アプライアンスに接続する Citrix ADC アプライアンスは、RPC ノード内のパスワードをチェックします。

GUI を使用して **RPC** ノードのパスワードを変更するには

1. ［**System**］> ［**Network**］> ［**RPC**］の順に選択します。
2. **RPC** ペインで、ノードを選択して ［**Edit**］をクリックします。
3. ［**Configure RPC Node**］に、新しいパスワードを入力します。
4. ［**Source IP Address**］に、ピアシステムノードとの通信に使用する既存のノードの IP アドレスを入力します。



5. [**Secure**] を選択し、[**OK**] をクリックします。

> 注
>
> セキュリティを強化するために、RPC ノードで [**Secure**] オプションを有効にすることをお勧めします。[**Secure**] オプションを有効にすると、アプライアンスは 1 つの ADC ノードから他の ADC ノードに送信されるすべての RPC 通信を暗号化して、RPC 通信を保護します。このセキュアな通信では、ポート番号 3008 を使用します。ADC ノード間のファイアウォールがポート番号 3008 をブロックしている場合は、ブロックを解除して続行します。そうしないと、構成の同期と構成の伝播が失敗する可能性があります。

CLI を使用して RPC ノードのパスワードを変更するには

コマンドラインで、次のコマンドを入力します:

```
set ns rpcNode <IPAddress> {
 -password }
  [-secure ( YES | NO )]
show ns rpcNode
<!--NeedCopy-->
```

例:


```
> set ns rpcNode 192.0.2.4 -password mypassword -secure YES
 Done
> show rpcNode
.
.
.
 IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
     SrcIP:  *           Secure:  ON
Done
>

<!--NeedCopy-->
```

FIPS アプライアンスの初回構成

April 25, 2022

> 注
>
> - FIPS FAQ は次の場所にあります：FIPS FAQ。

構成ユーティリティへの HTTPS アクセスおよびセキュアなリモートプロシージャコールには、証明書とキーのペアが必要です。RPC ノードは、構成およびセッション情報のシステム間通信に使用される内部システムエンティティです。アプライアンスごとに 1 つの RPC ノードが存在します。このノードに格納されるパスワードは、接続するアプライアンスによって提供されるパスワードと比較して調べられます。各アプライアンスがほかの Citrix ADC アプライアンスと通信するには、それらのアプライアンスについての知識（他のアプライアンスでの認証方法など）が必要です。RPC ノードはこの情報を保持しており、それにはほかの Citrix ADC アプライアンスの IP アドレスや、認証に使用されるパスワードなどが含まれます。

Citrix ADC MPX アプライアンス仮想アプライアンスでは、証明書とキーのペアは内部サービスに自動的にバインドされます。FIPS アプライアンスでは、FIPS カードのハードウェアセキュリティモジュール（HSM）に証明書とキーのペアをインポートする必要があります。そのためには、FIPS カードを構成し、証明書とキーのペアを作成して、それを内部サービスにバインドする必要があります。

CLI を使用してセキュアな HTTPS の構成

CLI を使用してセキュアな HTTPS を構成するには、次の手順を実行します

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化について詳しくは、「HSM の構成」を参照してください。

2. アプライアンスが高可用性セットアップの一部である場合は、SIM を有効にします。プライマリアプライアンスおよびセカンダリアプライアンス上での SIM の有効化について詳しくは、「高可用性セットアップでの FIPS アプライアンスの構成」を参照してください。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。コマンドプロンプトで入力します。

   ```
   import ssl fipskey serverkey -key ns-server.key -inform PEM
   ```

4. 証明書とキーのペアを追加します。コマンドプロンプトで入力します。

   ```
   add certkey server -cert ns-server.cert -fipskey serverkey
   ```

5. 前の手順で作成した証明書キーを次の内部サービスにバインドします。コマンドプロンプトで入力します。

   ```
   bind ssl service nshttps-127.0.0.1-443 -certkeyname server
   ```

   ```
   bind ssl service nshttps-::11-443 -certkeyname server
   ```

GUI を使用して安全な HTTPS を構成する

GUI を使用して安全な HTTPS を構成するには、次の手順に従います。

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化について詳しくは、「HSM の構成」を参照してください。
2. アプライアンスが高可用性セットアップの一部である場合は、セキュア情報システム（SIM）を有効にします。プライマリアプライアンスおよびセカンダリアプライアンス上での SIM の有効化について詳しくは、「高可用性セットアップでの FIPS アプライアンスの構成」を参照してください。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。FIPS キーのインポートについて詳しくは、「既存の FIPS キーのインポート」を参照してください。
4. [**Traffic Management**] **>** [**SSL**] **>** [**Certificates**] に移動します。
5. 詳細ペインで、[Install] をクリックします。
6. [Install Certificate] ダイアログボックスで、証明書の詳細を入力します。
7. [Create] をクリックしてから、[Close] をクリックします。
8. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
9. 詳細ペインの [Action] タブで、[Internal Services] をクリックします。
10. 一覧から`nshttps-127.0.0.1-443`を選択し、[Open] をクリックします。
11. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。
12. 一覧から`nshttps-::11-443`を選択し、[Open] をクリックします。
13. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。

14. [OK] をクリックします。

CLI を使用してセキュア RPC を構成する

CLI を使用してセキュア RPC を設定するには、次の手順に従います。

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化について詳しくは、「HSM の構成」を参照してください。
2. 安全な情報システム（SIM）を有効にします。プライマリアプライアンスおよびセカンダリアプライアンス上での SIM の有効化について詳しくは、「高可用性セットアップでの FIPS アプライアンスの構成」を参照してください。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。コマンドプロンプトで入力します。

   ```
   import ssl fipskey serverkey -key ns-server.key -inform PEM
   ```

4. 証明書とキーのペアを追加します。コマンドプロンプトで入力します。

   ```
   add certkey server -cert ns-server.cert -fipskey serverkey
   ```

5. 証明書とキーのペアを次の内部サービスにバインドします。コマンドプロンプトで入力します。

   ```
   bind ssl service nsrpcs-127.0.0.1-3008 -certkeyname server

   bind ssl service nskrpcs-127.0.0.1-3009 -certkeyname server

   bind ssl service nsrpcs-::1l-3008 -certkeyname server
   ```

6. セキュア RPC モードを有効にします。コマンドプロンプトで入力します。

   ```
   set ns rpcnode \<IP address\> -secure YES
   ```

   RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更」を参照してください。

GUI を使用してセキュア RPC を構成する

GUI を使用してセキュア RPC を設定するには、次の手順に従います。

1. アプライアンスの FIPS カードでハードウェアセキュリティモジュール（HSM）を初期化します。HSM の初期化について詳しくは、「HSM の構成」を参照してください。
2. 安全な情報システム（SIM）を有効にします。プライマリアプライアンスおよびセカンダリアプライアンス上での SIM の有効化について詳しくは、「高可用性セットアップでの FIPS アプライアンスの構成」を参照してください。
3. FIPS キーをアプライアンスの FIPS カードの HSM にインポートします。FIPS キーのインポートについて詳しくは、「既存の FIPS キーのインポート」を参照してください。
4. [**Traffic Management**] **>** [**SSL**] **>** [**Certificates**] に移動します。
5. 詳細ペインで、[Install] をクリックします。
6. [Install Certificate] ダイアログボックスで、証明書の詳細を入力します。

7. [Create] をクリックしてから、[Close] をクリックします。
8. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
9. 詳細ペインの [Action] タブで、[Internal Services] をクリックします。
10. 一覧から`nsrpcs-127.0.0.1-3008`を選択し、[Open] をクリックします。
11. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。
12. 一覧から`nskrpcs-127.0.0.1-3009`を選択し、[Open] をクリックします。
13. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。
14. 一覧から`nsrpcs-::11-3008`を選択し、[Open] をクリックします。
15. [Available] ペインの [SSL Settings] タブで、手順 7 で作成した証明書を選択して [Add] をクリックし、[OK] をクリックします。
16. [OK] をクリックします。
17. [**System**] > [**Network**] > [**RPC**] の順に選択します。
18. 詳細ペインで IP アドレスを選択して、[Open] をクリックします。
19. [Configure RPC Node] ダイアログボックスで、[Secure] を選択します。
20. [OK] をクリックします。

一般的なネットワークトポロジ

April 25, 2022

「Citrix ADC アプライアンスはネットワークのどこに適合しますか?」の「物理的な導入モード」の説明に従って、クライアントとサーバー間のインラインまたはワンアームのいずれかのモードで Citrix ADC アプライアンスを導入できます。インラインモードでは、一般的な導入タイプであるツーアームトポロジを使用します。

一般的なツーアームトポロジを設定します

ツーアームトポロジでは、1 つのネットワークインターフェイスはクライアントネットワークに接続され、もう 1 つのネットワークインターフェイスはサーバーネットワークに接続されます。これにより、すべてのトラフィックが仮想アプライアンスを通過するようになります。このトポロジでは、ハードウェアの再接続が必要になり、一時的にダウン時間が発生する場合もあります。ツーアームトポロジの基本的なバリエーションは複数サブネットと透過モードです。複数サブネットでは、通常、仮想アプライアンスがパブリックサブネット上に、サーバーがプライベートサブネット上に配置されます。透過モードでは、仮想アプライアンスとサーバーの両方がパブリックネットワーク上に配置されます。

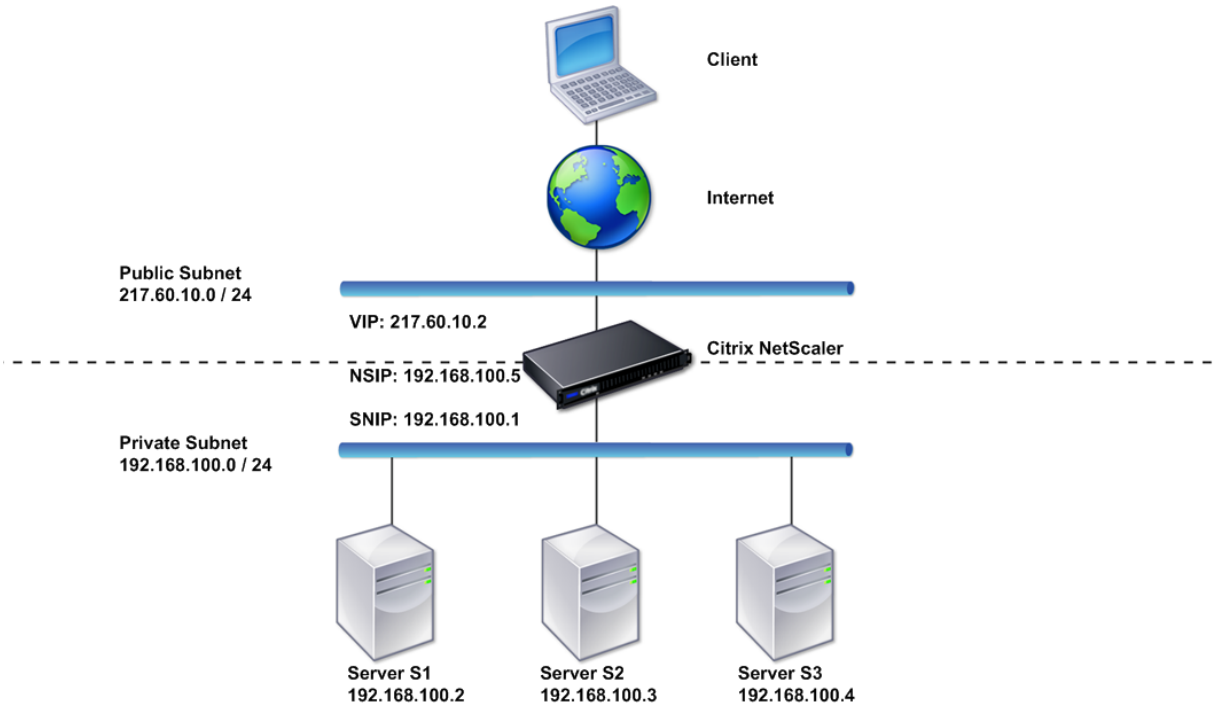
シンプルなツーアーム複数サブネットトポロジのセットアップ

最もよく使用されるトポロジの 1 つで、Citrix ADC アプライアンスはクライアントのサーバーの間にインラインに配置され、仮想サーバーはクライアント要求を処理するように設定されます。この構成は、クライアントとサーバーが異なるサブネット上にある場合に使用されます。たいていの場合、クライアントとサーバーは、それぞれパブリックサブネット上とプライベートサブネット上にあります。

たとえば、サーバー S1、S2、および S3 を管理するためにツーアームモードで展開されたアプライアンスについて考えてみましょう。アプライアンス上で HTTP の仮想サーバーが構成されており、各サーバー上で HTTP サービスが実行されています。これらのサーバーはプライベートサブネット上にあり、アプライアンスでこれらのサーバーと通信するための SNIP が構成されています。MIP の代わりに SNIP を使用するため、アプライアンスで USNIP（Use SNIP）オプションを有効にする必要があります。

次の図に示すように、VIP はパブリックサブネット 217.60.10.0 上にあり、NSIP、サーバー、および SNIP はプライベートサブネット 192.168.100.0/24 上にあります。

図 1: ツーアームモード、複数のサブネットのトポロジ図



複数のサブネットを持つツーアームモードで Citrix ADC アプライアンスを展開するには、次の手順に従います。

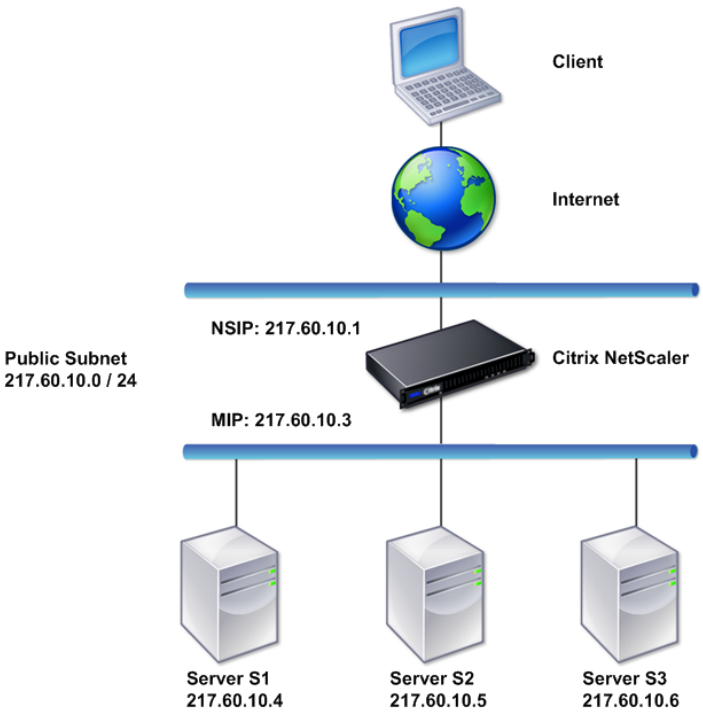
1. 「NetScaler IP アドレス（NSIP）の構成」の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 「サブネット IP アドレスの構成」の説明に従って、SNIP を構成します。
3. 「USNIP モードを有効または無効にするには」の説明に従って、USNIP オプションを有効にします。

4. 「仮想サーバーの作成」および「サービスの構成」の説明に従って、仮想サーバーとサービスを構成します。
5. 一方のネットワークインターフェイスをプライベートサブネットに、もう一方のインターフェイスをパブリックサブネットに接続します。

シンプルなツーアーム透過トポロジのセットアップ

クライアントが仮想サーバーの仲介なしでサーバーに直接アクセスする必要がある場合は、透過モードを使用します。クライアントはサーバーにアクセスできる必要があるので、サーバー IP アドレスはパブリックにする必要があります。次の図に示す例では、Citrix ADC アプライアンスがクライアントとサーバーの間に配置されています。そのため、トラフィックはアプライアンスを経由する必要があります。パケットをブリッジするため、L2 モードを有効にする必要があります。NSIP と MIP は、同じパブリックサブネット上（217.60.10.0/24）にあります。

図 2：ツーアーム、透過モードのトポロジ図



Citrix ADC アプライアンスをツーアームの透過モードで展開するには、次の手順に従います。

1. 「NetScaler IP アドレス（NSIP）の構成」の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 「レイヤー 2 モードの有効化と無効化」の説明に従って、L2 モードを有効にします。
3. 管理対象サーバーのデフォルトゲートウェイを、MIP として構成します。
4. ネットワークインターフェイスを、スイッチの適切なポートに接続します。

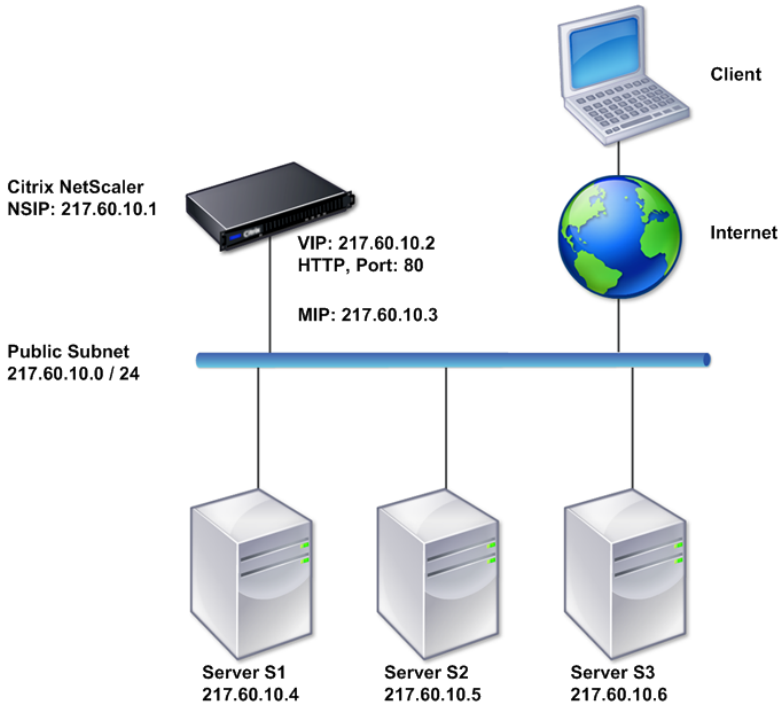
一般的なワンアームトポロジのセットアップ

ワンアームトポロジの 2 つの基本的なバリエーションは、1 つのサブネットを持つトポロジと複数のサブネットを持つトポロジです。

シンプルなワンアームシングルサブネットトポロジのセットアップ

クライアントとサーバーが同じサブネット上にある場合は、単一のサブネットでワンアームトポロジを使用できます。たとえば、サーバー S1、S2、および S3 を管理するためにワンアームモードで展開された Citrix ADC アプライアンスについて考えてみます。ADC アプライアンスで HTTP の仮想サーバーが構成されており、そのサーバー上で HTTP サービスが実行されています。次の図に示すように、Citrix ADC IP アドレス（NSIP）、マップされた IP アドレス（MIP）、およびサーバーの IP アドレスは同じパブリックサブネット上（217.60.10.0/24）にあります。

図 3：ワンアームモード、シングルサブネットのトポロジ図



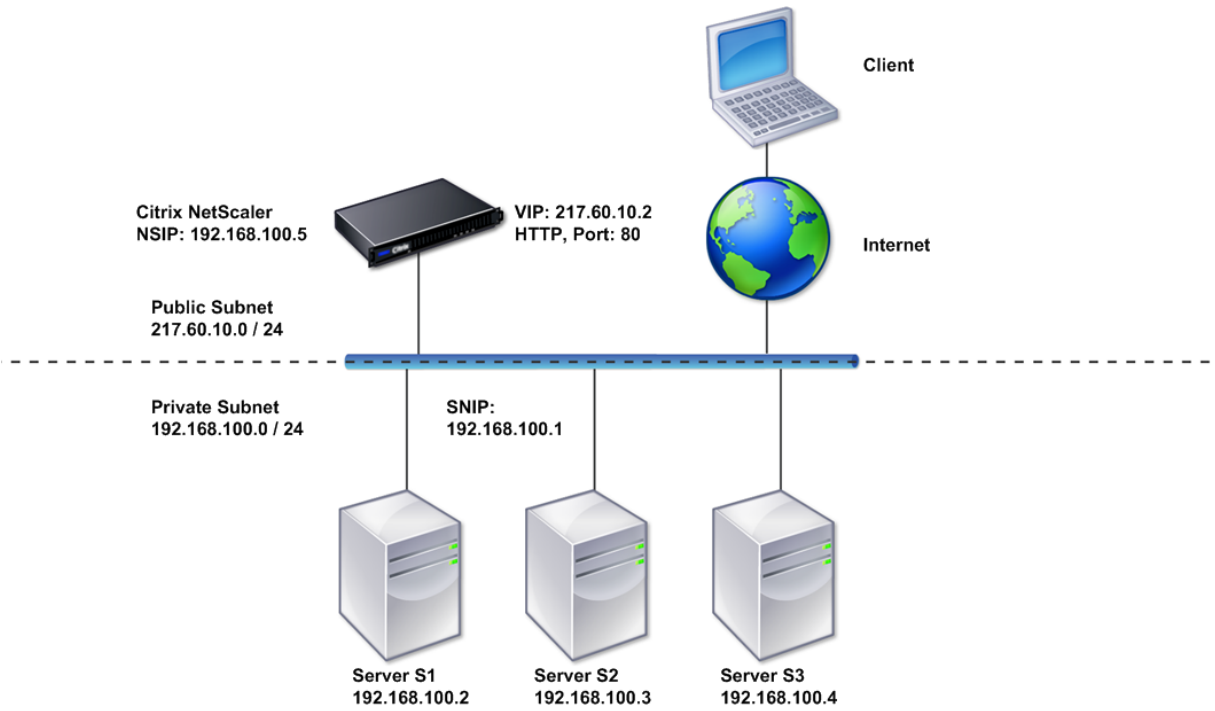
Citrix ADC アプライアンスを単一サブネットのワンアームモードで展開するには、次の手順に従います。

1. 「Citrix ADC IP アドレス（NSIP）の構成」の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 「仮想サーバーの作成」および「サービスの構成」の説明に従って、仮想サーバーとサービスを構成します。
3. 一方のネットワークインターフェイスをスイッチに接続します。

シンプルなワンアーム複数サブネットトポロジのセットアップ

クライアントとサーバーが異なるサブネット上にある場合は、複数のサブネットを持つワンアームトポロジを使用できます。たとえば、サーバー S1、S2、および S3 を管理するためにワンアームモードで導入された Citrix ADC アプライアンスについて考えてみましょう。これらのサーバーはネットワーク上のスイッチ SW1 に接続されています。アプライアンスで HTTP の仮想サーバーが構成されており、そのサーバー上で HTTP サービスが実行されています。これら 3 つのサーバーはプライベートサブネット上にあるので、SNIP（Subnet IP：サブネット IP）アドレスはこれらのサーバーと通信するように設定されています。アプライアンスが MIP の代わりに SNIP を使用するように、[Use Subnet IP address (USNIP)] を有効にする必要があります。次の図に示すように、仮想 IP アドレス (VIP) はパブリックサブネット上（217.60.10.0/24）にあります。NSIP、SNIP、およびサーバーの IP アドレスはプライベートサブネット上（192.168.100.0/24）にあります。

図 4：ワンアームモード、複数のサブネットのトポロジ図



複数のサブネットを持つワンアームモードで Citrix ADC アプライアンスを展開するには、次の手順に従います。

1. 「NetScaler IP アドレス（NSIP）の構成」の説明に従って、NSIP とデフォルトゲートウェイを構成します。
2. 「サブネット IP アドレスの構成」の説明に従って、SNIP を構成し、USNIP オプションを有効にします。
3. 「仮想サーバーの作成」および「サービスの構成」の説明に従って、仮想サーバーとサービスを構成します。
4. 一方のネットワークインターフェイスをスイッチに接続します。

システム管理設定

October 7, 2021

初期構成が整ったら、Citrix ADC アプライアンスの動作を定義し、接続管理を容易にする設定を構成できます。HTTP 要求と応答を処理するためのいくつかのオプションがあります。ルーティング、ブリッジング、および MAC ベースの転送モードは、Citrix ADC アプライアンスにアドレス指定されていないパケットを処理するために使用できます。ネットワークインターフェイスの特性を定義し、インターフェイスを集約できます。タイミングの問題を防ぐために、Citrix クロックをネットワークタイムプロトコル（NTP）サーバーと同期させることができます。Citrix ADC アプライアンスは、権限のあるドメインネームサーバー（ADNS）としてなど、さまざまな DNS モードで動作できます。システム管理用に SNMP を設定し、システムイベントの syslog ロギングをカスタマイズできます。展開する前に、構成が完全で正しいことを確認してください。

システム設定

April 21, 2022

システム設定の構成には、接続のキープアライブとサーバーオフロードを有効にするための HTTP ポートの設定、各サーバーの最大接続数の設定、接続あたりの最大要求件数の設定などの基本的なタスクが含まれます。プロキシ IP アドレスが適さない環境では、クライアント IP アドレス挿入を有効にして、HTTP Cookie バージョンを変更できます。

データ接続用のエフェメラルポートの代わりに、特定のポート範囲で FTP 接続が開かれるように Citrix ADC アプライアンスを構成することもできます。ファイアウォールですべてのポートを開くのは危険なので、この方法によってセキュリティが向上します。1,024～64,000 までの任意の範囲を設定できます。

展開前に、確認チェックリストを使用して構成内容を確認します。HTTP パラメーターと FTP ポート範囲を構成するには、Citrix ADC GUI を使用します。

次の表に記載された HTTP パラメーターの種類を変更できます。

パラメータータイプ: HTTP ポート情報

指定：管理対象サーバーが使用する Web サーバーの HTTP ポート。ポートを指定すると、アプライアンスは、指定されたポートと一致する宛先ポートを持つクライアント要求に対して要求のスイッチ操作を実行します。

> 注：着信したクライアント要求が、アプライアンスで指定されたサービスまたは仮想サーバー宛でない場合、要求の宛先ポートは、グローバルに構成されたいずれかの HTTP ポートと一致する必要があります。これにより、アプライアンスは接続のキープアライブとサーバーオフロードを実行できます。

パラメータータイプ: 制限

指定：各管理対象サーバーへの最大接続数、および、各接続を介して送信される要求の最大件数。たとえば、［Max Connections］が「500」に設定され、アプライアンスが 3 台のサーバーを管理している場合、3 台のサーバーそれぞれに対し、最大 500 個の接続を開くことができます。デフォルトでは、アプライアンスは管理する任意のサーバー

に対して任意の数の接続を作成できます。接続あたりの要求数を無制限に指定するには、［Max Requests］を「0」に設定します。

> 注：Apache HTTP サーバーを使用している場合は、［Max Connections］を、Apache httpd.conf ファイルの MaxClients パラメーターと同じ値に設定する必要があります。その他の Web サーバーの場合、このパラメーターの設定はオプションとなります。

パラメータータイプ: クライアント IP の挿入

指定：HTTP 要求ヘッダーへのクライアント IP アドレスの挿入を有効または無効にします。隣接するテキストボックスで、ヘッダーフィールドの名前を指定できます。アプライアンスが管理する Web サーバーが SNIP アドレスを受信すると、サーバーはそのアドレスをクライアント IP アドレスとして識別します。一部のアプリケーションでは、ログを記録するため、または Web サーバーが提供するコンテンツを動的に決定するために、クライアント IP アドレスが必要です。

クライアントから、アプライアンスが管理している 1 台、数台、またはすべてのサーバーに送信された HTTP ヘッダー要求に、実際のクライアント IP アドレスを挿入する機能を有効にできます。これによって、（Apache モジュール、ISAPI インターフェイス、または NSAPI インターフェイスを使用して）サーバーを少し変更するだけで、挿入されたアドレスにアクセスできるようになります。

パラメータータイプ: クッキーバージョン

指定：COOKIEINSERT パーシステンスが仮想サーバーに設定されている場合に使用される HTTP Cookie バージョン。デフォルトでは、インターネットで最も一般的な種類であるバージョン 0 を使用します。代わりにバージョン 1 を指定することも可能です。

パラメータータイプ: 要求/応答

指定：特定の種類の要求を処理し、HTTP エラー応答のログを有効または無効にするオプション。

パラメータータイプ: サーバーヘッダーの挿入

指定: Citrix ADC で生成された HTTP 応答にサーバーヘッダーを挿入します。

GUI を使用して HTTP パラメーターを設定するには、次の手順に従います。

1. ナビゲーションペインで、［**System**］を展開し、［**Settings**］をクリックします。
2. 詳細ペインで、［**Change HTTP Parameters**］をクリックします。
3. ［**Configure HTTP parameters**］ダイアログボックスで、上記の表に一覧表示された見出しの下に表示されている一部またはすべてのパラメーターの値を指定します。
4. ［**OK**］をクリックします。

GUI を使用して FTP ポート範囲を設定するには、次の手順に従います。

1. ナビゲーションペインで、［**System**］を展開し、［**Settings**］をクリックします
2. 詳細ペインの ［**Settings**］で、［**Change global system settings**］をクリックします。
3. ［**FTP Port Range**］の ［**Start Port**］および ［**End Port**］ボックスに、指定する範囲の最小ポート番号と最大ポート番号（たとえば、「5000」と「6000」）をそれぞれ入力します。
4. ［**OK**］をクリックします。

パケット転送モード

April 21, 2022

Citrix ADC アプライアンスは、アプライアンスが所有する IP アドレス宛ではないパケット（つまり NSIP、MIP、SNIP、構成済みサービス、または構成済み仮想サーバーの IP アドレス宛でないパケット）をルーティングまたはブリッジできます。デフォルトでは、L3 モード（ルーティング）が有効になり、L2 モード（ブリッジ）が無効になりますが、この構成は変更できます。アプライアンスがパケットを評価し、パケットの処理、ルーティング、ブリッジ、廃棄のいずれかを行う方法を次のフローチャートに示します。

図 1：レイヤー 2 モードとレイヤー 3 モード間の相互作用

アプライアンスは次のモードを使用して、受信したパケットを転送できます。

- レイヤー 2（L2）モード
- レイヤー 3（L3）モード
- MAC ベース転送モード

レイヤー **2** モードの有効化と無効化

レイヤー 2 モードは、レイヤー 2 フォワード（ブリッジ）機能を制御します。このモードを使用して、Citrix ADC アプライアンスをレイヤー 2 デバイスとして動作させ、自分宛ではないパケットをブリッジするように設定することができます。このモードを有効にした場合、パケットはどの MAC アドレスにも転送されません。これは、パケットがアプライアンスの任意のインターフェイスに着信することができ、各インターフェイスが独自の MAC アドレスを持っているからです。

レイヤー 2 モードを無効にした場合（デフォルト）、アプライアンスは、自分の MAC アドレス宛ではないパケットをドロップします。別のレイヤー 2 デバイスがアプライアンスと並列に設置されている場合は、レイヤー 2 モードを無効にしてブリッジ（レイヤー 2）ループを防ぐ必要があります。構成ユーティリティまたはコマンドラインを使用して、レイヤー 2 モードを有効にできます。

注：アプライアンスはスパニングツリープロトコルをサポートしていません。L2 モードが有効な場合に、ループを避けるために、アプライアンス上の 2 つのインターフェイスを同じブロードキャストドメインに接続しないでください。

CLI を使用してレイヤー **2** モードを有効または無効にするには

コマンドプロンプトで次のコマンドを入力し、レイヤー 2 モードを有効または無効にして、有効または無効になっていることを確認します。

- enable ns mode \< モード\>
- disable ns mode \< モード\>
- show ns mode

  例


```
    > enable ns mode l2
    Done
    > show ns mode

    Mode Acronym Status
    ------- ------- ------
    1) Fast Ramp FR ON
    2) Layer 2 mode L2 ON
    .
    .
    .
    Done
    >

    > disable ns mode l2
    Done
    > show ns mode

    Mode Acronym Status
    ------- ------- ------
    1) Fast Ramp FR ON
    2) Layer 2 mode L2 OFF
    .
    .
    .
    Done
    >
<!--NeedCopy-->
```

GUI を使用してレイヤー 2 モードを有効または無効にするには

1. ナビゲーションペインで、[**System**] を展開し、[**Settings**] をクリックします。
2. 詳細ペインの [**Modes** and **Features**] で [**Configure modes**] をクリックします。
3. [**Configure Modes**] ダイアログボックスで、[**Layer 2 Mode**] チェックボックスをオンにしてレイヤー 2 モードを有効にします。レイヤー 2 モードを無効にするには、チェックボックスをオフにします。
4. [**OK**] をクリックします。詳細ペインに「Enable/Disable Mode(s)?」メッセージが表示されます。
5. [**Yes**] をクリックします。

レイヤー **3** モードの有効化と無効化

レイヤー 3 モードは、レイヤー 3 フォワード機能を制御します。このモードを使用して、Citrix ADC アプライアンスがルーティングテーブルを参照して自分宛ではないパケットを転送するように設定できます。レイヤー 3 モードを有

効にした場合（デフォルト）、アプライアンスはルートテーブルのルックアップを実行して、アプライアンス所有の IP アドレス宛ではないすべてのパケットを転送します。レイヤー 3 モードを無効にした場合、アプライアンスはこれらのパケットをドロップします。

CLI を使用してレイヤ **3** モードを有効または無効にするには

コマンドプロンプトで次のコマンドを入力し、レイヤー 3 モードを有効または無効にして、有効または無効になっていることを確認します。

- enable ns mode \< モード\>
- disable ns mode \< モード\>
- show ns mode

例

```
> enable ns mode l3
Done
> show ns mode

Mode Acronym Status
------- ------- ------
1) Fast Ramp FR ON
2) Layer 2 mode L2 OFF
.
.
.
9) Layer 3 mode (ip forwarding) L3 ON
.
.
.
Done
>

> disable ns mode l3
Done
> show ns mode

Mode Acronym Status
------- ------- ------
1) Fast Ramp FR ON
2) Layer 2 mode L2 OFF
.
.
```



```
29      .
30      9) Layer 3 mode (ip forwarding) L3 OFF
31      .
32      .
33      .
34      Done
35      >
36  <!--NeedCopy-->
```

GUI を使用してレイヤー **3** モードを有効または無効にするには

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ペインの [Modes and Features] で [Configure modes] をクリックします。
3. [Configure Modes] ダイアログボックスで、[Layer 3 Mode (IP Forwarding)] チェックボックスをオンにしてレイヤー 3 モードを有効にします。To disable Layer 3 mode, clear the check box.
4. [OK] をクリックします。詳細ペインに「Enable/Disable Mode(s)?」メッセージが表示されます。
5. [Yes] をクリックします。

MAC ベースの転送モードを有効または無効にします

Citrix ADC アプライアンスがソースの MAC アドレスを記憶しているので、MAC ベース転送を使用して、パケットの転送時にトラフィックをより効率的に処理し、複数のルートや ARP ルックアップを防ぐことができます。複数のルックアップを防ぐため、アプライアンスは、ARP ルックアップを実行するすべての接続のソース MAC アドレスをキャッシュして、データを同じ MAC アドレスに返します。

MAC ベース転送は、VPN デバイスを使用している場合に便利です。これは、アプライアンスによって、特定の VPN を経由するすべてのトラフィックが同じ VPN デバイスを通過するようになるからです。

次の図は、MAC ベース転送のプロセスを示しています。

図 2: MAC ベースの転送プロセス

MAC ベース転送を有効にした場合、アプライアンスは次の MAC アドレスをキャッシュします。

- 受信接続のソース（ルーター、ファイアウォール、VPN デバイスなどの通信デバイス）
- 要求に応答するサーバー

サーバーがアプライアンスを介して応答する場合、アプライアンスは、応答パケットの宛先 MAC アドレスをキャッシュしたアドレスに設定し、トラフィックが対称的に流れるようにして、応答をクライアントに転送します。このプロセスでは、ルートテーブルのルックアップ機能と ARP ルックアップ機能が回避されます。ただし、アプライアンスが接続を開始した場合は、ルックアップ機能でルートと ARP テーブルが使用されます。MAC ベース転送を有効にするには、構成ユーティリティを使用するか、またはコマンドラインを使用します。

一部の展開環境では、着信および発信パスが、異なるルーターを経由する必要があります。このような状況では、MAC ベース転送がトポロジデザインに違反します。着信および発信パスが異なるルーターを経由する必要のあるグ

ローバルサーバー負荷分散（Global Server Load Balancing：GSLB）サイトの場合は、MAC ベース転送を無効にし、アプライアンスのデフォルトルーターを発信ルーターとして使用する必要があります。

MAC ベース転送を無効にして、レイヤー 2 またはレイヤー 3 接続を有効にした場合、ルートテーブルは、発信接続と着信接続に別のルーターを指定できます。MAC ベース転送を無効にするには、構成ユーティリティを使用するか、またはコマンドラインを使用します。

CLI を使用して **MAC** ベースの転送を有効または無効にするには

コマンドプロンプトで次のコマンドを入力し、MAC ベース転送を有効または無効にして、有効または無効になっていることを確認します。

- <enable ns mode \< モード\>
- <disable ns mode \< モード\>
- <show ns mode
  Example

  “‘ pre codeblock

  > enable ns mode mbf
  > Done
  > show ns mode

```
1   Mode                          Acronym           Status
2   -------                       -------           ------   1)  Fast
       Ramp                    FR                  ON   2)  Layer 2
       mode                 L2                  OFF   .   .   .   6)
       MAC-based forwarding        MBF                 ON   .   .   .
         Done   >
```

> disable ns mode mbf
> Done
> show ns mode

```
1   Mode                          Acronym           Status
2   -------                       -------           ------   1)  Fast
       Ramp                    FR                  ON   2)  Layer 2
       mode                 L2                  OFF   .   .   .   6)
       MAC-based forwarding        MBF                 OFF   .   .   .
          Done   >   <!--NeedCopy--> ```
```


GUI を使用して MAC ベースの転送を有効または無効にするには

1. ナビゲーションペインで、[**System**] を展開し、[**Settings**] をクリックします。
2. 詳細ペインの [**Modes and Features**] グループで [**Configure modes**] をクリックします。
3. [**Configure Modes**] ダイアログボックスで MAC ベース転送モードを有効にするには、[**MAC Based Forwarding**] チェックボックスをオンにします。MAC ベース転送を無効にするには、このチェックボックスをオフにします。
4. [**OK**] をクリックします。詳細ペインに「Enable/Disable Mode(s)?」メッセージが表示されます。
5. [**Yes**] をクリックします。

ネットワークインターフェイス

April 25, 2022

Citrix ADC インターフェイスにはスロット/ポート表示に番号が付けられています。個々のインターフェイスの特性を変更することに加えて、特定のホストグループだけにトラフィックが許可されるように VLAN を構成できます。リンクを高速チャネルに集約することもできます。

仮想 **LAN**

Citrix ADC アプライアンスは、（レイヤー 2）ポートと IEEE802.1Q タグ付き仮想 LAN（VLAN）をサポートします。VLAN 構成は、トラフィックを特定のワークステーショングループだけに制限しなければならない場合に便利です。IEEE 802.1q タグ付け機能を使用して複数の VLAN に属するように、ネットワークインターフェイスを構成できます。

構成した VLAN は、IP サブネットにバインドできます。これにより、（サブネット上のホストのデフォルトルーターとして構成されている場合）ADC アプライアンスは、これらの VLAN 間で IP 転送を実行します。

Citrix ADC アプライアンスは、次のタイプの VLAN をサポートします。

- デフォルト VLAN

  デフォルトでは、Citrix ADC アプライアンスのネットワークインターフェイスは、タグなしのネットワークインターフェイスとして、単一のポートベース VLAN に含まれています。このデフォルト VLAN は、VID が 1 であり、永続的に存在します。デフォルト VLAN を削除したり、その VID を変更したりすることはできません。

- ポートベース VLAN

  排他的なレイヤー 2 ブロードキャストドメインを共有するネットワークインターフェイスのセットは、ポートベース VLAN のメンバーシップを定義します。複数のポートベース VLAN を構成できます。インターフェイスをタグなしメンバーとして新しい VLAN に追加すると、デフォルト VLAN から自動的に削除されます。

- タグ付き VLAN

ネットワークインターフェイスは、VLAN のタグ付きまたはタグなしメンバーになることができます。各ネットワークインターフェイスは、唯一の VLAN（ネイティブ VLAN）のタグなしメンバーです。タグなしネットワークインターフェイスは、タグなしフレームとしてネイティブ VLAN のフレームを転送します。タグ付きネットワークインターフェイスは、複数の VLAN の一部になることができます。タグ付きを設定する場合は、リンクの両端で VLAN 設定が一致していることを確認してください。構成ユーティリティを使用し、VLAN のタグ付きメンバーとしてポートをバインドできる、タグ付き VLAN（nsvlan）を定義できます。この VLAN を構成するには ADC アプライアンスを再起動する必要があるので、ネットワークの初回構成中に実行する必要があります。

リンクアグリゲートチャネル

リンクアグリゲーションは、複数ポートからの着信データを、1 つの高速リンクに結合します。リンクアグリゲートチャネルを構成すると、Citrix ADC アプライアンスとほかの接続デバイス間の通信チャネルの容量と可用性が増加します。アグリゲートされたリンクは「チャネル」とも呼ばれます。

ネットワークインターフェイスをチャネルにバインドした場合、チャネルのパラメーターは、ネットワークインターフェイスのパラメーターよりも優先されます。ネットワークインターフェイスは、1 つのチャネルにのみバインドできます。ネットワークインターフェイスをリンクアグリゲートチャネルにバインドすると、VLAN 構成が変更されます。つまり、ネットワークインターフェイスをチャネルにバインドすると、ネットワークインターフェイスは以前属していた VLAN から削除され、デフォルト VLAN に追加されます。ただし、チャネルを元の VLAN や新しい VLAN にバインドすることができます。たとえば、ネットワークインターフェイス 1/2 と 1/3 を、ID が 2 の VLAN にバインドしていて、それらをリンクアグリゲートチャネル LA/1 にバインドした場合、ネットワークインターフェイスはデフォルト VLAN に移動されますが、VLAN 2 にバインドすることができます。

注：また、リンクアグリゲーション制御プロトコル（Link Aggregation Control Protocol：LACP）を使用して、リンクアグリゲーションを構成することもできます。詳しくは、「Link Aggregation Control Protocol を使用したリンクアグリゲーションの構成」を参照してください。

クロック同期

April 21, 2022

Citrix ADC アプライアンスを設定して、ローカルの時刻を、NTP（Network Time Protocol：ネットワークタイムプロトコル）サーバーの時刻と同期することができます。これにより、NetScaler のクロックの設定は、ネットワーク上のほかのサーバーと同じ日付と時刻になります。NTP は、UDP（User Datagram Protocol：ユーザーデータグラムプロトコル）ポート 123 を、トランスポートレイヤーとして使用します。NTP 構成ファイルで NTP サーバーを追加し、アプライアンスがこれらのサーバーから定期的に更新を受け取るようにする必要があります。

ローカルの NTP サーバーがない場合は、公式 NTP サイト（http://www.ntp.org）で、パブリックなオープンアクセス NTP サーバーの一覧を検索できます。

アプライアンスでクロック同期を構成するには、次の手順に従います：

1. コマンドラインにログオンし、shell コマンドを入力します。
2. シェルプロンプトで、ntp.conf ファイルを/etc ディレクトリから/nsconfig ディレクトリにコピーします。ファイルが/nsconfig ディレクトリに既に存在する場合は、ntp.conf ファイルから次のエントリが削除されていることを確認します。

   ```
   restrict localhost

   restrict 127.0.0.2
   ```

   これらのエントリは、デバイスをタイムサーバーとして実行する場合のみ必要となります。ただし、この機能は Citrix ADC アプライアンスではサポートされていません。
3. /nsconfig/ntp.conf を編集して、ファイルのサーバーと制限エントリの下に、必要な NTP サーバーの IP アドレスを入力します。
4. /nsconfig ディレクトリに rc.netscaler という名前のファイルがない場合は作成します。
5. 次のエントリを追加して、/nsconfig/rc.netscaler を編集します：`/usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &`

   このエントリは、ntpd サービスを開始し、ntp.conf ファイルをチェックして、メッセージを /var/log ディレクトリ。

   注：Citrix ADC アプライアンスとタイムサーバーの時間差が 1000 秒を超える場合、ntpd サービスは ADC ログへのメッセージで終了します。これを避けるには、強制的に時刻を同期する-g オプションを使用して ntpd を開始する必要があります。/nsconfig/rc.netscaler に次のエントリを追加します。

   ```
   /usr/sbin/ntpd -g -c /nsconfig/ntp.conf -l /var/log/ntpd.log &
   ```

   時間差が大きく、強制的に時刻を同期したくない場合は、日付を手動で設定してから ntpd を再び開始できます。アプライアンスとタイムサーバー間の時間差は、シェルで次のコマンドを実行することによって確認できます。

   ```
   1 ntpdate -q <IP address or domain name of the NTP server>
   2 <!--NeedCopy-->
   ```

6. アプライアンスを再起動して、クロック同期を有効にします。

   注：アプライアンスを再起動する前に時刻同期を開始したい場合は、手順 5 で rc.netscaler ファイルに追加した次のコマンドをシェルプロンプトで入力します。

   ```
   1 /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ ntpd.log &
   2 <!--NeedCopy-->
   ```


DNS の構成

July 15, 2022

ADNS（Authoritative Domain Name Server）、DNS プロキシサーバー、エンドリゾルバー、またはフォワーダーとして機能するように、Citrix ADC アプライアンスを構成できます。SRV レコード、AAAA レコード、A レコード、MX レコード、NS レコード、CNAME レコード、PTR レコード、SOA レコードなど、DNS リソースレコードを追加できます。また、アプライアンスは外部 DNS サーバーの負荷を分散できます。

アプライアンスをフォワーダーとして構成する方法が一般的です。この構成では、外部ネームサーバーを追加する必要があります。外部ネームサーバーを追加したら、構成が正しいことを確認する必要があります。

外部ネームサーバーを追加、削除、有効化、および無効化することができます。IP アドレスを指定してネームサーバーを作成するか、既存の仮想サーバーをネームサーバーとして設定できます。

ネームサーバーを追加する場合は、IP アドレスまたは VIP（Virtual IP：仮想 IP）アドレスを指定できます。IP アドレスを使用する場合、アプライアンスはラウンドロビン方式で、構成したネームサーバーに要求を負荷分散します。VIP を使用する場合は、任意の負荷分散方式を指定できます。

CLI を使用してネームサーバーを追加します

コマンドプロンプトで次のコマンドを入力し、ネームサーバーを追加して構成を確認します。

- <add dns nameServer \<IP\>
- <show dns nameServer \<IP\>

  例

  ‘‘‘

  ```
  add dns nameServer 10.102.29.10
  Done
  show dns nameServer 10.102.29.10
  1) 10.102.29.10 - State: DOWN
  Done
  ```

  ‘‘‘

GUI を使用してネームサーバーを追加する

1. [**Traffic Management**] > [**DNS**] > [**Name Servers**] の順に選択します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. [**Create Name Server**] ダイアログボックスで、[**IP Address**] を選択します。
4. [**IP Address**] ボックスにネームサーバーの IP アドレス（たとえば、「10.102.29.10」）を入力します。外部ネームサーバーを追加する場合は、[**Local**] チェックボックスをオフにします。

5. [作成] をクリックし、[閉じる] をクリックします。
6. 追加したネームサーバーが ［**Name Servers**］ペインに表示されることを確認します。
   “‘

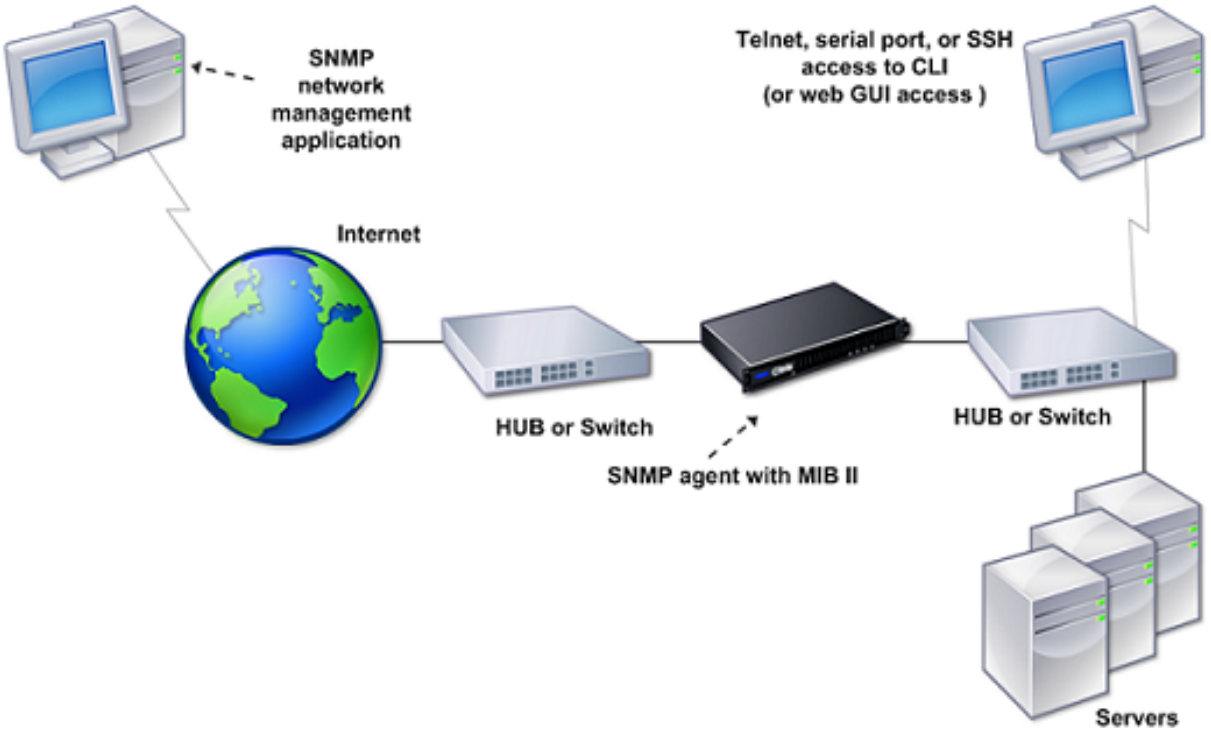
SNMP 構成

April 21, 2022

外部のコンピューターで実行されている SNMP（Simple Network Management Protocol：簡易ネットワーク管理プロトコル）ネットワーク管理アプリケーションは、Citrix ADC アプライアンスの SNMP エージェントにクエリを発行します。エージェントは、ネットワーク管理アプリケーションで要求されたデータを MIB（Management Information Base：管理情報ベース）で検索して、データをアプリケーションに送信します。

SNMP の監視では、トラップメッセージとアラームを使用します。SNMP トラップメッセージは、異常な状態を通知するためにエージェントが生成する非同期イベントです。このメッセージは、アラームによって通知されます。たとえば、CPU 使用率が 90% を超えたときに通知する場合は、その条件に対するアラームをセットアップできます。次の図は、SNMP が有効化および構成されている Citrix ADC アプライアンスを備えたネットワークを示しています。

図 1: Citrix ADC アプライアンスの SNMP



Citrix ADC の SNMP エージェントは、SNMP version 1 (SNMPv1)、SNMP version 2 (SNMPv2)、および SNMP version 3（SNMPv3）をサポートしています。エージェントはバイリンガルモードで動作しているので、SNMPv2 クエリ（Get-Bulk など）と SNMPv1 クエリを処理できます。また、SNMP エージェントは SNMPv2 に準拠したト

ラップを送信し、counter64 などの SNMPv2 データタイプをサポートしています。SNMPv1 マネージャー（ADC アプライアンスからの SNMP 情報を要求する、他のサーバー上のプログラム）は、SNMP クエリを処理するときに NS-MIB-smiv1.mib ファイルを使用します。SNMPv2 マネージャーは、NS-MIB-smiv2.mib ファイルを使用します。

Citrix ADC アプライアンスは、次のエンタープライズ固有の MIB をサポートします。

- 標準 MIB-2 グループのサブセット。MIB-2 グループの SYSTEM、IF、ICMP、UDP、および SNMP を提供します。
- システムエンタープライズ MIB。システム固有の設定と統計情報を提供します。

SNMP の設定には、SNMP エージェントにクエリを発行できるマネージャーを指定し、SNMP トラップメッセージを受信する SNMP トラップリスナーを追加し、SNMP アラームを設定する作業が含まれます。

SNMP マネージャーを追加する

SNMP Version 1、2、または 3 に準拠する管理アプリケーションを実行するワークステーションを構成して、アプライアンスにアクセスできます。このようなワークステーションは、「SNMP マネージャー」と呼ばれます。アプライアンスに SNMP マネージャーを指定しない場合、アプライアンスは、ネットワーク上のすべての IP アドレスから SNMP クエリを受け付けて応答します。1 つまたは複数の SNMP マネージャーを設定する場合、アプライアンスは、それらの特定の IP アドレスからのみ SNMP クエリを受け付けて応答します。SNMP マネージャーの IP アドレスを指定する場合は、ネットマスクパラメーターを使用して、サブネット全体からのアクセス権を付与できます。最大 100 個の SNMP マネージャーまたはネットワークを追加できます。CLI を使用して SNMP マネージャーを追加するには

コマンドプロンプトで次のコマンドを入力し、SNMP マネージャーを追加して構成を確認します。

```
add snmp manager <IPAddress> ... [-netmask <netmask>]
show snmp manager <IPAddress>
```

例:

```
1 add snmp manager 10.102.29.5 -netmask 255.255.255.255
2 Done
3 show snmp manager 10.102.29.5
4 10.102.29.5 255.255.255.255
5 Done
6 <!--NeedCopy-->
```

GUI を使用して **SNMP** マネージャーを追加するには:

1. ナビゲーションペインで ［**System**］、［**SNMP**］の順に展開して、［**Managers**］をクリックします。
2. 詳細ペインで、［**Add**］をクリックします。

3. [**Add SNMP Manager**] ダイアログボックスの [**IP Address**] ボックスに、管理アプリケーションを実行しているワークステーションの IP アドレス（たとえば、「10.102.29.5」）を入力します。
4. [**Create**] をクリックしてから、[**Close**] をクリックします。
5. 追加した SNMP トラップが、ペインの下部にある [**Details**] セクションに表示されていることを確認します。

SNMP トラップリスナーの追加

アラームを構成したら、アプライアンスによるトラップメッセージの送信先となるトラップリスナーを指定します。トラップリスナーの IP アドレスや宛先ポートなどのパラメーターを指定する以外に、トラップの種類（汎用または専用）と SNMP のバージョンを指定できます。

汎用または専用のトラップを受信するために、最大 20 のトラップリスナーを構成できます。

CLI を使用して SNMP トラップリスナーを追加するには

コマンドプロンプトで次のコマンドを入力し、SNMP トラップを追加して構成を確認します。

- `add snmp trap specific <IP>`
- `show snmp trap`

例:

```
Trap type: SPECIFIC
Destination IP: 10.102.29.3
TD: 0
Destination Port: 162
Source IP: NetScaler IP
Version: V2
Min-Severity: -
AllPartition: DISABLED
Community: public
<!--NeedCopy-->
```

GUI を使用して SNMP トラップリスナーを追加するには

1. ナビゲーションペインで [System]、[**SNMP**] の順に展開して、[**Traps**] をクリックします。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**Create SNMP Trap Destination**] ダイアログボックスの [**Destination IP Address**] ボックスに、IP アドレス（たとえば、「10.102.29.3」）を入力します。
4. [**Create**] をクリックしてから、[**Close**] をクリックします。
5. 追加した SNMP トラップが、ペインの下部にある [**Details**] セクションに表示されていることを確認します。

SNMP アラームを構成する

いずれかのアラームに該当するイベントが発生した場合にアプライアンスがトラップメッセージを生成するように構成できます。アラームを構成するには、アラームを有効にして、トラップを生成する重要度レベルを設定します。Critical、Major、Minor、Warning、および Informational という、5 つの重要度レベルがあります。アラームの重要度が、トラップに指定した重要度と一致する場合のみ、トラップが送信されます。

一部のアラームは、デフォルトで有効になっています。SNMP アラームを無効にすると、該当するイベントが発生してもアプライアンスはトラップメッセージを生成しません。たとえば、Login-Failure SNMP アラームを無効にすると、ログインが失敗してもアプライアンスはトラップメッセージを生成しません。

CLI を使用してアラームを有効または無効にするには

コマンドプロンプトで次のコマンドを入力し、アラームを有効または無効にして、有効または無効になっていることを確認します。

- set snmp alarm \< トラップ名\> \[-state ENABLED | DISABLED \]
- show snmp alarm \< トラップ名\>

  例

```
set snmp alarm LOGIN-FAILURE -state ENABLED
Done
show snmp alarm LOGIN-FAILURE
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
----- --------------- ---------------- ---- -------- --------- --------
LOGIN-FAILURE N/A N/A N/A ENABLED - ENABLED
Done
<!--NeedCopy-->
```

CLI を使用してアラームの重大度を設定するには

コマンドプロンプトで次のコマンドを入力し、アラームの重要度を設定して重要度が正しく設定されていることを確認します。

- set snmp alarm <trapName> [-severity <severity>]
- show snmp alarm <trapName>

例:

```
set snmp alarm LOGIN-FAILURE -severity Major
Done
```



```
show snmp alarm LOGIN-FAILURE
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
----- --------------- ---------------- ---- -------- --------- --------
LOGIN-FAILURE N/A N/A N/A ENABLED Major ENABLED
Done
<!--NeedCopy-->
```

GUI を使用してアラームを構成するには

1. ナビゲーションペインで ［**System**］、［SNMP］の順に展開して、［**Alarms**］をクリックします。
2. 詳細ペインでアラーム（たとえば、［LOGIN-FAILURE］）を選択し、［**Open**］をクリックします。
3. ［**Configure SNMP Alarm**］ダイアログボックスでアラームを有効にするには、［**State**］の一覧で［Enabled］を選択します。アラームを無効にするには、［Disabled］を選択します。
4. ［**Severity**］ドロップダウンリストで、重要度のオプション（たとえば、［Major］）を選択します。
5. ［**OK**］をクリックし、［**Close**］をクリックします。
6. ペインの下部にある［**Details**］セクションを表示し、SNMP アラームのパラメーターが正しく構成されていることを確認します。

構成を確認する

April 21, 2022

システムの設定が終了したら、次のチェックリストに記入して設定を確認します。

設定チェックリスト

- 実行中のビルド：
- 非互換性の問題はない（非互換性の問題は対象ビルドのリリースノートに記載されています）。
- ポート設定（速度、二重、フロー制御、監視）がスイッチのポートと同じである。
- ピーク時にすべてのサーバー側接続をサポートするように、SNIP アドレスが十分に設定されている。
  - 設定済みの SNIP IP アドレス数：__
  - 予想される同時サーバー接続数：
    [ ] 62,000 [ ] 124,000 [ ] Other____

トポロジ設定チェックリスト

ルートを使用して、他のサブネット上のサーバーを解決した。

入力したルート

- Citrix ADC アプライアンスが官民トポロジにある場合、リバース NAT が構成されています。
- ADC アプライアンスで設定されたフェールオーバー（高可用性）設定が、ワンアームまたはツーアーム構成で解決される。使用されないネットワークインターフェイスをすべて無効化：

- ADC アプライアンスが外部負荷分散装置の後ろに配置されている場合、外部負荷分散装置の負荷分散ポリシーが「least connection」ではない。

  外部負荷分散装置に設定されている負荷分散ポリシー：

- ADC アプライアンスがファイアウォールの前に配置されている場合、ファイアウォールのセッションタイムアウトが 300 秒以上に設定されている。

  注：Citrix ADC アプライアンスでの TCP アイドル接続のタイムアウトは 360 秒です。ファイアウォール上でも 300 秒以上のタイムアウトが設定されている場合、接続が先に閉じられないためにアプライアンスで TCP 接続の多重化が行われることがあります。

  セッションタイムアウトに設定されている値：_____

サーバー設定チェックリスト

- 「キープアライブ」がすべてのサーバーで有効になっている。

  キープアライブタイムアウトに設定されている値：_____

- デフォルトゲートウェイが正しい値に設定されている（デフォルトゲートウェイは、Citrix ADC アプライアンスまたはアップストリームルーターのいずれかである必要があります。）デフォルトゲートウェイは次のとおりです。

- サーバーのポート設定（速度、二重、フロー制御、監視）がスイッチのポート設定と同じである。

- Microsoft® Internet Information Server を使用している場合、バッファリングがサーバーで有効になっている。
- Apache Server を使用している場合、MaxConn（最大接続数）パラメーターがサーバーと Citrix ADC アプライアンスで設定されている。

  設定されている MaxConn（最大接続数）の値：

- Netscape Enterprise Server を使用する場合、接続パラメーターあたりの最大リクエスト数は Citrix ADC アプライアンスで設定されます。設定されている接続ごとの最大要求数の値:

ソフトウェア機能の設定チェックリスト

- レイヤー 2 モード機能を無効にする必要があるかどうか（別のレイヤー 2 デバイスが Citrix ADC アプライアンスと並行して動作している場合は無効にします。）

  有効または無効にする理由

- MAC ベース転送機能を無効にする必要があるかどうか（リターントラフィックが使用する MAC アドレスが異なる場合は、無効にする必要があります）。

  有効または無効にする理由

- ホストベースの再使用を無効にする必要があるかどうか（サーバーに仮想ホストがあるかどうか）。

  有効または無効にする理由

- サージ保護機能のデフォルト設定を変更する必要があるかどうか。

  設定を変更または保持する理由

アクセスチェックリスト

- クライアント側ネットワークから、システム IP の ping を実行できる。
- サーバー側ネットワークから、システム IP の ping を実行できる。
- 管理対象サーバーは、Citrix ADC を介して ping を実行できます。
- 管理対象サーバーから、インターネットホストの ping を実行できる。
- ブラウザーを介して、管理対象サーバーにアクセスできる。
- ブラウザーを使用して、管理対象サーバーからインターネットにアクセスできる。
- SSH を使用してシステムにアクセスできる。
- すべての管理対象サーバーへの管理者アクセスが機能している。

> 注: ping ユーティリティを使用している場合は、ping されるサーバーで ICMP ECHO を有効にしてください。そうしないと、ping が失敗します。

ファイアウォールチェックリスト

次のファイアウォール要件が満たされている。

- UDP 161（SNMP）
- UDP 162（SNMP トラップ）
- TCP/UDP 3010（GUI）
- HTTP 80（GUI）
- TCP 22（SSH）

Citrix ADC アプライアンスでトラフィックを負荷分散する

October 7, 2021

負荷分散機能は、クライアント要求を複数のサーバーに分散して、リソース使用率を最適化します。限られた数のサーバーが多数のクライアントにサービスを提供する実際のシナリオでは、サーバーが過負荷になり、サーバーファームのパフォーマンスが低下する可能性があります。Citrix ADC アプライアンスは、負荷分散基準を使用して、各クライアント要求を、要求が到着したときに処理するのに最適なサーバーに転送することにより、ボトルネックを防ぎます。

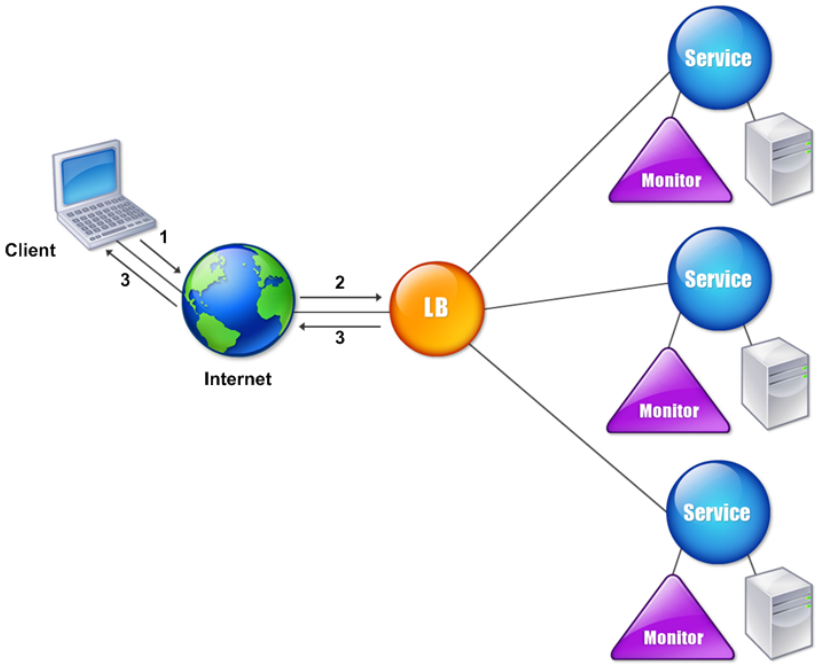
負荷分散を構成するには、サーバーファーム内の複数のサーバーをプロキシし、それらの間で負荷を分散する仮想サーバーを定義します。

クライアントがサーバーへの接続を開始すると、仮想サーバーはクライアント接続を終了し、選択したサーバーとの新しい接続を開始するか、サーバーとの既存の接続を再利用して負荷分散を実行します。負荷分散機能は、レイヤー 4（TCP および UDP）からレイヤー 7（FTP、HTTP、および HTTPS）までのトラフィック管理を提供します。

Citrix ADC アプライアンスは、負荷分散方法と呼ばれるいくつかのアルゴリズムを使用して、サーバー間で負荷を分散する方法を決定します。デフォルトの負荷分散方法は、最小接続方法です。

一般的な負荷分散の展開は、次の図で説明するエンティティで構成されます。

図 1：負荷分散アーキテクチャ



エンティティは次のように機能します。

- 仮想サーバー。IP アドレス、ポート、およびプロトコルで表されるエンティティ。仮想サーバーの IP アドレス（VIP）は通常、パブリック IP アドレスです。クライアントはこの IP アドレスに接続要求を送信します。仮想サーバーは、サーバーのバンクを表します。
- **Service**. サーバーまたはサーバー上で実行されているアプリケーションの論理表現。サーバーの IP アドレス、ポート、およびプロトコルを識別します。サービスは仮想サーバーにバインドされています。
- サーバーオブジェクト。IP アドレスで表されるエンティティ。サーバーオブジェクトは、サービスを作成するときに作成されます。サービスの IP アドレスは、サーバーオブジェクトの名前として使用されます。サーバーオブジェクトを作成してから、サーバーオブジェクトを使用してサービスを作成することもできます。
- モニター。サービスの状態を追跡するエンティティ。アプライアンスは、各サービスにバインドされたモニターを使用してサーバーを定期的にプローブします。サーバーが指定された応答タイムアウト内に応答せず、指定された数のプローブが失敗した場合、サービスは DOWN とマークされます。次に、アプライアンスは残りのサービス間で負荷分散を実行します。

負荷分散

April 21, 2022

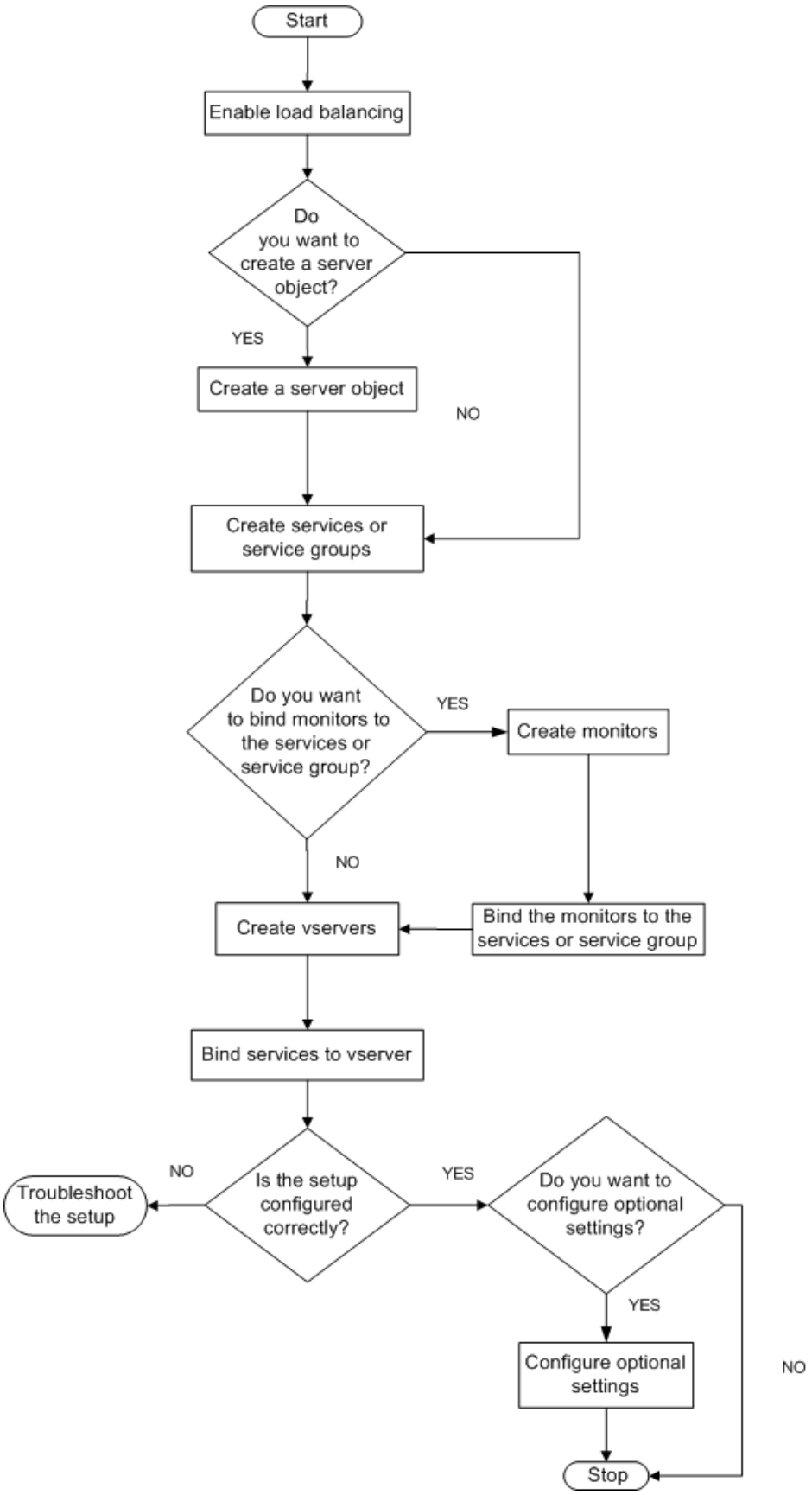
負荷分散を構成するには、まずサービスを作成する必要があります。次に、仮想サーバーを作成して、サービスをその仮想サーバーにバインドします。デフォルトでは、Citrix ADC アプライアンスはモニターを各サービスにバインドします。サービスをバインドしたら、すべての構成内容が正しいことを確認します。

> 注：構成を適用した後で、各エンティティがどのように実行されているかを示す統計情報を表示できます。統計ユーティリティ、または
> stat lb vserver コマンドを使用します。仮想サーバー名 >

オプションで、サービスに重要度（Weight）を割り当てることができます。割り当てられた重要度に基づいてサービスが負荷分散されます。ただし、負荷分散機能の導入時には詳細な重要度を構成せずに、基本的なパーシステンス設定（特定サーバーへの接続の保持）と構成保護設定のみを行えます。

次のフローチャートは、一連の構成タスクを示しています。

図 1：負荷分散を構成するための一連のタスク



負荷分散を有効にする

負荷分散を構成する前に、負荷分散機能が有効になっているか確認します。

CLI を使用して負荷分散を有効にするには

コマンドプロンプトで次のコマンドを入力し、負荷分散を有効にして構成を確認します。

- enable feature lb
- show feature

  例

  “‘ pre codeblock

  ```
  enable feature lb
  Done
  show feature
  ```

```
1   Feature                  Acronym          Status
2   -------                  -------          ------   1)  Web
       Logging               WL               OFF   2)  Surge
       Protection         SP               OFF   3)  Load Balancing
                 LB               ON   .   .   .   9)  SSL
       Offloading            SSL              ON   .   .   .  Done
       <!--NeedCopy--> ```
```

GUI を使用して負荷分散を有効にするには

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ペインの [Modes and Features] で、[Change basic features] をクリックします。
3. [Configure Basic Features] ダイアログボックスで、[Load Balancing] チェックボックスをオンにして [OK] をクリックします。
4. 「Enable/Disable Feature(s)?」メッセージが表示されたら、[はい] をクリックします。

サービスと仮想サーバーの構成

負荷分散するサービスを特定したら、負荷分散の初回構成を実装できます。これを行うには、サービスオブジェクトと負荷分散仮想サーバーを作成して、それらをバインドします。

CLI を使用して初期負荷分散構成を実装するには

コマンドプロンプトで次のコマンドを入力し、初回構成を実装して確認します。

- <add service \< 名前\> \<IP アドレス\> \< サービスタイプ\> \< ポート\>
- <add lb vserver \< 仮想サーバー名\> \< サービスタイプ\> \[\<IP アドレス\> \< ポート\>\]
- <bind lb vserver \< 名前\> \< サービス名\>
- <show service bindings \< サービス名\>

  例

```
> add service service-HTTP-1 10.102.29.5 HTTP 80
  Done
> add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
  Done
> bind lb vserver vserver-LB-1 service-HTTP-1
  Done
> show service bindings service-HTTP-1
    service-HTTP-1 (10.102.29.5:80) - State : DOWN

      1)      vserver-LB-1 (10.102.29.60:80) - State : DOWN
  Done
 <!--NeedCopy-->
```

GUI を使用して初期負荷分散構成を実装するには

1. [Traffic Management]［Load Balancing］の順に選択します。
2. 詳細ペインの［Getting Started］で、［Load Balancing wizard］をクリックし、ウィザードの説明に従って基本的な負荷分散セットアップを作成します。
3. ナビゲーションペインに戻り、［Load Balancing］を展開して、［Virtual Servers］をクリックします。
4. 構成した仮想サーバーを選択し、ページの下部に表示されるパラメーターが正しく構成されていることを確認します。
5. ［開く］をクリックします。
6. ［Services］タブの各サービスで［Active］チェックボックスがオンになっていることを確認し、各サービスが仮想サーバーにバインドされていることを確認します。

パーシステンス設定

April 25, 2022

仮想サーバーにより実行されるサービスへの接続を維持したい場合（電子商取引で使用される接続など）は、その仮想サーバーに対してパーシステンスを構成する必要があります。アプライアンスは、まず構成されている負荷分散方式に基づいてサーバーを選択しますが、それ以降は同じクライアントからのすべての要求を同じサーバーに転送します。

パーシステンスを構成すると、サーバーの初回選択時以降の要求で、負荷分散方式が無視されます。構成したパーシステンスの適用先サービスがダウンしている場合は、負荷分散方式に基づいて新しいサービスが選択され、同じクライアントからのそれ以降の要求はそのサービスに永続的に割り当てられます。選択したサービスが Out Of Service 状態の場合、未処理の要求の処理は続行されますが、新しい要求や接続は受け付けられません。シャットダウン期間が経過すると、既存の接続が閉じます。次の表は、設定できるパーシステンスの種類を示しています。

永続性タイプ	固定接続数
Source IP、SSL Session ID、Rule、DESTIP、SRCIPDESTIP	250K
CookieInsert、URL passive、Custom Server ID	メモリの上限。CookieInsert の場合、タイムアウトが 0 でなければ、メモリの上限に達するまで任意の数の接続が許可されます。

表 1. 同時持続的接続の数の制限

アプライアンスのリソース不足により構成済みのパーシステンスが維持できない場合は、負荷分散方式に基づいてサーバーが選択されます。パーシステンスは、その種類で構成された時間だけ保持されます。一部のパーシステンスの種類は、特定の仮想サーバーに固有です。次の表は、それらの関係を示しています。

Persistence TypeHeader 1	**HTTP**	**HTTPS**	**TCP**	**UDP/IP**	**SSL_Bridge**
接続元 IP	はい	はい	はい	はい	はい
CookieInsert	はい	はい	いいえ	いいえ	いいえ
SSL Session ID	いいえ	はい	いいえ	いいえ	はい
URL Passive	はい	はい	いいえ	いいえ	いいえ
Custom Server ID	はい	はい	いいえ	いいえ	いいえ
規則	はい	はい	いいえ	いいえ	いいえ
SRCIPDESTIP	-	-	はい	はい	-
DESTIP	-	-	はい	はい	-

表 2. 仮想サーバーのタイプごとに使用可能な永続性タイプ

仮想サーバーのグループに対して、パーシステンスを指定することもできます。グループに対してパーシステンスを有効にすると、クライアント要求を受信した仮想サーバーに関係なく、クライアント要求は同じサーバーに送信されます。パーシステンスの構成時間が経過すると、着信したクライアント要求に対して、グループの任意の仮想サーバーが選択されます。

一般的に使用される 2 つのパーシステンスの種類は、Cookie に基づくパーシステンスと URL のサーバー ID に基づくパーシステンスです。

Cookie に基づくパーシステンスの設定

Cookie に基づくパーシステンスを有効にすると、Citrix ADC アプライアンスは、HTTP 応答の Set-Cookie ヘッダーフィールドに、HTTP Cookie を追加します。Cookie には、HTTP 要求の送信先のサービスに関する情報が含まれています。クライアントは Cookie を保存して、それ以降のすべての要求に含めます。ADC は Cookie を使用して、これらの要求に対するサービスを選択します。HTTP または HTTPS タイプの仮想サーバーに対して、この種類のパーシステンスを使用できます。

Citrix ADC アプライアンスは、<NSC_XXXX>= <ServiceIP> <ServicePort> の Cookie を挿入します。

各項目の意味は次のとおりです:

- <<NSC_XXXX> は、仮想サーバー名から導出される仮想サーバー ID です。
- <<ServiceIP> は、サービスの IP アドレスの 16 進数値です。
- <<ServicePort> は、サービスのポートの 16 進数値です。

ADC は、Cookie を挿入するときに ServiceIP と ServicePort を暗号化し、Cookie を受け取ったときにこれらを復号化します。

注: クライアントが HTTP Cookie を保存できない場合は、以降の要求に HTTP Cookie が含まれなくなり、パーシステンスは適用されません。

デフォルトでは、ADC アプライアンスは Netscape 仕様に準拠して、HTTP Cookie バージョン 0 を送信します。また、RFC 2109 に準拠して、バージョン 1 を送信することもできます。

HTTP Cookie に基づくパーシステンスに対して、タイムアウト値を設定できます。以下の点に注意してください:

- HTTP Cookie バージョン 0 が使用されている場合、Citrix ADC アプライアンスは、世界協定時刻（GMT）とタイムアウト値の合計として計算される、Cookie の有効期限（HTTP Cookie の expires 属性）の絶対 GMT を挿入します。
- HTTP Cookie バージョン 1 が使用されている場合、ADC アプライアンスは相対有効期限（HTTP Cookie の Max-Age 属性）を挿入します。この場合、クライアントソフトウェアが実際の有効期限を計算します。

注: 現在インストールされているほとんどのクライアントソフトウェア（Microsoft Internet Explorer と Netscape ブラウザー）は、HTTP Cookie バージョン 0 を理解しますが、一部の HTTP プロキシは HTTP Cookie バージョン 1 を理解します。

タイムアウト値を 0 に設定すると、使用されている HTTP Cookie バージョンに関係なく、ADC アプライアンスは有効期限を指定しなくなります。この場合、有効期限はクライアントソフトウェアに依存し、そのような Cookie は、そのソフトウェアがシャットダウンすると、無効になります。この種類のパーシステンスはシステムリソースを消費しません。したがって、好きな数だけ永続的なクライアントを含めることができます。

管理者は HTTPCookie のバージョンを変更できます。

CLI を使用して **HTTPCookie** のバージョンを変更するには

コマンドプロンプトで次を入力します。

```
set ns param [-cookieversion ( 0 | 1 )]
<!--NeedCopy-->
```

例:

```
set ns param -cookieversion 1
<!--NeedCopy-->
```

GUI を使用して **HTTPCookie** のバージョンを変更するには

1. ［System］> ［Settings］に移動します。
2. 詳細ペインで、［Change HTTP Parameters］をクリックします。
3. ［Configure HTTP Parameters］ダイアログボックスの［Cookie］で、［Version 0］または［Version 1］を選択します。

注：パラメーターについて詳しくは、「Cookie に基づくパーシステンスの設定」を参照してください。

CLI を使用して **Cookie** に基づいて永続性を構成するには

コマンドプロンプトで次のコマンドを入力し、Cookie に基づくパーシステンスを構成して確認します。

```
set lb vserver <name> -persistenceType COOKIEINSERT

show lb vserver <name>
<!--NeedCopy-->
```

例:


```
set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
Done
show lb vserver vserver-LB-1
    vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
    .
    .
    .
    Persistence: COOKIEINSERT (version 0)
    Persistence Timeout: 2 min
    .
    .
    .
 Done
<!--NeedCopy-->
```

GUI を使用して **Cookie** に基づいて永続性を構成するには

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、パーシステンスを設定する仮想サーバー（たとえば、vserver-LB-1）を選択し、[Open] をクリックします。
3. [Configure Virtual Server (Load Balancing)] ダイアログボックスの [Method and Persistence] タブにある [Persistence] リストで、[COOKIEINSERT] を選択します。
4. [Time-out (min)] テキストボックスに、タイムアウト値（たとえば、「2」）を入力します。
5. [OK] をクリックします。
6. パーシステンスを設定した仮想サーバーを選択して、ペインの下部にある [Details] セクションを表示し、仮想サーバーが正しく構成されていることを確認します。

URL のサーバー ID に基づくパーシステンスの構成

Citrix ADC アプライアンスは、URL のサーバー ID に基づいて永続性を維持できます。「URL パッシブパーシステンス」と呼ばれる方法では、ADC はサーバー応答からサーバー ID を抽出して、クライアント要求の URL クエリに埋め込みます。サーバー ID は、16 進数で表記された IP アドレスとポートです。ADC は、以降のクライアント要求からサーバー ID を抽出し、それを使用してサーバーを選択します。

URL パッシブパーシステンスでは、ペイロード式またはポリシーインフラストラクチャ式を設定し、クライアント要求に含まれるサーバー ID の場所を指定する必要があります。式について詳しくは、「ポリシーの構成とリファレンス」を参照してください。

注：サーバー ID をクライアント要求から抽出できない場合、サーバーの選択は負荷分散方式に基づいて行われます。

例：ペイロード式

式「URLQUERY contains sid=」では、クライアント要求の URL クエリから、「sid=」の後のサーバー ID が抽出されます。したがって、URLhttp://www.citrix.com/index.asp?\\&sid;=c0a864100050のリクエストは、IP アドレス 10.102.29.10 とポート 80 のサーバーに送信されます。

タイムアウト値は、この種類のパーシステンスには影響しません。このパーシステンスは、サーバー ID がクライアント要求から抽出できる限り維持されます。この種類のパーシステンスはシステムリソースを消費しないため、保持されるクライアント数に制限はありません。

> 注：パラメーターについて詳しくは、「負荷分散」を参照してください。

CLI を使用して **URL** のサーバー **ID** に基づいて永続性を構成するには

コマンドプロンプトで次のコマンドを入力し、URL のサーバー ID に基づくパーシステンスを構成して確認します。

```
set lb vserver <name> -persistenceType URLPASSIVE

<show lb vserver <name>
<!--NeedCopy-->
```

例:

```
set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
Done
show lb vserver vserver-LB-1
    vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
    .
    .
    .
    Persistence: URLPASSIVE
    Persistence Timeout: 2 min
    .
    .
    .
Done
<!--NeedCopy-->
```

GUI を使用して **URL** のサーバー **ID** に基づいて永続性を構成するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、パーシステンスを設定する仮想サーバー（たとえば、vserver-LB-1）を選択し、［Open］をクリックします。

3. ［Configure Virtual Server (Load Balancing)］ダイアログボックスの［Method and Persistence］タブにある［Persistence］リストで、［URLPASSIVE］を選択します。
4. ［Time-out (min)］テキストボックスに、タイムアウト値（たとえば、「2」）を入力します。
5. ［Rule］ボックスに、有効な式を入力します。また、［Rule］ボックスの横にある［Configure］をクリックし、［Create Expression］ダイアログボックスを使用して式を作成します。
6. ［OK］をクリックします。
7. パーシステンスを設定した仮想サーバーを選択して、ペインの下部にある［Details］セクションを表示し、仮想サーバーが正しく構成されていることを確認します。

負荷分散設定を保護する機能の構成

April 21, 2022

正しく動作していない仮想サーバーに関する通知を提供するように URL リダイレクトを構成できます。また、プライマリ仮想サーバーが使用できなくなった場合にその役割を引き継ぐバックアップ仮想サーバーを構成することもできます。

URL リダイレクトの構成

HTTP または HTTPS タイプの仮想サーバーがダウンしたり無効になったりしたときに、アプライアンスのステータスを通信するためのリダイレクト URL を構成できます。この URL は、ローカルリンクでもリモートリンクでも構いません。アプライアンスでは、HTTP 302 リダイレクトが使用されます。

リダイレクトは、絶対 URL でも相対 URL でも構いません。構成したリダイレクト URL に絶対 URL が含まれている場合、着信した HTTP 要求で指定された URL に関係なく、その絶対 URL にリダイレクトされます。構成したリダイレクト URL にドメイン名のみが含まれている場合（相対 URL）、そのドメインに着信 URL を追記した場所にリダイレクトされます。

注：負荷分散仮想サーバーで、バックアップ仮想サーバーとリダイレクト URL の両方を構成した場合、バックアップ仮想サーバーがリダイレクト URL よりも優先されます。この場合は、プライマリおよびバックアップ仮想サーバーの両方がダウンしているときに、リダイレクトが使用されます。

CLI を使用してクライアント要求を URL へリダイレクトするように仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力し、クライアント要求が URL にリダイレクトされるように仮想サーバーを構成して確認します。

```
set lb vserver <name> -redirectURL <URL>

```



```
show lb vserver <name>
<!--NeedCopy-->
```

例:

```
    > set lb vserver vserver-LB-1 -redirectURL <http://www.newdomain.com
       /mysite/maintenance>
      Done
    > show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+666 ms)
        .
        .
        .
        Redirect URL: <http://www.newdomain.com/mysite/maintenance>
        .
        .
        .
      Done
     >
<!--NeedCopy-->
```

GUI を使用してクライアント要求を **URL** へリダイレクトするように仮想サーバーを構成するには

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、URL リダイレクトを構成する仮想サーバー（たとえば、vserver-LB-1）を選択し、[Open] をクリックします。
3. [Configure Virtual Server (Load Balancing)]ダイアログボックスで、[Advanced]タブの[Redirect URL]テキストボックスに、URL（たとえば、「http://www.newdomain.com/mysite/maintenance」）を入力して［OK］をクリックします。
4. サーバーに設定したリダイレクト URL が、ペインの下部にある［Details］セクションに表示されていることを確認します。

バックアップ仮想サーバーの設定

プライマリ仮想サーバーがダウンしているか無効である場合、アプライアンスは接続またはクライアント要求をバックアップ仮想サーバーに送信し、クライアントトラフィックをバックアップ仮想サーバーからサービスに転送できます。アプライアンスは、サイトの停止またはメンテナンスに関する通知メッセージをクライアントに送信することもできます。バックアップ仮想サーバーはプロキシであり、クライアントに対して透過的です。

仮想サーバーを作成したり、既存の仮想サーバーのオプションパラメーターを変更したりする場合は、バックアップ仮想サーバーを構成できます。また、既存のバックアップ仮想サーバーに対してバックアップ仮想サーバーを構成し、カスケードされたバックアップ仮想サーバーを作成することもできます。バックアップ仮想サーバーをカスケードする最大の深さは、10 です。アプライアンスは、起動しているバックアップ仮想サーバーを検索し、その仮想サーバーにアクセスしてコンテンツを提供します。

プライマリおよびバックアップ仮想サーバーがダウンしたり、それらのサーバーの処理要求数がしきい値に達したりしたときに、プライマリで URL がリダイレクトされるように構成できます。

注：バックアップ仮想サーバーが存在しない場合は、リダイレクト URL を構成しないとエラーメッセージが表示されます。バックアップ仮想サーバーとリダイレクト URL の両方が構成されている場合は、バックアップ仮想サーバーが優先されます。

CLI を使用してバックアップ仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力し、バックアップサーバーを構成して確認します。

```
set lb vserver <name> [-backupVserver <string>]

show lb vserver <name>
<!--NeedCopy-->
```

例：

```
    > set lb vserver vserver-LB-1 -backupVserver vserver-LB-2
      Done
    > show lb vserver vserver-LB-1
         vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
         State: DOWN
         Last state change was at Wed Jun 17 08:56:34 2009 (+661 ms)
         .
         .
         .
         Backup: vserver-LB-2
         .
         .
         .
      Done
     >
<!--NeedCopy-->
```


GUI を使用してバックアップ仮想サーバーをセットアップするには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、バックアップ仮想サーバーを設定する仮想サーバー（たとえば、vserver-LB-1）を選択し、［Open］をクリックします。
3. ［Configure Virtual Server (Load Balancing)］ダイアログボックスの［Advanced］タブにある［Backup Virtual Server］リストで、バックアップ仮想サーバー（たとえば、vserver-LB-2）を選択し、［OK］をクリックします。
4. 設定したバックアップ仮想サーバーが、ペインの下部にある［Details］セクションに表示されていることを確認します。

   注：ダウンしたプライマリサーバーが復帰した場合でも、その仮想サーバーをプライマリとして明示的に再設定するまでバックアップ仮想サーバーがプライマリサーバーとして動作するようにするには、［Disable Primary When Down］チェックボックスをオンにします。

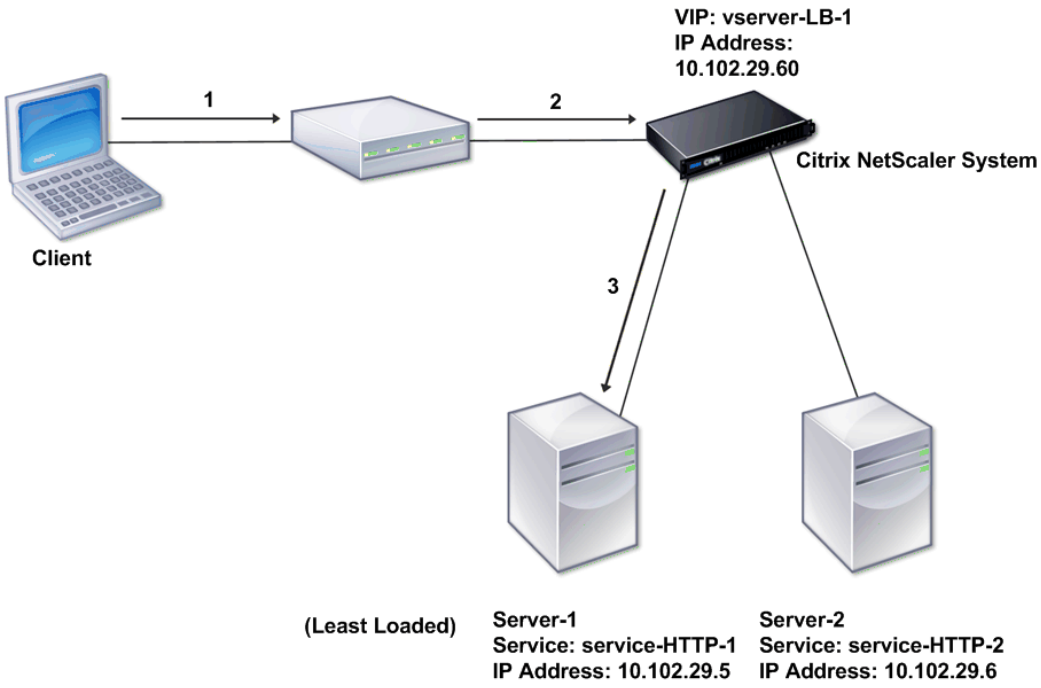
一般的な負荷分散シナリオ

April 25, 2022

負荷分散セットアップでは、Citrix ADC アプライアンスはクライアントとサーバーファームの間に論理的に配置され、サーバーへのトラフィックフローを管理します。

次の図は、基本的な負荷分散構成のトポロジを示しています。

図 1：基本的な負荷分散トポロジ

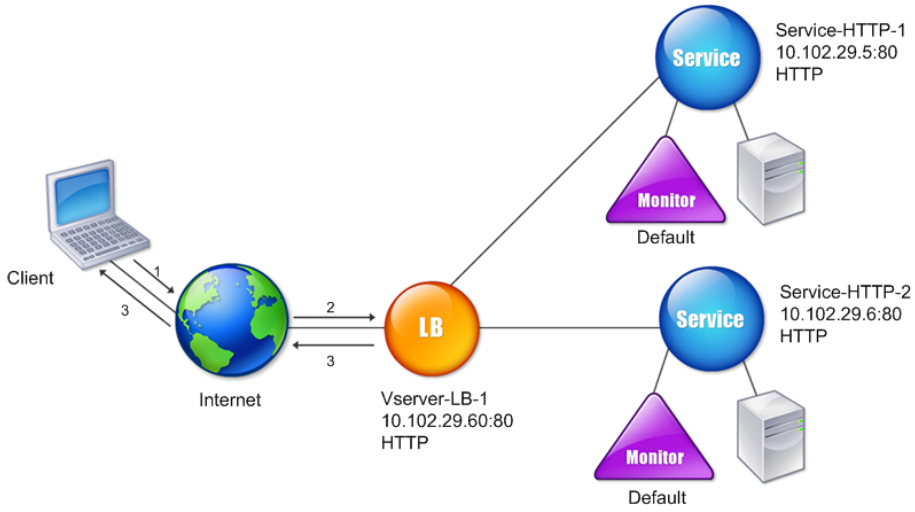


仮想サーバーは、クライアントからの要求に対してサービスを選択して割り当てます。サービス「service-HTTP-1」と「service-HTTP-2」が作成されて、「virtual server-LB-1」という仮想サーバーにバインドされている、前の図のシナリオについて考えてみましょう。virtual server-LB-1 は、クライアント要求を service-HTTP-1 または service-HTTP-2 に転送します。システムは Least Connections 負荷分散方式を使用して、各要求のサービスを選択します。次の表は、システムで設定する必要がある基本的なエンティティの名前と値を示しています。

表 1. LB 構成パラメーター値

次の図は、前の表で説明した負荷分散のサンプル値と、必須パラメーターを示しています。

図 2：負荷分散エンティティモデル



次の表に、コマンドラインインターフェイスを使用してこの負荷分散セットアップを構成するためのコマンドを示します。

タスク	コマンド
負荷分散を有効にする	enable feature lb
「service-HTTP-1」というサービスを作成する	add service service-HTTP-1 10.102.29.5 HTTP 80
「service-HTTP-2」というサービスを作成する	add service service-HTTP-2 10.102.29.6 HTTP 80
「vserver-LB-1」という仮想サーバーを作成する	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
「service-HTTP-1」というサービスを「vserver-LB-1」という仮想サーバーにバインドする	bind lb vserver vserver-LB-1 service-HTTP-1
「service-HTTP-2」というサービスを「vserver-LB-1」という仮想サーバーにバインドする	bind lb vserver vserver-LB-1 service-HTTP-2

表 2. 初期構成タスク

初期構成タスクの詳細については、「基本的な負荷分散のセットアップ」を参照してください。

タスク	コマンド
「vserver-LB-1」という仮想サーバーのプロパティを表示する	show lb vserver vserver-LB-1
「vserver-LB-1」という仮想サーバーの統計情報を表示する	stat lb vserver vserver-LB-1
「service-HTTP-1」というサービスのプロパティを表示する	show service service-HTTP-1
「service-HTTP-1」というサービスの統計情報を表示する	stat service service-HTTP-1
「service-HTTP-1」というサービスのバインド情報を表示する	show service bindings service-HTTP-1

表 3. 検証タスク

タスク	コマンド
「vserver-LB-1」という仮想サーバーにパーシステンスを構成する	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
「vserver-LB-1」という仮想サーバーに COOKIEINSERT パーシステンスを構成する	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
「vserver-LB-1」という仮想サーバーに URLPassive パーシステンスを構成する	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
クライアント要求を「vserver-LB-1」という仮想サーバー上の URL へリダイレクトするように仮想サーバーを構成する	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
「vserver-LB-1」という仮想サーバーにバックアップ仮想サーバーを設定する	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

表 4. カスタマイズタスク

パーシステンスの構成について詳しくは、「パーシステンス設定の選択と構成」を参照してください。クライアント要求を URL にリダイレクトするための仮想サーバーの構成およびバックアップ仮想サーバーのセットアップについて詳しくは、「負荷分散設定を保護する機能の構成」を参照してください。

ユースケース：Citrix ADC アプライアンスを使用して Web サイトに Secure および HttpOnly Cookie オプションを強制する方法

April 21, 2022

Web 管理者は、Secure、HttpOnly、またはセッション ID のフラグと Web アプリケーションによって生成される認証 Cookie の両方を強制する場合があります。HTTP 負荷分散仮想サーバーと Citrix ADC アプライアンスの書き換えポリシーを使用してこれらの 2 つのオプションを含めるように、Set-cookie ヘッダーを変更することができます。

- **HttpOnly** - Cookie のこのオプションにより、Web ブラウザーは HTTP または HTTPS プロトコルのみを使用して Cookie を返します。JavaScript の document.cookie 参照などの非 HTTP メソッドは、Cookie にアクセスできません。このオプションは、クロスサイトスクリプティングによる Cookie の盗難を防ぐのに役立ちます。

  > 注
  >
  > JavaScript やクライアント側の Java アプレットなどのクライアント側のスクリプトを使用して Web アプリケーションが Cookie のコンテンツにアクセスする必要がある場合は、HttpOnly オプションを使用できません。このドキュメントに記載されている方法を使用すると、サーバーで生成された Cookie のみを書き換えることができ、Citrix ADC アプライアンスで生成された Cookie は書き換えることができません。たとえば、AppFirewall、永続性、VPN セッションの Cookie などです。

- **Secure** - Cookie のこのオプションにより、送信が SSL で暗号化されている場合、Web ブラウザーは Cookie の値のみを返します。このオプションは、接続の傍受による Cookie の盗難を防ぐために使用できます。

> 注
>
> 次の手順は、VPN 仮想サーバーには適用されません。

CLI を使用して既存の HTTP 仮想サーバーで Secure フラグと HttpOnly フラグを強制するように Citrix ADC アプライアンスを構成するには

1. 書き換えアクションを作成します。

   この例は、Secure フラグと HttpOnly フラグの両方を設定するように構成されています。いずれかが欠落している場合は、必要に応じて他の組み合わせに変更してください。

```
add rewrite action act_cookie_Secure replace_all http.RES.
    full_Header ””Secure; HttpOnly; path=/”” -search ”regex(re!(
    path=/\\; Secure; HttpOnly)|(path=/\\; Secure)|(path=/\\;
    HttpOnly)|(path=/)!)” -bypassSafetyCheck YES
```



```
<!--NeedCopy-->
```

このポリシーは、次のすべてのインスタンスを置き換えます。「path=/」、「path=/; Secure」、「path=/; Secure; HttpOnly」、「path=/; HttpOnly」と「Secure; HttpOnly; path=/」。大文字と小文字が一致しない場合、この正規表現（regex）は失敗します。

2. アクションをトリガーするための書き換えポリシーを作成します。

```
add rewrite policy rw_force_secure_cookie ”http.RES.HEADER(”Set-Cookie”).EXISTS” act_cookie_Secure
<!--NeedCopy-->
```

3. 書き換えポリシーをセキュリティで保護する仮想サーバーにバインドします。Secureオプションを使用する場合は、SSL 仮想サーバーを使用する必要があります。

```
bind lb vserver mySSLVServer -policyName rw_force_secure_cookie -priority 100 -gotoPriorityExpression NEXT -type RESPONSE
<!--NeedCopy-->
```

例:

以下は、httpOnly フラグを設定する前の Cookie の例です

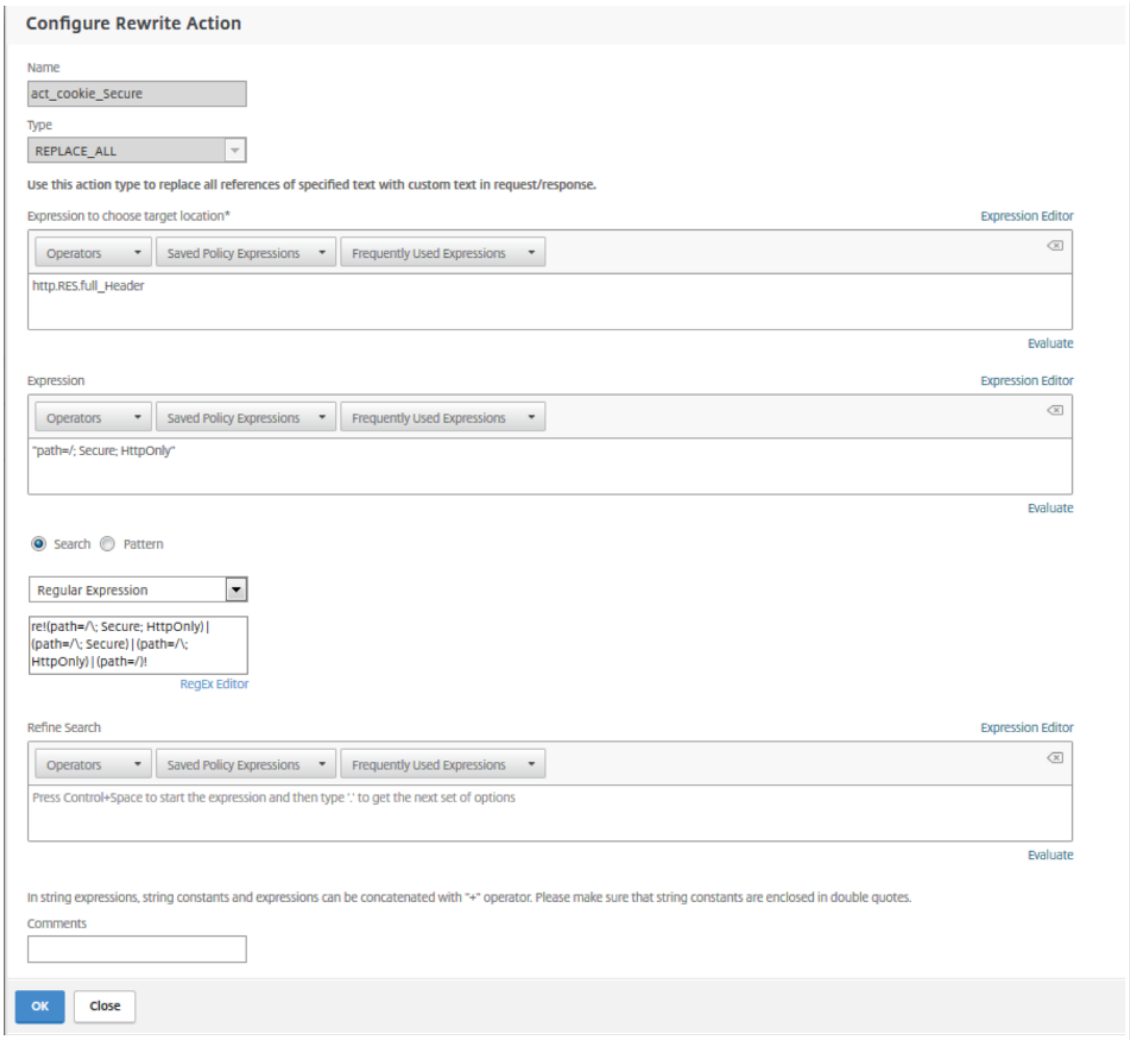
```
Set-Cookie: CtxsAuthId=C5614491; path=/Citrix/ProdWeb
<!--NeedCopy-->
```

以下は、httpOnly フラグを設定した後の Cookie の例です

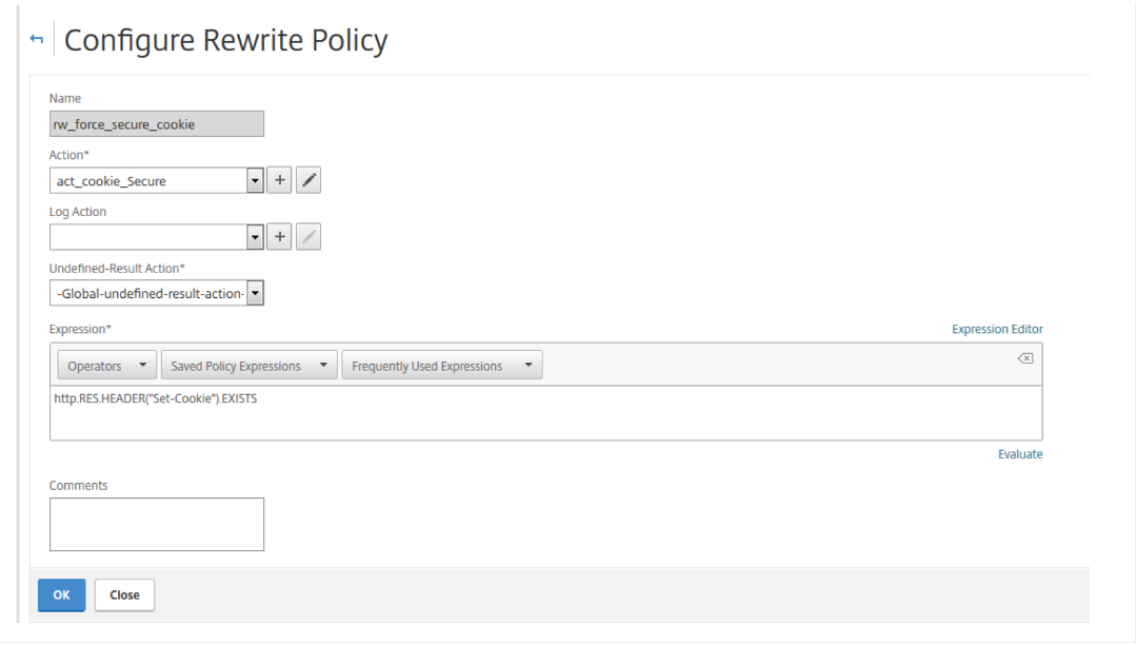
```
Set-Cookie: CtxsAuthId=C5614491; Secure; HttpOnly; path=/Citrix/ProdWeb/
<!--NeedCopy-->
```

GUI を使用して既存の HTTP 仮想サーバーで Secure フラグと HttpOnly フラグを強制するように Citrix ADC アプライアンスを構成するには

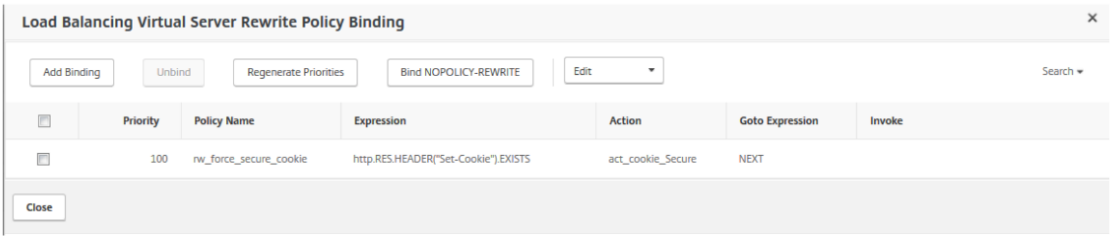
1. [**AppExpert**] > [**Rewrite**] > [**Actions**] の順に移動し、[**Add**] をクリックして新しい書き換えアクションを追加します。



2. [**AppExpert**] > [**Rewrite**] > [**Policies**] の順に移動し、[**Add**] をクリックして新しい書き換えアクションを追加します。



3. [**Traffic Management**] > [**Load Balancing**] > [**Virtual Servers**] に移動して書き換え（応答）ポリシーを対応する SSL 仮想サーバーにバインドします。



圧縮による負荷分散トラフィックの速度向上

April 21, 2022

圧縮は、帯域幅の使用を最適化する一般的な手段であり、ほとんどの Web ブラウザーで圧縮データがサポートされています。圧縮機能を有効にすると、Citrix ADC アプライアンスがクライアントからの要求をインターセプトして、そのクライアントが圧縮コンテンツに対応しているかどうかを判断します。また、アプライアンスがサーバーからの HTTP 応答を受信すると、そのコンテンツを調べて圧縮可能かどうかを決定します。コンテンツが圧縮可能な場合、アプライアンスはコンテンツを圧縮し、応答のヘッダーを変更して実行した圧縮の種類を示し、圧縮コンテンツをクライアントに転送します。

Citrix ADC 圧縮は、ポリシーベースの機能です。ポリシーは要求と応答をフィルタリングして圧縮される応答を特定し、各応答に適用する圧縮の種類を指定します。アプライアンスは、text/html、text/plain、text/xml、text/css、text/rtf、application/msword、application/vnd.ms-excel、application/vnd.ms-powerpoint などの一般的な MIME タイプを圧縮する複数の組み込みポリシーを提供します。また、カスタムポリシーを作成することもでき

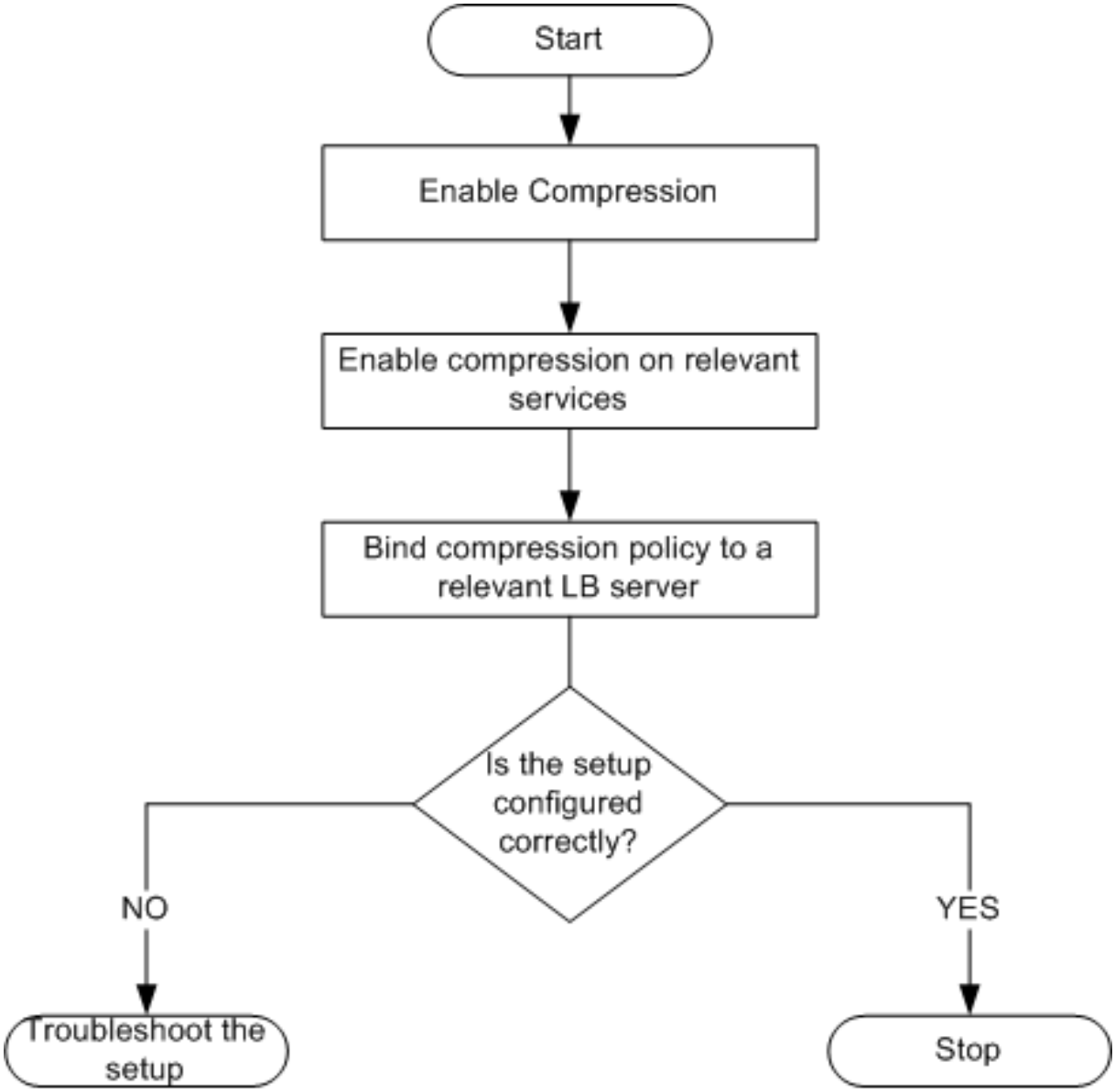
ます。アプライアンスは、application/octet-stream、binary、bytes などの圧縮済みの MIME タイプや、GIF、JPEG などの画像形式を圧縮しません。

圧縮を構成するには、グローバルな圧縮機能を有効にしてから、圧縮対象の応答を配信するサービスごとに圧縮を有効にする必要があります。負荷分散またはコンテンツスイッチ向けに仮想サーバーを構成済みの場合は、それらの仮想サーバーにポリシーをバインドする必要があります。それ以外の場合、アプライアンスを経由するすべてのトラフィックにポリシーが適用されます。

圧縮を構成するタスクの順序

次のフローチャートは、負荷分散セットアップで基本的な圧縮を構成するタスクの順序を示しています。

図 1: 圧縮を構成するためのタスクの順序



> 注：上図の手順では、負荷分散が構成済みであることが想定されています。

圧縮を有効化

デフォルトでは、圧縮が無効になっています。クライアントに送信される HTTP 応答の圧縮を許可するには、圧縮機能を有効にする必要があります。

CLI を使用して圧縮を有効にするには

コマンドプロンプトで次のコマンドを入力し、圧縮を有効化して構成を確認します。

- enable ns feature CMP
- show ns feature

```
> enable ns feature CMP



 Done

> show ns feature



        Feature                         Acronym             Status

        -------                         -------             ------

 1)     Web Logging                     WL                  ON

 2)     Surge Protection                SP                  OFF

 .


```



```
30      7) Compression Control CMP ON
31
32
33      8)      Priority Queuing                PQ                      OFF
34
35
36      .
37
38
39      Done
40
41  <!--NeedCopy-->
```

GUI を使用して圧縮を有効にするには

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ペインの [Modes and Features] で、[Change basic features] をクリックします。
3. [Configure Basic Features] ダイアログボックスで、[Compression] チェックボックスをオンにしてから [OK] をクリックします。
4. [Enable/Disable Feature(s)?] ダイアログボックスで、[Yes] をクリックします。

データを圧縮するサービスの設定

グローバルな圧縮設定を有効にしたら、圧縮対象のファイルを配信するサービスごとに圧縮を有効にする必要があります。

CLI を使用して特定のサービスの圧縮を有効にするには

コマンドプロンプトで次のコマンドを入力し、特定のサービスの圧縮を有効化して構成を確認します。

- set service \< 名前\> -CMP YES
- show service \< 名前\>

```
1  > show service SVC_HTTP1
2
3
4  SVC_HTTP1 (10.102.29.18:80) - HTTP
5
6
7  State: UP
8
```



```
9
10 Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)
11
12
13 Time since last state change: 0 days, 03:03:37.200
14
15
16 Server Name: 10.102.29.18
17
18
19 Server ID : 0   Monitor Threshold : 0
20
21
22 Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
23
24
25 Use Source IP: NO
26
27
28 Client Keepalive(CKA): NO
29
30
31 Access Down Service: NO
32
33
34 TCP Buffering(TCPB): NO
35
36
37 HTTP Compression(CMP): YES
38
39
40 Idle timeout: Client: 180 sec   Server: 360 sec
41
42
43 Client IP: DISABLED
44
45
46 Cacheable: NO
47
48
49 SC: OFF
50
51
52 SP: OFF
53
```



```
54
55 Down state flush: ENABLED
56
57 1)      Monitor Name: tcp-default
58
59
60 State: DOWN     Weight: 1
61
62
63 Probes: 1095    Failed [Total: 1095 Current: 1095]
64
65
66 Last response: Failure - TCP syn sent, reset received.
67
68
69 Response Time: N/A
70
71
72 Done
73
74 <!--NeedCopy-->
```

GUI を使用して特定のサービスの圧縮を有効にするには

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. 詳細ペインで圧縮を設定するサービス（たとえば、[service-HTTP-1]）を選択し、[Open] をクリックします。
3. [Advanced] タブの［Settings］で、[Compression] チェックボックスをオンにして［OK］をクリックします。
4. サービスが選択されている場合はペインの下部にある［**Details**］に［HTTP Compression(CMP): ON］が表示されていることを確認します。

仮想サーバーへの圧縮ポリシーのバインド

仮想サーバーに圧縮ポリシーをバインドすると、ポリシーは、その仮想サーバーに関連付けられたサービスによってのみ評価されます。仮想サーバーへの圧縮ポリシーのバインドは、[Configure Virtual Server (Load Balancing)] ダイアログボックスまたは［Compression Policy Manager］ダイアログボックスを使用して行います。このトピックには、[Configure Virtual Server（Load Balancing)］ダイアログボックスを使用して、圧縮ポリシーを負荷分散仮想サーバーにバインドする手順が含まれています。

コマンドラインを使用して仮想サーバーへの圧縮ポリシーをバインドまたはバインド解除するには

コマンドプロンプトで次のコマンドを入力し、負荷分散仮想サーバーへの圧縮ポリシーをバインドまたはバインド解除して構成を確認します。

- (bind|unbind) lb vserver \< 名前\> -policyName \< 文字列\>
- show lb vserver \< 名前\>

例:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp
Done
> showlbvserverlbvip

lbvip(8.7.6.6:80)-HTTPType:ADDRESS
State:UP
LaststatechangewasatThuMay2805:37:212009(+685ms)
Timesincelaststatechange:19days,04:26:50.470
EffectiveState:UP
ClientIdleTimeout:180sec
Downstateflush:ENABLED
DisablePrimaryVserverOnDown:DISABLED
PortRewrite:DISABLED
No.ofBoundServices:1(Total)1(Active)
ConfiguredMethod:LEASTCONNECTION
CurrentMethod:RoundRobin,Reason:Boundservice'sstatechangedtoUP
Mode:IP
Persistence:NONE
VserverIPandPortinsertion:OFF
Push:DISABLEDPushVServer:
PushMultiClients:NO
PushLabelRule:

BoundServiceGroups:
1)GroupName:Service-Group-1

1)Service-Group-1(10.102.29.252:80)-HTTPState:UPWeight:1

1)Policy:ns_cmp_msappPriority:0

Done

<!--NeedCopy-->
```


GUI を使用して負荷分散仮想サーバーへの圧縮ポリシーをバインドまたはバインド解除するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ペインで、圧縮ポリシーのバインドまたはバインド解除を行う仮想サーバー（たとえば、[Vserver-LB-1]）を選択し、［Open］をクリックします。
3. [Configure Virtual Server (Load Balancing)] ダイアログボックスで、[Policies] タブの [Compression] をクリックします。
4. 次のいずれかを行います：
   - 圧縮ポリシーをバインドする場合は、［Insert Policy］をクリックしてから仮想サーバーにバインドするポリシーを選択します。
   - 圧縮ポリシーをバインド解除する場合は、仮想サーバーからバインド解除するポリシーの名前を選択し、［Unbind Policy］をクリックします。
5. ［OK］をクリックします。

SSL による負荷分散トラフィックのセキュリティ保護

April 25, 2022

Citrix ADC SSL オフロード機能は、SSL トランザクションを行う Web サイトのパフォーマンスを、透過的に向上させます。SSL オフロードでは、CPU 負荷の高い SSL 暗号化および復号化タスクをローカル Web サーバーからアプライアンスにオフロードすることにより、SSL データの処理によるサーバーパフォーマンスの低下を引き起こすことなく、Web アプリケーションを安全に配信できます。SSL トラフィックを復号化すると、あらゆる標準サービスで処理できるようになります。SSL プロトコルは、さまざまな種類の HTTP および TCP データとシームレスに機能して、このようなデータを使用するトランザクションに、セキュリティ保護されたチャネルを提供します。

SSL を設定するには、まず SSL を有効にする必要があります。次に、アプライアンスで HTTP または TCP サービスおよび SSL 仮想サーバーを構成し、そのサービスを仮想サーバーにバインドします。証明書とキーのペアを追加して SSL 仮想サーバーにバインドする必要もあります。Outlook Web Access サーバーを使用する場合は、SSL サポートを有効にするアクションとそのアクションに適用するポリシーを作成する必要があります。SSL 仮想サーバーは、暗号化された着信トラフィックをインターセプトして、ネゴシエートしたアルゴリズムを使用してそのトラフィックを復号化します。復号化されたデータは、アプライアンス上のほかのエンティティに転送され、適切に処理されます。

SSL オフロードの詳細については、「SSL オフロードおよび SSL アクセラレーション」を参照してください。

SSL を設定するタスクの順序

SSL を設定するには、まず SSL を有効にする必要があります。次に、Citrix ADC アプライアンスで SSL 仮想サーバーと HTTP または TCP サービスを作成する必要があります。最後に、有効な SSL 証明書と設定済みのサービスを、SSL 仮想サーバーにバインドする必要があります。

SSL 仮想サーバーは、暗号化された着信トラフィックを傍受して、ネゴシエートしたアルゴリズムを使用してそのト

ラフィックを解読します。復号化されたデータは、Citrix ADC アプライアンス上のほかのエンティティに転送され、適切に処理されます。

次のフローチャートには、基本的な SSL オフロードセットアップを設定するタスクの順序が示されています。

図 1: SSL オフロードを構成するための一連のタスク

SSL オフロードを有効にする

まず、SSL 機能を有効にします。SSL 機能を有効にしなくてもアプライアンス上で SSL ベースのエンティティを設定できますが、それらのエンティティは SSL を有効にするまで動作しません。

CLI を使用して SSL を有効にする

コマンドプロンプトで次のコマンドを入力し、SSL オフロードを有効にして構成を確認します。

```
-  enable ns feature SSL
-  show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature ssl

Done


> show ns feature


Feature Acronym Status


------- ------- ------


1) Web Logging WL ON


2) SurgeProtection SP OFF


3) Load Balancing LB ON . . .


 9) SSL Offloading SSL ON


10) Global Server Load Balancing GSLB ON . .
```



```
28
29
30 Done >
31 <!--NeedCopy-->
```

GUI を使用して **SSL** を有効にする

次の手順を実行します:

1. ナビゲーションペインで、[**System**] を展開し、[**Settings**] をクリックします。
2. 詳細ペインの [**Modes and Features**] で、[**Change basic features**] をクリックします。
3. [**SSL Offloading**] チェックボックスをオンにして、[**OK**] をクリックします。
4. [**Enable/Disable Feature(s)?**] ダイアログボックスで、[**Yes**] をクリックします。

HTTP サービスを作成する

アプライアンス上の各サービスは、サーバー上の個々のアプリケーションとして機能します。構成したサービスは、アプライアンスがネットワーク上のサーバーにアクセスしてその状態を監視できるようになるまで無効状態になります。このトピックでは、HTTP サービスを作成する手順について説明します。

注: TCP トラフィックの場合、次の手順を実行しますが、HTTP サービスの代わりに TCP サービスを作成します。

CLI を使用して **HTTP** サービスを追加します

コマンドプロンプトで次のコマンドを入力し、HTTP サービスを追加して構成を確認します。

```
1 -  add service <name> (<IP> | <serverName>) <serviceType> <port>
2 -  show service <name>
3 <!--NeedCopy-->
```

例:

```
1 > add service SVC_HTTP1 10.102.29.18 HTTP 80
2
3
4  Done
5
6
7 > show service SVC_HTTP1
8
```



```
9
10          SVC_HTTP1 (10.102.29.18:80) - HTTP
11
12
13          State: UP
14
15
16          Last state change was at Wed Jul 15 06:13:05 2009
17
18
19          Time since last state change: 0 days, 00:00:15.350
20
21
22          Server Name: 10.102.29.18
23
24
25          Server ID : 0   Monitor Threshold : 0
26
27
28          Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
29
30
31          Use Source IP: NO
32
33
34          Client Keepalive(CKA): NO
35
36
37          Access Down Service: NO
38
39
40          TCP Buffering(TCPB): NO
41
42
43          HTTP Compression(CMP): YES
44
45
46          Idle timeout: Client: 180 sec   Server: 360 sec
47
48
49          Client IP: DISABLED
50
51
52          Cacheable: NO
53
```



```
54
55          SC: OFF
56
57
58          SP: OFF
59
60
61          Down state flush: ENABLED
62
63
64
65
66
67  1)      Monitor Name: tcp-default
68
69
70                  State: UP       Weight: 1
71
72
73                  Probes: 4       Failed [Total: 0 Current: 0]
74
75
76                  Last response: Success - TCP syn+ack received.
77
78
79                  Response Time: N/A
80
81
82   Done
83  <!--NeedCopy-->
```

GUI を使用して **HTTP** サービスを追加します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Servers**] の順に選択します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**Create Service**] ダイアログボックスで、サービスの名前、IP アドレス、およびポートを入力します（たとえば、SVC_HTTP1, 10.102.29.18、および 80）。
4. [**Protocol**] ボックスの一覧で、サービスの種類（ここでは [HTTP]）を選択します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。構成した HTTP サービスが、[Services] ページに表示されます。
6. 作成したサービスを選択して、ペインの下部にある [Details] セクションを表示し、パラメーターが正しく構

成されていることを確認します。

SSL ベースの仮想サーバーを追加する

基本的な SSL オフロードセットアップでは、SSL 仮想サーバーは暗号化されたトラフィックをインターセプトおよび復号化して、仮想サーバーにバインドされているサービスにクリアテキストメッセージを送信します。CPU 負荷の高い SSL 処理をアプライアンス側にオフロードすると、バックエンドサーバーでより多くの要求を処理できるようになります。

CLI を使用して SSL ベースの仮想サーバーを追加します

コマンドプロンプトで次のコマンドを入力し、SSL ベースの仮想サーバーを追加して構成を確認します。

```
-  add lb vserver <name> <serviceType> [<IPAddress> <port>]
-  show lb vserver <name>
<!--NeedCopy-->
```

> 注意：安全に接続するには、SSL ベースの仮想サーバーを有効にする前に、有効な SSL 証明書を SSL ベースの仮想サーバーにバインドする必要があります。

例：

```
> add lb vserver vserver-SSL-1 SSL 10.102.29.50 443
  Done


 > show lb vserver vserver-SSL-1


  vserver-SSL-1 (10.102.29.50:443) - SSL Type: ADDRESS


  State: DOWN[Certkey not bound] Last state change was at Tue Jun 16
      06:33:08 2009 (+176 ms)


  Time since last state change: 0 days, 00:03:44.120


  Effective State: DOWN Client Idle Timeout: 180 sec

```



```
19
20    Down state flush: ENABLED
21
22
23    Disable Primary Vserver On Down : DISABLED
24
25
26    No. of Bound Services : 0 (Total) 0 (Active)
27
28
29    Configured Method: LEASTCONNECTION Mode: IP
30
31
32    Persistence: NONE
33
34
35    Vserver IP and Port insertion: OFF
36
37
38    Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:
          Done
39  <!--NeedCopy-->
```

GUI を使用して **SSL** ベースの仮想サーバーを追加します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**Create Virtual Server (SSL Offload)**] ダイアログボックスで、仮想サーバーの名前、IP アドレス、およびポートを入力します。
4. [**Protocol**] ボックスの一覧で、仮想サーバーの種類（たとえば、[SSL]）を選択します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。
6. 作成した仮想サーバーを選択して、ペインの下部にある [Details] セクションを表示し、パラメーターが正しく構成されていることを確認します。証明書とキーのペアとサービスが仮想サーバーにバインドされていないため、仮想サーバーは DOWN としてマークされます。

> 注意：安全に接続するには、SSL ベースの仮想サーバーを有効にする前に、有効な SSL 証明書を SSL ベースの仮想サーバーにバインドする必要があります。

サービスの **SSL** 仮想サーバーへのバインド

SSL 仮想サーバーが復号化した受信データは、その仮想サーバーにバインドされたサービスに転送されます。

アプライアンスとサーバーの間のデータ転送は、暗号化したりクリアテキストで送信したりできます。アプライアンスとサーバーの間のデータ転送を暗号化する場合、トランザクション全体がエンドツーエンドで保護されることになります。エンドツーエンドのセキュリティのためのシステム構成について詳しくは、「SSL オフロードおよびアクセラレーション」を参照してください。

CLI を使用してサービスを仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力し、サービスを SSL 仮想サーバーにバインドして構成を確認します。

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver <name>
<!--NeedCopy-->
```

例:

```
> bind lb vserver vserver-SSL-1 SVC_HTTP1



  Done

 > show lb vserver vserver-SSL-1 vserver-SSL-1 (10.102.29.50:443) - SSL
     Type:

  ADDRESS State: DOWN[Certkey not bound]

  Last state change was at Tue Jun 16 06:33:08 2009 (+174 ms)

  Time since last state change: 0 days, 00:31:53.70

  Effective State: DOWN Client Idle

  Timeout: 180 sec

```



```
26
27    Down state flush: ENABLED Disable Primary Vserver On Down :
28
29
30    DISABLED No. of Bound Services : 1 (Total) 0 (Active)
31
32
33    Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver
          IP and
34
35
36    Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients:
         NO Push Label Rule:
37
38
39
40
41
42    1) SVC_HTTP1 (10.102.29.18: 80) - HTTP
43
44
45    State: DOWN Weight: 1
46
47
48    Done
49  <!--NeedCopy-->
```

GUI を使用してサービスを仮想サーバーにバインドする

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 詳細ペインで仮想サーバーを選択して、[**Open**] をクリックします。
3. [**Services**] タブの [**Active**] 列で、選択した仮想サーバーにバインドするサービスの横にあるチェックボックスをオンにします。
4. [**OK**] をクリックします。
5. ペインの下部にある [Details] セクションの [Number of Bound Services] カウンターが、仮想サーバーにバインドしたサービスの数だけ増加することを確認します。

証明書とキーのペアを追加します

SSL 証明書は、SSL キー交換および暗号化/復号化プロセスに含まれるエレメントです。証明書は、SSL サーバーのアイデンティティを確立するために SSL ハンドシェイク中に使用されます。Citrix ADC アプライアンスに含まれている、有効な既存の SSL 証明書を使用するか、独自の SSL 証明書を作成できます。アプライアンスは、最大 4096

ビットの RSA 証明書をサポートします。

次の曲線の ECDSA 証明書のみがサポートされています:

- prime256v1（ADC 上の P_256）
- secp384r1 (ADC 上の P_384）
- secp521r1 (ADC 上の P_521、VPX でのみサポートされます）
- secp224r1 (ADC 上の P_224、VPX でのみサポートされます）

> 注：信頼される証明機関から発行された有効な SSL 証明書を使用することをお勧めします。無効な証明書や自分で作成した証明書は、一部の SSL クライアントと互換性がありません。

SSL 処理に使用する前に、証明書を対応するキーとペアにする必要があります。次に、証明書とキーのペアを仮想サーバーにバインドすると、SSL 処理に使用できるようになります。

CLI を使用して証明書キーペアを追加します

> 注： ECDSA 証明書とキーのペアの作成については、「ECDSA 証明書とキーのペアの作成」を参照してください。

コマンドプロンプトで次のコマンドを入力し、証明書とキーのペアを作成して構成を確認します。

```
1 -  add ssl certKey <certkeyName> -cert <string> [-key <string>]
2 -  show sslcertkey <name>
3 <!--NeedCopy-->
```

例:

```
1 > add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key
2
3  Done
4
5
6 > show sslcertkey CertKey-SSL-1
7
8
9    Name: CertKey-SSL-1 Status: Valid,
10
11
12   Days to expiration:4811 Version: 3
13
14
15   Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer:
        C=US,ST=California,L=San
```



```
16
17
18     Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
19
20
21     Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17
           21:26:47 2022 GMT
22
23
24     Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,
           CN=d efault Public Key
25
26
27     Algorithm: rsaEncryption Public Key
28
29
30     size: 1024
31
32
33   Done
34  <!--NeedCopy-->
```

GUI を使用して証明書キーペアを追加します

次の手順を実行します:

1. [**Traffic Management**] **>** [**SSL**] **>** [**Certificates**] に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**Install Certificate**] ダイアログボックスの [Certificate-Key Pair Name] ボックスに、追加する証明書とキーのペアの名前（たとえば、「Certkey-SSL-1」）を入力します。
4. [**Details**] の [Certificate File Name] で、[**Browse (Appliance)**] をクリックして証明書を検索します。証明書とキーはともに、アプライアンスの/nsconfig/ssl/ディレクトリに保存されます。ローカルシステムにある証明書を使用するには、[Local] を選択します。
5. 使用する証明書を選択して、[**Select**] をクリックします。
6. [Private Key File Name] で、[**Browse (Appliance)**] をクリックして秘密キーファイルを検索します。ローカルシステムにある秘密キーを使用するには、[Local] を選択します。
7. 使用するキーを選択して、[**Select**] をクリックします。証明書とキーのペアで使用するキーを暗号化するには、暗号化に使用するパスワードを [Password] ボックスに入力します。
8. [インストール] をクリックします。
9. 証明書キーのペアをダブルクリックし、[証明書の詳細] ウィンドウで、パラメーターが正しく構成されて保存されていることを確認します。

SSL 証明書キーペアの仮想サーバーへのバインド

SSL 証明書とそれに対応するキーをペアにしたら、証明書とキーのペアを SSL 仮想サーバーにバインドして、SSL 処理に使用できるようにする必要があります。セキュリティで保護されたセッションでは、クライアントコンピューターとアプライアンス上の SSL ベースの仮想サーバーの間に接続を確立する必要があります。その後、仮想サーバーで着信トラフィックに対して SSL 処理が実行されます。したがって、アプライアンスで SSL 仮想サーバーを有効にする前に、有効な SSL 証明書を SSL 仮想サーバーにバインドする必要があります。

CLI を使用して、SSL 証明書キーペアを仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力し、SSL 証明書とキーのペアを仮想サーバーにバインドして構成を確認します。

```
-  bind ssl vserver <vServerName> -certkeyName <string>
-  show ssl vserver <name>
<!--NeedCopy-->
```

例:

```
> bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1

Done

> show ssl vserver Vserver-SSL-1



     Advanced SSL configuration for VServer Vserver-SSL-1:

     DH: DISABLED

     Ephemeral RSA: ENABLED Refresh Count: 0

     Session Reuse: ENABLED Timeout: 120 seconds

```



```
23
24      Cipher Redirect: ENABLED
25
26
27      SSLv2 Redirect: ENABLED
28
29
30      ClearText Port: 0
31
32
33      Client Auth: DISABLED
34
35
36      SSL Redirect: DISABLED
37
38
39      Non FIPS Ciphers: DISABLED
40
41
42      SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
43
44
45
46
47
48 1) CertKey Name: CertKey-SSL-1 Server Certificate
49
50
51 1) Cipher Name: DEFAULT
52
53
54    Description: Predefined Cipher Alias
55
56
57 Done
58 <!--NeedCopy-->
```

GUI を使用して、**SSL** 証明書キーペアを仮想サーバーにバインドします

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 証明書とキーのペアをバインドする仮想サーバー（たとえば、[Vserver-SSL-1]）を選択して、[Open] をクリックします。

3. [**Configure Virtual Server (SSL Offload)**] ダイアログボックスの [**SSL Settings**] タブにある [**Available**] で、仮想サーバーにバインドする証明書とキーのペアを選択します。次に、[追加] をクリックします。
4. [**OK**] をクリックします。
5. 選択した証明書キーのペアが [Configured] 領域に表示されていることを確認します。

Outlook Web Access に対するサポートの構成

Citrix ADC アプライアンスで Outlook Web Access（OWA）サーバーを使用している場合は、OWA サーバー宛の HTTP 要求に特別なヘッダーフィールド「FRONT-END-HTTPS: ON」を挿入するようにアプライアンスを構成して、URL リンクが「`http://`」ではなく「`https://`」として生成されるようにします。

注：OWA サポートは、HTTP ベースの SSL 仮想サーバーと SSL サービスで有効にできます。TCP ベースの SSL 仮想サーバーと SSL サービスでは OWA をサポートできません。

OWA サポートを構成するには、次の操作を実行します。

- OWA サポートを有効にする SSL アクションを作成します。
- SSL ポリシーを作成します。
- ポリシーを SSL 仮想サーバーにバインドします。

OWA サポートを有効にする **SSL** アクションを作成します

Outlook Web Access（OWA）サポートを有効にする前に、SSL アクションを作成する必要があります。SSL アクションは SSL ポリシーにバインドされ、着信データがポリシーで指定された規則と一致すると実行されます。

CLI を使用して **OWA** サポートを有効にする **SSL** アクションを作成します

コマンドプロンプトで次のコマンドを入力し、OWA サポートを有効にする SSL アクションを作成して構成を確認します。

```
1 -  add ssl action <name> -OWASupport ENABLED
2 -  show SSL action <name>
3 <!--NeedCopy-->
```

例：

```
1     > add ssl action Action-SSL-OWA -OWASupport enabled
2
3
4
```



```
5
6       Done
7
8
9     > show SSL action Action-SSL-OWA
10
11
12      Name: Action-SSL-OWA
13
14
15      Data Insertion Action: OWA
16
17
18      Support: ENABLED
19
20
21      Done
22  <!--NeedCopy-->
```

GUI を使用して **OWA** サポートを有効にする **SSL** アクションを作成します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。
2. 詳細ペインで、[**Actions**] タブ、[**Add**] をクリックします。
3. [**Create SSL Action**] ダイアログボックスの [Name] ボックスに、「Action-SSL-OWA」と入力します。
4. [Outlook Web Access] で、[**Enabled**] を選択します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。
6. [Action-SSL-OWA] が [**SSL Actions**] ページに表示されていることを確認します。

SSL ポリシーを作成する

SSL ポリシーは、ポリシーインフラストラクチャを使用して作成します。各 SSL ポリシーには SSL アクションがバインドされ、アクションは、着信トラフィックがポリシーで設定された規則と一致すると実行されます。

CLI を使用して **SSL** ポリシーを作成します

コマンドプロンプトで次のコマンドを入力し、SSL ポリシーを作成して構成を確認します。

```
1  -  add ssl policy <name> -rule <expression> -reqAction <string>
2  -  show ssl policy <name>
```



```
<!--NeedCopy-->
```

例:

```
> add ssl policy Policy-SSL-1 -rule ns_true -reqaction Action-SSL-OWA

Done

> show ssl policy Policy-SSL-1

Name: Policy-SSL-1 Rule: ns_true

Action: Action-SSL-OWA Hits: 0

Policy is bound to following entities

1) PRIORITY : 0

Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーを作成します

次の手順を実行します:

1. [**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**Create SSL Policy**] ダイアログボックスの [Name] ボックスに、SSL ポリシーの名前 (たとえば、「Policy-SSL-1」) を入力します。
4. [**Request** Action] で、このポリシーに関連付ける既存の SSL アクション (たとえば、[Action-SSL-OWA]) を選択します。ns_true 汎用式により、成功した SSL ハンドシェイクトラフィックのすべてにこのポリシーが適用されます。特定の応答に対してのみポリシーを適用する場合は、より高い詳細レベルのポリシーを作成できます。詳細なポリシー式の設定について詳しくは、「SSL アクションとポリシー」を参照してください。
5. [**Named Expressions**] で、組み込みの汎用式 ns_true を選択し、[**Add Expression**] をクリックします。式 ns_true が [Expression] ボックスに表示されます。
6. [**Create**] をクリックしてから、[**Close**] をクリックします。
7. ポリシーを選択して、ペイン下部にある [Details] セクションを表示し、ポリシーが正しく構成されていることを確認します。

SSL ポリシーを SSL 仮想サーバーにバインドします

Outlook Web Access に SSL ポリシーを設定したら、Outlook 着信トラフィックをインターセプトする仮想サーバーにポリシーをバインドします。着信データが SSL ポリシーで構成された規則と一致すると、そのポリシーに関連付けられたアクションが実行されます。

CLI を使用して SSL ポリシーを SSL 仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力し、SSL ポリシーを SSL 仮想サーバーにバインドして構成を確認します。

```
-  bind ssl vserver <vServerName> -policyName <string>
-  show ssl vserver <name>
<!--NeedCopy-->
```

例:

```
> bind ssl vserver Vserver-SSL-1 -policyName Policy-SSL-1

 Done

> show ssl vserver Vserver-SSL-1

Advanced SSL configuration for VServer Vserver-SSL-1:

DH: DISABLED

Ephemeral RSA: ENABLED

Refresh Count: 0

Session Reuse: ENABLED

Timeout: 120 seconds

Cipher Redirect: ENABLED

SSLv2 Redirect: ENABLED

ClearText Port: 0

Client Auth: DISABLED

```



```
SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) CertKey Name: CertKey-SSL-1 Server Certificate

1) Policy Name: Policy-SSL-1 Priority: 0

1) Cipher Name: DEFAULT Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーを **SSL** 仮想サーバーにバインドします

次の手順を実行します:

1. [**Traffic Management**] > [**SSL Offload**] > [**Virtual Servers**] の順に選択します。
2. 詳細ペインで仮想サーバー（たとえば、[Vserver-SSL-1]）を選択して、[**Open**] をクリックします。
3. [**Configure Virtual Server (SSL Offload)**] ダイアログボックスで [**Insert Policy**] をクリックし、SSL 仮想サーバーにバインドするポリシーを選択します。必要に応じて、[Priority] ボックスをダブルクリックして、新しい優先度を入力することもできます。
4. [**OK**] をクリックします。

一目でわかる機能

April 25, 2022

Citrix ADC の機能は、特定のニーズに対応するために、個別に構成することも、組み合わせて構成することもできます。一部の機能は複数のカテゴリに当てはまりますが、多数の Citrix ADC 機能は、一般に、アプリケーションスイッチングおよびトラフィック管理機能、アプリケーションアクセラレーション機能、アプリケーションセキュリティおよびファイアウォール機能、およびアプリケーション可視性機能として分類できます。

各機能を処理する順序については、「機能の処理順序」を参照してください。

アプリケーションスイッチングとトラフィック管理機能

April 25, 2022

以下は、アプリケーションスイッチングとトラフィック管理機能です。

SSL オフロード

Web サーバーから SSL 暗号化および解読を透過的にオフロードして、コンテンツ要求の処理用にサーバーのリソースを解放します。SSL はアプリケーションのパフォーマンスにとって大きな負担となり、多くの最適化方法が無効になることがあります。SSL オフロードおよび SSL アクセラレーションでは、Citrix Request Switching 技術のすべての利点を SSL トラフィックに適用できるため、エンドユーザーのパフォーマンスを低下させることなく、Web アプリケーションのセキュリティ保護されたデリバリーを実現できます。

詳しくは、「SSL オフロードおよびアクセラレーション」を参照してください。

アクセス制御リスト

着信パケットとアクセス制御リスト（ACL：Access Control List）を比較します。パケットが ACL 規則と一致した場合は、規則で指定されたアクションがパケットに適用されます。一致しない場合は、デフォルトアクション（ALLOW）が適用され、パケットは通常どおりに処理されます。アプライアンスが着信パケットと ACL を比較されるようにするには、ACL を適用する必要があります。すべての ACL はデフォルトで有効になっていますが、Citrix ADC アプライアンスが着信パケットを ACL と比較するためには、管理者が ACL を適用する必要があります。ルックアップテーブルに含める必要がなくても保持すべき ACL がある場合は、それを無効にしてから ACL を適用する必要があります。ADC アプライアンスは、着信パケットを無効な ACL と比較しません。

詳しくは、「アクセス制御リスト」を参照してください。

負荷分散

負荷分散の決定は、ラウンドロビン、最小接続数、加重最小帯域幅、加重最小パケット数、最小応答時間、および URL、ドメインソース IP、宛先 IP に基づくハッシュなど、さまざまなアルゴリズムに基づいて行われます。TCP と UDP プロトコルの両方がサポートされているので、Citrix ADC アプライアンスはこれらのプロトコルに基づくすべてのトラフィック（たとえば、HTTP、HTTPS、UDP、DNS、FTP、NNTP、および一般的なファイアウォールトラフィック）を負荷分散することができます。また、ADC アプライアンスは、ソース IP、Cookie、サーバー、グループ、または SSL セッションに基づくセッションパーシステンスを維持できます。サーバー、キャッシュ、ファイアウォール、およびそのほかのインフラストラクチャデバイスが正常に動作して適切なコンテンツがユーザーに提供されるように、カスタムの Extended Content Verification（ECV）を適用できます。また、ping、TCP、または HTTP URL を使用してヘルスチェックを実行したり、Perl スクリプトによるモニターを作成したりできます。
高度な WAN 最適化を提供するには、データセンターで展開されている CloudBridge アプライアンスを Citrix ADC アプライアンスで負荷分散することができます。これにより、帯域幅と同時セッションの数を大幅に改善できます。

詳しくは、「負荷分散」を参照してください。

トラフィックドメイン

トラフィックドメインを使用すると、単一の Citrix ADC アプライアンス内にいくつかの論理 ADC パーティションを作成できます。トラフィックドメインを使用すると、異なるアプリケーション用にネットワークトラフィックを分離できます。トラフィックドメインを使用すると、リソース間のやり取りが行われない分離環境を複数作成できます。特定のトラフィックドメインに属しているアプリケーションは、そのドメイン内のエンティティおよびプロセストラフィックとのみ通信します。あるトラフィックドメインに属しているトラフィックは、別のトラフィックドメインの境界を越えることはできません。そのため、アドレスが同じドメイン内で重複していない限り、アプライアンスで重複する IP アドレスを使用できます。

詳しくは、「トラフィックドメイン」を参照してください。

ネットワークアドレス変換

ネットワークアドレス変換（NAT）では、Citrix ADC アプライアンスを通過する IP パケットの送信元/宛先 IP アドレスや TCP/UDP ポート番号が変更されます。Citrix ADC アプライアンスで NAT を有効にすると、プライベートネットワークのセキュリティが強化されます。また、データが NetScaler を通過するときにプライベートネットワークの送信元 IP アドレスが変更されるため、インターネットなどのパブリックネットワークからプライベートネットワークが保護されます。

Citrix ADC アプライアンスでは、次の種類のネットワークアドレス変換がサポートされます。

INAT：受信 NAT（Inbound NAT：INAT）では、Citrix ADC アプライアンスで構成された IP アドレス（通常はパブリックアドレス）がサーバーの代わりに接続要求を待機します。アプライアンスがそのパブリック IP アドレスで要求パケットを受信した場合、Citrix ADC は、宛先 IP アドレスをサーバーのプライベート IP アドレスに置き換えます。つまり、アプライアンスはクライアントとサーバー間のプロキシとして機能します。INAT 構成には、Citrix ADC アプライアンスの IP アドレスとサーバーの IP アドレスの間の 1 対 1 の関係を定義する INAT 規則が含まれます。

RNAT：逆ネットワークアドレス変換（Reverse Network Address Translation：RNAT）では、サーバーによって開始されたセッションについて、Citrix ADC アプライアンスは、サーバーが生成したパケットの送信元 IP アドレスをアプライアンスで設定された IP アドレス（種類：SNIP）に置き換えます。これにより、サーバーが生成したパケットでサーバーの IP アドレスがさらされるのを防止します。RNAT 構成には、条件を指定する RNAT 規則が含まれます。アプライアンスは、条件に一致するパケットに対して RNAT 処理を実行します。

ステートレス **NAT46** 変換：ステートレス NAT46 は、セッション情報を Citrix ADC アプライアンスに保持せずに、IPv4 パケットと IPv6 パケットを相互に変換することにより、IPv4 ネットワークと IPv6 ネットワーク間の通信を実現します。ステートレス NAT46 構成には、IPv4-IPv6 INAT 規則と NAT46 IPv6 プレフィックスが含まれます。

ステートフル **NAT64** 変換：ステートフル NAT64 機能は、セッション情報を Citrix ADC アプライアンスに保持しながら、IPv6 パケットと IPv4 パケットを相互に変換することにより、IPv4 クライアントと IPv6 サーバー間の通信を実現します。ステートレス NAT64 構成には、NAT64 規則と NAT64 IPv6 プレフィックスが含まれます。

詳しくは、「ネットワークアドレス変換の構成」を参照してください。

マルチパス **TCP** のサポート

Citrix ADC アプライアンスは、マルチパス TCP（MPTCP）をサポートします。MPTCP は、ホスト間で使用可能な複数のパスを識別および使用して TCP セッションを保持する TCP/IP プロトコル拡張機能です。TCP プロファイルで MPTCP を有効にして仮想サーバーにバインドする必要があります。MPTCP が有効な場合、仮想サーバーは MPTCP ゲートウェイとして機能し、クライアントとの MPTCP 接続を、サーバーとの間で保持している TCP 接続に変換します。

詳しくは、「MPTCP（Multi-Path TCP）」を参照してください。

コンテンツスイッチ

ポリシーを切り替えるコンテンツの構成に基づいて要求を送信するサーバーを決定します。ポリシールールは、IP アドレス、URL、HTTP ヘッダーに基づいて設定できます。これにより、そのときのユーザー、使用されているエージェントの種類、ユーザーが要求したコンテンツなど、ユーザーとデバイスの特性に基づいて、スイッチを決定することができます。

詳しくは、「コンテンツスイッチ」を参照してください。

広域サーバー負荷分散（**Global Server Load Balancing**：**GSLB**）

NetScaler のトラフィック管理機能を拡張して、分散インターネットサイトとグローバル企業に対応します。設置場所が、複数のネットワークの場所や 1 箇所の複数のクラスターに分散していても、NetScaler は可用性を維持し、それらの間でトラフィックを分散します。インテリジェントな DNS 決定を行って、ダウンまたは過負荷状態のサイトにユーザーが割り当てられるのを防ぎます。近接ベースの GSLB 方式が有効な場合、NetScaler は、さまざまなサイトからクライアントのローカル DNS サーバー（LDNS）までの距離に基づいて、負荷分散の決定を行うことができます。距離ベースの GSLB 方式の最大の長所は、最も近い使用可能なサイトが選択されて、応答時間が短くなることです。

詳しくは、「グローバルサーバー負荷分散」を参照してください。

動的ルーティング

ルーターが、隣接するルーターからトポロジ情報、ルート、および IP アドレスを自動的に取得できるようにします。動的ルーティングが有効な場合、対応するルーティングプロセスはルート更新をリスンして、ルート情報を提供します。ルーティングプロセスはパッシブモードにすることもできます。ルーティングプロトコルを利用して、アップストリームルーターは Equal Cost Multipath 手法を使用し、2 台のスタンドアロン NetScaler 装置にホスティングされた同一の仮想サーバーに、トラフィックを負荷分散することができます。

詳しくは、「動的ルートの構成」を参照してください。

リンク負荷分散

複数の WAN リンクを負荷分散して、リンクフェールオーバーを提供し、ネットワークのパフォーマンスをさらに最適化して、ビジネスの継続性を保証します。インテリジェントなトラフィック制御とヘルスチェックを行って、アップストリームルーター間で効率的にトラフィックを分散することにより、ネットワーク接続の高い可用性を維持します。ポリシーとネットワーク状態に基づいて、着信トラフィックと送信トラフィックの両方をルーティングする最適な WAN リンクを特定し、高速な障害検出とフェールオーバーによって、WAN やインターネットリンク障害からアプリケーションを保護します。

詳しくは、「負荷分散のリンク」を参照してください。

TCP 最適化

TCP プロファイルを使用すると、TCP トラフィックを最適化できます。TCP プロファイルでは、NetScaler 仮想サーバーによる TCP トラフィックの処理方法を定義します。管理者は、組み込みの TCP プロファイルを使用するか、カスタムプロファイルを作成することができます。TCP プロファイルを定義した後、そのプロファイルを 1 つまたは複数の仮想サーバーにバインドできます。

TCP プロファイルで有効にできる主要な最適化機能のいくつかは次のとおりです。

- TCP Keep-Alive - リンクが切断されるのを防ぐために、指定された間隔で通信先の動作状態をチェックします。
- SACK（Selective Acknowledgment：選択的確認応答）- 特に LFN（Long Fat Network：広帯域高遅延ネットワーク）において伝送のパフォーマンスを向上させます。
- TCP ウィンドウスケーリング — LFN 経由の効率的なデータ転送を可能にします。

TCP プロファイルについて詳しくは、「TCP プロファイルの構成」を参照してください。

CloudBridge Connector

Citrix OpenCloud フレームワークの基本機能である Citrix NetScaler CloudBridge Connector は、クラウド拡張型のデータセンターの構築に使用されるツールです。OpenCloud Bridge により、ネットワークを再構成することなく、クラウド上の 1 つまたは複数の Citrix ADC アプライアンスまたは NetScaler 仮想アプライアンスをネットワークに接続することができます。クラウドがホストするアプリケーションは、組織内の単一ネットワーク上で実行されているかのように動作します。OpenCloud Bridge の主な目的は、企業がアプリケーションをクラウドに移行しながら、コストやアプライアンス障害のリスクを削減できるようにすることにあります。また、OpenCloud Bridge は、クラウド環境のネットワークセキュリティを向上します。OpenCloud Bridge は、クラウドインスタンス上の Citrix ADC アプライアンスまたは NetScaler 仮想アプライアンスを LAN 上の Citrix ADC アプライアンスまたは NetScaler 仮想アプライアンスに接続するレイヤー 2 ネットワークブリッジです。接続は、GRE（Generic Routing Encapsulation）プロトコルを使用するトンネルを介して確立されます。GRE プロトコルは、さまざまなネットワークプロトコルからのパケットをカプセル化し、別のプロトコル経由で転送するメカニズムを提供しています。IPSec（Internet Protocol Security：インターネットプロトコルセキュリティ）プロトコルは、OpenCloud Bridge のピア間の通信を確保します。

詳しくは、「CloudBridge」を参照してください。

DataStream

NetScaler DataStream 機能は、送信中の SQL クエリに基づいて要求を分散することで、データベース層で要求の割り振りを実行するインテリジェントなメカニズムを提供します。

データベースサーバーの前に NetScaler を導入すれば、アプリケーションサーバーまたは Web サーバーからのトラフィックを最適に分散することができます。管理者は、SQL クエリの情報と、データベース名、ユーザー名、文字セット、およびパケットサイズに基づいて、トラフィックをセグメント化できます。

負荷分散を構成して、負荷分散アルゴリズムに基づいて要求を割り振ることができます。または、ユーザー名、データベース名、コマンドパラメーターなどの SQL クエリパラメーターに基づくコンテンツスイッチを構成して、スイッチ条件を詳細に設定できます。さらに、モニターを構成してデータベースサーバーの状態を監視することもできます。

Citrix ADC アプライアンスの高度なポリシーインフラストラクチャには、要求の評価と処理に使用できる式が含まれています。高度な式により、MySQL データベースサーバーに関連付けられたトラフィックが評価されます。高度なポリシーの要求ベースの式（MYSQL.CLIENT および MYSQL.REQ で始まる式）を使用すると、コンテンツスイッチ仮想サーバーのバインドポイントで要求スイッチの意思決定を行うことが可能になり、応答ベースの式（MYSQL.RES で始まる式）を使用すると、ユーザー設定のヘルスモニターへのサーバー応答を評価することができます。

> 注：DataStream は、MySQL と MS SQL のデータベースでサポートされています。

詳しくは、「DataStream」を参照してください。

アプリケーションの速度向上機能

April 25, 2022

- AppCompress

  gzip 圧縮プロトコルを使った HTML とテキストファイルに対する透過的圧縮一般的な 4：1 の圧縮率で、データセンターの帯域幅要件が、最大 50% 削減されます。また、ユーザーのブラウザーに渡す必要があるデータ量が削減されるため、エンドユーザーの応答時間が大幅に短縮されます。

- キャッシュリダイレクト

  リバースプロキシ、透過プロキシ、またはフォワードプロキシのキャッシュファームに対するトラフィックのフローを管理します。すべての要求を検査して、キャッシュ不能な要求を特定し、固定接続を介してそれらを発信元のサーバーに直接送信します。キャッシュ不能な要求を発信元の Web サーバーへインテリジェントにリダイレクトすることによって、Citrix ADC アプライアンスはキャッシュリソースを解放し、キャッシュヒット率を上げながら、これらの要求に対する全体的な帯域幅消費と応答遅延を削減します。

  詳しくは、「キャッシュリダイレクト」を参照してください。

- AppCache

  静的および動的コンテンツの両方に対して、高速インメモリ HTTP/1.1 および HTTP/1.0 準拠の Web キャッシュを提供し、Web コンテンツとアプリケーションデータデリバリーを最適化します。このオンボードキャッシュは、着信要求がセキュリティで保護されたり、データが圧縮されたりしている場合でも、着信アプリケーション要求の結果を保存し、データを再利用して、同じ情報に対する今後の要求に対応します。オンボードキャッシュから直接データを提供することによって、静的および動的コンテンツ要求をサーバーに送信する必要がなくなるので、アプライアンスはページの再生成時間を削減できます。

  詳しくは、「統合キャッシュ」を参照してください。

- TCP バッファリング

  サーバーの応答をバッファリングして、そのクライアントの速度でクライアントに応答を提供し、より高速にサーバーをオフロードするので、Web サイトのパフォーマンスが向上します。

アプリケーションセキュリティとファイアウォール機能

September 27, 2022

以下は、セキュリティとファイアウォールの機能です。

コンテンツフィルタリング

レイヤー 7 レベルで、Web サイトを悪意のある攻撃から保護します。アプライアンスは、HTTP ヘッダーに基づくユーザー構成の規則に従って、各着信要求を検査し、ユーザーが構成したアクションを実行します。アクションとして、接続の再設定、要求のドロップ、ユーザーのブラウザーへのエラーメッセージの送信などを設定できます。これにより、不要な要求を除去して、サーバーが攻撃にさらされる危険性を減らすことができます。

この機能は、HTTP GET と POST 要求を分析して、既知の不正なシグニチャを排除できるので、HTTP ベースの攻撃からサーバーを防御できます。

詳しくは、「コンテンツフィルタリング」を参照してください。

レスポンダー

高度なフィルタリングのように機能し、アプライアンスからクライアントへの応答を生成するために使用できます。この機能の一般的な用途は、リダイレクト応答、ユーザー定義応答、およびリセットの生成です。

詳しくは、「レスポンダー」を参照してください。

リライト

HTTP ヘッダーと本文のテキストを変更します。再書き込み機能を使用して、HTTP 要求または応答に HTTP ヘッダーを追加したり、個別の HTTP ヘッダーを変更したり、HTTP ヘッダーを削除したりできます。また、要求と応答の HTTP ボディを変更することもできます。

アプライアンスは、要求を受信したり応答を送信したりするときに書き換え規則をチェックして、適切な規則を要求や応答に適用してから Web サーバーまたはクライアントコンピューターに渡します。

詳しくは、「書き換え」を参照してください。

優先度によるキューイング

ユーザー要求に優先度を付けて、要求ボリュームのサージ中に、最も重要なトラフィックが最初に処理されるようにします。要求 URL、Cookie、またはその他のさまざまな要因に基づいて、優先度を設定できます。アプライアンスは、構成された優先度に基づいて 3 層のキューに要求を入れて、サージ中やサイト攻撃中でも、ビジネスクリティカルなトランザクションをスムーズに処理できるようにします。

詳しくは、「優先度によるキューイング」を参照してください。

サージ保護

サーバーへのユーザー要求のフローを調整し、サーバー上のリソースへ同時にアクセスできるユーザー数を制御して、サーバーの容量に達した場合には、追加の要求をキューに入れます。接続を確立できるレートを制御することによって、アプライアンスは、サーバーに渡される要求のサージをブロックし、サイトがオーバーロード状態になるのを防ぎます。

詳しくは、「サージ保護」を参照してください。

Citrix Gateway

Citrix Gateway はアプリケーションアクセスのセキュリティを保護するソリューションで、詳細なアプリケーションレベルのポリシーと操作の制御機能を管理者に提供し、ユーザーがどこにいても作業できるようにすると同時に、アプリケーションとデータへのアクセスのセキュリティを保護します。IT 管理者は一拠点からツールを使用して、企業内外の規制順守および高度な情報セキュリティの確保を支援できます。それと同時に、ユーザーは役割、デバイス、およびネットワークに応じて最適化された単一のアクセスポイントを経由して、必要なエンタープライズアプリケーションとデータを使用できます。この 2 つの機能のユニークな組み合わせによって、今日のモバイルワーカーの生産性を最大限に向上させることができます。

詳しくは、「Ctirix Gateway」を参照してください。

アプリケーションファイアウォール

保護された各 Web サーバーと、その Web サーバー上の Web サイトに接続するユーザー間のトラフィックをフィルター処理して、クロスサイトスクリプティング攻撃、バッファーオーバーフロー攻撃、SQL インジェクション攻撃、強制的ブラウズなど、ハッカーやマルウェアによる悪用からアプリケーションを保護します。アプリケーションファイアウォールは、Web サーバーセキュリティに対する攻撃や、Web サーバーリソースの悪用の形跡がないかどうか、すべてのトラフィックを調べて適切なアクションを実行し、これらの攻撃を未然に防ぎます。

詳しくは、「アプリケーションファイアウォール」を参照してください。

アプリケーションの可視性機能

April 25, 2022

- Citrix Application Delivery Management

  Citrix Application Delivery Management（ADM）は、Web および HDX（ICA）トラフィック全体でエンドツーエンドのユーザーエクスペリエンスの可視性を提供する高性能コレクターです。Citrix ADC アプライアンスによって生成された HTTP および ICA AppFlow レコードを収集し、レイヤー 3 からレイヤー 7 の統計情報をカバーする分析レポートを作成します。Citrix ADM は、直前 5 分間のリアルタイムデータおよび直前 1 時間、1 日間、1 週間、1 か月間について収集された履歴データを詳細に分析します。

  HDX（ICA）分析ダッシュボードでは、HDX ユーザー、アプリケーション、デスクトップ、およびゲートウェイレベルの情報をドリルダウンできます。同様に、HTTP 分析では、Web アプリケーション、アクセスされた URL、クライアント IP アドレス、サーバー IP アドレス、およびそのほかのダッシュボードの概観を表示します。管理者は、ユースケースに合わせて、これらのダッシュボードからドリルダウンして問題点を明らかにできます。

- AppFlow を使用した拡張されたアプリケーションの可視性

  Citrix ADC アプライアンスは、データセンター内のすべてのアプリケーショントラフィックを一元的に制御します。これは、アプリケーションパフォーマンスの監視、分析、およびビジネスインテリジェンスアプリケーションにとって有効なフローとユーザーセッションレベルの情報を収集します。AppFlow は、RFC 5101 で定義されたオープンな IETF（Internet Engineering Task Force：インターネット技術標準化委員会）標準である IPFIX（Internet Protocol Flow Information eXport）形式を使用して、この情報を送信します。IPFIX（Cisco 社製 NetFlow の標準化バージョン）は、ネットワークフロー情報を監視するために幅広く使用されています。AppFlow は、新しい情報要素を定義してアプリケーションレベルの情報を表現します。

  トランスポートプロトコルとして UDP を使用して、AppFlow は フローレコードと呼ばれる収集されたデータを 1 つまたは複数の IPv4 コレクターに送信します。コレクターはフローレコードを集約し、リアルタイムレポートまたは履歴レポートを生成します。

  AppFlow は、HTTP、SSL、TCP、および SSL_TCP フローのトランザクションレベルでの可視性を実現します。監視対象のフロータイプのサンプリングとフィルタリングを行うことが可能です。

アプリケーショントラフィックのサンプリングとフィルタリングを行うことで監視するフロータイプを制限する場合は、AppFlow を仮想サーバー向けに有効化できます。AppFlow では、仮想サーバーの統計情報も提供しています。

また、AppFlow を特定のサービス向けに有効化してアプリケーションサーバーを表現し、そのアプリケーションサーバーへのトラフィックを監視することもできます。

詳しくは、「AppFlow」を参照してください。

- ストリーム分析

Web サイトやアプリケーションのパフォーマンスは、最も頻繁に要求されるコンテンツの配信をどのように最適化するかにより決まります。キャッシュや圧縮などの方法は、クライアントへのサービス配信の高速化に役立ちますが、最も頻繁に要求されるリソースを特定し、それらのリソースをキャッシュまたは圧縮できるようにする必要があります。Web サイトやアプリケーショントラフィックに関するリアルタイム統計を集計すれば、最も頻繁に使用されるリソースを特定できます。リソースごとのアクセス頻度や消費帯域幅などの統計によって、サーバーパフォーマンスとネットワーク使用率を改善するために、それらのリソースをキャッシュまたは圧縮する必要があるかどうかを判断できます。応答時間やアプリケーションへの同時接続数などの統計は、サーバー側のリソースを強化する必要があるかどうかを判断するのに役立ちます。

Web サイトやアプリケーションが頻繁に更新されない場合、統計データを収集する製品を使用して、その統計を手動で分析し、コンテンツの配信を最適化できます。ただし、最適化を手動で行わない場合や、Web サイトまたはアプリケーションのコンテンツが動的に生成される場合、統計データを収集するだけでなく、その統計に基づいてリソースの配信を自動的に最適化できるインフラストラクチャが必要です。Citrix ADC アプライアンスでは、この機能はストリーム分析機能によって提供されます。この機能は単一の Citrix ADC アプライアンス上で実行され、定義した条件に従ってリアルタイム統計を収集します。Citrix ADC ポリシーと共に使用すると、この機能によって自動的なリアルタイムトラフィックの最適化に必要なインフラストラクチャも提供されます。

詳しくは、「アクション分析」を参照してください。

Citrix ADC ソリューション

October 7, 2021

Citrix ADC ソリューションを使用すると、頻繁に展開される構成を簡単にセットアップできます。追加の解決策については、このスペースを随時確認してください。

このセクションでは、次のソリューションについて説明します。

- Citrix Virtual Apps and Desktops の Citrix ADC の設定
- グローバルサーバ負荷分散（GSLB）によるゾーンプリファレンス
- Citrix ADC におけるエニーキャストのサポート
- Citrix ADC を使用して AWS にデジタル広告プラットフォームをデプロイする

- Citrix ADC を使用した AWS でのクリックストリーム分析の強化
- Microsoft Windows Azure パックと Cisco ACI によって管理されるプライベートクラウドでの Citrix ADC

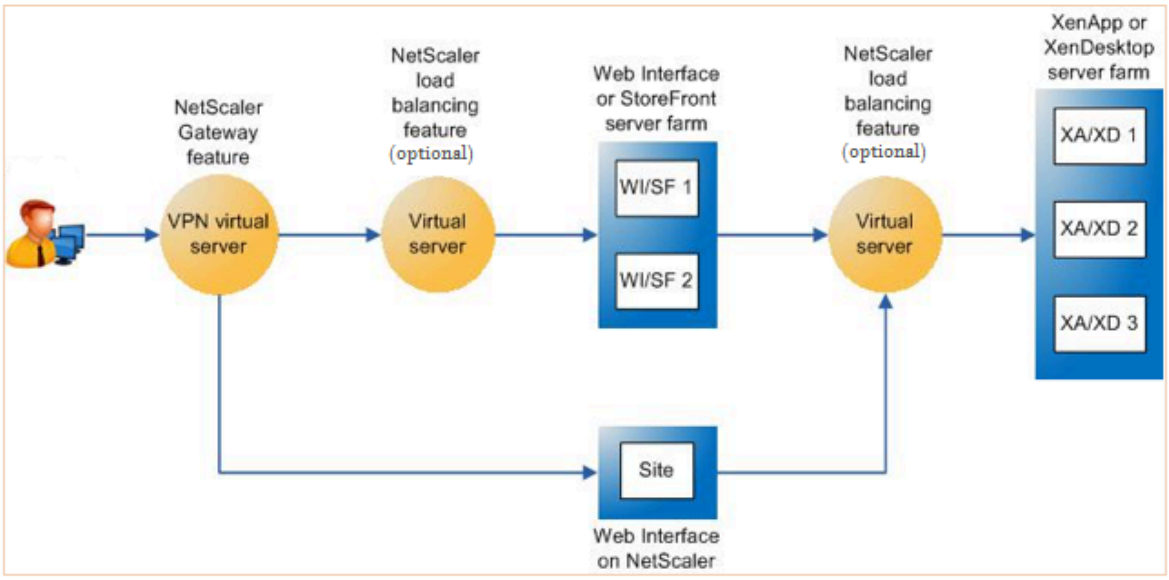
Citrix Virtual Apps and Desktops の Citrix ADC の設定

October 7, 2021

Citrix ADC アプライアンスは、Citrix Virtual Apps and Desktops アプリケーションへの負荷分散された安全なリモートアクセスを提供します。Citrix ADC 負荷分散機能を使用すると、Citrix Virtual Apps and Desktops サーバーにトラフィックを分散し、Citrix Gateway 機能を使用してサーバーへの安全なリモートアクセスを提供できます。Citrix Gateway 機能を使用して、サーバーへの安全なリモートアクセスを提供できます。

また、Citrix ADC は、トラフィックフローを加速して最適化し、Citrix Virtual Apps and Desktops の展開に役立つ可視性機能を提供します。

図 1: Citrix Virtual Apps and Desktops のセットアップでの Citrix ADC アプライアンス



上の図は、この展開に関係するコンポーネントを示しています。

- で接続する必要があります。ユーザー・アクセス用の URL を提供し、ユーザーを認証してセキュリティを提供します。
- **Citrix ADC** 負荷分散仮想サーバー。Web インターフェイスまたは StoreFront サーバーのトラフィックを負荷分散します。また、Citrix Virtual Apps およびデスクトップサーバーの前に負荷分散仮想サーバーを展

開して、XML ブローカーやデスクトップ Delivery Controller（DDC）サーバーなどの主要コンポーネントの負荷分散を行うこともできます。

- **Citrix ADC** 上の **Web** インターフェイスまたは **StoreFront** または **Web** インターフェイス。アプリケーションにアクセスするためのインターフェイスを提供します。

  注：Citrix ADC（WIonNS）上の Web インターフェイスは、Citrix ADC アプライアンスでホストされている Web インターフェイス製品のカスタマイズです。

- **Citrix Virtual Apps and Desktops**。ユーザーがアクセスするアプリケーションを提供します。

Citrix ADC GUI を使用して Citrix Virtual Apps and Desktops 用の Citrix ADC をセットアップするには

前提条件

- Citrix Virtual Apps デスクトップサーバーが構成され、利用可能になります。
- Citrix ADC サーバー上の Web インターフェイス、StoreFront、または Web インターフェイスが構成され、使用可能になります。
- あなたは、Citrix Gateway、Citrix ADC、Citrix Virtual Apps and Desktops、および Citrix ADC 上の StoreFront/Web Interface/Web Interface に関する実用的な知識を持っています。
- 仮想サーバーとサービスを構成し、サービスを仮想サーバーにバインドしていることを確認します。詳しくは、次のトピックを参照してください：
  - XenDesktop の負荷分散
  - XenApp の負荷分散

手順:

1. Citrix ADC アプライアンスにログオンし、［構成］タブで［**XenApp** および **XenDesktop** ］をクリックします。
2. [詳細] ウィンドウで、[開始] をクリックします。Citrix ADC 上にセットアップが存在する場合は、変更する各セクションに対応する「編集」リンクをクリックします。
3. 展開環境で、Citrix Virtual Apps and Desktops アプリケーションにアクセスするためのインターフェイスを提供する製品（StoreFront、Web インターフェイス、または Citrix ADC 上の Web インターフェイス）を選択します。
4. セキュアなリモートアクセスをセットアップします。
   a)「**NetScaler Gateway** の設定」セクションで、VPN 仮想サーバーの詳細を指定し、「続行」をクリックします。
   b) [サーバー証明書] セクションで、既存の証明書を選択するか、新しい証明書をインストールして、[続行] をクリックします。
   c) [認証] セクションで、使用するプライマリ認証メカニズムを構成し、サーバーの詳細を指定するか、既存のサーバーを使用して [続行] をクリックします。

d) **StoreFront** セクションで、アプリケーションにアクセスするためのインターフェイスを提供するサーバーの詳細を指定し、「続行」をクリックします。

e) StoreFront サーバーとして、次のいずれかを使用できます。

i. 複数の SF サーバを指している LB 仮想サーバ。
ii. Citrix ADC アプライアンスから直接アクセス可能な Web インターフェイスまたは StoreFront サーバー。
iii. Citrix ADC 上の Web インターフェイス。

5. [完了] をクリックして設定を完了します。

グローバルサーバ負荷分散（**GSLB**）によるゾーンプリファレンス

October 7, 2021

GSLB 電源ゾーンプリファレンスは、Citrix Virtual Apps and Desktops、StoreFront、および Citrix ADC を統合し、クライアントの場所に基づいて最も最適化されたデータセンターにクライアントがアクセスできるようにする機能です。

分散型 CitrixVirtual Apps and Desktops 展開では、複数の同等のリソースが複数のデータセンターから利用できる場合、StoreFront は最適なデータセンターを選択しない可能性があります。このような場合、StoreFront はランダムにデータセンターを選択します。要求を行うクライアントとの近さに関係なく、任意のデータセンター内の任意の Citrix Virtual Apps and Desktops サーバーに要求を送信できます。

HTTP 要求が CitrixGateway アプライアンスに到着すると、クライアントの IP アドレスが調べられます。実際のクライアント IP アドレスは、StoreFront に転送されるデータセンター設定リストを作成するために使用されます。Citrix ADC アプライアンスがゾーン優先ヘッダーを挿入するように構成されている場合、StoreFront 3.5 以降では、アプライアンスから提供された情報を使用して、配信コントローラのリストを並べ替え、クライアントと同じゾーン内の最適な配信 Controller に接続できます。StoreFront は、選択したデータセンターゾーンに最適なゲートウェイ VPN 仮想サーバーを選択し、この情報を適切な IP アドレスを使用して ICA ファイルに追加し、クライアントに送信します。次に、Storefront は、優先データセンターの配信コントローラーでホストされているアプリケーションを起動してから、他のデータセンターにある同等のコントローラーに接続しようとします。

このソリューションの構成の詳細については、ここをクリックしてください。

GSLB 電源ゾーン設定ソリューションに関するビデオ概要については、https://www.youtube.com/watch?v=Y8DELum0Xp0をクリックしてください。

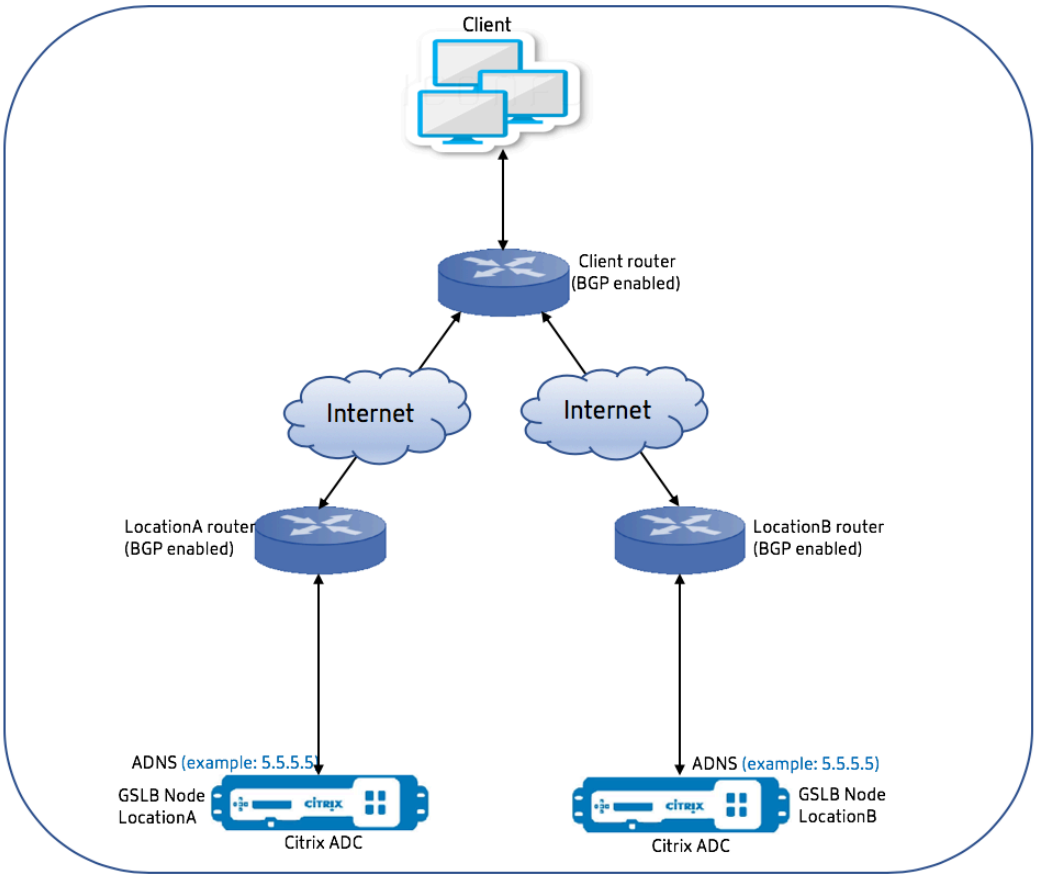
Citrix ADC におけるエニーキャストのサポート

October 7, 2021

エニーキャストは、サーバーのセットが IP アドレスを共有するタイプのネットワークです。クライアント要求は、ルーティングテーブルに基づいて、地形的に最も近いサーバーに送信されます。このルーティングにより、遅延の問題が軽減され、高可用性が確保され、ダウンタイムが最小限に抑えられます。

Citrix ADC は、グローバルサーバー負荷分散（GSLB）と DNS 機能を備えたエニーキャストネットワークをサポートします。

次の図は、Citrix ADC におけるエニーキャストのトポロジ図を示しています。



Anycast GSLB

Citrix ADC GSLB 機能は、ディザスタリカバリとともにグローバルに分散したサイト間の負荷分散を提供し、アプリケーションの継続的な可用性を保証します。

停止中、GSLB は、最も近いデータセンターまたは最もパフォーマンスの高いデータセンターにトラフィックをルーティングすることにより、即時の災害復旧を提供します。ただし、GSLB は以下を制御できません。

- DNS トラフィックが地理的に異なる場所にある GSLB ノードにルーティングされる方法。
- DNS クエリが GSLB ノードにルーティングされる間に、どのくらいのレイテンシーが追加されるか。

一般的な GSLB セットアップでは、各データセンターに、DNS クエリを受信するために、サイト固有の権限ドメインネームサーバー (ADNS) で構成された GSLB ノードがあります。各サイトの ADNS は、DNS リゾルバーのネーム

サーバーとして構成されます。GSLB ノードの数が増加すると、ネームサーバーレコードの数も増加します。このような場合、データセンターに障害が発生した場合、LDNS は別のネームサーバーで解決を再試行する必要があります。この再試行により、DNS 解決の遅延が増加します。
また、GSLB ノードが追加されるたびに、ネームサーバーレコードを更新する必要があります。

これらの欠点を克服するために、Anycast ADNS を使用することができます。エニーキャスト ADNS では、すべての GSLB ノードに単一の ADNS IP アドレスが使用され、DNS トラフィックは動的ルーティングを使用して GSLB ノードにルーティングされます。

たとえば、GSLB サイトがダウンしている場合、ルーティングテーブルが更新され、このサイトへのルートが削除されます。その結果、DNS クエリはダウンしているサイトに送信されません。その結果、再試行は行われません。

新しい GSLB ノードが追加されると、新しいノードには同じ ADNS IP アドレスが割り当てられます。動的ルーティングは、ルーティングアルゴリズムに基づいて、新しいサイトへのルートでルーティングテーブルを自動的に更新します。したがって、DNS ネームサーバーレコードを更新する必要はありません。新しい GSLB サイトの展開は、Anycast でより簡単かつ迅速に行われます。

エニーキャストモードで **ADNS IP** アドレスを設定する方法

Citrix ADC アプライアンスの ADNS IP でホストルーティングを有効にし、適切なルートヘルスインジェクション（RHI）レベルを設定します。ほとんどの場合、ADNS IP 上には仮想サーバーがないため、RHI レベルを NONE として選択する必要があります。ADNS IP 上でホストルートを有効にすると、カーネルルートになります。次に、選択の動的ルーティングを有効にし、カーネルルートを再配布するようにルーティングプロトコルを構成できます。

ADNS IP の設定：例

コマンドプロンプトで次を入力します。

```
add service adns_public 5.5.5.5 ADNS 53

set ip 5.5.5.5 -hostRoute ENABLED -vserverRHILevel ALL_VSERVERS
<!--NeedCopy-->
```

GSLB サイトの **BGP** 設定：例

```
Site1#sh run
!
hostname Site1
!
log syslog
```



```
log record-priority
!
ns route-install bgp
!
interface lo0
 ip address 127.0.0.1/8
 ipv6 address fe80::1/64
 ipv6 address ::1/128
!
interface vlan0
 ip address 10.102.148.94/25
 ipv6 address fe80::e84c:f4ff:fe74:4588/64
!
interface vlan2
 ip address 172.18.30.15/24
!
router bgp 5
 redistribute kernel -----> redistributing the kernel routes
 neighbor 172.18.30.30 remote-as 4
 neighbor 172.18.30.30 advertisement-interval 1
 neighbor 172.18.30.30 timers 4 16
!
End

Site1#
<!--NeedCopy-->
```

GSLB サイトルーティングテーブル：例

```
Site1#sh ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, I - Intranet
       * - candidate default

K       5.5.5.5/32 via 0.0.0.0 ---------------------------------------->
    Kernel Route for ADNS
C       10.102.148.0/25 is directly connected, vlan0
C       127.0.0.0/8 is directly connected, lo0
B       172.18.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
```



```
B       172.18.20.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
C       172.18.30.0/24 is directly connected, vlan2
B       192.168.3.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.5.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h

Gateway of last resort is not set
Site1#
<!--NeedCopy-->
```

Anycast DNS

Citrix ADC 上の DNS プロキシ仮想サーバーにエニーキャスト DNS を使用できます。複数の DNS ネームサーバが設定されている場合、DNS リゾルバはラウンドロビン方式に基づいて応答します。たとえば、リゾルバーが最初のサーバーから応答を受信しない場合、構成されたタイムアウト値の期限が切れると、リゾルバーは 2 番目のサーバーに切り替わります。1 台目のサーバから 2 台目のサーバへの切り替えは、DNS 解決のレイテンシを増やします。DNS リゾルバがエニーキャストで設定されている場合、この遅延は解消できます。

DNS の設定：例

コマンドプロンプトで次を入力します。

```
add lb vserver dns DNS 5.5.5.50 53

set ip 5.5.5.50 -hostRoute ENABLED -vserverRHILevel ALL_VSERVERS
<!--NeedCopy-->
```

Citrix ADC を使用して AWS にデジタル広告プラットフォームをデプロイする

October 7, 2021

デジタルプラットフォームの進化する性質により、幅広い広告アプリケーションが利用可能です。たとえば、ソーシャルメディア、ダイレクトメール、ビデオ、バナー、ポップ、インタースティシャル、リッチメディアなどです。広告主は速いペースでビデオ広告ネットワークを採用しており、広告トラフィックのほぼ 40%を占めています。しかし、現代のユーザーによるモバイルの使用が増えるにつれ、モバイルプラットフォームでの動画広告の掲載が大幅に増加しています。

デジタル広告プラットフォームはいくつかの課題に直面しています。いくつかの課題は次のとおりです。

- セキュリティの脅威
- 高い運用コスト
- インターネットを介してトラフィックを送信するために、さまざまなデバイスを利用できます。リアルタイム通信のさまざまなプロトコルには、次の課題があります。
  - webRTC
  - アダプティブストリーミング
  - WebRTC が UDPoverHTTP を使用するビデオ用 UDP

広告プラットフォームの複雑な動作に対処するために、AWS と十分に統合された機能と機能のスイート全体を備えた Citrix ADC ソリューションは、いつでもどこでもデジタル広告インベントリへの即時、安全、かつ信頼性の高いアクセスを提供します。Citrix ADC は、デジタルプラットフォーム向けの SaaS および Web アプリを提供する上で重要な役割を果たします。

Citrix ADC とのデジタル広告プラットフォームの統合

デジタル広告プラットフォームの概要

デジタル広告プラットフォームは、次の主要コンポーネントで構成されています。

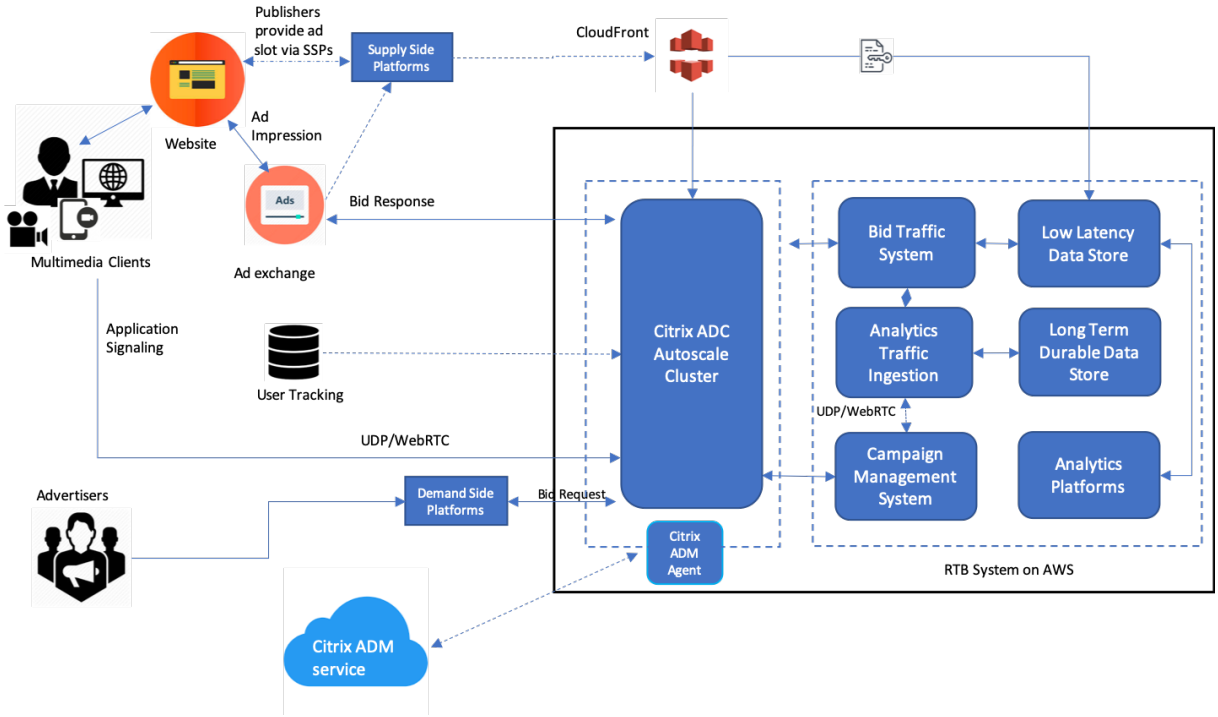
- Ad exchange
- Ad network
- デマンドサイドプラットフォーム（DSP）
- サプライサイドプラットフォーム（SSP）
- リアルタイムビッダー（RTB）システム

広告システムで行われるプロセスの概要は次のとおりです。

- 最初のトランザクションは、ユーザーが Web サイトにアクセスしたときに発生します。
- これにより、bid/advertisement アドエクスチェンジに連絡する広告サーバーまたはサイト運営者に送信されるリクエスト（ユーザーの人口統計情報を含む）。
- 広告パブリッシャーは、SSP を介してアドエクスチェンジに広告リクエストを送信します。
- Ad Exchange は、このリクエストとそれに付随するデータを DSP に送信して、インプレッションまたは広告リクエストが利用可能であることを通知します。したがって、複数の広告主がリアルタイムで自動的に入札を送信して、広告を掲載できます。
- 一方、広告主は DSP でキャンペーンを設定する必要があります。データ管理プラットフォーム（DMP）からのユーザーに関する情報を使用して、ユーザーに広告を配信するために支払う意思のある金額を評価します。
- DSP は、アドエクスチェンジに配信されるため、各広告インプレッションに対してこれらのリアルタイム入札を送信します。
- Ad Exchange または SSP によって設定された期間内に最も多く入札した入札者は、サイト運営者が広告を配信するための広告スロットを取得します。そうでなければ、彼らは彼らの主要な人口統計のための正しい広告を得る機会を失います。

デジタル広告プラットフォームが **Citrix ADC** とどのように統合されているか

次の図は、広告プラットフォームのさまざまなコンポーネントが Citrix ADC および CitrixApplication Delivery Management（ADM）と通信してオンライン広告を提供する方法を示しています。



Citrix ADC がどのように貢献するか

広告公開プロセスでは、Citrix ADC ソリューションは、一貫性のない入札トラフィックの流入を処理および処理するのに役立ちます。これは、すべてのトラフィックのエントリポイントとして機能し、アベイラビリティーゾーン全体のスケーラビリティーと可用性を確保します。広告トラフィックの弾力性に対応するために、Web アプリケーションとデータベースサーバーの前にある自動スケーリンググループに展開されます。

AWS の Citrix ADC ソリューションを使用した広告プラットフォームを使用すると、世界中でリアルタイムのパフォーマンス、高いスケーラビリティ、および高可用性を実現できます。リッチメディア、ビデオ、モバイル、ネイティブ広告をリアルタイムで売買できます。これにより、広告プラットフォームの実行に伴う全体的な運用コストと遅延が削減されます。これは、オートスケール、接続の多重化中にバックエンドサーバーを適切に削除し、エンドユーザーのトラフィックが影響を受けないようにする豊富な機能を備えた最高のパフォーマンスのプロキシです。Citrix ADC は、広告プラットフォームで使用される HTTP、UDP、WebRTC、および RTSP プロトコルの負荷分散をサポートしています。

Citrix ADC は、次の主要な属性を使用して AWS 環境に一貫して適合します。

- コンテンツの切り替え–ホスト名に基づいて適切なプラットフォームに切り替えます。
- セキュリティ保護– Web アプリケーションファイアウォール（WAF）機能、レート制限（クライアント IP を介して）、および DDoS 攻撃に対する保護を使用します。

- フロントエンドトラフィックとバックエンドトラフィックの両方の自動スケーリング。
- エンドツーエンドの可視性、および ADM を利用した ADC アプライアンス全体の異常検出。
- 低遅延。

CitrixADM の貢献方法

Citrix ADC は、Citrix ADM を利用して、デジタル広告プラットフォームが直面する次の課題を克服します。

- 期待されるパフォーマンスからの傾向の逸脱を特定する
- リアルタイムのアプリケーションパフォーマンス分析
- 容量監視

Citrix ADC および **ADM** との広告プラットフォーム統合の利点

Citrix ADC ソリューションは、デジタル広告プラットフォームベンダーに次の機能と利点を提供します。

低価格

- AWSAutoscaling サービスと統合された Citrix ADC VPX インスタンスは、フロントエンドとバックエンドのリソースを自動的にスケールアップまたはスケールダウンできます。これにより、広告プラットフォームの弾力性に対応するゼロタッチ構成が提供されます。
- 単一のポイントからすべてのタイプのトラフィックを配信する統合。

AWS 自動スケーリングの詳細については、「バックエンド AWS 自動スケーリングサービスの追加」を参照してください。

高可用性

- 1 つのアベイラビリティーゾーンが使用できなくなった場合、Citrix ADC はフォールトトレランス機能を適用して、トラフィックを中断することなく、別のアベイラビリティーゾーンのサーバーを自動検出します。
- また、クライアント接続の損失を回避してサーバーを正常に終了します。

詳細については、「AWS での高可用性の仕組み」を参照してください。

アプリケーション・パフォーマンス分析

Citrix ADM インテリジェント分析およびアプリケーションパフォーマンス分析により、次のことが保証されます。

- エンドユーザーエクスペリエンスを悩ませている問題（サーバー応答の異常、5XX エラーなど）を可視化します。
- 管理者に警告して、すぐに修正措置を講じてください。

詳細については、「アプリケーション分析のパフォーマンス指標」を参照してください。

ファイアウォールのセキュリティ

最も一般的なセキュリティの脆弱性は、ネットワークではなく Web アプリケーションで発生します。ボット、データの盗難、アプリケーション層の攻撃などの不正アクセスから Web アプリケーションを保護することが重要です。

Citrix ADC は、以下を含む包括的で統合されたレイヤー 4 からレイヤー 7 のセキュリティを提供します。

- Web App Firewall（WAF）は、定期的に更新されるボットシグネチャと動作ベースの検出により、Web アプリケーションを保護し、悪意のあるボットを識別して軽減します。
- 広告プラットフォームが圧倒されるのを防ぐためのレート制限。

詳細については、Citrix Web App Firewallを参照してください。

広告プラットフォームに適した **AWS** インスタンスタイプを選択します

次の 2 つの要因に応じて、ADC に適切な AWS インスタンスタイプを選択します。

- 広告プラットフォームに同時にアクセスするユーザーの数。
- プラットフォーム上の平均ユーザー数。

Citrix ADC は、c5、c5n、m5 などを含むさまざまな EC2 インスタンスにデプロイできます。広告プラットフォームの場合、次の AWS インスタンスタイプを使用します。

- c5 または c5n は、SSL の大量のトラフィックを処理するのに適しています。
- c5.large は、最大 1000 の SSL TPS を処理できます。

詳細については、VPX-AWS サポートマトリックスを参照してください。

Citrix ADC を使用した **AWS** でのクリックストリーム分析の強化

October 7, 2021

お客様は、モバイルアプリ、SaaS アプリなどのさまざまなアプリケーションを通じて企業製品にアクセスするようになってきています。したがって、アプリケーションはカスタマーエクスペリエンスデータの地雷になる可能性があります。顧客行動をオンラインで追跡するために、顧客中心の企業は、この顧客行動データを使用して、各顧客に対してデータドリブンなプロファイルを形成します。

クリックストリームは、ウェブサイトまたはモバイルアプリケーションでのユーザーアクション（クリック）を表すイベントのシーケンスまたはストリームです。ただし、クリックストリームの範囲はクリックを超えています。これには、商品検索、インプレッション、購入、およびビジネスに関連する可能性のあるイベントが含まれます。カスタマーエクスペリエンスのデータを収集して保存するだけではあまり価値がありません。非常に複雑なデータを、適切なタイミングで適切なベンダーにシームレスに配信する必要があります。企業は、データから価値を引き出し、意識的な意思決定を迅速に実行して、戦略を改善することができます。したがって、企業はクリックストリーム分析を使用して、アプリのカスタマーエクスペリエンスのジャーニーに関するインサイトを収集するようになっています。

このドキュメントでは、クリックストリームデータが最も重要である理由、収集、保存、分散、有意義で実用的な分析に変換される方法について、十分に理解しています。

Citrix ADC は Citrix ADM と統合し、Amazon Kinesis Data Firehose などの AWS サービスに付加価値を追加して、ユーザーのクリックストリームを中心に展開するクラス最高の分析ソリューションを企業に提供します。

この Citrix ADC ソリューションは、複雑なビジネス上の問題を効率的かつ非常に簡単に解決するのに役立ちます。Citrix ADC と AWSKinesis は、不十分に設計されたワークフローの問題を把握するのに役立ちます。Citrix ADM は、関連するフィルタを適用することで、Web アプリケーションおよびネットワークパフォーマンスに関連する問題を把握するのに役立ちます。Citrix ADC と CitrixADM および AWSKinesis を組み合わせることで、各フェーズでの大量のクリックストリームデータの流入を管理および分析できます。このソリューションは、可用性、拡張性、堅牢性を備え、継続的かつセキュアなデリバリーを保証します。したがって、実用的なインサイトを導き出すことができます。

企業がクリックストリーム分析を選択する理由

企業は主に、ユーザーがアプリケーションとのやりとりを理解し、アプリケーションの目標を改善するためのインサイトを得るために、クリックストリームを選択します。クリックストリーム分析は、ユーザーの行動、ナビゲーションの習慣などを追跡する情報検索ユースケースです。クリックストリーム分析では、次の情報が表示されます。

- 顧客がクリック頻度が高くどのリンクを、どの時点でクリックしているのですか。
- ウェブサイトにアクセスする前の訪問者はどこにいましたか?
- 訪問者は各ページにどれくらいの時間を費やしましたか?
- 訪問者がウェブブラウザの「戻る」ボタンをクリックしたのはいつですか?
- 訪問者がショッピングカートに追加（または削除）した商品はどれですか?
- 訪問者が私のウェブサイトを出たのはどのページからですか?

Amazon Kinesis を使用してクリックストリームデータを管理する分析サービス

Amazon Kinesis を使用して、クリックストリーム分析を実行できます。Amazon Kinesis によって、次のサービスでクリックストリーム分析ができます:

- Amazon Kinesis データファイヤーホース
- Amazon Kinesis データ分析
- Amazon Kinesis データストリーム

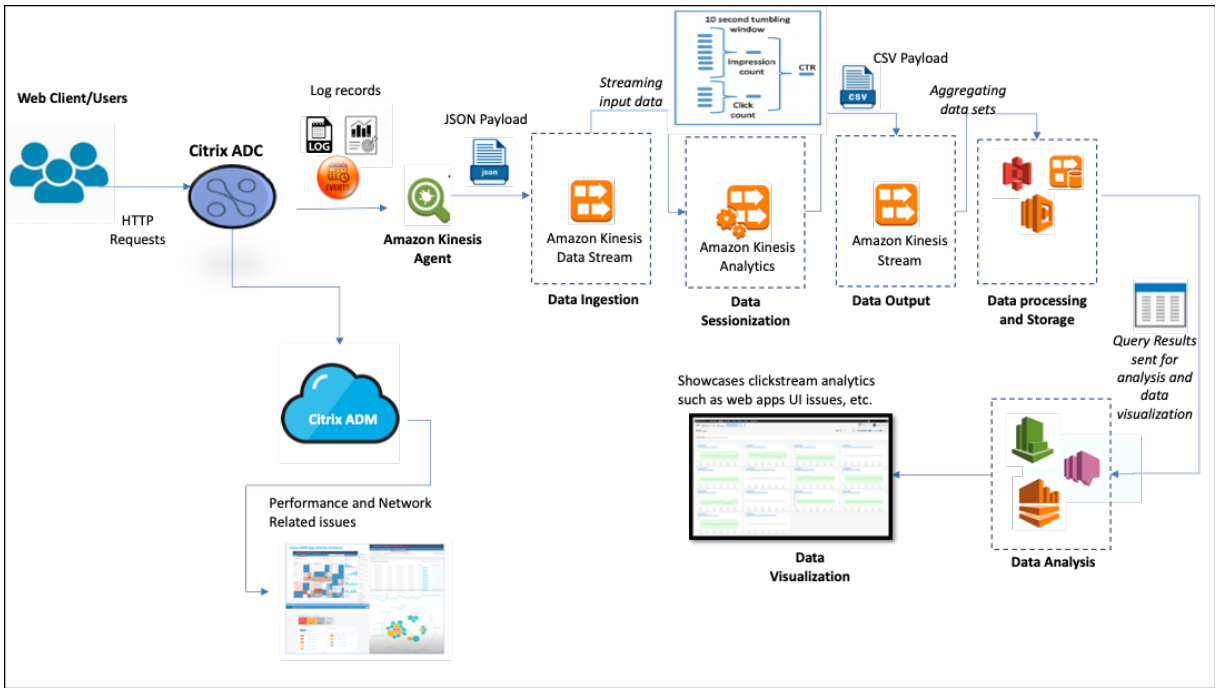
Amazon Kinesis を使用すると、あらゆる規模で膨大なデータセットを収集して分析できます。AWS Kinesis は、次のようなさまざまなソースからのデータを処理できます。

- モバイルおよびウェブアプリケーション（ゲーム、e コマースなど）
- IoT デバイス
- ソーシャルネットワーキングアプリケーション
- 金融取引サービス
- 地理空間サービス

Citrix ADC がクリックストリーム分析を可能にする方法

Citrix ADC ソリューションは、訪問したウェブサイト、消費された帯域幅、ナビゲーションフローなど、ユーザーのアクティビティに関する情報を安全に照合して提供します。企業は、この高いスループットと継続的なクリックストリームデータを分析して、次の効果を実証します。

- サイトレイアウト
- マーケティングキャンペーン
- 新しいアプリケーション機能



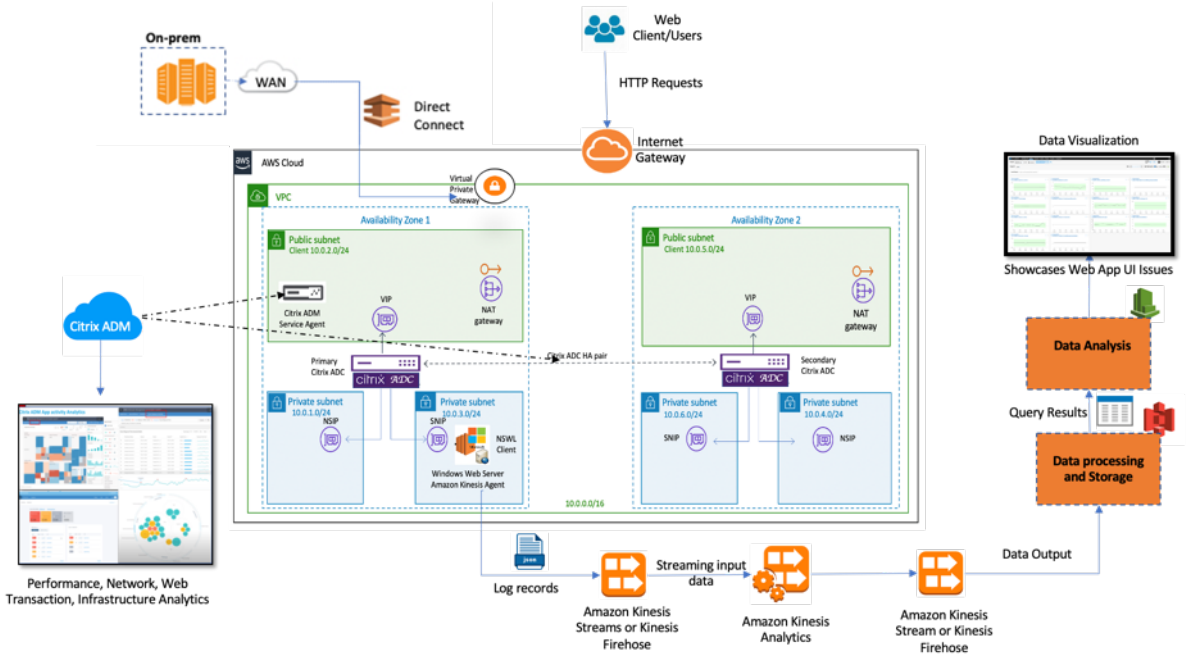
エンタープライズ環境に復元力のあるネットワーク保護を提供する Citrix ADC の機能により、計算量の多いタスクをオフロードし、このデータでセッションを実行することにより、サーバーのコストが大幅に削減されます。これにより、企業は常に高可用性、セキュリティ、低遅延でイベントをリアルタイムで識別できます。

構成情報については、「クリックストリーム分析用の Citrix ADC ソリューションを構成する」を参照してください。

Citrix ADC と Citrix ADM が AWS 環境をどのように補完するか

次の図は、AWS インフラストラクチャでクリックストリーム分析を実行するためのエンドツーエンドのユーザーワークフローを示しています。この図は、次のプロセスを理解するのに役立ちます。

- ユーザーが Citrix ADC と対話する方法
- Citrix ADC がユーザーのアクションをキャプチャしてクリックストリームデータを生成する方法
- クリックストリームデータが AWS サービスに配信される方法 (Amazon Kinesis)
- Amazon Kinesis がデータログを処理して保存し、有意義なクリックストリーム分析を生成する方法



Citrix ADC は、AWS 環境と Citrix ADM にシームレスに統合され、企業がクリックストリームデータの可変ボリュームと多様な性質と互換性を持たせるように支援します。ストリーミングの知識を簡単に読み込み、分析するサービスを提供します。また、特殊な要望に合わせてカスタムストリーミングナレッジアプリケーションを作成することもできます。

Amazon Kinesis

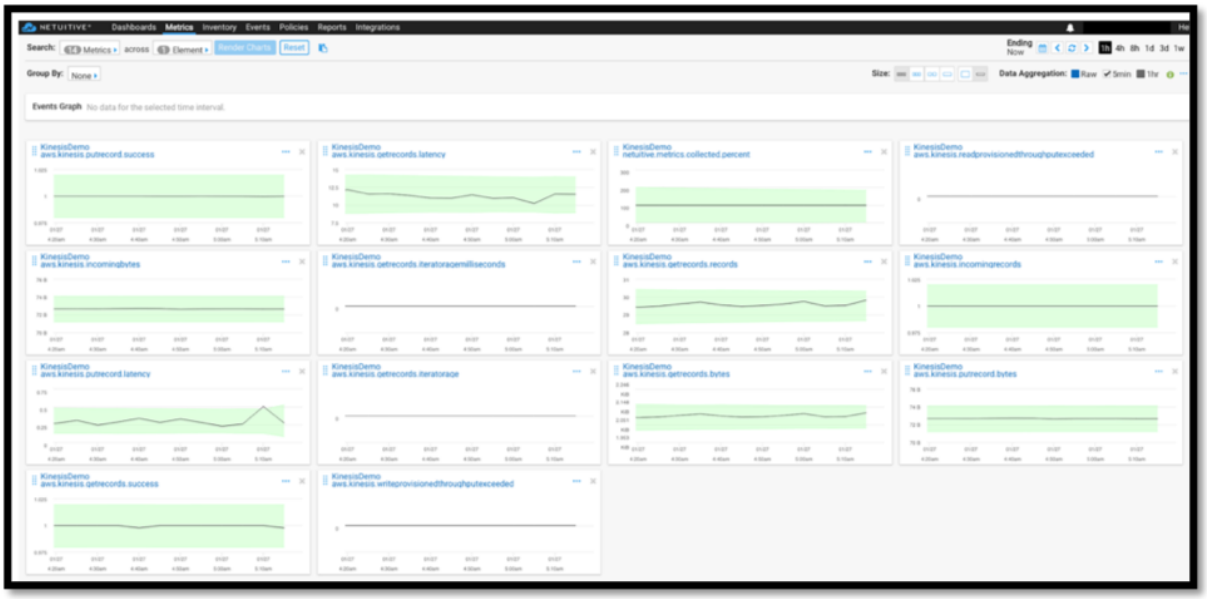
AWS 環境には、Citrix ADC によってキャプチャされたユーザーイベント、ログ、メトリックスの分析を実行するさまざまなサービスがあります。データには、ウェブサイトのクリックストリーム、財務トランザクション、ソーシャルメディアフィード、IT ログ、および位置追跡イベントが含まれます。

- Amazon Kinesis Data Streams は、複数のソースから 1 秒あたりのデータを継続的にキャプチャできる、スケーラブルで耐久性のあるリアルタイムデータストリーミングを含むシナリオで分析を実行します。
- Amazon Kinesis Data Analytics は、さまざまなデータセットの集約にかかる時間が短いため、セッション生成間のレイテンシーが短いシナリオで使用できます。
- Microsoft Windows 版 Amazon Kinesis エージェントは、入力データを収集、解析、フィルタリング、および Kinesis データストリームにストリーミングします。
- データがクラウドで稼働したら、正確なデータパイプラインを実装して、必要な結果を得ることができます。たとえば、この情報を Amazon Quick Sight で使用できます。Amazon Quick Sight は、ダッシュボードの作成に使用される視覚化ツールです。

AWS Kinesis ダッシュボードには、次のサービスがあります。

- ウェブアプリの UI の問題を紹介する
- 時間あたりのイベント、訪問者数、リファラーなど、Web 使用状況の指標をほぼリアルタイムで視覚化します。

- セッション単位の分析



Citrix ADM アナリティクス

CitrixADM と Citrix ADC を利用することで、すべてのビジネス環境にわたって一枚のガラスのビューを取得できます。Citrix ADC でキャプチャされたログは CitrixADM に送られ、CitrixADM は個々のアプリケーションを単一のエンティティとして扱います。次の ADM 機能を使用すると、貴重なインサイトを獲得し、問題を効果的にトラブルシューティングできます。

- インテリジェント・アナリティクス
- Web トランザクション分析
- 異常検出
- パフォーマンスおよびネットワーク関連の問題

次の ADM サービスダッシュボードは、問題を効果的にトラブルシューティングするための貴重なインサイトを得るのに役立ちます。

Citrix ADM とクリックストリーム分析との相関関係

クリックストリーム分析データを ADM 分析と関連付けて、アプリケーションのパフォーマンスを記述、予測、および改善できます。

Citrix ADM の詳細については、「Citrix ADM%20is%20A%20 集中型%20 管理%20 ソリューション

たとえば、組織がログを分析しているときに、ほとんどのユーザーがサイトを放棄していることに気づきました。しかし、このユーザー行動の根本的な原因を見つけるには、アプリケーションのどの部分が悪いのかを調べる必要があります。クリックストリーム分析データと ADM 分析を使用すると、次のインサイトを導き出して、ユーザーがサイトを放棄した理由を分析できます。

- レイテンシー、5xx エラーが原因でユーザーが放棄されていますか?
- SSL ハンドシェイクエラーはありますか。
- パフォーマンスやネットワークに関連する問題があるアプリケーションの一部はありますか。
- 404 エラーがあるか、ページの読み込み時間が応答するのに永遠にかかる、など。
- 顧客はサーバー応答の異常に直面していますか。

Citrix ADM サービスは、IT 管理者が次の機能を使用して問題を解決するための Web インサイトを提供します。

- Citrix ADC によって提供されるすべての Web アプリケーションの統合されたリアルタイムの監視を提供します。

- 観測性ツール（グローバルサービスグラフなど）を使用して、時間、レイテンシー、および通常のユーザーの動作に関するアプリケーションのパフォーマンスに関する全体的なビューを取得します。
- インテリジェントな分析を実行して、サーバー応答の異常を理解します。
- SSL インサイトは、5xx および 4xx エラーの解決に貢献します。
- 以下を含むすべての Web セッションの記録を維持するには
  - すべてのウェブトランザクションの詳細ログ
  - 関連ログを検索する検索機能
  - ADC・ツー・エンド・ユーザーを隔離する機能 ADC からサーバーへの問題

ADC がクリックストリーム分析用にエクスポートしたデータの種類

Citrix ADC は、次のようなさまざまな形式のデータを生成するさまざまなソースをキャプチャします。

- Web サーバーログ

  Web サーバロギング機能は、HTTP および HTTPS 要求のログをクライアントシステムに送信して、ストレージと取得を行います。これらのログには膨大な量のデータが含まれており、理解するのが難しく、そこから意味があります。分析ツールは、それを理解し、価値を引き出すのに役立ちます。設定の詳細については、このドキュメントの「**Web** ロギングの設定」セクションを参照してください。

- Syslogs

  syslog の主な用途は、システム管理です。プロアクティブな Syslog モニタリングは、インフラストラクチャ内のサーバーやその他のデバイスのダウンタイムを大幅に削減するため、効果があります。Syslog は、重要なネットワークの問題を特定し、プロアクティブに報告します。

- アクセスログ

  アクセスログには、Web サーバーで発生したイベントに関する情報が格納されます。たとえば、誰かがあなたのウェブサイトを訪問すると、ログが記録され、ウェブサーバー管理者に訪問者の IP アドレス、閲覧していたページ、ステータスコード、使用されたブラウザなどの情報が提供されます。ログを理解するための適切な知識が不足している場合、ログにアクセスするのは圧倒的かもしれません。
  システムを次のものに統合するようにプログラムできます。

  - シームレスな配信のための Citrix ADC
  - ビジネスに役立つ実用的なインサイトのための Kinesis

- 監査ログ

  監査ログ機能を使用すると、カーネルおよびユーザーレベルのデーモン内のさまざまなモジュールによって収集された Citrix ADC の状態とステータス情報をログに記録できます。

- エラーログ

  エラーログファイルは、管理者が Web サーバーで発生した特定のエラーに関する詳細情報を提供するための補助となります。

クリックストリーム分析用に **Citrix ADC** ソリューションを構成する

Web サーバーのログ機能を使用すると、HTTP および HTTPS 要求のログをクライアントシステムに送信して、ストレージと取得を行うことができます。

Web サーバーロギング用に Citrix ADC を構成するには、次のことを行う必要があります。

- Web ログ機能を有効にする
- Web ログサーバーが Citrix ADC で実行されるため、ログエントリを一時的に保存するようにバッファのサイズを構成します。

CLI を使用して Web サーバロギングを設定するには、次の手順を実行します。

1. Web サーバロギング機能を有効にします。

```
enable ns feature WL
<!--NeedCopy-->
```

2. [（オプション）] ログ情報を格納するためのバッファサイズを変更/設定します。

```
set ns weblogparam -bufferSizeMB 60
<!--NeedCopy-->
```

3. Citrix ADC Web ロギング（NSWL）クライアントをインストールします。詳細については、「Citrix ADC Web ロギング（NSWL）クライアントのインストール」を参照してください。
4. Windows で NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。
   a) nswl_win–.zip< release number > ファイルを展開して、< build number > NSWL クライアントをインストールする Windows システムにパッケージからコピーします。
   b) Windows システムでは、ファイルをディレクトリ (を参照 < NSWL-HOME>) に解凍します。ビン、サンプル、およびその他のディレクトリが抽出されます。
   c) コマンドプロンプトで、< NSWL-HOME >\ bin ディレクトリから次のコマンドを実行します。

```
nswl -install -f < path of the log.conf file >\log.conf
<!--NeedCopy-->
```

> 注:

> NSWL クライアントをアンインストールするには、コマンドプロンプトで < NSWL-HOME >\ bin ディレクトリから次のコマンドを実行します。

```
nswl -remove
<!--NeedCopy-->
```

5. NSWL クライアントをインストールしたら、NSWL 実行可能ファイルを使用して NSWL クライアントを構成します。これらの構成は、NSWL クライアント構成ファイル (log.conf) に保存されます。

   NSWL 実行可能ファイルが配置されているディレクトリから次のコマンドを実行します。

```
\ns\bin
<!--NeedCopy-->
```

6. NSWL クライアント構成ファイル（log.conf）で、クライアントシステムのコマンドプロンプトで次のコマンドを実行して、NSWL クライアントがログを収集する Citrix ADC IP アドレス（NSIP）を追加します。

```
nswl -addns -f < Path to the configuration(log.conf) file >\log.
    conf
<!--NeedCopy-->
```

7. Citrix ADC アプライアンスの NSIP（IP アドレス）、ユーザー名、nsroot およびパスワードを「インスタンス ID/設定したパスワード」として入力します。

   - NSWL クライアントは、NetScaler IP アドレス（NSIP）を NSWL 構成ファイルに追加した後、ADC に接続します。
   - ADC は、HTTP および HTTPS 要求ログエントリをクライアントに送信する前にバッファリングします。
   - クライアントは、エントリを保存する前に（log.conf ファイルを変更して）エントリをフィルタリングできます。

> 注
>
> Citrix ADC デフォルトパスワードを変更し、構成を続行します。次のコマンドを入力して、パスワードを変更します。

```
set system user nsroot -password <your password>
<!--NeedCopy-->
```


Amazon Kinesis エージェントの設定

AWS ウェブコンソールで次の手順を実行します。Amazon Kinesis エージェントを設定します。

1. 設定ファイル (appsettings.json) を作成してデプロイします。構成ファイルは、ソースをシンクに接続するソース、シンク、およびパイプのセットと、オプションの変換を定義します。

   次の例は、Windows アプリケーションログイベントを Kinesis Data Firehose にストリーミングするように Kinesis `appsettings.json` エージェントを設定する完全な設定ファイルです。

```
{

"Sources": [
    {

        "Id": "NSWLog",
        "SourceType": "DirectorySource",
        "Directory": "C:\Users\Administrator\Downloads\nswl_win
           -13.0-52.24\bin",
        "FileNameFilter": "*.log"
        "RecordParser": "TimeStamp",
        "TimestampFormat": "yyyy-MM-dddd HH:mm:ss.ffff", //
           Optional parameter required only by the timestamp
           record parser
        "TimeZoneKind": "UTC", //Local or UTC
        "SkipLines": 0 //Skip a number of lines at the beginning
           of each file
     }

    ],
"Sinks": [
    {

        "Id": "ApplicationLogKinesisFirehoseSink",
        "SinkType": "KinesisFirehose",
        "StreamName": "Delivery-ik-logs",
        "AccessKey": "Your Access Key",
        "SecretKey": "YourSecretKey",
        "Region": "ap-south-1"
     }

    ],
"Pipes": [
    {

```



```
32      "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink
            ",
33      "SourceRef": "ApplicationLogSource",
34      "SinkRef": "ApplicationLogKinesisFirehoseSink"
35       }
36
37      ],
38  "Telemetrics":
39      {
40
41      "off": "true"
42       }
43
44   }
45
46  <!--NeedCopy-->
```

2. データソースに KinesisAgent をセットアップしてデータを収集し、AmazonKinesis に継続的に送信します Firehose/Kinesis データ分析。詳細については、「Microsoft Windows 用 Amazon Kinesis エージェントの使用開始」を参照してください。
3. Amazon Kinesis Firehoseを使用して、エンドツーエンドのデータ配信ストリームを作成します。配信ストリームは、エージェントから宛先にデータを送信します。送信先には、Amazon Kinesis Analytics、Amazon Redshift、Amazon Elasticsearch サービス、Amazon S3 が含まれます。[ソース] で、[直接 **PUt** またはその他のソース] を選択して Kinesis Data Firehose 配信ストリームを作成します。
4. Amazon Kinesis アナリティクスで SQL クエリを使用して受信ログデータを処理します。
5. 処理されたデータを Kinesis アナリティクスから Amazon Elasticsearch サービスにロードして、データのインデックスを作成します。
6. Kibana や AWS QuickInsight Services などの可視化ツールを使用して、処理されたデータを分析し、視覚化します。

参照ドキュメント

- syslog メッセージの表示とエクスポート
- ハイブリッドマルチクラウド向け Citrix Networking
- Kinesis エージェントを使用した AWK Kinesis データストリームへの書き込み

Microsoft Windows Azure パックと Cisco ACI によって管理されるプライベートクラウドでの Citrix ADC

October 7, 2021

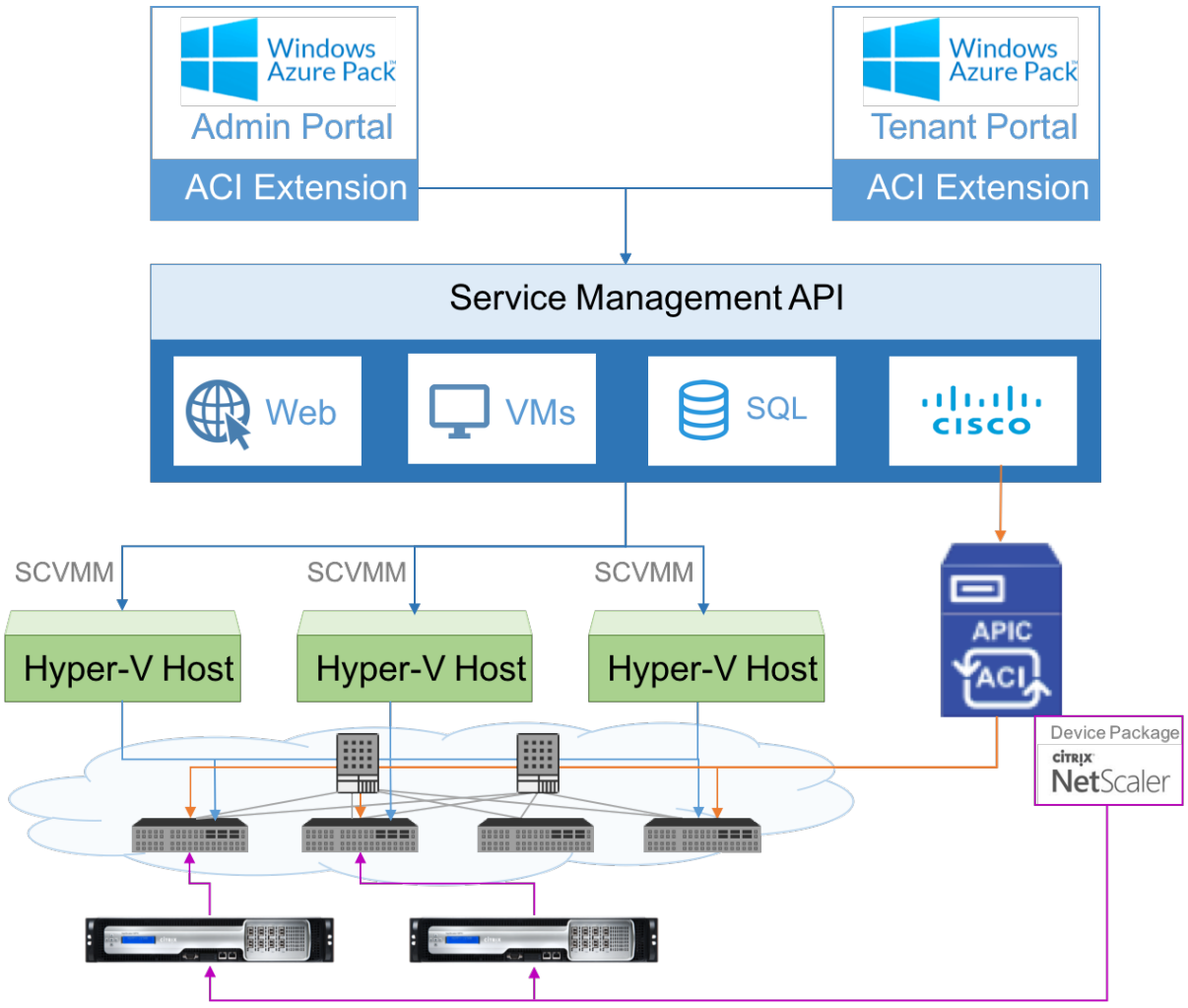
Citrix ADC アプライアンスを使用して、Microsoft Windows Azure パックで管理されるプライベートクラウドの負荷分散を行うことができます。プライベートクラウドのネットワークは、Cisco ACI および Citrix ADC を使用して自動化されます。

このソリューションには、Windows Azure パック (WAP) から Cisco APIC、システムセンターのバーチャルマシンマネージャー (SCVMM)、Cisco APIC から Citrix ADC への Cisco APIC など、多くの統合ポイントが含まれます。プライベートクラウドのテナントとして、NAT の有効化、ネットワークサービスのプロビジョニング、ロードバランサーの追加を行うことができます。

WAP は、管理者が ACI 登録、VIP 範囲、仮想マシンクラウドとの Citrix ADC デバイスの関連付け、テナントユーザーアカウントの作成などの管理タスクを実行できるテナントと管理者ポータルをサポートしています。テナントは WAP テナントポータルにログオンし、ネットワーク、ブリッジドメイン、仮想ルーティングと転送（VRF）を構成し、Citrix ADC 負荷分散と RNAT 機能を利用できます。

> 重要
>
> - このソリューションでは、Citrix ADC アプライアンスは基本的な負荷分散のみを提供します。
> - テナントは、同じネットワークに異なるポートを持つ複数の VIP アドレスを展開できますが、IP とポートの組み合わせが一意であることを確認する必要があります。
> - Citrix ADC デバイスパッケージは、シングルコンテキスト展開のみをサポートします。各テナントは、専用の Citrix ADC インスタンスを取得します。
> - WAP は、Citrix ADC MPX アプライアンスおよび Citrix ADC VPX 仮想アプライアンスをサポートします。これには、Citrix ADC SDX プラットフォーム上に展開された Citrix ADC VPX インスタンスを含みます。

次の図に、ソリューションの概要を示します。



前提条件

以下の点について確認してください：

- Cisco ACI コンポーネントと Citrix ADC の概念知識があります。
  - Cisco ACI とそのコンポーネントの詳細については、次の製品マニュアルを参照してください：http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html。
  - Citrix ADC の詳細については、http://docs.citrix.com/にある Citrix ADC 製品のドキュメントを参照してください。
- データセンター内の Cisco APIC を含め、Cisco ACI に必要なすべてのコンポーネントが設定および設定されています。Cisco ACI とそのコンポーネントの詳細については、次の製品マニュアルを参照してください：http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html。
- Microsoft の Windows Azure パックと Cisco ACI を統合する方法をご存じのはずです。製品のマニュアルを参照してください：http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/2-

x/virtualization/b_ACI_Virtualization_Guide_2_2_1.html。

- Microsoft の Windows Azure パックの概念的な知識があります。製品のマニュアルを参照してください: https://www.microsoft.com/en-in/cloud-platform/windows-azure-pack。
- Citrix ADC ソフトウェアリリース 11.1 以降がインストールされている。
- Citrix ADC は、Cisco ACI で設定し、Cisco APIC を使用して管理できるようにします。
- Cisco APIC から、次のことを確認します。
  - Cisco APIC と Citrix ADC の管理接続が確立されます。
  - Citrix ADC デバイスパッケージのバージョン *11.1 ～52.3* をアップロードし、Cisco APIC を使用して、Cisco ACI で Citrix ADC デバイスを登録します。
  - Cisco APIC の共通テナントで Citrix ADC アプライアンスを構成し、Cisco APIC に障害がないことを確認します。
  - VLAN プール、L3OutServicesDom、L3ExtOUt、リソースプールなど、すべての APIC 固有の設定が設定済みです。詳しくは、*Cisco* のドキュメントを参照してください。

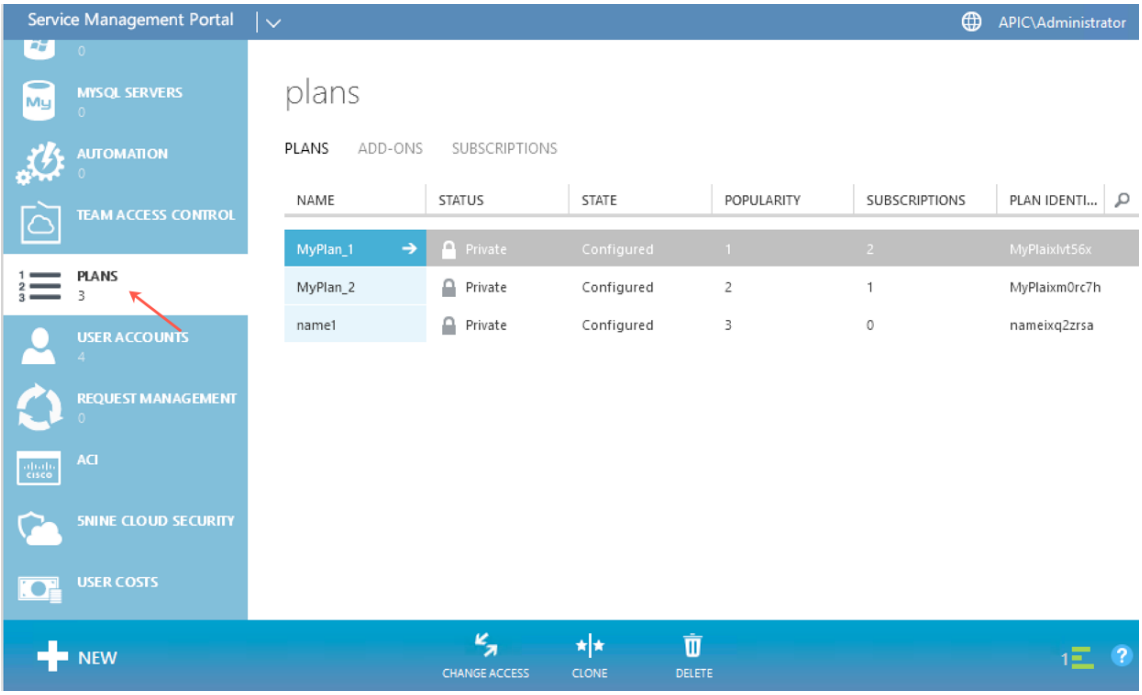
サービス管理ポータル（管理ポータル）でのプランでの **Citrix ADC** ロードバランサーの作成

October 7, 2021

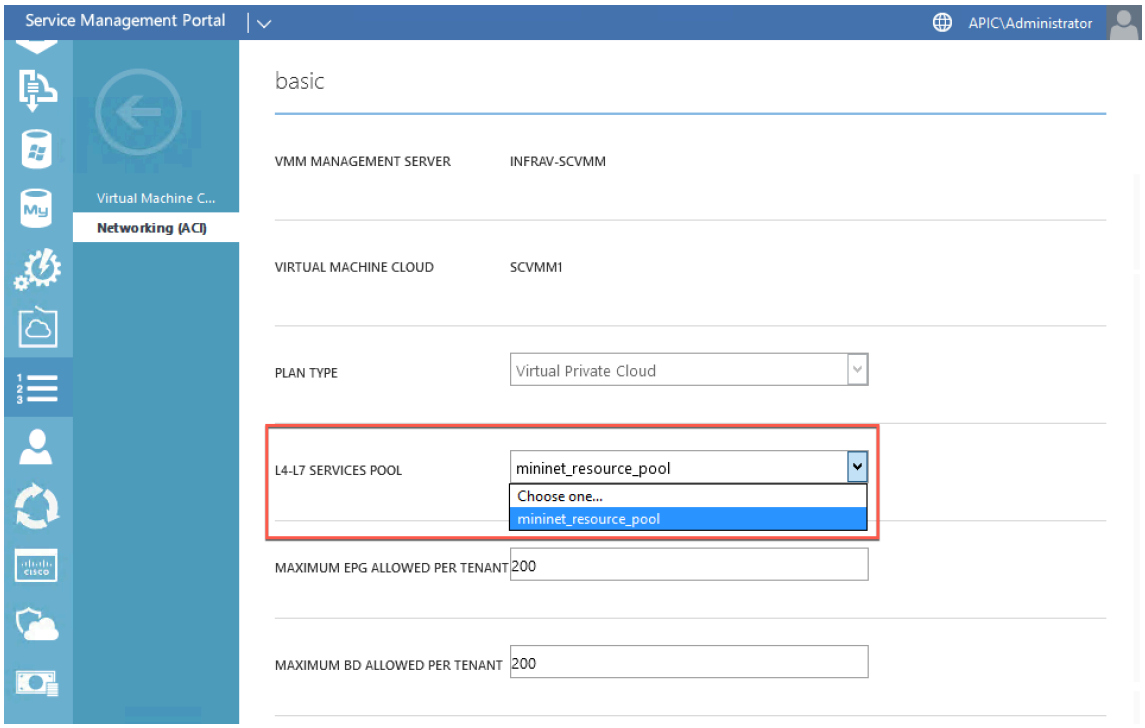
WAP のサービス管理ポータルは、管理者は、WAP と Cisco APIC を登録し、また、ホスティングプランを作成することができます。プランの一部として、VIP 範囲を指定し、Citrix ADC ロードバランサーをプランに関連付け、テナントユーザーアカウントを作成できます。

管理ポータルのプランで **Citrix ADC** ロードバランサーを作成するには:

1. サービス管理ポータル (管理ポータル) にログインします。
2. ナビゲーションペインで、[**PLANS**] を選択します。



3. [plans] ペインで、ロードバランサーを追加するプランを選択します。
4. 選択したプランのウィンドウで、[ネットワーク **(ACI)**] を選択します。
5. [ネットワーク（**ACI**）] ペインの [**L4-L7** サービスプール] ドロップダウンリストで、Cisco APIC で作成した L4-L7 リソースプールを選択します。



6. テナントユーザーアカウントを作成し、作成したプランにユーザーを関連付けます。

サービス管理ポータル (テナントポータル) を使用した Citrix ADC ロードバランサーの構成

October 7, 2021

WAP では、テナントがブリッジドメイン（BD）、VRF、およびネットワークを作成すると、テナントはサービス管理ポータル（テナントポータル）を介して Citrix ADC ロードバランサーを構成できます。

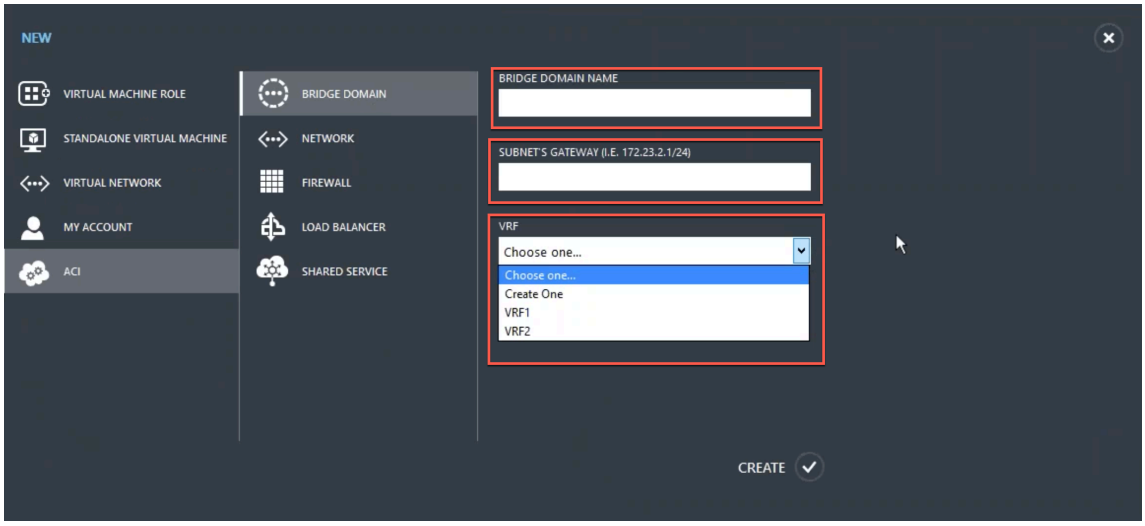
サービス管理ポータル (テナントポータル) で Citrix ADC ロードバランサーを構成するには

1. サービス管理ポータル (テナントポータル) にログオンします。
2. 次のように、ブリッジドメインと VRF を作成します。

   a. ナビゲーションペインで、[**ACI**] を選択します。

   b.「新規」をクリックします。

   c. [**NEW**] ペインで、[**BRIDGE DOMAIN**] を選択します。



   d. [**BRIDGE DOMAIN**] フィールドに、ブリッジドメイン名（BD01 など）を入力します。

   e.（オプション）**[SUBNET'S GATEWAY]** フィールドに、サブネットのゲートウェイを入力します（たとえば、192.168.1.1/24）。

   f. [**VRF**] フィールドで、すでにサブスクリプションの一部である VRF を選択するか、[**Create One**] を選択して VRF を作成します。

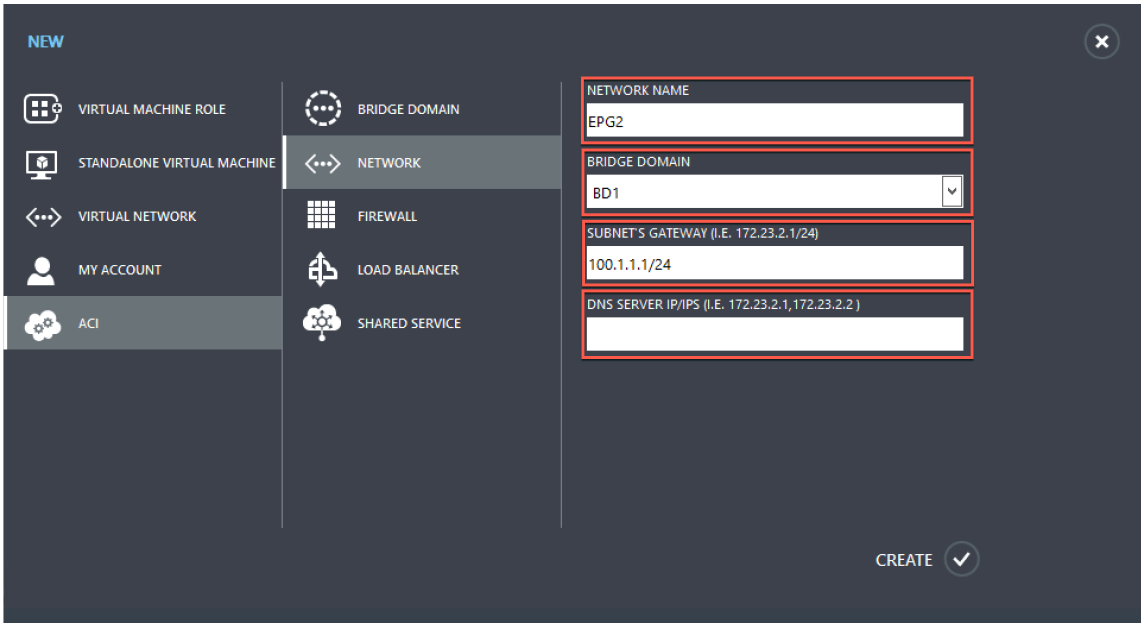
   g.［**CREATE**］をクリックします。

3. ネットワークを作成し、作成したブリッジドメインに関連付けます。以下を実行します：

   a. ナビゲーションペインで、[**ACI**] を選択します。

   b.「新規」をクリックします。

c. [新規] ペインで、[ネットワーク] を選択します。



d.「ネットワーク名」フィールドに、ネットワーク名（S01 など）を入力します。

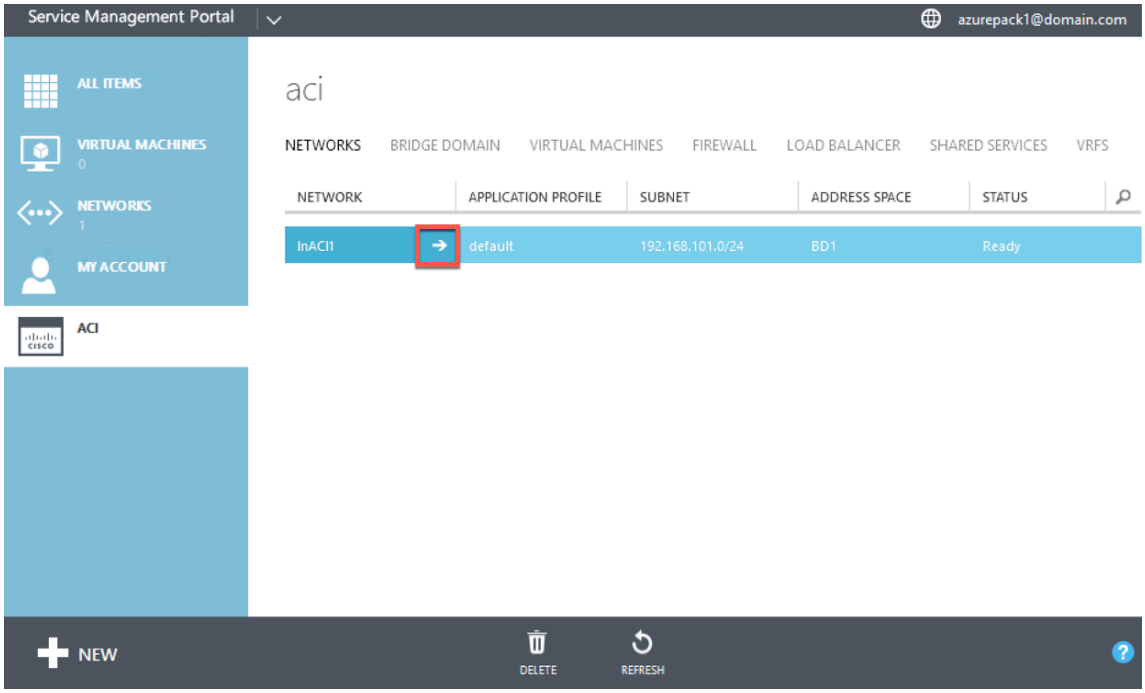
e. [**BRIDGE DOMAIN**] ドロップダウンリストで、作成したブリッジドメインを選択します。(たとえば、BD01) をクリックします。

f. サブネットの **GATEWAY** フィールドに、サブネットの Gateway アドレス（172.23.2.1/24 など）を入力します。

g. （任意）[**DNS SERVER IP/IPS**] フィールドに、DNS サーバの詳細を入力します。

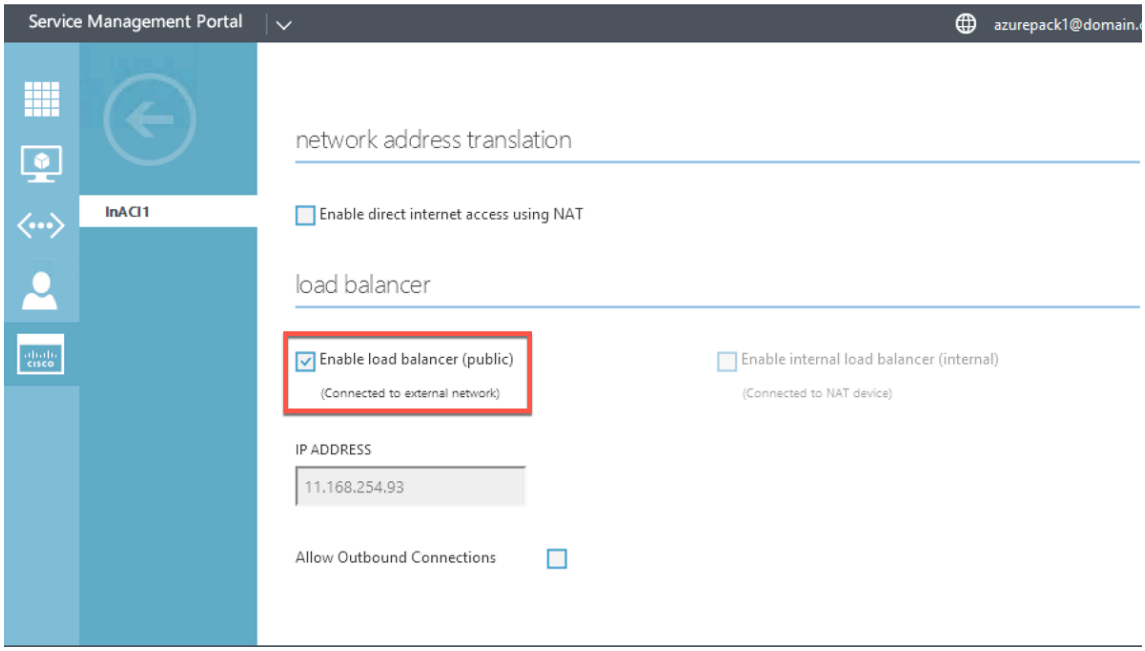
h. [**CREATE**] をクリックします。

4. [**ACI**] ペインで、[ネットワーク] を選択します。

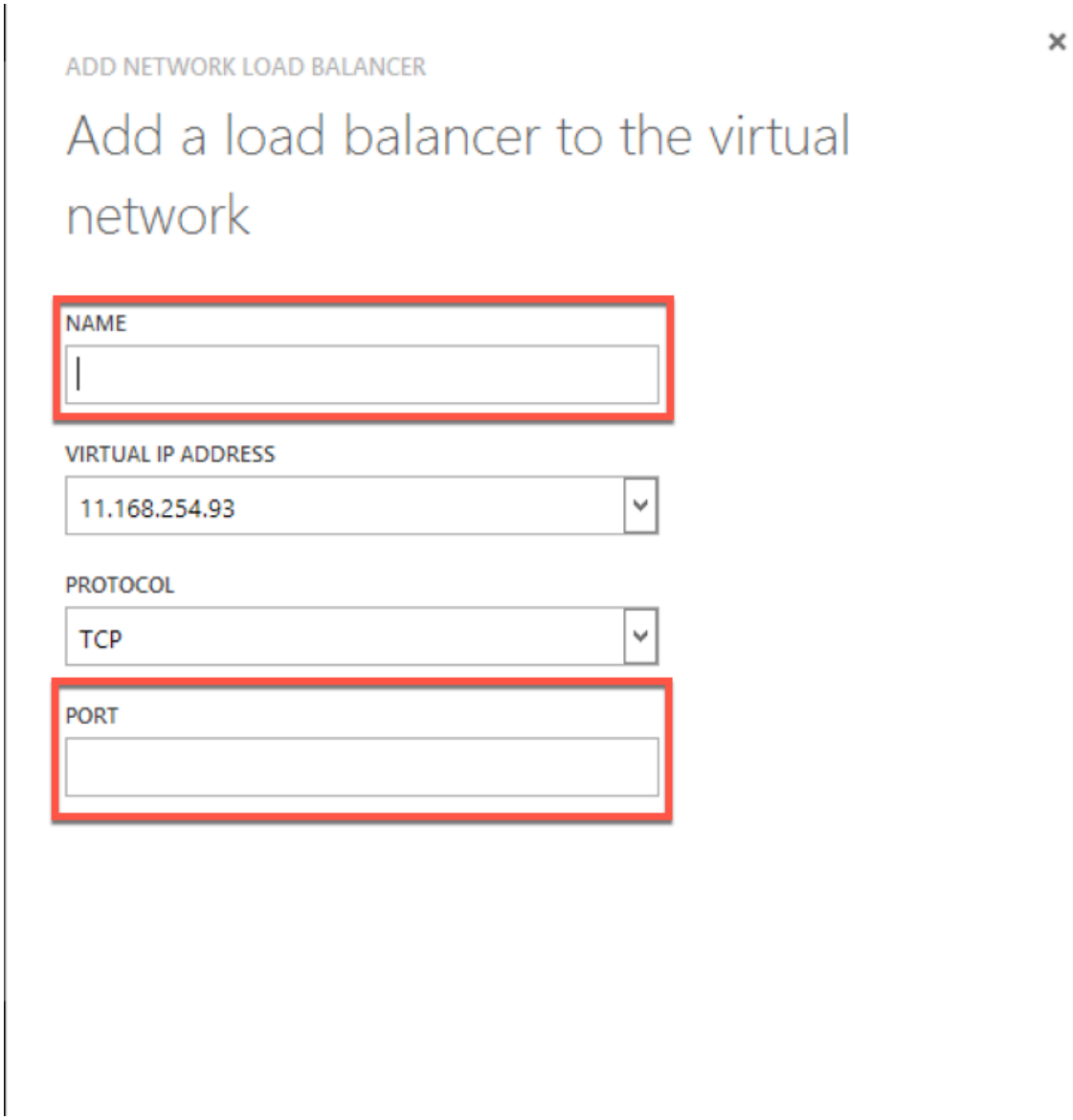


5. 作成したネットワークをダブルクリックします。次に、ネットワークペインで、[ロードバランサーを有効にする **(**パブリック**)**] を選択します。[**IP** アドレス] フィールドでは、管理者が管理者ポータルで設定した VIP 範囲から VIP が自動的に割り当てられます。詳細については、サービス管理ポータル（管理者ポータル）の「プランでの Citrix ADC ロードバランサーの作成」を参照してください。

6. 作成したネットワークをダブルクリックします。次に、ネットワークペインで、[ロードバランサーを有効にする **(**パブリック**)**] を選択します。[**IP** アドレス] フィールドでは、管理者が管理者ポータルで設定した VIP 範囲から VIP が自動的に割り当てられます。詳細については、サービス管理ポータル（管理者ポータル）の「プランでの Citrix ADC ロードバランサーの作成」を参照してください。



7. ネットワークペインで、[**Load Balancers**] タブを選択し、[追加] をクリックします。



8. [ネットワークロードバランサを追加] ウィンドウで、次の操作を行います。

   a. [**NAME**] フィールドに、ロードバランサーの名前を入力します。

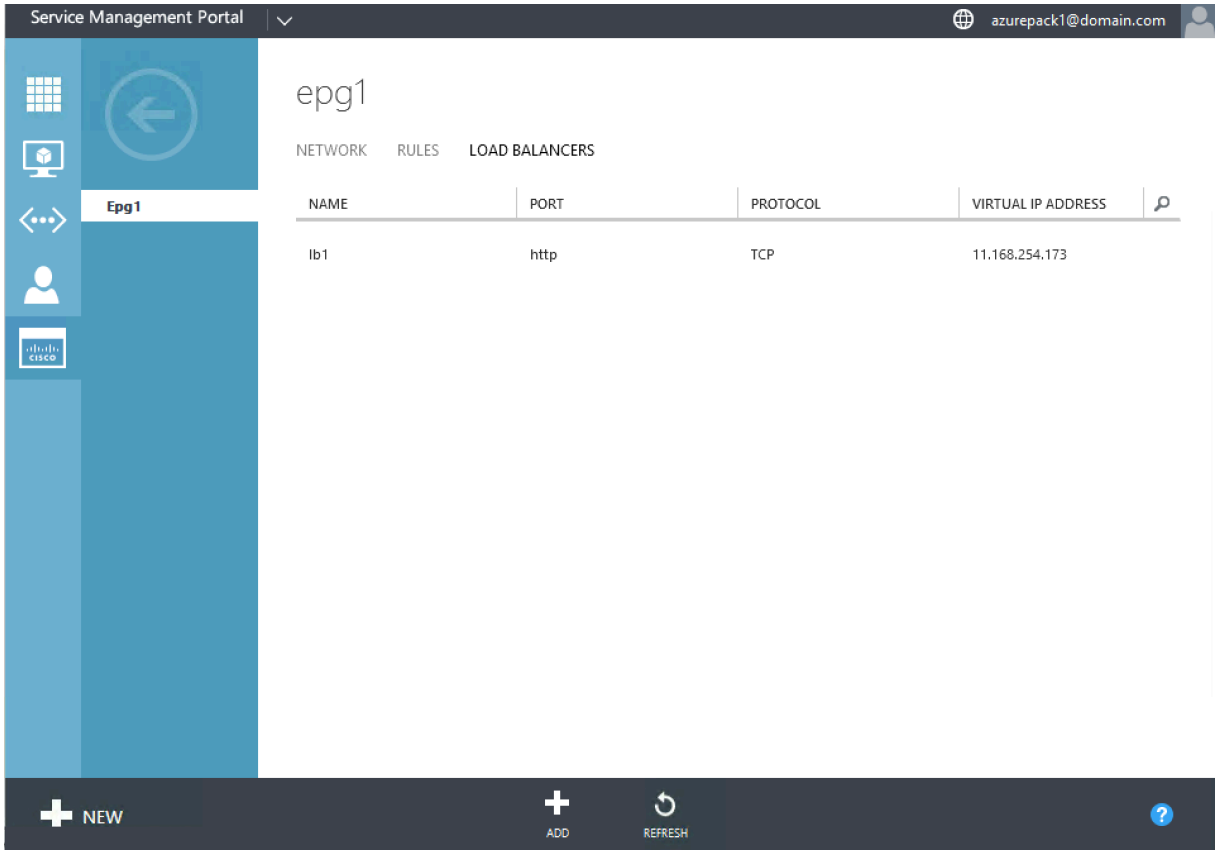
   b. オプションで、[**VIRTUAL IP** アドレス] フィールドで、以前に定義した VIP 範囲の VIP アドレスをロードバランサに割り当てます。

   c. 必要に応じて、[プロトコル] フィールドで [**TCP**] を選択します。

   d.「**PORT** 」フィールドにポート番号を入力します。

9. [**CREATE**] をクリックします。

   Citrix ADC ロードバランサーが「ロードバランサー」タブに表示され、**Citrix ADC** ロードバランサーがデータパスに対応している状態になります。



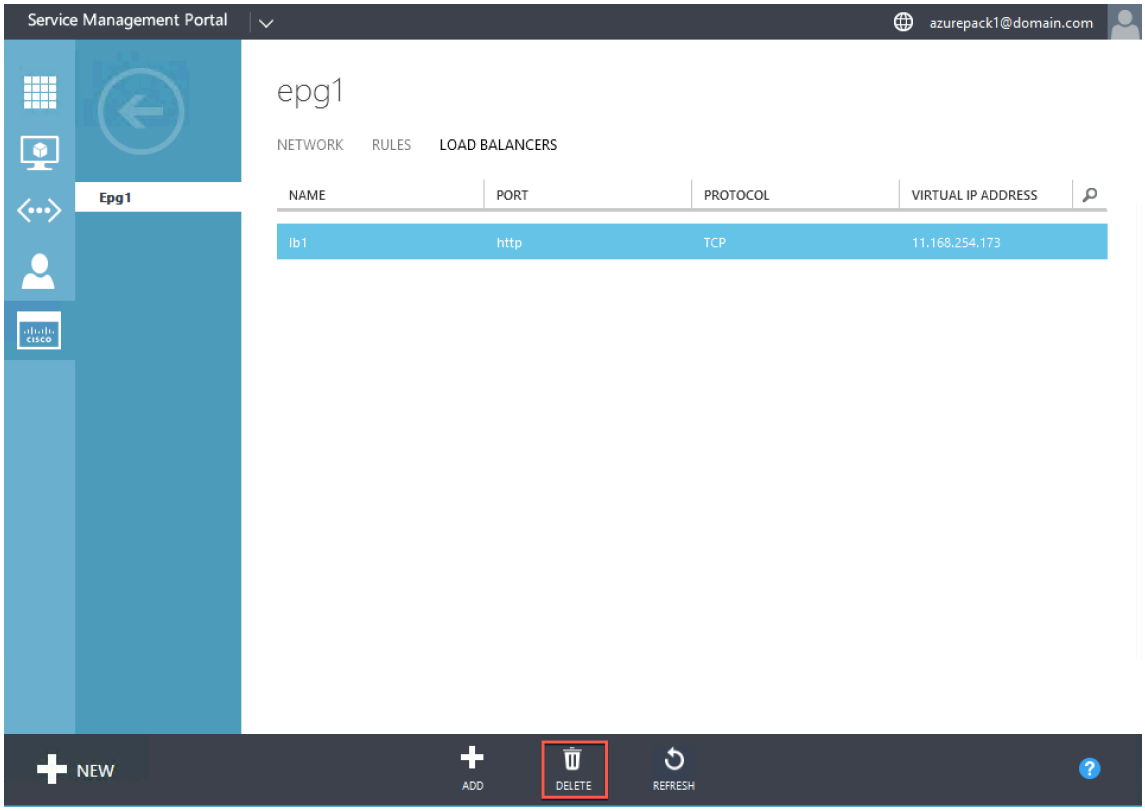
ネットワークからの **Citrix ADC** ロードバランサーの削除

October 7, 2021

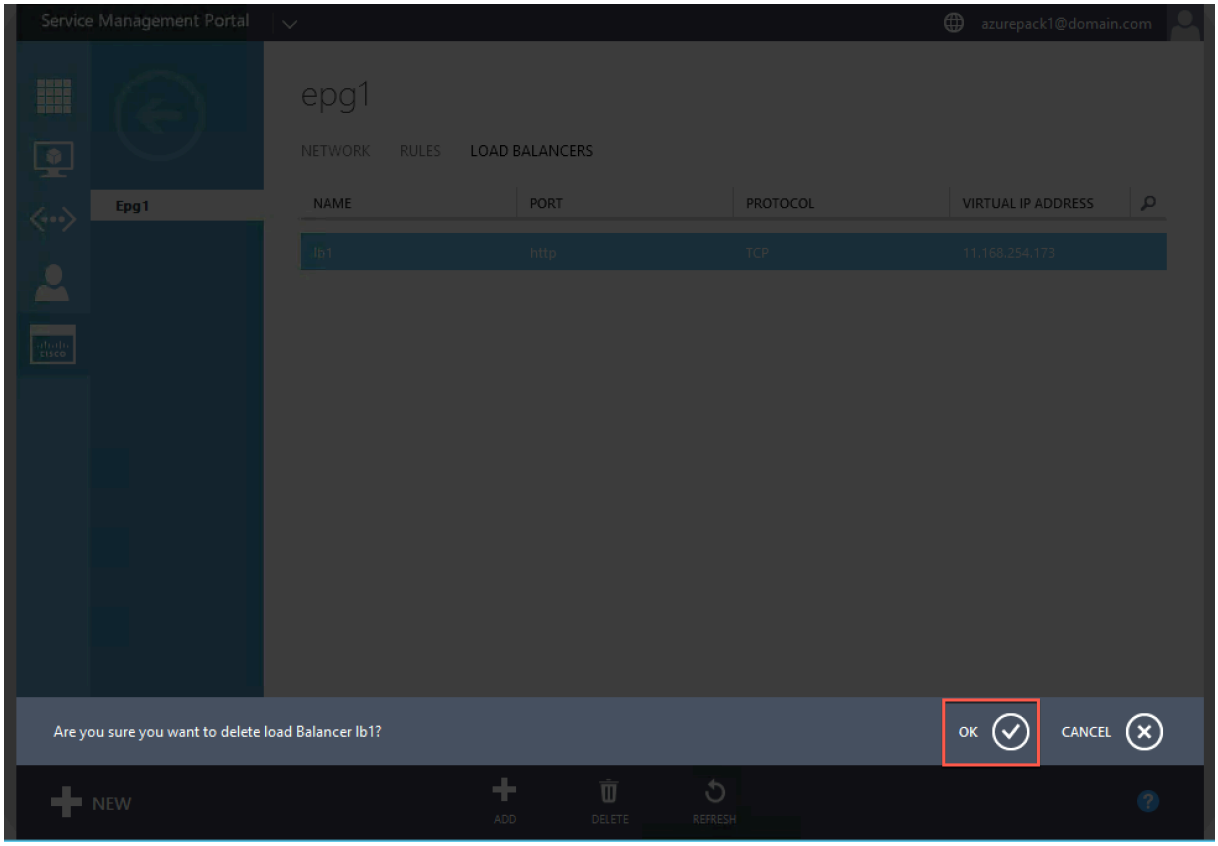
サービス管理ポータル（テナントポータル）を使用して、ネットワークから、作成した Citrix ADC ロードバランサーを削除できます。

ネットワークから **Citrix ADC** ロードバランサーを削除するには:

1. サービス管理ポータル (テナントポータル) にログオンします。
2. ナビゲーションペインで、[**ACI**] を選択します。
3. [**ACI**] ペインの [**NETWORKS**] タブで、作成したネットワークをクリックします。
4. 選択したネットワークのペインで、Citrix ADC ロードバランサーを選択し、［削除］をクリックします。



5. [**OK**] をクリックして、Citrix ADC ロードバランサーを削除します。



Kubernetes に基づくマイクロサービス向けの **Citrix** クラウドネイティブソリューション

October 7, 2021

企業がより迅速にイノベーションを起こし、顧客に近づくように変化するにつれて、企業は内部プロセスを再設計し、組織内の境界を打ち破っています。彼らは、同じチーム内の適切なスキルセットをまとめるためにサイロを削除しています。目標の 1 つは、スピード、敏捷性、効率性を備えたソフトウェアアプリケーションを作成して提供することです。この点で、マイクロサービスに基づく最新のアプリケーションアーキテクチャは、ますます多くの企業に採用されています。
マイクロサービスアーキテクチャを使用すると、個別にデプロイ、更新、スケーリングできる疎結合サービスのセットとしてアプリケーションを作成できます。

クラウドネイティブは、次の主要な属性を持つアプリケーションを構築およびデプロイするためにマイクロサービスアーキテクチャに依存するアプローチです。

- 疎結合のマイクロサービスまたはコンテナーとしてアプリケーションをデプロイします
- 非常に高度な自動化が含まれます
- アジャイル DevOps プロセスと継続的デリバリーワークフローを実装します
- 相互作用とコラボレーションのための API を中心に

Kubernetes はクラウドネイティブジャーニーにどのように役立つのでしょうか

必要なレベルの俊敏性と安定性を提供するために、クラウドネイティブアプリケーションには、高レベルのインフラストラクチャの自動化、セキュリティ、ネットワーキング、および監視が必要です。コンテナを大規模に効率的に管理できるコンテナオーケストレーションシステムが必要です。Kubernetes は、コンテナのデプロイとオーケストレーションの最も一般的なプラットフォームとして登場しました。Kubernetes は、開発者やオペレーターからコンテナーを実行、デプロイ、管理するという複雑なタスクを抽象化し、ノードのクラスター間でコンテナーを自動的にスケジュールします。Kubernetes とクラウドネイティブコンピューティングファンデーション（CNCF）エコシステムは、クラウドネイティブソリューションのプラットフォームを構築するのに役立ちます。

Kubernetes を使用する主な利点のいくつか:

- オンプレミス、ハイブリッド、またはパブリッククラウドインフラストラクチャのアプリケーション展開を簡素化します
- アプリケーションの開発と展開を加速します
- アプリケーションの敏捷性、柔軟性、およびスケーラビリティを向上させます

Citrix クラウドネイティブソリューションとは何ですか**?**

本番環境で Kubernetes を使用するメリットを最大化するには、Kubernetes をいくつかのツール、ベンダーソース、およびオープンソースコンポーネントと統合する必要があります。クラウドネイティブアプリケーションの本番

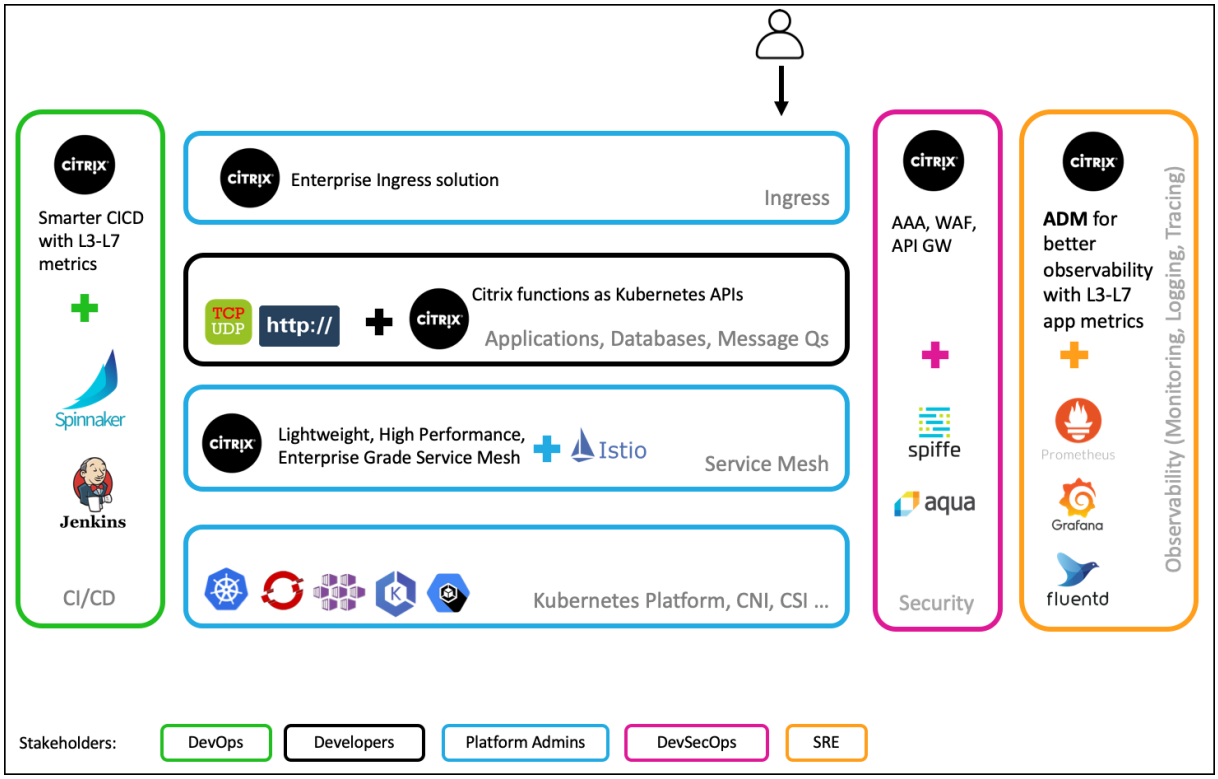
環境グレードの信頼性とセキュリティを確保することは、多くの組織が直面している課題です。
業界をリードする Citrix ADC のプロバイダーとして、Citrix は Kubernetes 本番環境の課題に対処するための Citrix クラウドネイティブソリューションを提供します。

Citrix クラウドネイティブソリューションは、Citrix ADC の高度なトラフィック管理、可観測性、および包括的なセキュリティ機能を活用して、エンタープライズグレードの信頼性とセキュリティを確保します。Kubernetes 環境のアプリケーショントラフィックを完全に可視化し、即座にフィードバックを提供し、アプリケーションのパフォーマンスに関する有意義な洞察を得るのに役立ちます。

次の表に、Ingress ソリューションを実装する際のさまざまな利害関係者の主な要件を示します。

利害関係者	職務権限	ニーズ
プラットフォーム管理者	Kubernetes クラスターの可用性を確保する	複数のクラスター、運用、およびプラットフォームのライフサイクル管理に展開されたアプリケーションを管理するためのより簡単な方法
DevOps	アプリケーションの本番環境への展開を加速します	との統合 CI/CD パイプライン、より迅速な展開のためのカナリアやブルーグリーンなどの展開手法のサポート
開発者	マイクロサービスの開発とテスト	トラフィックを Kubernetes クラスターに取り込む方法、トレースとデバッグ、アプリケーションのレート制限、アプリケーションの認証
SRE	サービスレベルアグリーメントを満たすためのアプリケーションの可用性を確保します	アプリケーションとインフラストラクチャ向けの高度なテレメトリ
SecOP	セキュリティコンプライアンスを確保する	安全な入力トラフィック、API 保護、Kubernetes クラスター内のマイクロサービス間の安全な通信のためのサービスメッシュ

次の図は、Citrix クラウドネイティブソリューションと、クラウドネイティブジャーニーで利害関係者が直面するさまざまな課題にどのように対処するかを説明しています。



Citrix クラウドネイティブソリューションには、次の主な利点があります。

- 開発者、SRE、devOps、ネットワークまたはクラスター管理者のニーズに応える高度な KubernetesIngress ソリューションを提供します。
- レガシーアプリケーションを Kubernetes 環境に移動する際に、TCP または UDP トラフィックに基づいて書き換える必要がなくなります。
- KubernetesAPI として公開されている Citrix ADC ポリシーを使用してアプリケーションを保護します。
- 南北トラフィックと East-West トラフィックの高性能マイクロサービスを展開するのに役立ちます。
- Citrix ADM サービスグラフを使用してすべてのマイクロサービスのオールインワンビューを提供します。
- TCP、UDP、HTTP、HTTPS、SSL などのさまざまな種類のトラフィックにわたるマイクロサービスのトラブルシューティングを高速化します。
- API を保護します。
- 自動化 CI/CD Canary デプロイメントのパイプライン。
- CNCF オープンソースツールとのすぐに使える統合を提供します。

Citrix クラウドネイティブソリューションのさまざまなコンポーネントの詳細については、次のリンクを参照してください。

- Kubernetes Ingress ソリューション
- サービスメッシュ
- 観測性のためのソリューション
- Kubernetes の API Gateway

Kubernetes Ingress ソリューション

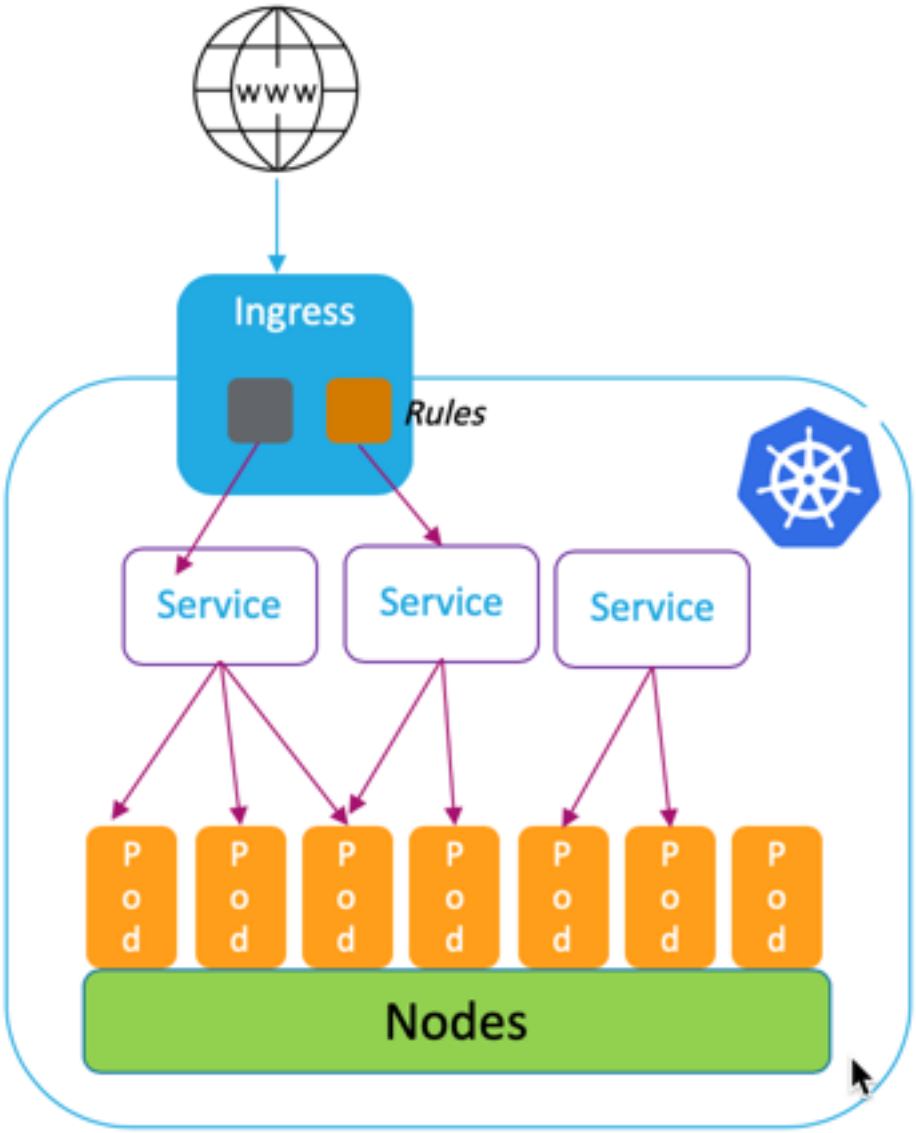
October 7, 2021

このトピックでは、Citrix が提供する Kubernetes Ingress ソリューションの概要と利点について説明します。

Kubernetes Ingress とは

Kubernetes クラスター内でアプリケーションを実行する場合、外部ユーザーが Kubernetes クラスターの外部からアプリケーションにアクセスする方法を提供する必要があります。Kubernetes は、安定した IP アドレスを使用して複数のサービスを公開する最も効果的な方法を提供する Ingress というオブジェクトを提供します。Kubernetes 入力オブジェクトは、常に 1 つ以上のサービスに関連付けられ、外部ユーザーがクラスター内で実行されているサービスにアクセスするための単一エントリポイントとして機能します。

次の図は、Kubernetes Ingress の動作を説明しています。



Kubernetes Ingress の実装は、次のコンポーネントで構成されています。

- 入力リソース。Ingress リソースを使用すると、クラスターの外部からアプリケーションにアクセスするためのルールを定義できます。
- 入力 **Controller** Ingress Controller は、Ingress で定義されているルールを解釈するクラスター内にデプロイされたアプリケーションです。Ingress Controller は、Ingress ルールをクラスタと統合した負荷分散アプリケーションの構成指示に変換します。ロードバランサーは、Kubernetes クラスター内で実行されるソフトウェアアプリケーション、またはクラスター外で実行されるハードウェアアプライアンスです。
- 入力デバイス。入力デバイスは、Citrix ADC CPX、VPX、MPX などの負荷分散アプリケーションで、入力 Controller が提供する構成指示に従って負荷分散を実行します。

Citrix の Kubernetes Ingress ソリューションは何ですか

このソリューションでは、Citrix ADC（Citrix ADC CPX、VPX、または MPX）を使用して Kubernetes クラスターにトラフィックを管理およびルーティングするために、Kubernetes Ingress Controller を実装しています。Citrix ingress controller は、Citrix ADC を Kubernetes 環境と統合し、イングレスルールに従って Citrix ADC CPX、VPX、または MPX を構成します。

標準的な Kubernetes Ingress ソリューションは、レイヤ 7（HTTP または HTTPS トラフィック）でのみロードバランシングを提供します。場合によっては、TCP、UDP、またはアプリケーションに依存する多くのレガシーアプリケーションを公開し、それらのアプリケーションの負荷分散の方法が必要になる場合があります。Citrix Kubernetes 入力ソリューションは、標準の HTTP または HTTPS 入力とは別に、TCP、TCP、SSL、および UDP トラフィックをサポートします。また、複数のクラウドまたはオンプレミスのデータセンター間でシームレスに動作します。

Citrix ADC は、書き換えポリシーやレスポンダーポリシーなどのエンタープライズグレードのトラフィック管理ポリシーを提供し、レイヤー 7 でトラフィックを効率的に負荷分散します。ただし、Kubernetes Ingress には、このようなエンタープライズレベルのトラフィック管理ポリシーがありません。Citrix の Kubernetes Ingress ソリューションを使用すると、Citrix が提供する CRD を使用して、Kubernetes 環境のアプリケーショントラフィックにリライトポリシーとレスポンダポリシーを適用できます。

Citrix の Kubernetes Ingress ソリューションは、CI/CD アプリケーションパイプラインの自動カナリア展開もサポートしています。このソリューションでは、Citrix ADC は Spinnaker プラットフォームと統合され、Kayenta を使用して Canary 展開を分析するための正確なメトリックを提供するソースとして機能します。メトリクスを分析した後、Kayenta はカナリアの集計スコアを生成し、カナリアバージョンを昇格または失敗することを決定します。Citrix ADC ポリシーインフラストラクチャを使用して、Canary バージョンへのトラフィック配信を規制することもできます。

次の表は、Citrix の Ingress ソリューションが Kubernetes Ingress よりも提供する利点をまとめたものです。

機能	Kubernetes Ingress	Citrix からの進入ソリューション
HTTP および HTTPS のサポート	はい	はい
URL ルーティング	はい	はい
TLS	はい	はい
負荷分散	はい	はい
TCP、TCP-SSL	いいえ	はい
UDP	いいえ	はい
HTTP/2	はい	はい
CI/CD ツールによるカナリア導入の自動サポート	いいえ	はい

機能	Kubernetes Ingress	Citrix からの進入ソリューション
Citrix ADC のリライトポリシーとレスポンダー ポリシーの適用のサポート	いいえ	はい
認証（オープン認証（OAuth）、相互 TLS（MTL））	いいえ	はい
Citrix のレート制限ポリシーの適用のサポート	いいえ	はい

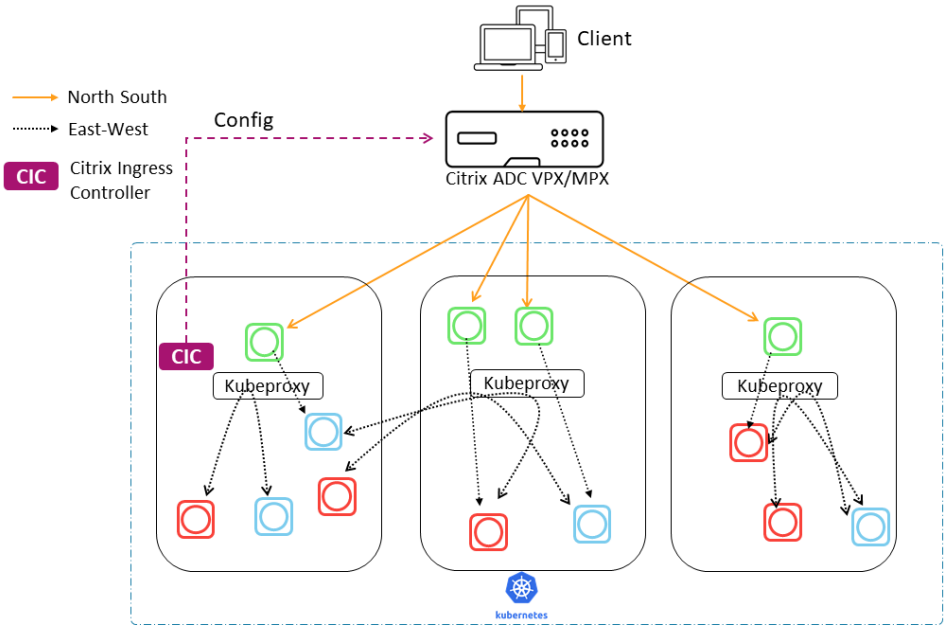
Kubernetes Ingress ソリューションの展開オプション

Citrix の Kubernetes Ingress ソリューションは、Citrix ADC および Kubernetes 環境の管理方法に応じて、柔軟なアーキテクチャを提供します。

統合入力（単一層）

統合された Ingress（単一層）アーキテクチャでは、Kubernetes クラスタの外に展開された Citrix MPX または VPX デバイスは、Citrix ingress controller を使用して Kubernetes 環境と統合されます。Citrix ingress controller は、Kubernetes クラスタにポッドとしてデプロイされ、マイクロサービスまたは Ingress リソースの変更に基づいて Citrix ADC の構成を自動化します。Citrix ADC デバイスは、インバウンドトラフィックに対して負荷分散、TLS 終了、HTTP または TCP プロトコルの最適化などの機能を実行し、Kubernetes クラスタ内の正しいマイクロサービスにトラフィックをルーティングします。このアーキテクチャは、同じチームが Kubernetes プラットフォームと、アプリケーション配信コントローラ (ADC) を含むその他のネットワークインフラストラクチャを管理するシナリオに最適です。

次の図は、Unified Ingress アーキテクチャを使用したデプロイメントを示しています。



統合された Ingress ソリューションには、次のような主な利点があります。

- 既存の Citrix ADC インフラストラクチャの機能を Kubernetes 環境に拡張する方法を提供します
- インバウンドトラフィックにトラフィック管理ポリシーを適用できます。
- ネットワークに精通した DevOps チームに適したシンプルなアーキテクチャを提供
- マルチテナンシーをサポート

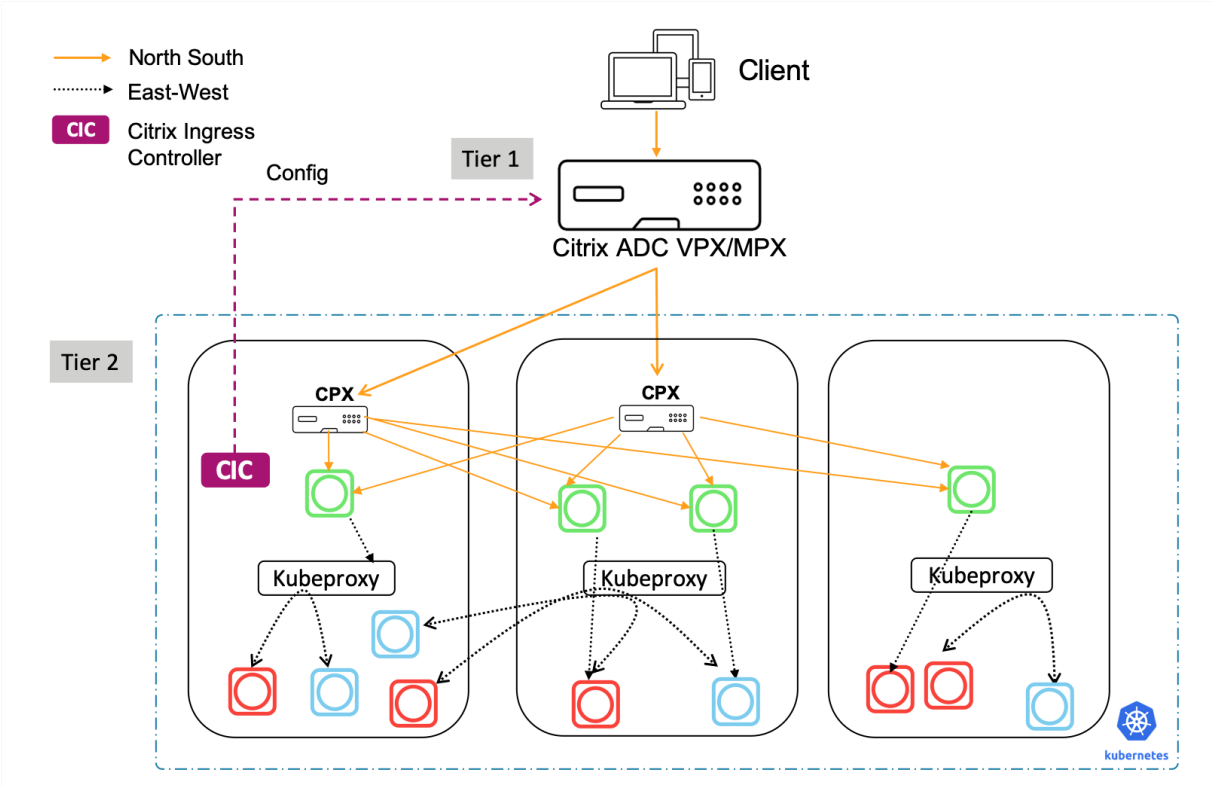
デュアル層入力

2 層アーキテクチャでは、Kubernetes クラスタの外側に展開された Citrix ADC（MPX または VPX）はティア 1 で動作し、クラスタ内で動作する Citrix ADC CPX への南北トラフィックの負荷分散を行います。Citrix ADC CPX は階層 2 で動作し、Kubernetes クラスタ内のマイクロサービスの負荷分散を実行します。

個別のチームが Kubernetes プラットフォームとネットワークインフラストラクチャを管理するシナリオでは、デュアルティアアーキテクチャが最適です。

ネットワークチームは、GSLB、ハードウェアプラットフォームでの TLS 終了、TCP 負荷分散などのユースケースに Tier1 の Citrix ADC を使用します。Kubernetes プラットフォームチームは、レイヤー 7（HTTP/HTTPS）負荷分散、相互 TLS、およびマイクロサービスの監視や監視のために、階層 2 の Citrix ADC（CPX）を使用できます。階層 2 の Citrix ADC（CPX）では、階層 1 の Citrix ADC とは異なるソフトウェアリリースバージョンを使用して、新しく利用可能な機能に対応できます。

次の図は、2 層アーキテクチャを使用した展開を示しています。



2 層の Ingress には、次の主な利点があります。

- 開発者またはプラットフォームチーム向けの高速アプリケーション開発を確保
- Kubernetes クラスター内のマイクロサービスに対して、開発者主導のトラフィック管理ポリシーを適用できるようにします。
- クラウドの拡張とマルチテナンシーを実現

詳細については、Citrix ingress controller ドキュメントを参照してください。

はじめに

Citrix の Kubernetes Ingress ソリューションを開始するには、次の例を試すことができます。

- Minikube の Citrix ADC CPX による入力トラフィックの負荷分散
- Citrix ADC CPX プロキシを使用した南北方向の入力トラフィックの負荷分散
- Citrix ADC CPX プロキシを使用した East-West マイクロサービストラフィックの負荷分散
- Citrix ADC CPX で Kubernetes 機能を詳しく知る

サービスメッシュ

October 7, 2021

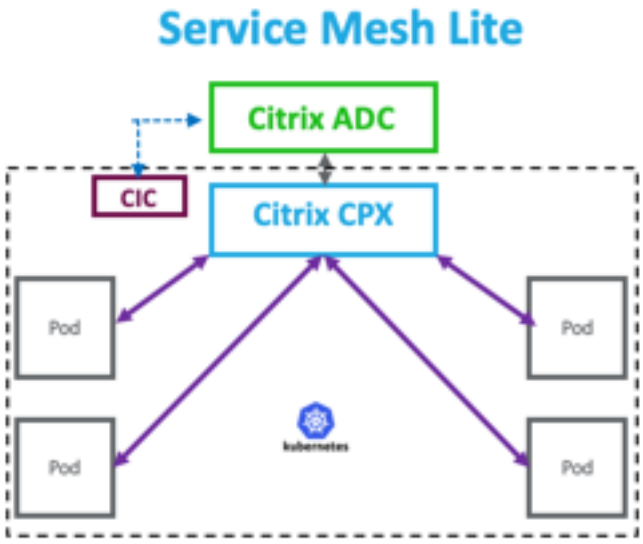
サービスメッシュは、API を使用してクラウドネイティブアプリケーションのサービス間通信を処理するためのインフラストラクチャレイヤーです。マイクロサービスを接続、保護、および監視する方法を提供します。Citrix では、サービスメッシュ要件を満たすために 2 つのソリューションを提供しています。

- サービスメッシュライト
- サービスメッシュ（Citrix ADC と Iistio との統合）

サービスメッシュライト

本格的なサービスメッシュの実装は複雑で、急な学習曲線が必要です。同様の利点を持つサービスメッシュを簡単に実装したい場合は、Citrix は、複雑さの低いサービスメッシュライトと呼ばれるソリューションを提供しています。このソリューションでは、Citrix ADC CPX は Kubernetes クラスタ内の集中型ロードバランサーとして実行され、マイクロサービス間で East-West トラフィックを負荷分散します。Citrix ADC CPX は、インバウンドおよびコンテナー間のトラフィックに対してポリシーを適用します。

次の図は、サービスメッシュ lite アーキテクチャを示しています。



詳細については、サービスメッシュ Lite のドキュメントを参照してください。

サービスメッシュ（**Citrix ADC** と **Iistio** との統合）

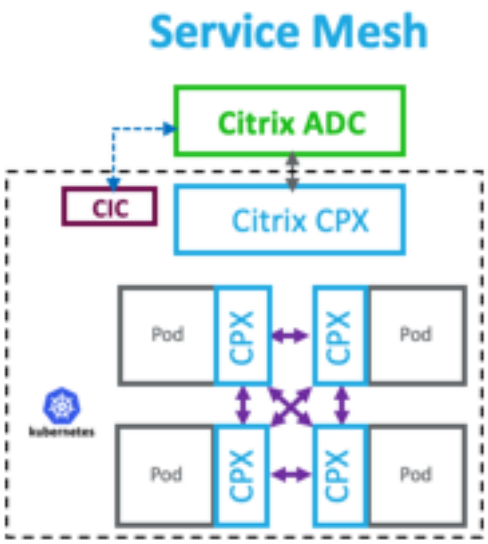
シトリックスは、Citrix ADC と Iistio を統合することにより、サービスメッシュソリューションを提供します。オープンソースでプラットフォームに依存しないサービスメッシュである Itio は、最も一般的なサービスメッシュ実装の 1 つです。Citrix ADC と Iistio を統合することで、Citrix ADC 機能を活用して、サービスメッシュ内のアプリケーションのトラフィックを保護および最適化できます。

Citrix ADC は、次の方法で Iistio と統合できます。

- Citrix ADC MPX、VPX、または CPX をサービスメッシュへの I スティオ入力ゲートウェイとして使用して、Kubernetes クラスタにトラフィックを公開します。
- アプリケーション間の通信を制御するために、サービスメッシュにアプリケーションコンテナを持つサイドカープロキシとしての Citrix ADC CPX。

統合を個別に使用することも、両方の方法を組み合わせて統合データプレーンソリューションを使用することもできます。

次の図は、サービスメッシュアーキテクチャを示しています。



Service Mesh は、安全性の高いアプリケーションに最適で、次のような利点もあります。

- コンテナごとにきめ細かい（モジュール化された）トラフィック管理を提供
- サイドカーの実装により、より豊かな観察性、分析、セキュリティ（相互 TLS）を保証
- 組み込みの Citrix ADC CPX を使用して、コンテナごとにカナリア展開を自動化
- クラウドの移植性をサポート
- アプリケーションによって実行される機能のいくつかをサイドカーにオフロードできます。
- サイドカーレイテンシを低減
- オープンソースのツールとの統合を提供
- 拡張性を提供

詳細については、Citrix ADC と Istio の統合に関するドキュメントを参照してください。

観測性のためのソリューション

August 12, 2022

マイクロサービスベースのアーキテクチャでは、サービス間の通信を可視化することは、効率的で耐障害性の高いアーキテクチャを構築するために不可欠です。従来のロギングと監視の方法では、マイクロサービスアーキテクチャの課題に対処できません。Citrix の監視ソリューションを使用すると、サービスが相互にやり取りしたときに何が起こっているのかを確認し、システムに関する有意義な洞察を得ることができます。

Citrix は、マイクロサービスアーキテクチャの観測可能性のニーズに対応するために、以下のソリューションを提供しています。

- Citrix ADM サービスグラフと分析
- Citrix ADC 観測性エクスポーター

Citrix ADM サービスグラフと分析

Citrix Application Delivery Management（ADM） は、複数のインスタンスで実行する必要がある管理ジョブを企業全体で可視化し、自動化できる一元管理ソリューションです。

マイクロサービスアーキテクチャでは、単一のエンドユーザー要求が複数のマイクロサービスにまたがる可能性があるため、トラブルシューティングは困難です。

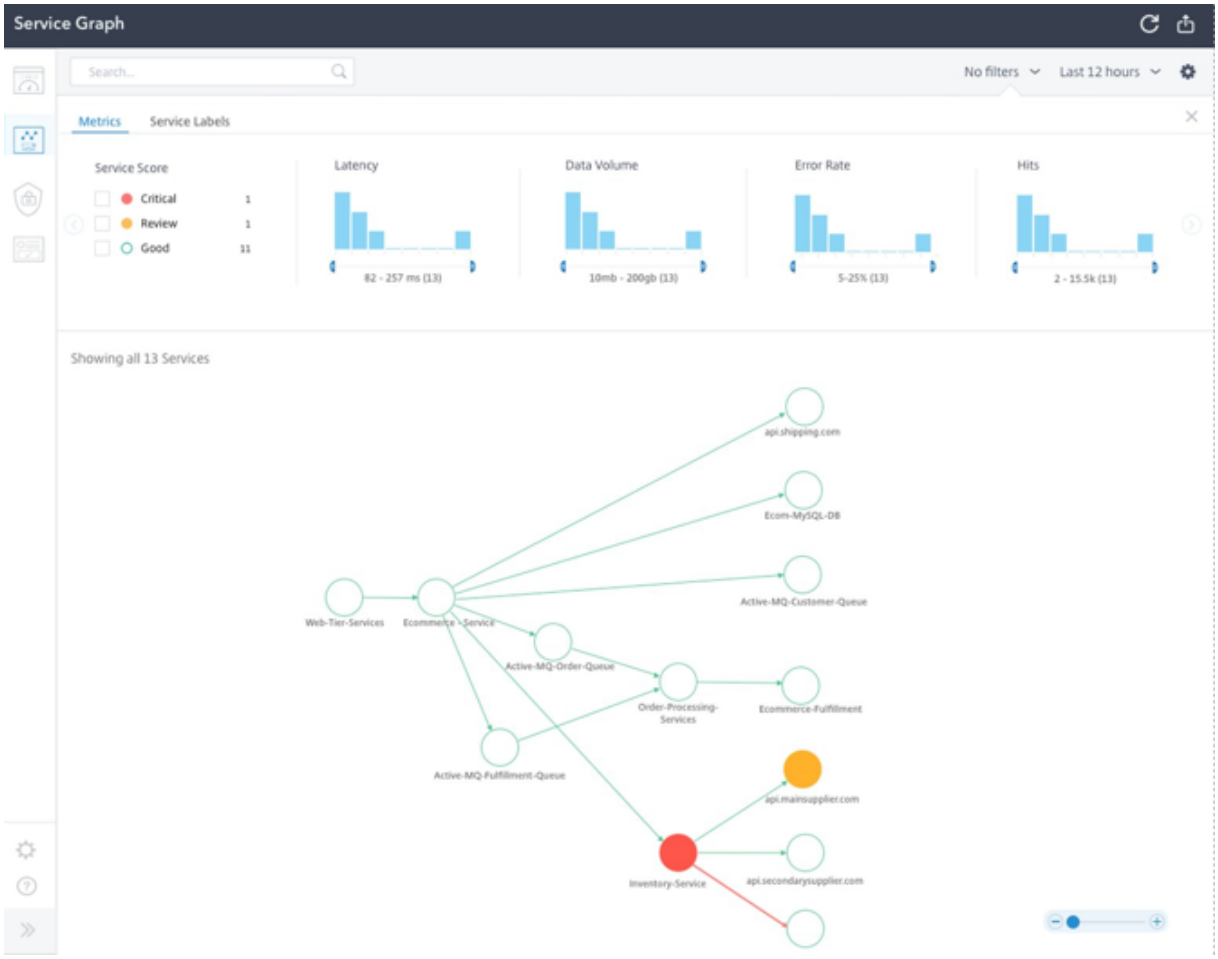
Citrix ADM のサービスグラフと分析は、マイクロサービス間の相互作用を可視化し、レイテンシーや HTTP エラーなどのさまざまなメトリックに基づいて問題を特定して修正するのに役立ちます。

また、Citrix ADM は、Citrix ADC から収集されたメトリックとトランザクションログに基づいて、高度な分析を提供します。

Citrix ADM ソリューションには次の利点があります。

- コンテナ、オンプレミス、クラウドにまたがるアプリケーションに、単一のパネルを提供
- マイクロサービスの監視可能性を向上させ、トラブルシューティングを高速化
- カナリア展開をサポート

次の図は、複数のマイクロサービスを含むアプリケーションのサンプルサービスグラフを示しています。



Citrix ADM サービスグラフと分析の設定方法の詳細については、[サービスグラフとアナリティクスのドキュメントを参照してください](/ja-jp/citrix-application-delivery-management-service/analytics.html) 。

Citrix ADC 観測性エクスポーター

Citrix ADC の観測性エクスポータは、Citrix ADC からメトリックとトランザクションを収集し、サポートされているエンドポイントに適した形式（JSON、AVRO など）に変換するコンテナです。Citrix ADC オブザーバビリティエクスポータによって収集されたデータを、目的のエンドポイントにエクスポートできます。データを分析することで、Citrix ADC によってプロキシされるアプリケーションについて、マイクロサービスレベルで貴重な洞察を得ることができます。

分散トレースのサポート

分散トレーサを使用すると、マイクロサービス間のデータフローを視覚化し、マイクロサービスアーキテクチャのボトルネックを特定するのに役立ちます。OpenTracing は、分散トレースを設計および実装するための API の仕様および標準セットです。

Citrix の観察性エクスポータは、Citrix ADC の分散トレースを実装しており、現在、分散トレーサとして Zipkin をサポートしています。

Zipkin で Elasticsearch と Kibana を使用すると、トレース解析を強化できます。Elasticsearch は、トレースデータを長期的に保持します。Kibana では、ログメッセージの探索と視覚化を行うツールを提供することで、データに対するより深い洞察を得ることができます。

トランザクション収集とストリーミングのサポート

Citrix ADC 監視性エクスポータは、トランザクションの収集とエンドポイントへのストリーミングをサポートしています。現在、Citrix ADC 観測性エクスポータは、トランザクションエンドポイントとして Elasticsearch と Kafka をサポートしています。

詳細については、Citrix ADC 監視性エクスポータのドキュメントを参照してください。

Citrix ingress controller YAML ファイルの注釈を使用した分析の有効化

Ingress または LoadBalancer 設定のサービスでスマートアノテーションとして定義されている分析プロファイルを使用して、分析を有効にできます。監視する必要がある特定のパラメータを定義するには、アプリケーションの Ingress またはサービス構成でパラメータを指定します。アノテーションを使用したアナリティクスの有効化の詳細については、「アノテーションを使用したアナリティクス」を参照してください。

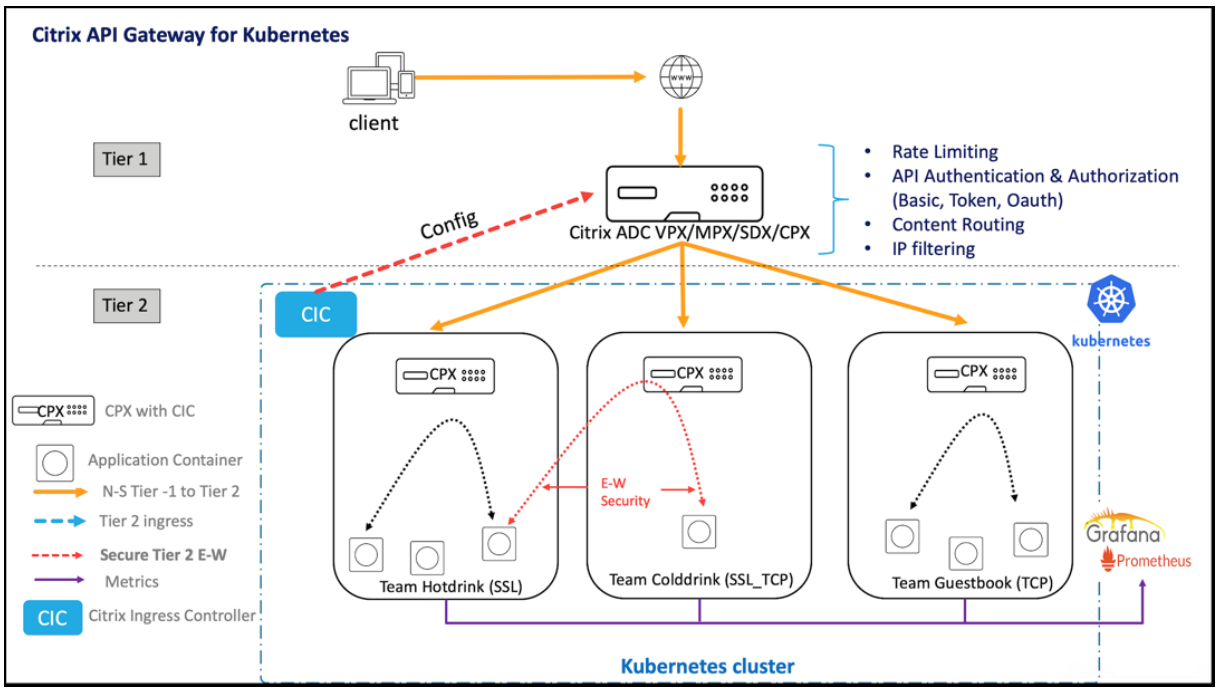
Kubernetes の API Gateway

October 7, 2021

API Gateway は、API の単一のエントリポイントとして機能し、システム内の複数の API およびマイクロサービスへの安全で信頼性の高いアクセスを保証します。

Citrix は、Kubernetes クラスターへの南北 API トラフィック用のエンタープライズグレードの APIGateway を提供します。
この API Gateway は、Citrix 入力 Controller と、オンプレミスまたはクラウド展開用の入力ゲートウェイとして展開された Citrix ADC（Citrix ADC MPX、VPX、または CPX）を介して Kubernetes と統合されます。

次の図は、API Gateway の 2 層トポロジを示しています。



Citrix が提供する APIGateway を使用すると、次の機能を実行できます。

- 認証ポリシーの適用
- サービスへのレート制限アクセス
- 高度なコンテンツルーティング
- リライトポリシーとレスポンダーポリシーを使用した HTTP トランザクションの柔軟で包括的な変換
- Web アプリケーションファイアウォールポリシーを適用する

API Gateway の仕組み

API ゲートウェイは、Citrix Ingress ゲートウェイの上に構築され、カスタムリソース定義（CRD）などの Kubernetes API 拡張機能を使用します。CRD を使用すると、同じインスタンスで Citrix ADC と API Gateway を自動的に構成できます。

Citrix は、APIGateway 用に以下の CRD を提供しています。

- 認証 CRD
- レート制限 CRD
- コンテンツルーティング CRD
- 書き換えとレスポンダー CRD
- WAF CRD

API Gateway を使用する主な利点

Citrix が提供する APIGateway の主な利点は次のとおりです。

- Citrix ADC の高度なトラフィック管理と包括的なセキュリティ機能を使用します。
- 複数のネットワーク機能を Citrix 入力 Gateway の 1 つのコンポーネントに統合することで、展開を最適化します。
- 複数のコンポーネントの導入に伴う運用の複雑さとコストを軽減します。
- 別々のコンポーネントを使用しながら、TCP または TLS 復号化の複数のホップを減らすことで、アプリケーショントラフィックのパフォーマンスを向上させます。
- YAML またはヘルムチャートを直接使用することで、Kubernetes 環境でのデプロイと統合を簡素化します。

API Gateway のデプロイ

CRD を使用して APIGateway 機能を構成する方法の詳細については、Citrix 入力 Controller ドキュメントを参照してください。

- 認証
- レート制限
- 高度なコンテンツルーティング
- ポリシーの書き換えと対応
- Web アプリケーションファイアウォールポリシー

Citrix ADM を使用して **Citrix** クラウドネイティブネットワークのトラブルシューティングを行う

February 21, 2022

概要

このドキュメントでは、Citrix ADM を使用して Kubernetes マイクロサービスアプリケーションを配信および監視する方法について説明します。また、CLI、サービスグラフ、トレースを使用して、プラットフォームと SRE チームがトラブルシューティングを行えるようにする方法についても説明します。

アプリケーションパフォーマンスとレイテンシーの概要

TLS 暗号化

TLS は、インターネット通信を保護するために設計された暗号化プロトコルです。TLS ハンドシェイクは、TLS 暗号化を使用する通信セッションを開始するプロセスです。TLS ハンドシェイク中、2 つの通信側はメッセージを交換して、相互の確認、相互の検証、使用する暗号化アルゴリズムの確立、セッションキーの合意を行います。TLS ハンドシェイクは、HTTPS の仕組みの基本的な部分です。

TLS と SSL ハンドシェイク

SSL (セキュア・ソケット・レイヤー) は、HTTP 用に開発されたオリジナルの暗号化プロトコルでした。TLS（トランスポート層セキュリティ）はしばらく前に SSL に取って代わりました。SSL ハンドシェイクは TLS ハンドシェイクと呼ばれるようになりましたが、「SSL」名は今でも広く使われています。

TLS ハンドシェイクはいつ発生しますか。

TLS ハンドシェイクは、ユーザーが HTTPS 経由で Web サイトに移動し、ブラウザーが最初に Web サイトのオリジンサーバーへのクエリを開始したときに実行されます。TLS ハンドシェイクは、API 呼び出しや DNS over HTTPS クエリなど、他の通信が HTTPS を使用する場合にも発生します。

TLS ハンドシェイクは、TCP ハンドシェイクを介して TCP 接続が開かれた後に発生します。

TLS ハンドシェイク中には何が起こりますか?

- TLS ハンドシェイク中、クライアントとサーバは共に次の処理を行います。
  - 使用する TLS のバージョン (TLS 1.0、1.2、1.3 など) を指定します。
  - 使用する暗号スイート (次のセクションを参照) を決定します。
  - サーバーの公開鍵と SSL 認証局のデジタル署名を使用して、サーバーの ID を認証します。
  - ハンドシェイクが完了したら、対称暗号化を使用するセッションキーを生成します。

TLS ハンドシェイクの手順を教えてください。

- TLS ハンドシェイクは、クライアントとサーバーによって交換される一連のデータグラム、つまりメッセージです。TLS ハンドシェイクには複数の手順が伴います。これは、クライアントとサーバーがハンドシェイクを完了し、さらに会話を可能にするために必要な情報を交換するためです。

TLS ハンドシェイク内の正確な手順は、使用する鍵交換アルゴリズムの種類と、両者がサポートする暗号スイートによって異なります。RSA 鍵交換アルゴリズムが最もよく使用されます。それは次のようになります。

1. `client hello` メッセージ: クライアントは「hello」メッセージをサーバーに送信してハンドシェイクを開始します。このメッセージには、クライアントがサポートしている TLS バージョン、サポートされている暗号スイート、「クライアントランダム」と呼ばれるランダムなバイト文字列が含まれます。
2. `server hello` メッセージ: クライアントの hello メッセージへの応答として、サーバーは、サーバーの SSL 証明書、サーバーが選択した暗号スイート、およびサーバーによって生成された別のランダムなバイト文字列である「server random」を含むメッセージを送信します。
3. 認証: クライアントは、サーバーの SSL 証明書を、その証明書を発行した認証局と照合します。これにより、サーバーが本人であり、クライアントがドメインの実際の所有者とやり取りしていることが確認されます。
4. premaster secret: クライアントは、ランダムなバイト文字列「premaster secret」をもう 1 つ送信します。premaster シークレットは公開鍵で暗号化され、サーバーは秘密鍵でのみ復号できます。クライアントはサーバーの SSL 証明書から公開鍵を取得します。)
5. 使用した秘密鍵: サーバーはプレマスターシークレットを復号化します。
6. 作成されたセッションキー: クライアントとサーバーの両方が、クライアントランダム、サーバーランダム、およびプレマスターシークレットからセッションキーを生成します。彼らは同じ結果になるはずです。
7. Client is ready: クライアントは、セッションキーで暗号化された「完了」メッセージを送信します。
8. サーバー準備完了: サーバーは、セッションキーで暗号化された「完了」メッセージを送信します。

9. 安全な対称暗号化を実現: ハンドシェイクが完了し、セッションキーを使用して通信が継続されます。

すべての TLS ハンドシェイクは非対称暗号化 (公開鍵と秘密鍵) を使用しますが、セッション鍵を生成する過程で秘密鍵を使用するわけではありません。たとえば、エフェメラルな Diffie-Hellman ハンドシェイクは次のように処理されます。

1. Client hello: クライアントは、プロトコルバージョン、クライアントランダム、および暗号スイートのリストを含むクライアント hello メッセージを送信します。
2. Server hello: サーバーは、SSL 証明書、選択した暗号スイート、およびサーバーランダムで応答します。前のセクションで説明した RSA ハンドシェイクとは対照的に、このメッセージにはサーバに次の内容も含まれています（ステップ 3)。
3. サーバーのデジタル署名: サーバーは秘密鍵を使用して、クライアントランダム、サーバーランダム、および DH パラメーター * を暗号化します。この暗号化されたデータはサーバーのデジタル署名として機能し、SSL 証明書の公開キーと一致する秘密キーがサーバーにあることを証明します。
4. デジタル署名の確認: クライアントは、公開鍵を使用してサーバーのデジタル署名を復号化し、サーバーが秘密鍵を制御していること、および本人であることを確認します。Client DH パラメータ: クライアントは DH パラメータをサーバに送信します。
5. クライアントとサーバがプレマスターシークレットを計算する： RSA ハンドシェイクのように、クライアントがプレマスターシークレットを生成してサーバに送信する代わりに、クライアントとサーバは交換した DH パラメータを使用して、一致するプレマスターシークレットを個別に計算します。
6. 作成されたセッションキー: クライアントとサーバーは、RSA ハンドシェイクと同様に、プレマスターシークレット、クライアントランダム、およびサーバーランダムからセッションキーを計算するようになりました。
   - クライアントの準備完了:RSA
     ハンドシェイクと同じ
   - サーバは準備完了です
   - セキュアな対称暗号化を実現

   *DH パラメーター:DH は Diffie-Hellman の略です。Diffie-Hellman アルゴリズムは、指数計算を使用して同じ premaster シークレットに到達します。サーバーとクライアントはそれぞれ計算用のパラメーターを提供し、これらを組み合わせると、両側で異なる計算が行われ、結果は等しくなります。

エフェメラルな Diffie-Hellman ハンドシェイクと他の種類のハンドシェイクの対比、およびこれらがどのように前方秘匿性を実現するかについての詳細は、この TLS プロトコルのドキュメントを参照してください。

暗号スイートって何ですか?

- 暗号スイートは、セキュアな通信接続を確立するために使用する暗号化アルゴリズムのセットです。暗号化アルゴリズムとは、データをランダムに見せるためにデータに対して実行される一連の数学的演算です。さまざまな暗号スイートが広く使用されており、TLS ハンドシェイクの重要な部分は、そのハンドシェイクにどの暗号スイートを使用するかを合意することです。

開始するには、「リファレンス: TLS プロトコルのドキュメント」を参照してください。

Citrix Application Delivery Management SSL ダッシュボード

Citrix Application Delivery Management（ADM）により、証明書管理のあらゆる側面が合理化されるようになりました。1 つのコンソールから、使われていない、または期限切れが近い証明書のタブは閉じたまま、正しい発行者、キーの強度、および正しいアルゴリズムを確保する自動化されたポリシーを作成することができます。Citrix ADM の SSL ダッシュボードとその機能の使用を開始するには、SSL 証明書とは何か、および Citrix ADM を使用して SSL 証明書を追跡する方法を理解する必要があります。

SSL トランザクションの一部であるセキュアソケットレイヤー (SSL) 証明書は、企業 (ドメイン) または個人を識別するデジタルデータフォーム (X509) です。この証明書には、サーバーとの安全なトランザクションを開始しようとするすべてのクライアントが確認できる公開キーコンポーネントが含まれます。対応する秘密キーは、Citrix Application Delivery Controller（ADC）アプライアンスに安全に配置され、非対称キー（または公開キー）の暗号化と復号化を完了するために使用されます。

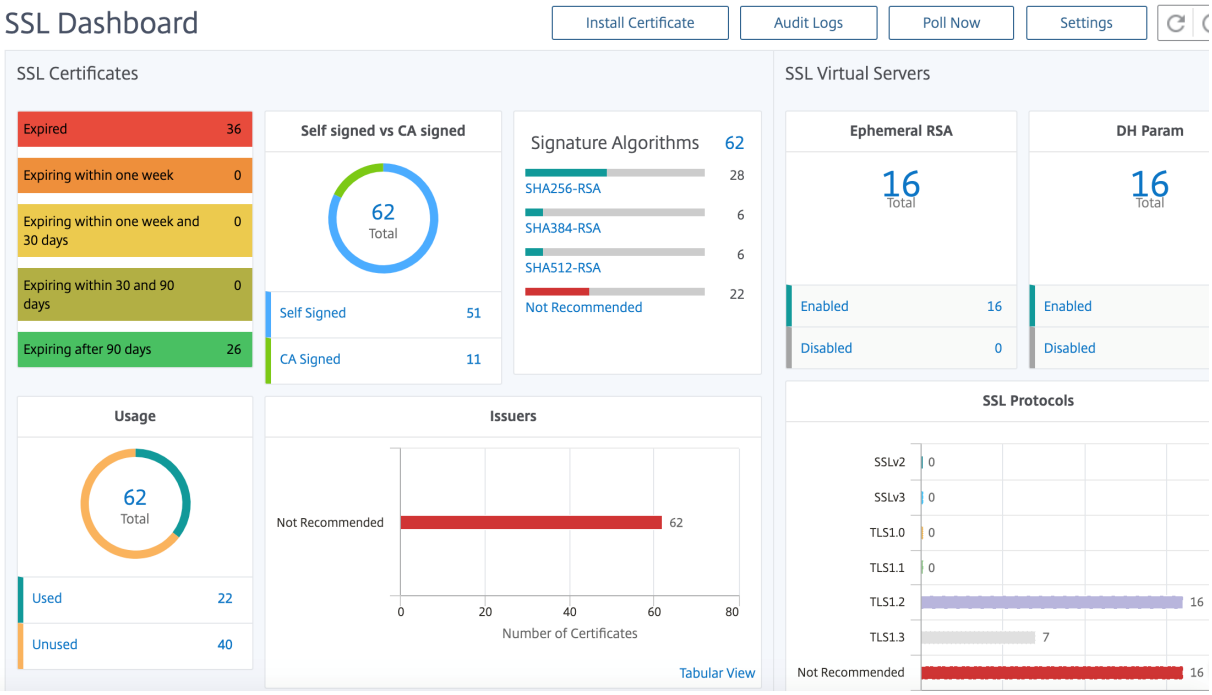
SSL 証明書およびキーは、次のいずれかの方法で入手できます。

- 認可された認証局 (CA) から
- Citrix ADC アプライアンスで新しい SSL 証明書とキーを生成する

Citrix ADM は、すべての管理対象 Citrix ADC インスタンスにインストールされた SSL 証明書を一元的に表示します。SSL Dashboard では、証明書の発行者、キーの強度、署名アルゴリズム、期限切れまたは未使用の証明書などを追跡するのに役立つグラフを表示できます。また、仮想サーバーで実行されている SSL プロトコルの分布および各サーバーで有効化されているキーも確認できます。

また、証明書の有効期限が近づいたときに通知を設定し、その証明書を使用する Citrix ADC インスタンスに関する情報を含めることもできます。

Citrix ADC インスタンスの証明書を CA 証明書にリンクできます。ただし、同じ CA 証明書にリンクする証明書のソースと発行元が同じであることを確認してください。証明書を CA 証明書にリンクしたら、それらのリンクを解除できます。



開始するには、SSL ダッシュボードのドキュメントを参照してください。

サード・パーティとの連携

アプリケーションのレイテンシーはミリ秒単位で測定され、使用するメトリクスに応じて 2 つのうちの 1 つを示すことができます。レイテンシを測定する一般的な方法は「ラウンドトリップ時間」(RTT) と呼ばれます。RTT は、データパケットがネットワーク上のある地点から別の地点に移動して、応答が送信元に返されるまでにかかる時間を計算します。もう 1 つは「最初のバイトまでの時間」(Time to first byte) (または TTFB) と呼ばれ、パケットがネットワーク上のある地点を出発してから宛先に到着するまでにかかる時間を記録します。RTT は、ネットワーク上の 1 つのポイントから実行でき、(TTFB のように) データ収集ソフトウェアを宛先ポイントにインストールする必要がないため、レイテンシの測定によく使用されます。

ADM サービスでは、アプリケーションの帯域幅の使用量とパフォーマンスをリアルタイムで監視することで、問題を簡単に特定し、潜在的な問題が顕在化してネットワーク上のユーザーに影響を及ぼす前に先制的に対処できます。このフローベースのソリューションは、インターフェイス、アプリケーション、カンバセーションごとに使用状況を追跡し、ネットワーク全体のアクティビティに関する詳細情報を提供します。

Splunk ツールを使う

インフラストラクチャとアプリケーションのパフォーマンスは相互に依存しています。全体像を把握するために、SignalFX はクラウドインフラストラクチャとその上で実行されるマイクロサービスとのシームレスな相関関係を提供します。メモリリーク、ノイズの多いネイバーコンテナー、その他のインフラストラクチャ関連の問題が原因でアプリケーションが動作した場合、SignalFX から通知されます。全体像を把握するために、コンテキスト内で Splunk のログとイベントにアクセスすることで、より詳細なトラブルシューティングと根本原因の分析が可能になります。

SignalFX マイクロサービス APM と Splunk によるトラブルシューティングの詳細については、DevOps 向け Splunk の情報をご覧ください 。

MongoDB サポート

MongoDB は、柔軟な JSON に似たドキュメントにデータを格納します。つまり、フィールドはドキュメントごとに異なり、データ構造は時間の経過とともに変化する可能性があります。

ドキュメントモデルはアプリケーションコード内のオブジェクトにマップされるため、データの操作が容易になります。

オンデマンドクエリ、インデックス作成、リアルタイム集約により、データにアクセスして分析するための強力な方法が提供されます。

MongoDB は中核をなす分散データベースであるため、高可用性、水平スケーリング、地理的分散が組み込まれており、使いやすくなっています。

MongoDB は、以下を実現するテクノロジー基盤により、最新のアプリケーションの要求を満たすように設計されています。

- ドキュメントデータモデル — データを操作する最適な方法を提供します。
- 分散システム設計 — データを必要な場所にインテリジェントに配置できます。
- どこにいても自由に実行できる統合エクスペリエンスにより、将来を見据えた作業が可能になり、ベンダーロックインを排除できます。

これらの機能により、MongoDB に支えられたインテリジェントな運用データプラットフォームを構築できます。詳細については、MongoDB のドキュメントを参照してください。

Ingress トラフィックを **TCP** または **UDP** ベースのアプリケーションに負荷分散する方法

Kubernetes 環境では、Ingress は Kubernetes クラスターの外部から Kubernetes サービスへのアクセスを許可するオブジェクトです。標準の Kubernetes Ingress リソースは、すべてのトラフィックが HTTP ベースであり、TCP、TCP-SSL、UDP などの HTTP ベース以外のプロトコルには対応していないと想定しています。したがって、DNS、FTP、LDAP などの L7 プロトコルに基づく重要なアプリケーションは、標準の Kubernetes Ingress を使用して公開することはできません。

Kubernetes の標準ソリューションは、LoadBalancer タイプのサービスを作成することです。詳細については、Citrix ADC のサービスタイプ LoadBalancer を参照してください 。

2 番目のオプションは、Ingress オブジェクトに注釈を付けることです。Citrix ingress controller を使用すると、TCP または UDP ベースの Ingress トラフィックの負荷を分散できます。Kubernetes Ingress リソース定義で以下のアノテーションを使用して 、TCP または UDP ベースの Ingress トラフィックの負荷を分散できます。

- ingress.citrix.com/insecure-service-type: このアノテーションにより、Citrix ADC プロトコルとして TCP、UDP、または ANY を使用した L4 負荷分散が可能になります。
- ingress.citrix.com/insecure-port: アノテーションは TCP ポートを構成します。このアノテーションは、非標準ポートでマイクロサービスアクセスが必要な場合に役立ちます。デフォルトでは、ポート 80 が設定されています。

詳細については、「Ingress トラフィックを TCP または UDP ベースのアプリケーションに負荷分散する方法」を参照してください。

TCP または **UDP** ベースのアプリケーションのパフォーマンスを監視し、改善する

アプリケーション開発者は、Citrix ADC のリッチモニター（TCP-ECV、UDP-ECV など）を使用して、TCP または UDP ベースのアプリケーションの状態を綿密に監視できます。ECV (拡張コンテンツ検証) モニターは、アプリケーションが予期したコンテンツを返しているかどうかを確認するのに役立ちます。

また、ソース IP などの永続化方法を使用することで、アプリケーションのパフォーマンスを向上させることができます。これらの Citrix ADC 機能は、Kubernetes のスマートアノテーションを通じて使用できます 。その一例を以下に挙げます。

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
    name: mongodb
    annotations:
        ingress.citrix.com/insecure-port:  “80”
        ingress.citrix.com/frontend-ip:  “192.168.1.1”
        ingress.citrix.com/csvserver:  ‘{
  “l2conn” :” on”  }
  ’
```



```
        ingress.citrix.com/lbvserver:  '{
  "mongodb-svc" :{
  "lbmethod" :" SRCIPDESTIPHASH"  }
  }
 '
        ingress.citrix.com/monitor:  '{
  "mongodbsvc" :{
  "type" :" tcp-ecv"  }
  }
 '
Spec:
    rules:
    - host: mongodb.beverages.com
      http:
        paths:
        - path: /
          backend:
            serviceName: mongodb-svc
            servicePort: 80
<!--NeedCopy-->
```

Citrix Application Delivery Management (ADM) サービス

Citrix ADM サービスには次の利点があります。

- 機敏性 — 運用、更新、使用が容易 Citrix ADM Service のサービスモデルはクラウド経由で利用できるため、提供される機能の操作、更新、および使用が簡単に行えます。更新の頻度と自動更新機能の組み合わせにより、Citrix ADC 展開が迅速に強化されます。
- タイム・ツ・バリューの短縮 — ビジネス目標の達成を迅速化従来のオンプレミス展開とは異なり、Citrix ADM Service は数回クリックするだけで使用できます。インストールと設定の時間を節約できるだけでなく、潜在的なエラーに時間とリソースを浪費することもありません。
- マルチサイト管理 — 複数のサイトデータセンターにまたがるインスタンスを 1 つのガラスで管理できます。Citrix ADM サービスを使用すると、さまざまな種類の展開環境にある Citrix ADC を管理および監視できます。オンプレミスとクラウドに展開された Citrix ADC は、ワンストップで管理できます。
- 運用効率 — 運用生産性を向上させる最適化および自動化された方法。Citrix ADM Service を使用すると、従来のハードウェア展開の保守とアップグレードにかかる時間、コスト、およびリソースを節約できるため、運用コストが削減されます。

Kubernetes アプリケーションのサービスグラフ

Citrix ADM のクラウドネイティブアプリケーション機能のサービスグラフを使用すると、次のことができます。

- エンド・ツー・エンドのアプリケーション全体のパフォーマンスを確保

- アプリケーションのさまざまなコンポーネントの相互依存性によって生じるボトルネックを特定
- アプリケーションのさまざまなコンポーネントの依存関係に関する洞察を集める
- Kubernetes クラスター内のサービスを監視する
- 問題のあるサービスを監視する
- パフォーマンスの問題に寄与する要因を確認する
- サービス HTTP トランザクションの詳細な可視性を表示
- HTTP、TCP、SSL メトリックの分析

Citrix ADM でこれらのメトリックを視覚化することで、問題の根本原因を分析し、必要なトラブルシューティングアクションを迅速に行うことができます。サービスグラフには、さまざまなコンポーネントサービス内のアプリケーションが表示されます。Kubernetes クラスター内で実行されるこれらのサービスは、アプリケーション内外のさまざまなコンポーネントと通信できます。

はじめに、「サービスグラフの設定」をご参照ください。

3 層 **Web** アプリケーションのサービスグラフ

アプリケーションダッシュボードのサービスグラフ機能を使用すると、次の項目を表示できます。

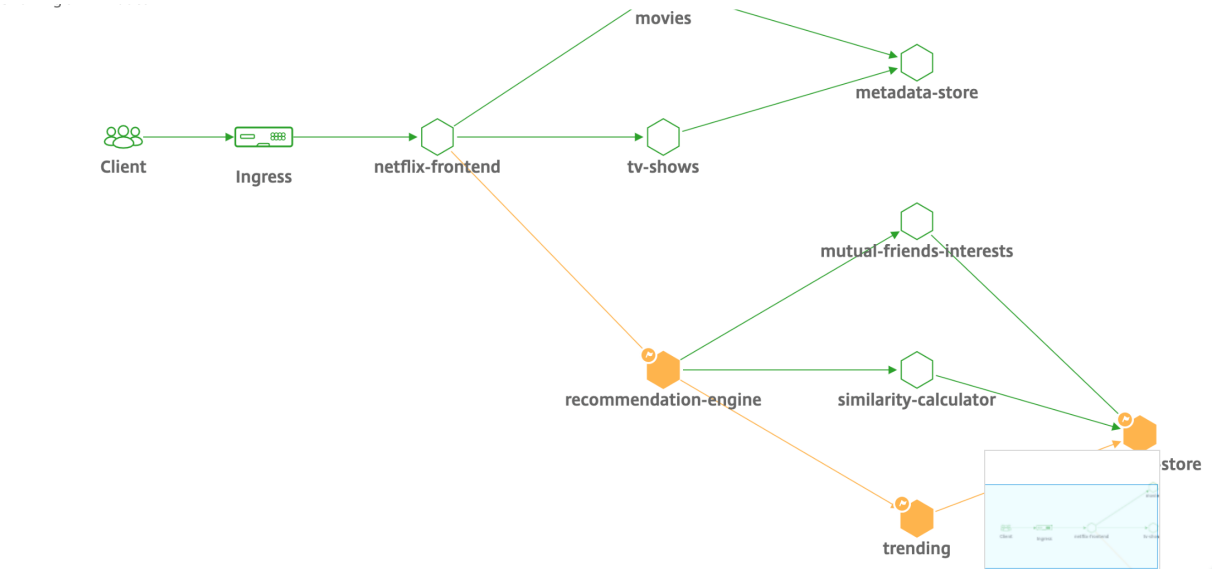
- アプリケーションの構成方法の詳細（コンテンツスイッチング仮想サーバーと負荷分散仮想サーバーを使用）
  - GSLB アプリケーションの場合、データセンター、ADC インスタンス、CS、および LB 仮想サーバーを表示できます。
- クライアントからサービスへのエンド・ツー・エンドのトランザクション
- クライアントがアプリケーションにアクセスしている場所
- クライアント要求が処理されるデータセンターの名前と、関連するデータセンター Citrix ADC メトリック（GSLB アプリケーションのみ）
- クライアント、サービス、仮想サーバーのメトリックの詳細
- エラーがクライアントまたはサービスからのものである場合
- 「緊急」、「レビュー」、「良好」などのサービスステータス。Citrix ADM は、サービスの応答時間とエラー数に基づいてサービスステータスを表示します。
  - 重大 **(**赤**)** -平均サービス応答時間が 200 ミリ秒を超え、エラーカウントが 0 より大きいことを示します。
  - **Review (**オレンジ**)** -平均サービス応答時間が 200 ミリ秒を超えるか、エラーカウントが 0 より大きいことを示します。
  - 良好 **(**緑**)** -エラーがなく、平均サービス応答時間が 200 ミリ秒未満であることを示します
- **Critical**、**Review**、**Good** などのクライアントのステータス。Citrix ADM は、クライアントネットワークの遅延とエラー数に基づいてクライアントのステータスを表示します。
  - **Critical (**赤**)**-平均クライアントネットワーク遅延が 200 ミリ秒を超え、エラーカウントが 0 より大きいことを示します
  - **Review (**オレンジ**)** -平均クライアントネットワーク遅延が 200 ミリ秒を超えるか、エラーカウントが 0 より大きいことを示します。
  - 良好 **(**緑**)** -エラーがなく、平均クライアントネットワーク遅延が 200 ミリ秒未満であることを示します。

- クリティカル、レビュー、良好（**Good**）などの仮想サーバのステータス。Citrix ADM は、アプリのスコアに基づいて仮想サーバーのステータスを表示します。
  - クリティカル **(**赤**)** -アプリのスコアが 40 未満になったことを示します
  - **Review (**オレンジ**)** -アプリのスコアが 40～75 の間であることを示します
  - **Good**（緑） -アプリのスコアが 75 を超えることを示します。

注意事項**:**

- サービスグラフには、負荷分散、コンテンツスイッチング、GSLB 仮想サーバーのみが表示されます。
- 仮想サーバがカスタムアプリケーションにバインドされていない場合、その詳細はそのアプリケーションのサービスグラフに表示されません。
- 仮想サーバと Web アプリケーションの間でアクティブなトランザクションが発生した場合にのみ、クライアントとサービスのメトリックをサービスグラフに表示できます。
- 仮想サーバーと Web アプリケーション間でアクティブなトランザクションを利用できない場合は、負荷分散、コンテンツスイッチング、GSLB 仮想サーバー、サービスなどの構成データに基づくサービスグラフにのみ詳細を表示できます。
- アプリケーション設定の更新がサービスグラフに反映されるまで 10 分かかる場合があります。

詳細については、「アプリケーション用サービスグラフ」を参照してください。



開始するには、Service Graph のドキュメントを参照してください。

Citrix ADC チームのトラブルシューティング

Citrix ADC プラットフォームのトラブルシューティングで最も一般的な属性のいくつかと、これらのトラブルシューティング手法がマイクロサービストポロジの Tier-1 展開にどのように適用されるかについて説明します。

Citrix ADC には、コマンドをリアルタイムで表示するコマンドラインインターフェイス（CLI）があり、ランタイム構成、静的、およびポリシー構成を決定するのに役立ちます。これは 「**SHOW**」コマンドで簡単に行えます。

SHOW-ADC CLI オペレーションを実行します。

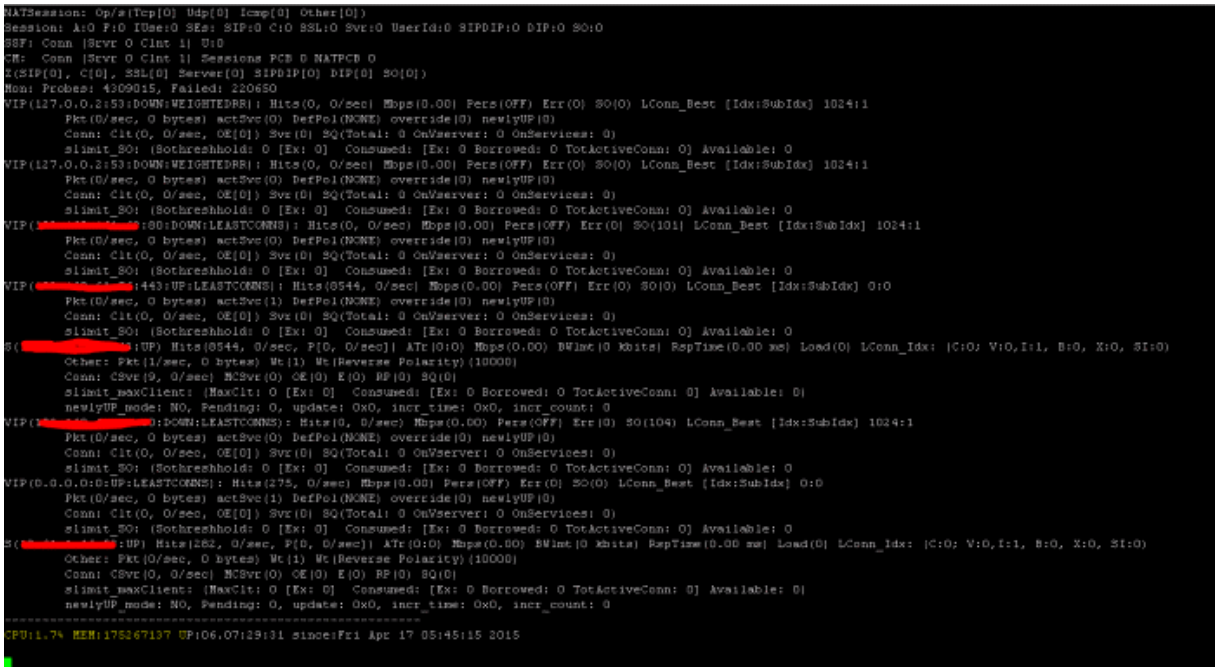
```
>Show running config (-summary -fullValues)

 Ability to search (grep command)
> “sh running config | -i grep vserver”

Check the version.
>Show license
   “sh license”
<!--NeedCopy-->
```

SSL 統計情報の表示

```
>Sh ssl
        System
        Frontend
        Backend
        Encryption
<!--NeedCopy-->
```



Citrix ADC には、7 秒のカウンタ間隔に基づいてすべてのオブジェクトの統計を列挙するコマンドがあります。これは 「**STAT**」コマンドによって容易になります。

Citrix ADC によるきめ細かな **L3-L7** テレメトリ

- システムレベル:ADC の CPU およびメモリ使用率。
- HTTP プロトコル:#Requests /レスポンス、GET/POST スプリット、N-S および E-W の HTTP エラー (サービスメッシュライトのみ、サイドカーはまもなく)。
- SSL: #Sessions および #Handshakes は、サービスメッシュ Lite 専用の N-S および E-W トラフィック用です。
- IP プロトコル:#Packets 受信/送信、#Bytes 受信/送信、#Truncated パケット、#IP アドレスルックアップ
- Citrix ADC AAA: #Active セッション
- インターフェイス:#Total マルチキャストパケット、#Total 転送バイト、および送受信された #Jumbo パケット
- 負荷分散仮想サーバーとコンテンツスイッチング仮想サーバー:#Packets、#Hits、#Bytes が受信/送信されました。

STAT-ADC CLI オペレーションを実行します。

```
1 >Statistics
2  “stat ssl”
3 <!--NeedCopy-->
```



```
> stat ns

System overview

Up since        Thu Apr 16 19:45:15 2015
Packet CPU usage (%)                1.60
Management CPU usage (%)            0.80
Memory usage (MB)                    165
InUse Memory (%)                   17.03
Last Transition time Th...015
System state                         UP
Master state                    Primary
# SSL cards UP                         0
# SSL cards present                    0

System Disks                   Used (%) Available
/flash Used (%)                       17      1168
/var Used (%)                         13     11246

Throughput Statistics                    Rate (/s)               Total
Megabits received                                2                288237
Megabits transmitted                             3                345685

TCP Connections                  Client   Server
All client connections              158      272
Established client connections      158      145

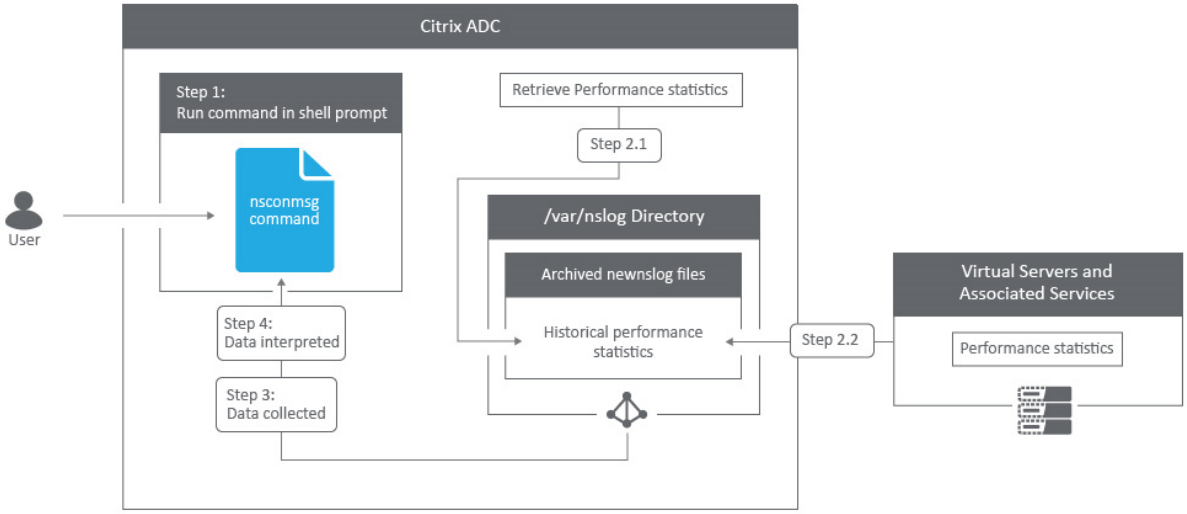
HTTP                                     Rate (/s)               Total
Total requests                                   0                191529
Total responses                                  0                263011
Request bytes received                        7007           1178810535
Response bytes received                     164477          12348432171

SSL                                      Rate (/s)               Total
```

Citrix ADC にはログアーカイブ構造があり、「**NSCONMSG**」コマンドを使用して特定のエラーをトラブルシューティングするときに統計とカウンタを検索できます。

NSCONMSG -メインログファイル **(ns** データ形式**)**

```
1     Cd /var/nslog
2
3      “Mac Moves”
4     nsconmsg -d current -g nic_err
5  <!--NeedCopy-->
```

Nstcpdump

nstcpdumpは、低レベルのトラブルシューティングに使用できます。nstcpdumpは、nstraceより詳細な情報を収集しません。ADC CLI を開き、shellと入力します。フィルタはnstcpdumpと使用できますが、ADC リソース固有のフィルタとともには使用できません。ダンプ出力は CLI 画面内で直接表示できます。

CTRL+C — これらのキーを同時に押すと、nstcpdumpが停止します。

`nstcpdump.sh dst host x.x.x.x` — 宛先ホストに送信されたトラフィックを表示します。

`nstcpdump.sh -n src host x.x.x.x` — 指定されたホストからのトラフィックを表示し、IP アドレスを名前に変換しない (-n)。

`nstcpdump.sh host x.x.x.x` — 指定したホスト IP との間で送受信されるトラフィックを表示します。

![Example nstcpdump](/en-us/citrix-adc/media/nstcpdump.png)

NSTRACE -パケットトレースファイル

NSTRACE は、ネットワークをトラブルシューティングするための低レベルのパケットデバッグツールです。これにより、アナライザツールを使用してさらに分析できるキャプチャファイルを保存できます。一般的なツールは、ネットワークアナライザと Wireshark の 2 つです。

![The nstrace output](/en-us/citrix-adc/media/nstrace.png)

```
> start nstrace -size 0
 Done
> stop nstrace
 Done
```

NSTRACE キャプチャファイルが ADC の /var/nstrace に作成されると、キャプチャファイルを Wireshark にインポートして、パケットキャプチャとネットワーク分析を行うことができます。

SYSCTL-詳細な **ADC** 情報**:** 説明、モデル、プラットフォーム、**CPU** など

```
    sysctl -a grep hw.physmem

    hw.physmem: 862306304
    netscaler.hw_physmem_mb: 822
<!--NeedCopy-->
```

aaad.debug-認証デバッグ情報のためにパイプをオープンする

```
process_radius Got RADIUS event
process_radius Received BAD_ACCESS_REJECT for: <username>
process_radius Sending reject.
send_reject_with_code Rejecting with error code 4001.
```

aaad.debug モジュールを使用した ADC または ADC ゲートウェイ経由での認証問題のトラブルシューティング方法の詳細については、aaad.debug のサポート記事を参照してください。

また、ADC のパフォーマンス統計やイベント・ログを直接取得することもできます。詳細については、ADC サポートドキュメントを参照してください。

SRE チームとプラットフォームチームのトラブルシューティング

Kubernetes トラフィックフロー

North/South:

- North/South トラフィックは、イングレス経由でユーザからクラスタに流れるトラフィックです。

East/West:

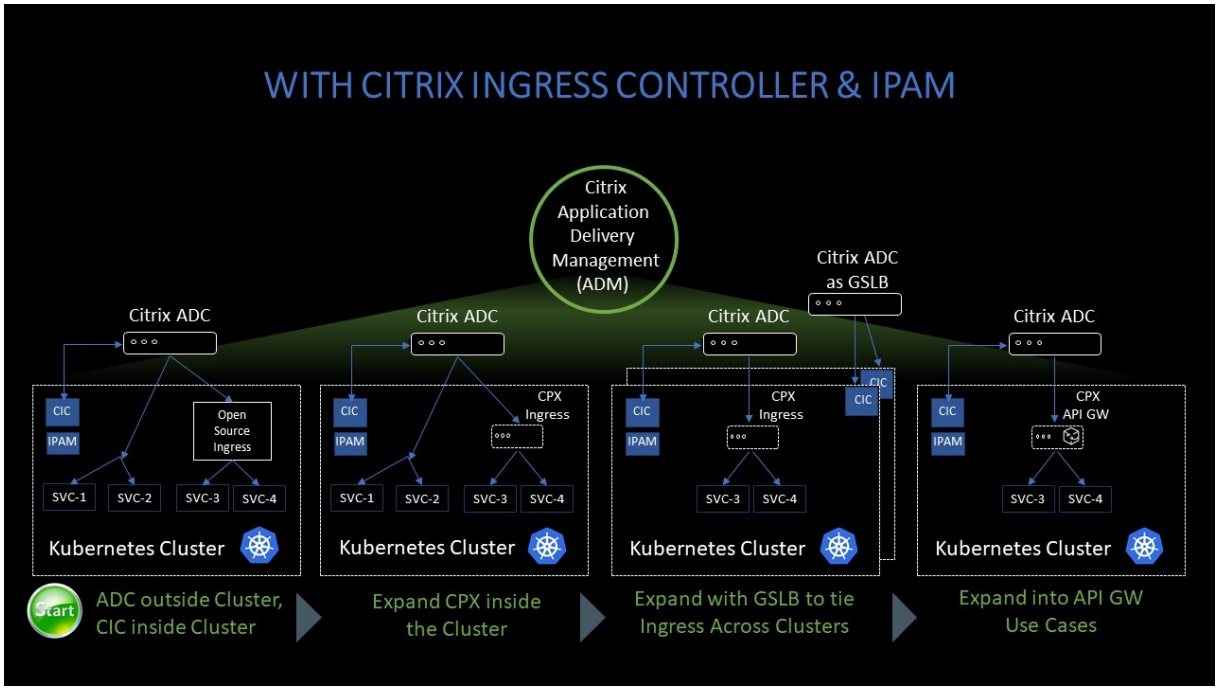
- East/West トラフィックは、Kubernetes クラスタの周りを流れるトラフィック (サービス間またはサービス間データストア) です。

Citrix ADC CPX が **Kubernetes** 環境で **east-west** トラフィックフローを負荷分散する方法

Kubernetes クラスターをデプロイしたら、ADM で Kubernetes 環境の詳細を指定して、クラスターを ADM と統合する必要があります。ADM は、サービス、エンドポイント、Ingress ルールなどの Kubernetes リソースの変更を監視します。

Kubernetes クラスターに ADC CPX インスタンスをデプロイすると、ADM に自動的に登録されます。登録プロセスの一環として、ADM は CPX インスタンスの IP アドレスと、NITRO REST API を使用してインスタンスに到達して構成できるポートについて学習します。

次の図は、ADC CPX が Kubernetes クラスターで East-West トラフィックフローを負荷分散する方法を示しています。



この例の説明を次に示します。

Kubernetes クラスターのノード 1 とノード 2 には、フロントエンドサービスとバックエンドサービスのインスタンスが含まれています。ADC CPX インスタンスがノード 1 とノード 2 にデプロイされると、ADC CPX インスタンスは自動的に ADM に登録されます。ADM で Kubernetes クラスターの詳細を設定して、Kubernetes クラスターを ADM に手動で統合する必要があります。

クライアントがフロントエンドサービスを要求すると、Ingress リソースは、2 つのノード上にあるフロントエンドサービスのインスタンスの間で要求を負荷分散します。フロントエンドサービスのインスタンスがクラスター内のバックエンドサービスからの情報を必要とする場合、要求はノード内の ADC CPX インスタンスに転送されます。この ADC CPX インスタンスは、クラスター内のバックエンドサービス間でリクエストの負荷を分散し、East-West トラフィックフローを提供します。

アプリケーション用の **ADM** サービスグラフ

Citrix ADM サービスグラフ機能を使用すると、すべてのサービスをグラフィカルに監視できます。この機能は、詳細な分析と有用なメトリックも提供します。次のサービスグラフを表示できます。

- すべての Citrix ADC インスタンスで構成されたアプリケーション
- Kubernetes アプリケーション
- 3 層の Web アプリケーション

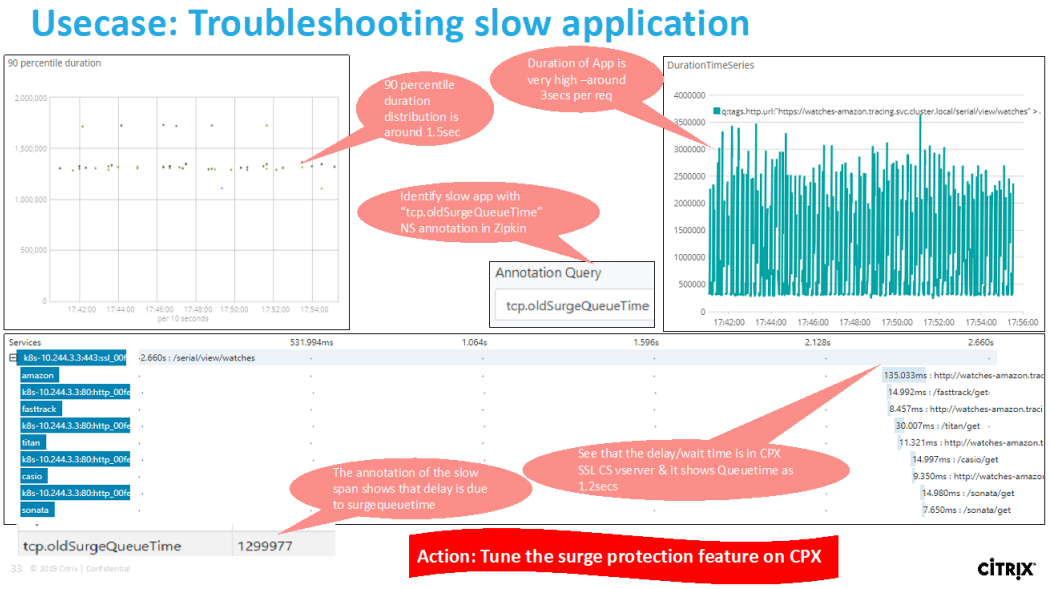
開始するには、サービスグラフの詳細を参照してください。

マイクロサービスアプリケーションカウンターの表示

サービスグラフには、Kubernetes クラスターに属するすべてのマイクロサービスアプリケーションも表示されます。ただし、サービスにマウスポインタを置くと、メトリクスの詳細が表示されます。

以下を表示できます:

- サービス名
- SSL、HTTP、TCP、SSL over HTTP、SSL などのサービスで使用されるプロトコル
- **Hits** — サービスによって受信されたヒットの総数
- サービス応答時間 — サービスから取得した平均応答時間。
  (応答時間 = クライアント RTT + 要求の最後のバイト — 要求の最初のバイト)
- エラー — 4xx、5xx などのエラーの総数
- **Data Volume** — サービスによって処理されるデータの総量
- 名前空間 — サービスの名前空間
- クラスター名 — サービスがホストされているクラスター名
- **SSL** サーバーエラー — サービスからの SSL エラーの合計



これらの特定のカウンターとトランザクションログは、サポートされているさまざまなエンドポイントを使用して、Citrix Observability Exporter（COE）で抽出できます。COE の詳細については、次のセクションを参照してください。

Citrix ADC 統計情報のエクスポーター

これは、Citrix ADC 統計情報をスクレイピングし、HTTP 経由で Prometheus にエクスポートするシンプルなサーバーです。その後、Prometheus をデータソースとして Grafana に追加して、Citrix ADC の統計情報をグラフィカルに表示できます。

Citrix ADC インスタンスの統計情報とカウンターを監視するために、`citrix-adc-metric-exporter` をコンテナまたはスクリプトとして実行できます。エクスポーターは、仮想サーバーへの総ヒット数、HTTP 要求レート、SSL 暗号化/復号化レートなど、Citrix ADC インスタンスから Citrix ADC 統計を収集し、Prometheus サーバーが統計情報を取得してタイムスタンプ付きで保存するまで保持します。その後、Grafana を Prometheus サーバーにポイントして、Citrix ADC 統計の分析に必要な統計情報の取得、プロット、アラームの設定、ヒートマップの作成、テーブルの生成などを行うことができます。

以下のセクションでは、図に示すような環境でエクスポータが動作するように設定する方法について詳しく説明します。エクスポーターがデフォルトでスクレイピングする Citrix ADC エンティティ/メトリックとその変更方法についても説明します。

Citrix ADC 用エクスポーターについて詳しくは、メトリクスエクスポーター GitHubを参照してください。

ADM サービス分散トレーシング

サービスグラフでは、分散トレーシングビューを使用して次の操作を実行できます。

- サービス全体のパフォーマンスを分析します。
- 選択したサービスとその相互依存サービス間の通信フローを視覚化します。
- エラーを示すサービスを特定し、エラーのあるサービスをトラブルシューティングする
- 選択したサービスと相互依存する各サービス間のトランザクション詳細を表示します。

ADM 分散トレーシングの前提条件

サービスのトレース情報を表示するには、次の操作を行う必要があります。

- East-West トラフィックを送信する間、アプリケーションが次のトレースヘッダーを保持していることを確認します。

- `x-request-id`
- `x-b3-traceid`
- `x-b3-spanid`
- `x-b3-parentspanid`
- `x-b3-sampled`
- `x-b3-flags`
- `x-ot-span-context`

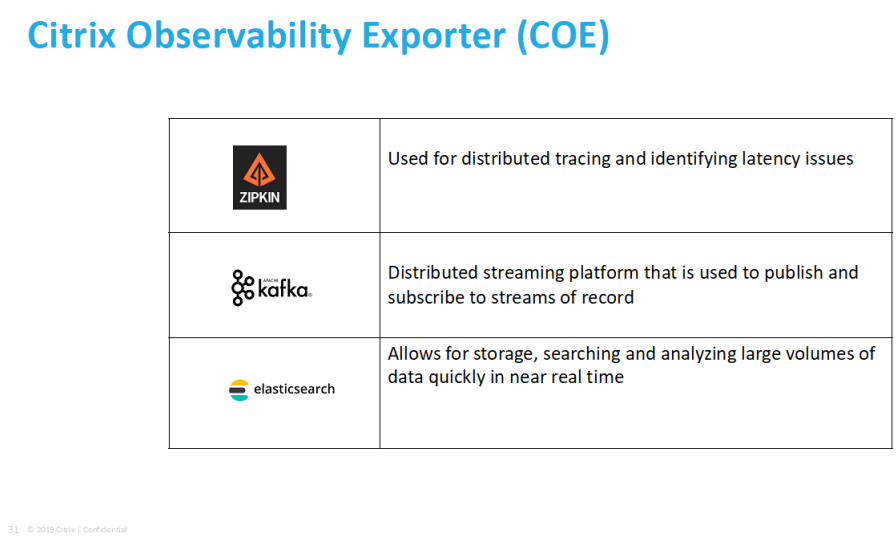
- CPX YAML ファイルを NS_DISTRIBUTED_TRACING で更新し、値を「はい」に設定します。
  はじめに、「分散トレーシング」を参照してください。

Citrix ADC オブザーバビリティエクスポーター（**COE**）の解析

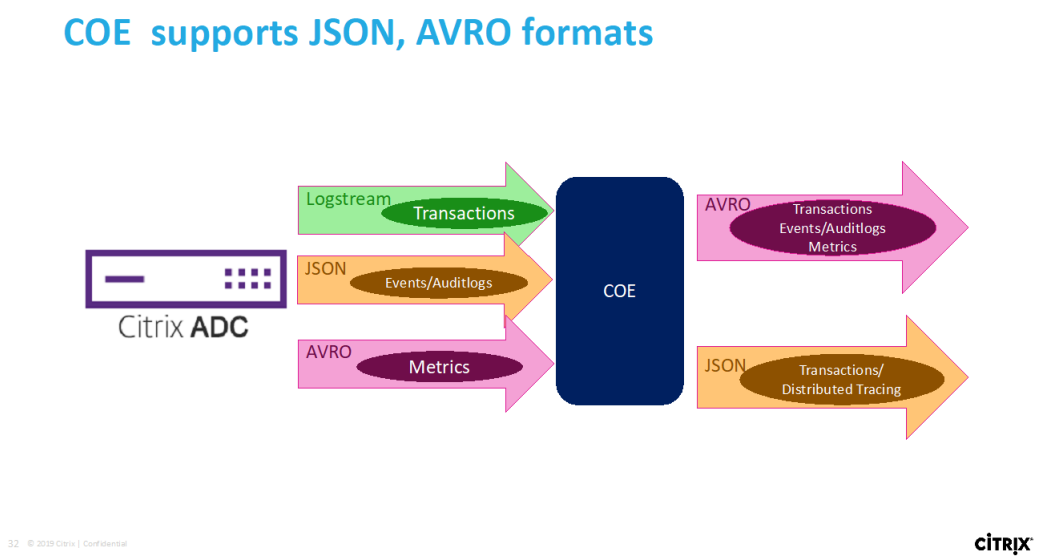
Citrix Observability Exporter は、Citrix ADC からメトリックとトランザクションを収集し、サポートされているエンドポイントに適した形式（JSON、AVRO など）に変換するコンテナです。Citrix Observability Exporter で収集したデータを、目的のエンドポイントにエクスポートできます。エンドポイントにエクスポートされたデータを分析することで、Citrix ADC によってプロキシされるアプリケーションについて、マイクロサービスレベルで貴重なインサイトを得ることができます。

COE の詳細については、COE GitHubを参照してください。

Elasticsearch をトランザクションエンドポイントとする **COE**



Elasticsearch がトランザクションエンドポイントとして指定されている場合、Citrix オブザーバビリティエクスポーターはデータを JSON 形式に変換します。Elasticsearch サーバーでは、Citrix オブザーバビリティエクスポーターが各 ADC の Elasticsearch インデックスを時間単位で作成します。これらのインデックスは、データ、時間、ADC の UUID、および HTTP データのタイプ (http_event または http_error) に基づいています。その後、Citrix オブザーバビリティエクスポーターは、各 ADC の Elastic 検索インデックスの下にデータを JSON 形式でアップロードします。通常のトランザクションはすべて http_event インデックスに配置され、異常は http_error インデックスに配置されます。



Zipkin による分散トレースのサポート

マイクロサービスアーキテクチャでは、1 つのエンドユーザー要求が複数のマイクロサービスにまたがる場合があるため、トランザクションの追跡やエラーの原因の修正が困難になります。このような場合、従来のパフォーマンス監視方法では、障害が発生した場所やパフォーマンスの低下の原因を正確に特定することはできません。リクエストを処理する各マイクロサービスに固有のデータポイントをキャプチャし、分析して有意義なインサイトを得る方法が必要です。

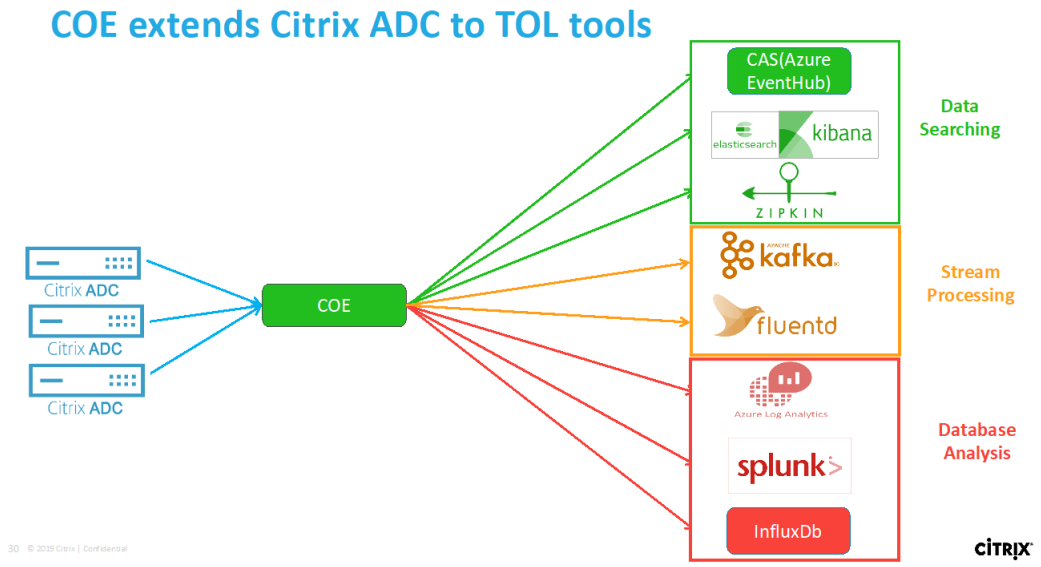
分散トレーシングは、トランザクションをエンドツーエンドで追跡し、複数のマイクロサービスにわたってトランザクションがどのように処理されているかを理解する方法を提供することで、この課題に対処します。

OpenTracing は、分散トレースを設計および実装するための API の仕様および標準セットです。分散トレーサを使用すると、マイクロサービス間のデータフローを視覚化し、マイクロサービスアーキテクチャのボトルネックを特定するのに役立ちます。

Citrix ADC オブザーバビリティエクスポーターは、Citrix ADC の分散トレースを実装しており、現在、分散トレーサーとして Zipkin をサポートしています 。

現在、Citrix ADC を使用してアプリケーションレベルでパフォーマンスを監視できます。Citrix ADC で Citrix Observability Exporter を使用すると、Citrix ADC CPX、MPX、または VPX によってプロキシされた各アプリケーションのマイクロサービスのトレースデータを取得できます。

はじめに、GitHub オブザーバビリティエクスポーターを参照してください。

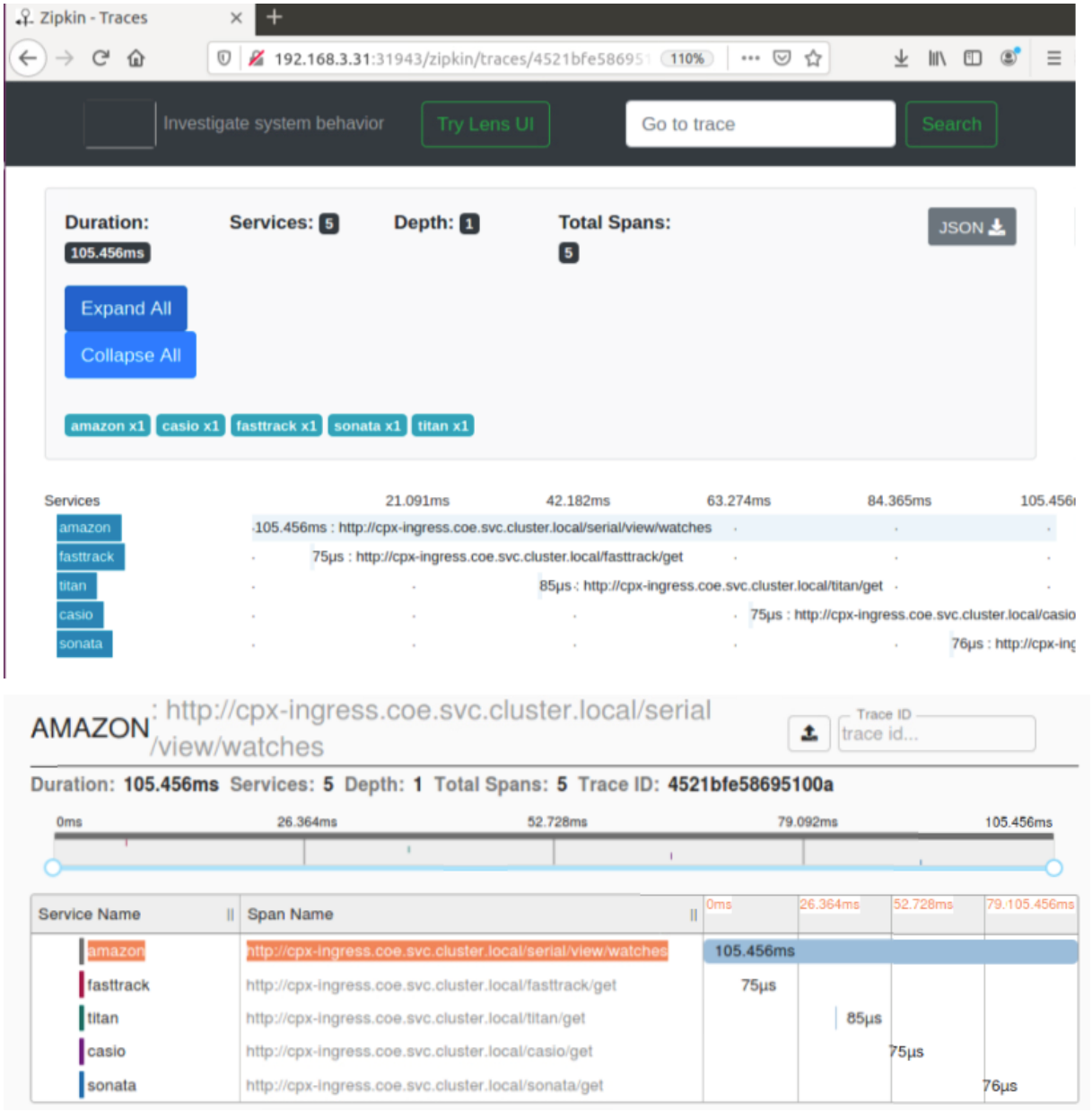


アプリケーションデバッグ用の **Zipkin**

Zipkin は、[Google の Dapper の論文に基づいたオープンソースの分散トレースシステムです](https://storage.googleapis.com/pub-tools-public-publication-data/pdf/36356.pdf)。Dapper は、本番環境での分散トレースのための Google のシステムです。Google はこれを彼らの論文「Google の開発者に複雑な分散システムの振る舞いに関するより多くの情報を提供するために Dapperを構築した」で説明している。トラブルシューティングを行う場合、特にシステムが複雑で分散している場合には、さまざまな角度からシステムを観察することが重要です。

次の Zipkin トレースデータは、Watches サンプルアプリケーションに関連する合計 5 つのスパンと 5 つのサービスを識別します。トレースデータには、5 つのマイクロサービスにまたがる特定のスパンデータが表示されます。

開始するには、Zipkinを参照してください。



最初のページ読み込みリクエストのアプリケーションレイテンシーを示す Zipkin スパンの例:

Services: amazon

Date Time	Relative Time	Annotation	Address
7/15/2020, 2:14:24 PM		Server Start	10.10.235.179:1719 (amazon)
7/15/2020, 2:14:24 PM	105.456ms	Server Finish	10.10.235.179:1719 (amazon)

Key	Value
component	py_zipkin
http.host	amazon:1719
http.method	GET
http.path	/serial/view/watches
http.url	http://cpx-ingress.coe.svc.cluster.local/serial/view/watches
Local Component	amazon
peer.address	10.10.235.190

データを見るための **Kibana**

Kibana は、Elasticsearch データを視覚化して Elastic Stack をナビゲートできるオープンなユーザーインターフェイスです。クエリのロードの追跡から、アプリ内でのリクエストの流れの把握まで、あらゆることを行えます。

アナリストでも管理者でも、Kibana は次の 3 つの重要な機能を提供することで、データをアクション可能にします。

- オープンソースの分析および可視化プラットフォーム。Kibana を使用して Elasticsearch データを探索し、美しいビジュアライゼーションとダッシュボードを構築できます。
- **Elastic** スタックを管理するための **UI**。セキュリティ設定の管理、ユーザーロールの割り当て、スナップショットの作成、データのロールアップなど、すべて Kibana UI から実行できます。
- **Elastic** のソリューションの一元化されたハブ。ログ分析からドキュメント検出、SIEM に至るまで、Kibana はこれらの機能やその他の機能にアクセスするためのポータルです。

Kibana は、Elasticsearch をデータソースとして使用するように設計されています。Elasticsearch は、Kibana

を一番上に置き、データを保存して処理するエンジンだと考えてください。

Kibana はホームページから、データを追加するための次のオプションを提供します。

- ファイルデータビジュアライザーを使用してデータをインポートします。
- 組み込みのチュートリアルを使用して、Elasticsearch へのデータフローをセットアップします。データに関するチュートリアルがない場合は、Beats の概要に移動して、Beats ファミリーの他のデータシッパーについて学習します。
- サンプルデータセットを追加して、自分でデータを読み込まなくても Kibana を試乗できます。
- REST APIまたはクライアントライブラリを使用して、Elasticsearch にデータのインデックスを作成します。

Kibana はインデックスパターンを使用して 、どの Elasticsearch インデックスを調べるかを指示します。ファイルをアップロードしたり、組み込みのチュートリアルを実行したり、サンプルデータを追加したりすると、無料でインデックスパターンが得られ、探索を開始するのに適しています。独自のデータをロードする場合は、Stack Managementでインデックスパターンを作成できます。

ステップ 1: Logstash のインデックスパターンを構成する

ステップ 2: インデックスを選択し、入力するトラフィックを生成します。

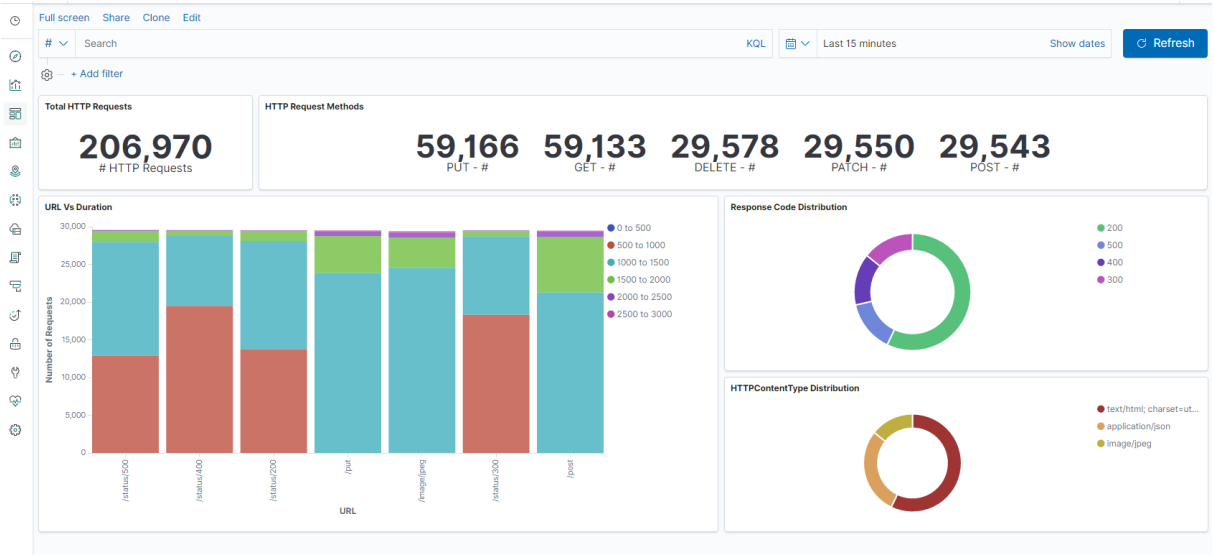
ステップ 3: ログフィードの非構造化データからアプリケーションを生成します。

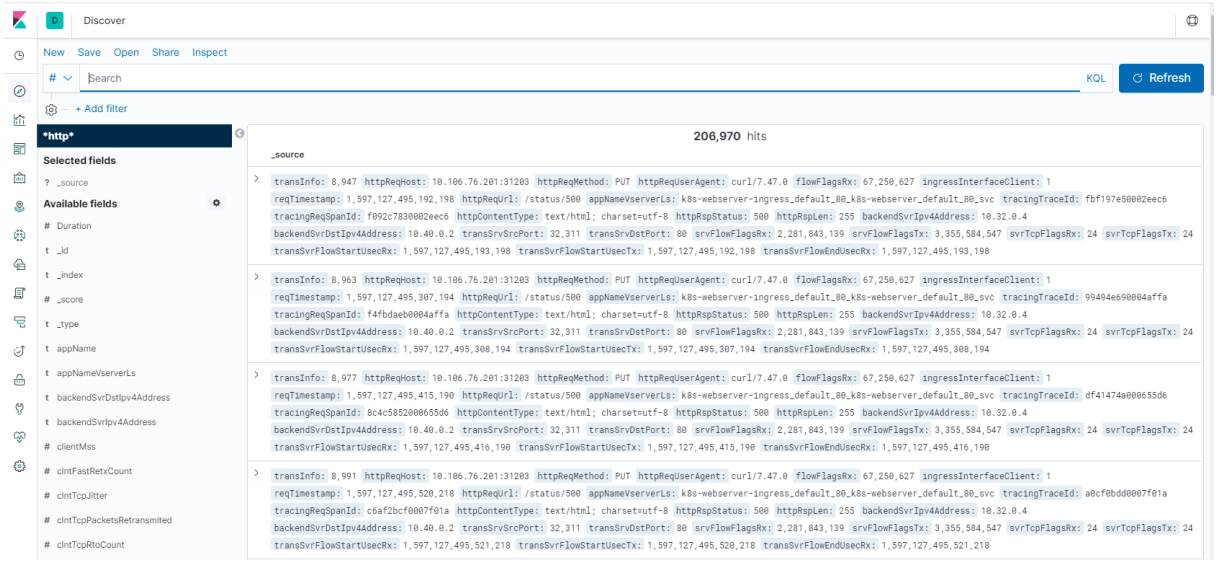
ステップ 4: Kibana は Logstash 入力をフォーマットしてレポートとダッシュボードを作成します。

- 時間範囲
- 表形式表示
- ヒット数はアプリケーションに基づきます。
  - 時刻 IP、エージェント、マシン.OS、レスポンスコード (200)、URL
  - 値によるフィルタリング

ステップ 5: 集計レポートでデータを視覚化します。

- チャート・レポート (円、グラフなど) での結果集計





Citrix ADC VPX インスタンスを展開する

April 7, 2022

> 注
>
> Citrix ADC または Citrix Gateway をインストールまたはアップグレードして 13.0 ビルド 61.xx 以降をリリースすると、Citrix ADM サービス接続がデフォルトで有効になります。詳細については、「データガバナンスと CitrixADM サービス接続」を参照してください。

Citrix ADC VPX 製品は、さまざまな仮想化およびクラウドプラットフォームでホストできる仮想アプライアンスです。

- Citrix Hypervisor
- VMware ESX
- Microsoft Hyper-V
- Linux KVM
- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

詳細については、Citrix ADC VPX のデータシートを参照してください。

SDX アプライアンスでの Citrix ADC VPX インスタンスのプロビジョニングの詳細については、「Citrix ADC インスタンスのプロビジョニング」を参照してください。

Citrix Application Delivery Management for Citrix ADC VPX

Citrix Application Delivery Management ソフトウェアは、管理者が企業全体の可視性を実現し、複数のインスタンスで実行する必要がある管理ジョブを自動化することで、運用を簡素化する一元管理ソリューションです。

Citrix ADC VPX インスタンスは、Citrix Gateway、Citrix ADC SDX、Citrix ADC CPX、Citrix SD-WAN などの他の Citrix アプリケーションネットワーク製品に加えて、Citrix ADC VPX インスタンスを管理および監視できます。アプリケーション配信管理ソフトウェアを使用すると、単一の統合コンソールからグローバルなアプリケーション配信インフラストラクチャ全体を管理、監視、トラブルシューティングできます。

詳細については、Citrix Application Delivery Management ドキュメントを参照してください。

サポート・マトリックスと使用ガイドライン

October 7, 2022

このドキュメントでは、Citrix ADC VPX インスタンスでサポートされているさまざまなハイパーバイザーと機能について説明します。また、使用上のガイドラインと既知の制限事項についても説明します。

表 1. Citrix Hypervisor 上の VPX インスタンス

Citrix Hypervisor のバージョン	システム ID	VPX モデル
8.2 は 13.0 64.x 以降をサポート、8.0、7.6、7.1	450000	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G

表 2. VMware ESXi ハイパーバイザー上の VPX インスタンス

ESXi バージョン	ESXi リリース日のMM/DD/YYYYフォーマット	ESXi ビルド番号	Citrix ADC VPX バージョン	システム ID	VPX モデル
ESXi 7.0 アップデート 3f	07/12/2022	20036589	13.0-86.x 以降	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 アップデート 3D	03/29/2022	19482537	13.0-86.x 以降	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 アップデート 3c	01/27/2022	19193900	13.0-85.x 以降	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi バージョン	ESXi リリース日のMM/DD/YYYYフォーマット	ESXi ビルド番号	Citrix ADC VPX バージョン	システム ID	VPX モデル
ESXi 7.0 アップデート 2d	09/14/2021	18538813	13.0-83.x 以降	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 update 2a	12/17/2020	17867351	13.0-82.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 update 1d	12/17/2020	17551050	13.0-82.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi バージョン	ESXi リリース日のMM/DD/YYYYフォーマット	ESXi ビルド番号	Citrix ADC VPX バージョン	システム ID	VPX モデル
ESXi 7.0 update 1c	12/17/2020	17325551	13.0-82.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 update 1b	10/06/2020	16850804	13.0-76.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0b	06/23/2020	16324942	13.0-71.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi バージョン	ESXi リリース日のMM/DD/YYYYフォーマット	ESXi ビルド番号	Citrix ADC VPX バージョン	システム ID	VPX モデル
ESXi 7.0 GA	04/02/2020	15843807	13.0-71.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 P04	11/19/2020	17167734	13.0-67.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 P03	08/20/2020	16713306	13.0-67.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi バージョン	ESXi リリース日のMM/DD/YYYYフォーマット	ESXi ビルド番号	Citrix ADC VPX バージョン	システム ID	VPX モデル
ESXi 6.7 P02	04/28/2020	16075168	13.0-67.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 P01	12/05/2019	15160138	13.0-67.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 Update 3	08/20/2019	14320388	13.0-58.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi バージョン	ESXi リリース日のMM/DD/YYYYフォーマット	ESXi ビルド番号	Citrix ADC VPX バージョン	システム ID	VPX モデル
ESXi 6.7 U2	04/11/2019	13006603	13.0-47.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.5 GA	11/15/2016	4564106	13.0-47.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.5 U1g	3/20/2018	7967591	13.0 47.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi バージョン	ESXi リリース日のMM/DD/YYYYフォーマット	ESXi ビルド番号	Citrix ADC VPX バージョン	システム ID	VPX モデル
ESXi 6.0 Update 3	2/24/2017	5050593	12.0-51.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.0 Express Patch 11	10/5/2017	6765062	12.0-56.x onwards	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

表 3. Microsoft Hyper-V 上の VPX

Hyper-V 版	システム ID	VPX モデル
2012, 2012 R2, 2016, 2019	450020	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000

表 4. 汎用 KVM 上の VPX インスタンス

汎用 KVM バージョン	システム ID	VPX モデル
RHEL 7.4、RHEL 7.5（Citrix ADC バージョン 12.1 50.x 以降から）、RHEL 7.6、RHEL 8.2、Ubuntu 16.04、Ubuntu 18.04、RHV 4.2	450070	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G。VPX 25G, VPX 40G, VPX 100G

注意事項**:**

KVM ハイパーバイザーを使用するときは、次の点を考慮してください。

- VPX インスタンスは、表 1-4 に記載されている Hypervisor リリースバージョンに対して認定されており、バージョン内のパッチリリースには適していません。ただし、VPX インスタンスは、サポートされているバージョンのパッチリリースとシームレスに動作することが期待されます。そうでない場合は、トラブルシューティングとデバッグのためのサポートケースを記録します。
- `ip link` コマンドを使用して RHEL 8.2 ネットワークブリッジを設定します。
- RHEL 7.6 を使用する前に、KVM ホストで以下のステップを完了します。
  1. /etc/default/grub を編集して”`kvm_intel.preemption_timer=0`”を`GRUB_CMDLINE_LINUX` 変数に追加します。
  2. コマンド”`## grub2-mkconfig -o /boot/grub2/grub.cfg`”で grub.cfg を再生成します。
  3. ホストマシンを再起動します。
- Ubuntu 18.04 を使用する前に、KVM ホストで以下のステップを完了してください。
  1. /etc/default/grub を編集して”`kvm_intel.preemption_timer=0`”を`GRUB_CMDLINE_LINUX` 変数に追加します。
  2. コマンド”`## grub-mkconfig -o /boot/grub/grub.cfg` “で grub.cfg を再生成します。
  3. ホストマシンを再起動します。

表 5. AWS 上の VPX インスタンス

AWS バージョン	システム ID	VPX モデル
-	450040	VPX 10、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX BYOL、VPX 8000、VPX 10G、VPX 15G、VPX 25G は、EC2 インスタンスタイプ（C5、M5、および C5n）での BYOL でのみ使用できます。

> 注:
> VPX 25G オファリングでは、AWS で希望する 25G のスループットは得られませんが、VPX 15G オファリングに比べて SSL トランザクションレートが高くなる可能性があります。

表 6. Azure 上の VPX インスタンス

Azure バージョン	システム ID	VPX モデル
-	450020	VPX 10、VPX 200、VPX 1000、VPX 3000、VPX BYOL

表 7. VPX 機能マトリックス

	VPX on XenServer		VPX on VMware ESX				VPX on Microsoft Hyper-V	VPX on generic KVM			VPX on AWS	VPX on Azure	VPX on GCP
Features	**PV**	**SR-IOV**	**PV**	**SR-IOV**	**Emulated**	**PCI Passthrough**	**PV**	**PV**	**SR-IOV**	**PCI Passthrough**			
Multi-PE Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Clustering Support	Yes	Yes[1]	Yes	Yes[1]	Yes	Yes	Yes	Yes	Yes[1]	Yes	No	No	No
VLAN Tagging	Yes	Yes	Yes	Yes	Yes	Yes	Yes (only on 2012R2)	Yes	Yes	Yes	No	No	No
Detecting Link Events	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	No[2]	Yes[3]	Yes[3]	No[2]	No[2]	No[2]
Interface Parameter Configuration	No	No	No	No	No	Yes	No	No	No	Yes	No	No	No
Static LA	Yes[2]	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	Yes[2]	Yes[2]	Yes[3]	Yes[3]	No	No	No
LACP	No	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]	No	No	No
Static CLAG	No	No	No	No	No	No	No	No	No	No	No	No	No
LACP CLAG	No	No	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]	No	No	No
Hot-plug	No	No	No	No	No	No	No	No	No	No	Yes	No	No

前の表で使用されている上付き番号 (1、2、3) は、それぞれの番号付けで次の点を指します。

1. SRIOV では、バックプレーンではなく、クライアント側およびサーバ側インターフェイス用のクラスタリングサポートを利用できます。

2. インターフェイスダウンイベントは、Citrix ADC VPX インスタンスには記録されません。
3. スタティック LA の場合、物理ステータスが DOWN のインターフェイスでトラフィックが送信される場合もあります。
4. LACP の場合、ピアデバイスは LACP タイムアウトメカニズムに基づいてインターフェイス DOWN イベントを認識します。
   - 短いタイムアウト：3 秒
   - 長いタイムアウト：90 秒
5. LACP では、VM 間でインターフェイスを共有しないでください。
6. ダイナミックルーティングの場合、リンクイベントが検出されないため、コンバージェンス時間はルーティングプロトコルによって異なります。
7. モニタ対象スタティックルート機能は、ルータの状態が VLAN ステータスに依存するため、モニタをスタティックルートにバインドしないと失敗します。VLAN ステータスは、リンクステータスによって異なります。
8. リンク障害がある場合、高可用性では部分的な障害検出は行われません。リンク障害があると、高可用性の分割脳の状態が発生する可能性があります。
   - VPX インスタンスからリンクイベント（無効/有効化、リセット）が生成された場合、リンクの物理ステータスは変わりません。静的 LA の場合、ピアによって開始されたトラフィックはすべてインスタンスでドロップされます。
   - VLAN タギング機能を機能させるには、次の手順を実行します。

   VMware ESX で、VMware ESX サーバの vSwitch でポートグループの VLAN ID を 1～4095 に設定します。VMware ESX サーバの vSwitch での VLAN ID の設定の詳細については、「VMware ESX Server 3 802.1Q VLAN ソリューション」を参照してください。

表 8. サポートされているブラウザー

オペレーティングシステム	ブラウザとバージョン
Windows 7	Internet Explorer-8, 9, 10, 11; Mozilla Firefox 3.6.25 以降; Google Chrome-15 以降
Windows 64 ビット	Internet Explorer-8、9; Google Chrome-15 以降
MAC	Mozilla Firefox-12 以降; Safari-5.1.3; Google Chrome-15 以降

使用ガイドライン

使用上のガイドラインに従ってください。

『**VMware vSphere 6.5** のパフォーマンスのベストプラクティス』の「**VMware ESXi CPU** に関する考慮事項」セ

クションを参照してください。ここに抽出があります:

- CPU/メモリの需要が高い仮想マシンを、オーバーコミットされたホスト/クラスタに置くことは推奨されません。
- ほとんどの環境では、ESXi は、仮想マシンのパフォーマンスに影響を与えることなく、かなりのレベルの CPU オーバーコミットメントを許可します。ホストでは、そのホスト内の物理プロセッサコアの総数よりも多くの vCPU を実行できます。
- ESXi ホストが CPU 飽和状態になった場合、つまり、仮想マシンおよびホスト上のその他の負荷がホストにあるすべての CPU リソースを要求すると、レイテンシの影響を受けやすいワークロードがうまく動作しない可能性があります。この場合は、たとえば、一部の仮想マシンをパワーオフしたり、別のホストに移行したり、DRS による仮想マシンの自動移行を許可したりして、CPU の負荷を軽減できます。
- 仮想マシンで ESXi Hypervisor の最新の機能セットを利用するには、最新のハードウェア互換性バージョンを使用することをお勧めします。ハードウェアと ESXi バージョンの互換性の詳細については、VMware のドキュメントを参照してください。
- Citrix ADC VPX は、レイテンシーに敏感で高性能な仮想アプライアンスです。期待されるパフォーマンスを実現するには、アプライアンスに vCPU 予約、メモリ予約、ホストでの vCPU ピン接続が必要です。また、ホスト上でハイパースレッディングを無効にする必要があります。ホストがこれらの要件を満たさない場合、高可用性フェイルオーバー、VPX インスタンス内の CPU スパイク、VPX CLI へのアクセスにおける低速化、ピットボスデーモンのクラッシュ、パケットドロップ、低スループットなどの問題が発生します。

Hypervisor は、次の 2 つの条件のいずれかが満たされると、過剰プロビジョニングと見なされます。

- ホストにプロビジョニングされた仮想コア (vCPU) の総数が、物理コア (pCPU) の総数を超えています。
- プロビジョニングされた仮想マシンの合計数は、pCPU の合計数よりも多くの vCPU を消費します。

  インスタンスが過剰プロビジョニングされている場合、ハイパーバイザーのスケジューリングオーバーヘッド、バグ、またはハイパーバイザーの制限により、ハイパーバイザーがインスタンスのリザーブドリソース（CPU、メモリなど）を保証しない場合があります。この動作により、Citrix ADC CPU リソースが不足し、使用ガイドラインの最初のポイントで説明されている問題が発生する可能性があります。管理者は、ホスト上でプロビジョニングされる vCPU の総数が pCPU の総数より少なくなるように、ホストのテナント数を減らすことをお勧めします。

  例

  ESX ハイパーバイザーの場合、`esxtop`コマンド出力で VPX vCPU の%RDY%パラメータが 0 より大きい場合、ESX ホストにスケジューリングオーバーヘッドがあると言われ、VPX インスタンスのレイテンシー関連の問題が発生する可能性があります。

  このような状況では、%RDY%が常に 0 に戻るように、ホストのテナンシーを減らします。または、ハイパーバイザーベンダーに連絡して、リソース予約が完了していない理由をトリアージします。

- ホットアドは、AWS 上の Citrix ADC を使用した PV および SRIOV インターフェイスでのみサポートされます。ENA インターフェイスを持つ VPX インスタンスはホットプラグをサポートしていないため、ホットプラ

グを試みるとインスタンスの動作が予測できない場合があります。

- AWS ウェブコンソールまたは AWS CLI インターフェイスを介したホット削除は、Citrix ADC の PV、SRIOV、および ENA インターフェイスではサポートされていません。ホット削除を試みると、インスタンスの動作が予測できなくなる可能性があります。

パケットエンジンの **CPU** 使用率を制御するコマンド

ハイパーバイザーおよびクラウド環境における VPX インスタンスのパケットエンジン（非管理）CPU 使用率の動作を制御するには、2 つのコマンド（`set ns vpxparam`および`show ns vpxparam`）を使用できます。

- `set ns vpxparam [-cpuyield (YES | NO | DEFAULT)] [-masterclockcpu1 (YES | NO)]`

  各 VM が、別の VM に割り当てられているが、使用されていない CPU リソースの使用を許可します。

  `Set ns vpxparam` パラメータ:

  -cpuyield: 割り当てられているが未使用の CPU リソースを解放または解放しません。

  - はい: 割り当てられているが未使用の CPU リソースを別の VM で使用できるようにします。
  - いいえ: 割り当てられている VM のすべての CPU リソースを予約します。このオプションは、ハイパーバイザーおよびクラウド環境で VPX CPU 使用率が高い割合を示します。
  - デフォルト: いいえ。

  > 注
  >
  > すべての Citrix ADC VPX プラットフォームで、ホストシステムの vCPU 使用率は 100% です。`set ns vpxparam -cpuyield YES`コマンドを入力して、この使用方法を上書きします。

  クラスタノードを「yield」に設定する場合は、CCO で次の追加設定を実行する必要があります。

  - クラスタが形成されると、すべてのノードに「yield=Default」が表示されます。
  - すでに「yield=Yes」に設定されたノードを使用してクラスターが形成されている場合、ノードは「DEFAULT」のイールドを使用してクラスターに追加されます。

  > 注:
  >
  > クラスタノードを「yield=YES」に設定する場合は、クラスタの形成後にのみ構成でき、クラスタが形成される前には設定できません。

  -masterclockcpu1: メインクロックソースを CPU0 (管理 CPU) から CPU1 に移動できます。このパラメータには、次のオプションがあります。

  - はい: 仮想マシンがメインクロックソースを CPU0 から CPU1 に移動できるようにします。
  - いいえ:VM はメインクロックソースに CPU0 を使用します。デフォルトでは、CPU0 がメインクロックソースです。

- `show ns vpxparam`

  現在の`vpxparam`設定を表示します。

その他の参考文献

- Citrix Ready 製品については、Citrix Ready Marketplaceにアクセスしてください
- Citrix Ready 製品サポートについては、よくある質問ページを参照してください。
- VMware ESX ハードウェアバージョンについては、VMware Tools のアップグレードを参照してください。

Optimize Citrix ADC VPX performance on VMware ESX, Linux KVM, and Citrix Hypervisors

December 7, 2021

The Citrix ADC VPX performance greatly varies depending on the hypervisor, allocated system resources, and the host configurations. To achieve the desired performance, first follow the recommendations in the VPX data sheet, and then further optimize it using the best practices provided in this document.

Citrix ADC VPX instance on VMware ESX hypervisors

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of Citrix ADC VPX instance on VMware ESX hypervisors.

- Recommended configuration on ESX hosts
- Citrix ADC VPX with E1000 network interfaces
- Citrix ADC VPX with VMXNET3 network interfaces
- Citrix ADC VPX with SR-IOV and PCI passthrough network interfaces

Recommended configuration on ESX hosts

To achieve high performance for VPX with E1000, VMXNET3, SR-IOV, and PCI passthrough network interfaces, follow these recommendations:

- The total number of virtual CPUs (vCPUs) provisioned on the ESX host must be less than or equal to the total number of physical CPUs (pCPUs) on the ESX host.
- Non-uniform Memory Access (NUMA) affinity and CPU affinity must be set for the ESX host to achieve good results.

– To find the NUMA affinity of a Vmnic, log in to the host locally or remotely, and type:

```
#vsish -e get /net/pNics/vmnic7/properties | grep NUMA
Device NUMA Node: 0
<!--NeedCopy-->
```

– To set NUMA and vCPU affinity for a VM, see VMware documentation.

Citrix ADC VPX with E1000 network interfaces

Perform the following settings on the VMware ESX host:

- On the VMware ESX host, create two vNICs from one pNIC vSwitch. Multiple vNICs create multiple Rx threads in the ESX host. This increases the Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase vNIC transmit (Tx) throughput, use a separate Tx thread in the ESX host per vNIC. Use the following ESX command:
  - For ESX version 5.5:

```
esxcli system settings advanced set -o /Net/NetTxWorldlet -i
<!--NeedCopy-->
```

  - For ESX version 6.0 onwards:

```
esxcli system settings advanced set -o /Net/NetVMTxType -i 1
<!--NeedCopy-->
```

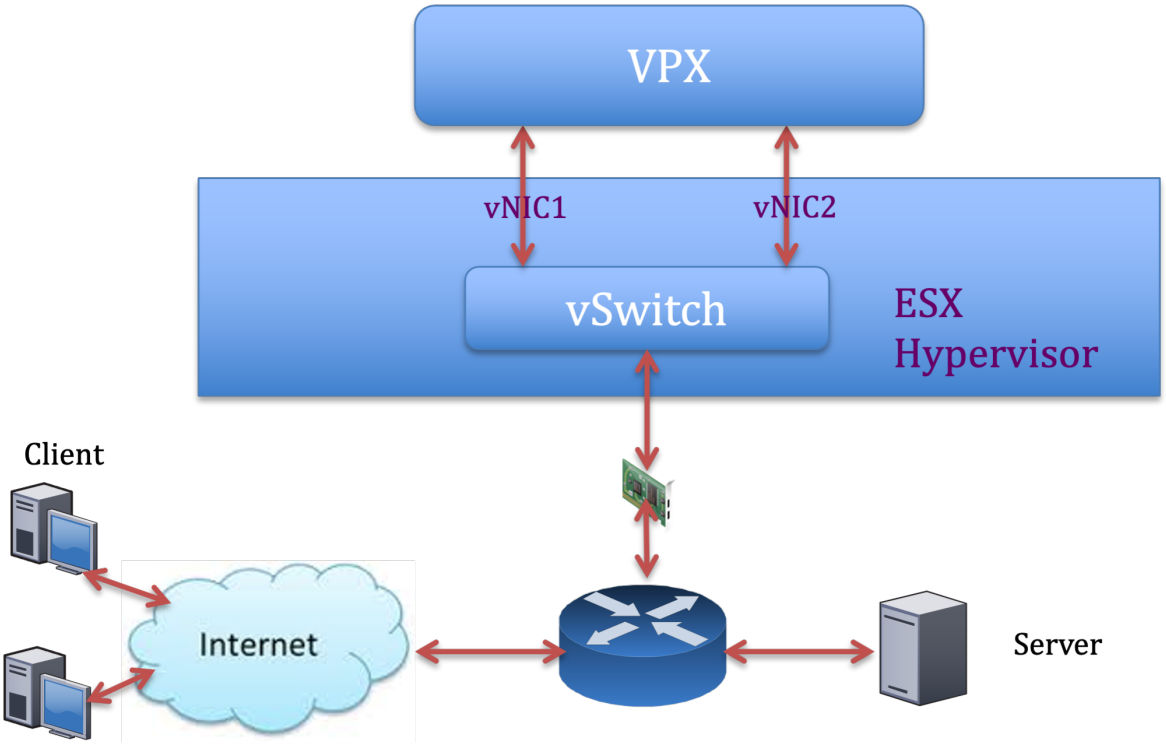
- To further increase the vNIC Tx throughput, use a separate Tx completion thread and Rx threads per device (NIC) queue. Use the following ESX command:

```
esxcli system settings advanced set -o /Net/
   NetNetqRxQueueFeatPairEnable -i 0
<!--NeedCopy-->
```


> Note:
>
> Make sure that you reboot the VMware ESX host to apply the updated settings.

Two vNICs per pNIC deployment

The following is a sample topology and configuration commands for the **Two vNICs per pNIC** model of deployment that delivers better network performance.



Citrix ADC VPX sample configuration:

To achieve the deployment shown in the preceding sample topology, perform the following configuration on the Citrix ADC VPX instance:

- On the client side, bind the SNIP (1.1.1.2) to network interface 1/1 and enable the VLAN tag mode.

```
1  bind vlan 2 -ifnum 1/1  - tagged
2  bind vlan 2 -IPAddress 1.1.1.2 255.255.255.0
3  <!--NeedCopy-->
```

- On the server side, bind the SNIP (2.2.2.2) to network interface 1/1 and enable the VLAN tag mode.


```
bind vlan 3 -ifnum 1/2  - tagged
bind vlan 3 -IPAddress 2.2.2.2 255.255.255.0
<!--NeedCopy-->
```

- Add an HTTP virtual server (1.1.1.100) and bind it to a service (2.2.2.100).

```
add lb vserver v1 HTTP 1.1.1.100 80 -persistenceType NONE -
   Listenpolicy None -cltTimeout 180
add service s1 2.2.2.100 HTTP 80 -gslb NONE -maxClient 0 -maxReq
   0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
   180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
bind lb vserver v1 s1
<!--NeedCopy-->
```

> Note:
>
> Make sure that you include the following two entries in the route table:
>
> - 1.1.1.0/24 subnet with gateway pointing to SNIP 1.1.1.2
> - 2.2.2.0/24 subnet with gateway pointing to SNIP 2.2.2.2

Citrix ADC VPX with VMXNET3 network interfaces

To achieve high performance for VPX with VMXNET3 network interfaces, do the following settings on the VMware ESX host:

- Create two vNICs from one pNIC vSwitch. Multiple vNICs create multiple Rx threads in the ESX host. This increases the Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase vNIC transmit (Tx) throughput, use a separate Tx thread in the ESX host per vNIC. Use the following ESX commands:
  - For ESX version 5.5:

```
esxcli system settings advanced set  - o /Net/NetTxWorldlet  - i
<!--NeedCopy-->
```

  - For ESX version 6.0 onwards:

```
 esxcli system settings advanced set -o /Net/NetVMTxType  - i 1
 <!--NeedCopy-->
```


On the VMware ESX host, perform the following configuration:

- On the VMware ESX host, create two vNICs from 1 pNIC vSwitch. Multiple vNICs create multiple Tx and Rx threads in the ESX host. This increases the Tx and Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase Tx throughput of a vNIC, use a separate Tx completion thread and Rx threads per device (NIC) queue. Use the following command:

```
esxcli system settings advanced set -o /Net/
   NetNetqRxQueueFeatPairEnable -i 0
<!--NeedCopy-->
```

- Configure a VM to use one transmit thread per vNIC, by adding the following setting to the VM's configuration:

```
ethernetX.ctxPerDev = "1"
<!--NeedCopy-->
```

For more information, see Best Practices for Performance Tuning of Telco and NFV Workloads in vSphere

> Note:
>
> Make sure that you reboot the VMware ESX host to apply the updated settings.

You can configure VMXNET3 as a **Two vNICs per pNIC** deployment. For more information, see Two vNICs per pNIC deployment.

Citrix ADC VPX with SR-IOV and PCI passthrough network interfaces

To achieve high performance for VPX with SR-IOV and PCI passthrough network interfaces, see Recommended configuration on ESX hosts.

Citrix ADC VPX instance on Linux-KVM platform

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of Citrix ADC VPX instance on Linux-KVM platform.

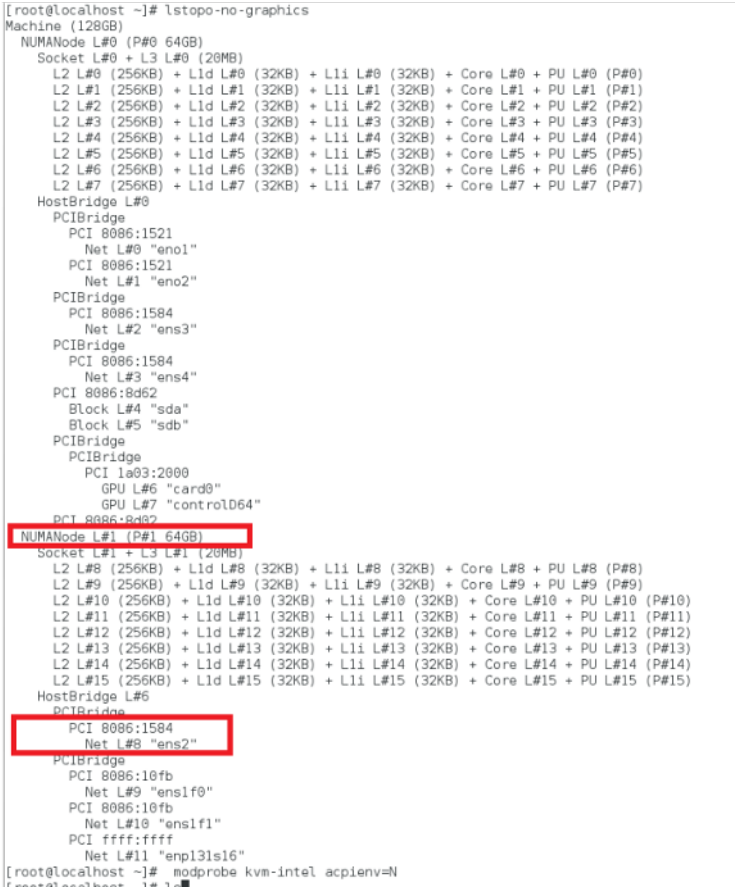
- Performance settings for KVM
- Citrix ADC VPX with PV network interfaces
- Citrix ADC VPX with SR-IOV and Fortville PCIe passthrough network interfaces

Performance settings for KVM

Perform the following settings on the KVM host:

Find the NUMA domain of the NIC using the `lstopo` command:

Make sure that memory for the VPX and the CPU is pinned to the same location.
In the following output, the 10G NIC “ens2” is tied to NUMA domain #1.



Allocate the VPX memory from the NUMA domain.

The `numactl` command indicates the NUMA domain from which the memory is allocated. In the following output, around 10 GB RAM is allocated from NUMA node #0.


```
[root@localhost ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 65429 MB
node 0 free: 55854 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 65536 MB
node 1 free: 52388 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
[root@localhost ~]#
```

To change the NUMA node mapping, follow these steps.

1. Edit the .xml of the VPX on the host.

```
/etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

2. Add the following tag:

```
<numatune>
<memory mode="strict" nodeset="1"/>    This is the NUMA domain
    name
</numatune>
<!--NeedCopy-->
```

3. Shut down the VPX.
4. Run the following command:

```
virsh define /etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

   This command updates the configuration information for the VM with the NUMA node mappings.

5. Power on the VPX. Then check the `numactl -hardware` command output on the host to see the updated memory allocations for the VPX.


```
[root@localhost ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 65429 MB
node 0 free: 65429 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 65536 MB
node 1 free: 55854 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
[root@localhost ~]#
```

Pin vCPUs of VPX to physical cores.

- To view the vCPU to pCPU mappings of a VPX, type the following command

```
virsh vcpupin <VPX name>
<!--NeedCopy-->
```

```
root@localhost qemu]#  virsh vcpupin NS-VPX-DVR
CPU: CPU Affinity
----------------------------------
  0: 8
  1: 9
  2: 10
  3: 11
```

The vCPUs 0–4 are mapped to physical cores 8–11.

- To view the current pCPU usage, type the following command:

```
mpstat -P ALL 5
<!--NeedCopy-->
```



```
[root@localhost qemu]# mpstat -P ALL 5
Linux 3.10.0-123.el7.x86_64 (localhost.localdomain)     05/17/2016      _x86_64_        (16 CPU)

02:26:20 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
02:26:25 PM  all    0.24    0.00    1.67    0.00    0.00    0.00    0.00   17.32    0.00   80.78
02:26:25 PM    0    0.20    0.00    1.00    0.00    0.00    0.00    0.00    0.00    0.00   98.80
02:26:25 PM    1    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    2    0.20    0.00    0.40    0.00    0.00    0.00    0.00    0.00    0.00   99.40
02:26:25 PM    3    0.00    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.80
02:26:25 PM    4    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    5    0.60    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.20
02:26:25 PM    6    0.40    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    7    1.62    0.00    1.42    0.00    0.00    0.00    0.00    0.00    0.00   96.96
02:26:25 PM    8    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM    9    0.00    0.00    7.60    0.00    0.00    0.00    0.00   92.40    0.00    0.00
02:26:25 PM   10    0.20    0.00    7.00    0.00    0.00    0.00    0.00   92.80    0.00    0.00
02:26:25 PM   11    0.00    0.00    8.60    0.00    0.00    0.00    0.00   91.40    0.00    0.00
02:26:25 PM   12    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   13    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   14    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   15    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
```

In this output, 8 is management CPU, and 9–11 are packet engines.

- To change the vCPU to pCPU pinning, there are two options.
  - Change it at runtime after the VPX boots up using the following command:

```
virsh vcpupin <VPX name> <vCPU id>  <pCPU number>
virsh vcpupin NetScaler-VPX-XML 0 8
virsh vcpupin NetScaler-VPX-XML 1 9
virsh vcpupin NetScaler-VPX-XML 2 10
virsh vcpupin NetScaler-VPX-XML 3 11
<!--NeedCopy-->
```

  - To make static changes to the VPX, edit the `.xml` file as before with the following tags:
    1. Edit the .xml file of the VPX on the host

```
/etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

    2. Add the following tag:

```
<vcpu placement='static' cpuset='8-11'>4</vcpu>
    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
```



```
7       </cputune>
8  <!--NeedCopy-->
```

3. Shut down the VPX.
4. Update the configuration information for the VM with the NUMA node mappings using the following command:

```
1  virsh define /etc/libvirt/qemu/ <VPX_name>.xml
2  <!--NeedCopy-->
```

5. Power on the VPX. Then check the `virsh vcpupin <VPX name>` command output on the host to see the updated CPU pinning.

Eliminate host interrupt overhead.

- Detect VM_EXITS using the `kvm_stat` command.

  At the hypervisor level, host interrupts are mapped to the same pCPUs on which the vCPUs of the VPX are pinned. This might cause vCPUs on the VPX to get kicked out periodically.

  To find the VM exits done by VMs running the host, use the `kvm_stat` command.

```
1  [root@localhost ~]# kvm_stat -1  | grep EXTERNAL
2  kvm_exit(EXTERNAL_INTERRUPT)  1728349 27738
3  [root@localhost ~]#
4  <!--NeedCopy-->
```

  A higher value in the order of 1+M indicates an issue.

  If a single VM is present, the expected value is 30–100 K. Anything more than that can indicate that there are one or more host interrupt vectors mapped to the same pCPU.

- Detect host interrupts and migrate host interrupts.

  When you run the `concatenate` command for the “/proc/interrupts” file, it displays all the host interrupt mappings. If one or more active IRQs map to the same pCPU, its corresponding counter increments.

  Move any interrupts that overlap with your Citrix ADC VPX’s pCPUs to unused pCPUs:

```
1  echo 0000000f > /proc/irq/55/smp_affinity
2  0000000f - - > it is a bitmap, LSBs indicates that IRQ 55 can
      only be scheduled on pCPUs 0 - 3
```



```
3   <!--NeedCopy-->
```

- Disable IRQ balance.

  Disable IRQ balance daemon, so that no rescheduling happens on the fly.

```
1   service irqbalance stop
2   service irqbalance show - To check the status
3   service irqbalance start - Enable if needed
4   <!--NeedCopy-->
```

  Make sure you run the `kvm_stat` command to ensure that there are not many counters.

Citrix ADC VPX with PV network interfaces

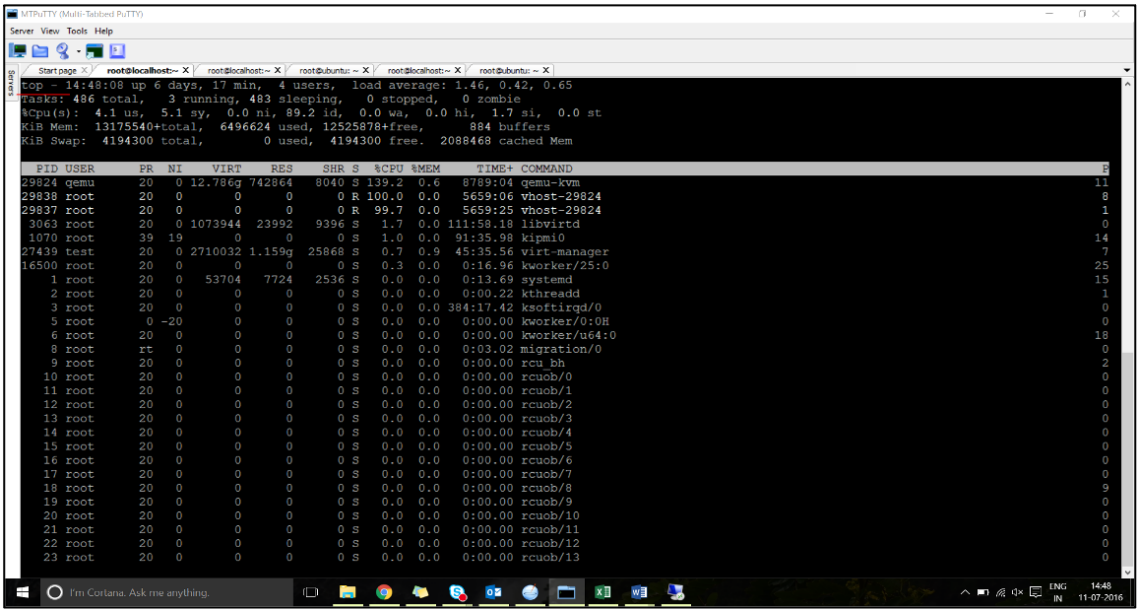
You can configure para-virtualization (PV), SR-IOV, and PCIe passthrough network interfaces as a **Two vNICs per pNIC** deployment. For more information, see Two vNICs per pNIC deployment.

For optimal performance of PV (virtio) interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot/NIC is tied to.
- The Memory and vCPU for the VPX must be pinned to the same NUMA domain.
- Vhost thread must be bound to the CPUs in the same NUMA domain.

Bind the virtual host threads to the corresponding CPUs:

1. Once the traffic is started, run the `top` command on the host.



2. Identify the virtual host process (named as `vhost-<pid-of-qemu>`) affinity.

3. Bind the vHost processes to the physical cores in the NUMA domain identified earlier using the following command:

```
taskset  - pc <core-id> <process-id>
<!--NeedCopy-->
```

Example:

```
taskset  - pc 12 29838
<!--NeedCopy-->
```

4. The processor cores corresponding to the NUMA domain can be identified with the following command:

```
[root@localhost ~]# virsh capabilities | grep cpu
<cpu>
    </cpu>
        <cpus num='8'>
            <cpu id='0' socket_id='0' core_id='0' siblings='0'/>
            <cpu id='1' socket_id='0' core_id='1' siblings='1'/>
            <cpu id='2' socket_id='0' core_id='2' siblings='2'/>
            <cpu id='3' socket_id='0' core_id='3' siblings='3'/>
            <cpu id='4' socket_id='0' core_id='4' siblings='4'/>
            <cpu id='5' socket_id='0' core_id='5' siblings='5'/>
            <cpu id='6' socket_id='0' core_id='6' siblings='6'/>
            <cpu id='7' socket_id='0' core_id='7' siblings='7'/>
        </cpus>

        <cpus num='8'>
        <cpu id='8' socket_id='1' core_id='0' siblings='8'/>
        <cpu id='9' socket_id='1' core_id='1' siblings='9'/>
        <cpu id='10' socket_id='1' core_id='2' siblings='10'/>
        <cpu id='11' socket_id='1' core_id='3' siblings='11'/>
        <cpu id='12' socket_id='1' core_id='4' siblings='12'/>
        <cpu id='13' socket_id='1' core_id='5' siblings='13'/>
        <cpu id='14' socket_id='1' core_id='6' siblings='14'/>
        <cpu id='15' socket_id='1' core_id='7' siblings='15'/>
        </cpus>

    <cpuselection/>
    <cpuselection/>
```



```
28
29 <!--NeedCopy-->
```

Bind the QEMU process to the corresponding physical core:

1. Identify the physical cores on which the QEMU process is running. For more information, see the preceding output.
2. Bind the QEMU process to the same physical cores to which you bind the vCPUs, using the following command:

```
1 taskset -pc 8-11 29824
2 <!--NeedCopy-->
```

Citrix ADC VPX with SR-IOV and Fortville PCIe passthrough network interfaces

For optimal performance of the SR-IOV and Fortville PCIe passthrough network interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot/NIC is tied to.
- The Memory and vCPU for the VPX must be pinned to the same NUMA domain.

Sample VPX XML file for vCPU and memory pinning for Linux KVM:

```
1     <domain type='kvm'>
2         <name>NetScaler-VPX</name>
3         <uuid>138f7782-1cd3-484b-8b6d-7604f35b14f4</uuid>
4         <memory unit='KiB'>8097152</memory>
5         <currentMemory unit='KiB'>8097152</currentMemory>
6         <vcpu placement='static'>4</vcpu>
7
8     <cputune>
9         <vcpupin vcpu='0' cpuset='8'/>
10        <vcpupin vcpu='1' cpuset='9'/>
11        <vcpupin vcpu='2' cpuset='10'/>
12        <vcpupin vcpu='3' cpuset='11'/>
13    </cputune>
14
15    <numatune>
16    <memory mode='strict' nodeset='1'/>
17    </numatune>
18
```



```
19      </domain>
20  <!--NeedCopy-->
```

Citrix ADC VPX instance on Citrix Hypervisors

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of Citrix ADC VPX instance on Citrix Hypervisors.

- Performance settings for Citrix Hypervisors
- Citrix ADC VPX with SR-IOV network interfaces
- Citrix ADC VPX with para-virtualized interfaces

Performance settings for Citrix Hypervisors

Find the NUMA domain of the NIC using the “xl” command:

```
1  xl info -n
2  <!--NeedCopy-->
```

Pin vCPUs of VPX to physical cores.

```
1  xl vcpu-pin <Netsclaer VM Name>  <vCPU id>  <physical CPU id>
2  <!--NeedCopy-->
```

Check binding of vCPUs.

```
1  xl vcpu-list
2  <!--NeedCopy-->
```

Allocate more than 8 vCPUs to Citrix ADC VMs.

For configuring more than 8 vCPUs, run the following commands from the Citrix Hypervisor console:

```
1  xe vm-param-set uuid=your_vms_uuid VCPUs-max=16
2  xe vm-param-set uuid=your_vms_uuid VCPUs-at-startup=16
3  <!--NeedCopy-->
```


Citrix ADC VPX with SR-IOV network interfaces

For optimal performance of the SR-IOV network interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot or NIC is tied to.
- Pin the Memory and vCPU for the VPX to the same NUMA domain.
- Bind the Domain-0 vCPU to the remaining CPU.

Citrix ADC VPX with para-virtualized interfaces

For optimal performance, two vNICs per pNIC and one vNIC per pNIC configurations are advised, as in other PV environments.

To achieve optimal performance of para-virtualized (netfront) interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot or NIC is tied to.
- Pin the memory and vCPU for the VPX to the same NUMA domain.
- Bind the Domain-0 vCPU to the remaining CPU of the same NUMA domain.
- Pin host Rx/Tx threads of vNIC to Domain-0 vCPUs.

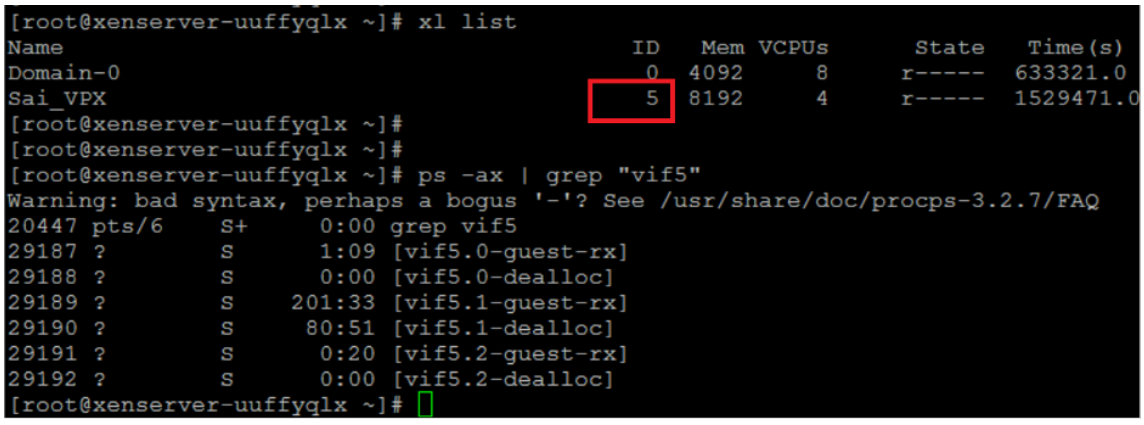
Pin host threads to Domain-0 vCPUs:

1. Find Xen-ID of the VPX by using the `xl list` command on the Citrix Hypervisor host shell.
2. Identify host threads by using the following command:

```
ps -ax | grep vif <Xen-ID>
<!--NeedCopy-->
```

In the following example, these values indicate:

- **vif5.0** - The threads for first interface allocated to VPX in XenCenter (management interface).
- **vif5.1** - The threads for second interface assigned to VPX and so on.



3. Pin the threads to Domain-0 vCPUs using the following command:

```
taskset  - pc <core-id> <process-id>
<!--NeedCopy-->
```

Example:

```
taskset -pc 1 29189
<!--NeedCopy-->
```

Citrix ADC VPX 構成をクラウドで Citrix ADC アプライアンスの最初の起動時に適用する

October 7, 2021

Citrix ADC VPX 構成は、クラウド環境での Citrix ADC アプライアンスの最初の起動時に適用できます。このステージは、このドキュメントでプレブートステージとして取り上げられています。したがって、ADC プールライセンスなどの特定のケースでは、特定の VPX インスタンスがはるかに短時間で起動されます。この機能は、Microsoft Azure、Google クラウドプラットフォーム、および AWS クラウドで使用できます。

ユーザーデータとは何ですか

クラウド環境で VPX インスタンスをプロビジョニングする場合、ユーザーデータをインスタンスに渡すオプションがあります。ユーザーデータを使用すると、一般的な自動設定タスクの実行、インスタンスの起動動作のカスタマイズ、インスタンスの起動後にスクリプトを実行できます。最初の起動時に、Citrix ADC VPX インスタンスは次のタスクを実行します。

- ユーザーデータを読み取ります。
- ユーザーデータで提供される構成を解釈します。
- 新しく追加された構成をブート時に適用します。

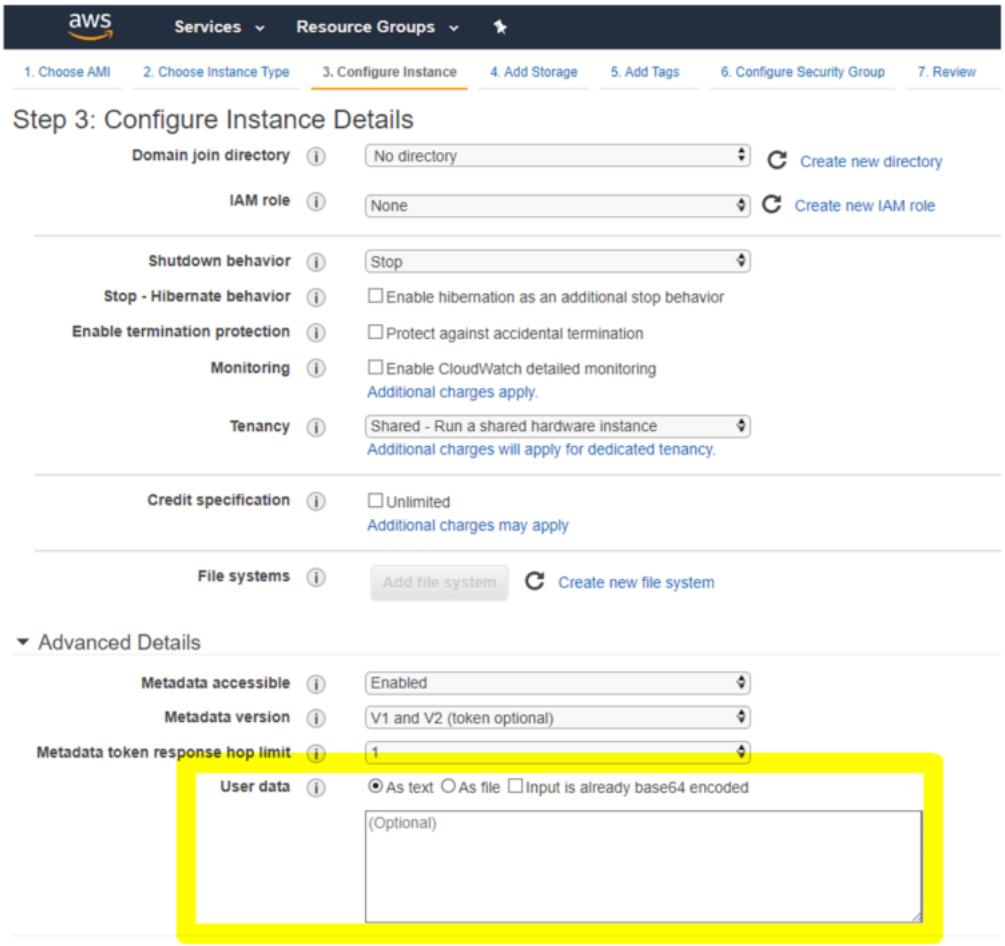
クラウドインスタンスでプレブートユーザーデータを提供する方法

プレブートユーザーデータを XML 形式でクラウドインスタンスに提供できます。クラウドによって、ユーザーデータを提供するためのインターフェースが異なります。

AWS コンソールを使用してプレブートユーザーデータを提供する

AWS コンソールを使用して Citrix ADC VPX インスタンスをプロビジョニングする場合は、[インスタンスの詳細の構成] **>** [詳細の詳細] に移動し、[ユーザーデータ] フィールドにプレブートユーザーデータ構成を指定します。

各手順の詳細については、「AWS ウェブコンソールを使用して AWS に Citrix ADC VPX インスタンスをデプロイする」を参照してください。
詳細については、インスタンスの起動に関するAWS ドキュメントを参照してください。



AWS CLI を使用してプレブートユーザーデータを提供する

AWS CLI で次のコマンドを入力します。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
```



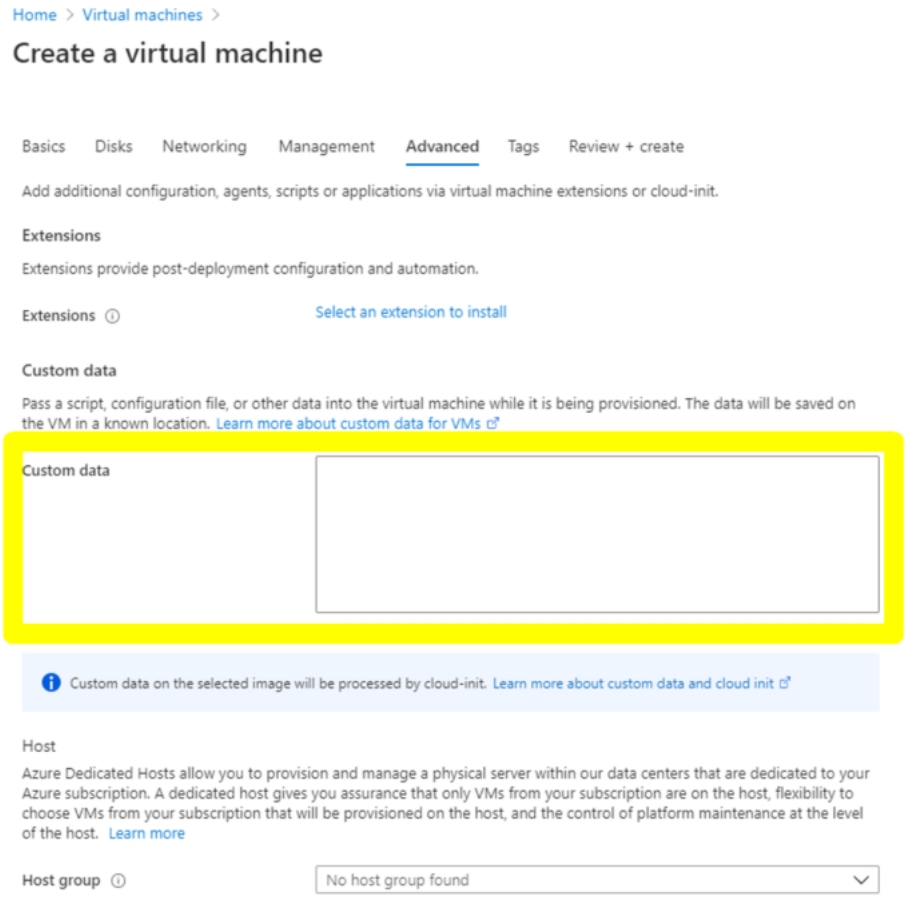
```
6       --key-name MyKeyPair \
7       --security-group-ids sg-0b0384b66d7d692f9 \
8       --user-data file://my_script.txt
9   <!--NeedCopy-->
```

詳細については、インスタンスの実行に関するAWS ドキュメントを参照してください。

詳細については、インスタンスユーザーデータの使用に関するAWS ドキュメントを参照してください。

Azure コンソールを使用してプリブートユーザーデータを提供する

Azure コンソールを使用して Citrix ADC VPX インスタンスをプロビジョニングする場合は、[仮想マシンの作成] > [詳細設定] タブに移動します。[カスタムデータ] フィールドに、プリブートユーザーデータの構成を指定します。



Azure CLI を使用してプリブートユーザーデータを提供する

Azure CLI で次のコマンドを入力します。


```
az vm create \
  --resource-group myResourceGroup \
  --name MyVm \
  --image debian \
  --custom-data MyCloudInitScript.txt \
<!--NeedCopy-->
```

例:

```
az vm create --resource-group MyResourceGroup -name MyVm --image debian
     --custom-data MyCloudInitScript.txt
<!--NeedCopy-->
```

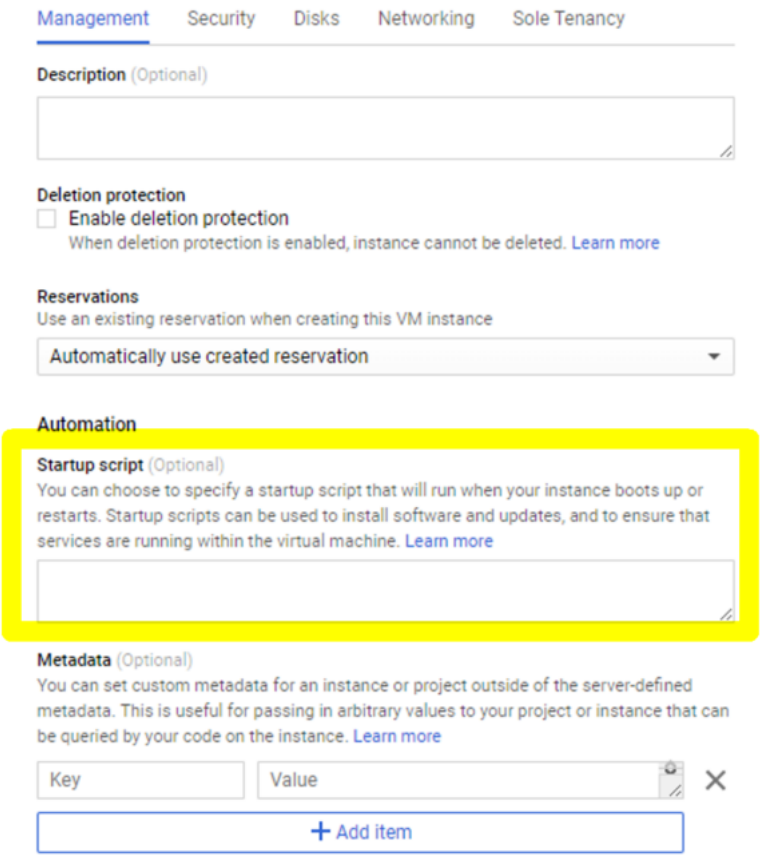
カスタムデータまたはプレブート設定を「—custom-data」パラメーターにファイルとして渡すことができます。この例では、ファイル名は **MyCloudInitScript.txt** です。

詳細については、Azure CLI のドキュメントを参照してください。

GCP コンソールを使用してプレブートユーザーデータを提供する

GCP コンソールを使用して Citrix ADC VPX インスタンスをプロビジョニングする場合は、インスタンスのプロパティを入力します。管理、セキュリティ、ディスク、ネットワーキング、単独テナンシを展開します。[管理] タブに移動します。[自動化] セクションで、[スタートアップスクリプト] フィールドにプリブートユーザーデータ設定を指定します。

GCP を使用した VPX インスタンスの作成の詳細については、「Google Cloud Platform での Citrix ADC VPX インスタンスの展開」を参照してください。



gcloud CLI を使用してプレブートユーザーデータを提供する

GCP CLI で次のコマンドを入力します。

```
gcloud compute instances create INSTANCE_NAMES --metadata-from-file=
    startup-script=LOCAL_FILE_PATH
<!--NeedCopy-->
```

metadata-from-file -に格納されているファイルから値またはユーザーデータを読み取ります。<LOCAL_FILE_PATH>。

詳細については、gcloud CLI ドキュメントを参照してください。

プレブートユーザーデータ形式

プレブートユーザーデータは XML 形式でクラウドインスタンスに提供する必要があります。起動時にクラウドインフラストラクチャを介して提供される Citrix ADC プレブートユーザーデータは、次の 4 つのセクションで構成されます。

- Citrix ADC 構成は<NS-CONFIG>タグで表されます。
- <NS-BOOTSTRAP>タグで表される Citrix ADC をカスタムブートストラップします。
- <NS-SCRIPTS>タグで表される Citrix ADC にユーザースクリプトを保存する。
- <NS-LICENSE-CONFIG>タグで表されるプールライセンス構成。

前の 4 つのセクションは、ADC のプレブート構成内で任意の順序で提供できます。
プリブートユーザーデータを提供しながら、次のセクションに示す書式に厳密に従うようにしてください。

> 注:
>
> 次の例に示すように、プレブートユーザーデータ構成全体を<NS-PRE-BOOT-CONFIG>タグで囲む必要があります。

例 **1**:

```
<NS-PRE-BOOT-CONFIG>
      <NS-CONFIG>           </NS-CONFIG>
      <NS-BOOTSTRAP>        </NS-BOOTSTRAP>
      <NS-SCRIPTS>          </NS-SCRIPTS>
      <NS-LICENSE-CONFIG>   </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

例 **2**:

```
<NS-PRE-BOOT-CONFIG>
     <NS-LICENSE-CONFIG> </NS-LICENSE-CONFIG>
     <NS-SCRIPTS>        </NS-SCRIPTS>
     <NS-BOOTSTRAP>      </NS-BOOTSTRAP>
     <NS-CONFIG>         </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

<NS-CONFIG>タグを使用して、プレブート段階で VPX インスタンスに適用する必要がある特定の Citrix ADC VPX 構成を指定します。この構成は、新しい「ns.conf」ファイルに格納されます。ADC が初めて起動されると、「ns.conf」ファイルに格納されている構成が VPX インスタンスに適用されます。

> 注:
>
> <NS-CONFIG>セクションには、有効な ADC CLI コマンドが必要です。CLI は、構文エラーまたは形式について検証されません。

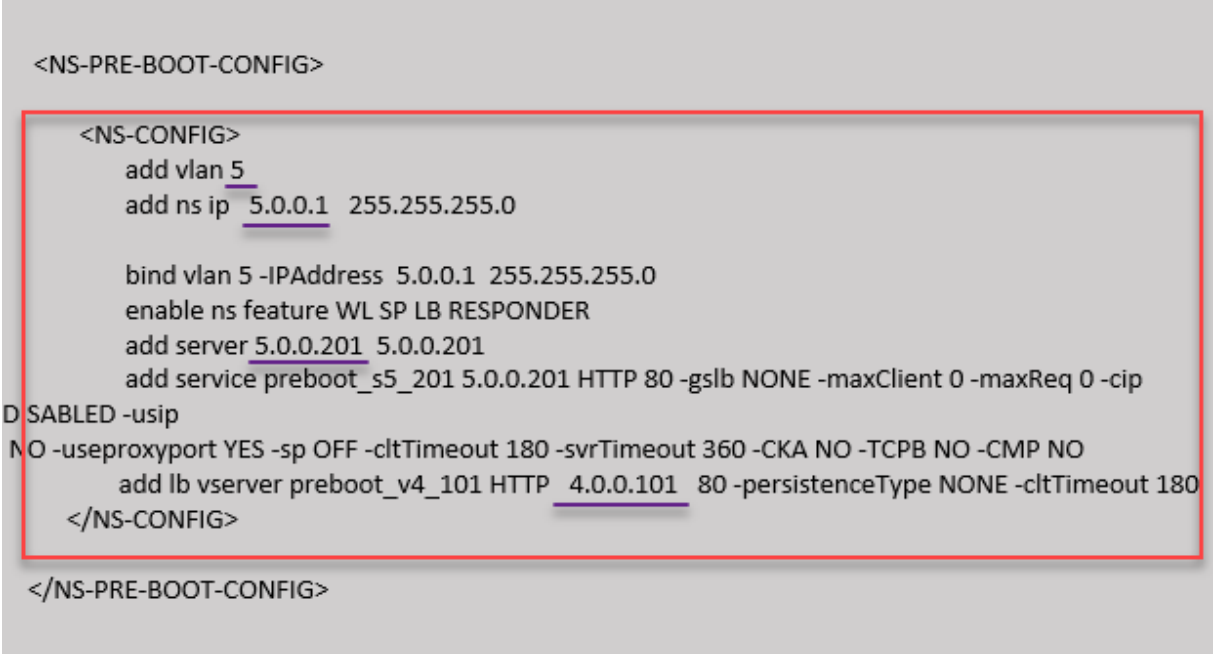
Citrix ADC 構成

<NS-CONFIG>タグを使用して、プレブート段階で VPX インスタンスに適用する必要がある特定の Citrix ADC VPX 構成を指定します。この構成は、新しい「ns.conf」ファイルに格納されます。ADC が初めて起動されると、「ns.conf」ファイルに格納されている構成が VPX インスタンスに適用されます。

> 注:
>
> <NS-CONFIG>セクションには、有効な ADC CLI コマンドが必要です。CLI は、構文エラーまたは形式について検証されません。

例:

次の例では、<NS-CONFIG>セクションに設定の詳細を示します。ID '5' の VLAN が設定され、SNIP（5.0.0.1）にバインドされます。負荷分散仮想サーバー (4.0.0.101) も構成されています。



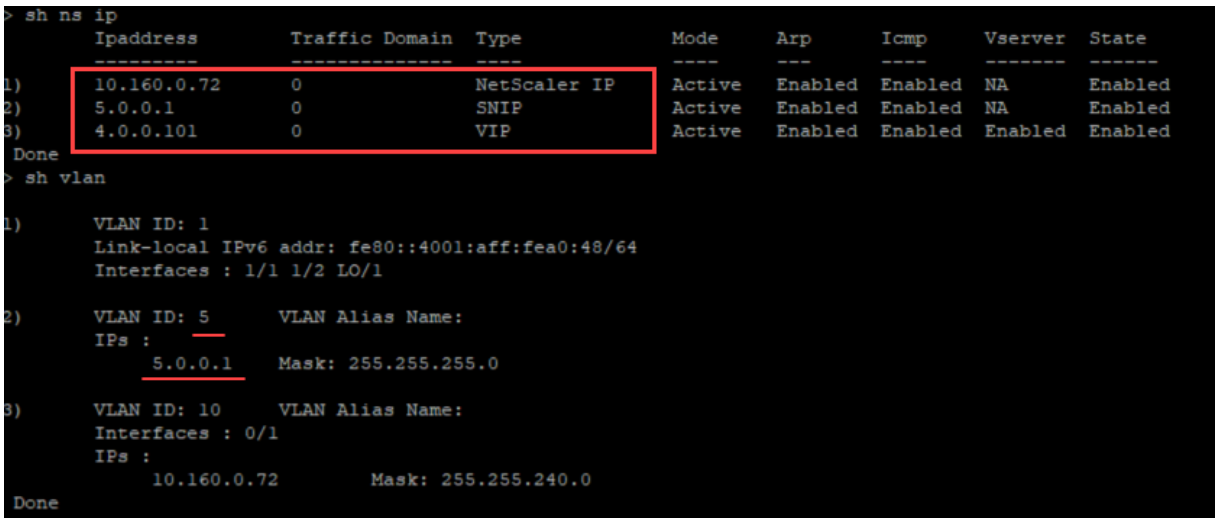
前のスクリーンショットに示した設定をここからコピーできます。

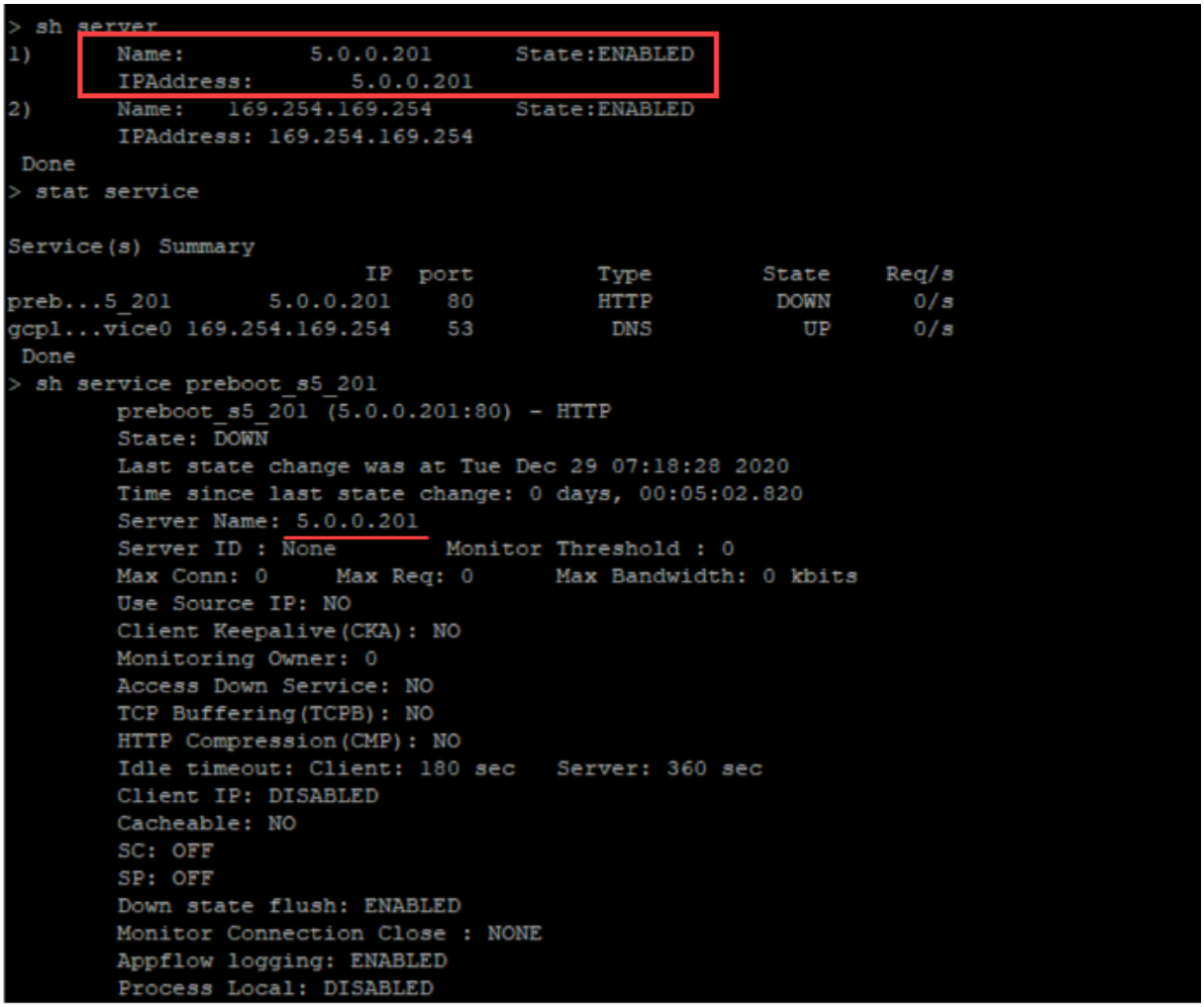
```
<NS-PRE-BOOT-CONFIG>
     <NS-CONFIG>
         add vlan 5
         add ns ip 5.0.0.1 255.255.255.0
         bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
         enable ns feature WL SP LB RESPONDER
         add server 5.0.0.201 5.0.0.201
         add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip
```



```
 9   NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -
        TCPB NO -CMP NO
10           add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
                 persistenceType NONE -cltTimeout 180
11        </NS-CONFIG>
12  </NS-PRE-BOOT-CONFIG>
13  <!--NeedCopy-->
```

Citrix ADC VPX インスタンスは、次の図に示すように、<NS-CONFIG>セクションに適用された構成を表示します。





ユーザースクリプト

<NS-SCRIPTS>タグを使用して、Citrix ADC VPX インスタンスに保存して実行する必要があるスクリプトを指定します。

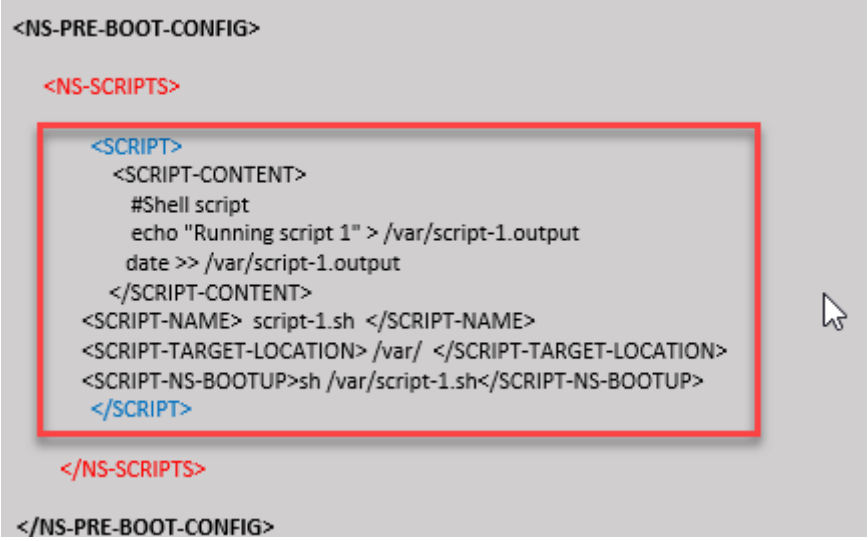
<NS-SCRIPTS>タグには多数のスクリプトを含めることができます。各スクリプトは<SCRIPT>タグ内に含める必要があります。
各<SCRIPT>セクションは 1 つのスクリプトに対応し、次のサブタグを使用してスクリプトの詳細をすべて含みます。

- **<SCRIPT-NAME>:**保存する必要のあるスクリプトファイルの名前を示します。
- **<SCRIPT-CONTENT>:**保存する必要のあるファイルの内容を示します。
- **<SCRIPT-TARGET-LOCATION>:**このファイルを保存する必要がある指定されたターゲットの場所を示します。ターゲットの場所が指定されていない場合、デフォルトでは、ファイルまたはスクリプトは「/nsconfig」ディレクトリに保存されます。
- **<SCRIPT-NS-BOOTUP>:**スクリプトの実行に使用するコマンドを指定します。

- セクションを使用する場合、<SCRIPT-NS-BOOTUP>セクションで提供されるコマンドは「/nsconfig/nsafter.sh」に格納され、「nsafter.sh」実行の一部としてパケットエンジンが起動した後にコマンドが実行されます。
- <SCRIPT-NS-BOOTUP>セクションを使用しない場合、スクリプトファイルは指定したターゲットの場所に保存されます。

例 **1**:

この例では、<NS-SCRIPTS>タグには script-1.sh というスクリプトの詳細が 1 つだけ含まれています。「script-1.sh」スクリプトは「/var」ディレクトリに保存されます。スクリプトには指定された内容が設定され、パケットエンジンの起動後に「sh /var/script-1.sh」コマンドで実行されます。



前のスクリーンショットに示した設定をここからコピーできます。

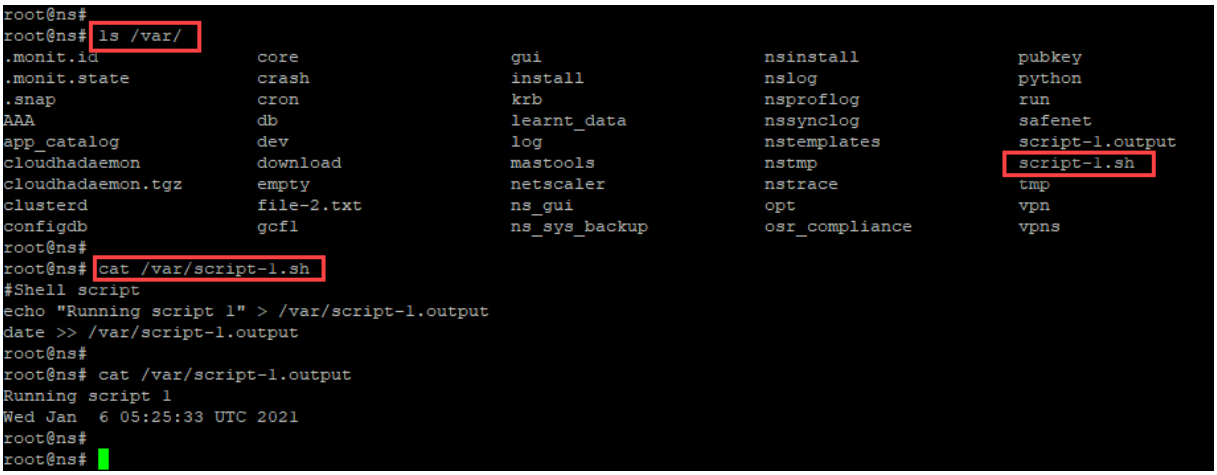
```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
    <SCRIPT>
            <SCRIPT-CONTENT>
                #Shell script
                echo ”Running script 1” > /var/script-1.output
                date >> /var/script-1.output
            </SCRIPT-CONTENT>

                <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
                <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION
                    >
                <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP
                    >
        </SCRIPT>
    </NS-SCRIPTS>
```



```
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

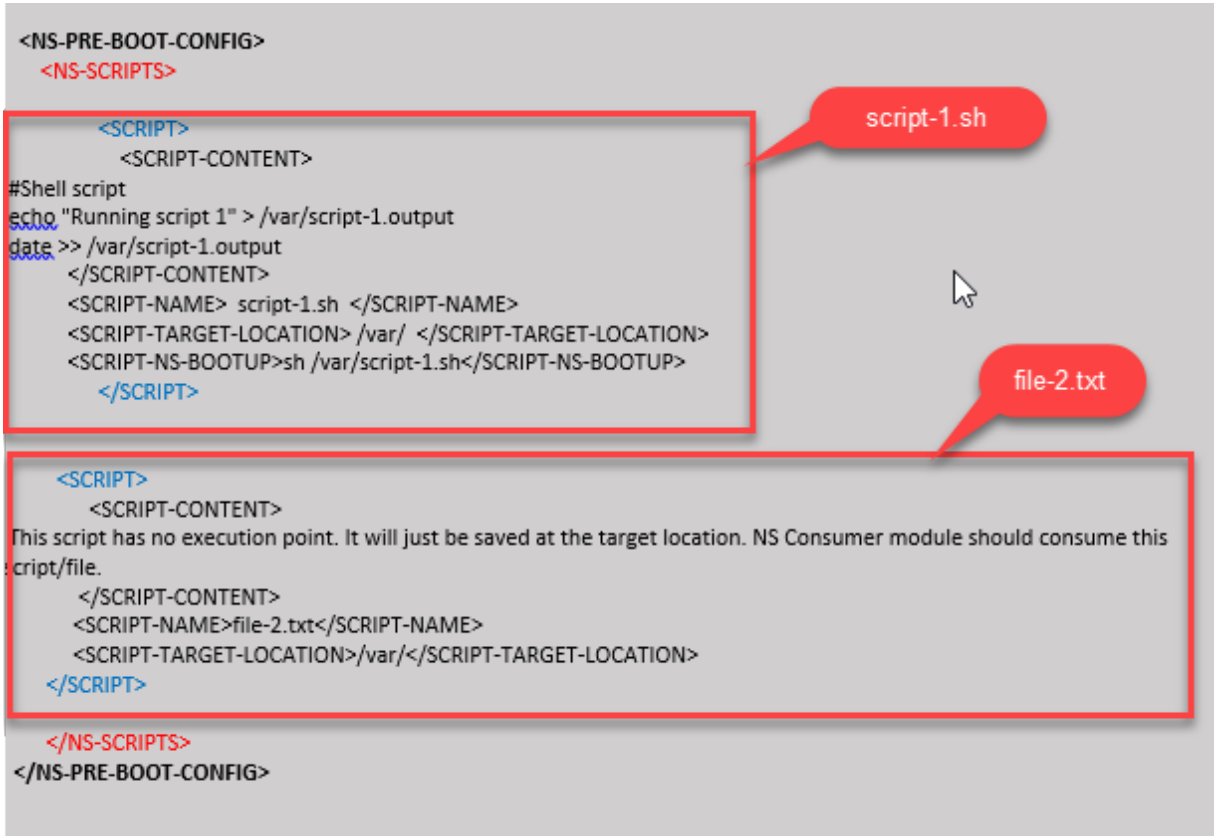
次のスナップショットでは、「script-1.sh」スクリプトが「/var/」ディレクトリに保存されていることを確認できます。”Script-1.sh” スクリプトが実行され、出力ファイルが適切に作成されます。



例 **2**:

次の例では、<NS-SCRIPTS>タグに 2 つのスクリプトの詳細が含まれています。

- 最初のスクリプトは「script-1.sh」として「/var」ディレクトリに保存されます。スクリプトは指定された内容で入力され、パケットエンジンの起動後にコマンド「sh /var/script-1.sh」で実行されます。
- 2 番目のスクリプトは、「/var」ディレクトリに「file-2.txt」として保存されます。このファイルには、指定されたコンテンツが入力されます。しかし、ブートアップ実行コマンド<SCRIPT-NS-BOOTUP>が提供されていないため、実行されません。




```
<NS-PRE-BOOT-CONFIG>
  <NS-SCRIPTS>

        <SCRIPT>
          <SCRIPT-CONTENT>
#Shell script
echo "Running script 1" > /var/script-1.output
date >> /var/script-1.output
      </SCRIPT-CONTENT>
      <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
      <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION>
      <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
        </SCRIPT>

    <SCRIPT>
      <SCRIPT-CONTENT>
This script has no execution point. It will just be saved at the target location. NS Consumer module should consume this
cript/file.
      </SCRIPT-CONTENT>
      <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
      <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
   </SCRIPT>

  </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```

前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
        <SCRIPT>
            <SCRIPT-CONTENT>
               #Shell script
               echo ”Running script 1” > /var/script-1.output
               date >> /var/script-1.output
            </SCRIPT-CONTENT>

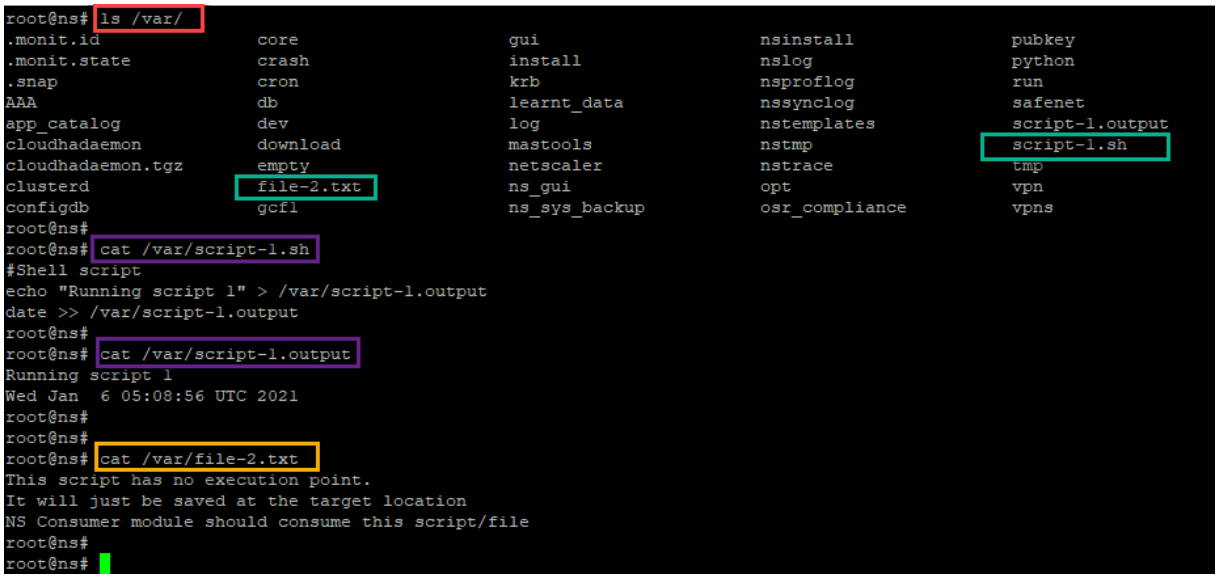
            <SCRIPT-NAME>  script-1.sh  </SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION> /var/  </SCRIPT-TARGET-LOCATION>
            <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
            </SCRIPT>

        <SCRIPT>
            <SCRIPT-CONTENT>
                This script has no execution point.
                It will just be saved at the target location
                NS Consumer module should consume this script/file
```



```
            </SCRIPT-CONTENT>
            <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

次のスナップショットでは、script-1.sh と file-2.txt が「/var/」ディレクトリ内に作成されていることを確認できます。Script-1.sh が実行され、出力ファイルが適切に作成されます。



ライセンス

VPX インスタンスの起動中に Citrix ADC プールライセンスを適用するには、<NS-LICENSE-CONFIG>タグを使用します。<NS-LICENSE-CONFIG>セクション内の<LICENSE-COMMANDS>タグを使用して、プールされたライセンスコマンドを指定します。これらのコマンドは構文的に有効である必要があります。

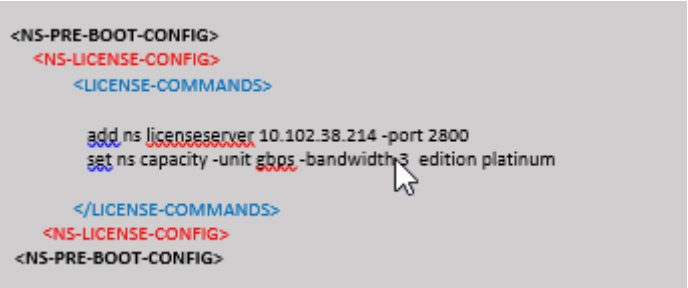
標準のプールライセンスコマンドを使用して、<LICENSE-COMMANDS>セクションで、ライセンスタイプ、容量、ライセンスサーバーなどのプールされたライセンスの詳細を指定できます。詳細については、「Citrix ADC プール容量ライセンスの構成」を参照してください。

<NS-LICENSE-CONFIG>を適用した後、VPX は起動時に要求されたエディションを起動し、VPX はライセンスサーバから構成されたライセンスをチェックアウトしようとします。

- ライセンスのチェックアウトが成功すると、構成された帯域幅が VPX に適用されます。
- ライセンスのチェックアウトに失敗した場合、約 10 ～12 分以内にライセンスはライセンスサーバから取得されません。その結果、システムがリブートし、ライセンスなしの状態になります。

例:

次の例では、<NS-LICENSE-CONFIG>を適用した後、VPX は起動時にプレミアムエディションを起動し、VPX はライセンスサーバ（10.102.38.214）から構成されたライセンスをチェックアウトしようとします。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>
   <NS-LICENSE-CONFIG>
        <LICENSE-COMMANDS>
            add ns licenseserver 10.102.38.214 -port 2800
            set ns capacity -unit gbps -bandwidth 3  edition platinum
        </LICENSE-COMMANDS>
    </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

次の図に示すように、「show license server」コマンドを実行し、ライセンスサーバー（10.102.38.214）が VPX に追加されていることを確認できます。

ブートストラッピング

<NS-BOOTSTRAP>タグを使用して、カスタムブートストラップ情報を指定します。<NS-BOOTSTRAP>セクション内では、<SKIP-DEFAULT-BOOTSTRAP>タグと<NEW-BOOTSTRAP-SEQUENCE>タグを使用できます。このセクションでは、デフォルトのブートストラップを回避するかどうかを Citrix ADC アプライアンスに通知します。デフォルトのブートストラップが回避される場合、このセクションでは、新しいブートストラップシーケンスを提供するオプションを提供します。

デフォルトのブートストラップ構成

Citrix ADC アプライアンスのデフォルトのブートストラップ構成は、次のインターフェイスの割り当てに従います。

- **Eth0** -特定の NSIP アドレスを持つ管理インターフェイス。
- **Eth1** -特定の VIP アドレスを持つクライアント向けインターフェイス。
- **Eth2** -特定の SNIP アドレスを持つサーバー側インターフェイス。

ブートストラップ構成をカスタマイズする

デフォルトのブートストラップシーケンスをスキップして、Citrix ADC VPX インスタンスに新しいブートストラップシーケンスを指定することができます。<NS-BOOTSTRAP>タグを使用して、カスタムブートストラップ情報を指定します。たとえば、管理インターフェイス（NSIP）、クライアント側インターフェイス（VIP）、およびサーバー側インターフェイス（SNIP）が常に特定の順序で提供されるデフォルトのブートストラップを変更できます。

次の表に、<SKIP-DEFAULT-BOOTSTRAP>および<NEW-BOOTSTRAP-SEQUENCE>タグで許可されるさまざまな値を使用したブートストラップ動作を示します。

SKIP-DEFAULT-BOOTSTRAP	NEW-BOOTSTRAP-SEQUENCE	ブートストラップ動作
はい	はい	デフォルトのブートストラップ動作はスキップされ、<NS-BOOTSTRAP>セクションで提供される新しいカスタムブートストラップシーケンスが実行されます。
はい	いいえ	デフォルトのブートストラップ動作はスキップされ、<NS-CONFIG>セクションで提供されているブートストラップコマンドが実行されます。

ブートストラップ構成は、次の 3 つの方法でカスタマイズできます。

- インターフェイスの詳細のみを入力します。
- IP アドレスとサブネットマスクとともにインターフェイスの詳細を指定します。
- <NS-CONFIG>セクションにブートストラップ関連のコマンドを入力します。

方法 **1:** インターフェイスの詳細のみを指定してカスタムブートストラップ

管理インターフェイス、クライアント向けインターフェイス、およびサーバ側インターフェイスは指定しますが、その IP アドレスとサブネットマスクは指定しません。IP アドレスとサブネットマスクは、クラウドインフラストラクチャのクエリによって設定されます。

AWS のカスタムブートストラップの例

次の例に示すように、カスタムブートストラップシーケンスを指定します。詳細については、「クラウドインスタンスでプレブートユーザーデータを提供する方法」を参照してください。Eth1 インターフェイスは管理インターフェイス（NSIP）、クライアントインターフェイス（VIP）として Eth0 インターフェイス、サーバインターフェイス（SNIP）として Eth2 インターフェイスが割り当てられます。<NS-BOOTSTRAP>セクションには、インターフェイスの詳細のみが含まれ、IP アドレスとサブネットマスクの詳細は含まれません。

```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

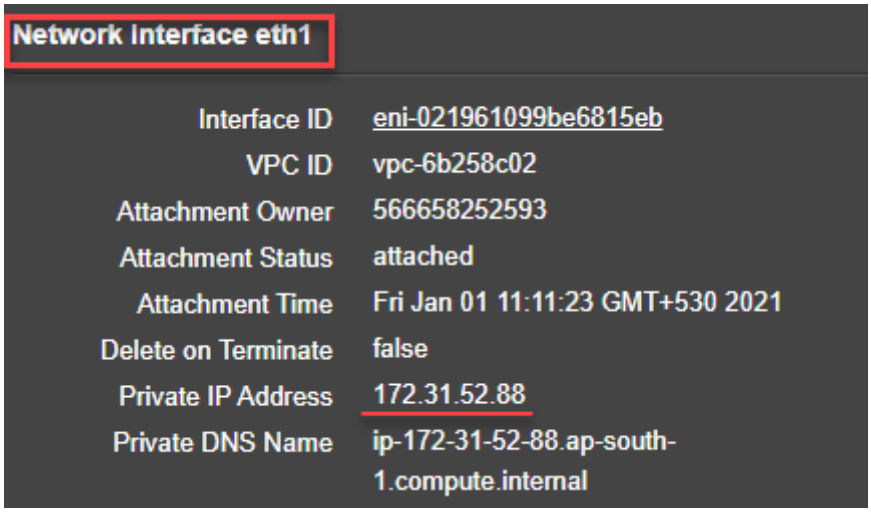
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

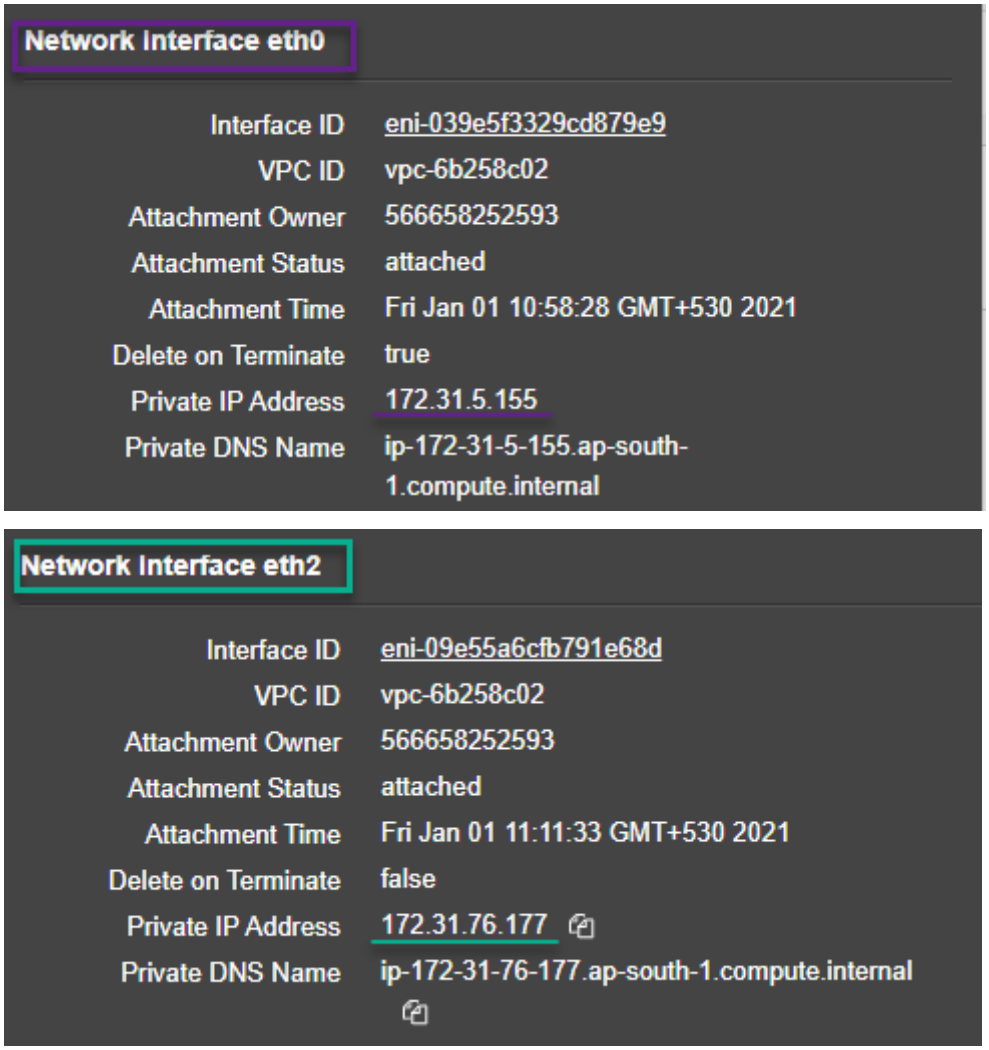
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

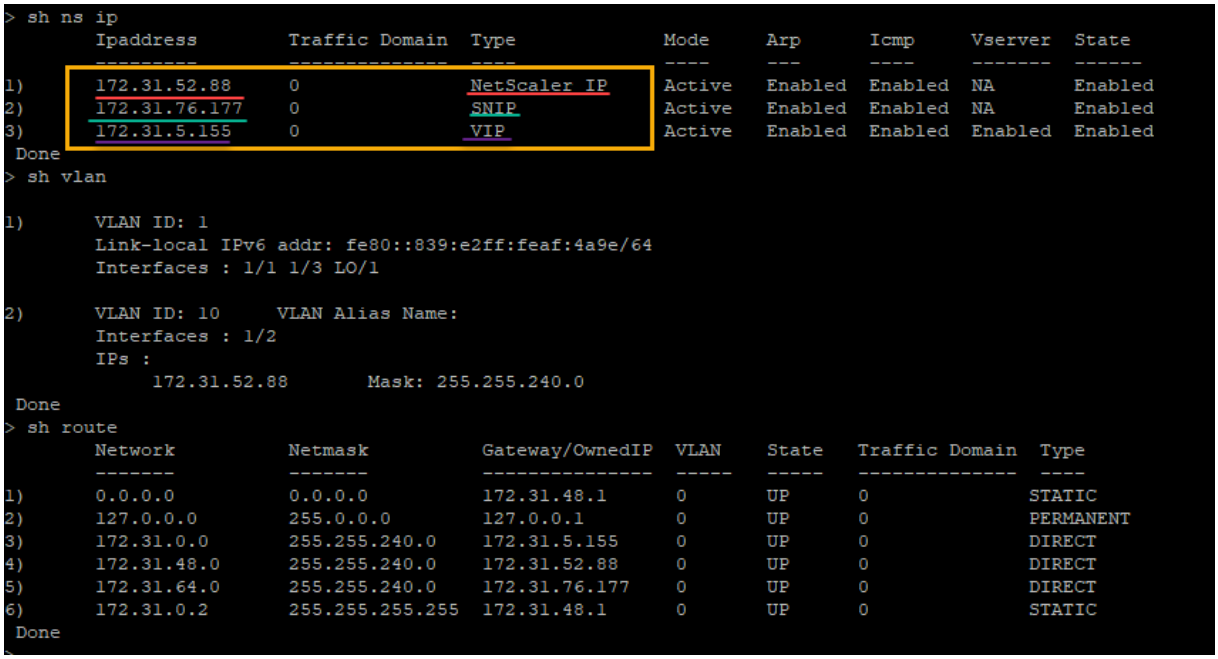
VM インスタンスの作成後、AWS ポータルで、ネットワークインターフェイスのプロパティを次のように確認できます。

1. [**AWS Portal] > [EC2** インスタンス] に移動し、カスタムブートストラップ情報を指定して作成したインスタンスを選択します。
2. [説明] タブでは、次の図に示すように、各ネットワークインタフェースのプロパティを確認できます。





ADC CLI で`show nsip`コマンドを実行し、**ADC** アプライアンスの最初の起動時に ADC VPX インスタンスに適用されたネットワークインターフェイスを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
```

Azure のカスタムブートストラップの例

次の例に示すように、カスタムブートストラップシーケンスを指定します。詳細については、「クラウドインスタンスでプレブートユーザーデータを提供する方法」を参照してください。Eth2 インターフェイスは、管理インターフェイス（NSIP）として、Eth1 インターフェイスをクライアントインターフェイス（VIP）として、Eth0 インターフェイスをサーバインターフェイス（SNIP）として割り当てます。<NS-BOOTSTRAP>セクションには、インターフェイスの詳細のみが含まれ、IP アドレスとサブネットマスクの詳細は含まれません。


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

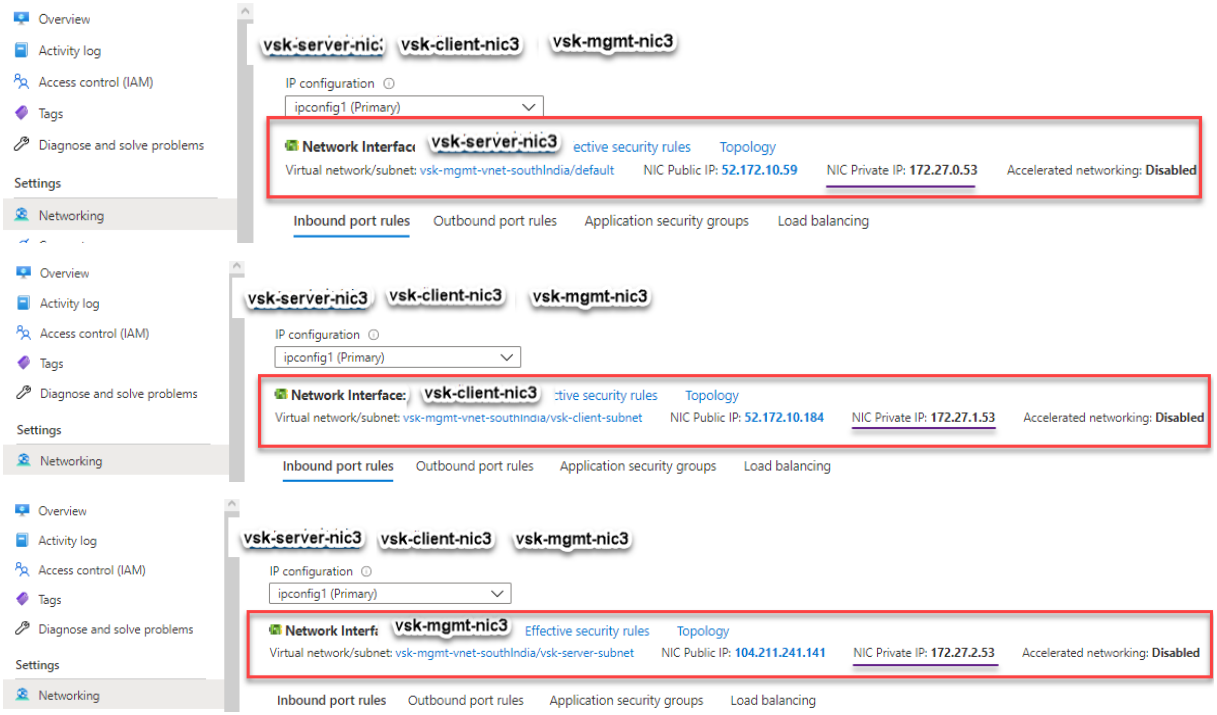
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

Citrix ADC VPX インスタンスが 3 つのネットワークインターフェイスで作成されていることがわかります。**Azure Portal > VM** インスタンス **>** ネットワークに移動し、次の図に示すように 3 つの NIC のネットワークプロパティを確認します。



ADC CLI で「show nsip」コマンドを実行し、<NS−BOOTSTRAP>セクションで指定された新しいブートストラッ

プシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
>
```

GCP のカスタムブートストラップの例

次の例に示すように、カスタムブートストラップシーケンスを指定します。詳細については、「クラウドインスタンスでプレブートユーザーデータを提供する方法」を参照してください。Eth1 インターフェイスは管理インターフェイス（NSIP）、クライアントインターフェイス（VIP）として Eth0 インターフェイス、サーバインターフェイス（SNIP）として Eth2 インターフェイスが割り当てられます。<NS-BOOTSTRAP>セクションには、インターフェイスの詳細のみが含まれ、IP アドレスとサブネットマスクの詳細は含まれません。


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

GCP ポータルで VM インスタンスを作成した後、ネットワークインターフェイスのプロパティを次のように確認できます。

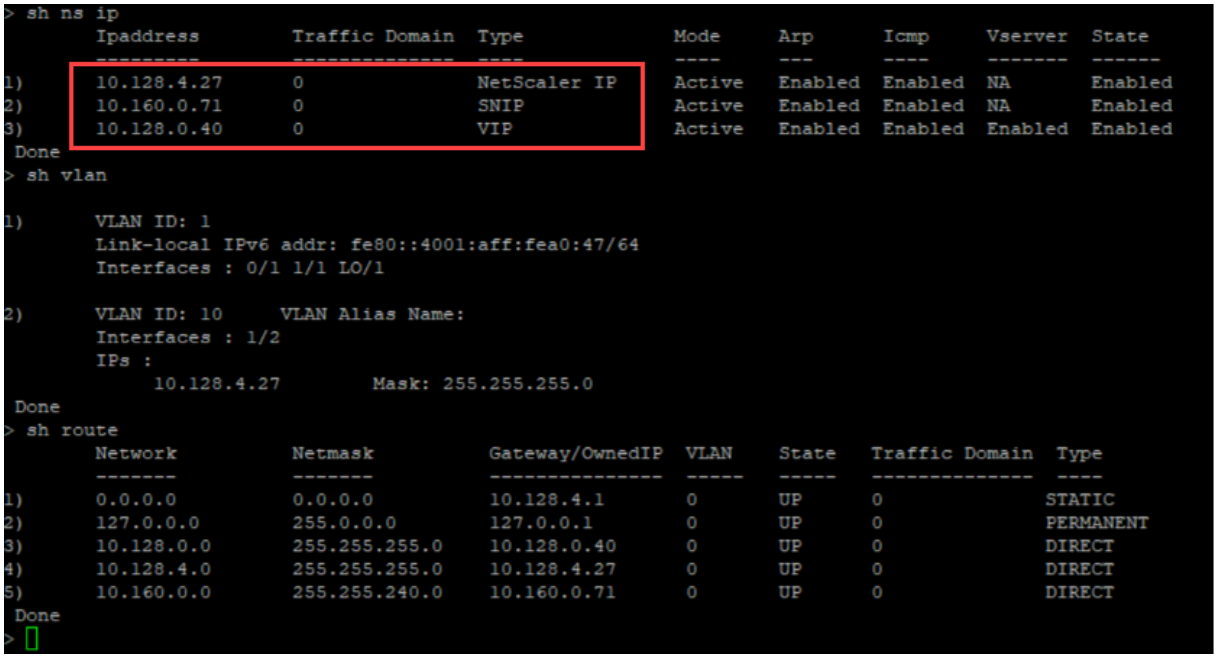
1. カスタムブートストラップ情報を指定して、作成したインスタンスを選択します。
2. ネットワークインターフェイスのプロパティに移動し、NIC の詳細を次のように確認します。

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	10.160.0.71	–	35.244.56.180 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	10.128.0.40	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	10.128.4.27	–	34.93.241.147 (ephemeral)	Premium		View details

Public DNS PTR Record
None

ADC CLI で`show nsip`コマンドを実行し、**ADC** アプライアンスの最初の起動時に ADC VPX インスタンスに適用されたネットワークインターフェイスを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -----
1)      10.128.4.27      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.160.0.71      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.128.0.40      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:47/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             10.128.4.27        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.4.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.40      0       UP      0               DIRECT
4)      10.128.4.0       255.255.255.0    10.128.4.27      0       UP      0               DIRECT
5)      10.160.0.0       255.255.240.0    10.160.0.71      0       UP      0               DIRECT
 Done
>
```

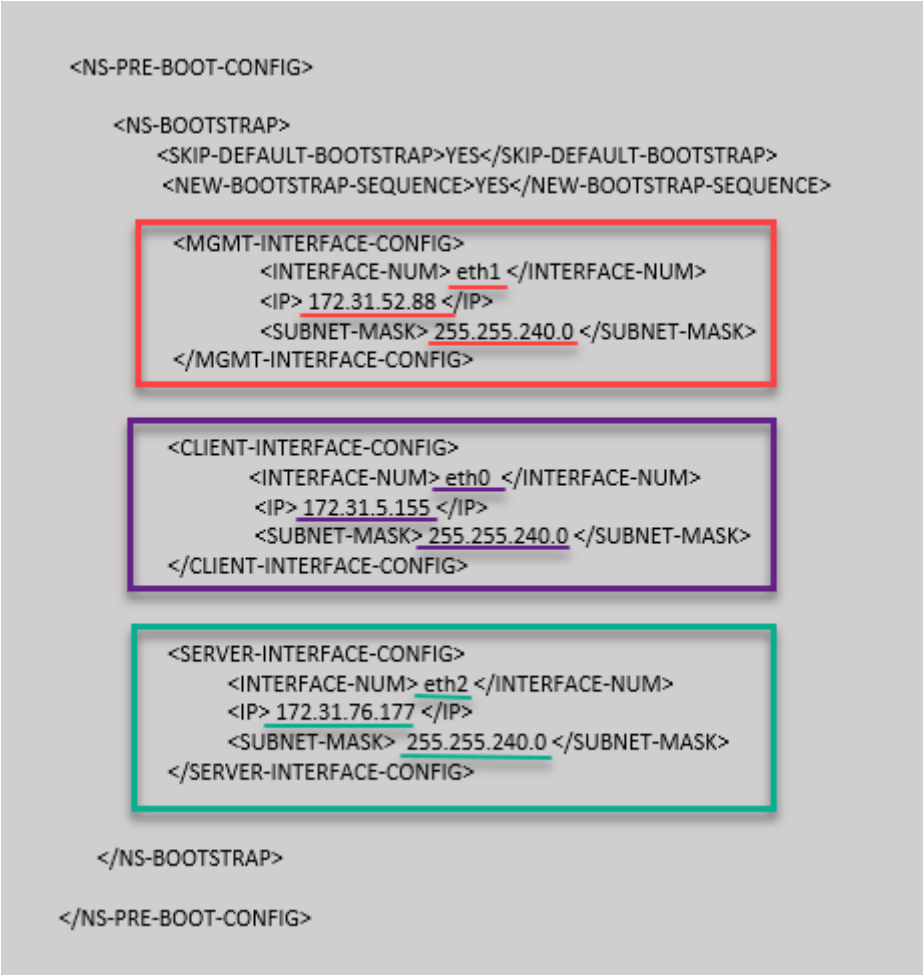
方法 **2:** インターフェイス、**IP** アドレス、およびサブネットマスクを指定してカスタムブートストラップ

管理インターフェイス、クライアント向けインターフェイス、およびサーバ向けインターフェイスと IP アドレスとサブネットマスクを指定します。

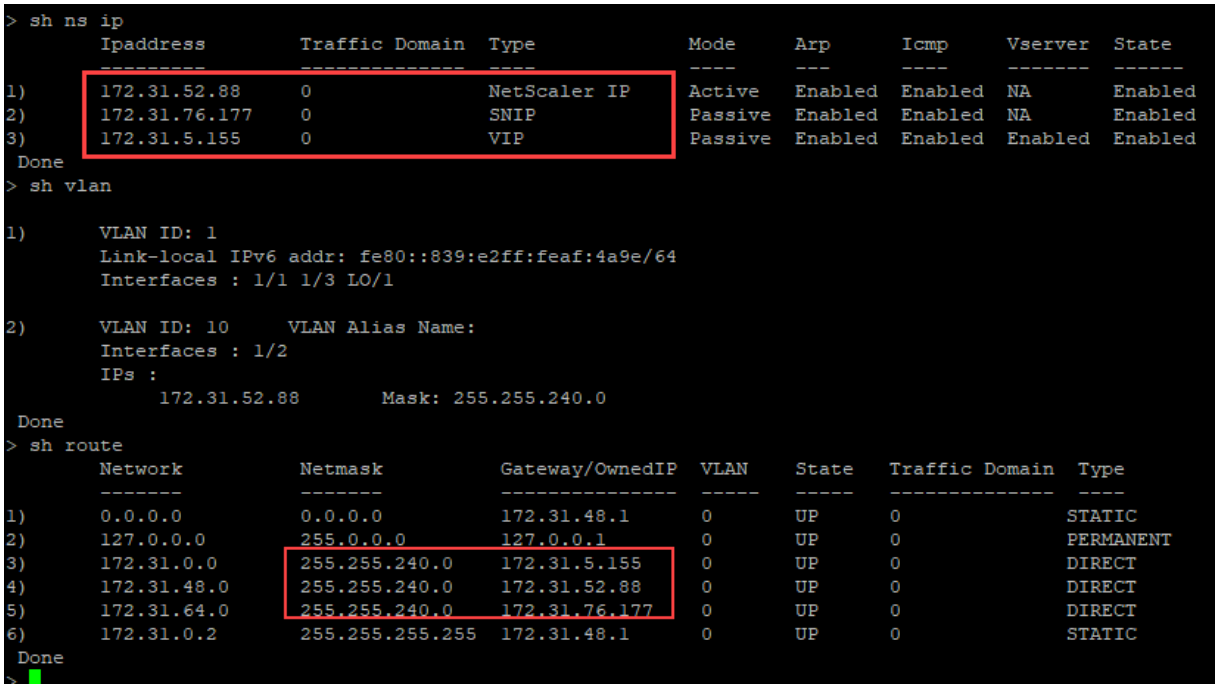
AWS のカスタムブートストラップの例

次の例では、デフォルトのブートストラップをスキップして、Citrix ADC アプライアンスの新しいブートストラップシーケンスを実行します。新しいブートストラップシーケンスでは、次の詳細を指定します。

- 管理インターフェイス：インターフェイス-Eth1、NSIP-172.31.52.88、およびサブネットマスク-255.255.240.0
- クライアント側インターフェイス：インターフェイス-Eth0、VIP-172.31.5.155、およびサブネットマスク-255.255.240.0。
- サーバー側インターフェイス：インターフェイス-Eth2、SNIP-172.31.76.177、サブネットマスク-255.255.240.0。



ADC CLI で`show nsip`コマンドを実行し、`<NS-BOOTSTRAP>`セクションで指定された新しいブートストラップシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -----
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

Azure のカスタムブートストラップの例

次の例では、ADC の新しいブートストラップシーケンスが記述され、デフォルトのブートストラップがスキップされます。インターフェイスの詳細と IP アドレスとサブネットマスクを次のように指定します。

- 管理インターフェイス（eth2）、NSIP（172.27.2.53）、およびサブネットマスク（255.255.255.0）
- クライアント側インターフェイス（eth1）、VIP（172.27.1.53）、およびサブネットマスク（255.255.255.0）
- サーバー側インターフェイス (eth0)、SNIP (172.27.0.53)、およびサブネットマスク (255.255.255.0)


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 172.27.2.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>

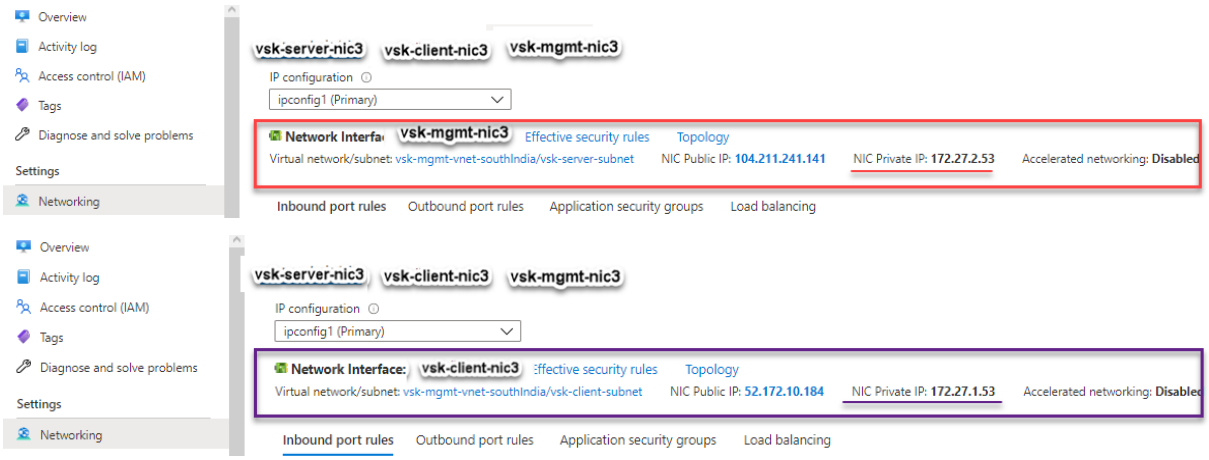
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 172.27.1.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
            </CLIENT-INTERFACE-CONFIG>

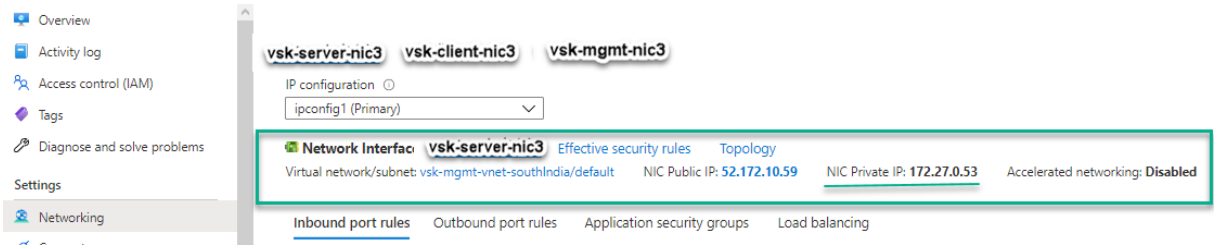
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 172.27.0.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

Citrix ADC VPX インスタンスが 3 つのネットワークインターフェイスで作成されていることがわかります。**Azure Portal > VM** インスタンス **>** ネットワークに移動し、次の図に示すように 3 つの NIC のネットワークプロパティを確認します。





ADC CLI で`show nsip`コマンドを実行し、<NS-BOOTSTRAP>セクションで指定された新しいブートストラップシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
```

GCP のカスタムブートストラップの例

次の例では、ADC の新しいブートストラップシーケンスが記述され、デフォルトのブートストラップがスキップされます。インターフェイスの詳細と IP アドレスとサブネットマスクを次のように指定します。

- 管理インターフェイス（eth2）、NSIP（10.128.4.31）、およびサブネットマスク（255.255.255.0）
- クライアント側インターフェイス（eth1）、VIP（10.128.0.43）、およびサブネットマスク（255.255.255.0）
- サーバ側インターフェイス（eth0）、SNIP（10.160.0.75）、およびサブネットマスク（255.255.255.0）


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>
        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 10.128.4.31 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 10.128.0.43 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </CLIENT-INTERFACE-CONFIG>
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 10.160.0.75 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

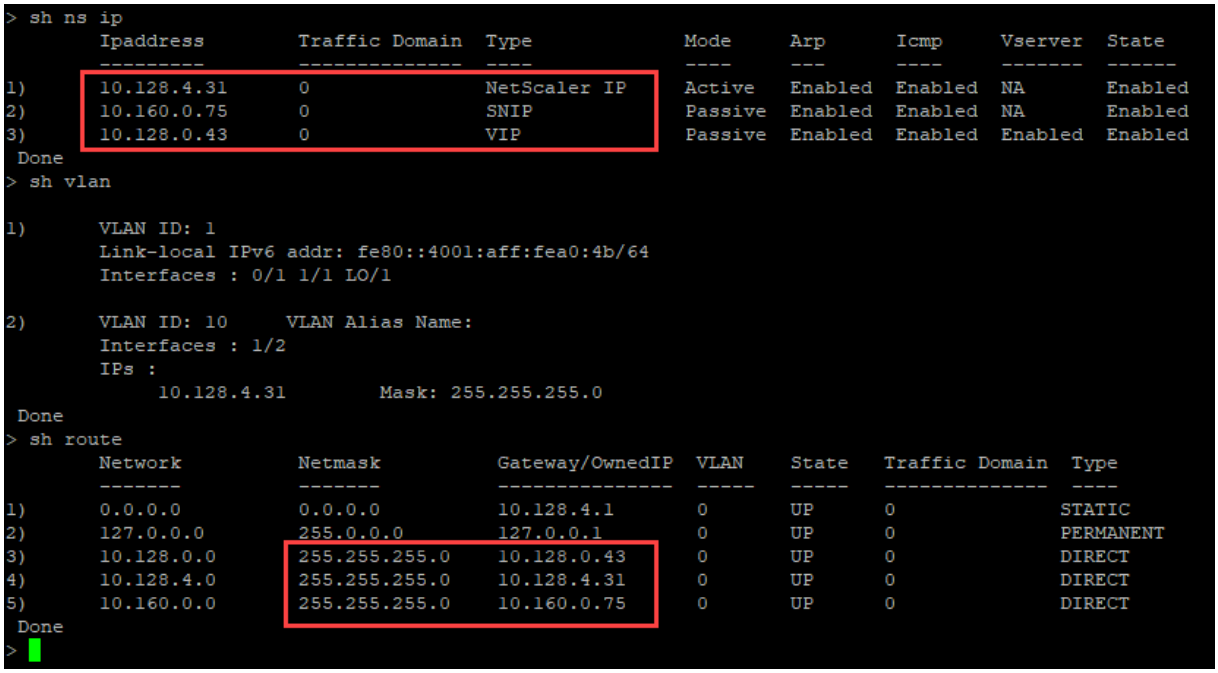
カスタムブートストラップを使用して GCP ポータルで VM インスタンスを作成した後、ネットワークインターフェイスのプロパティを次のように確認できます。

1. カスタムブートストラップ情報を指定して、作成したインスタンスを選択します。
2. ネットワークインターフェイスのプロパティに移動し、NIC の詳細を次のように確認します。

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	vsk-defnw-st-ip1 (10.160.0.75)	–	34.93.216.90 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	vsk-vpc-nw1-st-ip1 (10.128.0.43)	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	vsk-nw2-st-ip-1 (10.128.4.31)	–	34.93.202.214 (ephemeral)	Premium		View details

ADC CLI で`show nsip`コマンドを実行し、<NS-BOOTSTRAP>セクションで指定された新しいブートストラップシーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.4.31      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.160.0.75      0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      10.128.0.43      0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:4b/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             10.128.4.31        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.4.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.43      0       UP      0               DIRECT
4)      10.128.4.0       255.255.255.0    10.128.4.31      0       UP      0               DIRECT
5)      10.160.0.0       255.255.255.0    10.160.0.75      0       UP      0               DIRECT
 Done
>
```

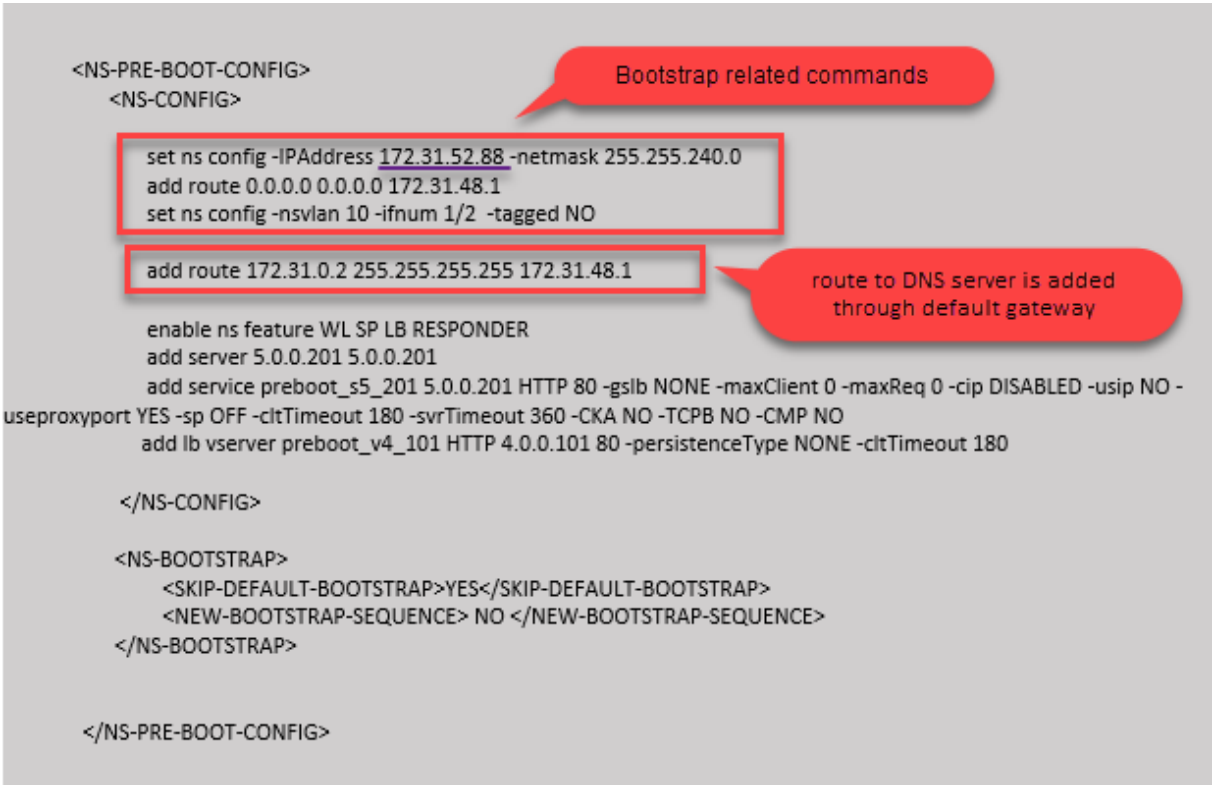
方法 **3: <NS-CONFIG>** セクションにブートストラップ関連のコマンドを指定して、カスタムブートストラップ

ブートストラップ関連のコマンドについては、<NS-CONFIG>セクションを参照してください。<NS-BOOTSTRAP>セクションで、<NS-CONFIG>セクションのブートストラップコマンドを実行するには、<NEW-BOOTSTRAP-SEQUENCE>を「No」に指定する必要があります。NSIP、デフォルトルート、および NSVLAN を割り当てるコマンドも指定する必要があります。さらに、使用するクラウドに関連するコマンドも提供します。

カスタムブートストラップを提供する前に、クラウドインフラストラクチャが特定のインターフェイス構成をサポートしていることを確認してください。

AWS のカスタムブートストラップの例

この例では、ブートストラップ関連のコマンドを<NS-CONFIG>セクションで提供しています。<NS-BOOTSTRAP>セクションは、デフォルトのブートストラップがスキップされ、<NS-CONFIG>セクションで提供されるカスタムブートストラップ情報が実行されることを示します。NSIP の作成、デフォルトルートの追加、および NSVLAN の追加を行うコマンドも指定する必要があります。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>

        set ns config -IPAddress 172.31.52.88 -netmask 255.255.240.0
        add route 0.0.0.0 0.0.0.0 172.31.48.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add route 172.31.0.2 255.255.255.255 172.31.48.1

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO - useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

    <NS-BOOTSTRAP>
```

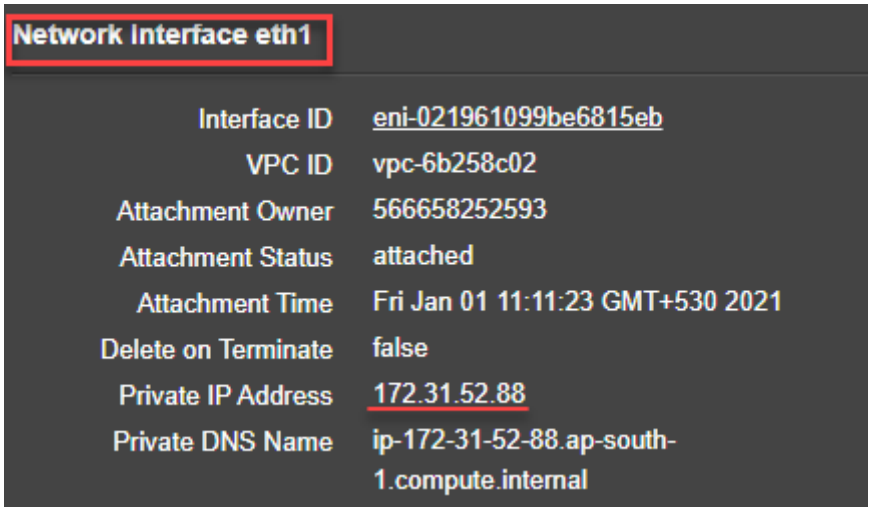


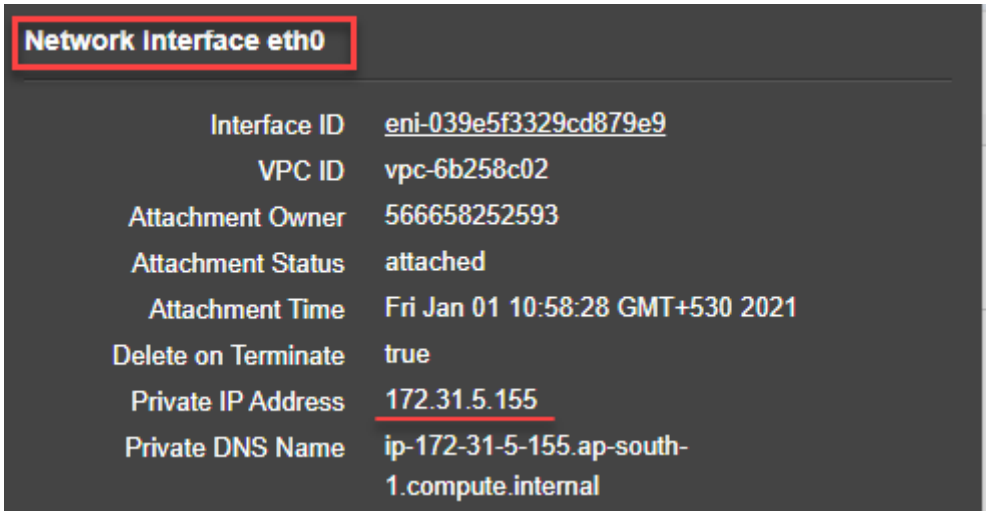
```
      <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
      <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
     </NS-BOOTSTRAP>

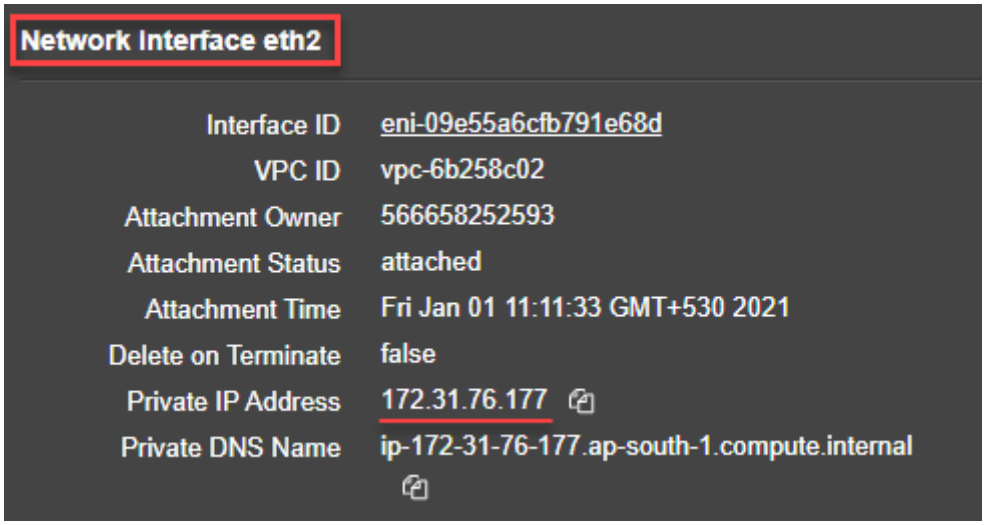

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

VM インスタンスの作成後、AWS ポータルで、ネットワークインターフェイスのプロパティを次のように確認できます。

1. [**AWS Portal] > [EC2** インスタンス] に移動し、カスタムブートストラップ情報を指定して作成したインスタンスを選択します。
2. [説明] タブでは、次の図に示すように、各ネットワークインタフェースのプロパティを確認できます。







ADC CLI で`show nsip`コマンドを実行し、**ADC** アプライアンスの最初の起動時に ADC VPX インスタンスに適用されたネットワークインターフェイスを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

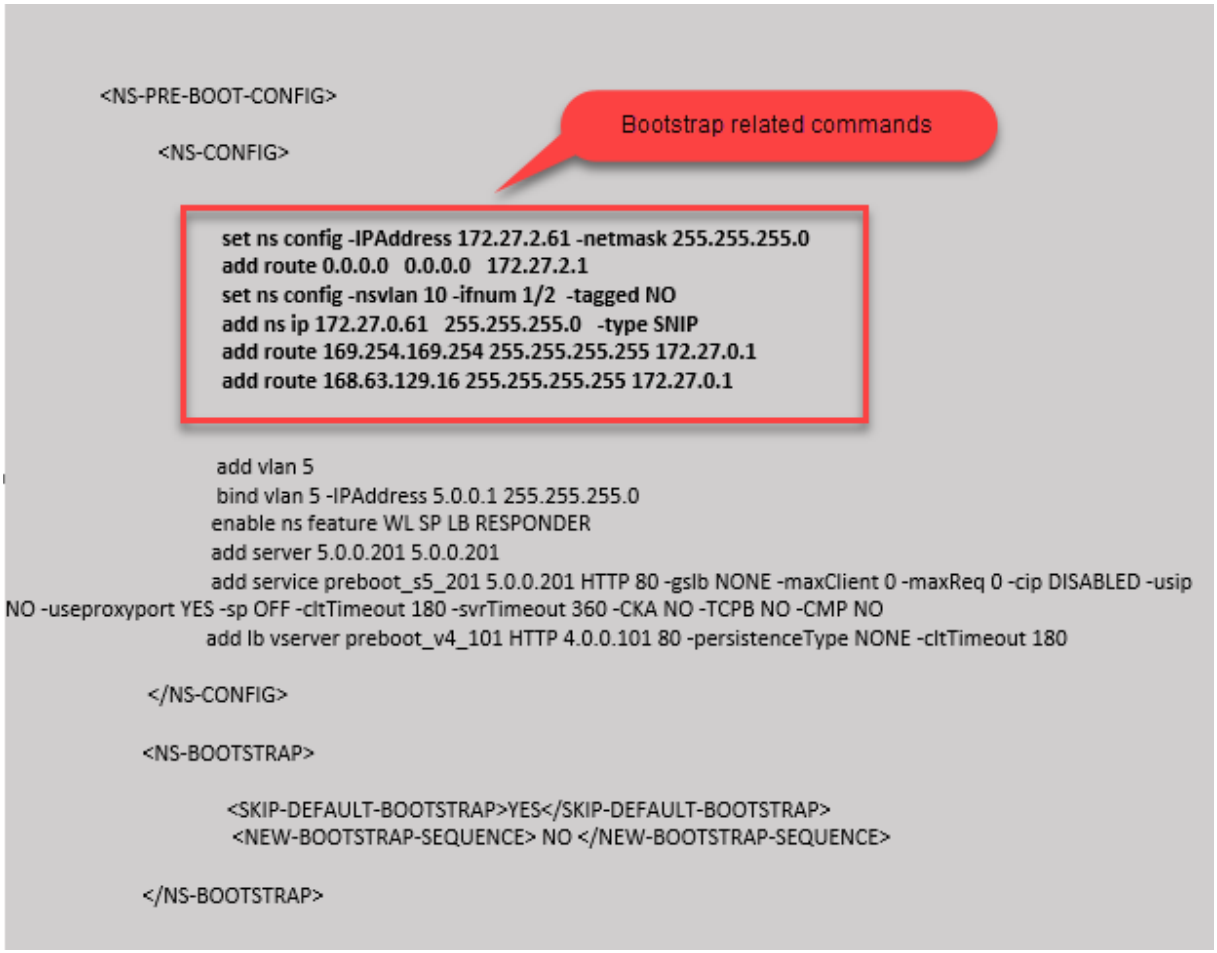
2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
4)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

Azure のカスタムブートストラップの例

この例では、ブートストラップ関連のコマンドを<NS-CONFIG>セクションで提供しています。<NS-BOOTSTRAP>セクションは、デフォルトのブートストラップがスキップされ、<NS-CONFIG>セクションで提供されるカスタムブートストラップ情報が実行されることを示します。

> 注:
>
> Azure クラウドの場合、インスタンスメタデータサーバー (IMDS) と DNS サーバーはプライマリインターフェイス (Eth0) を介してのみアクセスできます。したがって、Eth0 インターフェイスが管理インターフェイス（NSIP）として使用されない場合、Eth0 インターフェイスは、少なくとも IMDS または DNS アクセスを動作

させるには、SNIP として設定する必要があります。Eth0 のゲートウェイを経由する IMDS エンドポイント（169.254.169.254）および DNS エンドポイント（168.63.129.16）へのルートも追加する必要があります。




```
<NS-PRE-BOOT-CONFIG>

   <NS-CONFIG>

        set ns config -IPAddress 172.27.2.61 -netmask 255.255.255.0
        add route 0.0.0.0   0.0.0.0   172.27.2.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add ns ip 172.27.0.61   255.255.255.0   -type SNIP
        add route 169.254.169.254 255.255.255.255 172.27.0.1
        add route 168.63.129.16 255.255.255.255 172.27.0.1

        add vlan 5
        bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
```



```
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

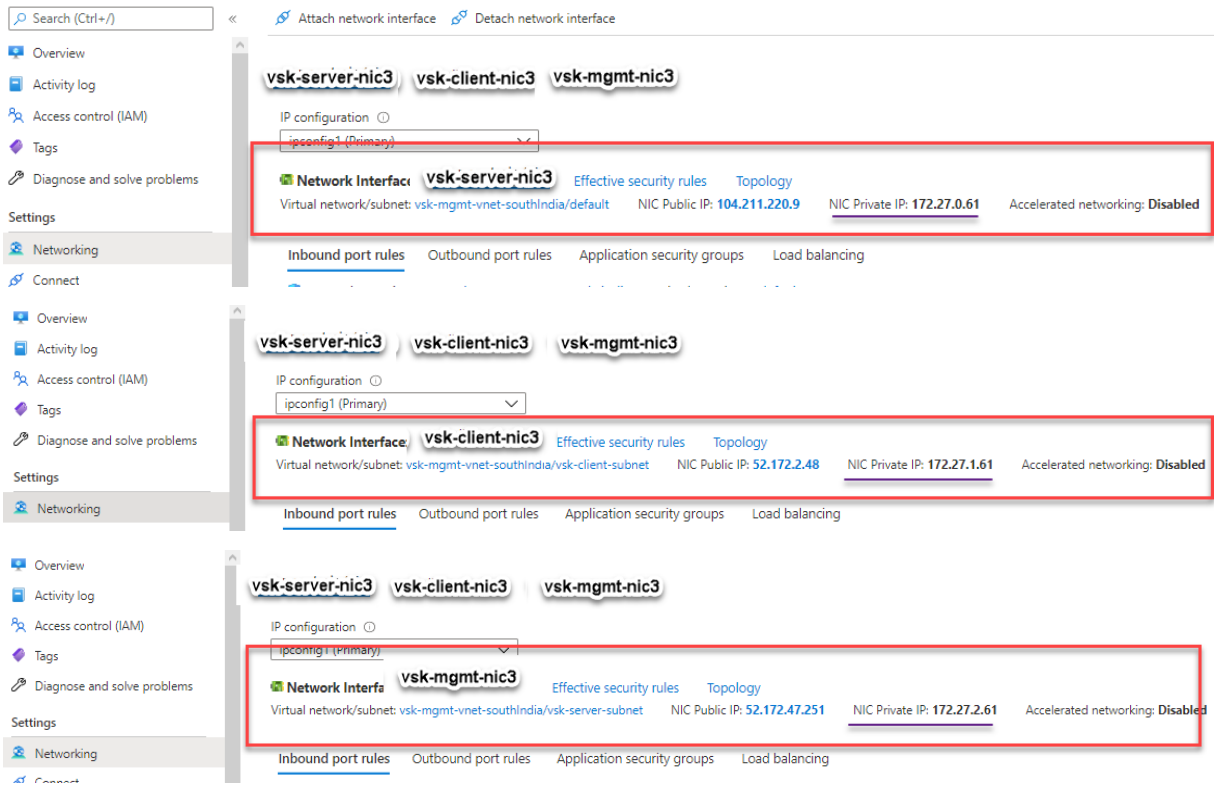
    <NS-BOOTSTRAP>

    <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
    <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>

    </NS-BOOTSTRAP>

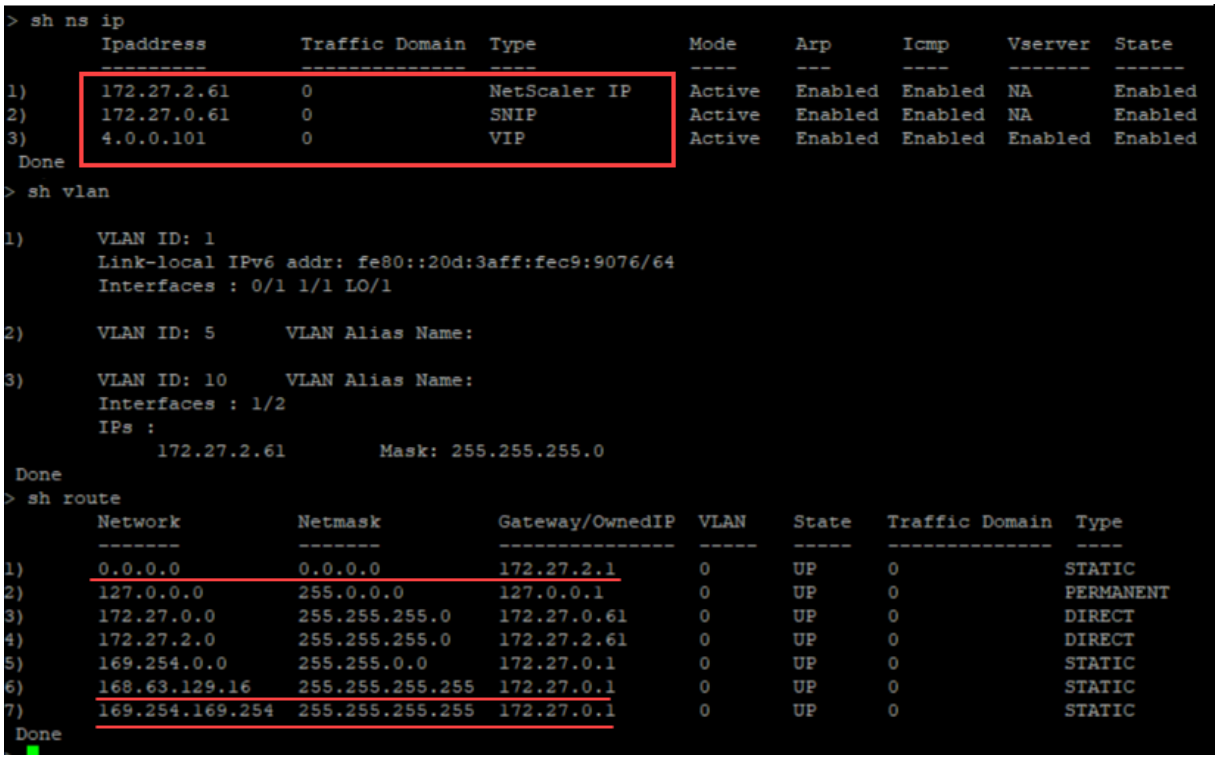
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

Citrix ADC VPX インスタンスが 3 つのネットワークインターフェイスで作成されていることがわかります。**Azure Portal > VM** インスタンス **>** ネットワークに移動し、次の図に示すように 3 つの NIC のネットワークプロパティを確認します。



ADC CLI で`show nsip`コマンドを実行し、<NS-BOOTSTRAP>セクションで指定された新しいブートストラップ

シーケンスが適用されていることを確認できます。「show route」コマンドを実行して、サブネットマスクを確認できます。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.61      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.61      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

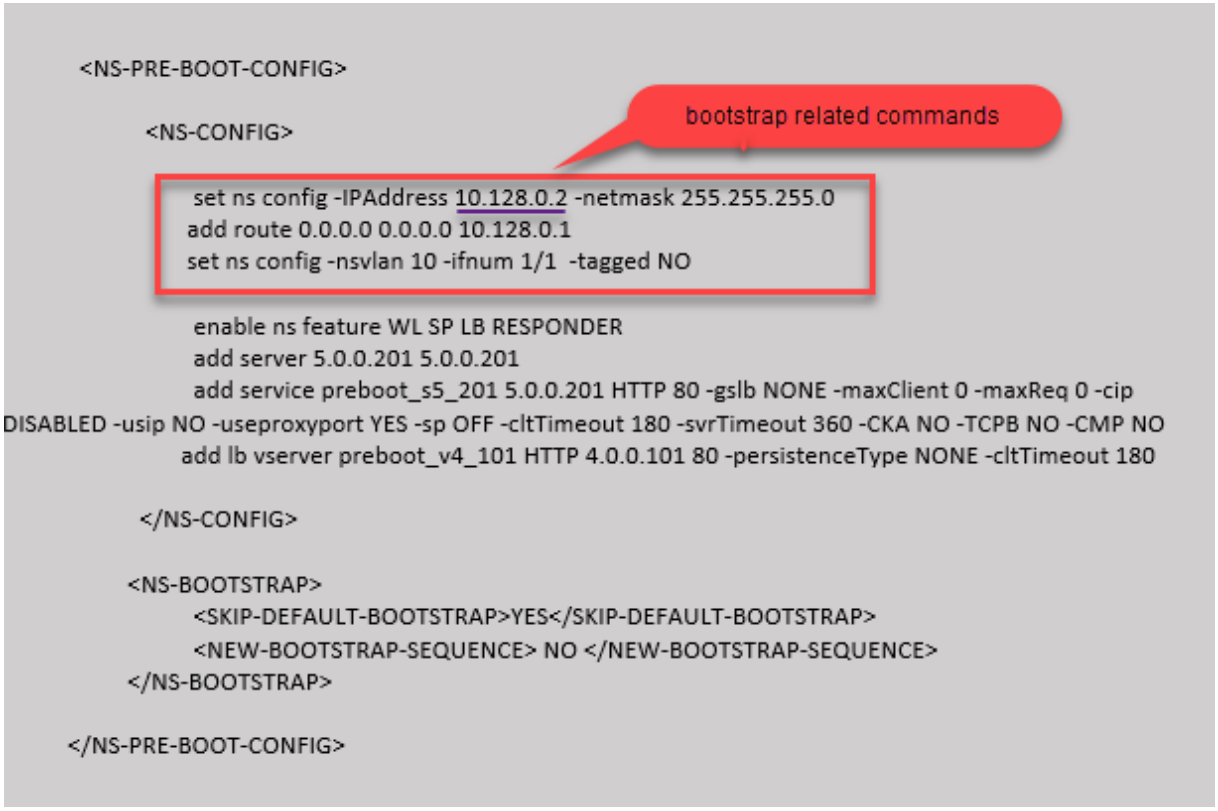
1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:9076/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 5      VLAN Alias Name:

3)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.61        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.61      0       UP      0               DIRECT
4)      172.27.2.0       255.255.255.0    172.27.2.61      0       UP      0               DIRECT
5)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
6)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
7)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
```

GCP のカスタムブートストラップの例

この例では、ブートストラップ関連のコマンドを<NS-CONFIG>セクションで提供しています。<NS-BOOTSTRAP>セクションは、デフォルトのブートストラップがスキップされ、<NS-CONFIG>セクションで提供されるカスタムブートストラップ情報が適用されることを示します。



前のスクリーンショットに示した設定をここからコピーできます。

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>

        set ns config -IPAddress 10.128.0.2 -netmask 255.255.255.0
        add route 0.0.0.0 0.0.0.0 10.128.0.1
        set ns config -nsvlan 10 -ifnum 1/1  -tagged NO

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

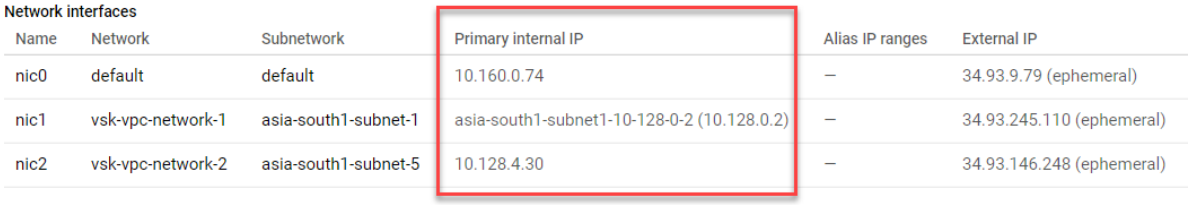
    <NS-BOOTSTRAP>
```



```
17          <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
18          <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
19      </NS-BOOTSTRAP>
20
21  </NS-PRE-BOOT-CONFIG>
22  <!--NeedCopy-->
```

カスタムブートストラップを使用して GCP ポータルで VM インスタンスを作成した後、ネットワークインターフェイスのプロパティを次のように確認できます。

1. カスタムブートストラップ情報を指定して、作成したインスタンスを選択します。
2. [Network Interface] プロパティに移動し、図に示すように NIC の詳細を確認します。



Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP
nic0	default	default	10.160.0.74	–	34.93.9.79 (ephemeral)
nic1	vsk-vpc-network-1	asia-south1-subnet-1	asia-south1-subnet1-10-128-0-2 (10.128.0.2)	–	34.93.245.110 (ephemeral)
nic2	vsk-vpc-network-2	asia-south1-subnet-5	10.128.4.30	–	34.93.146.248 (ephemeral)

ADC CLI で**`show nsip`**コマンドを実行し、**ADC** アプライアンスの最初の起動時に前の<NS-CONFIG>セクションで説明した設定が適用されていることを確認できます。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.0.2       0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:4a/64
        Interfaces : 0/1 1/2 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/1
        IPs :
             10.128.0.2         Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.0.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.2       0       UP      0               DIRECT
 Done
```

AWS および Azure での NIC のアタッチとデタッチによる影響

AWS と Azure には、ネットワークインターフェイスをインスタンスにアタッチし、インスタンスからネットワークインターフェイスをデタッチするオプションがあります。インターフェイスをアタッチまたはデタッチすると、インターフェイスの位置が変わることがあります。したがって、Citrix C VPX インスタンスからのインターフェイスのデタッチは控えることをお勧めします。カスタムブートストラップが構成されているときにインターフェイスをデタッチまたは接続すると、Citrix ADC VPX インスタンスは、管理インターフェイスの位置で新しく使用可能なインターフェイスのプライマリ IP を NSIP として再割り当てします。デタッチしたインターフェイスがこれ以上使用できな

い場合は、最初のインターフェイスが ADC VPX インスタンスの管理インターフェイスになります。
たとえば、Citrix ADC VPX インスタンスは、Eth0（SNIP）、Eth1（NSIP）、および Eth2（VIP）の 3 つのインターフェイスで起動されます。管理インターフェイスであるインスタンスから Eth1 インターフェイスをデタッチすると、ADC は次の使用可能なインターフェイス（Eth2）を管理インターフェイスとして設定します。これにより、ADC VPX インスタンスは、Eth2 インターフェイスのプライマリ IP を介してアクセスされます。Eth2 も使用できない場合は、残りのインターフェイス（Eth0）が管理インターフェイスになります。したがって、ADC VPX インスタンスへのアクセスは引き続き存在します。
Eth0（SNIP）、Eth1（VIP）、および Eth2（NSIP）のインターフェイスの別の割り当てについて考えてみましょう。Eth2（NSIP）をデタッチすると、Eth2 の後に新しいインターフェイスが使用できないため、最初のインターフェイス（Eth0）が管理インターフェイスになります。

ベアメタルサーバーに **Citrix ADC VPX** インスタンスをインストールします

October 7, 2021

ベアメタルは、クラウド環境に完全に統合された、物理的な分離を実現する完全に専用の物理サーバーです。シングルテナントサーバーとも呼ばれます。シングルテナンシーにより、ノイズの多いネイバー効果を回避できます。ベアメタルを使用すると、あなたが唯一のユーザーであるため、騒々しい隣人の影響を目撃することはありません。

ハイパーバイザーがインストールされたベアメタルサーバーは、サーバー上に仮想マシンを作成するための管理スイートを提供します。ハイパーバイザーはアプリケーションをネイティブに実行しません。その目的は、ワークロードを個別の仮想マシンに仮想化して、仮想化の柔軟性と信頼性を獲得することです。

ベアメタルサーバーに **Citrix ADC VPX** インスタンスをインストールするための前提条件

ベアメタルサーバーは、それぞれのハイパーバイザーのすべてのシステム要件を満たすクラウドベンダーから入手する必要があります。

ベアメタルサーバーに **Citrix ADC VPX** インスタンスをインストールします

ベアメタルサーバーに Citrix ADC VPX インスタンスをインストールするには、最初にクラウドベンダーから適切なシステムリソースを備えたベアメタルサーバーを入手する必要があります。そのベアメタルサーバーでは、ADC VPX インスタンスを展開する前に、Linux KVM、VMware ESX、Citrix Hypervisor、MicrosoftHyper-V などのサポートされているハイパーバイザーをインストールして構成する必要があります。

Citrix ADC VPX インスタンスでサポートされているさまざまなハイパーバイザーと機能のリストの詳細については、「サポートマトリックスと使用ガイドライン」を参照してください。

さまざまなハイパーバイザーに Citrix ADC VPX インスタンスをインストールする方法の詳細については、それぞれのドキュメントを参照してください。

- **Citrix Hypervisor**: Citrix Hypervisor への Citrix ADC VPX インスタンスのインストールを参照してください。
- **VMware ESX**: VMware ESX への Citrix ADC VPX インスタンスのインストールを参照してください。
- **Microsoft Hyper-V**: Microsoft Hyper-V サーバーへの Citrix ADC VPX インスタンスのインストールを参照してください。
- **Linux KVM** プラットフォーム: Linux-KVM プラットフォームへの Citrix ADC VPX インスタンスのインストールを参照してください。

Citrix Hypervisor への Citrix ADC VPX インスタンスのインストール

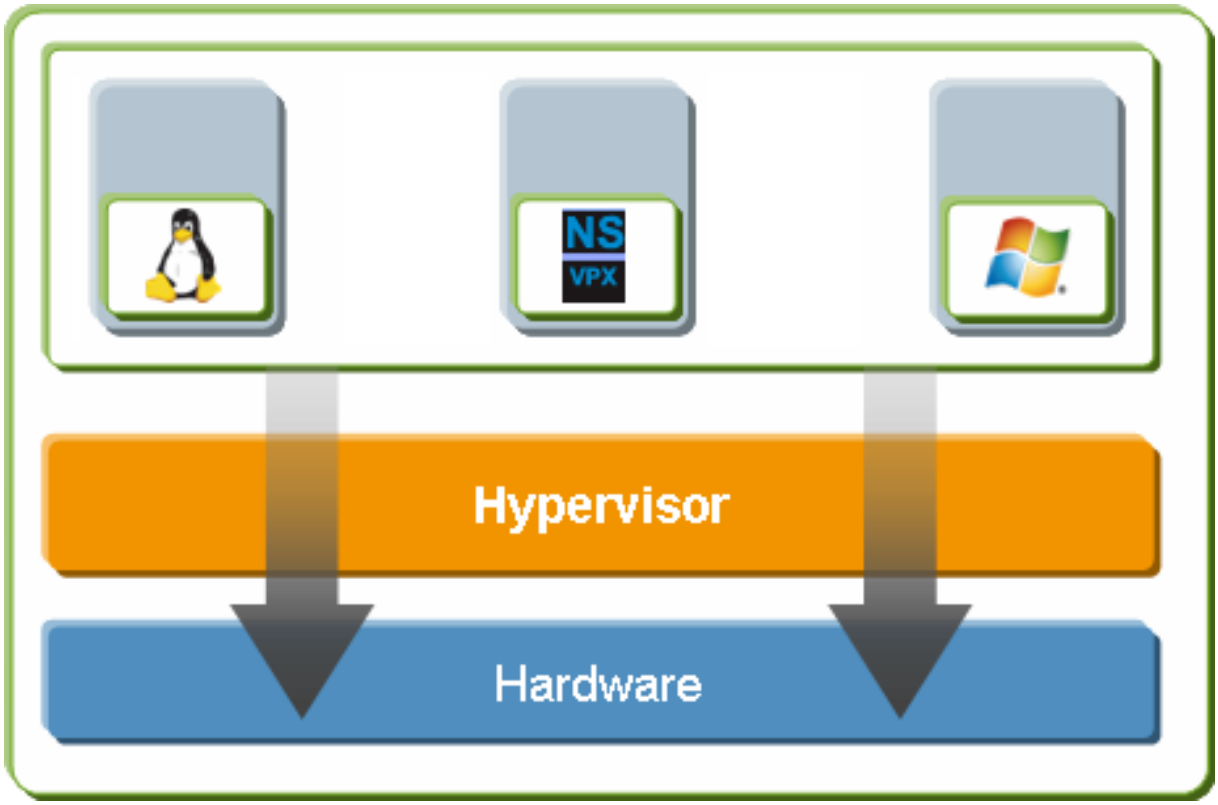
October 7, 2021

Citrix Hypervisor に VPX インスタンスをインストールするには、まず適切なシステムリソースを持つマシンにハイパーバイザーをインストールする必要があります。Citrix ADC VPX インスタンスのインストールを実行するには、Citrix XenCenter を使用します。Citrix XenCenter は、ネットワーク経由で Hypervisor ホストに接続できるリモートマシンにインストールする必要があります。

ハイパーバイザーの詳細については、Citrix Hypervisor のドキュメントを参照してください。

次の図は、Hypervisor 上の Citrix ADC VPX インスタンスのベアメタルソリューションアーキテクチャを示しています。

図。Citrix Hypervisor 上の Citrix ADC VPX インスタンス



Hypervisor に **Citrix ADC VPX** インスタンスをインストールするための前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- 最小要件を満たすハードウェアに Hypervisor バージョン 6.0 以降をインストールします。
- 最小システム要件を満たす管理ワークステーションに XenCenter をインストールします。
- 仮想アプライアンスのライセンスファイルを取得します。仮想アプライアンスのライセンスの詳細については、http://support.citrix.com/article/ctx122426で「*Citrix ADC VPX* ライセンスガイド」を参照してください。

Hypervisor のハードウェア要件

次の表は、Citrix ADC VPX インスタンスを実行する Hypervisor プラットフォームの最小ハードウェア要件を示しています。

表 **1**. nCore VPX インスタンスを実行する Hypervisor の最小システム要件

コンポーネント	条件
CPU	2 台以上の 64 ビット x86 CPU で、仮想化アシスト (インテル VT) が有効になっています。AMD プロセッサーはサポートされていません。Citrix ADC VPX インスタンスを実行するには、ハイパーバイザーホストで仮想化のハードウェアサポートを有効にする必要があります。仮想化サポートの BIOS オプションが無効になっていないことを確認してください。詳細については、BIOS のマニュアルを参照してください。
RAM	3GB
ディスク領域	40 GB のディスク容量を持つローカル接続ストレージ (PATA、SATA、SCSI)。注: ハイパーバイザーのインストールでは、Hypervisor ホストコントールドメインに 4 GB のパーティションが作成されます。残りの領域は、Citrix ADC VPX インスタンスおよびその他の仮想マシンで使用できます。
NIC	1 Gbps NIC x 1、推奨:2 つの 1 Gbps NIC

Hypervisor のインストールについては、http://support.citrix.com/product/xens/で Hypervisor のドキュメントを参照してください。

次の表は、Hypervisor が各 nCore VPX 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースの一覧です。

表 **2**. nCore VPX インスタンスの実行に必要な最小仮想コンピューティングリソース

コンポーネント	条件
メモリ	2GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	2

> 注:
> Citrix ADC VPX インスタンスを実稼働環境で使用する場合は、スケジューリング動作とネットワーク遅延を改善するために、仮想マシンのプロパティで CPU 優先度を最高レベルに設定することをお勧めします。

XenCenter システム要件

XenCenter は、Windows のクライアントアプリケーションです。Hypervisor ホストと同じマシンでは実行できません。最小システム要件と XenCenter のインストールの詳細については、次の Hypervisor ドキュメントを参照してください。

- システム要件
- インストール

XenCenter を使用して **Hypervisor** に **Citrix ADC VPX** インスタンスをインストールする

Hypervisor および XenCenter をインストールして構成したら、XenCenter を使用して Hypervisor に仮想アプライアンスをインストールできます。インストールできる仮想アプライアンスの数は、Hypervisor を実行しているハードウェアで使用可能なメモリ量によって異なります。

XenCenter を使用して初期 Citrix ADC VPX インスタンス（.xva イメージ）をハイパーバイザーにインストールした後、Command Center を使用して Citrix ADC VPX インスタンスをプロビジョニングできます。詳細については、Command Center ドキュメントを参照してください。

XenCenter を使用して Hypervisor に Citrix ADC VPX インスタンスをインストールするには、次の手順に従います。

1. ワークステーションで XenCenter を起動します。
2. ［サーバー］メニューの［追加］を選択します。
3. [新しいサーバーの追加] ダイアログボックスの [ホスト名] テキストボックスに、接続先のハイパーバイザーの IP アドレスまたは DNS 名を入力します。
4. ［User Name］ボックスと［Password］ボックスに管理者資格情報を入力して、［Connect］をクリックします。Hypervisor 名がナビゲーションペインに表示され、Hypervisor が接続されていることを示します。
5. ナビゲーションペインで、Citrix ADC VPX インスタンスをインストールするハイパーバイザーの名前をクリックします。
6. ［VM］メニューの［Import］を選択します。
7. [インポート] ダイアログボックスの [インポートファイル名] で、Citrix ADC VPX インスタンスの.xva イメージファイルを保存した場所を参照します。［Exported VM］オプションを選択したことを確認して、［Next］をクリックします。
8. 仮想アプライアンスをインストールする Hypervisor を選択し、次へをクリックします。
9. 仮想アプライアンスを保存するローカルストレージリポジトリを選択して［Import］をクリックし、インポート処理を開始します。
10. 必要に応じて、仮想ネットワークインターフェイスを追加、変更、または削除できます。完了したら［Next］をクリックします。

11. [完了] をクリックして、インポート処理を完了します。

    注記: インポートプロセスのステータスを表示するには、「ログ」(**Log**) タブをクリックします。

12. 別の仮想アプライアンスをインストールする場合は、手順 5 ～11 を繰り返します。

> 注
>
> VPX インスタンスの初期構成後、アプライアンスを最新のソフトウェアリリースにアップグレードする場合は、「システムソフトウェアのアップグレードまたはダウングレード」を参照してください。

シングルルート **I/O** 仮想化（**SR-IOV**）ネットワークインターフェイスを使用するように **VPX** インスタンスを構成する

October 7, 2021

XenServer に Citrix ADC VPX インスタンスをインストールして構成した後、SR-IOV ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

制限事項

XenServer は、SR-IOV インターフェイスで以下の機能をサポートしません。

- L2 モード切り替え
- クラスタリング
- 管理パーティション化［共有 VLAN モード］
- 高可用性［アクティブ/アクティブモード］
- ジャンボフレーム
- クラスター環境の IPv6 プロトコル

前提条件

XenServer ホストで、次のことを確認します。

- インテル 82599 NIC (NIC) をホストに追加します。
- **/etc/modprobe.d/blacklist.conf** ファイルに次のエントリを追加して、`ixgbevf`ドライバを一覧表示することを禁止します。

  blacklist ixgbevf

- **/etc/modprobe.d/blacklist.conf** ファイルで、以下のエントリを追加して、SR-IOV Virtual Functions（VF）を有効にします。

  options ixgbe max_vfs=*<number_of_VFs>*

ここで、*<number_VFs>* は、作成する SR-IOV VF の数です。

- SR-IOV サーバーが BIOS で有効になっていることを確認します。

IXGBE ドライバーのバージョン 3.22.3 をお勧めします。

XenServer ホストを使用して、**SR-IOV VF** を **VPX** インスタンスに割り当てます

SR-IOV ネットワークインターフェイスを Citrix ADC VPX インスタンスに割り当てるには、次の手順に従います。

1. XenServer ホストで、次のコマンドを使用して、SR-IOV VF を Citrix ADC VPX インスタンスに割り当てます。

   xe host-call-plugin plugin=iovirt host-uuid=*<Xen host UUID>* **fn=assign_free_vf args:uuid**=*<Netscalar VM UUID>* **args:ethdev**=*<interface name>* **args:mac=***<Mac addr>*

   各項目の意味は次のとおりです:

   - *<Xen host UUID>* は、XenServer ホストの UUID です。
   - ** <NetScaler VM UUID> は、Citrix ADC VPX インスタンスの UUID です。
   - *<interface name>* は、SR-IOV VF のインターフェイスです。
   - ** <MAC address > は SR-IOV VF の MAC アドレスです。

   > 注
   >
   > args: mac= パラメータで使用する MAC アドレスを指定します。指定しない場合、`iovirt`スクリプトはランダムに MAC アドレスを生成して割り当てます。また、リンクアグリゲーションモードで SR-IOV VF を使用する場合は、必ず MAC アドレスを 00:00:00:00:00 と指定します。

2. Citrix ADC VPX インスタンスを起動します。

XenServer ホストを使用して、**SR-IOV VF** を **VPX** インスタンスに割り当て解除します

正しくない SR-IOV VF を割り当てた場合、または割り当てられた SR-IOV VF を変更する場合は、SR-IOV VF を Citrix ADC VPX インスタンスに割り当て解除して再割り当てする必要があります。

Citrix ADC VPX インスタンスに割り当てられた SR-IOV ネットワークインターフェイスの割り当てを解除するには、次の手順に従います。

1. XenServer ホストで次のコマンドを使用して、SR-IOV VF を Citrix ADC VPX インスタンスに割り当てて、Citrix ADC VPX インスタンスを再起動します。

   xe host-call-plugin plugin=iovirt **host-uuid**=*<Xen_host_UUID>* **fn**=unassign_all **args:uuid**=*<Netscalar_VM_U*

   各項目の意味は次のとおりです:

   - **<Xen_host_UUID>-XenServer ホストの UUID です。

- *<Netscalar_VM_UUID>*-Citrix ADC VPX インスタンスの UUID

2. Citrix ADC VPX インスタンスを起動します。

SR-IOV インターフェイスで VLAN を構成する

> 重要
>
> SR-IOV VF を Citrix ADC VPX インスタンスに割り当てる際は、VF に MAC アドレス 00:00:00:00:00:00 を指定していることを確認してください。

リンクアグリゲーションモードで、SR-IOV 仮想機能を使用するには、作成した仮想機能のなりすましチェックを無効にする必要があります。XenServer ホストでなりすましチェックを無効にするには、以下のコマンドを使用します。

ip link set *<interface_name>* **vf** *<VF_id>* **spoofchk** off

各項目の意味は次のとおりです:

- <interface_name> は、インターフェイス名です。
- <VF_id> は、仮想機能 ID です。

作成したすべての仮想機能のスプーフィングチェックを無効にした後、Citrix ADC VPX インスタンスを再起動し、リンクアグリゲーションを構成します。手順については、「リンク集約の設定」を参照してください。

SR-IOV インターフェイスで VLAN を設定します

SR-IOV 仮想機能で VLAN を構成できます。手順については、「Configuring a VLAN」を参照してください。

> 重要
>
> XenServer ホストに VF インターフェイスの VLAN 設定を含めないようにしてください。

VMware ESX への Citrix ADC VPX インスタンスのインストール

April 7, 2022

VMware ESX に Citrix ADC VPX インスタンスをインストールする前に、VMware ESX Server が適切なシステムリソースを持つマシンにインストールされていることを確認してください。VMware ESXi に Citrix ADC VPX インスタンスをインストールするには、VMware vSphere クライアントを使用します。これらのクライアントソフトウェアは、ネットワーク経由で VMware ESX に接続できるリモートマシンにインストールする必要があります。

このセクションでは、以下のトピックについて説明します。

- 前提条件
- VMware ESX に Citrix ADC VPX インスタンスをインストールする

> 重要
>
> 標準の VMware Tools をインストールしたり、Citrix ADC VPX インスタンスで使用可能な VMware Tools バージョンをアップグレードしたりすることはできません。Citrix ADC VPX インスタンス用の VMware ツールは、Citrix ADC ソフトウェアリリースの一部として提供されます。

前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- 最小要件を満たすハードウェアに VMware ESX をインストールします。
- 最小システム要件を満たす管理用のワークステーションに VMware Client をインストールします。
- Citrix ADC VPX アプライアンスのセットアップファイルをダウンロードします。
- VMware ESX の物理ネットワークポートにラベルを付けます。
- VPX ライセンスファイルを入手します。Citrix ADC VPX インスタンスライセンスの詳細については、「ライセンスの概要」を参照してください。

VMware ESX のハードウェア要件

次の表では、Citrix ADC VPX nCore 仮想アプライアンスを実行している VMware ESX サーバーの最小システム要件について説明します。

表 1. Citrix ADC VPX インスタンスを実行している VMware ESX サーバーの最小システム要件

コンポーネント	条件
-	仮想化アシスト (Intel-VT) が有効になっている 64 ビット x86 CPU が 2 つ以上あります。Citrix ADC VPX インスタンスを実行するには、VMware ESX ホストで仮想化のハードウェアサポートを有効にする必要があります。仮想化サポートの BIOS オプションが無効になっていないことを確認してください。詳しくは、BIOS のドキュメントを参照してください。
RAM	3GB
ディスク領域	40GB 以上の空きディスク領域
ネットワーク	1 Gbps NIC (NIC) 1 つ、1 Gbps NIC を 2 つ推奨

VMware ESX のインストールについては、http://www.vmware.com/を参照してください。

次の表に、VMware ESX サーバが各 VPX nCore 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースを示します。

表 2. Citrix ADC VPX インスタンスの実行に必要な最小仮想コンピューティングリソース

コンポーネント	条件
メモリ	4GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	1. ESX では、VPX ハードウェアをバージョン 7 以上にアップグレードすると、最大 10 個の仮想ネットワークインターフェイスをインストールできます。
ディスク領域	20GB

> 注
>
> ハイパーバイザーに必要なディスク領域は含まれません。

VPX 仮想アプライアンスを本番環境で使用するには、完全なメモリ割り当てを予約する必要があります。少なくとも ESX の 1 つの CPU コアの速度に等しい CPU サイクル (MHz) を予約する必要があります。

VMware vSphere クライアントのシステム要件

VMware vSphere Client は、Windows および Linux の各オペレーティングシステムで実行できるクライアントアプリケーションです。VMware ESX サーバーと同じマシンでは実行できません。次の表は、最小システム要件を示しています。

表 3. VMware vSphere クライアントインストールの最小システム要件

コンポーネント	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で「vSphere 互換性マトリックス」PDF ファイルを検索してください。
CPU	750 MHz。1 ギガヘルツ (GHz) 以上推奨
RAM	1GB。2GB をお勧めします。
NIC (NIC)	100Mbps 以上の NIC。

OVF ツール **1.0** のシステム要件

Open Virtualization Format Tool（OVF Tool）は、Windows および Linux システムで実行できるクライアントアプリケーションです。VMware ESX サーバーと同じマシンでは実行できません。次の表は、最小システム要件を示

しています。

表 4. OVF ツールのインストールに必要な最小システム要件

コンポーネント	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。
CPU	最低 750MHz、1GHz 以上推奨
RAM	最小 1 GB、推奨 2 GB
NIC (NIC)	100Mbps 以上の NIC。

OVF のインストールについては、http://kb.vmware.com/で『OVF ツールユーザーガイド』の PDF ファイルを検索してください。

Citrix ADC VPX セットアップファイルのダウンロード

VMware ESX 用の Citrix ADC VPX インスタンスセットアップパッケージは、オープン仮想マシン（OVF）形式標準に準拠しています。これらのファイルは、Citrix Web サイトからダウンロードできます。ログオンするには、Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスし、[新規ユーザー] リンクをクリックし、指示に従って新しい Citrix アカウントを作成します。

ログオンしたら、Citrix のホームページから次のパスをナビゲートします。

Citrix.com > ダウンロード **> Citrix ADC >** 仮想アプライアンス。

次のファイルを、ESX サーバーと同じネットワーク上のワークステーションにコピーします。3 つのファイルをすべて同じフォルダーにコピーします。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（たとえば、nsvpx-esx-13.0-71.44_nc_64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf（たとえば、nsvpx-ESX-13.0-71.44_nc_64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（たとえば、nsvpx-esx-13.0-71.44_nc_64.mf）

VMware ESX の物理ネットワーク・ポートにラベルを付ける

VPX 仮想アプライアンスをインストールする前に、仮想アプライアンスに割り当てる予定のすべてのインターフェイスに、NS_NIC_1_1、NS_NIC_1_2 などの固有の形式でラベルを付けます。大規模な展開では、一意の形式でラベルを付けると、Windows や Linux など、他の仮想マシンで使用されるインターフェイス間で、VPX 仮想アプライアンスに割り当てられたインターフェイスをすばやく識別できます。このようなラベル付けは、異なるタイプの仮想マシンがインターフェイスを共有する場合に特に重要です。

VMware ESX サーバの物理ネットワーク・ポートにラベルを付けるには、次の手順に従います。

1. vSphere Client を使用して、VMware ESX サーバーにログオンします。
2. vSphere Client で、[Configuration] タブを選択して [Networking] をクリックします。
3. 右上隅の [Add Networking] をクリックします。
4. ネットワークの追加ウィザードの [接続の種類] で、[仮想マシン] を選択し、[次へ] をクリックします。
5. vSwitch 物理アダプタのリストをスクロールし、仮想アプライアンスのインターフェイス 1/1 にマップする物理ポートを選択します。
6. インターフェイスのラベルを入力します。たとえば、仮想アプライアンスのインターフェイス **1/1** に関連付けられている **vSwitch** の名前として **NS_NIC_1_1** を入力します。
7. [Next] をクリックして、vSwitch の作成を終了します。手順 2. からこの手順を繰り返して、仮想アプライアンスで使用されるインターフェイスを追加します。インターフェイスには、正しい形式で連続したラベルを付けます（例：NS_NIC_1_2）。

VMware ESX への **Citrix ADC VPX** インスタンスのインストール

VMware ESX をインストールして構成したら、VMware vSphere Client を使用して VMware ESX サーバーに仮想アプライアンスをインストールします。インストールできる仮想アプライアンスの数は、VMware ESX を実行するハードウェアで使用可能なメモリの量によって決まります。

VMware vSphere クライアントを使用して VMware ESX に Citrix ADC VPX インスタンスをインストールするには、次の手順を実行します。

1. ワークステーション上で VMware vSphere Client を起動します。
2. [**IP address / Name**] テキストボックスに、接続する VMware ESX サーバーの IP アドレスを入力します。
3. [ユーザー名] および [パスワード] テキストボックスに管理者の資格情報を入力し、[ログイン] をクリックします。
4. [**File**] メニューの [**Deploy OVF Template**] を選択します。
5. [**OVF** テンプレートのデプロイ] ダイアログボックスの [ファイルからデプロイ] で、Citrix ADC VPX インスタンスセットアップファイルを保存した場所を参照し、.ovf ファイルを選択して、[次へ] をクリックします。
6. 仮想アプライアンス OVF テンプレートに示されるネットワークを、ESX ホストで構成したネットワークにマップします。[**Next**] をクリックして、VMware ESX への仮想アプライアンスのインストールを開始します。インストールが完了すると、ポップアップウィンドウによって正常にインストールされたことが通知されます。
7. これで、Citrix ADC VPX インスタンスを起動する準備ができました。ナビゲーションペインで、インストールした Citrix ADC VPX インスタンスを選択し、右クリックメニューから [パワーオン] を選択します。
8. 仮想マシンが起動したら、コンソールから Citrix ADC IP、ネットマスク、およびゲートウェイアドレスを構成します。設定が完了したら、コンソールで [**Save and Quit**] オプションを選択します。
9. 別の仮想アプライアンスをインストールする場合は、ステップ 6 から繰り返します。

> 注
>
> デフォルトでは、Citrix ADC VPX インスタンスは E1000 ネットワークインターフェイスを使用します。

> インストール後、vSphere Client または vSphere Web Client を使用して VMware ESX 上の仮想アプライアンスを管理できます。
>
> VLAN タギング機能を機能させるには、VMware ESX で、VMware ESX サーバの vSwitch でポートグループの VLAN ID を All（4095）に設定します。VMware ESX サーバの vSwitch での VLAN ID の設定の詳細については、http://www.vmware.com/pdf/esx3_vlan_wp.pdfを参照してください。

VMware vMotion を使用して Citrix ADC VPX インスタンスを移行する

VMware vSphere vMotion を使用して、Citrix ADC VPX インスタンスを移行できます。

使用上のガイドラインに従ってください。

- VMware は、PCI パススルーおよび SR-IOV インターフェイスで構成された仮想マシンでは vMotion 機能をサポートしていません。
- サポートされているインターフェイスは、E1000 と VMXNET3 です。VPX インスタンスで vMotion を使用するには、サポートされているインターフェイスでインスタンスが設定されていることを確認します。
- VMware vMotion を使用してインスタンスを移行する方法の詳細については、VMware のドキュメントを参照してください。

VMXNET3 ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する

October 7, 2021

VMware ESX に Citrix ADC VPX インスタンスをインストールして構成したら、VMware vSphere Web クライアントを使用して、VMXNET3 ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

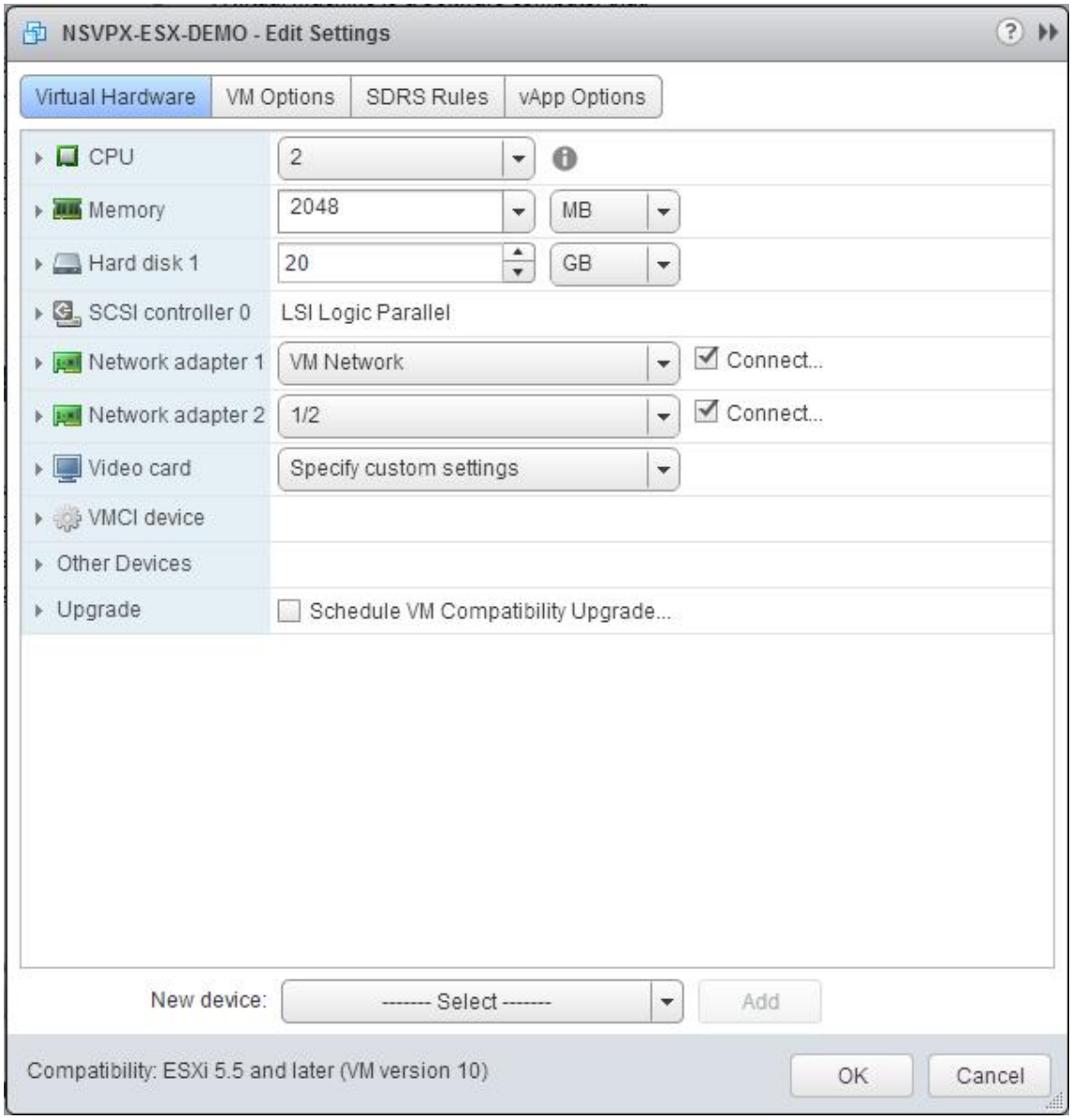
VMware vSphere Web クライアントを使用して、VMXNET3 ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成するには、次の手順に従います。

1. vSphere Web クライアントで［Hosts and Clusters］を選択します。
2. 以下のように、Citrix ADC VPX インスタンスの互換性設定を ESX にアップグレードします。

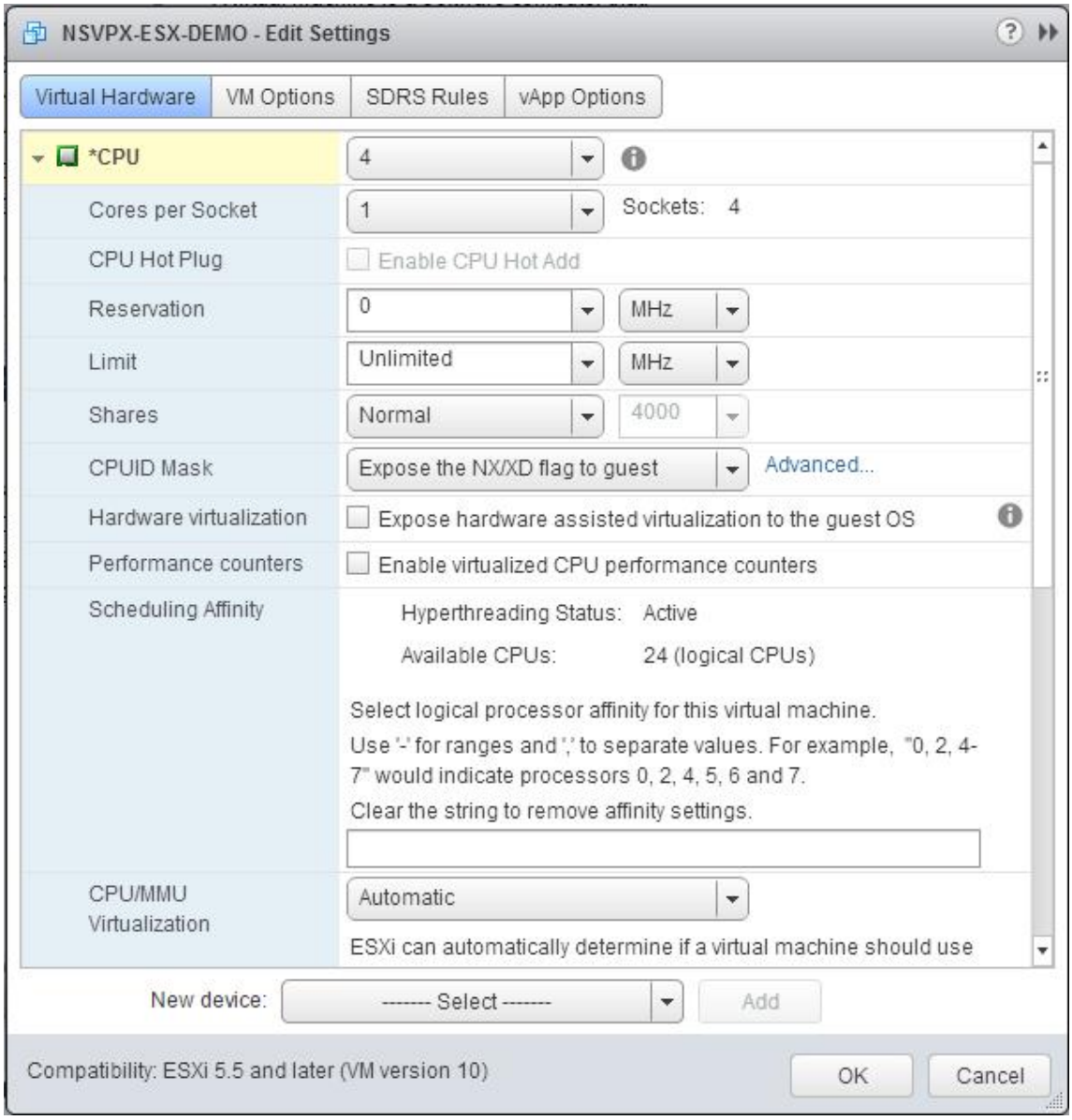
   a. Citrix ADC VPX インスタンスの電源を切ります。

   b. Citrix ADC VPX インスタンスを右クリックし、［互換性］>［仮想マシンの互換性のアップグレード］を選択します。

   c. 設定 VM 互換性] ダイアログボックスで、[OK] ドロップダウンリストに対応し、クリックからの ESXi 5.5 以降を選択します。
3. Citrix ADC VPX インスタンスを右クリックし、[設定の編集] をクリックします。



4. [<virtual_appliance> - Edit Settings] ダイアログボックスで [CPU] セクションをクリックします。



5. [CPU] セクションで、以下を更新します。

   - CPU の数
   - ソケット数
   - 予約
   - 上限
   - 共有

   値を次のように設定します。

   a. ［CPU］ボックスの一覧で、仮想アプライアンスに割り当てる CPU の数を選択します。

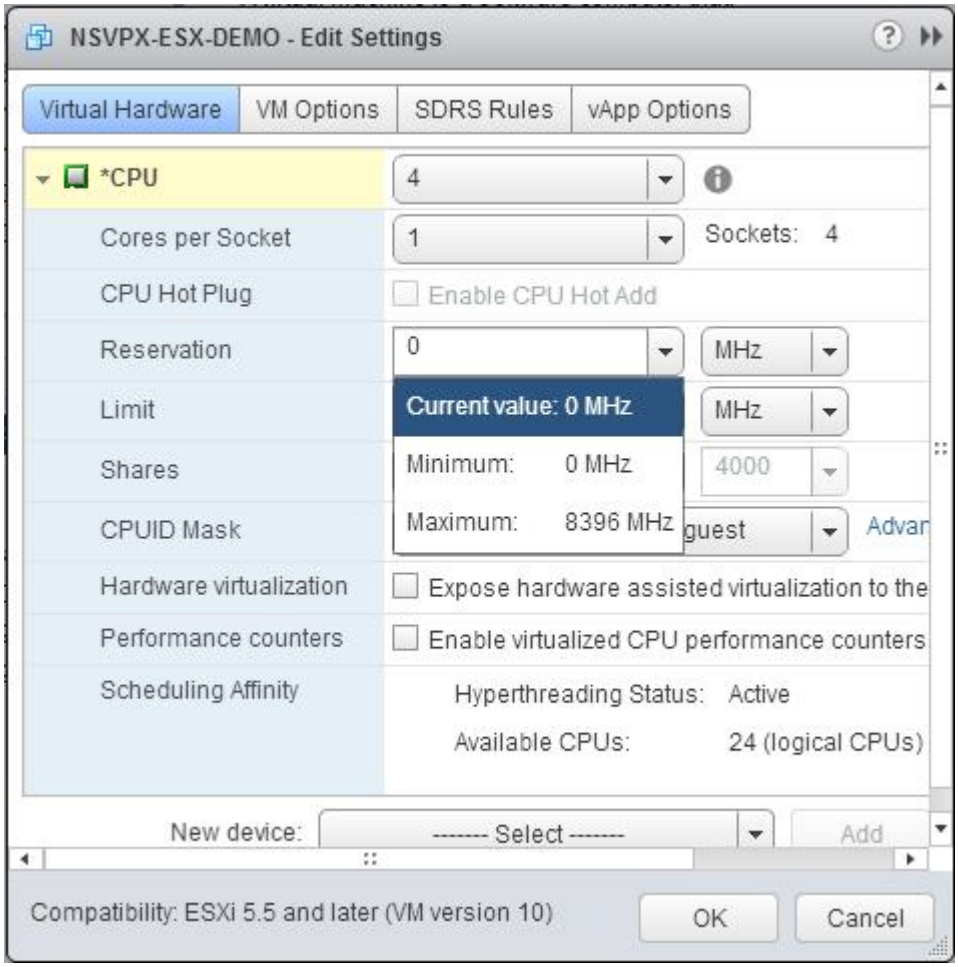
   b. [ソケットあたりのコア数] ドロップダウンリストで、ソケット数を選択します。

   c. （オプション）[CPU ホットプラグ] フィールドで、[CPU ホットアドを有効にする] チェックボックスをオ

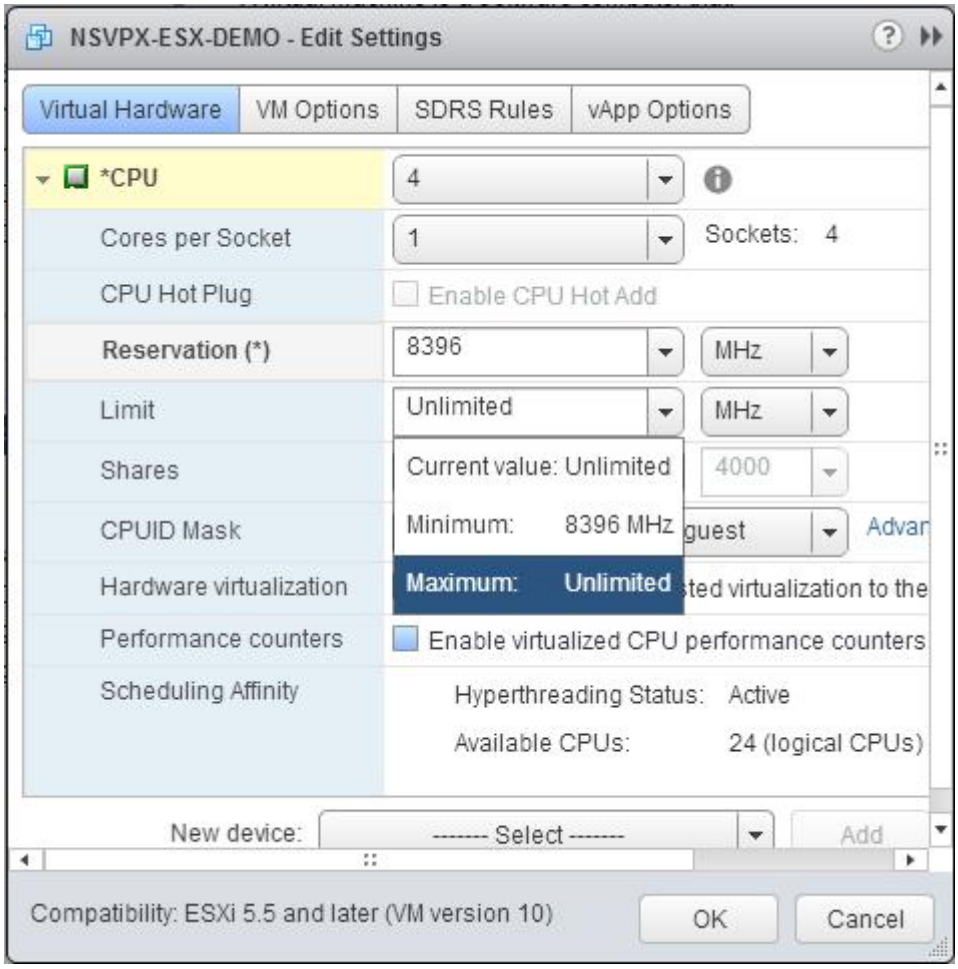
ンまたはオフにします。

注：デフォルト（無効）を受け入れることをお勧めします。

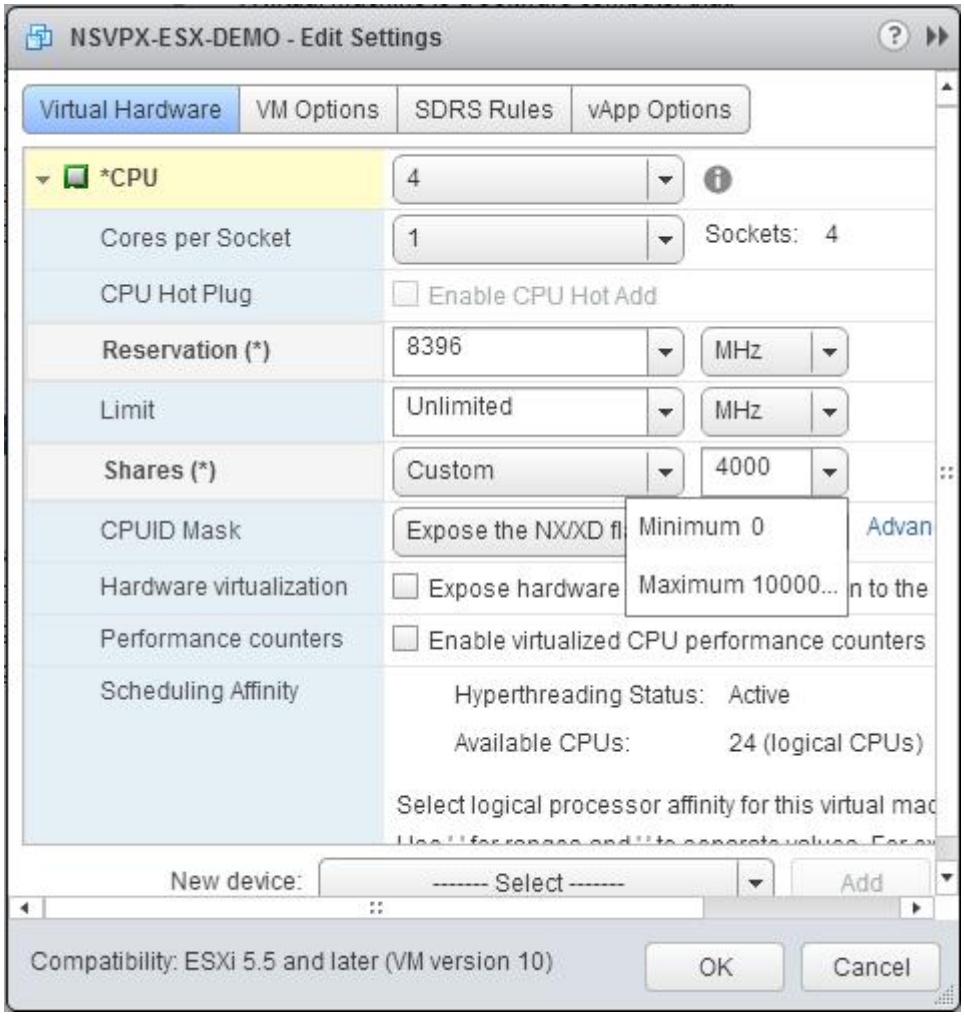
d. [予約] ドロップダウンリストで、最大値として表示される番号を選択します。



e. ［Limit］ボックスの一覧で、最大値として表示される数を選択します。



f. ［Shares］ボックスの一覧で、［Custom］と、最大値として表示される数を選択します。



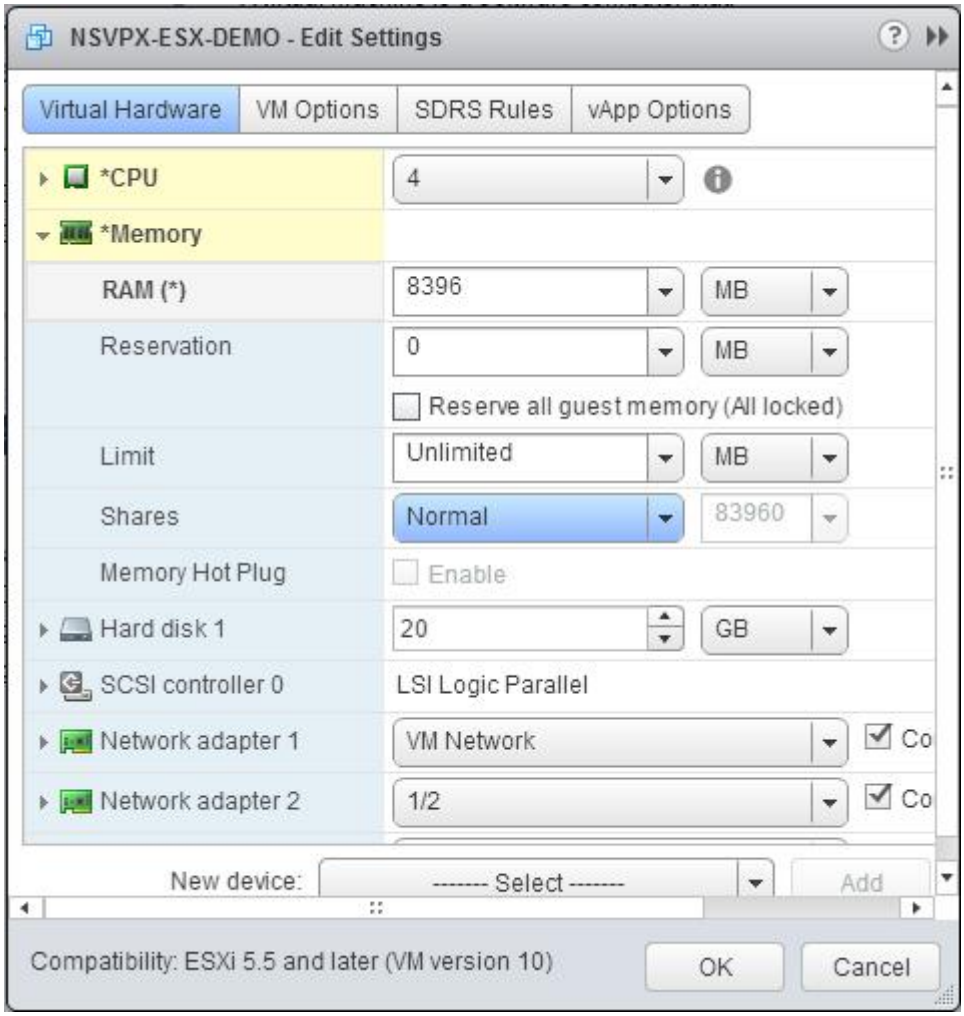
6. [メモリ] セクションで、次の項目を更新します。

   - RAM のサイズ
   - 予約
   - 上限
   - 共有

   値を次のように設定します。

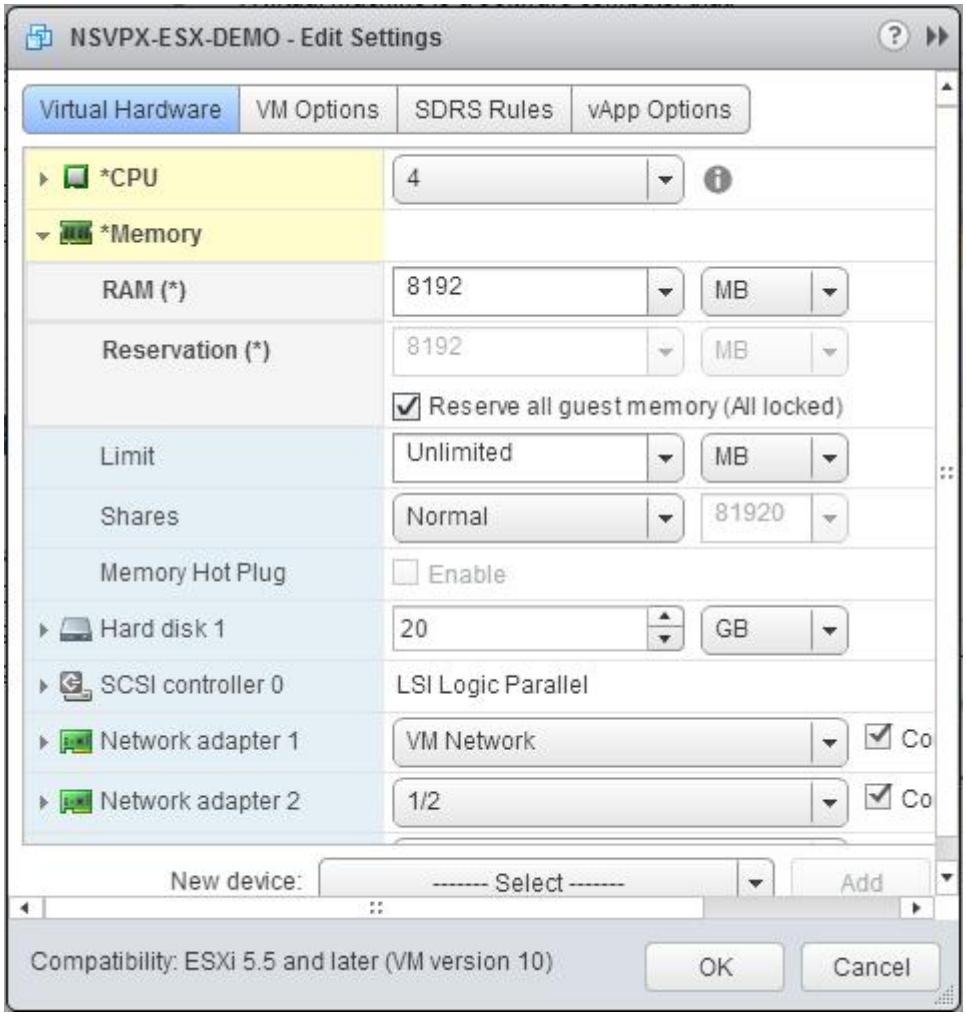
   a. [RAM] ドロップダウンリストで、RAM のサイズを選択します。これは vCPU の数 x 2 GB でなければなりません。たとえば、vCPU の数が 4 の場合、RAM は 4 x 2 GB = 8 GB でなければなりません。

   注：Citrix ADC VPX アプライアンスの Advanced エディションまたは Premium エディションでは、各 vCPU に 4GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。

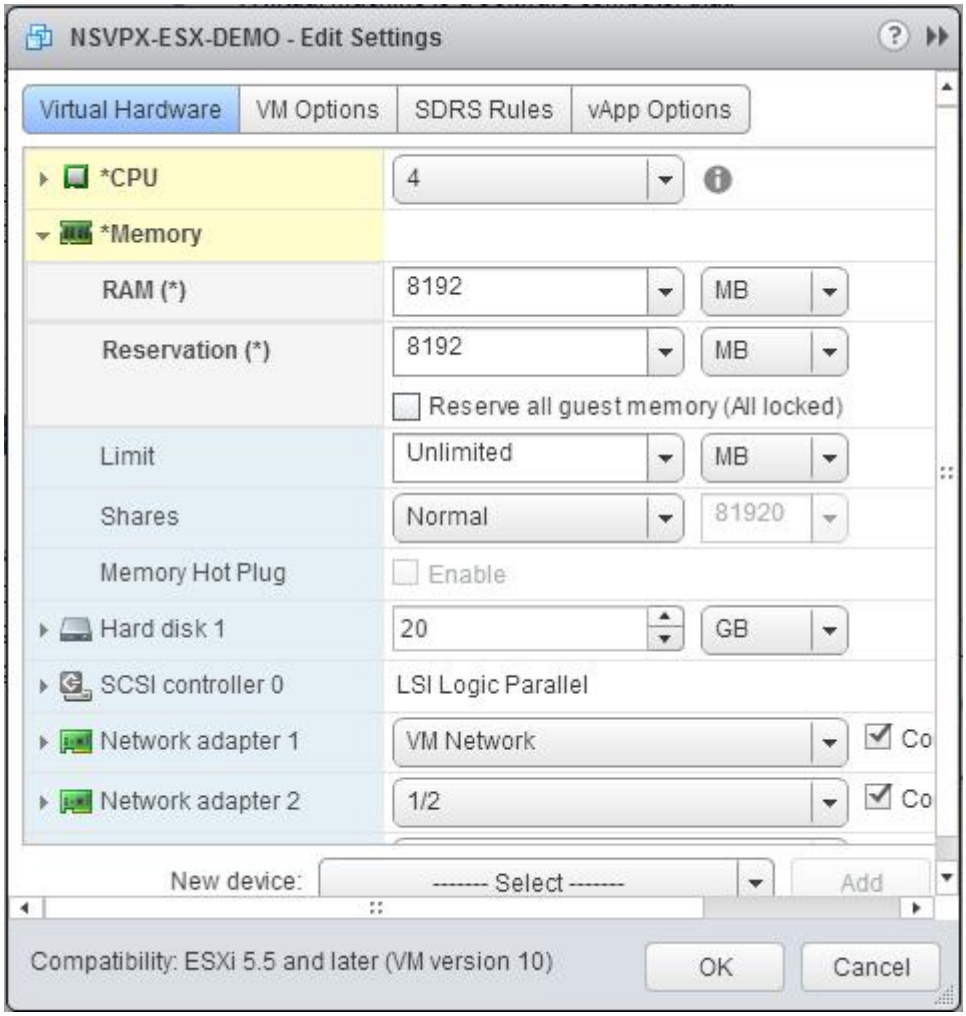


b. [予約] ドロップダウンリストで、メモリ予約の値を入力し、[すべてのゲストメモリを予約する (すべてのロック済み)] チェックボックスをオンにします。メモリ予約は vCPU の数 x 2 GB である必要があります。たとえば、vCPU の数が 4 の場合、メモリ予約は 4 x 2 GB = 8 GB である必要があります。

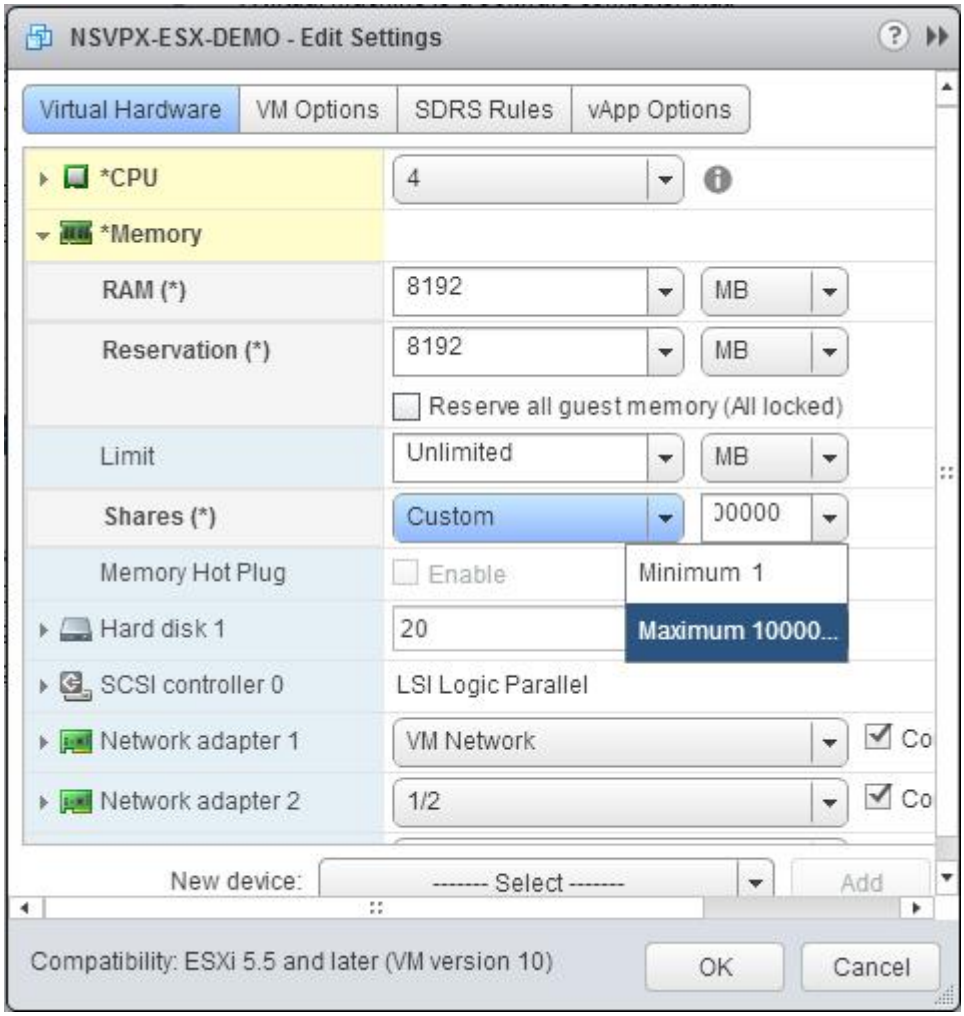
注: Citrix ADC VPX アプライアンスの Advanced エディションまたは Premium エディションでは、各 vCPU に 4GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。



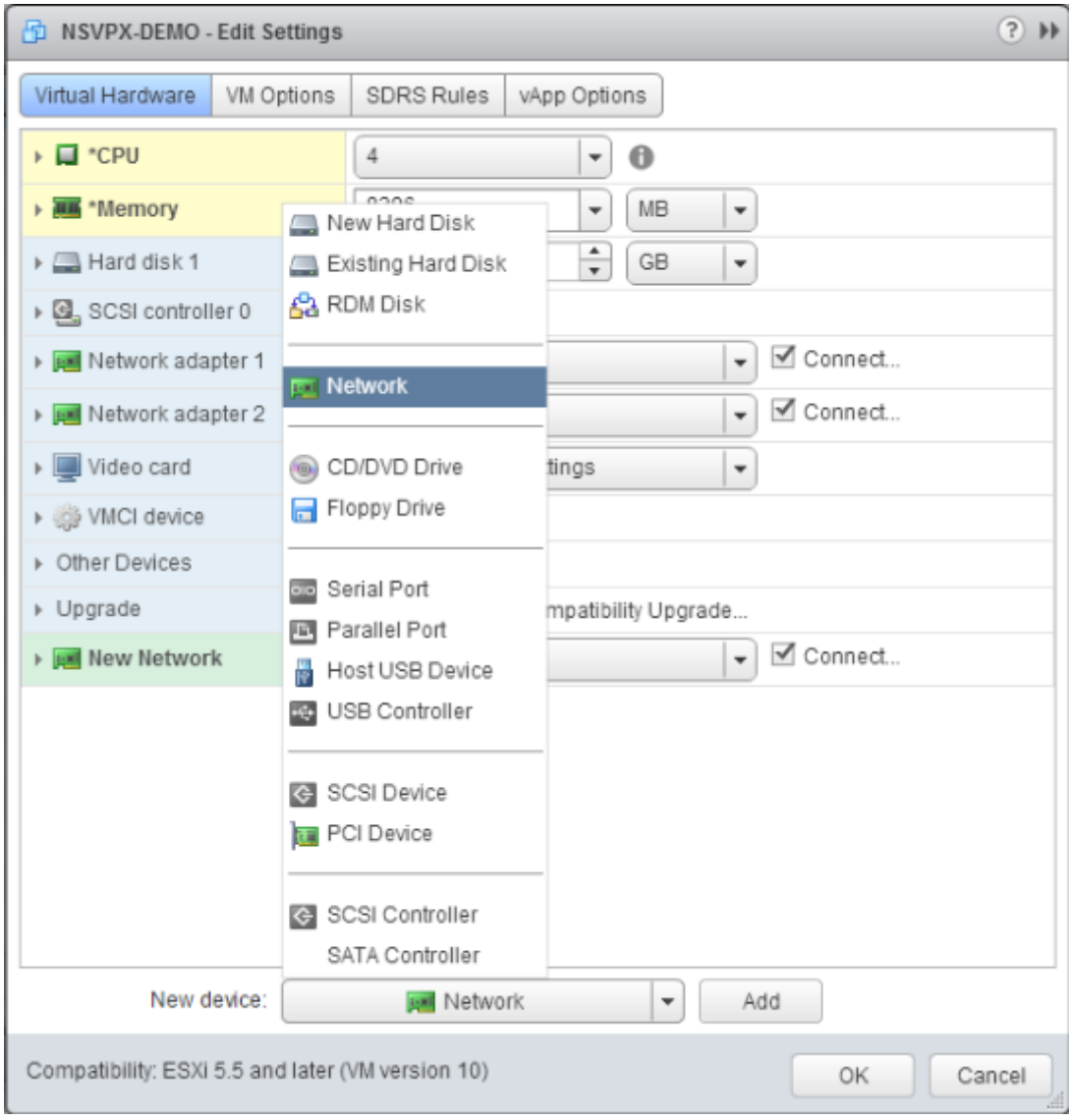
c. ［Limit］ボックスの一覧で、最大値として表示される数を選択します。



d.「共有」ドロップダウンリストで、「カスタム」を選択し、最大値として表示される数値を選択します。



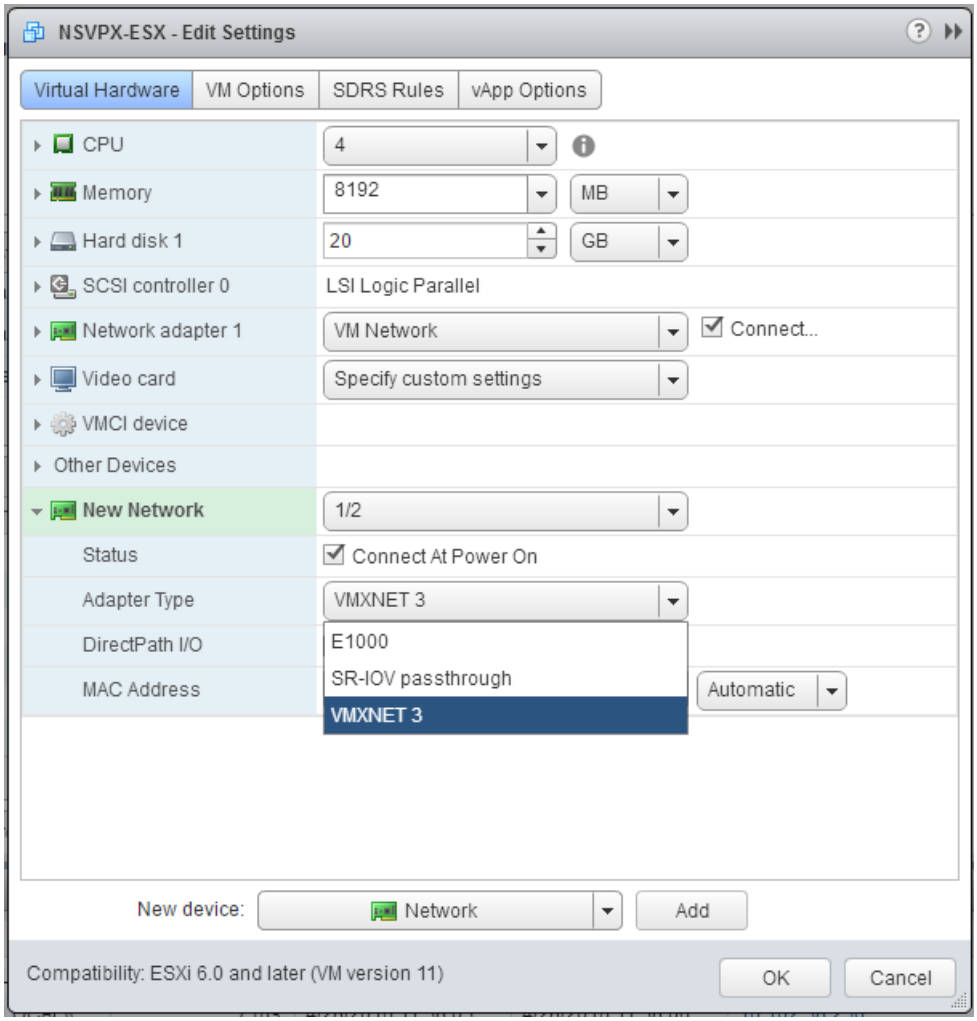
7. VMXNET3 ネットワークインターフェイスを追加します。[New device] ボックスの一覧で［Network］を選択し、[Add] をクリックします。



8. [New Network] セクションのドロップダウンリストからネットワークインターフェイスを選択し、次の操作を行います。

   a. ［Adapter Type］ボックスの一覧で、［VMXNET3］を選択します。

   > 重要
   >
   > デフォルトの E1000 ネットワークインターフェイスと VMXNET3 は共存できないため、E1000 ネットワークインターフェイスの削除を確認して、VMXNET3（0/1）を管理インターフェイスとして使用します。



9. [OK] をクリックします。
10. Citrix ADC VPX インスタンスの電源を入れます。
11. Citrix ADC VPX インスタンスの電源が入ったら、次のコマンドを使用して構成を確認できます。

    > インターフェイスの概要を表示

出力には、設定したすべてのインターフェイスが表示されている必要があります。

```
> show interface summary
-------------------------------------------------------------------------------

      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------------

1     0/1        1500       00:0c:29:89:1d:0e  NetScaler Vir...rface,
    VMXNET3
```



```
6  2     1/1        9000       00:0c:29:89:1d:18    NetScaler Vir...rface,
       VMXNET3
7  3     1/2        9000       00:0c:29:89:1d:22    NetScaler Vir...rface,
       VMXNET3
8  4     LO/1       9000       00:0c:29:89:1d:0e    Netscaler Loopback
      interface
```

> 注
>
> VMXNET3 インターフェイスを追加して Citrix ADC VPX アプライアンスを再起動すると、VMware ESX ハイパーバイザーによって NIC が VPX アプライアンスに提示される順序が変更されることがあります。そのため、ネットワークアダプター 1 が常に 0/1 のままであるとは限らず、その結果 VPX アプライアンスに対する管理接続が失われることがあります。この問題を回避するには、ネットワークアダプターの仮想ネットワークを変更します。
>
> これは VMware ESX ハイパーバイザの制限です。

SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する

October 7, 2021

VMware ESX に Citrix ADC VPX インスタンスをインストールして構成した後、VMware vSphere Web クライアントを使用して、シングルルート I/O 仮想化（SR-IOV）ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

制限事項

SR-IOV ネットワークインターフェイスで構成された Citrix ADC VPX には、以下の制限があります。

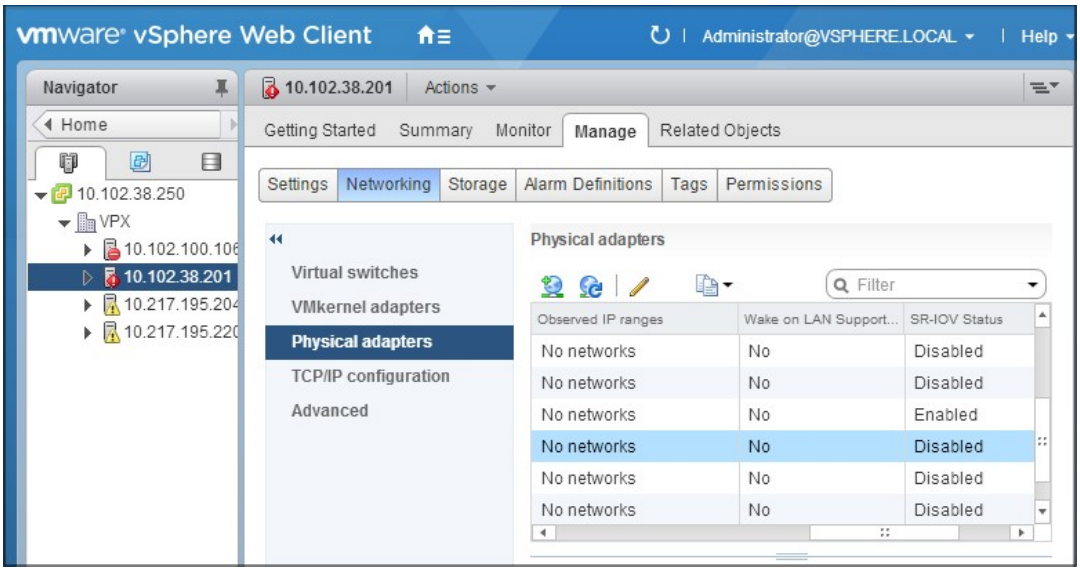
- 次の機能は、ESX VPX 上のインテル 82599 10G NIC を使用する SR-IOV インターフェイスではサポートされていません。
  - L2 モード切り替え
  - スタティックリンク集約および LACP
  - クラスタリング
  - 管理パーティション化［共有 VLAN モード］
  - 高可用性［アクティブ/アクティブモード］
  - ジャンボフレーム
  - IPv6
- KVM VPX 上のインテル 82599 10G NIC を搭載した SR-IOV インターフェイスでは、次の機能はサポートされていません。

- スタティックリンク集約および LACP
- L2 モード切り替え
- クラスタリング
- 管理パーティション化［共有 VLAN モード］
- 高可用性 [アクティブ-アクティブモード]
- ジャンボフレーム
- IPv6
- `ip link`コマンドによる SR-IOV VF インターフェイスのハイパーバイザでの VLAN 設定はサポートされていません

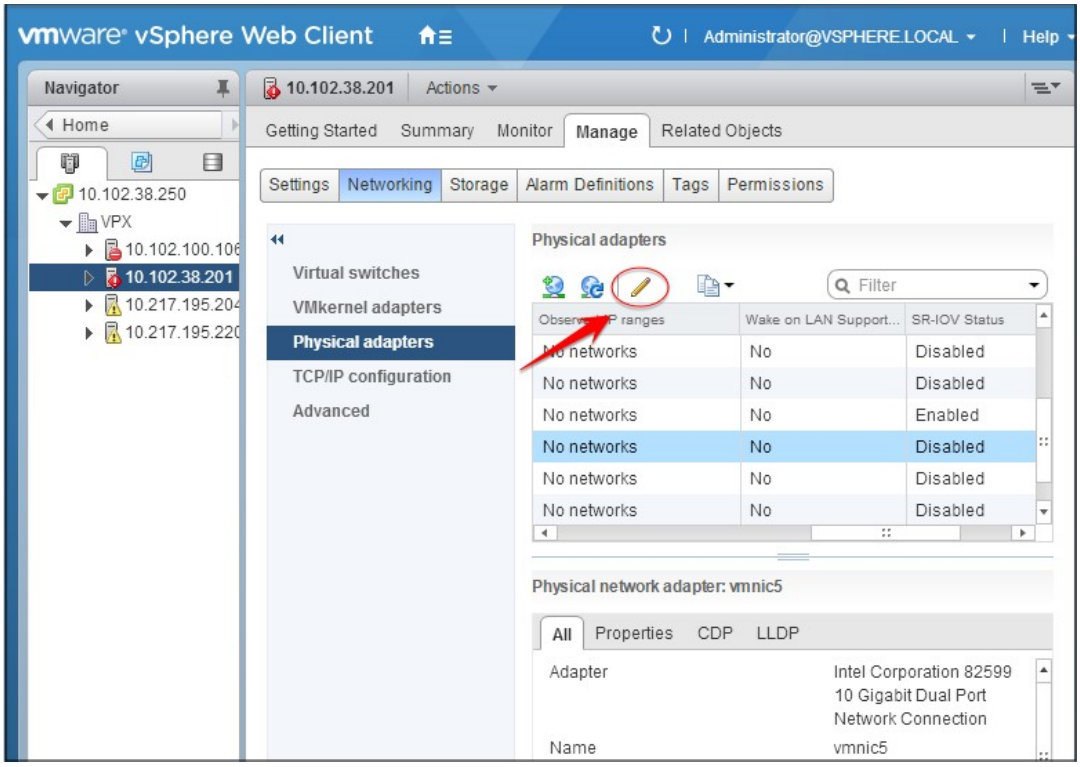
前提要件

以下の点について確認してください。

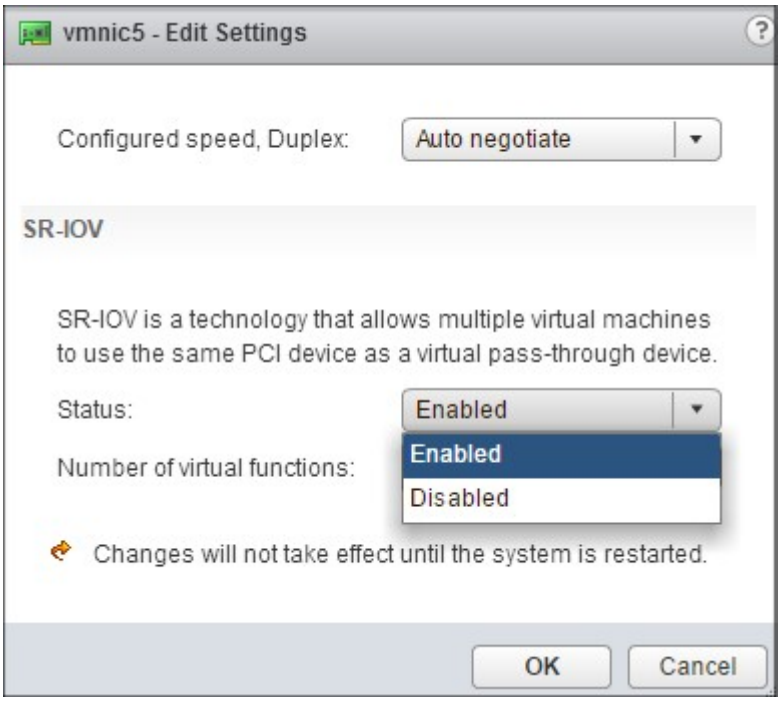
- インテル 82599 NIC (NIC) を ESX ホストに追加します。IXGBE ドライバーのバージョン 3.7.13.7.14iov をお勧めします。
- 次のように、ホスト物理アダプタで SR-IOV を有効にします。
  1. vSphere Web クライアントで、ホストに移動します。
  2. [管理] **>** [ネットワーク] タブで、[物理アダプタ] を選択します。［SR-IOV Status］フィールドに、物理アダプターが SR-IOV をサポートしているかどうかが表示されます。



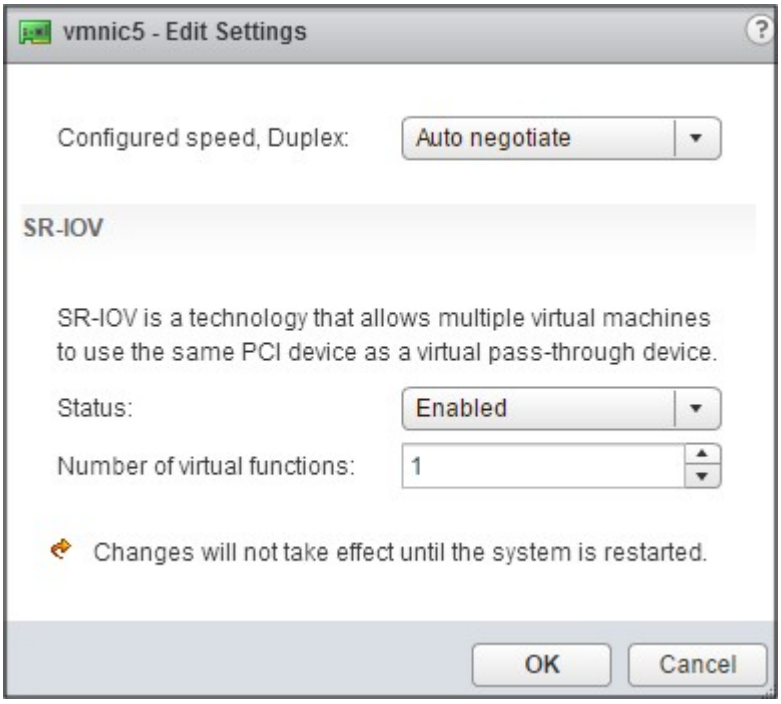
  3. 物理アダプタを選択し、鉛筆アイコンをクリックして [設定の編集] ダイアログボックスを開きます。



4.「SR-IOV」で、「ステータス」ドロップダウンリストから「有効」を選択します。



5. [仮想関数の数] フィールドに、アダプタに対して構成する仮想関数の数を入力します。



6. [**OK**] をクリックします。
7. ホストを再起動します。

- 分散仮想スイッチ (DVS) と`Portgroups`を作成します。手順については、VMware のドキュメントを参照してください。

> 注
>
> Citrix は、DVS および`Portgroups`でのみ SR-IOV 構成を認定しています。

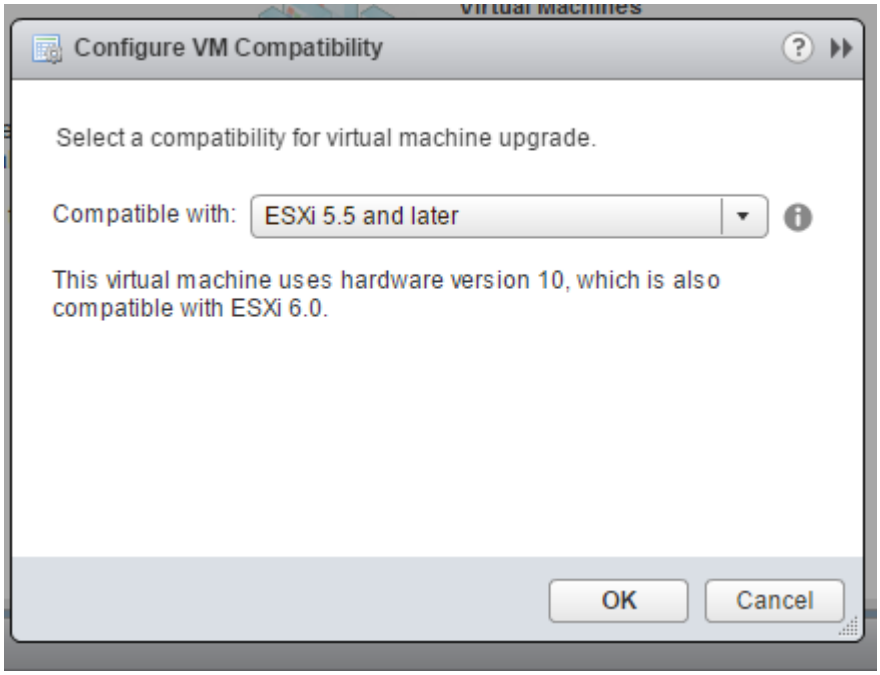
VMware vSphere Web クライアントを使用して **SR-IOV** ネットワークインターフェイスを使用するように **Citrix ADC VPX** インスタンスを構成するには:

1. vSphere Web Client で、[ホストとクラスター] を選択します。
2. 以下の手順に従って、Citrix ADC VPX インスタンスの互換性設定を ESX 5.5 以降にアップグレードします。

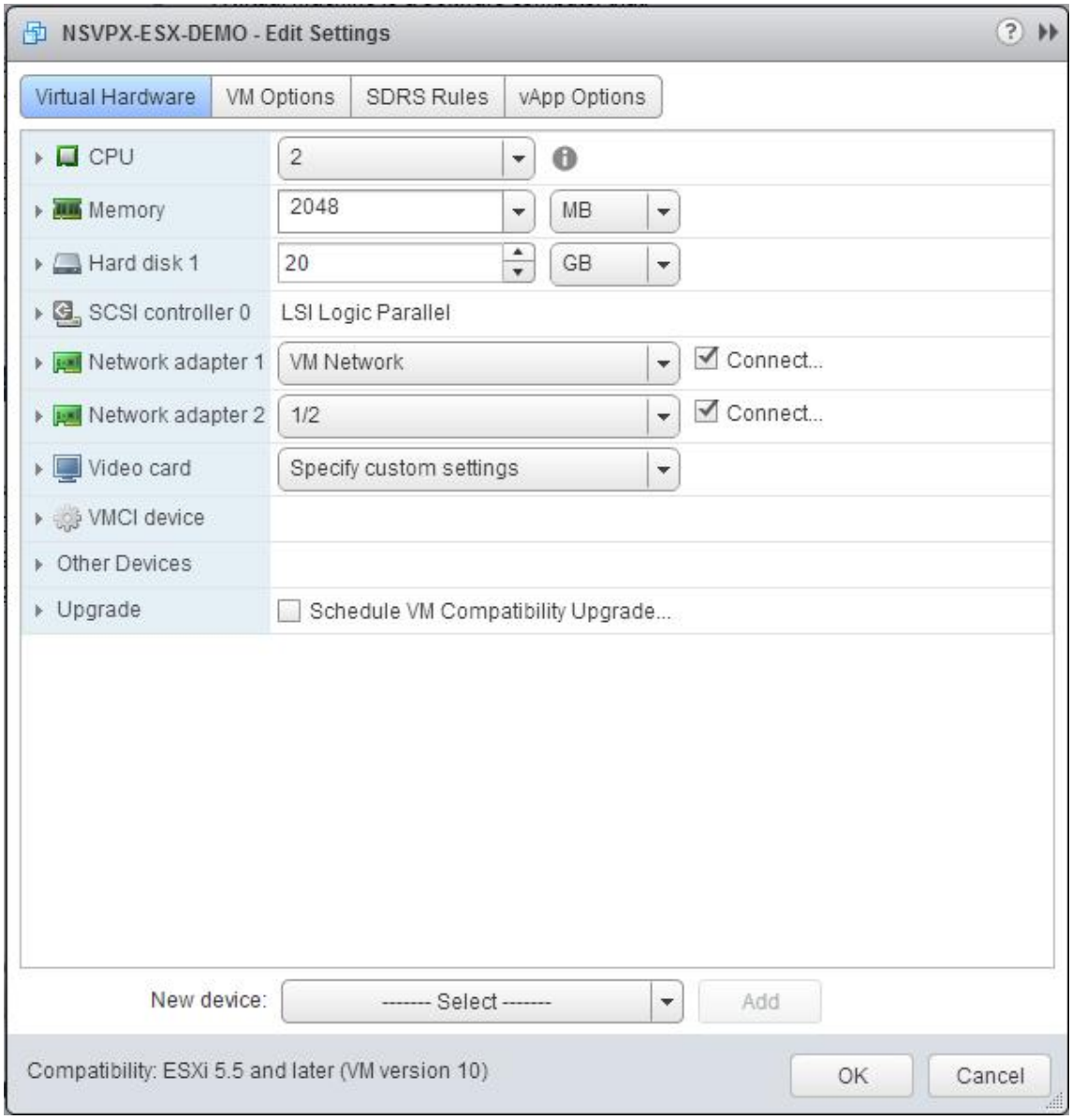
   a. Citrix ADC VPX インスタンスの電源を切ります。

   b. Citrix ADC VPX インスタンスを右クリックし、[互換性] **>** [仮想マシンの互換性のアップグレード] を選択します。

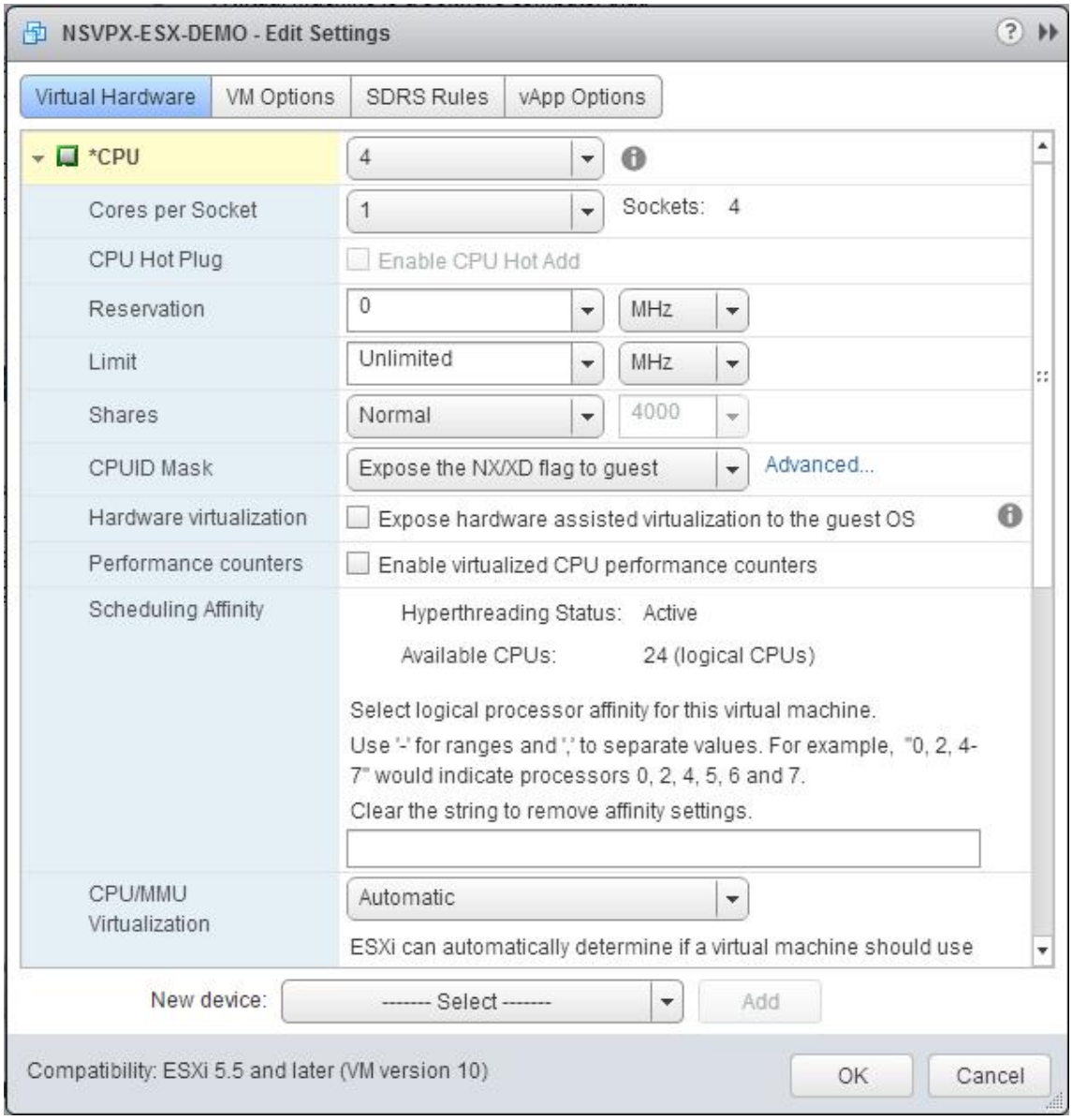
   c. 設定 **VM** 互換性] ダイアログボックスで 、**[OK]** ドロップダウンリスト に対応し、クリックから の **ESXi 5.5** 以降を選択します。



3. Citrix ADC VPX インスタンスを右クリックし、**[設定の編集]** をクリックします。



4. [**<virtual_appliance> 設定の編集]** ダイアログボックスで、**[**CPU**]** セクションをクリックします。



5. [**CPU**] セクションで、次の設定を更新します。

   - CPU の数
   - ソケット数
   - 予約
   - 上限
   - 共有

   値を次のように設定します。

   a. [**CPU**] ドロップダウンリストで、仮想アプライアンスに割り当てる CPU の数を選択します。

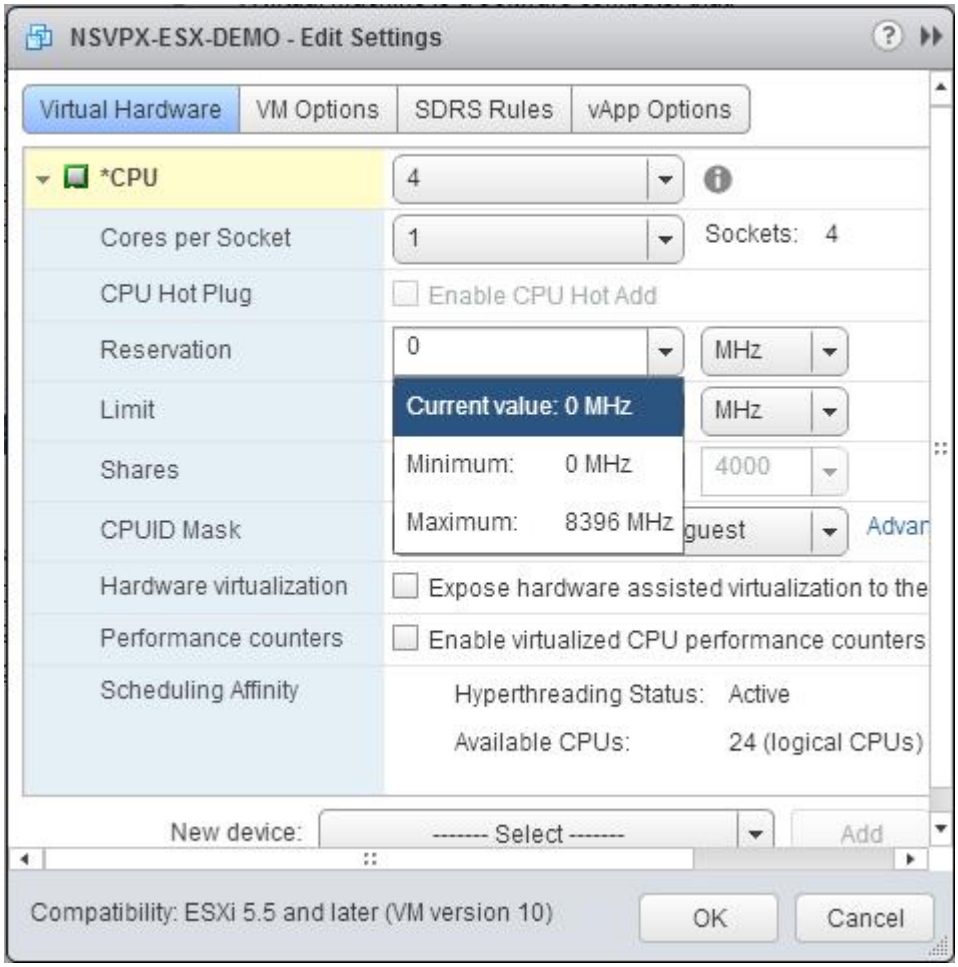
   b. [ソケットあたりのコア数] ドロップダウンリストで、ソケット数を選択します。

   c. (オプション) [**CPU** ホットプラグ] フィールドで、[**CPU** ホットアド**]** チェックボックスをオンまたはオフ

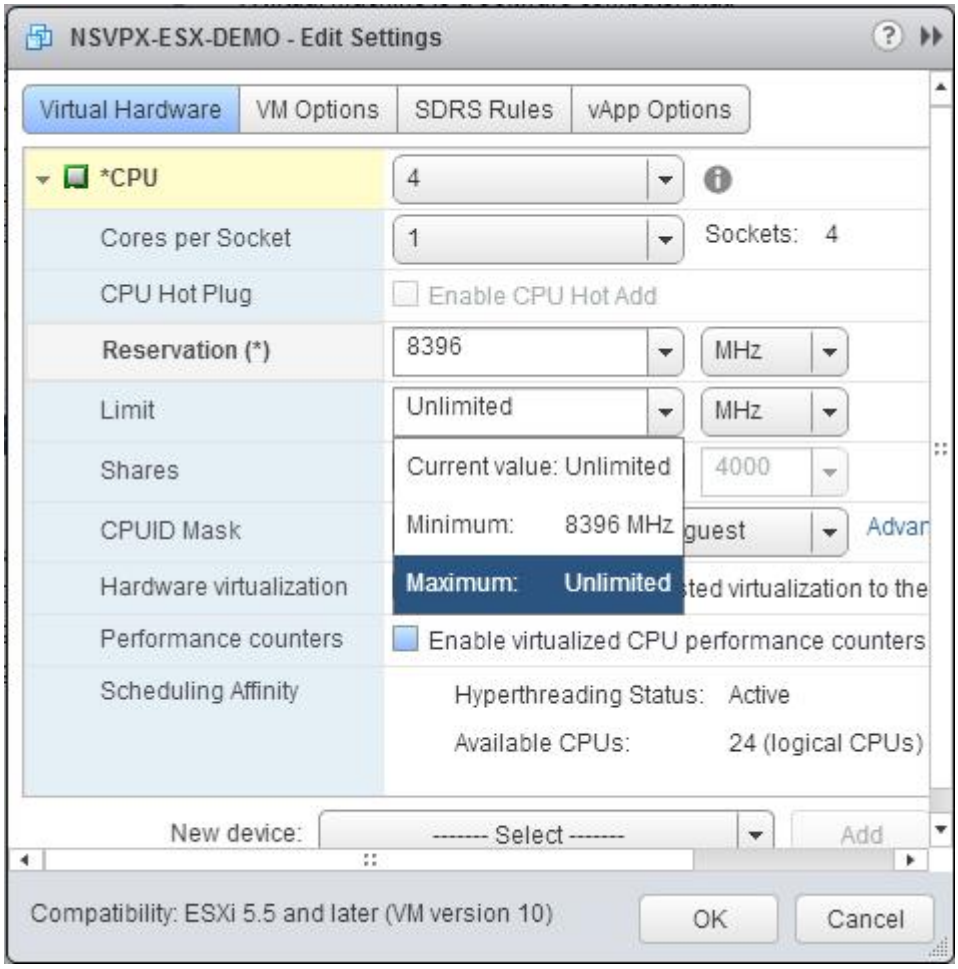
にします。

注：デフォルト（無効）を受け入れることをお勧めします。

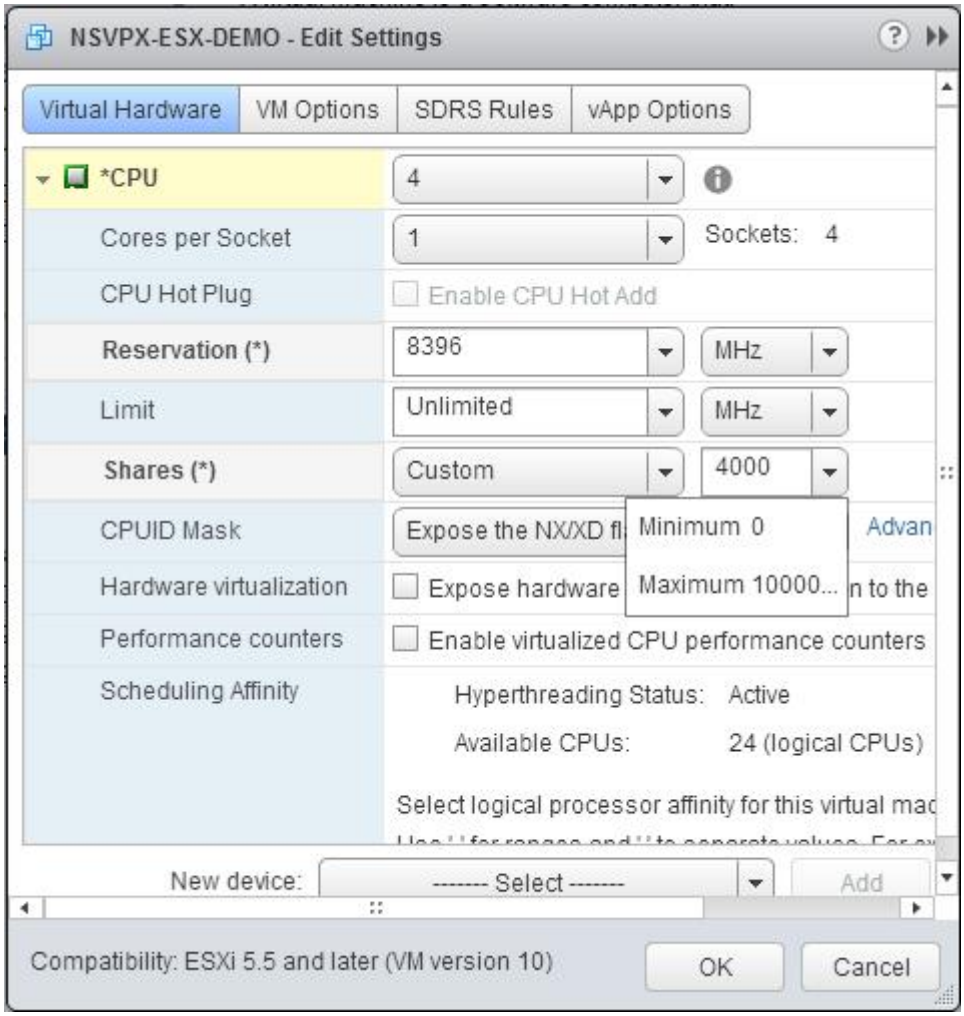
d. [予約] ドロップダウンリストで、最大値として表示される番号を選択します。



e. [制限] ドロップダウンリストで、最大値として表示されている数値を選択します。



f. [共有] ドロップダウンリストで、[カスタム] を選択し、最大値として表示される数を指定します。



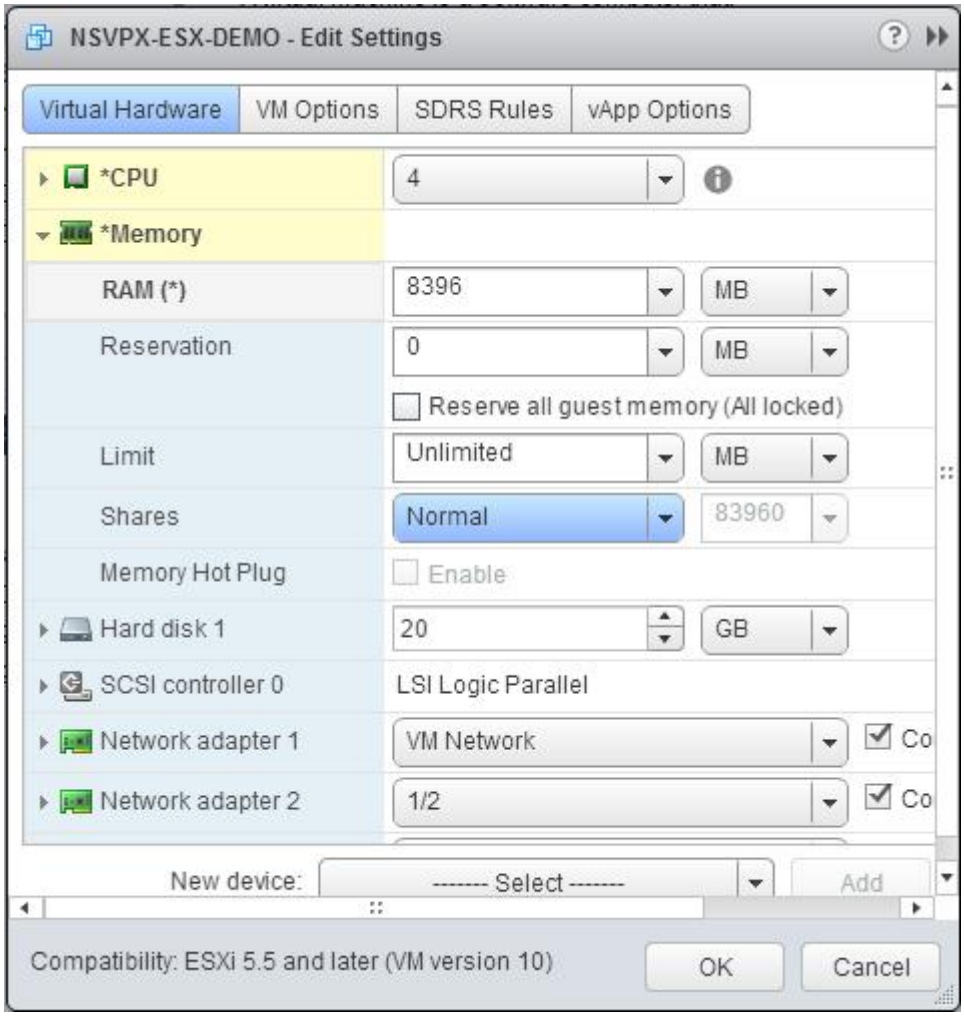
6. [メモリ] セクションで、次の設定を更新します。

   - RAM のサイズ
   - 予約
   - 上限
   - 共有

   値を次のように設定します。

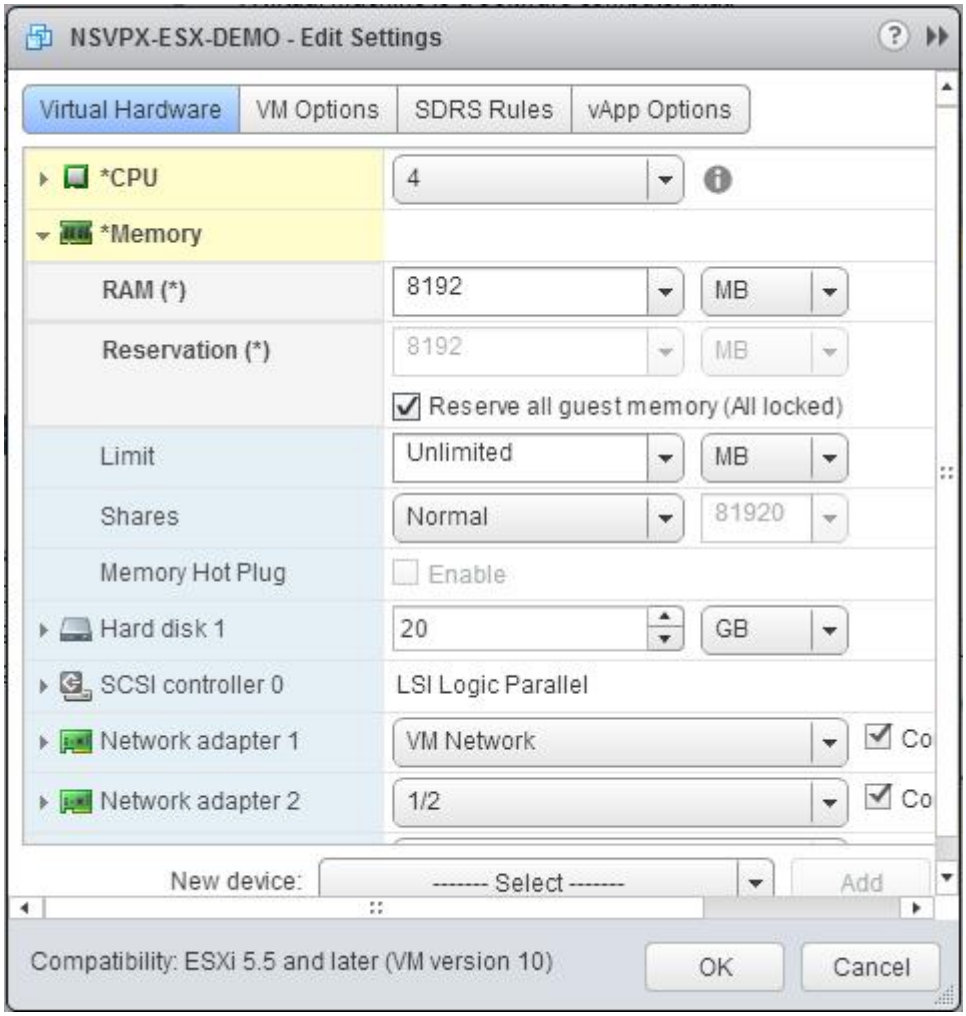
   a. [**RAM**] ドロップダウンリストで、RAM のサイズを選択します。これは vCPU の数 x 2 GB でなければなりません。たとえば、vCPU の数が 4 の場合、RAM = 4×2GB = 8GB になります。

   注： Citrix ADC VPX アプライアンスの高度なエディションまたはプレミアムエディションの場合は、各 vCPU に 4 GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。

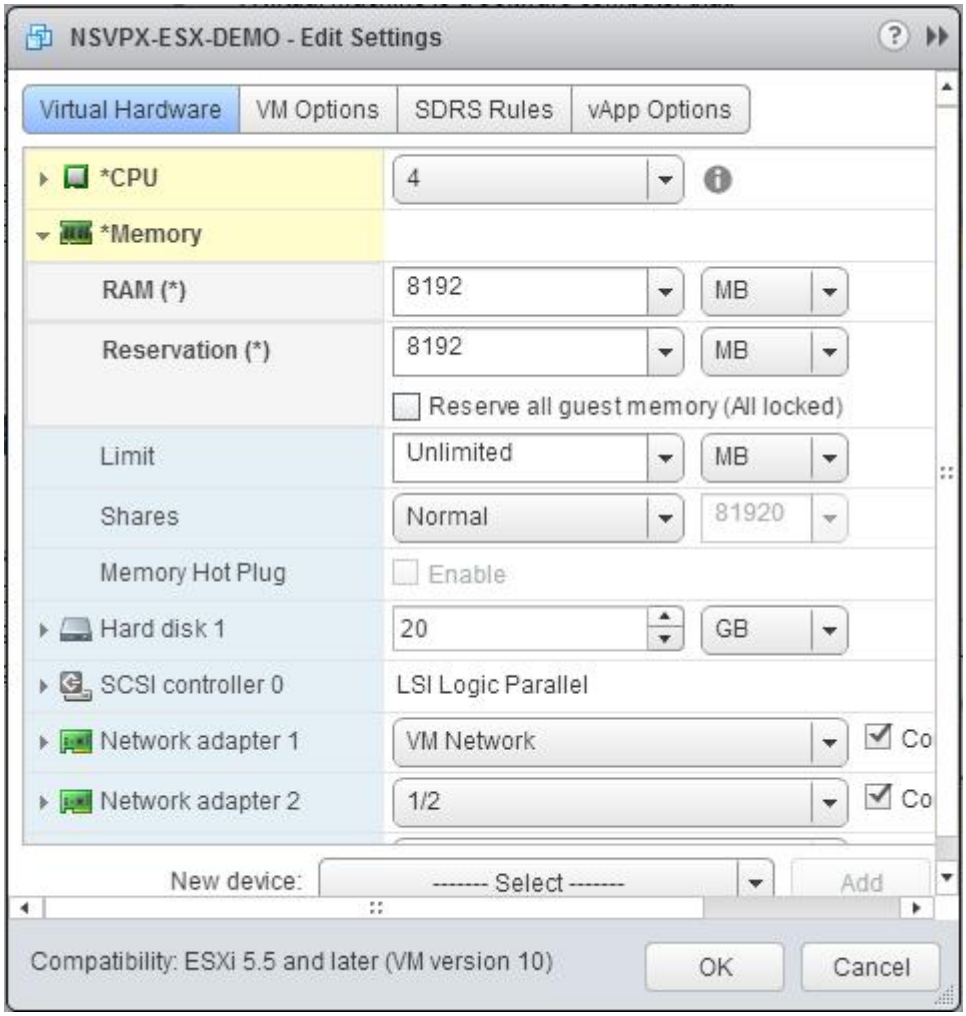


b. [予約] ドロップダウンリストで、メモリ予約の値を入力し、[すべてのゲストメモリを予約する **(**すべてのロック済み**)**] チェックボックスをオンにします。メモリ予約は vCPU の数 x 2 GB である必要があります。たとえば、vCPU の数が 4 の場合、メモリ予約は 4 x 2 GB = 8 GB である必要があります。

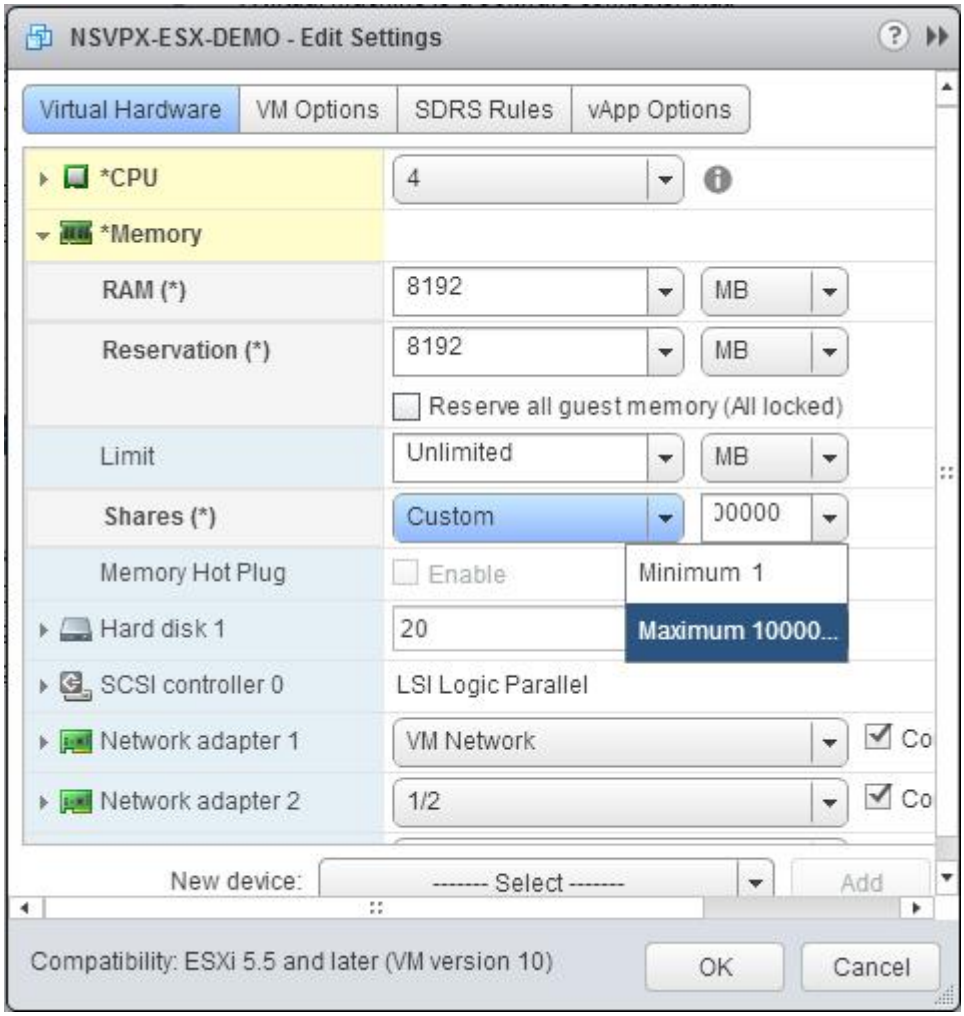
注： Citrix ADC VPX アプライアンスの高度なエディションまたはプレミアムエディションの場合は、各 vCPU に 4 GB の RAM を割り当ててください。たとえば、vCPU の数が 4 の場合、RAM = 4×4GB = 16GB になります。



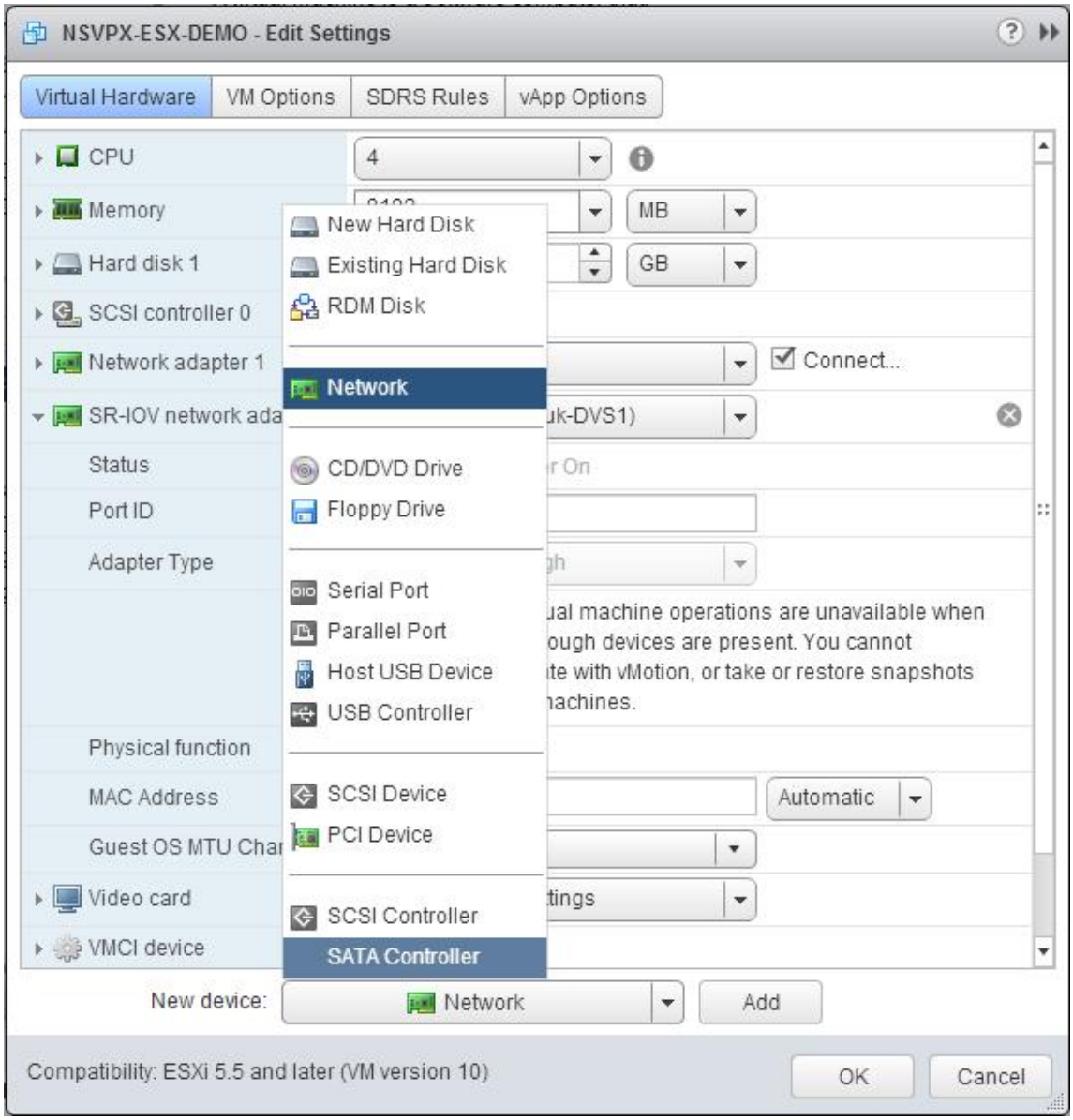
c. [制限] ドロップダウンリストで、最大値として表示されている数値を選択します。



d. [共有] ドロップダウンリストで、[カスタム] を選択し、最大値として表示される数値を選択します。

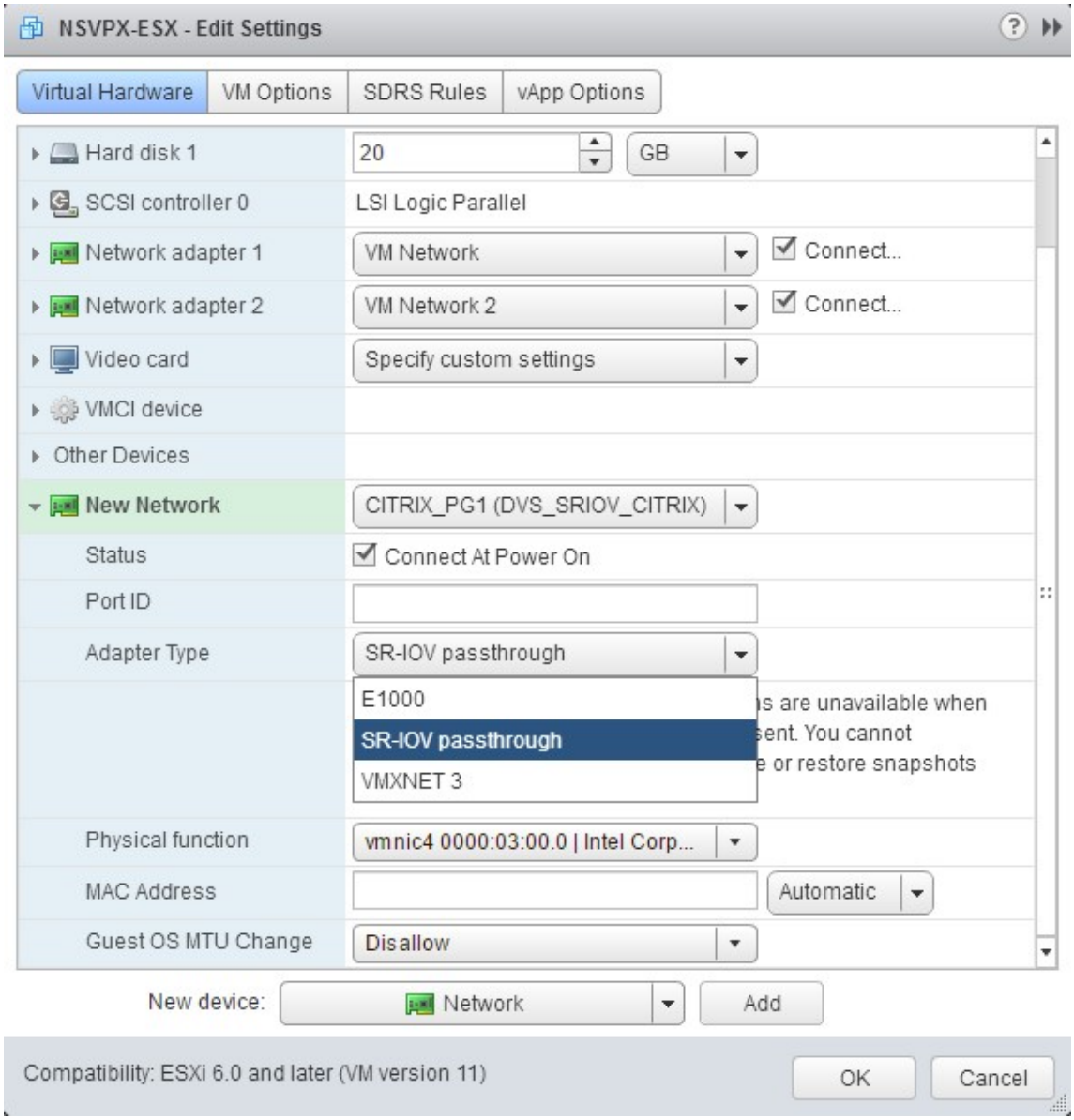


7. SR-IOV ネットワークインターフェイスを追加します。[新しいデバイス] ドロップダウンリストから [ネットワーク] を選択し、[追加] をクリックします。

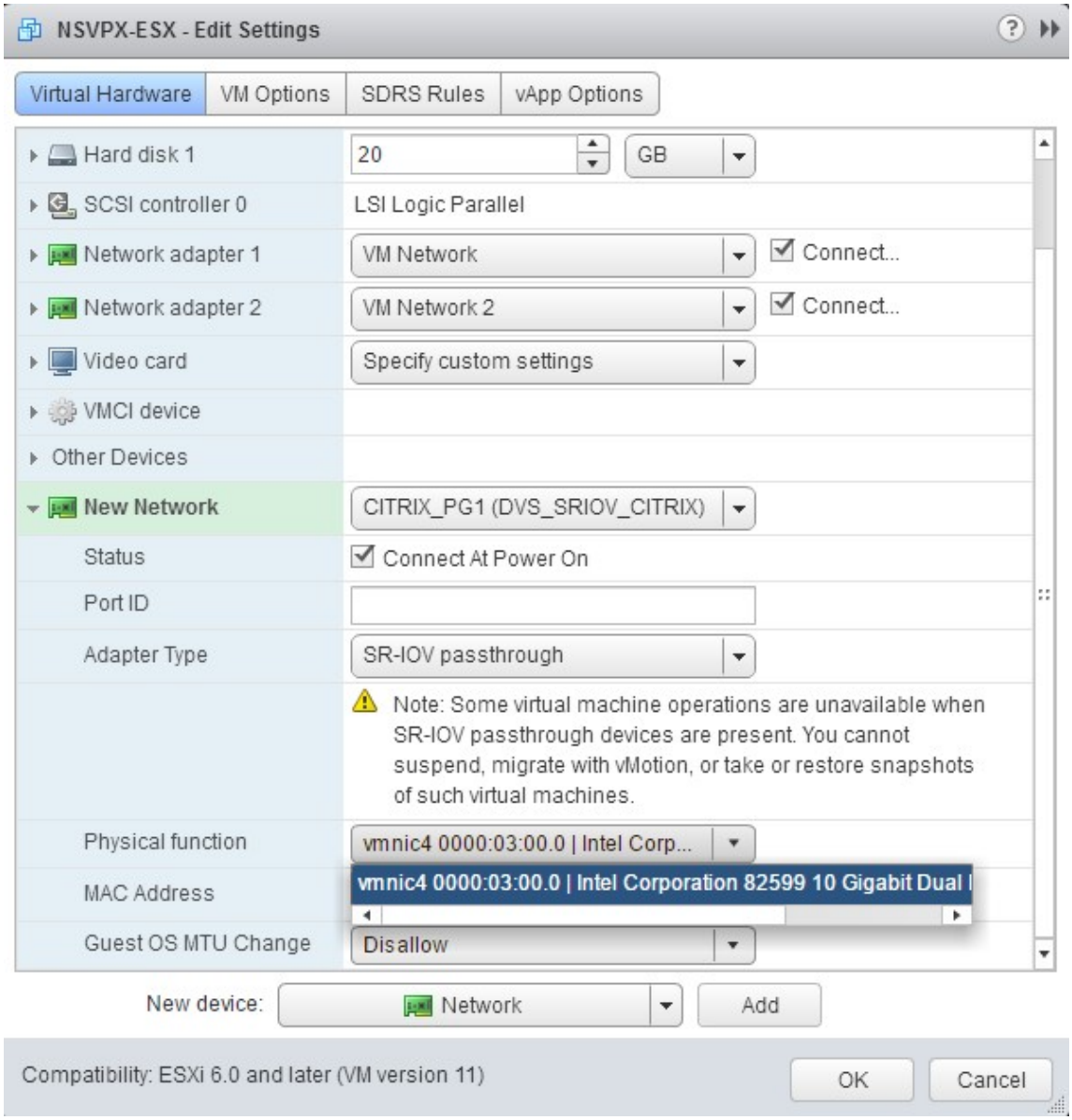


8. [新しいネットワーク] セクションで。ドロップダウンリストから、作成したPortgroupを選択し、次の操作を行います。

   a. [アダプタタイプ] ドロップダウンリストで、[**SR-IOV** パススルー] を選択します。

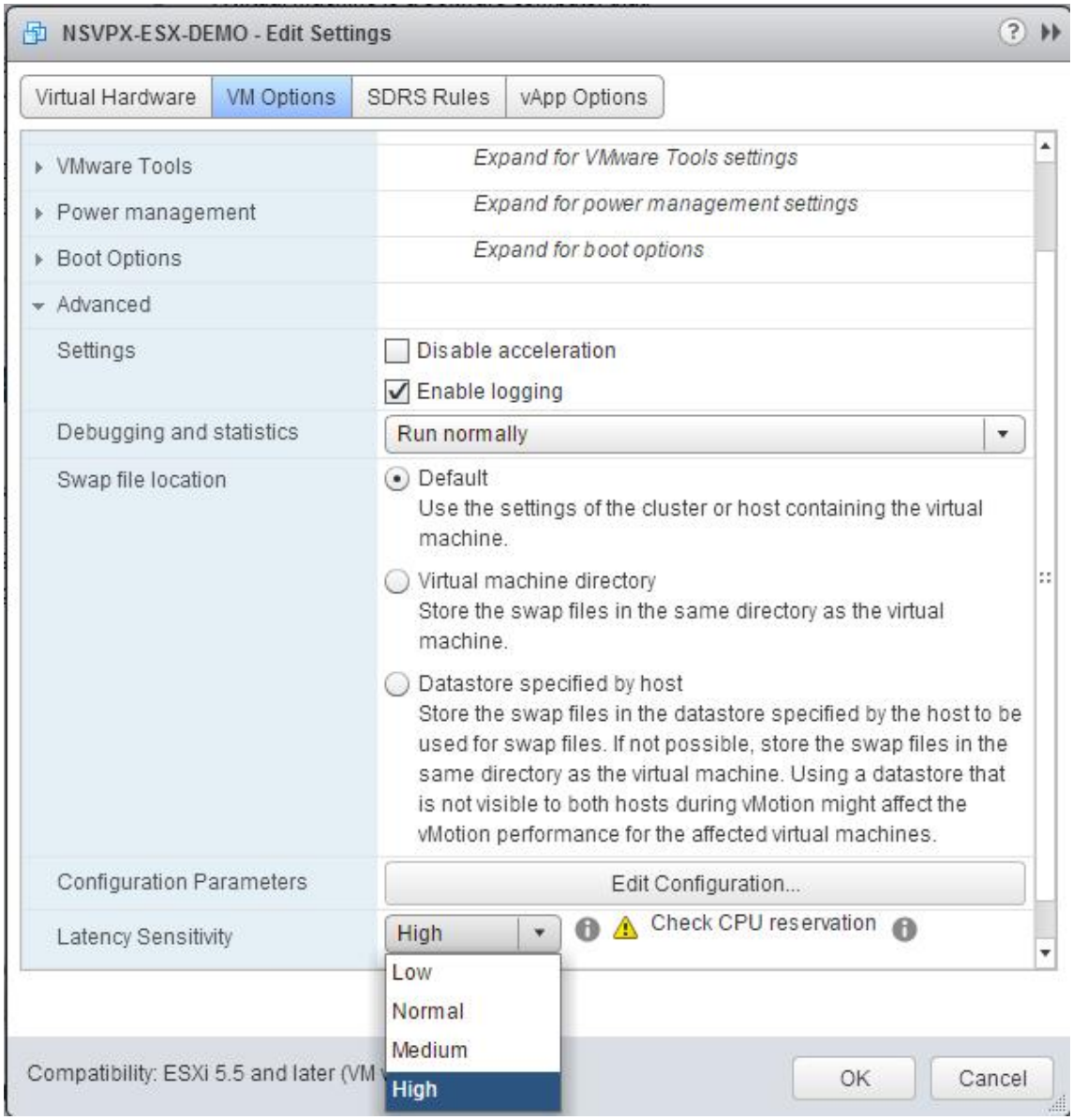


b. [物理機能] ドロップダウンリストで、`Portgroup`にマップされている物理アダプタを選択します。



c. [ゲスト **OS MTU** の変更] ドロップダウンリストで、[許可しない] を選択します。

9. [****<virtual_appliance>** 設定の編集**]** ダイアログボックスで、**[**VM オプション **] タブをクリックします。
10. [**VM** オプション] タブで、[詳細設定] セクションを選択します。[遅延感度] ドロップダウンリストから、[高] を選択します。



11. [**OK**] をクリックします。
12. Citrix ADC VPX インスタンスの電源を入れます。
13. Citrix ADC VPX インスタンスの電源が入ったら、次のコマンドを使用して構成を確認できます。

    > インターフェイスの概要を表示

出力には、設定したすべてのインターフェイスが表示されている必要があります。

```
1 > show interface summary
2 ---------------------------------------------------------------------------
3       Interface  MTU        MAC                Suffix
```



```
4 ----------------------------------------------------------------------------

5 1     0/1         1500        00:0c:29:1b:81:0b    NetScaler Virtual
     Interface
6 2     10/1        1500        00:50:56:9f:0c:6f    Intel 82599 10G VF
     Interface
7 3     10/2        1500        00:50:56:9f:5c:1e    Intel 82599 10G VF
     Interface
8 4     10/3        1500        00:50:56:9f:02:1b    Intel 82599 10G VF
     Interface
9 5     10/4        1500        00:50:56:9f:5a:1d    Intel 82599 10G VF
     Interface
10 6    10/5        1500        00:50:56:9f:4e:0b    Intel 82599 10G VF
     Interface
11 7    LO/1        1500        00:0c:29:1b:81:0b    Netscaler Loopback
     interface
12  Done
13 > show inter 10/1
14 1)       Interface 10/1 (Intel 82599 10G VF Interface) #1
15          flags=0xe460 <ENABLED, UP, UP, HAMON, 802.1q>
16          MTU=1500, native vlan=55, MAC=00:50:56:9f:0c:6f, uptime 0
               h21m53s
17          Actual: media FIBER, speed 10000, duplex FULL, fctl NONE,
               throughput 10000
18          LLDP Mode: NONE,                 LR Priority: 1024
19
20          RX: Pkts(838020742) Bytes(860888485431) Errs(0) Drops(2527)
               Stalls(0)
21          TX: Pkts(838149954) Bytes(860895860507) Errs(0) Drops(0) Stalls
               (0)
22          NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
23          Bandwidth thresholds are not set.
24  Done
```

Citrix ADC VPX を E1000 から SR-IOV または VMXNET3 ネットワークインターフェイスに移行する

October 7, 2021

2018 年 5 月 24 日

E1000 ネットワークインターフェイスを使用する既存の Citrix ADC VPX インスタンスで、SR-IOV または

VMXNET3 ネットワークインターフェイスを使用するように構成できます。

SR-IOV ネットワークインターフェイスを使用するように既存の Citrix ADC VPX インスタンスを構成するには、「SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する」を参照してください。

VMXNET3 ネットワークインターフェイスを使用するように既存の Citrix ADC VPX インスタンスを構成するには、VMXNET3 ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成するを参照してください。

PCI パススルーネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する

October 7, 2021

概要

VMware ESX Server に Citrix ADC VPX インスタンスをインストールして構成したら、vSphere Web クライアントを使用して、PCI パススルーネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

PCI パススルー機能では、ゲスト仮想マシンからホストに接続された物理 PCI および PCIe デバイスに直接アクセスできます。

前提条件

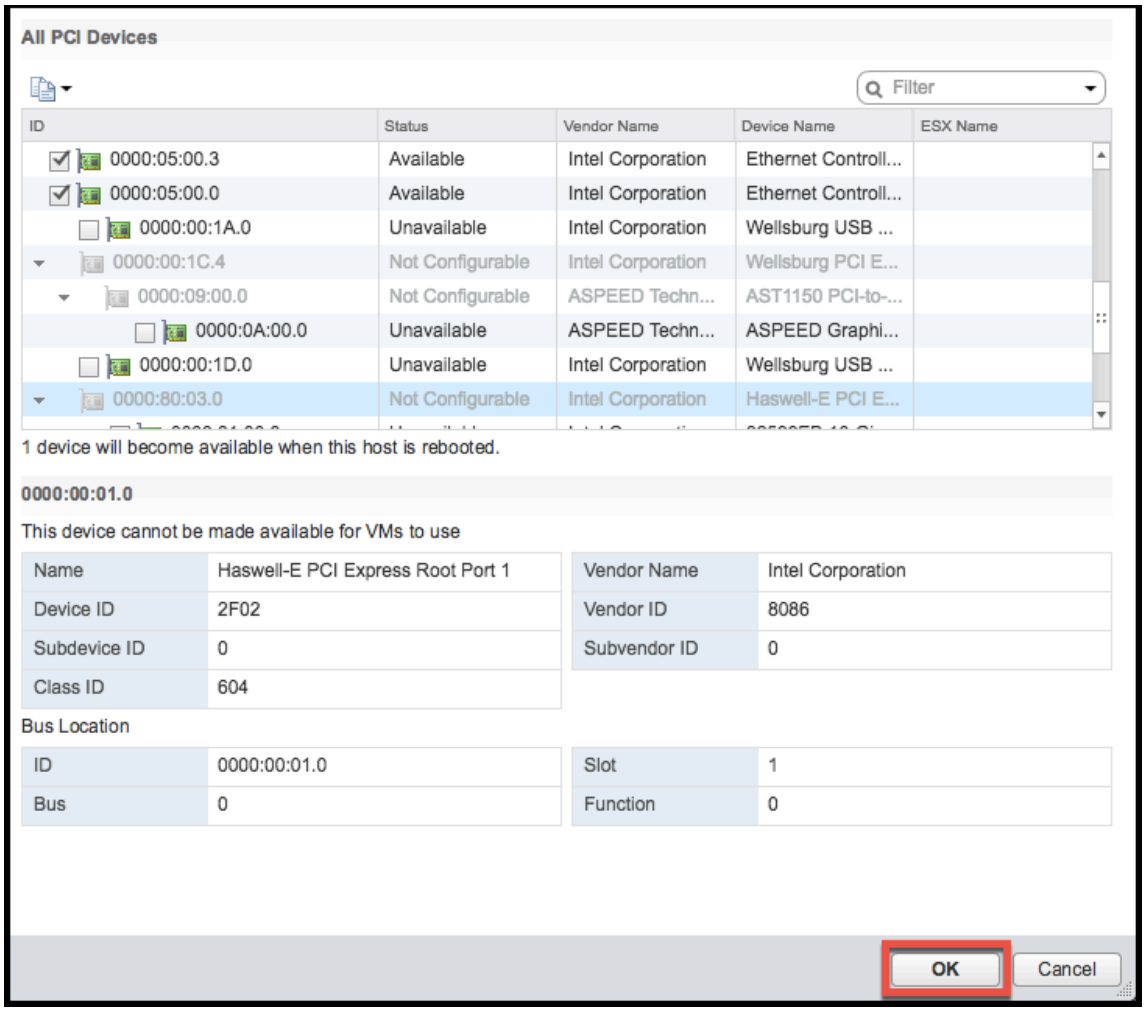
- ホストの Intel XL710 NIC のファームウェアバージョンは、5.04 です。
- ホストに接続され構成されている PCI パススルーデバイス
- サポートされている NIC:
  - Intel X710 10G NIC
  - Intel XL710 Dual Port 40G NIC
  - Intel XL710 Single Port 40G NIC

ホスト上のパススルー・デバイスの構成

仮想マシンでパススルー PCI デバイスを構成する前に、ホストマシン上でそれを構成する必要があります。ホストでパススルーデバイスを構成するには次の手順を実行します。

1. vSphere Web クライアントのナビゲーターパネルからホストを選択します。
2. 管理 > 設定 > **PCI** デバイスをクリックします。すべての利用可能なパススルーデバイスが表示されます。
3. 構成するデバイスを右クリックし、[**Edit**] をクリックします。

4. 「**PCI** デバイスの可用性の編集」ウィンドウが表示されます。
5. パススルーに使用するデバイスを選択し、[**OK**] をクリックします。

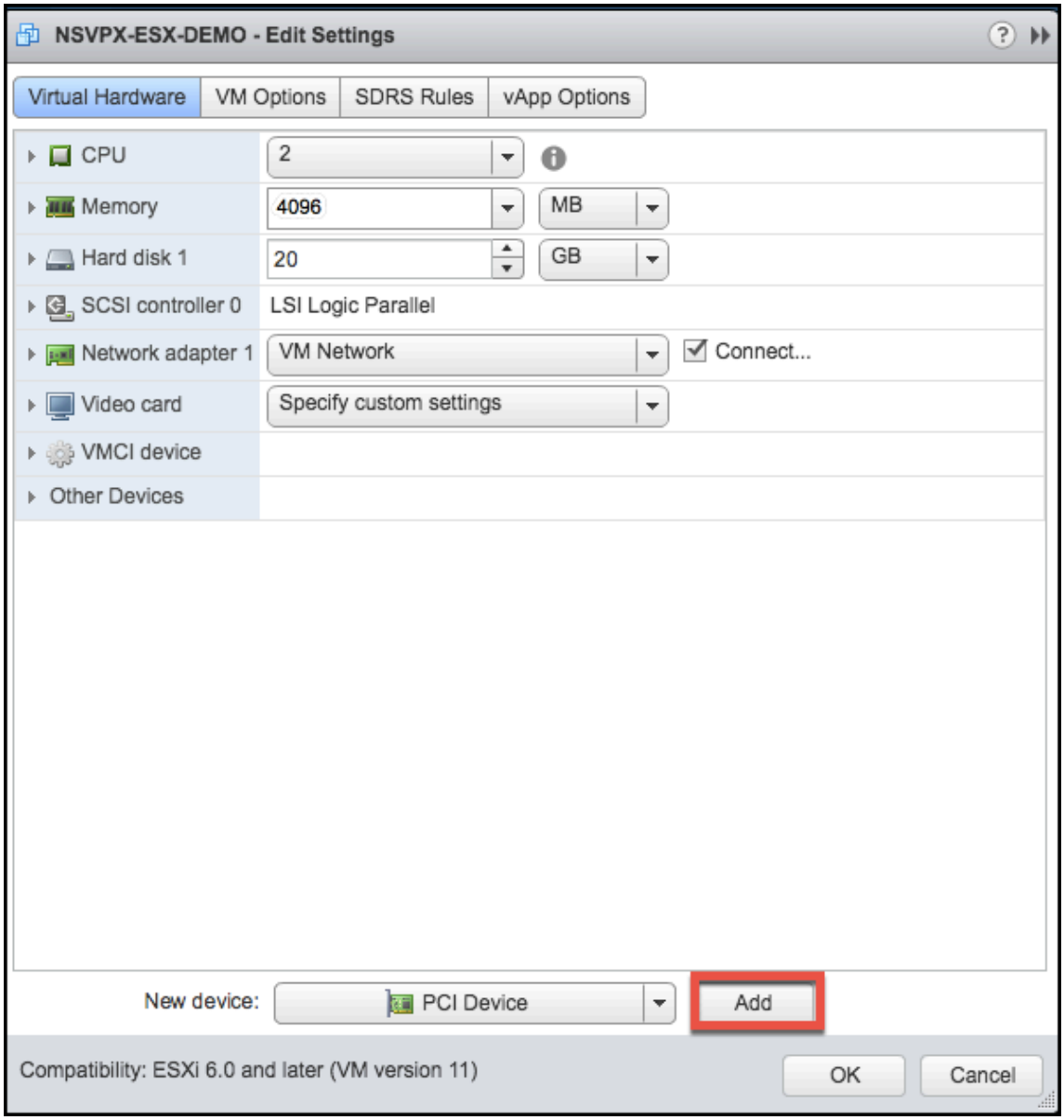


6. ホストマシンを再起動します。

Citrix ADC VPX インスタンスでのパススルーデバイスの構成

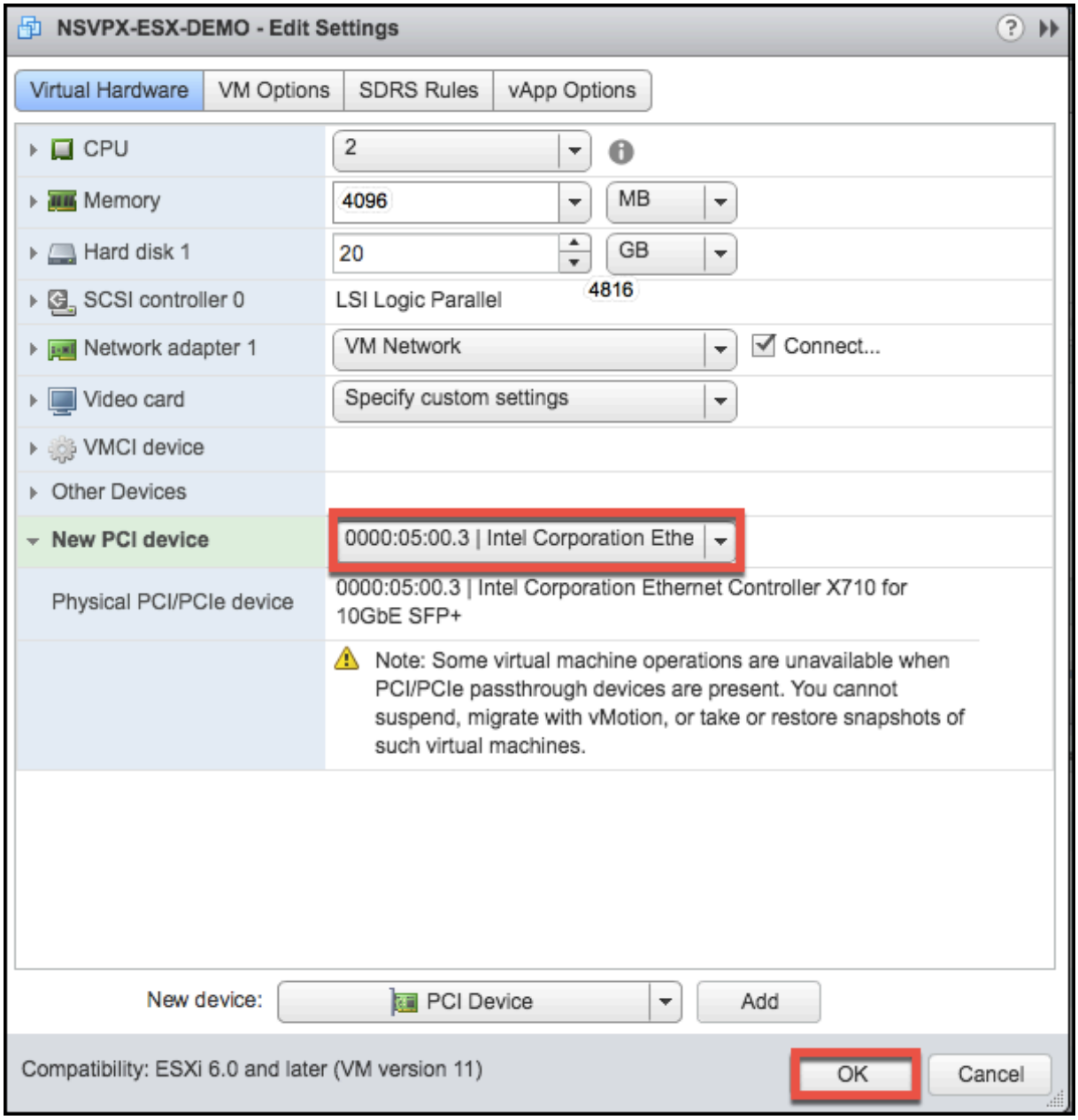
Citrix ADC VPX インスタンスでパススルー PCI デバイスを構成するには、次の手順に従います。

1. 仮想マシンの電源を切ります。
2. 仮想マシンを右クリックし、[設定の編集] を選択します。
3. [仮想ハードウェア] タブで、[新しい ****** デバイス] ドロップダウンメニューから **[PCI** デバイス] を選択し、**[**追加 ******] をクリックします。



4. [**New PCI device**] を展開し、ドロップダウンリストから仮想マシンに接続するパススルーデバイスを選択し、[**OK**] をクリックします。

> 注
>
> VMXNET3 ネットワークインターフェイスと PCI パススルーネットワークインターフェイスは共存できません。



1. ゲスト仮想マシンの電源を入れます。

PCI パススルーネットワークインターフェイスを使用するように Citrix ADC VPX を設定する手順は完了です。

Apply Citrix ADC VPX configurations at the first boot of the Citrix ADC appliance on VMware ESX hypervisor

December 7, 2021

You can apply the Citrix ADC VPX configurations during the first boot of the Citrix ADC appliance on the VMware ESX hypervisor. Therefore in certain cases, a specific setup or VPX instance is brought up in much lesser time.

For more information on Preboot user data and its format, see Apply Citrix ADC VPX configurations at the first boot of the Citrix ADC appliance in cloud.

> Note:
>
> To bootstrap using preboot user data in ESX, default gateway config must be passed in <NS-CONFIG> section. For more information on the content of the <NS-CONFIG> tag, see Sample-<NS-CONFIG>-section.

Sample <NS-CONFIG> section:

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

How to provide preboot user data on ESX hypervisor

You can provide preboot user data on ESX hypervisor in the following two ways:

- Using CD/DVD ISO
- Using OVF Property

Provide user data using CD/DVD ISO

You can use VMware vSphere client to inject user data into the VM as an ISO image using the CD/DVD drive.

Follow these steps to provide user data using CD/DVD ISO:

1. Create a file with file name `userdata` that contains the preboot user data content. For more information on the content of the <NS-CONFIG> tag, see Sample <NS-CONFIG> section.

   > Note: File name must be strictly used as `userdata`.

2. Store the `userdata` file in a folder, and build an ISO image using the folder.

   You can build an ISO image with `userdata` file by the following two methods:

   - Using any image processing tool such as PowerISO.
   - Using `mkisofs` command in Linux.

   The following sample configuration shows how to generate an ISO image using the `mkisofs` command in Linux.

```
root@ubuntu:~/sai/14jul2021# ls -l total 4
drwxr-xr-x 2 root root 4096 Jul 14 12:32 esx_preboot_userdata
root@ubuntu:~/sai/14jul2021#
root@ubuntu:~/sai/14jul2021# ls -l esx_preboot_userdata/total 4
-rw-r--r-- 1 root root 3016 Jul 14 12:32 userdata
root@ubuntu:~/sai/14jul2021# mkisofs -o esx_preboot_userdata.iso
    ./esx_preboot_userdata
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)
root@ubuntu:~/sai/14jul2021# ls -lh
total 356K
drwxr-xr-x 2 root root 4.0K Jul 14 12:32 esx_preboot_userdata
-rw-r--r-- 1 root root 352K Jul 14 12:34 esx_preboot_userdata.iso

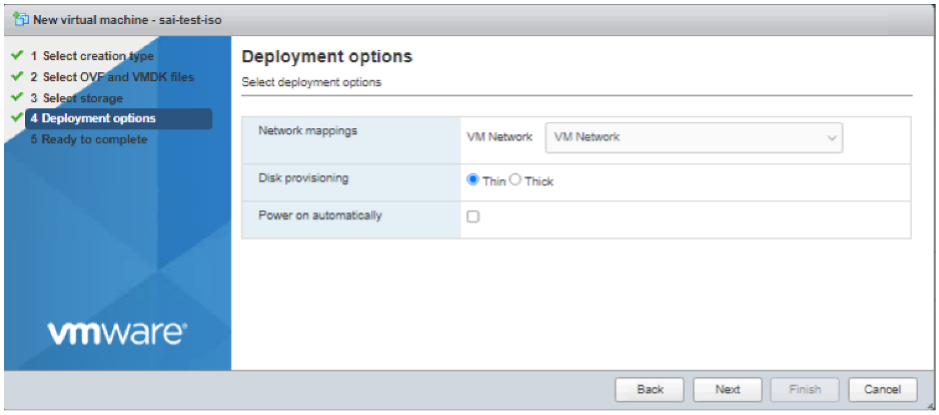
root@ubuntu:~/sai# ls preboot_userdata_155_193 userdata
root@ubuntu:~/sai# mkisofs -o preboot_userdata_155_193.iso ./
    preboot_userdata_155_193
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
```



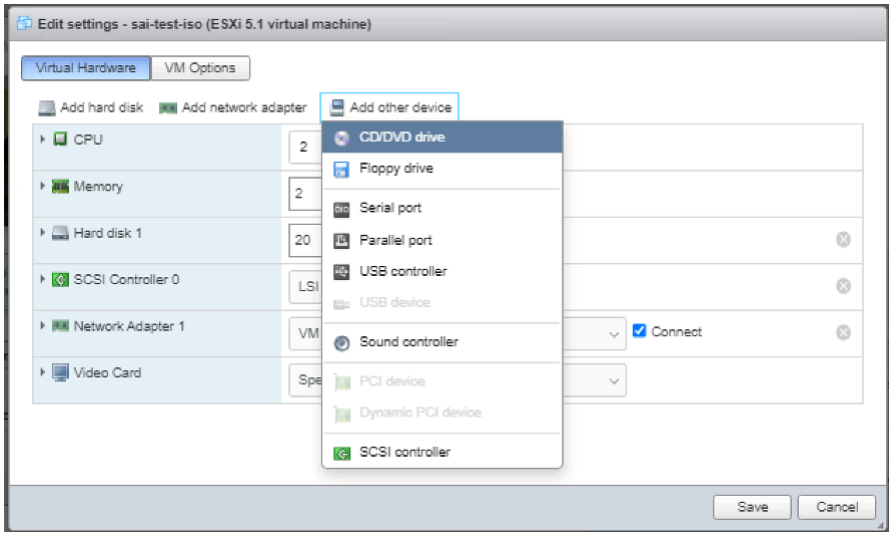
```
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)

<!--NeedCopy-->
```

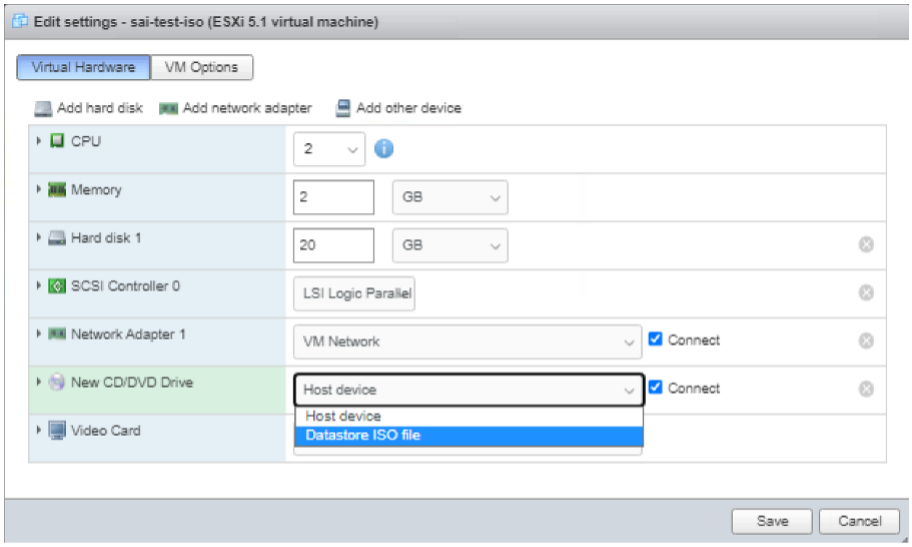
3. Provision the Citrix ADC VPX instance using standard deployment process to create the VM. But do not power on the VM automatically.



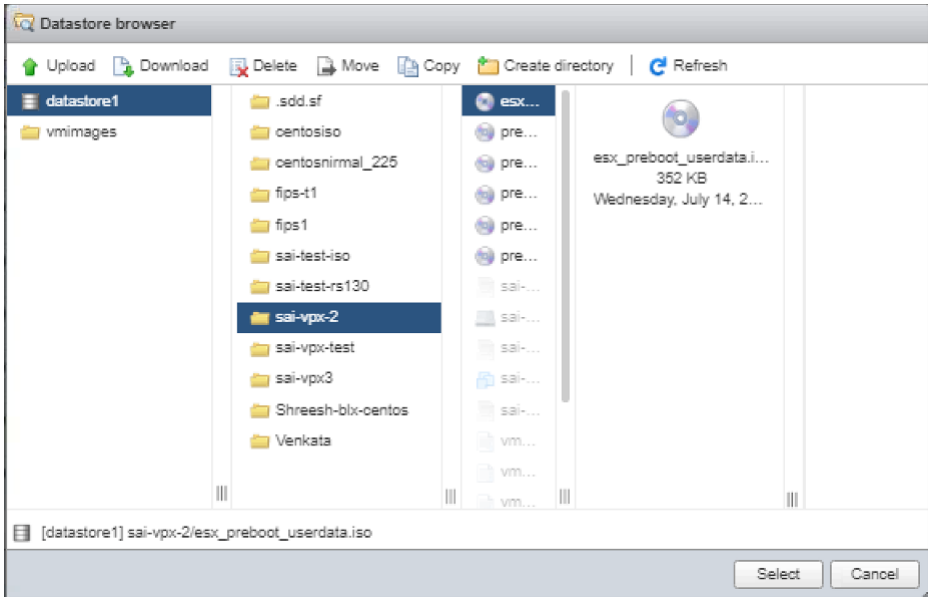
4. After the VM is successfully created, attach the ISO file as CD/DVD drive to the VM.



5. Navigate to **New CD/DVD Drive** and choose **Datastore ISO file** from the drop-down menu.



6. Select a Datastore in the vSphere Client.



7. Power on the VM.

Provide user data using OVF property

Follow these steps to provide user data using OVF property.

1. Create a file with user data content.


```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

2. Encode the user data content with Base64 encoding. You can perform the Base64 encoding using the following two methods:

   - In Linux, use the following command:

   ```
   base64 <userdata-filename>  >   <outuput-file>
   <!--NeedCopy-->
   ```

   Example:

   ```
   base64 esx_userdata.xml  > esx_userdata_b64
   <!--NeedCopy-->
   ```

   ```
   root@ubuntu:~/sai/14jul2021# base64 esx_userdata.xml > esx_userdata_b64
   root@ubuntu:~/sai/14jul2021#
   root@ubuntu:~/sai/14jul2021# cat esx_userdata_b64
   PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+CglhZGQgcm91dGUgMC4wLjAuMCAw
   LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQPgog
   ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1CT09U
   U1RSQVA+CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
   VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAgICAg
   ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAgICAg
   ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5FVC1N
   QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
   RS1DT05GSUc+CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg==
   ```

   - Use online tools to encode user data content, for example, Base64 Encode and Decode.

3. Include a **Product** section in the OVF template of a Citrix ADC VPX instance on ESX hypervisor.

   Sample Product section:


```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>

  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
<!--NeedCopy-->
```

4. Provide the base64 encoded user data as the `ovf:value` for `guestinfo.userdata` property in the Product section.

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true"
    ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
       CglhZGQgcm91dGUgMC4wLjAuMCAw
    LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQ

    ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C

    U1RSQVA+
       CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT

    VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg

    ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
```

```
15      ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5F
16      QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
           CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
17      RS1DT05GSUc+
           CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
           ==”>
18
19      <Label>Userdata</Label>
20      <Description> Userdata for ESX VPX </Description>
21    </Property>
22
23  </ProductSection>
24  <!--NeedCopy-->
```

5. Use the modified OVF template with Product section for the VM deployment.

```
Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
 Done
> sh ns ver
        NetScaler NS13.0: Build 83.9005.nc, Date: Jul 13 2021, 02:56:05   (64-bit)
 Done
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  S
tate
        ---------        --------------  ----             ----     ---      ----     -------  -
-----
1)      10.102.38.219    0               NetScaler IP     Active   Enabled  Enabled  NA       E
nabled
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State    Traffic Domain  Type
        -------          -------          ---------------  -----   -----    --------------  ----
1)      0.0.0.0          0.0.0.0          10.102.38.1      0       UP       0               STATI
C
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP       0               PERMA
NENT
3)      10.102.38.0      255.255.255.0    10.102.38.219    0       UP       0               DIREC
T
 Done
```

AWS 上の VMware クラウドに Citrix ADC VPX インスタンスをインストールする

October 7, 2021

AWS 上の VMware クラウド（VMC）を使用すると、希望する数の ESX ホストで AWS 上にクラウドソフトウェア定義データセンター（SDDC）を作成できます。AWS 上の VMC は、Citrix ADC VPX デプロイメントをサポートしています。VMC は、オンプレミスの vCenter と同じユーザー・インタフェースを提供します。これは、ESX ベースの Citrix ADC VPX 展開と同じように機能します。

前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- 1 つの VMware SDDC が、少なくとも 1 つのホストとともに存在している必要があります。
- Citrix ADC VPX アプライアンスのセットアップファイルをダウンロードします。
- 仮想マシンが接続する適切なネットワークセグメントを VMware SDDC 上に作成します。
- VPX ライセンスファイルを入手します。Citrix ADC VPX インスタンスライセンスの詳細については、http://support.citrix.com/article/ctx131110の「*Citrix ADC VPX* ライセンスガイド」を参照してください。

VMware クラウドのハードウェア要件

次の表に、VMware SDDC が各 VPX nCore 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースを示します。

表 1. Citrix ADC VPX インスタンスの実行に必要な最小仮想コンピューティングリソース

コンポーネント	条件
メモリ	2GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	VMware SDDC では、VPX ハードウェアをバージョン 7 以上にアップグレードすると、最大 10 個の仮想ネットワークインターフェイスをインストールできます。
ディスク領域	20GB

> 注
>
> ハイパーバイザーに必要なディスク領域は含まれません。

VPX 仮想アプライアンスを実稼働で使用するには、フルメモリ割り当てを予約する必要があります。

OVF ツール **1.0** のシステム要件

Open Virtualization Format Tool（OVF Tool）は、Windows および Linux システムで実行できるクライアントアプリケーションです。次の表は、最小システム要件を示しています。

表 2. OVF ツールのインストールに必要な最小システム要件

コンポーネント	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で「OVF ツールユーザーガイド」PDF ファイルを検索してください。
CPU	最低 750 MHz、1 GHz またはそれ以上の速度を推奨
RAM	最小 1 GB、推奨 2 GB
NIC	100Mbps 以上の NIC。

OVF のインストールについては、http://kb.vmware.com/で「OVF ツールユーザーガイド」の PDF ファイルを検索してください。

Citrix ADC VPX セットアップファイルのダウンロード

VMware ESX 用の Citrix ADC VPX インスタンスセットアップパッケージは、オープン仮想マシン（OVF）形式の標準に準拠しています。これらのファイルは、Citrix の Web サイトからダウンロードできます。ログオンするには、Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスしてください。[新しいユーザー] リンクをクリックし、指示に従って新しい Citrix アカウントを作成します。

ログオンしたら、Citrix のホームページから次のパスをナビゲートします。

Citrix.com > ダウンロード **> Citrix ADC >** 仮想アプライアンス。

次のファイルを、ESX サーバーと同じネットワーク上のワークステーションにコピーします。3 つのファイルをすべて同じフォルダーにコピーします。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（例えば、NSVPX-ESX-13.0-79.64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf（例えば、NSVPX-ESX-13.0-79.64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（例えば、NSVPX-ESX-13.0-79.64.mf）

VMware クラウドへの Citrix ADC VPX インスタンスのインストール

VMware SDDC をインストールして構成したら、SDDC を使用して VMware クラウドに仮想アプライアンスをインストールできます。インストールできる仮想アプライアンスの数は、SDDC で使用可能なメモリの量によって異なります。

VMware クラウドに Citrix ADC VPX インスタンスをインストールするには、次の手順に従います。

1. ワークステーションで VMware SDDC を開きます。
2. [ユーザー名] および [パスワード] テキストボックスに管理者の資格情報を入力し、[ログイン] をクリックします。

3. ［**File**］メニューの［**Deploy OVF Template**］を選択します。
4. ［**OVF** テンプレートの展開］ダイアログボックスの［ファイルから展開］で、Citrix ADC VPX インスタンスのセットアップファイルを保存した場所を参照し、.ovf ファイルを選択して［次へ］をクリックします。

   > 注：デフォルトでは、Citrix ADC VPX インスタンスは E1000 ネットワークインターフェイスを使用します。VMXNET3 インターフェイスで ADC を展開するには、E1000 ではなく VMXNET3 インターフェイスを使用するように OVF を変更します。

5. 仮想アプライアンス OVF テンプレートに表示されるネットワークを、VMware SDDC で設定したネットワークにマッピングします。［次へ］をクリックして、VMware SDDC への仮想アプライアンスのインストールを開始します。
6. これで、Citrix ADC VPX インスタンスを起動する準備ができました。ナビゲーションペインで、インストールした Citrix ADC VPX インスタンスを選択し、右クリックメニューから［電源オン］を選択します。コンソールポートをエミュレートするには、［**Console**］タブをクリックします。
7. 別の仮想アプライアンスをインストールする場合は、ステップ 6 からを繰り返します。
8. 管理ネットワークとして選択した同じセグメントから管理 IP アドレスを指定します。ゲートウェイには同じサブネットが使用されます。
9. VMware SDDC では、ネットワークセグメントに属するすべてのプライベート IP アドレスに対して NAT ルールとファイアウォールルールを明示的に作成する必要があります。

Microsoft の Hyper-V サーバーに Citrix ADC VPX インスタンスをインストールする

October 7, 2021

Microsoft Windows Server に Citrix ADC VPX インスタンスをインストールするには、まず適切なシステムリソースを持つマシンに、Hyper-V の役割を有効にして Windows Server をインストールする必要があります。Hyper-V の役割をインストールするときは、仮想ネットワークを作成するために Hyper-V で使用されるサーバー上の NIC を必ず指定してください。一部の NIC は、ホスト用に確保できます。Hyper-V マネージャーを使用して、Citrix ADC VPX インスタンスのインストールを実行します。

Hyper-V 用の Citrix ADC VPX インスタンスは、仮想ハードディスク（VHD）形式で配信されます。CPU、ネットワークインターフェイス、ハードディスクのサイズと形式などの要素について、デフォルト構成が格納されています。Citrix ADC VPX インスタンスをインストールすると、仮想アプライアンスでネットワークアダプタを構成し、仮想 NIC を追加してから、Citrix ADC IP アドレス、サブネットマスク、Gateway を割り当てて、仮想アプライアンスの基本構成を完了できます。

VPX インスタンスの初期構成後、アプライアンスを最新のソフトウェアリリースにアップグレードする場合は、Citrix ADC VPX スタンドアロンアプライアンスのアップグレードを参照してください。

> 注
>
> 中間システム間システム (ISIS) プロトコルは、HyperV-2012 プラットフォームでホストされている Citrix ADC VPX 仮想アプライアンスではサポートされていません。

Microsoft サーバーに **Citrix ADC VPX** インスタンスをインストールするための前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- Windows サーバーで Hyper-V の役割を有効にします。詳しくは、http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspxを参照してください。
- 仮想アプライアンスセットアップファイルをダウンロードします。
- Citrix ADC VPX インスタンスのライセンスファイルを取得します。Citrix ADC VPX インスタンスライセンスの詳細については、http://support.citrix.com/article/ctx131110の「*Citrix ADC VPX* ライセンスガイド」を参照してください。

Microsoft のサーバのハードウェア要件

次の表に、Microsoft サーバーの最小システム要件を示します。

表 1. Microsoft サーバの最小システム要件

コンポーネント	条件
CPU	1.4GHz 64 ビットプロセッサ
RAM	8GB
ディスク領域	32GB 以上

次の表に、各
Citrix ADC VPX インスタンスの仮想コンピューティングリソースを示します。

表 2. Citrix ADC VPX インスタンスの実行に必要な最小仮想コンピューティングリソース

コンポーネント	条件
RAM	4GB
仮想 CPU	2
ディスク領域	20GB
仮想ネットワークインターフェイス	1

Citrix ADC VPX セットアップファイルをダウンロードする

Hyper-V 用の Citrix ADC VPX インスタンスは、仮想ハードディスク（VHD）形式で配信されます。これらのファイルは、Citrix の Web サイトからダウンロードできます。ログインするには Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスし、［サインイン］**>**［マイアカウント］**>**［**Citrix** アカウントの作成］の順にクリックし、手順に従って Citrix アカウントを作成します。

Citrix ADC VPX インスタンスのセットアップファイルをダウンロードするには、次の手順に従います。

1. Web ブラウザーで、http://www.citrix.com/に移動します。
2. ユーザー名とパスワードを使用してサインインします。
3. ［ダウンロード］をクリックします。
4. ［製品の選択］ドロップダウンメニューで、［**Citrix ADC**（**NetScaler ADC**）］を選択します。
5. ［**Citrix ADC** リリース **X.X**］**>**［仮想アプライアンス］で［**Citrix ADC VPX** リリース **X.X**］をクリックします
6. 圧縮ファイルをサーバーにダウンロードします。

Microsoft のサーバーに **Citrix ADC VPX** インスタンスをインストールします

Microsoft Server で Hyper-V の役割を有効にして仮想アプライアンスファイルを抽出した後、Hyper-V マネージャーを使用して Citrix ADC VPX インスタンスをインストールできます。仮想マシンをインポートしてから仮想 NIC を構成し、Hyper-V によって作成された仮想ネットワークに関連付ける必要があります。

最大 8 つの仮想 NIC を構成できます。物理 NIC が DOWN になっても、同じホスト（サーバー）上の他の仮想アプライアンスと通信できるため、仮想アプライアンスでは仮想 NIC は UP と見なされます。

> 注
>
> 仮想アプライアンスの実行中は、設定を変更することができません。仮想アプライアンスをシャットダウンしてから変更を行います。

Hyper-V マネージャーを使用して **Microsoft** サーバーに **Citrix ADC VPX** インスタンスをインストールするには:

1. Hyper-V マネージャーを起動するには、[スタート] ボタンをクリックし、[管理ツール] をポイントして、[**Hyper-V** マネージャー] をクリックします。
2. ナビゲーションペインの [**Hyper-V Manage**r] で、Citrix ADC VPX インスタンスをインストールするサーバーを選択します。
3. [操作] メニューで、[仮想マシンのインポート] をクリックします。
4. ［仮想マシンのインポート］ダイアログボックスの［場所］で、Citrix ADC VPX インスタンスソフトウェアファイルを含むフォルダーのパスを指定し、［仮想マシンをコピーする（新しい一意の **ID** を作成する）］を選択します。このフォルダーは、Snapshots フォルダー、Virtual Hard Disks フォルダー、および Virtual Machines フォルダーを格納する親フォルダーです。

5. 注：圧縮ファイルを受け取った場合は、ファイルをフォルダーに抽出したことを確認してからフォルダーのパスを指定してください。
6. [インポート] をクリックします。
7. インポートした仮想アプライアンスが [仮想マシン] の下に表示されていることを確認します。
8. 別の仮想アプライアンスをインストールするには、手順 **2** ～**6** を繰り返します。

> 重要
>
> 手順 **4** で別のフォルダーにファイルを抽出することを確認します。

Hyper-V での **Citrix ADC VPX** インスタンスの自動プロビジョニング

Citrix ADC VPX インスタンスの自動プロビジョニングはオプションです。自動プロビジョニングを実行しない場合は、NetScaler 仮想アプライアンスによって IP アドレスなどを構成するためのオプションが提供されます。

Hyper-V で Citrix ADC VPX インスタンスを自動プロビジョニングするには、次の手順に従います。

1. 例に示されている説明に従い、XML ファイルを使用して ISO9660 準拠の ISO イメージを作成します。xml ファイルの名前が **userdata** であることを確認します。

```
<?xml version=”1.0” encoding=”UTF-8” standalone=”no”?>

<Environment xmlns:oe=`”http://schemas.dmtf.org/ovf/environment/1`
    ”

xmlns:xsi=`”http://www.w3.org/2001/XMLSchema-instance`”

oe:id=””

xmlns=`”http://schemas.dmtf.org/ovf/environment/1`”>

<PlatformSection>

<Kind>HYPER-V</Kind>

<Version>2013.1</Version>

<Vendor>CISCO</Vendor>

<Locale>en</Locale>

</PlatformSection>

<PropertySection>
```



```
24
25 <Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
       />
26
27 <Property oe:key="com.citrix.netscaler.platform" oe:value="NS1000V
       "/>
28
29 <Property oe:key="com.citrix.netscaler.orch_env" oe:value="cisco-
       orch-env"/>
30
31 <Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="
       10.102.100.122"/>
32
33 <Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
       255.255.255.128"/>
34
35 <Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="
       10.102.100.67"/></PropertySection>
36
37 </Environment>
38 <!--NeedCopy-->
```

2. ISO イメージを Hyper-V Server にコピーします。
3. インポートした仮想アプライアンスを選択し、[操作] メニューの [設定] を選択します。仮想アプライアンスを選択し、右クリックして [設定] を選択することもできます。選択した仮想アプライアンスの［設定］ウィンドウが表示されます。
4. [設定] ウィンドウの [ハードウェア] セクションで、[**IDE** コントローラ] をクリックします。
5. 右側のウィンドウで、[**DVD** ドライブ] を選択し、[追加] をクリックします。DVD ドライブは、左側のウィンドウ枠の [**IDE** コントローラ] セクションの下に追加されます。
6. 手順 5 で追加した **DVD** ドライブを選択します。右側のウィンドウウィンドウで、[イメージファイル] ラジオボタンを選択し、[参照] をクリックし、手順 2 で Hyper-V サーバーにコピーした ISO イメージを選択します。
7. [**Apply**］をクリックします。

> 注
>
> 仮想アプライアンスインスタンスは、次の場合にデフォルトの IP アドレスで起動されます。
>
> - DVD ドライブがアタッチされているのに、ISO ファイルが提供されていない。
> - ISO ファイルには、ユーザーデータファイルは含まれません。
> - ユーザデータファイル名または形式が正しくありません。

Citrix ADC VPX インスタンスで仮想 NIC を構成するには、次の手順に従います。

1. インポートした仮想アプライアンスを選択し、[操作] メニューの [設定] を選択します。

2. [****** の設定**] <virtual appliance name>** ダイアログボックスで、左側のウィンドウで **[**ハードウェアの追加 ******] をクリックします。
3. 右ペインで、デバイスのリストから［ネットワークアダプター］を選択します。
4. ［追加］をクリックします。
5. 左ペインに［**Network Adapter** (not connected)］が表示されていることを確認します。
6. 左ペインでネットワークアダプターを選択します。
7. 右側のペインで、[ネットワーク] メニューから、アダプタの接続先となる仮想ネットワークを選択します。
8. 使用する他のネットワークアダプタの仮想ネットワークを選択するには、手順 **6** と **7** を繰り返します。
9. [適用] をクリックし、[**OK**] をクリックします。

Citrix ADC VPX インスタンスを構成するには:

1. 前にインストールした仮想アプライアンスを右クリックし、［開始］を選択します。
2. 仮想アプライアンスをダブルクリックして、コンソールにアクセスします。
3. 仮想アプライアンスの Citrix ADC IP アドレス、サブネットマスク、Gateway を入力します。

仮想アプライアンスの基本構成が完了しました。Web ブラウザーで IP アドレスを入力して、仮想アプライアンスにアクセスします。

> 注
>
> 仮想マシン（VM）テンプレートを使用して、SCVMM を使用して Citrix ADC VPX インスタンスをプロビジョニングすることもできます。
>
> NetScaler VPX インスタンスで Microsoft Hyper-V NIC チーミングソリューションを使用する場合、詳細については、記事 CTX224494 を参照してください。

Linux-KVM プラットフォームへの Citrix ADC VPX インスタンスのインストール

October 7, 2021

Linux-KVM プラットフォーム用の Citrix ADC VPX を設定するには、グラフィカル仮想マシンマネージャ（仮想マネージャ）アプリケーションを使用できます。Linux-KVM コマンドラインを使用する場合は、`virsh` プログラムを使用できます。

KVM Module および QEMU のような仮想化ツールを使って、適切なハードウェアにホスト Linux オペレーティングシステムをインストールする必要があります。ハイパーバイザー上で展開できる仮想マシン（VM）の数はアプリケーション要件および選択されたハードウェアにより異なります。

Citrix ADC VPX インスタンスをプロビジョニングしたら、より多くのインターフェイスを追加できます。

制限事項と使用上のガイドライン

一般的な推奨事項

予測できない動作を回避するには、次の推奨事項を適用します。

- VPX 仮想マシンに関連付けられている VNet インターフェイスの MTU を変更しないでください。インターフェイスモードや CPU などの構成パラメータを変更する前に、VPX VM をシャットダウンします。
- VPX VM を強制的にシャットダウンしないでください。つまり、[強制オフ] コマンドは使用しないでください。
- ホスト Linux 上で指定された任意の構成は、Linux ディストリビューションの設定によってそのまま維持されたり、維持されなかったりします。これらの構成を維持するよう選択して、ホスト Linux オペレーティングシステムのリブートにおける一貫した動作を確保できます。
- Citrix ADC パッケージは、プロビジョニングされた Citrix ADC VPX インスタンスごとに一意である必要があります。

制限事項

- KVM 上で動作する VPX インスタンスのライブマイグレーションはサポートされていません。

Linux-KVM プラットフォームに Citrix ADC VPX インスタンスをインストールするための前提条件

October 7, 2021

Citrix ADC VPX インスタンスで実行されている Linux-KVM サーバーの最小システム要件を確認します。

CPU 要件:

- インテル VT-X プロセッサーに含まれるハードウェア仮想化機能を備えた 64 ビット x86 プロセッサー。

CPU が Linux ホストをサポートしているかどうかをテストするには、ホスト Linux シェルプロンプトで次のコマンドを入力します。

```
1 \*.egrep '^flags.\*(vmx|svm)' /proc/cpuinfo*
2 <!--NeedCopy-->
```

前の拡張機能の **BIOS** 設定が無効になっている場合は、BIOS でそれらを有効にする必要があります。

- ホスト Linux に 2 つ以上の CPU コアを指定します。
- プロセッサ速度に対する特定の推奨設定はありませんが、速度が速ければ速いほど VM アプリケーションのパフォーマンスはよくなります。

メモリ **(RAM)** 要件:

ホスト Linux カーネルに対して 4GB 以上。VM が必要とするメモリを追加します。

ハードディスク要件:

ホスト Linux カーネルおよび VM 要件の領域を計算します。単一の Citrix ADC VPX VM には、20 GB のディスク容量が必要です。

ソフトウェア要件

使用されるホストカーネルは、リリース 2.6.20 以降で、すべての仮想化ツールがある 64 ビットの Linux カーネルである必要があります。3.6.11-4 以降といったより新しいカーネルを推奨します。

Red Hat、CentOS、Fedora などの多くの Linux ディストリビューションでは、カーネルのバージョンと関連する仮想化ツールのテストが行われています。

ゲスト **VM** のハードウェア要件

Citrix ADC VPX は、IDE および virtIO ハードディスクタイプをサポートしています。ハードディスクの種類は、Citrix ADC パッケージの一部である XML ファイルで構成されています。

ネットワーク要件

Citrix ADC VPX は、virtIO 準仮想化、SR-IOV および PCI パススルーネットワークインターフェイスをサポートします。

サポートされるネットワークインターフェースの詳細については、以下を参照してください。

- 仮想マシンマネージャーを使用して Citrix ADC VPX インスタンスをプロビジョニングする
- SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する
- PCI パススルーネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する

ソースインターフェイスおよびモード

ソースデバイスの種類は、Bridge または MacVTap のいずれかにできます。MacvTap では、VEPA モード、ブリッジ、プライベート、パススルーの 4 つのモードが可能です。次のように、使用できるインターフェイスのタイプとサポートされているトラフィックタイプを確認します。

ブリッジ:

- Linux Bridge。
- 正しい設定を選択したり、`IPtable` サービスを無効にしたりしないと、ホスト Linux の`Ebtables`および`iptables`設定によってブリッジのトラフィックがフィルタリングされることがあります。

MacV タップ **(VEPA** モード**)**:

- パフォーマンスは Bridge より向上します。
- 同じ下位デバイスからのインターフェイスを VM 間で共有できます。
- 仮想マシン間通信（同じ
- 下位のデバイスは、アップストリームスイッチまたはダウンストリームスイッチが VEPA モードをサポートしている場合にのみ可能です。

MacVTap (プライベートモード**):**

- パフォーマンスは Bridge より向上します。
- 同じ下位デバイスからのインターフェイスを VM 間で共有できます。
- 同じ下位デバイスを使った内部 VM 通信を実行できません。

MacVTap (ブリッジモード**):**

- Bridge と比べてよい。
- 同じ下位デバイスのインターフェイスは、VM 間で共有できます。
- 下位のデバイスリンクがアップしている場合は、同じ下位デバイスを使用する VM 間通信が可能です。

MacVTap (パススルーモード**):**

- Bridge と比べてよい。
- 同じ下位デバイスのインターフェイスは、仮想マシン間で共有できません。
- 1 つの VM のみ、下位デバイスを使用できます。

> 注:VPX インスタンスで最適なパフォーマンスを得るには、ソースインターフェイスで`gro`および`lro`機能がオフになっていることを確認してください。

送信元インターフェイスのプロパティ

ソースインターフェイスの Generic-receive-offload（`gro`）および大規模受信オフロード（`lro`）機能をオフにします。`gro`および`lro`機能をオフにするには、ホスト Linux シェルプロンプトで次のコマンドを実行します。

```
ethtool -K eth6 gro off
ethool -K eth6 lro off
```

例:

```
[root@localhost ~]# ethtool -K eth6

                    Offload parameters for eth6:

                                        rx-checksumming: on

                                        tx-checksumming: on

                    scatter-gather: on
```



```
10
11                      tcp-segmentation-offload: on
12
13                      udp-fragmentation-offload: off
14
15                      generic-segmentation-offload: on
16
17                      generic-receive-offload: off
18
19                      large-receive-offload: off
20
21                      rx-vlan-offload: on
22
23                      tx-vlan-offload: on
24
25                      ntuple-filters: off
26
27                      receive-hashing: on
28
29      [root@localhost ~]#
30  <!--NeedCopy-->
```

例:

次の例のように、ホスト Linux ブリッジをソースデバイスとして使用する場合、ホストとゲスト VM を接続する仮想インターフェイスである VNet インターフェイスで`lro`機能をオフにする必要があります。

```
1       [root@localhost ~]# brctl show eth6_br
2
3       bridge name     bridge  id              STP enabled interfaces
4
5       eth6_br         8000.00e0ed1861ae         no         eth6
6
7                                                          vnet0
8
9                                                          vnet2
10
11      [root@localhost ~]#
12  <!--NeedCopy-->
```

上記の例では、2 つの仮想インターフェイスは eth6_br から派生し、vnet0 および vnet2 として表されます。次のコマンドを実行して、これらのインターフェイスの`gro`機能と`lro`機能をオフにします。


```
    ethtool -K vnet0 gro off
            ethtool -K vnet2 gro off
            ethtool -K vnet0 lro off
                    ethtool -K vnet2 lro off
<!--NeedCopy-->
```

無差別モード

次の機能を動作させるには、無差別モードを有効にする必要があります。

- L2 モード
- マルチキャストトラフィック処理
- ブロードキャスト
- IPV6 トラフィック
- 仮想 MAC
- 動的ルーティング

次のコマンドを使用して、無差別モードを有効にします。

```
[root@localhost ~]# ifconfig eth6 promisc
[root@localhost ~]# ifconfig eth6
eth6       Link encap:Ethernet  HWaddr 78:2b:cb:51:54:a3
           inet6 addr: fe80::7a2b:cbff:fe51:54a3/64 Scope:Link
           UP BROADCAST RUNNING PROMISC MULTICAST  MTU:9000  Metric:1
           RX packets:142961 errors:0 dropped:0 overruns:0 frame:0
           TX packets:2895843 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:14330008 (14.3 MB)  TX bytes:1019416071 (1.0 GB)

[root@localhost ~]#
<!--NeedCopy-->
```

必要なモジュール

ネットワークパフォーマンスを向上させるには、vhost_net モジュールが Linux ホストに存在することを確認します。vhost_net モジュールが存在することを確認するには、Linux ホストで次のコマンドを実行します。

```
lsmod | grep "vhost_net"
<!--NeedCopy-->
```


vhost_net がまだ実行されていない場合は、次のコマンドを入力して実行します。

```
modprobe vhost_net
<!--NeedCopy-->
```

OpenStack を使用して Citrix ADC VPX インスタンスをプロビジョニングする

October 7, 2021

OpenStack 環境で Citrix ADC VPX インスタンスをプロビジョニングするには、**Nova** ブートコマンド（OpenStack CLI）または Horizon（OpenStack ダッシュボード）を使用します。

VPX インスタンスをプロビジョニングします。オプションで、コンフィグドライブのデータを使用します。「構成ドライブ」とは、インスタンスの起動時に CD-ROM デバイスとしてアタッチされる特殊な構成ドライブを指します。この構成ドライブは、管理 IP アドレス、ネットワークマスク、デフォルトゲートウェイなど、ネットワーク構成を渡すためや、顧客スクリプトを注入するために使用できます。

Citrix ADC アプライアンスでは、デフォルトの認証メカニズムはパスワードベースです。現在、OpenStack 環境上の Citrix ADC VPX インスタンスでは、SSH キーペア認証メカニズムがサポートされています。

キーペア (公開鍵と秘密キー) は、公開鍵暗号化メカニズムを使用する前に生成されます。Horizon、Windows 用 Puttygen.exe、Linux 環境用`ssh-keygen` など、さまざまなメカニズムを使用して、キーペアを生成できます。キーペアの生成について詳しくは、それぞれの方式のオンラインドキュメントを参照してください。

キーペアが利用可能になったら、権限のあるユーザーがアクセスできる安全な場所に秘密鍵をコピーします。OpenStack では、Horizon または Nova ブートコマンドを使用して、VPX インスタンスにパブリックキーをデプロイできます。OpenStack を使用して VPX インスタンスをプロビジョニングすると、まず特定の BIOS 文字列を読み取って、インスタンスが OpenStack 環境で起動していることを検出します。この文字列は「OpenStack Foundation」であり、Red Hat Linux ディストリビューションの場合は、/etc/nova/release に保存されます。これは、KVM ハイパーバイザープラットフォームに基づくすべての OpenStack 実装で利用できる標準的なメカニズムです。ドライブには特定の OpenStack ラベルが必要です。

ネットワ ― ク構成、カスタムスクリプト、および SSH キーペアが提供されている場合は、構成ドライブが検出されると、インスタンスがそれらを読み取ろうとします。

ユーザーデータファイル

Citrix ADC VPX インスタンスは、ユーザーデータファイルとも呼ばれるカスタマイズされた OVF ファイルを使用して、ネットワーク構成、カスタムスクリプトを注入します。このファイルは、構成ドライブの一部として提供されます。次に、カスタマイズされた OVF ファイルの例を示します。


```
```
<?xml version=”1.0” encoding=”UTF-8” standalone=”no”?>
<Environment xmlns:oe=”http://schemas.dmtf.org/ovf/environment/1”
xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
oe:id=””
xmlns=”http://schemas.dmtf.org/ovf/environment/1”
xmlns:cs=”http://schemas.citrix.com/openstack”>
<PlatformSection>
<Kind></Kind>
<Version>2016.1</Version>
<Vendor>VPX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key=”com.citrix.netscaler.ovf.version” oe:value=”1.0”/>
<Property oe:key=”com.citrix.netscaler.platform” oe:value=”NSVPX”/>
<Property oe:key=”com.citrix.netscaler.orch_env” oe:value=”openstack-
   orch-env”/>
<Property oe:key=”com.citrix.netscaler.mgmt.ip” oe:value=”10.1.2.22”/>
<Property oe:key=”com.citrix.netscaler.mgmt.netmask” oe:value=”
   255.255.255.0”/>
<Property oe:key=”com.citrix.netscaler.mgmt.gateway” oe:value=”10.1.2.1
   ”/>
</PropertySection>
 <cs:ScriptSection>
   <cs:Version>1.0</cs:Version>
     <ScriptSettingSection xmlns=”http://schemas.citrix.com/openstack”
        xmlns:i=”http://www.w3.org/2001/XMLSchema-instance”>
         <Scripts>
           <Script>
                 <Type>shell</Type>
                 <Parameter>X Y</Parameter>
                <Parameter>Z</Parameter>
                <BootScript>before</BootScript>
                  <Text>
                        #!/bin/bash
                        echo ”Hi, how are you” $1 $2 >> /var/sample.txt
                  </Text>
           </Script>
           <Script>
                 <Type>python</Type>
                 <BootScript>after</BootScript>
                  <Text>
                      #!/bin/python
```

 
```

```
print(”Hello”);
                    </Text>
            </Script>
   <Script>
                  <Type>perl</Type>
                  <BootScript>before</BootScript>
                   <Text>
                        !/usr/bin/perl
my $name = ”VPX”;
print ”Hello, World $name !\n” ;
                    </Text>
            </Script>
            <Script>
                 <Type>nscli</Type>
                 <BootScript>after</BootScript>
                 <Text>
                  add vlan 33
bind vlan 33 -ifnum 1/2
                 </Text>
            </Script>
          </Scripts>
      </ScriptSettingSection>
 </cs:ScriptSection>
</Environment>
<!--NeedCopy--> ```
```

前の OVF ファイルでは、「PropertySection」は NetScaler ネットワーク構成に使用され、<cs:ScriptSection> はすべてのスクリプトを囲むために使用されます。<Scripts></Scripts> タグは、すべてのスクリプトをまとめるのに使われます。各スクリプトは <Script> </Script> タグの間に定義されています。各スクリプトタグには、従属するフィールドやタグがあります。

a) <Type>：スクリプトタイプの値を指定します。指定可能な値：Shell/Perl/Python/NSLCI（NetScaler CLI スクリプトの場合）

b) <Parameter>：スクリプトにパラメーターを指定します。各スクリプトでは、複数の <Parameter> タグを使用できます。

c) <BootScript>：スクリプト実行ポイントを指定します。このタグに指定できる値: 前/後。「before」は、PE がアップする前にスクリプトを実行することを指定します。「after」は、PE が起動した後にスクリプトが実行されることを指定します。

d) <Text>：スクリプトの内容を貼り付けます。

> 注
>
> 現在、VPX インスタンスはスクリプトのサニタイズを処理しません。管理者は、スクリプトの有効性を確認す

> る必要があります。
>
> すべてのセクションを表示する必要はありません。空の「PropertySection」を使用して最初のブート時に実行するスクリプトのみを定義するか、ネットワーク構成のみを定義するには空の「PropertySection」を使用します。
>
> OVF ファイル（ユーザーデータファイル）の必要なセクションが入力されたら、そのファイルを使用して VPX インスタンスをプロビジョニングします。

ネットワーク構成

ネットワーク構成の一部として、VPX インスタンスは以下を読み込みます。

- 管理 IP アドレス
- ネットワークマスク
- デフォルトゲートウェイ

パラメーターは、正常に読み取られると、インスタンスをリモートで管理できるように NetScaler 構成に移入されます。パラメーターが読み取られない場合、または構成ドライブが存在しない場合は、インスタンスが以下のデフォルトの処理を実行します。

- DHCP から IP アドレス情報を取得する。
- DHCP で障害が発生するか、タイムアウトした場合、インスタンスはデフォルトのネットワーク設定（192.168.100.1/16）で起動します。

カスタマースクリプト

VPX インスタンスでは、初期プロビジョニング中にカスタムスクリプトを実行できます。アプライアンスは、シェル、Perl、Python、および Citrix ADC CLI コマンドタイプのスクリプトをサポートしています。

SSH キーペア認証

VPX インスタンスは、インスタンスメタデータの一部として構成ドライブ内で利用可能なパブリックキーをその「authorized_keys」ファイルにコピーします。これにより、ユーザーが秘密キーを使用してインスタンスにアクセスできるようになります。

> 注
>
> SSH キーが提供されると、デフォルトの認証情報（nsroot/nsroot）は機能しなくなります。パスワードベースのアクセスが必要な場合は、それぞれの SSH プライベートキーでログオンし、手動でパスワードを設定します。

はじめに

OpenStack 環境で VPX インスタンスをプロビジョニングする前に、.tgz ファイルから.qcow2ファイルを抽出してビルドします。

qcow2 イメージからの OpenStack イメージ。次の手順を実行します:

1. 次のコマンドを入力して、.tqzファイルから.qcow2ファイルを抽出します。

```
tar xvzf <TAR file>
tar xvzf <NSVPX-KVM-12.0-26.2_nc.tgz>
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```

2. 次のコマンドを入力して、手順 1 で抽出した.qcoz2ファイルを使用して OpenStack イメージをビルドします。

```
openstack image create --container-format bare --property
   hw_disk_bus=ide --disk-format qcow2 --file <path to qcow2 file>
    --public <name of the OpenStack image>

glance image-create --name=”NS-VPX-12-0-26-2” --property
   hw_disk_bus=ide --ispublic=
true --container-format=bare --disk-format=qcow2< NSVPX-KVM
   -12.0-26.2_nc.qcow2
```

図 **1**: 次の図に、glance image-create コマンドの出力例を示します。

```
+------------------+------------------------------------------------------+
| Field            | Value                                                |
+------------------+------------------------------------------------------+
| checksum         | 154ade3fc7dca7d1706b1d03d7d97552                     |
| container_format | bare                                                 |
| created_at       | 2017-03-13T08:52:31Z                                 |
| disk_format      | qcow2                                                |
| file             | /v2/images/322c1e0f-cce8-4b7b-b53e-bd8152c388ed/file |
| id               | 322c1e0f-cce8-4b7b-b53e-bd8152c388ed                 |
| min_disk         | 0                                                    |
| min_ram          | 0                                                    |
| name             | VPX-KVM-12.0-26.2
| owner            | 58d17d81df5d4406afbb4fdab3a58d79                     |
| properties       | hw_disk_bus='ide'                                    |
| protected        | False                                                |
| schema           | /v2/schemas/image                                    |
| size             | 784338944                                            |
| status           | active                                               |
| updated_at       | 2017-03-13T08:52:43Z                                 |
| virtual_size     | None                                                 |
| visibility       | public                                               |
+------------------+------------------------------------------------------+
```


VPX インスタンスのプロビジョニング

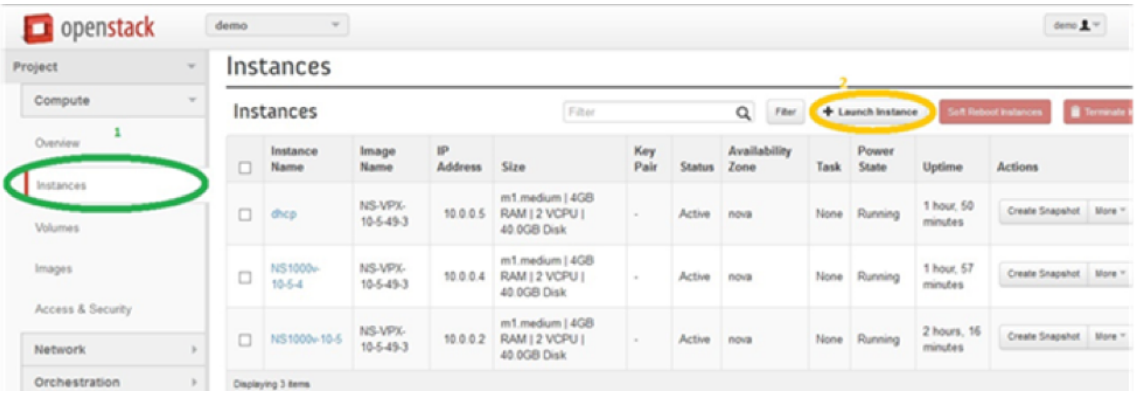
VPX インスタンスをプロビジョニングするには、次のいずれかの方法を使用します。

- Horizon（OpenStack ダッシュボード）
- Nova boot コマンド（OpenStack CLI）

OpenStack ダッシュボードを使用して VPX インスタンスをプロビジョニングする

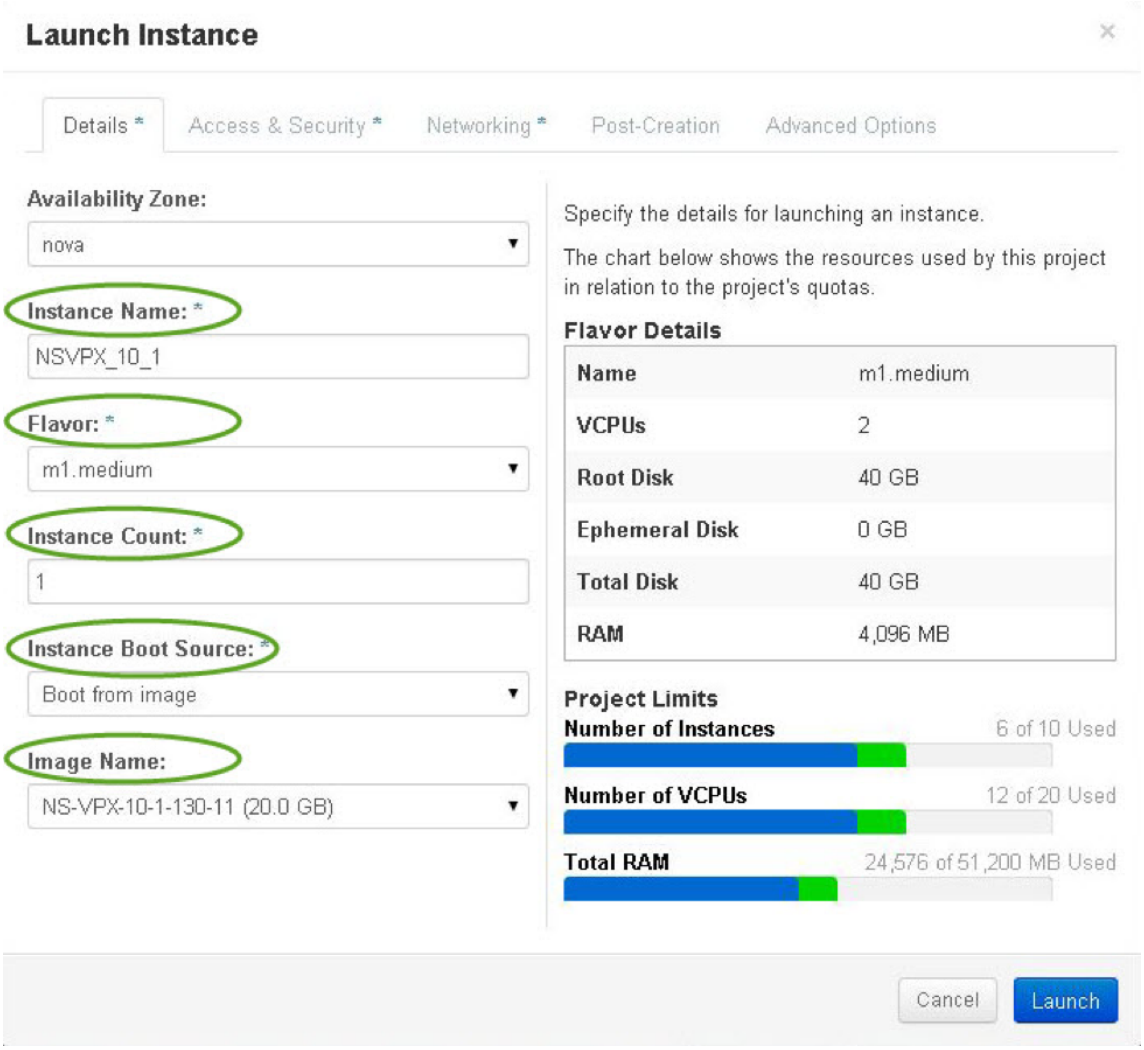
Horizon を使用して VPX インスタンスをプロビジョニングするには、次の手順に従います。

1. OpenStack ダッシュボードにログオンします。
2. ダッシュボードの左側にある [プロジェクト] パネルで、[インスタンス] を選択します。
3. [インスタンス] パネルで、[インスタンスの起動] をクリックして、[インスタンスの起動] ウィザードを開きます。



4. ［Launch Instance］ウィザードで、以下の情報を指定します。

   a) Instance Name - インスタンス名
   b) Flavor - インスタンスのフレーバー（種類）
   c) Instance Count - インスタンスの数
   d) Instance Boot Source - インスタンスの起動ソース
   e) イメージ名



5. 次の手順を実行して、Horizon を介して新しいキーペアか既存のキーペアを展開します。

   a) 既存のキーペアがない場合は、既存の方式を使用してキーを作成します。既存のキーがある場合は、この手順はスキップします。

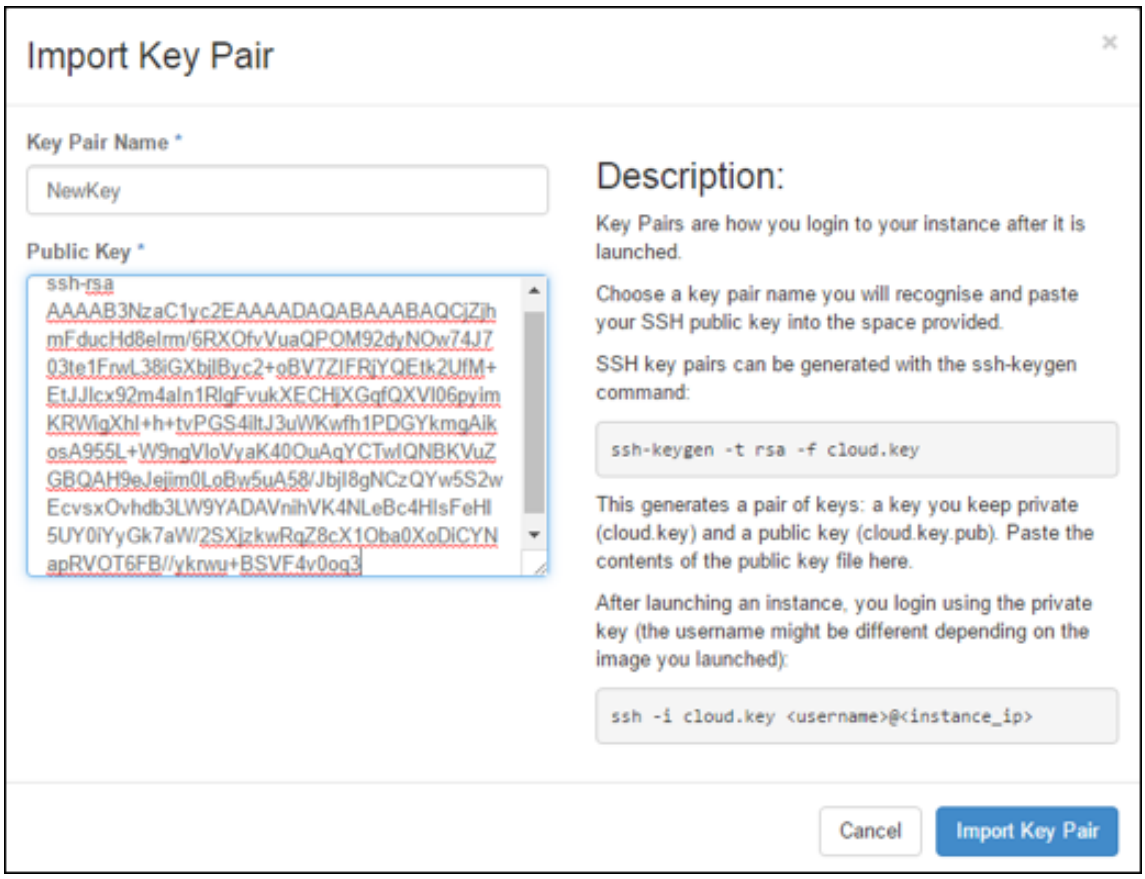
   b) 公開キーの内容をコピーします。

   c) [**Horizon] > [**インスタンス**] > [**新しいインスタンスの作成**]** の順に選択します。

   d) [アクセスとセキュリティ] をクリックします。

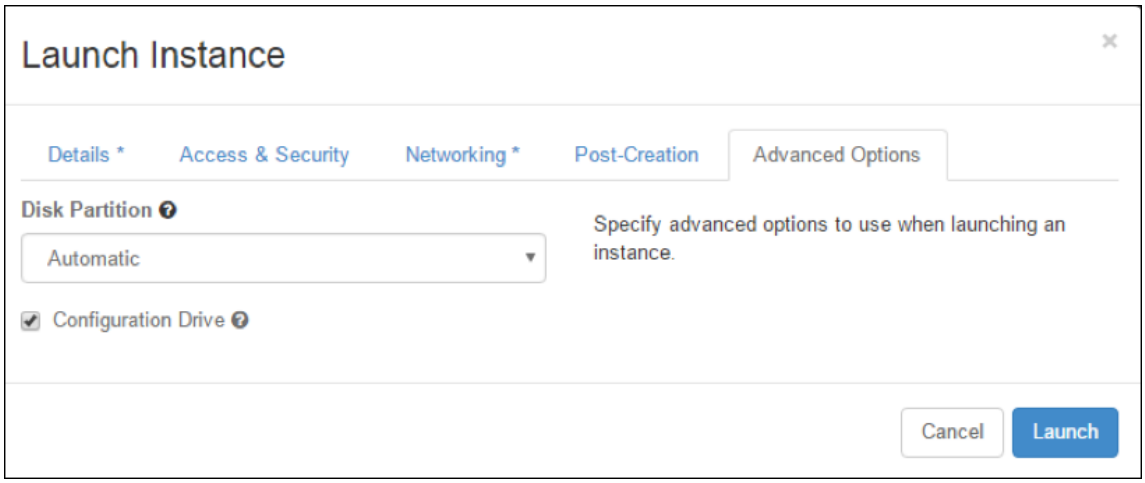
   e) [**Key Pair**] ドロップダウンメニューの隣にある [+] 記号をクリックし、表示されるパラメータの値を入力します。

   f) 公開鍵の内容を 公開鍵ボックスに貼り付け、鍵に名前を付け、[鍵 ペアのインポート] をクリックします。



6. ウィザードの [ポスト作成] タブをクリックします。[カスタマイズスクリプト] で、ユーザーデータファイルのコンテンツを追加します。ユーザーデータファイルには、VPX インスタンスの IP アドレス、ネットマスクとゲートウェイの詳細、およびカスタマースクリプトが含まれます。

7. キーペアを選択またはインポートした後、config-drive オプションをチェックし、**Launch** をクリックします。



OpenStack CLI を使用して **VPX** インスタンスをプロビジョニングする

OpenStack CLI を使用して VPX インスタンスをプロビジョニングするには、次の手順に従います。

1. qcow2 からイメージを作成するには、次のコマンドを入力します。

   ```
   openstack image create --container-format bare --property hw_disk_bus=
   ide --diskformat qcow2 --file NSVPX-OpenStack.qcow2 --public VPX-ToT-
   Image
   ```

2. インスタンスを作成するイメージを選択するには、次のコマンドを入力します。

   ```
   openstack image list | more
   ```

3. 特定のフレーバーのインスタンスを作成するには、次のコマンドを入力して、リストからフレーバー ID/名前を選択します。

   ```
   openstack flavor list
   ```

4. NIC を特定のネットワークに接続するには、次のコマンドを入力して、ネットワークリストからネットワーク ID を選択します。

   ```
   openstack network list
   ```

5. インスタンスを作成するには、次のコマンドを入力します。

```
1 openstack server create --flavor FLAVOR_ID --image IMAGE_ID --key-
     name KEY_NAME
2 --user-data USER_DATA_FILE_PATH --config-drive True --nic net-id=
     net-uuid
3 INSTANCE_NAME
4 openstack server create --image VPX-ToT-Image --flavor m1.medium
     --user-data
5 ovf.xml --config-drive True --nic net-id=2734911b-ee2b-48d0-a1b6-3
     efd44b761b9
6 VPX-ToT
```

図 **2**: 次の図は、出力例を示しています。


```
+--------------------------------------+--------------------------------------------------------------------------+
| Field                                | Value                                                                    |
+--------------------------------------+--------------------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                                   |
| OS-EXT-AZ:availability_zone          |                                                                          |
| OS-EXT-SRV-ATTR:host                 | None                                                                     |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | None                                                                     |
| OS-EXT-SRV-ATTR:instance_name        | instance-000001c2                                                        |
| OS-EXT-STS:power_state               | 0                                                                        |
| OS-EXT-STS:task_state                | scheduling                                                               |
| OS-EXT-STS:vm_state                  | building                                                                 |
| OS-SRV-USG:launched_at               | None                                                                     |
| OS-SRV-USG:terminated_at             | None                                                                     |
| accessIPv4                           |                                                                          |
| accessIPv6                           |                                                                          |
| addresses                            |                                                                          |
| adminPass                            | pFVvMtq7N8Z6                                                             |
| config_drive                         | True                                                                     |
| created                              | 2017-03-13T10:32:59Z                                                     |
| flavor                               | m1.medium (3)                                                            |
| hostId                               |                                                                          |
| id                                   | a1fe991e-3604-43a0-9dd6-59fa0f3749df                                     |
| image                                | VPX-ToT-Image (f0c2f9d1-08f2-4b2e-9943-2ee6bc2edbc7)                     |
| key_name                             | None                                                                     |
| name                                 |VPX-ToT                                                                   |
| os-extended-volumes:volumes_attached | []                                                                       |
| progress                             | 0                                                                        |
| project_id                           | 58d17d81df5d4406afbb4fdab3a58d79                                         |
| properties                           |                                                                          |
| security_groups                      | [{u'name': u'default'}]                                                  |
| status                               | BUILD                                                                    |
| updated                              | 2017-03-13T10:33:00Z                                                     |
| user_id                              | a6347b33916b4eb1b1f76360a9c8f935                                         |
+--------------------------------------+--------------------------------------------------------------------------+
```

仮想マシンマネージャーを使用して **Citrix ADC VPX** インスタンスをプロビジョニングする

October 7, 2021

Virtual Machine Manager は、VM ゲストを管理するためのデスクトップツールです。これによって新しい VM ゲストおよびさまざまな種類のストレージを作成し、仮想ネットワークを管理できます。組み込み VNC ビューアーにより VM ゲストのグラフィカルコンソールにアクセスして、ローカルまたはリモートでパフォーマンス統計を閲覧できます。

優先 Linux ディストリビューションをインストールした後、KVM 仮想化を有効にして、仮想マシンのプロビジョニングを処理できます。

仮想マシンマネージャーを使用して Citrix ADC VPX インスタンスをプロビジョニングする場合、次の 2 つのオプションがあります。

- 手動で IP アドレス、ゲートウェイ、およびネットマスクを入力する
- IP アドレス、Gateway、ネットマスクを自動的に割り当てる（自動プロビジョニング）

Citrix ADC VPX インスタンスをプロビジョニングするには、2 種類のイメージを使用できます。

- RAW
- QCOW2

Citrix ADC VPX RAW イメージを QCOW2 イメージに変換し、Citrix ADC VPX インスタンスをプロビジョニングできます。RAW イメージを QCOW2 イメージに変換するには、次のコマンドを入力します。

```
qemu-img convert -O qcow2 original-image.raw image-converted.qcow
```

次に例を示します:

```
qemu-img convert -O qcow2 NSVPX-KVM-11.1-12.5_nc.raw NSVPX-KVM-11.1-12.5_nc
.qcow
```

KVM での一般的な Citrix ADC VPX 展開には、次の手順が含まれます。

- Citrix ADC VPX インスタンスの自動プロビジョニングの前提条件の確認
- RAW イメージを使用した Citrix ADC VPX インスタンスのプロビジョニング
- QCOW2 イメージを使用した Citrix ADC VPX インスタンスのプロビジョニング
- Virtual Machine Manager を使用した VPX インスタンスへのインターフェイスの追加

Citrix ADC VPX インスタンスの自動プロビジョニングの前提条件の確認

自動プロビジョニングはオプション機能であり、CDROM ドライブからのデータの使用を伴います。この機能を有効にすると、初期セットアップ時に Citrix ADC VPX インスタンスの管理 IP アドレス、ネットワークマスク、デフォルト Gateway を入力する必要はありません。

VPX インスタンスを自動プロビジョニングする前に、次のタスクを完了する必要があります。

1. カスタマイズされたオープン仮想化形式 (OVF) XML ファイルまたはユーザーデータファイルを作成します。
2. オンラインアプリケーション（たとえば、PowerISO）を使用して、OVF ファイルを ISO イメージに変換します。
3. セキュアコピー (SCP) ベースのツールを使用して、ISO イメージを KVM ホストにマウントします。

サンプル **OVF XML** ファイル:

次に、OVF XML ファイルの内容の例を示します。このファイルをサンプルとして使用して、ファイルを作成することができます。

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2
3  <Environment xmlns:oe="`http://schemas.dmtf.org/ovf/environment/1"`
4
5  xmlns:xsi="`http://www.w3.org/2001/XMLSchema-instance"`
6
7  oe:id=""
8
9  xmlns="`http://schemas.dmtf.org/ovf/environment/1"`
10
11 xmlns:cs="`http://schemas.citrix.com/openstack">`
```



```
12
13 <PlatformSection>
14
15 <Kind></Kind>
16
17 <Version>2016.1</Version>
18
19 <Vendor>VPX</Vendor>
20
21 <Locale>en</Locale>
22
23 </PlatformSection>
24
25 <PropertySection>
26
27 <Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
28
29 <Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
30
31 <Property oe:key="com.citrix.netscaler.orch_env" oe:value="KVM"/>
32
33 <Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
34
35 <Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
      255.255.255.0"/>
36
37 <Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1
      "/>
38
39 </PropertySection>
40
41 </Environment>
42 <!--NeedCopy-->
```

前述の OVF XML ファイルでは、NetScaler ネットワーク構成に「PropertySection」が使用されています。ファイルを作成するときには、この例の最後で強調表示されている、パラメーターの値を指定します。

- 管理 IP アドレス
- ネットマスク
- Gateway

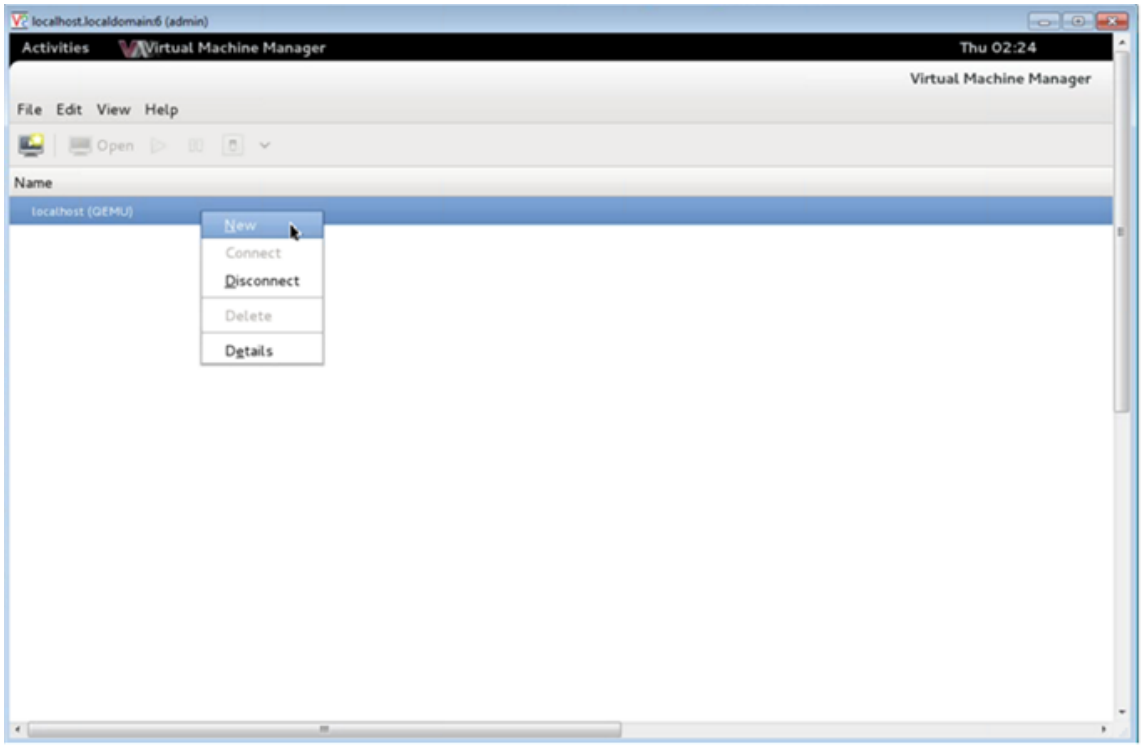
> 重要
>
> OVF ファイルが適切な XML 形式でない場合、VPX インスタンスには、ファイルに指定された値ではなく、デフォルトのネットワーク設定が割り当てられます。

RAW イメージを使用して **Citrix ADC VPX** インスタンスをプロビジョニングする

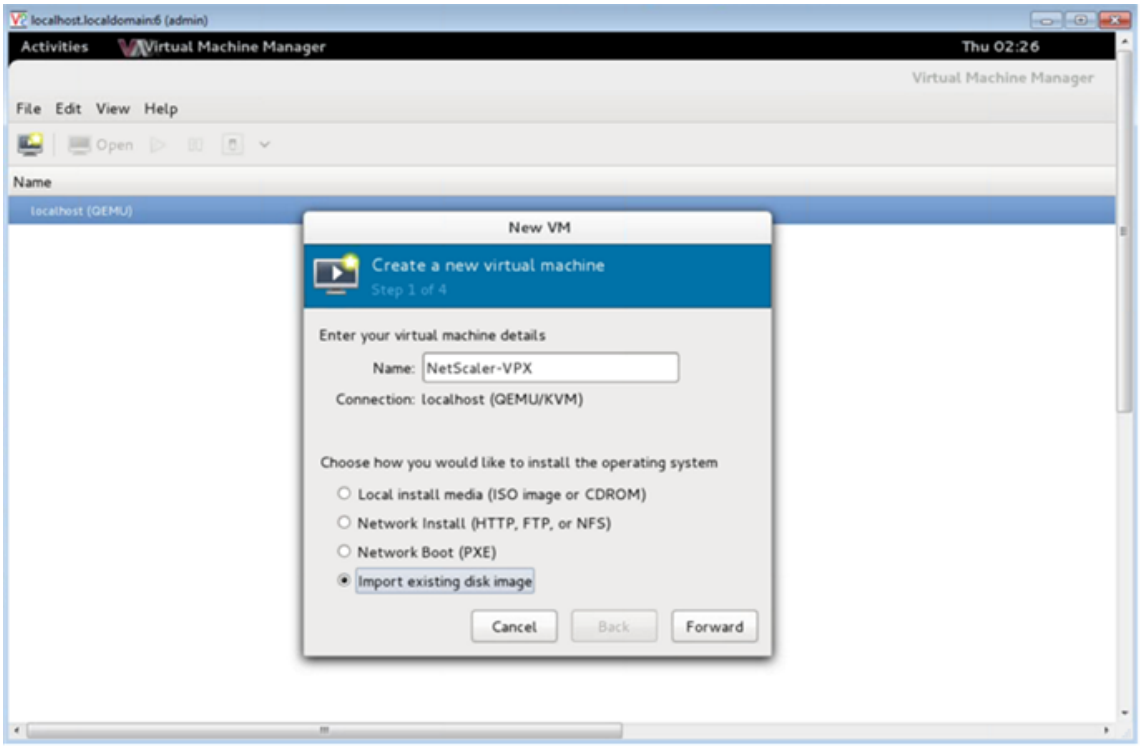
仮想マシンマネージャーでは、RAW イメージを使用して Citrix ADC VPX インスタンスをプロビジョニングできます。

仮想マシンマネージャーを使用して Citrix ADC VPX インスタンスをプロビジョニングするには、次の手順に従います。

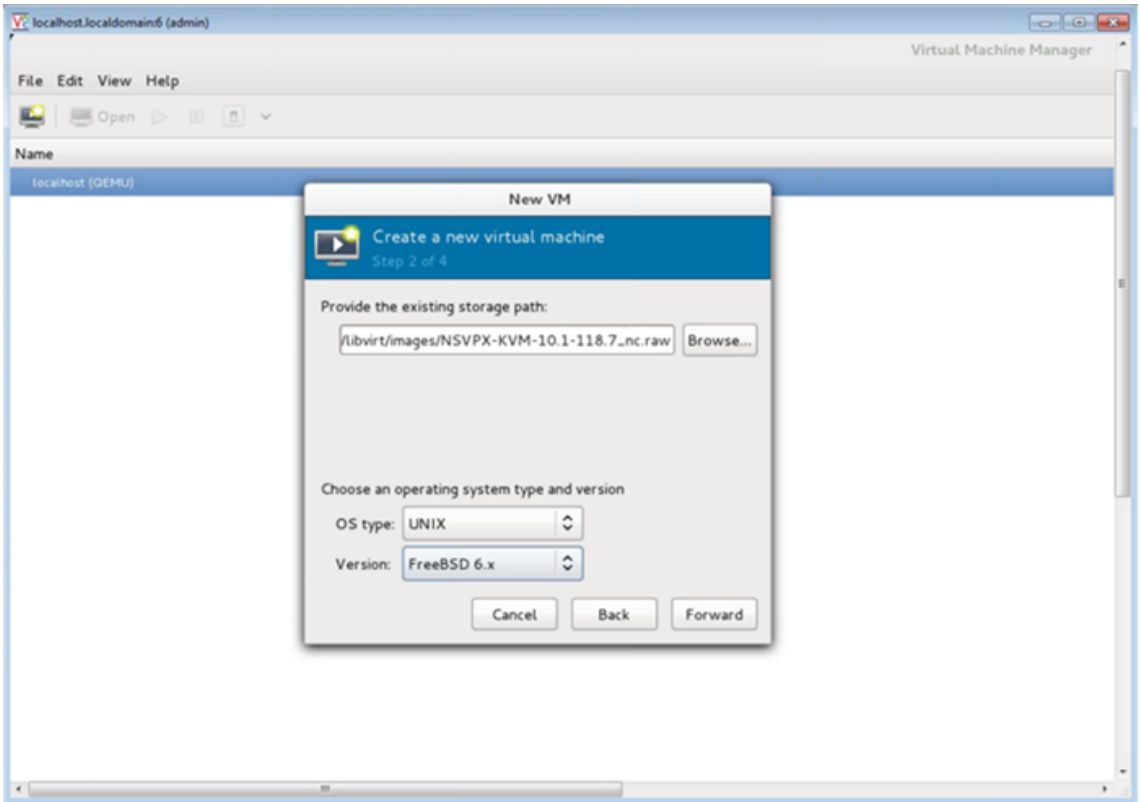
1. 仮想マシンマネージャー (アプリケーション > システムツール > バーチャルマシンマネージャー) を開き、[認証] ウィンドウにログオン資格情報を入力します。
2.  、ローカルホスト（**QEMU**）を右クリックして、新しい Citrix ADC VPX インスタンスを作成します。



3. ［名前］テキストボックスに、新しい仮想マシンの名前（NetScaler-VPX など）を入力します。
4. [新しい仮想マシン] ウィンドウの [オペレーティングシステムのインストール方法を選択する] で、[既存のディスクイメージのインポート] を選択し、[転送] をクリックします。

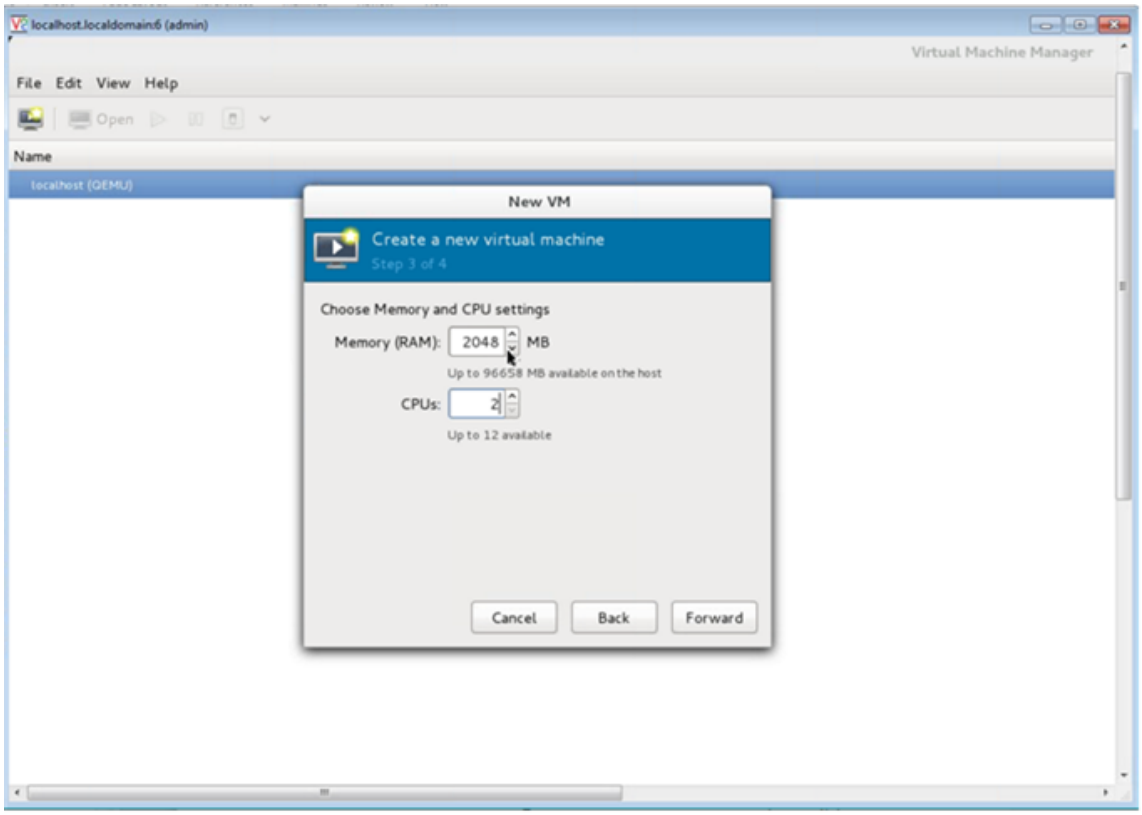


5. [既存のストレージパスを指定する] フィールドで、イメージへのパスをナビゲートします。オペレーティングシステム種類に UNIX、バージョンとして FreeBSD 6.x を選択します。次に、[進む] をクリックします。

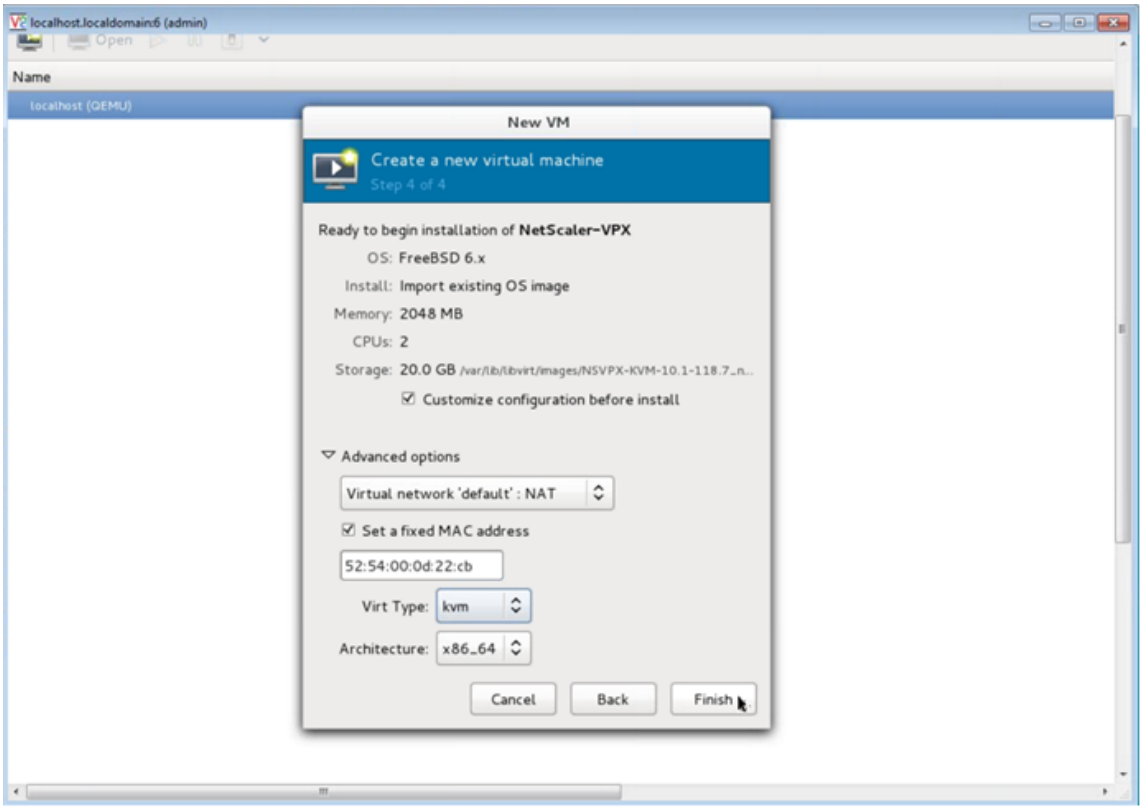


6. [メモリと **CPU** の設定の選択] で、次の設定を選択し、[転送] をクリックします。

   - メモリ (RAM) – 2048MB
   - CPU – 2

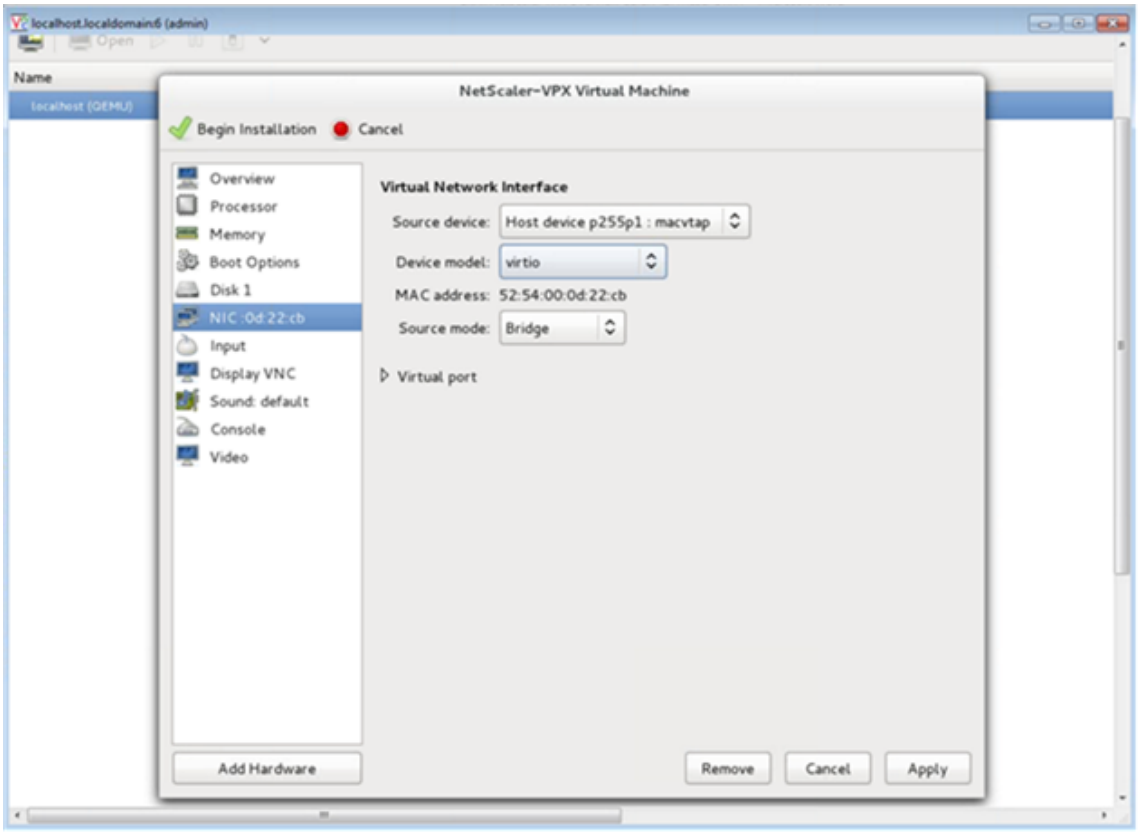


7. [インストール前に構成をカスタマイズする] チェックボックスをオンにします。オプションで、[詳細オプション] で MAC アドレスをカスタマイズできます。選択した **Virt** タイプが KVM で、選択されたアーキテクチャが x86_64 であることを確認します。[完了] をクリックします。



8. NIC を選択し、次の構成を指定します。

   - ソースデバイス: `ethX macvtap` またはブリッジ
   - デバイスモデル— `virtio`
   - ソースモード - Bridge



9. [**Apply**] をクリックします。
10. VPX インスタンスの自動プロビジョニングを行う場合は、このドキュメントの「**CDROM** ドライブを接続して自動プロビジョニングを有効にする」セクションを参照してください。それ以外の場合は、[インストレーションを開始] をクリックします。KVM で Citrix ADC VPX をプロビジョニングしたら、インターフェイスを追加できます。

QCOW2 イメージを使用して **Citrix ADC VPX** インスタンスをプロビジョニングする

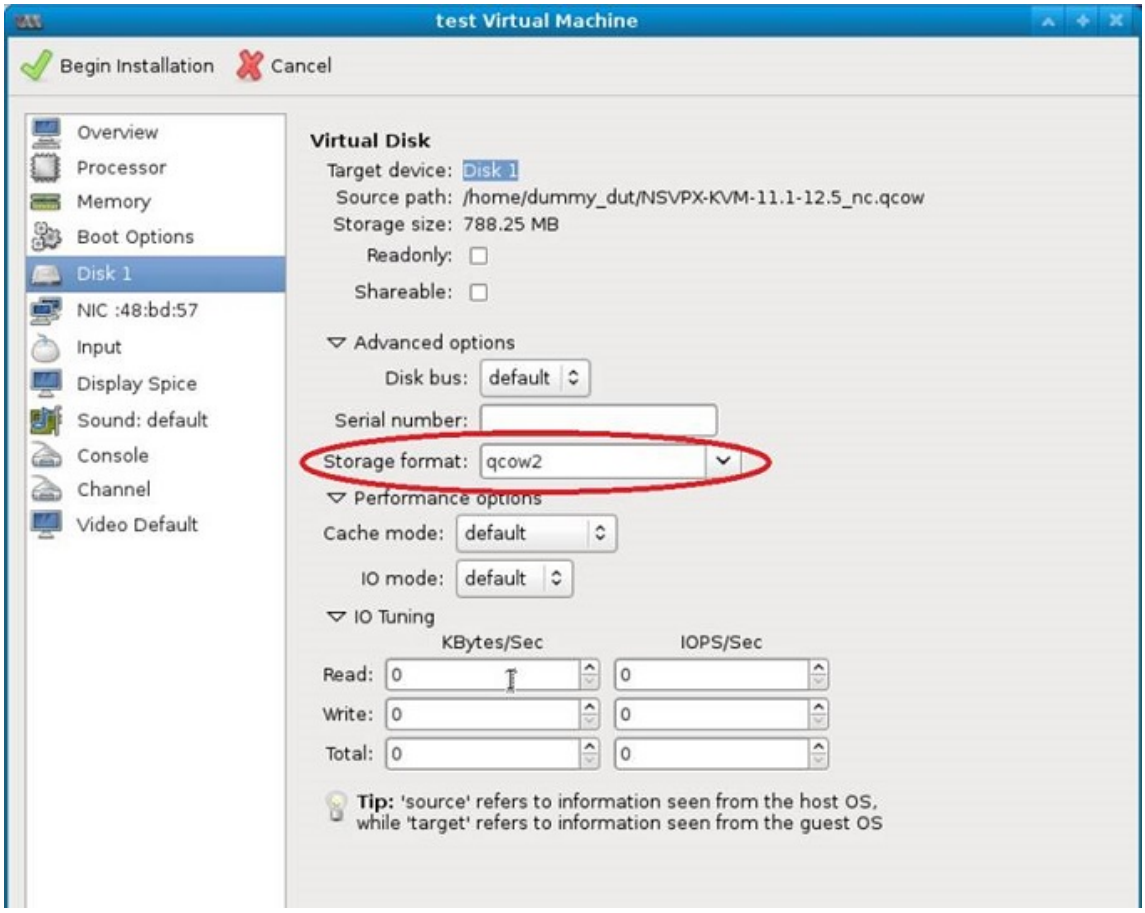
仮想マシンマネージャーを使用して、QCOW2 イメージを使用して Citrix ADC VPX インスタンスをプロビジョニングできます。

QCOW2 イメージを使用して Citrix ADC VPX インスタンスをプロビジョニングするには、次の手順に従います。

1. RAW イメージを使用した Citrix ADC ****VPX** インスタンスのプロビジョニングの手順 **1**～ステップ **8** に従います **。

   注：ステップ **5** で **qcow2** イメージを選択することを確認してください。

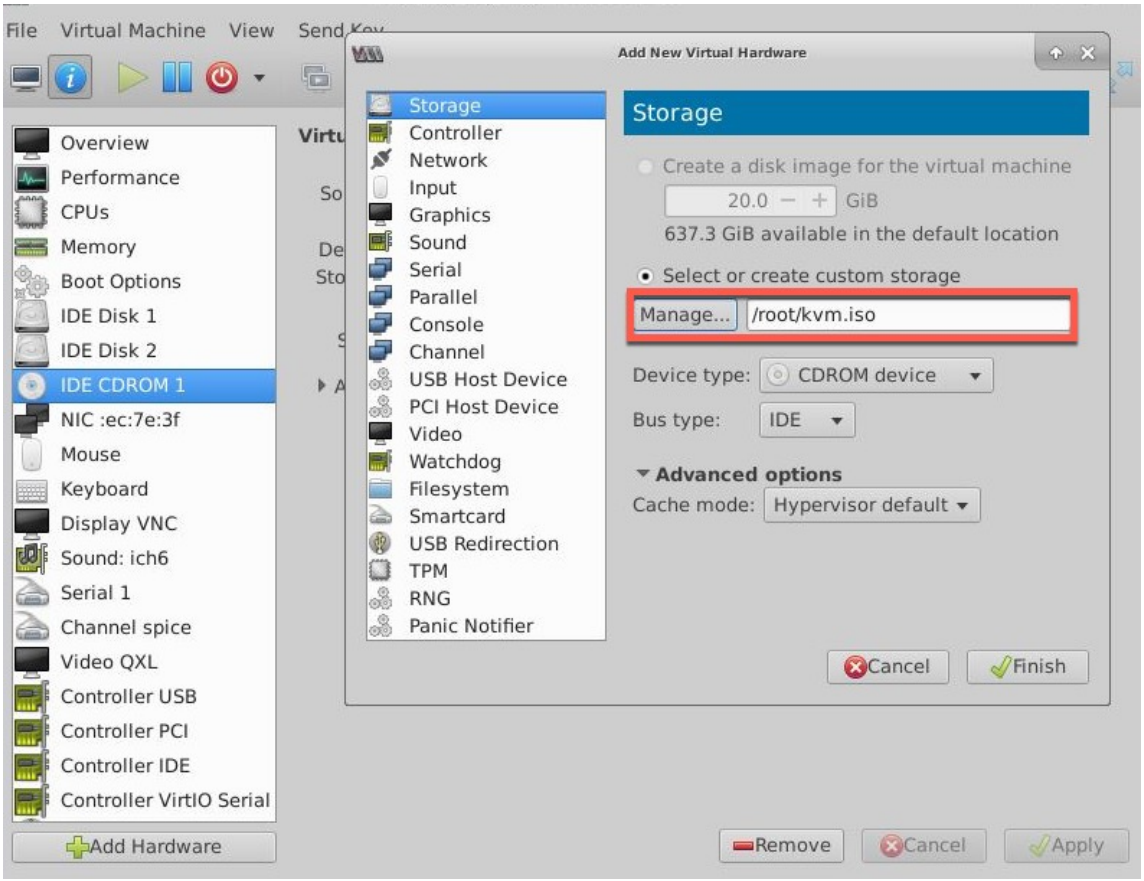
2. [ディスク **1**] を選択し、[詳細オプション] をクリックします。
3. [ストレージ形式] ドロップダウンリストから [**qcow2**] を選択します。



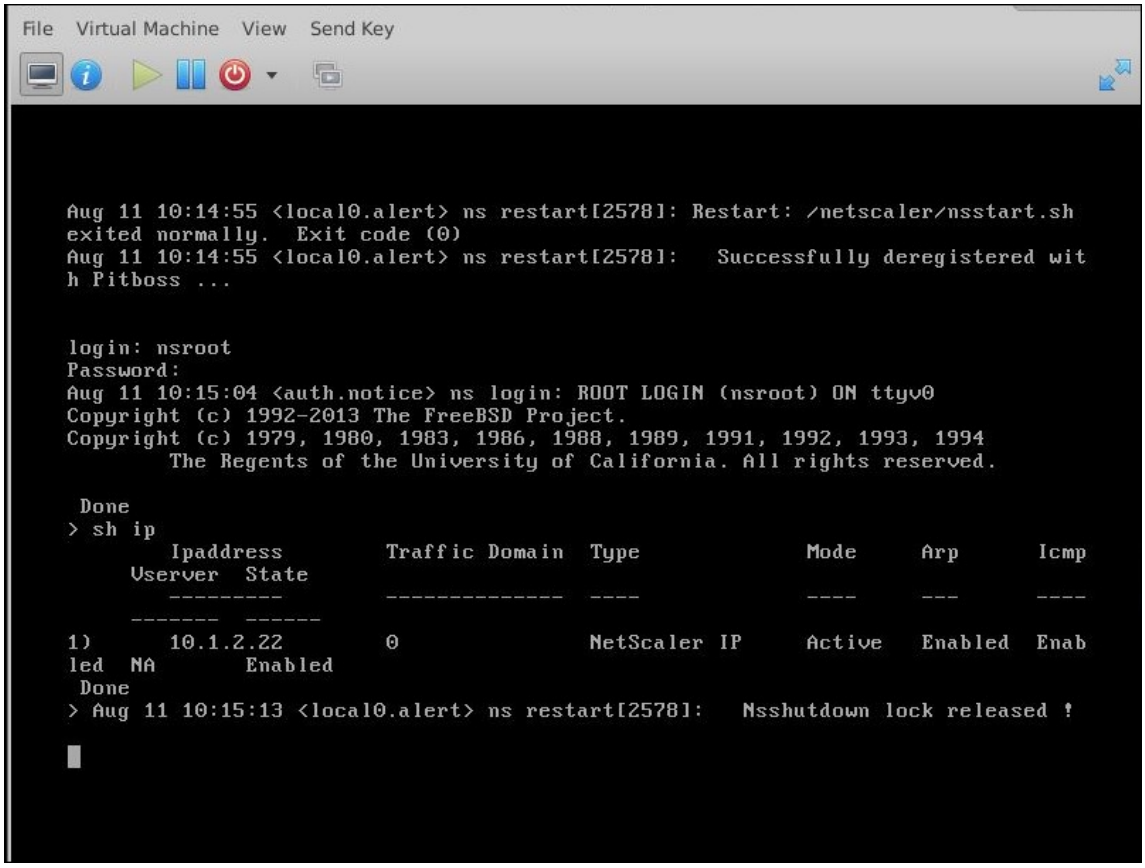
4. [**Apply**] をクリックし、次に [**Begin Installation**] をクリックします。KVM で Citrix ADC VPX をプロビジョニングしたら、インターフェイスを追加できます。

CD-ROM ドライブを接続して自動プロビジョニングを有効にする

1. [ハードウェアの追加] **>** [記憶域] **>** [デバイスの種類] **>** [**CD-ROM** デバイス] をクリックします。
2. [管理] をクリックし、[Citrix ADC VPX インスタンスの自動プロビジョニングの前提条件] セクションでマウントした正しい ISO ファイルを選択し、[完了] をクリックします。Citrix ADC VPX インスタンスのリソースの下に新しい CDROM が作成されます。



3. VPX インスタンスの電源をオンにすると、スクリーンショットの例で示すように、OVF ファイルで提供されているネットワーク構成を使用して自動プロビジョニングが行われます。



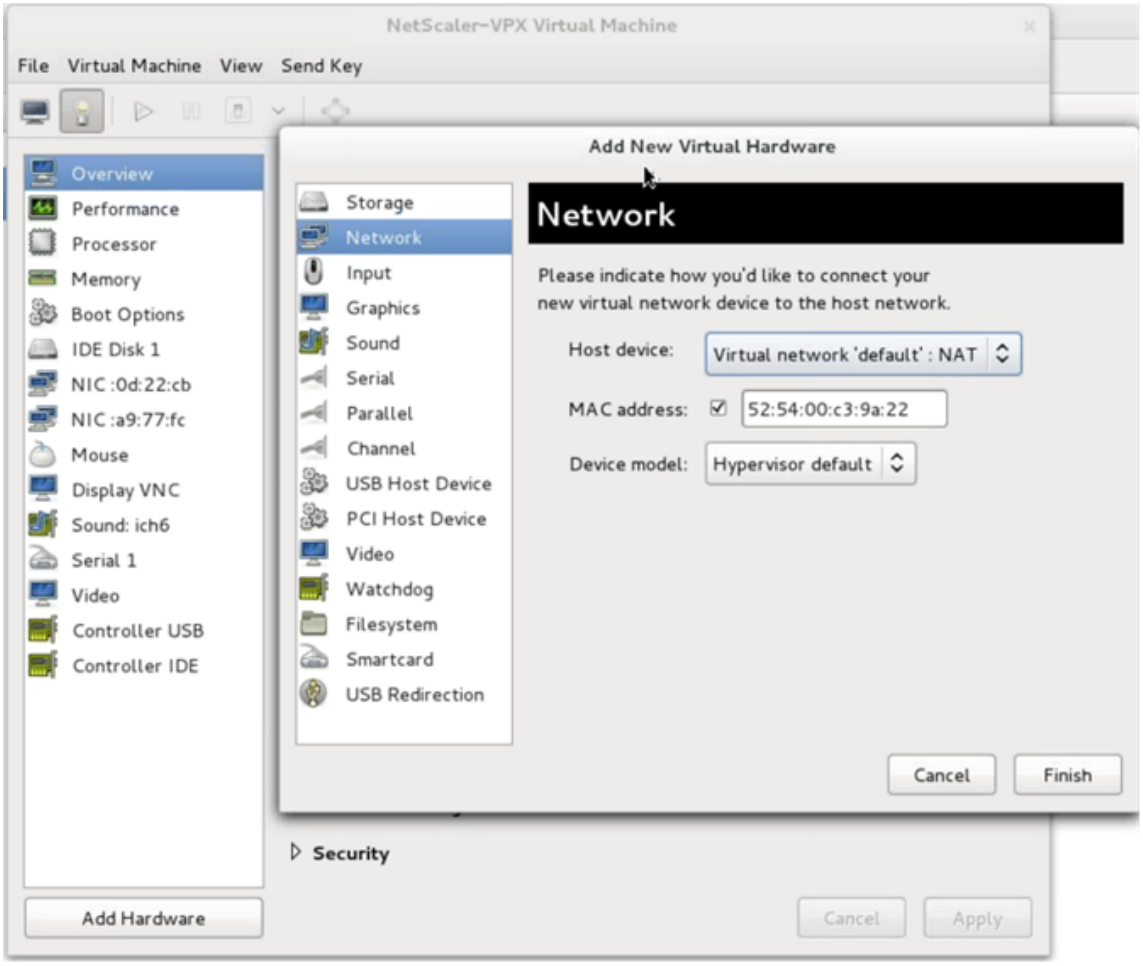
4. 自動プロビジョニングが失敗した場合、インスタンスはデフォルトの IP アドレス（192.168.100.1）で起動します。その場合は、初期設定を手動で完了する必要があります。詳細については、「ADC を初めて構成する」を参照してください。

仮想マシンマネージャーを使用して、**Citrix ADC VPX** インスタンスにインターフェイスを追加する

KVM で NetScaler VPX インスタンスをプロビジョニングしたら、インターフェイスを追加できます。

インターフェイスを追加するには、次の手順を実行します。

1. KVM の上で動作している NetScaler VPX インスタンスをシャットダウンします。
2. VPX インスタンスを右クリックし、ポップアップメニューから［**Open**］を選択します。
3.  、仮想ハードウェアの詳細を表示します。
4. [ハードウェアの追加] をクリックします。［**Add New Virtual Hardware**］ウィンドウで、ナビゲーションメニューから［**Network**］を選択します。



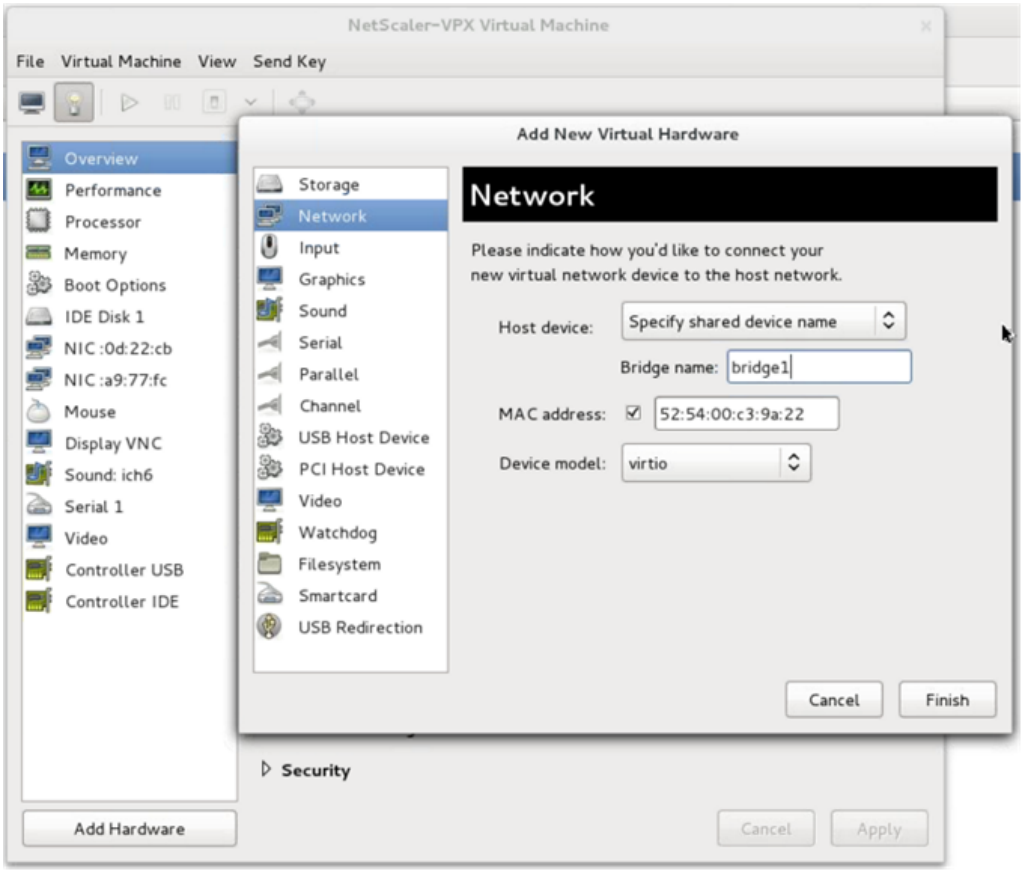
5. [**Host Device**] フィールドで、物理インターフェイスの種類を選択します。ホストデバイスの種類は、Bridge または MacVTap のいずれかにできます。macvTap の場合、VEPA モード、ブリッジ、プライベート、パススルーの 4 つのモードが可能です。

   a) Bridge の場合

      i. Host device - [Specify shared device name] オプションを選択します。

      ii. KVM ホストで構成される Bridge 名を指定します。

      注：KVM ホストで Linux Bridge が構成され、Bridge に物理インターフェイスが結合されて、Bridge が UP 状態になっている必要があります。



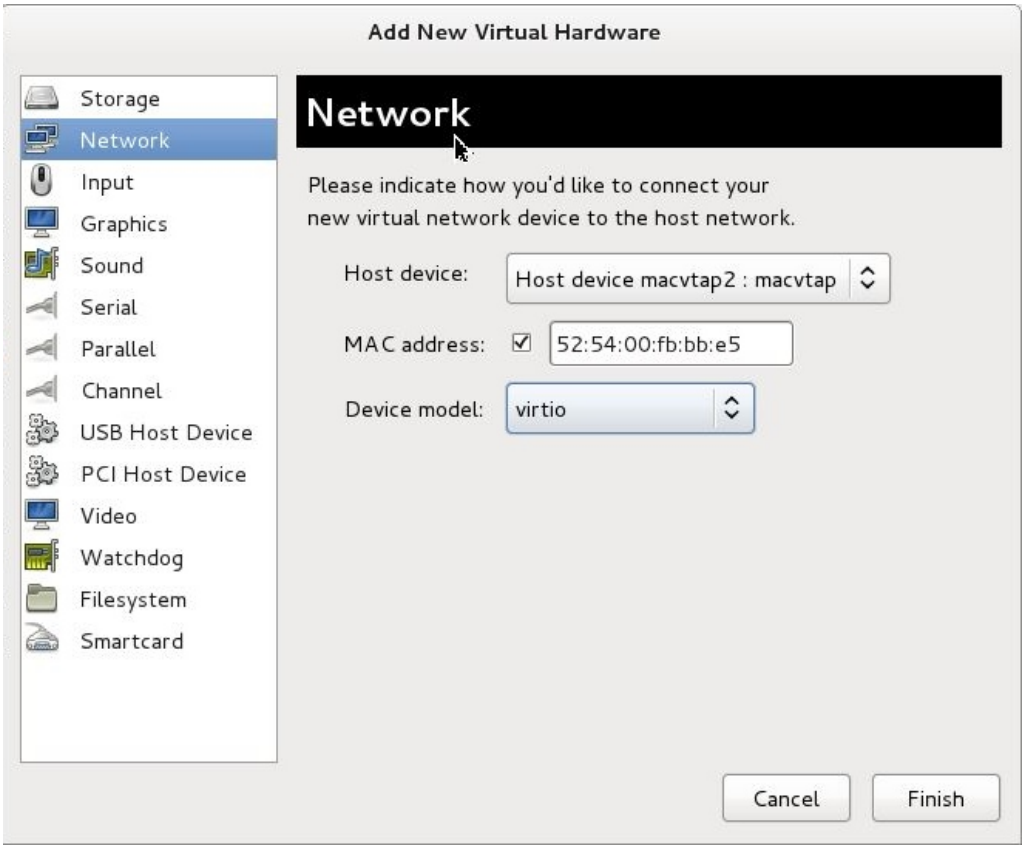
iii. デバイスモデル—`virtio`。

iv. [完了] をクリックします。

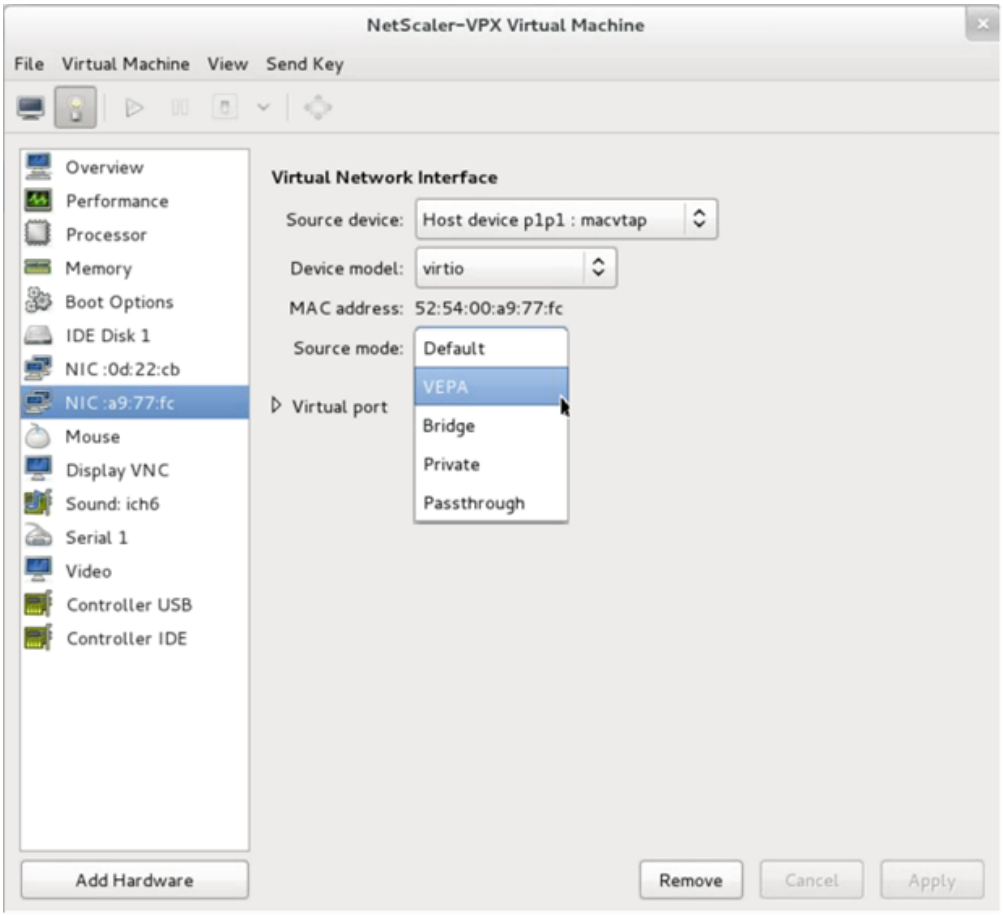
b) MacVtap 用

i. Host device - メニューからの物理インターフェイス

ii. デバイスモデル—`virtio`。



iii. ［完了］をクリックします。ナビゲーションペインで新しく追加された NIC を見ることができます。



iv. 新しく追加された NIC を選択して、この NIC の Source モードを選択します。利用可能なモードは VEPA、Bridge、Private、および Passthrough です。インターフェイスとモードについて詳しくは、「ソースインターフェイスおよびモード」を参照してください

v. [Apply] をクリックします。

6. VPX インスタンスを自動プロビジョニングする場合は、このドキュメントの「自動プロビジョニングを有効にするための構成ドライブの追加」セクションを参照してください。それ以外の場合は、VPX インスタンスの電源をオンにして、初期構成を手動で完了します。

> 重要
>
> スピード、デュプレックス、オートネゴシエーションなどの Interface パラメーター構成はサポートされません。

SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する

October 7, 2021

以下の NIC を使用して、シングルルート I/O 仮想化（SR-IOV）を使用して、Linux-KVM プラットフォーム上で動作する Citrix ADC VPX インスタンスを構成できます。

- インテル 82599 10G
- インテル X710 10G
- インテル XL710 40G
- インテル X722 10G

ここでは、次の操作の方法について説明します。

- SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する
- SR-IOV インターフェイスで静的 LA/LACP を構成する
- SR-IOV インターフェイスで VLAN を構成する

制限事項

インテル 82599、X710、XL710、X722 の NIC を使用する場合は、制限事項に留意してください。次の機能はサポートされません。

インテル **82599 NIC** の制限事項:

- L2 モード切り替え
- 管理パーティション化（共有 VLAN モード）
- 高可用性（アクティブ/アクティブモード）
- ジャンボフレーム。
- IPv6: SR-IOV インターフェイスが 1 つ以上ある場合は、VPX インスタンスで最大 30 個までの一意の IPv6 アドレスのみを設定できます。
- `ip link`コマンドによる SRIOV VF インターフェイスのハイパーバイザでの VLAN 設定はサポートされていません。
- スピート、デュプレックス、オートネゴシエーションなどの Interface パラメーター構成はサポートされません。

インテル **X710 10G**、インテル **XL710 40G**、インテル **X722 10G NIC** の制限事項:

- L2 モード切り替え
- 管理パーティション化（共有 VLAN モード）
- クラスタでは、XL710 NIC がデータ・インタフェースとして使用されている場合、ジャンボフレームはサポートされません。
- インターフェイスが切断され、再接続されると、インターフェイスリストが順序変更されます。
- スピート、デュプレックス、オートネゴシエーションなどの Interface パラメーター構成はサポートされません。
- インターフェイス名は、Intel X710 10G、Intel XL710 40G、Intel X722 10G NIC の場合は 40/X
- VPX インスタンスでは、最大 16 個のインテル XL710/X710/X722 SRIOV または PCI パススルーインターフェイスをサポートできます。

> 注**:** インテル X710 10G、インテル XL710 40G、インテル X722 10G NIC で IPv6 をサポートするには、KVM ホストで次のコマンドを入力して、仮想関数 (VF) で信頼モードを有効にする必要があります。
>
> ```
> ## ip link set <PNIC> <VF> trust on
> ```
>
> 例:
>
> ```
> ## ip link set ens785f1 vf 0 trust on
> ```

前提条件

SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する前に、次の前提条件タスクを完了してください。対応するタスクを完了する方法の詳細については、「NIC」列を参照してください。

タスク	インテル 82599 NIC	インテル X710、XL710、X722 NIC
1. NIC を KVM ホストに追加します。	-	-
2. 最新の Intel ドライバをダウンロードしてインストールします。	IXGBE ドライバー	I40E ドライバー
3. KVM ホスト上のドライバをブロックリストします。	/etc/modprobe.d/blacklist.conf ファイルに次のエントリを追加します。`blacklist ixgbevf`。IXGBE ドライバーのバージョン 4.3.15 を使用します（推奨）。	/etc/modprobe.d/blacklist.conf ファイルに次のエントリを追加します。`blacklist i40evf`。i40e ドライバーのバージョン 2.0.26 を使用します（推奨）。
4. KVM ホストで SR-IOV 仮想機能 (VF) を有効にします。次の 2 つの列の両方のコマンドで、`number_of_VFs` = 作成する仮想 VF の数。`device_name` = インターフェイス名です。	以前のバージョンのカーネル 3.8 を使用している場合は、次のエントリを /etc/modprobe.d/ixgbe ファイルに追加し、KVM ホストを再起動します。`options ixgbe max_vfs =<number_of_VFs>`。カーネル 3.8 以降を使用している場合は、次のコマンドを使用して VF を作成します。`echo <number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`。図 1 の例を参照してください。	以前のバージョンのカーネル 3.8 を使用している場合は、/etc/modprobe.d/i40e.conf ファイルに次のエントリを追加し、KVM ホストを再起動します。`options i40e max_vfs=<number_of_VFs>`。カーネル 3.8 以降を使用している場合は、次のコマンドを使用して VF を作成します。`echo<number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`。図 2 の例を参照してください。

タスク	インテル 82599 NIC	インテル X710、XL710、X722 NIC
5. rc.local ファイルに VF を作成するために使用したコマンドを追加して、VF を永続的にします。	図 3 の例を参照してください。	図 3 の例を参照してください。

> 重要
>
> SR-IOV VF を作成するときは、MAC アドレスを VF に割り当てないようにしてください。

図 **1**: インテル 82599 10G NIC の KVM ホストで SR-IOV VF を有効にする

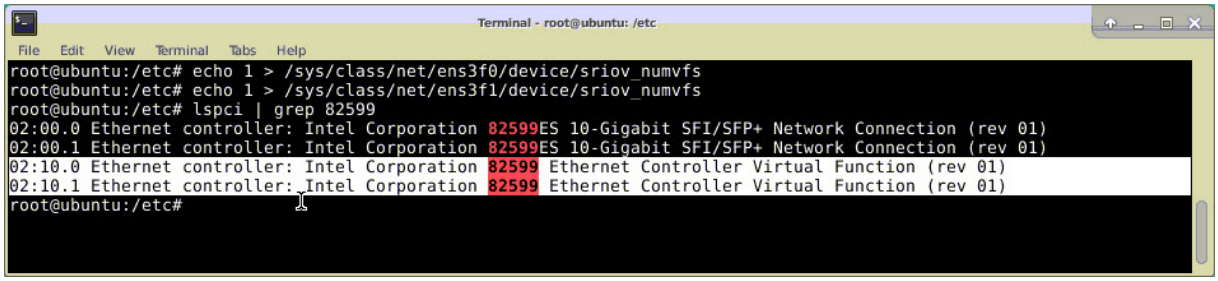


図 **2:** インテル X710 10G および XL710 40G NIC の KVM ホストで SR-IOV VF を有効にする

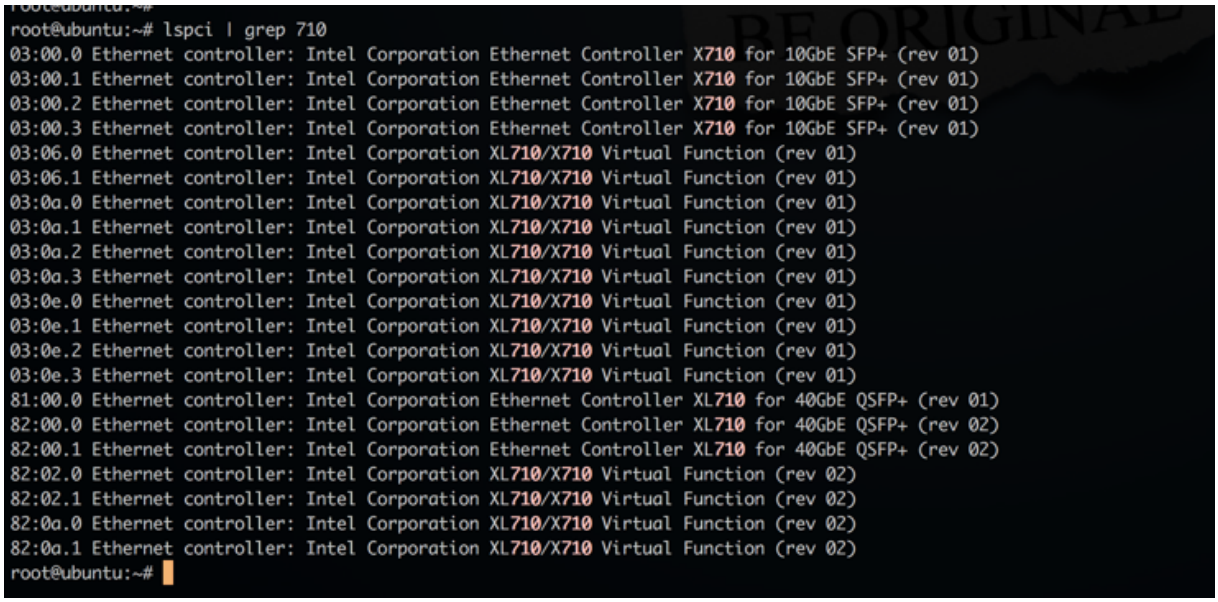


図 **3**: インテル X722 10G NIC の KVM ホストで SR-IOV VF を有効にする

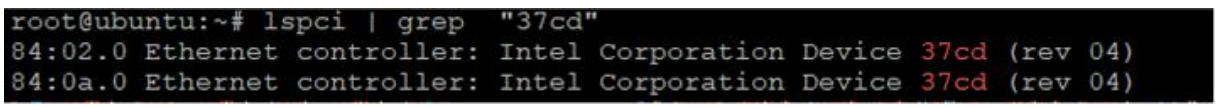
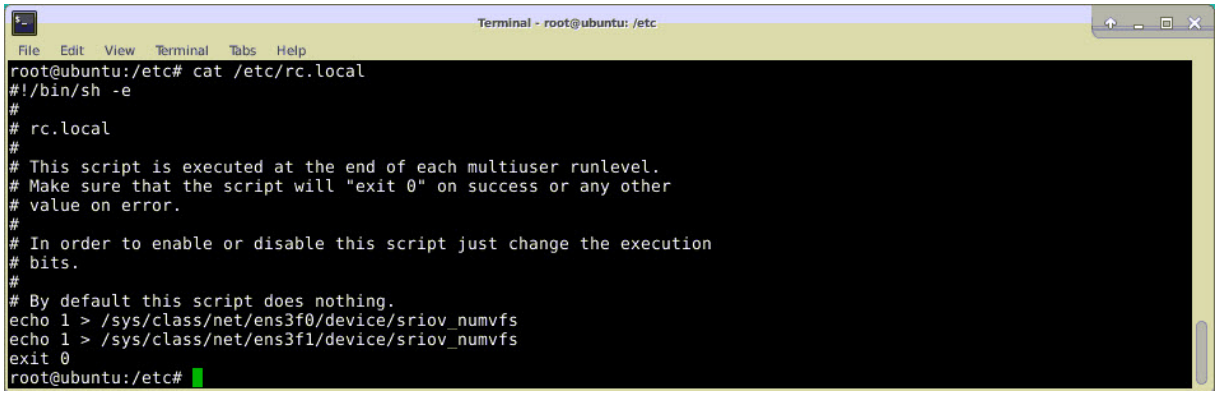


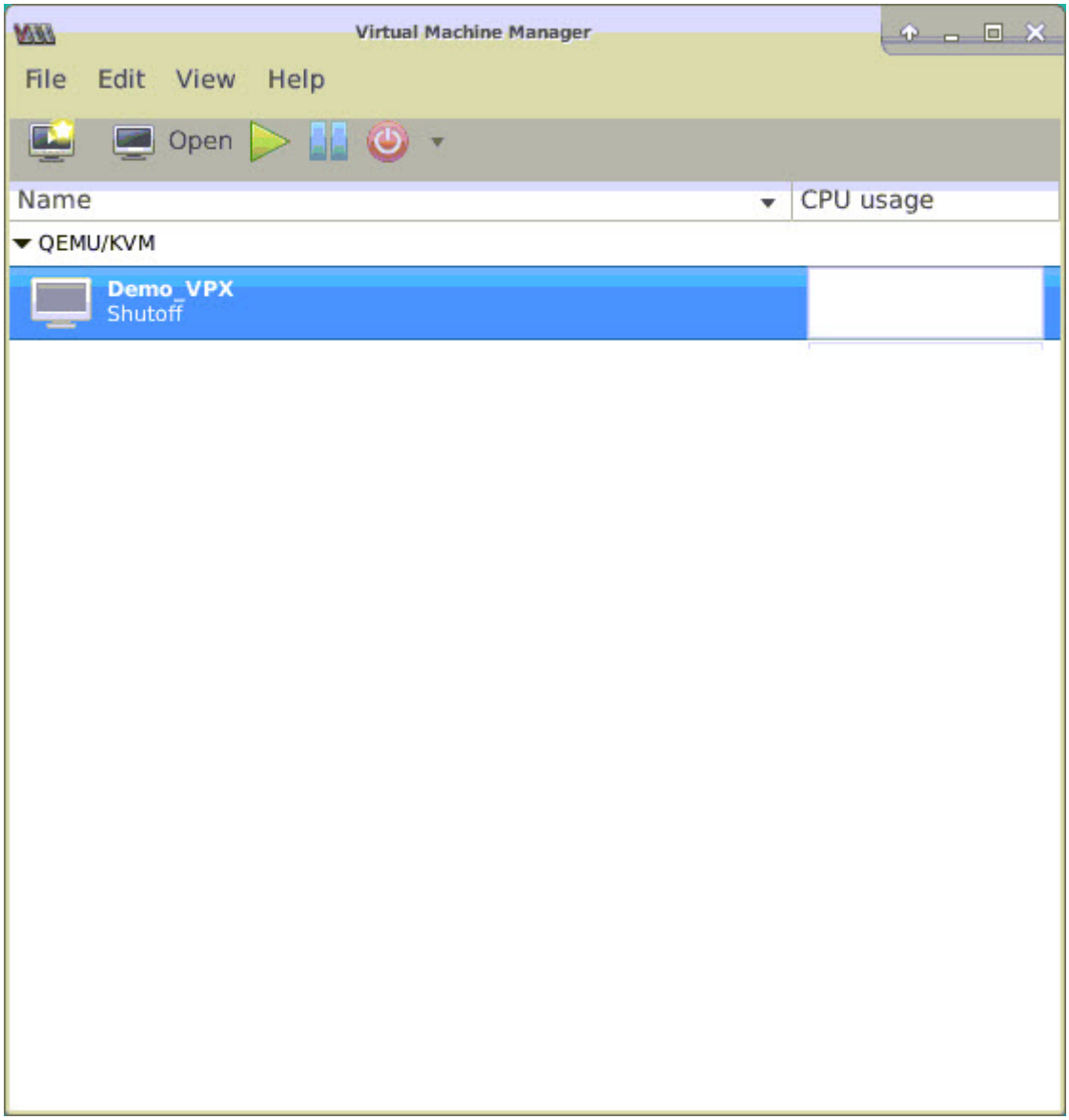
図 **4**: VF を永続的にする



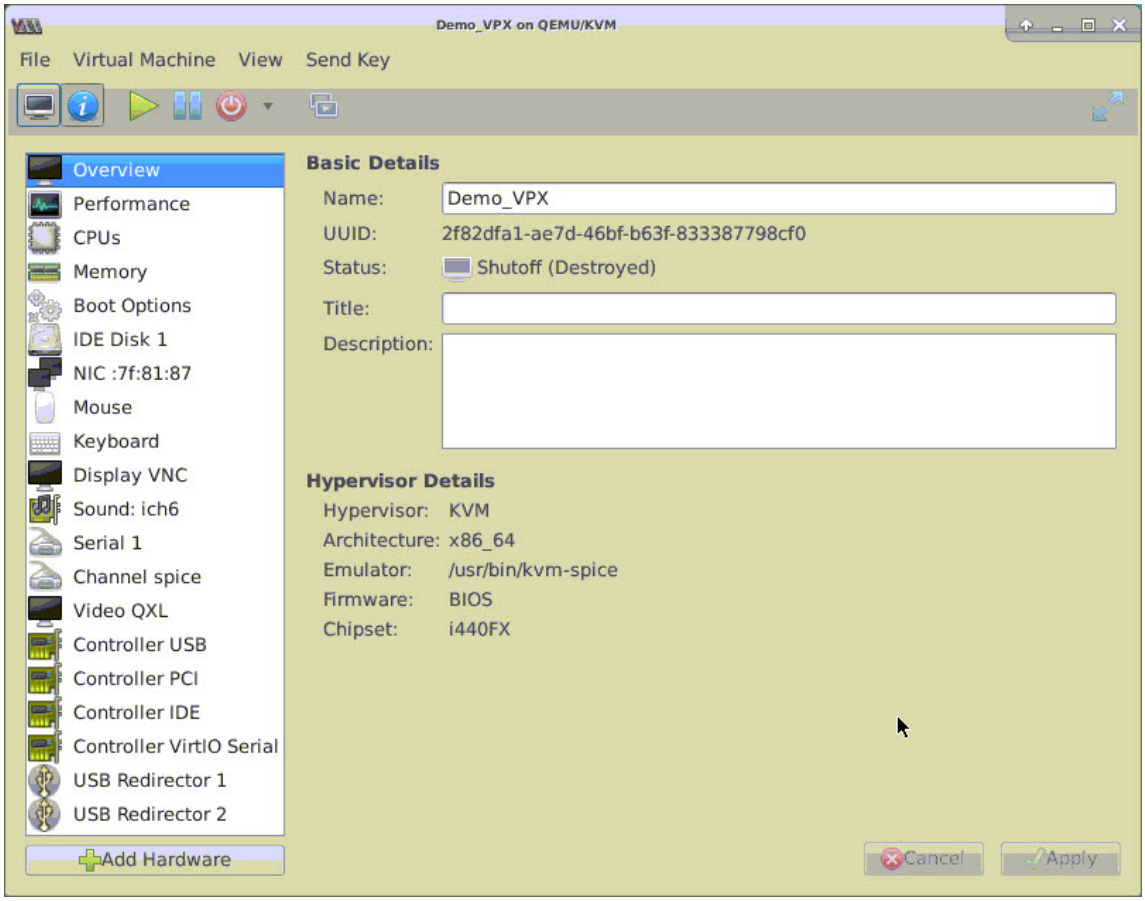
SR-IOV ネットワークインターフェイスを使用するように **Citrix ADC VPX** インスタンスを構成する

仮想マシンマネージャを使用して SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成するには、次の手順を実行します。

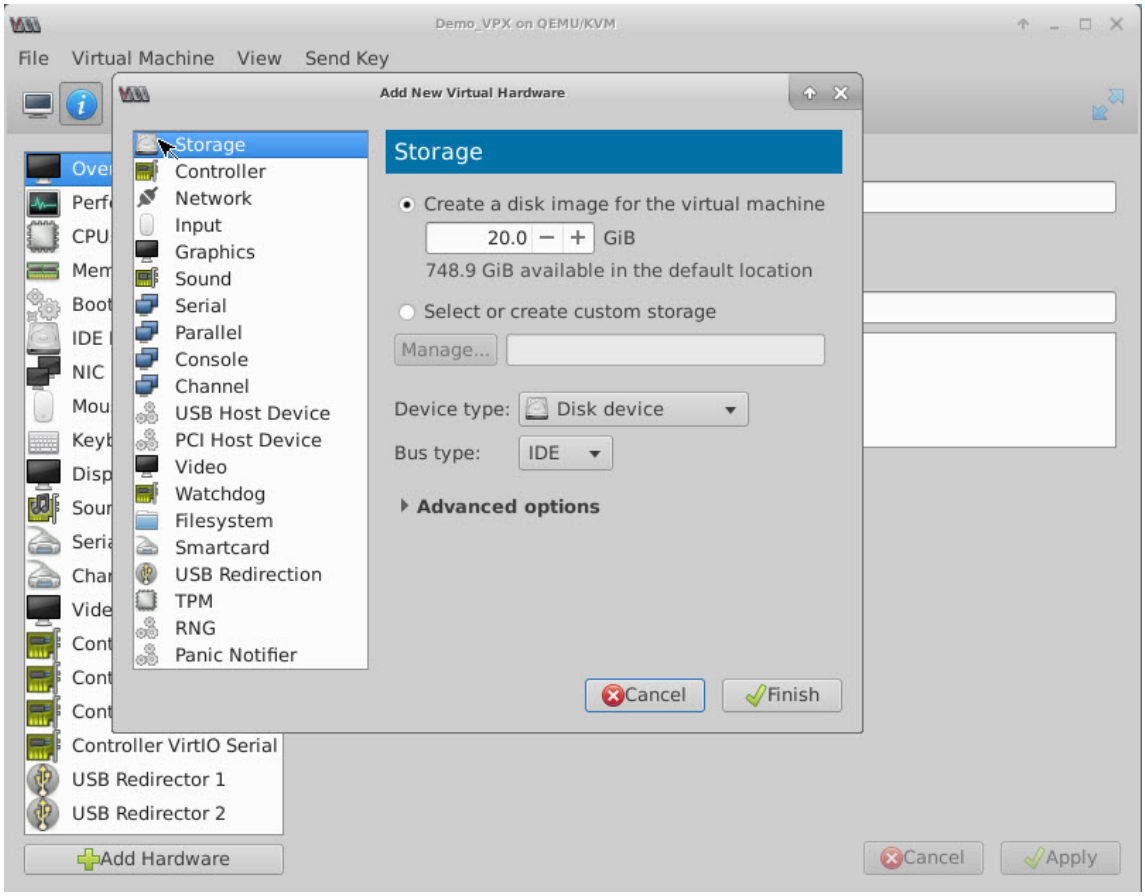
1. Citrix ADC VPX インスタンスの電源を切ります。
2. Citrix ADC VPX インスタンスを選択し、[開く] を選択します。



3. <virtual machine on KVM> ウィンドウで、**i** アイコンを選択します。



4. [ハードウェアの追加] を選択します。



5. [新しい仮想ハードウェアの追加] ダイアログボックスで、次の操作を行います。

   a) [PCI ホストデバイス] を選択します。

   b) [Host Device] セクションで、作成した VF を選択して、[Finish] をクリックします。

図 **4**: インテル 82599 10G NIC の VF

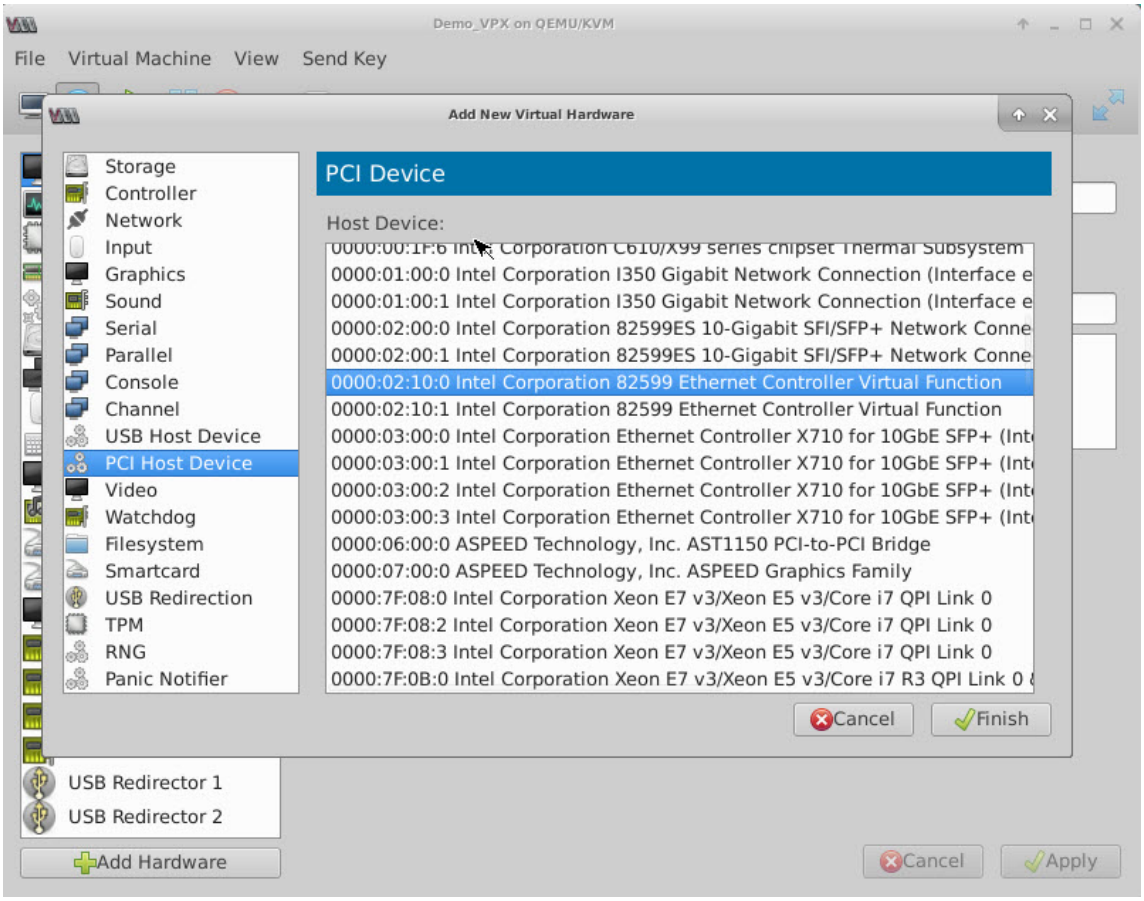


図 **5**: インテル XL710 40G NIC の VF

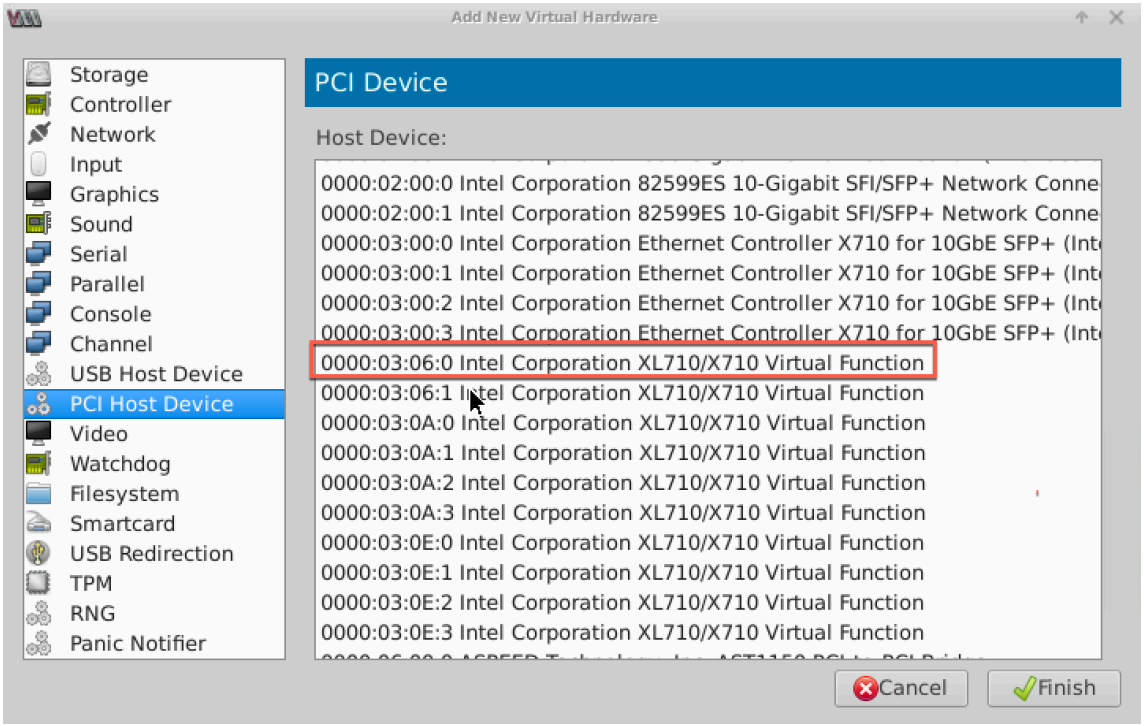
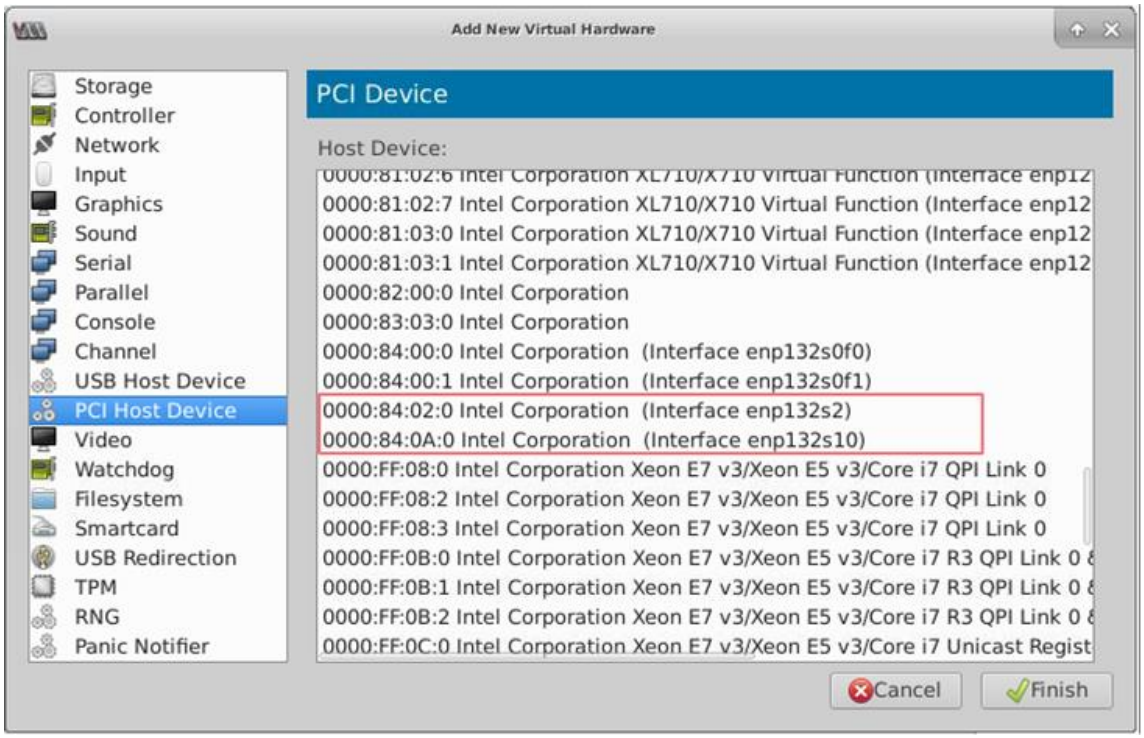


図 **6**: インテル X722 10G NIC の VF



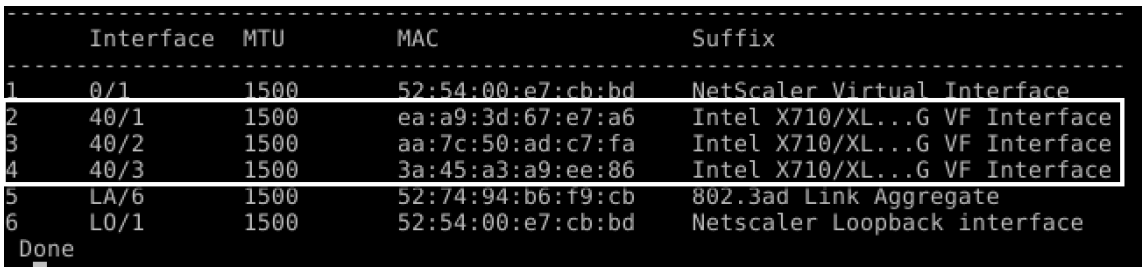
6. 手順 4 と 5 を繰り返し、作成した VF を追加します。
7. Citrix ADC VPX インスタンスの電源を入れます。
8. Citrix ADC VPX インスタンスの電源が入ったら、次のコマンドを使用して構成を確認します。

```
1 show interface summary
2 <!--NeedCopy-->
```

構成したすべてのインターフェイスが出力に表示されます。

図 **6**: インテル 82599 NIC の出力サマリー

図 **7**. インテル X710 および XL710 NIC の出力サマリー。



SR-IOV インターフェイスでスタティック **LA/LACP** を設定する

> 重要
>
> SR-IOV VF を作成する場合、MAC アドレスを VF に割り当てていないことを確認してください。

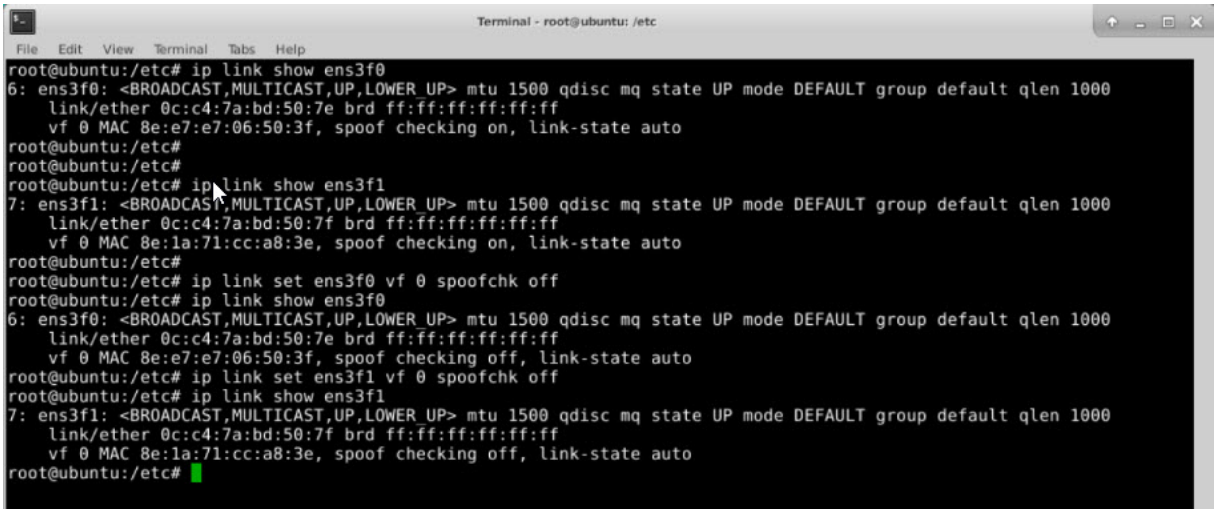
リンクアグリゲーションモードで、SR-IOV VF を使用するには、作成した VF のなりすましチェックを無効にします。KVM ホストでなりすましチェックを無効にするには、以下のコマンドを使用します。

```
*ip link set \<interface\\_name\> vf \<VF\\_id\> spoofchk off*
```

各項目の意味は次のとおりです:

- Interface_name – インターフェイス名です。
- VF_id – Virtual Function ID です。

例:



作成したすべての VF のなりすましチェックを無効にします。Citrix ADC VPX インスタンスを再起動し、リンクアグリゲーションを構成します。詳細な手順については、リンク集約の設定を参照してください。

SR-IOV インターフェイスで VLAN を構成する

SR-IOV VF で VLAN を構成できます。詳細な手順については 、VLAN の設定を参照してください。

> 重要
>
> KVM ホストに VF インターフェイスの VLAN 設定が含まれていないことを確認してください。

PCI パススルーネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する

October 7, 2021

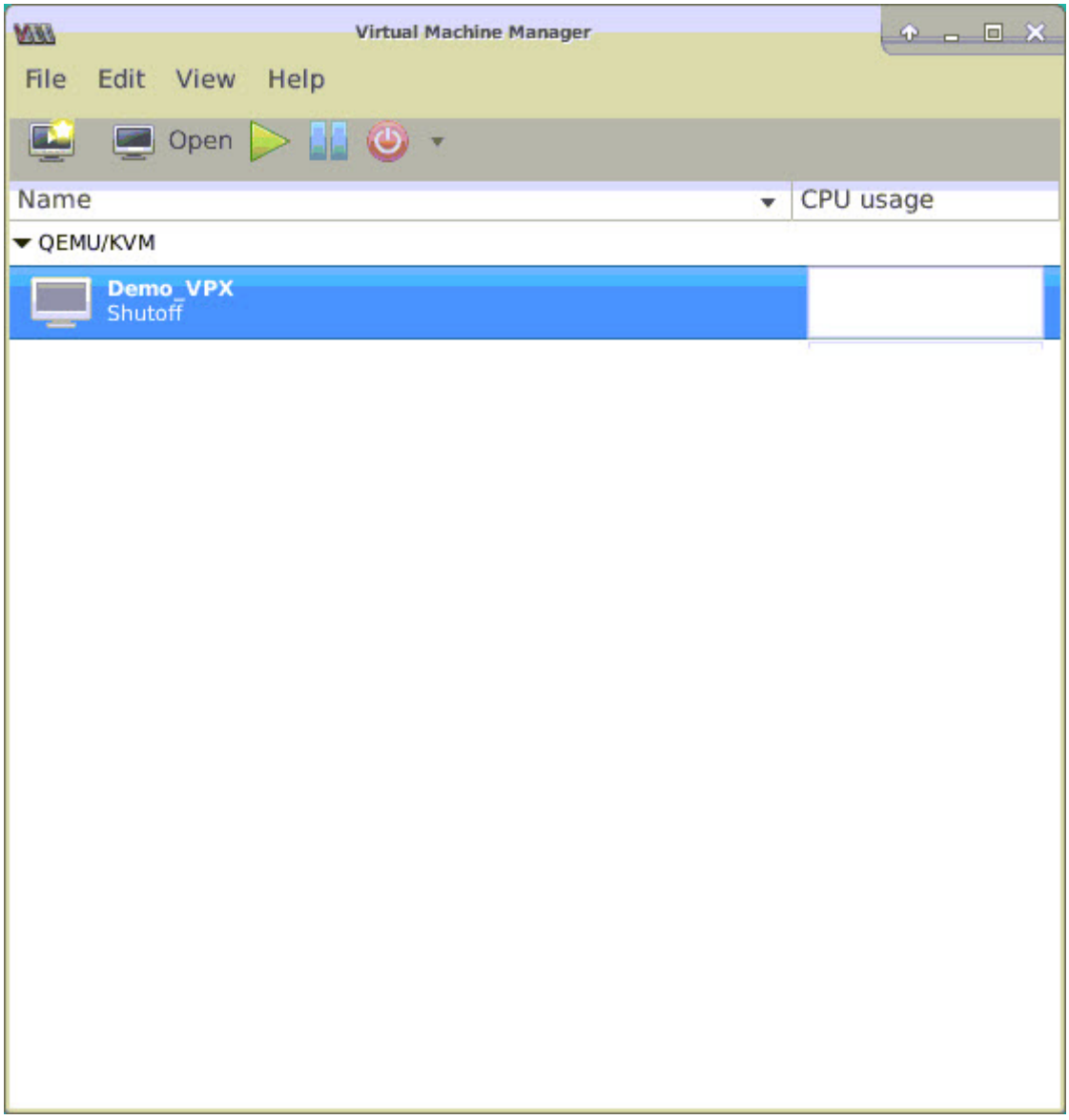
Linux-KVM プラットフォームに Citrix ADC VPX インスタンスをインストールして構成したら、仮想マシンマネージャーを使用して、PCI パススルーネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

前提条件

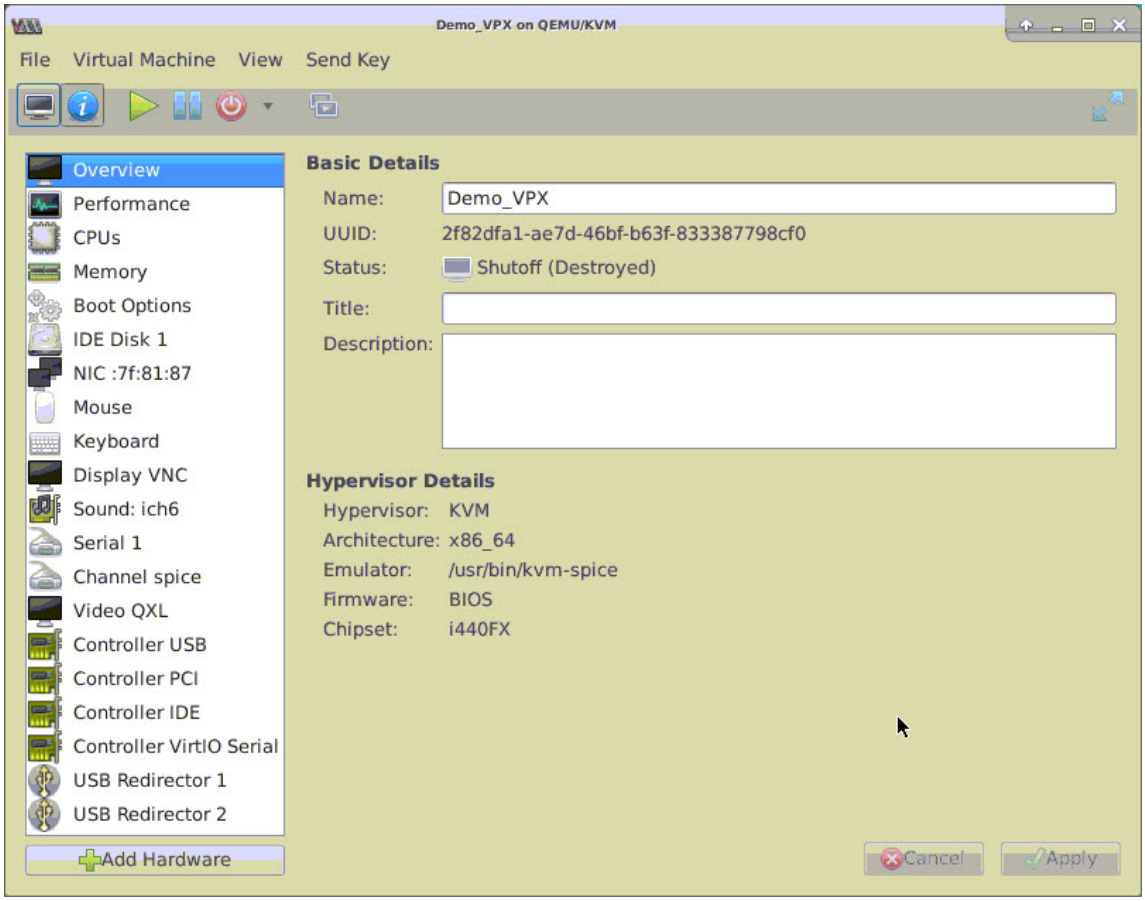
- KVM ホスト上のインテル XL710 NIC (NIC) のファームウェア・バージョンは 5.04 です。
- KVM ホストは、IOMMU（Input–Output Memory Management Unit）と Intel VT をサポートし、これらは KVM ホストの BIOS で有効になっています。KVM ホストで **IOMMU** を有効にするには、**/boot/grub2/grub.cfg** ファイルに次のエントリを追加します。
- 次のコマンドを実行して KVM ホストを再起動します。**grub2-MKConfig —o /boot/grub2/grub.cfg**

仮想マシンマネージャーを使用して **PCI** パススルーネットワークインターフェイスを使用するように **Citrix ADC VPX** インスタンスを構成するには:

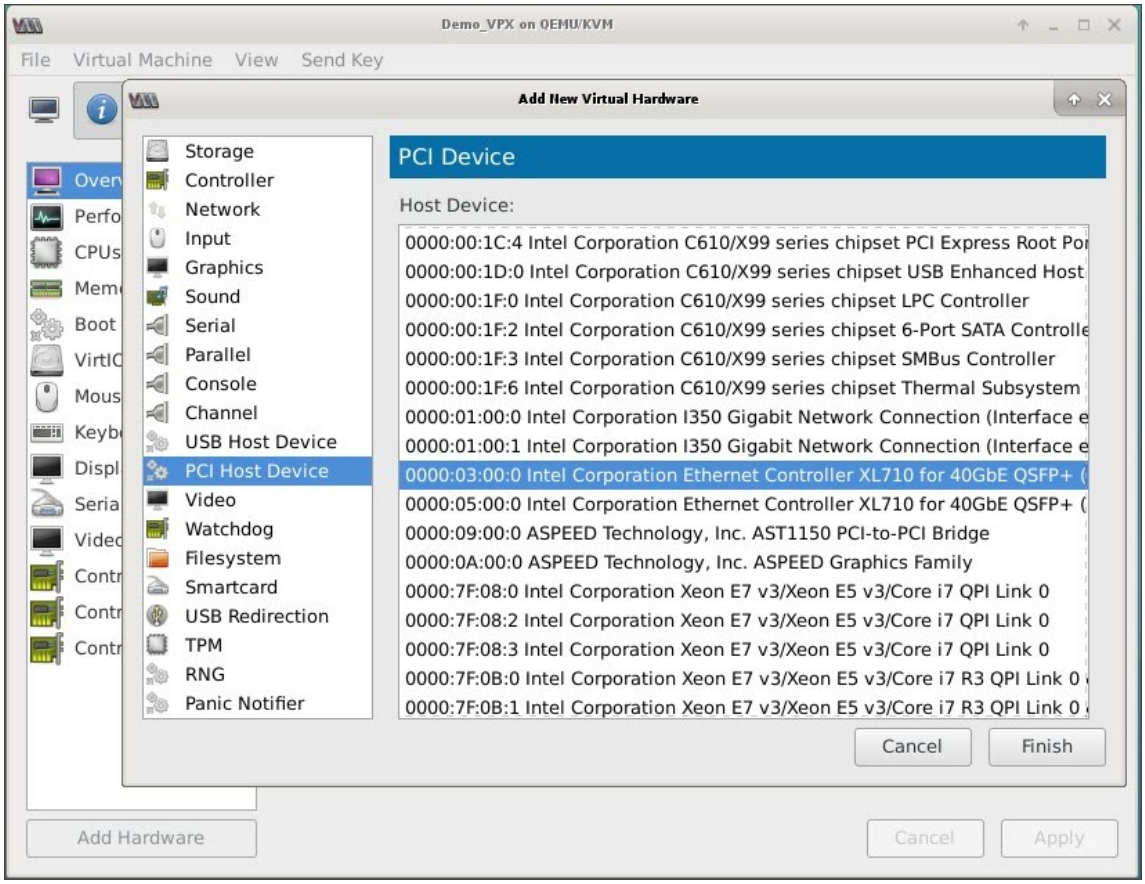
1. Citrix ADC VPX インスタンスの電源を切ります。
2. Citrix ADC VPX インスタンスを選択し、［開く］をクリックします。



3. **KVM>** の **virtual_machine** ウィンドウで、**i** アイコンをクリックします。



4. [ハードウェアの追加] をクリックします。
5. [新しい仮想ハードウェアの追加] ダイアログボックスで、次の操作を行います。
   a. [**PCI** ホストデバイス] を選択します。
   b. [ホストデバイス] セクションで、インテル XL710 物理機能を選択します。
   c. ［完了］をクリックします。

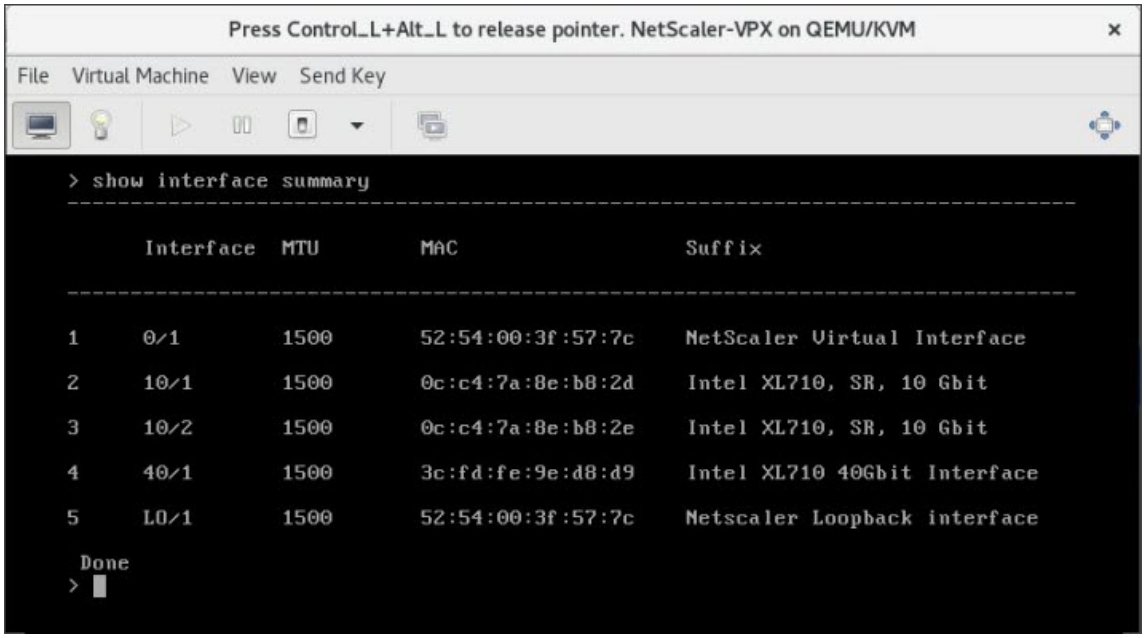


6. 手順 **4** と **5** を繰り返して、インテル XL710 物理関数を追加します。
7. Citrix ADC VPX インスタンスの電源を入れます。
8. Citrix ADC VPX インスタンスの電源が入ったら、次のコマンドを使用して構成を確認できます。

```
COMMAND

> show interface summary
```

出力には、設定したすべてのインターフェイスが表示されている必要があります。



virsh プログラムを使用して Citrix ADC VPX インスタンスをプロビジョニングする

October 7, 2021

virshプログラムは VM ゲストを管理するためのコマンドラインツールです。その機能性は Virtual Machine Manager に似ています。これにより VM Guest の状態（開始、停止、一時停止など）を変更でき、新しい Guests およびデバイスをセットアップして、既存の構成を編集できます。virshプログラムは、VM ゲスト管理操作のスクリプト作成にも役立ちます。

virshプログラムを使用して Citrix ADC VPX をプロビジョニングするには、次の手順に従います。

1. tar コマンドを使用して、Citrix ADC VPX パッケージを解凍します。nsvpx-kvm-*_nc.tgz パッケージには、次のコンポーネントが含まれています。

   - VPX 属性 [NSVPX-KVM-*_nc.xml] を指定するドメイン XML ファイル
   - NS-VM ディスクイメージ [Checksum.txt] のチェックサム
   - NS-VM ディスクイメージ [NSVPX-KVM-*_nc.raw]

   例:

```
tar -xvzf NSVPX-KVM-10.1-117_nc.tgz
NSVPX-KVM-10.1-117_nc.xml
NSVPX-KVM-10.1-117_nc.raw
checksum.txt
<!--NeedCopy-->
```


2. nsVPX-KVM-*_nc.xml XML ファイルを <DomainName>-nsVPX-KVM-*_nc.xml という名前のファイルにコピーします。<DomainName> は、仮想マシンの名前でもあります。例:

```
cp NSVPX-KVM-10.1-117_nc.xml NetScaler-VPX-NSVPX-KVM-10.1-117_nc.xml
<!--NeedCopy-->
```

3. <DomainName>-nsVPX-KVM-*_nc.xml ファイルを編集して、次のパラメータを指定します。

   - name - 名前を指定します。
   - Mac：MAC アドレスを指定します。
     注：ドメイン名と MAC アドレスは一意である必要があります。
   - source file：ディスクイメージの絶対ソースパスを指定します。ファイルパスは絶対パスである必要があります。RAW イメージファイルまたは QCOW2 イメージファイルのパスを指定することができます。

     RAW イメージファイルを指定する場合は、次の例のようにディスクイメージのソースパスを指定します。

     例:

```
<name>NetScaler-VPX</name>
<mac address='52:54:00:29:74:b3'/>
<source file='/root/NSVPX-KVM-10.1-117_nc.raw'/>
<!--NeedCopy-->
```

     次の例に示すように、絶対的な QCOW2 ディスクイメージのソースパスを指定し、ドライバの種類を **qcow2** として定義します。

     例:

```
<name>NetScaler-VPX</name>
<mac address='52:54:00:29:74:b3'/>
<driver name ='qemu' type='qcow2'/>
<source file='/root/NSVPX-KVM-10.1-117_nc.qcow'/>*
<!--NeedCopy-->
```

4. <DomainName>-nsVPX-KVM-*_nc.xml ファイルを編集して、ネットワークの詳細を設定します。

   - source dev - インターフェースを指定します。
   - mode - モードを指定します。デフォルトのインターフェイスは **Macvtap** ブリッジです。

例: モード:macvTap Bridge ターゲットインターフェイスをethxに設定し、モードをブリッジモデルタイプvirtioに設定

```
  <interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth0' mode='bridge'/>
      <target dev='macvtap0'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
         function='0x0'/>
    </interface>
<!--NeedCopy-->
```

ここで、eth0 は仮想マシンに接続された物理インターフェイスです。

5. 次のコマンドを使用して、<DomainName>-NSVPX-KVM-*_nc.xml ファイル内の仮想マシンの属性を定義します。virshdefine <DomainName>-NSVPX-KVM-*_nc.xml 例:

```
virsh define NS-VPX-NSVPX-KVM-10.1-117_nc.xml
<!--NeedCopy-->
```

6. 次のコマンドを入力して VM を起動します。virsh start [<DomainName>|<DomainUUID>] 例:

```
virsh start NetScaler-VPX
<!--NeedCopy-->
```

7. virshコンソールコンソールからゲスト VM を接続します [<DomainName>| <DomainUUID>|<DomainID>] 例:

```
virsh console NetScaler-VPX
<!--NeedCopy-->
```

virsh プログラムを使用して Citrix ADC VPX インスタンスにインターフェイスを追加する

KVM 上で Citrix ADC VPX をプロビジョニングした後、インターフェイスを追加できます。

インターフェイスを追加するには、次の手順を実行します。

1. KVM で実行されている Citrix ADC VPX インスタンスをシャットダウンします。
2. 次のコマンドを使用して <DomainName>-NSVPX-KVM-*_nc.xml ファイルを編集します。`virsh` edit [<DomainName> | <DomainUUID>]
3. の中に < DomainName>-NSVPX-KVM-*_nc.xml ファイルに、次のパラメータを追加します。

   a) **MacVtap** 用

   - Interface type - インターフェイスの種類として「direct」を指定します。
   - MAC アドレス：MAC アドレスを指定し、MAC アドレスがインターフェイス全体で一意であることを確認します。
   - source dev - インターフェイス名を指定します。
   - mode-モードを指定します。サポートされているモードは、ブリッジ、VEPA、プライベート、パススルーです。
   - モデルタイプ-モデルタイプを次のように指定します。`virtio`

   例:

   モード：MacVTap Pass-through

   ターゲットインタフェースを
   `ethx`、モードを
   ブリッジ、モデルタイプを次のように設定します。
   `virtio`

```
<interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth1' mode='passthrough'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

   ここで eth1 は仮想マシンに接続された物理インターフェイスです。

   b) ブリッジモードの場合

   注：KVM ホストで Linux Bridge が構成され、Bridge に物理インターフェイスが結合されて、Bridge が UP 状態になっている必要があります。

   - Interface type - インターフェイスの種類として「bridge」を指定します。
   - MAC アドレス：MAC アドレスを指定し、MAC アドレスがインターフェイス全体で一意であることを確認します。
   - source dev - ブリッジ名を指定します。
   - モデルタイプ-モデルタイプを次のように指定します。`virtio`

例：Bridge Mode

```
<interface type='bridge'>
      <mac address='52:54:00:2d:43:a4'/>
      <source bridge='br0'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

Citrix ADC VPX ゲスト仮想マシンの管理

October 7, 2021

仮想マシンマネージャと virshプログラムを使用して、仮想マシンゲストの起動または停止、新しいゲストとデバイスの設定、既存構成の編集、仮想ネットワークコンピューティング (VNC) によるグラフィカルコンソールへの接続などの管理タスクを実行できます。

仮想マシンマネージャーを使用して VPX ゲスト仮想マシンを管理する

- VM ゲストを一覧表示する

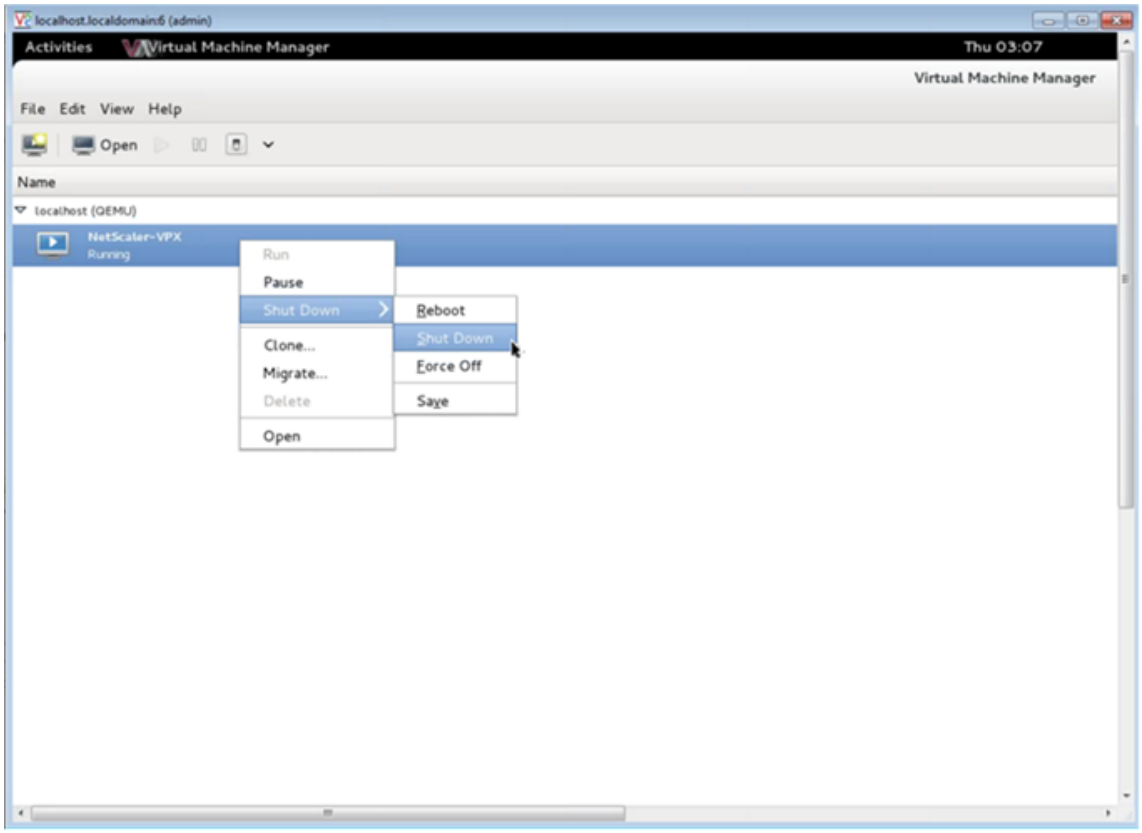
  Virtual Machine Manager のメインウィンドウには、接続される各 VM ホストサーバのすべての VM Guests の一覧が表示されます。各仮想マシンゲストエントリには、仮想マシンの名前と、アイコンに表示されるステータス（実行中、一時停止、またはシャットオフ）が含まれます。

- グラフィカルコンソールを開く

  VM Guest に対してグラフィカルコンソールを開いて、VNC 接続介して物理的ホストと通信するようにマシンと相互通信できます。Virtual Machine Manager でグラフィカルコンソールを開くには、VM Guest エントリーを右クリックして、ポップアップメニューで［オープン］オプションを選択します。

- ゲストの起動とシャットダウン

  Virtual Machine Manager から VM Guest を開始または停止できます。VM の状態を変更するには、VM Guest エントリーを右クリックして、ポップアップメニューで［Run］または［Shut Down］オプションのいずれかを選択します。



- ゲストを再起動する

  Virtual Machine Manager から VM Guest を再起動できます。VM を再起動するには、VM Guest エントリーを右クリックして、ポップアップメニューで［Shut Down］>［Reboot］を選択します。。

- ゲストを削除する

  デフォルトでは、VM Guest を削除すると XML 構成が消去されます。また、ゲストのストレージファイルを削除できます。これを実行して、完全にそうすることはゲストを消します。

  1. Virtual Machine Manager で、VM Guest エントリーを右クリックします。
  2. ポップアップメニューで［Delete from］を選択します。確認用のウィンドウが開きます。
     注:[削除] オプションは、VM ゲストがシャットダウンされている場合にのみ有効になります。
  3. ［削除］をクリックします。
  4. 完全にゲストを消去するには、［Delete Associated Storage Files］チェックボックスをオンにして、関連付けられた.raw ファイルを削除します。

`virsh` プログラムを使用して **Citrix ADC VPX** ゲスト仮想マシンを管理する

- VM ゲストとその現在の状態を一覧表示します。

  ゲストに関する情報を表示するために`virsh`を使用するには

  ```
  virsh list --all
  ```


コマンド出力はすべてのドメインとその状態を表示します。出力例:

```
Id Name                 State
----------------------------------
  0 Domain-0            running
  1 Domain-1            paused
  2 Domain-2            inactive
  3 Domain-3            crashed
<!--NeedCopy-->
```

- virshコンソールを開きます。

  ゲスト仮想マシンをコンソールから接続します。

  ```
  virsh console [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh console NetScaler-VPX
  ```

- ゲストの起動とシャットダウン。

  Guest は DomainName または Domain-UUID を使って開始できます。

  ```
  virsh start [<DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh start NetScaler-VPX
  ```

  ゲストをシャットダウンするには:

  ```
  virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh shutdown NetScaler-VPX
  ```

- ゲストを再起動する

  ```
  virsh reboot [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  例:

  ```
  virsh reboot NetScaler-VPX
  ```

  ゲストを削除する

  ゲスト仮想マシンを削除するには、ゲストをシャットダウンし、<DomainName>-NSVPX-KVM-*_nc.xml の定義を解除する必要があります。


```
virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
virsh undefine [<DomainName> | <DomainUUID>]
<!--NeedCopy-->
```

例:

```
virsh shutdown NetScaler-VPX
virsh undefine NetScaler-VPX
<!--NeedCopy-->
```

> 注:delete コマンドでは、手動で削除する必要があるディスクイメージファイルは削除されません。

OpenStack で SR-IOV を使用して Citrix ADC VPX インスタンスをプロビジョニングする

October 7, 2021

シングルルート I/O 仮想化（SR-IOV）テクノロジーを使用する高性能な Citrix ADC VPX インスタンスを OpenStack にデプロイできます。

SR-IOV テクノロジーを使用する Citrix ADC VPX インスタンスは、以下の 3 つのステップで展開できます。

- ホスト上で SR-IOV Virtual Functions（VF）を有効にします。
- VF を構成し、OpenStack で使用できるようにします。
- OpenStack で Citrix ADC VPX をプロビジョニングします。

前提条件

次のことを確実にします。

- インテル 82599 NIC (NIC) をホストに追加します。
- 最新の IXGBE ドライバーをダウンロードしてインストールします。
- ホスト上の IXGBEVF ドライバをブロックリストします。/etc/modprobe.d/blacklist.conf ファイルに次のエントリを追加します。ブロックリスト `ixgbevf`

> 注
>
> `ixgbe`ドライバーのバージョンは 5.0.4 以上でなければなりません。

ホストで **SR-IOV VF** を有効にする

SR-IOV VF を有効にするには、次のいずれかの手順を実行します。

- <number_of_VFs>3.8 より前のカーネルバージョンを使用している場合は、/etc/modprobe.d/ixgbe ファイルに次のエントリを追加し、ホストを再起動します。オプション ixgbe max_vfs=
- カーネル 3.8 以降のバージョンを使用している場合、以下のコマンドを使用して VF を作成します。

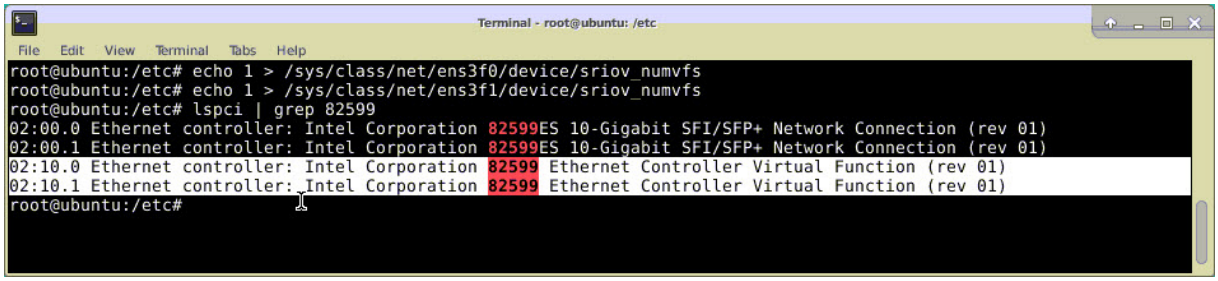
```
1     echo <number_of_VFs> > /sys/class/net/<device_name>/device/
         sriov_numvfs
2  <!--NeedCopy-->
```

各項目の意味は次のとおりです:

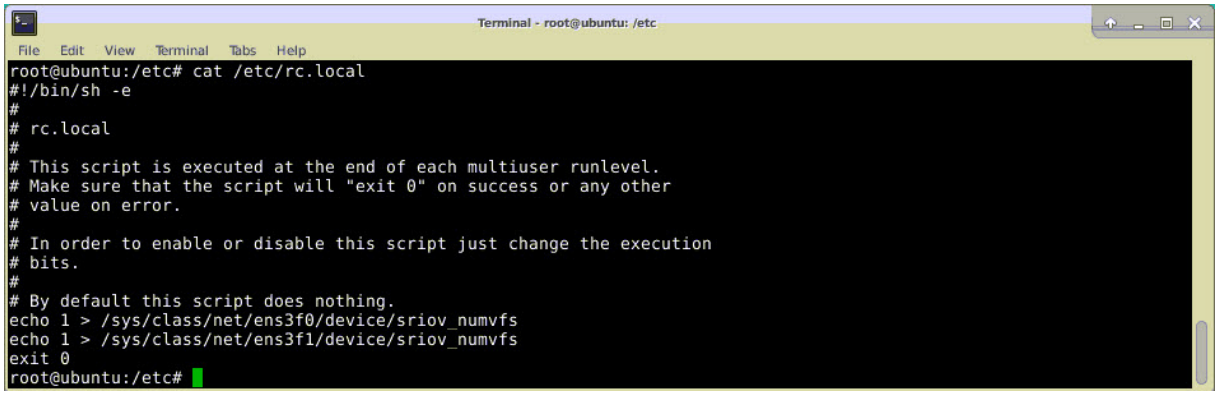
- number_of_VFs は、作成する Virtual Function の数です。
- device_name はインターフェイス名です。

> 重要
>
> SR-IOV VF を作成する場合、MAC アドレスを VF に割り当てないようにしてください。

次に、作成している 4 つの VF の例を示します。



VF を永続的にし、VF の作成に使用したコマンドを **rc.local** ファイルに追加します。rc.local ファイルの内容を示す例を次に示します。



詳細については、このインテル SR-IOV 構成ガイドを参照してください。

OpenStack で VF を設定して利用できるようにする

以下のリンクに記載されている手順に従って、OpenStack で SR-IOV を設定します：https://wiki.openstack.org/wiki/SR-IOV-Passthrough-For-Networking。

OpenStack で Citrix ADC VPX インスタンスをプロビジョニングする

OpenStack CLI を使用して、OpenStack 環境で Citrix ADC VPX インスタンスをプロビジョニングできます。

VPX インスタンスをプロビジョニングします。オプションで、コンフィグドライブのデータを使用します。「構成ドライブ」とは、インスタンスの起動時にアタッチされる特殊な構成ドライブを指します。この構成ドライブを使用して、インスタンスのネットワーク設定を構成する前に、管理 IP アドレス、ネットワークマスク、デフォルトゲートウェイなどのネットワーク構成情報をインスタンスに渡すことができます。

OpenStack が VPX インスタンスをプロビジョニングする場合、まず OpenStack を示す特定の BIOS 文字列（OpenStack ファウンデーション）を読み取ることによって、インスタンスが OpenStack 環境で起動していることを検出します。Red Hat Linux ディストリビューションの場合、この文字列は/etc/nova/release に保存されます。これは、KVM ハイパーバイザープラットフォームに基づくすべての OpenStack 実装で使用できる標準メカニズムです。ドライブには特定の OpenStack ラベルが必要です。構成ドライブが検出されると、インスタンスは nova boot コマンドで指定されたファイル名から次の情報を読み取ろうとします。以下の手順では、このファイルを「userdata.txt」と呼びます。

- 管理 IP アドレス
- ネットワークマスク
- デフォルトゲートウェイ

パラメーターが正しく読み取られると、それらの値が NetScaler スタックに適用されます。これにより、インスタンスをリモートから管理できるようになります。パラメーターが読み取られない場合、または構成ドライブが存在しない場合は、インスタンスが以下のデフォルトの処理を実行します。

- DHCP から IP アドレス情報を取得する。
- DHCP から情報を取得できない場合は、デフォルトのネットワーク構成として 192.168.100.1/16 を使用する。

CLI を使用して OpenStack で Citrix ADC VPX インスタンスをプロビジョニングする

OpenStack 環境で VPX インスタンスをプロビジョニングするには、OpenStack の CLI を使用します。次に、OpenStack で Citrix ADC VPX インスタンスをプロビジョニングする手順の概要を示します。

1. .tgz ファイルから **.** qcow2ファイルを抽出する
2. qcow2 イメージから OpenStack イメージを作成する
3. VPX インスタンスのプロビジョニング

OpenStack 環境で VPX インスタンスをプロビジョニングするには、次の手順を実行します。

1. 次のコマンドを入力して、.tqzファイルからqcow2ファイルを抽出します。

```
tar xvzf <TAR file>
tar xvzf NSVPX-KVM-12.0-26.2_nc.tgz
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
<!--NeedCopy-->
```

2. 次のコマンドを入力して、手順 1 で抽出した.qcoz2ファイルを使用して OpenStack イメージをビルドします。

```
glance image-create --name="<name of the OpenStack image>" --
    property hw_disk_bus=ide --is-public=true --container-format=
    bare --disk-format=qcow2< <name of the qcow2 file>

glance image-create --name="NS-VPX-12-0-26-2" --property
    hw_disk_bus=ide --is-public= true --container-format=bare --
    disk-format=qcow2< NSVPX-KVM-12.0-26.2_nc.qcow2
<!--NeedCopy-->
```

下図は、glance image-create コマンドの出力例です。


```
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 735dae4ea6e46e39ed3f0acfba02e755     |
| container_format | bare                                 |
| created_at       | 2017-02-16T10:03:29Z                 |
| disk_format      | qcow2                                |
| hw_disk_bus      | ide                                  |
| id               | aeaa13e9-b49b-411c-ab54-c61820a8e2f3 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | NSVPX-KVM-12.0-26.2                  |
| owner            | 06c41a73b32f4b48af55359fd7d3502c     |
| protected        | False                                |
| size             | 717946880                            |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2017-02-16T10:03:38Z                 |
| virtual_size     | None                                 |
| visibility       | private                              |
+------------------+--------------------------------------+
```

3. OpenStack イメージが作成されたら、Citrix ADC VPX インスタンスをプロビジョニングします。

```
1 nova boot --image NSVPX-KVM-12.0-26.2 --config-drive=true --
     userdata
2 ./userdata.txt --flavor m1. medium --nic net-id=3b258725-eaae-
3 455e-a5de-371d6d1f349f --nic port-id=218ba819-9f55-4991-adb6-
4 02086a6bdee2 NSVPX-10
5 <!--NeedCopy-->
```

前述のコマンドでは、userdata.txt は、VPX インスタンスの IP アドレス、ネットマスク、デフォルトゲートウェイなどの詳細を含むファイルです。ユーザーデータファイルは、ユーザーカスタマイズ可能なファイルです。NSVPX-KVM-12.0-26.2 は、プロビジョニングする仮想アプライアンスの名前です。—NIC port-id=218ba819-9f55-4991-adb6-02086a6bdee2 は OpenStack VF です。

次の図に、nova boot コマンドの出力例を示します。

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	
OS-EXT-SRV-ATTR:host	-
OS-EXT-SRV-ATTR:hypervisor_hostname	-
OS-EXT-SRV-ATTR:instance_name	instance-0000003c
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	-
OS-SRV-USG:terminated_at	-
accessIPv4	
accessIPv6	
adminPass	43EjPdM5shLz
config_drive	True
created	2017-02-20T11:53:37Z
flavor	m1.medium (3)
hostId	
id	6b9f6968-aab9-463c-b619-d58c73db3187
image	NSVPX-KVM-12.0-26.2 (a5478b8a-8435-48d1-b4a0-1494e2c8f8b1)
key_name	-
metadata	{}
name	NSVPX-10
os-extended-volumes:volumes_attached	[]
progress	0
security_groups	default
status	BUILD
tenant_id	06c41a73b32f4b48af55359fd7d3502c
updated	2017-02-20T11:53:38Z
user_id	418524f7101b4f0389ecbb36da9916b5

次の図は、userdata.txt ファイルのサンプルです。<PropertySection></PropertySection> タグ内の値は、ユーザーが設定可能な値で、IP アドレス、ネットマスク、デフォルトゲートウェイなどの情報を保持します。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1">
<PlatformSection>
<Kind>NOVA</Kind>
<Version>2013.1</Version>
<Vendor>Openstack</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
   />
<Property oe:key="com.citrix.netscaler.platform" oe:value="vpx"/>
citrix.com 4
<Property oe:key="com.citrix.netscaler.orch_env"
oe:value="openstack-orch-env"/>
```



```
<Property oe:key="com.citrix.netscaler.mgmt.ip"
oe:value="10.1.0.100"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask"
oe:value="255.255.0.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway"
oe:value="10.1.0.1"/>
</PropertySection>
</Environment>
<!--NeedCopy-->
```

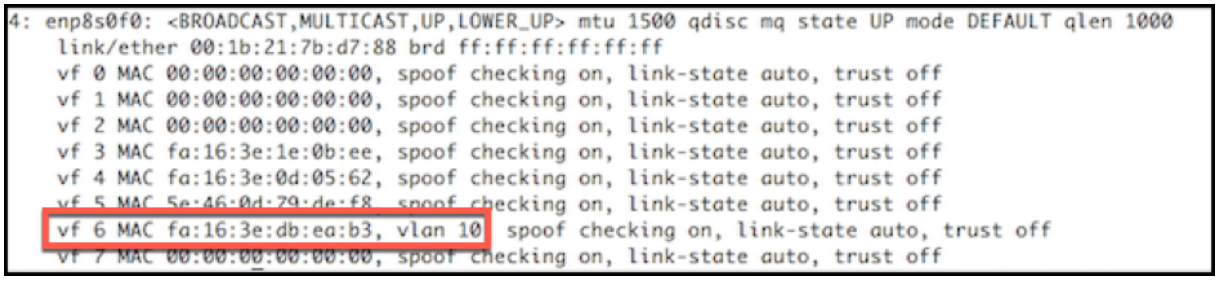
サポートされているその他の構成：ホストからの **SR-IOV VF** 上の **VLAN** の作成と削除

SR-IOV VF 上の VLAN を作成するには、次のコマンドを入力します。

```
ip link show enp8s0f0 vf 6 vlan 10
```

前述のコマンドでは、「enp8s0f0」は物理機能の名前です。

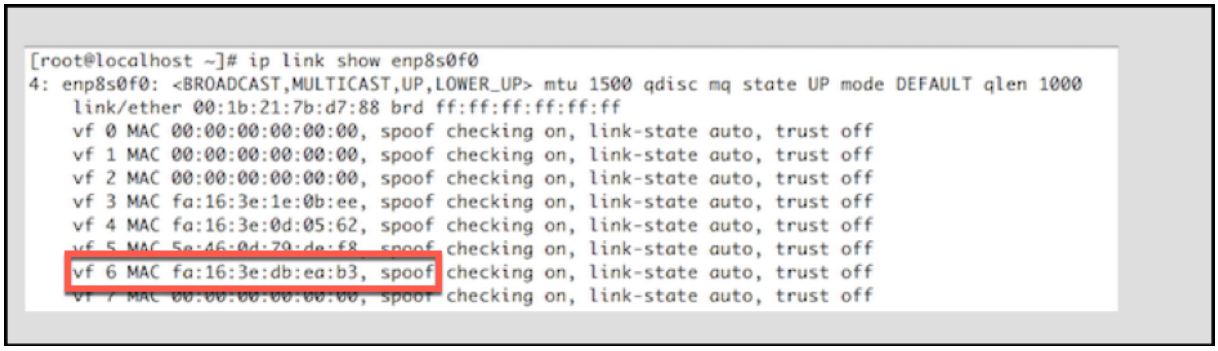
例：vf 6 で作成された VLAN 10



SR-IOV VF 上の VLAN を削除するには、次のコマンドを入力します。

```
ip link show enp8s0f0 vf 6 vlan 0
```

例：VLAN 10、vf 6 から削除された



以下の手順では、SRIOV テクノロジーを使用する Citrix ADC VPX インスタンスを OpenStack にデプロイする手順を完了します。

OVS DPDK ベースのホストインターフェイスを使用するように、KVM 上の Citrix ADC VPX インスタンスを構成する

October 7, 2021

KVM（Fedora および RHOS）で実行されている Citrix ADC VPX インスタンスを、データプレーン開発キット（DPDK）とともに Open vSwitch（OVS）を使用するように設定することで、ネットワークパフォーマンスを向上させることができます。このドキュメントでは、KVM ホスト上の OVS-DPDK によって公開される`vhost-user` ポートで動作するように Citrix ADC VPX インスタンスを構成する方法について説明します。

OVS は、オープンソースの Apache 2.0 ライセンスでライセンスされている多層仮想スイッチです。DPDK は、高速パケット処理のためのライブラリとドライバのセットです。

以下のバージョンの Fedora、RHOS、OVS、および DPDK は、Citrix ADC VPX インスタンスを設定するために認定されています。

Fedora	RHOS
Fedora 25	RHOS 7.4
OVS 2.7.0	OVS 2.6.1
DPDK 16.11.12	DPDK 16.11.12

前提条件

DPDK をインストールする前に、ホストに 1GB の巨大なページがあることを確認してください。

詳細については、この DPDK システム要件ドキュメントを参照してください。OVS DPDK ベースのホストインターフェイスを使用するように KVM で Citrix ADC VPX インスタンスを構成するために必要な手順の概要は次のとおりです。

- DPDK をインストールします。
- OVS を構築し、インストールします。
- OVS ブリッジを作成します。
- OVS ブリッジに物理インターフェイスを接続します。
- OVS データパスに`vhost-user`ポートを接続します。
- OVS-DPDK ベースの`vhost-user`ポートで KVM-VPX をプロビジョニングします

DPDK のインストール

DPDK をインストールするには、この Open vSwitch with DPDK ドキュメントに記載されている指示に従ってください。

OVS のビルドとインストール

OVS のダウンロードページから OVS をダウンロードします。次に、DPDK データパスを使用して OVS をビルドおよびインストールします。「Open vSwitch のインストール」ドキュメントに記載されている手順に従います。

詳細については、「DPDK 入門ガイド for Linux」を参照してください。

OVS ブリッジの作成

必要に応じて、Fedora コマンドか RHOS コマンドを入力して、OVS ブリッジを作成します。

Fedora コマンド:

```
> $OVS_DIR/utilities/ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0
    datapath_type=netdev
<!--NeedCopy-->
```

RHOS コマンド:

```
ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0 datapath_type=netdev
<!--NeedCopy-->
```

物理インターフェイスを OVS ブリッジに接続します

ポートを DPDK にバインドし、次の Fedora または RHOS コマンドを入力して OVS ブリッジにアタッチします。

Fedora コマンド:

```
> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface
    dpdk0 type=dpdk  options:dpdk-devargs=0000:03:00.0

> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface
    dpdk1 type=dpdk  options:dpdk-devargs=0000:03:00.1
<!--NeedCopy-->
```

RHOS コマンド:

```
ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface dpdk0 type=dpdk
    options:dpdk-devargs=0000:03:00.0
```

```
2
3
4 ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface dpdk1 type=dpdk
     options:dpdk-devargs=0000:03:00.1
5 <!--NeedCopy-->
```

オプションの一部として表示される dpdk-devargs は、それぞれの物理 NIC の PCI BDF を指定します。

OVS データパスに vhost-user ポートを接続する

OVS データパスにvhost-userポートを接続するには、次の Fedora または RHOS コマンドを入力します。

Fedora コマンド:

```
1 > $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user1 -- set
     Interface vhost-user1 type=dpdkvhostuser -- set Interface vhost-
     user1  mtu_request=9000
2
3 > $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user2 -- set
     Interface vhost-user2 type=dpdkvhostuser -- set Interface vhost-
     user2  mtu_request=9000
4
5 chmod g+w  /usr/local/var/run/openvswitch/vhost*
6 <!--NeedCopy-->
```

RHOS コマンド:

```
1 ovs-vsctl add-port ovs-br0 vhost-user1 -- set Interface vhost-user1
     type=dpdkvhostuser -- set Interface vhost-user1  mtu_request=9000
2
3 ovs-vsctl add-port ovs-br0 vhost-user2 -- set Interface vhost-user2
     type=dpdkvhostuser -- set Interface vhost-user2  mtu_request=9000
4
5 chmod g+w /var/run/openvswitch/vhost*
6 <!--NeedCopy-->
```

OVS-DPDK ベースの vhost-user ポートを持つ KVM-VPX のプロビジョニング

次の QEMU コマンドを使用して、CLI からのみ OVS-DPDK ベースのvhost-userポートを持つ Fedora KVM 上の VPX インスタンスをプロビジョニングできます。

Fedora コマンド:


```
qemu-system-x86_64 -name KVM-VPX -cpu host -enable-kvm -m 4096M \

-object memory-backend-file,id=mem,size=4096M,mem-path=/dev/hugepages,
   share=on -numa node,memdev=mem \

-mem-prealloc -smp sockets=1,cores=2 -drive file=<absolute-path-to-disc
   -image-file>,if=none,id=drive-ide0-0-0,format=<disc-image-format> \

-device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-0,id=ide0-0-0,
   bootindex=1 \

-netdev type=tap,id=hostnet0,script=no,downscript=no,vhost=on \

-device virtio-net-pci,netdev=hostnet0,id=net0,mac=52:54:00:3c:d1:ae,
   bus=pci.0,addr=0x3 \

-chardev socket,id=char0,path=</usr/local/var/run/openvswitch/vhost-
   user1> \

-netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce -device
   virtio-net-pci,mac=00:00:00:00:00:01,netdev=mynet1,mrg_rxbuf=on \

-chardev socket,id=char1,path=</usr/local/var/run/openvswitch/vhost-
   user2> \

-netdev type=vhost-user,id=mynet2,chardev=char1,vhostforce -device
   virtio-net

pci,mac=00:00:00:00:00:02,netdev=mynet2,mrg_rxbuf=on \

--nographic
<!--NeedCopy-->
```

RHOS の場合は、次のサンプル XML ファイルを使用して、virshを使用して Citrix ADC VPX インスタンスをプロビジョニングします。

```
<domain type='kvm'>

  <name>dpdk-vpx1</name>

  <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>

```



```
<memory unit='KiB'>16777216</memory>

<currentMemory unit='KiB'>16777216</currentMemory>

<memoryBacking>

  <hugepages>

    <page size='1048576' unit='KiB'/>

  </hugepages>

</memoryBacking>

<vcpu placement='static'>6</vcpu>

<cputune>

  <shares>4096</shares>

  <vcpupin vcpu='0' cpuset='0'/>

  <vcpupin vcpu='1' cpuset='2'/>

  <vcpupin vcpu='2' cpuset='4'/>

  <vcpupin vcpu='3' cpuset='6'/>

  <emulatorpin cpuset='0,2,4,6'/>

</cputune>

<numatune>

  <memory mode='strict' nodeset='0'/>

</numatune>

<resource>

  <partition>/machine</partition>

</resource>

<os>
```



```
52
53      <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
54
55      <boot dev='hd'/>
56
57    </os>
58
59    <features>
60
61      <acpi/>
62
63      <apic/>
64
65    </features>
66
67    <cpu mode='custom' match='minimum' check='full'>
68
69      <model fallback='allow'>Haswell-noTSX</model>
70
71      <vendor>Intel</vendor>
72
73      <topology sockets='1' cores='6' threads='1'/>
74
75      <feature policy='require' name='ss'/>
76
77      <feature policy='require' name='pcid'/>
78
79      <feature policy='require' name='hypervisor'/>
80
81      <feature policy='require' name='arat'/>
82
83  <domain type='kvm'>
84
85    <name>dpdk-vpx1</name>
86
87    <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>
88
89    <memory unit='KiB'>16777216</memory>
90
91    <currentMemory unit='KiB'>16777216</currentMemory>
92
93    <memoryBacking>
94
95      <hugepages>
96
```



```
97          <page size='1048576' unit='KiB'/>
98
99        </hugepages>
100
101     </memoryBacking>
102
103     <vcpu placement='static'>6</vcpu>
104
105     <cputune>
106
107       <shares>4096</shares>
108
109       <vcpupin vcpu='0' cpuset='0'/>
110
111       <vcpupin vcpu='1' cpuset='2'/>
112
113       <vcpupin vcpu='2' cpuset='4'/>
114
115       <vcpupin vcpu='3' cpuset='6'/>
116
117       <emulatorpin cpuset='0,2,4,6'/>
118
119     </cputune>
120
121     <numatune>
122
123       <memory mode='strict' nodeset='0'/>
124
125     </numatune>
126
127     <resource>
128
129       <partition>/machine</partition>
130
131     </resource>
132
133     <os>
134
135       <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
136
137       <boot dev='hd'/>
138
139     </os>
140
141     <features>
```



```
142
143       <acpi/>
144
145       <apic/>
146
147     </features>
148
149     <cpu mode='custom' match='minimum' check='full'>
150
151       <model fallback='allow'>Haswell-noTSX</model>
152
153       <vendor>Intel</vendor>
154
155       <topology sockets='1' cores='6' threads='1'/>
156
157       <feature policy='require' name='ss'/>
158
159       <feature policy='require' name='pcid'/>
160
161       <feature policy='require' name='hypervisor'/>
162
163       <feature policy='require' name='arat'/>
164
165       <feature policy='require' name='tsc_adjust'/>
166
167       <feature policy='require' name='xsaveopt'/>
168
169       <feature policy='require' name='pdpe1gb'/>
170
171       <numa>
172
173         <cell id='0' cpus='0-5' memory='16777216' unit='KiB' memAccess='
              shared'/>
174
175       </numa>
176
177     </cpu>
178
179     <clock offset='utc'/>
180
181     <on_poweroff>destroy</on_poweroff>
182
183     <on_reboot>restart</on_reboot>
184
185     <on_crash>destroy</on_crash>
```



```
186
187   <devices>
188
189     <emulator>/usr/libexec/qemu-kvm</emulator>
190
191     <disk type='file' device='disk'>
192
193       <driver name='qemu' type='qcow2' cache='none'/>
194
195       <source file='/home/NSVPX-KVM-12.0-52.18_nc.qcow2'/>
196
197       <target dev='vda' bus='virtio'/>
198
199       <address type='pci' domain='0x0000' bus='0x00' slot='0x07'
              function='0x0'/>
200
201     </disk>
202
203     <controller type='ide' index='0'>
204
205       <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
              function='0x1'/>
206
207     </controller>
208
209     <controller type='usb' index='0' model='piix3-uhci'>
210
211       <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
              function='0x2'/>
212
213     </controller>
214
215     <controller type='pci' index='0' model='pci-root'/>
216
217     <interface type='direct'>
218
219       <mac address='52:54:00:bb:ac:05'/>
220
221       <source dev='enp129s0f0' mode='bridge'/>
222
223       <model type='virtio'/>
224
225       <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
              function='0x0'/>
226
```



```
227     </interface>
228
229     <interface type='vhostuser'>
230
231       <mac address='52:54:00:55:55:56'/>
232
233       <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
            'client'/>
234
235       <model type='virtio'/>
236
237       <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
            function='0x0'/>
238
239     </interface>
240
241     <interface type='vhostuser'>
242
243       <mac address='52:54:00:2a:32:64'/>
244
245       <source type='unix' path='/var/run/openvswitch/vhost-user2' mode=
            'client'/>
246
247       <model type='virtio'/>
248
249       <address type='pci' domain='0x0000' bus='0x00' slot='0x05'
            function='0x0'/>
250
251     </interface>
252
253     <interface type='vhostuser'>
254
255       <mac address='52:54:00:2a:32:74'/>
256
257       <source type='unix' path='/var/run/openvswitch/vhost-user3' mode=
            'client'/>
258
259       <model type='virtio'/>
260
261       <address type='pci' domain='0x0000' bus='0x00' slot='0x06'
            function='0x0'/>
262
263     </interface>
264
265     <interface type='vhostuser'>
```



```
266
267         <mac address='52:54:00:2a:32:84'/>
268
269         <source type='unix' path='/var/run/openvswitch/vhost-user4' mode=
               'client'/>
270
271        <model type='virtio'/>
272
273         <address type='pci' domain='0x0000' bus='0x00' slot='0x09'
               function='0x0'/>
274
275       </interface>
276
277       <serial type='pty'>
278
279         <target port='0'/>
280
281       </serial>
282
283       <console type='pty'>
284
285         <target type='serial' port='0'/>
286
287       </console>
288
289       <input type='mouse' bus='ps2'/>
290
291       <input type='keyboard' bus='ps2'/>
292
293       <graphics type='vnc' port='-1' autoport='yes'>
294
295         <listen type='address'/>
296
297       </graphics>
298
299       <video>
300
301         <model type='cirrus' vram='16384' heads='1' primary='yes'/>
302
303         <address type='pci' domain='0x0000' bus='0x00' slot='0x02'
               function='0x0'/>
304
305       </video>
306
307       <memballoon model='virtio'>
```



```
308
309         <address type='pci' domain='0x0000' bus='0x00' slot='0x08'
               function='0x0'/>
310
311       </memballoon>
312
313     </devices>
314
315 </domain
316 <!--NeedCopy-->
```

注意事項

XML ファイルでは、サンプルファイルに示されているように、hugepage サイズは 1GB である必要があります。

```
1 <memoryBacking>
2
3     <hugepages>
4
5       <page size='1048576' unit='KiB'/>
6
7     </hugepages>
8 <!--NeedCopy-->
```

また、サンプルファイルでは、vhost-user1 は ovs-br0 にバインドされたvhostユーザーポートです。

```
1  <interface type='vhostuser'>
2
3        <mac address='52:54:00:55:55:56'/>
4
5        <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
            'client'/>
6
7        <model type='virtio'/>
8
9        <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
            function='0x0'/>
10
11     </interface>
12 <!--NeedCopy-->
```


Citrix ADC VPX インスタンスを起動するには、`virsh` コマンドの使用を開始します。

AWS での Citrix ADC VPX

December 7, 2021

Citrix ADC VPX インスタンスは、Amazon Web Services (AWS) で起動できます。Citrix ADC VPX アプライアンスは、AWS マーケットプレイスで Amazon Machine Image (AMI) として利用できます。AWS 上の Citrix ADC VPX インスタンスを使用すると、AWS クラウドコンピューティング機能を使用したり、Citrix ADC の負荷分散およびトラフィック管理機能をビジネスニーズに合わせて使用したりすることができます。VPX インスタンスは、物理 Citrix ADC アプライアンスのすべてのトラフィック管理機能をサポートし、スタンドアロンインスタンスまたは HA ペアで展開できます。VPX の機能の詳細については、VPX のデータシートを参照してください。

はじめに

VPX のデプロイを開始する前に、次の情報を理解しておく必要があります。

- AWS 用語
- AWS-VPX サポートマトリックス
- 制限事項と使用上のガイドライン
- 前提条件
- AWS での Citrix ADC VPX インスタンスの仕組み

AWS で Citrix ADC VPX インスタンスを展開する

AWS では、VPX インスタンスで次のデプロイタイプがサポートされています。

- スタンドアロン
- 高可用性 (アクティブ-パッシブ)
  - 同一ゾーン内での高可用性
  - Elastic IP を使用した異なるゾーンでの高可用性
  - プライベート IP を使用して、異なるゾーン間で高可用性
- アクティブ-アクティブ GSLB
- ADM を使用した自動スケーリング (アクティブ-アクティブ)

ハイブリッド展開

- AWS Outpost に Citrix ADC をデプロイする
- AWS の VMC に Citrix ADC をデプロイする

ライセンス

AWS 上の Citrix ADC VPX インスタンスにはライセンスが必要です。AWS で実行されている Citrix ADC VPX インスタンスでは、次のライセンスオプションを使用できます。

- 無料 (無制限)
- 毎時
- 年次
- BYOL
- 無料トライアル（AWS マーケットプレイスでは、すべての Citrix ADC VPX-AWS サブスクリプションを 21 日間無料で提供）

自動化

- Citrix ADM: スマートな展開
- AWS クイックスタート:AWS 上のウェブアプリケーション用 Citrix ADC VPX
- GitHub CFT: AWS デプロイ用の Citrix ADC テンプレートとスクリプト
- GitHub Ansible: AWS デプロイ用の Citrix ADC テンプレートとスクリプト
- GitHub Terraform: AWS デプロイ用の Citrix ADC テンプレートとスクリプト
- AWS パターンライブラリ (PL): Citrix ADC VPX

ブログ

- AWS 上の Citrix ADC がお客様のアプリケーションの安全な配信にどのように役立つのか
- Citrix ADC と AWS を使用したハイブリッドクラウドでのアプリケーションデリバリー
- Citrix は AWS ネットワーキングコンピテンシーパートナーです
- Citrix ADC: いつでもパブリッククラウドに対応
- Citrix ADC によるパブリッククラウドでのスケールアウトまたはスケールインが容易
- Citrix、AWS Outposts で ADC のデプロイメントの選択肢を拡大
- Citrix ADC と Amazon VPC イングレスルーティングの使用
- Citrix は、AWS での選択肢、パフォーマンス、シンプルなデプロイメントを提供します
- Citrix Web App Firewall のセキュリティ — 現在 AWS Marketplace で公開中
- Aria Systems が AWS で Citrix Web App Firewall を使用する方法

ビデオ

- ADM によるパブリッククラウドの Citrix ADC 導入の簡素化
- すぐに使用できる Terraform スクリプトを使用した、AWS での Citrix ADC VPX プロビジョニングと構成
- CloudFormation テンプレートを使用して AWS に Citrix ADC HA をデプロイする
- AWS クイックスタートを使用してアベイラビリティーゾーン全体に Citrix ADC HA
- AWS で Citrix ADC をデプロイする方法
- ADM を使用した Citrix ADC オートスケール
- Citrix ADC が AWS または AWS Autoscaling グループでバックエンドサーバーの自動スケーリングをサポート

お客様のケーススタディ

- テクノロジーソリューション-Xenit AB
- Citrix と AWS クラウドとのより良いビジネス方法 — Aria
- Citrix ADC と AWS の優位性をご覧ください
- Rain for Rent-お客様事件

解決方法

- Citrix ADC を使用して AWS にデジタル広告プラットフォームをデプロイする
- Citrix ADC を使用した AWS でのクリックストリーム分析の強化

サポート

- サポートケースを開く
- Citrix ADC サブスクリプションサービスについては、「AWS での VPX インスタンスのトラブルシューティング」を参照してください。サポートケースを登録するには、AWS アカウント番号とサポート PIN コードを見つけ、Citrix サポートにお問い合わせください。
- Citrix ADC カスタマーライセンス製品または BYOL については、有効なサポートおよびメンテナンス契約を結んでいることを確認してください。契約を結んでいない場合は、Citrix 担当者にお問い合わせください。

その他の参考資料

- AWS オンデマンドウェビナー-AWS 上の Citrix ADC
- AWS での Citrix ADC VPX のデプロイガイド

- SC2S/シークレットリージョンでの VPX Amazon Machine Image (AMI) の作成
- AWS 上の Citrix ADC
- Citrix ADC と AWS 検証済みのリファレンスデザイン
- Citrix ADC VPX のデータシート
- AWS Marketplace の Citrix ADC
- Citrix ADC は、AWS ネットワーキングパートナーソリューション（ロードバランサー）の一部です。
- AWS 上の VMware クラウド向け Citrix ADC
- AWS に関するよくある質問

AWS 用語

October 7, 2021

このセクションでは、よく使用される AWS の用語と語句のリストについて説明します。詳細については、「AWS 用語集」を参照してください。

用語	定義
Amazon マシンイメージ（AMI）	マシンイメージ。クラウド内の仮想サーバーであるインスタンスを起動するのに必要な情報を提供します。
Elastic Block Store	AWS クラウドで Amazon EC2 インスタンスと一緒に使用される、永続ブロックストレージボリュームを提供します。
Simple Storage Service（S3）	Internet 用のストレージ。Web 規模のコンピューティングを開発者が簡単に実施できるように設計されています。
Elastic Compute Cloud（EC2）	クラウドで、安全でサイズ変更できる処理能力を提供する Web サービスです。Web 規模のクラウドコンピューティングを開発者が簡単に実施できるように設計されています。
Elastic Load Balancing（ELB）	複数のアベイラビリティゾーンで、複数の EC2 インスタンスにまたがる受信アプリケーショントラフィックを分散します。これによってアプリケーションのフォールトトレランスが増加します。
エラスティックネットワークインターフェイス (ENI)	仮想プライベートクラウド（VPC）内のインスタンスにアタッチできる仮想ネットワークインターフェイス。

用語	定義
Elastic IP（EIP）アドレス	Amazon EC2 または Amazon VPC で割り当てられ、インスタンスにアタッチされた、静的パブリック IPv4 アドレスです。Elastic IP アドレスは特定のインスタンスではなく、お使いのアカウントに関連しています。ニーズの変化に応じて、割り当て、アタッチ、デタッチ、および解放が簡単にできるため、Elastic（融通が利く）と呼ばれています。
インスタンスの種類	Amazon EC2 では、さまざまなユースケースに対応できるよう最適化された幅広い種類のインスタンスを提供してます。インスタンスタイプを構成する CPU、メモリ、ストレージ、およびネットワーク機能の組み合わせはさまざまで、アプリケーションに合わせて最適なリソースの組み合わせを柔軟に選択できます。
Identity and Access Management（IAM）	AWS で ID が実行できること、または実行できないことを決定する許可ポリシーを持つ AWS の ID。IAM ロールを使うことで EC2 インスタンス上で実行されるアプリケーションが、AWS リソースに安全にアクセスできるようになります。高可用性セットアップで VPX インスタンスを展開する場合、IAM ロールは必須です。
インターネットゲートウェイ	ネットワークをインターネットに接続します。VPC 外部の IP アドレスのトラフィックをインターネットゲートウェイにルーティングできます。
キーペア	身元を電子的に証明するために使用する一連の資格情報。キーペアはプライベートキーとパブリックキーで構成されます。
ルートテーブル	関連付けられているサブネットからのトラフィックを制御するための一連のルーティング規則。1 つのルートテーブルに対して複数のサブネットを関連付けることができますが、各サブネットは一度に 1 つのルートテーブルにしか関連付けることができません。
セキュリティグループ	あるインスタンスに対して許可されている、名前が付けられた一連の受信方向のネットワーク接続。
サブネット	EC2 インスタンスをアタッチできる VPC の IP アドレス範囲の一部分。セキュリティと運用上の必要に応じて、サブネットを作成し、インスタンスをグループ分けできます。

用語	定義
Virtual Private Cloud（VPC）	定義した仮想ネットワーク内で AWS リソースを起動できる、AWS クラウドの論理的に隔離されたセクションをプロビジョニングする Web サービス。
Auto Scaling	ユーザー定義のポリシー、スケジュール、ヘルスチェックに基づいて Amazon EC2 インスタンスを自動的に起動または終了するウェブサービス。
クラウドの形成	関連する AWS リソースを 1 つの単位として一緒に作成および削除するテンプレートを書き込んだり変更したりするサービス。

VPX-AWS サポートマトリックス

December 7, 2021

次の表に、サポートされている VPX モデルと AWS リージョン、インスタンスタイプ、およびサービスを示します。

表 1: AWS でサポートされている VPX モデル

サポートされている VPX モデル
Citrix ADC VPX スタンダード/アドバンスト/プレミアムエディション-200 Mbps
Citrix ADC VPX スタンダード/アドバンスト/プレミアムエディション-1000 Mbps
Citrix ADC VPX スタンダード/アドバンスト/プレミアムエディション-3 Gbps
Citrix ADC VPX スタンダード/アドバンスト/プレミアムエディション-5 Gbps
Citrix ADC VPX スタンダード/アドバンスト/プレミアム-10 Mbps
Citrix ADC VPX エクスプレス-20 Mbps
Citrix ADC VPX-カスタマーライセンス
Citrix ADC（旧 NetScaler）VPX FIPS-カスタマーライセンス

表: サポートされている 2 つの AWS リージョン

サポートされている AWS リージョン
米国西部 (オレゴン) リージョン

サポートされている AWS リージョン
米国西部 (北カリフォルニア) リージョン
米国東部 (オハイオ) リージョン
米国東部 (バージニア北部) リージョン
アジア太平洋 (ムンバイ) リージョン
アジア太平洋 (ソウル) リージョン
カナダ (中部) リージョン
アジアパシフィック (シンガポール) リージョン
アジア太平洋 (シドニー) リージョン
アジア太平洋 (東京) リージョン
アジアパシフィック (香港) リージョン
カナダ (中部) リージョン
中国 (北京) リージョン
中国 (寧夏) 地域
欧州 (フランクフルト) 地域
欧州 (アイルランド) リージョン
欧州 (ロンドン) リージョン
欧州 (パリ) リージョン
欧州 (ミラノ) リージョン
南米 (サンパウロ) リージョン
AWS GovCloud (米国東部) リージョン
AWS GovCloud (米国西部) リージョン
AWS トップシークレット (C2S) リージョン
中東 (バーレーン) リージョン
アフリカ (ケープタウン)
C2S

表 3: サポートされている AWS インスタンスタイプ

サポートされる AWS インスタンスタイプ
t2.medium, t2.large, t2.x large, t2.2x large
m3.large, m3.x large, m3.2x large
c4.large, c4.large, c4.2x large, c4.4x large, c4.8x large
m4.large, m4.large, m4.2x large, m4.4x large, m4.10x large
m5.large, m5.x large, m5.2x large, m5.4x large, m5.12x large, m5.24x large
c5.large, c5.x large, c5.2x large, c5.4x large, c5.9x large, c5.18x large, c5.24x large
C5n.large, C5n.x large, C5n.2x large, C5n.4x large, C5n.9x large, C5n.18x large
D2.x large, D2.2x large, D2.4x large, D2.8x large
m5a.large, m5a.xlarge, m5a.2xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge

表 4: サポートされる AWS サービス

サポートされている AWS サービス
EC2: ADC インスタンスを起動します。
ラムダ： CFT からの Citrix ADC VPX インスタンスのプロビジョニング中に、Citrix ADC VPX NITRO API を呼び出します。
VPC と **VPC** イングレスルーティング**:** VPC は、ADC を起動できる分離されたネットワークを作成します。VPC 入力ルーティングは、ファイアウォールの負荷分散ソリューションで使用されます。
Route53： Citrix ADC Autoscale ソリューション内のすべての ADC VPX ノードにトラフィックを分散します。
ELB： Citrix ADC Autoscale ソリューション内のすべての ADC VPX ノードにトラフィックを分散します。
Cloudwatch： Citrix ADC VPX インスタンスのパフォーマンスとシステムパラメーターを監視します。
AWS Autoscaling: バックエンドサーバーの自動スケーリングに使用されます。
クラウドの形成： CloudFormation テンプレートは、Citrix ADC VPX インスタンスをデプロイするために使用されます。
Simple Queue Service (SQS): バックエンド自動スケーリングでスケールアップおよびスケールダウンイベントを監視します。
簡易通知サービス **(SNS):** バックエンド自動スケーリングでスケールアップおよびスケールダウンイベントを監視します。
ID とアクセス管理 **(IAM):** AWS のサービスとリソースへのアクセスを提供します。

サポートされている AWS サービス
AWS Outposts: AWS Outposts で Citrix ADC VPX インスタンスをプロビジョニングします。

Citrix では、次の AWS インスタンスタイプを推奨します。

- マーケットプレイスエディションまたは帯域幅ベースのプールライセンス用の M5 および C5n シリーズ
- vCPU ベースのプールライセンス用の C5n シリーズ

AWS マーケットプレイスでの VPX オファリング	AWS インスタンスの推奨事項
VPX 10、VPX エクスプレス 20、VPX 200	M5.xLarge
VPX 1000、VPX 3G、VPX 5G	M5.2xLarge

Citrix では、スループットに基づいて以下の AWS インスタンスタイプを推奨します。

プールライセンス付き VPX（帯域幅ライセンス）	AWS インスタンスの推奨事項
VPX 8G	C5n.4xLarge
VPX 10G、VPX 15G、VPX 25G	C5n.9xLarge

> 注:
>
> VPX 25G サービスでは、AWS で希望する 25G のスループットは得られませんが、SSL トランザクションレートが高くなる可能性があります。

5G を超えるスループットを実現するには、次の手順を実行します。

- AWS マーケットプレイスで **Citrix ADC VPX-**カスタマーライセンス **(BYOL)** オファリングを選択します。
- Citrix ADC GUI または CLI で［プールライセンス（帯域幅ライセンス）］を選択します。

1 秒あたりのパケット数、SSL トランザクションレートなどのさまざまなメトリックに基づいてインスタンスを判断するには、Citrix 連絡先に連絡してガイダンスを求めてください。vCPU ベースのプールライセンスとサイジングに関するガイダンスについては、Citrix サポートにお問い合わせください。

制限事項と使用上のガイドライン

October 7, 2021

Citrix ADC VPX インスタンスを AWS にデプロイする際には、以下の制限事項と使用上のガイドラインが適用されます。

- 開始する前に、AWS での Citrix ADC VPX インスタンスのデプロイの AWS 用語のセクションをお読みください。
- クラスタリング機能は、VPX ではサポートされていません。
- 高可用性セットアップを効果的に機能させるには、専用の NAT デバイスを管理インターフェイスに関連付けるか、EIP を NSIP に関連付けます。NAT について詳しくは、AWS ドキュメントの「NAT Instances」を参照してください。
- データトラフィックおよび管理トラフィックは、異なるサブネットに属する ENI で分離する必要があります。
- 管理 ENI には NSIP アドレスのみが存在する必要があります。
- セキュリティ上の理由により、EIP を NSIP に関連付ける代わりに NAT インスタンスを使用する場合は、VPC レベルでルーティングを適切に変更する必要があります。VPC レベルのルーティングの変更手順については、AWS ドキュメントの「シナリオ 2: パブリックサブネットとプライベートサブネットを持つ VPC」を参照してください。
- VPX インスタンスは、ある EC2 インスタンスタイプから別のインスタンスタイプへ（たとえば、m3.large から m3.xlarge へ）移動できます。
- AWS 上の VPX のストレージオプションについては、EBS は耐久性があり、インスタンスからデタッチした後でもデータが利用可能になるため、EBS をお勧めします。
- VPX への ENI の動的追加はサポートされていません。VPX インスタンスを再起動して更新を適用します。スタンドアロンインスタンスまたは HA インスタンスを停止し、新しい ENI を接続してからインスタンスを再起動することをお勧めします。
- 1 つの ENI に複数の IP アドレスを割り当てることができます。ENI あたりの IP アドレスの最大数は EC2 インスタンスタイプによって決まります。Elastic Network Interfacesの「インスタンスタイプごとのネットワークインターフェイスごとの IP アドレス」のセクションを参照してください。IP アドレスを ENI に割り当てる前に、AWS で割り当てる必要があります。詳細については、「Elastic ネットワークインターフェイス」を参照してください。
- Citrix ADC VPX インターフェイスでは、インターフェイスの有効化および無効化コマンドは使用しないことをお勧めします。
- Citrix ADC`set ha node \<NODE\_ID\> -haStatus STAYPRIMARY` と`set ha node \<NODE\_ID\> -haStatus STAYSECONDARY` コマンドはデフォルトで無効になっています。
- IPv6 は VPX ではサポートされていません。
- AWS の制限により、次の機能はサポートされていません。
  - GARP（Gratuitous ARP）

  - L2 モード
  - タグ付き VLAN
  - 動的ルーティング
  - 仮想 MAC

- RNAT が機能するには、送信元/宛先チェックが無効になっていることを確認します。詳細については、Elastic Network Interfacesの「ソース/デスティネーションチェックの変更」を参照してください。
- AWS での Citrix ADC VPX デプロイメントでは、一部の AWS リージョンで AWS インフラストラクチャが AWS API 呼び出しを解決できない場合があります。これは、Citrix ADC VPX インスタンスの非管理インターフェイスを介して API 呼び出しが発行された場合に発生します。
  回避策として、API 呼び出しを管理インターフェイスにのみ制限してください。これを行うには、VPX インスタンスに NSVLAN を作成し、適切なコマンドを使用して管理インターフェイスを NSVLAN にバインドします。
  例:
  ```
  set ns config -nsvlan <vlan id> -ifnum 1/1 -tagged NO
  save config
  ```
  プロンプトで VPX インスタンスを再起動します。nsvlanの設定の詳細については、NSVLAN の設定を参照してください。
- AWS コンソールでは、実際の使用量がはるかに低い場合でも、[**Monitoring**] タブの下に表示される VPX インスタンスの vCPU 使用率が高い（最大 100%）ことがあります。実際の vCPU 使用率を確認するには、**[**すべての **CloudWatch** メトリックスを表示**]** に移動します。詳細については、「Amazon CloudWatch を使用してインスタンスを監視する」を参照してください。

前提条件

January 25, 2022

AWS で VPX インスタンスを作成する前に、次のものがあることを確認してください。

- **AWS** アカウント:AWS 仮想プライベートクラウド (VPC) で Citrix ADC VPX AMI を起動します。AWS アカウントは www.aws.amazon.com で無料で作成できます。
- **AWS ID** およびアクセス管理（**IAM**）ユーザーアカウント: ユーザーの AWS サービスおよびリソースへのアクセスを安全にコントロールします。IAM ユーザーアカウントの作成方法の詳細については、「IAM ユーザーの作成 (コンソール)」を参照してください。IAM ロールは、スタンドアロンデプロイと高可用性デプロイの両方で必須です。

  AWS アカウントに関連付けられた IAM ロールには、さまざまなシナリオで次の IAM アクセス権限が必要です。

  同じ **AWS** ゾーン内の **HA** ペア**:**


```
”ec2:DescribeInstances”,
”ec2:AssignPrivateIpAddresses”,
”iam:SimulatePrincipalPolicy”,
”iam:GetRole”
<!--NeedCopy-->
```

異なる **AWS** ゾーンにまたがる **Elastic IP** アドレスを持つ **HA**

```
”ec2:DescribeInstances”,
”ec2:DescribeAddresses”,
”ec2:AssociateAddress”,
”ec2:DisassociateAddress”,
”iam:SimulatePrincipalPolicy”,
”iam:GetRole”
<!--NeedCopy-->
```

異なる **AWS** ゾーンのプライベート **IP** アドレスを持つ **HA** ペア**:**

```
”ec2:DescribeInstances”,
”ec2:DescribeRouteTables”,
”ec2:DeleteRoute”,
”ec2:CreateRoute”,
”ec2:ModifyNetworkInterfaceAttribute”,
”iam:SimulatePrincipalPolicy”,
”iam:GetRole”
<!--NeedCopy-->
```

異なる **AWS** ゾーンにわたるプライベート **IP** アドレスと **Elastic IP** アドレスの両方を持つ **HA** ペア**:**

```
”ec2:DescribeInstances”,
”ec2:DescribeAddresses”,
”ec2:AssociateAddress”,
”ec2:DisassociateAddress”,
”ec2:DescribeRouteTables”,
”ec2:DeleteRoute”,
”ec2:CreateRoute”,
”ec2:ModifyNetworkInterfaceAttribute”,
”iam:SimulatePrincipalPolicy”,
”iam:GetRole”
```



```
<!--NeedCopy-->
```

AWS バックエンドの自動スケーリング**:**

```
"ec2:DescribeInstances",
"autoscaling:*",
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:ListTopics",
"sns:Subscribe",
"sqs:CreateQueue",
"sqs:ListQueues",
"sqs:DeleteMessage",
"sqs:GetQueueAttributes",
"sqs:SetQueueAttributes",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
<!--NeedCopy-->
```

> 注:
>
> - 前述の機能を組み合わせて使用する場合は、各機能に IAM アクセス権限を組み合わせて使用します。
> - Citrix CloudFormation テンプレートを使用すると、IAM ロールが自動的に作成されます。このテンプレートでは、作成済みの IAM ロールを選択することはできません。
> - GUI から VPX インスタンスにログオンすると、IAM ロールに必要な権限を設定するよう求めるプロンプトが表示されます。権限をすでに構成している場合は、このプロンプトを無視してください。

- **AWS CLI**: ターミナルプログラムから AWS マネジメントコンソールが提供するすべての機能を使用する。詳細については、AWS CLI ユーザーガイドを参照してください。また、ネットワークインターフェイスの種類を SR-IOV に変更するには、AWS CLI も必要です。
- **Elastic Network Adapter (ENA)**: M5、C5 インスタンスなどの ENA ドライバー対応インスタンスタイプの場合、ファームウェアバージョンは 13.0 以降である必要があります。

AWS での Citrix ADC VPX インスタンスのしくみ

October 7, 2021

Citrix ADC VPX インスタンスは、AWS マーケットプレイスで AMI として使用でき、AWS VPC 内で EC2 インスタンスとして起動できます。Citrix ADC VPX AMI インスタンスには、少なくとも 2 つの仮想 CPU と 2 GB のメモリが

必要です。また、AWS VPC 内で起動される EC2 インスタンスは、複数のインターフェイス、インターフェイスごとに複数の IP アドレス、VPX 構成に必要なパブリックおよびプライベート IP アドレスも提供できます。各 VPX インスタンスには、少なくとも 3 つの IP サブネットが必要です。

- 管理サブネット
- クライアント側サブネット（VIP）
- バックエンド側サブネット（SNIP、MIP など）

AWS での標準の VPX インスタンスのインストールには、3 つのネットワークインターフェイスをお勧めします。

現在、AWS では、AWS VPC 内で実行しているインスタンスでのみ、マルチ IP 機能を使用できます。VPC 内の VPX インスタンスを使用して、EC2 インスタンスで実行しているサーバーの負荷を分散できます。Amazon VPC を使用すれば、独自の IP アドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイなどを含めて、仮想ネットワーク環境を作成および管理できます。

注: デフォルトでは、AWS アカウントごとに AWS リージョンごとに最大 5 つの VPC インスタンスを作成できます。Amazon のリクエストフォームを送信することで、より高い VPC 制限をリクエストhttp://aws.amazon.com/contact-us/vpc-requestできます。

図 **1**: AWS アーキテクチャでの Citrix ADC VPX インスタンスのデプロイメントの例

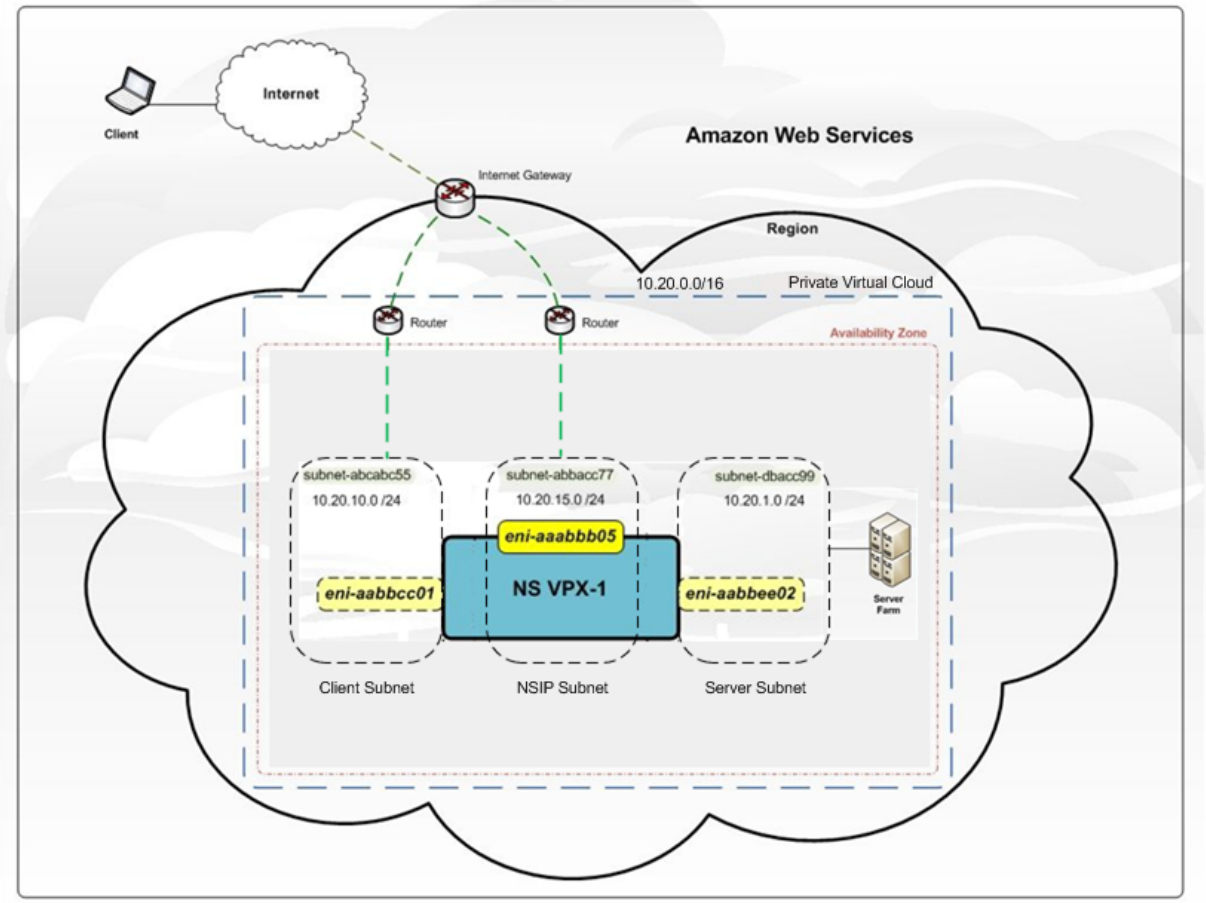


図 1 は、

Citrix ADC VPX デプロイメントを使用した AWS VPC の簡単なトポロジを示しています。AWS VPC は、以下の要素で構成されています。

1. VPC からの送受信トラフィックをルーティングするための単一のインターネットゲートウェイ。
2. インターネットゲートウェイとインターネット間のネットワーク接続。
3. 3 つのサブネット（管理、クライアント、サーバー用に 1 つずつ）。
4. インターネットゲートウェイと 2 つのサブネット（管理用とクライアント用）間のネットワーク接続。
5. VPC 内にデプロイされたスタンドアロンの Citrix ADC VPX インスタンス。VPX インスタンスには、各サブネットに 1 つずつ接続された ENI が 3 つあります。

Citrix ADC VPX スタンドアロンインスタンスを AWS にデプロイする

October 7, 2021

以下のオプションを使用して、Citrix ADC VPX スタンドアロンインスタンスを AWS にデプロイできます。

- AWS ウェブコンソール
- Citrix が作成した CloudFormation テンプレート
- AWS CLI

このトピックでは、Citrix ADC VPX インスタンスを AWS にデプロイする手順について説明します。

配置を開始する前に、次のトピックをお読みください。

- 前提条件
- 制限と使用上のガイドライン

AWS ウェブコンソールを使用して、Citrix ADC VPX インスタンスを AWS にデプロイする

AWS ウェブコンソールを使用して、Citrix ADC VPX インスタンスを AWS にデプロイできます。展開のプロセスには、次の手順が含まれます。

1. キーペアの作成
2. 仮想プライベートクラウド（VPC）の作成
3. サブネットの追加
4. セキュリティグループとセキュリティルールの作成
5. ルートテーブルの追加
6. インターネット Gateway の作成
7. Citrix ADC VPX インスタンスの作成
8. より多くのネットワークインターフェイスを作成してアタッチする
9. エラスティック IP の管理 NIC へのアタッチ
10. VPX インスタンスに接続する

ステップ **1:** キーペアを作成します。

Amazon EC2 は、キーペアを使用してログオン情報を暗号化および復号化します。インスタンスにログオンするには、キーペアを作成し、インスタンスを起動するときにキーペアの名前を指定し、インスタンスに接続するときにプライベートキーを指定する必要があります。

AWS Launch Instance ウィザードを使用してインスタンスを確認し、起動すると、既存のキーペアを使用するか、新しいキーペアを作成するように求められます。キーペアの作成方法の詳細については、「Amazon EC2 キーペア」を参照してください。

ステップ **2: VPC** を作成します。

Citrix ADC VPC インスタンスは AWS VPC 内にデプロイされます。VPC では、AWS アカウント専用の仮想ネットワークを定義できます。AWS VPC の詳細については、「Amazon VPC の使用開始」を参照してください。

Citrix ADC VPX インスタンス用の VPC を作成する際には、次の点に注意してください。

- AWS アベイラビリティーゾーンに AWSVPC を作成するには、単一のパブリックサブネットのみのオプションで VPC を使用します。
- 以下の サブネットを少なくとも **3** つ作成することをお勧めします。
  - 管理トラフィック用の 1 つのサブネット。管理 IP（NSIP）をこのサブネットに配置します。デフォルトでは、エラスティックネットワークインターフェイス（ENI）eth0 が管理 IP に使用されます。
  - クライアントアクセス（ユーザーから Citrix ADC VPX）トラフィック用の 1 つ以上のサブネット。クライアントが Citrix ADC 負荷分散仮想サーバーに割り当てられた 1 つ以上の仮想 IP（VIP）アドレスに接続します。
  - サーバーが VPX 所有のサブネット IP（SNIP）アドレスに接続するサーバーアクセス（VPX からサーバー）トラフィック用の 1 つ以上のサブネット。Citrix ADC 負荷分散と仮想サーバー、仮想 IP アドレス（VIP）、サブネット IP アドレス（SNIP）の詳細については、以下を参照してください。
  - すべてのサブネットは同じアベイラビリティーゾーンに存在する必要があります。

ステップ **3:** サブネットを追加します。

VPC ウィザードを使った場合、作成されたサブネットは 1 つのみです。要件に応じて、さらにサブネットを作成することもできます。サブネットをさらに作成する方法の詳細については、「VPC へのサブネットの追加」を参照してください。

ステップ **4:** セキュリティグループとセキュリティルールを作成します。

受信トラフィックと送信トラフィックを制御するには、セキュリティグループを作成し、そのグループに規則を追加します。グループを作成してルールを追加する方法の詳細については、「VPC のセキュリティグループ」を参照してください。

Citrix ADC VPX インスタンスの場合、EC2 ウィザードはデフォルトのセキュリティグループを提供します。このセキュリティグループは、AWS マーケットプレイスによって生成され、Citrix が推奨する設定に基づいています。ただし、要件に応じてさらにセキュリティグループを作成できます。

> 注
>
> ポート 22、80、443 は、SSH、HTTP、HTTPS アクセス用にセキュリティグループで開かれます。

ステップ **5:** ルートテーブルを追加します。

ルートテーブルには、ネットワークトラフィックの経路を判断する際に使用される、ルートと呼ばれる一連のルールが含まれます。VPC の各サブネットはルートテーブルに関連付ける必要があります。ルートテーブルの作成方法の詳細については、「ルートテーブル」を参照してください。

ステップ **6**: インターネット **Gateway** を作成します。

インターネット Gateway は、インターネットでルーティング可能なトラフィック用に VPC ルートテーブルでターゲットを提供し、パブリック IPv4 アドレスが割り当てられたインスタンスに対してネットワークアドレス変換（NAT）を実行するという 2 つの目的を果たします。

インターネットトラフィックに対して、インターネットゲートウェイを作成します。インターネットゲートウェイの作成方法の詳細については、「インターネットゲートウェイをアタッチする」を参照してください。

ステップ **7**: **AWS EC2** サービスを使用して **Citrix ADC VPX** インスタンスを作成します。

AWS EC2 サービスを使用して Citrix ADC VPX インスタンスを作成するには、以下の手順を実行します。

1. AWS ダッシュボードから、**[**コンピューティング**] > [EC2] > [**インスタンスの起動**] > [AWS** マーケットプレイス] に移動します。

   [**Launch Instance**] をクリックする前に、[Launch **Instance**] の下に表示されるメモをチェックして、リージョンが正しいことを確認します。

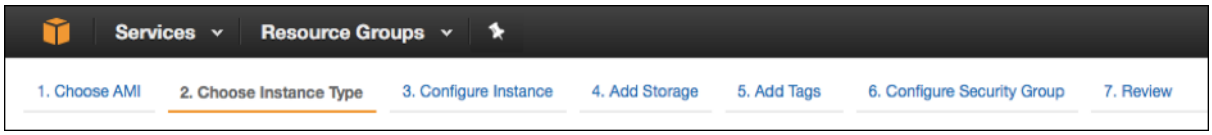
   

2. AWS マーケットプレイスの検索バーで、キーワード Citrix ADC VPX で検索します。
3. 展開するバージョンを選択し、[**Select**] をクリックします。Citrix ADC VPX バージョンでは、次のオプションがあります。

   - ライセンスバージョン
   - Citrix ADC VPX Express アプライアンス（これは、無料の仮想アプライアンスです, これは、Citrix ADC から提供されています 12.0 56.20.）
   - 自分のデバイスを持参

Launch Instance ウィザードが起動します。ウィザードに従って、インスタンスを作成します。ウィザードにより、次の操作が求められます。

- インスタンスの種類の選択
- インスタンスの構成
- ストレージの追加
- タグの追加
- セキュリティグループの構成
- 確認



ステップ **8**: より多くのネットワークインターフェイスを作成し、アタッチします。

VIP と SNIP 用に 2 つのネットワークインターフェイスを作成します。ネットワークインターフェイスの作成方法の詳細については、「ネットワークインターフェイスの作成」を参照してください。

ネットワークインターフェイスを作成したら、VPX インスタンスにアタッチする必要があります。インターフェイスを接続する前に、VPX インスタンスをシャットダウンし、インターフェイスを接続し、インスタンスの電源をオンにします。ネットワークインターフェイスの接続方法の詳細については、「インスタンスの起動時にネットワークインターフェイスをアタッチする」セクションを参照してください。

ステップ **9: Elastic IP** を割り当てて関連付けます。

EC2 インスタンスにパブリック IP アドレスを割り当てた場合、インスタンスが停止するまで割り当てが維持されます。その後、アドレスはプールに解放されます。インスタンスを再起動すると、新しいパブリック IP アドレスが割り当てられます。

対照的に、エラスティック IP（EIP）アドレスの場合は、インスタンスから割り当てが解除されるまで割り当ての状態が維持されます。

管理 NIC のエラスティック IP を割り当てて、関連付けます。Elastic IP アドレスを割り当てて関連付ける方法の詳細については、次のトピックを参照してください。

- Elastic IP アドレスの割り当て
- 実行中のインスタンスへの Elastic IP アドレスの関連付け

これらのステップは、AWS で Citrix ADC VPX インスタンスを作成する手順を完了します。インスタンスの準備が完了するまで数分かかる場合があります。インスタンスのステータスチェックに合格したことを確認します。この情報は、[Instances] ページの [**Status Check** s] 列で表示できます。

ステップ **10**: **VPX** インスタンスに接続します。

VPX インスタンスを作成したら、GUI と SSH クライアントを使用してインスタンスを接続します。

- GUI

Citrix ADC VPX インスタンスにアクセスするためのデフォルトの管理者資格情報は次のとおりです。

ユーザー名: `nsroot`

パスワード: ns ルートアカウントのデフォルトのパスワードは、Citrix ADC VPX インスタンスの AWS インスタンス ID に設定されます。最初のログオン時に、セキュリティ上の理由からパスワードを変更するように求められます。パスワードを変更した後、構成を保存する必要があります。構成が保存されずにインスタンスが再起動する場合は、デフォルトのパスワードでログオンする必要があります。プロンプトでパスワードを再度変更します。

- SSH クライアント

AWS マネジメントコンソールで、Citrix ADC VPX インスタンスを選択し、[接続] をクリックします。[インスタンスへの接続] ページの指示に従います。

AWS ウェブコンソールを使用して Citrix ADC VPX スタンドアロンインスタンスを AWS にデプロイする方法の詳細については、以下を参照してください。

- シナリオ: スタンドアロンインスタンス
- Citrix CloudFormation テンプレートを使用して AWS 上で Citrix NetScaler VPX インスタンスを構成する方法

Citrix CloudFormation テンプレートを使用して Citrix ADC VPX インスタンスを構成する

Citrix が提供する CloudFormation テンプレートを使用して、VPX インスタンスの起動を自動化できます。このテンプレートは、単一の Citrix ADC VPX インスタンスを起動したり、一対の Citrix ADC VPX インスタンスを使用して高可用性環境を作成したりする機能を提供します。

テンプレートは、AWS マーケットプレイスまたは GitHub から起動できます。

CloudFormation テンプレートには既存の VPC 環境が必要で、3 つのエラスティックネットワークインターフェイス（ENI）を使用して VPX インスタンスを起動します。CloudFormation テンプレートを開始する前に、次の要件を満たしていることを確認してください。

- AWS 仮想プライベートクラウド（VPC）
- VPC 内の 3 つのサブネット（管理用、クライアントトラフィック用、バックエンドサーバー用）
- インスタンスへの SSH アクセスを有効にする EC2 キーペア
- UDP 3003、TCP 3009 ～3010、HTTP、SSH ポートが開いているセキュリティグループ

前提条件を満たす方法の詳細については、「AWS ウェブコンソールを使用して AWS に Citrix ADC VPX インスタンスをデプロイする」セクションまたは AWS のドキュメントを参照してください。

このビデオでは 、AWS Marketplace で利用可能な Citrix CloudFormation テンプレートを使用して、Citrix ADC VPX スタンドアロンインスタンスを構成して起動する方法について説明します。

さらに、GitHub で利用可能な Citrix CloudFormation テンプレートを使用して、Citrix ADC VPX Express のスタンドアロンインスタンスを構成して起動します。

https://github.com/citrix/citrix-adc-aws-cloudformation/tree/master/templates/standalone/

IAM ロールは、スタンドアロンデプロイでは必須ではありません。ただし、今後の必要性のために、必要な権限を持つ IAM ロールを作成してインスタンスにアタッチすることをお勧めします。IAM ロールにより、スタンドアロンインスタンスは、必要に応じて SR-IOV を使用して高可用性ノードに簡単に変換されます。

必要な権限の詳細については、「SR-IOV ネットワークインターフェイスを使用するための Citrix ADC VPX インスタンスの構成」を参照してください。

> 注:
> AWS ウェブコンソールを使用して Citrix ADC VPX インスタンスを AWS にデプロイする場合、CloudWatch サービスはデフォルトで有効になります。Citrix CloudFormation テンプレートを使用して Citrix ADC VPX インスタンスを展開する場合、デフォルトのオプションは「はい」です。CloudWatch サービスを無効にする場合は、[いいえ] を選択します。詳細については、「Amazon CloudWatch を使用したインスタンスのモニタリング」を参照してください。

AWS CLI を使用して **Citrix ADC VPX** インスタンスを構成する

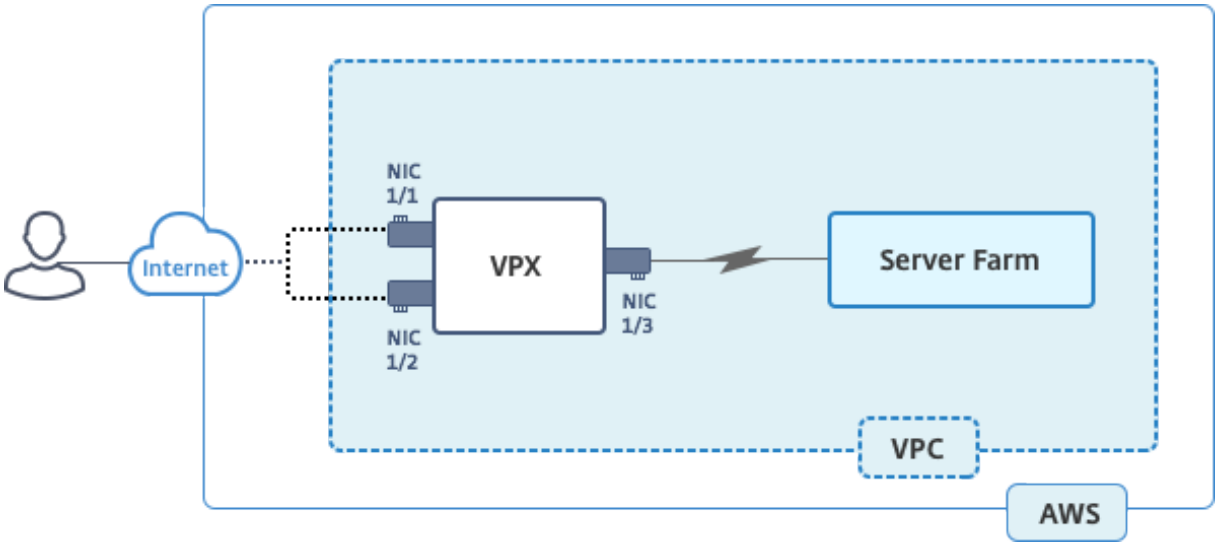
AWS CLI を使用してインスタンスを起動できます。詳細については、AWS コマンドラインインターフェイスのドキュメントを参照してください。

シナリオ: スタンドアロンインスタンス

October 7, 2021

このシナリオでは、AWS GUI を使用して、Citrix ADC VPX スタンドアロン EC2 インスタンスを AWS にデプロイする方法を示します。3 つの NIC を持つスタンドアロン VPX インスタンスを作成します。インスタンスは、負荷分散仮想サーバーとして構成されており、バックエンドサーバー（サーバーファーム）と通信します。この設定では、インスタンスとバックエンドサーバー間、およびパブリックインターネット上のインスタンスと外部ホスト間の必要な通信ルートを設定します。

VPX インスタンスをデプロイする手順の詳細については、「Citrix ADC VPX スタンドアロンインスタンスを AWS にデプロイする」を参照してください。

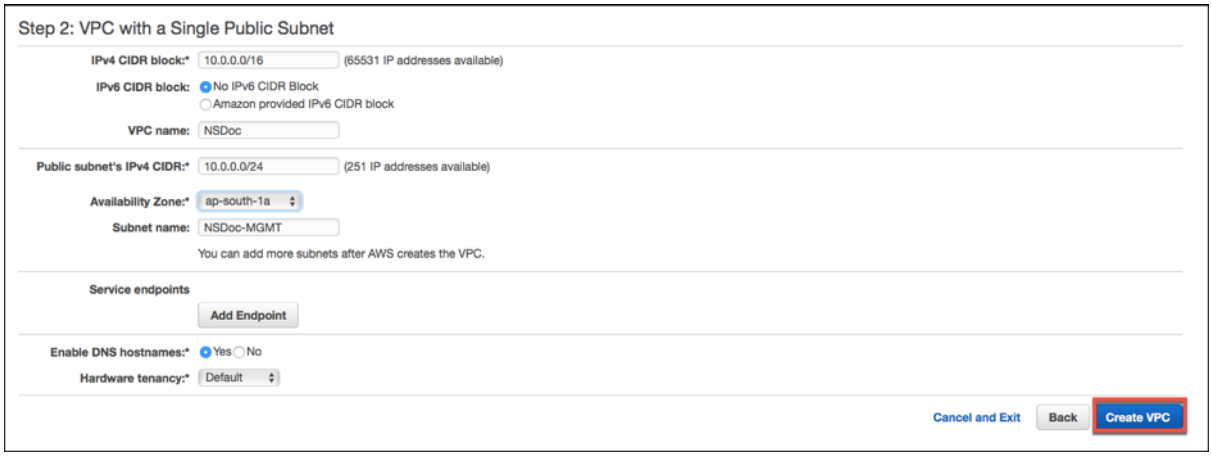


3 つの NIC を作成します。各 NIC は、IP アドレスのペア（パブリックとプライベート）を使用して構成できます。NIC は、次の目的に役立ちます。

NIC	目的	関連付けられている
eth0	NSIP（管理トラフィックを処理する）	パブリック IP アドレスとプライベート IP アドレス
eth1	クライアント側のトラフィック（VIP）をサービスする	パブリック IP アドレスとプライベート IP アドレス
eth2	バックエンド・サーバ（SNIP）との通信	パブリック IP アドレス (プライベート IP アドレスは必須ではありません)

ステップ 1: VPC を作成します。

1. AWS ウェブコンソールにログオンし、[ネットワークとコンテンツ配信] > [**VPC**] に移動します。[**VPC** ウィザードの開始**]** をクリックします。
2. 単一のパブリックサブネットを持つ **VPC** を選択し、[**Select**] をクリックします。
3. このシナリオでは、IP CIDR ブロックを 10.0.0.0/16 に設定します。
4. VPC の名前を指定します。
5. パブリックサブネットを 10.0.0.0/24 に設定します。（これは管理ネットワークです）。
6. アベイラビリティゾーンを選択してください。
7. サブネットの名前を付けます。
8. [**VPC** の作成] をクリックします。

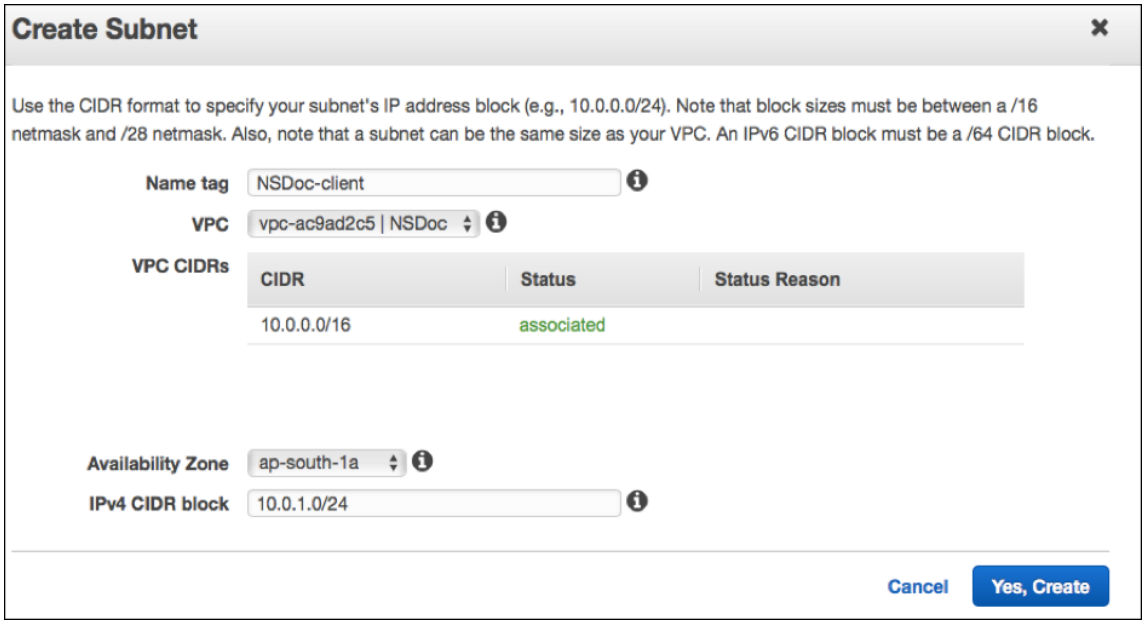


ステップ 2: 追加のサブネットを作成します。

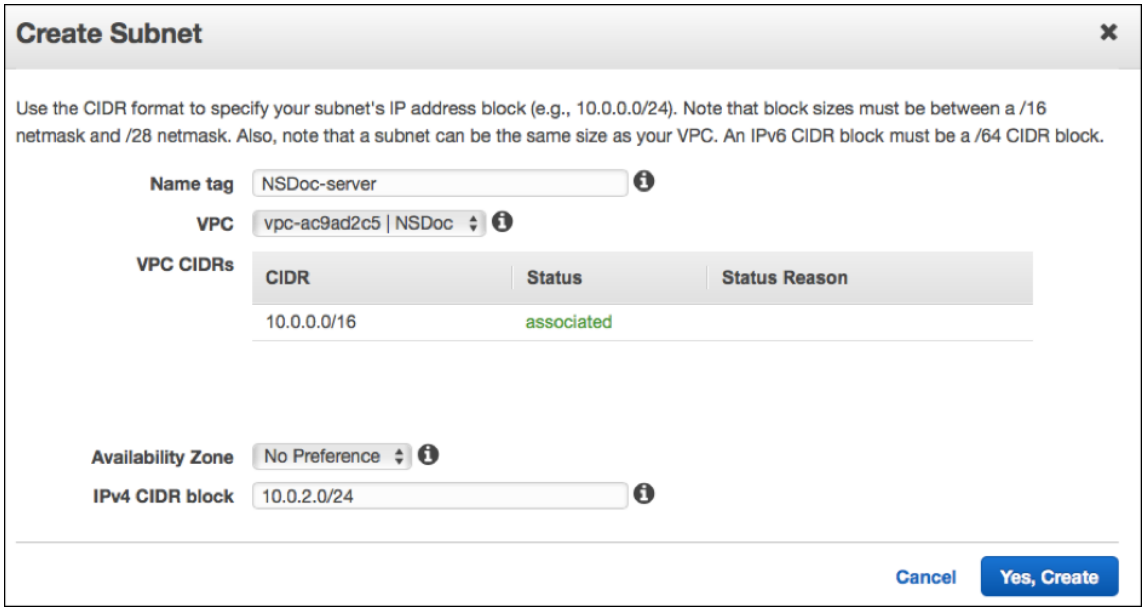
1. https://console.aws.amazon.com/vpc/で Amazon VPC コンソールを開きます。

2. ナビゲーションペインで、次の詳細を入力した後、[Subnets]、[Create Subnet] の順に選択します。

   - 名前タグ: サブネットの名前を指定します。
   - VPC: サブネットを作成する VPC を選択します。
   - アベイラビリティーゾーン: ステップ 1 で VPC を作成したアベイラビリティーゾーンを選択します。
   - IPv4 CIDR ブロック: サブネットの IPv4 CIDR ブロックを指定します。このシナリオでは、10.0.1.0/24 を選択します。



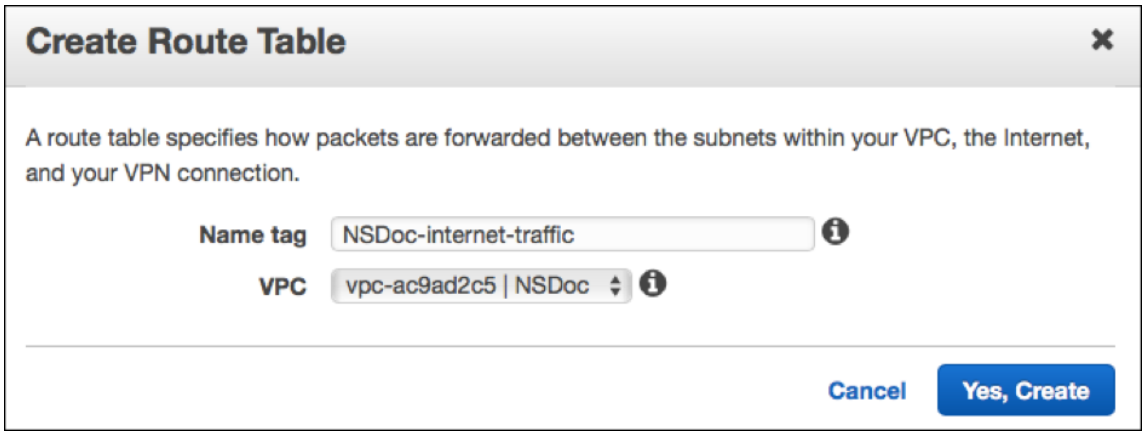
3. この手順を繰り返して、バックエンドサーバー用のサブネットをもう 1 つ作成します。



ステップ 3: ルートテーブルを作成します。

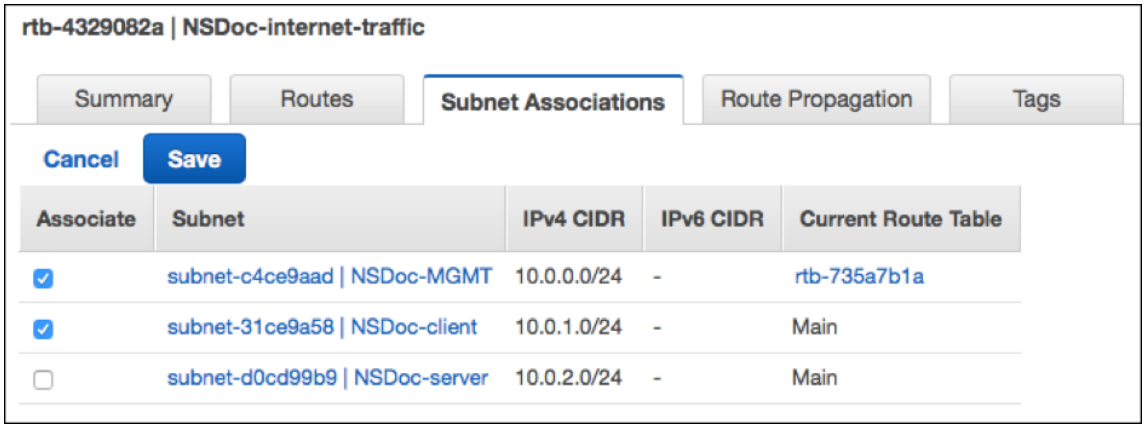
1. https://console.aws.amazon.com/vpc/で Amazon VPC コンソールを開きます。

2. ナビゲーションペインで、「ルートテーブル」**>**「ルートテーブルを作成」を選択します。
3. [Create Route Table] ウィンドウで、名前を追加し、ステップ 1 で作成した VPC を選択します。
4. [**Yes, Create**] をクリックします。

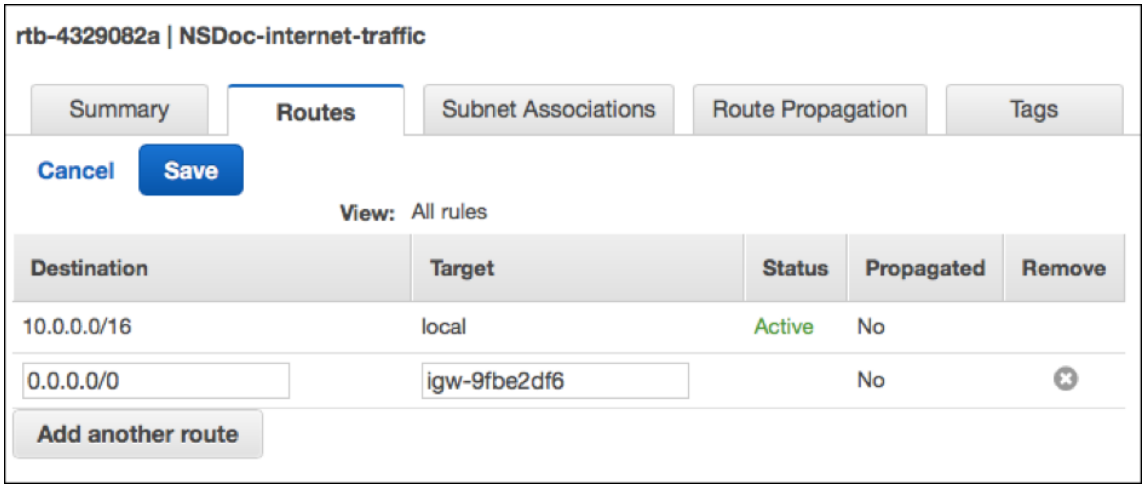


ルートテーブルは、この VPC 用に作成したすべてのサブネットに割り当てられます。これにより、あるサブネット内のインスタンスからのトラフィックのルーティングが別のサブネットのインスタンスに到達できるようになります。

5. [サブネットの関連付け] をクリックし、[編集] をクリックします。
6. 管理サブネットとクライアントサブネットをクリックし、[Save] をクリックします。これにより、インターネットトラフィック専用のルートテーブルが作成されます。



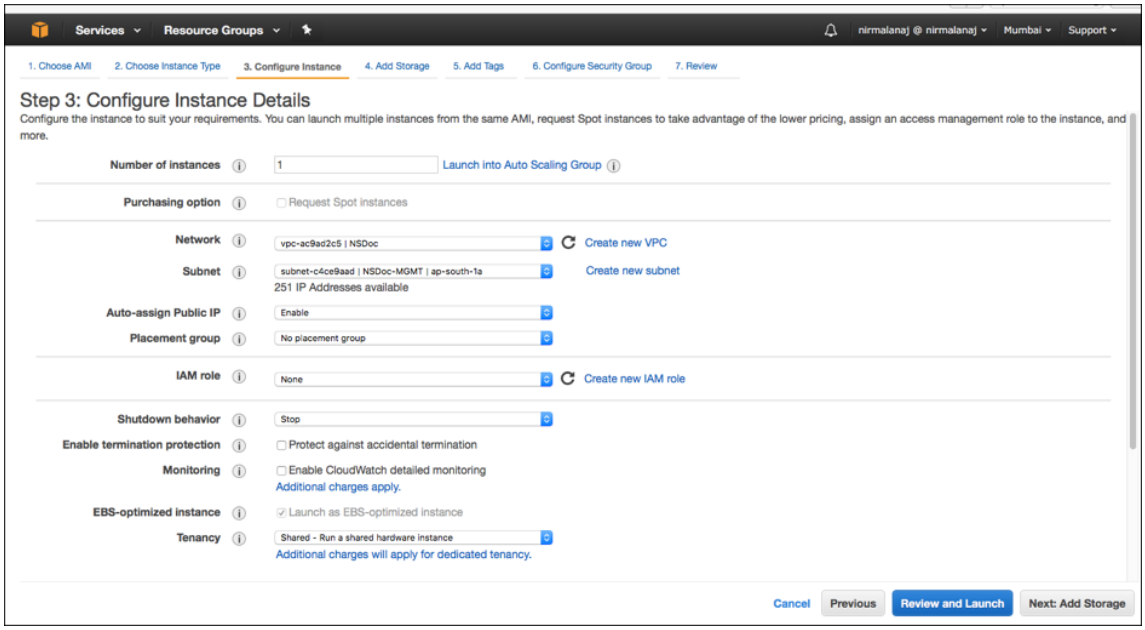
7. **[**ルート**] > [**編集**] > [**別のルートを追加**]** をクリックします。
8. [Destination] フィールドに 0.0.0.0/0 を追加し、[Target] フィールドをクリックして [igw] (<xxxx> VPC ウィザードによって自動的に作成されたインターネットゲートウェイ) を選択します。
9. [保存] をクリックします。



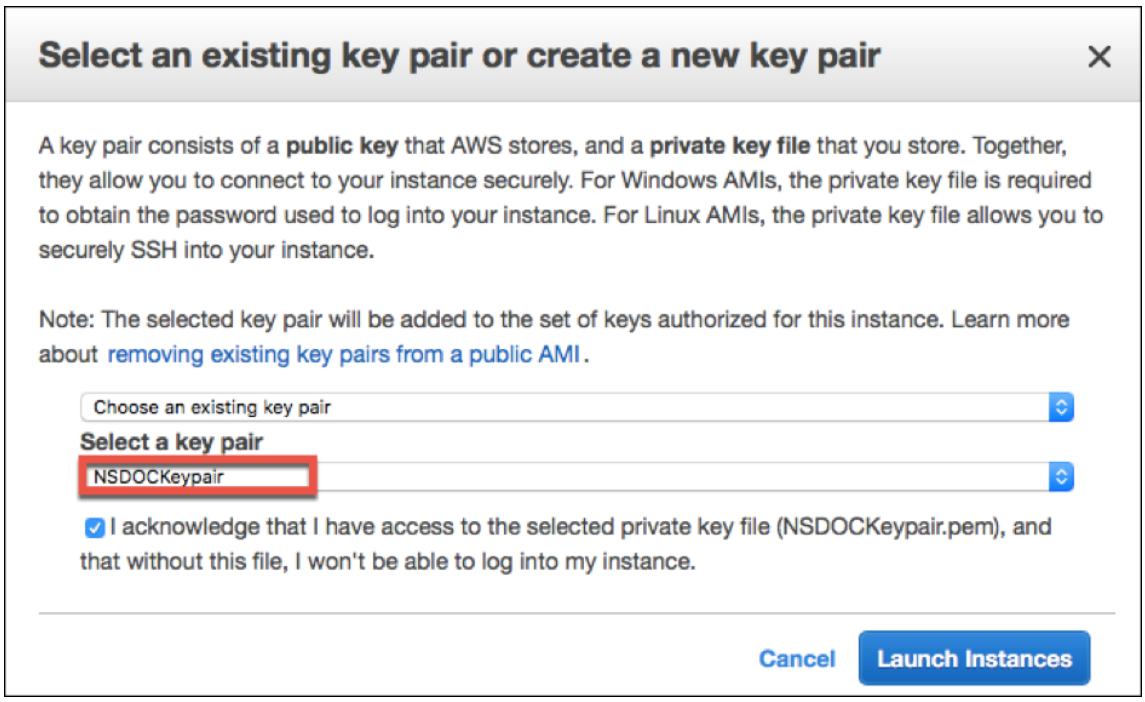
10. サーバー側のトラフィックのルートテーブルを作成する手順に従います。

ステップ 4: Citrix ADC VPX インスタンスを作成します。

1. AWS マネジメントコンソールにログオンし、[**Compute**] の下の [**EC2**] をクリックします。
2. [AWS マーケットプレイス] をクリックします。「AWS マーケットプレイスの検索」バーに「Citrix ADC VPX」と入力し、Enter キーを押します。使用可能な Citrix ADC VPX エディションが表示されます。
3. [選択] をクリックして、必要な Citrix ADC VPX エディションを選択します。EC2 インスタンスウィザードが起動します。
4. [インスタンスタイプの選択] ページで、[**m4**] を選択します。**Xlarge** (推奨) をクリックし、[次へ**:** インスタンスの詳細を設定] をクリックします。
5. [インスタンスの詳細の設定] ページで、次の項目を選択し、[次へ: ストレージの追加] をクリックします。
   - インスタンス数:1
   - ネットワーク: ステップ 1 で作成した VPC
   - サブネット: 管理サブネット
   - パブリック IP の自動割り当て: 有効



6. 「ストレージの追加」ページで、デフォルトのオプションを選択し、「次へ: タグの追加」をクリックします。
7. [Add Tags] ページで、インスタンスの名前を追加し、[Next: Configure Security Group] をクリックします。
8. [セキュリティグループの設定] ページで、デフォルトのオプション（AWS Marketplace によって生成され、Citrix Systems の推奨設定に基づいています）を選択し、[レビューして起動] をクリックします > 起動します。
9. 既存のキーペアを選択するか、新しいキーペアを作成して新しいキーペアを選択するように求められます。[Select a key pair] ドロップダウンリストから、前提条件として作成したキーペアを選択します（「前提条件」セクションを参照）。
10. チェックボックスをオンにしてキーペアを確認し、[Launch Instances] をクリックします。



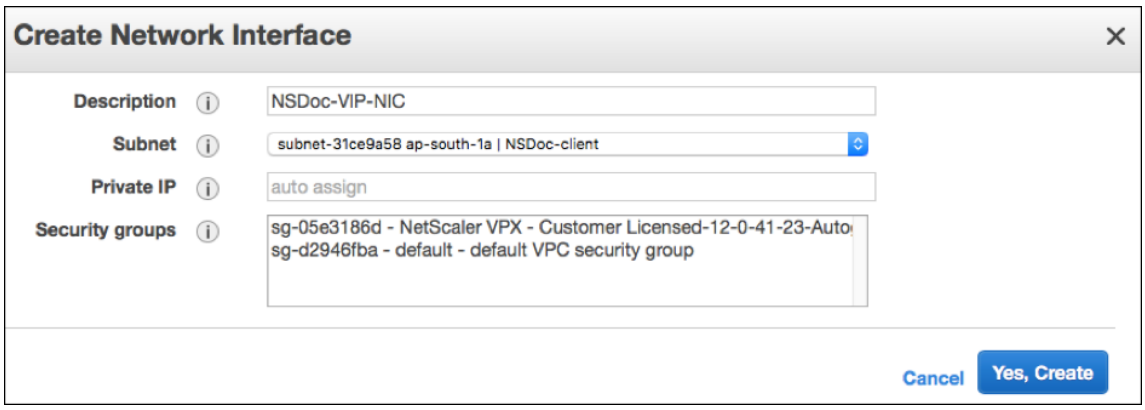
[Launch Instance Wizard] に [Launch Status] が表示され、インスタンスが完全に起動されるとインスタンスのリストに表示されます。

インスタンスをチェックし、AWS コンソールで [EC2] > [実行中のインスタンス] をクリックします。インスタンスを選択し、名前を追加します。インスタンスの状態が実行中で、ステータスチェックが完了していることを確認します。

ステップ 5：より多くのネットワークインターフェイスを作成し、アタッチします。

VPC を作成したとき、それに関連付けられたネットワークインターフェイスは 1 つだけです。次に、VIP と SNIP の 2 つのネットワークインターフェイスを VPC に追加します。

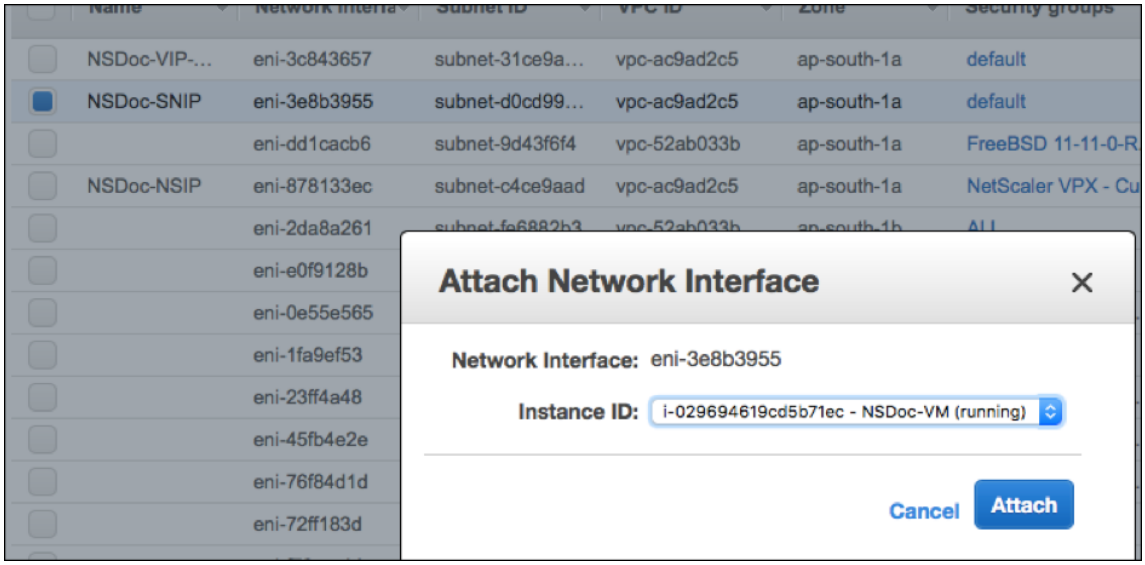
1. https://console.aws.amazon.com/ec2/で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[ネットワークインターフェイス] を選択します。
3.「ネットワークインタフェースの作成」を選択します。
4. [説明] に、わかりやすい名前を入力します。
5. [Subnet] で、VIP 用に以前に作成したサブネットを選択します。
6. プライベート IP の場合は、デフォルトのオプションのままにします。
7. [セキュリティグループ] で、グループを選択します。
8. [**Yes, Create**] をクリックします。



9. ネットワークインターフェースが作成されたら、インターフェースに名前を追加します。
10. この手順を繰り返して、サーバー側のトラフィック用のネットワークインターフェイスを作成します。

ネットワークインターフェイスをアタッチします。

1. ナビゲーションペインで、[ネットワークインターフェイス] を選択します。
2. ネットワークインターフェイスを選択し、[Attach] を選択します。
3. [Attach Network Interface] ダイアログボックスで、インスタンスを選択し、[Attach] を選択します。



ステップ 6: エラスティック IP を NSIP に接続します。

1. AWS マネジメントコンソールから、[ネットワークとセキュリティ] **> [Elastic IP**] に移動します。
2. アタッチする無料の EIP がないか確認してください。存在しない場合は、[新しいアドレスの割り当て] をクリックします。
3. 新しく割り当てられた IP アドレスを選択し、[アクション] **> [**アドレスの関連付け**]** を選択します。
4. [ネットワークインターフェイス] オプションボタンをクリックします。
5. [Network Interface] ドロップダウンリストから、管理 NIC を選択します。

6. [プライベート **IP**] ドロップダウンメニューから、AWS によって生成された IP アドレスを選択します。
7. [再関連付け] チェックボックスをオンにします。
8. [関連付け] をクリックします。

VPX インスタンスにアクセスします。

3 つの NIC を使用してスタンドアロンの Citrix ADC VPX インスタンスを構成した後、VPX インスタンスにログオンして、Citrix ADC 側の構成を完了します。次のオプションを使用します。

- GUI: ブラウザに管理 NIC のパブリック IP を入力します。ユーザー名として `nsroot` を使用し、パスワードとしてインスタンス ID（i-0c1ffe1d987817522）を使用してログオンします。

> 注
>
> 最初のログオン時に、セキュリティ上の理由からパスワードを変更するように求められます。パスワードを変更した後、構成を保存する必要があります。構成が保存されずにインスタンスが再起動する場合は、デフォルトのパスワードでログオンする必要があります。プロンプトでパスワードを再度変更し、構成を保存します。

- SSH：SSH クライアントを開き、次のように入力します。

```
ssh -i \<location of your private key\> ns root@\<public DNS of the instance
\>
```

パブリック DNS を見つけるには、インスタンスをクリックし、[接続] をクリックします。

関連情報:

- Citrix ADC が所有する IP アドレス（NSIP、VIP、および SNIP）を構成するには、Citrix ADC 所有の IP アドレスの構成を参照してください。
- Citrix ADC VPX アプライアンスの BYOL バージョンを構成しました。詳細については、VPX ライセンスガイド（http://support.citrix.com/article/CTX122426）

Citrix ADC VPX ライセンスのダウンロード

October 7, 2021

AWS マーケットプレイスから Citrix ADC VPX-カスタマーライセンスインスタンスを起動した後、ライセンスが必要です。VPX ライセンスの詳細については、ライセンスの概要を参照してください。

次の操作を実行する必要があります。

1. Citrix Web サイト内のライセンスポータルを使用して、有効なライセンスを生成します。
2. ライセンスをインスタンスにアップロードします。

有料マーケットプレイスインスタンスの場合は、ライセンスをインストールする必要はありません。正しい機能セットとパフォーマンスが自動的にアクティブ化されます。

モデル番号が VPX 5000 より大きい Citrix ADC VPX インスタンスを使用する場合、ネットワークスループットはインスタンスのライセンスで指定されたものとは異なることがあります。ただし、SSL スループットや 1 秒あたりの SSL トランザクションといった他の機能は改善されている場合があります。

`c4.8xlarge`インスタンスタイプでは 5 Gbps のネットワーク帯域幅が観測されます。

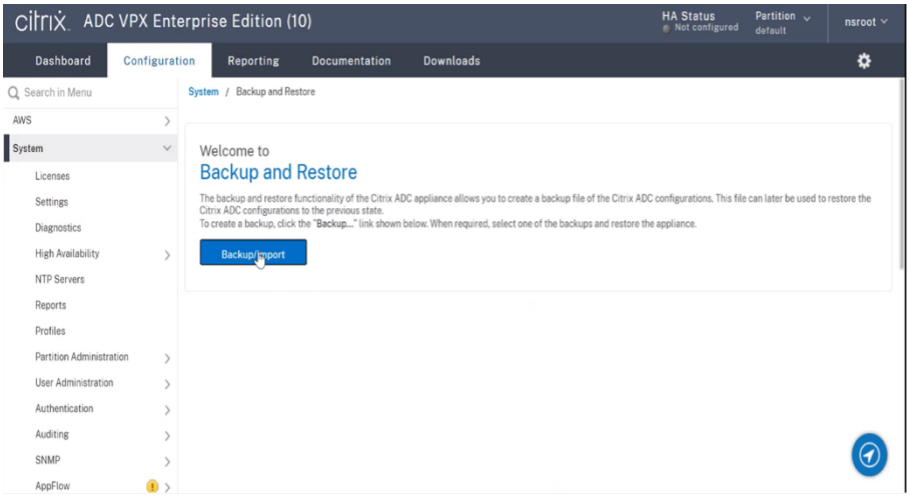
AWS サブスクリプションを BYOL に移行する方法

このセクションでは、AWS サブスクリプションから独自のライセンス (BYOL) に移行する手順、およびその逆について説明します。

AWS サブスクリプションを BYOL に移行するには、次の手順を実行します。

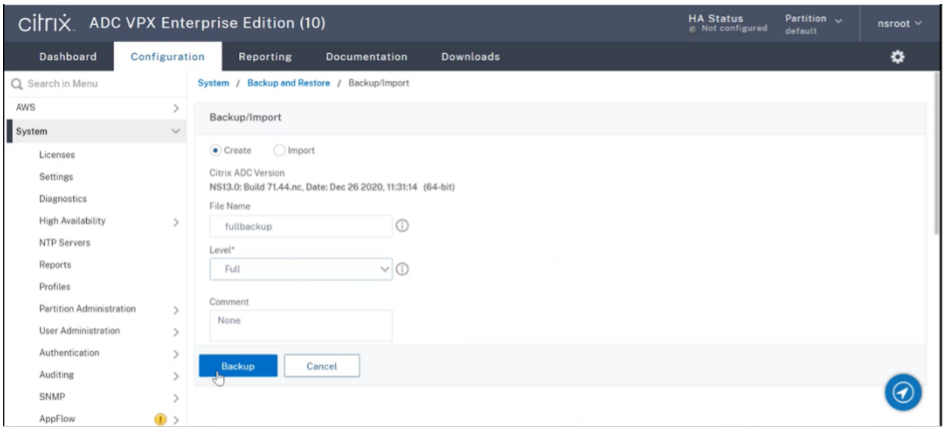
> 注
>
> ** ステップ **2** とステップ **3** は **Citrix** ADC VPX インスタンスで実行され、その他の手順はすべて AWS ポータルで実行されます。

1. Citrix ADC VPX-同じセキュリティグループ、IAM ロール、サブネットを持つ古い EC2 インスタンスと同じアベイラビリティーゾーンで、カスタマーライセンスを使用して BYOL EC2 インスタンスを作成します。新しい EC2 インスタンスには ENI インターフェイスが 1 つだけ必要です。
2. Citrix ADC GUI を使用して古い EC2 インスタンスのデータをバックアップするには、次の手順に従います。
   a) [システム] **>** [バックアップと復元] に移動します。
   b) [ようこそ] ページで、[バックアップ/インポート] をクリックしてプロセスを開始します。

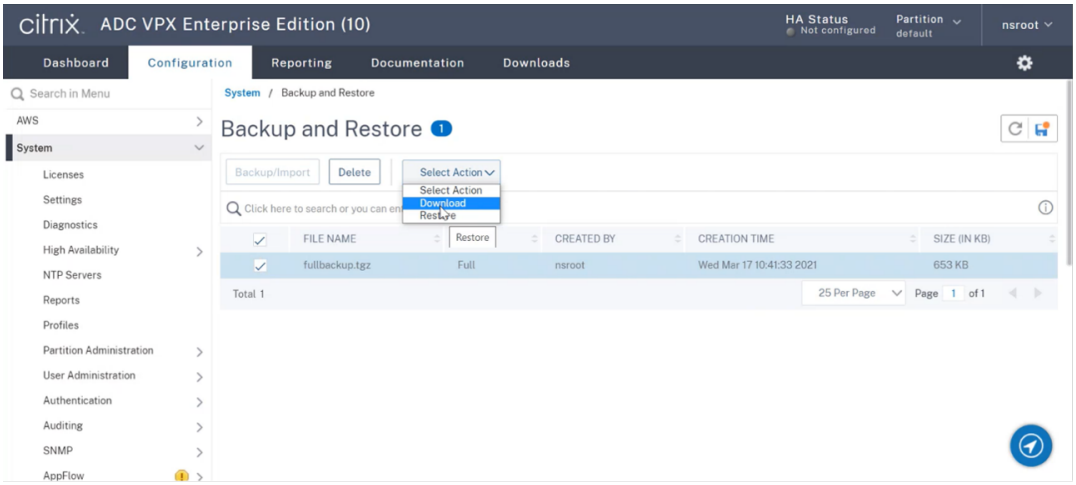


c) [バックアップ/インポート] ページで、次の詳細を入力します。

- **Name**： バックアップファイルの名前。
- **Level**： バックアップレベルを「フル」として選択します。
- [コメント]: バックアップの簡単な説明を入力します。



d) [バックアップ] をクリックします。バックアップが完了したら、ファイルを選択してローカルマシンにダウンロードできます。

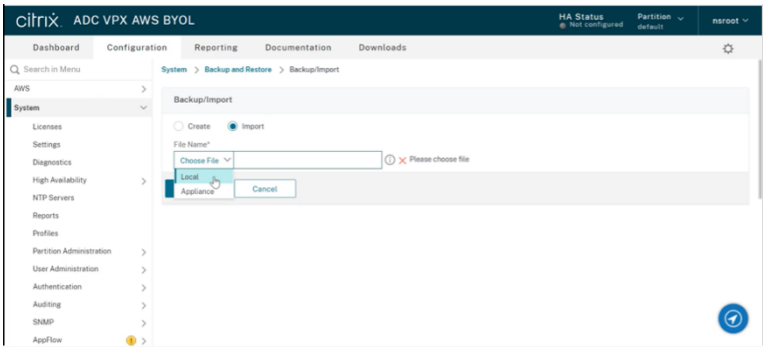


3. Citrix ADC GUI を使用して新しい EC2 インスタンスにデータを復元するには、次の手順に従います。

   a) [システム] **>** [バックアップと復元] に移動します。

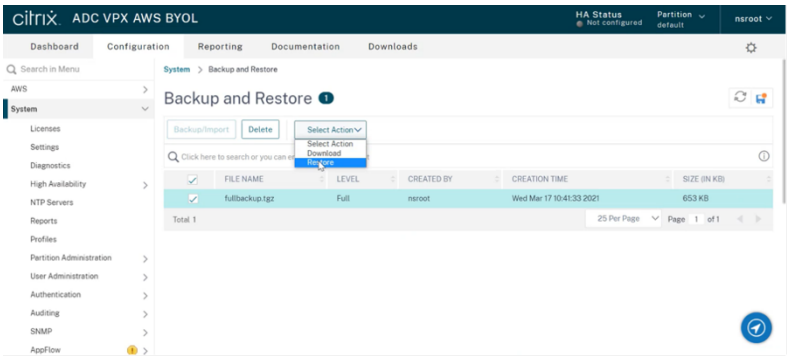
   b) [バックアップ/インポート] をクリックして、プロセスを開始します。

   c) [インポート] オプションを選択し、バックアップファイルをアップロードします。

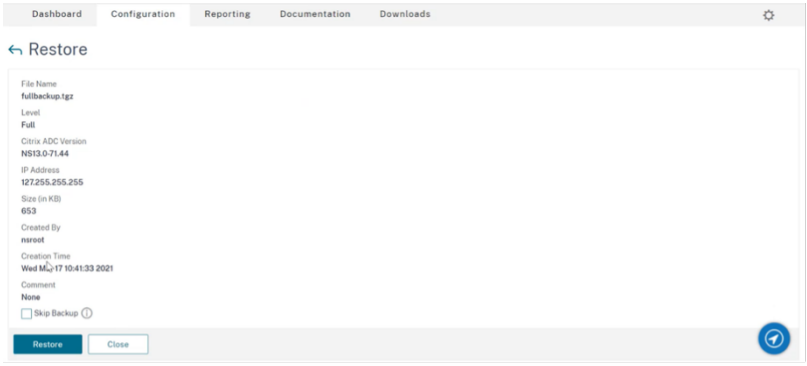


   d) ファイルを選択します。

   e) [アクションの選択] ドロップダウンメニューから、[復元] を選択します。



   f) [復元] ページで、ファイルの詳細を確認し、[復元] をクリックします。



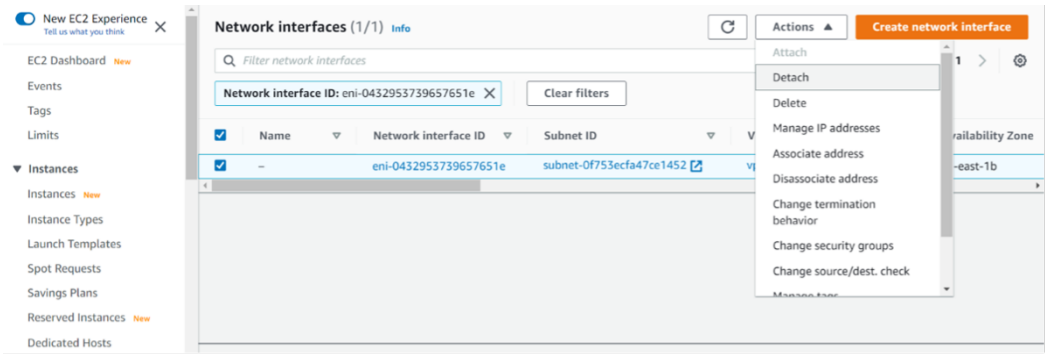
g) 復元後、EC2 インスタンスを再起動します。

4. 古い EC2 インスタンスから新しい EC2 インスタンスに、すべてのインターフェイス（NSIP アドレスがバインドされている管理インターフェイスを除く）を移動します。ネットワークインターフェイスを別の EC2 インスタンスに移動するには、次の手順を実行します。

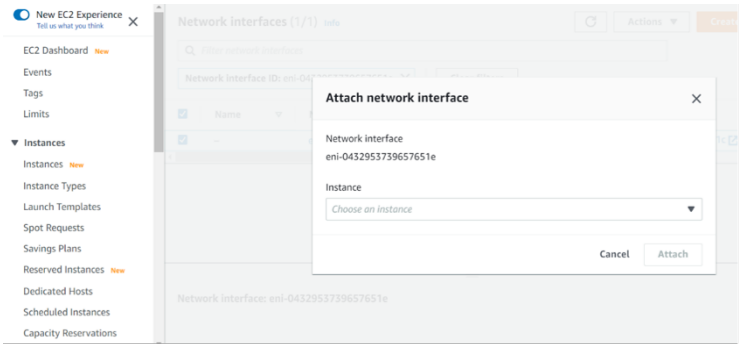
   a) **AWS** ポータルで、古い EC2 インスタンスと新しい EC2 インスタンスの両方を停止します。

   b) [ネットワークインターフェイス] に移動し、古い EC2 インスタンスにアタッチされたネットワークインターフェイスを選択します。

   c) [アクション] **> [**デタッチ**]** をクリックして **EC2** インスタンスをデタッチします。



   d) [アクション] **> [Attach]** の順にクリックして、ネットワークインターフェイスを新しい **EC2** インスタンスにアタッチします。ネットワークインターフェイスをアタッチする必要がある EC2 インスタンス名を入力します。



e) 接続されている他のすべてのインターフェイスについて、ステップ **1** からステップ **4** を実行します。シーケンスに従い、インターフェイスの順序を維持するようにしてください。つまり、まずインターフェイス 2 をデタッチして接続し、次にインターフェイス 3 をデタッチして接続します。

5. 古い EC2 インスタンスから管理インターフェイスをデタッチすることはできません。したがって、古い EC2 インスタンスの管理インターフェイス（プライマリネットワークインターフェイス）上のすべてのセカンダリ IP アドレス（存在する場合）を新しい EC2 インスタンスに移動します。IP アドレスをあるインターフェイスから別のインターフェイスに移動するには、次の手順を実行します。

   a) **AWS** ポータルで、古い EC2 インスタンスと新しい EC2 インスタンスの両方が **Stop** 状態であることを確認します。

   b) [ネットワークインターフェイス] に移動し、古い EC2 インスタンスにアタッチされた管理ネットワークインターフェイスを選択します。

   c) [アクション] **> [IP** アドレスの管理] の順にクリックし、割り当てられているすべてのセカンダリ IP アドレス（存在する場合）を書き留めます。

   d) 新しい EC2 インスタンスの管理ネットワークインターフェイスまたはプライマリインターフェイスに移動します。

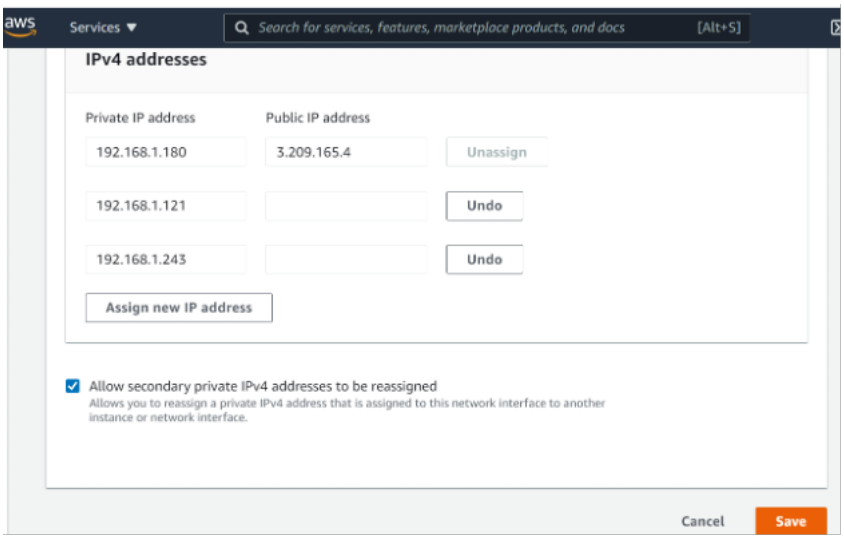
   e) [アクション] **> [IP** アドレスの管理] の順にクリックします。

   f) [**IPv4** アドレス] で、[新しい **IP** アドレスを割り当て] をクリックします。

   g) ステップ **3** で説明する IP アドレスを入力します。

   h) [セカンダリプライベート **IP** アドレスの再割り当てを許可する] チェックボックスをオンにします。

   i) [保存] をクリックします。



6. 新しい EC2 インスタンスを起動し、設定を確認します。すべての設定が移動されたら、要件に従って古い EC2 インスタンスを削除または保持できます。

7. 古い EC2 インスタンスの NSIP アドレスに EIP アドレスがアタッチされている場合は、古いインスタンスの NSIP アドレスを新しいインスタンスの NSIP アドレスに移動します。
8. 古いインスタンスに戻す場合は、古いインスタンスと新しいインスタンスの逆の方法で同じ手順を実行します。
9. サブスクリプションインスタンスから BYOL インスタンスに移行した後、ライセンスが必要です。ライセンスをインストールするには、次の手順に従います。
    - Citrix Web サイトのライセンスポータルを使用して、有効なライセンスを生成します。
    - ライセンスをインスタンスにアップロードします。詳細については、VPX ADC-新しいライセンスをインストールするを参照してください。

> 注
>
> BYOL インスタンスをサブスクリプションインスタンス (有料マーケットプレイスインスタンス) に移動する場合、ライセンスをインストールする必要はありません。正しい機能セットとパフォーマンスが自動的にアクティブ化されます。

制限事項

管理インターフェイスを新しい EC2 インスタンスに移動することはできません。したがって、管理インターフェイスを手動で構成することをお勧めします。詳細については、前の手順の手順 **5** を参照してください。新しい EC2 インスタンスは、古い EC2 インスタンスの正確なレプリカで作成されますが、新しい IP アドレスは NSIP アドレスだけです。

異なるアベイラビリティーゾーン内のサーバーの負荷分散

October 7, 2021

VPX インスタンスは、同じアベイラビリティーゾーンまたは以下で実行されているサーバーの負荷分散に使用できます。

- 同じ AWS VPC 内の別のアベイラビリティーゾーン（AZ）
- 別の AWS リージョン
- VPC 内の AWS EC2

VPX インスタンスが、VPX インスタンスがある AWS
VPC 外で実行されているサーバーの負荷を分散できるようにするには、次のように、EIP を使用してインターネット Gateway 経由でトラフィックをルーティングするようにインスタンスを設定します。

1. Citrix ADC VPX インスタンスで SNIP を構成するには、Citrix の CLI または GUI を使用します。
2. サーバー側のトラフィック用にパブリック向きのサブネットを作成して、トラフィックを AZ からルーティングできるようにします。
3. AWS GUI コンソールを使用して、インターネット Gateway ルートをルーティングテーブルに追加します。

4. 更新したルーティングテーブルをサーバー側のサブネットに関連付けます。
5. Citrix ADC SNIP アドレスにマップされているサーバー側のプライベート IP アドレスに EIP を関連付けます。

AWS での高可用性の仕組み

October 7, 2021

AWS 上の 2 つの Citrix ADC VPX インスタンスを高可用性（HA）アクティブ/パッシブのペアとして構成できます。1 つのインスタンスをプライマリノードとして構成し、もう 1 つをセカンダリノードとして設定すると、プライマリノードは接続を受け入れ、サーバーを管理します。セカンダリノードはプライマリを監視します。何らかの理由で 1 次ノードが接続を受け入れることができない場合は、2 次ノードが引き継ぎます。

AWS では、VPX インスタンスでは以下のデプロイタイプがサポートされています。

- 同一ゾーン内の高可用性
- 異なるゾーン間の高可用性

> 注
>
> 高可用性を機能させるには、両方の Citrix ADC VPX インスタンスに IAM ロールがアタッチされ、Elastic IP（EIP）アドレスが NSIP に割り当てられていることを確認します。NSIP が NAT インスタンスを介してインターネットに到達できる場合は、NSIP に EIP を割り当てる必要はありません。

同じゾーン内の高可用性

同じゾーン内の高可用性展開では、両方の VPX インスタンスのネットワーク構成が類似している必要があります。

次の 2 つのルールに従います。

ルール 1。1 つの VPX インスタンス上の NIC は、もう一方の VPX 内の対応する NIC と同じサブネットに存在する必要があります。両方のインスタンスには、次のものが必要です。

- 同じサブネット上の管理インターフェイス（管理サブネットと呼ばれます）
- 同じサブネット上のクライアントインターフェイス（クライアントサブネットと呼ばれる）
- 同じサブネット上のサーバーインターフェイス（サーバーサブネットと呼ばれます）

ルール 2。両方のインスタンスの管理 NIC、クライアント NIC、およびサーバ NIC のシーケンスが同じである必要があります。
たとえば、次のシナリオはサポートされていません。

VPX インスタンス 1

NIC 0: 管理
NIC 1: クライアント
NIC 2: サーバー

VPX インスタンス 2

NIC 0: 管理

NIC 1: サーバ

NIC 2: クライアント

このシナリオでは、インスタンス 1 の NIC 1 はクライアントサブネットにあり、インスタンス 2 の NIC 1 はサーバーサブネットにあります。HA が機能するには、両方のインスタンスの NIC 1 がクライアントサブネットまたはサーバーサブネット内にある必要があります。

13.0 41.xx から、フェールオーバー後にプライマリ HA ノードの NIC（クライアント側およびサーバ側の NIC）に接続されたセカンダリプライベート IP アドレスをセカンダリの HA ノードに移行することで、高可用性を実現できます。この展開は、以下のように管理されます。

- 両方の VPX インスタンスは、NIC 列挙に従って NIC の数とサブネットマッピングが同じです。
- 各 VPX NIC には、管理 IP アドレスに対応する最初の NIC を除き、追加のプライベート IP アドレスが 1 つあります。追加のプライベート IP アドレスは、AWS ウェブコンソールでプライマリプライベート IP アドレスとして表示されます。この文書では、この追加の IP アドレスをダミーの IP アドレスと呼びます)。
- ダミー IP アドレスは、Citrix ADC インスタンス上で VIP および SNIP として構成しないでください。
- 必要に応じて、その他のセカンダリプライベート IP アドレスを作成し、VIP および SNIP として設定する必要があります。
- フェールオーバー時に、新しいプライマリノードは設定された SNIP および VIP を検索し、前のプライマリに接続されている NIC から新しいプライマリ上の対応する NIC に移動します。
- Citrix ADC インスタンスでは、HA が機能するためには IAM アクセス許可が必要です。各インスタンスに追加された IAM ポリシーに次の IAM 権限を追加します。

```
”iam:GetRole”
”ec2:DescribeInstances”
”ec2:DescribeNetworkInterfaces”
”ec2:AssignPrivateIpAddresses”
```

> 注:`unassignPrivateIpAddress` は必須ではありません。

この方法は、従来の方法よりも高速です。古い方法では、HA はプライマリノードの AWS Elastic ネットワークインターフェイスからセカンダリノードへの移行に依存します。

従来の方法では、次のポリシーが必要です。

```
”iam:GetRole”
”ec2:DescribeInstances”
”ec2:DescribeAddresses”
”ec2:AssociateAddress”
”ec2:DisassociateAddress”
```


詳細については、「AWS に高可用性ペアをデプロイする」を参照してください。

異なるゾーン間の高可用性

独立したネットワーク構成（INC）モードで、高可用性のアクティブ/パッシブのペアとして、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンに 2 つの Citrix ADC VPX インスタンスを構成できます。フェイルオーバー時に、プライマリインスタンスの VIP の EIP (Elastic IP) がセカンダリに移行し、セカンダリが新しいプライマリとして引き継がれます。フェイルオーバープロセスでは、AWS API は以下を実行します。

- IPSetsが接続されている仮想サーバーをチェックします。
- 仮想サーバーがリッスンしている 2 つの IP アドレスから、パブリック IP が関連付けられている IP アドレスを検索します。1 つは仮想サーバーに直接接続され、もう 1 つは IP セットを介して接続されます。
- パブリック IP（EIP）を、新しいプライマリ VIP に属するプライベート IP に再関連付けします。

異なるゾーン間の HA には、次のポリシーが必要です。

```
”iam:GetRole”
”ec2:DescribeInstances”
”ec2:DescribeAddresses”
”ec2:AssociateAddress”
”ec2:DisassociateAddress”
```

詳細については、「AWS アベイラビリティーゾーン全体の高可用性」を参照してください。

展開を開始する前に

AWS での HA デプロイを開始する前に、次のドキュメントをお読みください。

- 前提条件
- 制限事項と使用上のガイドライン
- AWS で Citrix ADC VPX インスタンスを展開する
- 高可用性

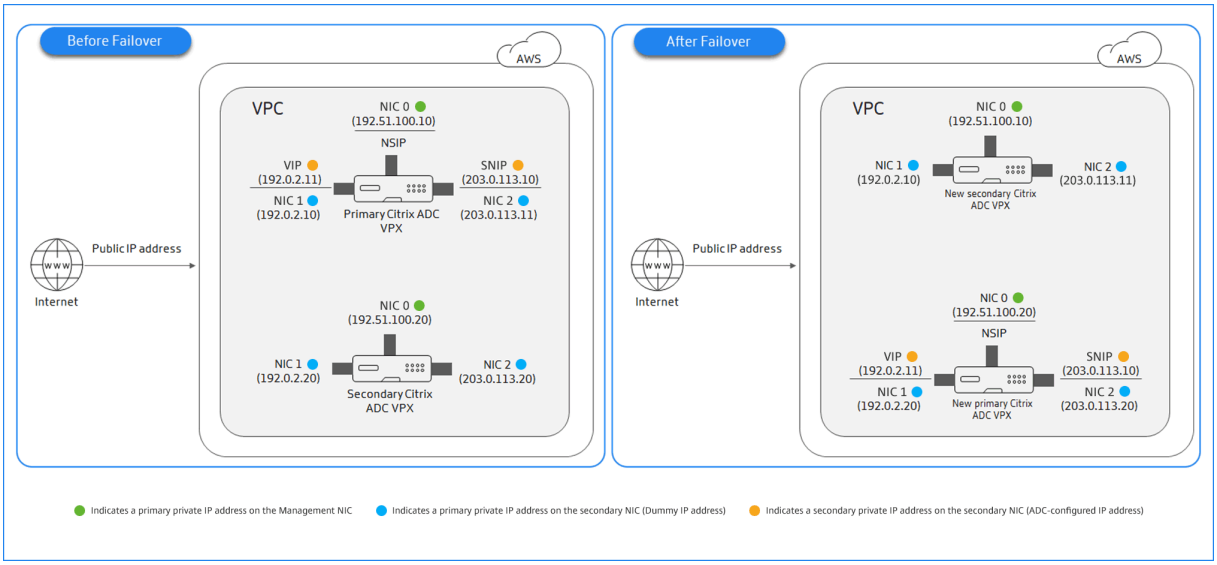
VPX HA ペアを同じ AWS アベイラビリティーゾーンにデプロイする

January 25, 2022

両方の VPX インスタンスが同じサブネット上にある同じ AWS ゾーンで、2 つの Citrix ADC VPX インスタンスを高可用性（HA）ペアとして AWS 上で構成できます。HA は、フェイルオーバー後に、プライマリ HA ノードの NIC（クライアント側およびサーバ側 NIC）に接続されているセカンダリプライベート IP アドレスをセカンダリ HA ノードに移行することで実現されます。セカンダリプライベート IP アドレスに関連付けられているすべての Elastic IP アドレスも移行されます。

次の図は、セカンダリプライベート IP アドレスを移行する HA フェールオーバーのシナリオを示しています。

図 1: プライベート IP マイグレーションを使用した、AWS 上の Citrix ADC VPX HA ペア



ドキュメントを開始する前に、次のドキュメントをお読みください。

- 前提条件
- 制限事項と使用上のガイドライン
- AWS で Citrix ADC VPX インスタンスを展開する
- 高可用性

VPX HA ペアを同じゾーンにデプロイする方法

VPX HA ペアを同じゾーンにデプロイする手順の概要を次に示します。

1. AWS で 2 つの VPX インスタンスを作成し、それぞれに NIC を 3 つ作成します。
2. AWS セカンダリプライベート IP アドレスをプライマリノードの VIP および SNIP に割り当てる
3. AWS セカンダリプライベート IP アドレスを使用してプライマリノードで VIP と SNIP を設定する
4. 両方のノードで HA を構成する

手順 **1**: 同じ **VPC** を使用して、それぞれ **3** つの **NIC**（イーサネット **0**、イーサネット **1**、イーサネット **2**）を持つ **2** つの **VPX** インスタンス（プライマリノードとセカンダリノード）を作成します

AWS ウェブコンソールを使用して、Citrix ADC VPX インスタンスを AWS にデプロイするに記載されている手順に従います。

手順 **2**：プライマリノードで、イーサネット **1**（クライアント **IP** または **VIP**）とイーサネット **2**（バックエンドサーバ **IP** または **SNIP**）にセカンダリプライベート **IP** アドレスを割り当てます

AWS コンソールは、設定された NIC にプライマリプライベート IP アドレスを自動的に割り当てます。VIP と SNIP には、セカンダリプライベート IP アドレスと呼ばれる、より多くのプライベート IP アドレスを割り当てます。

セカンダリプライベート IPv4 アドレスをネットワークインターフェイスに割り当てるには、次の手順を実行します。

1. https://console.aws.amazon.com/ec2/で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Network Interfaces] を選択し、インスタンスにアタッチされているネットワークインターフェイスを選択します。
3. [アクション]、[IP アドレスの管理] を選択します。
4. [IPv4 アドレス] で、[新しい IP の割り当て] を選択します。
5. インスタンスのサブネット範囲内にある特定の IPv4 アドレスを入力するか、フィールドを空白のままにして Amazon が IP アドレスを選択するようにします。
6. （オプション）セカンダリプライベート IP アドレスが別のネットワークインターフェイスにすでに割り当てられている場合に再割り当てを許可するには、[Allow reassignment] を選択します。
7. 「はい、更新」を選択します。

インスタンスの説明の下に、割り当てられたセカンダリプライベート IP アドレスが表示されます。

手順 **3**：セカンダリプライベート **IP** アドレスを使用して、プライマリノードで **VIP** と **SNIP** を構成します

SSH を使用してプライマリノードにアクセスします。ssh クライアントを開き、次のように入力します。

```
ssh -i <location of your private key> nsroot@<public DNS of the
    instance>
<!--NeedCopy-->
```

次に、VIP と SNIP を設定します。

VIP の場合は、次のように入力します。

```
add ns ip <IPAddress> <netmask> -type <type>
<!--NeedCopy-->
```

[SNIP] に、次のように入力します。

```
add ns ip <IPAddress> <netmask> -type SNIP
<!--NeedCopy-->
```


save configを入力して保存します。

設定された IP アドレスを表示するには、次のコマンドを入力します。

```
1 show ns ip
2 <!--NeedCopy-->
```

詳しくは、次のトピックを参照してください:

- 仮想 IP (VIP) アドレスの設定と管理
- NSIP アドレスの構成

ステップ **4:** 両方のインスタンスで **HA** を設定する

プライマリノードでシェルクライアントを開き、次のコマンドを入力します。

```
1 add ha node <id> <private IP address of the management NIC of the
     secondary node>
2 <!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
1 add ha node <id> < private IP address of the management NIC of the
     primary node >
2 <!--NeedCopy-->
```

save config と入力して、設定を保存します。

構成された HA ノードを表示するには、show ha nodeと入力します。

フェイルオーバー時に、前のプライマリノードで VIP および SNIP として構成されたセカンダリプライベート IP アドレスは、新しいプライマリノードに移行されます。

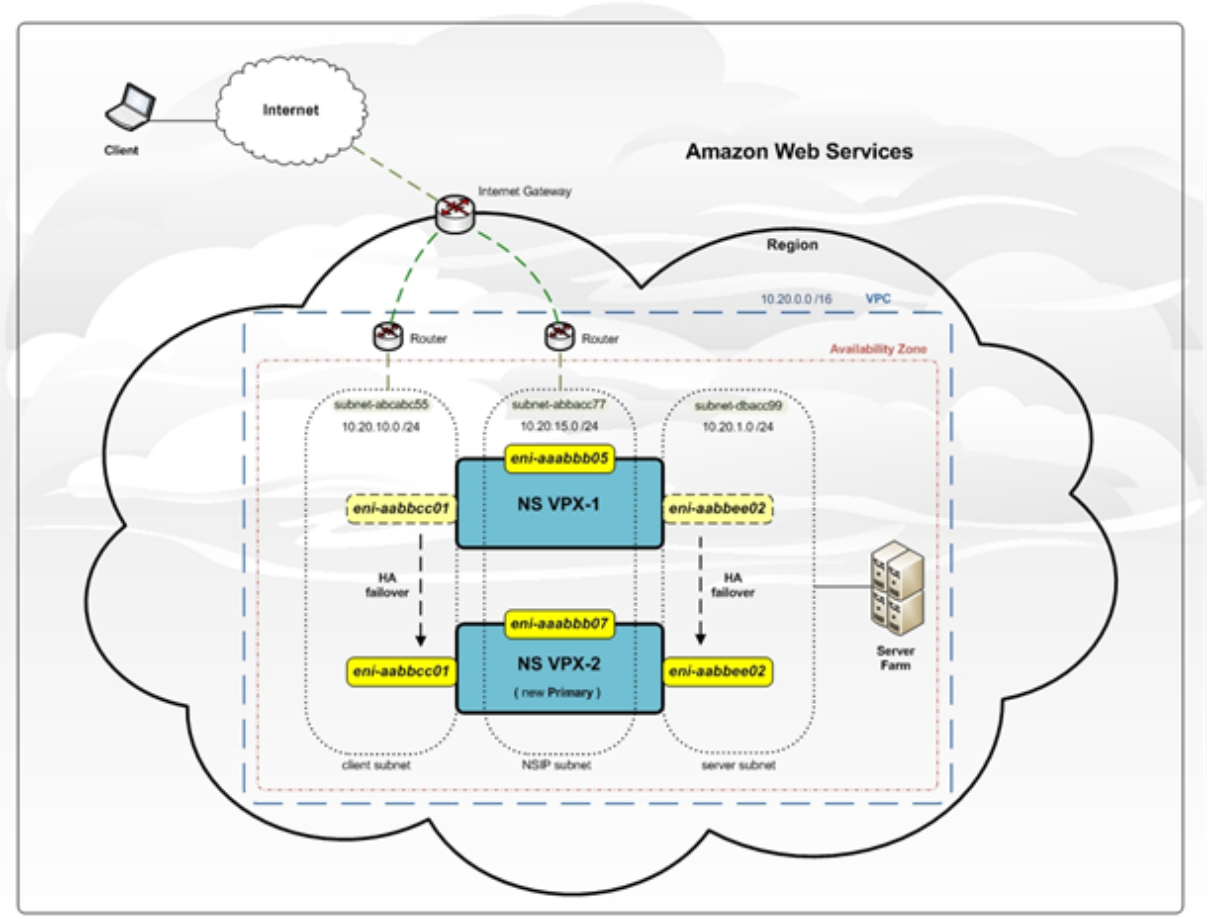
ノードでフェイルオーバーを強制するには、force HAfailover と入力します。

VPX HA ペアをデプロイするレガシーメソッド

13.0 41.x より前のリリースでは、AWS Elastic Network Interface（ENI）の移行により、同じゾーン内の HA が実現されていました。ただし、このメソッドは徐々に非推奨になっています。

次の図は、AWS 上の Citrix ADC VPX インスタンスの HA デプロイメントアーキテクチャの例を示しています。

図 1: AWS での Citrix ADC VPX HA ペア（ENI 移行を使用）



次のいずれかのオプションを使用して、2 つの VPX インスタンスを HA ペアとして AWS にデプロイできます。

- AWS マネジメントコンソールを使用して IAM ロールを持つインスタンスを手動で作成し、そのインスタンスに HA を設定します。
- または、Citrix CloudFormation テンプレートを使用して高可用性展開を自動化することもできます。

CloudFormation テンプレートは、HA ペアの作成に必要なステップ数を大幅に減らし、IAM ロールを自動的に作成します。このセクションでは、Citrix CloudFormation テンプレートを使用して Citrix ADC VPX HA（アクティブ-パッシブ）ペアをデプロイする方法を示します。

2 つの Citrix ADC VPX インスタンスを HA ペアとして展開する場合は、次の点に注意してください。

注意すべきポイント

- AWS の HA では、プライマリノードに少なくとも 2 つの ENI（1 つは管理用、もう 1 つはデータトラフィック用）があり、セカンダリノードには管理 ENI が 1 つ必要です。ただし、セキュリティ上の理由から、プライマリノードに 3 つの ENI を作成します。この設定では、プライベートネットワークとパブリックネットワークを分離できるためです（推奨）。
- セカンダリノードの ENI インターフェイスは常に 1 つで（管理用）、プライマリノードには最大 4 つの ENI を設定できます。

- 高可用性ペアの各 VPX インスタンスの NSIP アドレスは、インスタンスのデフォルト ENI で構成する必要があります。
- Amazon では、AWS ではブロードキャスト/マルチキャストパケットは許可されていません。その結果、HA セットアップでは、プライマリ VPX インスタンスに障害が発生すると、データプレーン ENI がプライマリ VPX インスタンスからセカンダリ VPX インスタンスに移行されます。
- デフォルト (管理) ENI を別の VPX インスタンスに移動することはできないため、クライアントとサーバーのトラフィック (データプレーントラフィック) にはデフォルトの ENI を使用しないでください。
- /var/log/ns.log の AWSCONFIG IOCTL NSAPI_HOTPLUG_INTF 成功出力 0 というメッセージは、2 つのデータ ENI がセカンダリインスタンス（新しいプライマリ）に正常にアタッチされたことを示しています。
- AWS の ENI デタッチ/アタッチにより、フェールオーバーには最大 20 秒かかる場合があります。
- フェールオーバー後、障害が発生したインスタンスは必ず再起動します。
- ハートビートパケットは、管理インターフェイスでのみ受信されます。
- プライマリ VPX インスタンスとセカンダリ VPX インスタンスの設定ファイルは、`nsroot` パスワードを含めて同期されます。セカンダリノードの`nsroot` パスワードは、HA 構成の同期後にプライマリノードのパスワードに設定されます。
- AWS API サーバーにアクセスするには、VPX インスタンスにパブリック IP アドレスが割り当てられているか、VPC のインターネットゲートウェイを指す VPC サブネットレベルでルーティングが正しく設定されている必要があります。
- ネームサーバー/DNS サーバーが VPC レベルの DHCP オプションにより構成されている必要があります。
- Citrix CloudFormation テンプレートでは、異なるアベイラビリティーゾーン間で HA セットアップは作成されません。
- Citrix CloudFormation テンプレートでは、INC モードは作成されません。
- AWS デバッグメッセージは、VPX インスタンスのログファイル /var/log/ns.log にあります。

Citrix CloudFormation テンプレートを使用して高可用性ペアをデプロイする

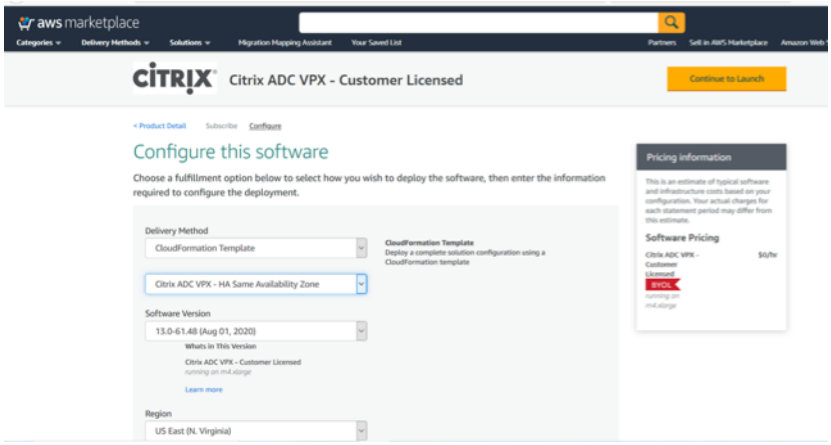
CloudFormation テンプレートを開始する前に、次の要件を満たしていることを確認してください。

- VPC
- VPC 内の 3 つのサブネット
- UDP 3003、TCP 3009—3010、HTTP、SSH ポートが開いているセキュリティグループ
- キーペア
- インターネットゲートウェイを作成する
- クライアントネットワークと管理ネットワークのルートテーブルを編集して、インターネットゲートウェイを指すようにする

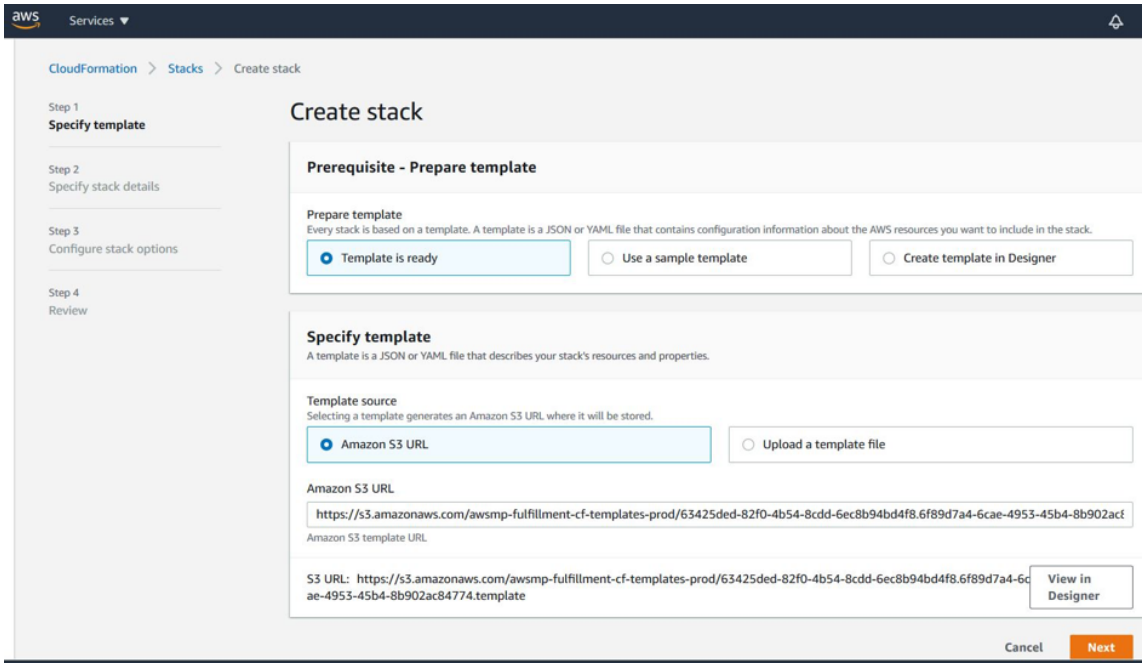
> 注
>
> Citrix CloudFormation テンプレートは、IAM ロールを自動的に作成します。既存の IAM ロールはテンプレートには表示されません。

Citrix CloudFormation テンプレートを起動するには、次の手順に従います。

1. AWS 認証情報を使用して AWS マーケットプレイスにログオンします 。
2. 検索フィールドに「**Citrix ADC VPX** 」と入力して Citrix ADC AMI を検索し、［実行］をクリックします。
3. 検索結果ページで、目的の Citrix ADC VPX 製品をクリックします。
4. 「価格設定」タブをクリックして、「価格情報」に移動します。
5. リージョンとフルフィルメントオプションを「**Citrix ADC VPX-**カスタマーライセンス」として選択します。
6. [続行] をクリックして購読します。
7. [購読] ページで詳細を確認し、[構成に進む] をクリックします。
8. **CloudFormation** テンプレートとして [配信方法] を選択します。
9. 必要な CloudFormation テンプレートを選択します。
10. [****** ソフトウェアのバージョンとリージョン] を選択し、**[**** 続行] をクリックして起動します。

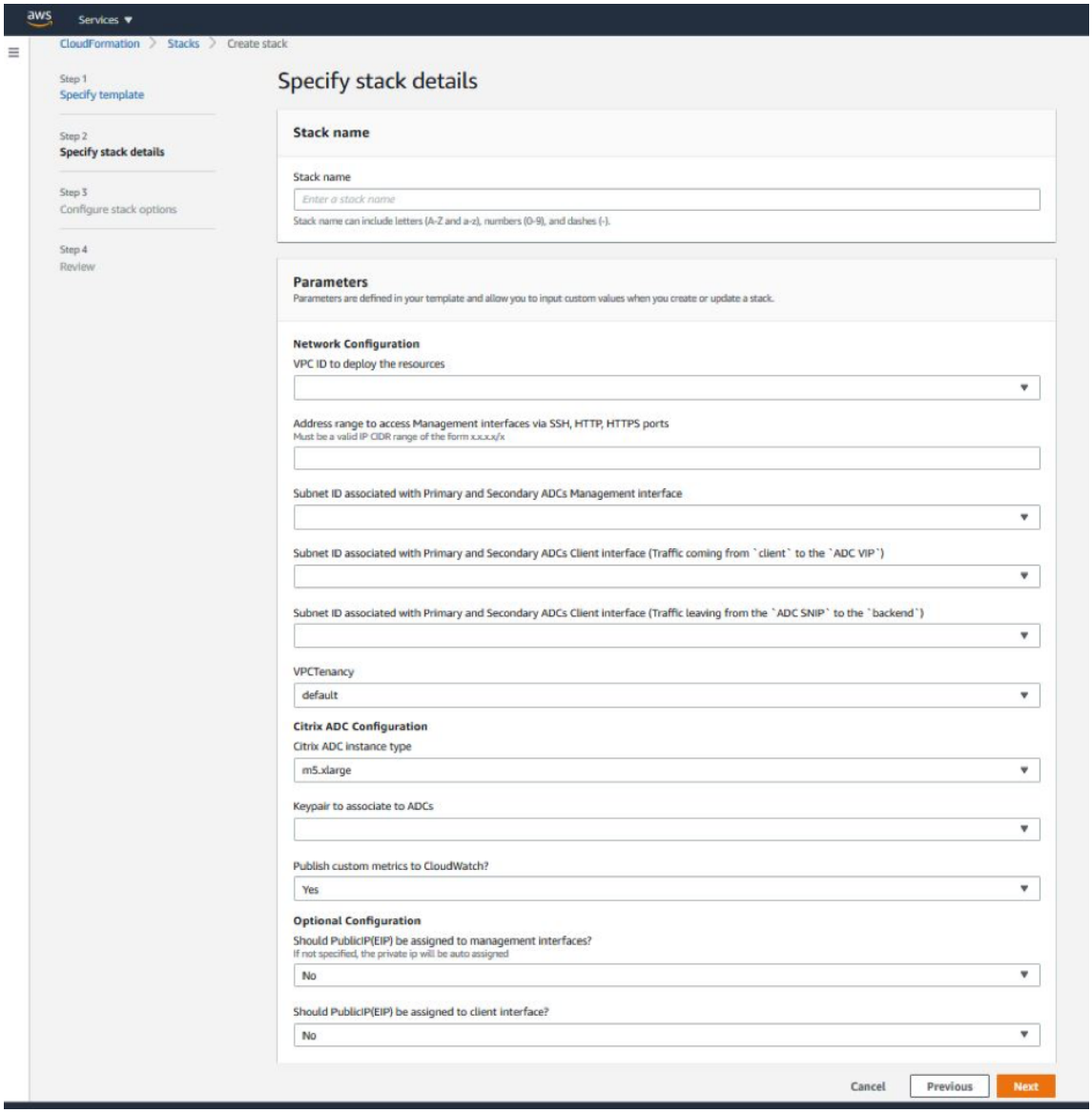


11. [アクションの選択] で、[**CloudFormation** の起動] を選択し、[起動] をクリックします。[スタックの作成] ページが表示されます。
12. ［次へ］をクリックします。



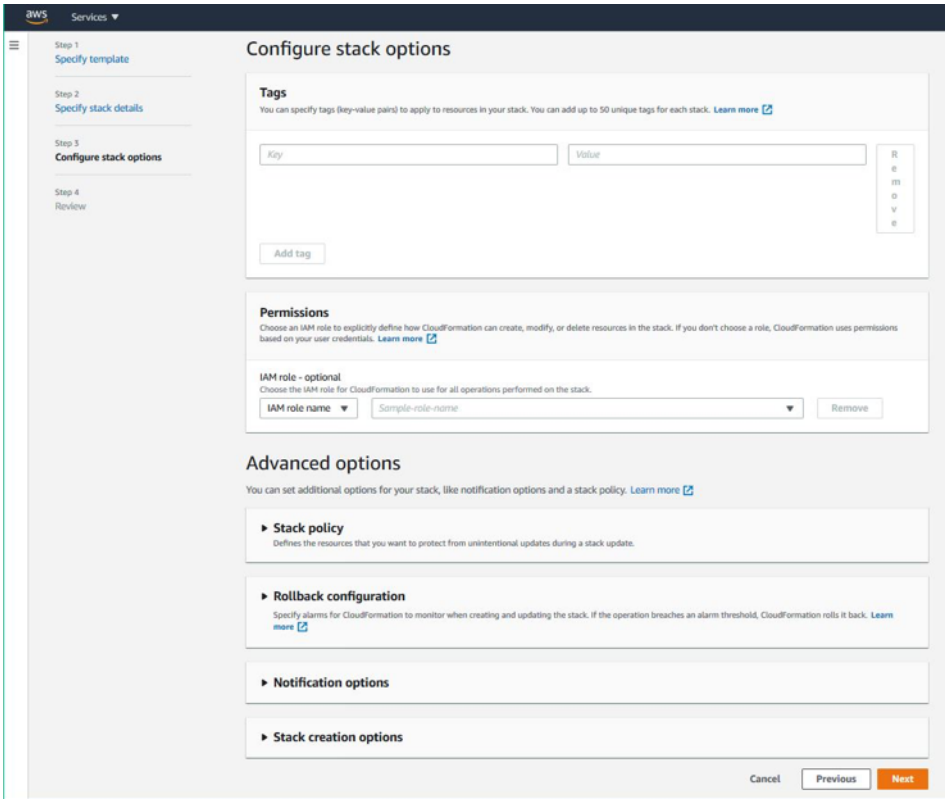
13. **[**スタックの詳細を指定] ページが表示されます。次の詳細を入力します。

    - スタック名を入力します。名前は 25 文字以内である必要があります。
    - [ネットワーク構成] で、次の操作を実行します。
        - [管理サブネットワーク]、[クライアントサブネットワーク**]**、および **[**サーバーサブネットワーク**]** を選択します。[VPC ID] で選択した VPC 内で作成した正しいサブネットワークを選択していることを確認します。
        - プライマリ管理 **IP**、セカンダリ管理 **IP**、クライアント **IP**、およびサーバ **IP** を追加します。IP アドレスは、それぞれのサブネットワークの同じサブネットに属している必要があります。または、テンプレートに IP アドレスが自動的に割り当てられるようにすることもできます。
        - **vpcTenancy** で [デフォルト] を選択します。
    - **Citrix ADC** 構成で、以下を実行します。
        - [インスタンスタイプ] で [**m5.xlarge**] を選択します。
        - [Key Pair] のメニューから、作成済みのキーペアを選択します。
        - デフォルトでは、カスタムメトリクスを **CloudWatch** にパブリッシュしますか? オプションが [はい**]** に設定されています。このオプションを無効にするには、[いいえ] を選択します。CloudWatch メトリクスの詳細については、「Amazon CloudWatch を使用してインスタンスを監視する」を参照してください。
    - [オプションの構成] で、次の操作を実行します。
        - デフォルトでは、**PublicIP**（**EIP**）を管理インターフェイスに割り当てる必要がありますかオプションが [いいえ] に設定されています。
        - デフォルトでは、**PublicIP (EIP)** をクライアントインターフェイスに割り当てる必要がありますかオプションが [いいえ] に設定されています。

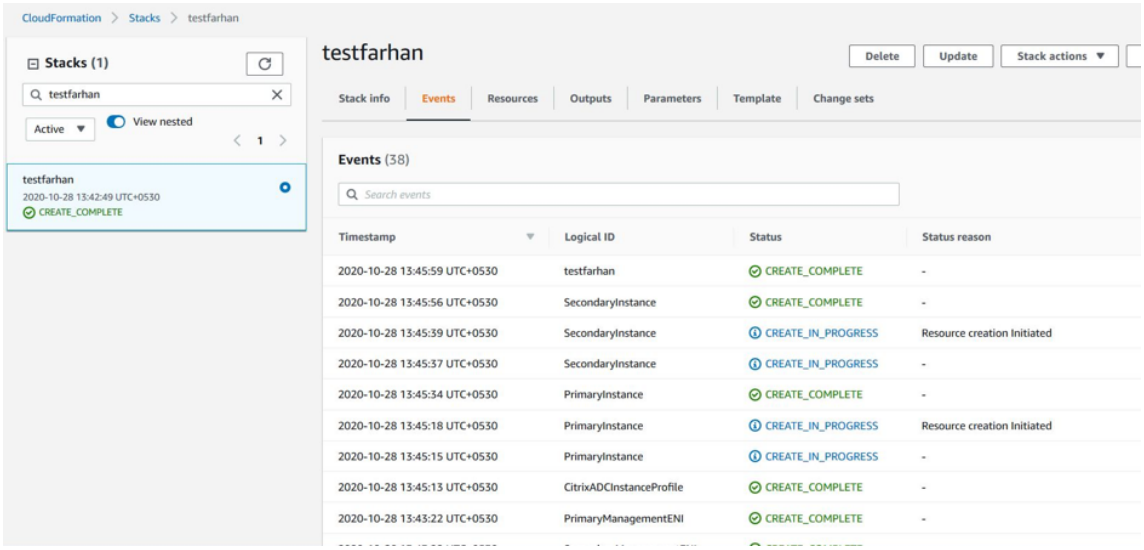


14. ［次へ］をクリックします。

15. **[**スタックオプションの設定**]** ページが表示されます。これはオプションのページです。



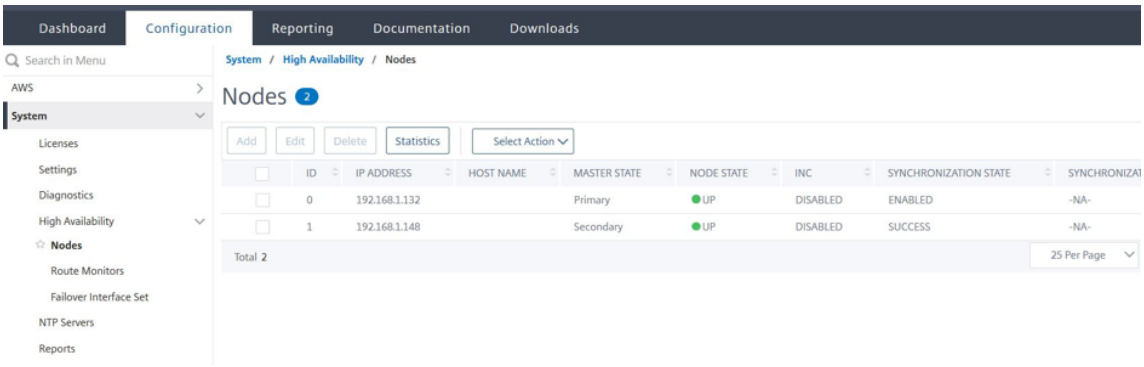
16. ［次へ］をクリックします。
17. [オプション] ページが表示されます。(これはオプションのページです)。［次へ］をクリックします。
18. [**Review**] ページが表示されます。しばらくして、設定を確認し、必要に応じて変更を加えます。
19. [**AWS CloudFormation** が **IAM** リソースを作成する可能性があることを承認します] を選択します。チェックボックスをオンにし、[スタックを作成] をクリックします。
20. **CREATE-IN-PROGRESS** が表示されます。ステータスが **CREATE-COMPLETE** になるまで待ちます。ステータスが **COMPLETE** に変更されない場合は、[**Events**] タブで失敗の原因を確認し、適切な構成でインスタンスを再作成します。



21. IAM リソースが作成されたら、[**EC2** マネジメントコンソール**] > [**インスタンス] に移動します。IAM ロールで作成された 2 つの VPX インスタンスがあります。プライマリノードとセカンダリノードは、それぞれ 3 つのプライベート IP アドレスと 3 つのネットワークインターフェイスを使用して作成されます。

22. ユーザー名`nsroot`とインスタンス ID をパスワードとしてプライマリノードにログオンします。GUI から、[システム**] > [**高可用性**] > [**ノード] に移動します。Citrix ADC VPX は、CloudFormation テンプレートによって HA ペアで既に構成されています。

23. Citrix ADC VPX HA ペアが表示されます。



Amazon CloudWatch を使用してインスタンスをモニタリングする

Amazon CloudWatch サービスを使用して、CPU とメモリの使用率、スループットなど、一連の Citrix ADC VPX メトリクスを監視できます。CloudWatch は AWS で実行されるリソースとアプリケーションをリアルタイムでモニタリングします。AWS マネジメントコンソールを使用して、Amazon CloudWatch ダッシュボードにアクセスできます。詳細については、「Amazon CloudWatch」を参照してください。

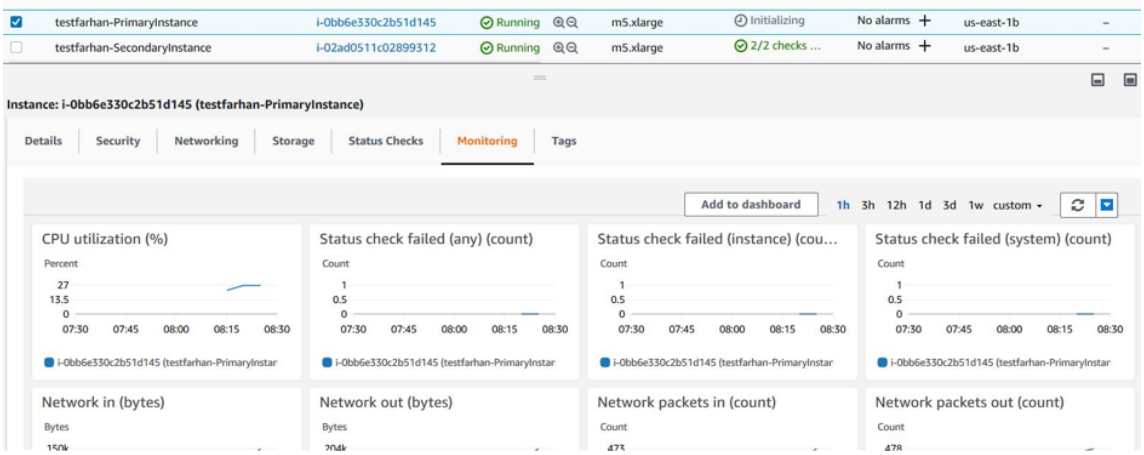
注意すべきポイント

- AWS ウェブコンソールを使用して AWS に Citrix ADC VPX インスタンスをデプロイすると、CloudWatch サービスはデフォルトで有効になります。
- Citrix CloudFormation テンプレートを使用して Citrix ADC VPX インスタンスをデプロイする場合、デフォルトのオプションは「はい」です。CloudWatch サービスを無効にする場合は、[いいえ] を選択します。
- メトリクスは、CPU (管理およびパケット CPU 使用率)、メモリ、およびスループット (インバウンドとアウトバウンド) で使用できます。

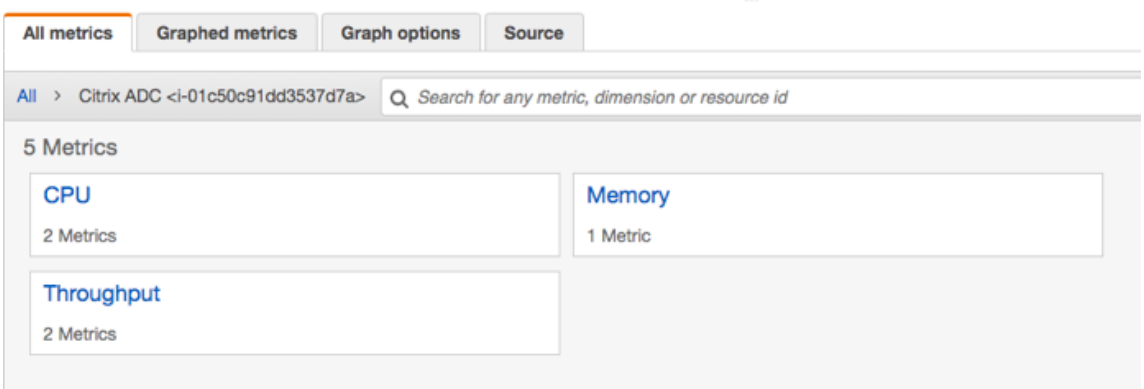
CloudWatch メトリクスの表示方法

インスタンスの CloudWatch メトリックスを表示するには、次の手順に従います。

1. **AWS** マネジメントコンソール **> EC2 >** インスタンスにログオンします。
2. インスタンスを選択します。
3. [監視] をクリックします。
4. [**CloudWatch** メトリックスをすべて表示] をクリックします。



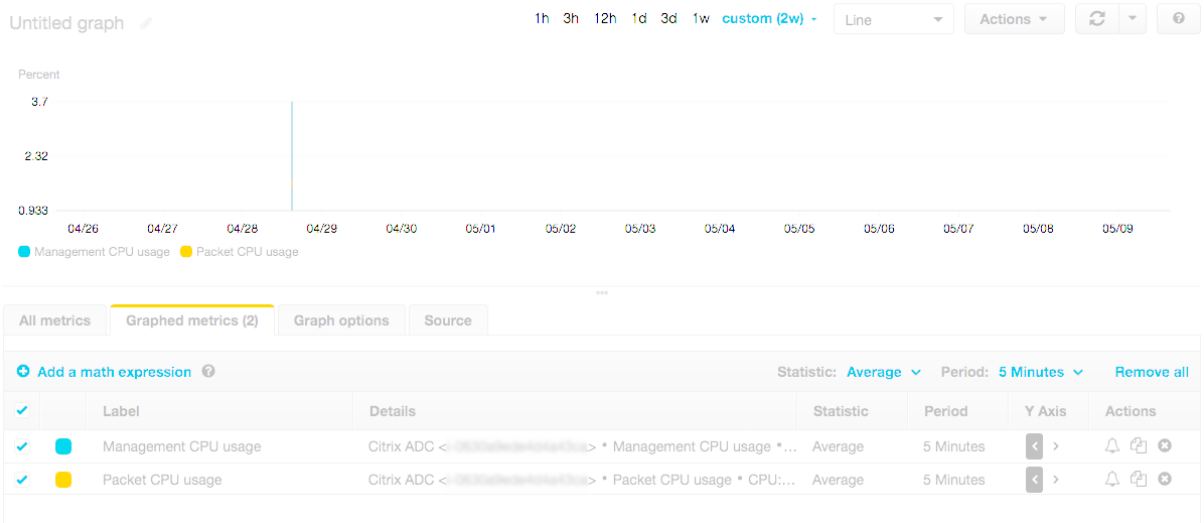
5. [すべてのメトリックス] で、インスタンス ID をクリックします。



6. 表示する指標をクリックし、期間（分、時間、日、週、月）を設定します。

7. グラフ化されたメトリクスをクリックして、使用量の統計を表示します。グラフをカスタマイズするには、[グラフ] オプションを使用します。

フィギュア。CPU 使用率に関するグラフ化されたメトリック



高可用性セットアップでの **SR-IOV** の設定

高可用性セットアップでの SR-IOV インターフェイスのサポートは、Citrix ADC リリース 12.0 57.19 以降から利用できます。SR-IOV を構成する方法の詳細については、「SR-IOV ネットワークインターフェイスを使用するように Citrix ADC VPX インスタンスを構成する」を参照してください。

関連リソース

AWS での高可用性の仕組み

異なる **AWS** アベイラビリティーゾーンでの高可用

January 25, 2022

独立ネットワーク構成（INC）モードでは、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンに 2 つの Citrix ADC VPX インスタンスを高可用性アクティブ/パッシブのペアとして構成できます。何らかの理由で 1 次ノードが接続を受け入れることができない場合は、2 次ノードが引き継ぎます。

高可用性の詳細については、「高可用性」を参照してください。INC の詳細については、「異なるサブネットでの高可用性ノードの設定」を参照してください。

注意すべきポイント

- 配置を開始する前に、次のドキュメントをお読みください。
  - AWS 用語
  - 前提条件
  - 制限事項と使用上のガイドライン
- VPX 高可用性ペアは、異なるサブネットの同じアベイラビリティーゾーンに存在することも、2 つの異なる AWS アベイラビリティーゾーンに存在することもできます。
- 管理（NSIP)、クライアントトラフィック（VIP)、バックエンドサーバー（SNIP）には異なるサブネットを使用することをお勧めします。
- フェイルオーバーを機能させるには、独立ネットワーク構成 (INC) モードで高可用性を設定する必要があります。
- 2 つのインスタンスでは、ハートビートに使用される UDP トラフィック用にポート 3003 が開いている必要があります。
- 残りの API が機能するように、両方のノードの管理サブネットは、内部 NAT を介してインターネットまたは AWS API サーバーにアクセスできる必要があります。
- IAM ロールには、パブリック IP または Elastic IP (EIP) 移行用の E2 アクセス権限と、プライベート IP 移行用の EC2 ルートテーブルのアクセス許可が必要です。

次の方法で AWS アベイラビリティーゾーン間で高可用性をデプロイできます。

- エラスティック IP アドレスの使用
- プライベート IP アドレスの使用

異なる **AWS** ゾーンに **Elastic IP** アドレスを使用した **VPX** 高可用性ペアをデプロイする

January 31, 2022

INC モードでエラスティック IP アドレスを使用して、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンで 2 つの Citrix ADC VPX インスタンスを設定できます。

高可用性の詳細については、「高可用性」を参照してください。INC の詳細については、「異なるサブネットでの高可用性ノードの設定」を参照してください。

異なる **AWS** ゾーンにわたる **EIP** アドレスを持つ **HA** のしくみ

フェールオーバー時には、プライマリインスタンスの VIP の EIP がセカンダリに移行し、セカンダリが新しいプライマリとして引き継がれます。フェイルオーバープロセスでは、AWS API は以下を行います。

1. `IPSets`が接続されている仮想サーバーをチェックします。

2. 仮想サーバーがリッスンしている 2 つの IP アドレスから、パブリック IP が関連付けられている IP アドレスを検索します。1 つは仮想サーバに直接接続され、もう 1 つは IP セットを介して接続されます。
3. パブリック IP (EIP) を、新しいプライマリ VIP に属するプライベート IP に再関連付けします。

> 注
>
> EIP を使用する際に、サービス拒否 (DoS) などの攻撃からネットワークを保護するために、AWS でセキュリティグループを作成して IP アクセスを制限できます。高可用性を実現するために、展開に従って EIP からプライベート IP 移動ソリューションに切り替えることができます。

異なる **AWS** ゾーン間でエラスティック **IP** アドレスを使用して **VPX** 高可用性ペアをデプロイする方法

VPX ペアを 2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンにデプロイする手順の概要を以下に示します。

1. Amazon 仮想プライベートクラウドを作成します。
2. 2 つの VPX インスタンスを、2 つの異なるアベイラビリティーゾーン、または同じゾーンで異なるサブネットにデプロイします。
3. 高可用性の構成
   a) 両方のインスタンスで INC モードで高可用性をセットアップします。
   b) 両方のインスタンスに IP セットを追加します 。
   c) 両方のインスタンスの IP セットを VIP にバインドします。
   d) プライマリ・インスタンスに仮想サーバを追加します。

ステップ 1 と 2 では、AWS コンソールを使用します。手順 3 では、Citrix ADC VPX GUI または CLI を使用します。

手順 **1**: Amazon 仮想プライベートクラウド (VPC) を作成します。

手順 **2**: 2 つの VPX インスタンスを 2 つの異なるアベイラビリティーゾーン、または同じゾーンで異なるサブネットにデプロイします。プライマリ VPX の VIP に EIP を接続します。

VPC を作成して AWS に VPX インスタンスをデプロイする方法の詳細については、「AWS への Citrix ADC VPX スタンドアロンインスタンスのデプロイ 」および「シナリオ: スタンドアロンインスタンス」を参照してください。

手順 **3**: 高可用性を構成します。Citrix ADC VPX CLI または GUI を使用して、高可用性をセットアップできます。

CLI を使用した高可用性の設定

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

   プライマリノードで、次の操作を行います。

   ```
   add ha node 1 <sec_ip> -inc ENABLED
   ```

   セカンダリノード:

   ```
   add ha node 1 <prim_ip> -inc ENABLED
   ```


<sec_ip>はセカンダリノードの管理 NIC のプライベート IP アドレスを指します。

<prim_ip>はプライマリノードの管理 NIC のプライベート IP アドレスを指します

2. 両方のインスタンスに IP セットを追加します。

   両方のインスタンスで以下のコマンドを入力します。

   ```
   add ipset <ipsetname>
   ```

3. IP セットを両方のインスタンスの VIP セットにバインドします。

   両方のインスタンスで以下のコマンドを入力します。

   ```
   add ns ip <secondary vip> <subnet> -type VIP
   ```

   ```
   bind ipset <ipsetname> <secondary VIP>
   ```

   > 注
   >
   > IP セットは、プライマリ VIP またはセカンダリ VIP にバインドできます。ただし、IP セットをプライマリ VIP にバインドする場合は、セカンダリ VIP を使用して仮想サーバに追加し、逆にセカンダリ VIP を使用します。

4. プライマリ・インスタンスに仮想サーバを追加します。

   次のコマンドを入力します。

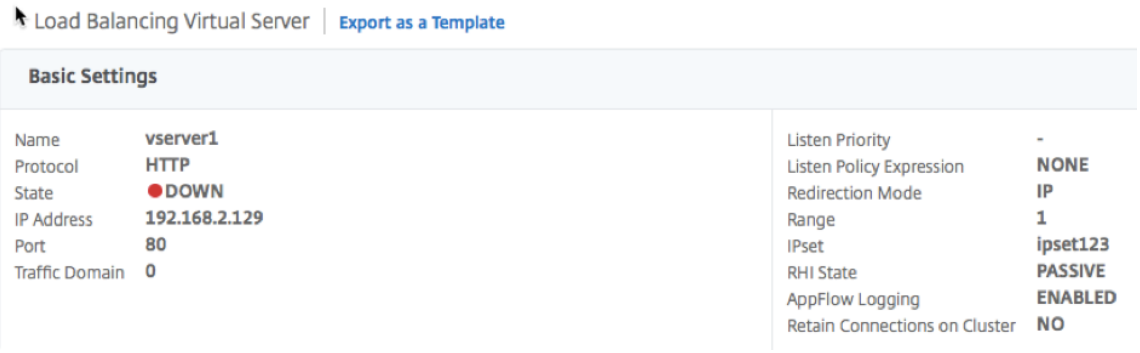
   ```
   add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port> -ipset \<ipset_name>
   ```

GUI を使用した高可用性の構成

1. 両方のインスタンスで INC モードで高可用性をセットアップする
2. ユーザー名nsrootとインスタンス ID をパスワードとしてプライマリノードにログオンします。
3. GUI から、[設定**] > [**** システム]**> [**** ハイアベイラビリティ] に移動します。[追加] をクリックします。
4. [リモートノード **IP** アドレス] フィールドに、セカンダリノードの管理 NIC のプライベート IP アドレスを追加します。
5. [セルフノードで **NIC (**独立ネットワーク構成**)** モードをオンにする] を選択します。
6. [リモートシステムログイン認証情報] で、セカンダリノードのユーザー名とパスワードを追加し、[作成] をクリックします。
7. セカンダリノードで手順を繰り返します。
8. IP セットを追加し、IP セットを両方のインスタンスの VIP セットにバインドします。
9. GUI から、[システム] > [ネットワーク] > [**IP**] > **[**追加**]** に移動します。
10. [IP アドレス]、[ネットマスク]、[IP タイプ (仮想 IP)] に必要な値を追加し、[作成] をクリックします。

11. システム > ネットワーク > **IP** セット > **Add** にナビゲートして下さい。IP セット名を追加し、[**Insert**] をクリックします。
12. [IPv4] ページで、仮想 IP を選択し、[挿入] をクリックします。[**Create**] をクリックして IP セットを作成します。
13. プライマリ・インスタンスに仮想サーバを追加する

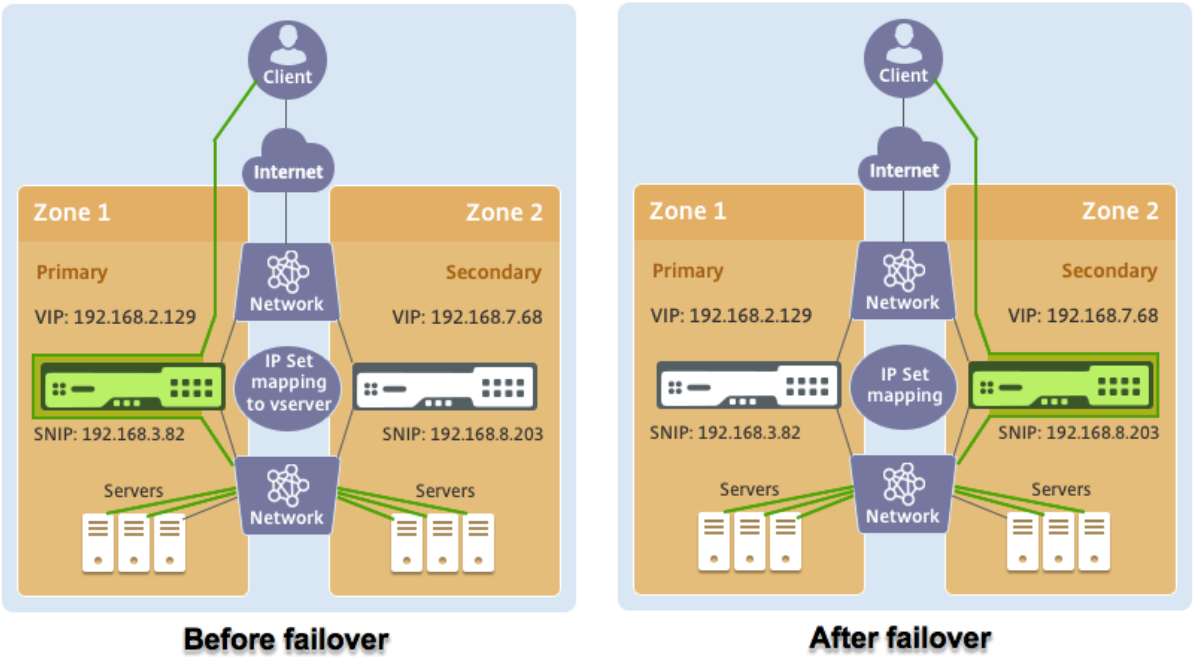
    GUI から、[構成]> [トラフィック管理] > [仮想サーバ] **>** [追加] に移動します。



シナリオ

このシナリオでは、1 つの VPC が作成されます。その VPC では、2 つのアベイラビリティーゾーンに 2 つの VPX インスタンスが作成されます。各インスタンスには、管理用、クライアント用、バックエンドサーバー用の 3 つのサブネットがあります。EIP はプライマリノードの VIP に接続されます。

図：この図は、AWS での INC モードでの Citrix ADC VPX の高可用性セットアップを示しています



このシナリオでは、CLI を使用して高可用性を設定します。

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

   プライマリノードとセカンダリノードで次のコマンドを入力します。

   プライマリ:

   ```
   add ha node 1 192.168.6.82 -inc enabled
   ```

   ここで、192.168.6.82 は、セカンダリノードの管理 NIC のプライベート IP アドレスを指します。

   セカンダリ:

   ```
   add ha node 1 192.168.1.108 -inc enabled
   ```

   ここで、192.168.1.108 は、プライマリノードの管理 NIC のプライベート IP アドレスを指します。

2. IP セットを追加し、IP セットを両方のインスタンスの VIP にバインドします。

   プライマリ:

   ```
   add ipset ipset123
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   bindipset ipset123 192.168.7.68
   ```

   セカンダリ:

   ```
   add ipset ipset123
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   bind ipset ipset123 192.168.7.68
   ```

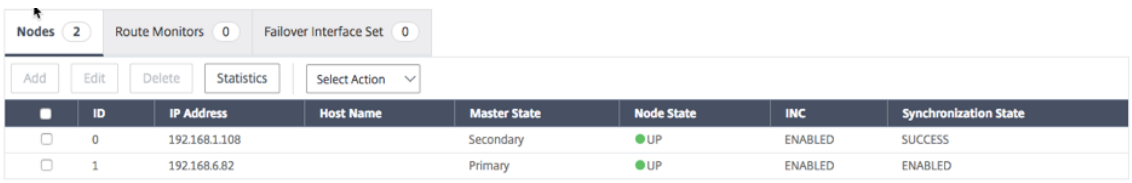
3. プライマリ・インスタンスに仮想サーバを追加します。

   以下のコマンドを実行します。

   ```
   add lbvserver vserver1 http 192.168.2.129 80 -ipset ipset123
   ```

4. 構成を保存します。

5. 強制フェールオーバーの後、セカンダリは新しいプライマリになります。



プライベート IP アドレスを使用した VPX 高可用性ペアを異なる AWS ゾーンにデプロイする

February 21, 2022

INC モードのプライベート IP アドレスを使用して、2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンで 2 つの Citrix ADC VPX インスタンスを設定できます。このソリューションは、Elastic IP アドレスを持つ既存のマルチゾーン VPX 高可用性ペアと簡単に統合できます。したがって、両方のソリューションを一緒に使用できます。

高可用性の詳細については、「高可用性」を参照してください。INC の詳細については、「異なるサブネットでの高可用性ノードの設定」を参照してください。

> 注:
>
> この展開は、Citrix ADC リリース 13.0 ビルド 67.39 以降でサポートされています。このデプロイは、AWS Transit Gateway および VPC ピアリングと互換性があります。

前提条件

AWS アカウントに関連付けられた IAM ロールに次の IAM アクセス権限があることを確認します。

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeAddresses",
                "ec2:AssociateAddress",
                "ec2:DisassociateAddress",
                "ec2:DescribeRouteTables",
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "iam:SimulatePrincipalPolicy",
                "iam:GetRole"
            ],
            "Resource": "*",
            "Effect": "Allow"
         }
```



```
22
23        ]
24    }
25
26
27  <!--NeedCopy-->
```

VPX 高可用性ペアをプライベート IP アドレスで異なる AWS ゾーンにデプロイする方法

次に、プライベート IP アドレスを使用して 2 つの異なるサブネットまたは 2 つの異なる AWS アベイラビリティーゾーンに VPX ペアをデプロイする手順の概要を示します。

1. Amazon 仮想プライベートクラウドを作成します。
2. 2 つの異なるアベイラビリティーゾーンに 2 つの VPX インスタンスをデプロイします。
3. 高可用性の構成
   a) 両方のインスタンスで INC モードで高可用性をセットアップします。
   b) クライアントインターフェイスを指すそれぞれのルートテーブルを VPC に追加します。
   c) プライマリ・インスタンスに仮想サーバを追加します。

ステップ 1 と 2 では、AWS コンソールを使用します。ステップ 3 では、Citrix ADC VPX GUI または CLI を使用します。

手順 **1**: Amazon 仮想プライベートクラウド (VPC) を作成します。

手順 **2**: 同じ数の ENI (ネットワークインターフェイス) を持つ 2 つの異なるアベイラビリティーゾーンに 2 つの VPX インスタンスをデプロイします。

VPC を作成して AWS に VPX インスタンスをデプロイする方法の詳細については、「AWS への Citrix ADC VPX スタンドアロンインスタンスのデプロイ」および「シナリオ: スタンドアロンインスタンス」を参照してください。

手順 **3**: Amazon VPC サブネットと重複しないサブネットを選択して、ADC VIP アドレスを設定します。VPC が 192.168.0.0/16 の場合、ADC VIP アドレスを設定するには、次の IP アドレス範囲から任意のサブネットを選択できます。

- 0.0.0.0-192.167.0.0
- 192.169.0.0-254.255.255.0

この例では、10.10.10.0/24 サブネットを選択し、このサブネットに VIP を作成しました。VPC サブネット以外の任意のサブネットを選択できます (192.168.0.0/16).

手順 **4**: VPC ルートテーブルから、プライマリノードのクライアントインターフェイス (VIP) を指すルートを追加します。

AWS CLI から、次のコマンドを入力します。


```
aws ec2 create-route --route-table-id rtb-2272532 --destination-cidr-
    block 10.10.10.0/24 --gateway-id <eni-client-primary>
<!--NeedCopy-->
```

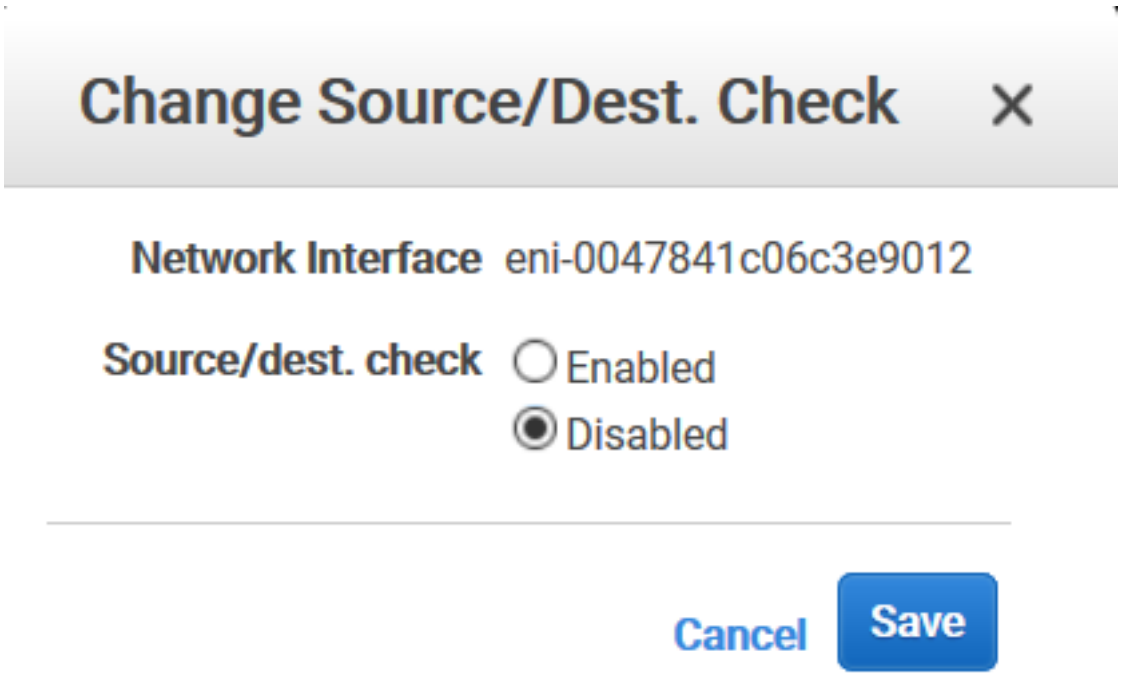
AWS GUI から、次の手順を実行してルートを追加します。

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、ルートテーブルを選択し、ルートテーブルを選択します。
3. [アクション] を選択し、[ルートの編集] をクリックします。
4. ルートを追加するには、[**Add route**] を選択します。[宛先] に、宛先 CIDR ブロック、単一の IP アドレス、またはプレフィックスリストの ID を入力します。ゲートウェイ ID には、プライマリノードのクライアントインターフェイスの ENI を選択します。

> 注
>
> プライマリ・インスタンスのクライアント ENI で **Source/Dest Check** を無効にする必要があります。

コンソールを使用してネットワークインターフェイスの source/destination チェックを無効にするには、次の手順を実行します。

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[ネットワークインターフェイス] を選択します。
3. プライマリクライアントインターフェイスのネットワークインターフェイスを選択し、[アクション] を選択し、[ソース/デストを変更] をクリックします。を確認してください。
4. ダイアログボックスで、[無効] を選択し、[保存] をクリックします。



手順 **5**：高可用性を構成します。Citrix ADC VPX CLI または GUI を使用して、高可用性をセットアップできます。

CLI を使用した高可用性の設定

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

   プライマリノードで、次の操作を行います。

   ```
   add ha node 1 <sec_ip> -inc ENABLED
   <!--NeedCopy-->
   ```

   セカンダリノード：

   ```
   add ha node 1 <prim_ip> -inc ENABLED
   <!--NeedCopy-->
   ```

   <sec_ip> セカンダリノードの管理 NIC のプライベート IP アドレスを参照します。

   <prim_ip> プライマリノードの管理 NIC のプライベート IP アドレスを参照します。

2. プライマリ・インスタンスに仮想サーバを追加します。選択したサブネット（10.10.10.0/24 など）から追加する必要があります。

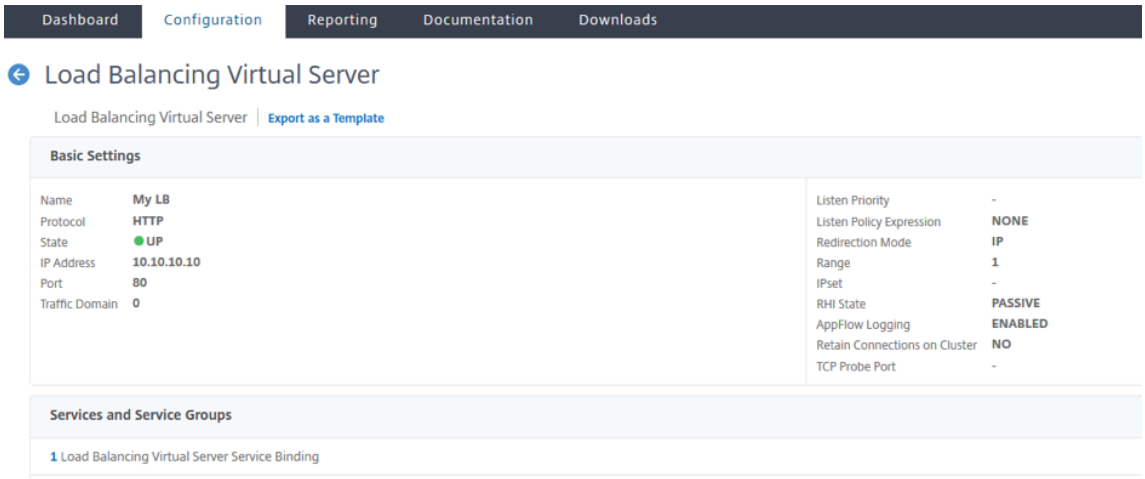
   次のコマンドを入力します。


```
add \<server\_type\> vserver \<vserver\_name\> \<protocol\> \<
   primary\_vip\> \<port\>
<!--NeedCopy-->
```

GUI を使用した高可用性の構成

1. 両方のインスタンスで INC モードで高可用性をセットアップする
2. ユーザー名nsrootとインスタンス ID をパスワードとしてプライマリノードにログオンします。
3. [構成] > [システム] > [高可用性] に移動し、[追加] をクリックします。
4. [リモートノード **IP** アドレス] フィールドに、セカンダリノードの管理 NIC のプライベート IP アドレスを追加します。
5. [セルフノードで **NIC (**独立ネットワーク構成**)** モードをオンにする] を選択します。
6. [リモートシステムログイン認証情報] で、セカンダリノードのユーザー名とパスワードを追加し、[作成] をクリックします。
7. セカンダリノードで手順を繰り返します。
8. プライマリ・インスタンスに仮想サーバを追加する

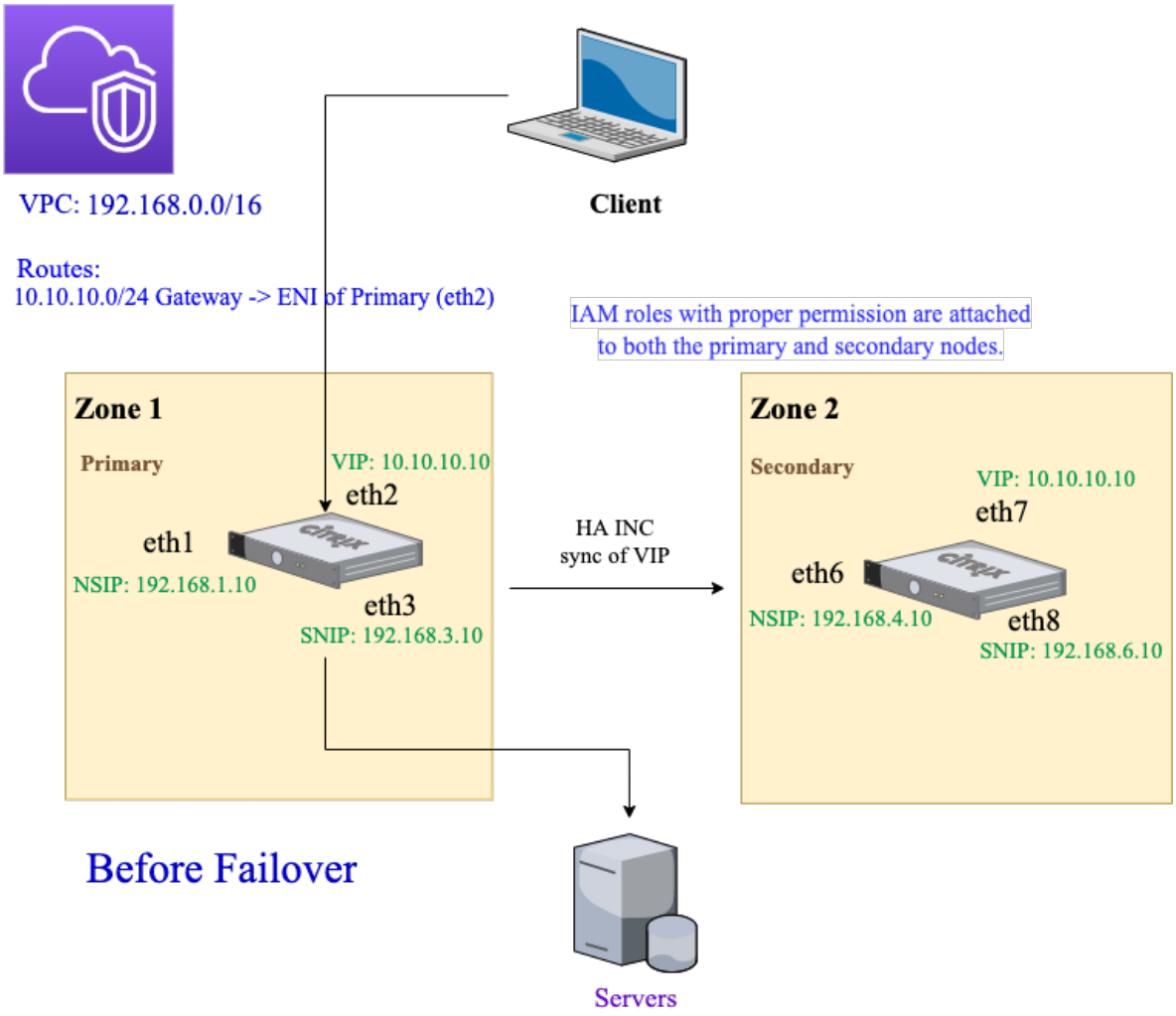
   [設定] > [トラフィック管理] > [仮想サーバー**] > [**追加**]** に移動します。

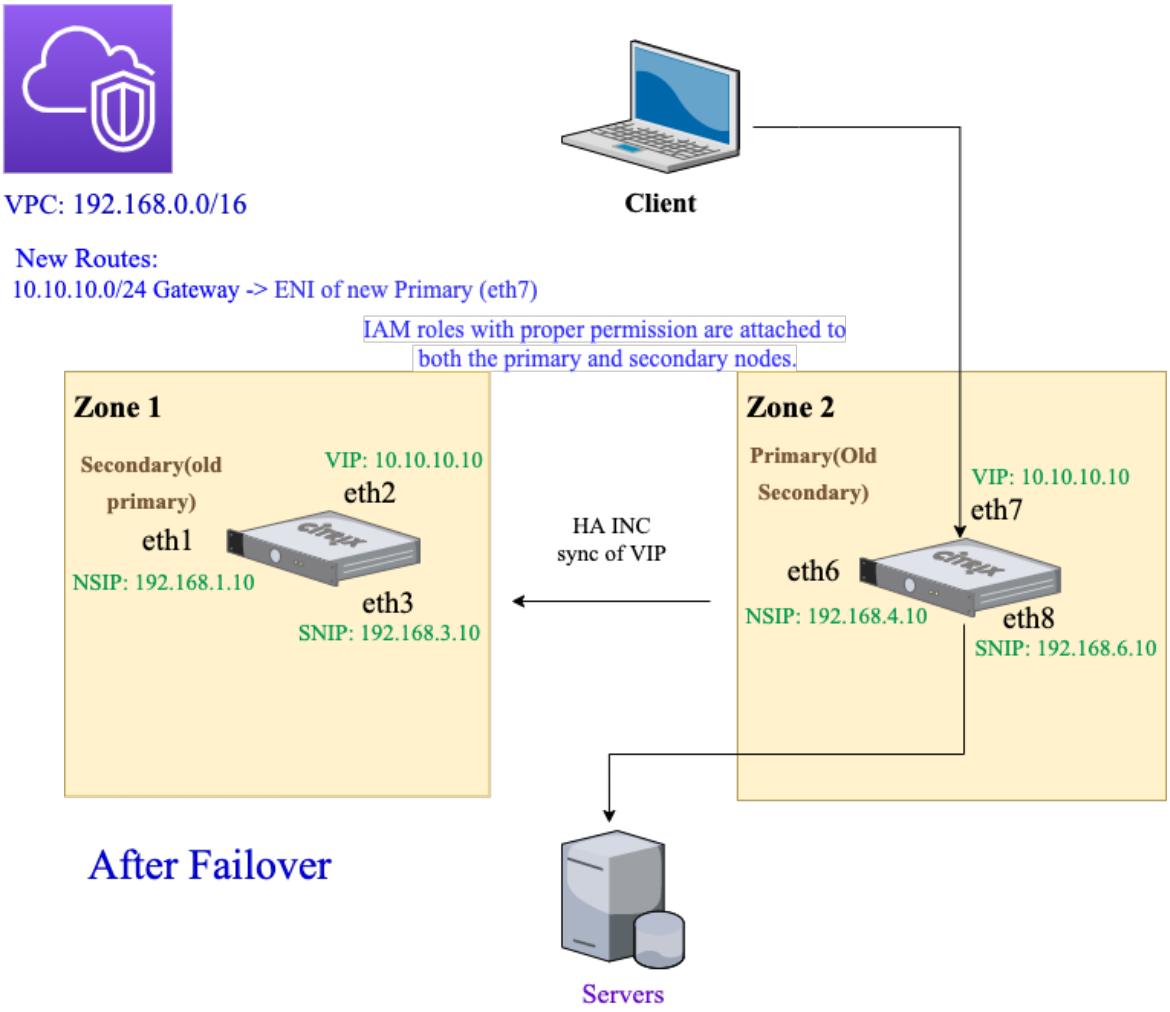


シナリオ

このシナリオでは、1 つの VPC が作成されます。その VPC では、2 つのアベイラビリティーゾーンに 2 つの VPX インスタンスが作成されます。各インスタンスには、管理用、クライアント用、バックエンドサーバー用の 3 つのサブネットがあります。

次の図は、AWS での INC モードでの Citrix ADC VPX 高可用性のセットアップを示しています。VPC の一部ではないカスタムサブネット 10.10.10.10 が VIP として使用されます。したがって、10.10.10.10 サブネットはアベイラビリティーゾーン全体で使用できます。





このシナリオでは、CLI を使用して高可用性を設定します。

1. 両方のインスタンスで INC モードで高可用性をセットアップします。

   プライマリノードとセカンダリノードで次のコマンドを入力します。

   プライマリノードで、次の操作を行います。

```
add ha node 1 192.168.4.10 -inc enabled
<!--NeedCopy-->
```

   ここで、192.168.4.10 は、セカンダリノードの管理 NIC のプライベート IP アドレスを指します。

   セカンダリノード:

```
add ha node 1 192.168.1.10 -inc enabled
<!--NeedCopy-->
```


ここで、192.168.1.10 は、プライマリノードの管理 NIC のプライベート IP アドレスを指します。

2. プライマリ・インスタンスに仮想サーバを追加します。

   次のコマンドを入力します。

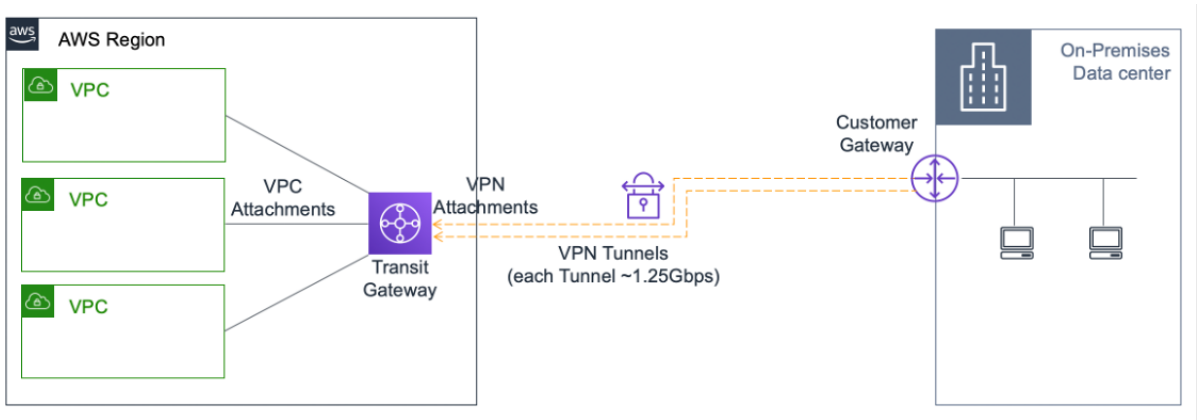
```
1 add lbvserver vserver1 http 10.10.10.10 80
2 <!--NeedCopy-->
```

3. 構成を保存します。
4. 強制フェールオーバーの後:

   - セカンダリインスタンスが新しいプライマリインスタンスになります。
   - プライマリ ENI を指す VPC ルートは、セカンダリクライアント ENI に移行します。
   - クライアントトラフィックは、新しいプライマリインスタンスに再開されます。

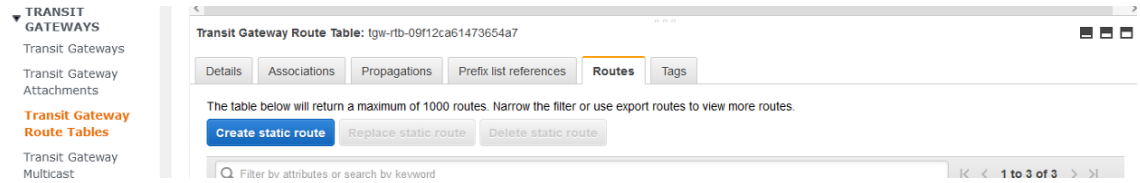
HA プライベート IP ソリューションの AWS Transit Gateway の設定

AWS Transit Gateway は、AWS VPC、リージョン、およびオンプレミスネットワーク全体で、内部ネットワーク内でプライベート VIP サブネットをルーティング可能にする必要があります。VPC は AWS Transit Gateway に接続する必要があります。AWS Transit Gateway ルートテーブル内の VIP サブネットまたは IP プールの静的ルートが作成され、VPC をポイントします。

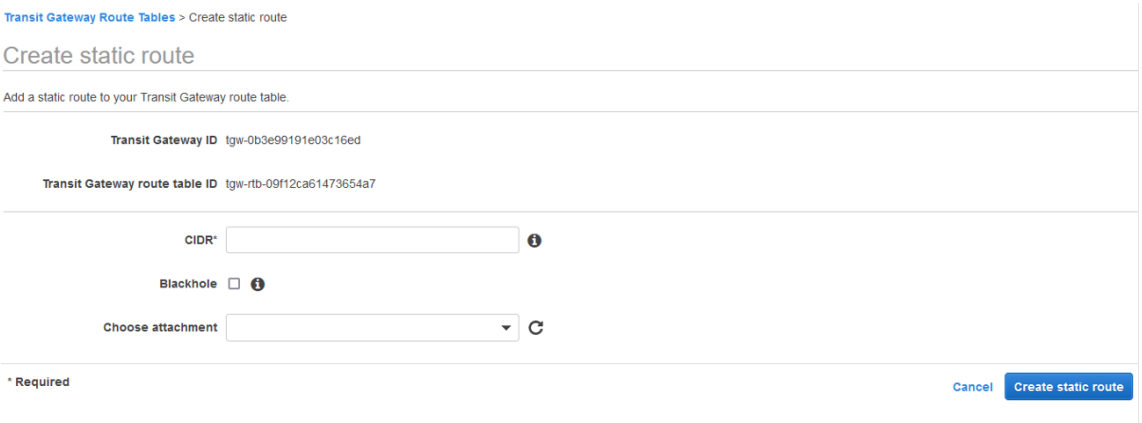


AWS Transit Gateway を設定するには、次の手順に従います。

1. Amazon VPC コンソールを開きます。
2. ナビゲーションペインで、[**Transit Gateway** ートテーブル] を選択します。
3. [ルート] タブを選択し、[静的ルートの作成] をクリックします。



4. CIDR がプライベート VIP サブネットを指し、アタッチメントが ADC VPX を持つ VPC を指す静的ルートを作成します。



5. [スタティックルートの作成] をクリックし、[閉じる] を選択します。

AWS アウトポストに Citrix ADC VPX インスタンスをデプロイする

October 7, 2021

AWS Outposts は、お客様のサイトにデプロイされた AWS コンピューティングおよびストレージ容量のプールです。Outposts は、オンプレミスの場所に AWS インフラストラクチャとサービスを提供しています。AWS は、この容量を AWS リージョンの一部として運用、監視、管理します。オンプレミスと AWS クラウドで同じ Citrix ADC VPX インスタンス、AWS API、ツール、インフラストラクチャを使用して、一貫性のあるハイブリッドエクスペリエンスを実現できます。

アウトポストにサブネットを作成し、EC2 インスタンス、EBS ボリューム、ECS クラスター、RDS インスタンスなどの AWS リソースを作成するときにサブネットを指定できます。Outposts サブネット内のインスタンスは、プライベート IP アドレスを使用して AWS リージョンの他のインスタンスと通信します。これらはすべて、同じ Amazon 仮想プライベートクラウド（VPC）内にあります。

詳細については、AWS Outposts ユーザーガイドを参照してください。

AWS アウトポストの仕組み

AWS Outposts は、お客様の Outposts と AWS リージョン間の一貫した接続を維持して運用できるように設計されています。このリージョンとオンプレミス環境のローカルワークロードに接続するには、Outpost をオンプレミス

ネットワークに接続する必要があります。オンプレミスのネットワークは、リージョンとインターネットへの WAN アクセスを提供する必要があります。また、インターネットは、オンプレミスのワークロードまたはアプリケーションが存在するローカルネットワークへの LAN または WAN アクセスを提供する必要があります。

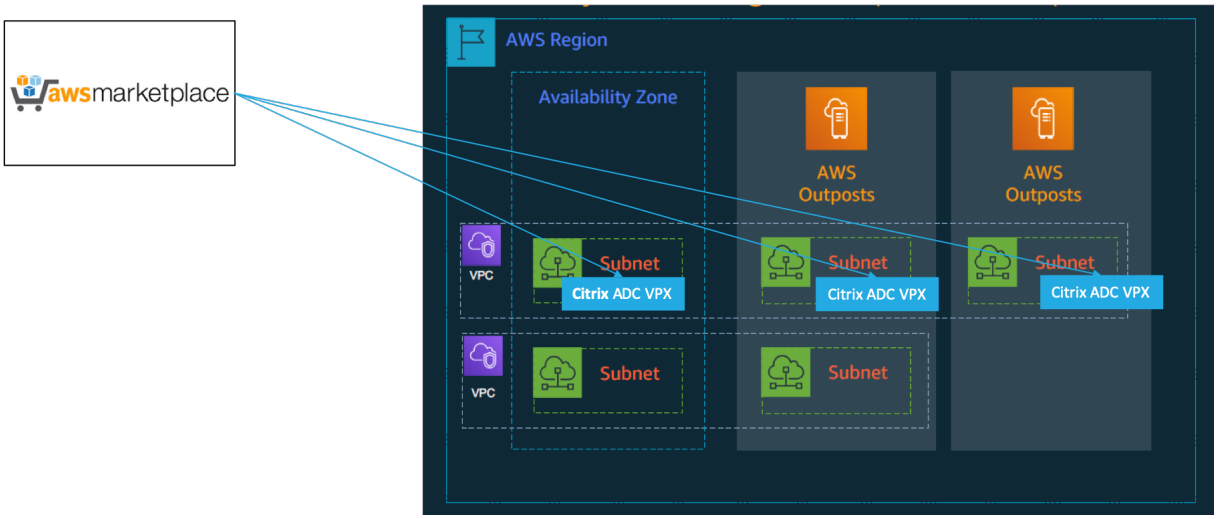
前提要件

- サイトに AWS Outpost をインストールする必要があります。
- AWS Outposts のコンピューティング容量とストレージ容量が使用可能である必要があります。

AWS Outposts の注文方法の詳細については、次の AWS ドキュメントを参照してください。
https://aws.amazon.com/blogs/aws/aws-outposts-now-available-order-your-racks-today/

AWS ウェブコンソールを使用して、**AWS** アウトポストに **Citrix ADC VPX** インスタンスをデプロイする

次の図は、Citrix ADC VPX インスタンスのアウトポストへの簡単な展開を示しています。AWS マーケットプレイスに存在する Citrix ADC AMI は、アウトポストにもデプロイされます。



AWS ウェブコンソールにログインし、以下の手順を実行して、ADC VPX EC2 インスタンスを AWS アウトポストにデプロイします。

1. キーペアを作成します。
2. 仮想プライベートクラウド（VPC）を作成します。
3. サブネットをさらに追加します。
4. セキュリティグループとセキュリティルールを作成します。
5. ルートテーブルを追加します。
6. インターネット Gateway を作成します。
7. AWS EC2 サービスを使用して ADC VPX インスタンスを作成します。
   AWS ダッシュボードから、[コンピューティング] **>** **[EC2]** **>** [インスタンスの起動] **>** **[AWS** マーケットプレイス] に移動します。

8. より多くのネットワークインターフェイスを作成し、接続します。
9. 管理 NIC にエラスティック IP を接続します。
10. VPX インスタンスに接続します。

各手順の詳細については、「AWS ウェブコンソールを使用して AWS に Citrix ADC VPX インスタンスをデプロイする」を参照してください。

同じアベイラビリティーゾーンのデプロイ内での高可用性については、「AWS に高可用性ペアをデプロイする」を参照してください。

バックエンドの **AWS Autoscaling** サービスを追加する

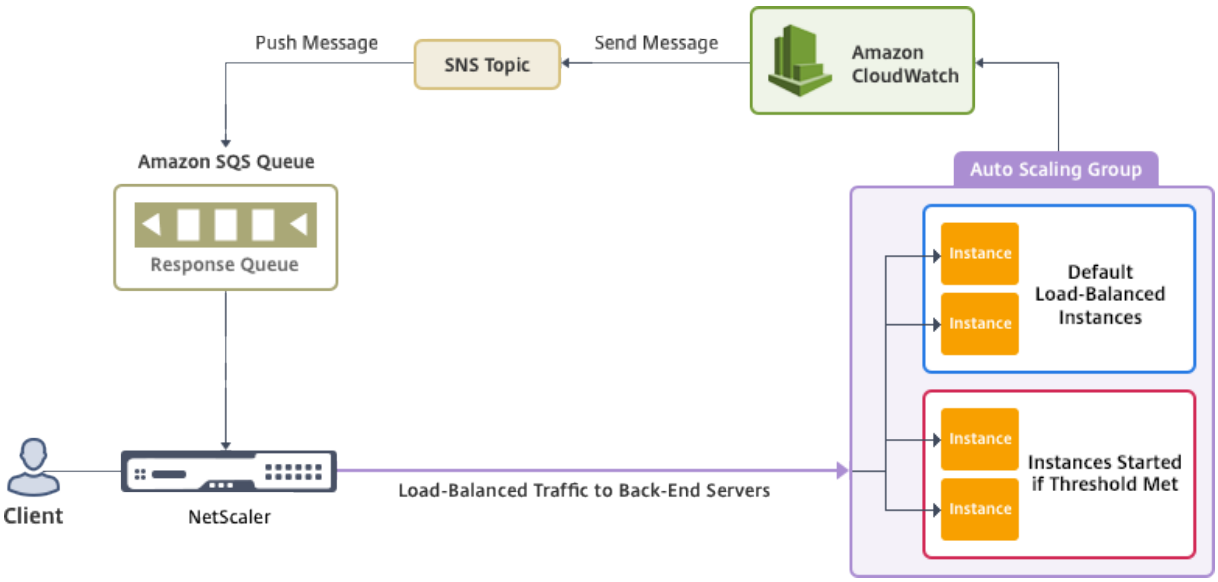
October 7, 2021

クラウドでアプリケーションを効率的にホストすると、アプリケーションの必要に応じて、リソースを簡単にコスト効率よく管理できます。需要の増大に対応するには、ネットワークリソースをスケールアップする必要があります。アイドル状態のリソースに不要なコストを費やさないようにするには、需要が低下しているかどうかに関わらず、スケールダウンが必要です。特定の時間に必要な数のインスタンスのみをデプロイすることで、アプリケーションの実行コストを最小限に抑えるには、トラフィック、メモリ、CPU の使用量などを常に監視する必要があります。しかし、トラフィックを手動で監視するのは煩雑です。アプリケーション環境を動的にスケールアップまたはスケールダウンするには、トラフィックの監視プロセスを自動化し、必要に応じてリソースを増減する必要があります。

AWS Auto Scaling サービスと統合され、Citrix ADC VPX インスタンスには次の利点があります。

- 負荷分散と管理: 需要に応じて、サーバーのスケールアップとスケールダウンを自動構成します。VPX インスタンスは、バックエンドサブネット内の Autoscale グループを自動検出し、ユーザーは Autoscale グループを選択して負荷を分散できます。このすべては、VPX インスタンスの仮想 IP アドレスとサブネット IP アドレスを自動構成することによって行われます。
- 高可用性: 複数のアベイラビリティーゾーンおよび負荷分散サーバーにまたがる Autoscale グループを検出します。
- ネットワーク可用性の向上: VPX インスタンスは次の機能をサポートします。
  - VPC ピアを使用した異なる VPC のバックエンドサーバー
  - 同じ配置グループのバックエンドサーバー
  - 異なるアベイラビリティゾーンのバックエンドサーバー
- 正常な接続終了**:** グレースフルタイムアウト機能を使用して、Autoscale サーバーを正常に削除し、スケールダウンアクティビティが発生したときにクライアント接続が失われないようにします。

図: Citrix ADC VPX インスタンスを使用した AWS Autoscaling サービス



この図は、AWS Auto Scaling サービスと Citrix ADC VPX インスタンス（負荷分散仮想サーバー）との互換性を示しています。詳しくは、次の AWS のトピックを参照してください。

- Autoscaling グループ
- CloudWatch
- Simple Notification Service（SNS）
- Simple Queue Service (Amazon SQS)

はじめに

Citrix ADC VPX インスタンスで Autoscaling を使用する前に、次のタスクを完了する必要があります。

1. 次のトピックをお読みください。
   - 前提条件
   - 制限と使用上のガイドライン
2. 要件に応じて、AWS で Citrix ADC VPX インスタンスを作成します。
   - Citrix ADC VPX スタンドアロンインスタンスの作成方法の詳細については、「AWS への Citrix ADC VPX スタンドアロンインスタンスのデプロイ 」および「シナリオ：スタンドアロンインスタンス」を参照してください。
   - VPX インスタンスを HA モードでデプロイする方法の詳細については、「AWS に高可用性ペアをデプロイする」を参照してください。

> 注:
>
> AWS で Citrix ADC VPX インスタンスを作成する場合は、CloudFormation テンプレートを使用することをお勧めします。
>
> 管理用（NSIP）、クライアント側 LB 仮想サーバー（VIP）用、サブネット IP（NSIP）用の 3 つのイン

> ターフェイスを作成することをお勧めします。

3. AWS Autoscale グループを作成します。既存の Auto Scaling 設定がない場合は、次の操作を行う必要があります。

   a) 起動構成を作成する

   b) Autoscaling グループの作成

   c) Autoscaling グループの確認

   詳しくは、http://docs.aws.amazon.com/autoscaling/latest/userguide/GettingStartedTutorial.htmlを参照してください。

4. AWS Autoscale グループでは、少なくとも 1 つのスケールダウンポリシーを指定する必要があります。Citrix ADC VPX インスタンスは、ステップスケーリングポリシーのみをサポートします。自動スケールグループでは、簡易スケーリングポリシーとターゲット追跡スケーリングポリシーはサポートされていません。

Citrix ADC VPX インスタンスへの AWS Autoscaling サービスの追加

GUI を使用して、ワンクリックで Autoscaling サービスを VPX インスタンスに追加できます。以下の手順を実行して、Autoscaling サービスを VPX インスタンスに追加します。

1. nsrootの資格情報を使用して、VPX インスタンスにログオンします。
2. Citrix ADC VPX インスタンスに初めてログオンすると、デフォルトのクラウドプロファイルページが表示されます。ドロップダウンメニューから AWS Auto Scaling グループを選択し、[**Create**] をクリックしてクラウドプロファイルを作成します。クラウドプロファイルを後で作成する場合は、[スキップ] をクリックします。

   クラウドプロファイルの作成時に留意すべきポイント**:** デフォルトでは、CloudFormation テンプレートによって以下の IAM ロールが作成され、アタッチされます。

```
{


    ”Version”: ”2012-10-17”,

    ”Statement”: [

     {


            ”Action”: [

                ”ec2:DescribeInstances”,

```



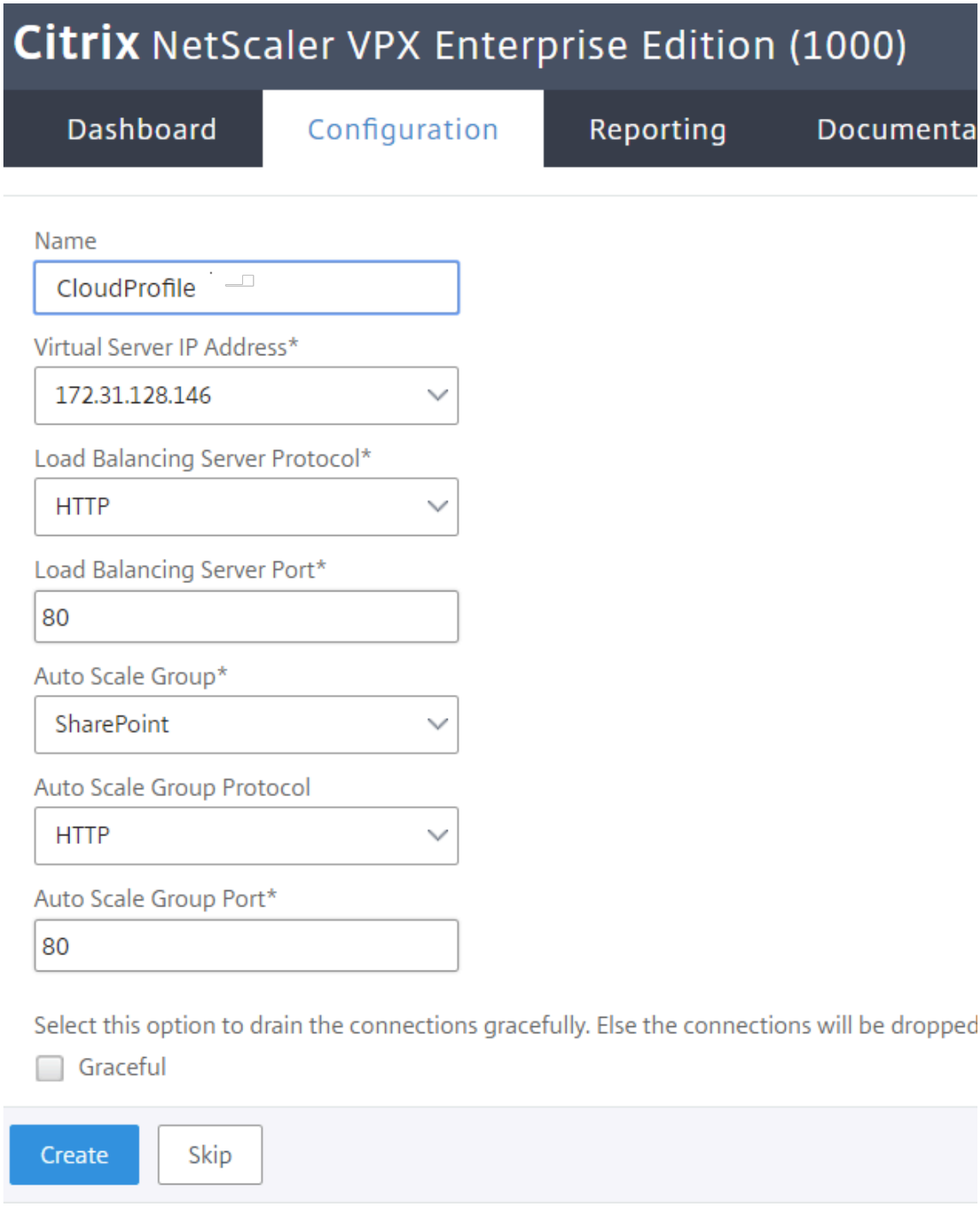
```
15                  "ec2:DescribeNetworkInterfaces",
16
17                  "ec2:DetachNetworkInterface",
18
19                  "ec2:AttachNetworkInterface",
20
21                  "ec2:StartInstances",
22
23                  "ec2:StopInstances",
24
25                  "ec2:RebootInstances",
26
27                  "autoscaling:*",
28
29                  "sns:*",
30
31                  "sqs:*"
32
33               "iam: SimulatePrincipalPolicy"
34
35               "iam: GetRole"
36
37              ],
38
39              "Resource": "*",
40
41              "Effect": "Allow"
42
43          }
44
45
46      ]
47
48  }
49
50 <!--NeedCopy-->
```

インスタンスの IAM ロールに適切な権限があることを確認します。

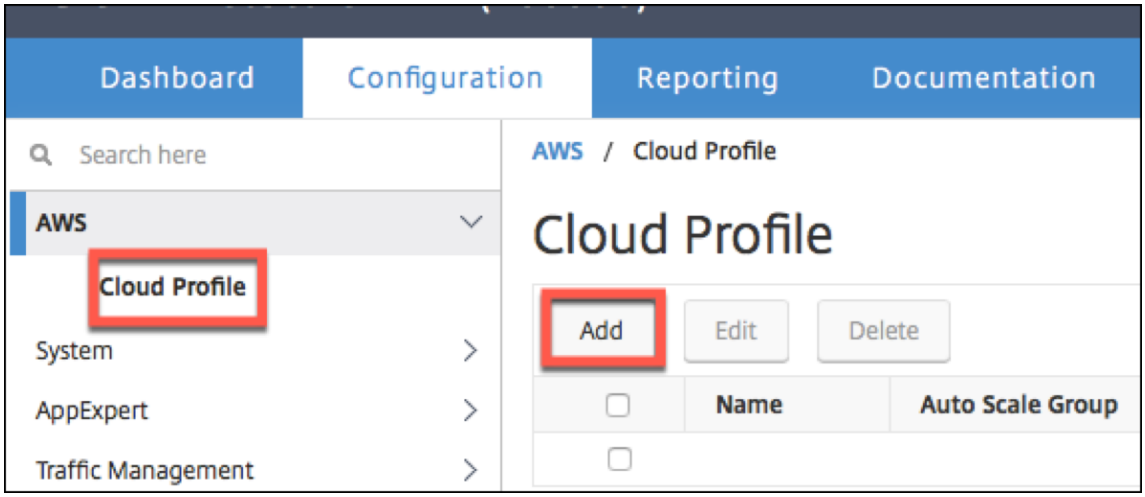
- 仮想サーバーの IP アドレスは、VPX インスタンスで使用可能な空き IP アドレスから自動入力されます。https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/MultipleIP.html#ManageMultipleIP
- Autoscale グループは、AWS アカウントで設定された Autoscale グループから事前に設定されています。http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroup.html。

- Autoscaling Group プロトコルとポートを選択するときに、サーバーがそれらのプロトコルとポートをリッスンし、サービスグループで正しいモニタをバインドします。デフォルトでは、TCP モニターが使用されます。
- SSL プロトコルタイプ Autoscaling の場合、クラウドプロファイルを作成した後、証明書がないためにロードバランシング仮想サーバーまたはサービスグループがダウンしています。証明書は、仮想サーバまたはサービスグループに手動でバインドできます。
- Autoscale サーバを正常に削除するには、[グレースフルタイムアウト] オプションを選択します。このオプションが選択されていない場合、サーバはロードがダウンした直後に Autoscale グループが削除されます。これにより、既存の接続クライアントのサービスが中断される可能性があります。Graceful を選択してタイムアウトを発生させるのはスケールダウンの場合です。VPX インスタンスはサーバーをすぐに削除するのではなく、サーバーの 1 つに正常な削除のマークを付けます。この間、インスタンスではこのサーバーへの新しい接続は許可されません。既存の接続は、タイムアウトが発生するまで処理され、タイムアウト後に VPX インスタンスがサーバーを削除します。

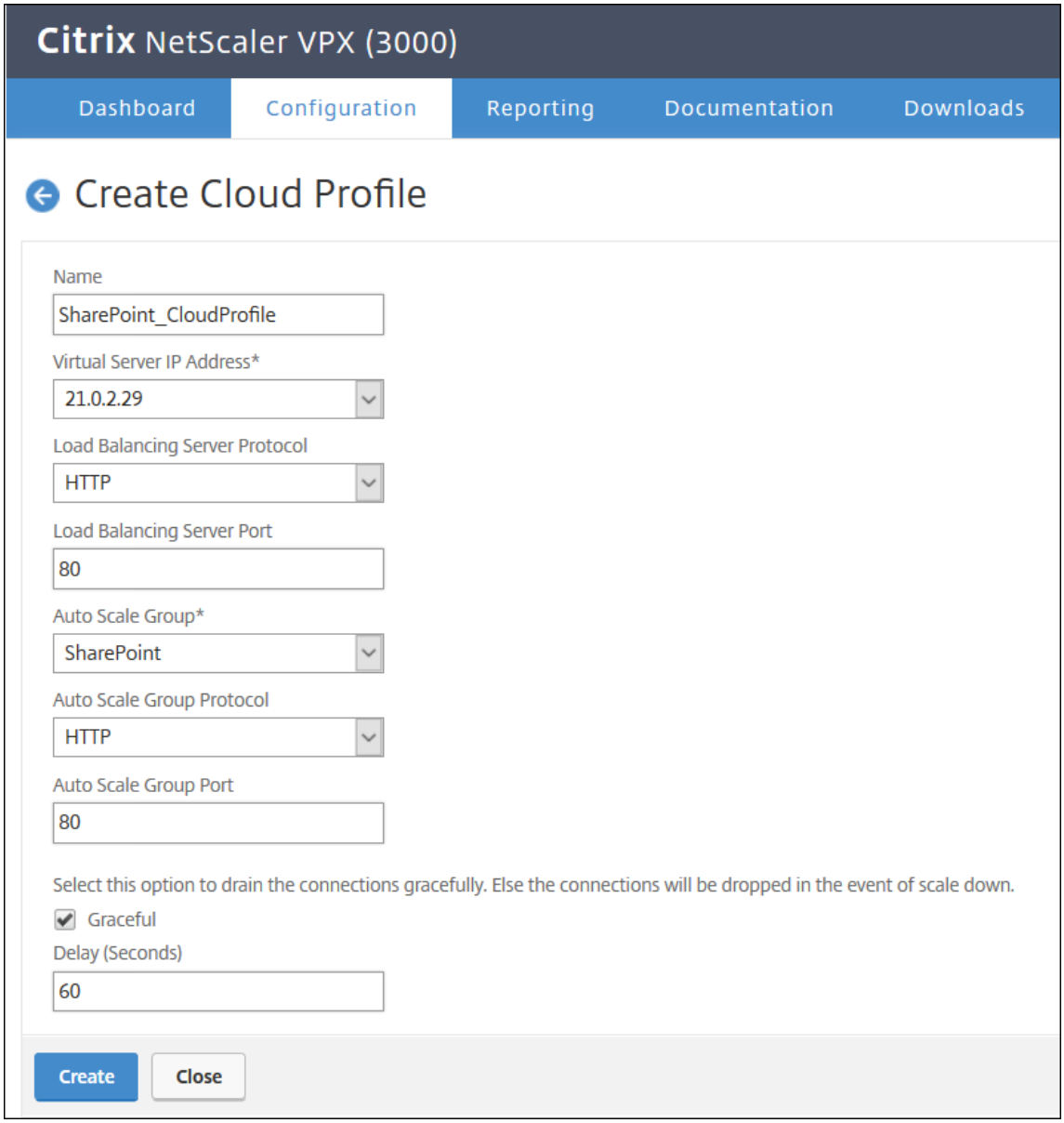
図: デフォルトのクラウドプロファイルページ



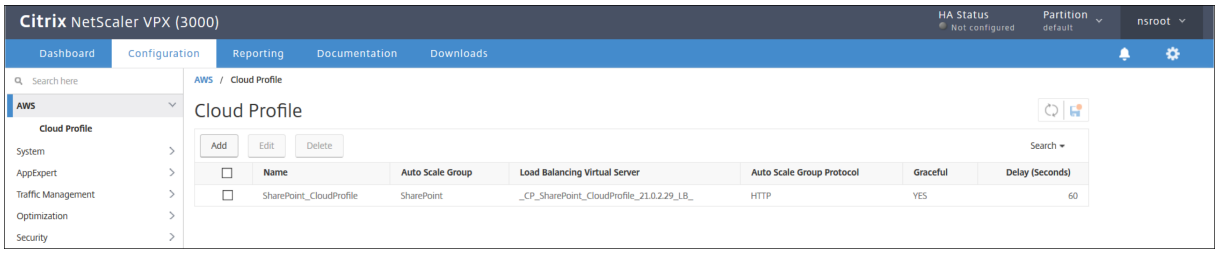
3. クラウドプロファイルを作成する場合は、初回ログオン後、GUI で [システム] **> [AWS] > [**クラウドプロファイル] に移動し、[追加] をクリックします。



[クラウドプロファイルの作成] 設定ページが表示されます。



クラウドプロファイルは、Citrix ADC 負荷分散仮想サーバーと、Autoscaling グループのサーバーとしてメンバーを持つサービスグループを作成します。バックエンドサーバーは、VPX インスタンスで構成された SNIP を介して到達可能である必要があります。



> 注
>
> AWS コンソールで Autoscale 関連の情報を表示するには、[**EC2**] > [ダッシュボード] > [Auto Scaling] > [Auto Scaling] > [**Auto Scaling** グループ] の順に選択します。

SR-IOV ネットワークインターフェイスを使用するように **Citrix ADC VPX** インスタンスを構成する

October 7, 2021

> 注
>
> 高可用性セットアップでの SR-IOV インターフェイスのサポートは、Citrix ADC リリース 12.0 57.19 以降から利用できます。

AWS で Citrix ADC VPX インスタンスを作成した後、AWS CLI を使用して、SR-IOV ネットワークインターフェイスを使用するように仮想アプライアンスを構成できます。

Citrix ADC VPX 3G および 5G の Citrix ADC VPX AWS マーケットプレイスエディションを除き、すべての Citrix ADC VPX モデルでは、ネットワークインターフェイスのデフォルト構成で SR-IOV が有効になっていません。

設定を開始する前に、次のトピックをお読みください。

- 前提条件
- 制限事項および使用上のガイドライン

このセクションでは、以下のトピックについて説明します。

- インターフェイスタイプを SR-IOV に変更します。
- 高可用性セットアップでの SR-IOV の設定

インターフェイスタイプを **SR-IOV** に変更します

show interface summary コマンドを実行すると、ネットワークインターフェイスのデフォルト設定を確認できます。

例 **1**: 次の CLI スクリーンキャプチャは、Citrix ADC VPX AWS Marketplace エディションの 3G および 5G で SR-IOV がデフォルトで有効になっているネットワークインターフェイスの構成を示しています。

```
> show interface summary
-------------------------------------------------------------------------------
      Interface  MTU       MAC                Suffix
-------------------------------------------------------------------------------
1       1/1      1500      0a:1e:2e:17:a2:37  Intel 82599 10G VF Interface
2       LO/1     1500      0a:1e:2e:17:a2:37  Netscaler Loopback interface
 Done
```

例 **2:** 次の CLI 画面キャプチャは、SR-IOV が有効になっていないネットワークインターフェイスのデフォルト設定を示しています。

```
 Done
> sh int s
-------------------------------------------------------------------------------
      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------------
1     1/1        1500       12:fc:04:c5:d0:12  NetScaler Virtual Interface
2     LO/1       1500       12:fc:04:c5:d0:12  Netscaler Loopback interface
 Done
>
```

インターフェイスの種類を SR-IOV に変更する方法の詳細については、http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.htmlを参照してください

インターフェイスタイプを **SR-IOV** に変更するには

1. AWS で実行されている Citrix ADC VPX インスタンスをシャットダウンします。
2. ネットワークインターフェイスで SR-IOV を有効にするには、次のコマンドを AWS CLI に入力します。

   ```
   $ aws ec2 modify-instance-attribute --instance-id \<instance\\_id\> --sriov-net-support simple
   ```

3. SR-IOV が有効にされたかどうか確認するには、次のコマンドを AWS CLI に入力します。

   ```
   $ aws ec2 describe-instance-attribute --instance-id \<instance\\_id\> --attribute sriovNetSupport
   ```

   例 **3:** AWS CLI を使用して、ネットワークインターフェイスの種類が SR-IOV に変更されました。

   ```
   aws ec2 modify-instance-attribute --instance-id i-008c1230aaf303bee --sriov-net-support simple
   aws ec2 describe-instance-attribute --instance-id   i-008c1230aaf303bee  --attribute sriovNetSupport
   {
       "InstanceId": "i-008c1230aaf303bee",
       "SriovNetSupport": {
           "Value": "simple"
       }
   }
   ```

   SR-IOV が有効になっていない場合、SriovNetSupport の値は存在しません。

   例 **4:** 次の例では、SR-IOV サポートが有効になっていません。

   ```
   {
       "InstanceId": "i-0c3e84cfa65b04cc8",
       "SriovNetSupport": {}
   }
   ```

4. VPX インスタンスの電源を入れます。ネットワークインターフェイスの変更されたステータスを確認するには、CLI で「show interface summary」と入力します。

例 **5:** 次の画面キャプチャは、SR-IOV が有効になっているネットワークインターフェイスを示しています。インターフェイス 10/1、10/2、10/3 で SR-IOV が有効にされています。

```
> show interface summary
-------------------------------------------------------------------------------
      Interface  MTU       MAC               Suffix
-------------------------------------------------------------------------------
1     10/1       1500      0a:1e:2e:17:a2:37 Intel 82599 10G VF Interface
2     10/2       1500      0a:df:17:0a:fe:83 Intel 82599 10G VF Interface
3     10/3       1500      0a:de:5d:31:bf:c3 Intel 82599 10G VF Interface
4     LO/1       1500      0a:1e:2e:17:a2:37 Netscaler Loopback interface
 Done
```

これらの手順では、SR-IOV ネットワークインターフェイスを使用するように VPX インスタンスを構成する手順を完了します。

高可用性セットアップで **SR-IOV** を構成する

高可用性は、Citrix ADC リリース 12.0 ビルド 57.19 以降の SR-IOV インターフェイスでサポートされています。

高可用性セットアップが手動で、または Citrix ADC バージョン 12.0 56.20 以前の Citrix CloudFormation テンプレートを使用してデプロイされた場合、高可用性セットアップにアタッチされた IAM ロールには次の権限が必要です。

- ec2:DescribeInstances
- ec2:DescribeNetworkInterfaces
- ec2:DetachNetworkInterface
- ec2:AttachNetworkInterface
- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- autoscaling:*
- sns: *
- sqs:*
- IAM: プリンシパルポリシーのシミュレーション
- IAM:GetRole

デフォルトでは、Citrix ADC バージョン 12.0 57.19 用の Citrix CloudFormation テンプレートによって、必要な権限が IAM ロールに自動的に追加されます。

> 注
>
> SR-IOV インターフェイスを使用したハイアベイラビリティのセットアップには、約 100 秒のダウンタイムが発生します。

関連リソース:

IAM ロールの詳細については、AWS ドキュメントを参照してください。

AWS ENA で拡張ネットワーキングを使用するよう Citrix ADC VPX インスタンスを構成する

October 7, 2021

AWS で Citrix ADC VPX インスタンスを作成した後、AWS CLI を使用して、AWS Elastic Network Adapter（ENA）を使用した拡張ネットワーキングを使用するように仮想アプライアンスを構成できます。

AWS ENA と組み合わせると、拡張ネットワーキングは、より高い帯域幅、高いパケット/秒（PPS）パフォーマンス、一貫して低いインスタンス間レイテンシーを提供します。

設定を開始する前に、次のトピックをお読みください。

- 前提条件
- 制限事項および使用上のガイドライン

ENA 対応インスタンスでは、次の HA 設定がサポートされています。

- プライベート IP アドレスは、同じアベイラビリティーゾーン内で移動できます。
- Elastic IP アドレスは、アベイラビリティーゾーン間で移動できます。

AWS で Citrix ADC VPX インスタンスをアップグレードする

October 7, 2021

EC2 インスタンスタイプ、スループット、ソフトウェアエディション、および AWS で実行されている Citrix ADC VPX のシステムソフトウェアをアップグレードできます。一部のアップグレード方法では、高可用性構成を使用してダウンタイムを最小限に抑えることができます。

> 注:
>
> - Citrix ADC VPX AMI（ユーティリティライセンスとカスタマーライセンスの両方を含む）用の Citrix ADC ソフトウェアリリース 10.1.e-124.1308.e 以降では、M1 および M2 インスタンスファミリーはサポートされません。
> - VPX インスタンスのサポートが変更されたため、10.1.e-124 以降のリリースから 10.1.123.x 以前のリリースへのダウングレードはサポートされていません。
> - ほとんどのアップグレードでは、新しい AMI を起動する必要はなく、アップグレードは現在の Citrix ADC AMI インスタンスで実行できます。新しい Citrix ADC AMI インスタンスにアップグレードする場合は、高可用性構成方法を使用します。

AWS で Citrix ADC VPX インスタンスの EC2 インスタンスタイプを変更する

Citrix ADC VPX インスタンスでリリース 10.1.e-124.1308.e 以降が実行されている場合は、次のように AWS コンソールから EC2 インスタンスタイプを変更できます。

1. VPX インスタンスを停止します。
2. AWS コンソールで EC2 インスタンスの種類を変更します。
3. インスタンスを起動します。

上記の手順は、Release 10.1.e-124.1308.e よりも前の NetScaler VPX インスタンスでも使用できます。ただし、EC2 インスタンスの種類を M3 に変更することはできません。その場合は、Citrix ADC ソフトウェアを 10.1.e-124 以降のリリースにアップグレードするには、まず標準の Citrix ADC アップグレード手順 () に従って、上記の手順を実行する必要があります。

AWS での Citrix ADC VPX インスタンスのスループットまたはソフトウェアエディションのアップグレード

ソフトウェアエディション（Standard エディションから Premium エディションへのアップグレードなど）またはスループット（たとえば、200 Mbps から 1000 Mbps へのアップグレードなど）をアップグレードするには、インスタンスのライセンスによって異なります。

カスタマーライセンスの使用 (自分のライセンスの持ち込み)

カスタマーライセンスを使用している場合は、Citrix Web サイトから新しいライセンスを購入してダウンロードし、VPX インスタンスにライセンスをインストールできます。Citrix Web サイトからライセンスをダウンロードおよびインストールする方法については、『VPX ライセンスガイド』を参照してください。

ユーティリティライセンスの使用（時間単位のユーティリティライセンス）

AWS では課金ベースのインスタンスの直接アップグレードがサポートされていません。料金ベースの Citrix ADC VPX インスタンスのソフトウェアエディションまたはスループットをアップグレードするには、必要なライセンスと容量を持つ新しい AMI を起動し、古いインスタンス構成を新しいインスタンスに移行します。これは、このページの「Citrix ADC 高可用性構成を使用した新しい Citrix ADC AMI インスタンスへのアップグレード」の説明に従って、Citrix ADC 高可用性構成を使用して実現できます 。

AWS での Citrix ADC VPX インスタンスのシステムソフトウェアのアップグレード

10.1.e-124.1308.e 以降のリリースを実行している VPX インスタンスをアップグレードする必要がある場合は、「Citrix ADC アプライアンスのアップグレードとダウングレード」の標準の Citrix ADC アップグレード手順に従ってください。

10.1.e-124.1308.e より古いリリースを実行している VPX インスタンスを 10.1.e-124.1308.e 以降のリリースにアップグレードする必要がある場合は、まずシステムソフトウェアをアップグレードしてから、インスタンスタイプを次のように M3 に変更します。

1. VPX インスタンスを停止します。
2. AWS コンソールで EC2 インスタンスの種類を変更します。
3. インスタンスを起動します。

Citrix ADC 高可用性構成を使用して、新しい **Citrix ADC AMI** インスタンスへのアップグレード

新しい Citrix ADC AMI インスタンスへのアップグレードの高可用性方式を使用するには、次のタスクを実行します。

- AWS Marketplace で、EC2 インスタンスの種類、ソフトウェアエディション、スループット、またはソフトウェアリリースを指定して新しいインスタンスを作成します。
- 古いインスタンス（アップグレード前）と新しいインスタンスとの間に高可用性を構成します。これにより、古いインスタンスの構成内容が新しいインスタンスに同期されます。
- 古いインスタンスから新しいインスタンスへの強制高可用性フェールオーバーを実行します。これにより、新しいインスタンスがプライマリノードとして設定され、新しいトラフィックを受信し始めます。
- 古いインスタンスを停止して、再構成するか AWS から削除します。

前提条件と考慮すべきポイント

- AWS 上の 2 つの Citrix ADC VPX インスタンス間で高可用性がどのように機能するかを理解してください。AWS 上の 2 つの Citrix ADC VPX インスタンス間の高可用性構成の詳細については、「AWS での高可用性ペアのデプロイ」を参照してください。
- 新しいインスタンスは、古いインスタンスと同じアベイラビリティゾーン内に作成し、同じセキュリティグループおよびサブネットが設定されている必要があります。
- 高可用性のセットアップでは、両インスタンスのユーザーの AWS IAM (Identity and Access Management) アカウントに関連付けられたアクセスキーと秘密キーが必要です。正しいキー情報を使用して VPX インスタンスを作成しないと、高可用性のセットアップに失敗します。VPX インスタンスの IAM アカウントの作成の詳細については、「前提条件」を参照してください。
  - 新しいインスタンスを作成するには EC2 コンソールを使用する必要があります。AWS の 1-Click 起動は使用できません。これは、アクセスが許可されず、秘密キーを入力できないためです。
  - 新しいインスタンスには ENI インターフェイスが 1 つだけ必要です。

高可用性構成を使用して Citrix ADC VPX インスタンスをアップグレードするには、次の手順に従います。

1. 古いインスタンスと新しいインスタンスの間で高可用性を構成します。2 つの Citrix ADC VPX インスタンス間で高可用性を構成するには、各インスタンスのコマンドプロンプトで次のように入力します。
   - `add ha node <nodeID> <IPaddress of the node to be added>`
   - `save config`

例:

古いインスタンスのコマンドプロンプトで、次のように入力します。

```
add ha node 30 192.0.2.30
Done
<!--NeedCopy-->
```

新しいインスタンスのコマンドプロンプトで、次のように入力します。

```
add ha node 10 192.0.2.10
Done
<!--NeedCopy-->
```

以下の点に注意してください:

- この高可用性セットアップで、古いインスタンスがプライマリノードで新しいインスタンスがセカンダリノードになります。
- NSIP アドレスは古いインスタンスから新しいインスタンスにコピーされません。このため、アップグレード完了時に新しいインスタンスには異なる管理 IP アドレスが設定されます。
- 新しいインスタンスの`nsroot`アカウントパスワードは、HA 同期後に古いインスタンスのアカウントパスワードに設定されます。

AWS 上の 2 つの Citrix ADC VPX インスタンス間の高可用性構成の詳細については、「AWS での高可用性ペアのデプロイ」を参照してください。

2. HA フェールオーバーを強制します。高可用性構成でフェイルオーバーを強制するには、いずれかのインスタンスのコマンドプロンプトで次のように入力します。

```
force HA failover
<!--NeedCopy-->
```

強制フェールオーバーにより、古いインスタンスの ENI が新しいインスタンスに移行され、トラフィックが新しいインスタンス（新しいプライマリノード）に流れます。また、古いインスタンス（新しいセカンダリノード）が再起動します。

次の警告メッセージが表示された場合は、N を入力して操作を中止します。

```
[WARNING]:Force Failover may cause configuration loss, peer health
    not optimum. Reason(s):
```



```
2 HA version mismatch
3 HA heartbeats not seen on some interfaces
4 Please confirm whether you want force-failover (Y/N)?
5 <!--NeedCopy-->
```

この警告メッセージは、2 つの VPX インスタンスのシステムソフトウェアで高可用性がサポートされていない場合に表示されます。このため、強制フェールオーバー時に古いインスタンスの構成情報が新しいインスタンスに同期されません。

この問題に対する回避策を次に示します。

a) 古いインスタンスの Citrix ADC シェルプロンプトで、次のコマンドを入力して、構成ファイル（ns.conf）のバックアップを作成します。

```
copy /nsconfig/ns.conf to /nsconfig/ns.conf.bkp
```

b) バックアップ構成ファイル（ns.conf.bkp）から次の行を削除します。

- `set ns config -IPAddress <IP> -netmask <MASK>`

例: `set ns config -IPAddress 192.0.2.10 -netmask 255.255.255.0`

c) 古いインスタンスのバックアップ構成ファイル (ns.conf.bkp) を新しいインスタンスの /nsconfig ディレクトリにコピーします。

d) 新しいインスタンスの Citrix ADC シェルプロンプトで、次のコマンドを入力して、古いインスタンスの構成ファイル（ns.conf.bkp）を新しいインスタンスにロードします。

- `batch -f /nsconfig/ns.conf.bkp`

e) 新しいインスタンスに設定を保存します。

- `save conifg`

f) いずれかのノードのコマンドプロンプトで、次のコマンドを入力してフェールオーバーを強制し、強制フェールオーバー操作を確認する警告メッセージに Y を入力します。

- `force ha failover`

例:

```
1          > force ha failover
2
3 WARNING]:Force Failover may cause configuration loss, peer health
     not optimum.
4          Reason(s):
5          HA version mismatch
6          HA heartbeats not seen on some interfaces
7          Please confirm whether you want force-failover (Y/N)? Y
```



```
8  <!--NeedCopy-->
```

3. HA 設定を削除して、2 つのインスタンスが HA 設定に含まれないようにします。これを行うには、まずセカンダリノードの高可用性構成を削除して、次にプライマリノードの高可用性を削除します。

   2 つの Citrix ADC VPX インスタンス間の HA 構成を削除するには、各インスタンスのコマンドプロンプトで次のように入力します。

```
1      > remove ha node <nodeID>
2      >  save config
3  <!--NeedCopy-->
```

   AWS 上の 2 つの VPX インスタンス間の高可用性設定の詳細については、「AWS に高可用性ペアをデプロイする」を参照してください。

   例:

古いインスタンス (新しいセカンダリノード) のコマンドプロンプトで、次のように入力します。

```
1          > remove ha node 30
2            Done
3          > save config
4            Done
5  <!--NeedCopy-->
```

新しいインスタンス（新しいプライマリノード）のコマンドプロンプトで、次のように入力します。

```
1          > remove ha node 10
2            Done
3          > save config
4            Done
5  <!--NeedCopy-->
```

AWS での VPX インスタンスのトラブルシューティング

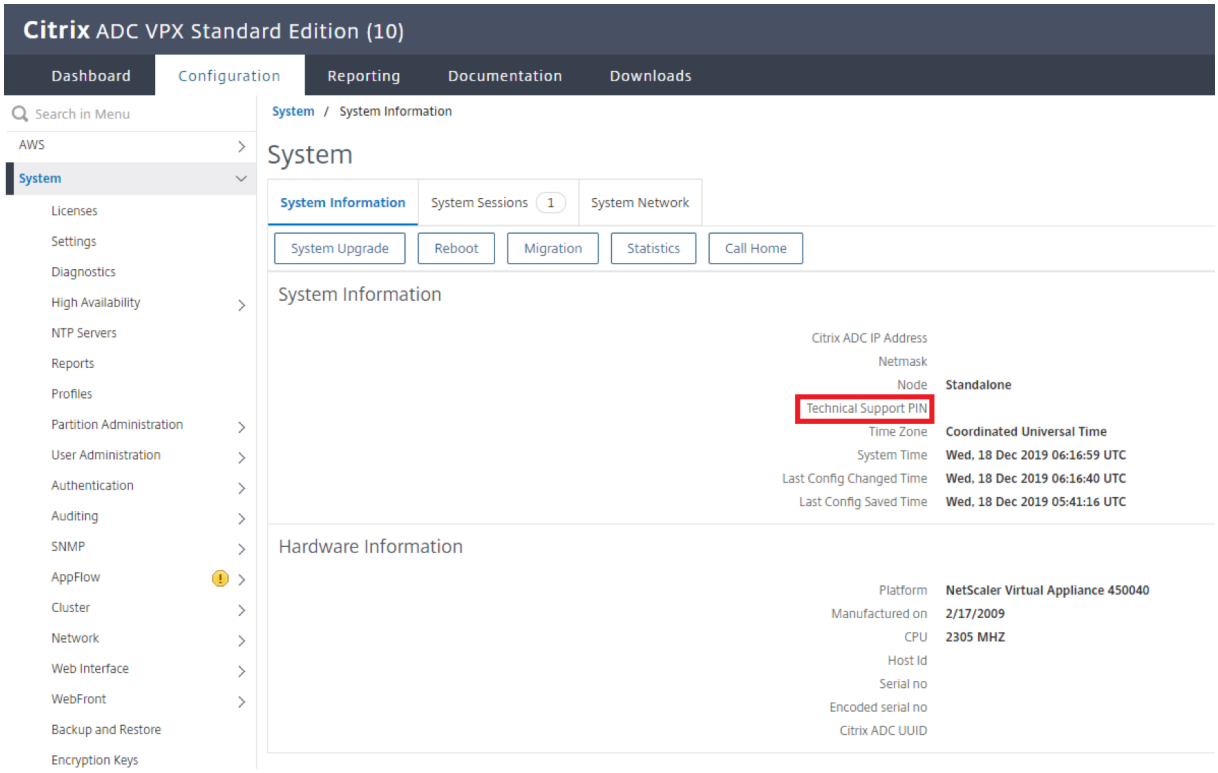
October 7, 2021

Amazon では、Citrix ADC VPX インスタンスへのコンソールアクセスは提供していません。トラブルシューティングを行うには、AWS GUI を使用してアクティビティログを表示する必要があります。デバッグできるのは、ネット

ワークが接続されている場合だけです。インスタンスのシステムログを表示するには、インスタンスを右クリックして [System Log] を選択します。

Citrix は、AWS で AWSMarketplace ライセンスの Citrix ADC VPX インスタンス（時間料金付きのユーティリティライセンス）のサポートを提供します。サポートケースを登録するには、AWS アカウント番号とサポート PIN コードを見つけ、Citrix サポートにお問い合わせください。また、お名前とメールアドレスを入力するよう求められます。サポート PIN を検索するには、VPX GUI にログオンし、［システム］ページに移動します。

サポート PIN を示すシステムページの例を次に示します。



AWS に関するよくある質問

February 21, 2022

- **Citrix ADC VPX** インスタンスは、**AWS** で暗号化されたボリュームをサポートしていますか**?**

  暗号化と復号化は Hypervisor レベルで行われるため、どのインスタンスでもシームレスに動作します。暗号化されたボリュームの詳細については、次の AWS ドキュメントを参照してください。

  https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html

- **AWS** で **Citrix ADC VPX** インスタンスをプロビジョニングする最良の方法は何ですか**?**

  次のいずれかの方法で、AWS で Citrix ADC VPX インスタンスをプロビジョニングできます。

- AWS マーケットプレイスでの AWS CloudFormation テンプレート（CFT）
- Citrix ADM
- AWS クイックスタート
- GitHub の Citrix AWS CFT
- GitHub の Citrix Terraform スクリプト
- GitHub の Citrix Ansible プレイブック
- AWS EC2 起動ワークフロー

使用するオートメーションツールに基づいて、一覧表示されたオプションのいずれかを選択できます。

オプションの詳細については、「AWS での Citrix ADC VPX」を参照してください。

- **AWS** で **Citrix ADC VPX** インスタンスをアップグレードするには**?**

  AWS で Citrix ADC VPX インスタンスをアップグレードするには、「AWS での Citrix ADC VPX インスタンスのアップグレード」の手順に従って、システムソフトウェアをアップグレードするか、新しい Citrix ADC VPX Amazon Machine Image（AMI）にアップグレードします。

  Citrix ADC VPX インスタンスをアップグレードする推奨される方法は、ジョブを使用した Citrix ADC インスタンスのアップグレードの手順に従って、ADM サービスを使用することです。

- **AWS** での **Citrix ADC VPX** の **HA** フェイルオーバー時間はどれくらいですか**?**

  - AWS アベイラビリティーゾーン内での Citrix ADC VPX の高可用性フェイルオーバーには約 3 秒かかります。
  - AWS アベイラビリティーゾーン全体の Citrix ADC VPX の HA フェイルオーバーには、約 5 秒かかります。

- テクニカルサポート **PIN** を提供する **Citrix ADC VPX** マーケットプレイスのサブスクリプションをご利用のお客様には、どのレベルのサポートが提供されますか**?**

  デフォルトでは、テクニカルサポート PIN を提供するお客様には「ソフトウェアの選択」サービスが提供されます。

- **Elastic IP** デプロイを使用した異なるゾーンでの高可用性では 、アプリケーションごとに複数の **IPSet** を作成する必要がありますか。

  はい。複数の VIP が複数の EIP にマッピングされた複数のアプリケーションがある場合は、複数の IPSet が必要です。したがって、HA フェールオーバー中に、EIP のすべてのプライマリ VIP マッピングがセカンダリ（新しいプライマリ）VIP に変更されます。

- 異なるゾーン展開で高可用性で **INC** モードが有効になるのはなぜですか。

  アベイラビリティーゾーン全体の HA ペアは、異なるネットワークにあります。HA 同期の場合、ネットワーク構成を同期してはいけません。これは、HA ペアで INC モードを有効にすることによって実現されます。

- アベイラビリティーゾーンが同じ **VPC** 内にある場合、あるアベイラビリティーゾーンの **HA** ノードは、別のアベイラビリティーゾーンのバックエンドサーバーと通信できますか。

はい。同じ VPC の異なるアベイラビリティーゾーンにあるサブネットには、SNIP 経由でバックエンドサーバーのサブネットを指す追加のルートを追加することで到達できます。たとえば、AZ1 の ADC の SNIP サブネットが 192.168.3.0/24、AZ2 のバックエンドサーバーのサブネットが 192.168.6.0/24 の場合、AZ1 に存在する Citrix ADC アプライアンスに 192.168.6.0 255.255.0 192.168.3.1 としてルートを追加する必要があります。

- **Elastic IP** を使用した異なるゾーン間での高可用性とプライベート **IP**デプロイを使用した異なるゾーン間での高可用性は一緒に機能しますか 。

  はい、両方の設定を同じ HA ペアに適用できます。

- プライベート **IP** デプロイを使用した異なるゾーン間の高可用性で 、**VPC** 内に複数のルートテーブルを持つ複数のサブネットがある場合、**HA** ペアのセカンダリノードは、**HA** フェールオーバー中にチェックされるルートテーブルについてどのように認識しますか。

  セカンダリノードはプライマリ NIC を認識し、VPC 内のすべてのルートテーブルを検索します。

- **AWS** で **VPX** のデフォルトイメージを使用する場合の**/var**パーティションのサイズはどれくらいですか? ディスク容量を増やすには?

  ディスクイメージを小さく保つために、ルートディスクのサイズは 20 GB に制限されています。

  /var/core/または/var/crash/ディレクトリ領域を増やす場合は、追加のディスクを接続します。/varサイズを大きくするには、現在のところ、重要なコンテンツを新しいディスクにコピーした後、追加のディスクを接続して/varにシンボリックリンクを作成する必要があります。

- **vCPU** にアクティブ化され、割り当てられるパケットエンジンは何台ありますか。

  パケットエンジン（PE）は、ライセンスされた vCPU の数によって制限されます。Citrix ADC デーモンは特定の vCPU に固定されず、非 PE vCPU で実行される可能性があります。AWS によると、C5.9xLarge は 72 GB のメモリを持つ 36vCPU インスタンスです。プールライセンスでは、Citrix ADC VPX インスタンスが最大数の PE でデプロイされます。この場合、19 PE がコア 1 ～19 で実行されます。ただし、ADC 管理プロセスは CPU 20～31 から実行されます。

- **ADC** の適切な **AWS** インスタンスを決定するには?

  1. スループット、PPS、SSL 要件、平均パケットサイズなどのユースケースと要件を理解します。
  2. VPX 帯域幅オファリングや vCPU ベースのライセンスなど、要件を満たす適切な ADC 製品とライセンスを選択します。
  3. 選択したオファリングに基づいて、AWS インスタンスを決定します。

  例**:**

  5 Gbps ライセンスでは、5 つのデータパケットエンジンが有効になります。したがって、vCPU 要件は 6（管理の場合は 5+1）です。ただし、6 つの vCPU インスタンスは利用できません。したがって、5 Gbps の帯域幅をサポートするネットワークを選択すれば、8 vCPU はそのスループットに到達するのに十分です。たとえば、5 Gbps のライセンスの最大 PE 割り当てを有効にするには、5 Gbps 帯域幅ライセンスに m5.2xlarge

を選択する必要があります。ただし、スループットによって制限されない vCPU ライセンスを使用すると、m5.xlarge インスタンス自体を使用して 5 Gbps のスループットが得られる可能性があります。

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
m5.large	2	8	EBS-Only	Up to 10	Up to 4,750
m5.xlarge	4	16	EBS-Only	Up to 10	Up to 4,750
m5.2xlarge	8	32	EBS-Only	Up to 10	Up to 4,750
m5.4xlarge	16	64	EBS-Only	Up to 10	4,750

- **AWS** の **ADC** には **3** つの **NIC 3** サブネットのデプロイメントが必須ですか?

  `Three NICs-three subnets`は、管理、クライアント、およびサーバーネットワーク用の推奨展開です。この展開により、トラフィックの分離と VPX パフォーマンスが向上します。2 つの NIC-2 サブネット、および 1 つの nic-One サブネットは、他の使用可能なオプションです。Citrix では、2 つの NIC（1 つのサブネットデプロイメント）など、複数の NIC が AWS でサブネットを共有することはお勧めしません。非対称ルーティングなどのネットワークの問題につながる可能性があるためです。詳細については、「AWS でネットワークインターフェイスを設定するためのベストプラクティス」を参照してください。

Microsoft Azure で Citrix ADC VPX インスタンスを展開する

January 25, 2022

Microsoft Azure Resource Manager（ARM）に Citrix ADC VPX インスタンスを展開する場合、次の両方の機能セットを使用してビジネスニーズを満たすことができます。

- Azure クラウドコンピューティング機能
- Citrix ADC 負荷分散とトラフィック管理機能

Citrix ADC VPX インスタンスは、スタンドアロンインスタンスとして、またはアクティブ/スタンバイモードの高可用性ペアとして ARM にデプロイできます。

Citrix ADC VPX インスタンスを Microsoft Azure にデプロイするには、次の 2 つの方法があります。

- Azure マーケットプレイスを通じて。Citrix ADC VPX 仮想アプライアンスは、Microsoft Azure Marketplace でイメージとして使用することができます。
- GitHub で利用可能な Citrix ADC Azure Resource Manager（ARM）json テンプレートを使用します。詳細については、Citrix NetScaler ソリューションテンプレートの GitHub リポジトリを参照してください。

Microsoft Azure スタックは、ローカルデータセンターに Microsoft Azure パブリッククラウドサービスを提供し、組織がハイブリッドクラウドを構築できるようにするハードウェアとソフトウェアの統合プラットフォームです。これで、Microsoft Azure スタックに Citrix ADC VPX インスタンスをデプロイできます。

前提要件

Azure に Citrix VPX インスタンスを展開する前に、前提条件となる知識が必要です。

- Azure の用語とネットワークの詳細に精通しています。詳細については、Azure の用語を参照してください。
- Citrix ADC アプライアンスの知識。Citrix ADC アプライアンスの詳細については、Citrix ADC を参照してください。
- Citrix ADC ネットワークに関する知識。ネットワーキング トピックを参照してください。

Azure での **Citrix ADC VPX** インスタンスのしくみ

オンプレミス展開では、Citrix ADC VPX インスタンスには少なくとも 3 つの IP アドレスが必要です。

- 管理 IP アドレス。NSIP アドレスと呼ばれます。
- サーバーファームとやり取りするためのサブネット IP（SNIP）アドレス
- クライアント要求を受け付ける仮想サーバー IP（VIP）アドレス

詳細については、「Microsoft Azure 上の Citrix ADC VPX インスタンスのネットワークアーキテクチャ」を参照してください。

> 注
>
> VPX 仮想アプライアンスは、2 つ以上のインテル VT-X コアと 2 GB を超えるメモリを持つ任意のインスタンスタイプにデプロイできます。システム要件の詳細については、Citrix ADC VPX のデータシートを参照してください。現在、Citrix ADC VPX インスタンスはインテルプロセッサーのみをサポートしています。

Azure デプロイメントでは、次の 3 つの方法で Azure で Citrix ADC VPX インスタンスをプロビジョニングできます。

- マルチ NIC マルチ IP アーキテクチャ
- Single NIC multi IP アーキテクチャ
- 単一の NIC シングル IP

ニーズに応じて、サポートされている次のアーキテクチャタイプを使用できます。

マルチ **NIC** マルチ **IP** アーキテクチャ

このデプロイタイプでは、VPX インスタンスに複数のネットワークインターフェイス (NIC) をアタッチできます。NIC には、静的または動的パブリック IP アドレスとプライベート IP アドレスを 1 つ以上割り当てることができます。

詳細については、次のユースケースを参照してください。

- 複数の IP アドレスと NIC を使用して高可用性セットアップを構成する
- PowerShell コマンドを使用して、複数の IP アドレスと NIC を使用して高可用性セットアップを構成する

> 注
>
> Azure 環境での MAC 移動やインターフェイスのミュートを回避するには、ADC VPX インスタンスのデータインターフェイス（タグなし）ごとに VLAN を作成し、Azure で NIC のプライマリ IP をバインドすることをお勧めします。詳細については、CTX224626 の記事を参照してください。

Single NIC multi IP アーキテクチャ

この展開タイプでは、複数の IP 構成（静的または動的パブリック IP アドレスおよびプライベート IP アドレスに割り当てられている）に関連付けられた 1 つのネットワークインターフェイス（NIC）。
詳細については、次のユースケースを参照してください。

- Citrix ADC VPX スタンドアロンインスタンスに複数の IP アドレスを構成する
- PowerShell コマンドを使用して、Citrix ADC VPX スタンドアロンインスタンスに複数の IP アドレスを構成する

単一の **NIC** シングル **IP**

この展開タイプでは、単一の IP アドレスに関連付けられた 1 つのネットワークインターフェイス（NIC）で、NSIP、SNIP、および VIP の機能を実行するために使用されます。

詳細については、次のユースケースを参照してください。

- Citrix ADC VPX スタンドアロンインスタンスを構成する

> 注
>
> 単一 IP モードは Azure 展開環境でのみ使用することができます。このモードは、オンプレミス、AWS、またはその他のタイプのデプロイの Citrix ADC VPX インスタンスでは使用できません。

Citrix ADC VPX ライセンス

Azure 上の Citrix ADC VPX インスタンスにはライセンスが必要です。Azure で実行されている Citrix ADC VPX インスタンスでは、次のライセンスオプションを使用できます。

- サブスクリプションベースのライセンス：Citrix ADC VPX アプライアンスは、Azure Marketplace で有料インスタンスとして使用できます。サブスクリプションベースライセンスは使った分を支払うオプションです。ユーザーは時間単位で課金されます。Azure Marketplace では、次の VPX モデルとライセンスタイプを使用できます。

VPX モデル	ライセンスの種類	推奨インスタンス
VPX10	スタンダード、アドバンス、プレミアム	Standard_D2s_v4

VPX モデル	ライセンスの種類	推奨インスタンス
VPX200	スタンダード、アドバンス、プレミアム	Standard_D2s_v4
VPX1000*	スタンダード、アドバンス、プレミアム	Standard_D4s_v4
VPX3000*	スタンダード、アドバンス、プレミアム	Standard_D4s_v4

> *: VPX 1000 および VPX 3000 モデルの場合、Citrix ADC VPX インスタンスでアクセラレーテッドネットワーキングを有効にして、目的のパフォーマンスを得る必要があります。アクセラレーションネットワークの構成について詳しくは、「Azure アクセラレーションネットワークを使用するように Citrix ADC VPX インスタンスを構成する」を参照してください。

Citrix では、サブスクリプションベースのライセンスインスタンスに関するテクニカルサポートを提供しています。サポートケースを申し出するには、「Azure での Citrix ADC サポート-時間単位のサブスクリプションライセンス」を参照してください。

- 自分のライセンスを持参 **(BYOL)**: 自分のライセンス (BYOL) を持ち込む場合は、http://support.citrix.com/article/CTX122426の VPX ライセンスガイドを参照してください。次の操作を実行する必要があります。
  - Citrix Web サイト内のライセンスポータルを使用して、有効なライセンスを生成します。
  - ライセンスをインスタンスにアップロードします。

VPX モデル	ライセンスの種類	推奨インスタンス
VPX10	スタンダード、アドバンス、プレミアム	Standard_D2s_v4
VPX200	スタンダード、アドバンス、プレミアム	Standard_D2s_v4
VPX1000*	スタンダード、アドバンス、プレミアム	Standard_D4s_v4
VPX3000*	スタンダード、アドバンス、プレミアム	Standard_D4s_v4
VPX5000*	スタンダード、アドバンス、プレミアム	Standard_D8s_v4
VPX8000*	スタンダード、アドバンス、プレミアム	Standard_D8s_v4

VPX モデル	ライセンスの種類	推奨インスタンス
VPX10000*	スタンダード、アドバンス、プレミアム	Standard_D16s_v4

> *: VPX 1000 から VPX 10000 モデルまで、Citrix ADC VPX インスタンスでアクセラレーテッドネットワーキングを有効にして、目的のパフォーマンスを得る必要があります。アクセラレーションネットワークの構成について詳しくは、「Azure アクセラレーションネットワークを使用するように Citrix ADC VPX インスタンスを構成する」を参照してください。

- **Citrix ADC VPX** チェックイン/チェックアウトライセンス：詳細については、「Citrix ADC VPX チェックイン/チェックアウトライセンス」を参照してください。

> 注
>
> Azure スタック環境では、**BYOL** が唯一のライセンスオプションです。

NetScaler リリース 12.0 56.20 以降、オンプレミスおよびクラウド展開用の VPX Express にはライセンスファイルは必要ありません。Citrix ADC VPX Express の詳細については、Citrix ADC ライセンスの概要の「Citrix ADC VPX Express ライセンス」セクションを参照してください。

> 注
>
> Azure Marketplace から購入したサブスクリプションベースの時間単位のライセンスに関係なく、まれに Azure にデプロイされた Citrix ADC VPX インスタンスに、デフォルトの Citrix ADC ライセンスが割り当てられます。これは、Azure インスタンスメタデータサービス (IMDS) の問題が原因で発生します。
>
> Citrix ADC VPX インスタンスの構成を変更する前にウォームリスタートを行い、正しい Citrix ADC VPX ライセンスを有効にします。

制限事項

ARM で Citrix ADC VPX 負荷分散ソリューションを実行すると、次の制限が課されます。

- Azure アーキテクチャでは、以下の NetScaler 機能をサポートしていません。
  - IPv6
  - Gratuitous ARP (GARP)
  - L2 モード
  - タグ付き VLAN
  - 動的ルーティング
  - 仮想 MAC
  - USIP
  - ジャンボフレーム
  - クラスタリング

> 注
>
> Citrix Application Delivery Management（ADM）Autoscale 機能（クラウド展開）では、ADC インスタンスはすべてのライセンスでクラスタリングをサポートします。詳細については、「Citrix ADM を使用した Microsoft Azure での Citrix ADC VPX の自動スケーリング」を参照してください。

- Citrix ADC VPX 仮想マシンをシャットダウンして一時的に割り当て解除しなければならないことが予想される場合は、仮想マシンの作成中に静的な内部 IP アドレスを割り当てます。静的内部 IP アドレスを割り当てないと、Azure が再起動のたびに異なる IP アドレスを仮想マシンに割り当てる可能性があり、仮想マシンにアクセスできなくなる場合があります。
- Azure 展開では、次の Citrix ADC VPX モデルのみがサポートされています: VPX 10、VPX 200、VPX 1000、および VPX 3000。詳細については、Citrix ADC VPX のデータシートを参照してください。

  モデル番号が VPX 3000 より大きい Citrix ADC VPX インスタンスを使用する場合、ネットワークスループットはインスタンスのライセンスで指定されたものとは異なることがあります。ただし、SSL スループットや SSL トランザクション/秒など、その他の機能が向上する可能性があります。
- 仮想マシンのプロビジョニング時に Azure によって生成される「展開 ID」は、ARM のユーザーには表示されません。展開 ID を使用して Citrix ADC VPX アプライアンスを ARM に展開することはできません。
- Citrix ADC VPX インスタンスは、初期化時に 20 MB/秒のスループットとスタンダードエディションの機能をサポートします。
- 高速ネットワーキングが有効になっている Azure 上の Citrix ADC VPX インスタンスは、パフォーマンスが向上します。Azure アクセラレーションネットワーキングは、リリース 13.0 ビルド 76.x 以降の Citrix ADC VPX インスタンスでサポートされています。ADC VPX で高速ネットワークを有効にするには、高速ネットワークをサポートする Azure インスタンスタイプを使用することをお勧めします。
- XenApp および XenDesktop 展開の場合、VPX インスタンス上の VPN 仮想サーバーは次のモードで構成できます。
  - 基本モード。`ICAOnly` VPN 仮想サーバーパラメーターが ON に設定されます。基本モードは、ライセンスされていない Citrix ADC VPX インスタンスで完全に機能します。
  - SmartAccess モード。`ICAOnly` VPN 仮想サーバーパラメーターが OFF に設定されています。SmartAccess ess モードは、ライセンスされていない Citrix ADC VPX インスタンス上の 5 人の Citrix ADC AAA セッションユーザーに対してのみ機能します。

> 注
>
> SmartControl 機能を構成するには、Citrix ADC VPX インスタンスにプレミアムライセンスを適用する必要があります。

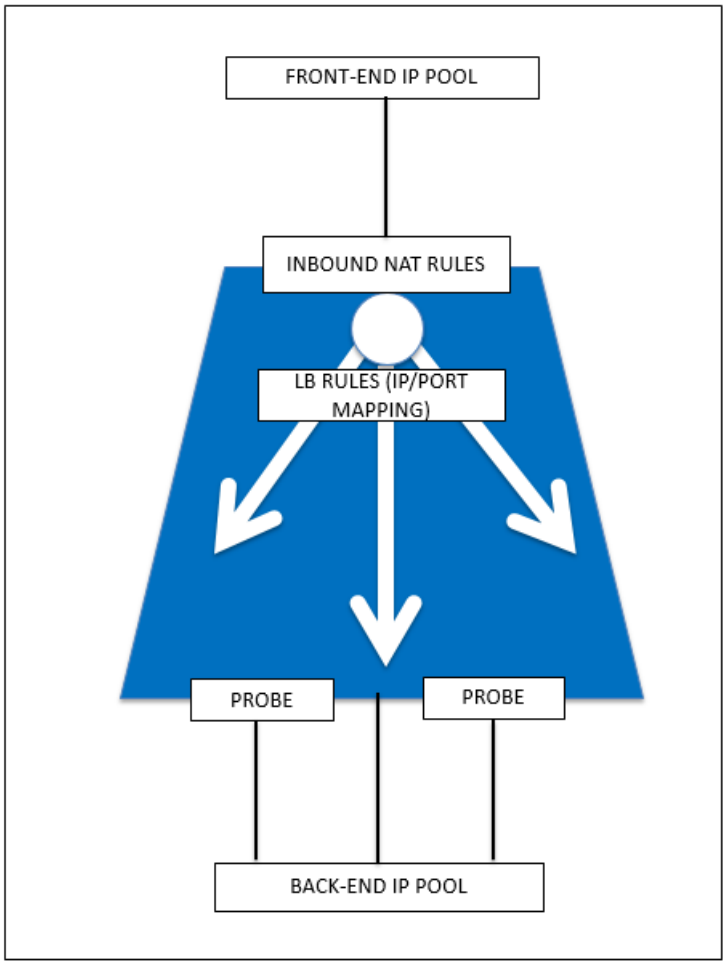
Azure 用語集

October 7, 2021

Citrix ADC VPX Azure のドキュメントで使用されている Azure 用語の一部を以下に示します。

1. Azure ロードバランサー — Azure ロードバランサーは、ネットワーク内のコンピューター間で着信トラフィックを分散するリソースです。トラフィックは、ロードバランサーセット内に定義された仮想マシンに分配されます。ロードバランサーには、外部ロードバランサー、インターネットに接続するロードバランサー、または内部ロードバランサーがあります。
2. Azure Resource Manager (ARM) — ARM は、Azure のサービスの新しい管理フレームワークです。Azure Load Balancer は、ARM ベースの API およびツールを使用して管理されます。
3. バックエンドアドレスプール — 負荷が分散される仮想マシンの NIC (NIC) に関連付けられた IP アドレスです。
4. BLOB-バイナリラージオブジェクト — Azure ストレージに格納できるファイルまたはイメージのようなバイナリオブジェクト。
5. フロントエンド IP 構成 — Azure ロードバランサーには、仮想 IP (VIP) とも呼ばれる 1 つ以上のフロントエンド IP アドレスを含めることができます。これらの IP アドレスがトラフィックの入口として使用されます。
6. インスタンスレベルのパブリック IP (ILPIP) — ILPIP は、仮想マシンまたはロールインスタンスが存在するクラウドサービスではなく、仮想マシンまたはロールインスタンスに直接割り当てることができるパブリック IP アドレスです。これは、クラウドサービスに割り当てられた VIP（仮想 IP）に代わるものではありません。これは、仮想マシンまたはロールインスタンスに直接接続するために使用できる追加の IP アドレスです。

   注：以前には、ILPIP は PIP（「パブリック IP」の略）と呼ばれていました。
7. インバウンド NAT ルール — ロードバランサーのパブリックポートを、バックエンドアドレスプール内の特定の仮想マシンのポートにマッピングするルールが含まれます。
8. IP-Config：個々の NIC に関連付けられた IP アドレスのペア（パブリック IP とプライベート IP）として定義できます。IP-Config では、パブリック IP アドレスが NULL の場合があります。各 NIC には、最大 255 までの IP 構成を関連付けることができます。
9. 負荷分散ルール：特定のフロントエンド IP とポートの組み合わせを、バックエンド IP アドレスとポートの組み合わせのセットにマップする規則プロパティ。ロードバランサーリソースの単一の定義を使用して複数のロードバランサー規則を定義でき、その各規則は、フロントエンド IP およびポートと、仮想マシンに関連付けられたバックエンド IP およびポートの組み合わせを示します。



10. ネットワークセキュリティグループ — 仮想ネットワーク内の仮想マシンインスタンスへのネットワークトラフィックを許可または拒否するアクセス制御リスト (ACL) ルールのリストが含まれます。NSG は、サブネット、またはそのサブネット内の個々の仮想マシンインスタンスに関連付けることができます。ネットワークセキュリティグループがサブネットに関連付けられている場合、ACL ルールはそのサブネット内のすべての仮想マシンインスタンスに適用されます。さらに、ネットワークセキュリティグループをその仮想マシンに直接関連付けることで、個々の仮想マシンへのトラフィックをさらに制限できます。

11. プライベート IP アドレス — Azure 仮想ネットワーク内の通信に使用され、VPN Gateway を使用してネットワークを Azure に拡張するときのオンプレミスネットワークで使用されます。プライベート IP アドレスを使用すると、Azure リソースは、VPN ゲートウェイまたは ExpressRoute 回路を経由して、インターネットで到達できる IP アドレスを使用せずに、仮想ネットワークまたはオンプレミスネットワーク内の他のリソースと通信できます。Azure Resource Manager 展開モデルでは、プライベート IP アドレスは次の種類の Azure リソースに関連付けられます - 仮想マシン、内部ロードバランサー（ILB)、およびアプリケーションゲートウェイ。

12. Probes — これには、バックエンドアドレスプール内の仮想マシンインスタンスの可用性をチェックするために使用されるヘルスプローブが含まれます。個別の仮想マシンが一定時間ヘルスプローブに応答しない場合、それはトラフィック供用から除外されます。プローブを使用すると、仮想インスタンスのヘルスを追跡できま

す。ヘルスプローブが失敗した場合、仮想インスタンスはローテーションから自動的に除外されます。

13. パブリック IP アドレス (PIP) — PIP は、Azure のパブリック向けサービスを含むインターネットとの通信に使用され、仮想マシン、インターネット向けロードバランサー、VPN ゲートウェイ、およびアプリケーションゲートウェイに関連付けられます。

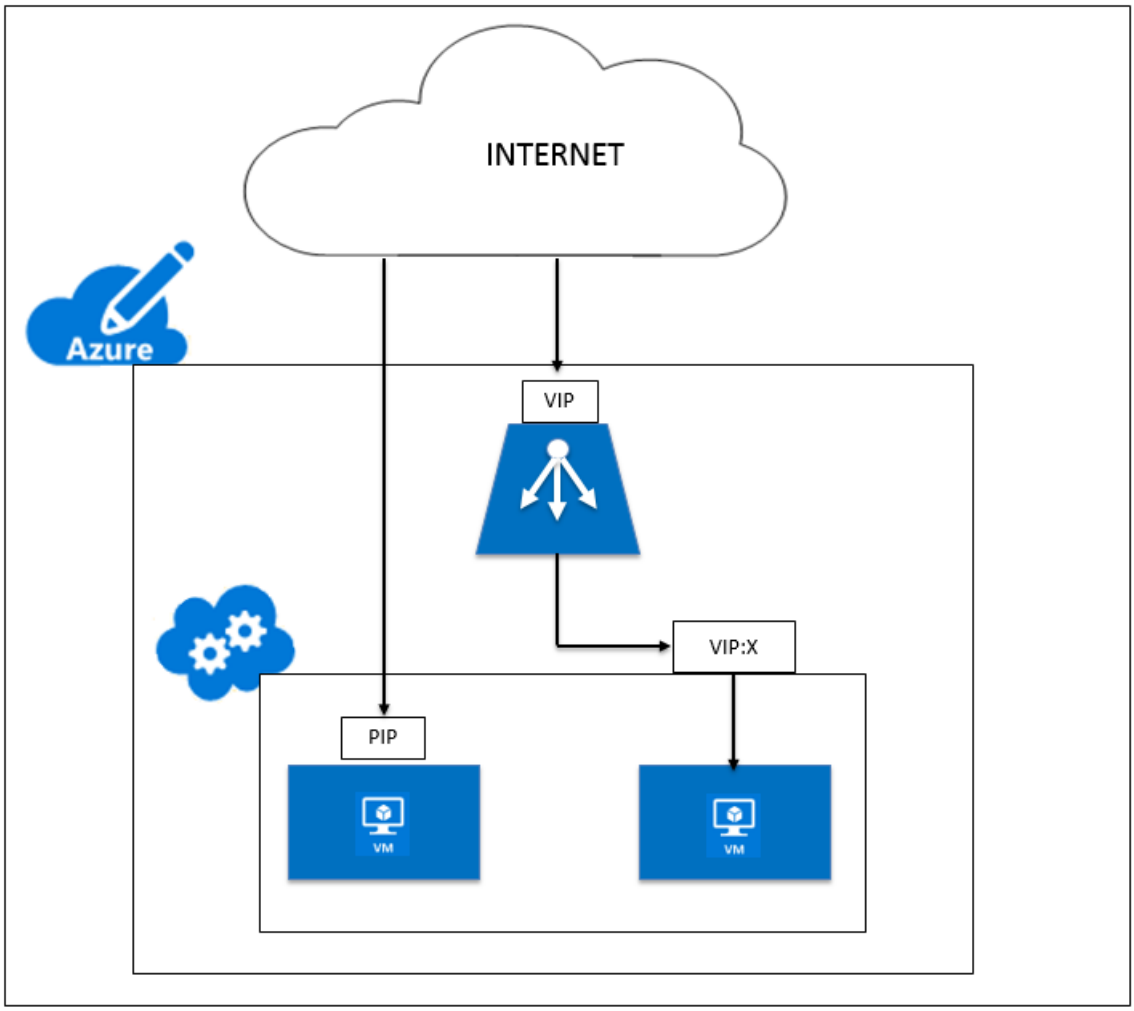
14. リージョン-国境を越えず、1 つ以上のデータセンターを含む地理内のエリア。価格設定、地域サービスおよびタイプは、リージョンレベルで公開されます。リージョンは通常、（最大で数百マイル離れた）別のリージョンと対にされ、リージョンペアを形成します。障害回復シナリオおよび高可用性シナリオでは、リージョンペアをメカニズムとして使用できます。また、一般に場所とも呼ばれます。

15. リソースグループ-リソースマネージャのコンテナは、アプリケーションに関連するリソースを保持します。リソースグループには、アプリケーションのリソースをすべて含めることも、論理的にグループにまとめられたリソースだけを含めることもできます。

16. ストレージアカウント — Azure ストレージアカウントを使用すると、Azure Storage の Azure BLOB、キュー、テーブル、およびファイルサービスにアクセスできます。ストレージアカウントは、Azure ストレージデータオブジェクトに一意の名前空間を提供します。

17. 仮想マシン — オペレーティングシステムを実行する物理コンピュータのソフトウェア実装。同じハードウェア上で複数の仮想マシンを同時に実行できます。Azure には、いろいろなサイズの仮想マシンが用意されています。

18. 仮想ネットワーク-Azure 仮想ネットワークは、クラウド内の独自のネットワークを表現したものです。それはサブスクリプション専用の Azure クラウドの論理的隔離です。IP アドレスブロック、DNS 設定、セキュリティポリシー、およびこのネットワーク内のルートテーブルを全面的に制御できます。さらに VNet のサブネットに分割したり、Azure IaaS 仮想マシンおよびクラウドサービス（PaaS ロールインスタンス）を起動したりすることもできます。また、Azure で利用できる接続性オプションの 1 つを使用して、仮想ネットワークをオンプレミスネットワークに接続できます。本質的には、Azure が提供するエンタープライズスケールの利点を持つ IP アドレスブロック上の全面的なコントロールを使用して、ネットワークを Azure に拡張できます。



Microsoft の Azure 上の Citrix ADC VPX インスタンスのネットワークアーキテクチャ

October 7, 2021

Azure Resource Manager（ARM）では、Citrix ADC VPX 仮想マシン（VM）が仮想ネットワークに存在します。仮想ネットワークの特定のサブネットに単一のネットワークインターフェイスを作成でき、VPX インスタンスに接続できます。ネットワークセキュリティグループを使用して、Azure 仮想ネットワーク内の VPX インスタンスとの間のネットワークトラフィックをフィルタリングできます。ネットワークセキュリティグループには、VPX インスタンスへのインバウンドネットワークトラフィックまたは VPX インスタンスからのアウトバウンドネットワークトラフィックを許可または拒否するセキュリティルールが含まれています。詳細については、「セキュリティグループ」を参照してください。

ネットワークセキュリティグループは、Citrix ADC VPX インスタンスへの要求をフィルタリングし、VPX インスタ

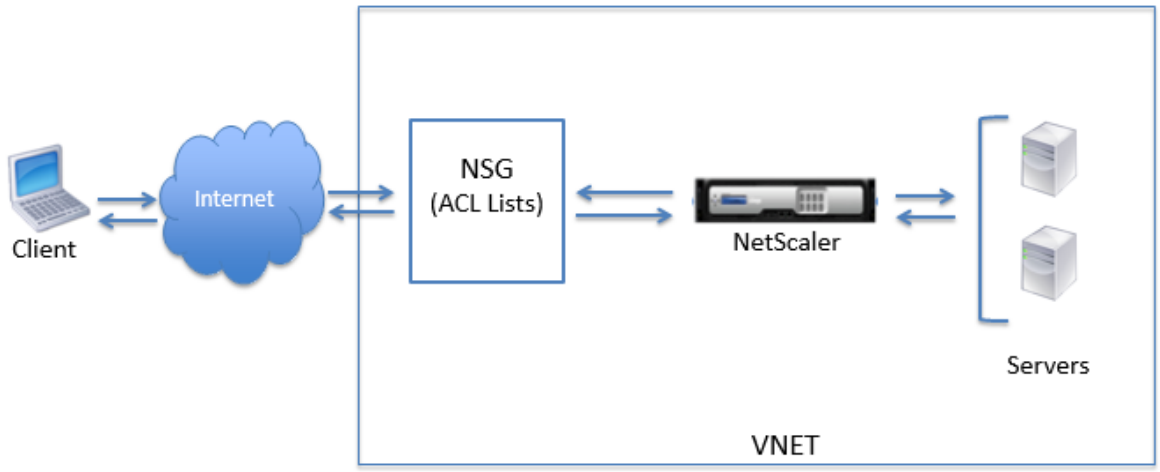
ンスはそれらをサーバーに送信します。サーバーからの応答は、逆の順序で同じパスをたどります。ネットワークセキュリティグループは、単一の VPX VM をフィルタリングするように構成することも、サブネットと仮想ネットワークを使用して、複数の VPX インスタンスを展開するトラフィックをフィルタリングすることもできます。

NIC には、ネットワーク構成の詳細（仮想ネットワーク、サブネット、内部 IP アドレス、パブリック IP アドレスなど）が含まれます。

ARM では、単一の NIC と 1 つの IP アドレスでデプロイされた仮想マシンにアクセスするために使用される、次の IP アドレスを知っておくとよいでしょう。

- パブリック IP（PIP）アドレスは、NetScaler VM の仮想 NIC 上で直接構成されるインターネット側 IP アドレスです。これにより、外部ネットワークから VM に直接アクセスできます。
- Citrix ADC IP（NSIP とも呼ばれる）アドレスは、仮想マシン上で構成された内部 IP アドレスです。これはルーティング不可能です。
- 仮想 IP アドレス（VIP）は、NSIP とポート番号を使用して構成されます。クライアントは PIP アドレスから NetScaler サービスにアクセスし、要求が NetScaler VPX VM または Azure ロードバランサーの NIC に到達すると、VIP が内部 IP（NSIP）および内部ポート番号に変換されます。
- 内部 IP アドレスは、仮想ネットワークのアドレス空間プールにある、VM のプライベート内部 IP アドレスです。この IP アドレスは、外部ネットワークから到達できません。この IP アドレスは、静的に設定しない限り、デフォルトで動的です。インターネットからのトラフィックは、ネットワークセキュリティグループで作成されたルールに従って、このアドレスにルーティングされます。ネットワークセキュリティグループは NIC と統合して、仮想マシンで設定されたサービスに応じて、適切なタイプのトラフィックを NIC の適切なポートに選択的に送信します。

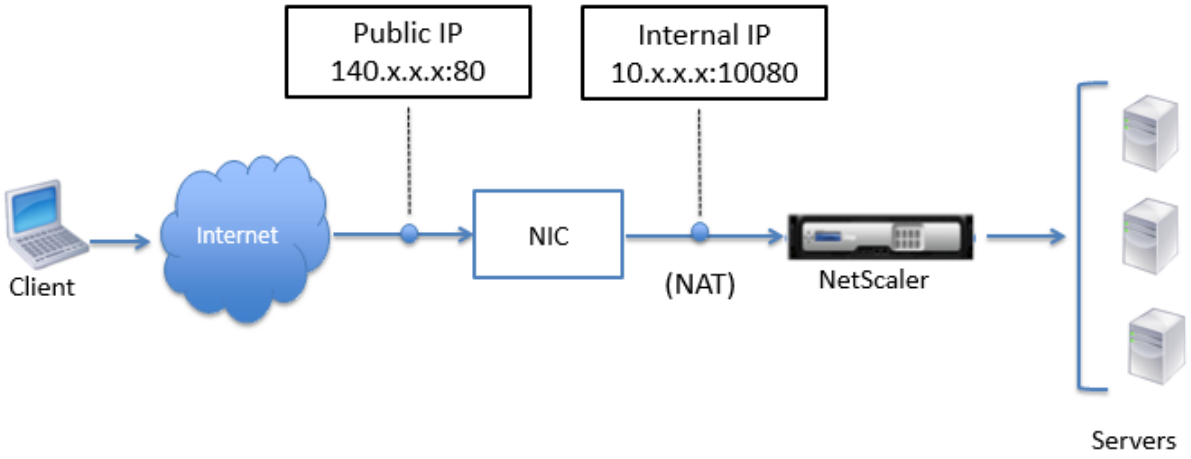
以下の図は、ARM でプロビジョニングされた NetScaler VPX インスタンスを介したクライアントからサーバーへのトラフィックフローを示しています。



ネットワークアドレス変換によるトラフィックフロー

Citrix ADC VPX インスタンス（インスタンスレベル）のパブリック IP（PIP）アドレスをリクエストすることもできます。この直接 PIP を VM レベルで使用する場合、ネットワークトラフィックを傍受する受信および送信規則を定義する必要はありません。インターネットからの着信要求が VM で直接受信されます。Azure はネットワークアドレス変換（NAT）を実行し、VPX インスタンスの内部 IP アドレスにトラフィックを転送します。

以下の図は、Azure がネットワークアドレス変換を実行し、NetScaler 内部 IP アドレスをマップする方法を示しています。



この例では、ネットワークセキュリティグループに割り当てられたパブリック IP は 140.x.x.x で、内部 IP アドレスは 10.x.x.x です。インバウンドルールとアウトバウンドルールが定義されている場合、パブリック HTTP ポート 80 はクライアントリクエストを受信するポートとして定義され、対応するプライベートポート 10080 は Citrix ADC VPX インスタンスがリッスンするポートとして定義されます。クライアント要求はパブリック IP アドレス 140.x.x.x で受信されます。Azure がネットワークアドレス変換を実行して、PIP を内部 IP アドレス 10.x.x.x（ポート 10080）にマップし、クライアント要求を転送します。

> 注
>
> 高可用性の Citrix ADC VPX VM は、負荷分散トラフィックを制御するためのインバウンドルールが定義されている外部または内部のロードバランサーによって制御されます。外部トラフィックは最初にこれらのロードバランサによって代行受信され、トラフィックは設定されたロードバランシング規則に従って迂回されます。ロードバランサには、バックエンドプール、NAT ルール、および健全性プローブが定義されています。

ポートの使用に関する注意事項

Citrix ADC VPX インスタンスの作成中または仮想マシンのプロビジョニング後に、ネットワークセキュリティグループでより多くのインバウンドルールとアウトバウンドルールを構成できます。各受信および送信規則は、パブリックポートおよびプライベートポートに関連付けられています。

ネットワークセキュリティグループルールを設定する前に、使用できるポート番号に関する次のガイドラインに注意

してください。

1. Citrix ADC VPX インスタンスは、以下のポートを予約します。インターネットからの要求にパブリック IP アドレスを使用する場合、これらをプライベートポートとして定義することはできません。

   ポート 21, 22, 80, 443, 8080, 67, 161, 179, 500, 520, 3003, 3008, 3009, 3010, 3011, 4001, 5061, 9000, 7000.

   ただし、VIP などのインターネットに直接接続するサービスで標準ポート（ポート 443 など）を使用する場合は、ネットワークセキュリティグループを使用してポートマッピングを作成する必要があります。これにより、標準ポートがこの VIP サービス用に NetScaler で構成された別のポートにマップされます。

   たとえば、VIP サービスが VPX インスタンスのポート 8443 で実行されているが、パブリックポート 443 にマッピングされているとします。したがって、ユーザーがパブリック IP を介してポート 443 にアクセスすると、要求はプライベートポート 8443 に送信されます。

2. パブリック IP アドレスでは、ポートマッピングが動的に解放される、パッシブ FTP や ALG のようなプロトコルをサポートしていません。

3. 高可用性は、Azure ロードバランサーで構成された PIP ではなく、VPX インスタンスに関連付けられたパブリック IP アドレス（PIP）を使用するトラフィックでは機能しません。

> 注
>
> Azure Resource Manager では、Citrix ADC VPX インスタンスは、パブリック IP アドレス（PIP）と内部 IP アドレスの 2 つの IP アドレスに関連付けられます。外部トラフィックは PIP に接続しますが、内部 IP アドレスまたは NSIP はルーティング不可能です。VPX で VIP を設定するには、内部 IP アドレスと使用可能な空きポートのいずれかを使用します。VIP の構成に PIP を使用してはいけません。

Citrix ADC VPX スタンドアロンインスタンスの構成

October 7, 2021

仮想マシンを作成し、他のリソースを構成することで、Azure Resource Manager（ARM）ポータルで単一の Citrix ADC VPX インスタンスをスタンドアロンモードでプロビジョニングできます。

はじめに

次の項目があることを確認します。

- Microsoft Azure ユーザーアカウント
- Microsoft Azure Resource Manager へのアクセス
- Microsoft Azure SDK
- Microsoft のアズ PowerShell

Microsoft Azure ポータルのページで 、ユーザー名とパスワードを指定して Azure Resource Manager ポータルにログオンします。

> 注
>
> ARM ポータルで、1 つのペインでオプションをクリックすると、右側に新しいペインが開きます。ペイン間を移動してデバイスを構成します。

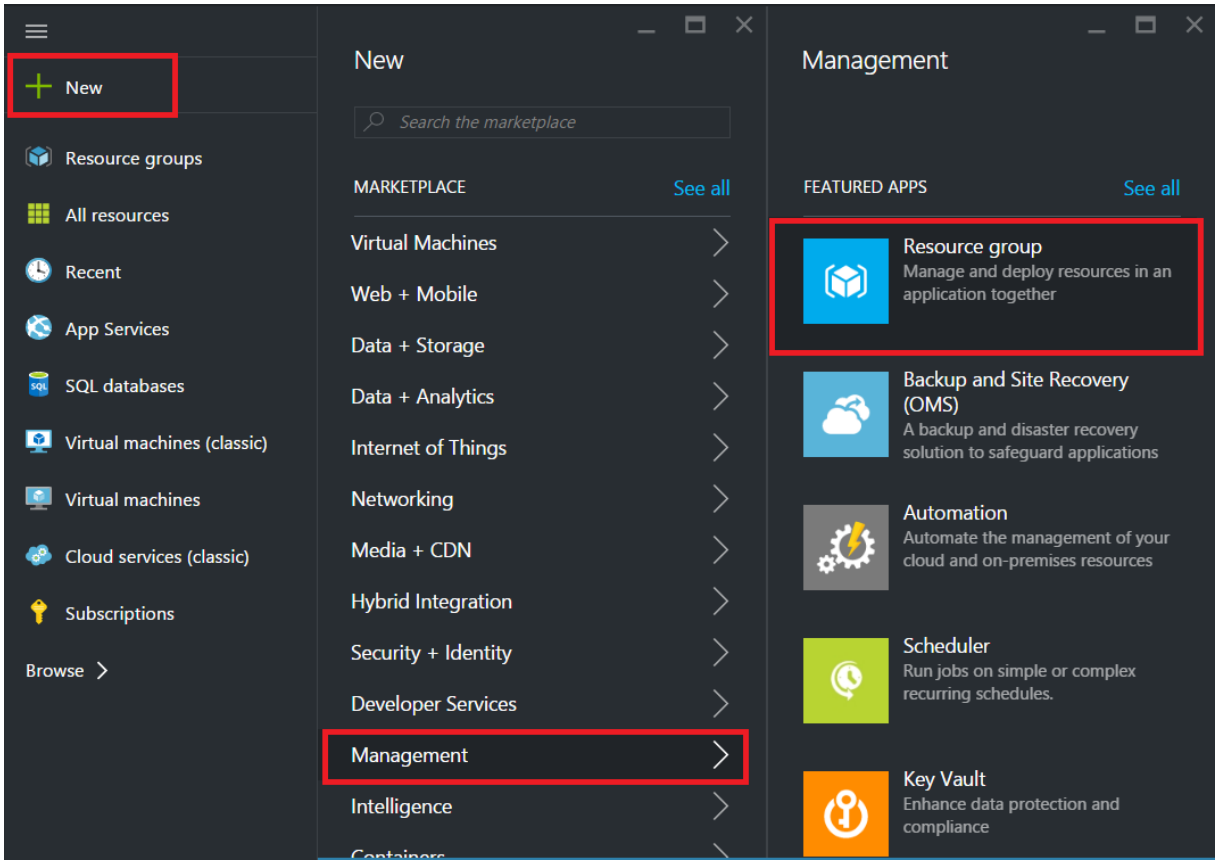
設定手順の要約

1. リソースグループの構成
2. ネットワークセキュリティグループの構成
3. 仮想ネットワークインターフェイスとそのサブネットの構成
4. ストレージアカウントの構成
5. 可用性セットの構成
6. Citrix ADC VPX インスタンスを構成します。

リソースグループの構成

すべてのリソースのコンテナとなる新しいリソースグループを作成します。リソースグループを使用して、リソースをグループとして展開、管理、および監視します。

1. [新規作成] > [管理] > [リソースグループ] をクリックします。
2. [リソースグループ] ペインで、次の詳細を入力します。
   - リソースグループ名
   - リソースグループの場所
3. ［作成］をクリックします。

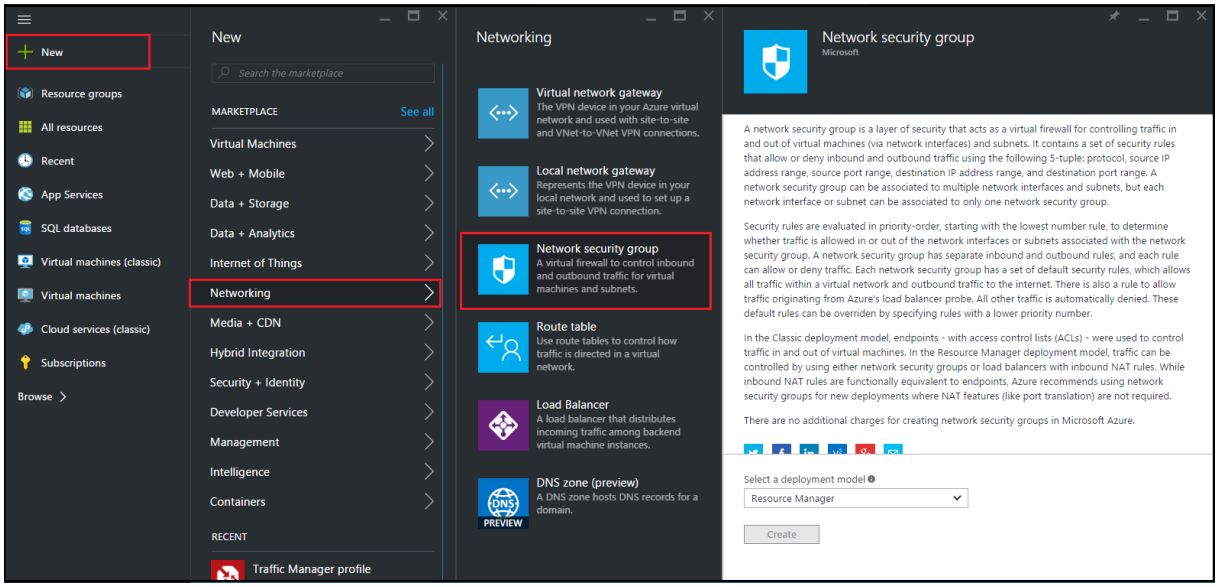


ネットワークセキュリティグループの構成

仮想ネットワーク内の着信トラフィックと発信トラフィックを制御するインバウンドルールとアウトバウンドルールを割り当てるネットワークセキュリティグループを作成します。ネットワークセキュリティグループを使用すると、単一の仮想マシンのセキュリティルールを定義したり、仮想ネットワークサブネットのセキュリティルールを定義したりできます。

1. [新規] > [ネットワーク] > [ネットワークセキュリティグループ] をクリックします。
2. [ネットワークセキュリティグループの作成] ウィンドウで、次の詳細を入力し、[作成] をクリックします。
   - Name - セキュリティグループの名前を入力します
   - Resource group - ボックスの一覧からリソースグループを選択します

> 注
>
> 正しい場所を選択していることを確認します。場所が異なれば、ボックスの一覧に表示されるリソースの一覧も異なります。

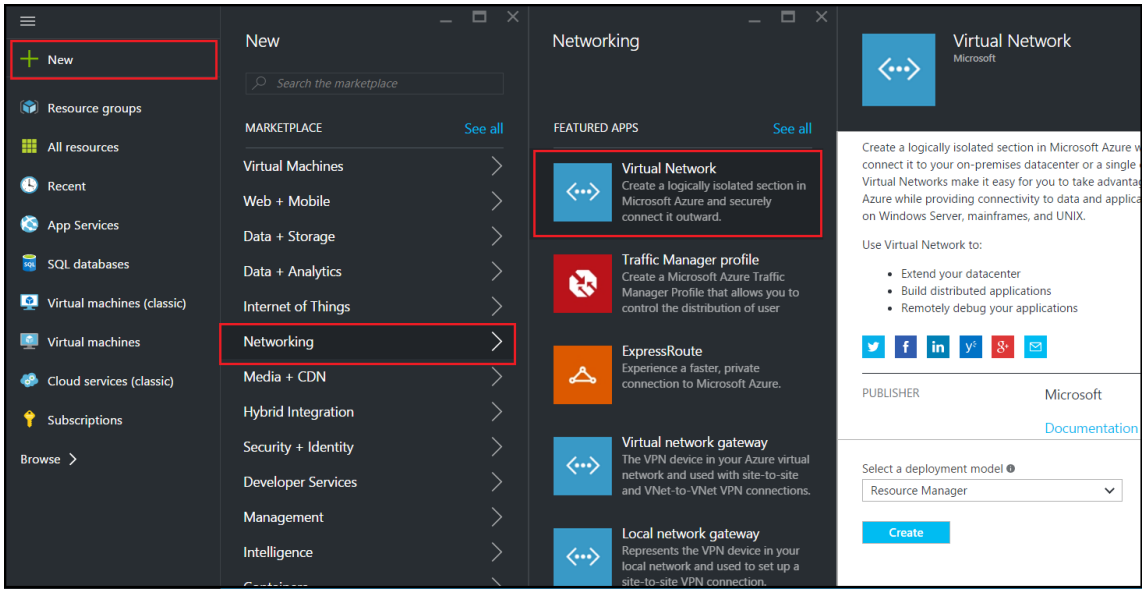


仮想ネットワークとサブネットを構成する

ARM の仮想ネットワークは、サービスのセキュリティを強化し、隔離するものです。同じ仮想ネットワークに属する VM およびサービスは、互いにアクセスできます。

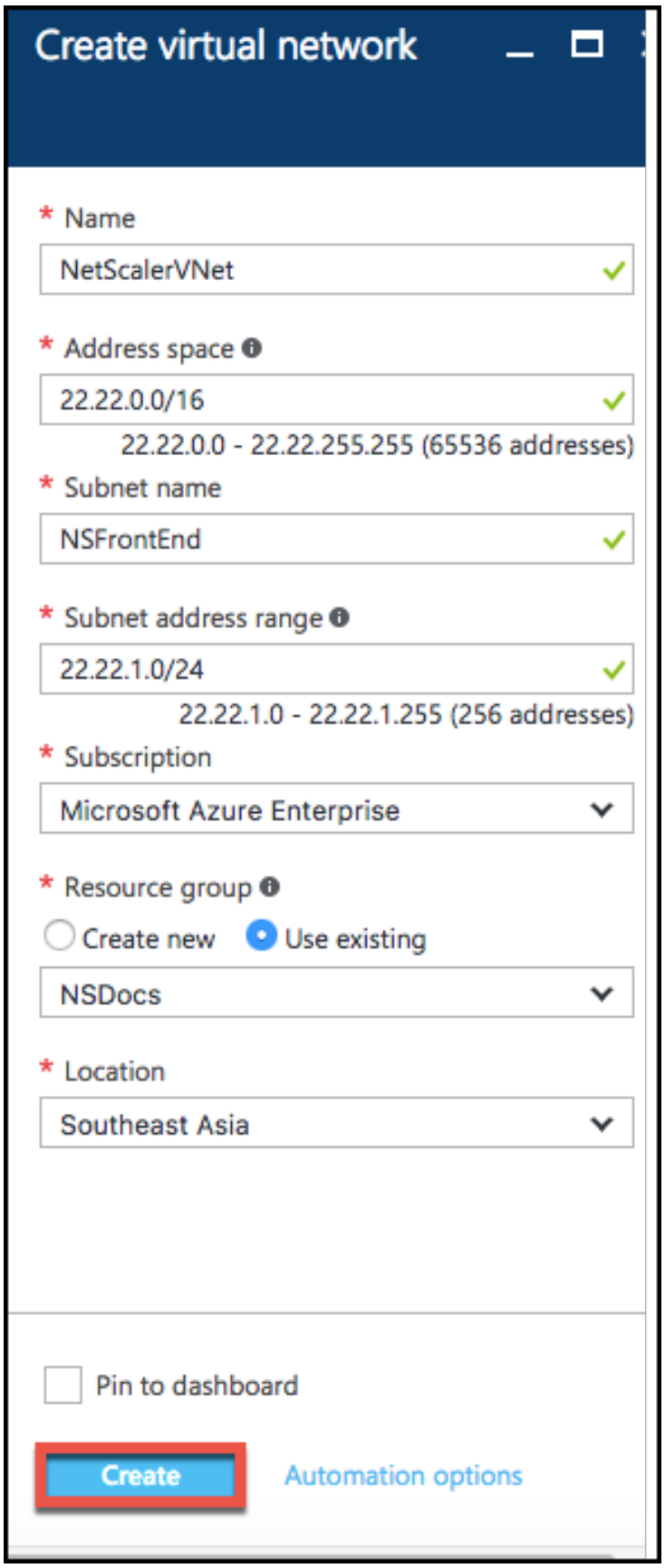
仮想ネットワークとサブネットを作成する手順は次のとおりです。

1. [新規] > [ネットワーク] > [仮想ネットワーク] をクリックします。
2. [仮想ネットワーク] ペインで、展開モードが [リソースマネージャ] であることを確認し、[作成] をクリックします。



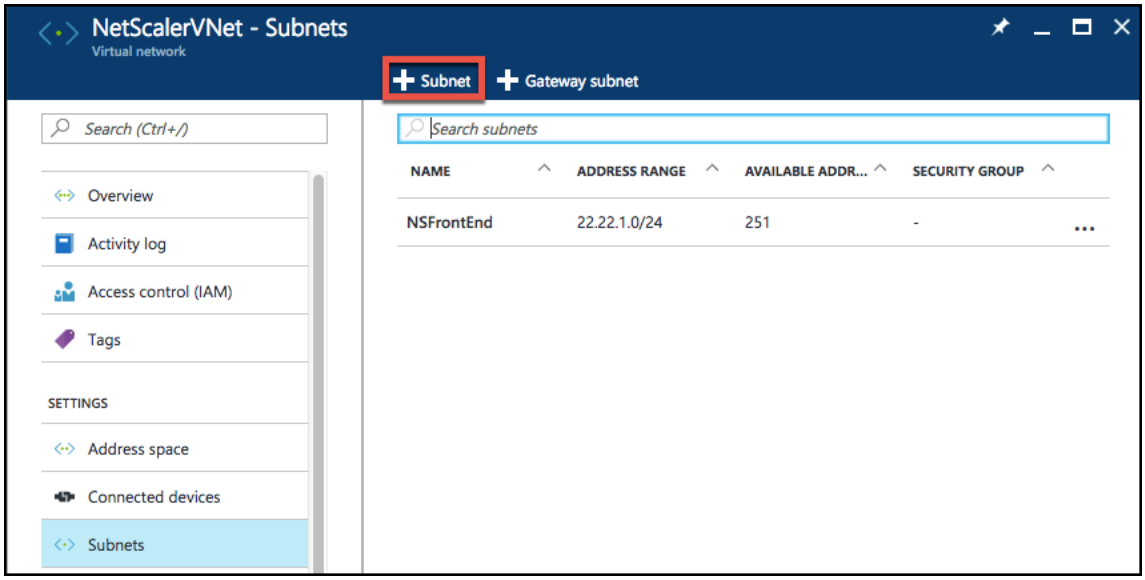
3. 仮想ネットワークの作成ウィンドウで、次の値を入力し、作成をクリックします。

- 仮想ネットワークの名前
- Address space - 仮想ネットワークの予約済 IP アドレスブロックを入力します
- Subnet-最初のサブネットの名前を入力します (この手順の後半で 2 番目のサブネットを作成します)。
- Subnet address range - サブネットの予約済 IP アドレスブロックを入力します
- Resource group - ボックスの一覧から以前に作成したリソースグループを選択します。



2 番目のサブネットを設定する

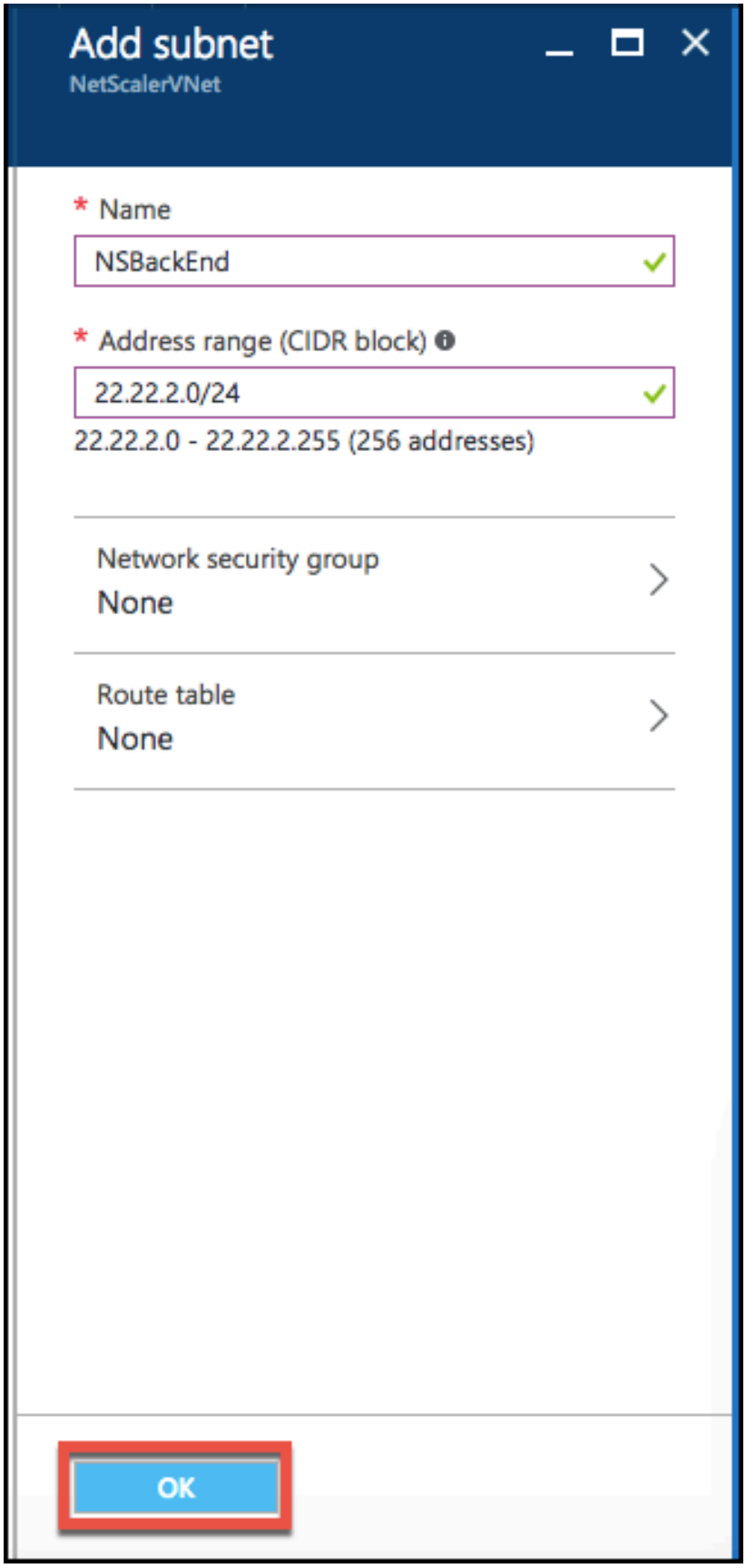
1. [すべてのリソース] ペインから新しく作成した仮想ネットワークを選択し、[設定] ペインで [サブネット] をクリックします。



2. **+Subnet** をクリックし、次の詳細を入力して 2 番目のサブネットを作成します。

   - 2 番目のサブネットの名前
   - Address range - サブネットの予約済 IP アドレスブロックを入力します
   - ネットワークセキュリティグループ-ドロップダウンリストからネットワークセキュリティグループを選択します。

3. ［作成］をクリックします。

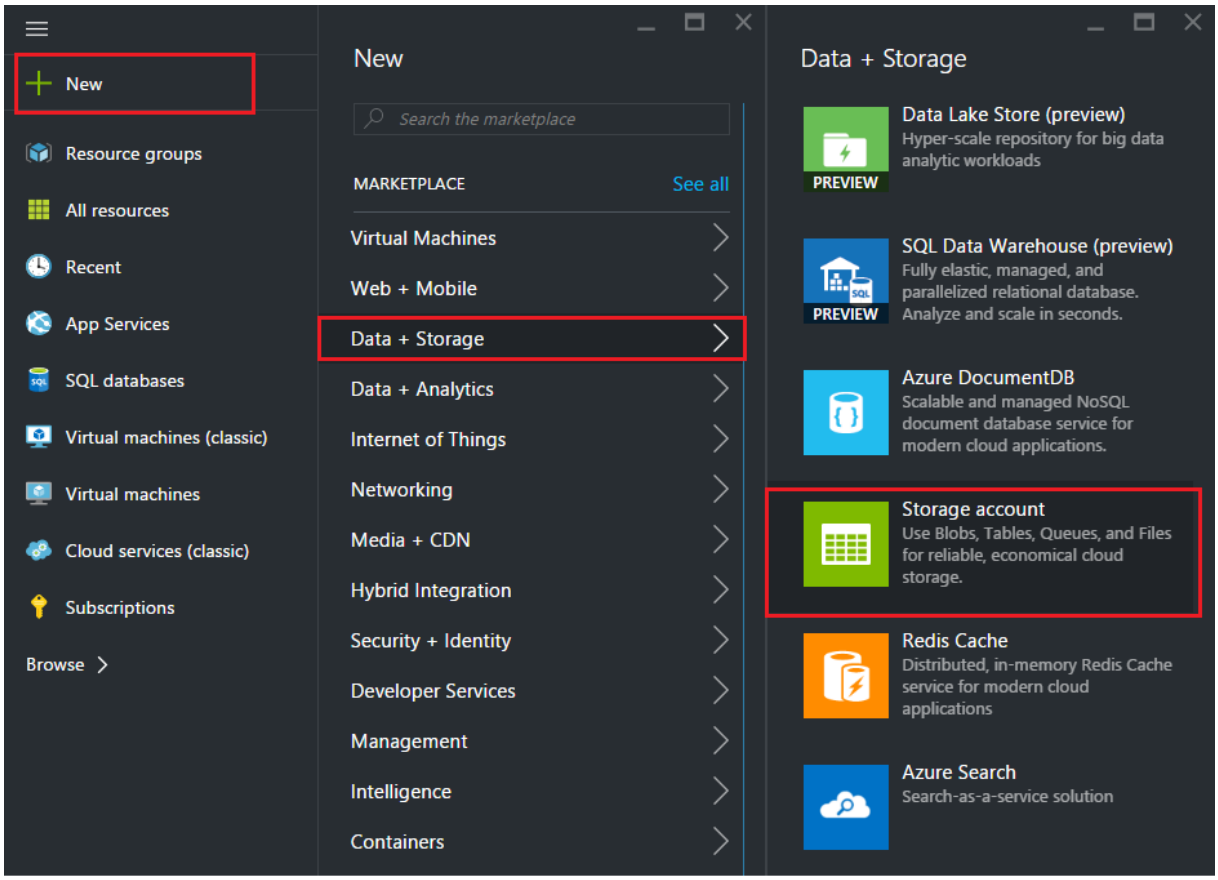


ストレージアカウントの構成

ARM IaaS インフラストラクチャストレージには、BLOB、表、キューおよびファイルの形式でデータを保存できるすべてのサービスが含まれます。ARM では、これらの形式のストレージデータを使用してアプリケーションを作成することもできます。

ストレージアカウントを作成してすべてのデータを保存します。

1. [**+ 新規**] > [データ + ストレージ] > [ストレージアカウント **] をクリックします。
2. [ストレージアカウントの作成] ウィンドウで、次の詳細を入力します。
   - アカウントの名前
   - デプロイメントモード-必ず リソースマネージャを選択してください
   - Account kind-ドロップダウンリストから [汎用] を選択します。
   - Replication-ドロップダウン・リストから [ローカル冗長ストレージ] を選択します。
   - Resource group - ボックスの一覧から新しく作成したリソースグループを選択します
3. [作成] をクリックします。

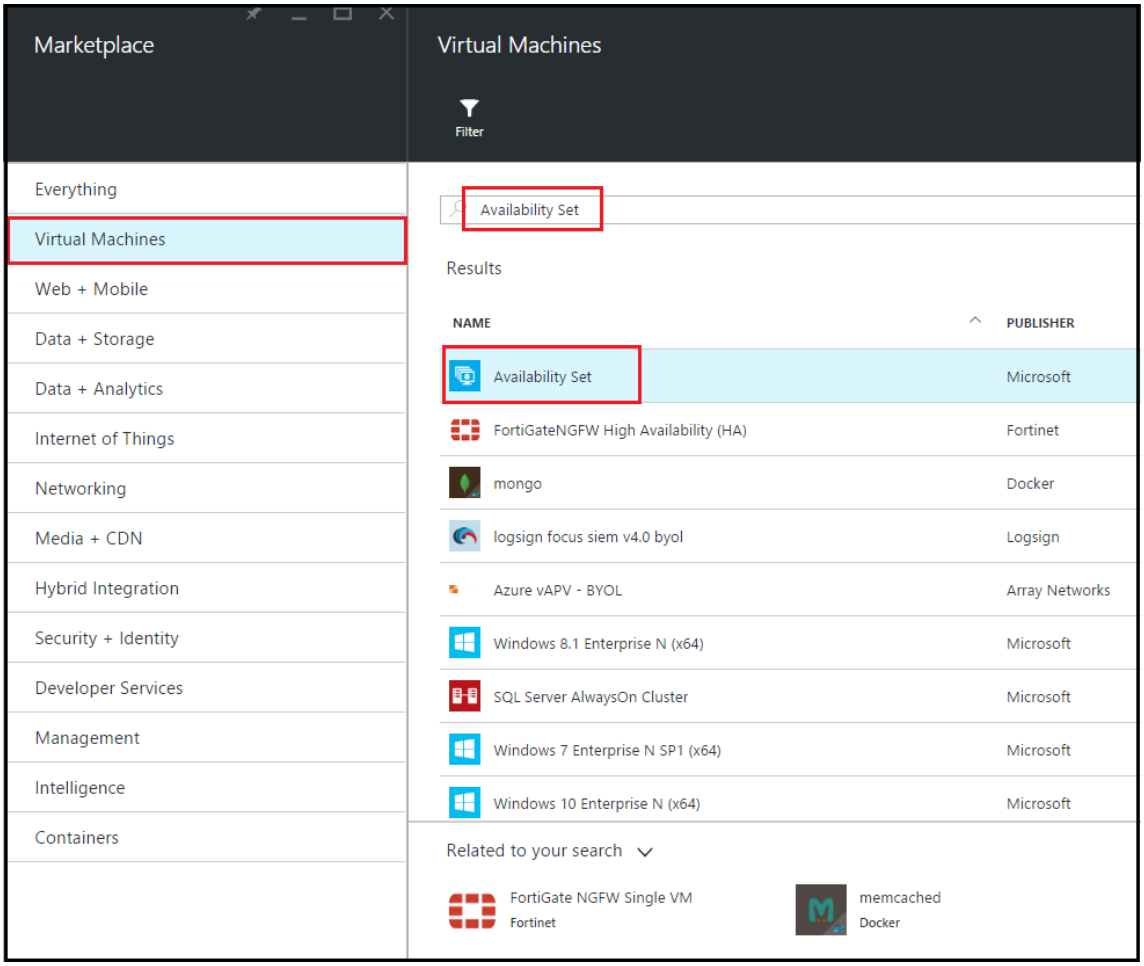


可用性セットの構成

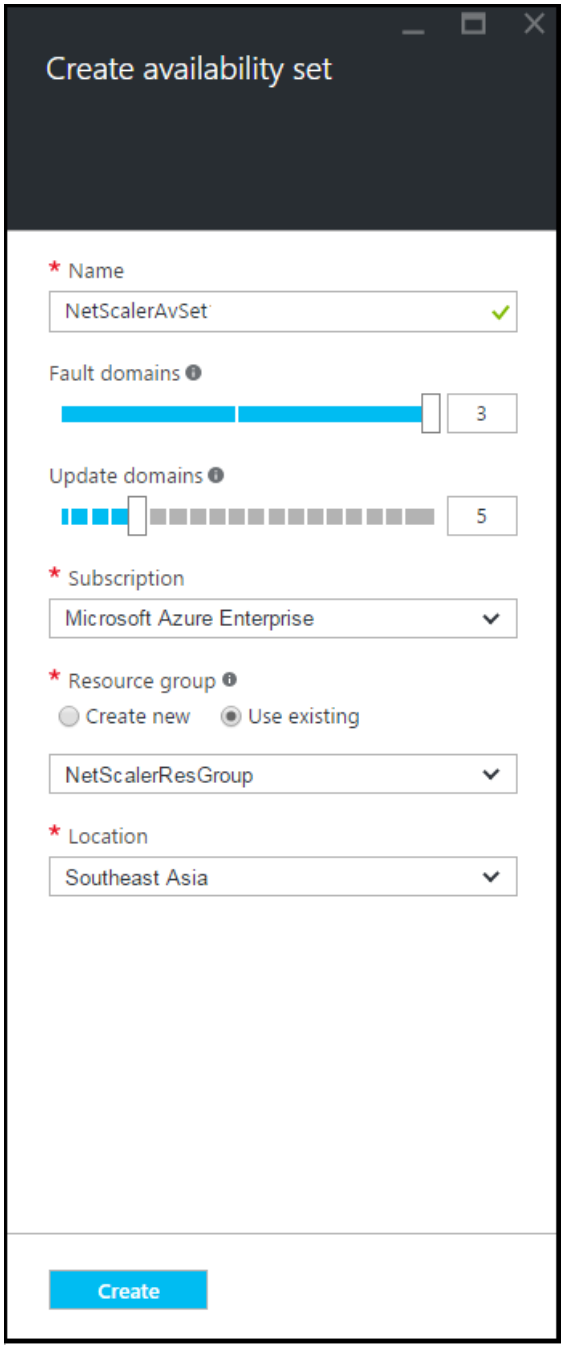
可用性セットは、計画的または計画外のメンテナンスの場合に、少なくとも 1 つの VM が稼働し続けることを保証します。同じ可用性セットに属する 2 台以上の VM は、異なるフォールトドメインに配置されて、サービスの冗長性を

確保します。

1. [+ 新規作成] をクリックします。
2. [MARKETPLACE] ペインで [すべて表示] をクリックし、[仮想マシン] をクリックします。
3. 可用性セットを検索し、表示されたリストから [可用性セットエンティティ] を選択します。



4. [作成] をクリックし、 [可用性セットの作成] ウィンドウで、次の詳細を入力します。
   - セットの名前
   - Resource group - ボックスの一覧から新しく作成したリソースグループを選択します
5. [作成] をクリックします。



Citrix ADC VPX インスタンスの構成

仮想ネットワーク内に Citrix ADC VPX インスタンスを作成します。Azure Marketplace から Citrix ADC VPX イメージを取得し、Azure Resource Manager ポータルを使用して Citrix ADC VPX インスタンスを作成します。

Citrix ADC VPX インスタンスの作成を開始する前に、インスタンスが存在する必要なサブネットを持つ仮想ネットワークが作成されていることを確認してください。仮想マシンのプロビジョニング時に仮想ネットワークを作成することもできますが、柔軟性に欠けるため別のサブネットを作成することはできません。仮想ネットワークの作成につ

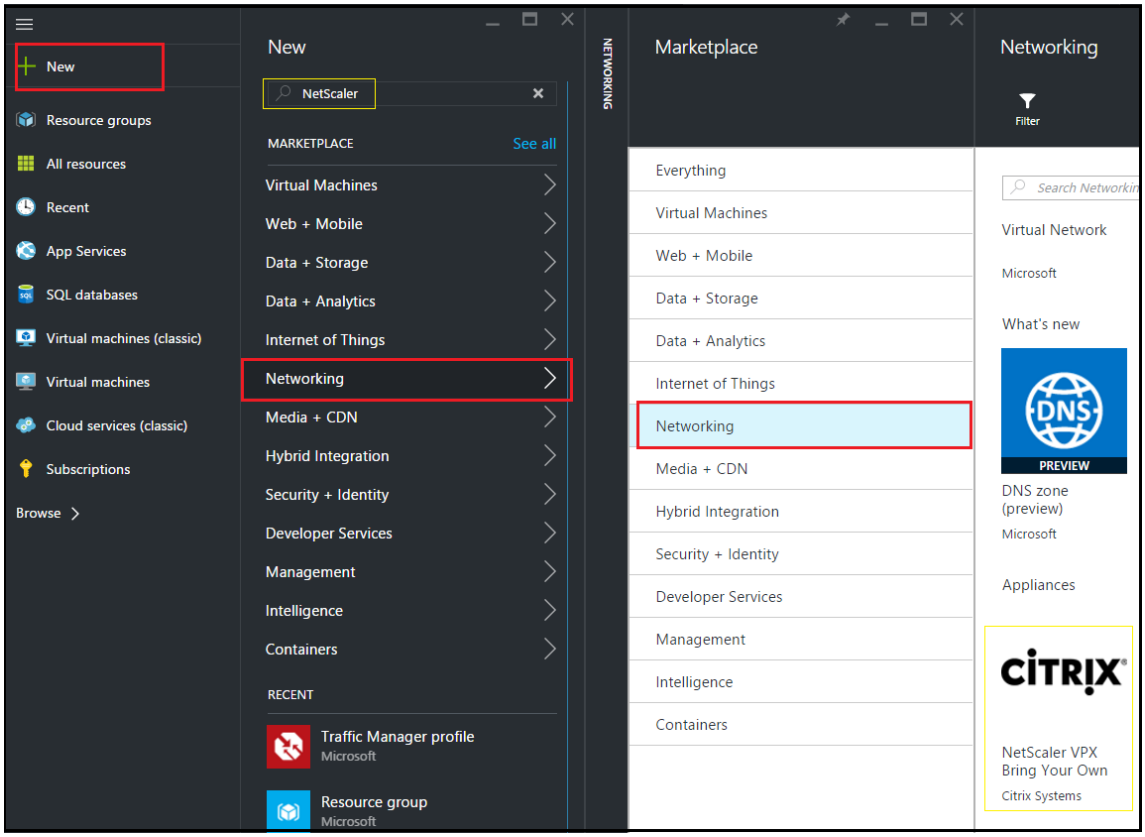
いては、http://azure.microsoft.com/en-us/documentation/articles/create-virtual-network/を参照してください。

必要に応じて、仮想マシンがインターネットリソースにアクセスできるようにする DNS サーバーと VPN 接続を構成します。

> 注
>
> Citrix ADC VPX VM をプロビジョニングする前に、リソースグループ、ネットワークセキュリティグループ、仮想ネットワークなどのエンティティを作成して、Provisioning 中にネットワーク情報を使用できるようにすることをお勧めします。

1. [+ 新規] > [ネットワーク] をクリックします。
2. ［すべて表示］をクリックし、［ネットワーク］ペインで ［**Citrix ADC 13.0**］をクリックします。
3. ソフトウェアプランのリストから［**Citrix ADC 13.0 VPX** 独自のライセンスを持ち込む］を選択します。

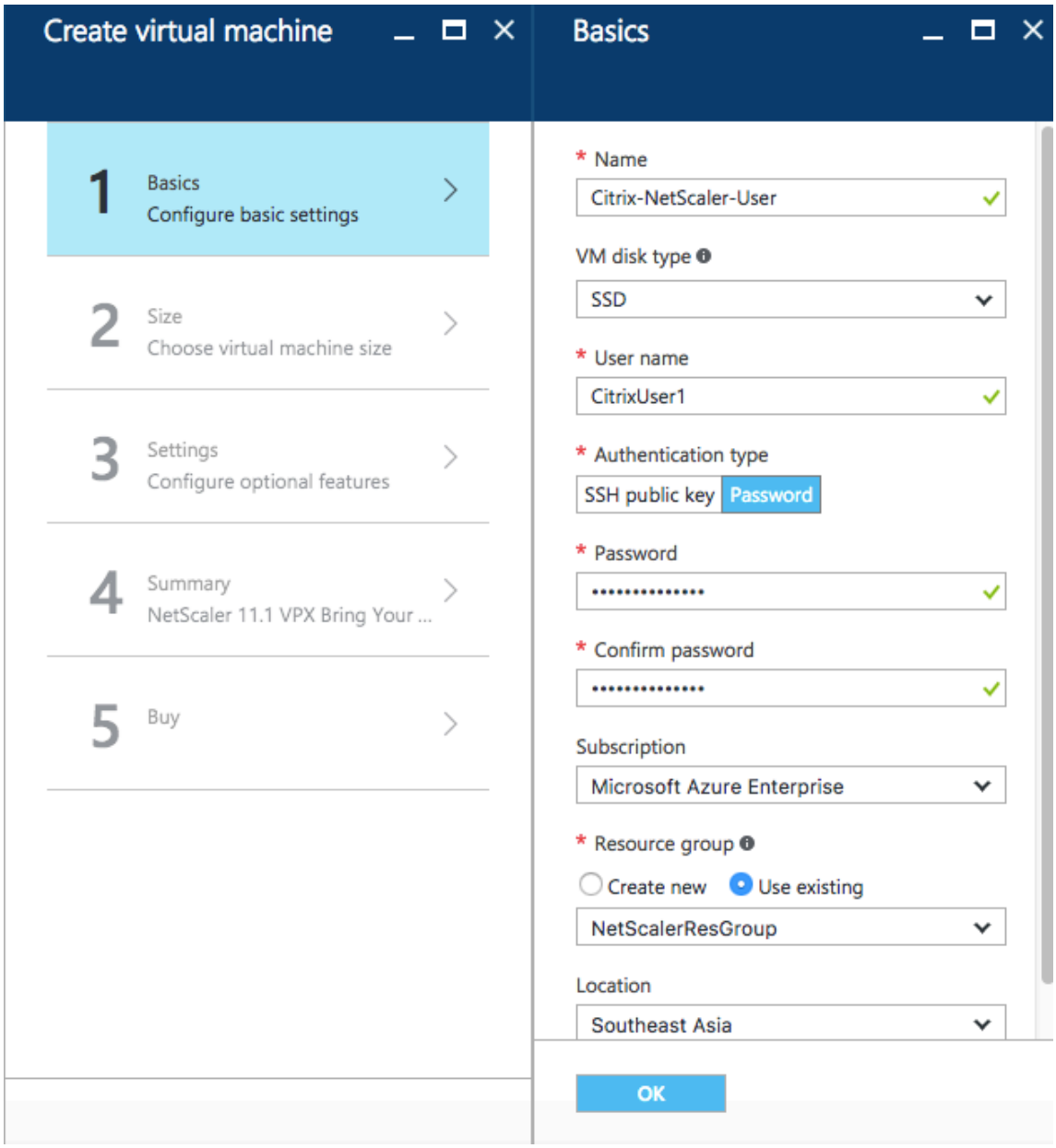
   ARM ポータル上で任意のエンティティをすばやく検出する方法として、Azure Marketplace 検索ボックスにエンティティの名前を入力して、Enter キーを押す方法もあります。Citrix NetScaler イメージを検出するには、検索ボックスに「NetScaler」と入力します。



> 注

> 最新のイメージを選択するようにしてください。Citrix NetScaler イメージは、名前にリリース番号がついていることがあります。

4. [**Citrix ADC VPX** 自分のライセンスを持参] ページのドロップダウンリストから [リソースマネージャー] を選択し、[作成] をクリックします。



5. [仮想マシンの作成] ウィンドウで、各セクションで必要な値を指定して、仮想マシンを作成します。各セクションで [**OK**] をクリックして、設定を保存します。

基本:

- 名前-Citrix ADC VPX インスタンスの名前を指定します。
- VM disk type - ボックスの一覧から SSD （デフォルト値）または HDD を選択します。

- User name and Password - 作成したリソースグループのリソースにアクセスするためのユーザー名およびパスワードを指定します
- Authentication Type - ［SSH Public Key］または［Password］を選択します
- Resource group - ボックスの一覧から作成したリソースグループを選択します。

ここではリソースグループを作成できますが、Azure Resource Manager で［Resource groups］からリソースグループを作成して、そのグループをボックスの一覧から選択することをお勧めします

> 注
>
> Azure スタック環境では、基本パラメータに加えて、次のパラメータを指定します。
>
> - Azure スタックドメイン
> - Azure スタックテナント (オプション)
> - Azure クライアント (オプション)
> - Azure クライアントシークレット (オプション)

サイズ:

基本設定で選択した仮想マシンディスクタイプ、SDD、または HDD に応じて、ディスクサイズが表示されます。

- 必要に応じてディスクサイズを選択し、[**Select**] をクリックします。

設定:

- デフォルトのディスクタイプ（[Standard]）を選択します
- Storage account - ストレージアカウントを選択します
- Virtual network - 仮想ネットワークを選択します
- Subnet - サブネットアドレスを設定します
- Public IP address - IP アドレス割り当ての種類を選択します
- Network security group - 作成したセキュリティグループを選択します。セキュリティグループで、受信規則および送信規則が構成されていることを確認します。
- 可用性セット-ドロップダウンメニューボックスから可用性セットを選択します。

要約:

構成設定が検証され、［Summary］ページに検証の結果が表示されます。検証が失敗すると、［Summary］ページに障害の理由が表示されます。個別のセクションに戻り、必要に応じて変更します。検証に合格したら、「**OK**」をクリックします。

購入:

「購入」ページでオファーの詳細と法的条件を確認し、「購入」をクリックします。

高可用性を展開するには、Citrix ADC VPX 独立したインスタンスを同じ可用性セットと同じリソースグループに 2 つ作成し、アクティブ/スタンバイ構成で展開します。

Citrix ADC VPX スタンドアロンインスタンスの複数の IP アドレスを構成する

October 7, 2021

このセクションでは、Azure Resource Manager（ARM）で複数の IP アドレスを使用してスタンドアロン Citrix ADC VPX インスタンスを構成する方法について説明します。VPX インスタンスには 1 つまたは複数の NIC を接続でき、各 NIC には 1 つ以上の静的または動的なパブリック IP アドレスとプライベート IP アドレスを割り当てることができます。複数の IP アドレスを NSIP、VIP、SNIP などとして割り当てることができます。

詳細については、Azure のドキュメント「Azure ポータルを使用して仮想マシンに複数の IP アドレスを割り当てる」を参照してください。

PowerShell コマンドを使用する場合は、「PowerShell コマンドを使用してスタンドアロンモードで Citrix ADC VPX インスタンスの複数の IP アドレスを構成する」を参照してください。

使用例

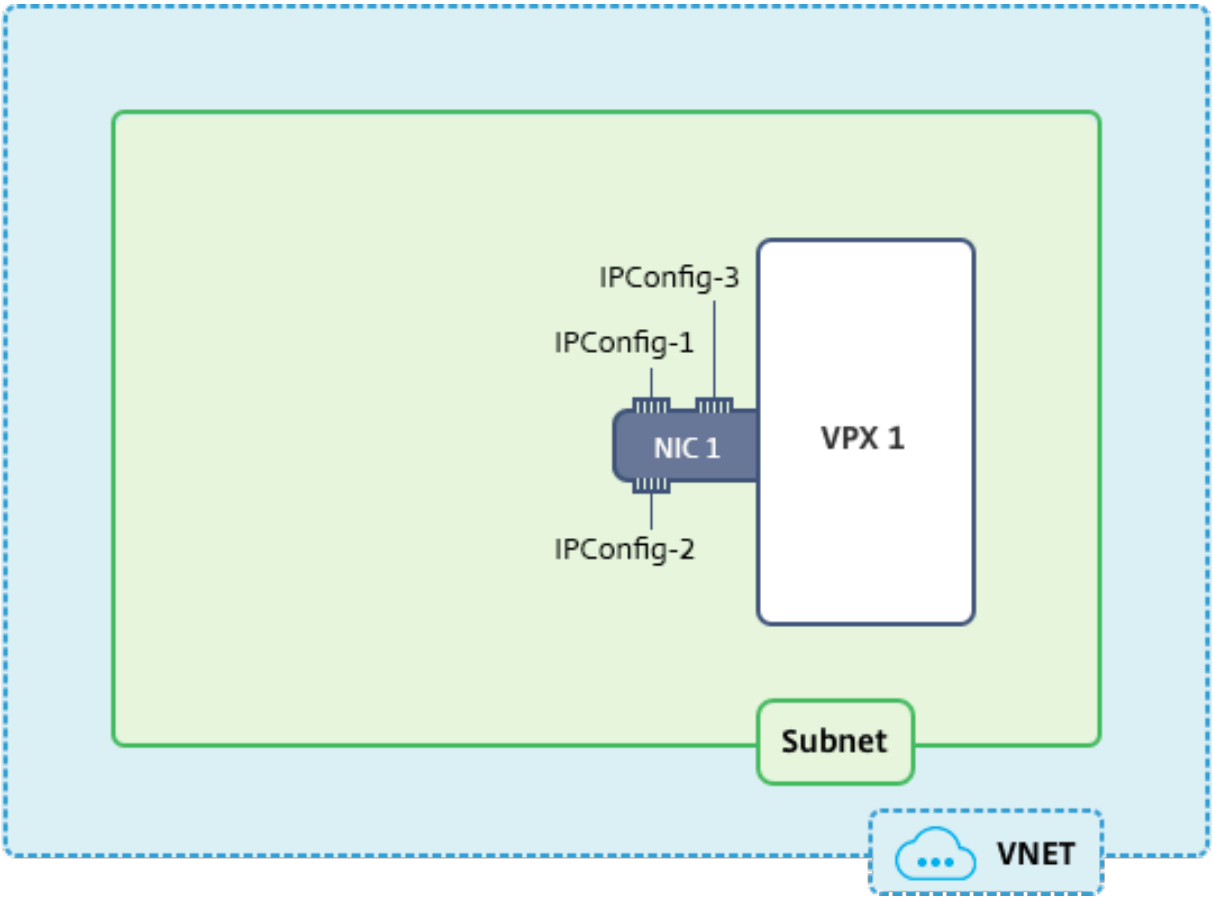
この使用例では、スタンドアロンの Citrix ADC VPX アプライアンスは、仮想ネットワーク（VNET）に接続された単一の NIC で構成されます。NIC は、表に示すように、3 つの IP 構成（ipconfig）に関連付けられ、各サーバは異なる目的で使用されます。

IP コンフィグ	関連付けられている	目的
ipconfig1	静的パブリック IP アドレス、静的プライベート IP アドレス	管理トラフィックを提供する
ipconfig2	静的パブリック IP アドレス、静的プライベートアドレス	クライアント側のトラフィックを提供する
ipconfig3	静的プライベート IP アドレス	バックエンドサーバーと通信する

> 注
>
> `IPConfig-3`はパブリック IP アドレスに関連付けられていません。

図: トポロジ

次の図はこの使用例を視覚的に示しています。



> 注
>
> マルチ NIC、マルチ IP Azure Citrix ADC VPX 展開では、プライマリ（最初の）NIC のプライマリ（最初）`IPConfig` に関連付けられたプライベート IP が、アプライアンスの管理 NSIP として自動的に追加されます。`IPConfigs`に関連付けられた残りのプライベート IP アドレスは、必要に応じて、`add ns ip` コマンドを使用して VIP または SNIP として VPX インスタンスに追加する必要があります。

はじめに

始める前に、次のリンクに示す手順に従って VPX インスタンスを作成します。

Citrix ADC VPX スタンドアロンインスタンスの構成

このユースケースでは、NSDoc0330VM VPX インスタンスが作成されます。

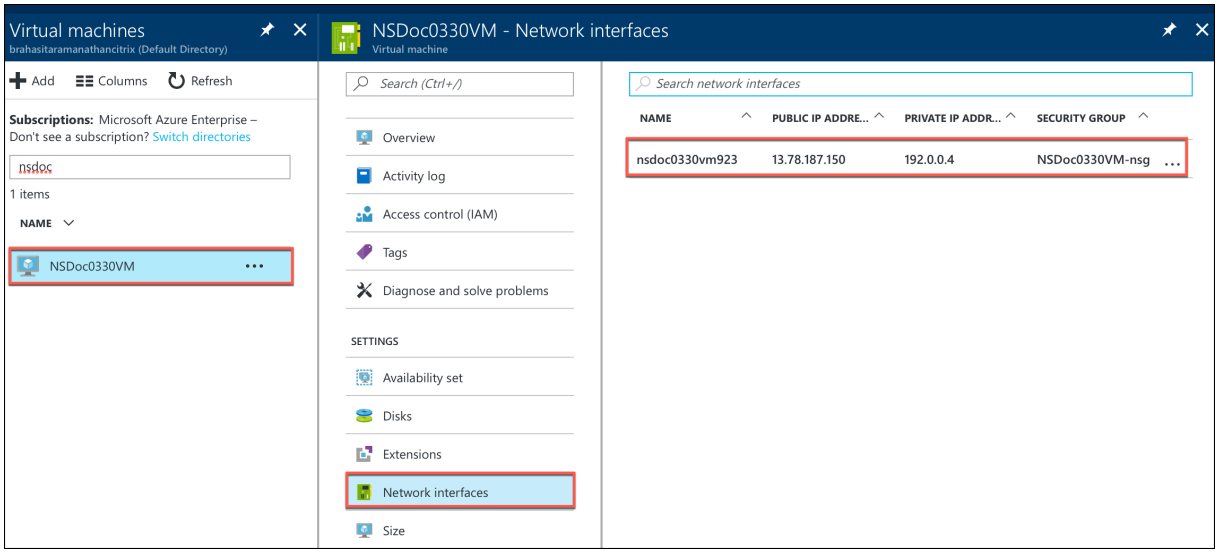
スタンドアロンモードで **Citrix ADC VPX** インスタンスの複数の **IP** アドレスを構成する手順。

スタンドアロンモードで Citrix ADC VPX アプライアンスの複数の IP アドレスを構成するには：

1. VM への IP アドレス追加
2. Citrix ADC が所有する IP アドレスを構成する

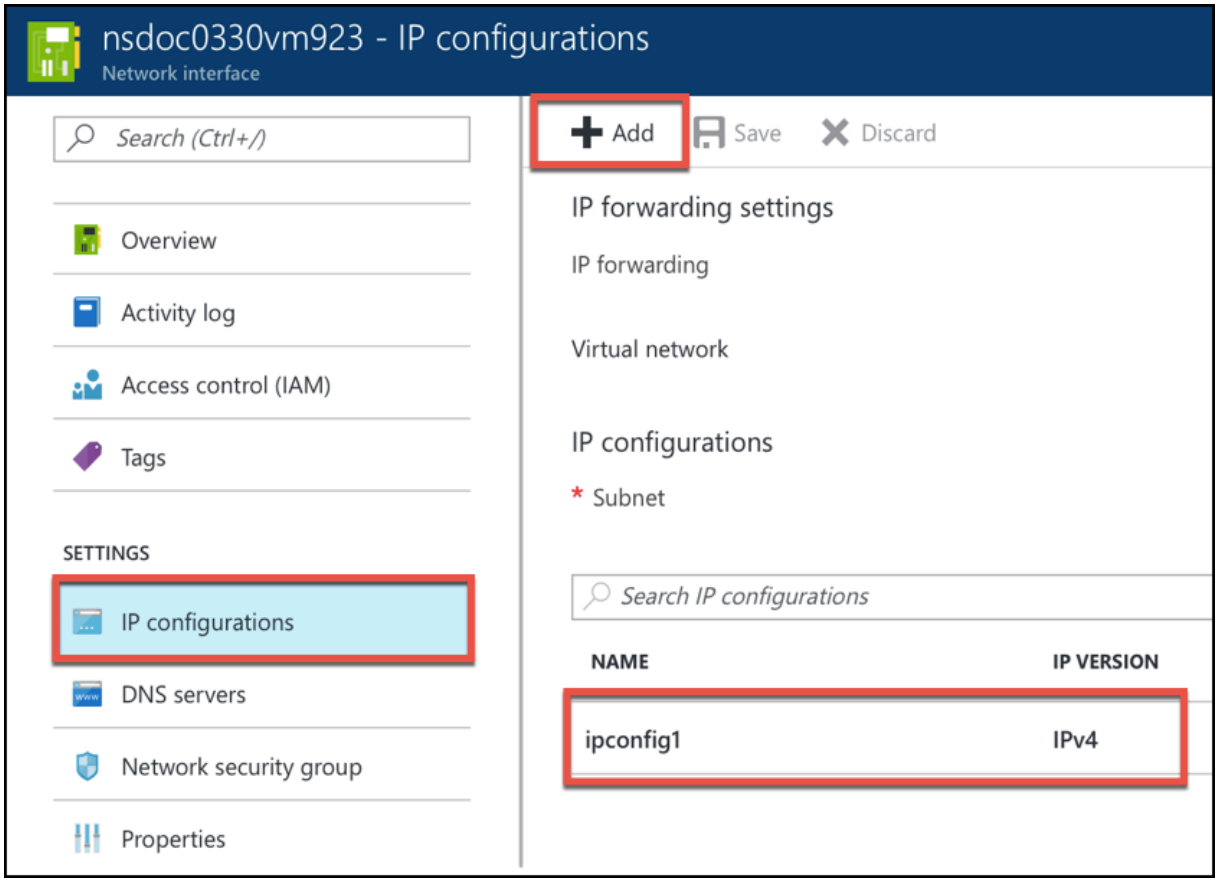
手順 **1**: 仮想マシンに IP アドレスを追加する

1. ポータルで、[その他のサービス] をクリックし、フィルターボックスに「仮想マシン」と入力し、[仮想マシン] をクリックします。
2. [仮想マシン] ブレードで、IP アドレスを追加する仮想マシンをクリックします。表示される仮想マシンブレードの [ネットワークインターフェイス] をクリックし、ネットワークインターフェイスを選択します。



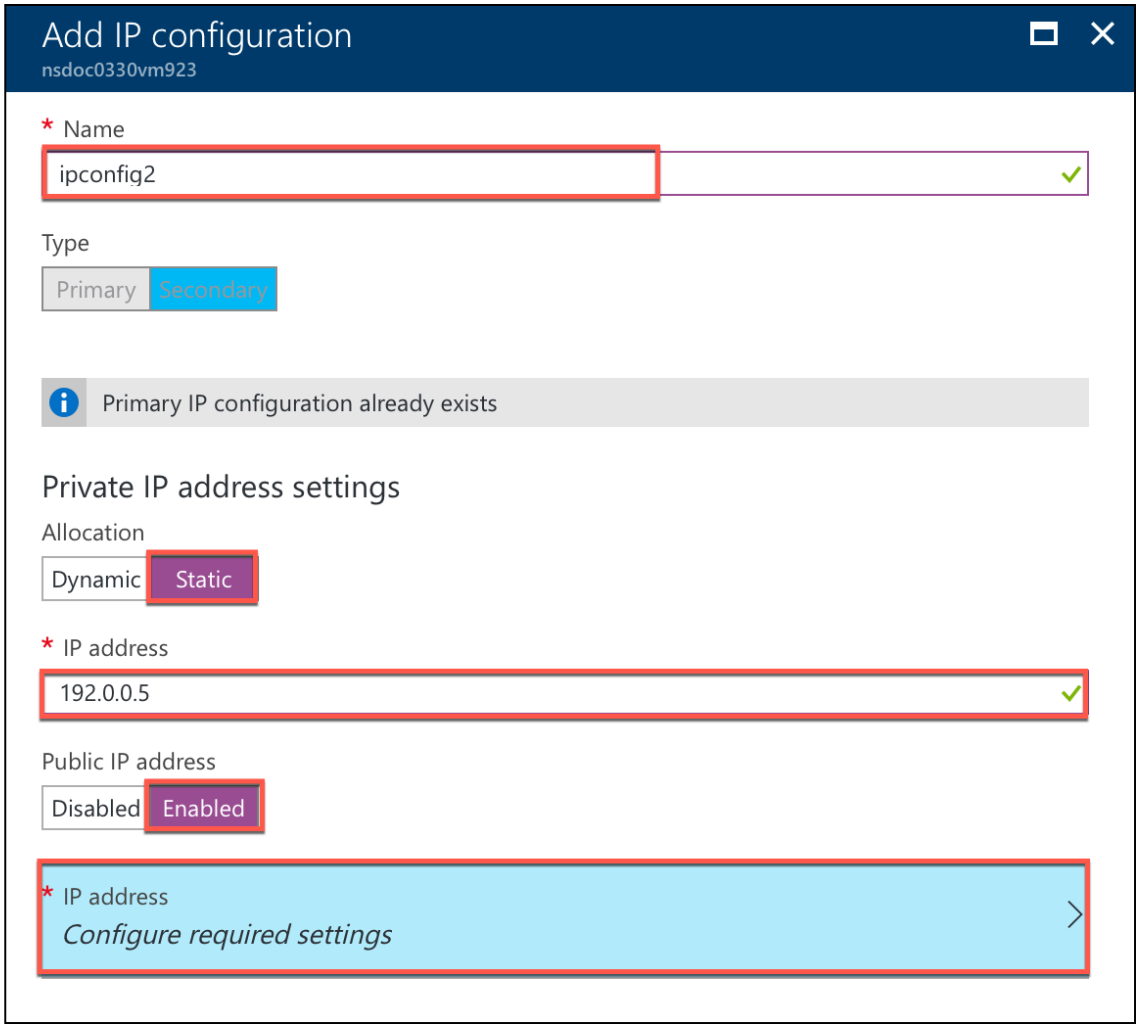
選択した NIC のブレードで、[**IP** 構成] をクリックします。仮想マシン **ipconfig1** の作成時に割り当てられていた既存の IP 構成が表示されます。この使用例では、ipconfig1 に割り当てられている IP アドレスが静的アドレスであることを確認します。次に、さらに 2 つの IP 構成、ipconfig2（VIP）と ipconfig3（SNIP）を作成します。

さらに`ipconfigs`を作成するには、**Add** を作成します。



[**IP** 構成の追加] ウィンドウで、[名前] を入力し、割り当て方法として [静的] を指定し、IP アドレス (このユースケースでは 192.0.0.5) を入力し、[パブリック **IP** アドレス] を有効にします。

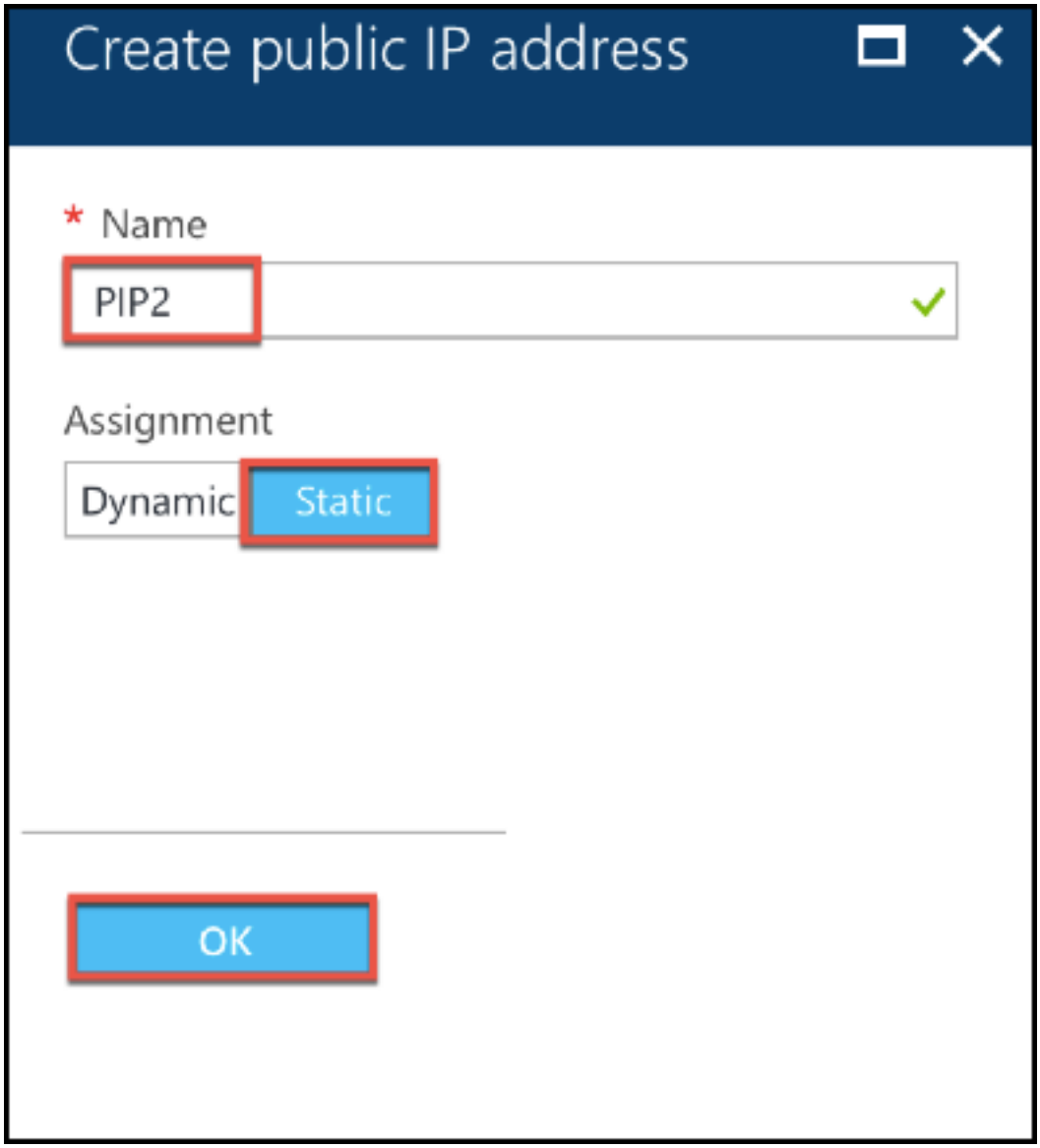
> 注
>
> 静的なプライベート IP アドレスを追加する前に、IP アドレスの可用性をチェックし、その IP アドレスが、NIC の接続先と同じサブネットに属していることを確認します。



次に、[必要な設定の構成] をクリックして ipconfig2 の静的パブリック IP アドレスを作成します。

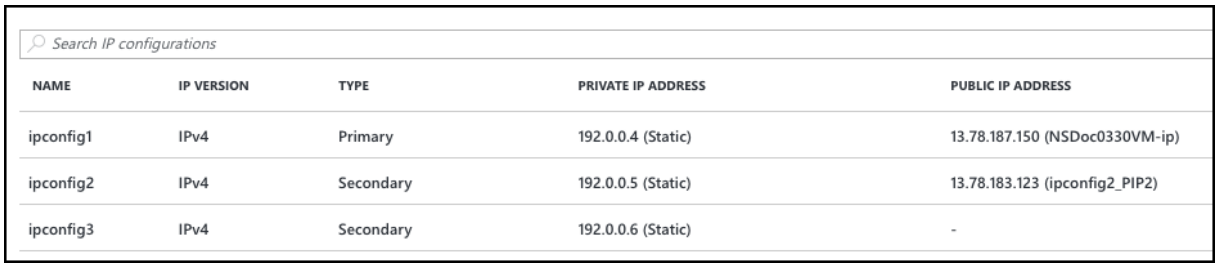
デフォルトでは、パブリック IP アドレスは動的なアドレスです。VM に常に同じパブリック IP アドレスを使用させるために、静的なパブリック IP アドレスを作成します。

[パブリック IP アドレスの作成] ブレードで、[名前] を追加し、[割り当て] で [静的] をクリックします。[**OK**] をクリックします。



> 注
>
> 割り当て方式を静的に設定している場合でも、パブリック IP リソースに割り当てられる実際の IP アドレスを指定することはできません。代わりに、リソースが作成された Azure の場所で利用可能な IP アドレスプールから割り当てられます。

手順に従って、もう 1 つの IP 構成 ipconfig3 を追加します。パブリック IP アドレスは必須ではありません。



ステップ **2**: **Citrix ADC** が所有する **IP** アドレスを構成する

GUI または`add ns ip`コマンドを使用して、Citrix ADC が所有する IP アドレスを設定します。詳細については、「Citrix ADC 所有の IP アドレスの構成」を参照してください。

複数の **IP** アドレスと **NIC** を使用した高可用性セットアップの構成

January 31, 2022

Microsoft Azure デプロイメントでは、Azure ロードバランサー（ALB）を使用して、2 つの Citrix ADC VPX インスタンスの高可用性構成を実現します。これは、ALB でヘルスプローブを構成することによって実現されます。ALB は、プライマリインスタンスとセカンダリインスタンスの両方に 5 秒ごとにヘルスプローブを送信することで、各 VPX インスタンスを監視します。

この設定では、プライマリノードだけがヘルスプローブに応答し、セカンダリノードは応答しません。プライマリがヘルスプローブに応答を送信すると、ALB はインスタンスへのデータトラフィックの送信を開始します。プライマリインスタンスで連続した 2 つのヘルスプローブが見つからない場合、ALB はそのインスタンスにトラフィックをリダイレクトしません。フェイルオーバー時は、新しいプライマリがヘルスプローブへの応答を開始し、ALB はそのプライマリにトラフィックをリダイレクトします。標準の VPX 高可用性フェイルオーバー時間は 3 秒です。トラフィックスイッチングにかかる合計フェールオーバー時間は、最大 13 秒です。

Azure のアクティブ/パッシブ高可用性（HA）セットアップで、複数の NIC を持つ一対の Citrix ADC VPX インスタンスを展開できます。各 NIC に複数の IP アドレスを設定できます。

マルチ NIC 高可用性展開では、次のオプションを使用できます。

- Azure 可用性セットを使用した高可用性
- Azure アベイラビリティーゾーンを使用した高可用性

Azure アベイラビリティセットとアベイラビリティーゾーンの詳細については、Azure のドキュメント「Linux 仮想マシンの可用性の管理」を参照してください。

可用性セットを使用した高可用性

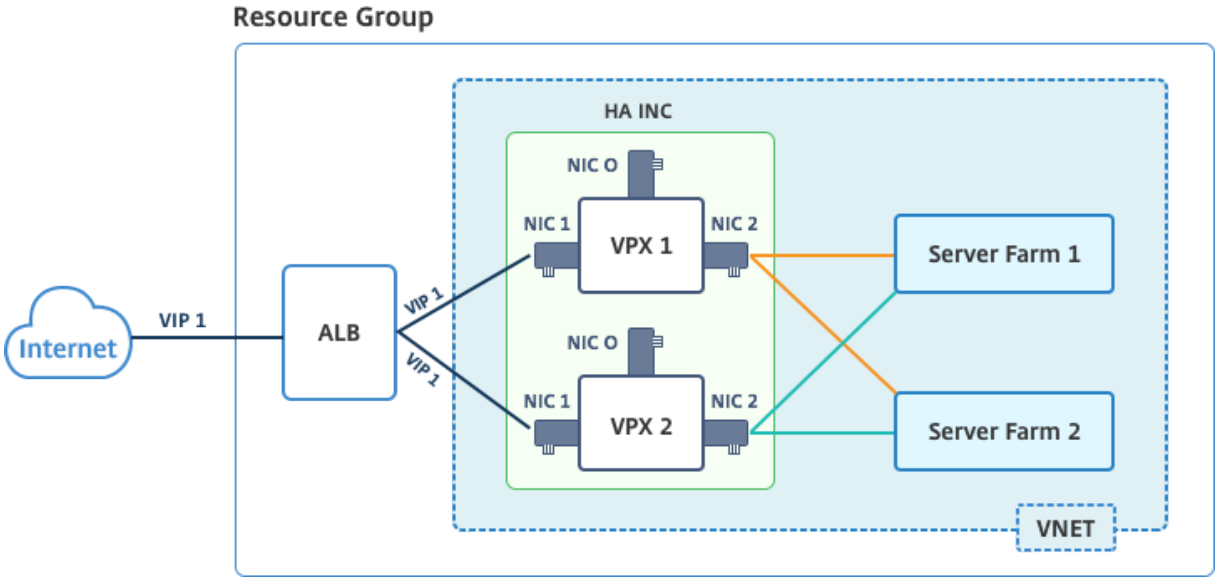
可用性セットを使用した高可用性セットアップは、次の要件を満たす必要があります。

- HA Independent Network Configuration（INC）構成
- Direct Server Return（DSR）モードの Azure Load Balancer（ALB）

すべてのトラフィックはプライマリノードを通過します。セカンダリノードは、プライマリノードが失敗するまでスタンバイモードを維持します。

> 注
>
> Azure クラウドでの CitrixVPX 高可用性展開を機能させるには、2 つの VPX ノード間で移動できるフローティングパブリック IP（PIP）が必要です。Azure ロードバランサー (ALB) は、フローティング PIP を提供します。このフローティング PIP は、フェールオーバーが発生した場合に自動的に第 2 ノードに移動されます。

図: Azure 可用性セットを使用した高可用性デプロイアーキテクチャの例



アクティブ/パッシブ展開では、ALB フロントエンドパブリック IP (PIP) アドレスが各 VPX ノードに VIP アドレスとして追加されます。HA-INC 構成では、これらの VIP アドレスはフローティングされ、SNIP アドレスはインスタンス固有のアドレスとなります。

VPX ペアをアクティブ-パッシブ高可用性モードで展開するには、次の 2 つの方法があります。

- **Citrix ADC VPX** 標準の高可用性テンプレート: 3 つのサブネットと 6 つの NIC のデフォルトオプションを使用して HA ペアを構成するには、このオプションを使用します。
- **Windows PowerShell** コマンド: サブネットと NIC の要件に従って HA ペアを構成するには、このオプションを使用します。

このトピックでは、Citrix テンプレートを使用してアクティブ/パッシブ HA セットアップで VPX ペアを展開する方法について説明します。PowerShell コマンドを使用する場合は、PowerShell コマンドを使用して複数の IP アドレスと NIC を使用した HA セットアップの構成を参照してください。

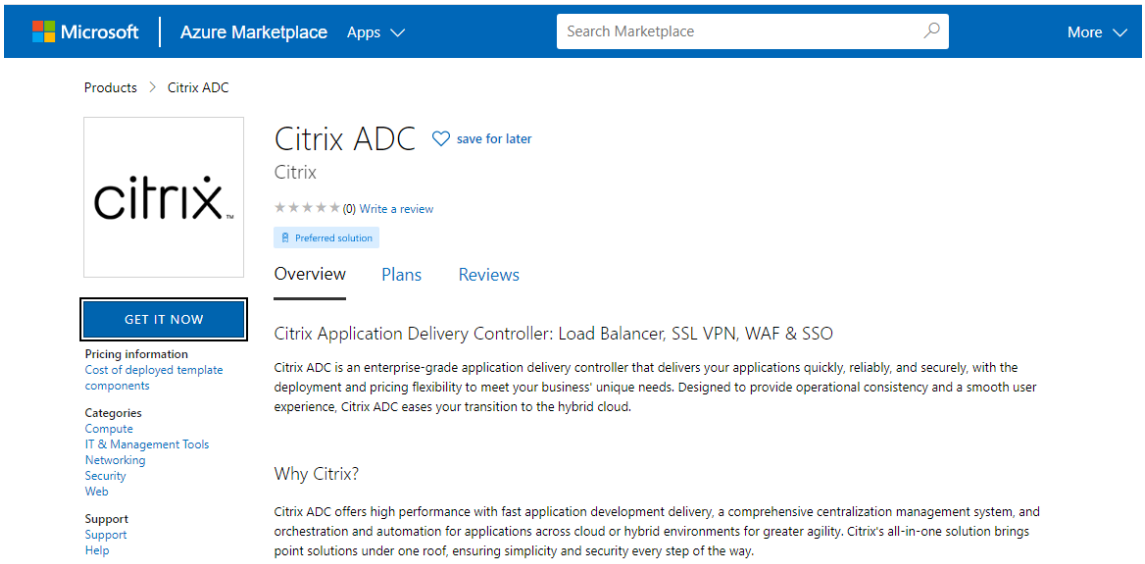
Citrix 高可用性テンプレートを使用して **HA-INC** ノードを構成する

標準テンプレートを使用すると、VPX インスタンスのペアを HA-INC モードで迅速かつ効率的に展開できます。テンプレートは、3 つのサブネットと 6 つの NIC を持つ 2 つのノードを作成します。サブネットは管理、クライアント、サーバー側のトラフィック用です。各サブネットには、両方の VPX インスタンスに対して 2 つの NIC があります。

Citrix ADC HA ペアテンプレートは、Azure マーケットプレイスで入手できます。

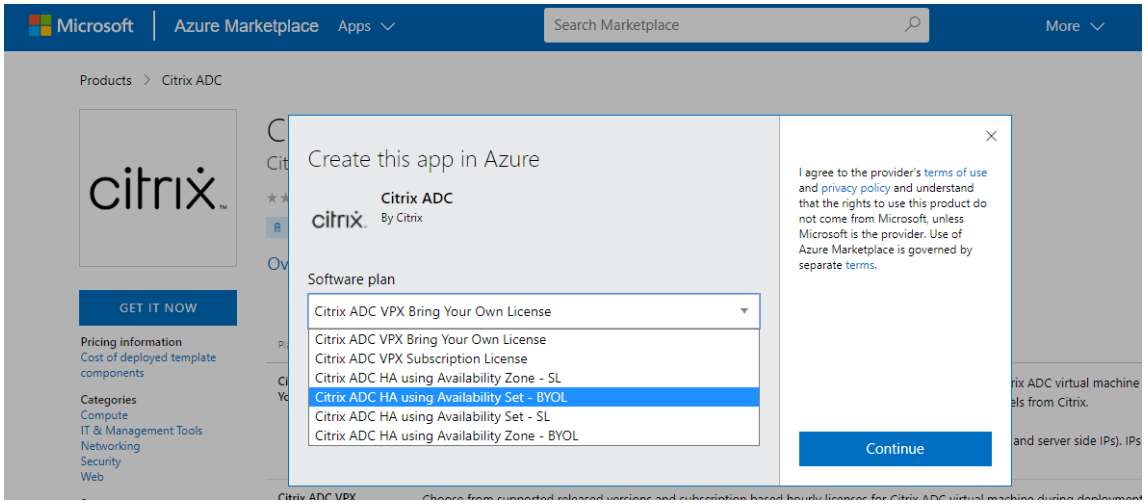
次の手順を実行して、Azure 可用性セットを使用して、テンプレートを起動し、高可用性 VPX ペアをデプロイします。

1. Azure Marketplace から、Citrix ADC を検索し ます。

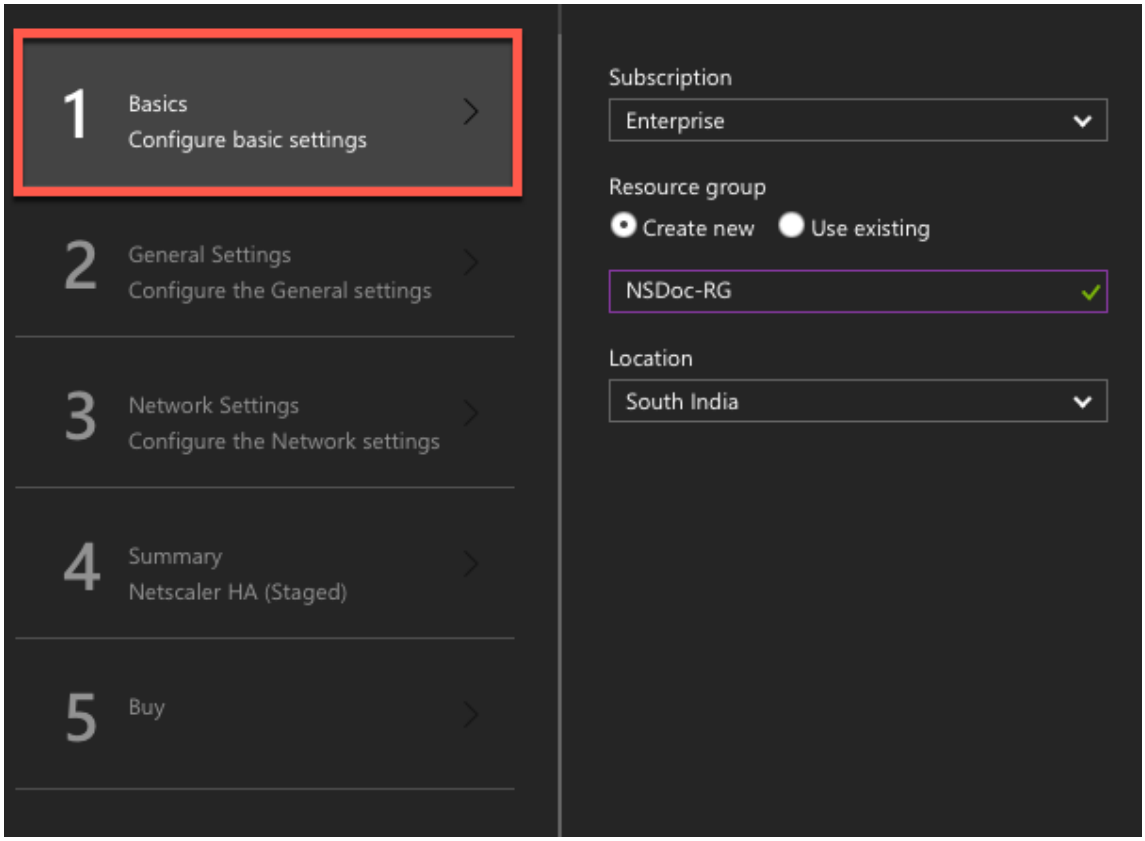


2. [今すぐ入手] をクリックします。

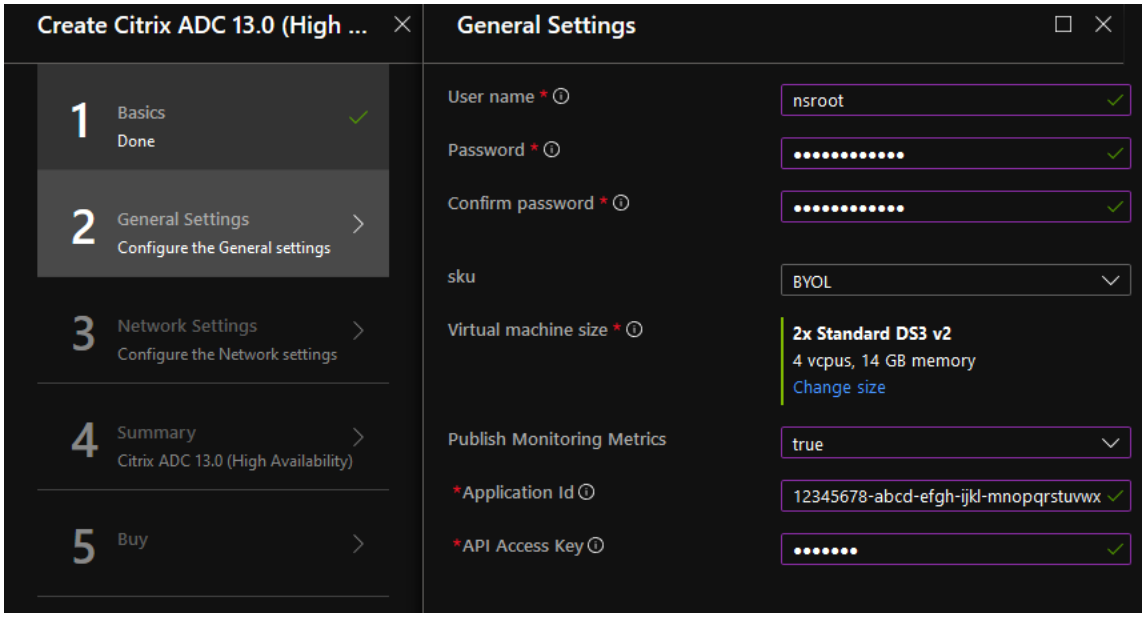
3. ライセンスとともに必要な HA 展開を選択し、[続行] をクリックします。



4.「基本」ページが表示されます。リソースグループを作成し、「**OK**」を選択します。



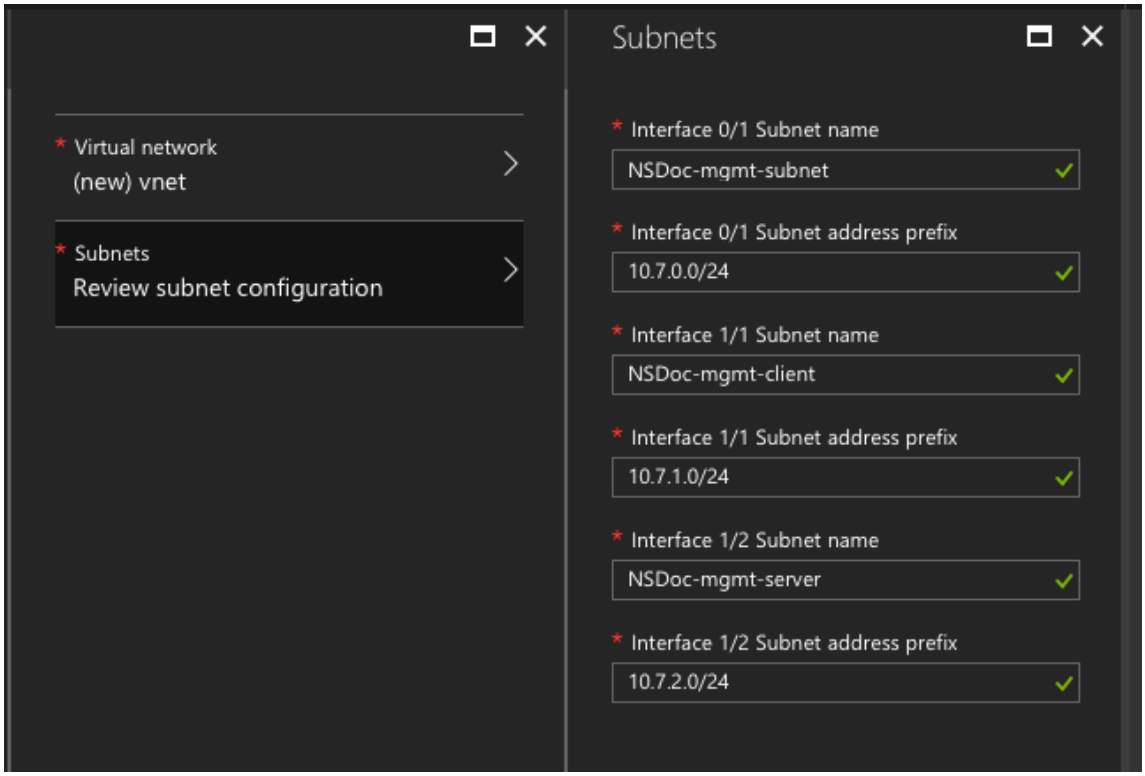
5. [一般設定] ページが表示されます。詳細を入力し、[**OK**] を選択します。



> 注:
>
> デフォルトでは、**Publishing MonitoringMetrics** オプションは **false** に設定されています。このオプションを有効にする場合は、**true** を選択します。

> リソースにアクセスできる AzureActive Directory（ADD）アプリケーションとサービスプリンシパルを作成します。新しく作成された AAD アプリケーションにコントリビュータロールを割り当てます。詳細については、「ポータルを使用してリソースにアクセスできる Azure Active Directory アプリケーションおよびサービスプリンシパルを作成する」を参照してください。

6. [ネットワーク設定] ページが表示されます。VNet とサブネットの構成を確認し、必要な設定を編集して、[**OK**] を選択します。

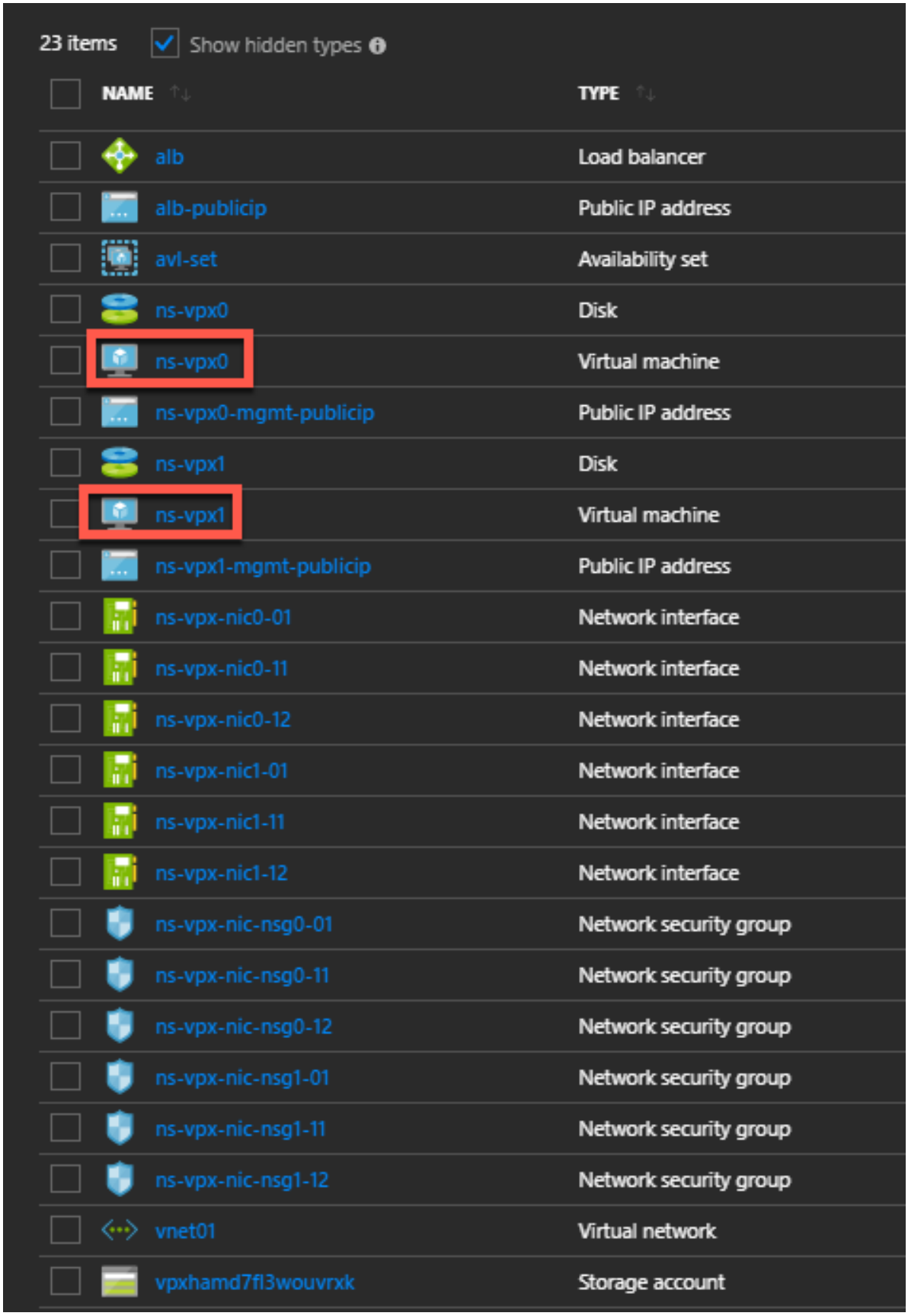


7. ［概要］ページが開きます。構成を確認し、適宜編集します。［**OK**］を選択して確定します。

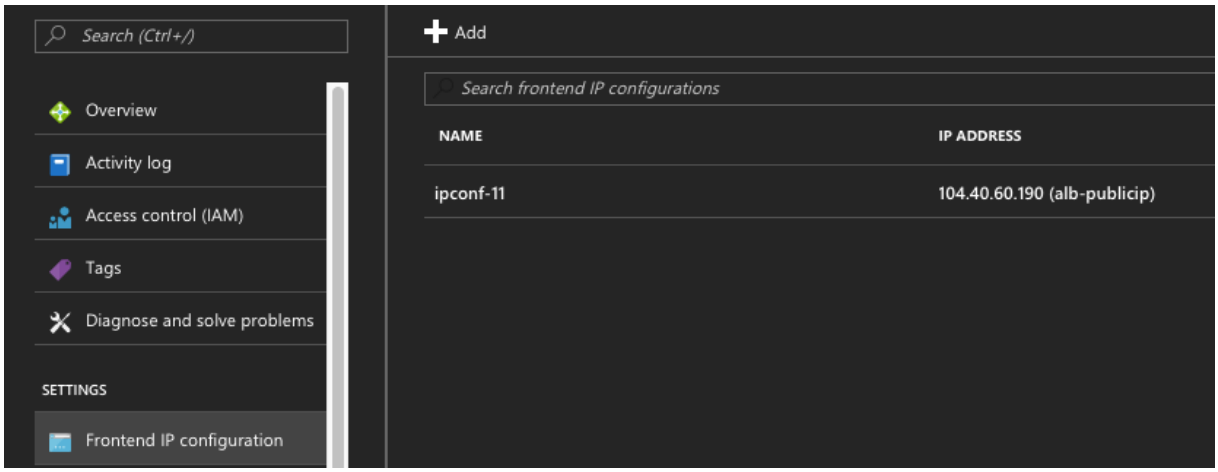
8. 「購入」ページが表示されます。[購入] を選択して、展開を完了します。

必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了後、Azure ポータルでリソースグループを選択し、LB ルール、バックエンドプール、ヘルスプローブなどの構成の詳細を表示します。高可用性ペアは、ns-vpx0 および ns-vpx1 と表示されます。

追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。



次に、プライマリノードで **ALB** のフロントエンドパブリック **IP**（**PIP**）アドレスを使用して負荷分散仮想サーバーを構成する必要があります。ALB PIP を検索するには、[ALB] > [フロントエンド **IP** 構成] を選択します。



負荷分散仮想サーバーを構成する方法の詳細については、「リソース」セクションを参照してください。

リソース:

次のリンクは、HA の展開と仮想サーバーの構成に関連する追加情報を提供します。

- 異なるサブネットでの高可用性ノードの構成
- 基本的な負荷分散の設定

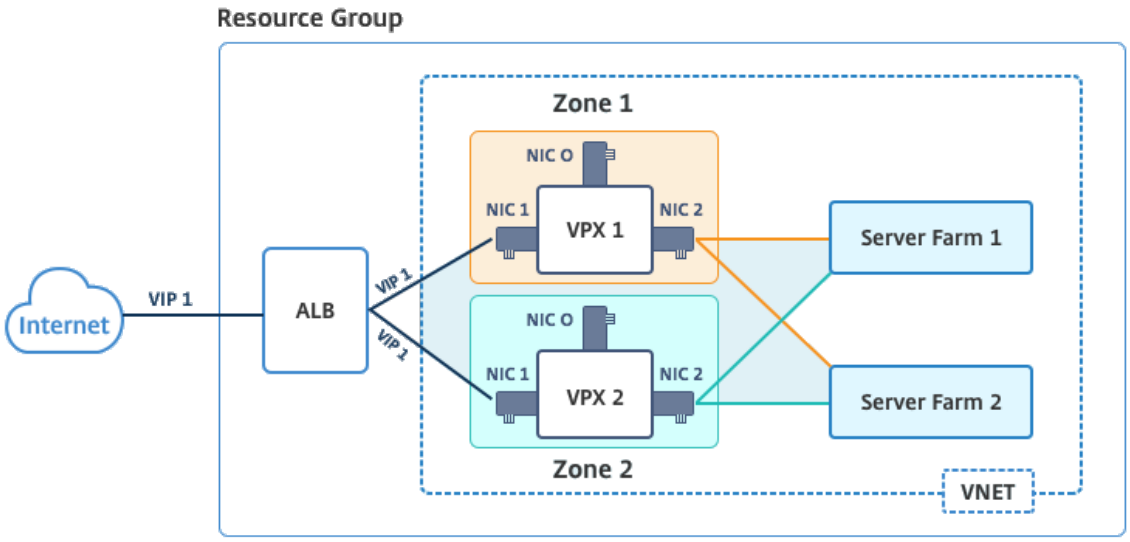
関連リソース:

- PowerShell コマンドを使用して複数の IP アドレスと NIC を使用して高可用性セットアップを構成する
- Azure でのアクティブスタンバイ HA デプロイメントでの GSLB の構成

アベイラビリティーゾーンを使用した高可用性

Azure アベイラビリティーゾーンは、Azure リージョン内の障害分離された場所であり、冗長な電源、冷却、ネットワーキングを提供し、回復力を高めます。特定の Azure リージョンだけがアベイラビリティーゾーンをサポートします。詳細については、Azure のドキュメント [Azure のアベイラビリティーゾーンとは何ですか] を参照してください。

図:Azure アベイラビリティーゾーンを使用した高可用性デプロイアーキテクチャの例



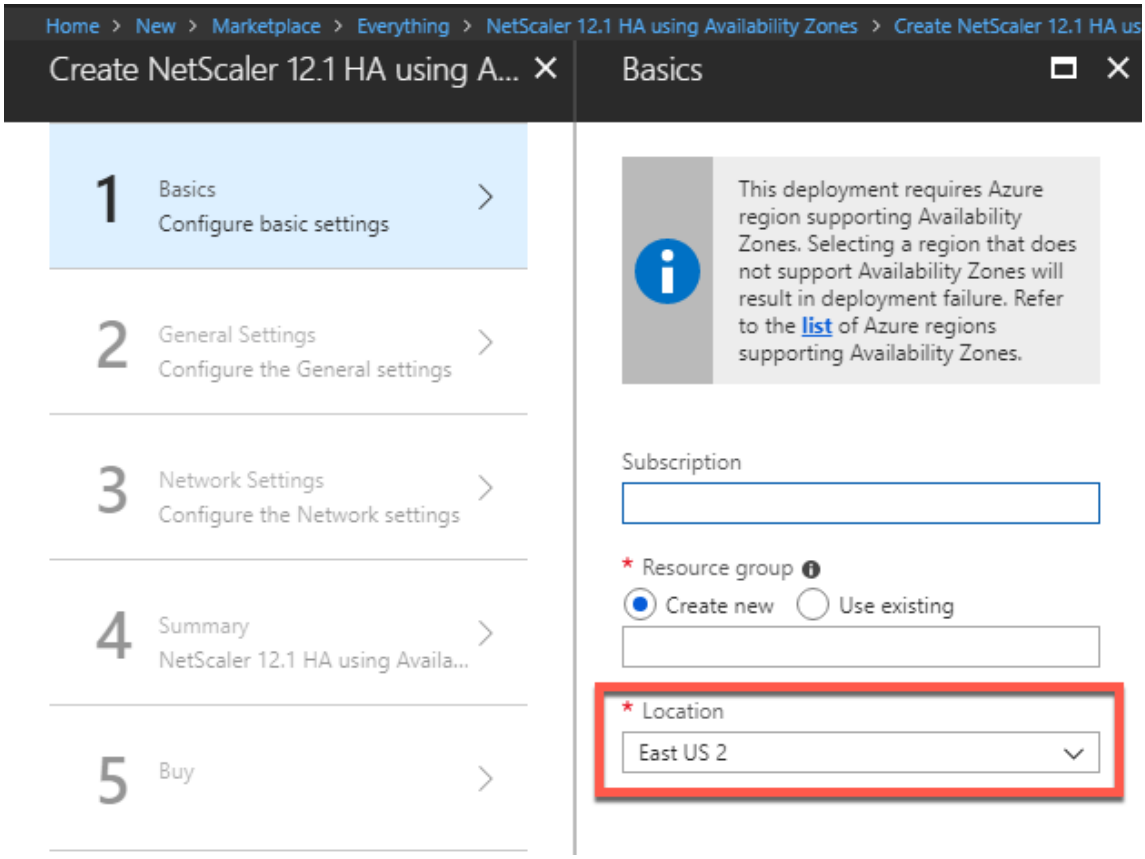
Azure マーケットプレイスで利用可能な「アベイラビリティーゾーンを使用した NetScaler 13.0 HA」というテンプレートを使用して、VPX ペアを高可用性モードで展開できます。

Azure アベイラビリティーゾーンを使用してテンプレートを起動し、高可用性 VPX ペアをデプロイするには、次の手順を実行します。

1. Azure Marketplace から、Citrix ソリューションテンプレートを選択して開始します。

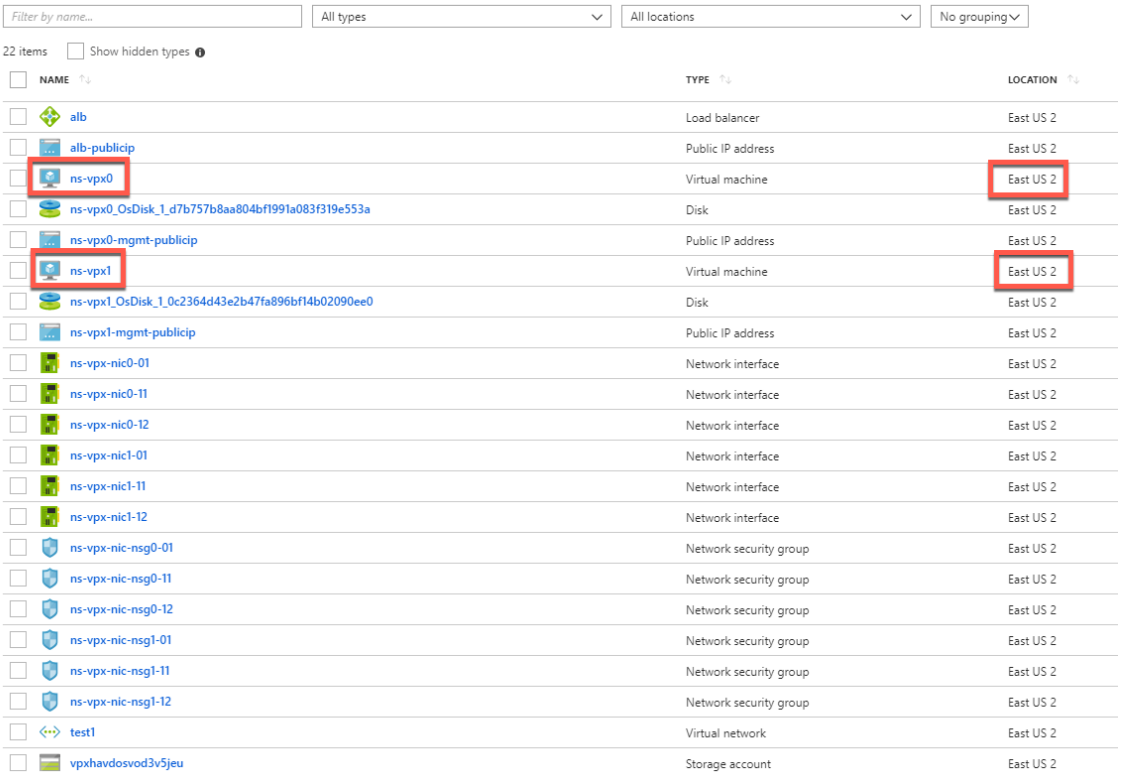
2. デプロイメントの種類が「リソースマネージャー」であることを確認し、「作成」を選択します。
3. 「基本」ページが表示されます。詳細を入力し、[**OK**] をクリックします。

> 注: アベイラビリティーゾーンをサポートする Azure リージョンを選択してください。アベイラビリティーゾーンをサポートするリージョンの詳細については、Azure のドキュメントを参照してください。Azure のアベイラビリティーゾーンは何ですか。



4. [一般設定] ページが表示されます。詳細を入力し、[**OK**] を選択します。
5. [ネットワーク設定] ページが表示されます。VNet とサブネットの構成を確認し、必要な設定を編集して、[**OK**] を選択します。
6. [概要] ページが開きます。構成を確認し、適宜編集します。[**OK**] を選択して確定します。
7. 「購入」ページが表示されます。[購入] を選択して、展開を完了します。

   必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了後、リソースグループを選択して、LB ルール、バックエンドプール、正常性プローブなどの構成の詳細を Azure Portal に表示します。高可用性ペアは、ns-vpx0 および ns-vpx1 と表示されます。また、[場所] 列の下に場所が表示されます。



追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。

Azure モニターのメトリックを使用してインスタンスを監視する

Azure モニターデータプラットフォームのメトリックを使用して、CPU、メモリ使用率、スループットなどの一連の Citrix ADC VPX リソースを監視できます。Metrics サービスは、Azure で実行される Citrix ADC VPX リソースをリアルタイムで監視します。メトリックエクスプローラーを使用して、収集されたデータにアクセスできます。詳細については、「Azure Monitor メトリックスの概要」を参照してください。

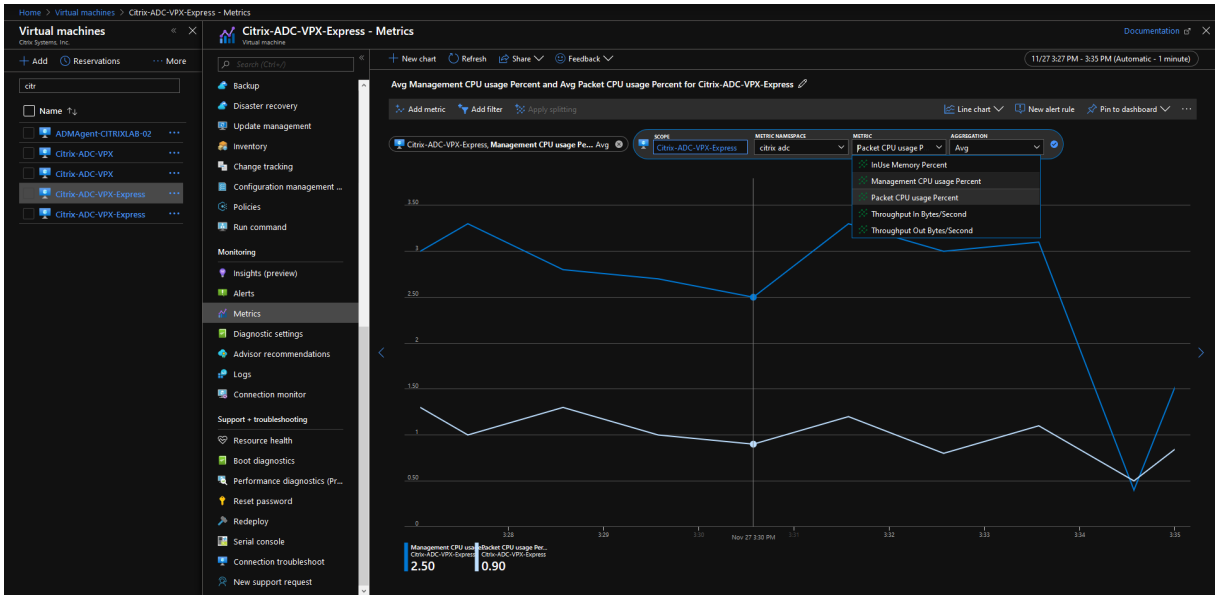
注意事項

- AzureMarketplace オファーを使用して Citrix ADC VPX インスタンスを Azure にデプロイすると、メトリックサービスはデフォルトで無効になります。
- Metrics サービスは AzureCLI ではサポートされていません。
- メトリックは、CPU（管理およびパケット CPU 使用率）、メモリ、およびスループット（インバウンドおよびアウトバウンド）で使用できます。

Azure モニターでメトリックを表示する方法

インスタンスの Azure モニターでメトリックスを表示するには、次の手順を実行します。

1. **Azure Portal > Virtual Machines** にログオンします。
2. プライマリノードである仮想マシンを選択します。
3. [監視] セクションで、[メトリック] をクリックします。
4. [メトリックネームスペース] ドロップダウンメニューから、[**Citrix ADC**] をクリックします。
5. メトリックドロップダウンメニューの すべてのメトリックの下で、表示したいメトリックをクリックします。
6. 同じグラフに別の 指標を表示するには、[指標の追加] をクリックします。グラフオプションを使用して、グラフをカスタマイズします。



PowerShell コマンドを使用して複数の **IP** アドレスと **NIC** を使用して高可用性セットアップを構成する

October 7, 2021

Azure のアクティブ/パッシブ高可用性（HA）セットアップで、複数の NIC を持つ一対の Citrix ADC VPX インスタンスを展開できます。各 NIC に複数の IP アドレスを設定できます。

アクティブ/パッシブ展開では以下が必要です。

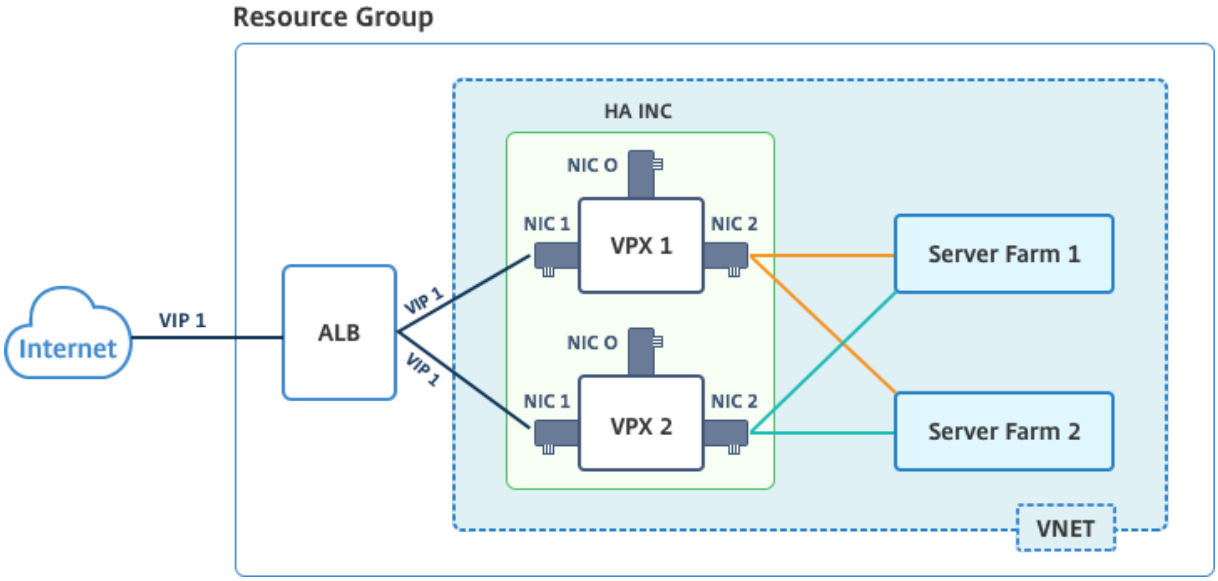
- HA Independent Network Configuration（INC）構成
- Direct Server Return（DSR）モードの Azure Load Balancer（ALB）

すべてのトラフィックはプライマリノードを通過します。セカンダリノードは、プライマリノードが失敗するまでスタンバイモードを維持します。

> 注
>
> Azure クラウド上の Citrix ADC VPX 高可用性展開を機能させるには、2 つの高可用性ノード間で移動できる

> フローティングパブリック IP（PIP）が必要です。Azure ロードバランサー (ALB) は、フローティング PIP を提供します。このフローティング PIP は、フェールオーバーが発生した場合に自動的に第 2 ノードに移動されます。

図: アクティブ-パッシブ展開アーキテクチャの例



アクティブ/パッシブ展開では、ALB フローティングパブリック IP（PIP）アドレスが各 VPX ノードに VIP アドレスとして追加されます。HA-INC 構成では、これらの VIP アドレスはフローティングされ、SNIP アドレスはインスタンス固有のアドレスとなります。

ALB は 5 秒ごとにヘルスプローブを送信して各 VPX インスタンスを監視し、定期的にヘルスプローブ応答を送信するトラフィックのみをそのインスタンスにリダイレクトします。そのため、HA セットアップでは、プライマリノードがヘルスプローブに応答し、セカンダリノードは応答しません。プライマリインスタンスが 2 つの連続したヘルスプローブを見逃した場合、ALB はそのインスタンスにトラフィックをリダイレクトしません。フェイルオーバー時は、新しいプライマリがヘルスプローブへの応答を開始し、ALB はそのプライマリにトラフィックをリダイレクトします。標準の VPX 高可用性フェイルオーバー時間は 3 秒です。トラフィック切り替えにかかる合計フェイルオーバー時間は、最大で 13 秒になる可能性があります。

VPX ペアをアクティブ-パッシブ HA セットアップで展開するには、次の 2 つの方法があります。

- **Citrix ADC VPX** 標準高可用性テンプレート: 3 つのサブネットと 6 つの NIC のデフォルトオプションを使用して HA ペアを構成するには、このオプションを使用します。
- **Windows PowerShell** コマンド: サブネットと NIC の要件に従って HA ペアを構成するには、このオプションを使用します。

このトピックでは、PowerShell コマンドを使用してアクティブ/パッシブ HA セットアップで VPX ペアを展開する方法について説明します。Citrix ADC VPX 標準 HA テンプレートを使用する場合は、複数の IP アドレスと NIC を使用した HA セットアップの構成を参照してください。

PowerShell コマンドを使用して **HA-INC** ノードを構成する

シナリオ**:HA-INC PowerShell** の展開

このシナリオでは、表に示されているトポロジを使用して、Citrix ADC VPX ペアを展開します。各 VPX インスタンスには 3 つの NIC があり、各 NIC は異なるサブネットに展開されます。各 NIC には IP 構成が割り当てられます。

ALB	VPX1	VPX2
ALB はパブリック IP 3 (pip3) に関連付けられています。	管理 IP は IPConfig1 を使用して構成され、これには 1 つのパブリック IP (パイプ 1) と 1 つのプライベート IP (12.5.2.24) が含まれます。nic1、Mgmt サブネット = 12.5.2.0/24	管理 IP は IPConfig5 を使用して構成され、これには 1 つのパブリック IP (パイプ 3) と 1 つのプライベート IP (12.5.2.26) が含まれます。nic4、Mgmt サブネット = 12.5.2.0/24
設定された LB ルールおよびポートは、HTTP（80）、SSL（443）、ヘルスプローブ（9000）です。	クライアント側の IP は IPConfig3 を使用して構成され、これには 1 つのプライベート IP (12.5.1.27) が含まれます。	クライアント側の IP は IPConfig7 で構成され、これには 1 つのプライベート IP (12.5.1.28) が含まれます。
-	サーバー側の IP は IPConfig4 を使用して構成され、これには 1 つのプライベート IP (12.5.3.24)、nic3、バックエンドサブネット = 12.5.3.0/24 が含まれます。	サーバー側の IP は IPConfig8 を使用して構成され、これには 1 つのプライベート IP (12.5.3.28)、nic6、バックエンドサブネット = 12.5.3.0/24 が含まれます。
-	NSG のルールとポートは、SSH (22)、HTTP (80)、HTTPS (443)	-

パラメータ設定

このシナリオでは、次のパラメータ設定が使用されます。

$locName = “East Asia”

$rgName = “MulitIP-MultiNIC-RG”

$nicName1= “VM1-NIC1”

$nicName2 = “VM1-NIC2”

$nicName3= “VM1-NIC3”

$nicName4 = “VM2-NIC1”

$nicName5= “VM2-NIC2”

$nicName6 = “VM2-NIC3”

$vNetName = “Azure-MultiIP-ALB-vnet”

$vNetAddressRange= “12.5.0.0/16”

$frontEndSubnetName= “frontEndSubnet”

$frontEndSubnetRange= “12.5.1.0/24”

$mgmtSubnetName= “mgmtSubnet”

$mgmtSubnetRange= “12.5.2.0/24”

$backEndSubnetName = “backEndSubnet”

$backEndSubnetRange = “12.5.3.0/24”

$prmStorageAccountName = “multiipmultinicbstorage”

$avSetName = “multiple-avSet”

$vmSize= “Standard_DS4_V2”

$publisher = “Citrix”

$offer = “netscalervpx-120”

$sku = “netscalerbyol”

$version=”latest”

$pubIPName1=”VPX1MGMT”

$pubIPName2=”VPX2MGMT”

$pubIPName3=”ALBPIP”

$domName1=”vpx1dns”

$domName2=”vpx2dns”

$domName3=”vpxalbdns”

$vmNamePrefix=”VPXMultiIPALB”

$osDiskSuffix1=”osmultiipalbdiskdb1”

$osDiskSuffix2=”osmultiipalbdiskdb2”

$lbName= “MultiIPALB”

$frontEndConfigName1= “FrontEndIP”

$backendPoolName1= “BackendPoolHttp”

$lbRuleName1= “LBRuleHttp”

$healthProbeName= “HealthProbe”

$nsgName=”NSG-MultiIP-ALB”

$rule1Name=”Inbound-HTTP”

$rule2Name=”Inbound-HTTPS”

$rule3Name=”Inbound-SSH”

展開を完了するには、PowerShell コマンドを使用して次の手順を完了します。

1. リソースグループ、ストレージアカウント、高可用性セットの作成
2. ネットワークセキュリティグループの作成と規則の追加
3. 仮想ネットワークと 3 つのサブネットの作成
4. パブリック IP アドレスの作成
5. VPX1 の IP 構成の作成
6. VPX2 の IP 構成の作成
7. VPX1 の NIC の作成
8. VPX2 の NIC の作成
9. VPX1 の作成
10. VPX2 の作成
11. ALB の作成

リソースグループ、ストレージアカウント、および可用性セットを作成します。

```
New-AzureRmResourceGroup -Name $rgName -Location $locName


$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $rgName -Type Standard_LRS
     -Location $locName


$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $rgName -Location $locName
```

ネットワークセキュリティグループを作成し、ルールを追加します。

```
$rule1 = New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -
    Description ”Allow HTTP” -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 101


```



```
-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 80


$rule2 = New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -
    Description "Allow HTTPS" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 110


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 443


$rule3 = New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -
    Description "Allow SSH" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 120


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 22


$nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
    Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

仮想ネットワークと **3** つのサブネットを作成します。

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontEndSubnetName -AddressPrefix $frontEndSubnetRange (this
    parameter value should be as per your requirement)


$mgmtSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $mgmtSubnetName
     -AddressPrefix $mgmtSubnetRange


$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backEndSubnetName -AddressPrefix $backEndSubnetRange


$vnet =New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
    $rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
    $frontendSubnet,$backendSubnet, $mgmtSubnet
```



```
11
12
13 $subnetName =”frontEndSubnet”
14
15
16 $subnet1=$vnet.Subnets|?{
17  $_.Name -eq $subnetName }
18
19
20
21 $subnetName=”backEndSubnet”
22
23
24 $subnet2=$vnet.Subnets|?{
25  $_.Name -eq $subnetName }
26
27
28
29 $subnetName=”mgmtSubnet”
30
31
32 $subnet3=$vnet.Subnets|?{
33  $_.Name -eq $subnetName }
```

パブリック **IP** アドレスを作成します。

```
1 $pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
     $rgName -DomainNameLabel $domName1 -Location $locName -
     AllocationMethod Dynamic
2
3 $pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
     $rgName -DomainNameLabel $domName2 -Location $locName -
     AllocationMethod Dynamic
4
5 $pip3=New-AzureRmPublicIpAddress -Name $pubIPName3 -ResourceGroupName
     $rgName -DomainNameLabel $domName3 -Location $locName -
     AllocationMethod Dynamic
```

VPX1 の **IP** 構成を作成します。

```
1 $IpConfigName1 = ”IPConfig1”
2
```



```
3
4  $IPAddress = ”12.5.2.24”
5
6
7  $IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
       Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip1
       -Primary
8
9
10 $IPConfigName3=”IPConfig-3”
11
12
13 $IPAddress=”12.5.1.27”
14
15
16 $IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
       Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary
17
18
19 $IPConfigName4 = ”IPConfig-4”
20
21
22 $IPAddress = ”12.5.3.24”
23
24
25 $IPConfig4 = New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
       Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

VPX2 の **IP** 構成を作成します。

```
1  $IpConfigName5 = ”IPConfig5”
2
3
4  $IPAddress=”12.5.2.26”
5
6
7  $IPConfig5=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName5 -
       Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip2
       -Primary
8
9
10 $IPConfigName7=”IPConfig-7”
11
```



```
12
13 $IPAddress=”12.5.1.28”
14
15
16 $IPConfig7=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName7 -
       Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary
17
18
19 $IPConfigName8=”IPConfig-8”
20
21
22 $IPAddress=”12.5.3.28”
23
24
25 $IPConfig8=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName8 -
       Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

VPX1 用の **NIC** を作成します。

```
1 $nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig1 -
      NetworkSecurityGroupId $nsg.Id
2
3
4 $nic2=New-AzureRmNetworkInterface -Name $nicName2 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig3 -
      NetworkSecurityGroupId $nsg.Id
5
6
7 $nic3=New-AzureRmNetworkInterface -Name $nicName3 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig4 -
      NetworkSecurityGroupId $nsg.Id
```

VPX2 用の **NIC** を作成します。

```
1 $nic4=New-AzureRmNetworkInterface -Name $nicName4 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig5 -
      NetworkSecurityGroupId $nsg.Id
2
3
4 $nic5=New-AzureRmNetworkInterface -Name $nicName5 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig7 -
```



```
    NetworkSecurityGroupId $nsg.Id
5
6
7 $nic6=New-AzureRmNetworkInterface -Name $nicName6 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig8 -
    NetworkSecurityGroupId $nsg.Id
```

VPX1 を作成します。

この手順には、次の下位手順が含まれています。

- VM 設定オブジェクトの作成
- 資格情報、OS、イメージの設定
- NIC の追加
- OS ディスクの指定と VM の作成

```
1  $suffixNumber = 1
2
3  $vmName=$vmNamePrefix + $suffixNumber
4
5  $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
       AvailabilitySetId $avSet.Id
6
7  $cred=Get-Credential -Message "Type the name and password for VPX
       login."
8
9  $vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
       ComputerName $vmName -Credential $cred
10
11 $vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
       $publisher -Offer $offer -Skus $sku -Version $version
12
13 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.
       Id -Primary
14
15 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.
       Id
16
17 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.
       Id
18
19 $osDiskName=$vmName + "-" + $osDiskSuffix1
20
```



```
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/" + $osDiskName + ".vhd"

$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri $osVhdUri -CreateOption fromImage

Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -Name $sku

New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location $locName
```

VPX2 を作成します。

```
```
$suffixNumber=2

$vmName=$vmNamePrefix + $suffixNumber

$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avSet.Id

$cred=Get-Credential -Message "Type the name and password for VPX login."

$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -ComputerName $vmName -Credential $cred

$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName $publisher -Offer $offer -Skus $sku -Version $version

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic4.Id -Primary

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic5.Id

```

 
```

```
25
26 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic6.Id
27
28
29 $osDiskName=$vmName + "-" + $osDiskSuffix2
30
31
32 $osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     + $osDiskName + ".vhd"
33
34
35 $vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage
36
37
38 Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku
39
40
41 New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
    $locName
42 <!--NeedCopy--> ```
```

NIC に割り当てられたプライベート IP アドレスとパブリック IP アドレスを表示するには、次のコマンドを入力します。

```
1 ```
2 $nic1.IPConfig
3
4
5 $nic2.IPConfig
6
7
8 $nic3.IPConfig
9
10
11 $nic4.IPConfig
12
13
14 $nic5.IPConfig
15
16
17 $nic6.IPConfig
```



```
<!--NeedCopy--> ```
```

Azure の負荷分散 **(ALB)** を作成します。

この手順には、次の下位手順が含まれています。

- フロントエンド IP 構成を作成する
- ヘルスプローブの作成
- バックエンドアドレスプールの作成
- 負荷分散規則（HTTP および SSL）の作成
- フロントエンド IP 設定、バックエンドアドレスプール、および LB ルールを使用して ALB を作成します。
- IP 構成をバックエンドプールに関連付ける

```
$frontEndIP1=New-AzureRmLoadBalancerFrontendIpConfig -Name $frontEndConfigName1
 -PublicIpAddress $pip3

$healthProbe=New-AzureRmLoadBalancerProbeConfig -Name $healthProbeName
-Protocol Tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

$beAddressPool1=New-AzureRmLoadBalancerBackendAddressPoolConfig -Name
$backendPoolName1

$lbRule1=New-AzureRmLoadBalancerRuleConfig -Name $lbRuleName1 -FrontendIpConfigur
 $frontEndIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -
Protocol Tcp -FrontendPort 80 -BackendPort 80 -EnableFloatingIP

$lb=New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name $lbName -
Location $locName -FrontendIpConfiguration $frontEndIP1 -LoadBalancingRule
 $lbRule1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe

$nic2.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb.
BackendAddressPools[0])

$nic5.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb.
BackendAddressPools[0])

$lb=$lb |Set-AzureRmLoadBalancer

$nic2=$nic2 | Set-AzureRmNetworkInterface

$nic5=$nic5 | Set-AzureRmNetworkInterface
```

Citrix ADC VPX ペアを正常に展開したら、各 VPX インスタンスにログオンして HA-INC、SNIP アドレス、VIP アドレスを構成します。

1. 次のコマンドを入力して HA ノードを追加します。

```
add ha node 1 PeerNodeNSIP -inc Enabled
```


2. クライアント側 NIC のプライベート IP アドレスを VPX1（NIC2）および VPX2（NIC5）の SNIP として追加する

```
add nsip privateIPofNIC2 255.255.255.0 -type SNIP
add nsip privateIPofNIC5 255.255.255.0 -type SNIP
```

3. ALB のフロントエンド IP アドレス (パブリック IP) を持つプライマリノードに負荷分散仮想サーバーを追加します。

```
add lb virtual server v1 HTTP FrontEndIPofALB 80
```

関連リソース:

Azure でのアクティブスタンバイ HA デプロイメントでの GSLB の構成

Azure アクセラレーションネットワークを使用するように Citrix ADC VPX インスタンスを構成する

October 7, 2021

高速ネットワーキングにより、仮想マシンへのシングルルート I/O 仮想化 (SR-IOV) 仮想機能 (VF) NIC が有効になり、ネットワークのパフォーマンスが向上します。この機能は、信頼性の高いストリーミングと低い CPU 使用率でより高いスループットでデータを送受信する必要がある負荷の高いワークロードで使用できます。
NIC が高速ネットワーキングで有効になっている場合、Azure は NIC の既存のパラ仮想化 (PV) インターフェイスと SR-IOV VF インターフェイスをバンドルします。SR-IOV VF インターフェイスのサポートにより、Citrix ADC VPX インスタンスのスループットが有効になり、向上します。

高速ネットワーキングには、次の利点があります。

- 低レイテンシ
- 1 秒あたりのパケット数 (pps) のパフォーマンスが向上
- スループットの強化
- ジッタの低減
- CPU 使用率の低下

> 注
>
> Azure アクセラレーションネットワーキングは、リリース 13.0 ビルド 76.29 以降の Citrix ADC VPX インスタンスでサポートされています。

前提条件

- VM のサイズが Azure アクセラレーションネットワーキングの要件と一致していることを確認します。
- 任意の NIC で高速ネットワーキングを有効にする前に、VM（個別または可用性セット内）を停止します。

制限事項

高速ネットワーキングは、一部のインスタンスタイプでのみ有効にできます。詳細については、「サポートされるインスタンスタイプ」を参照してください。

高速ネットワーキングでサポートされる **NIC**

Azure は Mellanox ConnectX3 および ConnectX4 NIC を SR-IOV モードで高速ネットワーキングに提供している。

Citrix ADC VPX インターフェイスで高速ネットワークが有効になっている場合、Azure は ConnectX3 または ConnectX4 インターフェイスのいずれかを Citrix ADC VPX アプライアンスの既存の PV インターフェイスにバンドルします。

仮想マシンにインターフェイスをアタッチする前に高速ネットワークを有効にする方法の詳細については、「高速ネットワークを使用したネットワークインターフェイスの作成」を参照してください。

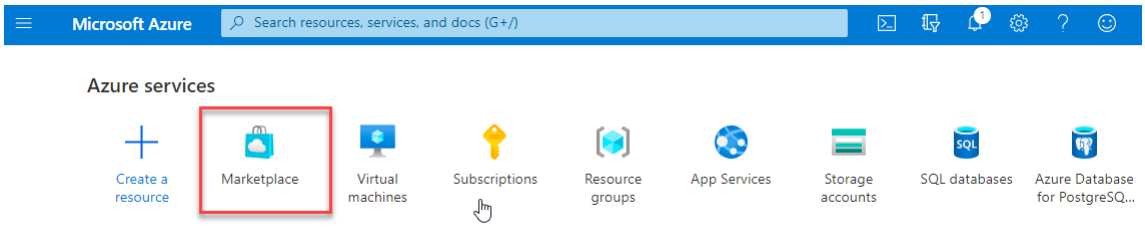
仮想マシンの既存のインターフェイスで高速ネットワーキングを有効にする方法の詳細については、「仮想マシンで既存のインターフェイスを有効にする」を参照してください。

Azure コンソールを使用して **Citrix ADC VPX** インスタンスで高速ネットワーキングを有効にする方法

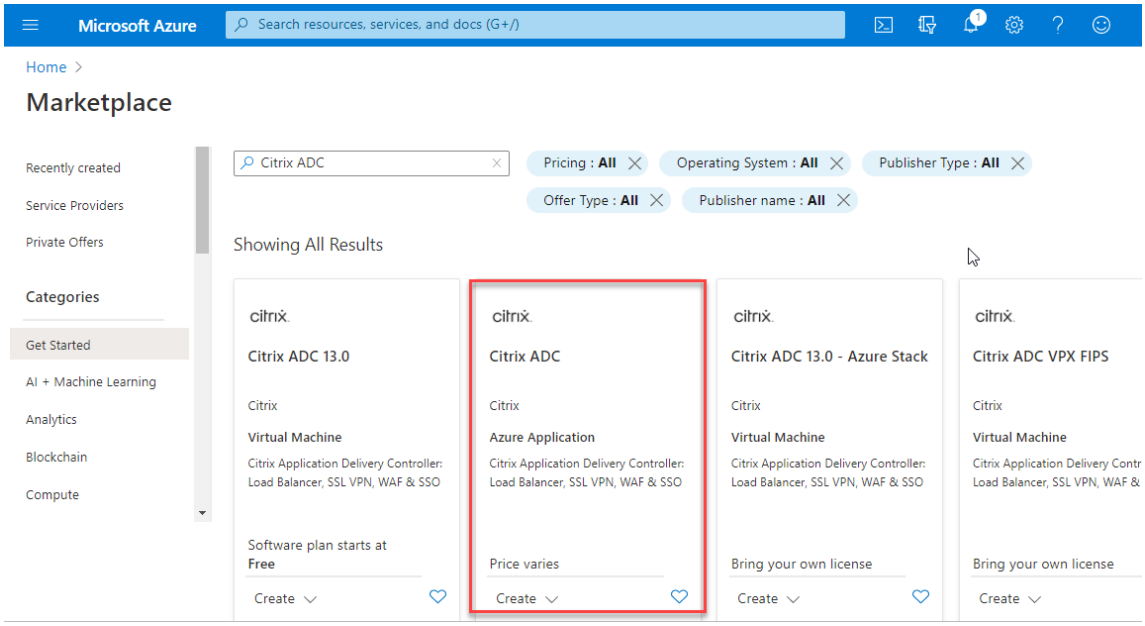
Azure コンソールまたは Azure PowerShell を使用して、特定のインターフェイスで高速ネットワークを有効にできます。

Azure のアベイラビリティセットまたはアベイラビリティーゾーンを使用して高速ネットワークを有効にするには、次の手順を実行します。

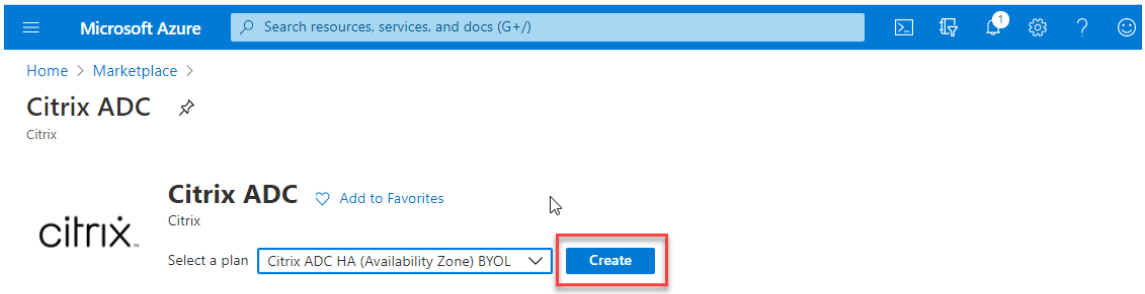
1. Azure ポータルにログインし、Azureマーケットプレイスにナビゲートします。



2. **Azure** マーケットプレイスから、**Citrix ADC** を検索します。

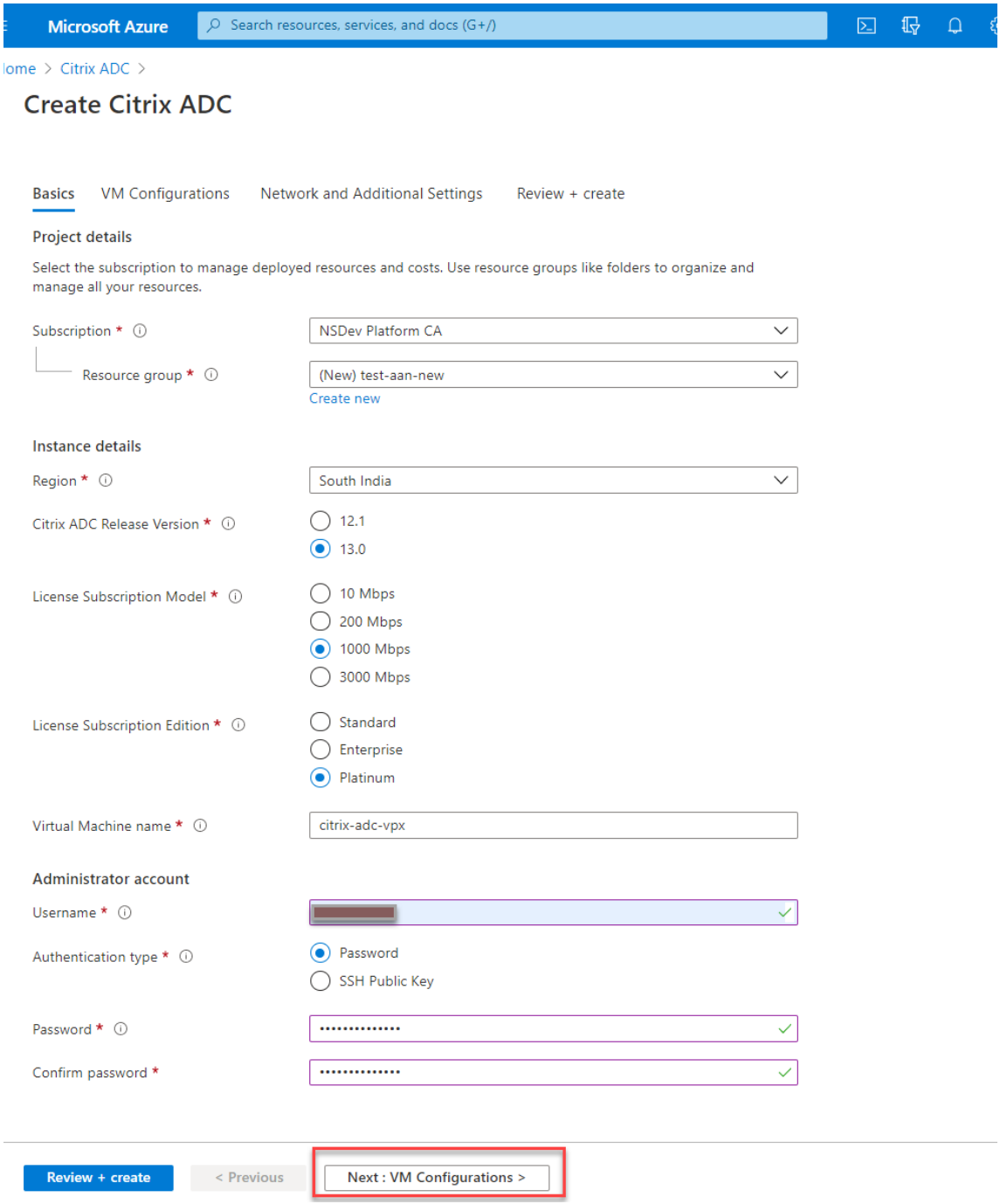


3. ライセンスとともに FIPS 以外の Citrix ADC プランを選択し、［作成］をクリックします。



［**Citrix ADC** の作成］ページが表示されます。

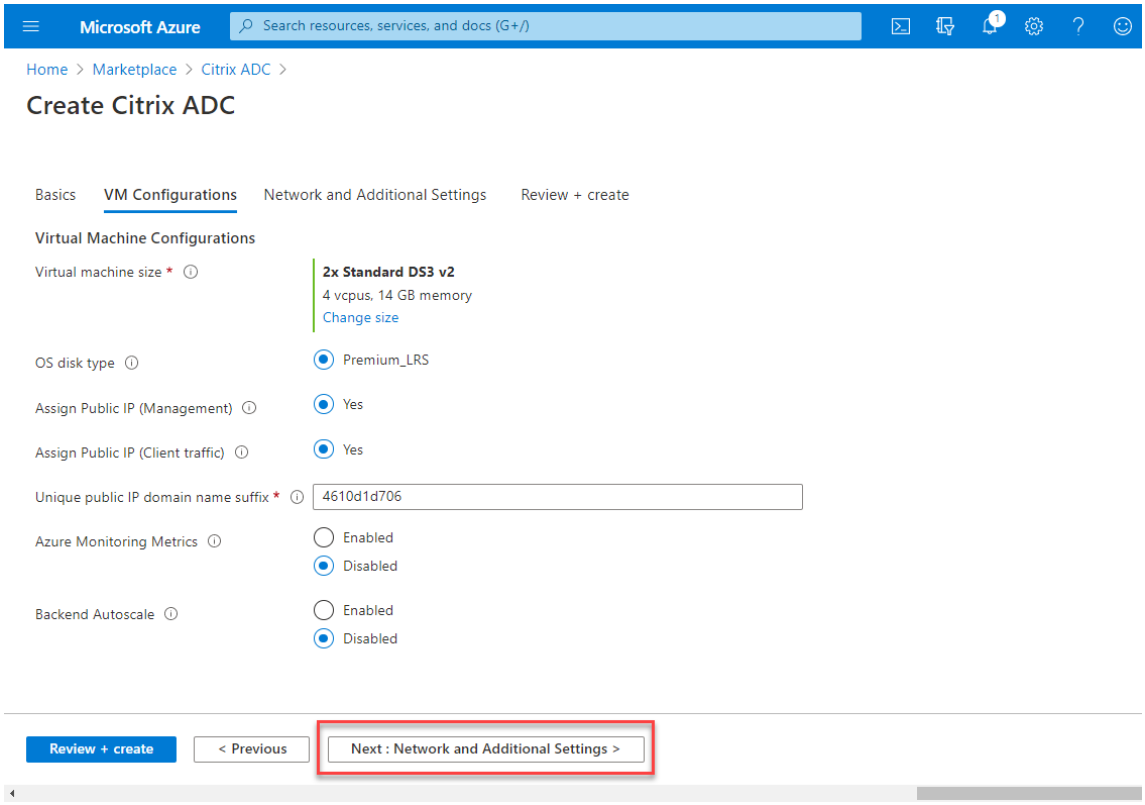
4. [基本] タブで、リソースグループを作成します。[パラメータ] タブで、地域、管理者ユーザー名、管理者パスワード、ライセンスタイプ（VM SKU）、およびその他のフィールドの詳細を入力します。



5. [次へ] をクリックします: VM 構成 > 。

   [**VM** 構成] ページで、次の手順を実行します。

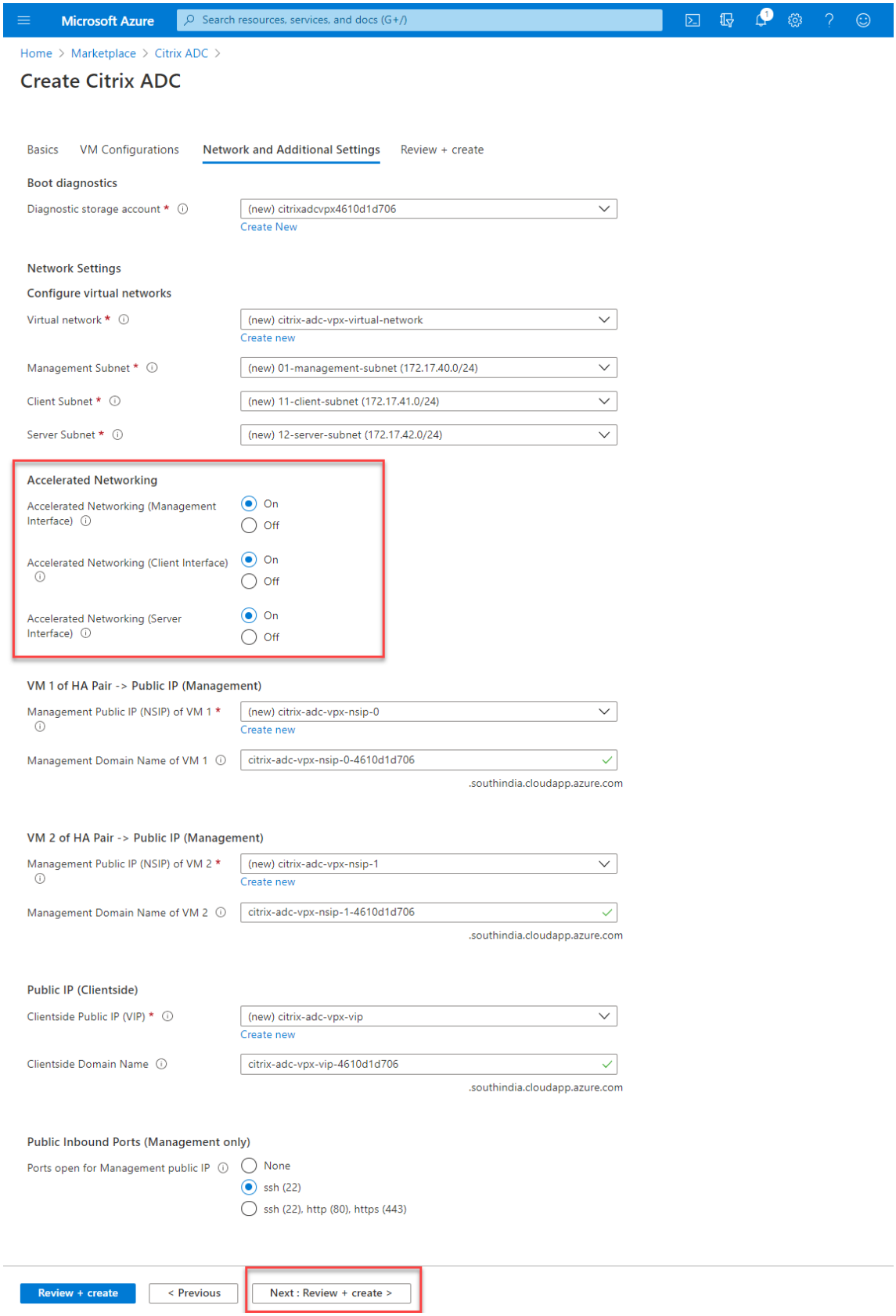
   a) パブリック IP ドメイン名のサフィックスを設定します。
   b) **Azure** モニタリングメトリクスを有効または無効にします。
   c) バックエンドオートスケールを有効または無効にします。



6. 次へ**:** ネットワークと追加設定をクリックします。

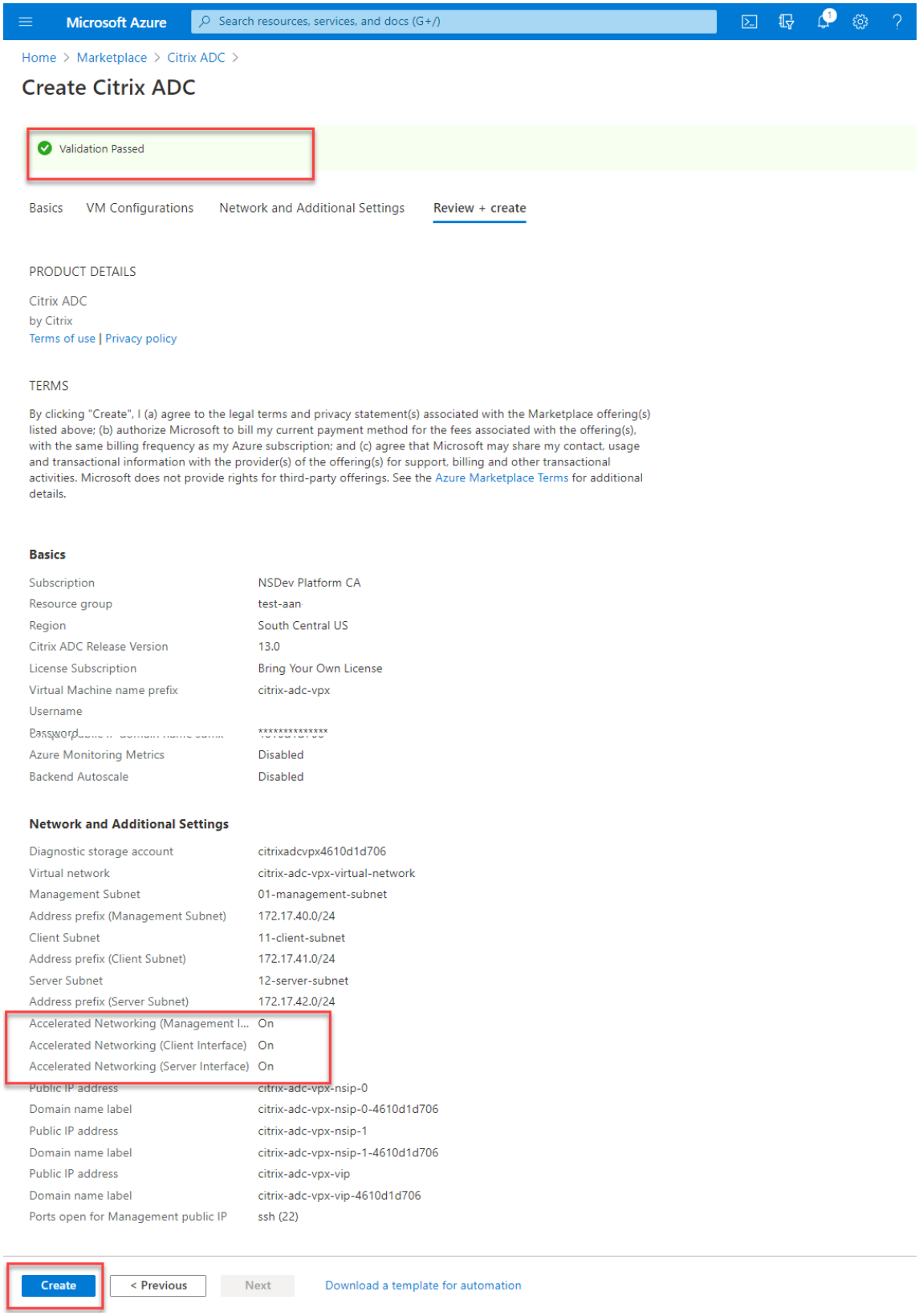
   [ネットワークとその他の設定**]** ページで、ブート診断アカウントを作成し、ネットワーク設定を構成します。

   [高速ネットワーキング] セクションには、管理インターフェイス、クライアントインターフェイス、およびサーバインターフェイスについて、アクセラレーションネットワーキングを個別に有効または無効にするオプションがあります。

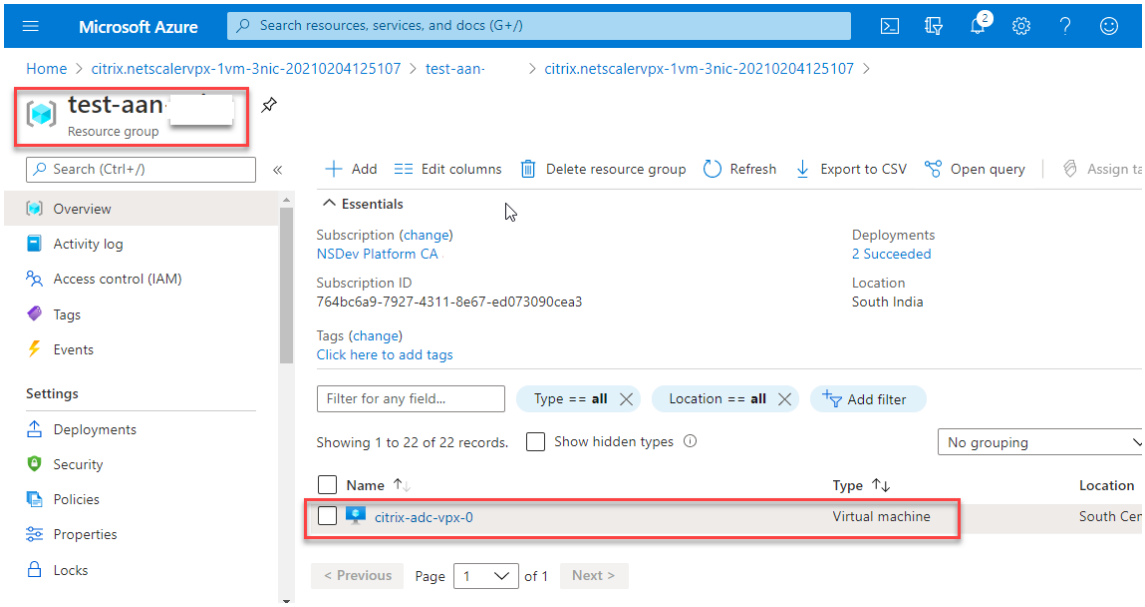


7. [次へ] をクリックします: レビュー + 作成する > 。

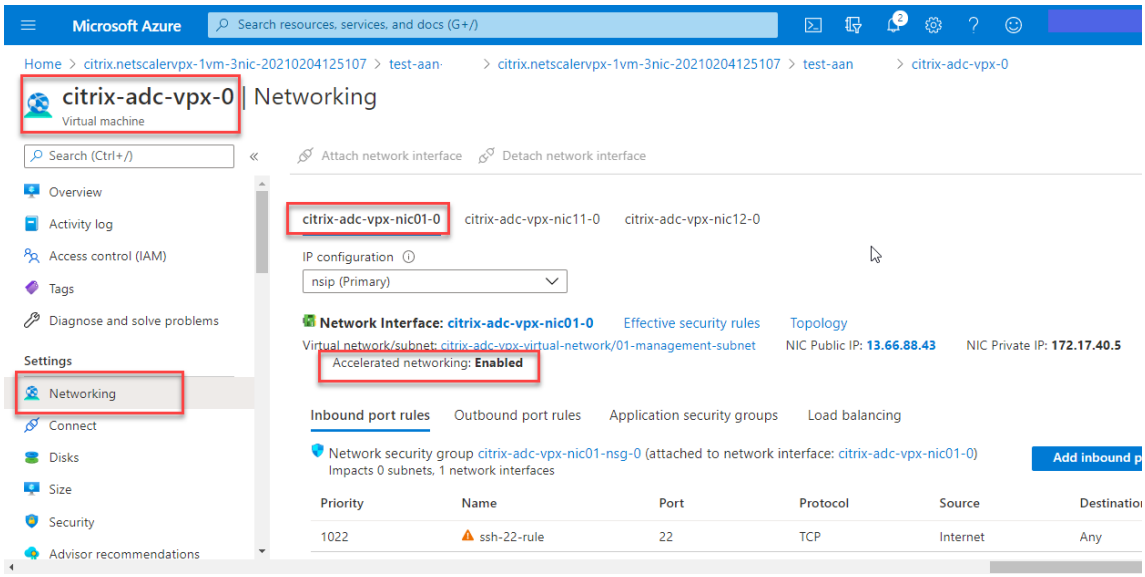
   検証が成功したら、基本設定、仮想マシンの構成、ネットワーク、および追加設定を確認し、[作成] をクリックします。Azure リソースグループが必要な構成で作成されるまでに時間がかかる場合があります。



8. デプロイが完了したら、リソースグループを選択して構成の詳細を表示します。



9. 高速ネットワーク構成を確認するには、[仮想マシン] **>** [ネットワーク] を選択します。アクセラレートネットワーキングのステータスは、NIC ごとに [有効] または [無効] と表示されます。



Azure PowerShell を使用して高速ネットワーキングを有効にする

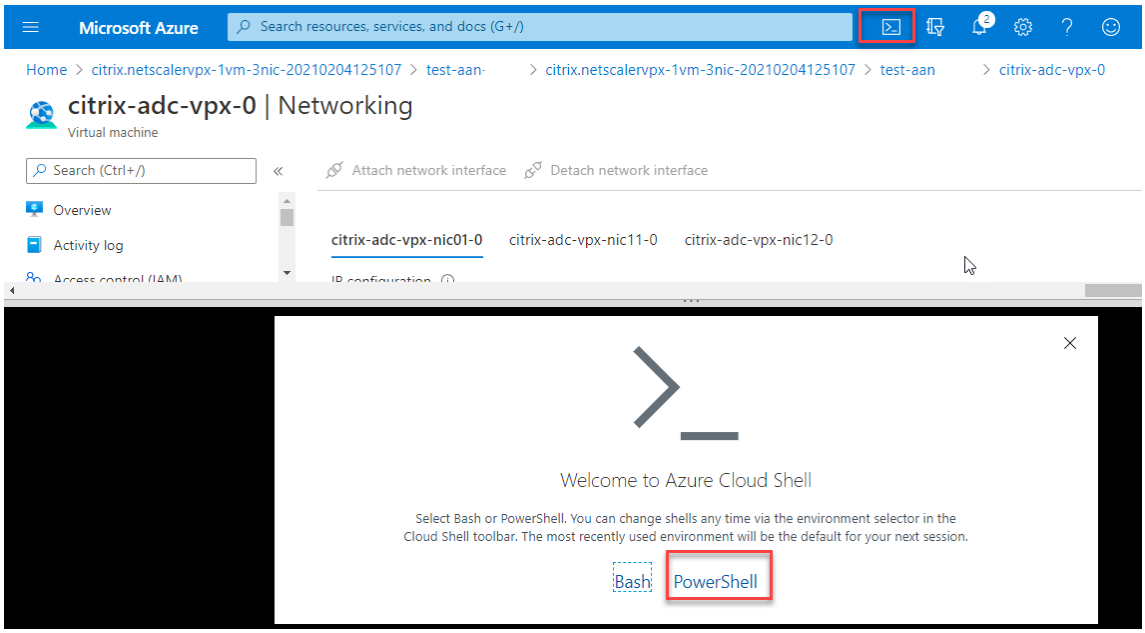
仮想マシンの作成後に高速ネットワークを有効にする必要がある場合は、Azure PowerShell を使用して有効化できます。

> 注:
>
> Azure PowerShell を使用した高速ネットワーキングを有効にする前に、仮想マシンを停止してください。

Azure PowerShell を使用して高速ネットワークを有効にするには、次の手順を実行します。

1. **Azure** ポータルに移動し、右上隅にある **PowerShell** アイコンをクリックします。

   > 注:
   >
   > Bash モードの場合は、PowerShell モードに切り替えます。



2. コマンドプロンプトで、次のコマンドを実行します。

```
az network nic update --name <nic-name> --accelerated-networking [true | false] --resource-group <resourcegroup-name>
<!--NeedCopy-->
```

アクセラレートネットワーキングパラメータは、次のいずれかの値を受け入れます。

- **True**: 指定した NIC で高速ネットワーキングを有効にします。
- **False**: 指定された NIC のアクセラレーションネットワーキングを無効にします。

特定の **NIC** で高速ネットワーキングを有効にするには、次の手順を実行します。

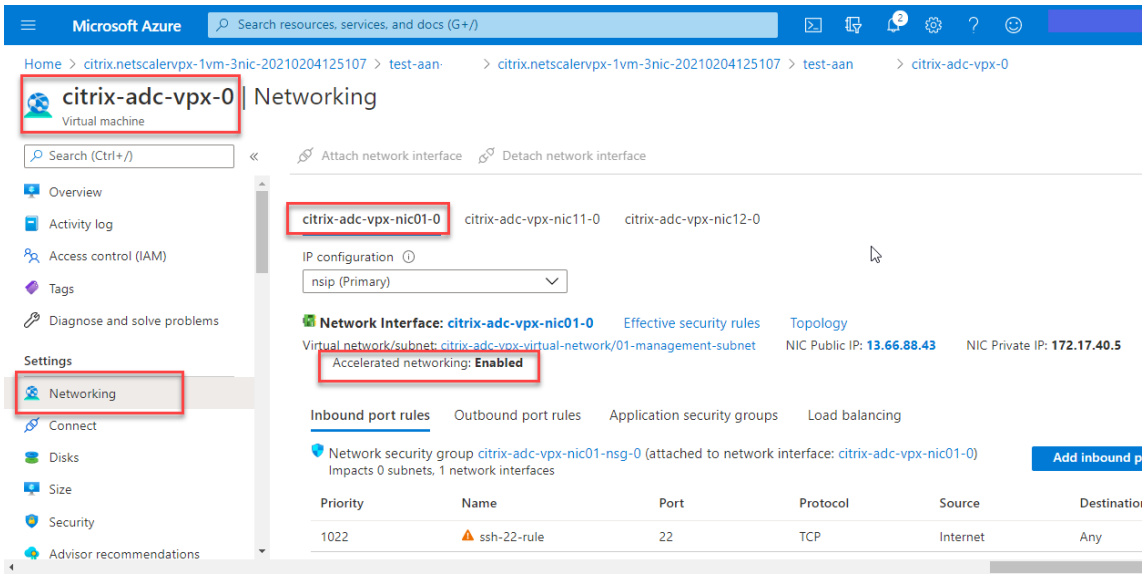
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking true --resource-group rsgp1-aan
<!--NeedCopy-->
```

特定の **NIC** で高速ネットワーキングを無効にするには、次の手順を実行します。

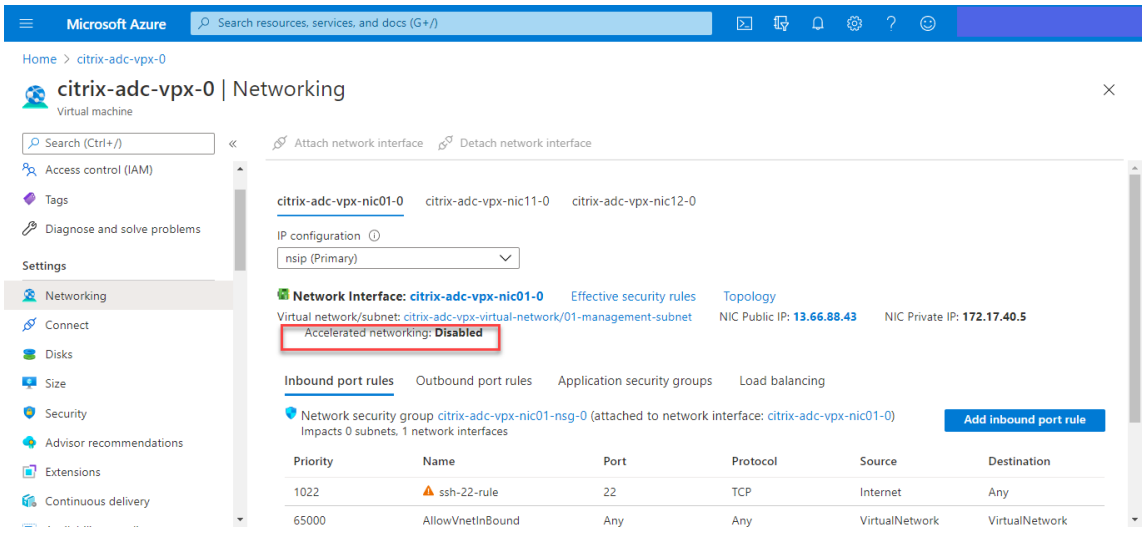
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking false --resource-group rsgp1-aan
<!--NeedCopy-->
```

3. デプロイの完了後に [アクセラレーテッドネットワーク] ステータスを確認するには、[**VM] > [**ネットワーク] に移動します。

   次の例では、アクセラレートネットワーキングが有効になっていることを確認します。



   次の例では、アクセラレートネットワーキングが無効になっていることがわかります。



Citrix ADC FreeBSD Shell を使用してインターフェイス上で高速ネットワークを検証するには

Citrix ADC FreeBSD シェルにログインし、次のコマンドを実行して高速ネットワークのステータスを確認できます。

ConnectX3 NIC の例**:**

次の例は、Mellanox ConnectX3 NIC の「ifconfig」コマンド出力を示しています。「50/n」は、Mellanox ConnectX3 NIC の VF インターフェイスを示します。0/1 と 1/1 は、Citrix ADC VPX インスタンスの PV インターフェイスを示します。PV インターフェイス（1/1）と CX3 VF インターフェイス（50/1）の両方が同じ MAC アドレス（00:22:48:1c: 99:3 e）を持つことがわかります。これは、2 つのインターフェイスが一緒にバンドルされていることを示します。

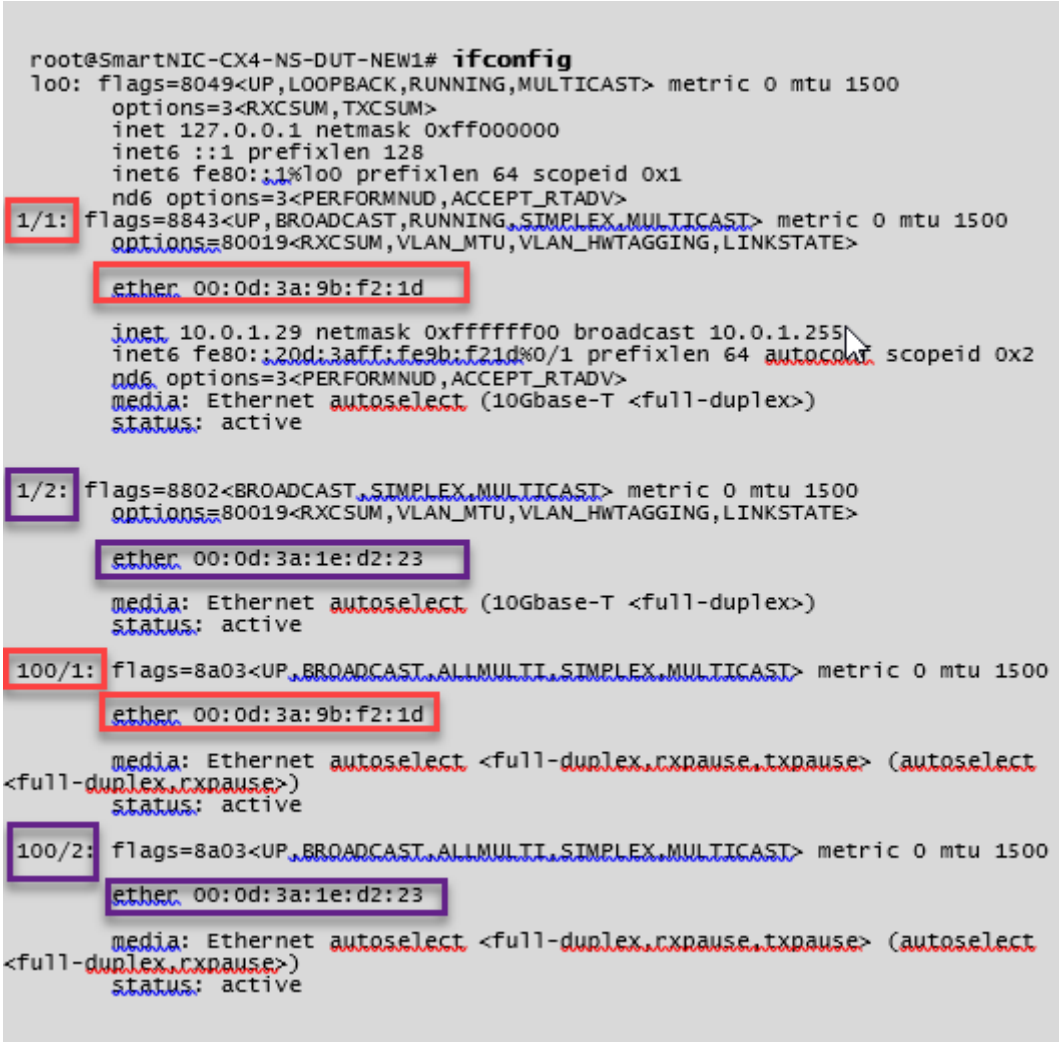
```
root@nvr-us-cx3# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
0/1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:0d:3a:98:71:be
        inet 172.16.27.11 netmask 0xffffff00 broadcast 172.16.27.255
        inet6 fe80::20d:3aff:fe98:71be%0/1 prefixlen 64 autoconf scopeid 0x2
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
1/1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
50/1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500

options=900b8<VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,VLAN_HWFILTER,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (<unknown subtype>)
        status: active
```

ConnectX4 NIC の例:

次の例は、Mellanox ConnectX4 NIC の「ifconfig」コマンド出力を示しています。「100/n」は、Mellanox ConnectX4 NIC の VF インターフェイスを示します。0/1、1/1、および 1/2 は、Citrix ADC VPX インスタンスの PV

インターフェイスを示します。
PV インターフェイス（1/1）と CX4 VF インターフェイス（100/1）の両方が同じ MAC アドレス（00:0 d: 3a: 9b: f 2:1 d）を持つことがわかります。これは、2 つのインターフェイスが一緒にバンドルされていることを示します。同様に、PV インターフェイス（1/2）と CX4 VF インターフェイス（100/2）は同じ MAC アドレス（00:0 d: 3a: 1:D 2:23）を持ちます。



ADC CLI を使用してインターフェイスで高速ネットワーキングを検証するには

ConnectX3 NIC の例**:**

次の show interface コマンドの出力は、PV インターフェイス 1/1 が SR-IOV VF NIC である仮想機能 50/1 にバンドルされていることを示しています。1/1 と 50/1 の NIC の両方の MAC アドレスは同じです。高速ネットワーキングが有効になると、1/1 インターフェイスのデータは、ConnectX3 インターフェイスである 50/1 インターフェイスのデータパスを介して送信されます。PV インターフェイス (1/1) の「show interface」出力が VF (50/1) を指すことがわかります。同様に、VF インターフェイス (50/1) の「show interface」出力は PV インターフェイス (1/1) を指します。


```
> show interface 1/1

        Interface 1/1 (NetScaler Virtual Interface, SmartNIC, VF 50/1 Datapath 50/1) #1
        flags=0xe060 <ENABLED, UP, UP, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m07s
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.

Done

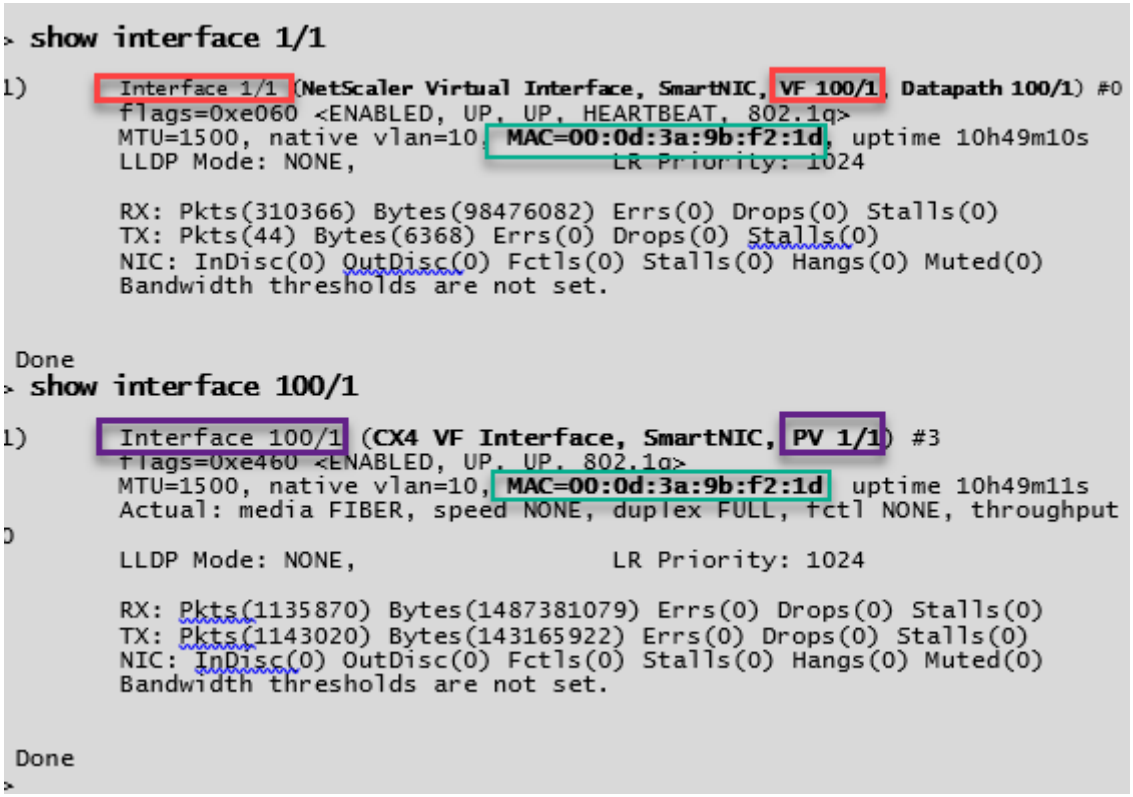
> show interface 50/1

      Interface 50/1 (CX3 VF Interface, SmartNIC, PV 1/1) #2
        flags=0xe460 <ENABLED, UP, UP, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m08s
        Actual: media NONE, speed 50000, duplex FULL, fctl NONE, throughput 50000
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
```

ConnectX4 NIC の例**:**

次の show interface コマンドの出力は、PV インターフェイス 1/1 が SR-IOV VF NIC である仮想機能 100/1 にバンドルされていることを示しています。1/1 と 100/1 の NIC の両方の MAC アドレスは同じです。高速ネットワーキングが有効になると、1/1 インターフェイスのデータは、ConnectX4 インターフェイスである 100/1 インターフェイスのデータパスを介して送信されます。PV インターフェイス (1/1) の「show interface」出力が VF (100/1) を指すことがわかります。同様に、VF インターフェイス (100/1) の「show interface」出力は PV インターフェイス (1/1) を指します。



Citrix ADC での注意点

- PV インターフェイスは、必要なすべての操作のプライマリインターフェイスまたはメインインターフェイスと見なされます。設定は PV インターフェイスでのみ実行する必要があります。
- VF インターフェイスでのすべての「set」操作は、以下を除いてブロックされます。
  - インターフェイスを有効にする
  - インターフェイスを無効にする
  - インターフェイスをリセット
  - 統計をクリアする

  > 注:
  >
  > VF インターフェイスでは操作を実行しないことをお勧めします。
- `show interface`コマンドを使用して、PV インターフェイスと VF インターフェイスとのバインディングを確認できます。

PV インターフェースへの VLAN の構成

PV インターフェイスが VLAN にバインドされている場合、関連するアクセラレーション VF インターフェイスも PV インターフェイスと同じ VLAN にバインドされます。この例では、PV インターフェイス（1/1）は VLAN（20）にバ

インドされています。PV インターフェイス（1/1）にバンドルされている VF インターフェイス（100/1）も VLAN 20 にバインドされます。

例**:**

1. VLAN を作成します。

```
add vlan 20
<!--NeedCopy-->
```

2. VLAN を PV インターフェースにバインドします。

```
bind vlan 20  - ifnum 1/1

show vlan

1)  VLAN ID: 1
    Link-local IPv6 addr: fe80::20d:3aff:fe9b:f21d/64
    Interfaces : LO/1

2)  VLAN ID: 10     VLAN Alias Name:
    Interfaces : 0/1 100/1
    IPs : 10.0.1.29  Mask: 255.255.255.0

3)  VLAN ID: 20     VLAN Alias Name:
    Interfaces :  1/1 100/2

<!--NeedCopy-->
```

> 注
>
> アクセラレーションされた VF インターフェイスでは VLAN バインディング操作は許可されていません。

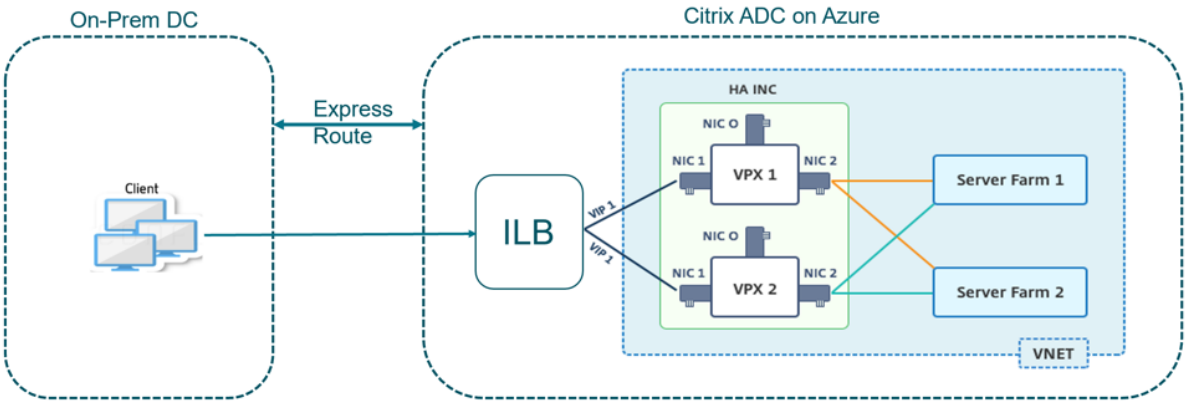
```
bind vlan 1 -ifnum 100/1
ERROR: Operation not permitted
<!--NeedCopy-->
```

Azure ILB で **Citrix** 高可用性テンプレートを使用して **HA-INC** ノードを構成する

January 25, 2022

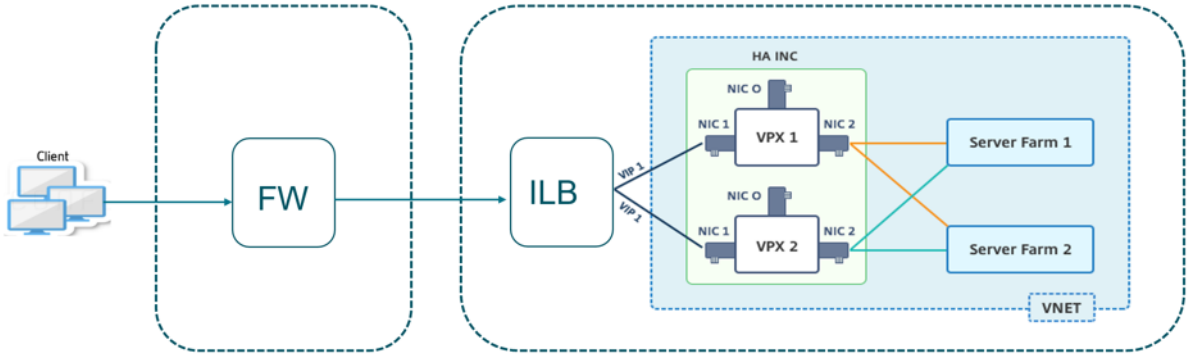
イントラネットアプリケーション用の標準テンプレートを使用すると、HA-INC モードで一対の VPX インスタンスを迅速かつ効率的にデプロイできます。Azure 内部ロードバランサー (ILB) は、図 1 に示すように、フロントエンドに内部 IP アドレスまたはプライベート IP アドレスを使用します。このテンプレートでは、3 つのサブネットと 6 つの NIC を持つ 2 つのノードが作成されます。サブネットは、管理、クライアント、およびサーバー側のトラフィック用で、各サブネットはデバイスごとに異なる NIC に属します。

図 1: 内部ネットワーク内のクライアント用の Citrix ADC HA ペア



この展開は、図 2 に示すように、Citrix ADC HA ペアがファイアウォールの内側にある場合にも使用できます。パブリック IP アドレスはファイアウォールに属し、ILB のフロントエンド IP アドレスに NAT されます。

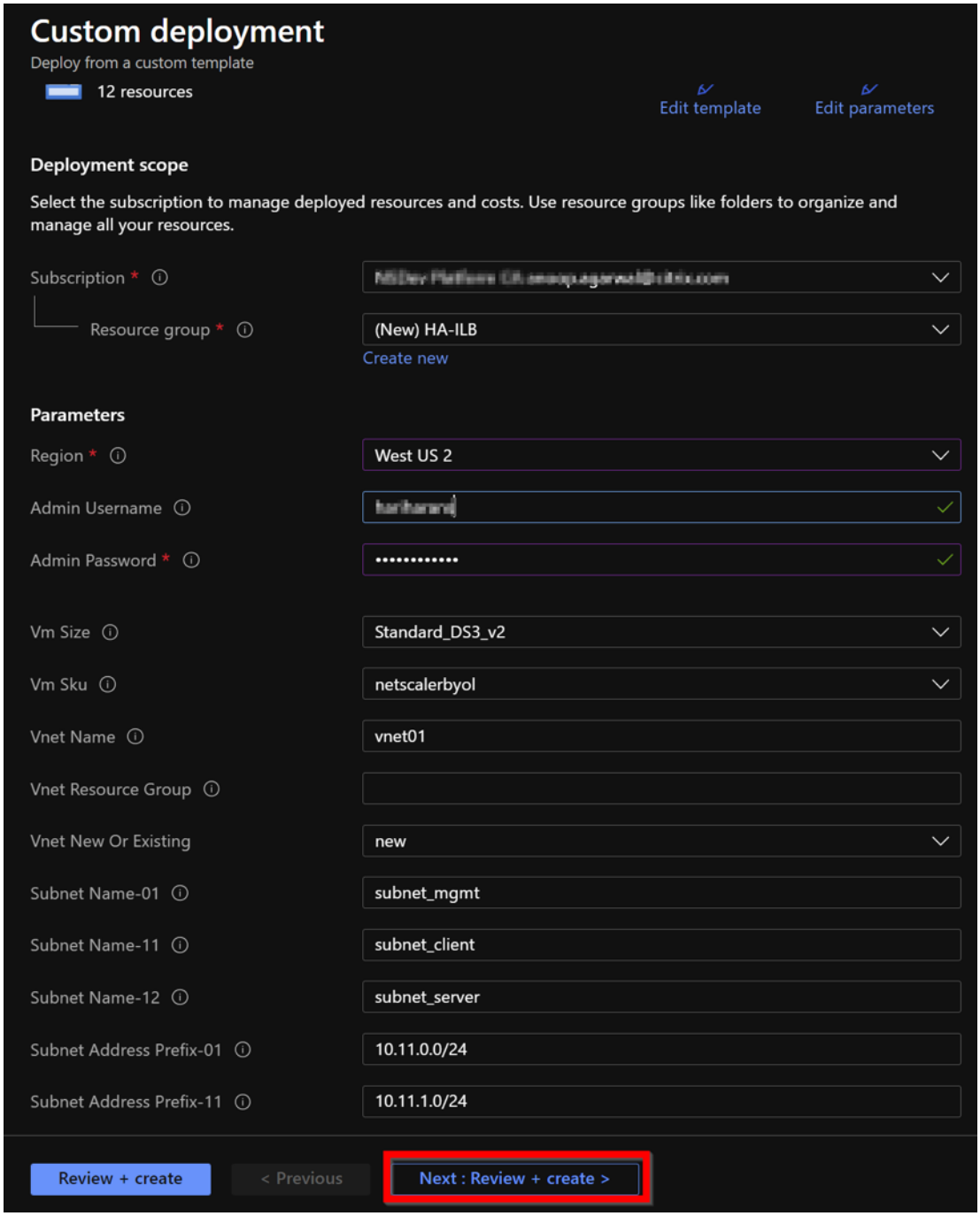
図 2: パブリック IP アドレスを持つファイアウォールと Citrix ADC HA のペア



イントラネットアプリケーションの Citrix ADC HA ペアテンプレートは、Azure ポータルで入手できます。

次の手順を実行してテンプレートを起動し、Azure 可用性セットを使用して高可用性 VPX ペアをデプロイします。

1. Azure Portal から、[カスタム展開] ページに移動します。
2. [基本] ページが表示されます。リソースグループを作成します。[パラメータ] タブで、[リージョン]、[管理者ユーザー名]、[管理者パスワード]、[ライセンスタイプ] (VM sku)、およびその他のフィールドの詳細を入力します。

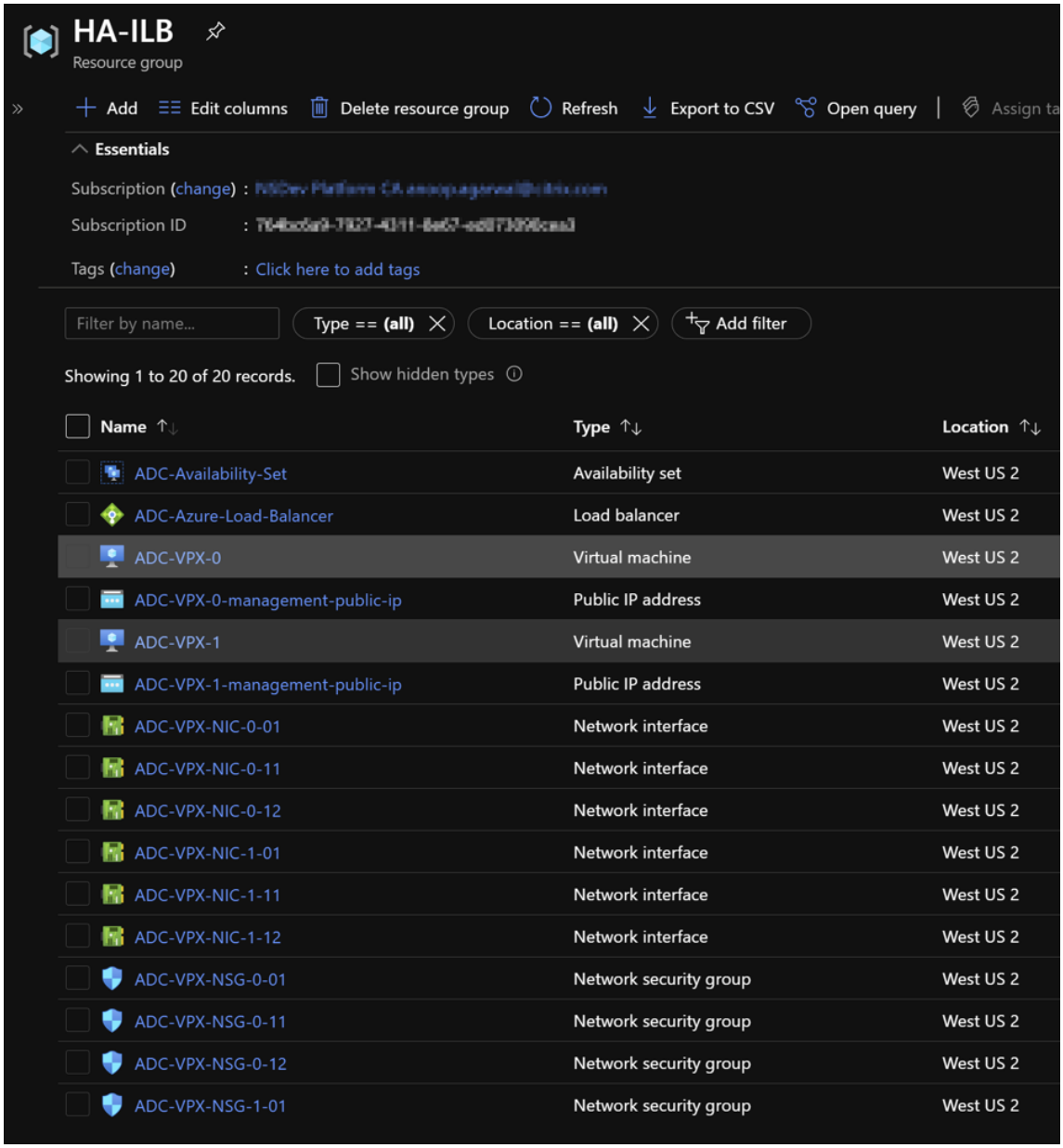


3. [** 次へ: レビュー + 作成 **] をクリックします。

   必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了後、Azure ポータルでリソースグループを選択し、LB ルール、バックエンドプール、ヘルスプローブなどの構成の詳細を表示します。高可用性ペアは ADC-VPX-0 と ADC-VPX-1 として表示されます。

   追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。

必要な構成が完了すると、次のリソースが作成されます。



4. 次の構成を検証するには、**ADC-VPX-0** ノードと **ADC-VPX-1** ノードにログオンする必要があります。

   - 両方のノードの NSIP アドレスは管理サブネットに存在する必要があります。
   - プライマリ（ADC-VPX-0）ノードとセカンダリ（ADC-VPX-1）ノードには、2 つの SNIP アドレスが表示される必要があります。一方の SNIP（クライアントサブネット）は ILB プローブへの応答に使用され、もう 1 つの SNIP（サーバーサブネット）はバックエンドサーバー通信に使用されます。

   > 注
   >
   > HA-INC モードでは、ADC-VPX-0VM と ADC-VPX-1VM の SNIP アドレスは、両方が同じである従来のオンプレミス ADC HA 展開とは異なり、同じサブネット内では異なります。

> VPX ペア SNIP が異なるサブネットにある場合、または VIP が SNIP と同じサブネット内にない場合に展開をサポートするには、Mac ベース転送（MBF）を有効にするか、各 VIP の静的ホストルートを各 VPX ノードに追加する必要があります。

プライマリノード（ADC-VPX-0）

```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.5        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.4        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.5 (ADC-VPX-0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:20:26 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.4
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

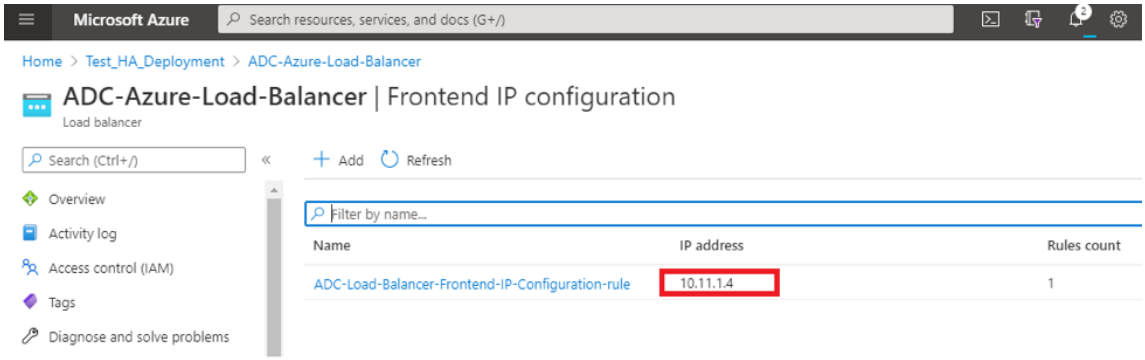
セカンダリノード（ADC-VPX-1）

```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.4        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.6        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

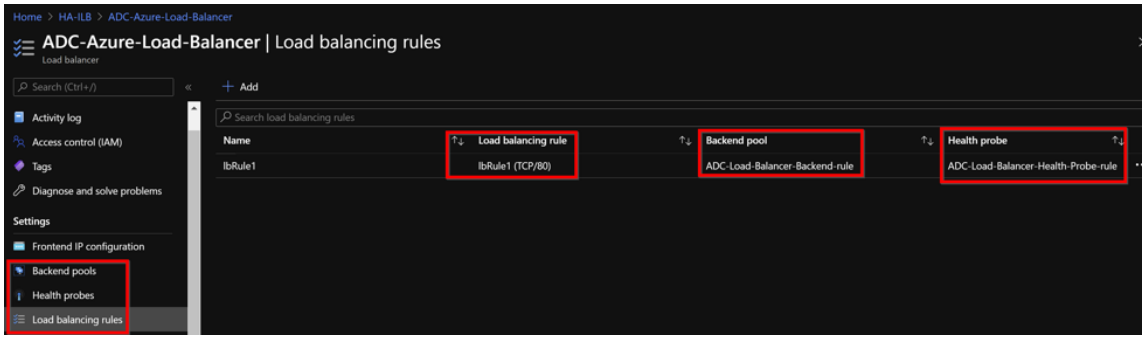


```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.4 (ADC-VPX-1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:24:18 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.5
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

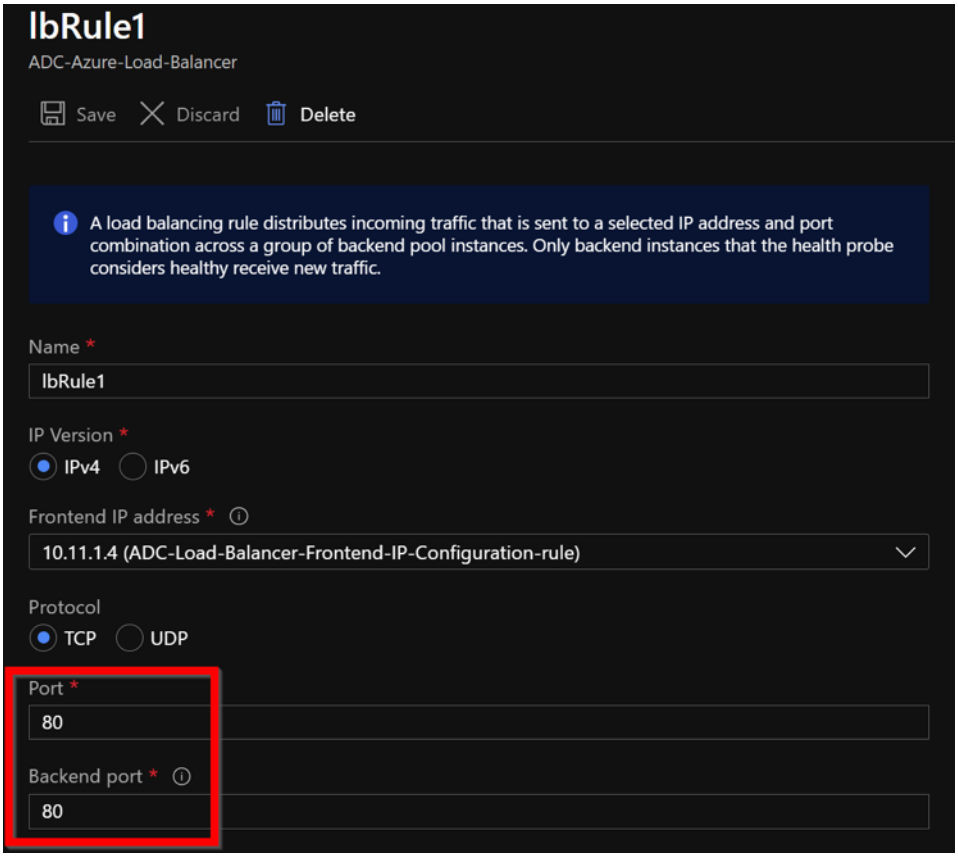
5. プライマリノードとセカンダリノードが UP で、同期ステータスが **SUCCESS** になったら、プライマリノード (ADC-VPX-0) の負荷分散仮想サーバーまたはゲートウェイ仮想サーバーを、ADC Azure ロードバランサーのプライベートフローティング IP (FIP) アドレスで構成する必要があります。詳細については、「サンプル設定」セクションを参照してください。
6. ADC Azure 負荷分散サーバーのプライベート IP アドレスを見つけるには、**Azure portal > ADC Azure Load Balancer > Frontend IP configuration** に移動します。



7. **Azure Load Balancer** の設定ページでは、ARM テンプレートのデプロイが LB ルール、バックエンドプール、およびヘルスプローブの作成に役立ちます。

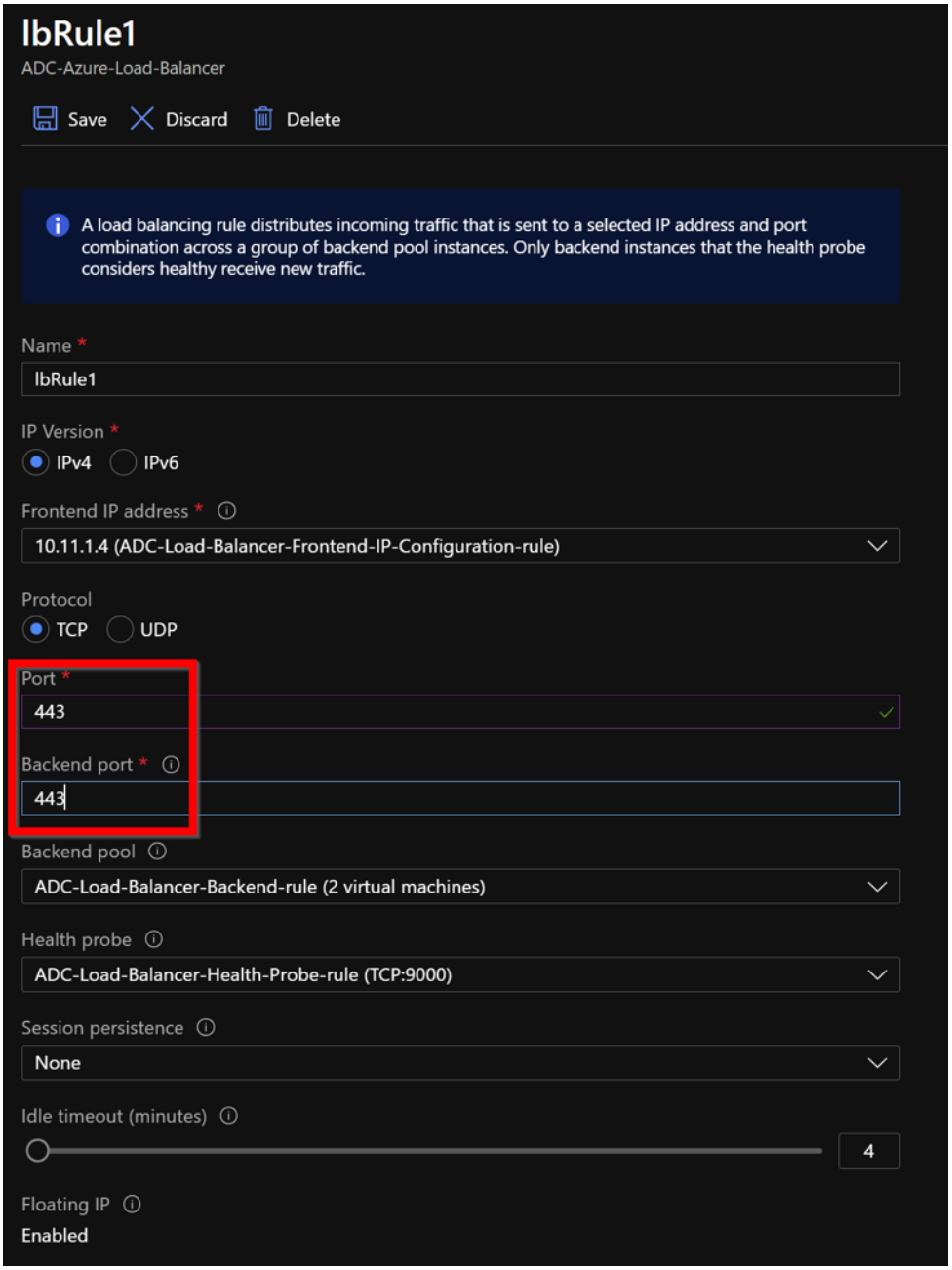


   - LB ルール (lbrule1) はデフォルトでポート 80 を使用します。



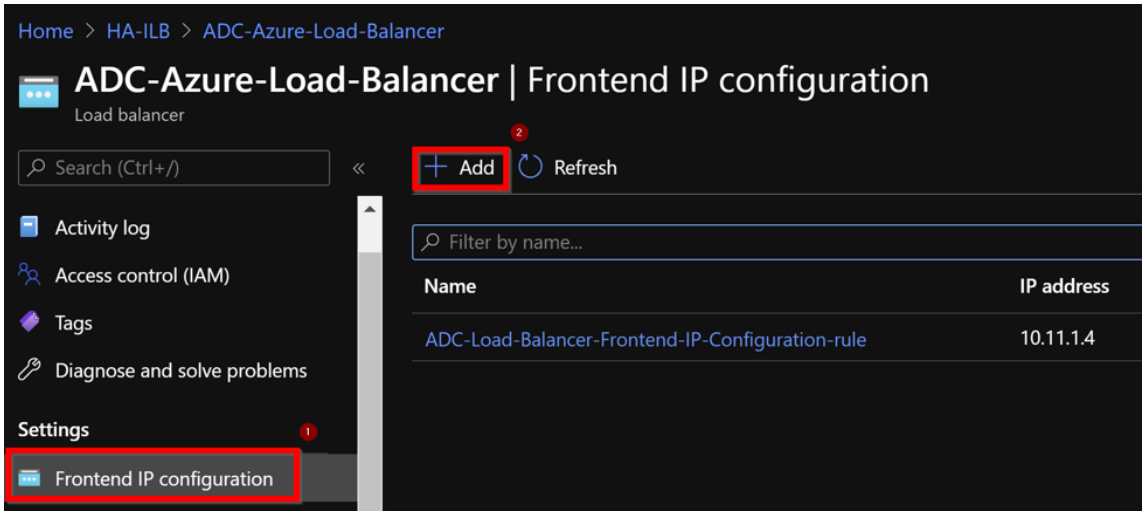
- ポート 443 を使用するようにルールを編集し、変更を保存します。

  > 注
  >
  > セキュリティを強化するため、LB 仮想サーバーまたはゲートウェイ仮想サーバーには SSL ポート 443 を使用することをお勧めします。

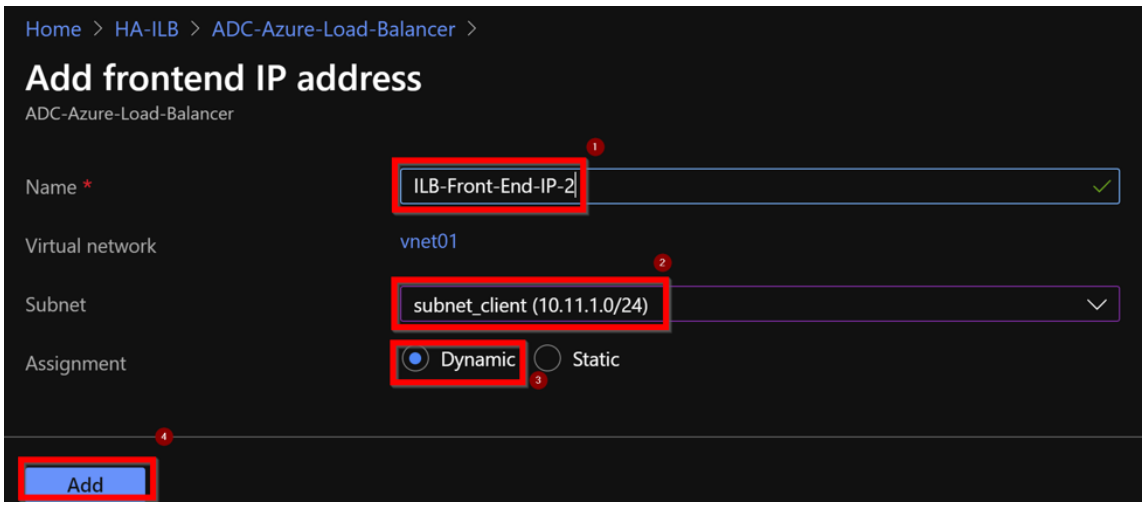


ADC に VIP アドレスを追加するには、次の手順に従います。

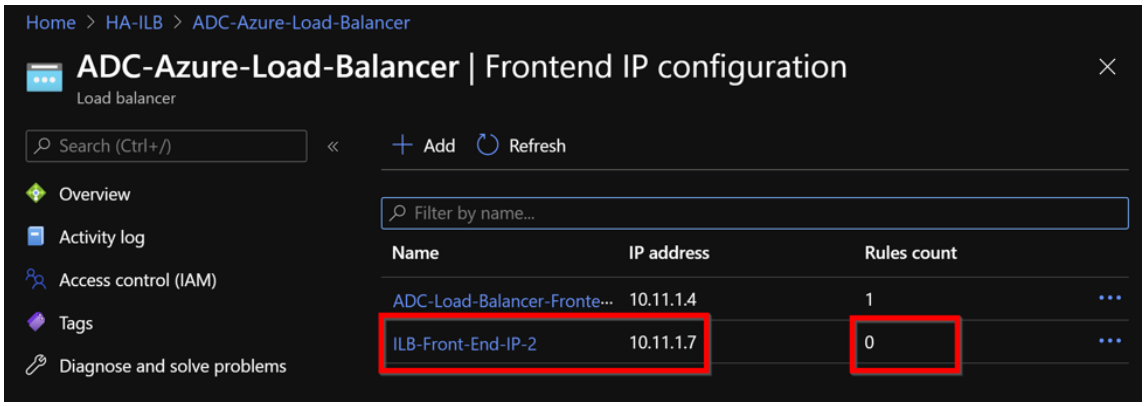
1. **Azure Load Balancer > Frontend IP** 構成に移動し、[追加] をクリックして新しい内部ロードバランサー IP アドレスを作成します。



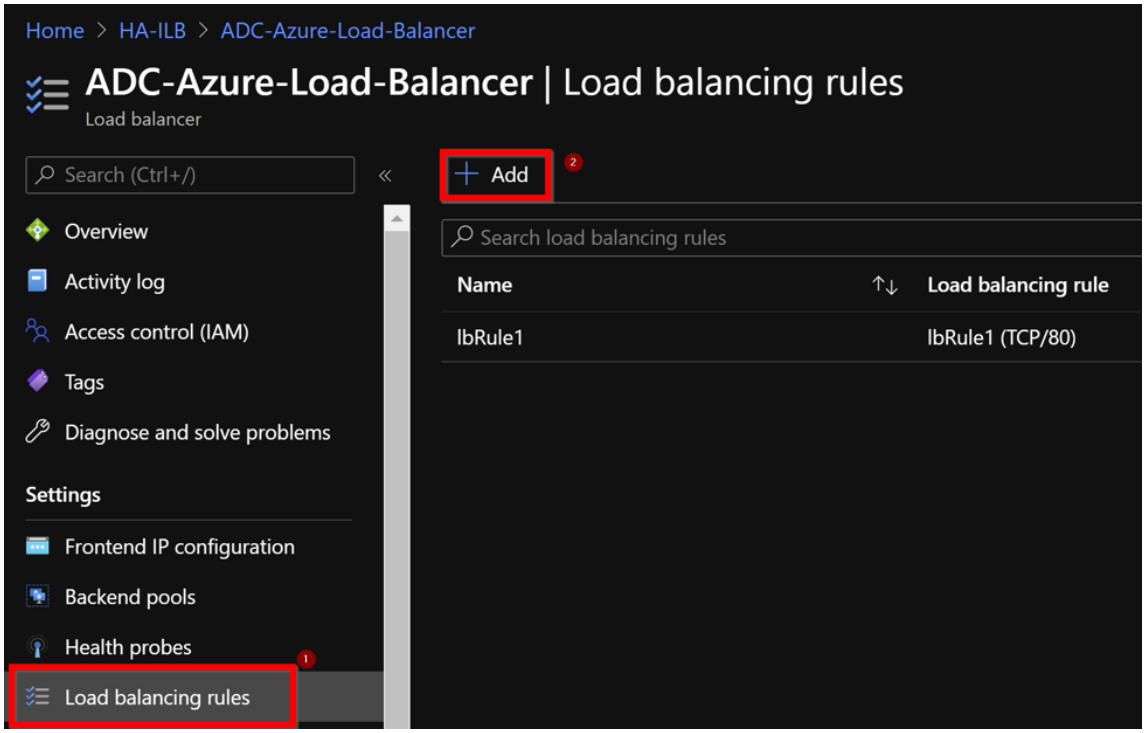
2. **[Add frontend IP address**] ページで、名前を入力し、クライアントサブネットを選択し、動的 IP アドレスまたは静的 IP アドレスを割り当てて、[**Add**] をクリックします。



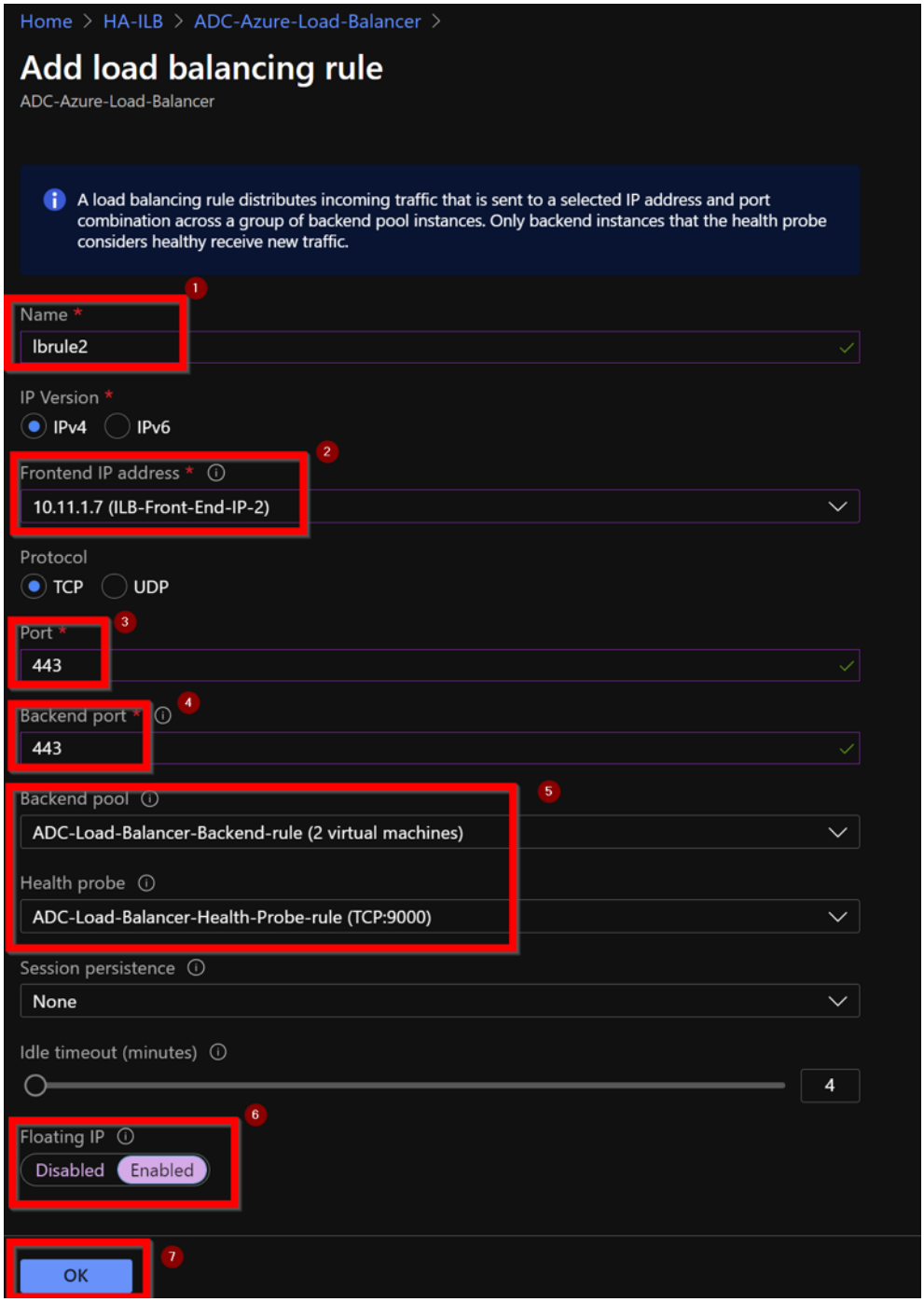
3. フロントエンド IP アドレスは作成されますが、LB ルールは関連付けられていません。新しい負荷分散ルールを作成し、フロントエンド IP アドレスに関連付けます。



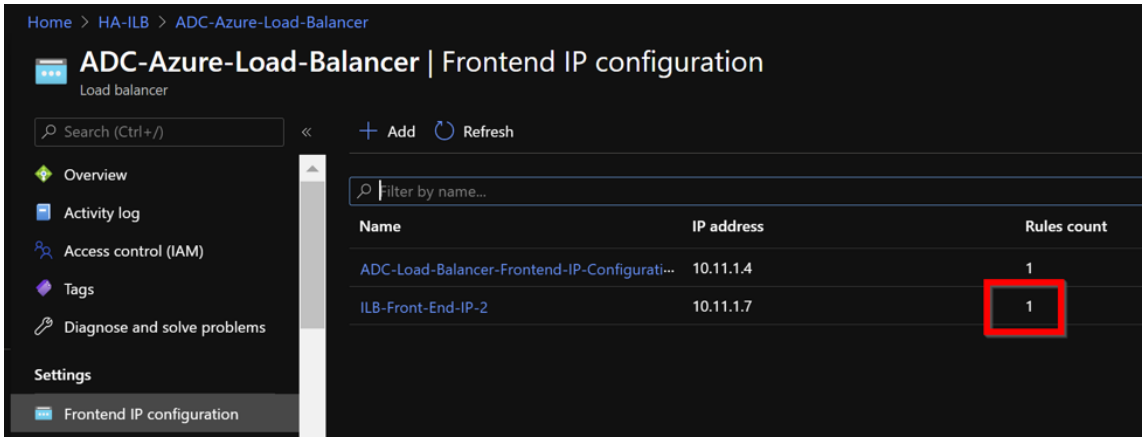
4. [**Azure** ロードバランサー] ページで、[負荷分散ルール] を選択し、[追加] をクリックします。



5. 新しいフロントエンド IP アドレスとポートを選択して、新しい LB ルールを作成します。[フローティング **IP**] フィールドは [有効] に設定する必要があります。



6. これで、フロントエンド **IP** 設定に、適用されている LB ルールが表示されます。



設定例

ゲートウェイ VPN 仮想サーバーと負荷分散仮想サーバーを設定するには、プライマリノード (ADC-VPX-0) で次のコマンドを実行します。設定はセカンダリノード（ADC-VPX-1）に自動的に同期されます。

ゲートウェイのサンプル構成

```
enable feature aaa LB SSL SSLVPN
enable ns mode MBF
add vpn vserver vpn_ssl SSL 10.11.1.4 443
add ssl certKey ckp -cert wild-cgwsanity.cer -key wild-cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

負荷分散のサンプル構成

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 10.11.1.7 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

ILB の内部 IP アドレスに関連付けられている完全修飾ドメイン名 (FQDN) を使用して、負荷分散または VPN 仮想サーバーにアクセスできるようになりました。

負荷分散仮想サーバーの構成方法の詳細については、「**Resources**」セクションを参照してください。

リソース:

次のリンクには、HA の導入と仮想サーバの設定に関する追加情報が表示されます。

- 異なるサブネットに高可用性ノードを構成する
- 基本的な負荷分散の設定

関連リソース:

- PowerShell コマンドを使用して、複数の IP アドレスと NIC を使用して高可用性セットアップを構成する
- Azure でのアクティブスタンバイ HA デプロイメントでの GSLB の構成

インターネット向けアプリケーション用の **Citrix** 高可用性テンプレートを使用して **HA-INC** ノードを構成します

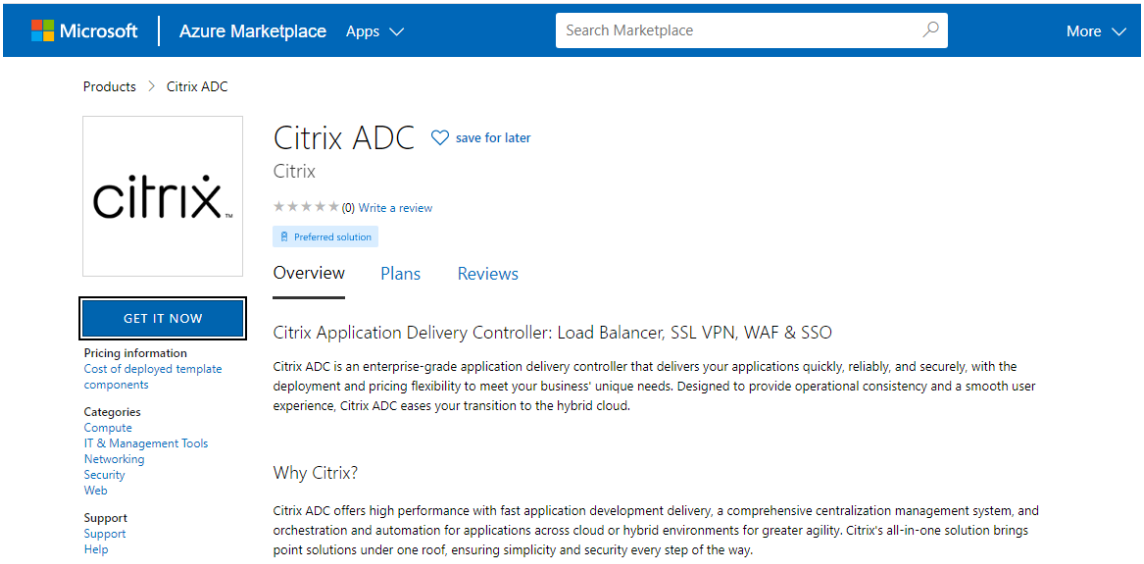
December 7, 2021

インターネット向けアプリケーションの標準テンプレートを使用すると、VPX インスタンスのペアを HA-INC モードで迅速かつ効率的に展開できます。Azure ロードバランサー（ALB）は、フロントエンドにパブリック IP アドレスを使用します。テンプレートは、3 つのサブネットと 6 つの NIC を持つ 2 つのノードを作成します。サブネットは、管理、クライアント、およびサーバー側のトラフィック用です。各サブネットには、両方の VPX インスタンス用に 2 つの NIC があります。

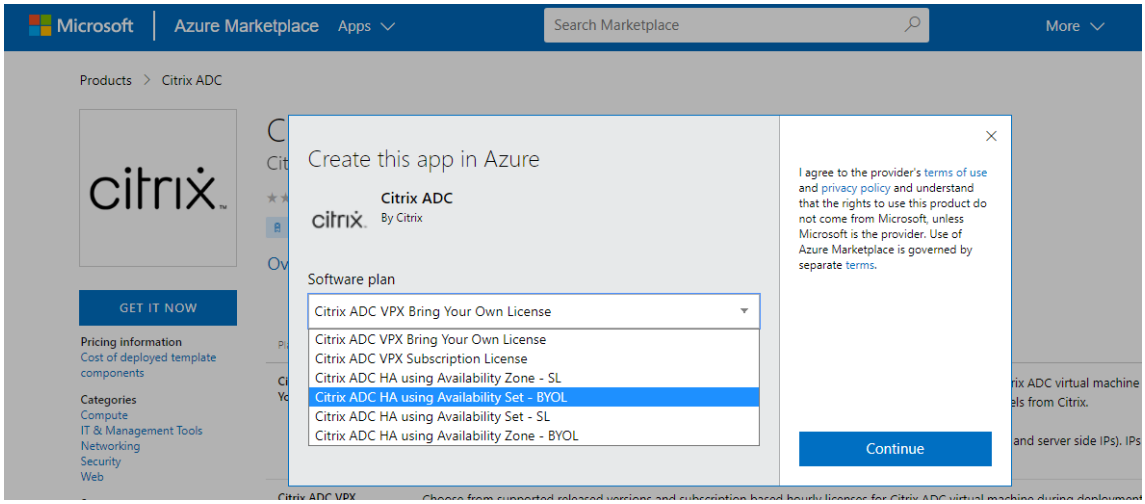
インターネット向けアプリケーションの Citrix ADC HA ペアテンプレートは、Azure Marketplace で入手できます。

次の手順を実行してテンプレートを起動し、Azure 可用性セットまたは可用性ゾーンを使用して高可用性 VPX ペアをデプロイします。

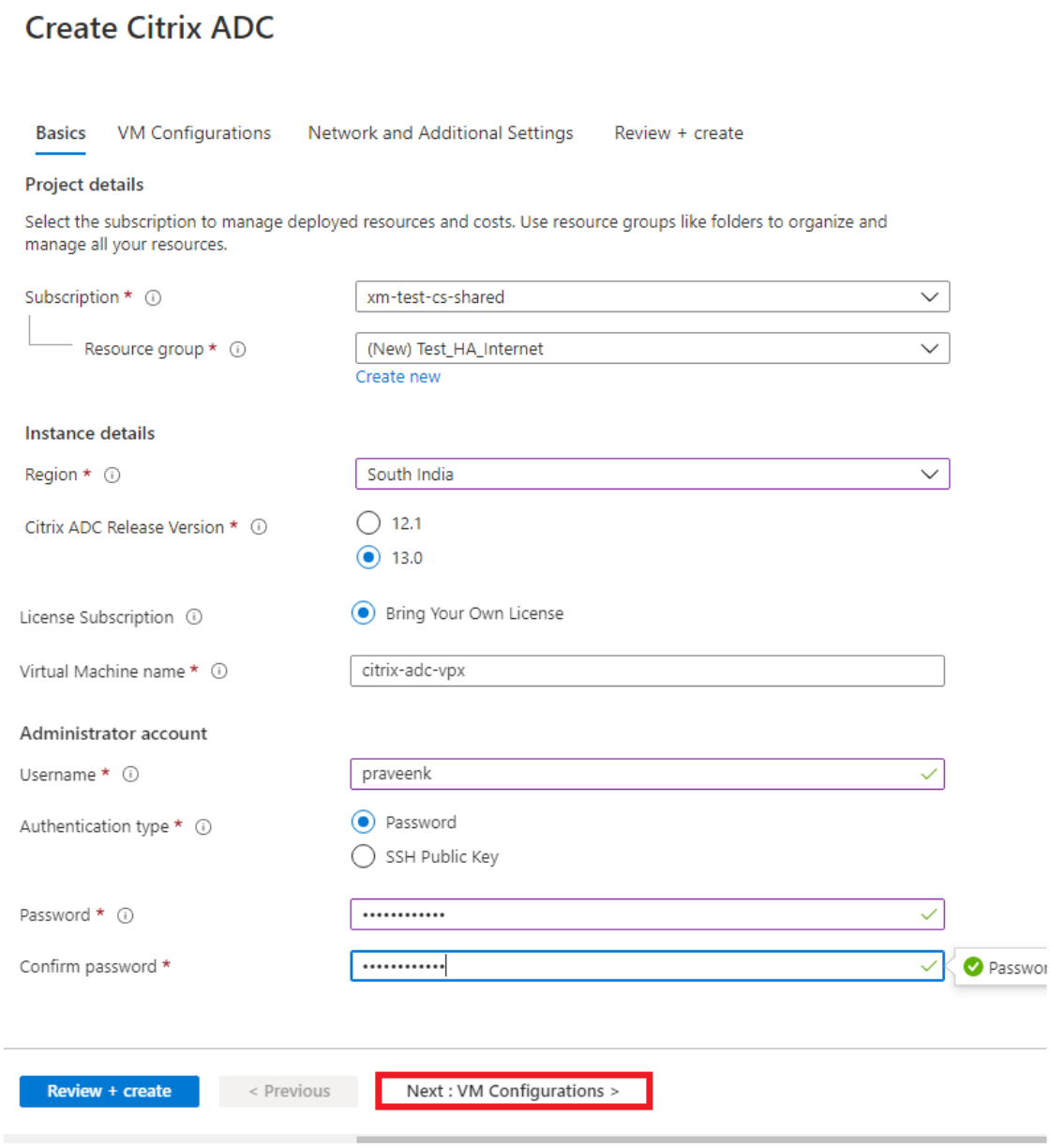
1. Azure Marketplace から、Citrix ADC を検索し ます。
2. [今すぐ入手] をクリックします。



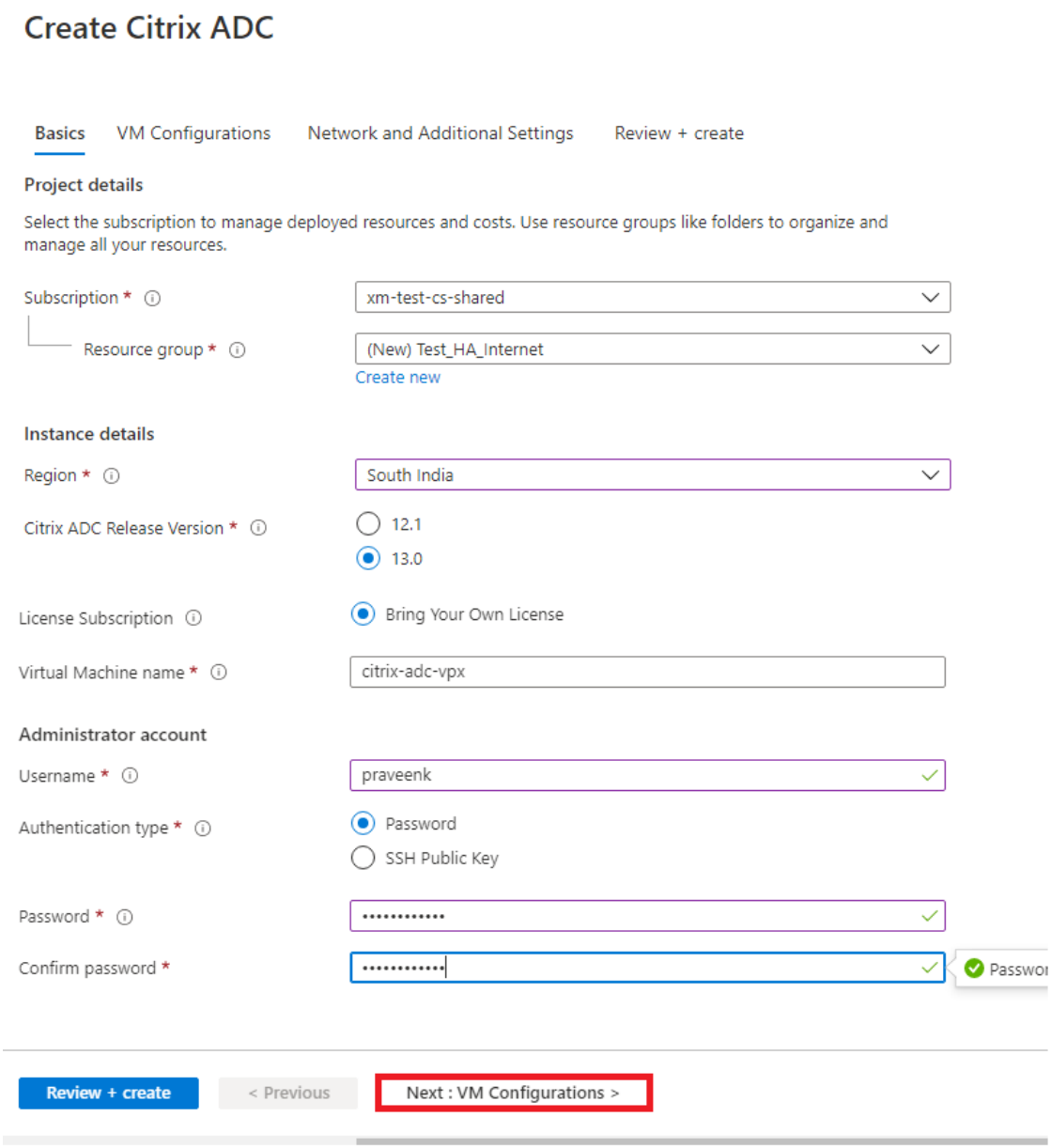
3. ライセンスとともに必要な HA 展開を選択し、[続行] をクリックします。



4. 「基本」ページが表示されます。リソースグループを作成します。[パラメータ] タブで、地域、管理者ユーザー名、管理者パスワード、ライセンスタイプ（VM SKU）、およびその他のフィールドの詳細を入力します。



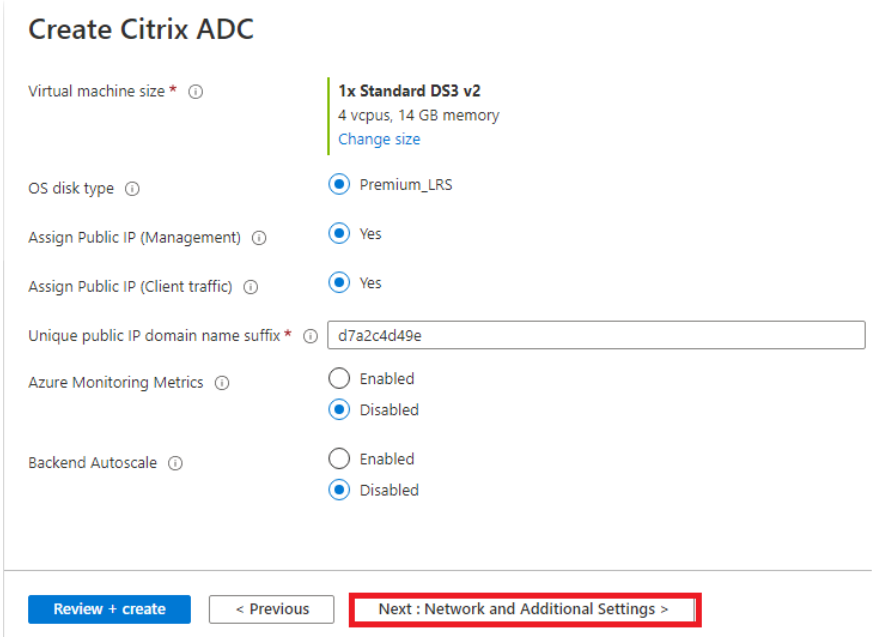
5. [次へ] をクリックします: VM 構成 > 。



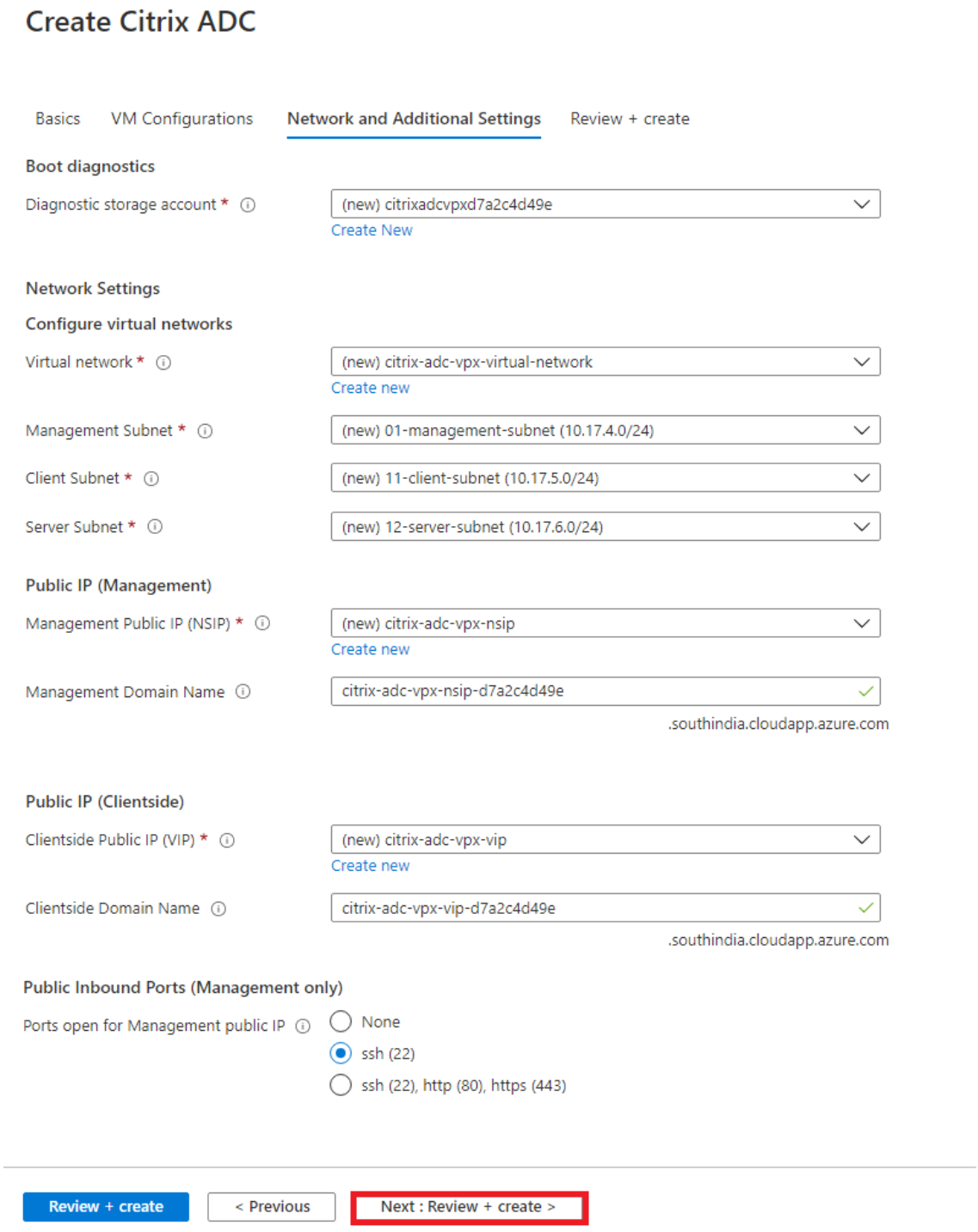
6. [**VM** 構成] ページで、次の手順を実行します。

   - パブリック IP ドメイン名のサフィックスを構成する
   - **Azure MonitoringMetrics** を有効または無効にする
   - バックエンド自動スケールを有効または無効にする

7. [次へ] をクリックします: ネットワークと追加設定 >



8. **[ネットワークと追加の設定]** ページで、ブート診断アカウントを作成し、ネットワーク設定を構成します。



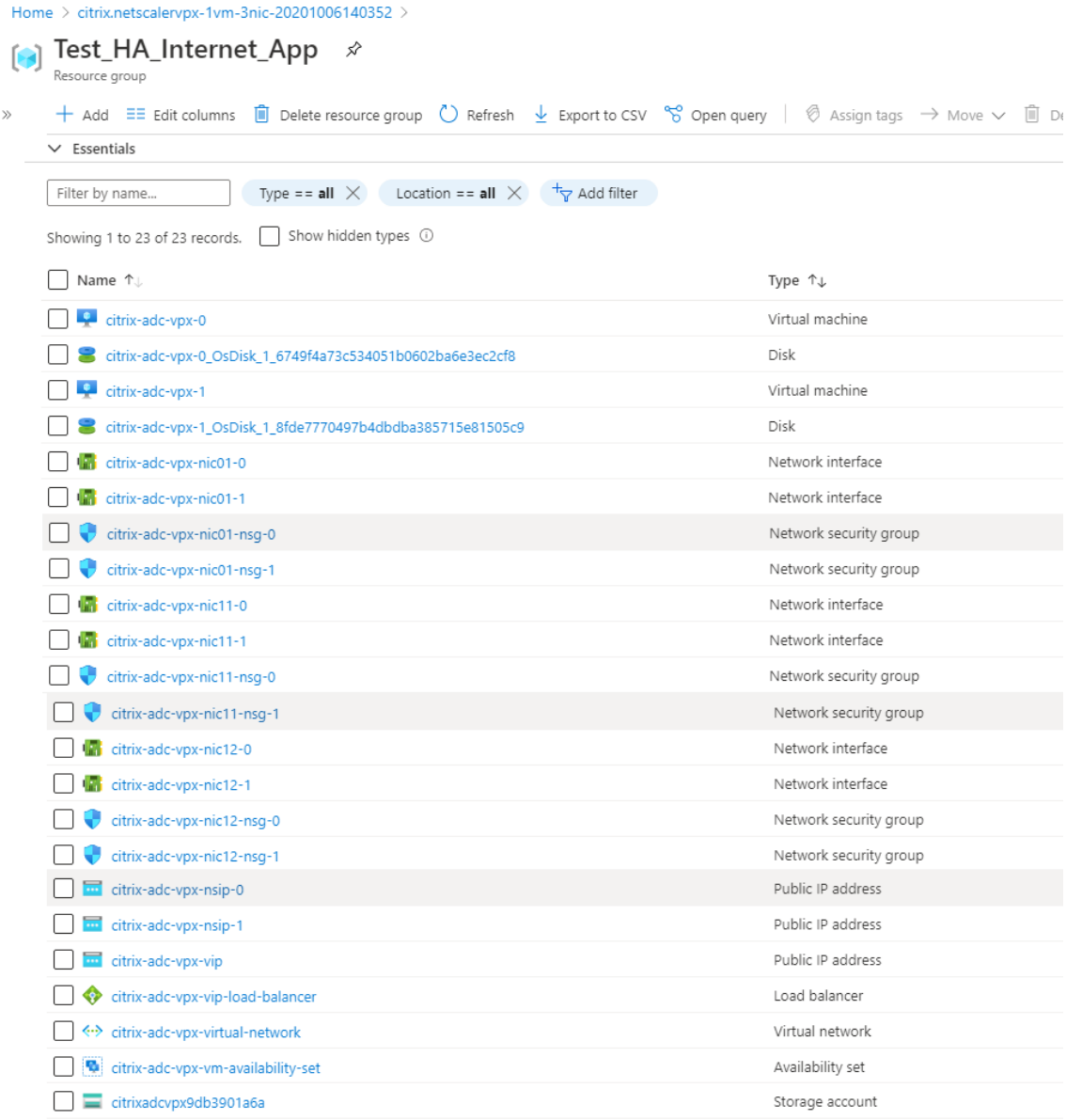
9. [次へ] をクリックします: レビュー + 作成する > 。
10. 基本設定、VM 構成、ネットワーク、および追加設定を確認し、[作成] をクリックします。

    必要な構成で Azure リソースグループが作成されるまで時間がかかることがあります。完了後、Azure ポー

タルでリソースグループを選択して、LB ルール、バックエンドプール、ヘルスプローブなどの構成の詳細を確認します。高可用性ペアは、**citrix-adc-vpx-0** および **citrix-adc-vpx-1** として表示されます。

追加のセキュリティルールやポートを作成するなど、HA セットアップでさらに変更が必要な場合は、Azure Portal から実行できます。

必要な構成が完了すると、次のリソースが作成されます。



11. 次の構成を検証するには、**citrix-adc-vpx-0** ノードと citrix-adc-vpx-1 ノードにログオンする必要があります。

    - 両方のノードの NSIP アドレスは、管理サブネット内にある必要があります。
    - プライマリ（citrix-adc-vpx-0）ノードとセカンダリ（citrix-adc-vpx-1）ノードで、2 つの SNIP アドレスを確認する必要があります。1 つの SNIP（クライアントサブネット）は ALB プローブへの応答に

使用され、もう 1 つの SNIP（サーバーサブネット）はバックエンドサーバー通信に使用されます。

> 注
>
> HA-INC モードでは、citrix-adc-vpx-0VM と citrix-adc-vpx-1VM の SNIP アドレスが異なります。これは、両方が同じである従来のオンプレミス ADC 高可用性展開とは異なります。

プライマリノード（citrix-adc-vpx-0）

```
> sh ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  ------
1)      10.18.0.4        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.4 (ns-vpx0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:34:21 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.5
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
```

セカンダリノード（citrix-adc-vpx-1）

```
> show ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  ------
1)      10.18.0.5        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
>
```

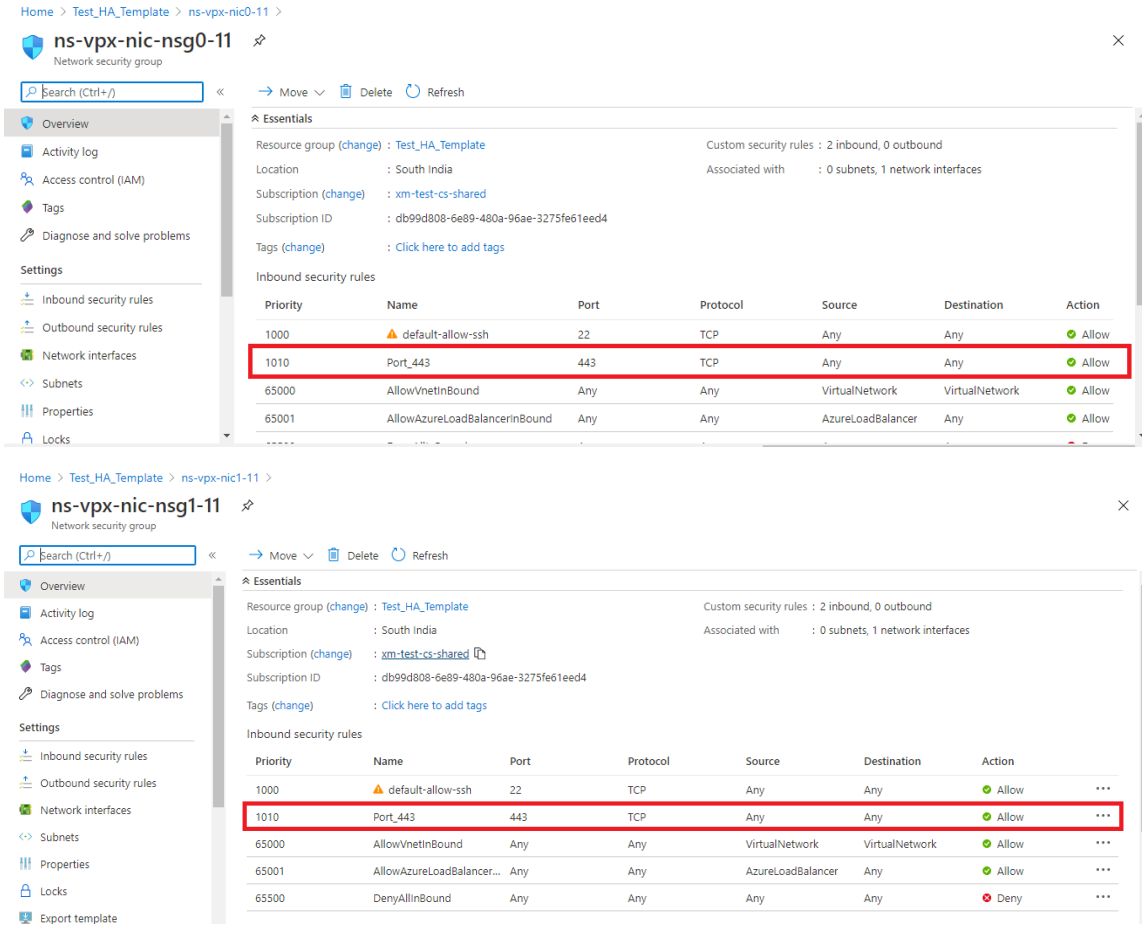


```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.5 (ns-vpx1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:23:51 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.4
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

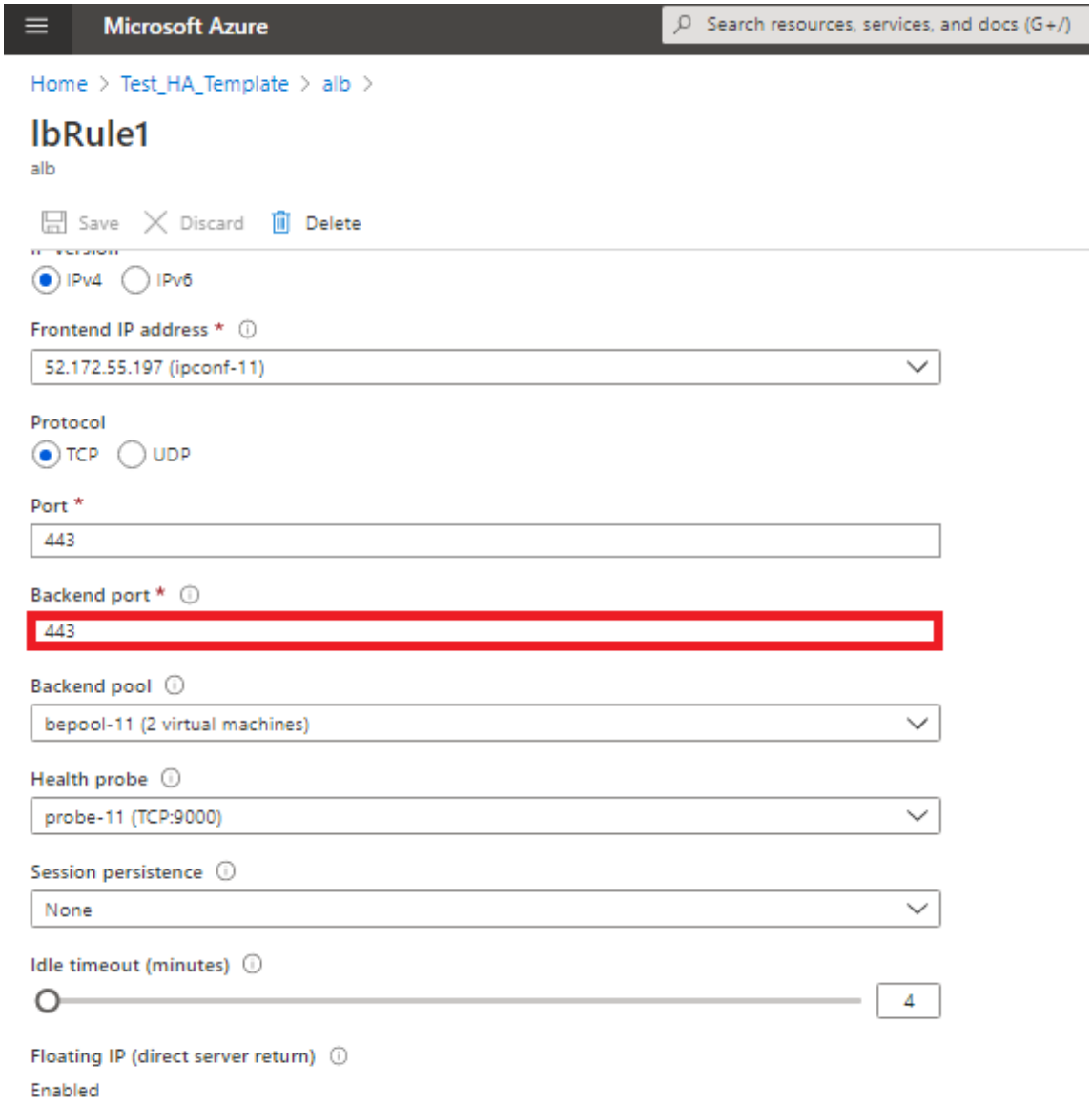
12. プライマリノードとセカンダリノードが UP になり、同期ステータスが **SUCCESS** になったら、ALB 仮想のパブリック IP アドレスを使用して、プライマリノード（citrix-adc-vpx-0）の負荷分散仮想サーバーまたはゲートウェイ仮想サーバーを構成する必要があります。サーバ。詳細については、「サンプル設定」セクションを参照してください。

13. ALB 仮想サーバーのパブリック IP アドレスを見つけるには、**Azure portal > Azure Load Balancer > Frontend IP configuration** に移動します。

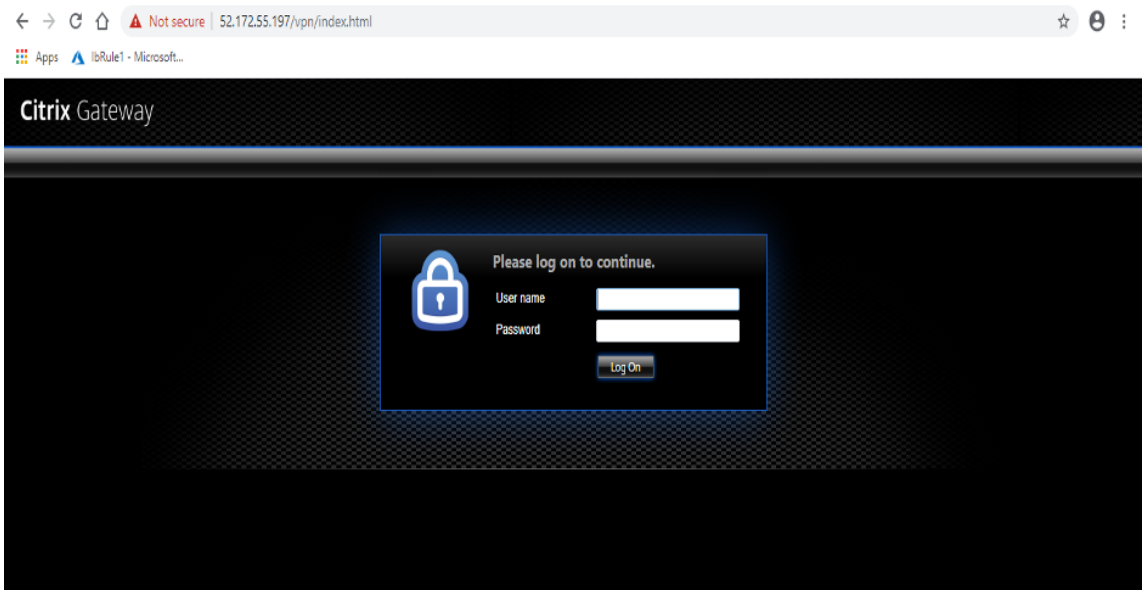
14. 両方のクライアントインターフェイスのネットワークセキュリティグループに、仮想サーバーポート 443 のインバウンドセキュリティルールを追加します。



15. アクセスする ALB ポートを構成し、指定したポートのインバウンドセキュリティルールを作成します。バックエンドポートは、負荷分散仮想サーバーポートまたは VPN 仮想サーバーポートです。



16. これで、ALB パブリック IP アドレスに関連付けられた FQDN を使用して、負荷分散仮想サーバーまたは VPN 仮想サーバーにアクセスできます。



構成例

ゲートウェイ VPN 仮想サーバーと負荷分散仮想サーバーを構成するには、プライマリノード（ADC-VPX-0）で次のコマンドを実行します。構成は、セカンダリノード（ADC-VPX-1）に自動同期します。

ゲートウェイのサンプル構成

```
enable feature aaa LB SSL SSLVPN
add ip 52.172.55.197 255.255.255.0 -type VIP
add vpn vserver vpn_ssl SSL 52.172.55.197 443
add ssl certKey ckp -cert cgwsanity.cer -key cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

負荷分散のサンプル構成

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 52.172.55.197 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

これで、ILB の内部 IP アドレスに関連付けられた完全修飾ドメイン名（FQDN）を使用して、負荷分散または VPN 仮想サーバーにアクセスできます。

負荷分散仮想サーバーを構成する方法の詳細については、「リソース」セクションを参照してください。

リソース:

次のリンクは、HA の展開と仮想サーバーの構成に関連する追加情報を提供します。

- 仮想サーバーを作成する
- 基本的な負荷分散の設定

Azure 外部および内部ロードバランサーで同時に高可用性セットアップを構成する

October 7, 2021

Azure の高可用性ペアは、外部ロードバランサーと内部ロードバランサーの両方を同時にサポートします。

Azure 外部ロードバランサーと内部ロードバランサーの両方を使用して高可用性ペアを構成するには、次の 2 つのオプションがあります。

- Citrix ADC アプライアンス上で 2 つの LB 仮想サーバーを使用する。
- 1 つの LB 仮想サーバーと IP セットを使用する。単一の LB 仮想サーバは、IPSet によって定義された複数の IP にトラフィックを処理します。

外部ロードバランサーと内部ロードバランサーを同時に使用して Azure で高可用性ペアを構成するには、次の手順を実行します。

手順 1 と 2 については、Azure ポータルを使用します。手順 3 および 4 では、Citrix ADC VPX GUI または CLI を使用します。

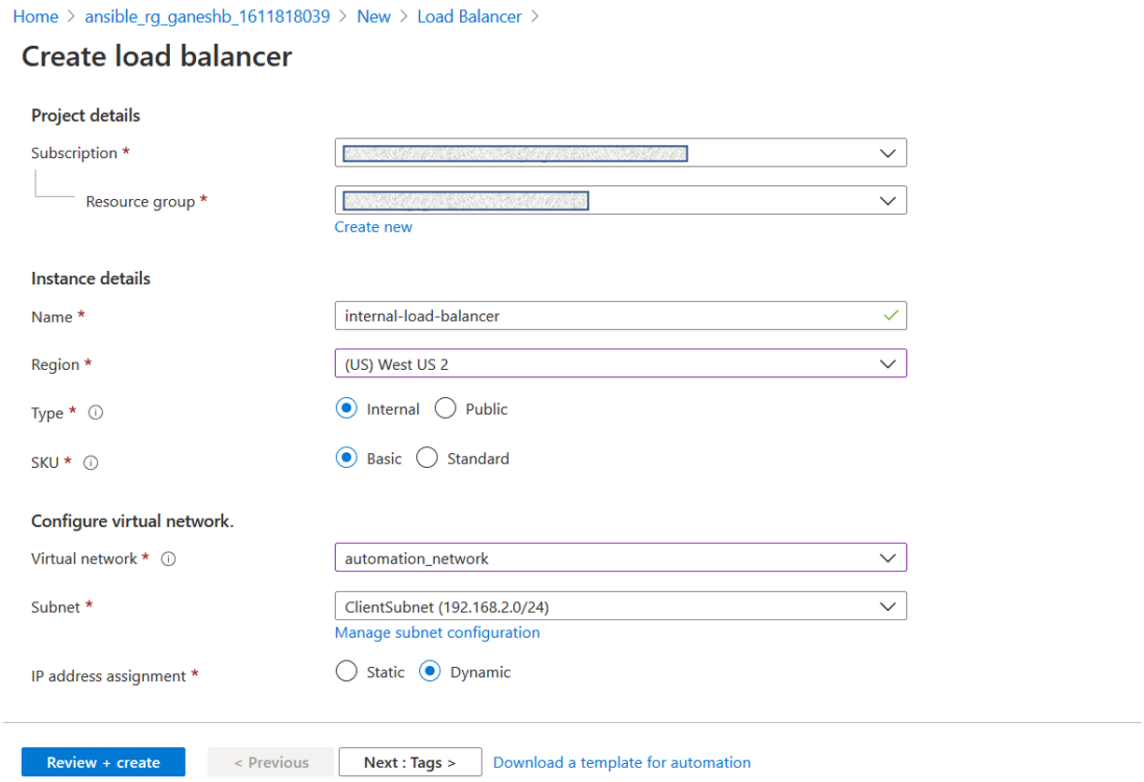
手順 **1.** Azure ロードバランサー (外部ロードバランサーまたは内部ロードバランサー) を構成します。

Azure 外部ロードバランサーを使用した高可用性セットアップの構成の詳細については、「複数の IP アドレスと NIC を使用した高可用性セットアップを構成する」を参照してください。

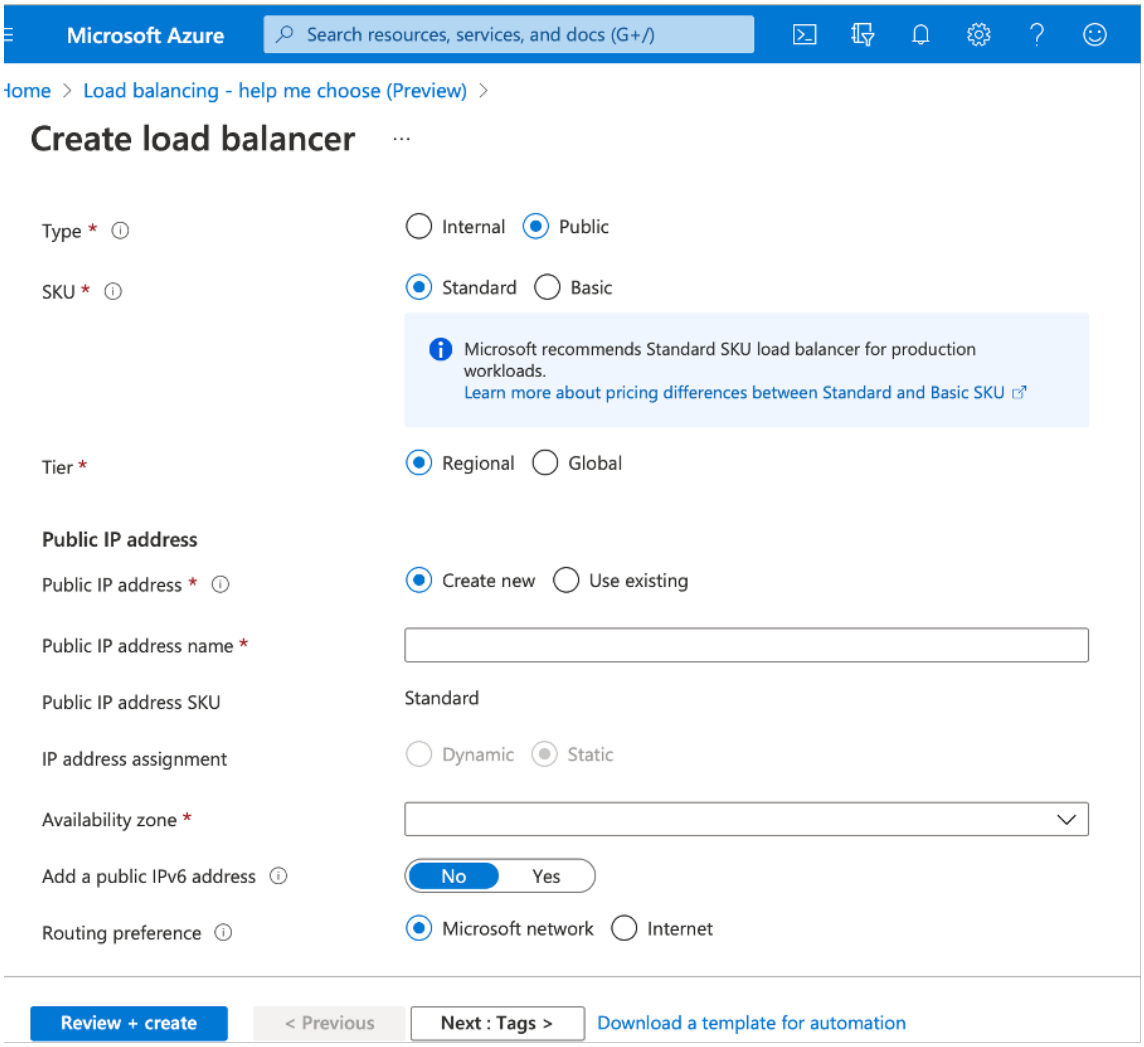
Azure 内部ロードバランサーを使用した高可用性セットアップの構成の詳細については、「Azure ILB で Citrix 高可用性テンプレートを使用して HA-INC ノードを構成する」を参照してください。

手順 **2.** リソースグループに追加のロードバランサー (ILB) を作成します。ステップ 1 では、外部ロードバランサーを作成した場合は、内部ロードバランサーを作成し、逆に作成します。

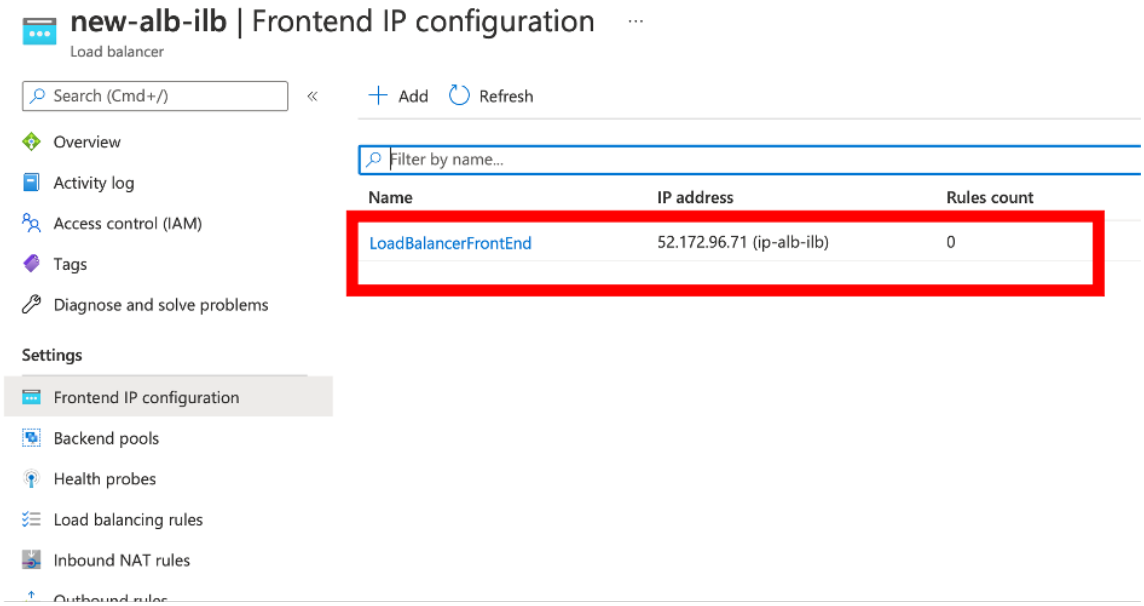
- 内部ロードバランサーを作成するには、ロードバランサのタイプを [内部] として選択します。[サブネット] フィールドで、Citrix ADC クライアントサブネットを選択する必要があります。競合がない限り、そのサブネットに静的 IP アドレスを指定することもできます。それ以外の場合は、ダイナミック IP アドレスを選択します。



- 外部ロードバランサーを作成するには、ロードバランサの種類を [パブリック] として選択し、ここにパブリック IP アドレスを作成します。



1. Azure Load Balancer を作成したら、フロントエンド **IP** 設定に移動し、ここに示す IP アドレスを書き留めます。ステップ 3 のように ADC 負荷分散仮想サーバーを作成するときは、この IP アドレスを使用する必要があります。



2. **Azure Load Balancer** の設定ページで、ARM テンプレートのデプロイは、LB ルール、バックエンドプール、およびヘルスプローブの作成に役立ちます。
3. 高可用性ペアのクライアント NIC を ILB のバックエンドプールに追加します。
4. ヘルスプローブの作成（TCP、9000 ポート）
5. 次の 2 つのロードバランシングルールを作成します。
   - ポート 80 の HTTP トラフィック（Webapp ユースケース）の 1 つの LB ルール。ルールでは、バックエンドポート 80 も使用する必要があります。作成したバックエンドプールとヘルスプローブを選択します。フローティング IP を有効にする必要があります。
   - ポート 443 の HTTPS または CVAD トラフィックに対する別の LB ルール。プロセスは HTTP トラフィックと同じです。

手順 **3.** Citrix ADC アプライアンスのプライマリノードで、ILB の負荷分散仮想サーバーを作成します。

1. 負荷分散仮想サーバーを追加します。

```
add lb vserver <name> <serviceType> [<ILB Frontend IP address>] [<
    port>]
<!--NeedCopy-->
```

例:

```
add lb vserver vserver_name HTTP 52.172.96.71 80
<!--NeedCopy-->
```


> 注:
>
> ステップ 2 で作成した追加のロードバランサーに関連付けられた、ロードバランサーのフロントエンド IP アドレスを使用します。

2. サービスを負荷分散仮想サーバーにバインドします。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

詳細については、「基本的な負荷分散の設定」を参照してください。

ステップ **4:** ステップ 3 の代わりに、IPSet を使用して ILB の負荷分散仮想サーバーを作成できます。

1. 仮想サーバー IP (VIP) タイプの IP アドレスを追加します。

```
add nsip <ILB Frontend IP address> -type <type>
<!--NeedCopy-->
```

例:

```
add nsip 52.172.96.71 -type vip
<!--NeedCopy-->
```

2. プライマリノードとセカンダリノードの両方に IPSet を追加します。

```
add ipset <name>
<!--NeedCopy-->
```

例:

```
add ipset ipset1
<!--NeedCopy-->
```


3. IP アドレスを IP セットにバインドします。

```
bind ipset <name> <ILB Frontend IP address>
<!--NeedCopy-->
```

例:

```
bind ipset ipset1 52.172.96.71
<!--NeedCopy-->
```

4. 既存の LB 仮想サーバーを IPSet を使用するように設定します。

```
set lb vserver <vserver name> -ipset <ipset name>
<!--NeedCopy-->
```

例:

```
set lb vserver vserver_name -ipset ipset1
<!--NeedCopy-->
```

詳細については、「マルチ IP 仮想サーバーの構成」を参照してください。

Azure VMware ソリューションに Citrix ADC VPX インスタンスをインストールする

December 7, 2021

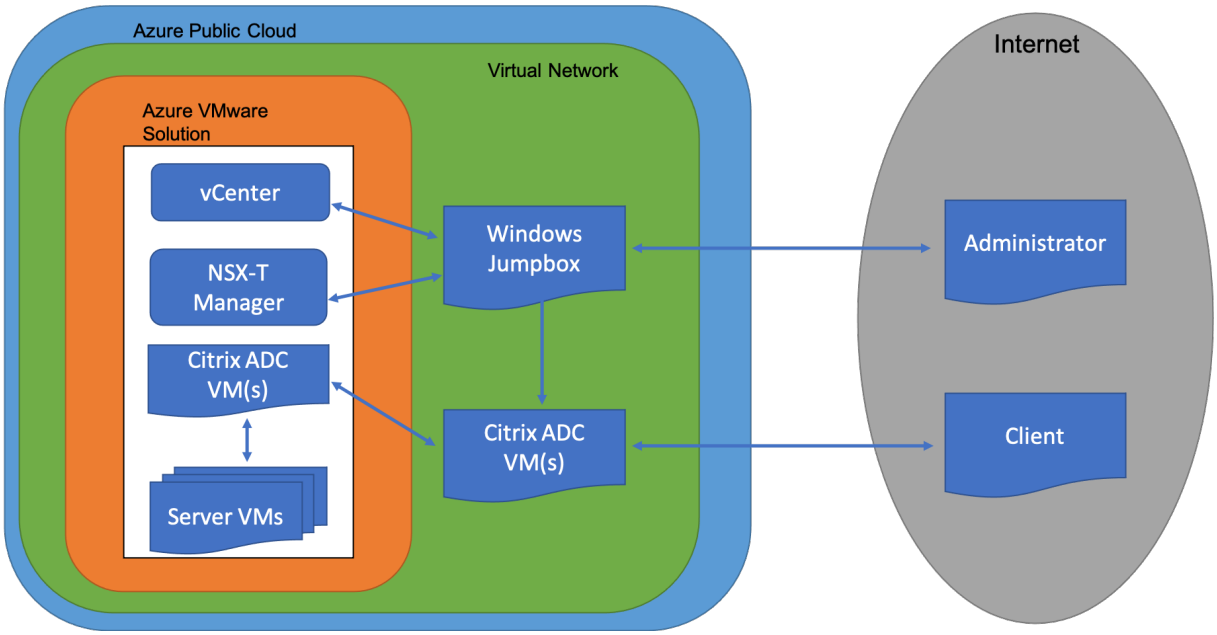
Azure VMware ソリューション (AVS) は、専用のベアメタル Azure インフラストラクチャから構築された vSphere クラスタを含むプライベートクラウドを提供します。最初のデプロイメントは最小で 3 台のホストですが、追加ホストは一度に 1 つずつ追加でき、クラスタごとに最大 16 台のホストを追加できます。プロビジョニングされたすべてのプライベートクラウドには、vCenter Server、vSAN、vSphere、NSX-T があります。

Azure 上の VMware クラウド (VMC) を使用すると、必要な数の ESX ホストを使用して Azure 上にクラウドソフトウェア定義データセンター (SDDC) を作成できます。Azure 上の VMC は、Citrix ADC VPX デプロイメントをサポートしています。VMC は、オンプレミスの vCenter と同じユーザー・インタフェースを提供します。これは、ESX ベースの Citrix ADC VPX 展開と同様に機能します。

次の図は、管理者またはクライアントがインターネット経由でアクセスできる Azure パブリッククラウド上の Azure VMware ソリューションを示しています。管理者は、Azure VMware ソリューションを使用して、ワークロードまた

はサーバー仮想マシンを作成、管理、および構成できます。管理者は、Windows ジャンプボックスから AVS の Web ベースの vCenter および NSX-T マネージャにアクセスできます。vCenter を使用して Azure VMware Solution 内に Citrix ADC VPX インスタンス（スタンドアロンまたは高可用性ペア）とサーバー仮想マシンを作成し、NSX-T Manager を使用して対応するネットワークを管理できます。AVS 上の Citrix ADC VPX インスタンスは、オンプレミスの VMware ホストのクラスタと同様に機能します。AVS は、同じ仮想ネットワーク内に作成された Windows ジャンプボックスから管理されます。

クライアントは、ADC の VIP に接続することによってのみ AVS サービスにアクセスできます。Azure VMware ソリューション外の別の Citrix ADC VPX インスタンスは、同じ Azure 仮想ネットワーク内にある別の Citrix ADC VPX インスタンスは、Azure VMware ソリューション内の Citrix ADC VPX インスタンスの VIP をサービスとして追加するのに役立ちます。要件に応じて、インターネット上でサービスを提供するように Citrix ADC VPX インスタンスを構成できます。



前提条件

仮想アプライアンスのインストールを開始する前に、次の操作を行います。

- Azure VMware ソリューションとその前提条件の詳細については、Azure VMware ソリューションのドキュメントを参照してください。
- Azure VMware ソリューションのデプロイの詳細については、「Azure VMware ソリューションのプライベートクラウドをデプロイする」を参照してください。
- Azure VMware ソリューションにアクセスして管理するための Windows ジャンプボックス仮想マシンの作成の詳細については、「Azure VMware ソリューションのプライベートクラウドにアクセスする」を参照してください。
- Windows ジャンプボックス仮想マシンで、Citrix ADC VPX アプライアンスのセットアップファイルをダウンロードします。

- 仮想マシンが接続する VMware SDDC 上に、適切な NSX-T ネットワークセグメントを作成します。詳細については、「Azure VMware ソリューションでのネットワークセグメントの追加」を参照してください。
- VPX ライセンスファイルを入手します。
- Azure VMware Solution プライベートクラウドに作成または移行された仮想マシン (VM) は、ネットワークセグメントに接続する必要があります。

VMware クラウドのハードウェア要件

次の表に、VMware SDDC が各 VPX nCore 仮想アプライアンスに対して提供する必要がある仮想コンピューティングリソースを示します。

表 1. Citrix ADC VPX インスタンスの実行に必要な最小仮想コンピューティングリソース

コンポーネント	条件
メモリ	2GB
仮想 CPU（VCPU）	2
仮想ネットワークインターフェイス	VMware SDDC では、VPX ハードウェアをバージョン 7 以上にアップグレードすると、最大 10 個の仮想ネットワークインターフェイスをインストールできます。
ディスク領域	20GB

> 注
>
> ハイパーバイザーに必要なディスク領域は含まれません。

VPX 仮想アプライアンスを実稼働で使用するには、フルメモリ割り当てを予約する必要があります。

OVF ツール 1.0 のシステム要件

Open Virtualization Format Tool（OVF Tool）は、Windows および Linux システムで実行できるクライアントアプリケーションです。次の表に、OVF ツールをインストールするためのシステム要件を示します。

表 2. OVF ツールのインストールに関するシステム要件

コンポーネント	条件
オペレーティングシステム	VMware からの詳細な要件については、http://kb.vmware.com/で「OVF ツールユーザーガイド」PDF ファイルを検索してください。
CPU	最低 750 MHz、1 GHz またはそれ以上の速度を推奨

コンポーネント	条件
RAM	最小 1 GB、推奨 2 GB
NIC	100Mbps 以上の NIC。

OVF のインストールについては、http://kb.vmware.com/で「OVF ツールユーザーガイド」の PDF ファイルを検索してください。

Citrix ADC VPX セットアップファイルのダウンロード

VMware ESX 用の Citrix ADC VPX インスタンスセットアップパッケージは、オープン仮想マシン（OVF）形式の標準に準拠しています。これらのファイルは、Citrix の Web サイトからダウンロードできます。ログオンするには、Citrix アカウントが必要です。Citrix アカウントをお持ちでない場合は、http://www.citrix.comのホームページにアクセスしてください。[新しいユーザー] リンクをクリックし、指示に従って新しい Citrix アカウントを作成します。

ログオンしたら、Citrix のホームページから次のパスをナビゲートします。

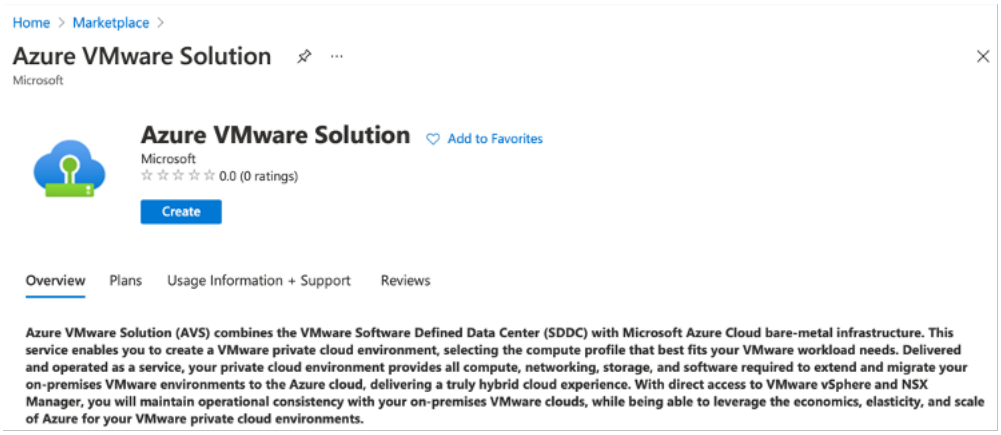
Citrix.com > ダウンロード **> Citrix ADC >** 仮想アプライアンス。

次のファイルを、ESX サーバーと同じネットワーク上のワークステーションにコピーします。3 つのファイルをすべて同じフォルダーにコピーします。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（例えば、NSVPX-ESX-13.0-79.64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf（例えば、NSVPX-ESX-13.0-79.64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（例えば、NSVPX-ESX-13.0-79.64.mf）

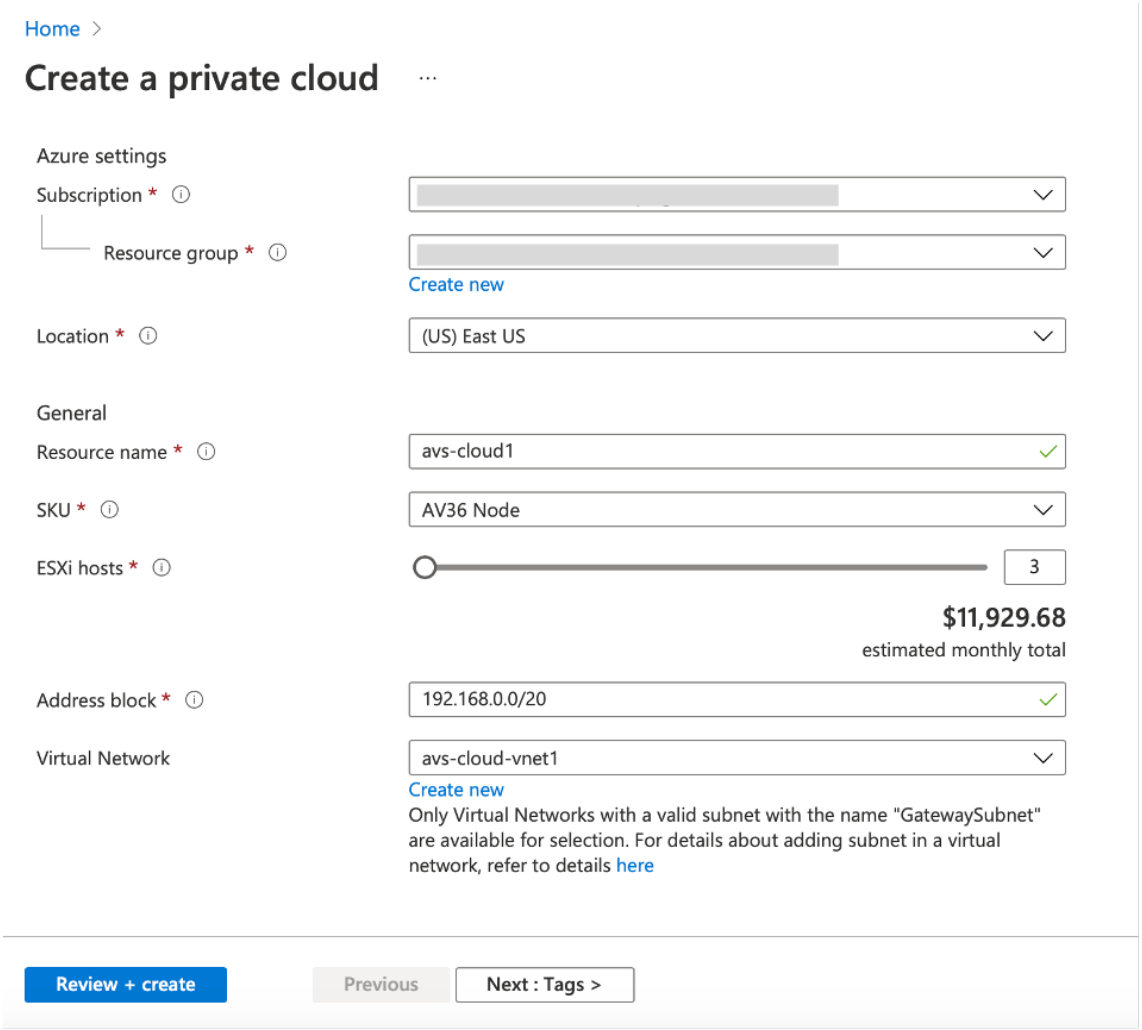
Azure VMware ソリューションをデプロイする

1. Microsoft Azure ポータルにログインし、Azureマーケットプレイスに移動します。
2. **Azure** マーケットプレイスから **AzureVMware** ソリューションを検索し、[作成] をクリックします。



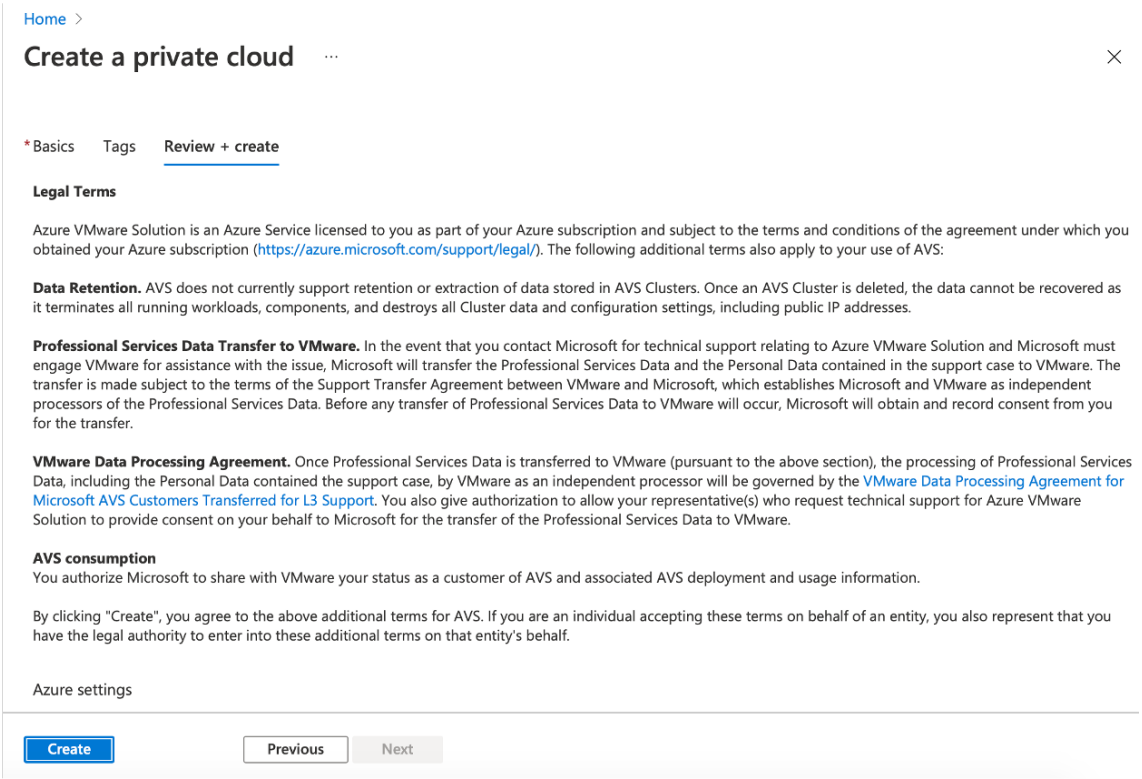
3. [プライベートクラウドの作成] ページで、次の詳細を入力します。

   - プライベートクラウドのデフォルトクラスタを作成するには、最低 3 つの ESXi ホストを選択します。
   - [**Address** ブロック] フィールドには、**/22** アドレス空間を使用します。
   - 仮想ネットワークの場合、CIDR 範囲が、オンプレミスまたはその他の Azure サブネット (仮想ネットワーク) またはゲートウェイサブネットと重複していないことを確認します。
   - ゲートウェイサブネットは、プライベートクラウドとの接続のルーティングを表現するために使用されます。

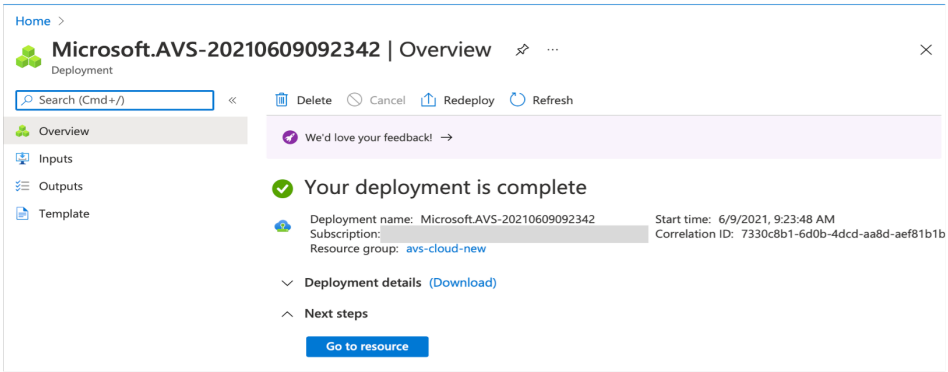


4. [レビュー**] + [**作成] をクリックします。

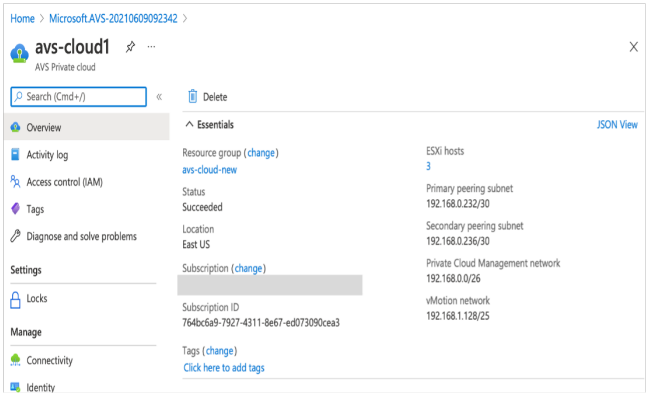
5. 設定を確認します。設定を変更する必要がある場合は、[前へ] をクリックします。



6. [作成] をクリックします。プライベートクラウドのプロビジョニングプロセスが開始されます。プライベートクラウドのプロビジョニングには最大 2 時間かかることがあります。



7. [リソースに移動] をクリックして、作成されたプライベートクラウドを確認します。



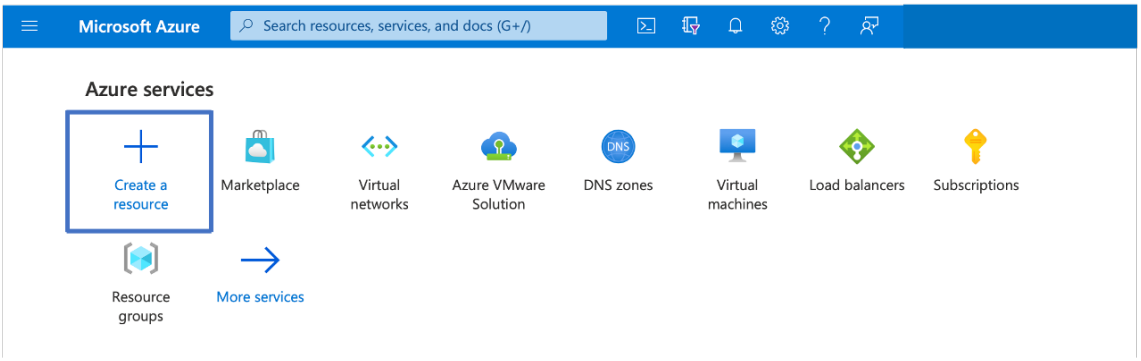
> 注
>
> このリソースにアクセスするには、Windows でジャンプボックスとして機能する仮想マシンが必要です。

Windows を実行している **Azure** 仮想マシンに接続する

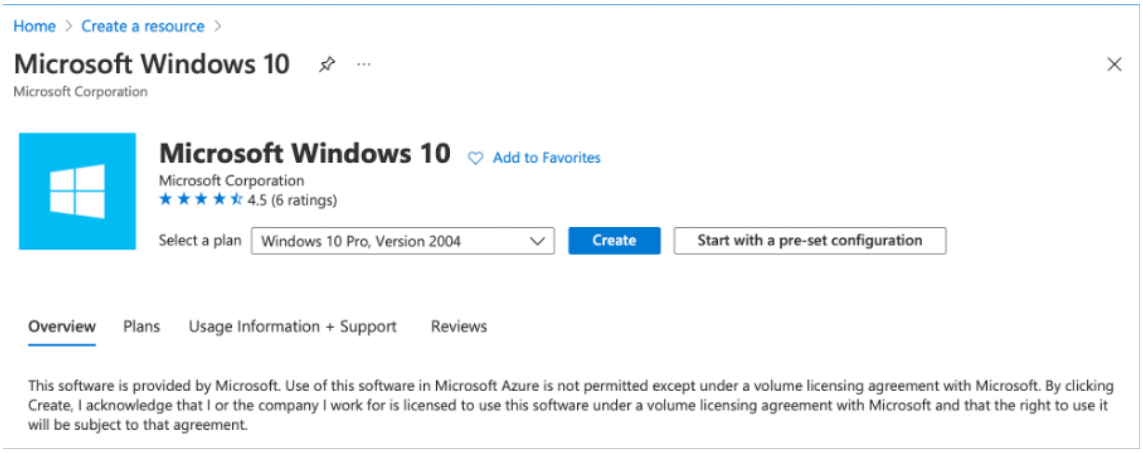
この手順では、Azure ポータルを使用して、Windows Server 2019 を実行する仮想マシン (VM) を Azure にデプロイする方法について説明します。VM の動作を確認するには、仮想マシンに RDP し、IIS Web サーバーをインストールします。

作成したプライベートクラウドにアクセスするには、同じ仮想ネットワーク内に Windows ジャンプボックスを作成する必要があります。

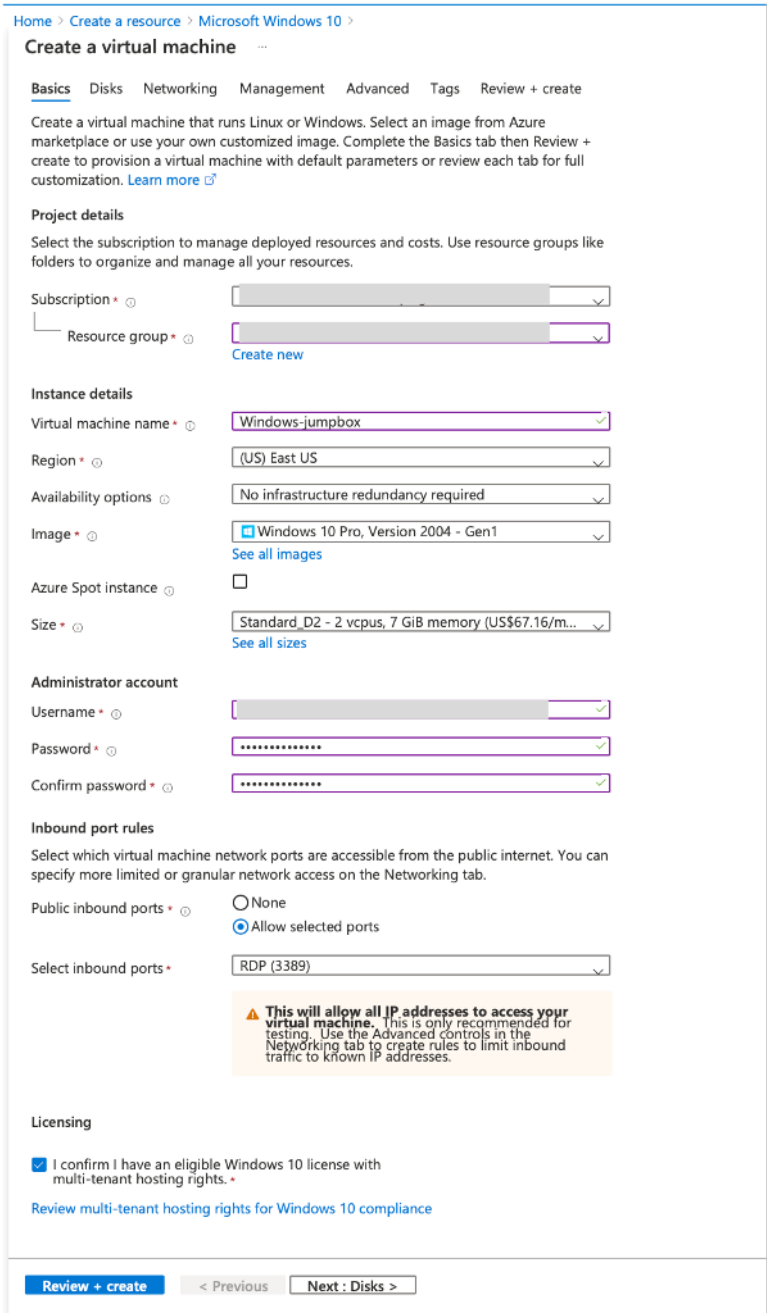
1. **Azure** ポータルに移動し、[リソースの作成] をクリックします。



2. **Microsoft Windows 10** を検索し、[作成] をクリックします。



3. Windows Server 2019 を実行する仮想マシン (VM) を作成します。[仮想マシンの作成] ページが表示されます。[基本] タブにすべての詳細を入力し、[ライセンス] チェックボックスをオンにします。残りのデフォルトのままにして、ページの下部にある [**Review + create**] ボタンを選択します。



4. 検証の実行後、ページの下部にある [作成] ボタンを選択します。
5. デプロイが完了したら、[リソースに移動] を選択します。
6. 作成した Windows 仮想マシンに移動します。Windows 仮想マシンのパブリック IP アドレスを使用し、RDP を使用して接続します。

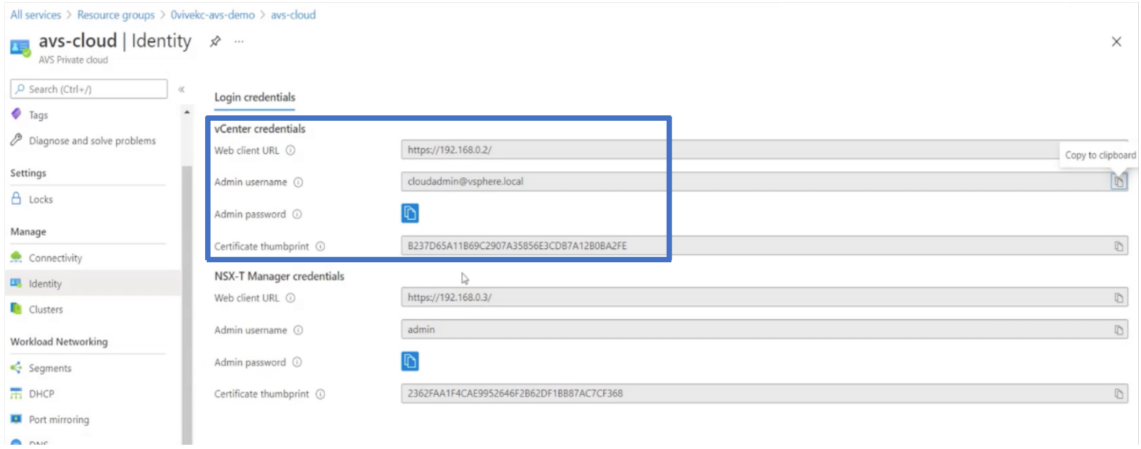
   Azure ポータルの [接続] ボタンを使用して、Windows デスクトップからリモートデスクトップ (RDP) セッションを開始します。まず仮想マシンに接続し、次にサインオンします。

   Mac から Windows 仮想マシンに接続するには、Microsoft リモートデスクトップなどの Mac 用 RDP クラ

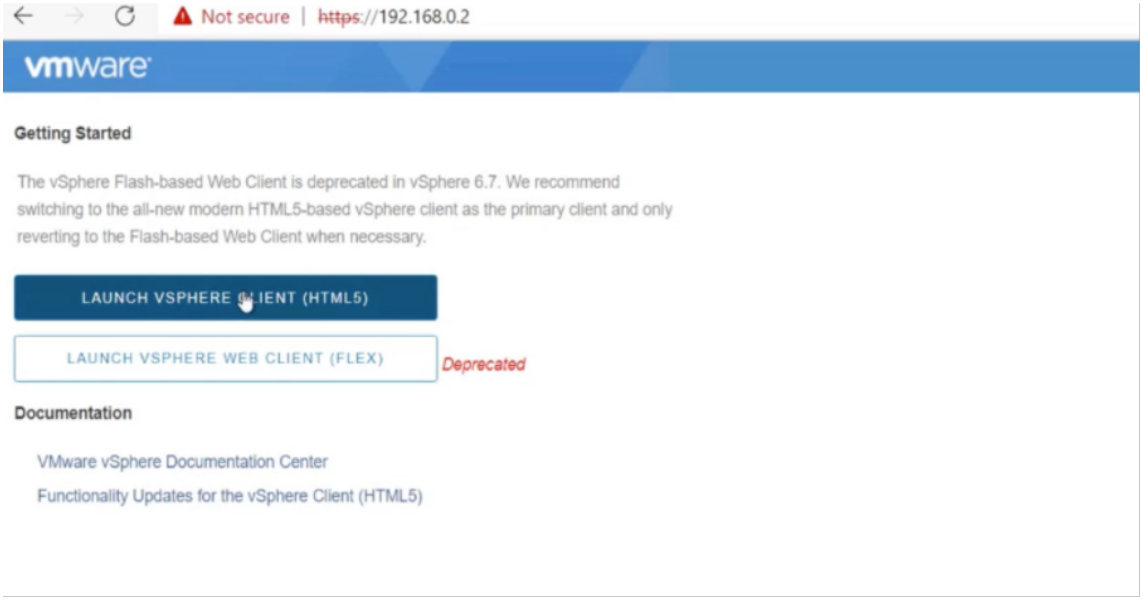
イアントをインストールする必要があります。詳細については、「Windows を実行する Azure 仮想マシンに接続してサインオンする方法」を参照してください。

プライベートクラウド **vCenter** ポータルにアクセスする

1. Azure VMware ソリューションのプライベートクラウドで、[管理] で [アイデンティティ] を選択します。vCenter の認証情報を書き留めます。

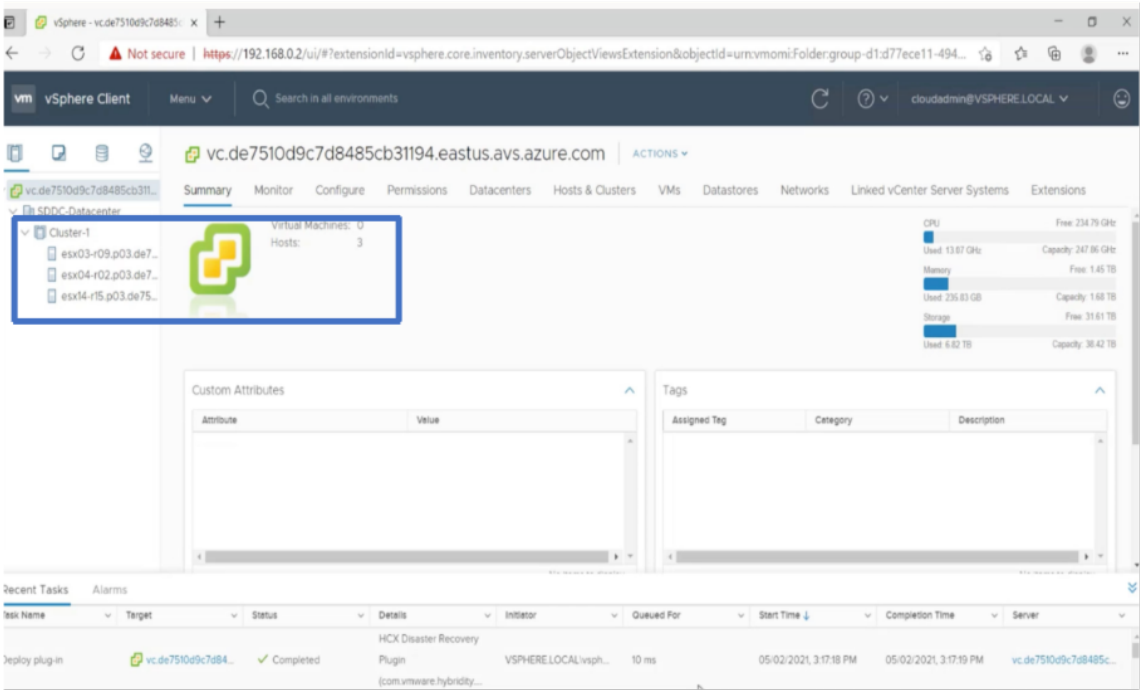


2. vCenter Web クライアントの URL を入力して、vSphere クライアントを起動します。



3. Azure VMware ソリューションプライベートクラウドの vCenter 認証情報を使用して VMware vSphere にログインします。

4. vSphere クライアントでは、Azure ポータルで作成した ESXi ホストを確認できます。



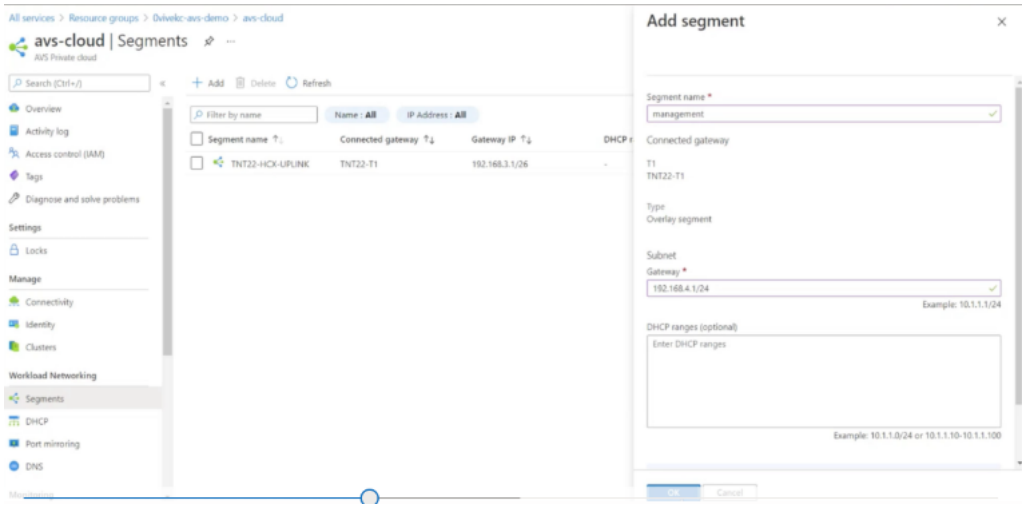
詳細については、「プライベートクラウド vCenter ポータルへのアクセス」を参照してください。

Azure ポータルで NSX-T セグメントを作成します

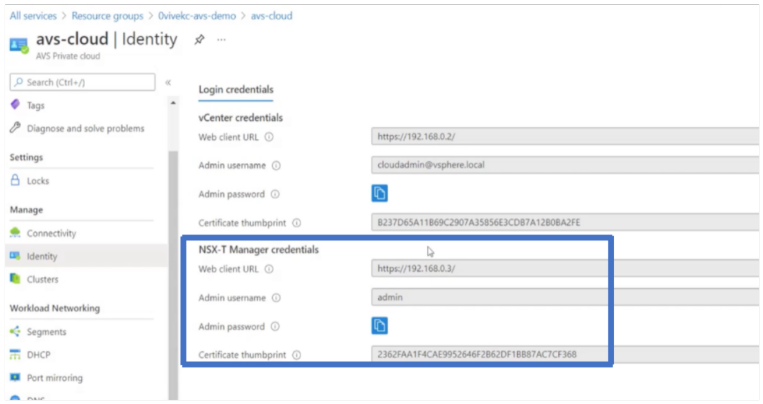
NSX-T セグメントは、Azure ポータルの Azure VMware ソリューションコンソールから作成および構成できます。これらのセグメントはデフォルトの Tier-1 ゲートウェイに接続され、これらのセグメントのワークロードは East-West および North-South 接続を取得します。セグメントを作成すると、NSX-T Manager および vCenter

に表示されます。

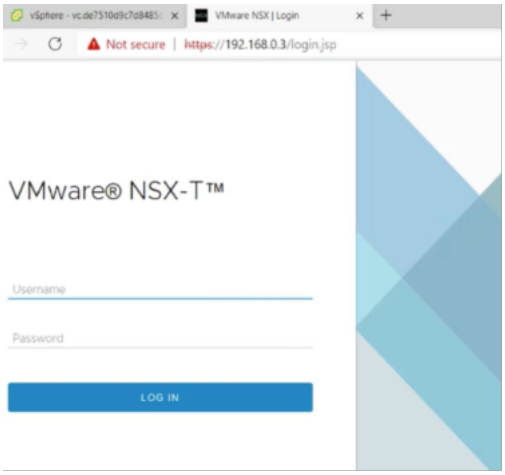
1. Azure VMware ソリューションのプライベートクラウドで、[ワークロードネットワーキング] で、[セグメント] **> [**追加] の順に選択します。新しい論理セグメントの詳細を入力し、「**OK**」を選択します。クライアント、管理、およびサーバーインターフェイスに対して 3 つの別々のセグメントを作成できます。



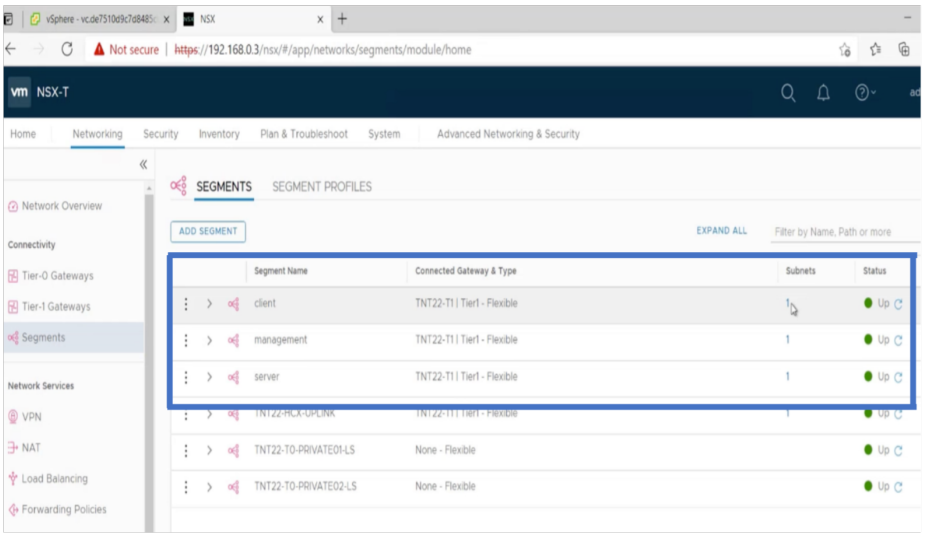
2. Azure VMware ソリューションのプライベートクラウドで、[管理] で [アイデンティティ] を選択します。NSX-T マネージャのクレデンシャルを書き留めます。



3. NSX-T Web クライアント URL を入力して VMware NSX-T マネージャを起動します。



4. NSX-T マネージャの [ネットワーク] **>** **[**セグメント] の下に、作成したすべてのセグメントが表示されます。サブネットを確認することもできます。



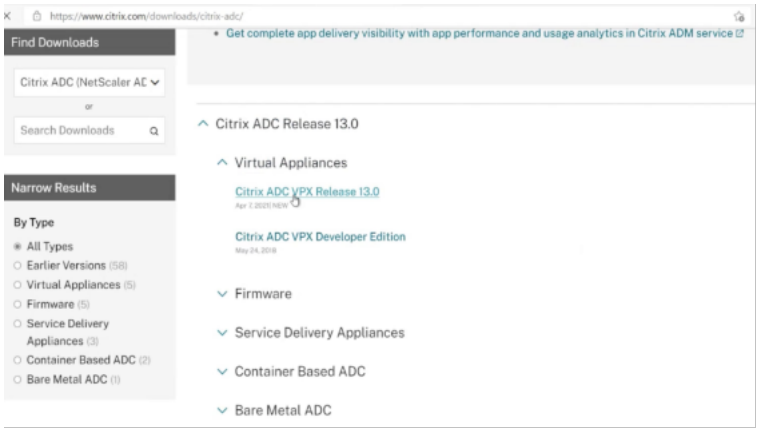
詳細については、「Azure ポータルで NSX-T セグメントを作成する」を参照してください。

VMware クラウドへの **Citrix ADC VPX** インスタンスのインストール

VMware ソフトウェア定義データセンター（SDDC）をインストールして構成したら、SDDC を使用して VMware クラウドに仮想アプライアンスをインストールできます。インストールできる仮想アプライアンスの数は、SDDC で使用可能なメモリの量によって異なります。

VMware クラウドに Citrix ADC VPX インスタンスをインストールするには、Windows ジャンプボックス仮想マシンで次の手順を実行します。

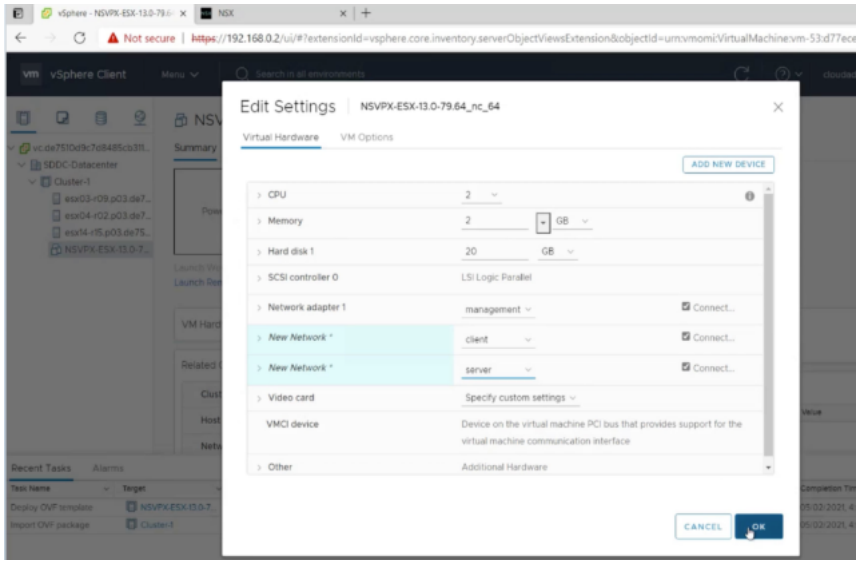
1. Citrix ダウンロードサイトから ESXi ホストの Citrix ADC VPX インスタンスセットアップファイルをダウンロードします。



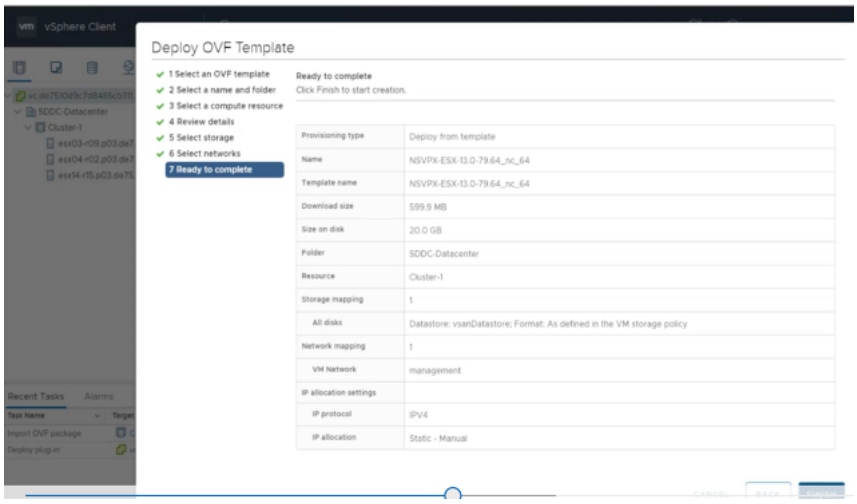
2. Windows のジャンプボックスで VMware SDDC を開きます。
3. [ユーザー名] フィールドと [パスワード] フィールドに管理者の資格情報を入力し、[ログイン] をクリックします。
4. [**File**] メニューの [**Deploy OVF Template**] を選択します。
5. [**OVF** テンプレートのデプロイ] ダイアログボックスの [ファイルからの展開] フィールドで、Citrix ADC VPX インスタンスセットアップファイルを保存した場所を参照し、.ovf ファイルを選択し、[次へ] をクリックします。

   > メモ
   >
   > デフォルトでは、Citrix ADC VPX インスタンスは E1000 ネットワークインターフェイスを使用します。VMXNET3 インターフェイスで ADC を展開するには、E1000 ではなく VMXNET3 インターフェイスを使用するように OVF を変更します。VMXNET3 インターフェイスの可用性は Azure インフラストラクチャによって制限され、Azure VMware ソリューションでは利用できない場合があります。

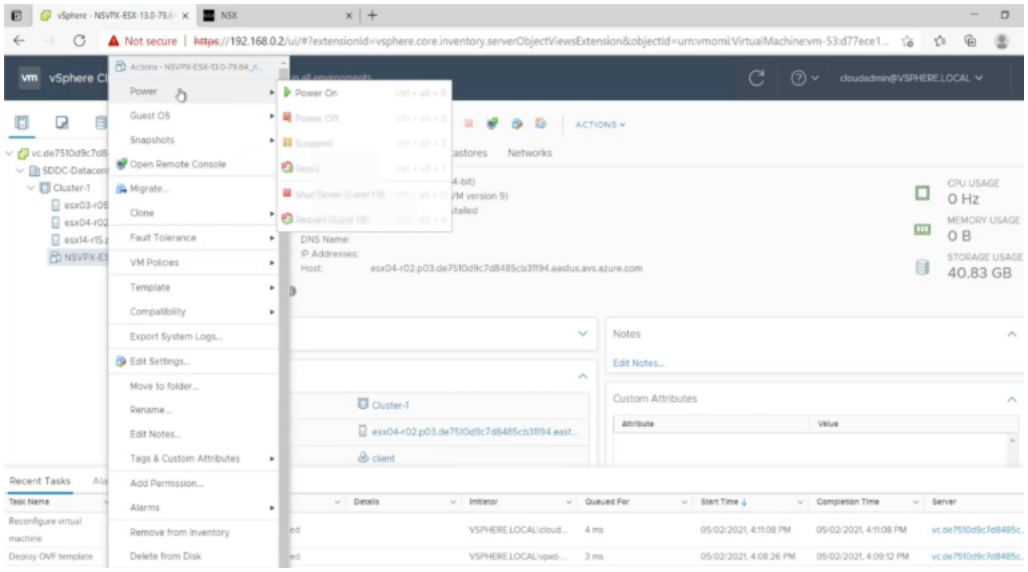
6. 仮想アプライアンス OVF テンプレートに表示されるネットワークを、VMware SDDC で設定したネットワークにマッピングします。[**OK**] をクリックします。



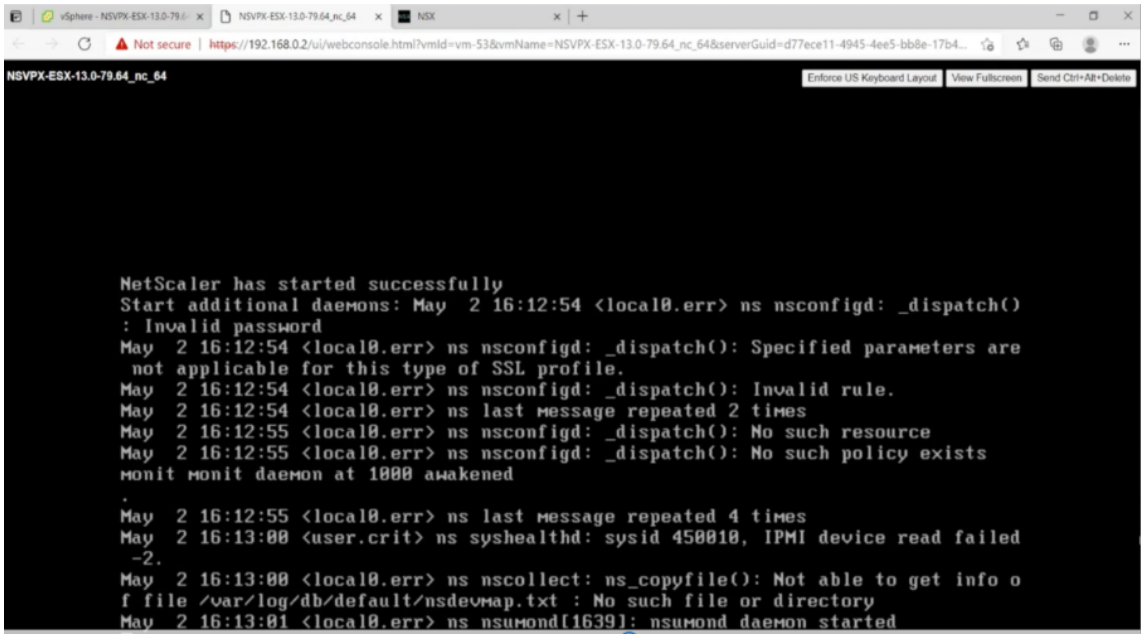
7. [完了] をクリックして VMware SDDC への仮想アプライアンスのインストールを開始します。



8. これで、Citrix ADC VPX インスタンスを起動する準備ができました。ナビゲーションペインで、インストールした Citrix ADC VPX インスタンスを選択し、右クリックメニューから［電源オン］を選択します。コンソールポートをエミュレートするには、［**Console**］タブをクリックします。



9. これで、vSphere クライアントから Citrix ADC 仮想マシンに接続されています。



10. SSH キーを使用して Citrix ADC アプライアンスにアクセスするには、CLI で次のコマンドを入力します。

```
ssh nsroot@<management IP address>
<!--NeedCopy-->
```

例:

```
ssh nsroot@192.168.4.5
<!--NeedCopy-->
```

11. ADC の設定は、`show ns ip`コマンドを使用して確認できます。

    ![`show nsip`コマンドを使って検証する] (/en-us/citrix-adc/media/avs-show-nsip.png)

Azure オートスケール設定を追加する

April 25, 2022

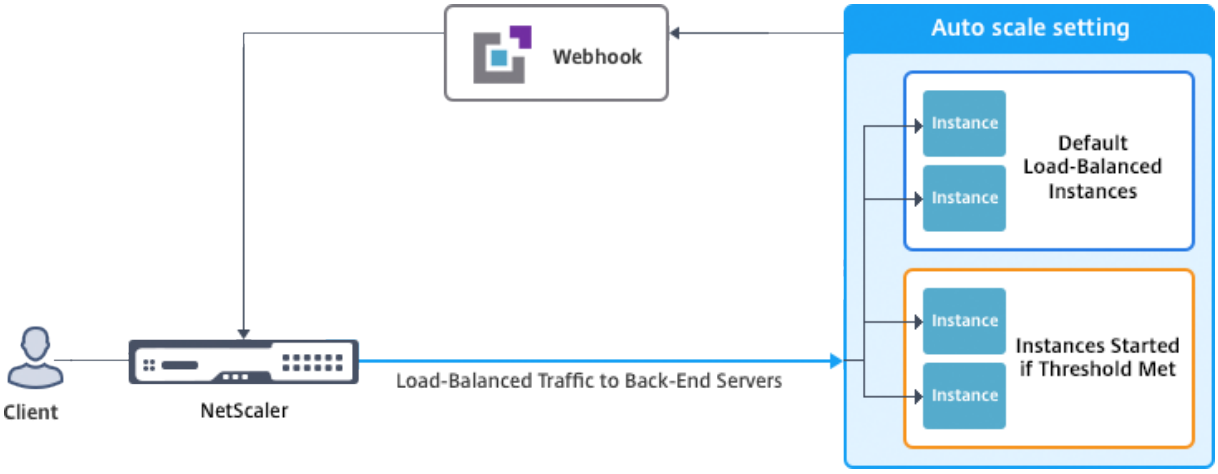
クラウドでアプリケーションを効率的にホストすると、アプリケーションの必要に応じて、リソースを簡単にコスト効率よく管理できます。需要の増大に対応するには、ネットワークリソースをスケールアップする必要があります。需要が低下するかどうかにかかわらず、アイドル状態のリソースの不要なコストを回避するためにスケールダウンする必要があります。アプリケーションを実行するコストを最小限に抑えるには、トラフィック、メモリ、CPU の使用率などを常に監視する必要があります。しかし、トラフィックを手動で監視するのは煩雑です。アプリケーション環

境を動的にスケールアップまたはスケールダウンするには、トラフィックの監視プロセスを自動化し、必要に応じてリソースを増減する必要があります。

Azure での VPX マルチ IP スタンドアロンおよび高可用性のデプロイには、Azure 仮想マシンスケールセット (VMSS) で Autoscale を使用できます。

Azure 仮想マシンスケールセット（VMSS）および自動スケール機能と統合され、Citrix ADC VPX インスタンスには次の利点があります。

- 負荷分散と管理: 需要に応じて、サーバーのスケールアップとスケールダウンを自動構成します。VPX インスタンスは、VPX インスタンスと同じリソースグループ内のバックエンドサブネット内の VMSS Autoscale 設定を自動検出し、ユーザーが VMSS Autoscale 設定を選択して負荷を分散できるようにします。このすべては、VPX インスタンスで Citrix ADC 仮想 IP アドレスとサブネット IP アドレスを自動構成することによって行われます。
- 高可用性: 同じリソースグループおよび負荷分散サーバー内の Autoscale グループを検出します。
- ネットワークの可用性の向上：VPX インスタンスは、異なる仮想ネットワーク（VNet）上のバックエンドサーバーをサポートします。



詳細については、次の Azure トピックを参照してください。

- 仮想マシンのスケールセットのドキュメント
- Microsoft Azure 仮想マシン、クラウドサービス、および Web アプリケーションの Autoscale の概要

はじめに

1. Azure 関連の使用に関するガイドラインを参照してください。詳細については、「Microsoft Azure での Citrix ADC VPX インスタンスの展開」を参照してください。
2. 要件（スタンドアロンまたは高可用性デプロイ）に応じて、Azure 上に 3 つのネットワークインターフェイスを使用して 1 つまたは複数の Citrix ADC VPX インスタンスを作成します。
3. VPX インスタンスの 0/1 インターフェイスのネットワークセキュリティグループで TCP 9001 ポートを開きます。VPX インスタンスはこのポートを使用して、スケールアウトおよびスケールイン通知を受信します。

4. 同じリソースグループに Azure 仮想マシンスケールセット (VMSS) を作成します。既存の VMSS 設定がない場合は、次のタスクを実行します。

   a）VMSS の作成

   b) VMSS でオートスケールを有効にする

   c) VMSS Autoscale 設定でスケールインおよびスケールアウトポリシーを作成する

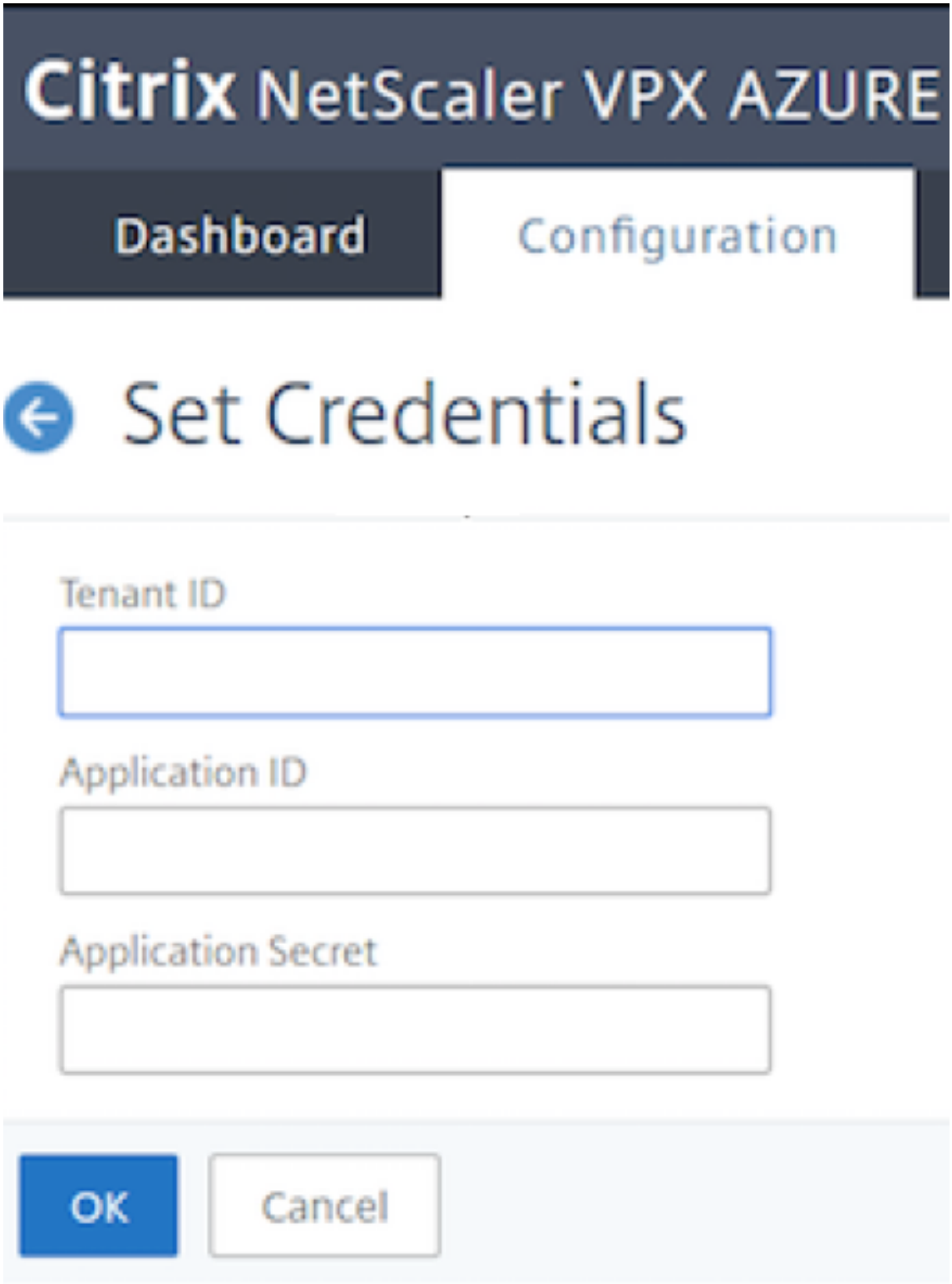
   詳細については、「Azure 仮想マシンのスケールセットを使用した Autoscale の概要」を参照してください。

5. リソースにアクセスできる AzureActive Directory（ADD）アプリケーションとサービスプリンシパルを作成します。新しく作成された AAD アプリケーションにコントリビュータロールを割り当てます。詳細については、「ポータルを使用してリソースにアクセスできる Azure Active Directory アプリケーションおよびサービスプリンシパルを作成する」を参照してください。

VMSS を **Citrix ADC VPX** インスタンスに追加する

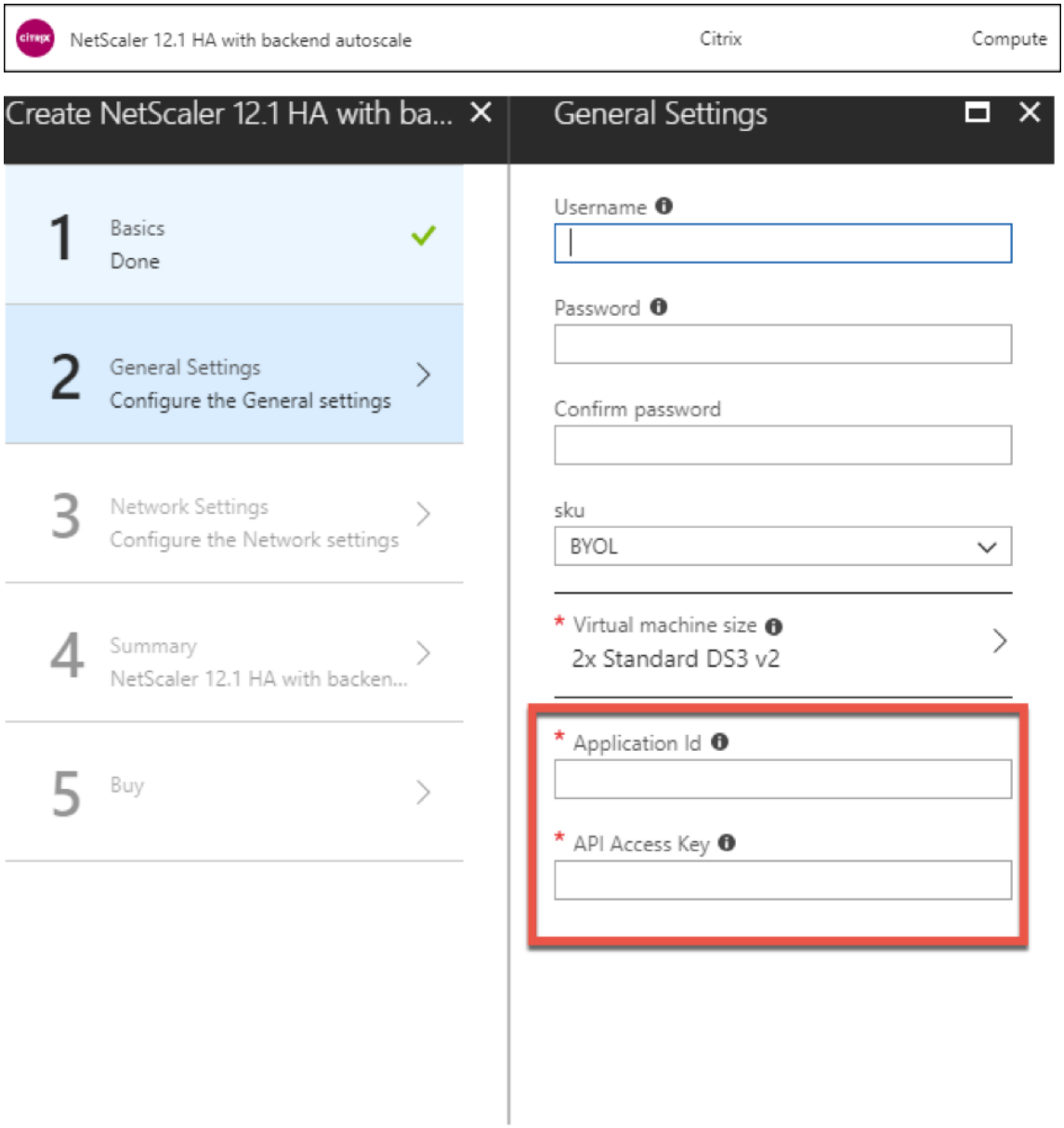
GUI を使用して、ワンクリックで VPX インスタンスに Autoscale 設定を追加できます。VPX インスタンスに Autoscale 設定を追加するには、次の手順を実行します。

1. VPX インスタンスにログオンします。
2. Citrix ADC VPX インスタンスに初めてログオンすると、[認証情報の設定] ページが表示されます。Autoscale 機能を機能させるために必要な Azure 認証情報を追加します。



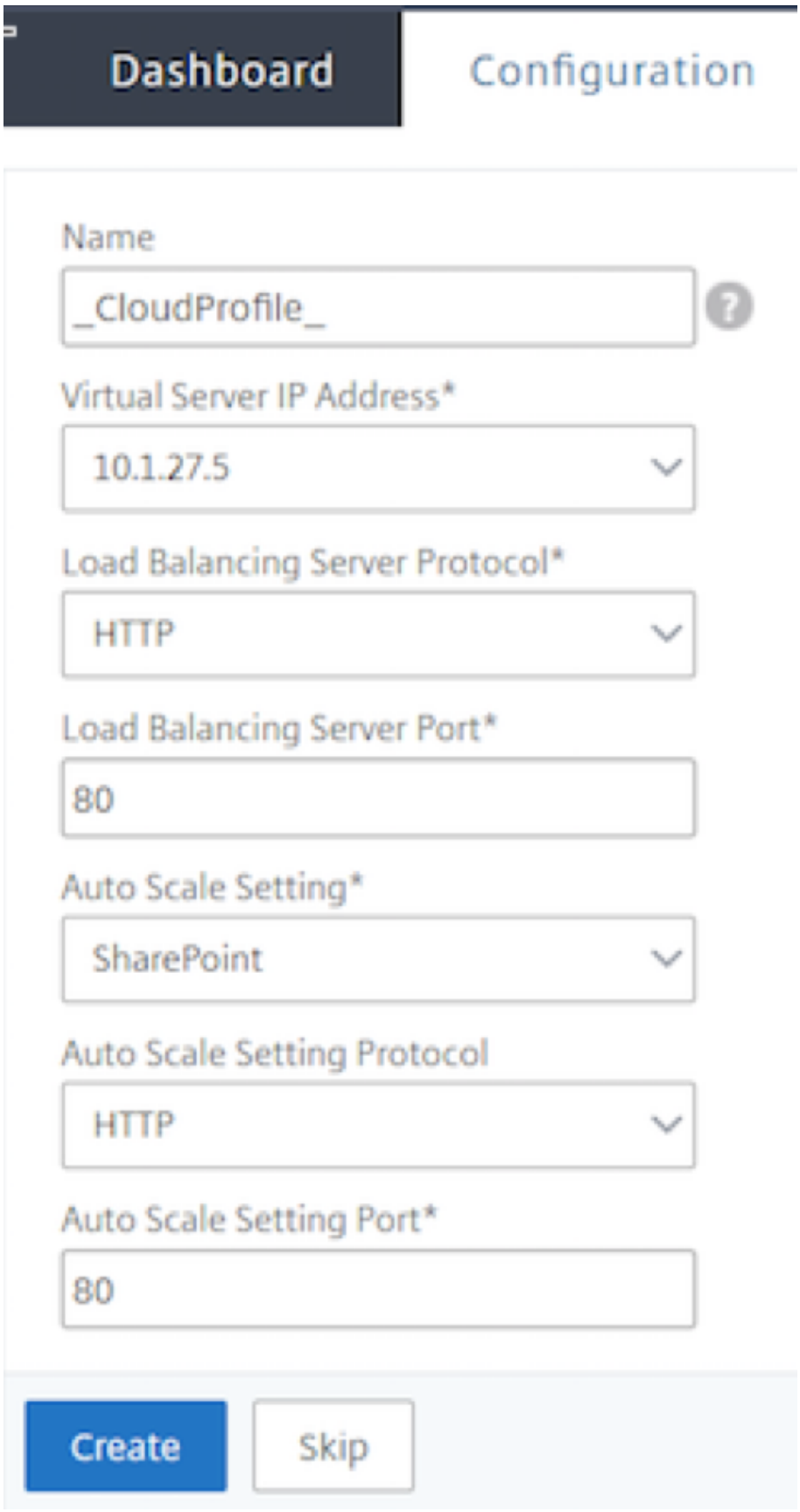
[認証情報の設定] ページが表示されるのは、アプリケーション ID と API アクセスキーが設定されていない場合や、正しいアプリケーション ID と API アクセスキー (アプリケーションシークレットと同じ) が Azure ポータルで設定されていない場合のみです。

Azure Marketplace から「バックエンド Autoscale を使用した NetScaler 12.1 HA」オファーを展開すると、Azure ポータルは Azure サービスプリンシパル認証情報（アプリケーション ID と API アクセスキー）の入力を求められます。



アプリケーション ID の作成方法については、「アプリケーションの追加」および「アクセスキーまたはアプリケーションシークレットの作成 」を参照してください。Web API にアクセスするようにクライアントアプリケーションを構成する」を参照してください。

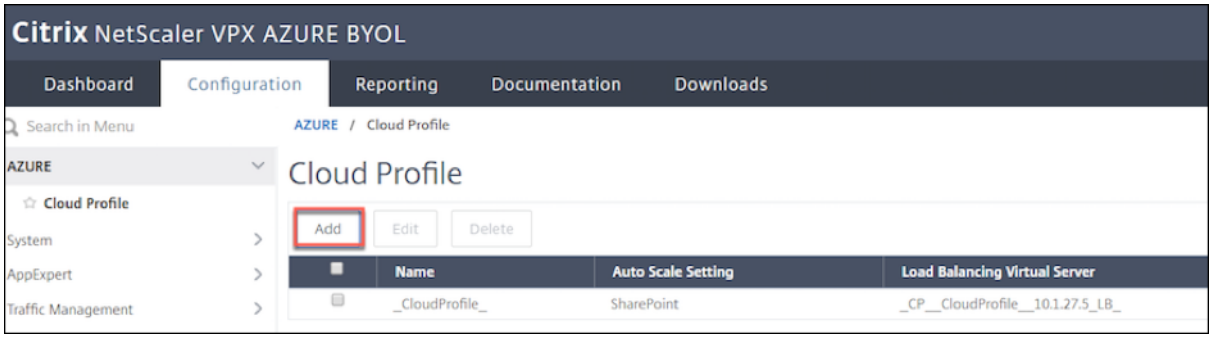
3. デフォルトのクラウドプロファイルページで、次の例に示すように詳細を入力し、[Create] をクリックします。



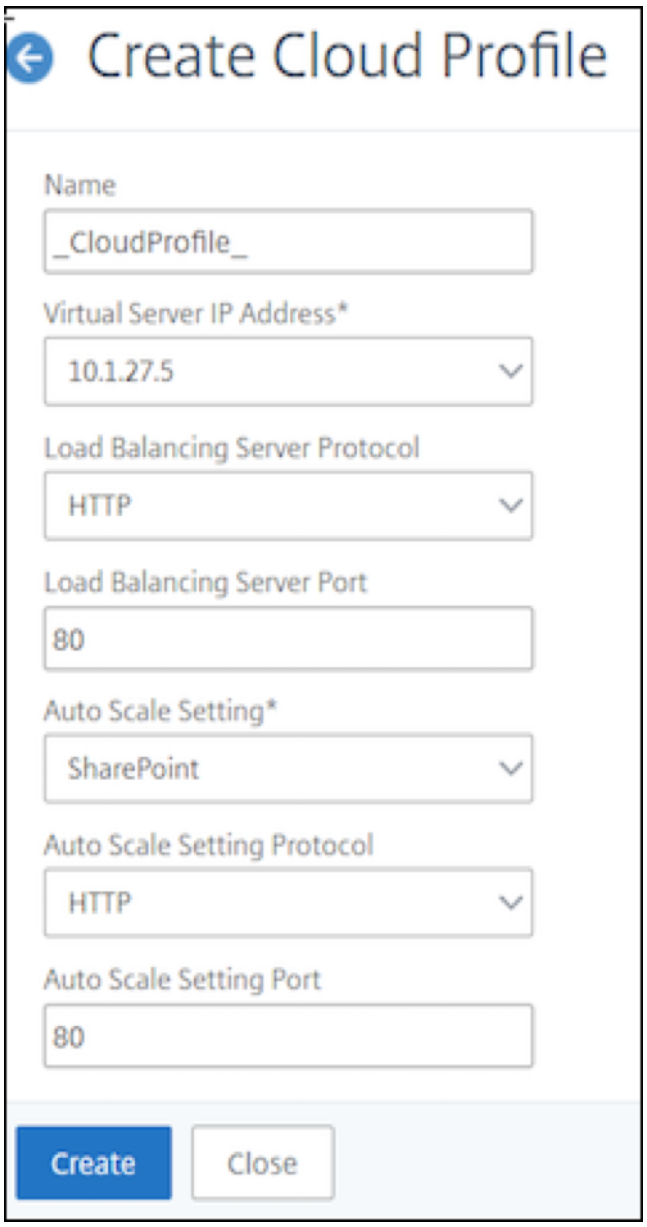
クラウドプロファイルの作成時に留意すべきポイント

- 仮想サーバーの IP アドレスは、VPX インスタンスで使用可能な空き IP アドレスから自動的に設定されます。詳細については、「Azure ポータルを使用して仮想マシンに複数の IP アドレスを割り当てる」を参照してください。
- Autoscale 設定は、Azure アカウントの現在のリソースグループで構成されている VMSS Autoscale 設定から事前に入力されます。詳細については、「Azure 仮想マシンのスケールセットを使用した Autoscale の概要」を参照してください。
- Auto Scaling Group のプロトコルとポートを選択するときに、サーバーがそれらのプロトコルとポートをリッスンし、サービスグループで正しいモニタをバインドしていることを確認します。デフォルトでは、TCP モニターが使用されます。
- SSL プロトコルタイプの Autos Scaling では、クラウドプロファイルを作成した後、証明書がないためにロードバランシング仮想サーバーまたはサービスグループがダウンします。証明書は、仮想サーバまたはサービスグループに手動でバインドできます。

初めてログオンした後、クラウドプロファイルを作成する場合は、GUI で [システム] > [Azure] > [クラウドプロファイル] の順に選択し、[追加] をクリックします。



[クラウドプロファイルの作成] 設定ページが表示されます。



クラウドプロファイルは、Citrix ADC 負荷分散（LB）仮想サーバー（仮想サーバー）と、Auto Scaling グループのサーバーとしてメンバー（サーバー）を持つサービスグループを作成します。バックエンドサーバーは、VPX インスタンスで構成された SNIP を介して到達可能である必要があります。

Azure Portal で Autoscale 関連の情報を表示するには、[すべてのサービス] > [仮想マシンスケールセット] > [仮想マシンスケールセットの選択] > [スケーリング] の順に選択します。

Citrix ADC VPX デプロイメント用の Azure タグ

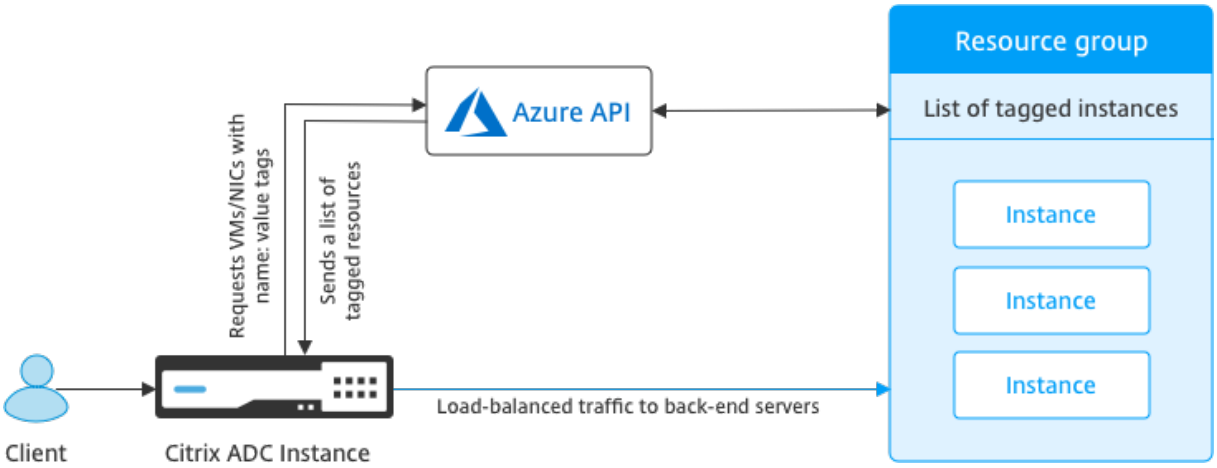
October 7, 2021

Azure クラウドポータルでは、名前: 値のペア (Dept: Finance など) でリソースにタグを付けて、リソースグループ間、およびポータル内でサブスクリプション間でリソースを分類して表示できます。タグ付けは、課金、管理、または自動化のためにリソースを整理する必要がある場合に役立ちます。

VPX デプロイメントにおける **Azure** タグの仕組み

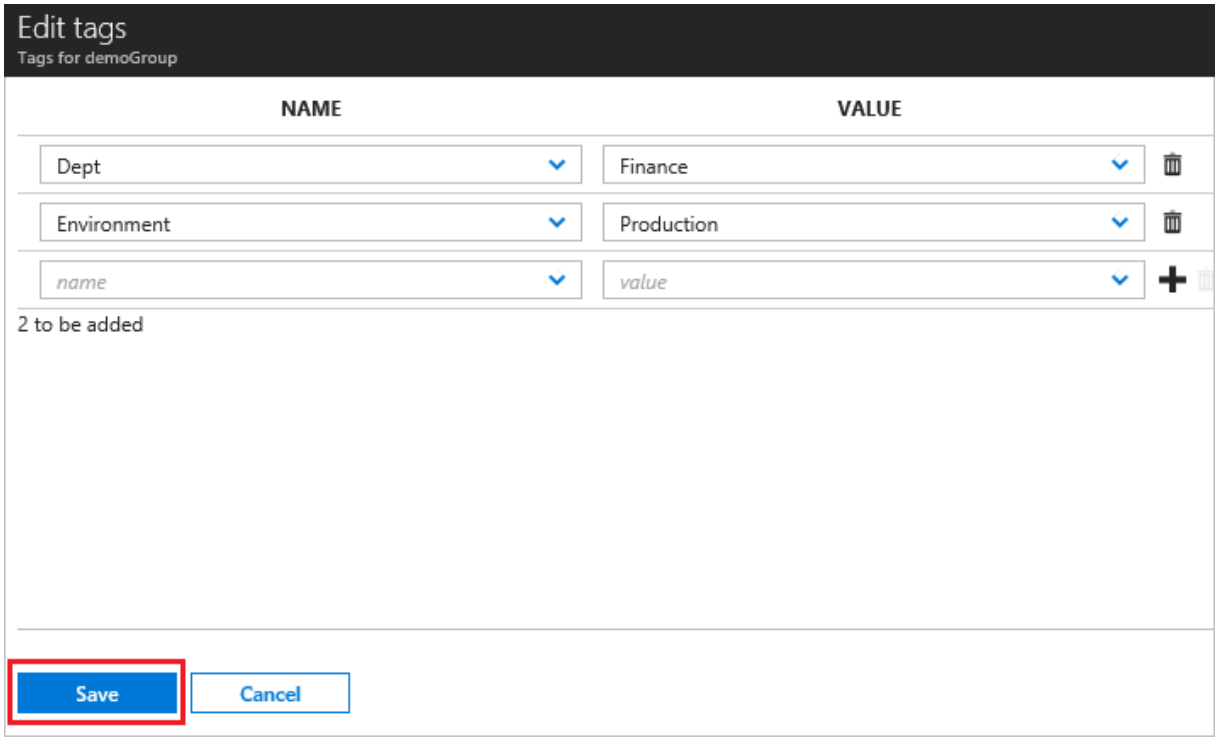
Azure Cloud にデプロイされた Citrix ADC VPX スタンドアロンおよび高可用性インスタンスで、Azure タグに関連付けられた負荷分散サービスグループを作成できるようになりました。VPX インスタンスは、Azure 仮想マシン（バックエンドサーバー）とネットワークインターフェイス（NIC）、またはその両方をそれぞれのタグで常に監視し、それに応じてサービスグループを更新します。

VPX インスタンスは、タグを使用してバックエンドサーバーの負荷分散を行うサービスグループを作成します。インスタンスは、特定のタグ名とタグ値でタグ付けされたすべてのリソースを Azure API に照会します。割り当てられたポーリング期間（デフォルトでは 60 秒）に応じて、VPX インスタンスは定期的に Azure API をポーリングし、VPX GUI で割り当てられたタグ名とタグ値で利用可能なリソースを取得します。適切なタグが付いた VM または NIC が追加または削除されるたびに、ADC はそれぞれの変更を検出し、サービスグループに VM または NIC の IP アドレスを自動的に追加または削除します。



はじめに

Citrix ADC 負荷分散サービスグループを作成する前に、Azure のサーバーにタグを追加します。タグは、仮想マシンまたは NIC に割り当てることができます。



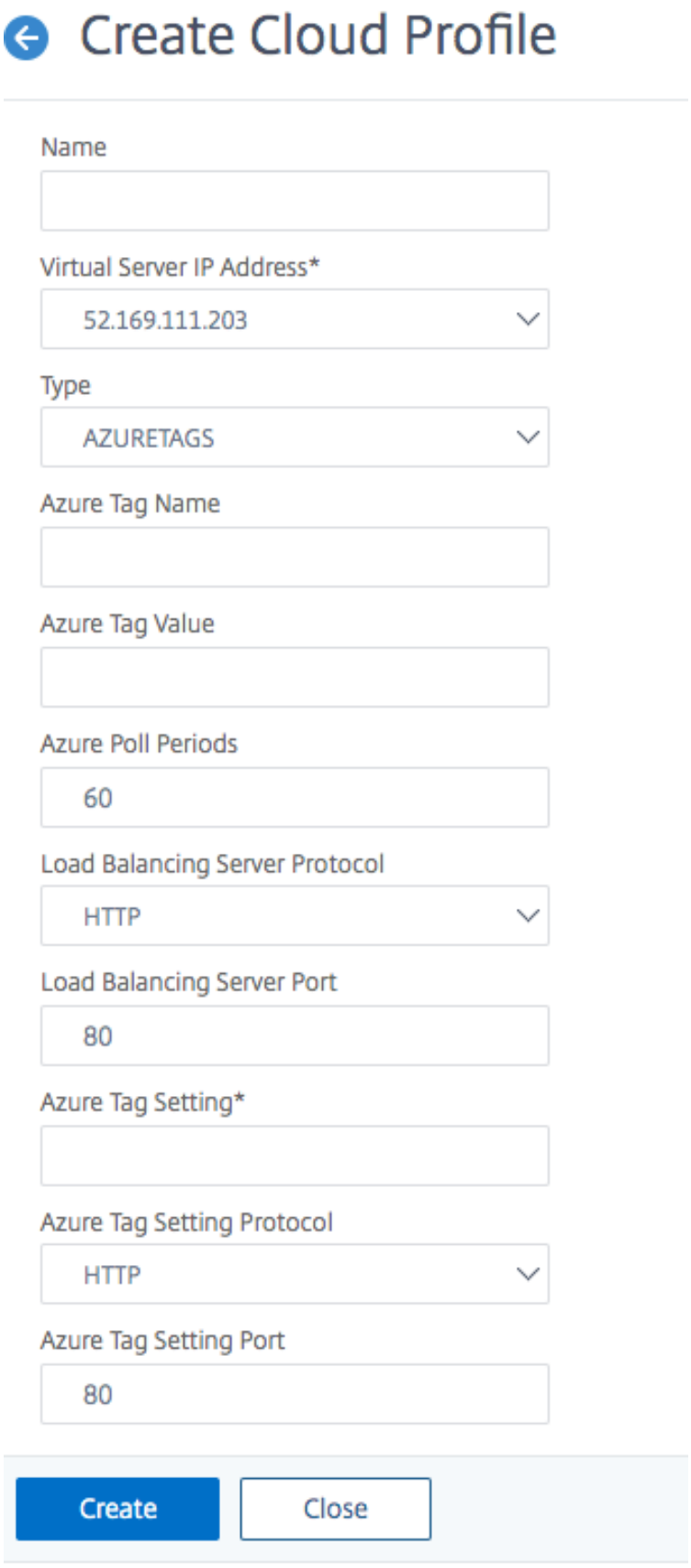
Azure タグの追加の詳細については、Microsoft ドキュメント「タグを使用して Azure リソースを整理する」を参照してください。

> 注:
> Azure タグ設定を追加する ADC CLI コマンドは、数字またはアルファベットのみで始まるタグ名とタグ値をサポートし、他のキーボード文字は使用できません。

VPX GUI を使用して **Azure** タグ設定を追加する方法

VPX GUI を使用して Azure タグクラウドプロファイルを VPX インスタンスに追加できます。これにより、インスタンスは指定されたタグを使用してバックエンドサーバーの負荷を分散できます。次の手順を実行します:

1. VPX GUI から、**[構成] > [Azure] > [**クラウドプロファイル] に移動します。
2. [追加] をクリックして、クラウドプロファイルを作成します。クラウドプロファイルウィンドウが開きます。



1. 次のフィールドに値を入力します。
   - 名前: プロフィールの名前を追加します。
   - 仮想サーバーの IP アドレス：仮想サーバーの IP アドレスは、VPX インスタンスで使用可能な空き IP アドレスから自動的に設定されます。詳細については、「Azure ポータルを使用して仮想マシンに複数の IP アドレスを割り当てる」を参照してください。
   - タイプ: メニューから「AZURETAGS」を選択します。
   - Azure タグ名:Azure ポータルで仮想マシンまたは NIC に割り当てた名前を入力します。
   - Azure タグ値:Azure ポータルの VM または NIC に割り当てた値を入力します。
   - Azure のポーリング期間: 既定では、ポーリング期間は 60 秒 (最小値) です。要件に応じて変更することができます。
   - [Load Balancing Server Protocol]: ロードバランサーがリッスンするプロトコルを選択します。
   - [負荷分散サーバーポート]: ロードバランサーがリッスンするポートを選択します。
   - Azure タグ設定: このクラウドプロファイル用に作成されるサービスグループの名前。
   - Azure タグ設定プロトコル: バックエンドサーバーがリッスンするプロトコルを選択します。
   - Azure タグ設定ポート: バックエンドサーバーがリッスンするポートを選択します。
2. ［作成］をクリックします。

タグ付けされた仮想マシンまたは NIC に対して、ロードバランサ仮想サーバとサービスグループが作成されます。ロードバランサー仮想サーバーを表示するには、VPX GUI から、トラフィック管理 **>** ロードバランシング **>** 仮想サーバーに移動します。

VPX CLI を使用して **Azure** タグ設定を追加する方法

Citrix ADC CLI で次のコマンドを入力して、Azure タグのクラウドプロファイルを作成します。

```
add cloud profile `<profile name>` -type azuretags -vServerName `<
    vserver name>` -serviceType HTTP -IPAddress `<vserver IP address>` -
    port 80 -serviceGroupName `<service group name>` -
    boundServiceGroupSvcType HTTP -vsvrbindsvcport 80 -azureTagName `<
    Azure tag specified on Azure portal>` -azureTagValue `<Azure value
    specified on the Azure portal>` -azurePollPeriod 60

<!--NeedCopy-->
```

> 重要
> すべての設定を保存する必要があります。保存しないと、インスタンスの再起動後に設定が失われます。「save config」と入力します。

例 **1**:「myTagName/myTagValue」ペアでタグ付けされたすべての Azure VM/NIC の HTTP トラフィック用のクラウドプロファイルのサンプルコマンドを次に示します。


```
add cloud profile MyTagCloudProfile -type azuretags -vServerName
    MyTagVServer -serviceType HTTP -IPAddress 40.115.116.57 -port 80 -
    serviceGroupName MyTagsServiceGroup -boundServiceGroupSvcType HTTP -
    vsvrbindsvcport 80 -azureTagName myTagName -azureTagValue myTagValue
     -azurePollPeriod 60
Done
<!--NeedCopy-->
```

クラウドプロファイルを表示するには、`show cloudprofile`と入力します。

例 **2**: 次の CLI コマンドは、例 1 で新しく追加されたクラウドプロファイルに関する情報を出力します。

```
show cloudprofile
1)    Name:  MyTagCloudProfile Type:  azuretags         VServerName:
    MyTagVServer ServiceType:  HTTP       IPAddress:  52.178.209.133
    Port:  80               ServiceGroupName:  MyTagsServiceGroup
    BoundServiceGroupSvcType:  HTTP
       Vsvrbindsvcport:  80     AzureTagName:  myTagName AzureTagValue:
           myTagValue AzurePollPeriod:  60     GraceFul:  NO
          Delay:  60
<!--NeedCopy-->
```

クラウドプロファイルを削除するには、「rm cloud profile <`cloud profile name`>」と入力します。

例 **3**: 次のコマンドは、例 1 で作成したクラウドプロファイルを削除します。

```
> rm cloudprofile MyTagCloudProfile
  Done
<!--NeedCopy-->
```

トラブルシューティング

問題: ごくまれに、「rm cloud profile」CLI コマンドで、削除されたクラウドプロファイルに関連付けられているサービスグループおよびサーバーの削除に失敗することがあります。これは、削除されるクラウドプロファイルのポーリング期間が経過する秒前にコマンドが発行された場合に発生します。

解決方法: 残りのサービスグループごとに次の CLI コマンドを入力して、残りのサービスグループを手動で削除します。


```
#> rm servicegroup <serviceGroupName>

<!--NeedCopy-->
```

残りの各サーバに対して次の CLI コマンドを入力して、残りのサーバもそれぞれ削除します。

```
#> rm server <name>
<!--NeedCopy-->
```

問題:CLI を使用して VPX インスタンスに Azure タグ設定を追加すると、ウォームリブート後も HA ペアノードで rain_tags プロセスが実行され続けます。

解決方法: ウォームリブート後に、セカンダリノードでプロセスを手動で終了します。セカンダリ HA ノードの CLI からシェルプロンプトに出ます。

```
#> shell

<!--NeedCopy-->
```

rain_tags プロセスを強制終了するには、次のコマンドを使用します。

```
# PID=`ps -aux | grep rain_tags | awk '{
 print $2 }
 '`; kill -9 $PID

<!--NeedCopy-->
```

問題: バックエンドサーバーは正常であるにもかかわらず、VPX インスタンスから到達できず、DOWN として報告されることがあります。
解決方法: VPX インスタンスが、バックエンドサーバーに対応するタグ付き IP アドレスに到達できることを確認します。タグ付きの NIC の場合、これは NIC の IP アドレスです。タグ付きの VM の場合、これは仮想マシンのプライマリ IP アドレスです。VM/NIC が別の Azure VNet 上に存在する場合は、VNet ピアリングが有効になっていることを確認します。

Citrix ADC VPX インスタンスで GSLB を構成する

October 7, 2021

グローバルサーバー負荷分散（GSLB）用に構成された Citrix ADC アプライアンスは、WAN の障害点から保護することにより、ディザスタリカバリとアプリケーションの継続的な可用性を提供します。GSLB は、クライアント要求を最も近い、または最もパフォーマンスの高いデータセンター、または停止が発生した場合に存続しているデータセンターに送信することにより、データセンター間で負荷を分散できます。

このセクションでは、Windows PowerShell コマンドを使用して、Microsoft Azure 環境の 2 つのサイトの VPX インスタンスで GSLB を有効にする方法について説明します。

> 注
>
> GSLB の詳細については、「グローバルサーバーの負荷分散」を参照してください。

Azure 上の Citrix ADC VPX インスタンスで GSLB を構成するには、次の 2 つの手順を実行します。

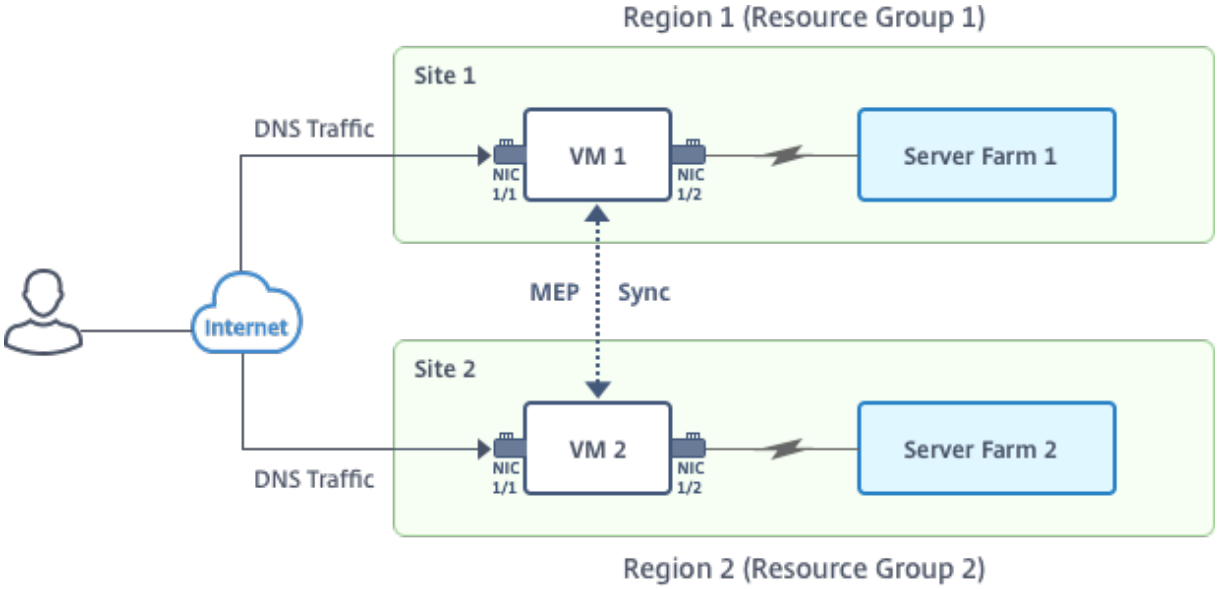
1. 各サイトに、複数の NIC と複数の IP アドレスを持つ VPX インスタンスを作成します。
2. VPX インスタンスで GSLB を有効にします。

> 注
>
> 複数の NIC および IP アドレスの構成の詳細については、「PowerShell コマンドを使用してスタンドアロンモードで Citrix ADC VPX インスタンスの複数の IP アドレスを構成する」を参照してください。

シナリオ

このシナリオには、2 つのサイト（Site 1 と Site 2）が含まれています。各サイトの VM（VM1 と VM2）には、複数の NIC、複数の IP アドレス、および GSLB が構成されています。

図。2 つのサイト（Site 1 と Site 2）に実装された GSLB セットアップ



このシナリオでは、各 VM には 3 つの NIC（NIC 0/1、1/1、1/2）が設定されています。各 NIC に複数のプライベートおよびパブリック IP アドレスを設定できます。これらの NIC は次の目的で構成されています。

- NIC 0/1：管理トラフィックを提供する

- NIC 1/1：クライアント側のトラフィックを提供する
- NIC 1/2：バックエンドサーバーと通信する

このシナリオで各 NIC に設定された IP アドレスの詳細については、「IP 構成の詳細」セクションを参照してください。

パラメーター

このドキュメントのこのシナリオのサンプルパラメーター設定は、次のとおりです。必要な場合は、異なる設定を使用できます。

```
$location="West Central US"

$vnetName="NSVPX-vnet"

$RGName="multiIP-RG"

$prmStorageAccountName="multiipstorageaccnt"

$avSetName="MultiIP-avset"

$vmSize="Standard_DS3_V2"
<!--NeedCopy-->
```

注: VPX インスタンスの最小要件は、2 つの vCPU と 2 GB RAM です。

```
$publisher="citrix"

$offer="netscalervpx111"

$sku="netscalerbyol"

$version="latest"

$vmNamePrefix="MultiIPVPX"

$nicNamePrefix="MultiipVPX"

$osDiskSuffix="osdiskdb"

$numberOfVMs=1

```



```
$ipAddressPrefix=”10.0.0.”

$ipAddressPrefix1=”10.0.1.”

$ipAddressPrefix2=”10.0.2.”

$pubIPName1=”MultiIP-pip1”

$pubIPName2=”MultiIP-pip2”

$IpConfigName1=”IPConfig1”

$IPConfigName2=”IPConfig-2”

$IPConfigName3=”IPConfig-3”

$IPConfigName4=”IPConfig-4”

$frontendSubnetName=”default”

$backendSubnetName1=”subnet_1”

$backendSubnetName2=”subnet_2”

$suffixNumber=10
<!--NeedCopy-->
```

仮想マシンの作成

PowerShell コマンドを使用して、ステップ 1 ～10 に従って、複数の NIC と複数の IP アドレスを使用して VM1 を作成します。

1. リソースグループの作成
2. ストレージアカウントの作成
3. 可用性セットの作成
4. 仮想ネットワークの作成
5. パブリック IP アドレスの作成
6. NIC の作成
7. VM 設定オブジェクトの作成
8. VM の資格情報の取得と OS プロパティの設定

9. NIC の追加
10. OS ディスクの指定と VM の作成

すべての手順とコマンドを完了して VM1 を作成した後で、これらの手順を繰り返して VM2 固有のパラメーターで VM2 を作成します。

リソースグループの作成

```
New-AzureRMResourceGroup -Name $RGName -Location $location
<!--NeedCopy-->
```

ストレージアカウントの作成

```
$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $RGName -Type Standard_LRS
     -Location $location
<!--NeedCopy-->
```

可用性セットの作成

```
$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $RGName -Location $location
<!--NeedCopy-->
```

仮想ネットワークの作成

1. サブネットを追加します。

```
$subnet1=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontendSubnetName -AddressPrefix ”10.0.0.0/24”
$subnet2=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName1 -AddressPrefix ”10.0.1.0/24”
$subnet3=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName2 -AddressPrefix ”10.0.2.0/24”
<!--NeedCopy-->
```

2. 仮想ネットワークオブジェクトを追加します。

```
$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
    $RGName -Location $location -AddressPrefix 10.0.0.0/16 -Subnet
    $subnet1, $subnet2, $subnet3
<!--NeedCopy-->
```

3. サブネットを取得します。

```
$frontendSubnet=$vnet.Subnets|?{
 $\_.Name -eq $frontendSubnetName }

$backendSubnet1=$vnet.Subnets|?{
 $\_.Name -eq $backendSubnetName1 }

$backendSubnet2=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName2 }

<!--NeedCopy-->
```

パブリック **IP** アドレスの作成

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
<!--NeedCopy-->
```

NIC の作成

NIC 0/1 の作成

```
$nic1Name=$nicNamePrefix + $suffixNumber + ”-Mgmnt”
$ipAddress1=$ipAddressPrefix + $suffixNumber
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    SubnetId $frontendSubnet.Id -PublicIpAddress $pip1 -PrivateIpAddress
     $ipAddress1 -Primary
$nic1=New-AzureRMNetworkInterface -Name $nic1Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig1
<!--NeedCopy-->
```


NIC 1/1 の作成

```
$nic2Name $nicNamePrefix + $suffixNumber + ”-frontend”
$ipAddress2=$ipAddressPrefix1 + ($suffixNumber)
$ipAddress3=$ipAddressPrefix1 + ($suffixNumber + 1)
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    PublicIpAddress $pip2 -SubnetId $backendSubnet1.Id  -
    PrivateIpAddress $ipAddress2  -Primary
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    SubnetId $backendSubnet1.Id  -PrivateIpAddress $ipAddress3
nic2=New-AzureRMNetworkInterface -Name $nic2Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig2, $IpConfig3
<!--NeedCopy-->
```

NIC 1/2 の作成

```
$nic3Name=$nicNamePrefix + $suffixNumber + ”-backend”
$ipAddress4=$ipAddressPrefix2 + ($suffixNumber)
$IPConfig4=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    SubnetId $backendSubnet2.Id -PrivateIpAddress $ipAddress4 -Primary
$nic3=New-AzureRMNetworkInterface -Name $nic3Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig4
<!--NeedCopy-->
```

VM 設定オブジェクトの作成

```
$vmName=$vmNamePrefix
$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id
<!--NeedCopy-->
```

資格情報の取得と **OS** プロパティの設定

```
$cred=Get-Credential -Message ”Type the name and password for VPX login
    .”
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version
```



```
<!--NeedCopy-->
```

NIC の追加

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
    Primary
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.Id
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id
<!--NeedCopy-->
```

OS ディスクの指定と **VM** の作成

```
$osDiskName=$vmName + "-" + $osDiskSuffix
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     +$osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $RGName -Location
    $location
<!--NeedCopy-->
```

> 注
>
> 「PowerShell コマンドを使用したマルチ NIC 仮想マシンの作成」に記載されている手順 1～10 を繰り返して、VM2 に固有のパラメータを使用して VM2 を作成します。

IP 構成の詳細

次の IP アドレスを使用します。

表 **1.** VM1 で使用する IP アドレス

NIC	プライベート IP	パブリック IP（PIP）	説明
0/1	10.0.0.10	PIP1	NSIP（管理 IP）として構成
1/1	1.10	PIP2	SNIP/GSLB サイト IP として設定

NIC	プライベート IP	パブリック IP（PIP）	説明
-	10.0.1.11	-	LB サーバ IP として設定。パブリック IP アドレスは必須ではありません
1/2	2.10	-	モニタプローブをサービスに送信するための SNIP として設定。パブリック IP は必須ではありません。

表 **2.** VM2 で使用する IP アドレス

NIC	内部 IP	パブリック IP（PIP）	説明
0/1	20.0.0.10	PIP4	NSIP（管理 IP）として構成
1/1	20.0.1.10	PIP5	SNIP/GSLB サイト IP として設定
-	20.0.1.11	-	LB サーバ IP として設定。パブリック IP アドレスは必須ではありません
1/2	20.0.2.10	-	モニタプローブをサービスに送信するための SNIP として設定。パブリック IP は必須ではありません。

VM1 および VM2 用の Citrix ADC VPX CLI で作成された IP アドレスと初期 LB 構成を示す、このシナリオの構成例を次に示します。

VM1 の設定例を次に示します。

```
1 add ns ip 10.0.1.10 255.255.255.0 -mgmtAccess ENABLED
2 Add nsip 10.0.2.10 255.255.255.0
3 add service svc1 10.0.1.10 ADNS 53
4 add lb vserver v1 HTTP 10.0.1.11 80
5 add service s1 10.0.2.120 http 80
```



```
Add service s2 10.0.2.121 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

VM2 の設定例を次に示します。

```
add ns ip 20.0.1.10 255.255.255.0  -mgmtAccess ENABLED
Add nsip 20.0.2.10 255.255.255.0
add service svc1 20.0.1.10 ADNS 53
add lb vserver v1 HTTP 20.0.1.11 80
Add service s1 20.0.2.90 http 80
Add service s2 20.0.2.91 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

GSLB サイトおよびその他の設定を構成する

次のトピックで説明するタスクを実行して、2 つの GSLB サイトおよびその他の必要な設定を構成します。

Global Server Load Balancing

詳細については、次のサポート記事https://support.citrix.com/article/CTX110348を参照してください。

VM1 および VM2 での GSLB 設定の例を次に示します。

```
enable ns feature LB GSLB
add gslb site site1 10.0.1.10 -publicIP PIP2
add gslb site site2 20.0.1.10 -publicIP PIP5
add gslb service site1_gslb_http_svc1 10.0.1.11 HTTP 80 -publicIP PIP3
    -publicPort 80 -siteName site1
add gslb service site2_gslb_http_svc1 20.0.1.11 HTTP 80 -publicIP PIP6
    -publicPort 80 -siteName site2
add gslb vserver gslb_http_vip1 HTTP
bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
<!--NeedCopy-->
```

Azure で実行されている Citrix ADC VPX インスタンスで GSLB を構成しました。

Citrix ADC VPX インスタンスで GSLB を構成する方法の詳細については、以下の画像をクリックして、Microsoft Azure での Citrix ADC GSLB の構成に関するビデオをご覧ください。

アクティブ/スタンバイの高可用性セットアップで **GSLB** を構成する

June 24, 2022

Azure のアクティブ/スタンバイ HA 展開には、次の 3 つの手順でグローバルサーバー負荷分散（GSLB）を構成できます。

1. 各 GSLB サイトで VPX HA ペアを作成します。HA ペアの作成方法については、「複数の IP アドレスと NIC を使用した高可用性設定の構成」を参照してください。
2. Azure Load Balancer（ALB）をフロントエンド IP アドレスと、GSLB および DNS トラフィックを許可する規則で構成します。

   この手順には、次の下位手順が含まれています。これらの下位手順の完了に使用する PowerShell コマンドについては、このセクションのシナリオを参照してください。

   a. GSLB サイトのフロントエンド`IPconfig`を作成します。

   b. HA 内のノードの NIC 1/1 の IP アドレスを持つバックエンドアドレスプールを作成します。

   c. 次のような負荷分散規則を作成します。

   ```
   TCP/3009  -  gslb communication
   TCP/3008 - gslb communication
   UDP/53 - DNS communication
   ```

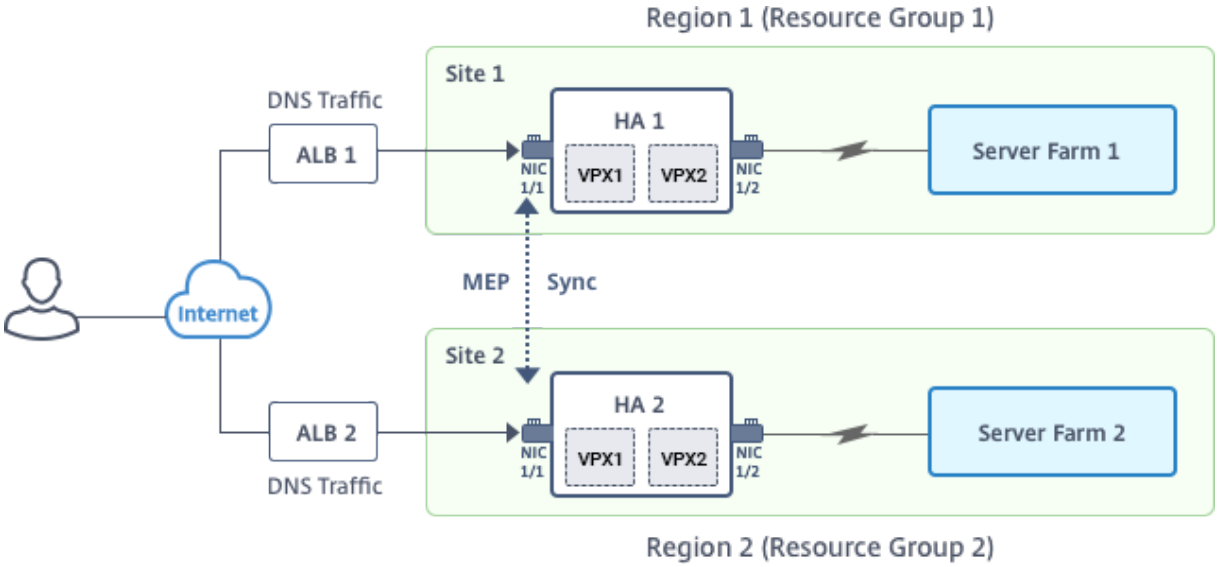
   d. バックエンドアドレスプールと手順 c で作成した LB 規則を関連付けます。

   e. 両方の HA ペアのノードの NIC 1/1 のネットワークセキュリティグループを更新して、TCP 3008、TCP 3009、および UDP 53 ポートのトラフィックを許可します。
3. 各 HA ペアで GSLB を有効にします。

シナリオ

このシナリオには、2 つのサイト（Site 1 と Site 2）が含まれています。各サイトの HA ペア（HA1 と HA2）には、複数の NIC、複数の IP アドレス、および GSLB が構成されています。

図: Azure でのアクティブ-スタンバイ HA デプロイメントでの GLSB



このシナリオでは、各 VM には 3 つの NIC（NIC 0/1、1/1、1/2）が設定されています。これらの NIC は次の目的で構成されています。

NIC 0/1：管理トラフィックを提供する

NIC 1/1：クライアント側のトラフィックを提供する

NIC 1/2：バックエンドサーバーと通信する

パラメーター設定

ALB のサンプルパラメーター設定は、次のとおりです。必要な場合は、異なる設定を使用できます。

```
$locName="South east Asia"

$rgName="MulitIP-MultiNIC-RG"

$pubIPName4="PIPFORGSLB1"

$domName4="vpxgslbdns"

$lbName="MultiIPALB"

$frontEndConfigName2="FrontEndIP2"

$backendPoolName1="BackendPoolHttp"

$lbRuleName2="LBRuleGSLB1"
```



```
16
17 $lbRuleName3=”LBRuleGSLB2”
18
19 $lbRuleName4=”LBRuleDNS”
20
21 $healthProbeName=”HealthProbe”
```

フロントエンド **IP** アドレスとルールを使用して **ALB** を構成し、**GSLB** と **DNS** トラフィックを許可する

手順 1：**GSLB** サイト **IP** 用のパブリック **IP** を作成する

```
1 $pip4=New-AzureRmPublicIpAddress -Name $pubIPName4 -ResourceGroupName
     $rgName -DomainNameLabel $domName4 -Location $locName -
     AllocationMethod Dynamic
2
3
4 Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName | Add-
     AzureRmLoadBalancerFrontendIpConfig -Name $frontEndConfigName2 -
     PublicIpAddress $pip4 | Set-AzureRmLoadBalancer
```

手順 2：**LB** ルールを作成し、既存の **ALB** を更新します。

```
1 $alb = get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName
2
3
4 $frontendipconfig2=Get-AzureRmLoadBalancerFrontendIpConfig -
     LoadBalancer $alb -Name $frontEndConfigName2
5
6
7 $backendPool=Get-AzureRmLoadBalancerBackendAddressPoolConfig -
     LoadBalancer $alb -Name $backendPoolName1
8
9
10 $healthprobe=Get-AzureRmLoadBalancerProbeConfig -LoadBalancer $alb -
     Name $healthProbeName
11
12
13 $alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName2 -
     BackendAddressPool $backendPool -FrontendIPConfiguration
     $frontendipconfig2 -Protocol ”Tcp” -FrontendPort 3009 -BackendPort
```



```
    3009 -Probe $healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer
14
15
16 $alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName3 -
    BackendAddressPool $backendPool -FrontendIPConfiguration
    $frontendipconfig2 -Protocol "Tcp" -FrontendPort 3008 -BackendPort
    3008 -Probe $healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer
17
18
19 $alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName4 -
    BackendAddressPool $backendPool -FrontendIPConfiguration
    $frontendipconfig2 -Protocol "Udp" -FrontendPort 53 -BackendPort 53
    -Probe $healthprobe -EnableFloatingIP  | Set-AzureRmLoadBalancer
```

各高可用性ペアで **GSLB** を有効にします

各 ALB（ALB 1 と ALB 2）で 2 つのフロントエンド IP アドレスを設定しました。1 つめの IP アドレスは LB 仮想サーバー、もう 1 つは GSLB サイトの IP です。

HA 1 には次のフロントエンド IP アドレスがあります。

- frontendiPOFalb1 (LB 仮想サーバー用)
- PIPFORGSLB1（GSLB IP）

HA 2 には次のフロントエンド IP アドレスがあります。

- frontendiPOFALB2 (LB 仮想サーバー用)
- PIPFORGSLB2（GSLB IP）

このシナリオでは、次のコマンドを使用します。

```
1  enable ns feature LB GSLB
2
3  add service dnssvc PIPFORGSLB1 ADNS 53
4
5  add gslb site site1 PIPFORGSLB1 -publicIP PIPFORGSLB1
6
7  add gslb site site2 PIPFORGSLB2 -publicIP PIPFORGSLB2
8
9  add gslb service site1_gslb_http_svc1 FrontEndIPofALB1 HTTP 80 -
    publicIP FrontEndIPofALB1 -publicPort 80 -siteName site1
10
```



```
add gslb service site2_gslb_http_svc1 FrontEndIPofALB2 HTTP 80 -
   publicIP FrontEndIPofALB2 -publicPort 80 -siteName site2

add gslb vserver gslb_http_vip1 HTTP

bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
```

関連リソース:

Citrix ADC VPX インスタンスで GSLB を構成する

Global Server Load Balancing

Citrix Gateway アプライアンスのアドレスプールのイントラネット IP を構成する

October 7, 2021

場合によっては、Citrix Gateway プラグインを使用して接続するユーザーは、Citrix ADC ゲートウェイアプライアンス用に一意の IP アドレスが必要です。グループのアドレスプール（IP プールとも呼ばれます）を有効にすると、Citrix Gateway アプライアンスは各ユーザーに一意の IP アドレスエイリアスを割り当てることができます。アドレスプールは、イントラネット IP (IIP) アドレスを使用して構成します。

Azure にデプロイされた Citrix Gateway アプライアンスでアドレスプールを構成するには、次の 2 ステップの手順に従います。

- アドレスプールで使用されるプライベート IP アドレスを Azure に登録する
- Citrix Gateway アプライアンスでのアドレスプールの構成

Azure ポータルにプライベート IP アドレスを登録する

Azure では、複数の IP アドレスを持つ Citrix ADC VPX インスタンスをデプロイできます。次の 2 つの方法で IP アドレスを VPX インスタンスに追加できます。

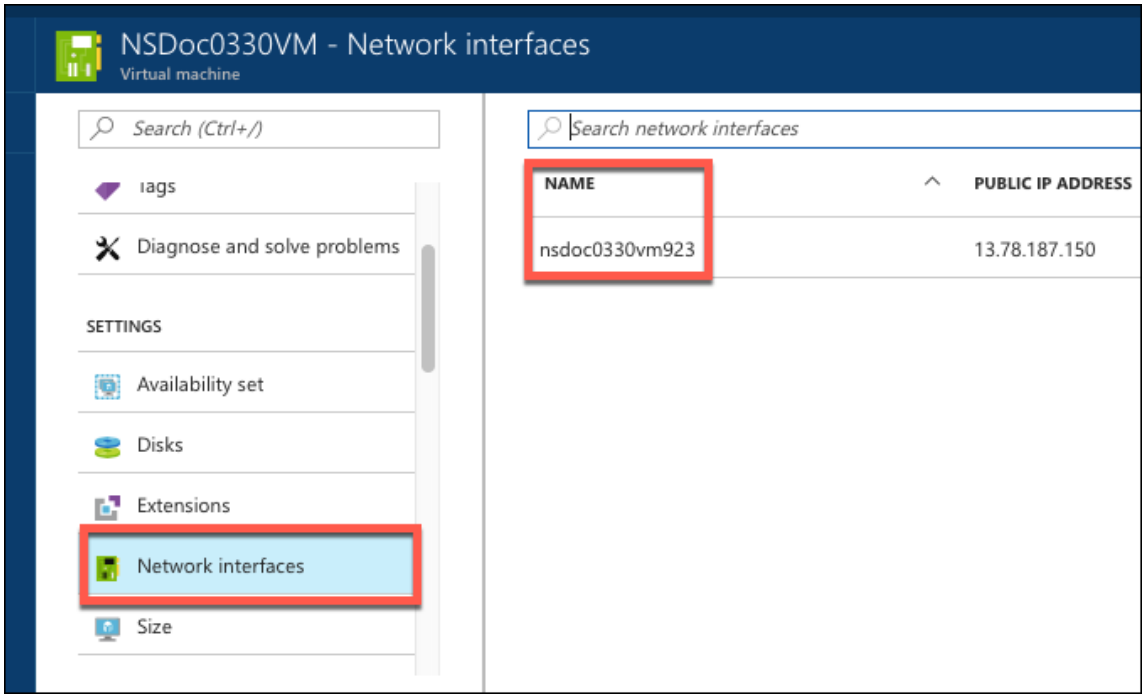
a. VPX インスタンスの Provisioning 中

VPX インスタンスのプロビジョニング中に複数の IP アドレスを追加する方法の詳細については、「Citrix ADC スタンドアロンインスタンスの複数の IP アドレスを構成する」を参照してください。VPX インスタンスのプロビジョニング中に PowerShell コマンドを使用して IP アドレスを追加するには、PowerShell コマンドを使用してスタンドアロンモードで Citrix ADC VPX インスタンスの複数の IP アドレスを構成するを参照してください。

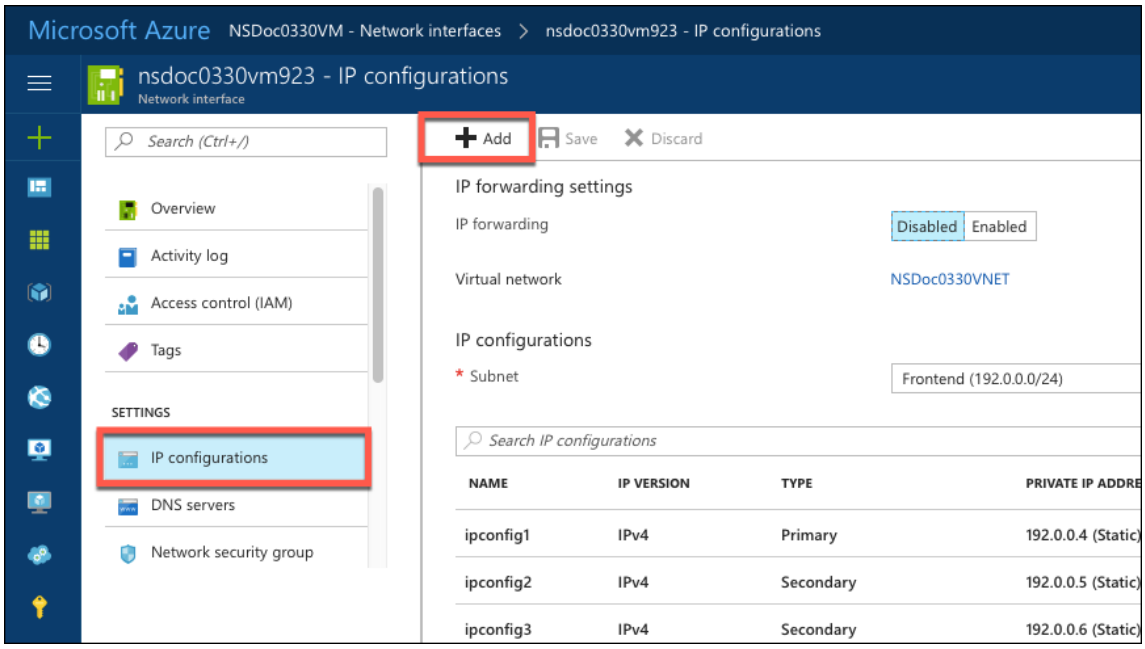
b. VPX インスタンスをプロビジョニング後

VPX インスタンスをプロビジョニングしたら、以下の手順に従って Azure ポータルにプライベート IP アドレスを登録します。プライベート IP アドレスを Citrix Gateway アプライアンスのアドレスプールとして構成します。

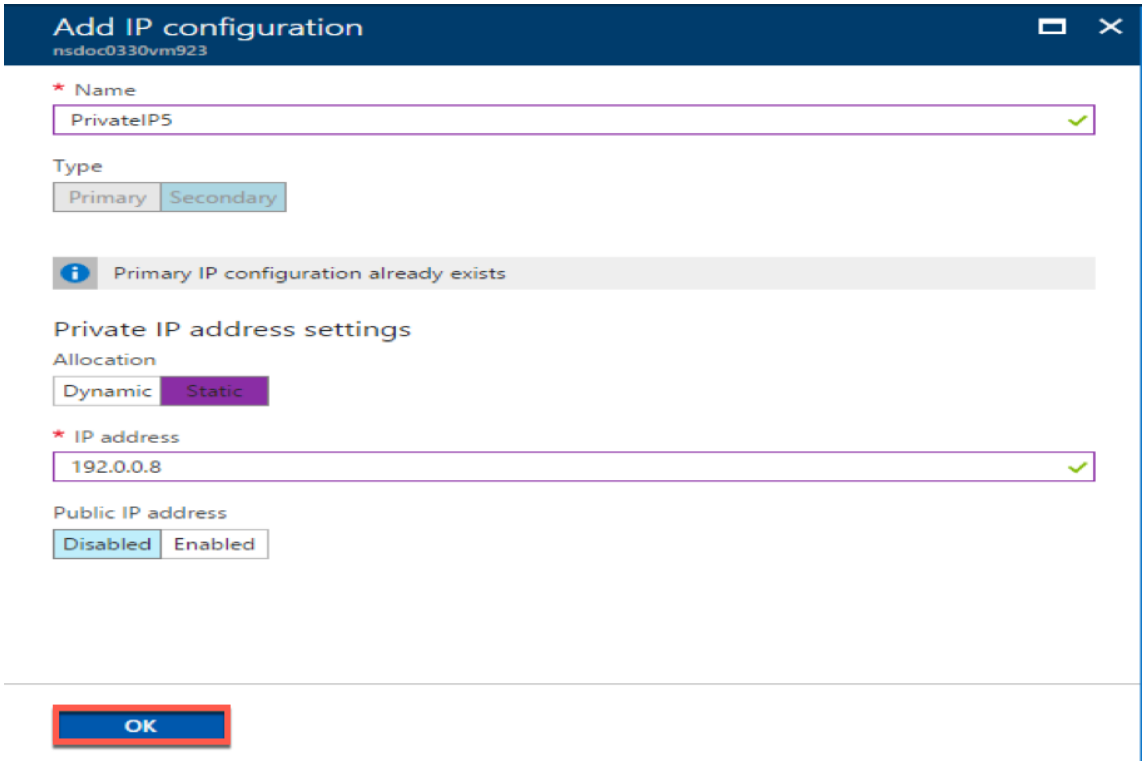
1. Azure Resource Manager（ARM）から、すでに作成された Citrix ADC VPX インスタンス > ネットワークインターフェイスに移動します。登録する IIP が属しているサブネットにバインドされているネットワークインターフェイスを選択します。



2. [**IP** 構成] をクリックし、[追加] をクリックします。



3. 以下の例のように必要な詳細を入力し、[**OK**] をクリックします。



Citrix Gateway アプライアンスでアドレスプールを構成する

Citrix Gateway でアドレスプールを構成する方法の詳細については、「アドレスプールの構成」を参照してください。

制限:IIP アドレスの範囲をユーザーにバインドすることはできません。アドレスプールで使用されるすべての IIP アドレスを登録する必要があります。

PowerShell コマンドを使用して、**Citrix ADC VPX** スタンドアロンインスタンスの複数の **IP** アドレスを構成する

October 7, 2021

Azure 環境では、Citrix ADC VPX 仮想アプライアンスを複数の NIC で展開できます。各 NIC に複数の IP アドレスを設定できます。このセクションでは、PowerShell コマンドを使用して、単一の NIC と複数の IP アドレスを使用して Citrix ADC VPX インスタンスを展開する方法について説明します。複数 NIC と複数 IP の展開にも同じスクリプトを使用できます。

> 注
>
> このドキュメントでは、IP-config は、個々の NIC に関連付けられている IP アドレス、パブリック IP、プライ

> ベート IP のペアを指します。詳細については、Azure の用語のセクションを参照してください 。

使用例

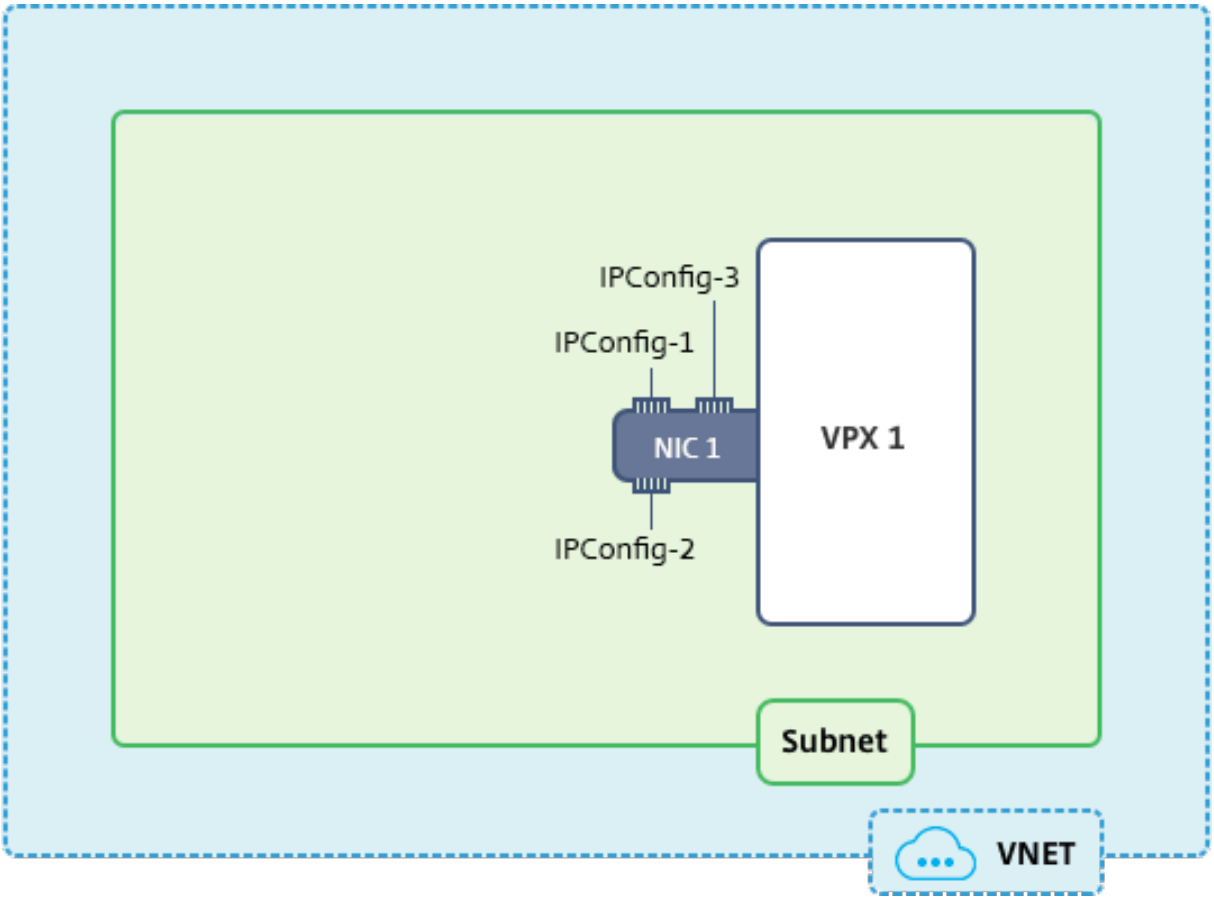
この使用例では、1 つの NIC が仮想ネットワーク（VNET）に接続されています。この NIC には、次の表に示す 3 つの IP 構成が関連付けられています。

IP コンフィグ	関連付けられている
IPConfig-1	静的パブリック IP アドレス; 静的プライベート IP アドレス
IPConfig-2	静的パブリック IP アドレス; 静的プライベートアドレス
IPConfig-3	静的プライベート IP アドレス

> 注
>
> IPConfig-3 は、パブリック IP アドレスに関連付けられていません。

図: トポロジ

次の図はこの使用例を視覚的に示しています。



> 注
>
> マルチ NIC、マルチ IP Azure Citrix ADC VPX 展開では、プライマリ（最初の）NIC のプライマリ（最初）`IPConfig`に関連付けられたプライベート IP アドレスが、アプライアンスの管理 NSIP アドレスとして自動的に追加されます。`IPConfigs`に関連付けられた残りのプライベート IP アドレスは、要件に応じて`add ns ip`コマンドを使用して、VPX インスタンスに VIP または SNIP として追加する必要があります。

スタンドアロンモードで Citrix ADC VPX 仮想アプライアンスの複数の IP アドレスを構成するために必要な手順の概要を次に示します。

1. リソースグループの作成
2. ストレージアカウントの作成
3. 可用性セットの作成
4. ネットワークサービスグループの作成
5. 仮想ネットワークの作成
6. パブリック IP アドレスの作成
7. IP 構成の割り当て
8. NIC の作成
9. Citrix ADC VPX インスタンスの作成
10. NIC 構成のチェック

11. VPX 側の構成のチェック

スクリプト

パラメーター

このドキュメントのこの使用例のサンプルパラメーター設定は、次のとおりです。必要な場合は、異なる設定を使用できます。

$locName=”westcentralus”

$rgName=”Azure-MultiIP”

$nicName1=”VM1-NIC1”

$vNetName=”Azure-MultiIP-vnet”

$vNetAddressRange=”11.6.0.0/16”

$frontEndSubnetName=”frontEndSubnet”

$frontEndSubnetRange=”11.6.1.0/24”

$prmStorageAccountName=”multiipstorage”

$avSetName=”multiip-avSet”

$vmsize= “Standard_DS4_v2” (このパラメータは最大 4 つの NIC を持つ仮想マシンを作成します。)

注**:** VPX インスタンスの最小要件は、2 つの vCPU と 2 GB RAM です。

$ パブリッシャー =「Citrix」

$offer=”netscalervpx110-6531”（異なる offer を使用できます）

$sku=”netscalerbyol”（offer によって、異なる SKU にすることができます）

$version=”latest”

$pubIPName1=”PIP1”

$pubIPName2=”PIP2”

$domName1=”multiipvpx1”

$domName2=”multiipvpx2”

$vmNamePrefix=”VPXMultiIP”

$osDiskSuffix=”osmultiipalbdiskdb1”

ネットワークセキュリティグループ **(NSG)** 関連情報:

$nsgName=”NSG-MultiIP”

$rule1Name=”Inbound-HTTP”

$rule2Name=”Inbound-HTTPS”

$rule3Name=”Inbound-SSH”

$IpConfigName1=”IPConfig1”

$IPConfigName2=”IPConfig-2”

$IPConfigName3=”IPConfig-3”

1. リソースグループの作成

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2. ストレージアカウントの作成

```
$prmStorageAccount = New-AzureRMStorageAccount -Name $prmStorageAccountName
 -ResourceGroupName $rgName -Type Standard_LRS -Location $locName
```

3. 可用性セットの作成

```
$avSet = New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
$rgName -Location $locName
```

4. ネットワークセキュリティグループの作成

1. 規則を追加します。トラフィックを処理するポートのネットワークセキュリティグループにルールを追加する必要があります。

   ```
   $rule1=New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -Description
    ”Allow HTTP”-Access Allow -Protocol Tcp -Direction Inbound -Priority
   101 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
    * -DestinationPortRange 80
   $rule2=New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -Description
    ”Allow HTTPS”-Access Allow -Protocol Tcp -Direction Inbound -Priority
   110 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
    * -DestinationPortRange 443
   $rule3=New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -Description
    ”Allow SSH”-Access Allow -Protocol Tcp -Direction Inbound -Priority
   120 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
    * -DestinationPortRange 22
   ```

2. ネットワークセキュリティグループオブジェクトを作成します。


```
$nsg=New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

5. 仮想ネットワークの作成

1. サブネットを追加します。

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $frontEndSubnetName
 -AddressPrefix $frontEndSubnetRange
```

2. 仮想ネットワークオブジェクトを追加します。

```
$vnet=New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
$rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
$frontendSubnet
```

3. サブネットを取得します。

```
$subnetName="frontEndSubnet"
$subnet1=$vnet.Subnets|?{ $_.Name -eq $subnetName }
```

6. パブリック **IP** アドレスの作成

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
$rgName -DomainNameLabel $domName1 -Location $locName -AllocationMethod
Static
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
$rgName -DomainNameLabel $domName2 -Location $locName -AllocationMethod
Static
```

> 注
>
> 使用する前にドメイン名の可用性をチェックします。
>
> IP アドレスの割り当て方法は動的または静的にできます。

7. IP 構成の割り当て

この使用例では、IP アドレスを割り当てる前に次の点を検討します。

- IPConfig-1 が VPX1 の subnet1 に属していること
- IPConfig-2 が VPX1 の subnet 1 に属していること
- IPConfig-3 が VPX1 の subnet 1 に属していること

> 注
>
> 複数の IP 構成を 1 つの NIC に割り当てるときには、1 つの構成をプライマリとして割り当てる必要があります。

```
$IPAddress1=”11.6.1.27”
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress1 -PublicIpAddress $pip1
     -Primary
$IPAddress2=”11.6.1.28”
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress2 -PublicIpAddress $pip2
$IPAddress3=”11.6.1.29”
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress3 -Primary
```

サブネットの要件に合う有効な IP アドレスを使用して、その可用性をチェックします。

8. NIC の作成

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
$rgName -Location $locName -IpConfiguration $IpConfig1,$IpConfig2,$IPConfig3
 -NetworkSecurityGroupId $nsg.Id
```

9. Citrix ADC VPX インスタンスの作成

1. 変数を初期化します。

   ```
   $suffixNumber = 1
   $vmName = $vmNamePrefix + $suffixNumber
   ```

2. VM Config オブジェクトを作成します。

   ```
   $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
    $avSet.Id
   ```

3. 資格情報、OS、イメージを設定します。

   ```
   $cred=Get-Credential -Message ”Type the name and password for VPX login
   .”
   $vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -ComputerName
    $vmName -Credential $cred
   $vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName $publisher
    -Offer $offer -Skus $sku -Version $version
   ```


4. NIC を追加します。

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
Primary
```

> 注
>
> マルチ NIC VPX 展開では、1 つの NIC がプライマリである必要があります。したがって、その NIC を VPX インスタンスに追加するときに「-Primary」を追加する必要があります。

5. OS ディスクを指定して、VM を作成します。

```
$osDiskName=$vmName + "-"+ $osDiskSuffix1
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString()+ "vhds/"+
 $osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
$osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
$locName
```

10. NIC 構成のチェック

VPX インスタンスの起動後、次のコマンドを使用して VPX NIC の IPConfigs に割り当てられている IP アドレスを確認できます。

```
$nic.IPConfig
```

11. VPX 側の構成のチェック

Citrix ADC VPX インスタンスが起動すると、IPconfig プライマリ NIC のプライマリに関連付けられたプライベート IP アドレスが NSIP アドレスとして追加されます。残りのプライベート IP アドレスは、要件に従って、VIP または SNIP アドレスとして追加する必要があります。次のコマンドを使用します。

```
add nsip <Private IPAddress><netmask> -type VIP/SNIP
```

スタンドアロンモードで Citrix ADC VPX インスタンスの複数の IP アドレスを構成しました。

Azure デプロイ用の追加の PowerShell スクリプト

October 13, 2021

このセクションでは、Azure PowerShell で次の構成を実行できる PowerShell コマンドレットについて説明します。

- Citrix ADC VPX スタンドアロンインスタンスのプロビジョニング
- Azure 外部ロードバランサーを使用して高可用性セットアップで Citrix ADC VPX ペアをプロビジョニングする
- Azure 内部ロードバランサーを使用した高可用性セットアップでの Citrix ADC VPX ペアのプロビジョニング

PowerShell コマンドを使用して実行できる構成については、次のトピックも参照してください。

- PowerShell コマンドを使用して複数の IP アドレスと NIC を使用して高可用性セットアップを構成する
- Citrix ADC VPX インスタンスで GSLB を構成する
- NetScaler アクティブスタンバイ高可用性セットアップで GSLB を構成する
- PowerShell コマンドを使用して、スタンドアロンモードで Citrix ADC VPX インスタンスの複数の IP アドレスを構成する
- スタンドアロン VPX インスタンス用に複数の Azure VIP を構成する

Citrix ADC VPX スタンドアロンインスタンスのプロビジョニング

1. リソースグループの作成

   リソースグループには、ソリューションのすべてのリソースを含めることも、グループとして管理するリソースのみを含めることもできます。ここで指定した場所は、そのリソースグループ内のリソースの既定の場所です。ロードバランサーを作成する場合、すべてのコマンドで同じリソースグループを使用してください。

   ```
   $rgName="<resource group name>"
   $locName="<location name, such as West US>
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   たとえば、

   ```
   $rgName = "ARM-VPX"
   $locName = "West US"
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. ストレージアカウントの作成

   ストレージアカウントには、小文字と数字のみを含む一意の名前を選択する必要があります。

   ```
   $saName="<storage account name>"
   $saType="<storage account type>"で次のいずれかを指定します: Standard_LRS、Standard_GRS、Standard_RAGRS、またはPremium_LRS
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
   ```


たとえば、

```
$saName=”vpxstorage”
$saType=”Standard_LRS”
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. 可用性セットの作成

   可用性セットにより、メンテナンス時などのダウンタイム中でも仮想マシンを使用し続けることができます。可用性セットが構成されたロードバランサーでは、アプリケーションをいつでも使用できます。

   ```
   $avName=”<availability set name>”

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -
   Location $locName
   ```

4. 仮想ネットワークを作成する

   以前に作成されたサブネットがない場合、少なくとも 1 つのサブネットを持つ新しい仮想ネットワークを追加します。

   ```
   $FrontendAddressPrefix=”10.0.1.0/24”
   $BackendAddressPrefix=”10.0.2.0/24”
   $vnetAddressPrefix=”10.0.0.0/16”
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet
    -AddressPrefix $FrontendAddressPrefix
   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet
    -AddressPrefix $BackendAddressPrefix
   New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName -
   Location $locName -AddressPrefix $vnetAddressPrefix -Subnet $frontendSubnet
   ,$backendSubnet
   ```

   たとえば、

   ```
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       frontendSubnet -AddressPrefix $FrontendAddressPrefix

   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       backendSubnet -AddressPrefix $BackendAddressPrefix

   ```



```
New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName
     -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet
    $frontendSubnet,$backendSubnet
<!--NeedCopy-->
```

5. **NIC** を作成する

   NIC を作成し、その NIC を Citrix ADC VPX インスタンスに関連付けます。上記の手順で作成されたフロントエンドサブネットは 0 でインデックス付けされ、バックエンドサブネットは 1 でインデックス付けされます。次の 3 つのいずれかの方法で NIC を作成します。

   a) パブリック *IP* アドレスを持つ *NIC*

   ```
   $nicName="<name of the NIC of the VM>"

   $pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
   $rgName -Location $locName -AllocationMethod Dynamic

   $nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
   $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
   PublicIpAddressId $pip.Id
   ```

   b）パブリック *IP* および *DNS* ラベルを持つ *NIC*

   ```
   $nicName="<name of the NIC of the VM>"

   $domName="<domain name label>"

   $pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
   $rgName -DomainNameLabel $domName -Location $locName -AllocationMethod
   Dynamic
   ```

   $domName を割り当てる前に、次のコマンドを使用して、それが利用できるかどうかを確認します。

   ```
   Test-AzureRmDnsAvailability -DomainQualifiedName $domName -Location
   $locName

   $nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
   $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
   PublicIpAddressId $pip.Id
   ```

   たとえば、

   ```
   $nicName="frontendNIC"

   $domName="vpxazure"

   ```



```
5 $pip = New-AzureRmPublicIpAddress -Name $nicName -
      ResourceGroupName $rgName -DomainNameLabel $domName -Location
      $locName -AllocationMethod Dynamic
6
7 $nic = New-AzureRmNetworkInterface -Name $nicName -
      ResourceGroupName $rgName -Location $locName -SubnetId $vnet.
      Subnets[0].Id -PublicIpAddressId $pip.Id
8 <!--NeedCopy-->
```

c) 動的パブリックアドレスと静的プライベート *IP* アドレスを持つ *NIC*

仮想マシンに追加するプライベート（静的）IP アドレスが、指定したサブネットのアドレスと同じ範囲である必要があります。

```
$nicName="<name of the NIC of the VM>"

$staticIP="<available static IP address on the subnet>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
$rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
$rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
PublicIpAddressId $pip.Id -PrivateIpAddress $staticIP
```

6. 仮想オブジェクトの作成

```
$vmName="<VM name>"

$vmSize="<VM size string>"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
$rgName

$vm=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
 $avset.Id
```

7. **Citrix ADC VPX** イメージの取得

```
$pubName="<Image publisher name>"

$offerName="<Image offer name>"

$skuName="<Image SKU name>"

$cred=Get-Credential -Message "Type the name and password of the local
administrator account."
```

VPX へのログインに使用する認証情報を提供する

```
$vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName $vmName -
Credential $cred -Verbose
```



```
$vm=Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -Offer
$offerName -Skus $skuName -Version ”latest”

$vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id
```

たとえば、

```
$pubName=”citrix”
```

次のコマンドを使用すると、Citrix からのすべてのオファーが表示されます。

```
1 Get-AzureRMVMImageOffer -Location $locName -Publisher $pubName |
     Select Offer
2
3 $offerName=”netscalervpx110-6531”
4 <!--NeedCopy-->
```

次のコマンドは、特定のオファー名についてパブリッシャーから提供される SKU を知るために使用します。

```
Get-AzureRMVMImageSku -Location $locName -Publisher $pubName -Offer
$offerName | Select Skus
```

8. 仮想マシンの作成

```
$diskName=”<name identifier for the disk in Azure storage, such as
OSDisk>”
```

たとえば、

```
1  $diskName=”dynamic”
2
3  $pubName=”citrix”
4
5  $offerName=”netscalervpx110-6531”
6
7  $skuName=”netscalerbyol”
8
9  $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
       Name $saName
10
11 $osDiskUri=$storageAcc.PrimaryEndpoints.Blob.ToString() + ”vhds/”
       + $diskName  + ”.vhd”
12
13 $vm=Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri
        -CreateOption fromImage
14 <!--NeedCopy-->
```


MarketPlace に存在するイメージから VM を作成する場合、次のコマンドを使用して VM プランを指定します。

```
Set-AzureRmVMPlan -VM $vm -Publisher $pubName -Product $offerName -Name
 $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm
```

Azure 外部ロードバランサーを使用して高可用性セットアップで **Citrix ADC VPX** ペアをプロビジョニングする

Azure のユーザー資格情報を使用して AzureRmAccount にログオンします。

1. リソースグループの作成

   ここで指定した場所は、そのリソースグループ内のリソースの既定の場所です。ロードバランサーを作成する場合、すべてのコマンドで同じリソースグループを使用してください。

   ```
   $rgName=”<resource group name>”

   $locName=”<location name, such as West US>”

   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   たとえば、

   ```
   $rgName = ”ARM-LB-NS”

   $locName = ”West US”

   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. ストレージアカウントの作成

   ストレージアカウントには、小文字と数字のみを含む一意の名前を選択する必要があります。

   ```
   $saName=”<storage account name>”

   $saType=”<storage account type>”で次のいずれかを指定します: Standard_LRS、Standard_GRS、Standard_RAGRS、またはPremium_LRS

   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -
   Type $saType -Location $locName
   ```

   たとえば、


```
$saName=”vpxstorage”

$saType=”Standard_LRS”

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. 可用性セットの作成

   可用性セットが構成されたロードバランサーでは、アプリケーションをいつでも使用できます。

   ```
   $avName=”<availability set name>”

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -
   Location $locName
   ```

4. 仮想ネットワークを作成する

   以前に作成されたサブネットがない場合、少なくとも 1 つのサブネットを持つ新しい仮想ネットワークを追加します。

   ```
   $vnetName = ”LBVnet”

   $FrontendAddressPrefix=”10.0.1.0/24”

   $BackendAddressPrefix=”10.0.2.0/24”

   $vnetAddressPrefix=”10.0.0.0/16”

   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       frontendSubnet -AddressPrefix $FrontendAddressPrefix

   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       backendSubnet -AddressPrefix $BackendAddressPrefix

   $vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
       $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
       Subnet $frontendSubnet,$backendSubnet
   <!--NeedCopy-->
   ```

   注意: 要件に従って「AddressPrefix」パラメータ値を選択します。

   フロントエンドおよびバックエンドサブネットを、前の手順で作成した仮想ネットワークに割り当てます。

フロントエンドサブネットが配列 VNet の最初の要素である場合、subnetId は $vnet.subnets [0] .Id でなければなりません。

フロントエンドサブネットが配列の 2 番目の要素である場合、subnetId は $vnet.subnets [1] .Id というようにする必要があります。

5. フロントエンド **IP** アドレスを構成し、バックエンドアドレスプールを作成する

   受信ロードバランサーネットワークトラフィック用のフロントエンド IP アドレスを構成し、負荷分散トラフィックを受信するバックエンドアドレスプールを作成します。

```
$pubName=”PublicIp1”

$publicIP1 = New-AzureRmPublicIpAddress -Name $pubName -
    ResourceGroupName $rgName -Location $locName -AllocationMethod
    Static -DomainNameLabel nsvpx
<!--NeedCopy-->
```

   注: DomainNameLabel の値が使用可能かどうか確認してください。

```
$FIPName = ”ELBFIP”

$frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name
    $FIPName -PublicIpAddress $publicIP1

$BEPool = ”LB-backend-Pool”

$beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -
    Name $BEPool
<!--NeedCopy-->
```

6. ヘルスプローブの作成

   ポート 9000、間隔 5 秒で TCP ヘルププローブを作成します。

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name
    HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -
    ProbeCount 2
<!--NeedCopy-->
```

7. 負荷分散ルールの作成

   負荷分散するサービスごとに LB ルールを作成します。

たとえば、

次の例を使用して、HTTP サービスを負荷分散できます。

```
$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name ”HTTP-LB” -
    FrontendIpConfiguration $frontendIP1 -BackendAddressPool
    $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort
     80 -BackendPort 80
<!--NeedCopy-->
```

8. インバウンド **NAT** ルールの作成

   負荷分散していないサービスに対する NAT 規則を作成します。

   たとえば、Citrix ADC VPX インスタンスへの SSH アクセスを作成する場合などです。

   注:2 つの NAT ルールでプロトコル frontendPort-backendPort トリプレットが同じであってはなりません。

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH1     -FrontendIpConfiguration $frontendIP1 -Protocol
    TCP -FrontendPort 22 -BackendPort 22

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -
    FrontendPort 10022 -BackendPort 22
<!--NeedCopy-->
```

9. ロードバランサーエンティティの作成

   すべてのオブジェクト（NAT 規則、ロードバランサー規則、プローブ構成）を一度に追加してロードバランサーを作成します。

```
$lbName=”ELB”

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name
    $lbName -Location $locName -InboundNatRule $inboundNATRule1,
    $inboundNATRule2 -FrontendIpConfiguration $frontendIP1 -
    LoadBalancingRule $lbrule1 -BackendAddressPool $beAddressPool1
    -Probe $healthProbe
<!--NeedCopy-->
```


10. **NIC** を作成する

    2 つの NIC を作成し、各 NIC を各 VPX インスタンスに関連付けます

    a）NIC1 を VPX1 に

    たとえば、

```
$nicName=”NIC1”

$lbName=”ELB”

$bePoolIndex=0

* Rule indexes starts from 0.

$natRuleIndex=0

$subnetIndex=0

* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
    $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -Subnet $vnet.
    Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.
    BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule
    $lb.InboundNatRules[$natRuleIndex]
<!--NeedCopy-->
```

    b）NIC2 を VPX2 に

    たとえば、

```
$nicName=”NIC2”

$lbName=”ELB”

$bePoolIndex=0

$natRuleIndex=1

```



```
* Second Inbound NAT (SSH) rule we need to use

`$subnetIndex=0

* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
    $rgName

$nic2=New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -Subnet $vnet.
    Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.
    BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule
    $lb.InboundNatRules[$natRuleIndex]
<!--NeedCopy-->
```

11. **Citrix ADC VPX** インスタンスの作成

    同じリソースグループと可用性セットの一部として 2 つの Citrix ADC VPX インスタンスを作成し、外部ロードバランサーにアタッチします。

    a) Citrix ADC VPX インスタンス 1

    たとえば、

```
$vmName="VPX1"

$vmSize="Standard_A3"

$pubName="citrix"

$offerName="netscalervpx110-6531"

$skuName="netscalerbyol"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
    $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used
    to login to VPX instance"

```



```
$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName
    $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/
    " + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName
     -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
    $vm1
<!--NeedCopy-->
```

b) Citrix ADC VPX インスタンス 2

たとえば、

```
$vmName="VPX2"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be
    used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
```



```
    $vmName -Credential $cred -Verbose
12
13 $vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"
14
15 $vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id
16
17 $diskName="dynamic"
18
19 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName
20
21 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
    " + $diskName  + ".vhd"
22
23 $vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage
24
25 Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
     -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
    $vm2
28 <!--NeedCopy-->
```

12. 仮想マシンの構成

    両方の Citrix ADC VPX インスタンスが起動したら、SSH プロトコルを使用して両方の Citrix ADC VPX インスタンスに接続し、仮想マシンを構成します。

    a）アクティブ-アクティブ：両方の Citrix ADC VPX インスタンスのコマンドラインで同じ設定コマンドを実行します。

    b）アクティブ-パッシブ：両方の Citrix ADC VPX インスタンスのコマンドラインでこのコマンドを実行します。

    ```
    add ha node ##nodeID <nsip of other Citrix ADC VPX>
    ```

    アクティブ-パッシブモードでは、プライマリノードでのみ構成コマンドを実行します。

Azure 内部ロードバランサーを使用した高可用性セットアップでの **Citrix ADC VPX** ペアのプロビジョニング

Azure のユーザー資格情報を使用して AzureRmAccount にログオンします。

1. リソースグループの作成

ここで指定した場所は、そのリソースグループ内のリソースの既定の場所です。ロードバランサーを作成する場合、すべてのコマンドで同じリソースグループを使用してください。

```
$rgName=”\<resource group name\>”

$locName=”\<location name, such as West US\>”

New-AzureRmResourceGroup -Name $rgName -Location $locName
```

たとえば、

```
$rgName = ”ARM-LB-NS”

$locName = ”West US”

New-AzureRmResourceGroup -Name $rgName -Location $locName
<!--NeedCopy-->
```

2. ストレージアカウントの作成

ストレージアカウントには、小文字と数字のみを含む一意の名前を選択する必要があります。

```
$saName=”<storage account name>”

$saType=”<storage account type>”で次のいずれかを指定します: Standard_LRS、Standard_GRS、Standard_RAGRS、またはPremium_LRS

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
```

たとえば、

```
$saName=”vpxstorage”

$saType=”Standard_LRS”

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. 可用性セットの作成

可用性セットが構成されたロードバランサーでは、アプリケーションをいつでも使用できます。

```
$avName=”<availability set name>”

New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName
```


4. 仮想ネットワークを作成する

   以前に作成されたサブネットがない場合、少なくとも 1 つのサブネットを持つ新しい仮想ネットワークを追加します。

```
$vnetName = ”LBVnet”

$vnetAddressPrefix=”10.0.0.0/16”

$FrontendAddressPrefix=”10.0.1.0/24”

$BackendAddressPrefix=”10.0.2.0/24”

$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
     $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
    Subnet $frontendSubnet,$backendSubnet`

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    backendSubnet -AddressPrefix $BackendAddressPrefix
<!--NeedCopy-->
```

   注意: 要件に従って「AddressPrefix」パラメータ値を選択します。

   フロントエンドおよびバックエンドサブネットを、前の手順で作成した仮想ネットワークに割り当てます。

   フロントエンドサブネットが配列 VNet の最初の要素である場合、subnetId は $vnet.subnets [0] .Id でなければなりません。

   フロントエンドサブネットが配列の 2 番目の要素である場合、subnetId は $vnet.subnets [1] .Id というようにする必要があります。

5. バックエンドアドレスプールの作成

   `$beaddresspool= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name ”LB-backend”`

6. **NAT** ルールの作成

   負荷分散していないサービスに対する NAT 規則を作成します。

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name ”Inboundnatrule1” -FrontendIpConfiguration $frontendIP -
    Protocol TCP -FrontendPort 3441 -BackendPort 3389
```



```
2
3 $inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
      Name ”RDP2” -FrontendIpConfiguration $frontendIP -Protocol TCP
      -FrontendPort 3442 -BackendPort 3389
4 <!--NeedCopy-->
```

要件に応じて、フロントエンドポートおよびバックエンドポートを使用します。

7. ヘルスプローブの作成

   ポート 9000、間隔 5 秒で TCP ヘルッププローブを作成します。

```
1 $healthProbe = New-AzureRmLoadBalancerProbeConfig -Name ”
      HealthProbe”  ” -Protocol tcp -Port 9000 -IntervalInSeconds 5 -
      ProbeCount 2
2 <!--NeedCopy-->
```

8. 負荷分散ルールの作成

   負荷分散するサービスごとに LB ルールを作成します。

   次に例を示します:

   次の例を使用して、HTTP サービスを負荷分散できます。

```
1 $lbrule = New-AzureRmLoadBalancerRuleConfig -Name ”lbrule1” -
      FrontendIpConfiguration $frontendIP -BackendAddressPool
      $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort
      80 -BackendPort 80
2 <!--NeedCopy-->
```

   要件に応じて、フロントエンドポートおよびバックエンドポートを使用します。

9. ロードバランサーエンティティの作成

   すべてのオブジェクト（NAT 規則、ロードバランサー規則、プローブ構成）を一度に追加してロードバランサーを作成します。

```
1 $NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgname -Name
      ”InternalLB” -Location $locName -FrontendIpConfiguration
      $frontendIP -InboundNatRule $inboundNATRule1,$inboundNatRule2 -
      LoadBalancingRule $lbrule -BackendAddressPool $beAddressPool -
      Probe $healthProbe
2 <!--NeedCopy-->
```


10. **NIC** を作成する

    2 つの NIC を作成し、各 NIC を各 Citrix ADC VPX インスタンスに関連付ける

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic1-be -Location $locName -PrivateIpAddress
    10.0.2.6 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules[0]
<!--NeedCopy-->
```

    この NIC は、Citrix の Citrix ADC VPX 1 用です。プライベート IP は、追加されたサブネットと同じサブネット内に存在する必要があります。

```
$backendnic2= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic2-be -Location $locName -PrivateIpAddress
    10.0.2.7 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules[1].
<!--NeedCopy-->
```

    この NIC は Citrix ADC VPX 用です 2. `Private IPAddress`パラメーターには、要件に応じて任意のプライベート IP を設定できます。

11. **Citrix ADC VPX** インスタンスの作成

    同じリソースグループと可用性セットの一部に 2 つの VPX インスタンスを作成し、内部ロードバランサーにアタッチします。

    a) Citrix ADC VPX インスタンス 1

    たとえば、

```
$vmName=”VPX1”

$vmSize=”Standard_A3”

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

```



```
$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $backendnic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
<!--NeedCopy-->
```

b) Citrix ADC VPX インスタンス 2

たとえば、

```
$vmName="VPX2"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be
```



```
    used to login to VPX instance ”
10
11 $vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
    $vmName -Credential $cred -Verbose
12
13 $vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version ”latest”
14
15 $vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $backendnic2.Id
16
17 $diskName=”dynamic”
18
19 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName
20
21 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + ”vhds2/
    ” + $diskName  + ”.vhd”
22
23 $vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage
24
25 Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
     -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
    $vm2
28 <!--NeedCopy-->
```

12. 仮想マシンの構成

    両方の Citrix ADC VPX インスタンスが起動したら、SSH プロトコルを使用して両方の Citrix ADC VPX インスタンスに接続し、仮想マシンを構成します。

    a）アクティブ-アクティブ：両方の Citrix ADC VPX インスタンスのコマンドラインで同じ設定コマンドを実行します。

    b）アクティブ-パッシブ：両方の Citrix ADC VPX インスタンスのコマンドラインでこのコマンドを実行します。

    ```
    add ha node ##nodeID <nsip of other Citrix ADC VPX>
    ```

    アクティブ-パッシブモードでは、プライマリノードでのみ構成コマンドを実行します。

Azure に関するよくある質問

October 7, 2021

- **Azure Marketplace** からインストールされた **Citrix ADC VPX** インスタンスのアップグレード手順は、オンプレミスのアップグレード手順とは異なりますか**?**

  いいえ。標準の Citrix ADC VPX アップグレード手順を使用して、Microsoft Azure クラウドの Citrix ADC VPX インスタンスを Citrix ADC VPX リリース 11.1 以降にアップグレードできます。GUI または CLI プロシージャを使用してアップグレードできます。新規インストールの場合は、Microsoft Azure クラウド用の Citrix ADC VPX イメージを使用します。

  Citrix ADC VPX アップグレードビルドをダウンロードするには、［**Citrix** ダウンロード］**>** ［**Citrix ADC**ファームウェア］の順に選択します。

- **Azure** でホストされている **Citrix ADC VPX** インスタンスで観察される **MAC** 移動とインターフェイスミュートを修正するにはどうすればよいですか?

  Azure マルチ NIC 環境では、デフォルトでは、すべてのデータインターフェイスに MAC 移動とインターフェイスのミュートが表示されることがあります。Azure 環境での MAC 移動やインターフェイスのミュートを回避するには、ADC VPX インスタンスのデータインターフェイス（タグなし）ごとに VLAN を作成し、Azure で NIC のプライマリ IP をバインドすることをお勧めします。

  詳細については、CTX224626 の記事を参照してください。

Google Cloud Platform への Citrix ADC VPX インスタンスのデプロイ

January 25, 2022

Citrix ADC VPX インスタンスを Google Cloud Platform（GCP）にデプロイできます。GCP の VPX インスタンスを使用すると、GCP クラウドコンピューティング機能を活用し、ビジネスニーズに合わせて Citrix の負荷分散機能とトラフィック管理機能を使用できます。VPX インスタンスを GCP にスタンドアロンインスタンスとしてデプロイできます。シングル NIC 構成とマルチ NIC 構成の両方がサポートされています。

サポートされる機能

Premium、Advanced、Standard のすべての機能は、使用されているライセンス/バージョンタイプに基づいて GCP でサポートされます。

制限事項

- IPv6 はサポートされていません。

ハードウェア要件

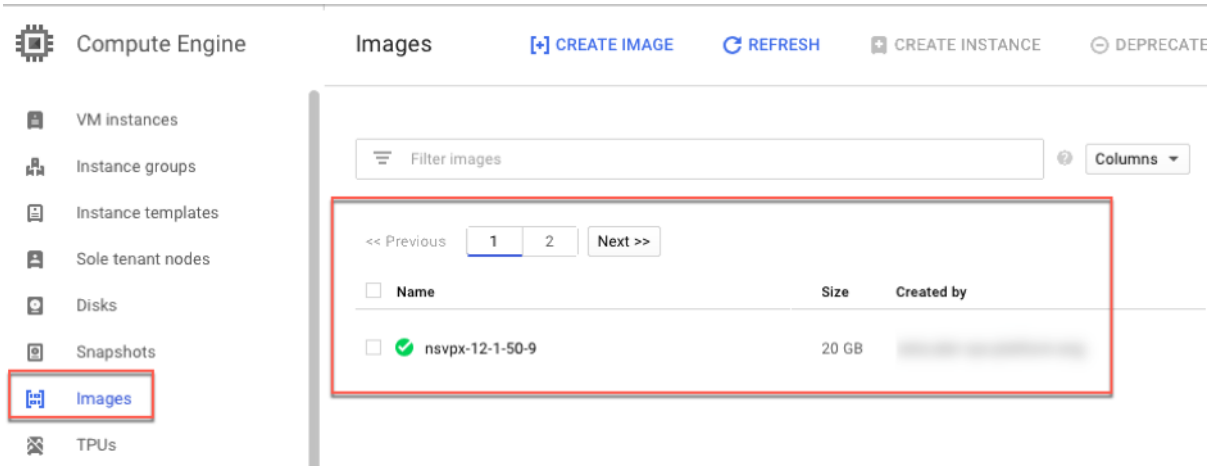
GCP の VPX インスタンスには、最低 2 つの vCPU と 4 GB の RAM が必要です。

前提条件

1. デバイスに「gcloud」ユーティリティをインストールします。このユーティリティは、次のリンクで見つけることができます。https://cloud.google.com/sdk/install
2. Citrix ダウンロードサイトから NSVPX-GCP イメージをダウンロードします。
3. 「https://cloud.google.com/storage/docs/uploading-objects」の手順に従って、ファイル（nsvpx-GCP-12.1-50.9_nc_64.tar.GZ）を Google のストレージバケットにアップロードします。
4. gcloud ユーティリティで次のコマンドを実行して、イメージを作成します。

```
gcloud compute images create <IMAGE_NAME> --source-uri=gs://<
    STORAGE_BUCKET_NAME>/<FILE_NAME>.tar.gz --guest-os-features=
    MULTI_IP_SUBNET
<!--NeedCopy-->
```

画像が作成されるまでに少し時間がかかる場合があります。イメージが作成されると、GCP コンソールの [コンピュート] > [**Compute Engine**] の下に表示されます。



注意すべきポイント

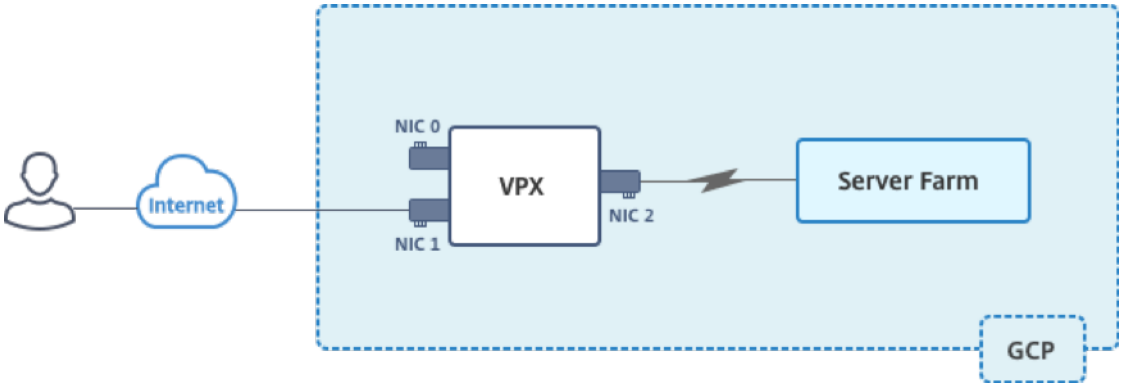
デプロイを開始する前に、次の GCP 固有の点を考慮してください。

- インスタンスの作成後、ネットワークインターフェイスを追加または削除することはできません。
- マルチ NIC デプロイの場合は、NIC ごとに個別の VPC ネットワークを作成します。1 つの NIC を関連付けることができるネットワークは 1 つだけです。

- シングル NIC インスタンスの場合、GCP コンソールはデフォルトでネットワークを作成します。
- 2 つ以上のネットワークインターフェースを持つインスタンスには、最低 4 つの vCPU が必要です。
- IP 転送が必要な場合は、インスタンスの作成と NIC の設定中に IP 転送を有効にする必要があります。

シナリオ：マルチ **NIC**、マルチ **IP** スタンドアロン **VPX** インスタンスをデプロイする

このシナリオでは、GCP に Citrix VPX スタンドアロンインスタンスを展開する方法について説明します。このシナリオでは、複数の NIC を持つスタンドアロン VPX インスタンスを作成します。インスタンスはバックエンドサーバー (サーバーファーム) と通信します。



次の目的に応える NIC を 3 つ作成します。

NIC	目的	VPC ネットワークに関連付けられている
NIC 0	管理トラフィック（Citrix ADC IP）にサービスを提供する	管理ネットワーク
NIC 1	クライアント側のトラフィック（VIP）をサービスする	クライアントネットワーク
NIC 2	バックエンド・サーバ（SNIP）との通信	バックエンドサーバーネットワーク

また、インスタンスとバックエンドサーバー間、およびパブリックインターネット上のインスタンスと外部ホスト間の必要な通信ルートを設定します。

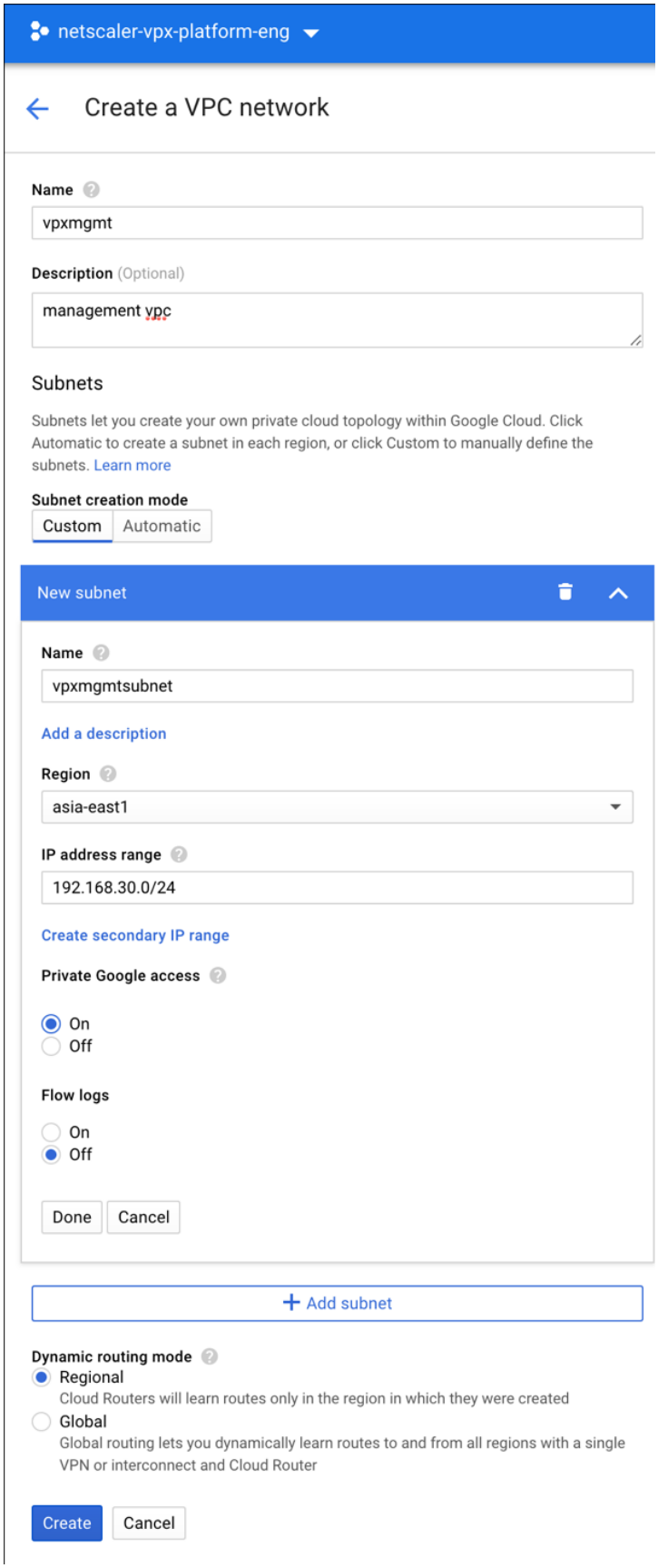
導入手順の概要

1. 3 つの異なる NIC に対して 3 つの VPC ネットワークを作成します。
2. ポート 22、80、443 のファイアウォールルールを作成します。
3. 3 つの NIC でインスタンスを作成する

注:VPC ネットワークを作成したのと同じリージョンにインスタンスを作成します。

手順 **1**: **VPC** ネットワークを作成します。

管理 NIC、クライアント NIC、およびサーバー NIC に関連付けられた 3 つの VPC ネットワークを作成します。VPC ネットワークを作成するには、**Google** コンソール **>** ネットワーク **> VPC** ネットワーク **> VPC** ネットワークの作成にログオンします。スクリーン・キャプチャに示されている必須フィールドに入力し、「作成」をクリックします。

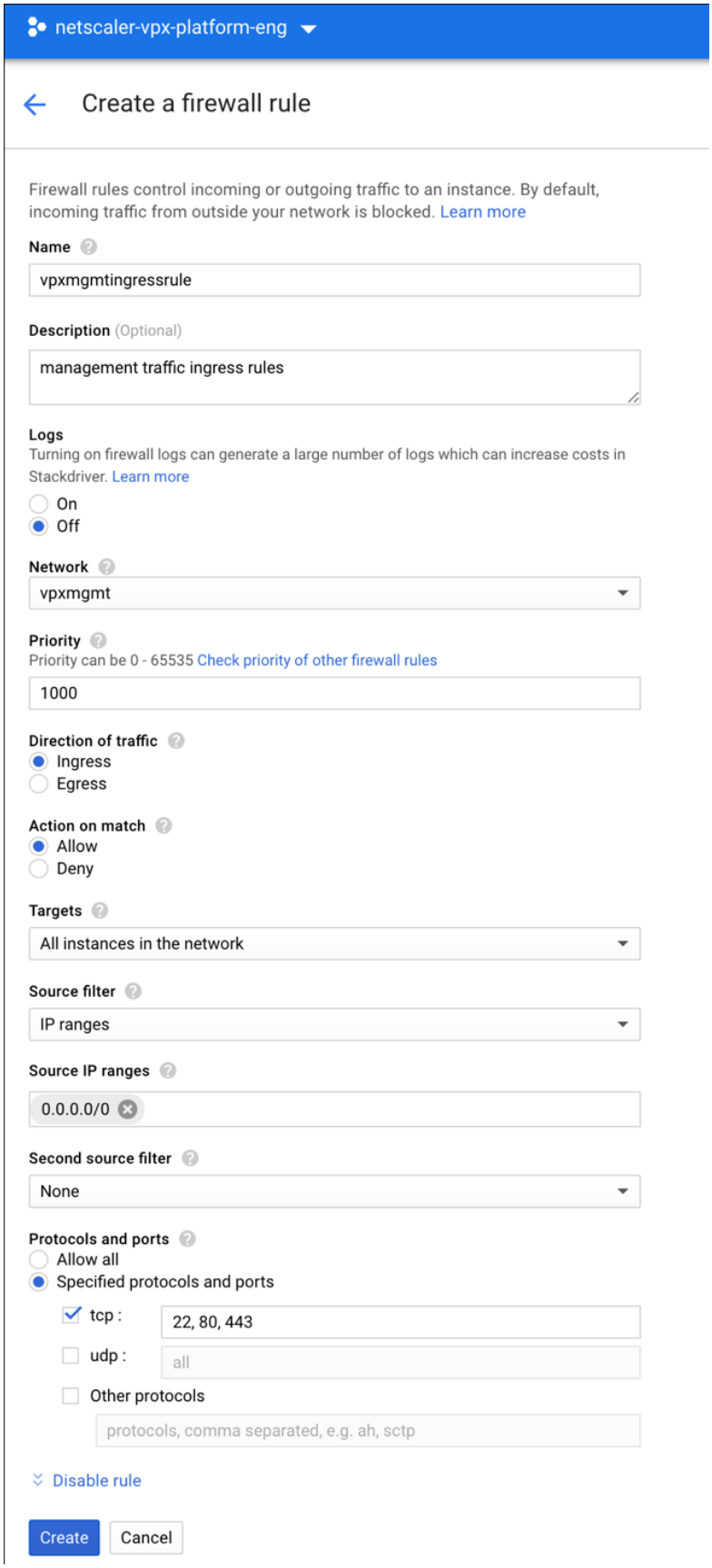


同様に、クライアント側およびサーバー側 NIC 用の VPC ネットワークを作成します。

注:3 つの VPC ネットワークはすべて同じリージョンにある必要があります。このシナリオでは asia-east1 になります。

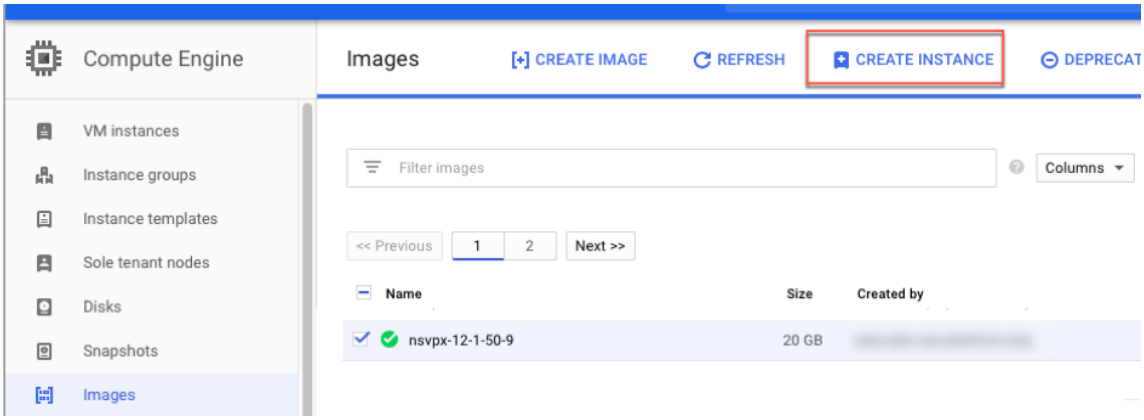
手順 **2**：ポート **22**、**80**、および **443** のファイアウォールルールを作成します。

VPC ネットワークごとに SSH（ポート 22）、HTTP（ポート 80）、HTTPS（ポート 443）のルールを作成します。ファイアウォールルールの詳細については、「ファイアウォールルールの概要」を参照してください。



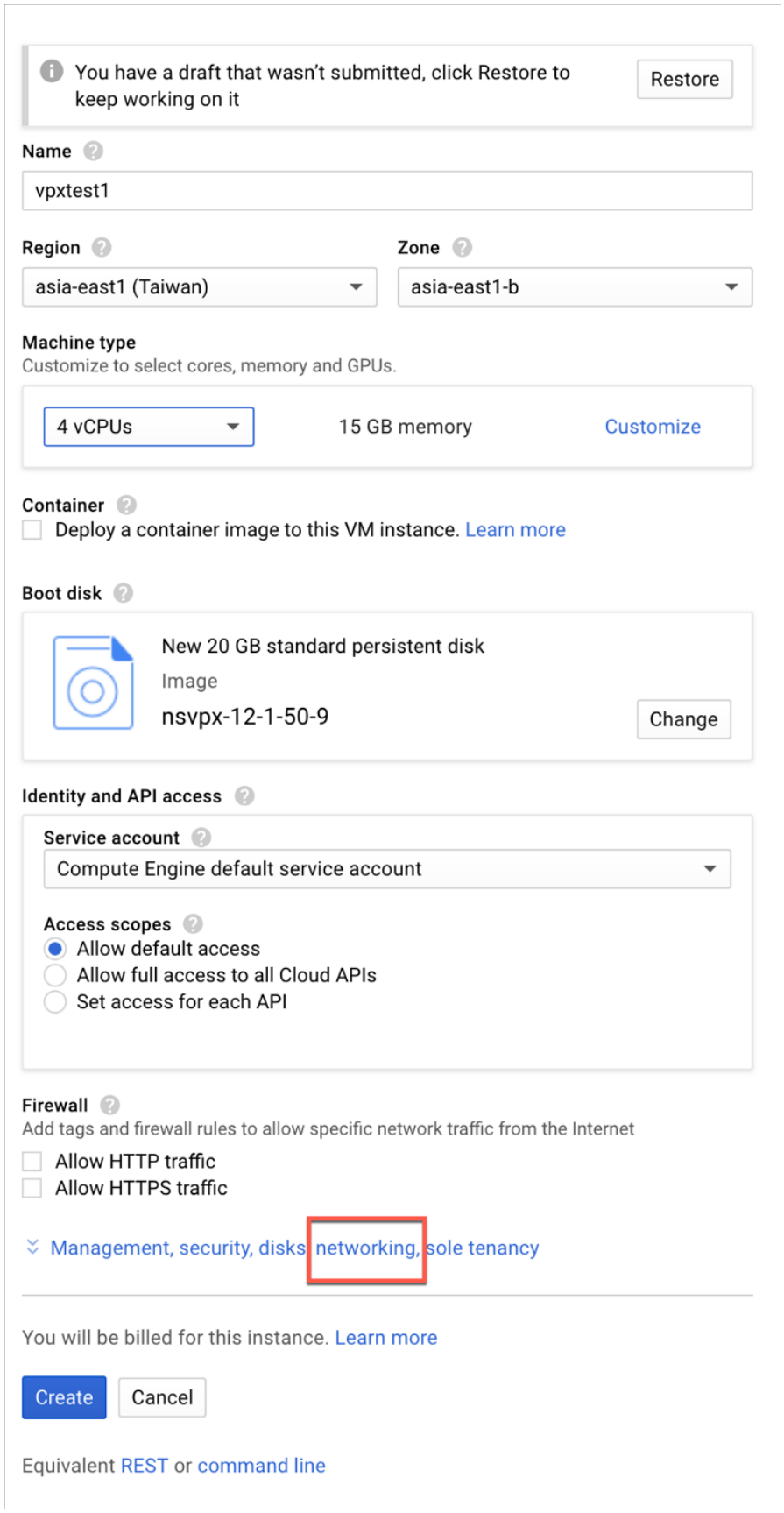
手順 **3**：**VPX** インスタンスを作成します。

1. GCP コンソールにログオンします。
2. [コンピューティング] で、[Compute Engine] にカーソルを合わせ 、[
3. イメージを選択し、[インスタンスの作成] をクリックします。

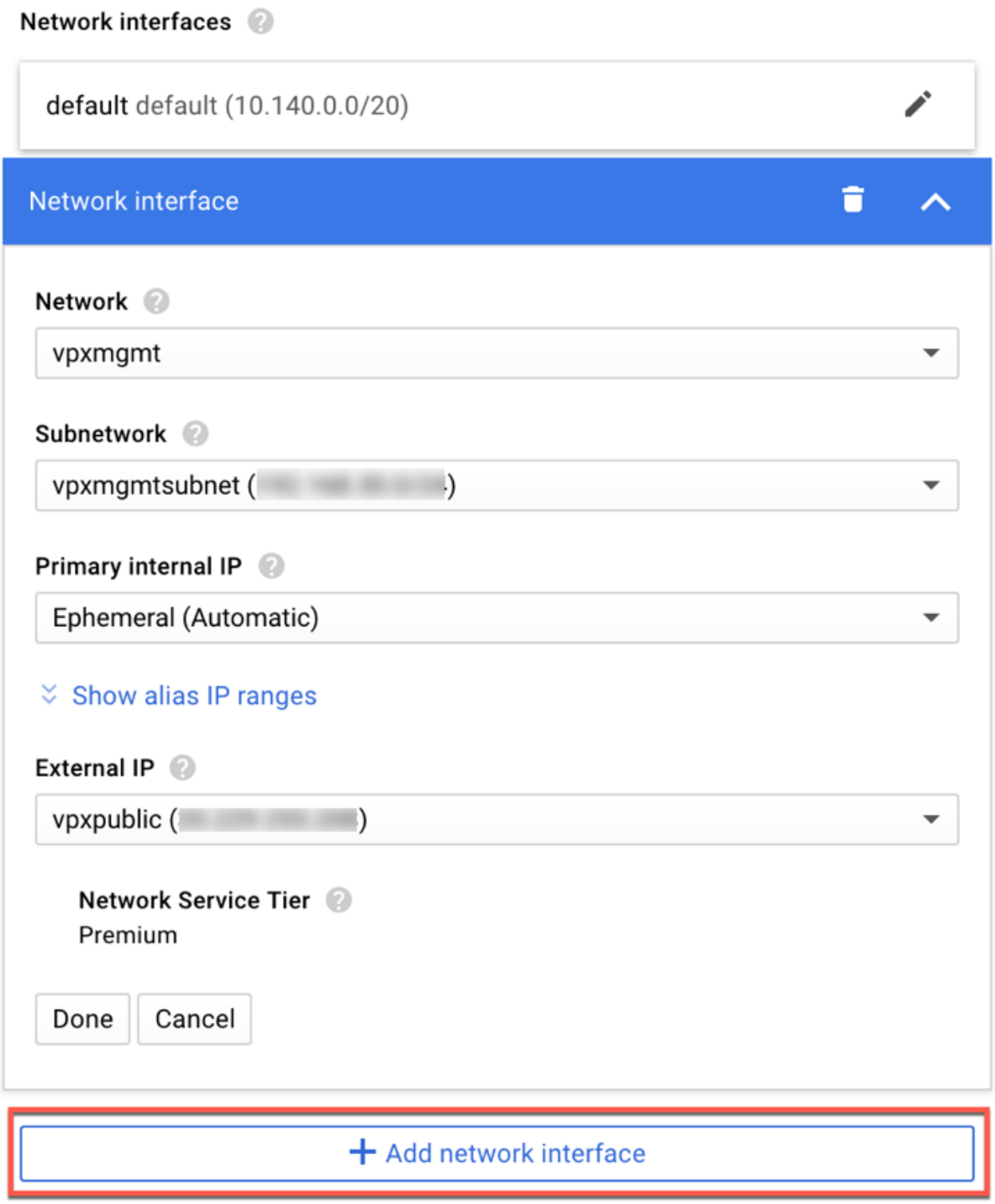


4. 複数の NIC をサポートするには、4 つの vCPU を持つインスタンスを選択します。
5. [管理、セキュリティ、ディスク、ネットワーキング、単独テナンシー] から [ネットワーク] オプションをクリックして、NIC を追加します。

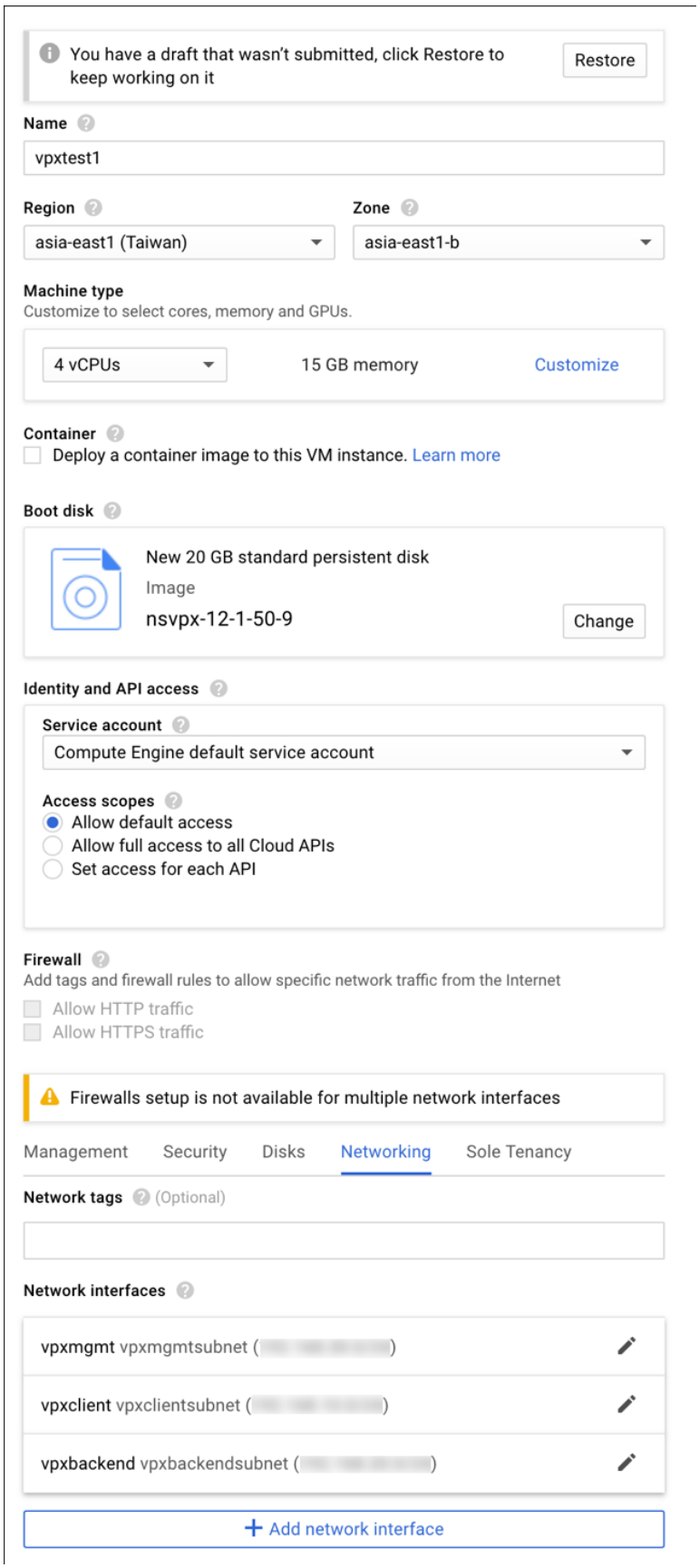
   注: コンテナイメージは、GCP の VPX インスタンスではサポートされていません。



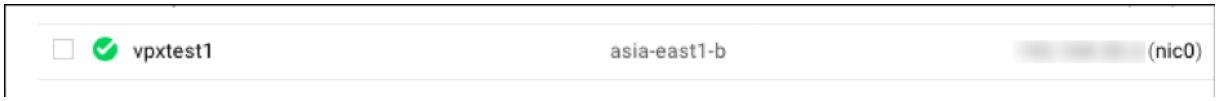
6. [ネットワークインターフェイス] で、[編集] アイコンをクリックして、デフォルトの NIC を編集します。この NIC は管理 NIC です。
7. [ネットワークインターフェイス] ウィンドウの [ネットワーク] で、管理 NIC 用に作成した VPC ネットワークを選択します。
8. 管理 NIC の場合は、静的外部 IP アドレスを作成します。[外部 IP] リストで、[**IP** アドレスの作成] をクリックします。
9. [新しい静的 **IP** アドレスを予約する] ウィンドウで、名前と説明を追加し、[予約] をクリックします。
10. [ネットワークインターフェイスの追加] をクリックして、クライアント側およびサーバー側のトラフィック用の NIC を作成します。



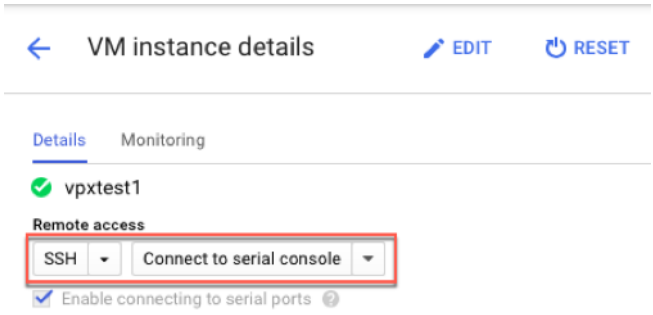
すべての NIC を作成したら、[作成] をクリックして VPX インスタンスを作成します。



インスタンスは [**VM** インスタンス] の下に表示されます。

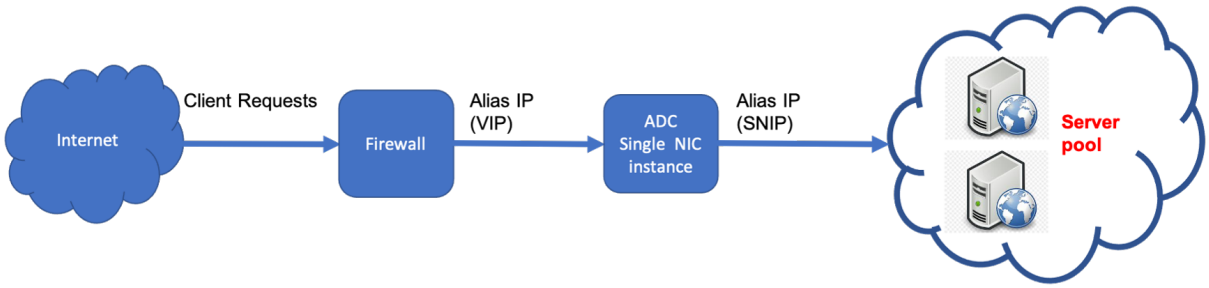


GCP SSH またはシリアルコンソールを使用して VPX インスタンスを構成および管理します。



シナリオ**:** シングル **NIC** のスタンドアロン **VPX** インスタンスをデプロイする

このシナリオでは、GCP に単一の NIC を使用して Citrix VPX スタンドアロンインスタンスを展開する方法について説明します。エイリアス IP アドレスは、この展開を実現するために使用されます。



1 つの NIC (NIC0) を作成して、次の目的を果たします。

- 管理ネットワーク内の管理トラフィック（Citrix ADC IP）を処理します。
- クライアントネットワーク内のクライアント側トラフィック (VIP) を処理します。
- バックエンドサーバーネットワーク内のバックエンドサーバー (SNIP) と通信します。

次の間の必要な通信ルートを設定します。

- インスタンスとバックエンドサーバー。
- パブリックインターネット上のインスタンスと外部ホスト。

導入手順の概要

1. NIC0 用の VPC ネットワークを作成します。
2. ポート 22、80、および 443 のファイアウォールルールを作成します。
3. 1 つの NIC でインスタンスを作成します。

4. VPX にエイリアス IP アドレスを追加します。
5. VPX に VIP と SNIP を追加します。
6. 負荷分散仮想サーバーを追加します。
7. インスタンスにサービスまたはサービスグループを追加します。
8. サービスまたはサービスグループをインスタンス上の負荷分散仮想サーバーにバインドします。

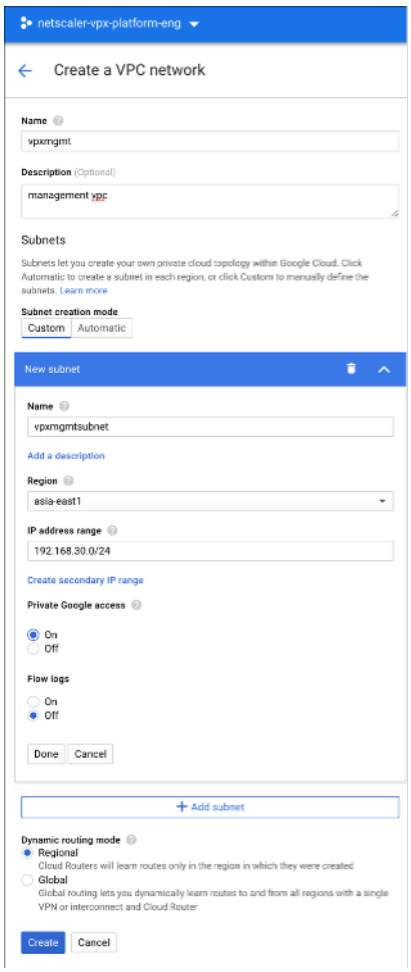
> 注:
>
> VPC ネットワークを作成したのと同じリージョンにインスタンスを作成します。

手順 **1**：**1** つの **VPC** ネットワークを作成します。

NIC0 に関連付ける VPC ネットワークを 1 つ作成します。

VPC ネットワークを作成するには、次の手順を実行します。

1. **GCP** コンソール **>** ネットワーク **> VPC** ネットワーク **> VPC** ネットワークの作成にログオンします。
2. 必須フィールドに入力し、[**Create**] をクリックします。



手順 **2**：ポート **22**、**80**、および **443** のファイアウォールルールを作成します。

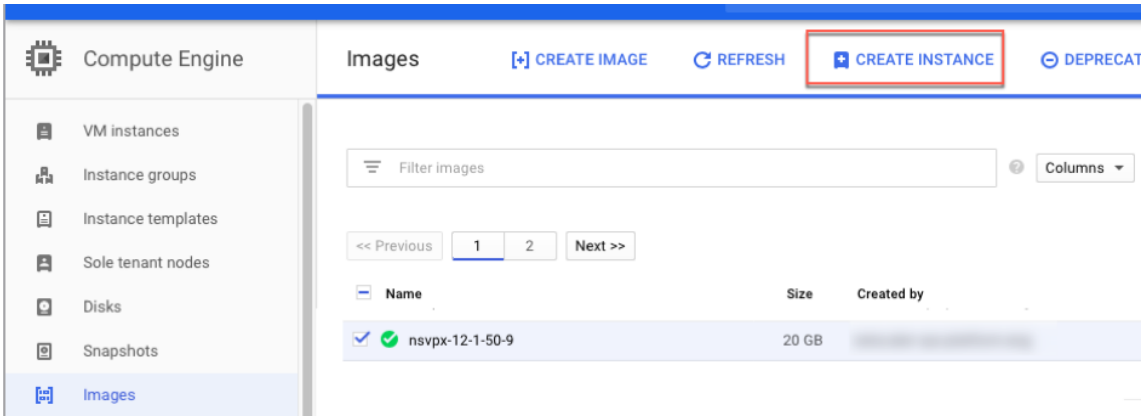
VPC ネットワークの SSH（ポート 22）、HTTP（ポート 80）、HTTPS（ポート 443）のルールを作成します。ファ

イアウォールルールの詳細については、「ファイアウォールルールの概要」を参照してください。

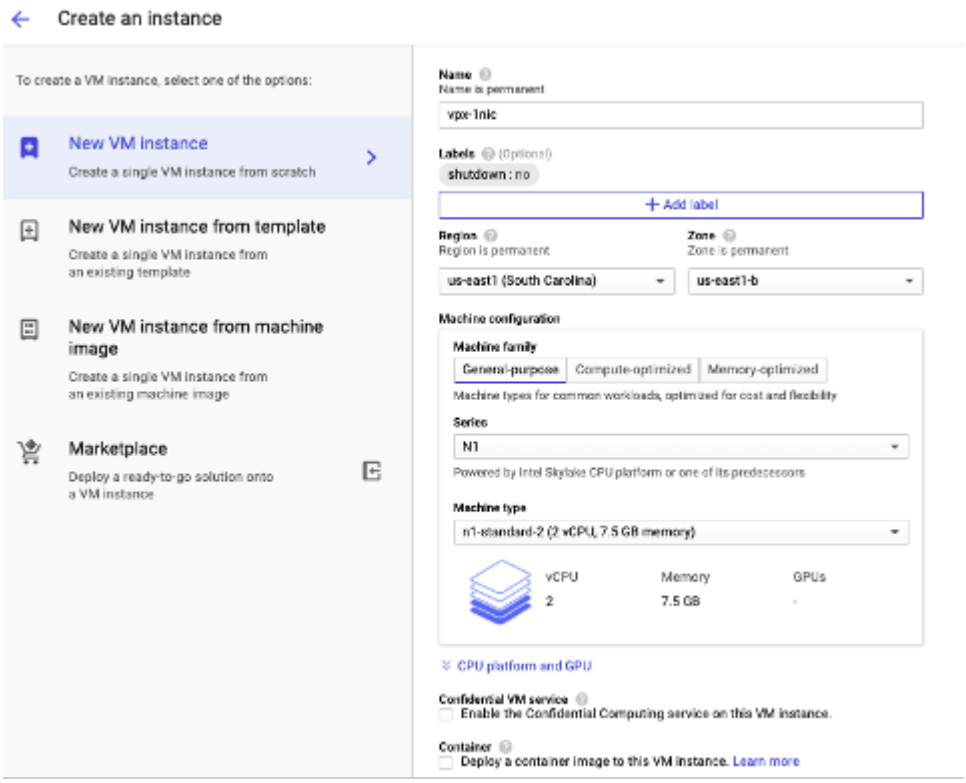
手順 **3**：**1** つの **NIC** でインスタンスを作成します。

単一の NIC でインスタンスを作成するには、次の手順を実行します。

1. **GCP** コンソールにログオンします。
2. [コンピュート] で [**Compute Engine**] にカーソルを合わせ 、[
3. イメージを選択し、[インスタンスの作成] をクリックします。



4. 2 つの vCPU を持つインスタンスタイプを選択します（ADC の最小要件）。



5. [管理、セキュリティ、ディスク、** ネットワーク] ウィンドウから [ネットワーク **] タブをクリックします。
6. [ネットワークインターフェイス] で、[編集] アイコンをクリックして、デフォルトの NIC を編集します。
7. [ネットワークインターフェイス] ウィンドウの [ネットワーク] で、作成した VPC ネットワークを選択します。
8. 静的外部 IP アドレスを作成できます。[外部 **IP** アドレス] で、**[**IP** アドレスの作成 **] をクリックします。
9. [静的アドレスを予約] ウィンドウで、名前と説明を追加し、[予約] をクリックします。
10. [作成] をクリックして VPX インスタンスを作成します。
    新しいインスタンスが [VM インスタンス] の下に表示されます。

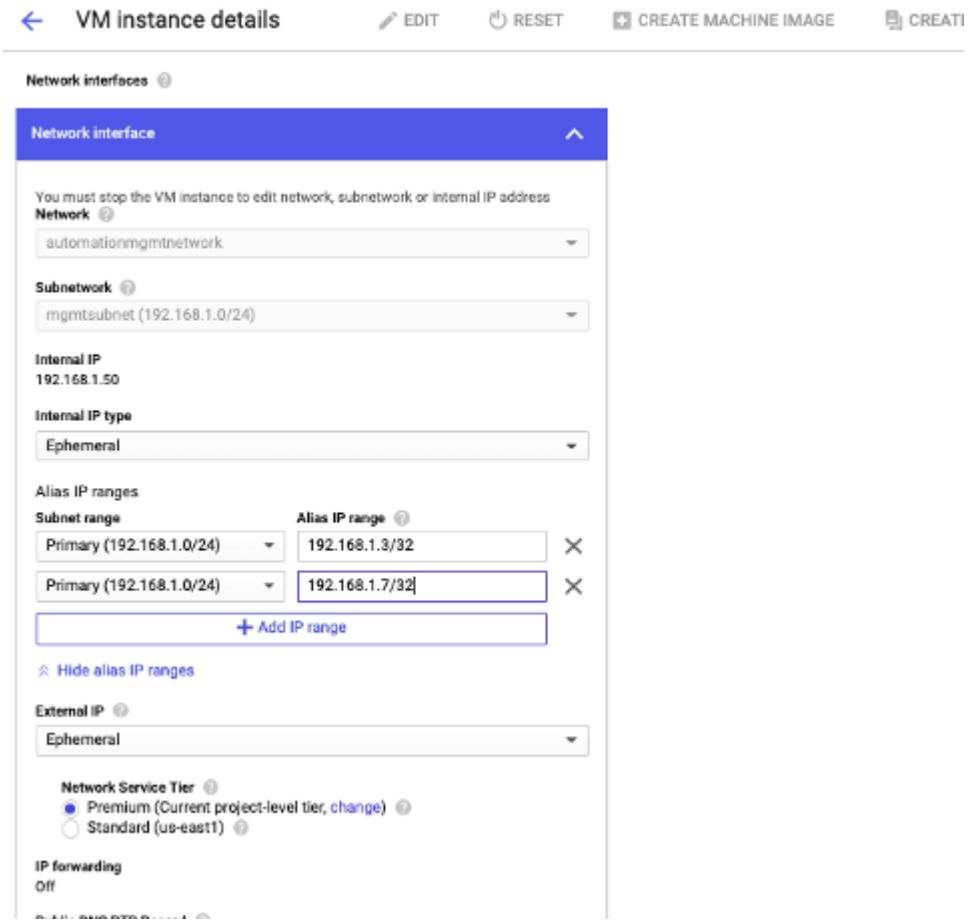
手順 **4**：**VPX** インスタンスにエイリアス **IP** アドレスを追加します。

VIP アドレスと SNIP アドレスとして使用する VPX インスタンスに 2 つのエイリアス IP アドレスを割り当てます。

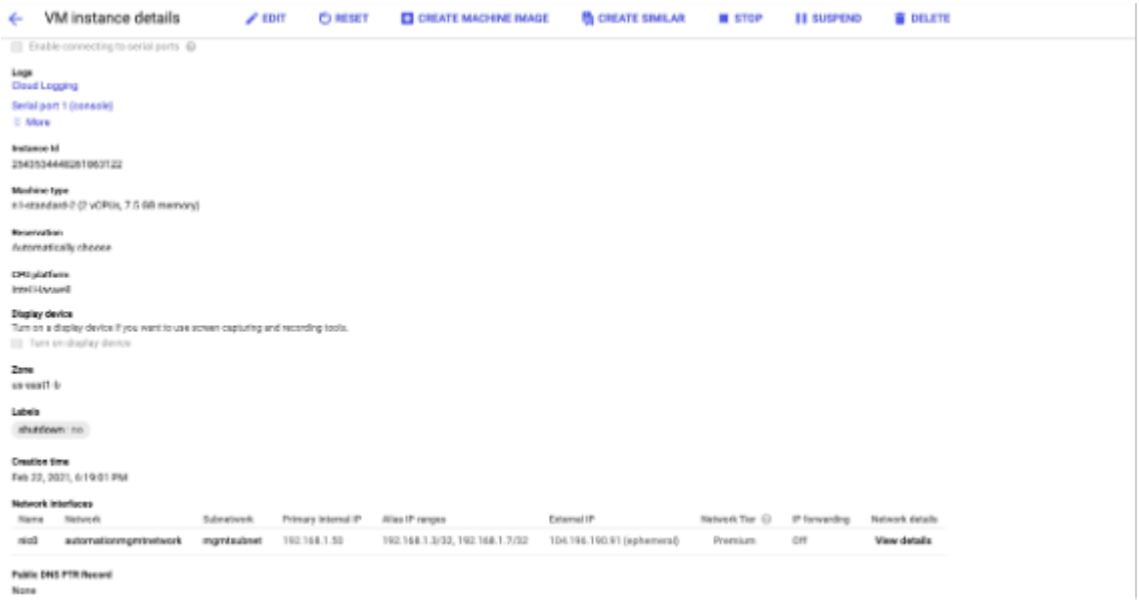
> 注：
>
> VPX インスタンスのプライマリ内部 IP アドレスを使用して VIP または SNIP を構成しないでください。

エイリアス IP アドレスを作成するには、次の手順を実行します。

1. VM インスタンスに移動し、[編集] をクリックします。
2. [ネットワークインターフェイス] ウィンドウで、NIC0 インターフェイスを編集します。
3. [エイリアス **IP** 範囲] フィールドに、エイリアス IP アドレスを入力します。



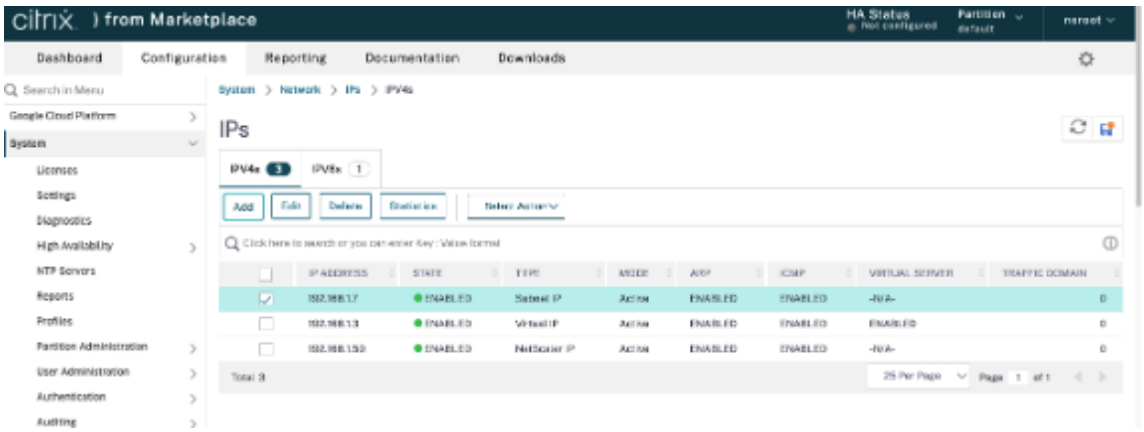
4. [完了]、[保存] の順にクリックします。
5. **VM** インスタンスの詳細ページでエイリアス IP アドレスを確認します。



手順 **5**：**VPX** インスタンスに **VIP** と **SNIP** を追加します。

VPX インスタンスで、クライアントエイリアス IP アドレスとサーバーエイリアス IP アドレスを追加します。

1. Citrix ADC GUI で、[システム] **>** [ネットワーク] **>** [**IP**] **>** [**IPv4**] に移動し、[追加] をクリックします。



2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。

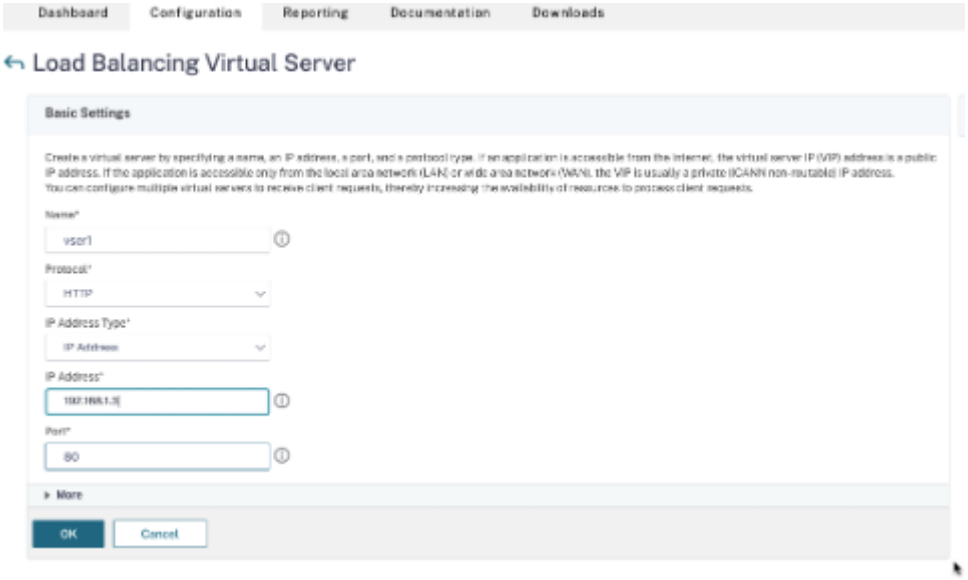
   - VM インスタンスで VPC サブネットに設定されたクライアントエイリアス IP アドレスとネットマスクを入力します。
   - [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   - [作成] をクリックします。

3. サーバーエイリアス IP (SNIP) アドレスを作成するには、次の手順を実行します。

   - VM インスタンスの VPC サブネットに設定されたサーバーエイリアス IP アドレスとネットマスクを入力します。
   - [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   - [作成] をクリックします。

手順 **6**：負荷分散仮想サーバーを追加します。

1. Citrix ADC GUI で、［構成］**>**［トラフィック管理］**>**［負荷分散］**>**［仮想サーバー］に移動し、［追加］をクリックします。
2. [名前]、[プロトコル]、[IP アドレスタイプ (IP アドレス)]、[IP アドレス] (クライアントエイリアス IP)、および [ポート] に必要な値を追加します。
3. **OK** をクリックして、負荷分散仮想サーバーを作成します。



手順 **7**：**VPX** インスタンスにサービスまたはサービスグループを追加します。

1. Citrix ADC GUI から、［構成］**>**［トラフィック管理］**>**［負荷分散］**>**［サービス］に移動し、［追加］をクリックします。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、「**OK**」をクリックします。

手順 **8.** サービス**/**サービスグループをインスタンス上の負荷分散仮想サーバーにバインドします。

1. GUI から、[設定**] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバ] に移動します。
2. 手順 **6** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] ウィンドウで、[負荷分散仮想サーバーサービスのバインドなし] をクリックします。
4. ステップ **7** で設定したサービスを選択し、[バインド（Bind）] をクリックします。

VPX インスタンスを **GCP** にデプロイした後の注意点

- ユーザー名`nsroot`とインスタンス ID をパスワードとして VPX にログオンします。プロンプトで、パスワードを変更し、設定を保存します。
- テクニカルサポートバンドルを収集するには、慣例`show techsupport`ではなくコマンド`shell /netscaler/showtech_cloud.pl`を実行します。

- GCP コンソールから Citrix ADC VM を削除した後、関連する Citrix ADC 内部ターゲットインスタンスも削除します。これを行うには、gcloud CLI に移動し、次のコマンドを入力します。

```
1  gcloud compute -q  target-instances delete <instance-name>-
       adcinternal --zone <zone>
2  <!--NeedCopy-->
```

> 注: <instance-name>-adcinternal は、削除する必要があるターゲットインスタンスの名前です。

Citrix ADC VPX ライセンス

GCP 上の Citrix ADC VPX インスタンスにはライセンスが必要です。GCP で実行されている Citrix ADC VPX インスタンスでは、以下のライセンスオプションを使用できます。

- サブスクリプションベースのライセンス：Citrix ADC VPX アプライアンスは、GCP マーケットプレイスで有料インスタンスとして利用できます。サブスクリプションベースのライセンスは、従量課金制のオプションです。ユーザーは時間単位で課金されます。GCP マーケットプレイスでは、以下の VPX モデルとライセンスエディションが利用可能です。

VPX モデル	ライセンスエディション
—	—
VPX10	スタンダード、アドバンス、プレミアム

- 自分のライセンスを持ち込む **(BYOL)**: 自分のライセンス (BYOL) を持参する場合は、http://support.citrix.com/article/CTX122426の「VPX ライセンスガイド」を参照してください。次の操作を実行する必要があります。
  - Citrix Web サイト内のライセンスポータルを使用して、有効なライセンスを生成します。
  - ライセンスをインスタンスにアップロードします。
- **Citrix ADC VPX** チェックイン/チェックアウトライセンス：詳細については、「Citrix ADC VPX チェックイン/チェックアウトライセンス」を参照してください。

オンプレミスおよびクラウド展開用の VPX Express では、ライセンスファイルは不要です。Citrix ADC VPX Express の詳細については、Citrix ADC ライセンスの概要の「Citrix ADC VPX Express ライセンス」セクションを参照してください。

Citrix ADC VPX インスタンスを展開するための GDM テンプレート

Citrix ADC VPX Google デプロイメントマネージャー（GDM）テンプレートを使用して、GCP に VPX インスタンスを展開できます。詳細については、Citrix ADC GDM テンプレートを参照してください。

Citrix ADC マーケットプレイスのイメージ

GDM テンプレート内のイメージを使用して、Citrix ADC アプライアンスを起動できます。

次の表は、GCP マーケットプレイスで利用可能な画像の一覧です。

解除	イメージ名	イメージの場所
13.0	citrix-adc-vpx-10-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-enterprise-13-0-83-29
13.0	citrix-adc-vpx-10-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-platinum-13-0-83-29
13.0	citrix-adc-vpx-10-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-standard-13-0-83-29
13.0	citrix-adc-vpx-200-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-0-83-29
13.0	citrix-adc-vpx-200-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-platinum-13-0-83-29
13.0	citrix-adc-vpx-200-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-standard-13-0-83-29
13.0	citrix-adc-vpx-1000-advanced-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-advanced-13-0-83-29
13.0	citrix-adc-vpx-1000-premium-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-premium-13-0-83-29

解除	イメージ名	イメージの場所
13.0	citrix-adc-vpx-1000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-standard-13-0-83-29
13.0	citrix-adc-vpx-3000-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-enterprise-13-0-83-29
13.0	citrix-adc-vpx-3000-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-platinum-13-0-83-29
13.0	citrix-adc-vpx-3000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-standard-13-0-83-29
13.0	citrix-adc-vpx-5000-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-enterprise-13-0-83-29
13.0	citrix-adc-vpx-5000-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-0-83-29
13.0	citrix-adc-vpx-5000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-standard-13-0-83-29
13.0	citrix-adc-vpx-byol-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-0-83-29
13.0	citrix-adc-vpx-express-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-0-83-29

解除	イメージ名	イメージの場所
13.0	citrix-adc-vpx-waf-1000-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-waf-1000-13-0-83-29

リソース

- 複数のネットワークインターフェースを持つインスタンスの作成
- VM インスタンスの作成と起動

関連情報

- VPX 高可用性ペアを Google Cloud Platform にデプロイする

Google Cloud Platform に VPX 高可用性ペアを展開する

October 7, 2021

Google Cloud Platform（GCP）上の 2 つの Citrix ADC VPX インスタンスを、高可用性（HA）アクティブ-パッシブペアとして構成できます。1 つのインスタンスをプライマリノードとして構成し、もう 1 つをセカンダリノードとして設定すると、プライマリノードは接続を受け入れ、サーバーを管理します。セカンダリノードはプライマリを監視します。何らかの理由で 1 次ノードが接続を受け入れることができない場合、2 次ノードが引き継ぎます。

HA の詳細については、「ハイアベイラビリティ」を参照してください。

ノードは同じリージョンにある必要がありますが、同じゾーンまたは異なるゾーンにある可能性があります。詳細については、「リージョンとゾーン」を参照してください。

各 VPX インスタンスには、少なくとも 3 つの IP サブネット（Google VPC ネットワーク）が必要です。

- 管理サブネット
- クライアント側サブネット（VIP）
- バックエンド側サブネット（SNIP、MIP など）

標準の VPX インスタンスには 3 つのネットワークインターフェイスをお勧めします。

VPX 高可用性ペアは、次の方法でデプロイできます。

- 外部静的 IP アドレスの使用
- プライベート IP アドレスの使用

GCP に VPX 高可用性ペアを展開するための GDM テンプレート

Citrix ADC Google デプロイメントマネージャー（GDM）テンプレートを使用して、GCP に VPX 高可用性ペアを展開できます。詳細については、Citrix ADC GDM テンプレートを参照してください。

GCP での VPX 高可用性ペアの転送ルールのサポート

転送ルールを使用して、GCP に VPX 高可用性ペアをデプロイできます。

転送ルールの詳細については、「転送ルールの概要」を参照してください。

前提条件

- 転送ルールは、VPX インスタンスと同じリージョンにある必要があります。
- ターゲットインスタンスは、VPX インスタンスと同じゾーンにある必要があります。
- プライマリノードとセカンダリノードの両方のターゲットインスタンスの数が一致する必要があります。

例**:**

`us-east1` リージョンに高可用性ペアがあり、プライマリ VPX が `us-east1-b` ゾーンにあり、セカンダリ VPX が `us-east1-c` ゾーンにあります。`us-east1-b` ゾーンにターゲットインスタンスがあるプライマリ VPX に対して転送ルールが設定されます。`us-east1-c` ゾーンでセカンダリ VPX のターゲットインスタンスを構成して、フェイルオーバー時に転送ルールを更新します。

制限事項

VPX 高可用性デプロイメントでは、バックエンドでターゲットインスタンスを使用して構成された転送ルールのみがサポートされます。

Google Cloud Platform に外部の静的 IP アドレスを指定した VPX 高可用性ペアをデプロイする

October 7, 2021

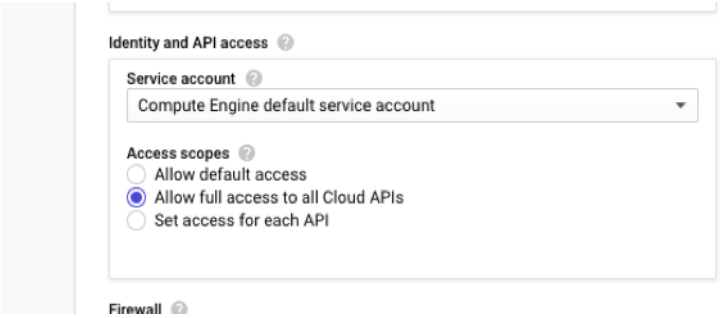
VPX ハイアベイラビリティペアは、外部の静的 IP アドレスを使用して GCP にデプロイできます。プライマリノードのクライアント IP アドレスは、外部の静的 IP アドレスにバインドする必要があります。フェールオーバー時に、外部静的 IP アドレスがセカンダリノードに移動され、トラフィックが再開されます。

静的外部 IP アドレスは、プロジェクトを解放するまでプロジェクト用に予約されている外部 IP アドレスです。IP アドレスを使用してサービスにアクセスする場合、その IP アドレスを予約して、プロジェクトのみが使用できるようにすることができます。詳細については、「静的外部 IP アドレスの予約」を参照してください。

HA の詳細については、「ハイアベイラビリティ」を参照してください。

はじめに

- Google Cloud Platform への Citrix ADC VPX インスタンスのデプロイで説明されている制限、ハードウェア要件、注意点をお読みください。この情報は、HA 配置にも適用されます。
- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。
- インスタンスの作成中に、すべての Cloud API へのフルアクセスを許可します。



- GCP サービスアカウントに関連付けられた IAM ロールに次の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [

”compute.addresses.use”,
 “compute.forwardingRules.list” ,
 “compute.forwardingRules.setTarget” ,
 “compute.instances.setMetadata”
”compute.instances.addAccessConfig”,
”compute.instances.deleteAccessConfig”,
”compute.instances.get”,
”Compute.instances.list”,
”compute.networks.useExternalIp”,
”compute.subnetworks.useExternalIp”,
 “compute.targetInstances.list” ,
 “compute.targetInstances.use” ,
”compute.zones.list”,
]
<!--NeedCopy-->
```

- 管理インターフェイス以外のインターフェイスにエイリアス IP アドレスを設定している場合は、GCP サービスアカウントに次の追加の IAM 権限があることを確認します。

```
”compute.instances.updateNetworkInterface”
<!--NeedCopy-->
```


- プライマリノードで GCP 転送ルールを構成した場合は、「GCP での VPX 高可用性ペアの転送ルールのサポート」に記載されている制限と要件を読み、フェールオーバー時に新しいプライマリに更新します。

Google Cloud Platform に VPX HA ペアを展開する方法

HA 展開手順の概要を次に示します。

1. 同じリージョンに VPC ネットワークを作成します。たとえば、アジア東です。
2. 同じリージョンに 2 つの VPX インスタンス（プライマリノードとセカンダリノード）を作成します。それらは、同じゾーンまたは異なるゾーンに存在することができます。たとえば、アジア東-1a、アジア東-Ib。
3. Citrix ADC GUI または ADC CLI コマンドを使用して、両方のインスタンスで高可用性設定を構成します。

手順 1. VPC ネットワークを作成する

要件に基づいて VPC ネットワークを作成します。管理 NIC、クライアント NIC、サーバー NIC に関連付けるために 3 つの VPC ネットワークを作成することをお勧めします。

VPC ネットワークを作成するには、次の手順を実行します。

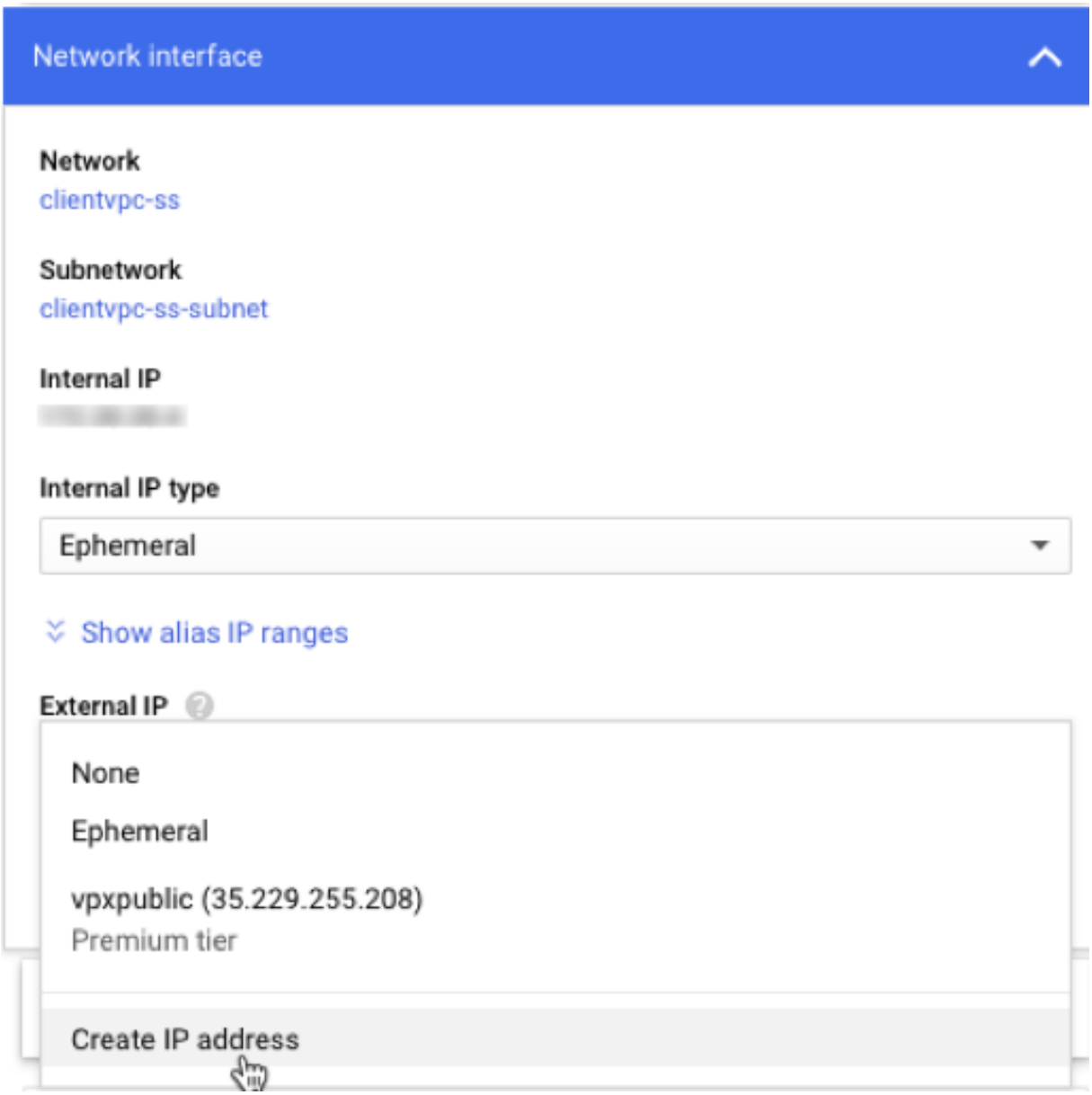
1. **Google** コンソールにログインし、**[**ネットワーク**] > [VPC** ネットワーク**] > [VPC** ネットワークの作成**]** をクリックします。
2. 必須フィールドに入力し、[**Create**] をクリックします。

詳細については、「Google Cloud Platform での Citrix ADC VPX インスタンスのデプロイ」の「**VPC** ネットワークの作成」セクションを参照してください。

手順 2. 2 つの VPX インスタンスを作成する

シナリオ：マルチ NIC、マルチ IP スタンドアロン VPX インスタンスをデプロイする手順に従って、2 つの VPX インスタンスを作成します。

> 重要
>
> プライマリノードのクライアント IP アドレス (VIP) に静的外部 IP アドレスを割り当てます。既存の予約済み IP アドレスを使用するか、新しい予約済み IP アドレスを作成できます。静的外部 IP アドレスを作成するには、[ネットワークインターフェイス**] > [**外部 **IP**] に移動し、[**IP** アドレスの作成] をクリックします。



フェールオーバー後、古いプライマリが新しいセカンダリになると、スタティック外部 IP アドレスは古いプライマリから移動し、新しいプライマリに接続されます。詳細については、Google Cloud ドキュメント「静的外部 IP アドレスを予約する」を参照してください。

VPX インスタンスを構成したら、VIP アドレスと SNIP アドレスを構成できます。詳細については、「Citrix ADC 所有の IP アドレスの構成」を参照してください。

手順 **3.** 高可用性の構成

Google Cloud Platform でインスタンスを作成した後、CLI 用 Citrix ADC GUI を使用して HA を構成できます。

GUI を使用した **HA** の設定

ステップ **1.** 両方のインスタンスで INC モードで高可用性をセットアップします。

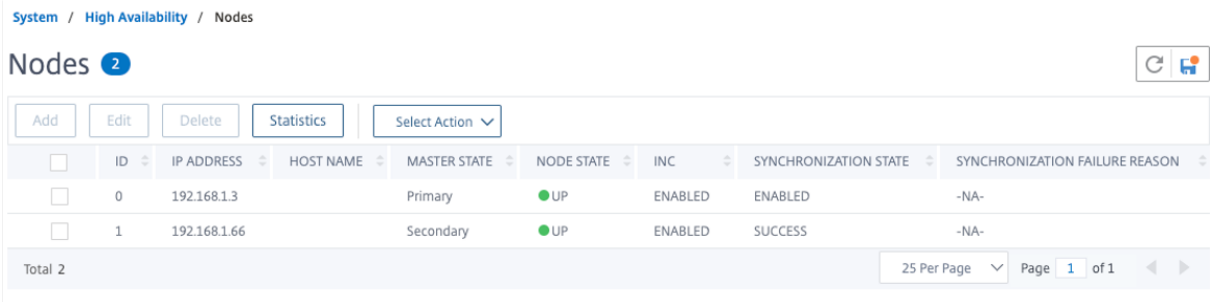
プライマリノードで、次の手順を実行します。

1. GCP Console からノードのユーザー名nsrootとインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノードの **IP** アドレス] フィールドに、セカンダリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

セカンダリノードで、次の手順を実行します。

1. GCP Console nsroot からノードのユーザー名とインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノード **IP** アドレス] フィールドに、プライマリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [作成] をクリックします。

先に進む前に、[**Nodes**] ページにセカンダリノードの同期状態が **SUCCESS** と表示されていることを確認してください。



> 注
>
> これで、セカンダリノードは、プライマリノードと同じログオン資格情報を持ちます。

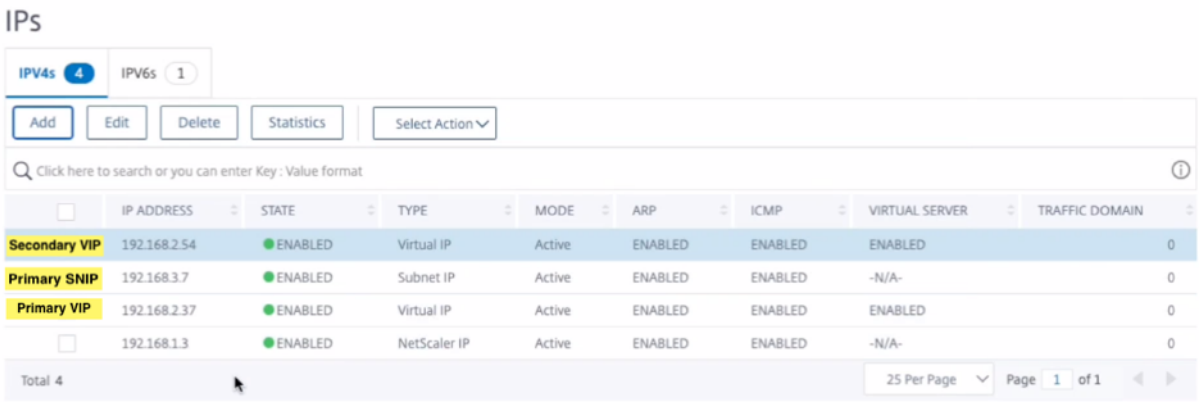
ステップ **2.** 両方のノードに仮想 IP アドレスとサブネット IP アドレスを追加します。

プライマリノードで、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** [IP] **>** [IPv4] に移動し、[追加] をクリックします。
2. 次の手順に従って、プライマリ VIP アドレスを追加します。
   a) プライマリ・インスタンスのクライアント側インターフェイスの内部 IP アドレスと、VM インスタンスのクライアント・サブネットに対して構成されたネットマスクを入力します。

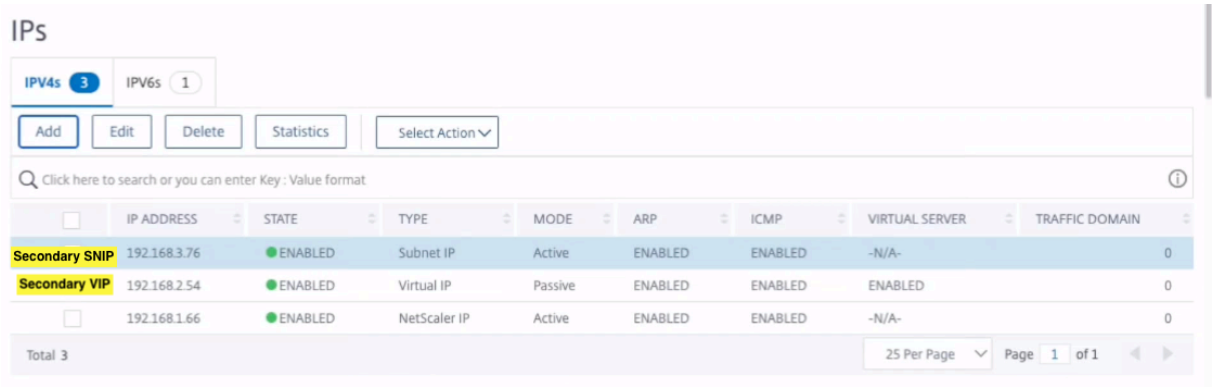
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
c) [作成] をクリックします。

3. 次の手順に従って、プライマリ SNIP アドレスを追加します。
   a) プライマリ・インスタンスのサーバ側インターフェイスの内部 IP アドレスと、プライマリ・インスタンスのサーバ・サブネットに対して構成されたネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) [作成] をクリックします。
4. 次の手順に従って、セカンダリ VIP アドレスを追加します。
   a) セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスと、VM インスタンスのクライアントサブネットに設定されたネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   c) [作成] をクリックします。



セカンダリノードで、次の手順を実行します。

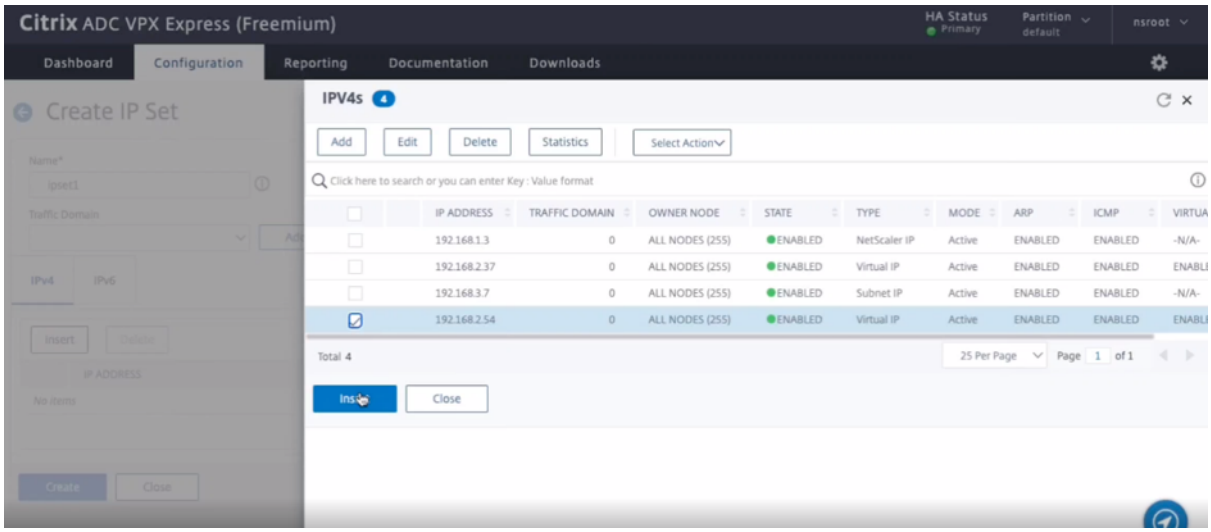
1. [システム] **>** [ネットワーク] **>** **[IP]** **>** **[IPv4]** に移動し、[追加] をクリックします。
2. 次の手順に従って、セカンダリ VIP アドレスを追加します。
   a) セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスと、VM インスタンスのクライアントサブネットに設定されたネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
3. 次の手順に従って、セカンダリ SNIP アドレスを追加します。
   a) セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスと、セカンダリインスタンスのサーバサブネットに設定されたネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) [作成] をクリックします。



ステップ **3**. IP セットを追加し、両方のインスタンスで IP セットをセカンダリ VIP にバインドします。

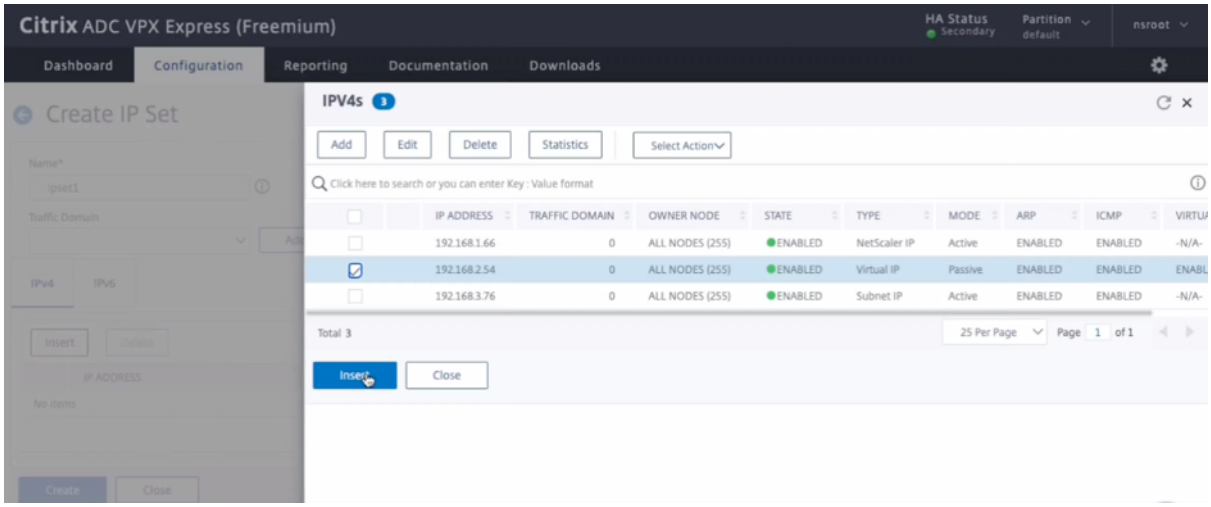
プライマリノードで、次の手順を実行します。

1. **System > Network > IP Sets > Add** に移動します。
2. IP セット名を追加し、[**Insert**] をクリックします。
3. [**IPv4**] ページで、仮想 IP（セカンダリ VIP）を選択し、[挿入] をクリックします。
4. [**Create**] をクリックして IP セットを作成します。



セカンダリノードで、次の手順を実行します。

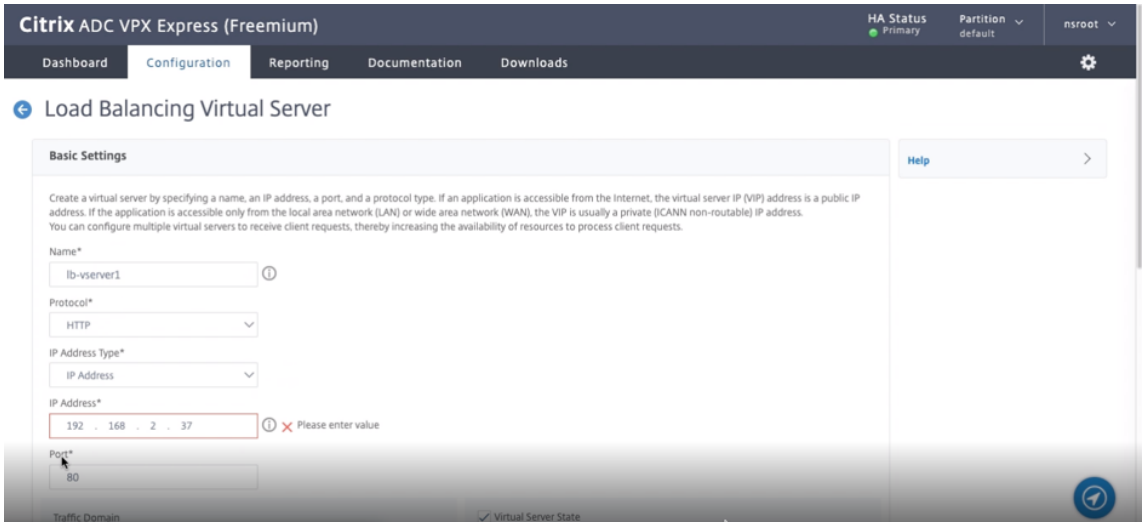
1. **System > Network > IP Sets > Add** に移動します。
2. IP セット名を追加し、[**Insert**] をクリックします。
3. [**IPv4**] ページで、仮想 IP（セカンダリ VIP）を選択し、[挿入] をクリックします。
4. [**Create**] をクリックして IP セットを作成します。



> 注
>
> IP セット名は、両方のインスタンスで同じである必要があります。

ステップ **4**。プライマリインスタンスに負荷分散仮想サーバーを追加します。

1. [設定] **>** [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [追加]
2. [名前]、[プロトコル]、[IP アドレスタイプ (IP アドレス)]、[IP アドレス] (プライマリ VIP)、および [ポート] に必要な値を追加します。



3. [詳細] クリックします。[**IP** 範囲 **IP** セット設定] に移動し、ドロップダウンメニューから [**IPSet**] を選択し、ステップ **3** で作成した IPSet を指定します。
4. **OK** をクリックして、負荷分散仮想サーバーを作成します。

ステップ **5**. プライマリノードにサービスまたはサービスグループを追加します。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

ステップ **6.** サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **4** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] タブで、[負荷分散仮想サーバーサービスバインドなし] をクリックします。
4. 手順 **5** で構成したサービスを選択し、[バインド] をクリックします。

構成を保存します。強制フェールオーバーの後、セカンダリは新しいプライマリになります。古いプライマリ VIP の外部スタティック IP は、新しいセカンダリ VIP に移動します。

CLI を使用した高可用性の設定

ステップ **1.** 両方のインスタンスで INC モードで高可用性をセットアップします。

プライマリノードで、次のコマンドを入力します。

```
add ha node 1 <sec_ip> -inc ENABLED
<!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

sec_ipは、セカンダリノードの管理 NIC の内部 IP アドレスを指します。

prim_ipは、プライマリノードの管理 NIC の内部 IP アドレスを指します。

ステップ **2.** 両方のノードに仮想 IP とサブネット IP を追加します。

プライマリノードで、次のコマンドを入力します。

```
add ns ip <primary_vip> <subnet> -type VIP

add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <primary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

primary_vipは、プライマリ・インスタンスのクライアント側インターフェイスの内部 IP アドレスを指します。

secondary_vipは、セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスを指します。

primary_snipは、プライマリ・インスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

セカンダリノードで、次のコマンドを入力します。

```
add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <secondary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

secondary_vipは、セカンダリインスタンスのクライアント側インターフェイスの内部 IP アドレスを指します。

secondary_snipは、セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

ステップ **3**. IP セットを追加し、両方のインスタンスで IP セットをセカンダリ VIP にバインドします。

プライマリノードで、次のコマンドを入力します。

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
<!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
<!--NeedCopy-->
```

> 注
>
> IP セット名は、両方のインスタンスで同じである必要があります。

ステップ **4**。プライマリ・インスタンスに仮想サーバを追加します。

次のコマンドを入力します。

```
add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port
    > -ipset <ipset_name>
<!--NeedCopy-->
```

ステップ **5**. プライマリインスタンスにサービスまたはサービスグループを追加します。

次のコマンドを入力します。


```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

ステップ **6.** サービス/サービスグループをプライマリインスタンス上の負荷分散仮想サーバーにバインドします。

次のコマンドを入力します。

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> 注:
>
> 設定を保存するには、コマンド`save config`を入力します。そうしないと、インスタンスの再起動後に設定が失われます。

ステップ **7.** 設定を確認します。

プライマリクライアント NIC に接続されている外部 IP アドレスが、フェールオーバー時にセカンダリに移動することを確認します。

1. 外部 IP アドレスに cURL 要求を行い、それが到達可能であることを確認します。
2. プライマリインスタンスで、フェイルオーバーを実行します。

   GUI から、[設定] **>** [システム] **>** [高可用性] **>** [アクション] **>** [強制フェールオーバー] に移動します。

   CLI から、次のコマンドを入力します。

   ```
   force ha failover -f
   <!--NeedCopy-->
   ```

   GCP コンソールで、セカンダリインスタンスに移動します。外部 IP アドレスは、フェールオーバー後にセカンダリのクライアント NIC に移動されている必要があります。
3. 外部 IP に cURL 要求を発行し、再び到達可能であることを確認します。

プライベート **IP** アドレスを持つ **VPX** 高可用性ペアを **Google Cloud Platform** にデプロイする

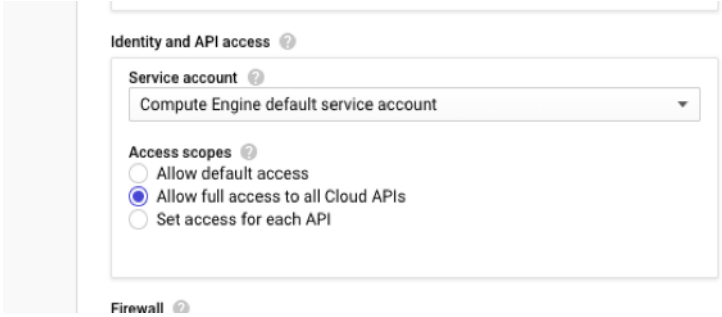
March 8, 2022

プライベート IP アドレスを使用して、VPX 高可用性ペアを GCP にデプロイできます。クライアント IP（VIP）は、プライマリノードのエイリアス IP アドレスとして設定する必要があります。フェールオーバー時に、クライアント IP アドレスがセカンダリノードに移動され、トラフィックが再開されます。

高可用性の詳細については、「高可用性」を参照してください。

はじめに

- Google Cloud Platform への Citrix ADC VPX インスタンスのデプロイで説明されている制限、ハードウェア要件、注意点をお読みください。この情報は、高可用性展開にも適用されます。
- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。
- インスタンスの作成中に、すべての Cloud API へのフルアクセスを許可します。



- GCP サービスアカウントに次の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [
 “compute.forwardingRules.list” ,
 “compute.forwardingRules.setTarget” ,
 “compute.instances.setMetadata” ,
”compute.instances.get”,
”compute.instances.list”,
”compute.instances.updateNetworkInterface”,
 “compute.targetInstances.list” ,
 “compute.targetInstances.use” ,
”compute.zones.list”,
]
<!--NeedCopy-->
```

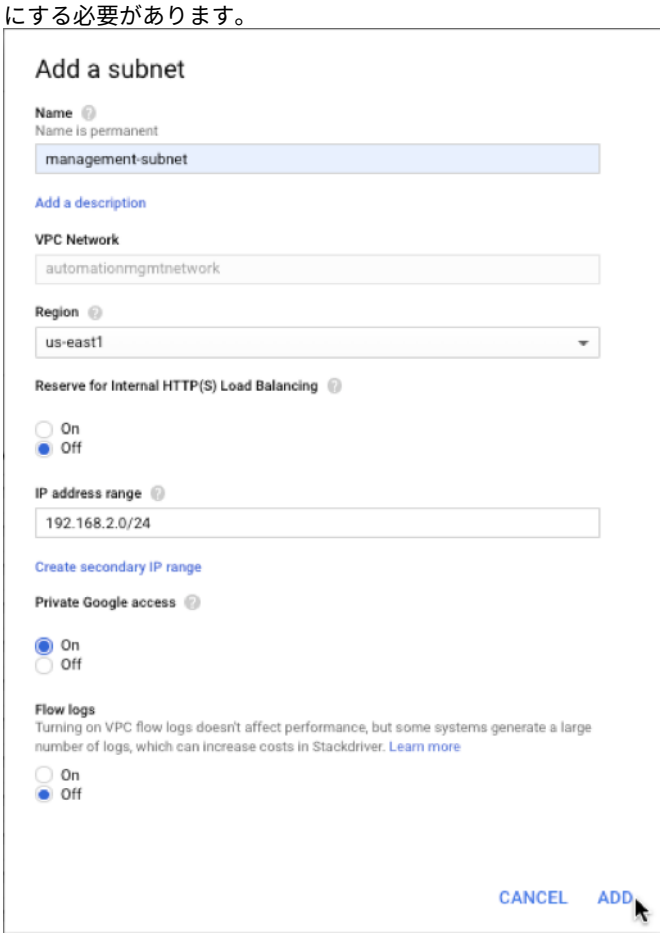
- 管理インターフェイス以外のインターフェイスに外部 IP アドレスを設定している場合は、GCP サービスアカウントに次の追加の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [
”compute.addresses.use”
```



```
3   ”compute.instances.addAccessConfig”,
4   ”compute.instances.deleteAccessConfig”,
5   ”compute.networks.useExternalIp”,
6   ”compute.subnetworks.useExternalIp”,
7   ]
8   <!--NeedCopy-->
```

- 仮想マシンにインターネットアクセスがない場合は、管理サブネットでプライベート **Google Access** を有効にする必要があります。



- プライマリノードで GCP 転送ルールを構成した場合は、「GCP での VPX 高可用性ペアの転送ルールのサポート」に記載されている制限と要件を読み、フェールオーバー時に新しいプライマリに更新します。

VPX 高可用性ペアを Google Cloud Platform にデプロイする方法

ここでは、高可用性展開手順の概要を示します。

1. 同じリージョンに VPC ネットワークを作成します。たとえば、アジア東です。
2. 同じリージョンに 2 つの VPX インスタンス (プライマリノードとセカンダリノード) を作成します。同じゾーンまたは異なるゾーンに配置できます。たとえば、アジア東-1a、アジア東-Ib。
3. Citrix ADC GUI または ADC CLI コマンドを使用して、両方のインスタンスで高可用性設定を構成します。

手順 **1**：**VPC** ネットワークを作成する

要件に基づいて VPC ネットワークを作成します。管理 NIC、クライアント NIC、サーバー NIC に関連付けるために 3 つの VPC ネットワークを作成することをお勧めします。

VPC ネットワークを作成するには、次の手順を実行します。

1. **Google** コンソール **>** ネットワーク **> VPC** ネットワーク **> VPC** ネットワークの作成にログオンします。
2. 必須フィールドに入力し、[**Create**] をクリックします。

詳細については、「Google Cloud Platform での Citrix ADC VPX インスタンスのデプロイ」の「**VPC** ネットワークの作成」セクションを参照してください。

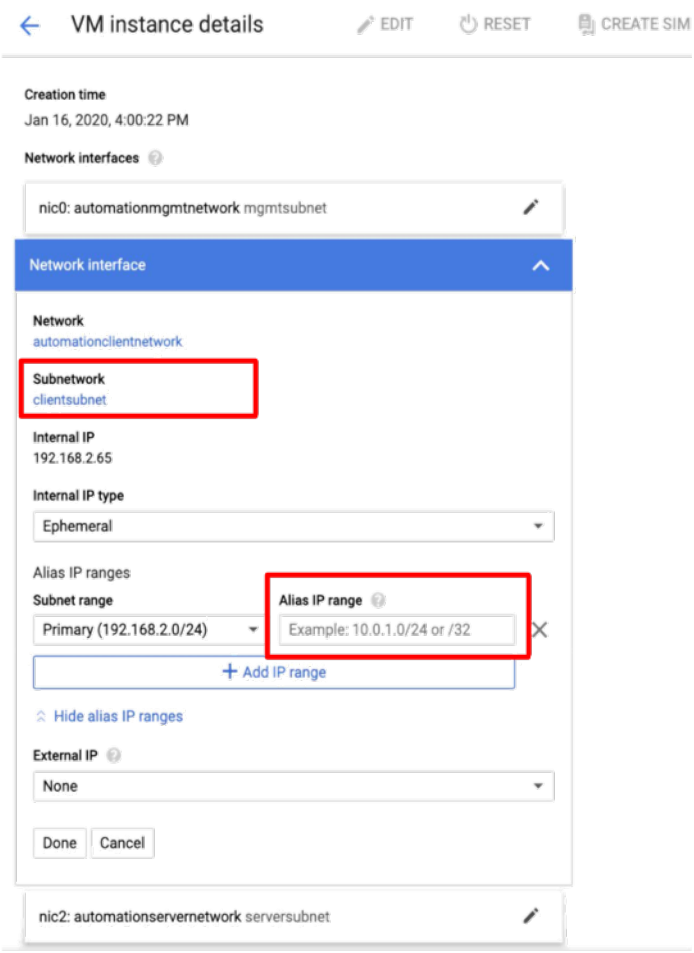
手順 **2**：**2** つの **VPX** インスタンスを作成する

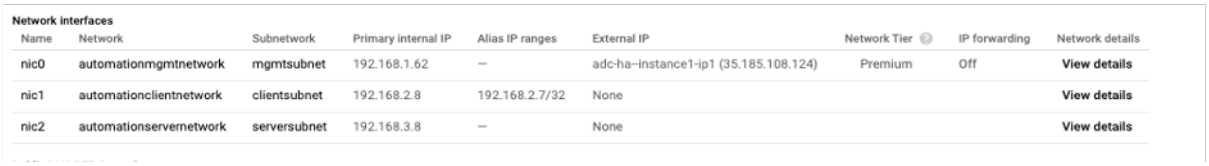
シナリオ：Multi-NIC、Multi-IP スタンドアロン VPX インスタンスを展開するの手順に従って、VPX インスタンスを 2 つ作成します。

> 重要：
>
> クライアントエイリアス IP アドレスをプライマリノードに割り当てます。VPX インスタンスの内部 IP アドレスを使用して VIP を構成しないでください。

クライアントエイリアス IP アドレスを作成するには、次の手順を実行します。

1. VM インスタンスに移動し、[編集] をクリックします。
2. [ネットワークインターフェイス（Network Interface）] ウィンドウで、クライアントインターフェイスを編集します。
3. [エイリアス **IP** 範囲（Alias IP range）] フィールドに、クライアントエイリアス IP アドレスを入力します。





Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	automationmgmtnetwork	mgmtsubnet	192.168.1.62	–	adc-ha--instance1-ip1 (35.185.108.124)	Premium	Off	View details
nic1	automationclientnetwork	clientsubnet	192.168.2.8	192.168.2.7/32	None			View details
nic2	automationservernetwork	serversubnet	192.168.3.8	–	None			View details

フェールオーバー後、古いプライマリが新しいセカンダリになると、エイリアス IP アドレスは古いプライマリから移動し、新しいプライマリに接続されます。

VPX インスタンスを構成したら、仮想 (VIP) アドレスとサブネット IP (SNIP) アドレスを構成できます。詳細については、「Citrix ADC 所有の IP アドレスの構成」を参照してください。

手順 **3**：高可用性の構成

Google Cloud Platform でインスタンスを作成した後、Citrix ADC GUI または CLI を使用して高可用性を構成できます。

GUI を使用した高可用性の構成

ステップ **1.** 両方のノードで INC Enabled モードで高可用性を設定します。

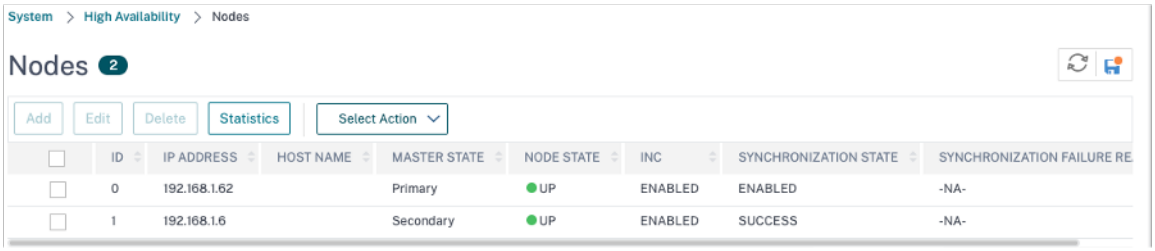
プライマリノードで、次の手順を実行します。

1. GCP Console `nsroot` からノードのユーザー名とインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノードの **IP** アドレス] フィールドに、セカンダリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [**Create**] をクリックします。

セカンダリノードで、次の手順を実行します。

1. GCP Console `nsroot` からノードのユーザー名とインスタンス ID をパスワードとしてインスタンスにログオンします。
2. 構成 **>** システム **>** 高可用性 **>** ノードに移動し、追加をクリックします。
3. [リモートノード **IP** アドレス] フィールドに、プライマリノードの管理 NIC のプライベート IP アドレスを入力します。
4. [セルフノードで **INC (**独立ネットワーク構成**)** モードをオンにする] チェックボックスをオンにします。
5. [**Create**] をクリックします。

先に進む前に、[**Nodes**] ページにセカンダリノードの同期状態が **SUCCESS** と表示されていることを確認してください。



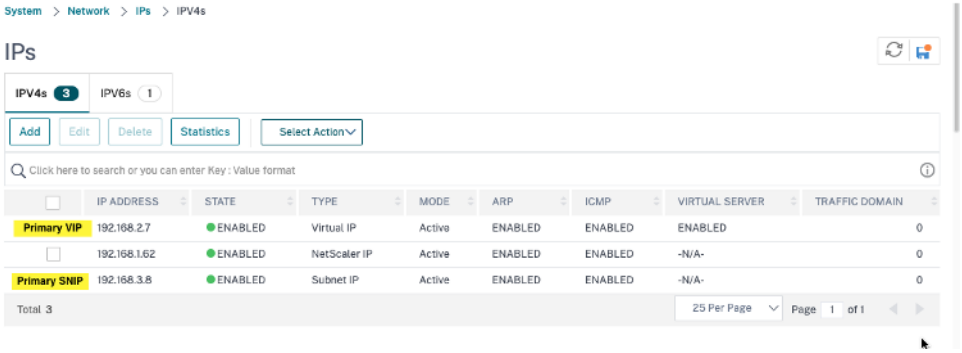
> 注
>
> これで、セカンダリノードは、プライマリノードと同じログオン資格情報を持ちます。

ステップ **2.** 両方のノードに仮想 IP アドレスとサブネット IP アドレスを追加します。

プライマリノードで、次の手順を実行します。

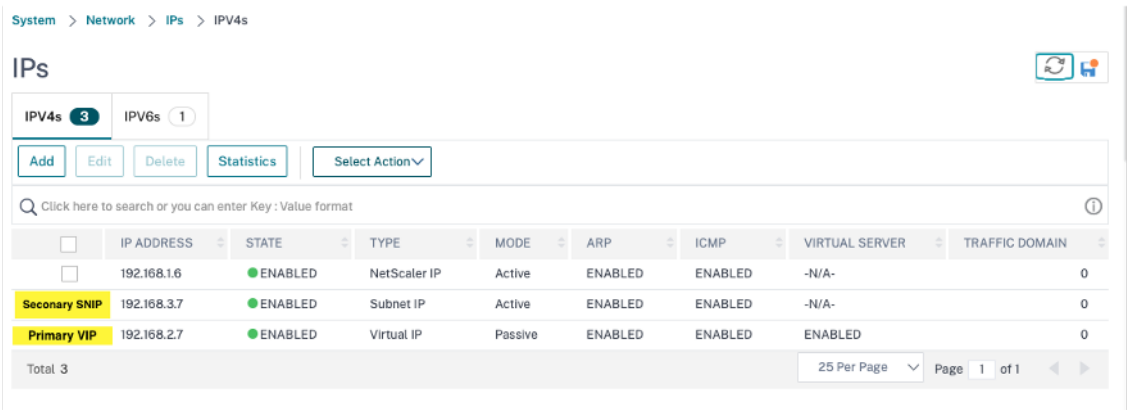
1. [システム**] > [**ネットワーク**] > [IP] > [IPv4**] に移動し、[追加] をクリックします。
2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。
   a) 仮想マシンインスタンスのクライアントサブネットに設定されたエイリアス IP アドレスとネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Virtual IP**] を選択します。
   c) [**Create**] をクリックします。
3. サーバ IP (SNIP) アドレスを作成するには、次の手順を実行します。

a) プライマリ・インスタンスのサーバ側インターフェイスの内部 IP アドレスと、サーバ・サブネットに設定されたネットマスクを入力します。
b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
c) [**Create**] をクリックします。



セカンダリノードで、次の手順を実行します。

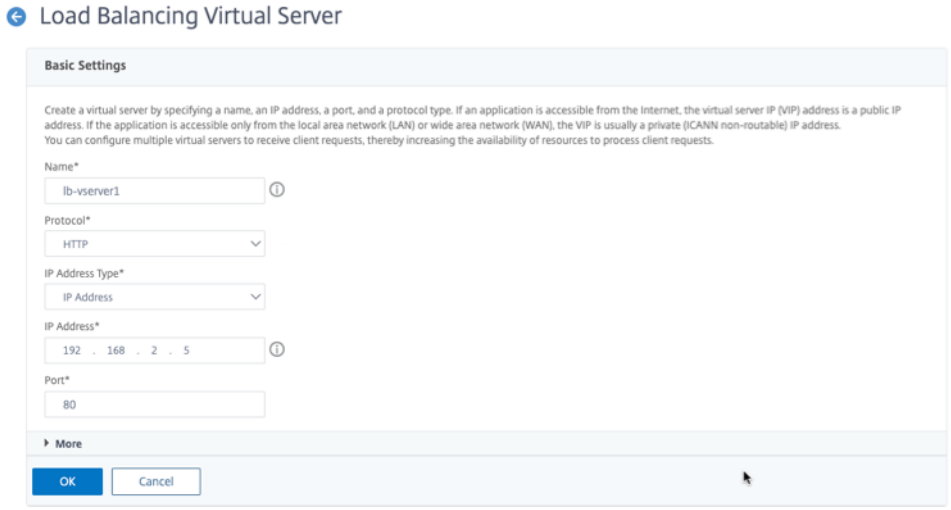
1. [システム] **>** [ネットワーク] **>** [**IP**] **>** [**IPv4**] に移動し、[追加] をクリックします。
2. クライアントエイリアス IP（VIP）アドレスを作成するには、次の手順を実行します。
   a) プライマリ VM インスタンスのクライアントサブネットに対して構成された Alias IP アドレスとネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) [**Create**] をクリックします。
3. サーバ IP (SNIP) アドレスを作成するには、次の手順を実行します。
   a) セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスと、サーバサブネットに設定されたネットマスクを入力します。
   b) [**IP Type**] フィールドで、ドロップダウンメニューから [**Subnet IP**] を選択します。
   c) [**Create**] をクリックします。



ステップ **3**. プライマリノードに負荷分散仮想サーバーを追加します。

1. [設定] **>** [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] **>** [追加] に移動します。

2. 名前、プロトコル、IP アドレスタイプ（IP アドレス）、IP アドレス（プライマリクライアントエイリアス IP アドレス）、およびポートに必要な値を追加し、「**OK**」をクリックします。



ステップ **4**. プライマリノードにサービスまたはサービスグループを追加します。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

ステップ **5**. サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **3** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] タブで、[負荷分散仮想サーバーサービスバインドなし] をクリックします。
4. 手順 **4** で構成したサービスを選択し、[バインド] をクリックします。

ステップ **5**. 構成を保存します。

強制フェールオーバーの後、セカンダリは新しいプライマリになります。古いプライマリからのクライアントエイリアス IP (VIP) とサーバエイリアス IP (SNIP) が新しいプライマリに移動します。

CLI を使用した高可用性の設定

ステップ **1**. Citrix ADC CLI を使用して、両方のインスタンスで **INC** 対応モードで高可用性を設定します。

プライマリノードで、次のコマンドを入力します。

```
1 add ha node 1 <sec_ip> -inc ENABLED
2 <!--NeedCopy-->
```

セカンダリノードで、次のコマンドを入力します。


```
1 add ha node 1 <prim_ip> -inc ENABLED
2 <!--NeedCopy-->
```

sec_ipは、セカンダリノードの管理 NIC の内部 IP アドレスを指します。

prim_ipは、プライマリノードの管理 NIC の内部 IP アドレスを指します。

ステップ **2.** 両方のノードに VIP と SNIP を追加します。

プライマリノードで次のコマンドを入力します。

```
1 add ns ip <primary_client_alias_ip> <subnet> -type VIP
2
3 <!--NeedCopy-->
```

> 注:
>
> 仮想マシンインスタンスのクライアントサブネットに設定されたエイリアス IP アドレスとネットマスクを入力します。

```
1 add ns ip <primary_snip> <subnet> -type SNIP
2 <!--NeedCopy-->
```

primary_snipは、プライマリインスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

セカンダリノードで次のコマンドを入力します。

```
1 add ns ip <primary_client_alias_ip> <subnet> -type VIP
2 <!--NeedCopy-->
```

> 注
>
> プライマリ VM インスタンスのクライアントサブネットに対して構成された Alias IP アドレスとネットマスクを入力します。

```
1 add ns ip <secondary_snip> <subnet> -type SNIP
2 <!--NeedCopy-->
```

secondary_snipは、セカンダリインスタンスのサーバ側インターフェイスの内部 IP アドレスを指します。

> 注:
>
> VM インスタンスのサーバサブネットに設定された IP アドレスとネットマスクを入力します。

ステップ **3**. プライマリノードに仮想サーバを追加します。

次のコマンドを入力します。

```
add <server_type> vserver <vserver_name> <protocol> <
   primary_client_alias_ip> <port>
<!--NeedCopy-->
```

ステップ **4**. プライマリノードにサービスまたはサービスグループを追加します。

次のコマンドを入力します。

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

ステップ **5**. サービスまたはサービスグループをプライマリノードの負荷分散仮想サーバーにバインドします。

次のコマンドを入力します。

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> 注:
>
> 設定を保存するには、コマンド`save config`を入力します。そうしないと、インスタンスの再起動後に設定が失われます。

バックエンド **GCP** 自動スケーリングサービスの追加

October 7, 2021

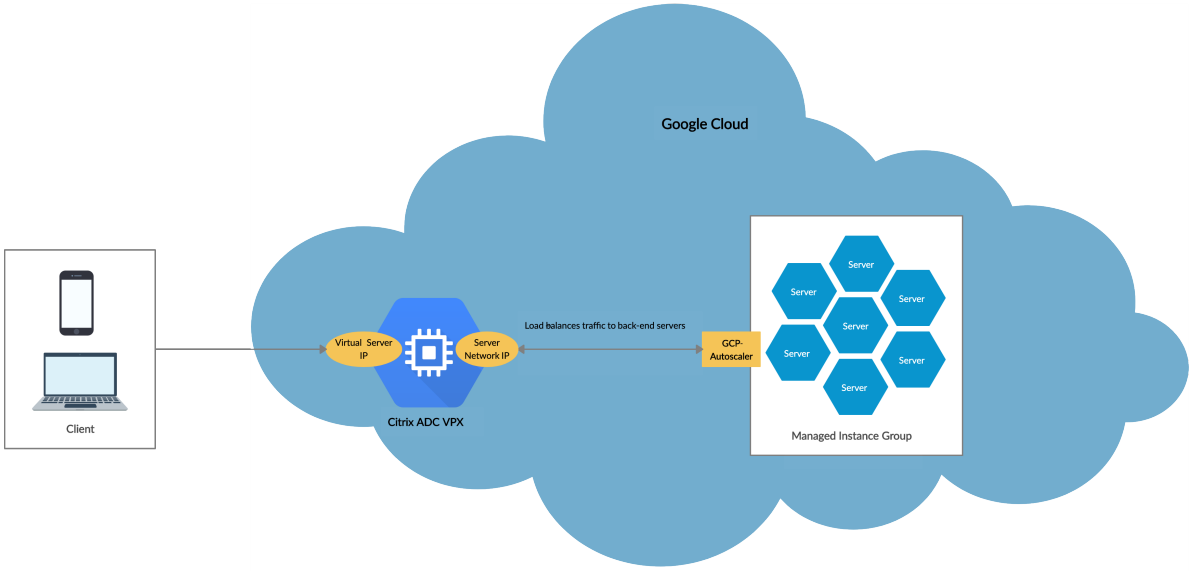
クラウドでアプリケーションを効率的にホスティングするには、アプリケーションのニーズに応じて、簡単で費用対効果の高いリソース管理が必要です。増大する需要に対応するには、ネットワークリソースを拡大する必要があります。需要が縮小したら、十分に活用されていないリソースの不要なコストを避けるために、スケールダウンする必要があります。アプリケーションを実行するコストを最小限に抑えるには、トラフィック、メモリ、CPU の使用率などを常に監視する必要があります。しかし、トラフィックを手動で監視するのは煩雑です。アプリケーション環境を動

的にスケールアップまたはスケールダウンするには、トラフィックの監視プロセスを自動化し、必要に応じてリソースを増減する必要があります。

GCP オートスケーリングサービスと統合された Citrix ADC VPX インスタンスには次の利点があります。

- 負荷分散と管理: 需要に応じて、サーバーのスケールアップとスケールダウンを自動構成します。VPX インスタンスは、バックエンドサブネット内の管理対象インスタンスグループを自動的に検出し、負荷を分散する管理対象インスタンスグループを選択できます。仮想アドレスとサブネット IP アドレスは、VPX インスタンスで自動的に設定されます。
- 高可用性: 複数のゾーンにまたがるマネージドインスタンスグループを検出し、サーバーの負荷分散を行います。
- ネットワーク可用性の向上: VPX インスタンスは次の機能をサポートします。
  - 同じ配置グループのバックエンドサーバー
  - 異なるゾーンにあるバックエンドサーバー

この図は、負荷分散仮想サーバーとして機能する Citrix ADC VPX インスタンスで、GCP Auto Scaling サービスがどのように機能するかを示しています。

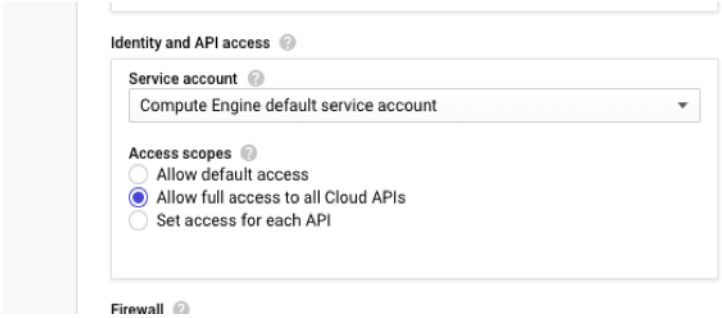


はじめに

Citrix ADC VPX インスタンスで Autoscaling を使用する前に、次のタスクを完了する必要があります。

- 要件に応じて、GCP 上に Citrix ADC VPX インスタンスを作成します。
  - Citrix ADC VPX インスタンスの作成方法の詳細については、「Google Cloud Platform での Citrix ADC VPX インスタンスの展開」を参照してください。
  - VPX インスタンスを HA モードでデプロイする方法の詳細については、「VPX 高可用性ペアを Google Cloud Platform にデプロイする」を参照してください。
- GCP プロジェクトで クラウドリソースマネージャー **API** を有効にします。

- インスタンスの作成中に、すべての Cloud API へのフルアクセスを許可します。



- GCP サービスアカウントに次の IAM 権限があることを確認します。

```
REQUIRED_INSTANCE_IAM_PERMS = [

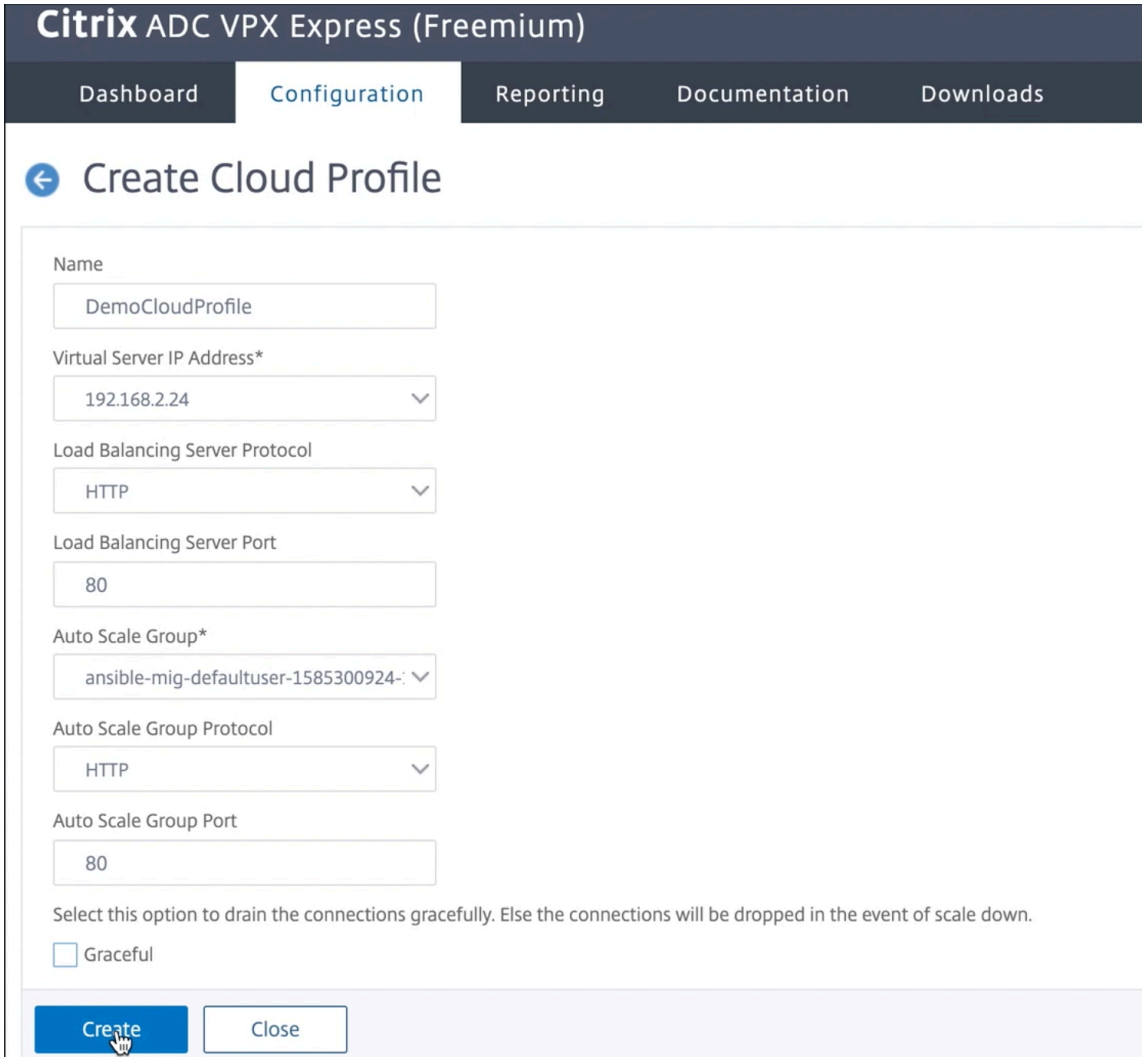
”compute.instances.get”,
”compute.zones.list”,
”compute.instanceGroupManagers.list”,
”compute.instanceGroupManagers.get”
]
<!--NeedCopy-->
```

- Auto Scaling を設定するには、次の項目が設定されていることを確認します。
  - インスタンス・テンプレート
  - 管理対象インスタンスグループ
  - 自動スケーリングポリシー

Citrix ADC VPX インスタンスへの GCP 自動スケーリングサービスの追加

GUI を使用して、ワンクリックで Autoscaling サービスを VPX インスタンスに追加できます。以下の手順を実行して、Autoscaling サービスを VPX インスタンスに追加します。

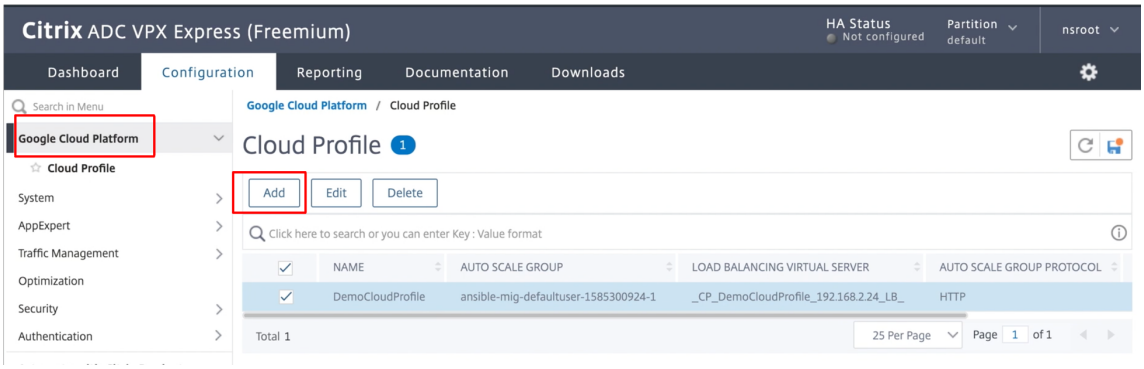
1. nsrootの資格情報を使用して、VPX インスタンスにログオンします。
2. Citrix ADC VPX インスタンスに初めてログオンすると、デフォルトのクラウドプロファイルページが表示されます。ドロップダウンメニューから GCP 管理対象インスタンスグループを選択し、[**Create**] をクリックしてクラウドプロファイルを作成します。



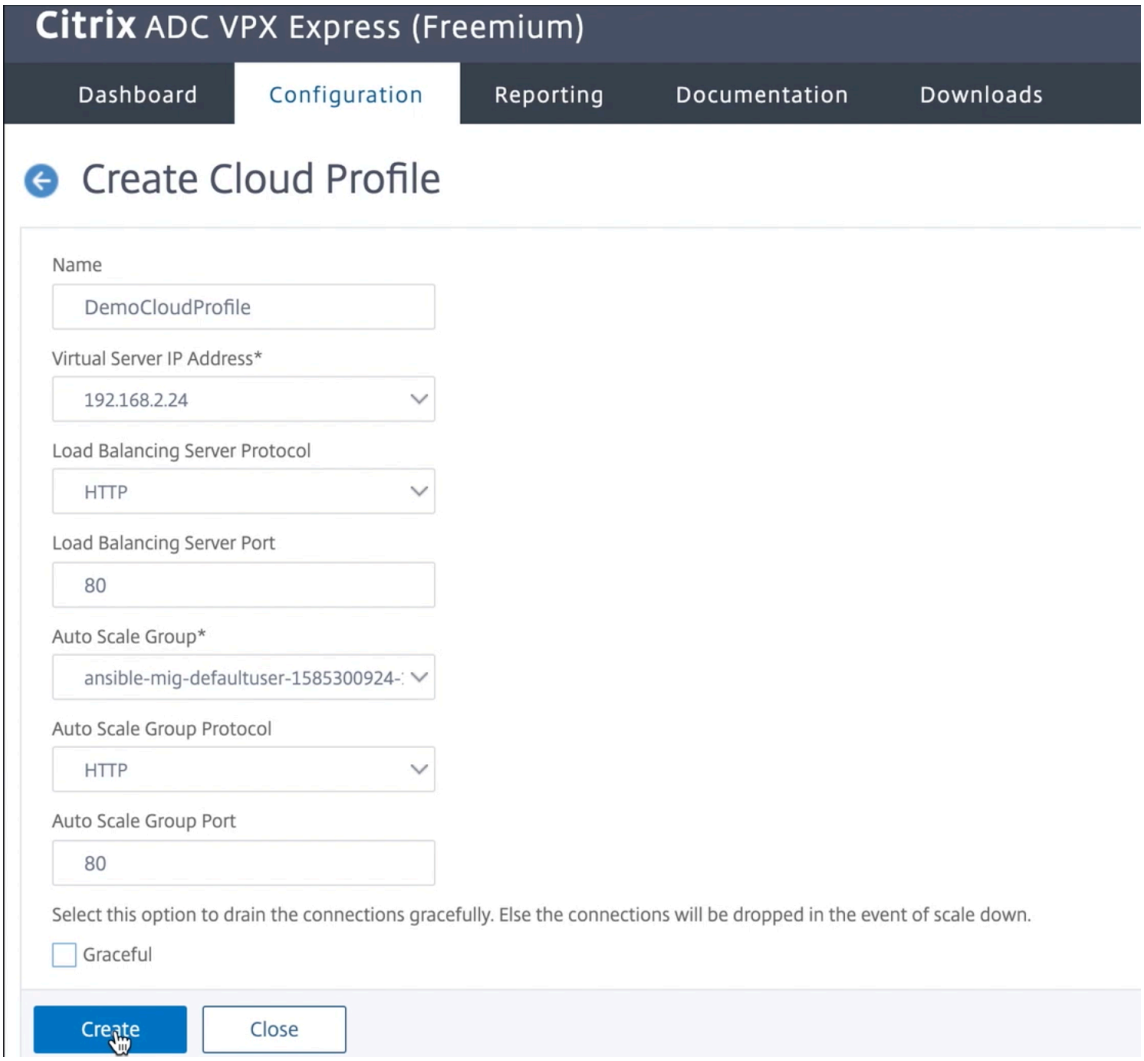
- [**Virtual Server IP Address**] フィールドは、インスタンスに関連付けられたすべての IP アドレスから自動的に入力されます。
- **Autoscale** グループは、GCP アカウントで設定されたマネージドインスタンスグループから事前設定されています。
- [自動スケールグループプロトコル] と [自動スケールグループポート] を選択するときは、サーバが構成済みのプロトコルとポートでリッスンしていることを確認します。サービスグループに適切なモニタをバインドします。デフォルトでは、TCP モニターが使用されます。
- サポートされていないため、[**Graceful**] チェックボックスをオフにします。

> 注:
>
> SSL プロトコルタイプの Auto Scaling では、クラウドプロファイルの作成後に、証明書がないために負荷分散仮想サーバーまたはサービスグループがダウンしています。証明書は、仮想サーバまたはサービスグループに手動でバインドできます。

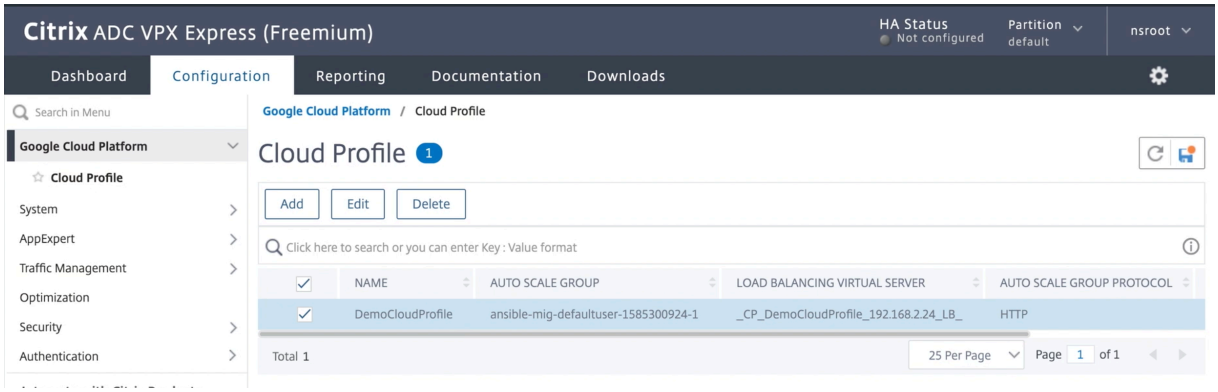
3. クラウドプロファイルを作成する場合は、初回ログオン後、GUI で [システム] **> [Google Cloud Platform] > [**クラウドプロファイル] に移動し、[追加] をクリックします。



[クラウドプロファイルの作成] 設定ページが表示されます。



クラウドプロファイルは、Citrix ADC 負荷分散仮想サーバーと、管理対象インスタンスグループのサーバーとしてメンバーを持つサービスグループを作成します。バックエンドサーバーは、VPX インスタンスで構成された SNIP を介して到達可能である必要があります。



GCP での Citrix ADC VPX インスタンスの VIP スケーリングサポート

October 7, 2021

NetScaler アプライアンスはクライアントとサーバーの間に設置され、クライアント要求とサーバー応答は Citrix ADC アプライアンスを経由します。一般的な設置では、アプライアンス上で構成された仮想サーバーによって接続ポイントが提供され、クライアントはこれを使用してアプライアンスの背後にあるアプリケーションにアクセスします。展開に必要なパブリック仮想 IP（VIP）アドレスの数は、ケースバイケースで異なります。

GCP アーキテクチャでは、インスタンスの各インターフェイスが異なる VPC に接続されるように制限します。GCP 上の VPC はサブネットの集合であり、各サブネットはリージョンのゾーンにまたがることができます。さらに、GCP には次の制限があります。

- パブリック IP アドレス数と NIC の数が 1:1 でマッピングされています。NIC に割り当てることができるパブリック IP アドレスは 1 つだけです。
- 大容量のインスタンスタイプには最大 8 つの NIC しか接続できません。

たとえば、n1-standard-2 インスタンスは 2 つの NIC しか持つことができず、追加できるパブリック VIP は 2 つに制限されています。詳細については、「VPC リソースクォータ」を参照してください。

Citrix ADC VPX インスタンスでより大規模なパブリック仮想 IP アドレスを実現するために、インスタンスのメタデータの一部として VIP アドレスを構成できます。ADC VPX インスタンスは、GCP によって提供される転送ルールを内部的に使用して、VIP のスケーリングを実現します。ADC VPX インスタンスは、構成された VIP に高可用性も提供します。
メタデータの一部として VIP アドレスを設定した後、転送ルールの作成に使用するのと同じ IP を使用して LB 仮想サーバを設定できます。したがって、転送ルールを使用して、GCP の ADC VPX インスタンスでパブリック VIP アドレスを使用する際のスケールの制限を緩和できます。

転送ルールの詳細については、「転送ルールの概要」を参照してください。

HA の詳細については、「ハイアベイラビリティ」を参照してください。

注意事項

- Google は、各仮想 IP 転送ルールに対して追加費用を請求します。実際のコストは、作成されるエントリの数によって異なります。関連するコストは、Google の価格設定ドキュメントから確認できます。
- 転送ルールは、パブリック VIP にのみ適用されます。展開でプライベート IP アドレスが VIP として必要な場合は、エイリアス IP アドレスを使用できます。
- 転送ルールは、LB 仮想サーバーを必要とするプロトコルに対してのみ作成できます。VIP は、その場で作成、更新、または削除できます。同じ VIP アドレスを持つが、プロトコルが異なる新しい負荷分散仮想サーバーを追加することもできます。

はじめに

- Citrix ADC VPX インスタンスは、GCP にデプロイする必要があります。
- 外部 IP アドレスは予約する必要があります。詳細については、「静的外部 IP アドレスの予約」を参照してください。
- GCP サービスアカウントに次の IAM 権限があることを確認します。

```
REQUIRED_IAM_PERMS = [
”compute.addresses.list”,
”compute.addresses.get”,
”compute.addresses.use”,
”compute.forwardingRules.create”,
”compute.forwardingRules.delete”,
”compute.forwardingRules.get”,
”compute.forwardingRules.list”,
”compute.instances.use”,
”compute.subnetworks.use”,
”compute.targetInstances.create”
”compute.targetInstances.get”
”compute.targetInstances.use”,
]

<!--NeedCopy-->
```

Citrix ADC VPX インスタンスでの **VIP** スケーリング用の外部 **IP** アドレスを構成する

1. Google Cloud コンソールで、[**VM** インスタンス] ページに移動します。
2. 新しい VM インスタンスを作成するか、既存のインスタンスを使用します。
3. インスタンス名をクリックします。**VM** インスタンスの詳細ページで、[編集] をクリックします。

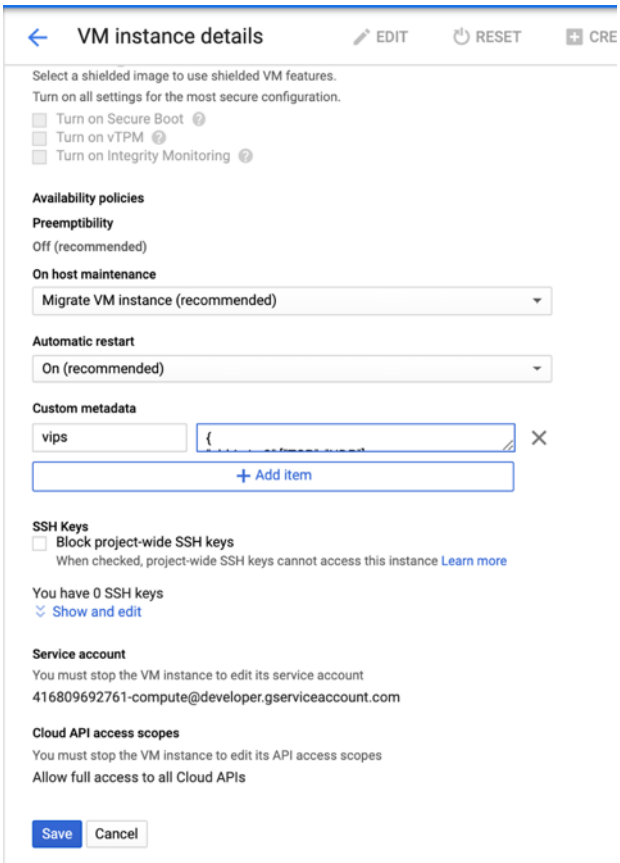
4. 次のように入力して、カスタムメタデータを更新します。

   - キー = VIP
   - 値 = 次の JSON 形式で値を指定します。

     {
     “Name of external reserved IP”: [list of protocols],
     }

   GCP は次のプロトコルをサポートしています。

   - AH
   - ESP
   - ICMP
   - SCT
   - TCP
   - UDP



詳細については、「カスタムメタデータ」を参照してください。

カスタムメタデータの例:

{
“external-ip1-name”:[“TCP”, “UDP”],

“external-ip2-name”:[“ICMP”, “AH”]
}

この例では、ADC VPX インスタンスは、IP、プロトコルペアごとに 1 つの転送ルールを内部的に作成します。メタデータエントリは、転送ルールにマッピングされます。この例では、メタデータエントリに対して作成される転送ルールの数を把握するのに役立ちます。

次の 4 つの転送ルールが作成されます。

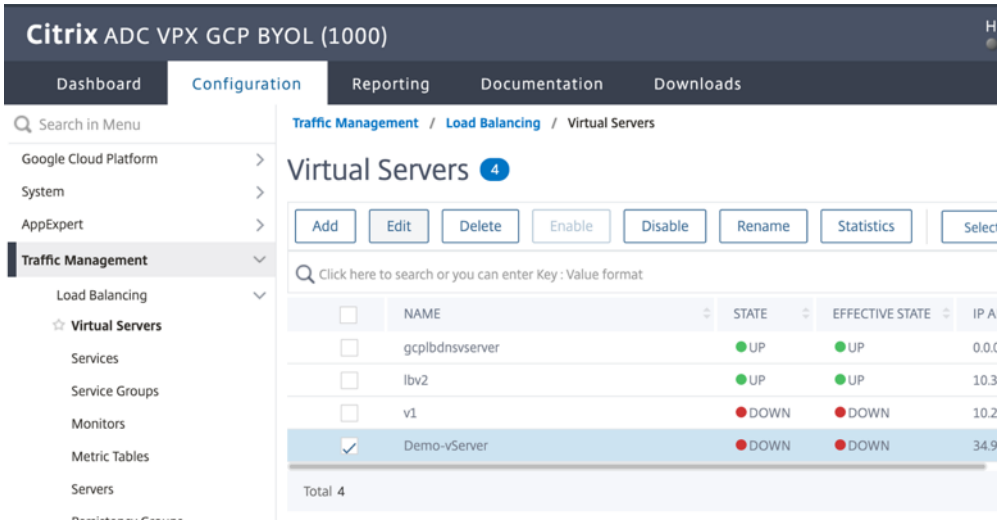
a) 外部 ip1 名と TCP
b) 外部 ip1 名と UDP
c) external-ip2-name と ICMP
d) external-ip2-name と AH

5. ［保存］をクリックします。

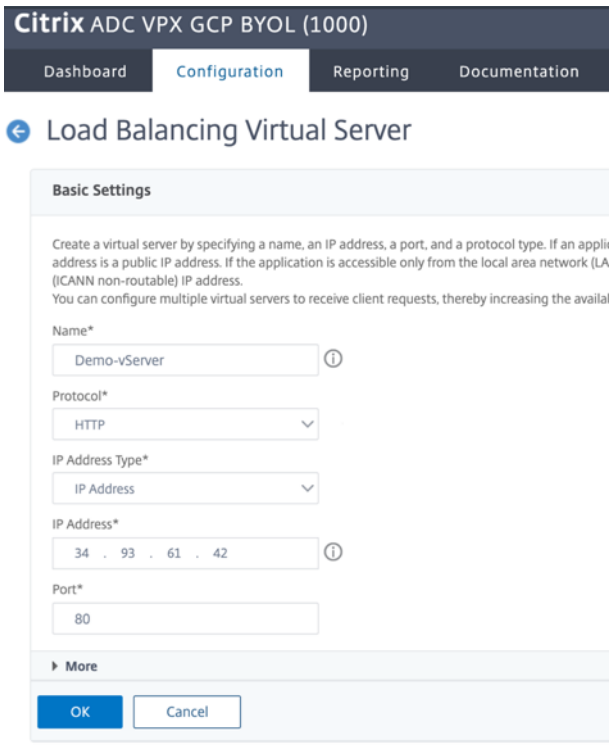
Citrix ADC VPX インスタンスで外部 **IP** アドレスを使用した負荷分散仮想サーバーのセットアップ

ステップ **1.** 負荷分散仮想サーバーを追加します。

1. ［設定］**>** **[**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**] > [**追加**]**

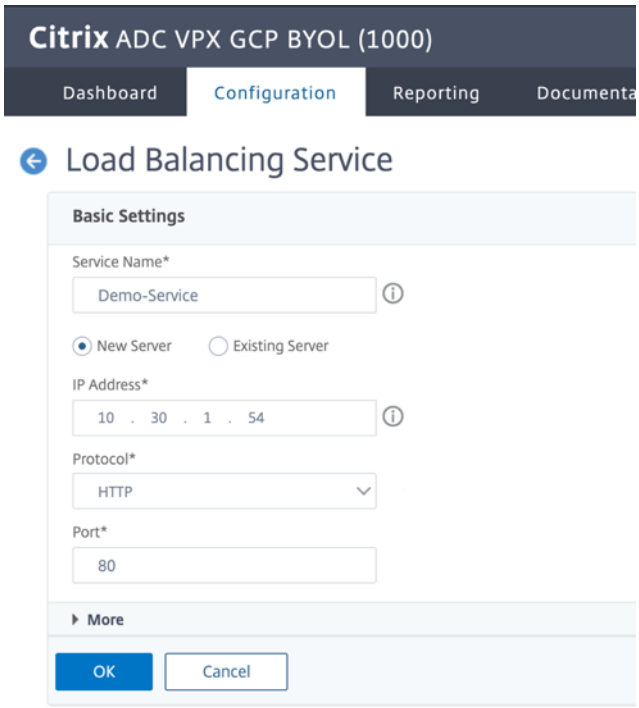


2. 名前、プロトコル、IP アドレスタイプ（IP アドレス）、IP アドレス（ADC で VIP として追加される転送ルールの外部 IP アドレス）、およびポートに必要な値を追加し、「**OK**」をクリックします。



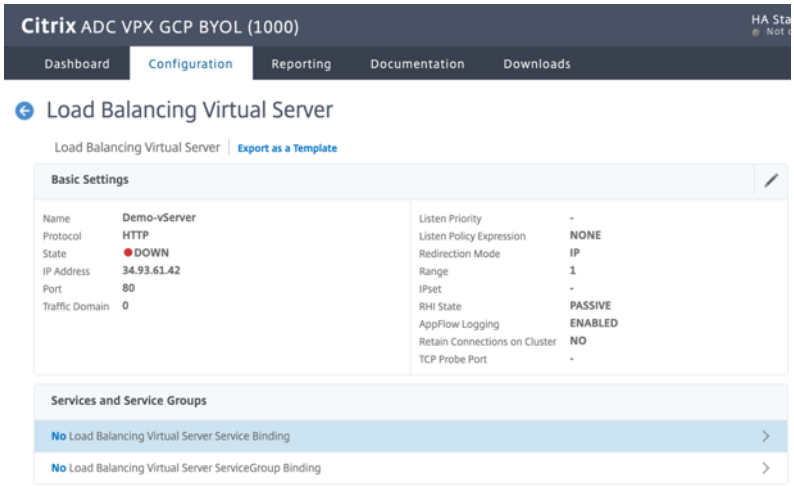
ステップ **2**. サービスまたはサービスグループを追加します。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**サービス**] >[**追加**]** に移動します。
2. サービス名、IP アドレス、プロトコル、およびポートに必要な値を追加し、[**OK**] をクリックします。

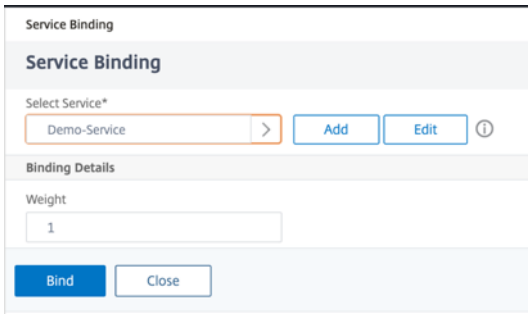


ステップ **3**. サービスまたはサービスグループを負荷分散仮想サーバーにバインドします。

1. **[設定] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。
2. 手順 **1** で構成した負荷分散仮想サーバーを選択し、[編集] をクリックします。
3. [サービスとサービスグループ] ページで、[負荷分散仮想サーバーサービスのバインドなし] をクリックします。



4. 手順 **3** で構成したサービスを選択し、[バインド] をクリックします。



5. 構成を保存します。

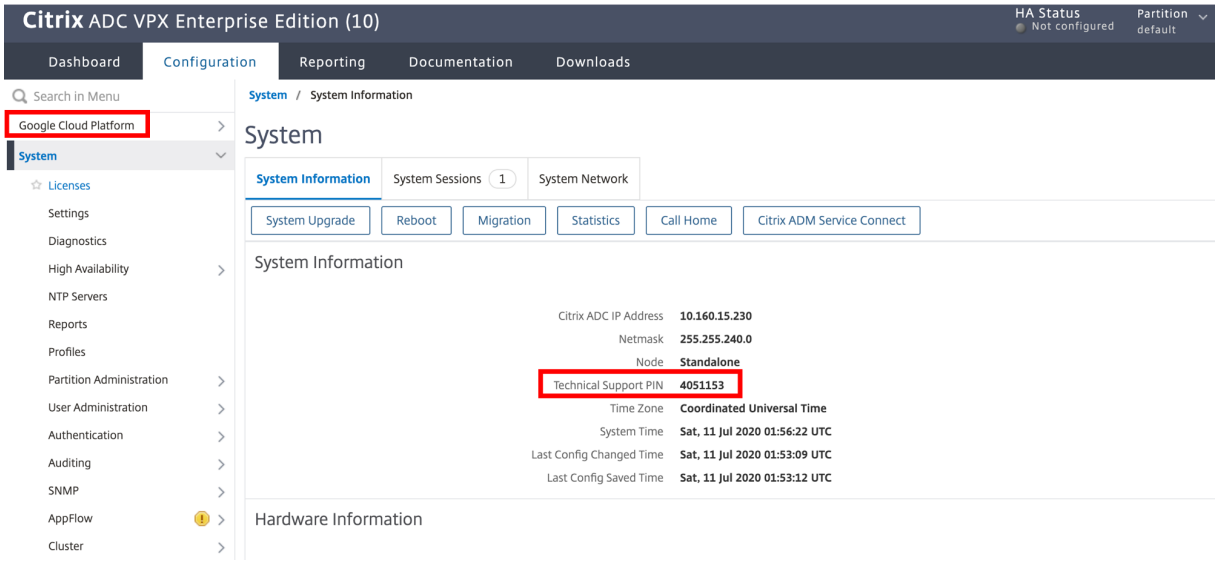
GCP での VPX インスタンスのトラブルシューティング

October 7, 2021

Google Cloud Platform（GCP）は、Citrix ADC VPX インスタンスへのコンソールアクセスを提供します。デバッグできるのは、ネットワークが接続されている場合だけです。インスタンスのシステムログを表示するには、コンソールにアクセスし、システムログファイルを確認します。

Citrix は、GCP 上で料金ベースの Citrix ADC VPX インスタンス（時間単位料金のユーティリティライセンス）をサポートしています。サポートケースを登録するには、GCP アカウント番号とサポート PIN コードを確認して、Citrix サポートにお問い合わせください。名前と E メールアドレスの入力を求められます。サポート PIN を検索するには、VPX GUI にログオンし、［システム］ページに移動します。

サポート PIN を示すシステムページの例を次に示します。



Citrix ADC VPX インスタンスのジャンボフレーム

October 7, 2021

Citrix ADC VPX アプライアンスは、最大 9216 バイトの IP データを含むジャンボフレームの送受信をサポートします。ジャンボフレームでは、標準の IP MTU サイズ（1500 バイト）を使用するよりも効率的に大きなファイルを送信することができます。

Citrix ADC アプライアンスは、次の展開シナリオでジャンボフレームを使用できます。

- ジャンボで受信/ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それをジャンボフレームで送信します。
- 非ジャンボで受信/ジャンボで送信。アプライアンスがデータを通常のフレームで受信し、それをジャンボフレームで送信します。
- ジャンボで受信/非ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それを通常のフレームで送信します。

詳細については、「Citrix ADC アプライアンスでのジャンボフレームサポートの構成」を参照してください。

ジャンボフレームのサポートは、次の仮想化プラットフォームで実行されている Citrix ADC VPX アプライアンスで利用できます。

- VMware ESX
- Linux-KVM プラットフォーム
- Citrix XenServer
- Amazon Web Services（AWS）

VPX アプライアンスのジャンボフレームは、MPX アプライアンスのジャンボフレームと同様に機能します。ジャンボフレームおよびそのユースケースについて詳しくは、「MPX アプライアンスでのジャンボフレームの構成」を参照してください。MPX アプライアンスでのジャンボフレームの使用例は、VPX アプライアンスにも適用されます。

VMware ESX で実行中の **VPX** インスタンスのジャンボフレームを構成する

VMware ESX サーバーで実行されている Citrix ADC VPX アプライアンスでジャンボフレームを構成するには、次のタスクを実行します。

1. VPX アプライアンスのインターフェイスまたはチャネルの MTU を 1501～9000 の範囲の値に設定します。CLI または GUI を使用して MTU サイズを設定します。VMwareESX で実行されている Citrix ADC VPX アプライアンスは、最大 9000 バイトの IP データを含むジャンボフレームの送受信をサポートします。
2. 管理アプリケーションを使用して、VMware ESX サーバーの対応する物理インターフェイスで同じ MTU サイズを設定します。VMware ESX の物理インターフェイスでの MTU サイズの設定の詳細については、http://vmware.com/を参照してください。

Linux-KVM サーバーで実行されている **VPX** インスタンスのジャンボフレームを構成する

Linux-KVM サーバーで実行されている Citrix ADC VPX アプライアンスでジャンボフレームを構成するには、次のタスクを実行します。

1. VPX アプライアンスのインターフェイスまたはチャネルの MTU を 1501～9216 の範囲の値に設定します。Citrix ADC VPX CLI または GUI を使用して、MTU サイズを設定します。
2. 管理アプリケーションを使用して、Linux-KVM サーバーの対応する物理インターフェイスで同じ MTU サイズを設定します。Linux-KVM の物理インターフェイスでの MTU サイズの設定の詳細については、http://www.linux-kvm.org/を参照してください。

Citrix XenServer で実行中の **VPX** インスタンスのジャンボフレームを構成する

次のタスクを実行して、Citrix XenServer で実行されている Citrix ADC VPX アプライアンスでジャンボフレームを構成します。

1. XenCenter を使用して XenServer に接続します。
2. MTU を変更する必要があるネットワークを使用するすべての VPX インスタンスをシャットダウンします。
3. [ネットワーク] タブで、ネットワーク-ネットワーク 0/1/2 を選択します。
4. [プロパティ] を選択し、MTU を編集します。

XenServer でジャンボフレームを構成した後、ADC アプライアンスでジャンボフレームを構成できます。詳細については、「Citrix ADC アプライアンスでのジャンボフレームサポートの構成」を参照してください。

AWS で実行中の VPX インスタンスのジャンボフレームを設定する

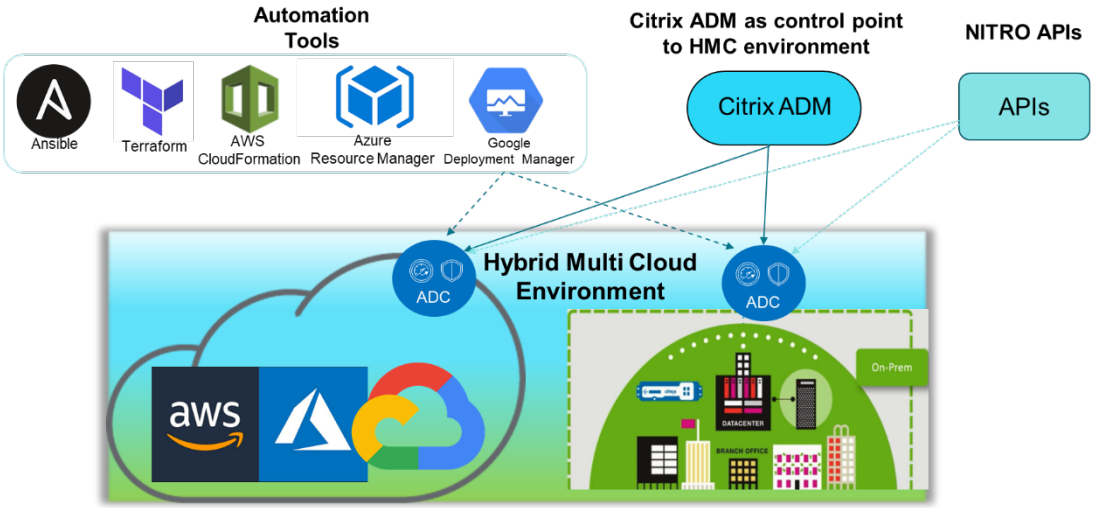
Azure 上の VPX では、ホストレベルの構成は不要です。VPX でジャンボフレームを構成するには、Citrix ADC アプライアンスでのジャンボフレームサポートの構成に記載されている手順に従います。

Citrix ADC の導入と構成を自動化する

October 7, 2021

Citrix ADC は、ADC の展開と構成を自動化するための複数のツールを提供します。このドキュメントでは、ADC 構成の管理に使用できるさまざまな自動化ツールの概要とさまざまな自動化リソースへの参照を提供します。

次の図は、ハイブリッドマルチクラウド（HMC）環境での Citrix ADC 自動化の概要を示しています。



Citrix ADM を使用して Citrix ADC を自動化する

Citrix ADM は、分散 ADC インフラストラクチャへの自動化制御ポイントとして機能します。Citrix ADM は、ADC アプライアンスのプロビジョニングからアップグレードまでの包括的な自動化機能のセットを提供します。ADM の主な自動化機能は次のとおりです。

- AWS での Citrix ADC VPX インスタンスのプロビジョニング
- Azure での Citrix ADC VPX インスタンスのプロビジョニング
- StyleBook
- 構成ジョブ
- 構成監査
- ADC のアップグレード
- SSL 証明書の管理
- 統合- GitHub、ServiceNow、イベント通知の統合

自動化に関する **Citrix ADM** ブログとビデオ

- StyleBooks を使用したアプリケーションの移行
- ADC 構成を CI/CD ADM スタイルブックの使用
- ADM によるパブリッククラウド Citrix ADC の導入の簡素化
- Citrix ADM サービスが Citrix ADC アップグレードをより簡単にサポートする 10 の方法

Citrix ADM は、IT 自動化全体の一部として Citrix ADM と Citrix ADC を統合するさまざまな機能の API も提供します。詳細については、「Citrix ADM サービス API」を参照してください。

Terraform を使用して **Citrix ADC** を自動化する

Terraform は、クラウド、インフラストラクチャ、またはサービスをプロビジョニングおよび管理するためのコードアプローチとしてインフラストラクチャを採用するツールです。Citrix ADC テラフォームリソースは、GitHub で使用できます。詳細なドキュメントと使用法については、GitHub を参照してください。

- 負荷分散や GSLB などのさまざまなユースケース向けに ADC を構成する Citrix ADC Terraform モジュール
- AWS に ADC をデプロイするための Terraform クラウドスクリプト
- Azure に ADC をデプロイするための Terraform クラウドスクリプト

ADC 自動化のための **Terraform** に関するビデオ

- Terraform を使用して Citrix ADC の導入を自動化する
- Terraform を使用して AWS の HA セットアップで ADC をプロビジョニングおよび構成します

Ansible を使用して **Citrix ADC** を自動化する

Ansible は、オープンソースのソフトウェアプロビジョニング、構成管理、およびアプリケーション展開ツールであり、インフラストラクチャをコードとして有効にします。Citrix ADC Ansible モジュールとサンプルプレイブックは、GitHub で使用できます。詳細なドキュメントと使用法については、GitHub を参照してください。

- ADC を構成するための Ansible モジュール
- Ansible で ADC を自動化-ホワイトペーパー
- ADM 用の Ansible モジュール

Citrix は認定された AnsibleAutomation パートナーです。Red Hat Ansible オートメーションプラットフォームのサブスクリプションをお持ちのユーザーは、Red Hat オートメーションハブから Citrix ADCコレクションにアクセスできます。

Terraform と **Ansible** の自動化ブログ

- アプリの配信とセキュリティのための Terraform と Ansible の自動化

ADC 展開用のパブリッククラウドテンプレート

パブリッククラウドテンプレートは、パブリッククラウドでのデプロイメントのプロビジョニングを簡素化します。さまざまな環境で、さまざまな Citrix ADC テンプレートを使用できます。使用法の詳細については、それぞれの GitHub リポジトリを参照してください。

AWS CFT:

- AWS で Citrix ADC VPX をプロビジョニングするための CFT

Azure Resource Manager（**ARM**）テンプレート:

- Azure で Citrix ADC VPX をプロビジョニングするための ARM テンプレート

Google Cloud Deployment Manager（**GDM**）テンプレート:

- Google で Citrix ADC VPX をプロビジョニングするための GDM テンプレート

テンプレートのビデオ

- CloudFormation テンプレートを使用して AWS に Citrix ADC HA をデプロイする
- AWS クイックスタートを使用して、アベイラビリティーゾーン間で Citrix ADC HA をデプロイする
- GDM テンプレートを使用した GCP での Citrix ADC HA の導入

AWS クイックスタート

- CitrixWAF クイックスタート
- AWS 上の Web アプリケーション向け Citrix ADC VPX の AWS クイックスタート

NITRO API

Citrix ADC NITRO プロトコルを使用すると、Representational State Transfer（REST）インターフェイスを使用して、Citrix ADC アプライアンスをプログラムで構成および監視できます。そのため、NITRO アプリケーションはあらゆるプログラミング言語で開発することができます。Java、.NET、または Python で開発する必要のあるアプリケーションの場合、NITRO API は、個別のソフトウェア開発キット（SDK）としてパッケージ化された関連ライブラリを通じて公開されます。

- NITRO API ドキュメント
- Citrix ADC API リファレンス
- NITRO API を使用した ADC ユースケース構成の例

よくある質問

April 7, 2022

次のセクションでは、Citrix アプリケーション Delivery Controller（ADC）VPX に基づいて FAQ を分類するのに役立ちます。

- 機能と機能
- 暗号化
- 価格設定と梱包
- Citrix ADC VPX Express
- ハイパーバイザー
- キャパシティプランニングまたはサイジング
- システム要件
- その他の技術的なよくある質問

機能と機能

Citrix ADC VPX とは何ですか**?**

Citrix ADC VPX は、業界標準のサーバーにインストールされた Hypervisor でホストできる仮想 ADC アプライアンスです。

Citrix ADC VPX には、すべての **Web** アプリケーション最適化機能が **ADC** アプライアンスとして含まれていますか

はい。Citrix ADC VPX には、すべての負荷分散、トラフィック管理、アプリケーションアクセラレーション、アプリケーションセキュリティ（Citrix ADC Gateway および Citrix アプリケーションファイアウォールを含む）、およびオフロード機能が含まれています。Citrix ADC の機能と機能の完全な概要については、「アプリケーションの配信方法」を参照してください。

Citrix アプリケーションファイアウォールを **Citrix ADC VPX** で使用する場合、制限はありますか**?**

Citrix ADC VPX 上の Citrix アプリケーションファイアウォールは、Citrix ADC アプライアンスと同じセキュリティ保護を提供します。Citrix アプリケーションファイアウォールのパフォーマンスまたはスループットは、プラットフォームによって異なります。

Citrix ADC VPX 上の **Citrix ADC Gateway** と **Citrix ADC** アプライアンスの **Citrix ADC Gateway** の間に違いはありますか**?**

機能的には、それらは同じです。Citrix ADC VPX 上の Citrix ADC Gateway は、Citrix ADC ソフトウェアリリース 9.1 で利用可能なすべての Citrix ADC Gateway 機能をサポートします。ただし、Citrix ADC アプライアンスは専用

の SSL アクセラレーションハードウェアを提供するため、Citrix ADC VPX インスタンスよりも優れた SSL VPN スケーラビリティを提供します。

Hypervisor 上で実行できるという明らかな違い以外に、**Citrix ADC VPX** は **Citrix ADC** 物理アプライアンスとどのように違いますか

顧客に行動の違いが見られる主な領域は 2 つあります。1 つ目は、Citrix ADC VPX は多くの Citrix ADC アプライアンスと同じパフォーマンスを提供できないことです。2 つ目は、Citrix ADC アプライアンスは独自の L2 ネットワーク機能を組み込んでいるが、Citrix ADC VPX は L2 ネットワークサービスのために Hypervisor に依存しているということです。一般的に、Citrix ADC VPX の展開方法は制限されません。物理 Citrix ADC アプライアンスに構成されている特定の L2 機能は、基盤となる Hypervisor で構成する必要があります。

Citrix ADC VPX は、アプリケーションデリバリー市場でどのように役割を果たしていますか**?**

Citrix ADC VPX は、アプリケーション配信市場のゲームを次のように変えます。

- Citrix ADC アプライアンスをさらに手頃な価格にすることで、Citrix ADC VPX は、あらゆる IT 組織が Citrix ADC アプライアンスを展開できるようにします。これは、最もミッションクリティカルな Web アプリケーションだけでなく、すべての Web アプリケーション用です。
- Citrix ADC VPX を使用すると、データセンター内でネットワーキングと仮想化をさらに統合できます。Citrix ADC VPX は、仮想サーバーでホストされている Web アプリケーションを最適化するためだけに使用することはできません。また、Web アプリケーションの配信自体を、どこでも簡単かつ迅速に展開できる仮想化サービスにすることができます。IT 組織は、Web アプリケーション配信インフラストラクチャのプロビジョニング、自動化、チャージバックなどのタスクに標準的なデータセンタープロセスを使用します。
- Citrix ADC VPX は、物理アプライアンスだけを使用する場合は実用的ではない新しい展開アーキテクチャを開きます。Citrix ADC VPX および Citrix ADC MPX アプライアンスは、圧縮やアプリケーションファイアウォール検査などのプロセッサ負荷の高いアクションを処理するために、各アプリケーションの個々のニーズに合わせてベースで使用できます。データセンターエッジでは、Citrix ADC MPX アプライアンスは、初期トラフィック分散、SSL 暗号化または復号化、サービス拒否（DoS）攻撃防止、グローバル負荷分散など、大量のネットワーク全体のタスクを処理します。高性能の Citrix ADC MPX アプライアンスと展開しやすい Citrix ADC VPX 仮想アプライアンスを組み合わせることで、データセンター全体のコストを削減しながら、最新の大規模データセンター環境に比類のない柔軟性とカスタマイズ機能を提供します。

Citrix ADC VPX はシトリック **Citrix** デリバリーセンター戦略にどのように適合していますか

Citrix ADC VPX を利用することで、Citrix デリバリーセンターの全サービスを仮想化されたサービスとして利用できます。Citrix XenCenter で利用可能な強力な管理、プロビジョニング、監視、およびレポート機能によって、Citrix デリバリーセンター全体がメリットを得られます。これは、ほぼすべての環境に迅速に導入でき、どこからでも一元的に管理できます。1 つの統合された仮想化アプリケーション配信インフラストラクチャにより、組織はデスクトップ、クライアント/サーバーアプリケーション、Web アプリケーションを配信できます。

暗号化

Citrix ADC VPX は **SSL** オフロードをサポートしていますか**?**

はい。ただし、Citrix ADC VPX はすべての SSL 処理をソフトウェアで行うため、Citrix ADC VPX は Citrix ADC アプライアンスと同じ SSL パフォーマンスを提供しません。Citrix ADC VPX は、毎秒最大 750 の新しい SSL トランザクションをサポートできます。

Citrix ADC VPX をホストするサーバーにインストールされているサードパーティの **SSL** カードは、**SSL** 暗号化または復号化を高速化しますか**?**

なしサードパーティの SSL カードをサポートしていると、Citrix ADC VPX を特定のハードウェア実装に関連付けることはできません。これにより、データセンター内の任意の場所で Citrix ADC VPX を柔軟にホストする組織の能力が大幅に低下します。Citrix ADC MPX アプライアンスは、Citrix ADC VPX が提供するよりも高い SSL スループットが必要な場合に使用する必要があります。

Citrix ADC VPX は、物理的な **Citrix ADC** アプライアンスと同じ暗号化暗号をサポートしていますか**?**

VPX は、ECDSA を除くすべての暗号化暗号を物理 Citrix ADC アプライアンスとしてサポートします。

Citrix ADC VPX SSL トランザクションスループットとは何ですか**?**

SSL トランザクションのスループットについては、Citrix ADC VPX のデータシートを参照してください 。

価格設定と梱包

Citrix ADC VPX はどのようにパッケージ化されていますか

Citrix ADC VPX の選択は、Citrix ADC アプライアンスの選択に似ています。まず、お客様は、機能要件に基づいて Citrix ADC エディションを選択します。次に、スループット要件に基づいて、特定の Citrix ADC VPX 帯域幅層を選択します。Citrix ADC VPX は、スタンダード、アドバンスエディション、およびプレミアムエディションで利用できます。Citrix ADC VPX は、10Mbps（VPX 10）から 100Gbps（VPX 100G）まで対応している。詳細については、Citrix ADC VPX のデータシートを参照してください。

Citrix ADC VPX の価格はすべての **Hypervisor** で同じですか**?**

はい。

すべての **Hypervisor** で **VPX** に同じ **Citrix ADC SKU** が使用されていますか**?**

はい。

Citrix ADC VPX ライセンスをある **Hypervisor** から別の **Hypervisor** に移動できますか（たとえば、**VMware** から **Hyper-V** へ）**?**

はい。Citrix ADC VPX ライセンスは、基盤となる Hypervisor から独立しています。Citrix ADC VPX 仮想マシンをある Hypervisor から別の Hypervisor に移動する場合は、新しいライセンスを取得する必要はありません。ただし、既存の Citrix ADC VPX ライセンスを再ホストする必要がある場合があります。

Citrix ADC VPX インスタンスはアップグレードできますか**?**

はい。スループット制限と Citrix ADC ファミリエディションの両方をアップグレードできます。両方のタイプのアップグレードのアップグレード SKU が利用可能です。

Citrix ADC VPX を高可用性ペアに展開する場合、必要なライセンスはいくつですか

Citrix ADC 物理アプライアンスと同様に、Citrix ADC 高可用性構成には 2 つのアクティブなインスタンスが必要です。したがって、お客様は 2 つのライセンスを購入する必要があります。

Citrix ADC VPX Express および 90 日間の無料トライアル

Citrix ADC VPX Express には、**Citrix ADC** 標準機能がすべて含まれていますか? **Citrix ADC Gateway**、および **Citrix Virtual Apps**（旧 **XenApp**）**Web** インターフェイスおよび **XML** ブローカーの負荷分散は含まれていますか

はい。Citrix ADC VPX Express には、Citrix ADC 標準のすべての機能が含まれています。Citrix ADC リリース 12.0～56.20 以降、Citrix は VPX Express の動作を変更しました。

Citrix ADC VPX Express には、**Citrix ADC** 標準機能がすべて含まれていますか? **Citrix ADC Gateway** と、**Citrix Virtual Apps Web** インターフェイスと **XML** ブローカーの負荷分散が含まれていますか

Citrix ADC リリース 12.0～56.20 以降、VPX Express は、ゲートウェイ機能を除く Citrix ADC スタンダードエディション機能セットを提供します。12.0—56.20 以前のリリースでは、VPX Express には標準エディションのすべての機能が含まれています。

Citrix ADC VPX Express にはライセンスが必要ですか

新しい Citrix ADC VPX Express リリース（12.0～56.20 以降）では、VPX Express は無料で、インストールにライセンスファイルを必要とせず、コミットメントもありません。すでに VPX Express ライセンスをお持ちの場合、以前の VPX Express の動作は保持されます。VPX *Express* ライセンスファイルが削除され、12.0—56.20 以降のリリースが使用されている場合、新しい VPX Express の動作が有効になります。

Citrix ADC VPX Express ライセンスは期限切れになりますか

新しい VPX Express では、いいえ。ライセンスと有効期限はありません。VPX Express ライセンスをすでにお持ちの場合、ライセンスはダウンロード後 1 年で期限切れになります。

Citrix ADC VPX Express には、**5** つの無料の **Citrix ADC** ゲートウェイ同時実行ライセンスが含まれていますか

はい、VPX Express ライセンスを所有している場合は可能です。

顧客がダウンロードできる **Citrix ADC VPX Express** の数に制限はありますか

ファイブ。

Citrix ADC VPX Express は、**Citrix ADC MPX** アプライアンスと同じ暗号化暗号をサポートしていますか

一般的な可用性のために、Citrix ADC アプライアンスでサポートされている同じ強力な暗号化暗号はすべて、Citrix ADC VPX および Citrix ADC VPX Express で利用できます。これは、同じ輸出入規制の対象となります。

Citrix ADC VPX Express のテクニカルサポートケースを報告できますか

なしテクニカルサポートケースを提出するには、VPX-10、VPX-200、VPX-1000、VPX-3000 などの小売 Citrix ADC VPX ライセンスが必要です。ただし、Citrix ADC VPX Express ユーザーは、Citrix ADC VPX ナレッジセンターの両方を自由に使用でき、Z ディスカッションフォーラムを使用してコミュニティにヘルプをリクエストできます。

Citrix ADC VPX Express を製品版にアップグレードできますか**?**

はい。必要な小売 Citrix ADC VPX ライセンスを購入し、対応するライセンスを Citrix ADC VPX Express インスタンスに適用するだけです。

ハイパーバイザー

Citrix ADC VPX はどの **VMware** バージョンをサポートしていますか

Citrix ADC VPX は、バージョン 3.5 以降では、VMware ESX と ESXi の両方をサポートしています。詳細については、「サポートマトリックスと使用上のガイドライン」を参照してください。

VMware の場合、**VPX** に割り当てることができる仮想ネットワーク・インタフェースはいくつですか**?**

最大 10 個の仮想ネットワークインターフェイスを Citrix ADC VPX に割り当てることができます。

vSphere から、**Citrix ADC VPX** コマンドラインにどのようにアクセスできますか

VMware vSphere クライアントは、コンソールタブから Citrix ADC VPX コマンドラインへの組み込みアクセスを提供します。また、任意の SSH または Telnet クライアントを使用してコマンドラインにアクセスすることもできます。Citrix ADC VPX の NSIP アドレスは、SSH または Telnet クライアントで使用できます。

Citrix ADC VPX GUI にはどのようにアクセスできますか

Citrix ADC VPX GUI にアクセスするには、任意のブラウザーのアドレスフィールドに、Citrix ADC VPX の NSIP（たとえば、`http://NSIP address`）を入力します。

同じ **VMware ESX** にインストールされている **2** つの **Citrix ADC VPX** インスタンスを高可用性セットアップで構成できますか**?**

はい、でもお勧めできません。ハードウェア障害は、両方の Citrix ADC VPX インスタンスに影響します。

2 つの異なる **VMware ESX** システム上で実行されている **2** つの **Citrix ADC VPX** インスタンスを、高可用性セットアップで構成できますか**?**

はい。これは、高可用性セットアップで推奨されます。

VMware の場合、インターフェイス関連のイベントは **Citrix ADC VPX** でサポートされていますか

なしインターフェイス関連のイベントはサポートされていません。

VMware の場合、タグ付き **VLAN** は **Citrix ADC VPX** でサポートされていますか**?**

はい。Citrix ADC タグ付き VLAN は、リリース 11.0 以降の Citrix ADC VPX でサポートされています。詳細については、Citrix のドキュメントを参照してください。

VMware の場合、リンクアグリゲーションと **LACP** は **Citrix ADC VPX** でサポートされていますか**?**

なしリンクアグリゲーションと LACP は、Citrix ADC VPX ではサポートされていません。リンクアグリゲーションは VMware レベルで設定する必要があります。

Citrix ADC VPX ドキュメントにはどのようにアクセスするのですか

このドキュメントは、Citrix ADC VPX GUI から入手できます。ログイン後、[ドキュメント] タブを選択します。

キャパシティプランニングまたはサイジング

Citrix ADC VPX で期待できるパフォーマンスは何ですか

Citrix ADC VPX は、優れたパフォーマンスを提供します。Citrix ADC VPX を使用して達成可能な特定のパフォーマンスレベルについては、Citrix ADC VPX のデータシートを参照してください 。

サーバーの **CPU** パワーが変化することを考えると、**Citrix ADC** インスタンスの最大パフォーマンスをどのように見積もることができますか**?**

より高速な CPU を使用すると（ライセンスで許可されている最大値まで）パフォーマンスが向上しますが、低速の CPU を使用すると、パフォーマンスが確実に制限されます。

Citrix ADC VPX の帯域幅またはスループットの制限は、インバウンドのみのトラフィック、またはインバウンドとアウトバウンドの両方のトラフィックですか

Citrix ADC VPX 帯域幅制限は、要求トラフィックか応答トラフィックかにかかわらず、Citrix ADC への着信トラフィックにのみ適用されます。これは、Citrix ADC VPX-1000（たとえば）が 1 Gbps のインバウンドトラフィックと 1 Gbps のアウトバウンドトラフィックの両方を同時に処理できることを示します。インバウンドおよびアウトバウンドトラフィックは、要求および応答トラフィックと同じではありません。Citrix ADC では、エンドポイントからのトラフィック（リクエストトラフィック）とオリジンサーバーからのトラフィック（レスポンストラフィック）の両方が「インバウンド」（つまり、Citrix ADC に着信）です。

同じサーバー上で **Citrix ADC VPX** 複数のインスタンスを実行できますか**?**

はい。ただし、物理サーバーにホストで実行されている合計ワークロードをサポートするのに十分な CPU と I/O 容量があることを確認してください。そうしなければ Citrix ADC VPX のパフォーマンスに影響する可能性があります。

Citrix ADC VPX の複数のインスタンスが物理サーバーで実行されている場合、**Citrix ADC VPX** インスタンスごとの最小ハードウェア要件は何ですか**?**

各 Citrix ADC VPX インスタンスには、2 GB の物理 RAM、20 GB のハードディスク容量、および 2 つの vCPU を割り当てる必要があります。

Citrix ADC VPX と他のアプリケーションを同じサーバーでホストできますか**?**

はい。たとえば、Citrix ADC VPX、Citrix Virtual Apps Web インターフェイス、および Citrix Virtual Apps XML ブローカーはすべて仮想化でき、同じサーバー上で実行できます。最高のパフォーマンスを得るには、実行中のすべてのワークロードをサポートするのに十分な CPU および I/O 容量が物理ホストにあることを確認します。

単一の **Citrix ADC VPX** インスタンスに **CPU** コアを追加すると、そのインスタンスのパフォーマンスが向上しますか**?**

ライセンスに応じて、Citrix ADC VPX インスタンスは現在、最大 4 つの vCPU を使用できます。より多くの CPU を使用できる Citrix ADC VPX インスタンスに CPU を追加すると、パフォーマンスが向上します。

Citrix ADC VPX がアイドル状態のにもかかわらず、**CPU** の **90**%以上を消費しているように見えるのはなぜですか**?**

これは正常な動作であり、Citrix ADC アプライアンスは同じ動作を示します。Citrix ADC VPX CPU 使用率の実程度を確認するには、Citrix ADC CLI で stat CPU コマンドを使用するか、Citrix ADC GUI から Citrix ADC VPX CPU 使用率を表示します。Citrix ADC パケット処理エンジンは、やるべき作業がない場合でも、常に「仕事を探している」ことです。したがって、CPU を制御し、それを解放しないためにすべてを行います。Citrix ADC VPX がインストールされたサーバーでは、Citrix ADC VPX が CPU 全体を消費しているような外観になります（Hypervisor の観点から）。CLI または GUI を使用した「Citrix ADC 内部」からの CPU 使用率を見ると、Citrix ADC VPX CPU 容量が使用されている様子が表示されます。

システム要件

Citrix ADC VPX 最小ハードウェア要件を教えてください**?**

システム要件については、Citrix ADC VPX のデータシートを参照してください 。

Citrix ADC VPX には以下が必要です。

- プロセッサ要件:Intel Xeon 搭載デュアルコアサーバ。
- 使用可能なメモリ： 4 GB の RAM と 20 GB のハードドライブ。重要な展開では、システムが非常にメモリに制約のある環境で動作するため、VPX に 2 GB RAM を使用することは推奨しません。これにより、スケール、パフォーマンス、または安定性に関連する問題が発生する可能性があります。
- Hypervisor： Citrix Hypervisor 5.6 以降。VMware ESX/ESXi 3.5 以降、Hyper-V を搭載した Windows Server 2008 R2。
- 接続性: 最低 100 Mbps。1 Gbps を推奨。
- Hypervisor と互換性のある NIC。

> 注
>
> AMD プロセッサーはサポートされていません。

Intel VT-X って何ですか

これらの機能は、「ハードウェアアシスト」または「仮想化アシスト」とも呼ばれ、ゲスト OS によって実行される機密または特権 CPU 命令をハイパーバイザにトラップします。これにより、Hypervisor でのゲスト OS（Citrix ADC VPX 用の BSD）のホスティングが簡単になります。

VT-X はどれくらい一般的ですか

事実上、過去 2 年以内に出荷されたすべてのサーバが VT-X をサポートしている可能性があります。多くのサーバには、BIOS で仮想化支援が無効になっている状態で出荷されます。Citrix ADC VPX を実行できないと仮定する前に、サーバーでこの設定を変更する必要があるかどうかを確認してください。

Citrix ADC VPX ハードウェア互換性リスト（**HCL**）はありますか**?**

サーバーが Intel VT-X をサポートしている限り、Citrix ADC VPX は、基盤となる Hypervisor と互換性のあるサーバーで実行する必要があります。サポートされるプラットフォームの包括的なリストについては、Hypervisor HCL を参照してください。

Citrix ADC VPX はどのバージョンの **Citrix ADC OS** をベースにしていますか

Citrix ADC VPX は、Citrix ADC 9.1 以降のリリースをベースにしています。

Citrix ADC VPX は **BSD** 上で動作するので、**BSD Unix** がインストールされているサーバーでネイティブに実行できますか

いいえ。Citrix ADC VPX を実行するには、Hypervisor が必要です。Hypervisor のサポートの詳細については、Citrix ADC VPX のデータシートを参照してください。

その他の技術的なよくある質問

複数の **NIC** を持つ物理サーバでのリンクアグリゲーションは機能しますか**?**

LACP はサポートされていません。Citrix Hypervisor では、静的リンクアグリゲーションがサポートされ、4 つのチャネルと 7 つの仮想インターフェイスの制限があります。VMware の場合、静的リンクアグリゲーションは Citrix ADC VPX 内ではサポートされていませんが、VMware レベルで構成できます。

MAC ベースの転送 **(MBF)** は **VPX** でサポートされていますか**? Citrix ADC** アプライアンスの実装から変更はありますか**?**

MBF はサポートされており、Citrix ADC アプライアンスと同じように動作します。Hypervisor は基本的に、Citrix ADC VPX から受信したすべてのパケットを外部に切り替え、逆に切り替えます。

Citrix ADC VPX のアップグレードプロセスはどのように実行されますか**?**

アップグレードは、Citrix ADC アプライアンスの場合と同じ方法で実行されます。カーネルファイルをダウンロードし、GUI で install ns またはアップグレードユーティリティを使用します。

VPX にデフォルトイメージを使用する場合の**/var** パーティションのサイズはどれくらいですか? ディスク容量を増やすには

ディスクイメージを小さく保つために、ルートディスクのサイズは 20 GB に制限されています。

/var/core/または/var/crash/ディレクトリ領域を増やす場合は、追加のディスクを接続します。/varサイズを大きくするには、現在のところ、重要なコンテンツを新しいディスクにコピーした後、追加のディスクを接続して/varにシンボリックリンクを作成する必要があります。

NetScaler VPX ビルドの番号付けと、他のビルドとの相互運用性に関して、どのようなことが期待できますか**?**

Citrix ADC VPX には、9.1 と同様のビルド番号が付けられています。Cl (クラシック) と 9.1. Nc (nCore) リリース。例えば 9.1_97.3.vpx、9.1_97.3.nc、9.1_97.3.cl。

Citrix ADC VPX を **Citrix ADC** アプライアンスを使用した高可用性セットアップの一部にすることはできますか**?**

サポートされていない構成です。

Citrix ADC VPX に表示されるすべてのインターフェイスは、**Hypervisor** 上のインターフェイスの数に直接関係していますか

いいえ。Hypervisor 上に物理 NIC が 1 つしかない Citrix ADC VPX 構成ユーティリティを使用して、最大 7 つのインターフェイス（VMware では 10）を追加できます。

Citrix Hypervisor XenMotion または **VMware VMotion** または **Hyper-V** ライブマイグレーションを使用して、**Citrix ADC VPX** アクティブなインスタンスを移動できますか

Citrix ADC VPX は、XenMotion または Hyper-V ライブマイグレーションをサポートしていません。vMotion は、Citrix ADC 12.1 リリース以降でサポートされています。詳細については、「リリースノート」を参照してください。

ライセンスサーバーの概要

October 7, 2021

Citrix は、組織のニーズを満たすために、MPX および VPX アプライアンス向けの幅広い製品エディションとライセンスモデルを提供しています。

Citrix ADC アプライアンスを適切に動作させるには、Citrix ADC ファミリエディションライセンスの 1 つが必要です。ADC 製品ラインには、次の 3 つのファミリ・エディションがあります。

- Standard Edition

- Advanced Edition
- Premium Edition

詳細については、Citrix ADC のデータシートを参照してください。

Citrix ADC エディションを選択した後、MPX および VPX ライセンス製品のいずれかを選択できます。永続およびサブスクリプション（年間および時間単位のサブスクリプション）、vCPU と帯域幅、オンプレミスとクラウドなどの基準に基づきます。

Citrix ADC VPX ライセンス

以下は、VPX 固有のライセンスです。

Citrix ADC VPX Express ライセンス

Citrix ADC リリース 12.0 56.20 以降、オンプレミスおよびクラウド展開用の VPX Express はライセンスファイルを必要とせず、次の機能を備えています。

- 20Mbps の帯域幅
- Citrix Gateway および L4 および L7 防御を除く、すべての ADC 標準ライセンス機能
- 最大 250 の SSL セッション
- 20Mbps の SSL スループット

VPX Express ライセンスを次の 2 つのオプションにアップグレードできます。

1. スタンドアロンの Citrix ADC VPX ライセンス
2. VPX インスタンスの Citrix ADC プール容量ライセンス。詳細については、「Citrix ADC プール容量」を参照してください。

> 重要
>
> クラスタリングは、VPX パブリッククラウドのスタンダードエディションと VPX Express ライセンスで利用できます。

Citrix ADC VPX プール容量ライセンス

Citrix Application Delivery Management（ADM）を使用して、共通の帯域幅とインスタンスプールで構成されるライセンスフレームワークを作成できます。詳細については、「Citrix ADC プール容量」を参照してください。

関連リソース

The Citrix Licensing System

Citrix ADC VPX ライセンスを割り当てる方法

クラウドでの **VPX** ライセンス

VPX デプロイメントは、Azure、AWS、Google などのパブリッククラウドプロバイダーでサポートされています。詳しくは、次のドキュメントを参照してください。

- VPX-Azure ライセンス
- VPX-AWS ライセンス
- VPX-GCP ライセンス

ライセンスの割り当てと適用

October 7, 2021

CitrixMPX および VPXADC GUI では、ハードウェアシリアル番号（HSN）またはライセンスアクセスコードを使用して、ライセンスを割り当てることができます。または、ローカルコンピューターにライセンスが既に存在する場合は、それをアプライアンスにアップロードできます。

ライセンスの返却や再割り当てなど、他のすべての機能については、ライセンスポータルを使用する必要があります。オプションで、ライセンスの割り当てにライセンスポータルを引き続き使用できます。詳細については、「citrix.com の [My Account] でライセンスの管理を使用する」を参照してください。

Citrix ライセンスガイド

Citrix ライセンスガイドでは、Citrix ADC アプライアンスへのライセンスのインストールおよび他の Citrix 製品へのライセンスのインストールについても説明しています。詳細については、「Citrix ライセンスガイド」を参照してください。

前提条件

> 注:
>
> 高可用性ペアのアプライアンスごとに個別のライセンスを購入します。同じタイプのライセンスが両方のアプライアンスにインストールされていることを確認してください。たとえば、あるアプライアンスのプレミアムライセンスを購入する場合、もう一方のアプライアンス用に別のプレミアムライセンスを購入する必要があります。

ハードウェアシリアル番号またはライセンスアクセスコードを使用してライセンスを割り当てるには、次の手順を実行します。

- アプライアンスを介してパブリックドメインにアクセスできる必要があります。たとえば、アプライアンスは www.citrix.com にアクセスできる必要があります。ライセンス割り当てソフトウェアは、ライセンスの Citrix ライセンスポータルに内部的にアクセスします。パブリックドメインにアクセスするには:

  - プロキシサーバーを使用するか、DNS サーバーをセットアップします。
  - Citrix ADC アプライアンスで Citrix ADC IP（NSIP）アドレスまたはサブネット IP（SNIP）アドレスを設定します。

- ライセンスがハードウェアにリンクされているか、有効なライセンスアクセスコードが必要です。ライセンスを購入すると、Citrix からライセンスアクセスコードが電子メールで送信されます。

GUI を使用したライセンスの割り当て

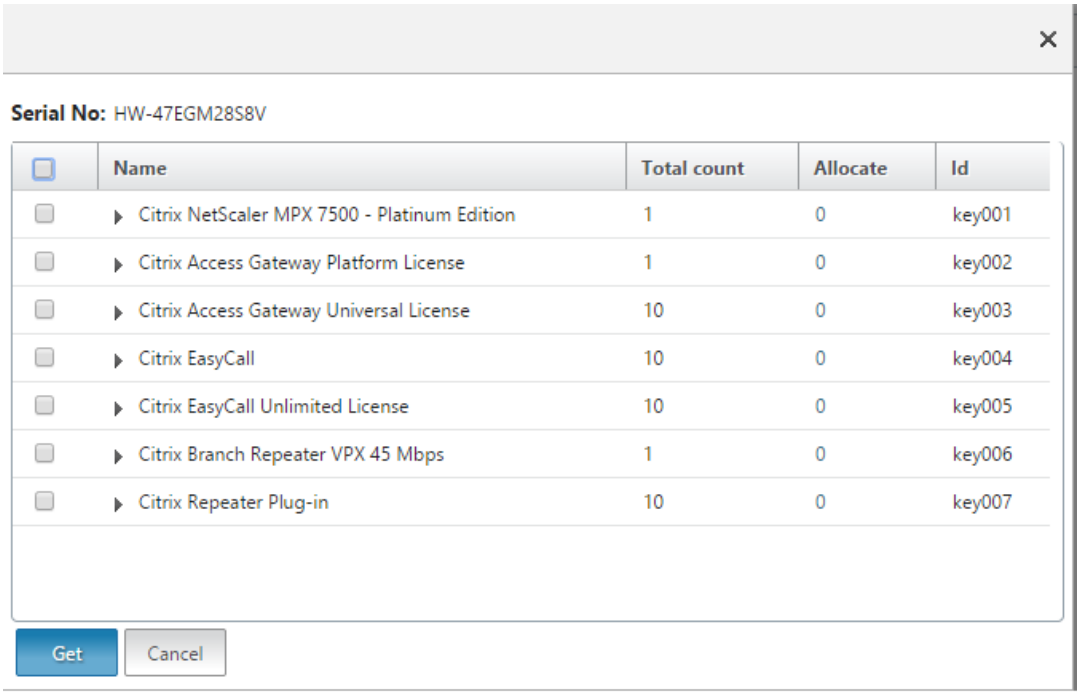
ライセンスがすでにハードウェアにリンクされている場合は、ライセンス割り当てプロセスでハードウェアシリアル番号を使用できます。それ以外の場合は、ライセンスアクセスコードを入力する必要があります。

展開の必要に応じて、ライセンスを部分的に割り当てることができます。たとえば、ライセンスファイルに 10 のライセンスが含まれていても、現在の要件が 6 つのライセンスのみである場合、ここで 6 つのライセンスを割り当てて、後でさらにライセンスを割り当てることができます。ライセンスファイルに存在するライセンスの合計数を超えるライセンスを割り当てることはできません。

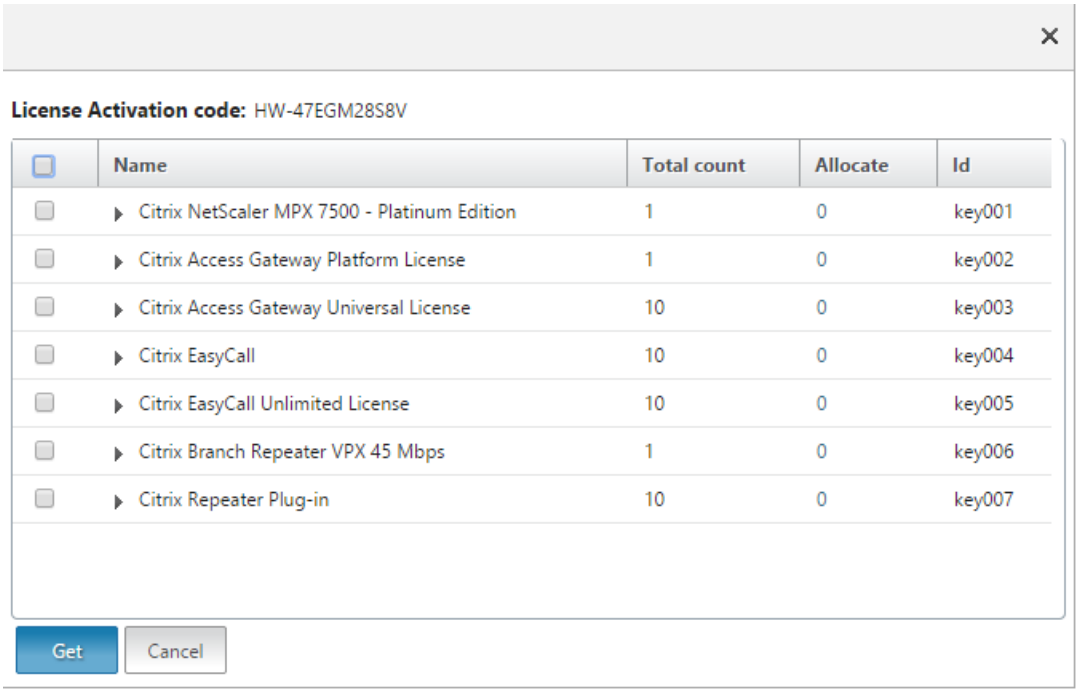
ライセンスを割り当てるには

1. Web ブラウザーに Citrix ADC アプライアンスの IP アドレスを入力します(例: `http://192.168.100.1`)。
2. [ユーザー名] と [パスワード] に、管理者の資格情報を入力します。
3. [**Configuration**] タブで、[**System**] **>** [**Licenses**] の順に移動します。
4. 詳細ウィンドウで、[ライセンスの管理] をクリックし、[新しいライセンスの追加] をクリックして、次のいずれかのオプションを選択します。
   - シリアル番号を使用する: ソフトウェアは内部でアプライアンスのシリアル番号を取得し、この番号を使用してライセンスを表示します。
   - ライセンスアクセスコードを使用する: Citrix は、購入したライセンスのライセンスアクセスコードを電子メールで送信します。テキストボックスにライセンスアクセスコードを入力します。

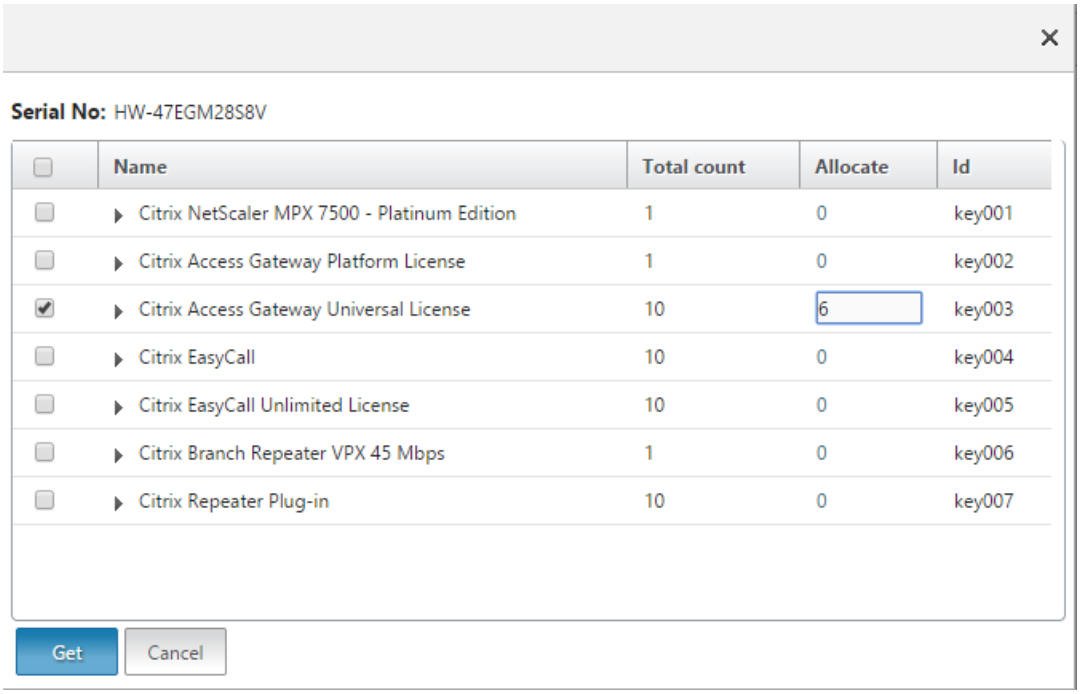
     Citrix ADC アプライアンスでインターネット接続を構成しない場合は、プロキシサーバーを使用できます。[**Connect through Proxy Server**] チェックボックスを選択し、プロキシサーバーの IP アドレスとポートを指定します。
5. [**Get Licenses**] をクリックします。選択したオプションに応じて、次のいずれかのダイアログボックスが表示されます。
   - [ハードウェアシリアル番号] を選択した場合は、次のダイアログボックスが表示されます。



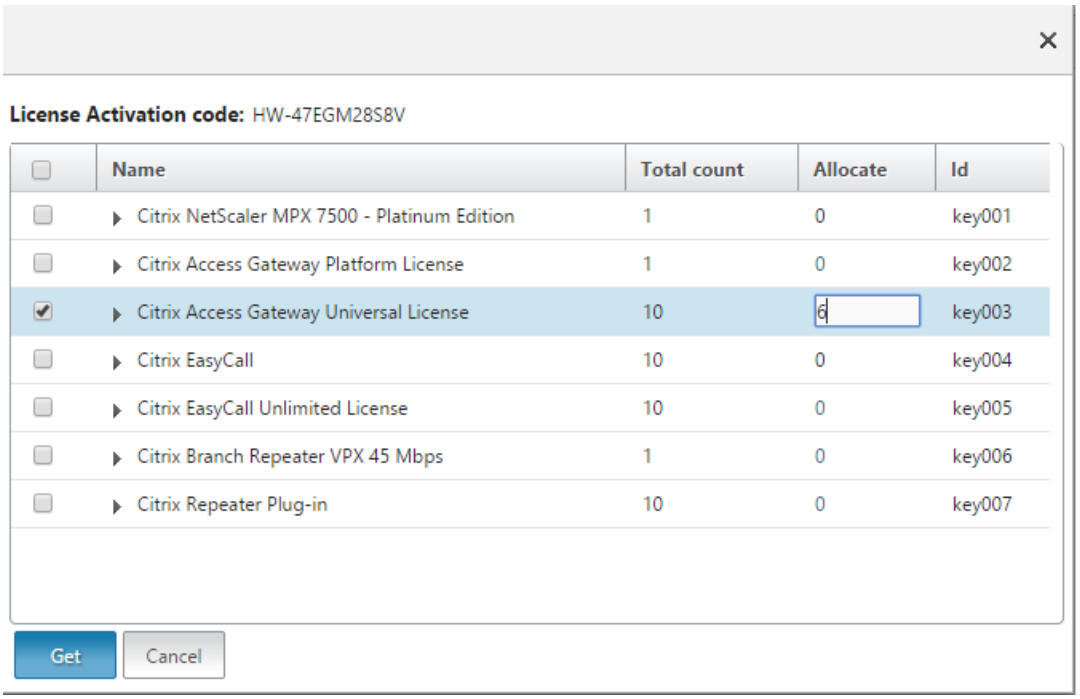
- ライセンスアクセスコードを選択すると、次のダイアログボックスが表示されます。



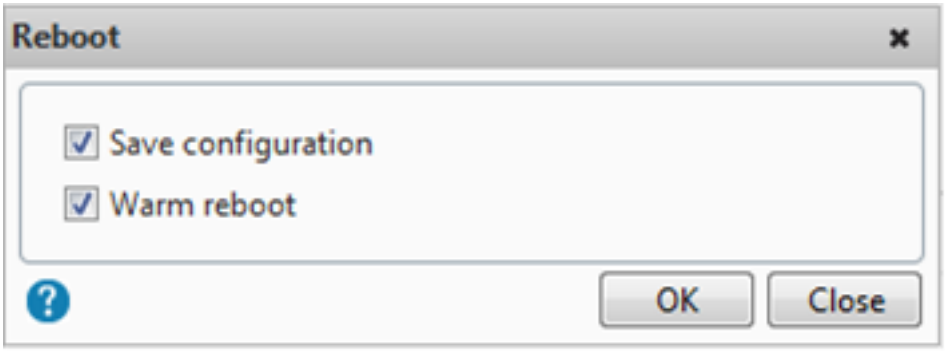
6. ライセンスの割り当てに使用するライセンスファイルを選択します。
7. [**Allocate**] 列に、割り当てるライセンスの数を入力します。次に、[取得] をクリックします。
   - [ハードウェアシリアル番号] を選択した場合は、次のスクリーンショットに示すように、ライセンスの数を入力します。



- ライセンスアクセスコードを選択した場合は、次のスクリーンショットに示すように、ライセンスの数を入力します。



8. ライセンスを有効にするには、[restart] をクリックします。
9. [再起動] ダイアログボックスで、[**OK**] をクリックして変更を続行するか、[閉じる] をクリックして変更をキャンセルします。



ライセンスをインストールする

ライセンスポータルにアクセスしてライセンスファイルをローカルコンピュータにダウンロードした場合は、アプライアンスにライセンスをアップロードする必要があります。

GUI を使用してライセンスファイルをインストールするには

1. Web ブラウザーに Citrix ADC アプライアンスの IP アドレスを入力します(例:`http://192.168.100.1`)。
2. [User Name] ボックスと [Password] ボックスに管理者資格情報を入力します。
3. [構成] タブで、[システムライセンス] に移動します。
4. 詳細ペインで、[**Manage Licenses**] をクリックします。
5. [**Add New License**] をクリックし、[**Upload license files from a local computer**] を選択します。
6. [**Browse**] をクリックします。ライセンスファイルの場所に移動し、ライセンスファイルを選択して、[**Open**] をクリックします。
7. [restart] をクリックしてライセンスを適用します。
8. [再起動] ダイアログボックスで、[**OK**] をクリックして変更を続行するか、[閉じる] をクリックして変更をキャンセルします。

CLI を使用してライセンスをインストールするには

1. PuTTY などの **SSH** クライアントを使用して、**ADC** アプライアンスへの **SSH** 接続を開きます。
2. 管理者の資格情報を使用して ADC アプライアンスにログオンします。
3. シェルプロンプトに切り替え、`nsconfig` ディレクトリにライセンスサブディレクトリが存在しない場合は作成し、1 つ以上の新しいライセンスファイルをこのディレクトリにコピーします。

   例

```
1 login: nsroot
2 Password: nsroot
```



```
3 Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
4 Done
5 > shell
6 Last login: Mon Aug  4 03:51:42 from 10.103.25.64
7 root@ns# mkdir /nsconfig/license
8 root@ns# cd /nsconfig/license
9 <!--NeedCopy-->
```

1 つまたは複数の新しいライセンスファイルをこのディレクトリにコピーします。

注：コマンドラインインターフェイスを使用してライセンスをインストールする場合、Citrix ADC アプライアンスは再起動オプションの入力を求めません。restart -w コマンドを実行してシステムをウォームリスタートするか、restart コマンドを実行してシステムを正常に再起動します。

ライセンスされた機能の確認

機能を使用する前に、ライセンスがその機能をサポートしていることを確認してください。

CLI を使用してライセンスされた機能を検証するには

1. PuTTY などの **SSH** クライアントを使用して、**ADC** アプライアンスへの **SSH** 接続を開きます。
2. 管理者の資格情報を使用して ADC アプライアンスにログオンします。
3. コマンドプロンプトで sh ns license コマンドを入力して、ライセンスでサポートされている機能を表示します。

   例

```
1 sh ns license
2         License status:
3                             Web Logging: YES
4                         Surge Protection: YES
5                           .......
6
7                           HTML Injection: YES
8  Done
9 <!--NeedCopy-->
```

GUI を使用してライセンスされた機能を検証するには

1. Web ブラウザで、ADC アプライアンスの IP アドレスを入力します（http://192.168.100.1など）。
2. ［User Name］ボックスと［Password］ボックスに管理者資格情報を入力します。

3. ユーザー名とパスワードを入力し、「ログイン」をクリックします。
4. ナビゲーションペインで、[システム] を展開し、[ライセンス] をクリックします。ライセンスされた機能の横に緑色のチェックマークが表示されます。

機能を有効または無効にする

Citrix ADC アプライアンスを初めて使用する場合、その機能を使用するには、その機能を有効にする必要があります。機能を有効にする前に設定すると、警告メッセージが表示されます。設定は保存されますが、この機能が有効になった後にのみ適用されます。

CLI を使用して機能を有効にするには

コマンドプロンプトで次のコマンドを入力して、機能を有効にして構成を確認します。

- enable feature <FeatureName>
- show feature

例

```
enable feature lb cs
done
 >show feature

        Feature                     Acronym
           Status
        -------                     -------
           ------
 1)     Web Logging                 WL                  OFF
 2)     Surge Protection            SP                  ON
 3)     Load Balancing              LB                  ON
 4)     Content Switching           CS                  ON
 5)     Cache Redirection           CR                  ON
 .
 .
 .
 24)    NetScaler Push              push                OFF
 Done
<!--NeedCopy-->
```

次に、ロードバランシング（lb）およびコンテンツスイッチング（cs）を有効にする例を示します。

特定の機能でライセンスキーを使用できない場合は、その機能に関する次のエラーメッセージが表示されます。

エラー: 機能がライセンスされていません

注：オプション機能を有効にするには、機能固有のライセンスが必要です。たとえば、Citrix NetScaler アドバンスエディションのライセンスを購入してインストールしたとします。ただし、統合キャッシュ機能を有効にするには、AppCache ライセンスを購入してインストールする必要があります。

CLI を使用して機能を無効にするには

コマンドプロンプトで次のコマンドを入力して、機能を無効にし、構成を確認します。

- disable feature <FeatureName>
- show feature

例

次に、ロードバランシング（LB）を無効にする例を示します。

```
> disable feature lb
 Done
> show feature

        Feature                       Acronym
           Status
        -------                       -------
           ------
 1)     Web Logging                   WL                    OFF
 2)     Surge Protection              SP                    ON
 3)     Load Balancing                LB                    OFF
 4)     Content Switching             CS                    ON
 .
 .
 .
 24)    NetScaler Push                push                  OFF
 Done
>
<!--NeedCopy-->
```

ライセンスの有効期限情報を確認する

GUI または CLI を使用して Citrix ADC ライセンスの有効期限情報を確認できます。

GUI を使用して **Citrix ADC** ライセンスの有効期限情報を確認するには：

[設定] **>** [システム] **>** [ライセンス] に移動します。

ADC ライセンスの有効期限が 30 日以下の場合、GUI アラートが表示されます。

CLI を使用してライセンスの有効期限情報を確認するには:

コマンド「shownslicense」を入力します。

```
> sh license
    License status:

    Web Logging: YES
    Surge Protection: YES

    Web Logging: YES
    Surge Protection: YES

    ...

Days to expiry: 204

Done
>
<!--NeedCopy-->
```

ライセンスの有効期限が切れると、Citrix ADC アプライアンスは SNMP アラームを生成します “NS_LICENSE_EXPIRY,” 有効期限イベントがコンソールに記録されます。

ライセンスの有効期限が切れると、Citrix ADC アプライアンスは自動的に再起動してライセンスを取り消します。

Citrix ADC アプライアンスで Citrix サービスプロバイダー（CSP）ライセンスが使用されている場合、アプライアンスはライセンスを取り消すために自動的に再起動しません。ただし、ユーザーがアプライアンスを再起動すると、ライセンスなしとして再起動されます。

ライセンスをアップグレードする

Citrix ADC アプライアンスは、より容量の大きいライセンスを購入することで、あるファミリエディションから別のファミリエディションに、またあるキャパシティ範囲から別のファミリエディションにアップグレードできます。

アップグレードには次の 2 つのタイプがあります。

- エディションのアップグレード：スタンダードからアドバンス、スタンダードからプレミアム、アドバンスからプレミアム。エディションのアップグレードは、同じ帯域幅内である必要があります。
- 容量のアップグレード：vCPU と帯域幅の両方について、容量を小さいものから大きいものにアップグレードできます。容量のアップグレードは、同じエディション（スタンダード、アドバンス、プレミアム）でのみ実行できます。

容量とエディションの両方をアップグレードする場合は、最初に容量をアップグレードし、アプライアンスを再起動してから、エディションをアップグレードします。

例：VPX 10 Mbps StandardEdition ライセンスを VPX200 Mbps Premium Edition にアップグレードするには、アップグレードを 2 つのステップで実行する必要があります。

- VPX を 10Mbps StandardEdition から 200Mbps StandardEdition にアップグレードします。
- VPX を 200Mbps StandardEdition から 200Mbps PremiumEdition にアップグレードします。

> 注
>
> Citrix Application Delivery Management（ADM）を使用して、共通の帯域幅とインスタンスプールで構成されるライセンスフレームワークを作成できます。詳細については、「Citrix ADC プール容量」を参照してください。

関連リソース

- The Citrix Licensing System
- Citrix ADC VPX ライセンスの割り当て方法

データガバナンス

August 12, 2022

Citrix ADM サービスコネクトって何ですか

Citrix Application Delivery Management（ADM）サービス接続は、Citrix ADC MPX、SDX、VPX インスタンス、および Citrix Gateway アプライアンスの Citrix ADM サービスへのシームレスなオンボーディングを可能にする機能です。この機能により、Citrix ADC インスタンスまたは Citrix Gateway アプライアンスは、Citrix ADM サービスに自動的かつ安全に接続し、システム、使用状況、およびテレメトリデータをそのサービスに送信できます。このデータに基づいて、Citrix ADM サービス上の Citrix ADC インフラストラクチャに関する洞察と推奨事項が得られます。

Citrix ADM サービス接続機能を使用して、Citrix ADC インスタンスまたは Citrix Gateway アプライアンスを Citrix ADM サービスにオンボーディングします。オンプレミスでもクラウドでも、すべての Citrix ADC および Citrix Gateway アセットを管理することもできます。また、パフォーマンスの問題、高いリソース使用率、重大なエラーなどをすばやく特定するのに役立つ豊富な可視性機能へのアクセスもメリットがあります。Citrix ADM サービスは、Citrix ADC インスタンスおよびアプリケーションに幅広い機能を提供します。Citrix ADM サービスの詳細については、「Citrix Application Delivery Management サービス」を参照してください。

> 重要
>
> - Citrix Gateway アプライアンスは、Citrix ADM サービス接続機能もサポートしています。より簡単にするために、Citrix Gateway アプライアンスは連続するセクションで明示的に呼び出されません。

Citrix ADM サービスとは何ですか

Citrix ADM サービスは、Citrix ADC インスタンスの管理、監視、調整、自動化、およびトラブルシューティングを支援するクラウドベースのソリューションです。また、Citrix ADC インスタンス、およびアプリケーションの状態、パフォーマンス、セキュリティに関する分析的洞察と精選された機械学習ベースの推奨事項も提供します。詳細については、Citrix ADM サービスの概要を参照してください。

Citrix ADM サービス接続はどのように有効になっていますか

Citrix ADC または Gateway をリリース 13.0 ビルド 61.xx 以降にインストールまたはアップグレードすると、Citrix ADM サービス接続がデフォルトで有効になります。

Citrix ADM サービス接続を使用してキャプチャされるデータはどれですか**?**

Citrix ADM サービス接続を使用すると、次の詳細がキャプチャされます。

- **Citrix ADC** の詳細
  - シリアル ID
  - エンコードされたシリアル ID
  - ホスト ID
  - UUID

  - 管理 IP アドレス
  - ホスト名
  - バージョン
  - ビルドタイプ
  - 構築
  - ライセンスの種類
  - Hypervisor
  - 導入タイプ (スタンドアロン/HA)
  - プラットフォームタイプ
  - プラットフォームの説明
  - システム ID
  - ADC で有効になっているモード
  - ADC で有効になっている機能

- ライセンス情報
  - Citrix ADC でライセンスされている機能
  - ライセンス番号

- 主な使用状況メトリック
  - システム日時
  - CPU の使用率
  - 管理 CPU パーセンテージ
  - スループット
  - SSL 新規セッション
  - SSL 暗号化スループット
  - SSL 復号化スループット
  - システム稼働時間

- 構成
  - ns.conf ファイル

> 注
>
> Citrix ADM サービス接続は、Citrix ADC アプライアンスから Citrix ADM サービスに`ns.conf`ファイルを送信する前に、暗号化されたパスワードまたはハッシュされたパスワードを匿名化します。Citrix ADM サービスコネクトは、「-encrypted」または「-passcrypt」パラメータをチェックし、関連する暗号化またはハッシュされた値を「XXXX」に置き換えます。Citrix ADM サービス接続は、`ns.conf`ファイルをエンコードして圧縮し、Citrix ADM サービスエンドポイントに送信します。

- 重大なエラーの詳細
  - ハードディスク障害
  - SSL カードの障害

  - 電源ユニット (PSU) の障害
  - フラッシュドライブの障害
  - ウォームリブート
  - 持続的なメモリ使用量が 90% を超えるか、メモリリークが発生している
  - レート制限の持続的な低下

- 診断の詳細

  > 注:
  >
  > ADM 診断ツールでは、次の診断詳細が使用されます。詳しくは、Citrix ADM の「診断ツール 」トピックを参照してください。

  - ADC CLI ステータス
  - ADC DNS ステータス
  - ADM エンドポイント「adm.cloud.com」へのネットワーク接続ステータス
  - ADM エンドポイント「agent.adm.cloud.com」へのネットワーク接続ステータス
  - ADM トラストサービス「trust.citrixnetworkapi.net」へのネットワーク接続ステータス
  - ADM ダウンロードサイト「download.citrixnetworkapi.net」へのネットワーク接続ステータス

データはどのように使用されますか

データを収集することにより、Citrix ADC インストールに関するタイムリーかつ詳細な洞察を提供できます。これには、次のようなものがあります。

- 主要な指標。CPU、メモリ、スループット、SSL スループットに関する主要なメトリックスの詳細、および Citrix ADC インスタンスでの異常な動作を強調表示します。
- 重大なエラー。Citrix ADC インスタンスで発生した可能性のある重大なエラー。
- 導入アドバイザリ。スタンドアロンモードで展開されているがスループットが高く、単一障害点に対して脆弱な Citrix ADC インスタンスを特定します。
- 診断ツール。ADC インスタンスを Citrix ADM にオンボーディングすると、ADC インスタンスが正常にオンボーディングされない問題がいくつか発生することがあります。この問題をトラブルシューティングするには、診断ツールを手動で使用するか、ADM GUI で診断情報を確認します。詳細については、「診断ツール」を参照してください。

収集されたデータはどのくらいの期間保持されますか

収集されたデータはすべて 13 か月以内に保持されます。

Citrix ADC から Citrix ADM サービス接続機能を無効にしてサービスの使用を終了することにした場合、以前に収集されたデータは 30 日後に削除されます。

データはどこに保存され、どの程度安全ですか

Citrix ADM サービスコネクトによって収集されたすべてのデータは、米国、欧州連合、オーストラリアおよびニュージーランド（ANZ）の 3 つの地域のいずれかに保存されます。詳細については、地理的考慮事項を参照してください。

データは、データベース層で厳密なテナント分離を使用して安全に保存されます。

Citrix ADM サービス接続を無効にする方法は

Citrix ADM サービス接続によるデータ収集を無効にする場合は、「Citrix ADM サービス接続を有効または無効にする方法」を参照してください。

Citrix ADC アプライアンス用の Citrix ADM サービス接続の概要

February 21, 2022

Citrix ADM サービスは、Citrix ADC インスタンスの管理、監視、調整、自動化、およびトラブルシューティングを支援するクラウドベースのソリューションです。また、アプリケーションの正常性、パフォーマンス、およびセキュリティに関する分析的洞察と精選された機械学習ベースの推奨事項も提供します。詳細については、「Citrix Application Delivery Management サービス」を参照してください。

Citrix Application Delivery Management（ADM）サービス接続は、Citrix ADC インスタンスの Citrix ADM サービスへのシームレスなオンボーディングを可能にする機能です。この機能により、Citrix ADC インスタンスと Citrix ADM サービスが総合的なソリューションとして機能し、お客様にさまざまなメリットを提供します。

Citrix ADM サービス接続機能を使用すると、Citrix ADC インスタンスは Citrix ADM サービスに自動的に接続し、システム、使用状況、およびテレメトリデータをサービスに送信できます。このデータに基づいて、Citrix ADM サービスは、Citrix ADC および Gateway インフラストラクチャに関する次のような洞察と推奨事項を提供します。

- 脆弱な ADC アプライアンスを強調するセキュリティアドバイザリの洞察。
- アップグレードのアドバイザリーインサイトは、メンテナンス終了と寿命の終了に達している、または到達しようとしている ADC アプライアンスを強調します。
- パフォーマンスの問題、高いリソース使用率、重大なエラーをすばやく特定します。

Citrix ADM サービスの機能を活用するために、Citrix ADC インスタンスを Citrix ADM サービスにオンボードすることを選択できます。オンボーディングプロセスでは ADM サービスコネクトが使用され、エクスペリエンスをスムーズかつ高速化します。

> 注意事項
>
> - Citrix ADM サービス接続が、Citrix ADC MPX、SDX、VPX インスタンスおよび Citrix Gateway アプライアンスで利用できるようになりました。
> - この Citrix ADM サービス接続機能を使用する Citrix ADM サービスのイニシアチブは、ADM サービス

接続ベースのロータッチオンボーディングです。詳細については、「Citrix ADM サービス接続を使用した Citrix ADC インスタンスのロータッチオンボーディング」を参照してください。

- ADC インスタンスで ADM サービス接続が有効になっている場合、特定の診断情報が自動的に ADM サービスに送信されます。

詳しくは、「データガバナンス」を参照してください。

重要

Citrix ADM サービス接続は、プローブデータの収集に失敗し、次の条件を満たす場合、ADC アプライアンスを ADM サービスにオンボーディングする際には役立ちません。

- `NSinternal`ユーザーアカウントが無効になっています。
- SSH 公開キーが設定されていません。

上記のシナリオを克服するには、次のいずれかを実行することをお勧めします。

- `set ns param -internaluserlogin ENABLED`を使用して、`internaluser`ユーザアカウントを有効にします。
- 公開キー認証を設定します。詳細については、「SSH キーとパスワードなしで Citrix ADC アプライアンスにアクセスする」を参照してください。

Citrix ADM サービスは、**Citrix ADM** サービスとのサポートをどのように接続しますか

これは、Citrix ADC の Citrix ADM サービス接続機能が Citrix ADM サービスとどのように相互作用するかについての高レベルのワークフローです。

1. Citrix ADC アプライアンスの Citrix ADM サービス接続機能は、定期的なプローブ要求を使用して Citrix ADM サービスと自動接続します。
2. このリクエストには、システム、使用状況、およびテレメトリデータが含まれており、これを使用して Citrix ADM サービスにより、Citrix ADC インフラストラクチャに関する洞察と推奨事項が提供されます。同様に、パフォーマンスの問題、高いリソース使用率、重大なエラーを迅速に特定できます。
3. 洞察と推奨事項を表示し、ADC インスタンスを Citrix ADM サービスにオンボーディングして Citrix ADC インスタンスの管理を開始することを決定できます。
4. オンボーディングを決定すると、Citrix ADM サービス接続機能により、オンボーディングをシームレスに完了できます。

Citrix ADM サービス接続はどのバージョンの **Citrix ADC** でサポートされていますか

Citrix ADM サービス接続は、すべての Citrix ADC プラットフォームとすべてのアプライアンスモデル（MPX、VPX、SDX）でサポートされています。Citrix ADC リリース 13.0 ビルド 61.xx 以降、Citrix ADC アプライアンスではデフォルトで Citrix ADM サービス接続が有効になります。

Citrix ADM サービス接続を有効にするにはどうすればよいですか

既存の Citrix ADC ユーザーで、Citrix ADC リリース 13.0 ビルド 61.xx にアップグレードする場合、Citrix ADM サービス接続はアップグレードプロセスの一部としてデフォルトで有効になります。

Citrix ADC リリース 13.0 ビルド 61.xx をインストールする Citrix ADC 新規顧客の場合、Citrix ADM サービス接続はインストールプロセスの一部としてデフォルトで有効になります。

> 注
>
> 新しい Citrix ADC アプライアンスとは異なり、既存の Citrix ADC アプライアンスは Citrix Insight Service（CIS）または Call Home を介してルートを検索します。

Citrix ADM サービス接続を有効または無効にするにはどうすればよいですか

Citrix ADM サービス接続は、CLI、GUI、または NITRO API メソッドから有効または無効にできます。

CLI を使用する

CLI を使用して Citrix ADM サービス接続を有効にするには

コマンドプロンプトで入力します。

```
1 set adm parameter  - admserviceconnect ENABLED
```

CLI を使用して Citrix ADM サービス接続を無効にするには

コマンドプロンプトで入力します。

```
1 set adm parameter  - admserviceconnect DISABLED
```

> 重要
>
> Citrix ADC がリリース 13.0 ビルド 61.xx の場合、Citrix ADC サービス接続を有効または無効にするパラメータ名は「自動接続」です。たとえば、サービス接続を有効にするには、`set adm parameter - autoconnect ENABLED` コマンドを使用します。

GUI の使用

Citrix ADC GUI を使用して Citrix ADM サービス接続を無効にするには

1. Web ブラウザーに Citrix ADC アプライアンスの IP アドレスを入力します（例：http://192.0.2.10）。
2. ［**User Name**］と ［**Password**］に管理者の資格情報を入力します。

3. [システム] **>** [設定] **>** [**ADM** パラメータの構成] に移動します。
4. [**ADM** パラメーターの構成] ページで、[**Citrix ADM** サービス接続を有効にする] ダイアログボックスの選択を解除し、[**OK**] をクリックします。

NITRO API を使用する

Citrix ADM サービス接続を無効にするには、**NITRO** コマンドを使用します。

- Citrix ADC リリース 13.0 ビルド 61.xx では、次のコマンドを使用して、Citrix ADM サービス接続を有効または無効にできます。
  - `curl -X PUT -H "Content-Type:application/json"http://192.0.2.10/nitro/v1/config/admparameter> -d '{ "admparameter":{ "autoconnect":"enabled" } } ' -u nsroot:Test@1`
- Citrix ADC リリース 13.0 ビルド 64.xx から、「自動接続」パラメータ名が`admserviceconnect`に変更されます。次のコマンドを使用して、Citrix ADM サービス接続を無効にできます。
  - `curl -X PUT -H "Content-Type:application/json" http://192.0.2.10/nitro/v1/config/admparameter -d '{ "admparameter":{ "admserviceconnect":"disabled" } } ' -u nsroot:Test@1`

診断ツール

ADC インスタンスを Citrix ADM にオンボーディングすると、ADC インスタンスが正常にオンボーディングされない問題がいくつか発生することがあります。この問題をトラブルシューティングするには、診断ツールを手動で使用するか、ADM GUI で診断情報を確認します。

- ADM Service Connect を使用してキャプチャされた詳細について詳しくは、「データガバナンス」を参照してください。
- 診断ツールについて詳しくは、「診断ツール」を参照してください。

Citrix ADM 組み込みエージェントの動作

Citrix ADC リリース 13.0 ビルド 61.xx 以降から、Citrix ADC インスタンスで使用可能な Citrix ADM 組み込みエージェントは ADM サービスと通信します。各 ADC インスタンスで手動で初期化する必要なく通信します。ADM サービスとの通信が確立された後、組み込みエージェントは、定期的に最新のソフトウェアバージョンに自動アップグレードすることにより、常緑を維持します。

以前は、ADM サービスとの通信を確立したり、定期的な自動アップグレードを行うために、`mastools`コマンドを使用して ADC インスタンスで組み込みエージェントを初期化する必要がありました。
詳細については、「インスタンスを管理するための ADC 組み込みエージェントの設定」を参照してください。

参照ドキュメント

Citrix ADM サービス接続の詳細については、次のトピックを参照してください。

- データガバナンス: データガバナンス。
- Citrix ADM サービス: Citrix Application Delivery Management サービス。

Citrix ADC アプライアンスのアップグレードとダウングレード

April 7, 2022

> 注
>
> Citrix ADC または Citrix Gateway をインストールまたはアップグレードして 13.0 ビルド 61.xx 以降をリリースすると、Citrix ADM サービス接続がデフォルトで有効になります。詳細については、「データガバナンスと CitrixADM サービス接続」を参照してください。

Citrix ADC 13.0 は、機能が強化された新機能と更新された機能を提供します。機能拡張についてはすべて、リリース発表に付属のリリースノートに記載されています。ソフトウェアをアップグレードする前に、リリースノートのドキュメントをお読みください。

このセクションでは、**Citrix ADC GUI** または **CLI** を使用して、**Citrix ADC** アプライアンス（**MPX** および **VPX**）ファームウェアをアップグレードおよびダウングレードする方法について説明します。

Citrix ADM を使用して **Citrix ADC** アプライアンスをアップグレードすることもできます。詳しくは、次のトピックを参照してください:

- Citrix ADM サービスが Citrix ADC アップグレードをより簡単にサポートする 10 の方法
- Citrix ADM サービスを使用して Citrix ADC インスタンスをアップグレードする
- Citrix ADM ソフトウェアを使用して Citrix ADC インスタンスをアップグレードする

Citrix ADC SDX アプライアンスのアップグレードの詳細については、「シングルバンドルアップグレード」を参照してください。

はじめに

December 7, 2021

アップグレードまたはダウングレードプロセスを開始する前に、次の点を確認してください。

- Citrix ADC アプライアンスのアップグレードに割り当てられた時間。組織の変更管理手順に従います。アップグレードの実行に 2 倍の時間を割り当てます。各 Citrix ADC アプライアンスをアップグレードするのに十分な時間を割り当てます。

- 組織のサポート契約を評価します。Citrix テクニカルサポートまたは Citrix 認定パートナーのサポートについては、アプライアンスのシリアル番号、サポート契約、連絡先の詳細を文書化します。
- ライセンスフレームワークとライセンスの種類。ソフトウェアエディションのアップグレードには、次のような新しいライセンスが必要になる場合があります。
  - 標準版から上級版へのアップグレード、または
  - スタンダードエディションからプレミアムエディション、または
  - プレミアムエディションへのアドバンスエディション。

  バージョン 13.0 にアップグレードしても、既存の Citrix ADC ライセンスは引き続き機能します。詳細については、「ライセンス」を参照してください。
- 新規および非推奨のコマンド、パラメータ、および SNMP OIDを確認します。
- Citrix ADC MPX ハードウェアおよびソフトウェアの互換性マトリクスを確認します。
- Citrix ADC Gateway のログオンページがカスタマイズされている場合は、UI テーマがデフォルトに設定されていることを確認します。
- LOM をアップグレードする場合は、LOM ファームウェアのアップグレードページを確認します。
- Citrix ADC ダウンロードから Citrix ADC ファームウェアをダウンロードします。Citrix ADC ファームウェアをダウンロードする詳細な手順については、Citrix ADC リリースパッケージをダウンロードするを参照してください。
- バックアップファイル。構成ファイル、カスタマイズファイル、証明書、モニタースクリプト、ライセンスファイルなどのバックアップを手動で実行するか、Citrix ADC CLI または GUI- バックアップと復元を使用したバックアップについては、次のドキュメントを参照してください。
  - バックアップ用のその他の一般的なカスタマイズファイルについては、次のリストを参照してください。
    * /nsconfig/monitors/*.pl
    * /nsconfig/htmlinjection/*
    * /nsconfig/rc.netscaler
  - カスタマイズフォルダをバックアップします。/var/customizationsこれは通常下です。カスタマイズの例として、ロゴ付きのログオンページがあります。カスタマイズフォルダーをコピーした後、アプライアンスをアップグレードする前に、Citrix ADC アプライアンスからフォルダーを削除する必要があります。カスタマイズを適してアップグレードすると、いくつかの問題が発生する可能性があります。

  > Citrix:
  > 上記のバックアップ手順を確認することを強くお勧めします。Citrix ADC アプライアンスで更新が完了しない場合のアクションプランを用意します。
- アップグレードを実行する前に、Citrix ADC アプライアンス用の/var および/flash ディレクトリに十分な領域があることを確認します。/var には 5 GB の空き領域が必要です (アップグレードバンドルの場合は 1 GB + アップグレードプロセスに 4 GB)
  /flash には、新しいカーネル上でコピーするのに十分な領域が必要です。これは 140 MB から 160MB の間で

異なります。少なくとも 250 MB の空き領域があることを確認します。
/var のディスク領域をクリアする方法の詳細については、「Citrix ADC アプライアンスの問題をログに記録するために/var ディレクトリの空き領域を解放する方法」を参照してください。
/flash のディスク領域のクリアの詳細については、https://support.citrix.com/article/CTX133587を参照してください。

- Citrix ADC アプライアンスの整合性を検証します。Citrix ADC ハードウェアアプライアンスを使用している場合は、`fsck`を実行して、ディスクチェックを実行し、Citrix Citrix ADC ハードドライブの整合性を検証することを強くお勧めします。エラーが発生した場合は、ハードディスクドライブをリセットし、ディスクチェックコマンドを繰り返します。エラーメッセージが再び表示される場合は、Citrix サポートに連絡して問題をさらに調査してください。
  - fsck コマンドを使用して、ハードドライブのディスクの整合性を検証します。詳細については、CTX122845を参照してください。
  - 診断バンドルファイルを使用し、分析のためにログを Citrix Insight Service にアップロードして、Citrix ADC アプライアンスの整合性を検証します。詳細については、「テクニカルサポートバンドルの収集方法」を参照してください。
- Citrix ADC VPX サポートマトリックスと使用ガイドラインを確認してください。
- FAQ セクションを確認してください。
- 一度に 1 つのメジャーリリースにアップグレードすることをお勧めします。最新バージョンに直接アップグレードしないでください。

  たとえば、Citrix ADC アプライアンスがリリース 12.0 にあり、リリース 13.0 にアップグレードする場合は、まずアプライアンスをリリース 12.1 にアップグレードし、次にリリース 13.0 にアップグレードする必要があります。
- テスト環境でアップグレード手順を確認します。

Citrix ADC アプライアンスをアップグレードまたはダウングレードするための前提条件の詳細については、以下のサポート記事を参照してください。

- CTX220371: Citrix ADC アップグレード前とアップグレード後に記事を読む必要がある

アップグレードに関する考慮事項-SNMP 構成

October 7, 2021

SNMP アラームのタイムアウトパラメータは、アラーム設定に影響を与えない内部オプションです。

これらの SNMP アラーム設定に変更を加えない場合でも、実行構成 (sh running) および保存された構成 (ns.conf) の SNMP アラーム設定に Timeout パラメータが表示されることがあります。

タイムアウト設定の問題が修正されたリリースビルドにアップグレードすると、SNMP 構成が誤ってデフォルト値にリセットされます。

次の SNMP アラーム（設定されている場合）は、アップグレード中に影響を受けます。

- APPFW-BUFFER-OVERFLOW
- APPFW-COOKIE
- APPFW-CSRF-TAG
- APPFW-DENY-URL
- APPFW-FIELD-CONSISTENCY
- APPFW-FIELD-FORMAT
- APPFW-POLICY-HIT
- APPFW-REFERER-HEADER
- APPFW-SAFE-COMMERCE
- APPFW-SAFE-OBJECT
- APPFW-SQL
- APPFW-START-URL
- APPFW-VIOLATIONS-TYPE
- APPFW-XML-ATTACHMENT
- APPFW-XML-DOS
- APPFW-XML-SCHEMA-COMPILE
- APPFW-XML-SOAP-FAULT
- APPFW-XML-SQL
- APPFW-XML-VALIDATION
- APPFW-XML-WSI
- APPFW-XML-XSS
- APPFW-XSS
- CLUSTER-BACKPLANE-HB-MISSING
- CLUSTER-NODE-HEALTH
- CLUSTER-NODE-QUORUM
- CLUSTER-VERSION-MISMATCH
- COMPACT-FLASH-ERRORS
- CONFIG-CHANGE
- CONFIG-SAVE
- HA-BAD-SECONDARY-STATE
- HA-NO-HEARTBEATS
- HA-SYNC-FAILURE
- HA-VERSION-MISMATCH
- HARD-DISK-DRIVE-ERRORS
- HA-STATE-CHANGE
- HA-STICKY-PRIMARY

- PORT-ALLOC-FAILED
- SYNFLOOD

これらの SNMP アラーム構成は、Citrix ADC を次のリリースビルドにアップグレードすると影響を受けます。

- リリース 11.1 ビルド 61.2 以降
- リリース 12.0 ビルド 61.0 以降
- リリース 12.1 ビルド 30.1 以降
- リリース 13.0 ビルド 51.4 以降

例

CLUSTER-NODE-HEALTHSNMP アラームの例を考えてみましょう。

```
1 CLUSTER-NODE-HEALTH SNMP alarm is set up by using the Citrix ADC
     command line:
2
3 > set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -
     severity Major
4
5 > save config
6 <!--NeedCopy-->
```

この SNMP アラーム構成は、保存された構成ファイル（ns.conf）に次のように表示されます。

```
1 set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -severity
     Major -timeout 86400
2
3 <!--NeedCopy-->
```

上記のいずれかのリリースビルドへのアップグレード中に、ns.log ファイルに次のエラーが表示されます。

```
1 May 23 09:14:46 <local0.err> ns nsconfigd: __init_config_filter(): (
     null) line 0: No such argument [-timeout]>> set snmp alarm CLUSTER-
     NODE-HEALTH -time 111 -state DISABLED -severity Major -timeout
     86400.
2 <!--NeedCopy-->
```

アップグレード後、SNMP アラーム設定はデフォルト値にリセットされます。

回避方法

この問題を回避するには、以下のいずれかを行います。

- アップグレードする前に、保存された構成ファイル（ns.conf）の SNMP 構成からタイムアウト設定を削除してください。
- アップグレード後、timeout パラメータなしで SNMP アラームを再設定します。

Citrix ADC リリースパッケージをダウンロードする

October 7, 2021

Citrix ADC リリースパッケージをダウンロードするには、次の手順を実行します。

1. Web ブラウザで Citrix ADC ダウンロードページを開きます 。
2. [Citrix ADC ダウンロード] ページで、更新する **Citrix ADC** リリースを展開します。
3. 適切なカテゴリの 1 つを展開し、Citrix ADC ビルドリンクをクリックします。たとえば、Citrix ADC ファームウェアのバージョンをダウンロードするには、[ファームウェア] を展開し、ダウンロードする Citrix ADC ビルドをクリックします。
4. 選択した Citrix ADC ビルドページで、[ビルド] セクションを展開し、[ファイルのダウンロード] をクリックして Citrix ADC ビルドパッケージをダウンロードします。

> 注:
> チェックサムは、ダウンロードしたビルドパッケージが Web サイトでホストされている実際のパッケージと一致することを確認するために提供されています。チェックサムは、正しいビットがあることを確認するための重要なチェックです。

Citrix ADC スタンドアロンアプライアンスのアップグレード

October 7, 2021

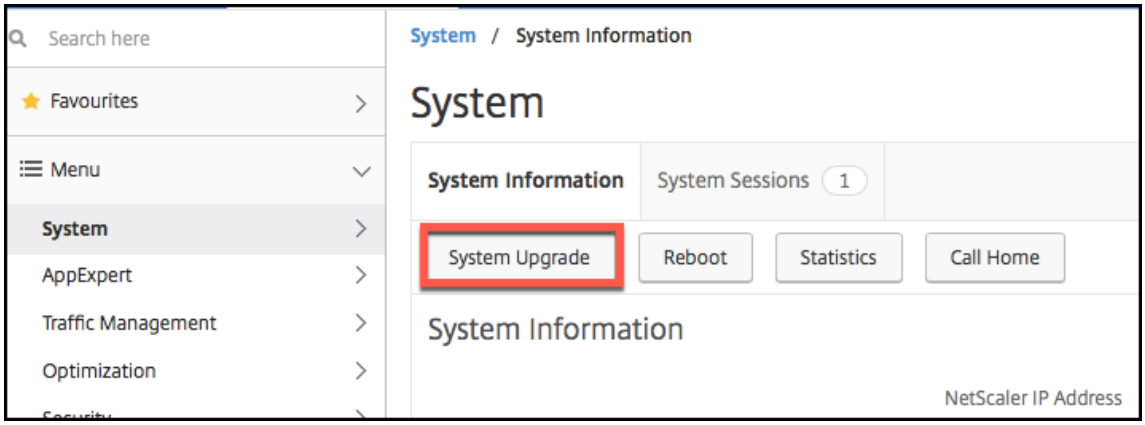
システムソフトウェアをアップグレードする前に、「開始する前に 」セクションを読み、必要なファイルのバックアップや Citrix ADC ファームウェアのダウンロードなどの前提条件を完了してください。

GUI を使用して Citrix ADC スタンドアロンアプライアンスをアップグレードする

GUI を使用してスタンドアロンの Citrix ADC をリリース 13.0 にアップグレードする手順は、次のとおりです。

1. Web ブラウザで、Citrix ADC IP アドレスを入力します（例: `http://10.102.29.50`）。
2. [ユーザー名] と [パスワード] に、管理者の資格情報 (nsroot/nsroot) を入力し、**[**ログオン] をクリックします。

3. GUI で、[システムのアップグレード] をクリックします。



4. 「ファイルの選択」メニューから、適切なオプション（「ローカル」または「アプライアンス」）を選択します。Appliance オプションを使用する場合は、まず Citrix ADC にファームウェアをアップロードする必要があります。WinSCP などの任意のファイル転送方法を使用して、Citrix ADC ファームウェアをアプライアンスにアップロードできます。
5. 正しいファイルを選択し、[アップグレード] をクリックします。
6. 指示に従って、ソフトウェアをアップグレードします。
7. プロンプトが表示されたら、[再起動] を選択します。

アップグレード後、アプライアンスにアクセスする前に、すべてのブラウザインスタンスを閉じ、コンピュータのキャッシュをクリアします。

CLI を使用した Citrix ADC スタンドアロンアプライアンスのアップグレード

CLI を使用してスタンドアロン Citrix ADC をリリース 13.0 にアップグレードする手順は、次のとおりです。

この手順では、<release> と<releasenumber> はダウングレードするリリースバージョンを表し、<targetbuildnumber>はダウングレードするビルド番号を表します。この手順には、アップグレード中に /etc ディレクトリにプッシュされた更新が失われないようにするためのオプションの手順が含まれています。

1. PuTTy などの SSH クライアントを使用して、アプライアンスへの SSH 接続を開きます。
2. 管理者の資格情報を使用して、アプライアンスにログオンします。構成実行を保存します。プロンプトで、次のように入力します。

   ```
   save config
   ```

3. 次のコマンドを実行して、シェルプロンプトに切り替えます。

   ```
   shell
   ```

4. ns.conf ファイルのコピーを作成します。シェルプロンプトで、次のように入力します。

   - `cd /nsconfig`

- cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber>

構成ファイルは別のコンピュータにバックアップする必要があります。

5. （オプション）/etc ディレクトリ内の次のファイルの一部を変更し、永続性を維持するために /nsconfig にコピーした場合、アップグレード中に /etc ディレクトリにプッシュされた更新はすべて失われる可能性があります。

   - ttys
   - resolv.conf
   - sshd_config
   - host.conf
   - newsyslog.conf
   - host.conf
   - httpd.conf
   - rc.conf
   - syslog.conf
   - crontab
   - monitrc

   これらの更新内容が失われないようにするには、/var/nsconfig_backupディレクトリを作成し、カスタマイズしたファイルをこのディレクトリに移動します。つまり、次のコマンドを実行して、/etc ディレクトリで変更し、/nsconfig にコピーしたファイルを移動します。

   ```
   cp /nsconfig/<filename> /var/nsconfig_backup
   ```

   例:

   ```
   cp /nsconfig/syslog.conf /var/nsconfig_backup
   ```

6. インストールパッケージの場所を作成します。シェルプロンプトで次のように入力します。

   - cd /var/nsinstall
   - cd <releasenumber>

     > 注:
     >
     > 目的のリリース番号ディレクトリが存在しない場合は、次のコマンドを使用して作成します。
     >
     > mkdir <releasenumber>
     >
     > 例:
     >
     > mkdir 13.0

   - mkdir build_<targetbuildnumber>
   - cd build_<targetbuildnumber>

7. WinSCP などのファイル転送方法を使用して、ダウンロード済みの Citrix ADC ファームウェアを上記の手順で作成したビルドディレクトリにコピーします。Citrix ADC ファームウェアのダウンロードの詳細については、「開始する前に」セクションを参照してください。
8. インストールパッケージの内容を展開します。例:

   ```
   tar -xvzf build-13.0-37.2_nc_64.tgz
   ```

9. installns スクリプトを実行して、新しいバージョンのシステムソフトウェアをインストールします。

   ```
   ./installns
   ```

10. プロンプトが表示されたら、Citrix ADC を再起動します。
11. (オプション) [開始前] セクションで ns.conf ファイルのコピーを作成した場合は 、次の手順を実行します。

    a) /var/nsconfig_backupと/etc内のファイルを手動で比較し、/etcで適切な変更を加えます。
    b) 永続性を維持するには、/etc内の更新されたファイルを /nsconfig に移動します。
    c) アプライアンスを再起動して、変更を有効にします。

以下は、Citrix ADC ファームウェアのアップグレードの例です。

```
login: nsroot

Password: nsroot

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

Done

> save config

> shell

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

root@NSnnn# cd /var/nsinstall

root@NSnnn# cd 13.0

root@NSnnn# mkdir build_43.1

root@NSnnn# cd build_43.1

root@NSnnn# ftp <FTP server IP address>

ftp> mget build-13.0-41.1_nc.tgz
```



```
26
27 ftp> bye
28
29 root@NSnnn# tar xzvf build-13.0-41.1_nc.tgz
30
31 root@NSnnn# ./installns
32
33 installns version (13.0-41.1) kernel (ns-13.0-41.1_nc.gz)
34
35 ...
36
37 Copying ns-13.0-41.1_nc.gz to /flash/ns-13.0-41.1_nc.gz ...
38
39 ...
40
41 Installation has completed.
42
43 Reboot NOW? [Y/N] Y
```

NITRO API を使用して **Citrix ADC** スタンドアロンアプライアンスをアップグレードする

NITRO API を使用して Citrix ADC をアップグレードまたはダウングレードするには、Citrix ADC アップグレードとダウングレードを単一の API で自動化するを参照してください。

アップグレード後、**Citrix ADC** アプライアンスのエンティティステータスを確認します

Citrix ADC アプライアンスをアップグレードした後、次のエンティティのステータスを確認します。

- 仮想サーバーは UP 状態です
- モニターは UP 状態です
- GSLB サイトは問題なく同期します
- すべての証明書がアプライアンスに存在します
- すべてのライセンスがアプライアンスに存在します

Citrix ADC 13.0 ソフトウェアアップデートの確認とインストール

アップデートが利用可能になったら、Citrix ADC ソフトウェアをアップデートして、パフォーマンスを向上させます。Citrix ADC アップデートには、機能の向上、パフォーマンスの修正、または機能強化が含まれます。リリースノートを読んで、アップデートで利用可能な修正と機能強化を確認してください。ソフトウェア更新プログラムを確認してインストールするには、次の手順を実行します。

1. Citrix ADC ホームページで、右上隅にある［**nsroot** ］メニューから［更新の確認］をクリックします。

2. [利用可能な最新のシステムソフトウェアの更新] ページで、インストール可能なソフトウェア更新プログラムを確認します。
3. [ダウンロード] をクリックして、Citrix のダウンロード Web サイトからインストールパッケージをダウンロードします 。
4. ソフトウェアパッケージをダウンロードしたら、CLI または GUI の手順で更新プログラムをインストールします。

> 注
>
> [**Check for Update**] リンクにアクセスできるのは、HTTPS プロトコルではなく、HTTP プロトコルを使用して GUI にログインした場合だけです。

関連リソース

次のリソースは、Citrix ADC アプライアンスのアップグレードまたはダウングレードに関する関連情報を提供します。

- ビデオチュートリアル- CLI を使用して Citrix ADC をアップグレードする方法

Citrix ADC スタンドアロンアプライアンスのダウングレード

October 7, 2021

CLI または GUI を使用して、スタンドアロン Citrix ADC の以前のリリースにダウングレードできます。

> 注:
>
> ダウングレード時に設定が失われることがあります。ダウングレードの前後の構成を比較してから、不足しているエントリを手動で再入力します。

CLI を使用して Citrix ADC アプライアンスをダウングレードする

以下の手順に従って、リリース 13.0 を実行する Citrix ADC スタンドアロンアプライアンスを以前のリリースにダウングレードします。

この手順では、<release> と<releasenumber> はダウングレードするリリースバージョンを表し、<targetbuildnumber>はダウングレードするビルド番号を表します。

1. PuTTY などの SSH クライアントを使用して、Citrix ADC への SSH 接続を開きます。
2. 管理者の資格情報を使用して、Citrix ADC にログオンします。構成実行を保存します。プロンプトで、次のように入力します。

   save config

3. ns.conf ファイルのコピーを作成します。シェルプロンプトで、次のように入力します。

   a) `cd /nsconfig`
   b) `cp ns.conf ns.conf.NS<currentbuildnumber>`

   構成ファイルのコピーを別のコンピュータにバックアップする必要があります。

4. <releasenumber> 構成ファイル（ns.conf.NS <releasenumber>）を ns.conf にコピーします。シェルプロンプトで、次のように入力します。

```
cp ns.conf.NS<releasenumber> ns.conf
<!--NeedCopy-->
```

> 注:
>
> `ns.conf.NS<releasenumber>` は、システムソフトウェアをリリースバージョンから現在のリリースバージョン`<releasenumber>` にアップグレードしたときに自動的に作成されるバックアップ構成ファイルです。
>
> ダウングレード時に設定が失われる可能性があります。アプライアンスの再起動後、ステップ 3 で保存した設定を構成実行と比較し、ダウングレード前に設定された機能とエンティティを調整します。変更を行った後、構成実行構成を保存します。
>
> 重要:
>
> ルーティングが有効になっている場合は、手順 5 を実行します。それ以外の場合は、手順 6 に進みます。

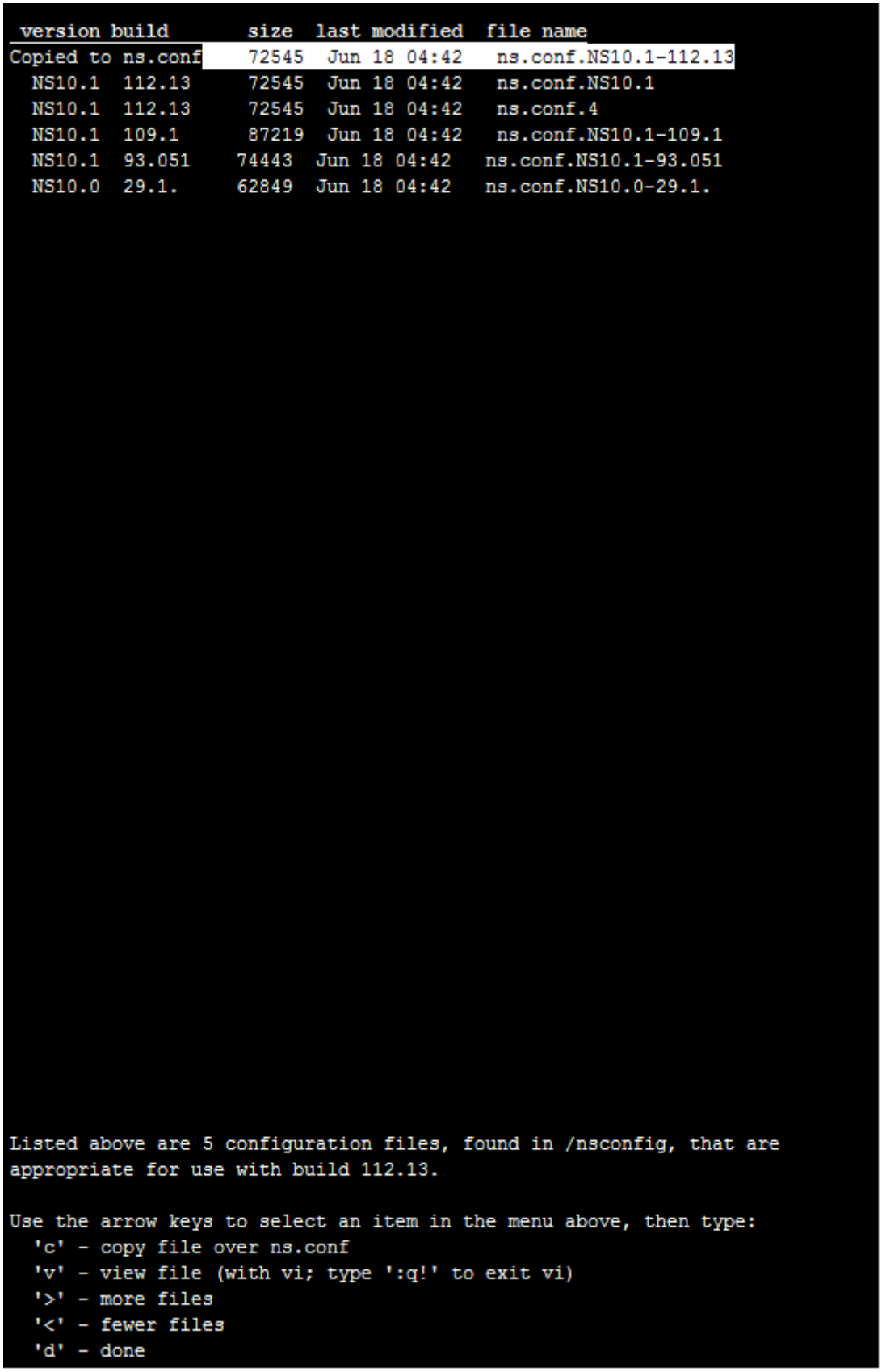
5. ルーティングが有効になっている場合、ZebOS.conf ファイルに構成が含まれます。シェルプロンプトで、次のように入力します。

```
cd /nsconfig
cp ZebOS.conf ZebOS.conf.NS
cp ZebOS.conf.NS<targetreleasenumber> ZebOS.conf
<!--NeedCopy-->
```

6. ディレクトリを/var/nsinstall/<releasenumber>nsinstallに変更するか、存在しない場合は作成します。
7. ディレクトリをbuild_<targetbuildnumber>に変更するか、存在しない場合は作成します。
8. インストールパッケージ（build-<release>-<targetbuildnumber>.tgz）をこのディレクトリにダウンロードまたはコピーして、インストールパッケージの内容を抽出します。
9. `installns` スクリプトを実行して、新しいバージョンのシステムソフトウェアをインストールします。スクリプトは `/etc` ディレクトリを更新します。

ダウングレードするビルドの構成ファイルがアプライアンスに存在する場合は、その構成をロードするように求められます。

図 1：設定ファイルが存在する場合はメニューをダウングレード



フラッシュドライブの空き容量が不足して新しいビルドをインストールできない場合、Citrix ADC はインストールを中止します。手動でフラッシュドライブをクリーンアップし、インストールを再起動します。

例:

```
login: nsroot

Password: nsroot

Last login: Mon Apr 24 02:06:52 2017 from 10.102.29.9

Done

> save config

> shell

root@NSnnn# cp ns.conf.NS10.5 ns.conf

root@NSnnn# cd /var/nsinstall

root@NSnnn# mkdir 10.5nsinstall

root@NSnnn# cd 10.5nsinstall

root@NSnnn# mkdir build_57

root@NSnnn# cd build_57

root@NSnnn# ftp 10.102.1.1

ftp> mget build-10.5-57_nc.tgz

ftp> bye

root@NSnnn# tar -xzvf build-10.1-125_nc.tgz

root@NSnnn# ./installns

installns version (10.5-57) kernel (ns-10.5-57.gz)

...

...
```



```
40
41 ...
42
43 Copying ns-10.5-57.gz to /flash/ns-10.5-57_nc.gz ...
44
45 Changing /flash/boot/loader.conf for ns-10.5-57 ...
46
47
48
49 Installation has completed.
50
51
52
53 Reboot NOW? [Y/N] Y
54 <!--NeedCopy-->
```

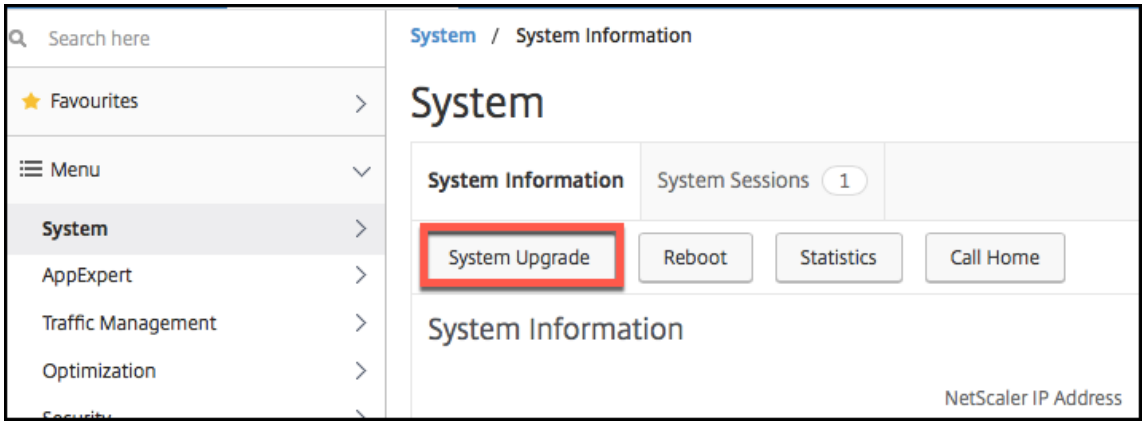
GUI を使用して Citrix ADC アプライアンスをダウングレードする

GUI のアップグレードウィザードを使用して、リリース 13.0 を実行している Citrix ADC アプライアンスを以前のリリースにダウングレードできます。

> メモ:
>
> GUI を使用して、リリース 13.0 を実行している Citrix ADC アプライアンスをリリース 10.5 以前に直接ダウングレードすることはできません。ダウングレードには CLI の使用をお勧めします。
>
> Citrix ADC のリリースライフサイクルの詳細については、製品マトリックスサイトを参照してください 。
>
> 一度に 1 つのメジャーリリースにダウングレードすることをお勧めします。
>
> たとえば、Citrix ADC アプライアンスがリリース 13.0 であり、リリース 12.0 にダウングレードする場合は、アプライアンスを最初にリリース 12.1 にダウングレードしてから、リリース 12.0 にダウングレードする必要があります。

GUI を使用して、リリース 13.0 を実行している Citrix ADC アプライアンスを以前のリリースにダウングレードするには、以下の手順に従います。

1. Web ブラウザで、Citrix ADC IP アドレスを入力します（例: `http://10.102.29.50`）。
2. [ユーザー名とパスワード] に管理者の資格情報を入力し、[ログオン] をクリックします。
3. GUI で、[システムのアップグレード] をクリックします。



4. 「ファイルの選択」メニューから、適切なオプション（「ローカル」または「アプライアンス」）を選択します。アプライアンスオプションを使用する場合は、最初にファームウェアを Citrix ADC にアップロードする必要があります。WinSCP などの任意のファイル転送方法を使用して、Citrix ADC ファームウェアをアプライアンスにアップロードできます。
5. 正しいファイルを選択し、[アップグレード] をクリックします。
6. 指示に従ってソフトウェアをダウングレードします。
7. プロンプトが表示されたら、[再起動] を選択します。

ダウングレード後、アプライアンスにアクセスする前に、すべてのブラウザーインスタンスを閉じ、コンピューターのキャッシュをクリアします。

関連リソース

次のリソースは、Citrix ADC アプライアンスのアップグレードまたはダウングレードに関する関連情報を提供します。

- ビデオチュートリアル- CLI を使用して Citrix ADC をアップグレードする方法

高可用性ペアのアップグレード

October 7, 2021

高可用性セットアップにおける Citrix ADC アプライアンスの要件の 1 つは、セットアップの両方のアプライアンスに同じ Citrix ADC ソフトウェアリリースをインストールすることです。したがって、あるアプライアンス上のソフトウェアをアップグレードする場合は、両方のアプライアンスでソフトウェアがアップグレードされていることを確認してください。

スタンドアロンアプライアンスまたは高可用性ペアの各アプライアンスを同じ手順でアップグレードできますが、高可用性ペアのアップグレードには追加の考慮事項が適用されます。

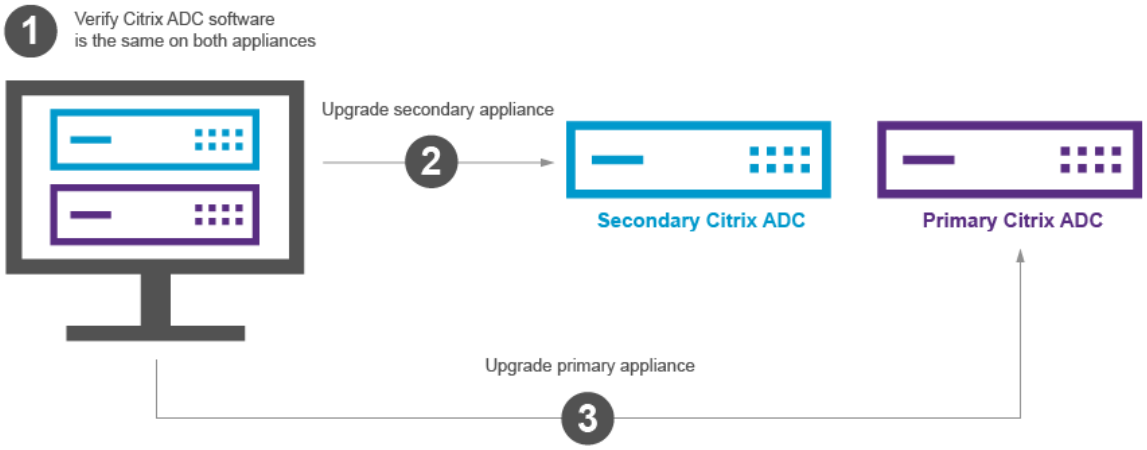
高可用性ペアで Citrix ADC ファームウェアのアップグレードを開始する前に、「開始する前に 」セクションに記載されている前提条件をお読みください。また、いくつかの HA 固有の点を考慮する必要があります。

注意事項

- 最初にセカンダリノードをアップグレードしてから、プライマリノードをアップグレードします。プライマリアプライアンスの前にセカンダリアプライアンスでソフトウェアをアップグレードすると、アップグレード処理が問題なく完了します。
- 高可用性（HA）セットアップの両方のノードが異なる Citrix ADC ソフトウェアリリースを実行している場合、次の機能は無効になります。
  - HA 構成の同期
  - HA コマンドの伝播
  - 状態サービス情報の HA 同期
  - セッションの接続ミラーリング（接続フェイルオーバー）
  - 永続セッション情報の HA 同期
- 高可用性（HA）セットアップの両方のノードが同じリリースの異なるビルドを実行しているが、両方のビルドの内部 HA バージョンが異なる場合、上記の機能は無効になります。上記の機能は、高可用性（HA）セットアップの両方のノードが同じリリースの異なるビルドを実行しているが、両方のビルドの内部 HA バージョンが同じである場合に正常に機能します。

  Citrix ADC ビルドで内部 HA バージョンが変更されているかどうかを確認するには、リリースノートの「要点」セクションを参照してください。
- HA 構成の 2 つのノードで異なる Citrix ADC ソフトウェアリリースが実行されている場合、または 2 つのノードで同じリリースの異なるビルドが実行されている場合、[HA files] コマンドの [All] モードでのファイルの同期は正常に実行されます。詳細については、「高可用性セットアップでの構成ファイルの同期」を参照してください。

図。高可用性ペアのアップグレード



Citrix ADC CLI または GUI を使用してアップグレードできます。

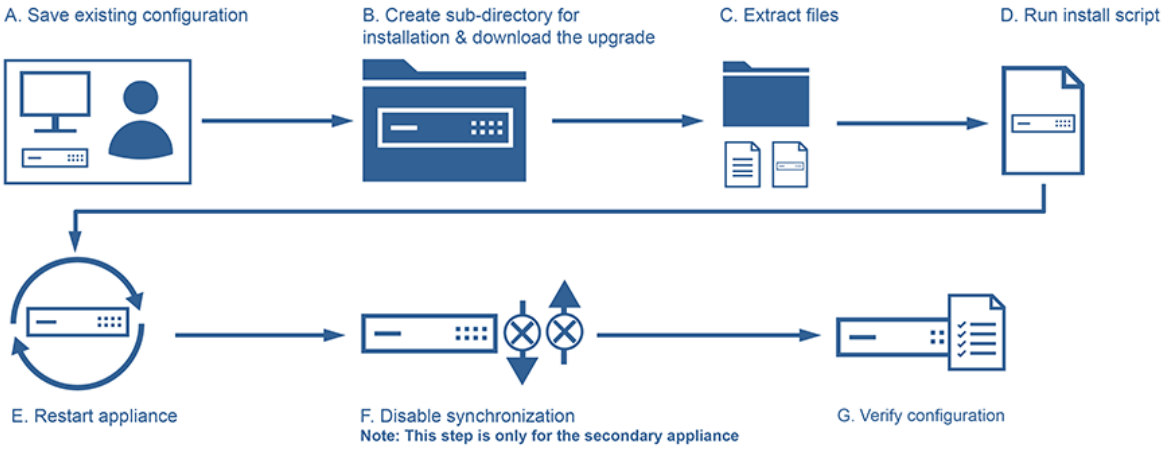
CLI を使用した高可用性ペアのアップグレード

アップグレードプロセスには、次の手順が含まれます。

1. セカンダリアプライアンス上のソフトウェアのアップグレード
2. プライマリアプライアンス上のソフトウェアのアップグレード
3. セカンダリアプライアンスの同期

セカンダリアプライアンス上のソフトウェアのアップグレード

次の図は、セカンダリアプライアンスでソフトウェアをアップグレードする手順を示しています。



1. PuTTY などの SSH ユーティリティを使用してセカンダリ NetScaler アプライアンスにログオンし、NetScaler IP（NSIP）を指定します。nsroot 認証情報を使用してアプライアンスにログオンします。
2. アプライアンスのコマンドラインインターフェイスから、次のコマンドを入力して既存の設定を保存します。
   save config
3. シェルプロンプトに切り替えます。

```
login as: username
Using keyboard-interactive authentication.
Password:
Last login: Wed Jun 24 14:59:16 2015 from 10.252.252.65
Done
> shell
Copyright (c) 1992-20

<!--NeedCopy-->
```


4. 次のコマンドを実行して、デフォルトのインストール・ディレクトリに変更します。# cd /var/nsinstall
5. 次のコマンドを実行して、nsinstall ディレクトリの一時サブディレクトリを作成します。**# mkdir x_xnsinstall**

   > 注：x_x というテキストは、将来の構成のために NetScaler バージョンに名前を付けるために使用されます。たとえば、NetScaler 9.3 のインストールファイルのディレクトリは、9_3nsinstall と呼ばれます。フォルダ名にピリオド（.）を使用しないでください。アップグレードが失敗する可能性があります。

6. **x_xnsinstall** ディレクトリに移動します。
7. 必要なインストールパッケージとドキュメント・バンドル（「ns-x.0-xx.x-doc.tgz」など）を、手順 4 で作成した一時ディレクトリにダウンロードします。

   > 注:
   >
   > 一部のビルドには、インストールする必要がないため、ドキュメントバンドルがありません。
   >
   > GUI の [ドキュメント] タブをクリックして、ドキュメントにアクセスします。

8. インストールスクリプトを実行する前に、ファイルを抽出してアプライアンスに配置する必要があります。Citrix Web サイトからダウンロードしたバンドルを解凍するには、次のコマンドを使用します。**tar-zxvf ns-x.0-xx.x-doc.tgz**。以下に、使用するパラメーターの簡単な説明を示します。

   x: ファイルの抽出

   v: ファイル名を 1 つずつ抽出して出力します。

   z: ファイルは「gzip」ファイルです。

   f: 操作に次の tar アーカイブを使用します。

9. 次のコマンドを実行して、ダウンロードしたソフトウェアをインストールします。#. /インストール

   > 注：アプライアンスに新しいカーネルファイルをインストールするための十分なディスク容量がない場合、インストールプロセスによってフラッシュドライブの自動クリーンアップが実行されます。

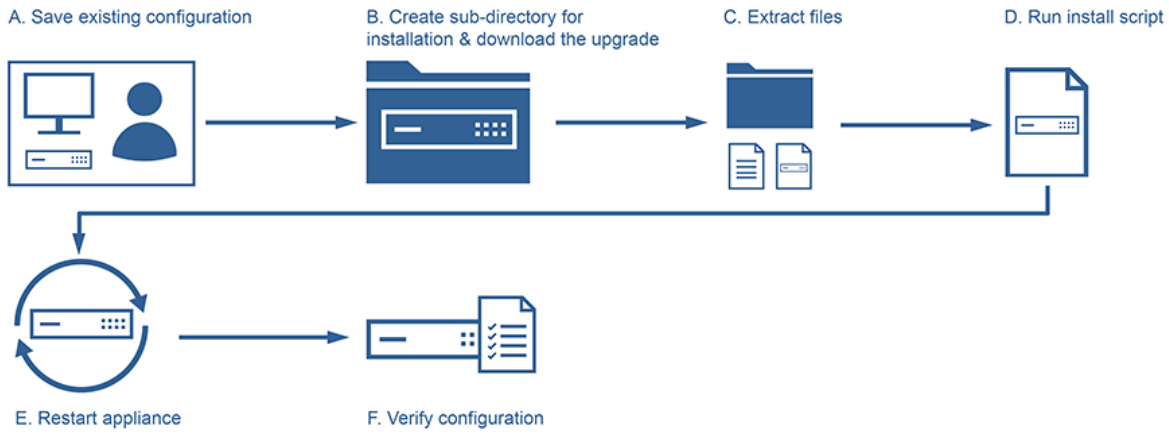
10. インストールプロセスが完了すると、アプライアンスの再起動を求めるメッセージが表示されます。y を押してアプライアンスを再起動します。
11. nsroot 認証情報を使用して、アプライアンスのコマンドライン・インターフェイスにログオンします。
12. から次のコマンドを実行して、NetScaler アプライアンスの状態を表示します。**show ha node** 前述のコマンドの出力は、アプライアンスがセカンダリノードであり、同期が無効になっていることを示しています。
13. 次のコマンドを実行して、プライマリアプライアンスとして強制フェールオーバーとテイクオーバーを実行します。強制フェールオーバー

次に、新しいプライマリノードの設定例を示します。


```
login: nsroot
Password: nsroot
Last login: Monday Apr  17 08:37:26 2017 from 10.102.29.9
Done
show ha node
        2 nodes:
1)      Node ID:      0
        IP:        10.0.4.2
        Node State: UP
        Master State: Primary
        ...
        Sync State: AUTO DISABLED
        Propagation: AUTO DISABLED
        ...
Done
<!--NeedCopy-->
```

プライマリアプライアンス上のソフトウェアのアップグレード

次の図は、プライマリアプライアンスでソフトウェアをアップグレードする手順を示しています。



> 注:「セカンダリアプライアンスでのソフトウェアのアップグレード」の手順を完了すると、元のプライマリアプライアンスはセカンダリアプライアンスになります。

1. PuTTY などの SSH ユーティリティを使用して、セカンダリ NetScaler アプライアンスにログオンします。nsroot 認証情報を使用してアプライアンスにログオンします。上記のセクションで説明した手順と同じ手順を実行して、インストールプロセスを完了します。前のセクション（セカンダリアプライアンスのソフトウェアのアップグレード）のステップ 2 ～9 で述べた手順と同じ手順を実行する必要があります。
2. インストールプロセスが完了すると、アプライアンスの再起動を求めるメッセージが表示されます。y を押してアプライアンスを再起動します。

3. nsroot 認証情報を使用して、アプライアンスのコマンドライン・インターフェイスにログオンします。
4. 次のコマンドを実行して、NetScaler アプライアンスの状態を表示します。**show ha node**。上記のコマンドの出力は、アプライアンスがセカンダリノードであり、ノード状態のステータスが UP としてマークされていることを示しているはずです。
5. 次のコマンドを実行して、強制フェイルオーバーを実行し、アプライアンスがプライマリアプライアンスであることを確認します。強制フェイルオーバー
6. アプライアンスがプライマリアプライアンスであることを確認します。

   次に、新しいプライマリノードと新しいセカンダリノードの設定例を示します。

```
show ha node
    Node ID:      0
    IP:   10.0.4.11
    Node State: UP
    Master State: Primary
    ...
    ...
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    ...
    ...
    Local node information
    Critical Interfaces: 1/1
Done

Show ha node
    Node ID:      0
    IP:   10.0.4.2
    Node State: UP
    Master State: Secondary
    ..
    ..
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
```



```
33      . .
34      . .
35      Local node information:
36      Critical Interfaces: 1/1
37  Done
38  <!--NeedCopy-->
```

GUI を使用した高可用性ペアのアップグレード

次の手順に従って、ADC GUI を使用して高可用性セットアップで Citrix ADC ペアをアップグレードします。Citrix ADC アプライアンス CITRIX-ADC-A（プライマリ）および CITRIX-ADC-B（セカンダリ）の高可用性セットアップの例を検討してください。

1. セカンダリノードをアップグレードします。管理者資格情報を使用してセカンダリノード GUI にログオンし、「GUI を使用した Citrix ADC スタンドアロンアプライアンスのアップグレード」の説明に従ってアップグレードを実行します。
2. 強制フェイルオーバー。「ノードのフェイルオーバーの強制実行」の説明に従って、GUI を使用してセカンダリノードで強制フェールオーバーを実行します。

   フェイルオーバー操作後、セカンダリノードがプライマリとして引き継ぎ、プライマリノードが新しいセカンダリノードになります。強制フェイルオーバー。

   - CITRIX-ADC-B が新しいプライマリになります
   - CITRIX-ADC-A が新しいセカンダリになります
3. 元のプライマリノード（新しいセカンダリノード）をアップグレードします。新しいセカンダリノード GUI（CITRIX-ADC-A）にログオンし、GUI を使用した Citrix ADC スタンドアロンアプライアンスのアップグレードの説明に従ってアップグレードを実行します。
4. 強制フェイルオーバー。「ノードのフェイルオーバーの強制実行」の説明に従って、GUI を使用して新しいセカンダリノード（CITRIX-ADC-A）で強制フェールオーバーを実行します。

   この 2 回目のフェイルオーバー操作の後、両方のノードの状態は、HA アップグレード操作を開始する前と同じ状態に戻ります。強制フェイルオーバー。

   - CITRIX-ADC-A がプライマリになります
   - CITRIX-ADC-B がセカンダリになります
5. アップグレードプロセスを確認します。両方のノードの GUI にログオンします。**System > High Availability** に移動して詳細ページで両方のノードの HA 状態を確認します。また、GUI の上部ペインに表示されるアップグレードされたリリースの詳細を確認します。

関連リソース

次のリソースは、Citrix ADC の高可用性セットアップのアップグレードに関する関連情報を提供します。

- ビデオチュートリアル- GUI を使用して Citrix ADC HA ペアをアップグレードする方法

ダウンタイムゼロのアップグレードを実行するための高可用性を実現するサービス中ソフトウェアアップグレードのサポート

October 7, 2021

高可用性セットアップの定期的なアップグレードプロセスでは、ある時点で、両方のノードが異なるソフトウェアビルドを実行します。これら 2 つのビルドは、同じまたは異なる内部高可用性バージョン番号を持つことができます。

両方のビルドで高可用性バージョン番号が異なる場合、既存のデータ接続に対する接続のフェールオーバー (有効になっていても) はサポートされません。つまり、既存のデータ接続はすべて失われ、ダウンタイムにつながります。

この問題に対処するために、サービスソフトウェアアップグレード（ISSU）で高可用性セットアップを使用できます。ISSU では、アップグレードプロセスの強制フェールオーバー操作手順に代わる移行機能が導入されています。移行機能では、既存の接続を尊重し、強制フェールオーバー操作が含まれます。

移行操作が実行されると、新しいプライマリノードは常に既存の接続に関連するトラフィック（要求と応答）を受信しますが、それらを古いプライマリノードに誘導します。古いプライマリノードはデータトラフィックを処理し、送信先に直接送信します。

拡張 **ISSU** のしくみ

高可用性セットアップでの通常のアップグレードプロセスは、次の順次ステップで構成されます。

1. セカンダリノードをアップグレードします。この手順には、セカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。
2. フェイルオーバーを強制します。強制フェールオーバーを実行すると、アップグレードしたセカンダリノードがプライマリノードになり、プライマリノードがセカンダリノードになります。
3. 新しいセカンダリノードをアップグレードします。この手順には、新しいセカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。

手順 1 から手順 3 までの時間枠では、両方のノードが異なるソフトウェアビルドを実行します。これら 2 つのビルドは、同じまたは異なる内部高可用性バージョンを持つことができます。

両方のビルドで高可用性バージョン番号が異なる場合、既存のデータ接続に対する接続のフェールオーバー (有効になっていても) はサポートされません。つまり、既存のデータ接続はすべて失われ、ダウンタイムにつながります。

高可用性セットアップの ISSU アップグレードプロセスは、次の手順で構成されます。

1. セカンダリノードをアップグレードします。この手順には、セカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。

2. **ISSU** 移行オペレーション。この手順には、強制フェールオーバー操作が含まれており、既存の接続を処理します。移行操作を実行すると、新しいプライマリノードは常に既存の接続に関連するトラフィック（要求と応答）を受信しますが、GRE トンネルで設定された SYNC VLAN を介して古いプライマリノードに誘導します。古いプライマリノードはデータトラフィックを処理し、送信先に直接送信します。ISSU 移行操作は、既存のすべての接続が閉じられると完了します。
3. 新しいセカンダリノードをアップグレードします。この手順には、新しいセカンダリノードのソフトウェアアップグレードとノードの再起動が含まれます。

はじめに

高可用性セットアップで ISSU プロセスの実行を開始する前に、次の前提条件と制限事項に従ってください。

- 高可用性セットアップの両方のノードで SYNC VLAN が設定されていることを確認します。詳細については、「VLAN へのハイアベイラビリティ同期トラフィックの制限」を参照してください。
- Microsoft Azure は GRE トンネリングをサポートしていないため、ISSU は Microsoft Azure クラウドではサポートされていません。
- 高可用性構成の伝播および同期は、ISSU では機能しません。
- ISSU は、IPv6 高可用性セットアップではサポートされません。
- ISSU は、次のセッションではサポートされません。
  - ジャンボフレーム
  - IPv6 セッション
  - 大規模 NAT (LSN)

構成の手順

ISSU には、高可用性セットアップの通常のアップグレードプロセスにおける強制フェールオーバー動作に代わる移行機能が含まれています。移行機能では、既存の接続を尊重し、強制フェールオーバー操作が含まれます。

高可用性セットアップの ISSU プロセス中に、セカンダリノードをアップグレードした直後に移行操作を実行します。移行操作は、2 つのノードのいずれかから実行できます。

CLI のプロシージャ

CLI を使用して高可用性移行操作を実行するには、次の手順を実行します。

コマンドプロンプトで次のように入力します。

```
start ns migration
<!--NeedCopy-->
```


GUI の手順

GUI を使用して高可用性移行操作を実行するには、次の手順を実行します。

[システム] に移動し、[システム情報] タブ、[移行] タブ、[移行の開始] の順にクリックします。

ISSU 統計を表示する

高可用性セットアップで現在の ISSU プロセスを監視するための ISSU 統計を表示できます。ISSU 統計には、次の情報が表示されます。

- ISSU 移行操作の現在のステータス
- ISSU 移行操作の開始時刻
- ISSU 移行操作の終了時間
- ISSU ロールバック操作の開始時刻

CLI または GUI を使用して、いずれかの HA ノードの ISSU 統計を表示できます。

CLI のプロシージャ

CLI を使用して ISSU 統計を表示するには、次のようにします。

コマンドプロンプトで次のように入力します。

```
1 show ns migration
2 <!--NeedCopy-->
```

GUI の手順

GUI を使用して ISSU 統計を表示するには:

システムに移動し 、[システム情報] タブをクリックし、[移行] タブをクリックし、[表示の移行をクリックしてください。

ISSU プロセスのロールバック

高可用性 (HA) セットアップでは、インサービスソフトウェアアップグレード (ISSU) プロセスのロールバックがサポートされるようになりました。ISSU ロールバック機能は、ISSU 移行動作中の HA セットアップが安定していないか、期待どおりに最適レベルで実行されていないことを観察する場合に役立ちます。

ISSU ロールバックは、ISSU 移行操作が進行中の場合に適用されます。ISSU 移行処理がすでに完了している場合、ISSU ロールバックは機能しません。つまり、ISSU 移行動作が進行中の場合は、ISSU ロールバック動作を実行する必要があります。

ISSU ロールバック動作がトリガーされたときの ISSU 移行動作の状態に応じて、ISSU ロールバックの動作が異なります。

- **ISSU** 移行操作中に、強制フェールオーバーがまだ行われていません。ISSU ロールバックによって ISSU 移行動作が停止され、両方のノードに格納されている ISSU 移行に関連する内部データが削除されます。現在のプライマリノードはプライマリノードとして残り、既存の接続と新しい接続に関連するデータトラフィックを処理し続けます。
- **ISSU** 移行操作中に強制フェールオーバーが発生しました。ISSU 移行操作中に高可用性フェールオーバーが発生した場合、新しいプライマリノード（N1 など）は、新しい接続に関連するトラフィックを処理します。古いプライマリノード（新しいセカンダリノード、N2 など）は、古い接続（ISSU 移行操作前の既存の接続）に関連するトラフィックを処理します。

  ISSU ロールバックは、ISSU 移行動作を停止し、強制フェールオーバーをトリガーします。新しいプライマリノード（N2）は、新しい接続に関連するトラフィックの処理を開始します。新しいプライマリノード（N2）は、古い接続（ISSU 移行操作の前に確立された既存の接続）に関連するトラフィックも引き続き処理します。つまり、ISSU 移行操作の前に確立された既存の接続は失われません。

  新しいセカンダリノード（N1）は、既存の接続（ISSU 移行操作中に作成された新しい接続）をすべて削除し、トラフィックを処理しません。つまり、ISSU 移行操作の強制フェールオーバー後に確立された既存の接続は、永久に失われます。

構成の手順

Citrix ADC CLI または GUI を使用して、ISSU のロールバック操作を実行できます。

CLI のプロシージャ

CLI を使用して ISSU ロールバック操作を実行するには、次の手順を実行します。

コマンドプロンプトで次のように入力します。

```
stop ns migration
<!--NeedCopy-->
```

GUI の手順

GUI を使用して ISSU ロールバック操作を実行するには、次の手順を実行します。

[システム] に移動し、[システム情報] タブ、[移行] タブ、[移行の停止] の順にクリックします。

インサービスソフトウェアアップグレードプロセスの **SNMP** トラップ

高可用性セットアップの In Service Software Upgrade（ISSU; インサービスソフトウェアアップグレード）プロセスは、ISSU 移行操作の開始および終了時に次の SNMP トラップメッセージをサポートします。

SNMP トラップ	説明
移行を開始しました	この SNMP トラップは、ISSU 移行動作の開始時に生成され、設定された SNMP トラップリスナーに送信されます。
移行完了	この SNMP トラップは、ISSU 移行操作が完了すると、設定された SNMP トラップリスナーに生成され、送信されます。

プライマリノード（ISSU プロセスの開始前）は、常にこれらの 2 つの SNMP トラップを生成し、設定済みの SNMP トラップリスナーに送信します。

ISSU SNMP トラップに関連付けられた SNMP アラームはありません。つまり、これらのトラップは、SNMP アラームに関係なく生成されます。必要なのは、トラップ SNMP リスナーの設定だけです。

SNMP トラップロスナの構成の詳細については、Citrix ADC での SNMP トラップを参照してください。

高可用性ペアのダウングレード

October 7, 2021

コマンドラインインターフェイスを使用して、高可用性ペアの任意のリリースにダウングレードできます。GUI は、ダウングレードプロセスをサポートしていません。

高可用性ペアの Citrix ADC ペアのシステムソフトウェアをダウングレードするには、まずセカンダリノードでソフトウェアをダウングレードし、次にプライマリノードでダウングレードする必要があります。各ノードを個別にダウングレードする方法については、Citrix ADC スタンドアロンアプライアンスのダウングレードを参照してください。

> 重要
>
> ダウングレード時に設定が失われることがあります。ダウングレード前とダウングレード後の設定を比較し、不足しているエントリを手動で再入力する必要があります。

インストール、アップグレード、ダウングレードプロセスに関連する問題のトラブルシューティング

January 31, 2022

インストール、アップグレード、またはダウングレード処理の完了後にアプライアンスが期待どおりに動作しない場合は、まず問題の最も一般的な原因をチェックします。

トラブルシューティングのリソース

最適な結果を得るには、Citrix ADC インストール、アップグレード、ダウングレードに関連する問題のトラブルシューティングを行うために、次のリソースを使用します。

- アプライアンスからの設定ファイル。高可用性ペアの場合、両方のアプライアンスからの構成ファイル。
- アプライアンスの次のファイル:
  - 関連する newnslog ファイル。
  - ns.log ファイル。
  - メッセージファイル。
- ネットワークトポロジダイアグラム。

問題と解決策

次に、インストール、アップグレード、ダウングレードに関する最も一般的な問題、およびそれらを解決するためのヒントを示します。

1. 問題

   ハードウェアとソフトウェアの非互換性が原因で、Citrix ADC MPX アプライアンスのアップグレードが失敗します。

   解像度

   Citrix ADC MPX ハードウェアとソフトウェアの互換性マトリクスを参照し 、ソフトウェアリリースが Citrix ADC MPX ハードウェアでサポートされているかどうかを確認します。

2. 問題

   Citrix ADC VPX アプライアンスとハイパーバイザーの非互換性が原因で、Citrix ADC VPX アプライアンスのアップグレードが失敗します。

   解像度

   Citrix ADC VPX アプライアンスおよびハイパーバイザーの互換性マトリクスを参照し 、Citrix ADC VPX アプライアンスモデルがハイパーバイザーでサポートされているかどうかを確認します。

3. 問題

   ハードウェアとソフトウェアの非互換性が原因で、Citrix ADC MPX アプライアンスのアップグレードが失敗します。

   解像度

   Citrix ADC アプライアンスの整合性を検証します。Citrix ADC ハードウェアアプライアンスをお持ちの場合は、`fsck`を実行してディスクチェックを実行し、Citrix ADC ハードディスクの整合性を検証することをお勧めします。

   詳細については、「Citrix ADC アプライアンスのファイルシステムの整合性を検証する方法」を参照してください。

4. 問題

   GUI ストールを使用した Citrix ADC アプライアンスのアップグレード。

   解像度

   ブラウザを更新して、アップグレードが進行中であるかどうかを確認します。

5. 問題

   /var ディレクトリのスペースが不足しているため、Citrix ADC アプライアンスのアップグレードが失敗する

   解像度

   /var ディレクトリ上のスペースを解放します。詳細については、「/var ディレクトリの空き領域を増やす方法」を参照してください。

6. 問題

   ソフトウェアのダウングレード後、Citrix ADC にアクセスできない

   原因

   ソフトウェアのダウングレード処理中に、既存のリリースとビルドの設定ファイルが以前のリリースおよびビルドの設定ファイルと一致しない場合、アプライアンスは設定をロードできず、デフォルトの IP アドレスがアプライアンスに割り当てられます。

   解像度

   - コンソールからアプライアンスにアクセスできることを確認します。
   - アプライアンスの NSIP アドレスとルートを確認します。
     - IP アドレスがデフォルトの 192.168.100.1 IP アドレスに変更されている場合は、必要に応じて IP アドレスを変更します。
     - アプライアンスがアクセス可能であることを確認します。

7. 問題

   アップグレード中に、同期用のコマンドを実行すると、次のメッセージが表示されます。

コマンドはセカンダリノードで失敗しましたが、プライマリノードで成功しました。

解像度

高可用性 (HA) 同期の実行中は、依存コマンド (set /unset /bind /unbind) を実行しないでください。

8. 問題

アップグレードプロセス中に、force failover コマンドを実行しても、トラフィックは新しいプライマリノードを通過しません。

解像度

- ネットワークトポロジとスイッチの設定に関する問題をチェックします。
- set L2param-garpreply ENABLED コマンドを実行して、GARP 応答を有効にします。
- まだ使用されていない場合は、仮想 MAC を使用してみてください。
- プライマリノードから sendarp —a コマンドを実行します。

9. 問題

Citrix ADC アプライアンスをアップグレードまたはダウングレードした後、SSH を介したアプライアンスへの接続が失敗します。

解像度

Citrix ADC アプライアンスで次の操作を実行します。

- /nsconfig/ssh/ssh_host_*で古いまたは安全でないホストキーを削除します。
- /nsconfig/sshd_config でカスタム SSHD 構成を確認し、それがまだ関連性と互換性があるかどうかを確認します。Google NMT それに応じて、カスタム SSHD 構成の名前を変更するか削除します。
- Citrix ADC アプライアンスをコールドリブートします

10. 問題

HA ペアでは、force HA フェールオーバーコマンドを実行した後も、デバイスはリブートを続けます。アップグレード後にセカンダリデバイスが起動しない。

解像度

/var ディレクトリがいっぱいかどうかを確認します。その場合は、古いインストールファイルを削除します。df —h コマンドを実行して、使用可能なディスク領域を表示します。

11. 問題

HA ペアをアップグレードすると、ノードの 1 つが UNKNOWN 状態として表示されます。

解像度

- 両方のノードが同じビルドを実行しているかどうかを確認します。ビルドが同じではなく、HA ノードにバージョンが一致しない場合、show ha node コマンドを実行すると、一部のフィールドが UNKNOWN と表示されます。

- セカンダリアプライアンスが到達可能かどうかを確認します。

12. 問題

    Citrix ADC をアップグレードすると、負荷分散仮想サーバーとサービスのほとんどがダウンしていることがインターフェイスに表示されます。

    解像度

    セカンダリアプライアンスで SNIP アドレスがアクティブであることを確認します。また、show service コマンドを入力して、サービスが実行されているかどうかを確認します。

13. 問題

    アップグレードを実行すると、セカンダリアプライアンス上のすべての仮想サーバがダウンします。

    解像度

    次のコマンドを実行して、HA 状態と HA 同期を有効にします。

    - ノードのハーステートの有効化を設定する
    - ノードの非同期有効化を設定する

    HA を無効にすることは推奨されません。

14. 問題

    ダウングレードを実行した後、Citrix ADC が正常に起動しません。

    解像度

    正しいライセンスがインストールされているかどうかを確認します。

15. 問題

    HA ペアでは、アップグレードの実行後に一部の機能が同期化されません。

    解像度

    sync ha file misc コマンドを実行して、プライマリノードからセカンダリノードに設定ファイルを同期します。

16. 問題

    再起動中に、次のエラーメッセージが表示されます。

    ns.conf の 1 つ以上のコマンドが失敗したどうしたらいいですか?

    解像度

    ns.conf ファイルのコマンドが 255 バイトの制限を超えていないことを確認します。25 バイト制限に対して長すぎるポリシーを作成するコマンドでは、パターンセットを使用してポリシーを短縮できます。

    例:


```
add cs policy p11 -rule 'HTTP.REQ.URL.ENDSWITH_ANY("
    ctx_file_extensions")'
Done
<!--NeedCopy-->
```

ctx_file_extensions は、多数の拡張子をカバーするデフォルトのパッチセットです。デフォルトのパターンセットに加えて、ユーザー定義のパターンセットを作成できます。次のコマンドを実行して、パッチセットを追加します。

```
add patset <name>
<!--NeedCopy-->
```

注意: パッチセットは、リリース 9.3 以降でのみサポートされます。

17. 問題

    Citrix ADC VPX アプライアンスをアップグレードする場合、/var の空き容量を増やすように指示されます。どのファイルを削除すればよいですか?

    解像度

    /var/tmp/ ディレクトリから古いインストールファイルを削除します。また、/flash から不要なファイルを削除します。

18. 問題

    セカンダリアプライアンスで force HA フェールオーバーコマンドを実行しても、グラフィカルユーザーインターフェイス（GUI）に接続できません。

    解像度

    コマンドラインインターフェイスを使用してセカンダリアプライアンスにログオンし、set ns ip <IP>-gui enabled コマンドを実行して GUI へのアクセスを有効にします。

19. 問題

    アップグレードの実行後、Java アプレット (アップグレードウィザードまたはライセンスウィザード) をロードする GUI 上のリンクをクリックすると、次のエラーメッセージが表示されます。**GUI** のバージョンがカーネルのバージョンと一致しません。このインスタンスを閉じて、**Java** プラグインのキャッシュをクリアして再度開いてください。

    解像度

    - GUI を使用して Citrix ADC にログオンします。
    - ［Citrix ADC Gateway］>［グローバル設定］に移動します。
    - [設定] の [グローバル設定の変更] をクリックします。

- 詳細ペインの [クライアントエクスペリエンス] で、[UI テーマ] リストから [既定] を選択します。
- [OK] をクリックします。

20. 問題

    何らかの理由で Citrix ADC アプライアンスのアップグレードが失敗した場合、バックアップファイルを使用してアプライアンスを復元する方法は?

    解像度

    アップグレードが失敗した場合は、バックアップされたファイルを使用して、アプライアンスを以前のバージョンの Citrix ADC アプライアンスに復元します。詳細については、「Citrix ADC アプライアンスのバックアップと復元」を参照してください。

    Citrix ADC クラスターセットアップのバックアップと復元の詳細については、「クラスター設定のバックアップと復元」を参照してください。

21. 問題

    Citrix ADC アプライアンスのアップグレードに失敗した後にライセンスが失われた場合、どのように問題を解決するのですか?

    解像度

    ライセンスが不足している場合、またはライセンスを再割り当てする場合は、次のトピックの「ライセンスの概要」を参照してください。

> 注
>
> これらのトラブルシューティング手順は、複数のリリース間でソフトウェアをダウングレードするときの設定が失われる問題にも適用されます。
>
> その他の問題については、リリースノート、ナレッジセンターの記事、FAQ を参照してください。

よくある質問

October 7, 2021

Citrix ADC ファームウェアのアップグレードに関する質問への回答については、「インストール、アップグレード、ダウングレードに関するよくある質問 」を参照してください。

新規および非推奨のコマンド、パラメータ、**SNMP OID**

October 7, 2021

この項では、新しいコマンド、パラメータ、および SNMP OID を一覧表示します。

新しいコマンド

次の表に、リリース 13.0 の新しいコマンドを示します。

[コマンド] 領域	コマンド
認証、承認、監査	add aaa ssoprofile; rm aaa ssoprofile; show aaa ssoprofile; set aaa ssoprofile; add authentication citrixAuthAction; rm authentication citrixAuthAction; set authentication citrixAuthAction; unset authentication citrixAuthAction; show authentication citrixAuthAction; lock aaa user; set aaa otpparameter; show aaa otpparameter; add authentication emailAction; rm authentication emailAction; set authentication emailAction; show authentication emailAction; add authentication noAuthAction; rm authentication noAuthAction; set authentication noAuthAction; show authentication noAuthAction; add authentication captchaAction; rm authentication captchaAction; set authentication captchaAction; show authentication captchaAction; add authentication adfsProxyProfile; rm authentication adfsProxyProfile; set authentication adfsProxyProfile; show authentication adfsProxyProfile
AppFlow	bind appflow action; unbind appflow action
アプリケーションファイアウォール	stat rnat6 and stat MapBmr

[コマンド] 領域	コマンド
コンテンツ検査	add contentInspection profile; rm contentInspection profile; set contentInspection profile; show contentInspection profile; add contentInspection callout; rm contentInspection callout; set contentInspection callout; show contentInspection callout; count contentInspection callout; set contentInspection parameter; unset contentInspection parameter; show contentInspection parameter
LSN	add lsn appsattributes; rm lsn appsattributes; set lsn appsattributes; show lsn appsattributes
ネットワーク	add rnat; rename rnat; bind rnat; unbind rnat; rm rnat
SSL	add ssl caCertGroup; bind ssl caCertGroup; rm ssl caCertGroup; unbind ssl caCertGroup; show ssl caCertGroup
システム	enable system autorestorefeature ; rm system restorepoint; show system restorepoint; migrate ns; show ns timezone; disable system autorestorefeature; create system restorepoint
URL フィルタリング	add urlfiltering Categorization; clear urlfiltering Categorization; show urlfiltering Categorization
VPN	add vpn urlPolicy; rm vpn urlPolicy; set vpn urlPolicy; show vpn urlPolicy; stat vpn urlPolicy; rename vpn urlPolicy; add vpn urlAction; rm vpn urlAction; set vpn urlAction; show vpn urlAction; rename vpn urlAction

新しいパラメータ

コマンドグループ: 認証、承認、および監査

コマンド:

- set aaa parameter [-maxKBQuestions]
- unset aaa parameter [-maxKBQuestions]
- show aaa parameter [-maxKBQuestions]

コマンドグループ: 管理パーティション

コマンド:

- add ns ip6 [-advertiseOnDefaultPartition]
- set ns ip6 [-advertiseOnDefaultPartition]
- show ns ip6[-advertiseOnDefaultPartition]
- add ns ip [-advertiseOnDefaultPartition]
- set ns ip [-advertiseOnDefaultPartition]
- show ns ip[-advertiseOnDefaultPartition]

コマンドグループ: AppFlow

コマンド:

- bind appflow action [-analyticsProfile]
- unbind appflow action [-analyticsProfile]
- show appflow action [-analyticsProfile]
- set appflow param [-gxSessionReporting] [-usageRecordInterval] [-metrics] [-events [-auditlogs] [-observationPointId] [-distributedTracing] [-distTracingSamplingRate] [-tcpAttackCounterInterval]
- show appflow param [-observationPointId] [-subscriberIdObfuscationAlgo] [-gxSessionReporting] [-usageRecordInterval] [-metrics] [-events] [-auditlogs [-distributedTracing] [-distTracingSamplingRate] [-tcpAttackCounterInterval]

コマンドグループ: アプリケーションファイアウォール

コマンド:

- add appfw profile [-postBodyLimitSignature]
- add appfw profile [-rfcprofile]
- set appfw profile [-postBodyLimitSignature] [-rfcprofile]
- show appfw profile [-postBodyLimitSignature] [-rfcprofile]
- set appfw settings [-malformedReqAction]
- show appfw settings [-malformedReqAction]
- import appfw signatures [-preservedefactions]

コマンドグループ: 監査

コマンド:

- add audit syslogAction [-ContentInspectionLog]
- set audit syslogAction [-ContentInspectionLog]

- unset audit syslogAction [-ContentInspectionLog]
- show audit syslogAction [-ContentInspectionLog]
- add audit nslogAction [-ContentInspectionLog]
- set audit nslogAction [-ContentInspectionLog]
- unset audit nslogAction [-ContentInspectionLog]
- show audit nslogAction [-ContentInspectionLog]
- set audit syslogParams [-ContentInspectionLog]
- unset audit syslogParams [-ContentInspectionLog]
- show audit syslogParams [-ContentInspectionLog]
- set audit nslogParams [-ContentInspectionLog]
- unset audit nslogParams [-ContentInspectionLog]
- show audit nslogParams [-ContentInspectionLog]

コマンドグループ: 認証

コマンド:

- add authentication ldapAction [-KBAttribute] [-alternateEmailAttr]
- set authentication ldapAction [-KBAttribute] [-alternateEmailAttr]
- show authentication ldapAction [-KBAttribute] [-alternateEmailAttr]
- add authentication tacacsAction [-Attributes]
- set authentication tacacsAction [-Attributes]
- show authentication tacacsAction [-Attributes]
- add authentication samlAction [-Attributes] [-storeSAMLResponse]
- set authentication samlAction [-Attributes] [-storeSAMLResponse]
- show authentication samlAction [-Attributes] [-storeSAMLResponse]
- add authentication vserver [-certkeyNames]
- set authentication vserver [-certkeyNames]
- show authentication vserver [-certkeyNames]- show authentication loginSchema [-feature]- - add authentication OAuthIDPProfile [-configservice] [-signatureAlg] [-Attributes][-sendPassword]
- set authentication OAuthIDPProfile [-configservice] [-signatureAlg] [-Attributes] [-sendPassword]
- show authentication OAuthIDPProfile [-configservice] [-signatureAlg] [-Attributes] [-sendPassword]
- add authentication pushService [-CertEndpoint] [-signingKeyName] [-signingKey] [-clientID] [-clientSecret] [-CustomerID] [-refreshInterval] [-trustService]
- set authentication pushService [-CertEndpoint] [-signingKeyName] [-signingKey] [-clientID] [-clientSecret] [-CustomerID] [-refreshInterval] [-trustService]
- show authentication pushService [-clientID] [-clientSecret] [-CustomerID] [-CertEndpoint] [-refreshInterval] [-pushServiceStatus] [-trustService] [-pushCloudServerStatus] [-signingKeyName] [-signingKey]
- show authentication adfsProxyProfile [-adfsTrustStatus]

コマンドグループ: 基本

コマンド:

- add server [-queryType]
- show server [-queryType] [-serviceGroupEntName2] [-svcitmPriority] [-svcitmActSvcs] [-svcitmBoundSvcs] [-weight]
- add service [-contentInspectionProfileName]
- set service[-contentInspectionProfileName]
- unset service [-contentInspectionProfileName]- show service [-contentInspectionProfileName]-bind serviceGroup [-nameServer] [-dbsTTL]
- show serviceGroup [-numOfCurConnections] [-numOfLastConnections] [-nameServer] [-dbsTTL] [-svcitmActSvcs] [-svcitmPriority] [-svcitmBoundSvcs]

コマンドグループ: キャッシュリダイレクト

コマンド:

- add cr vserver [-UseOriginIpPortForCache]
- set cr vserver [-UseOriginIpPortForCache]
- show cr vserver [-UseOriginIpPortForCache]

コマンドグループ: クラスタリング

コマンド:

- show cluster instance [-heterogeneousFlag]

コマンドグループ: コンテンツの切り替え

コマンド:

- add cs vserver [-persistenceType] [-persistMask] [-v6persistmasklen] [-timeout] [-cookieName] [-persistenceBackup] [-backupPersistenceTimeout]
- set cs vserver [-persistenceType] [-persistMask] [-v6persistmasklen] [-timeout] [-cookieName] [-persistenceBackup] [-backupPersistenceTimeout]
- show cs vserver [-persistenceType] [-persistMask] [-v6persistmasklen] [-timeout] [-cookieName] [-persistenceBackup] [-backupPersistenceTimeout]

コマンドグループ: DNS

コマンド:

- set dns nameServer [-type]
- rm dns nameServer [-type]
- enable dns nameServer [-type]
- disable dns nameServer [-type]
- show dns action64
- set dns parameter [-maxUDPPacketSize]
- show dns parameter [-maxUDPPacketSize]

コマンドグループ: GSLB

コマンド:

- show gslb service [-gslbsvcHealth] [-glsbsvcHealthdescr]
- show gslb parameter [-builtin]

コマンドグループ: 高可用性

コマンド:

- show HA node [-haSyncFailureReason]

コマンドグループ: ICA

コマンド:

- set ica parameter [-HDXInsightNonNSAP]
- show ica parameter [-HDXInsightNonNSAP]

コマンドグループ: 負荷分散

コマンド:

- add lb vserver [-adfsProxyProfile]
- set lb vserver [-adfsProxyProfile]
- unset lb vserver [-adfsProxyProfile]
- show lb vserver [-adfsProxyProfile]
- set lb parameter [-dbsTTL]
- show lb parameter [-dbsTTL]

コマンドグループ: LSN

コマンド:

- add lsn logprofile [-analyticsProfile] [-logSessDeletion]
- set lsn logprofile [-analyticsProfile] [-logSessDeletion]
- show lsn logprofile [-analyticsProfile] [-logSessDeletion]
- bind lsn appsprofile [-appsattributesname]
- unbind lsn appsprofile [-appsattributesname]
- show lsn appsprofile [-appsattributesname]

コマンドグループ: Network

コマンド:

- add route [-vlan]
- rm route [-vlan]
- add netProfile [-proxyProtocol] [-proxyProtocoltxversion]
- set netProfile [-proxyProtocol] [-proxyProtocoltxversion]
- show netProfile [-proxyProtocol] [-proxyProtocoltxversion]

- set rnat [-name]
- unset rnat [-name]
- add rnat [-name]
- rename rnat [-name] [-newName]
- bind rnat [-name] [-natIP]
- unbind rnat [-name] [-natIP]
- rm rnat [-name]
- show rnat [-name] [-stateflag]

コマンドグループ: Policy

コマンド:

- show policy expression
- show policy patset
- show policy urlset [-imported]
- import policy urlset [-subdomainExactMatch]

コマンドグループ: RDP

コマンド:

- add rdp clientprofile [-randomizeRDPFilename] [-rdpLinkAttribute]
- set rdp clientprofile [-randomizeRDPFilename] [-rdpLinkAttribute]
- show rdp clientprofile [-randomizeRDPFilename] [-rdpLinkAttribute]
- add rdp serverprofile [-rdpRedirection]
- set rdp serverprofile [-rdpRedirection]
- show rdp serverprofile [-rdpRedirection]

コマンドグループ: SSL

コマンド:

- create ssl ecdsaKey [-pkcs8]
- ssl certKey [-deletefromdevice]
- clear ssl certKey [-ocspstaplingCache]
- add ssl action [-caCertGrpName]
- show ssl action [-clientCertVerification] [-caCertGrpName]
- show ssl ocspResponder [-port]
- create ssl rsakey [-pkcs8]
- set ssl parameter [-ndcppComplianceCertCheck]
- show ssl parameter [-ndcppComplianceCertCheck]
- set ssl vserver [-preload] [-preload]
- show ssl dtlsProfile
- add ssl profile [-preload]

- set ssl profile [-preload]
- show ssl profile [-preload]

コマンドグループ: システム

コマンド:

- create system backup [-useLocalTimezone] [-includekernel]
- show system backup [-useLocalTimezone]
- add ns tcpProfile [-taillossprobe]
- set ns tcpProfile [-taillossprobe]
- show ns tcpProfile [-taillossprobe]
- add ns icapProfile [-insertHTTPRequest]
- set ns icapProfile [-insertHTTPRequest]
- show ns icapProfile [-insertHTTPRequest]
- add ns httpProfile [-markTraceReqInval]
- set ns httpProfile [-markTraceReqInval]
- unset ns httpProfile [-markTraceReqInval]
- show ns httpProfile[- markTraceReqInval]
- save ns config all
- set ns param [-mgmthttpport] [-mgmthttpsport] [-proxyProtocol]
- show ns param [-mgmthttpport] [-mgmthttpsport] [-proxyProtocol]
- set ns httpParam [-ignoreConnectCodingScheme]
- show ns httpParam [-ignoreConnectCodingScheme]
- add ns icapProfile [-logAction]
- set ns icapProfile [-logAction]
- show ns icapProfile [-logAction]

コマンドグループ: サブスクライバー

コマンド:

- set subscriber gxInterface [-healthCheck] [-healthCheckTTL] [-cerRequestTimeout] [-negativeTTLLimitedSuccess] [-purgeSDBonGxFailure] [-gxReportingAvp1] [-gxReportingAvp1VendorId] [-gxReportingAvp1Type] [-gxReportingAvp2] [-gxReportingAvp2VendorId] [-gxReportingAvp2Type] [-gxReportingAvp3] [-gxReportingAvp3VendorId] [-gxReportingAvp3Type] [-gxReportingAvp4] [-gxReportingAvp4VendorId] [-gxReportingAvp4Type] [-gxReportingAvp5] [-gxReportingAvp5VendorId] [-gxReportingAvp5Type]
- show subscriber gxInterface [-healthCheck] [-healthCheckTTL] [-cerRequestTimeout] [-negativeTTLLimitedSuccess] [-purgeSDBonGxFailure] [-gxReportingAvp1] [-gxReportingAvp1VendorId] [-gxReportingAvp1Type] [-gxReportingAvp2] [-gxReportingAvp2VendorId] [-gxReportingAvp2Type] [-gxReportingAvp3] [-gxReportingAvp3VendorId] [-gxReportingAvp3Type] [-gxReportingAvp4] [-gxReportingAvp4VendorId] [-gxReportingAvp4Type] [-gxReportingAvp5] [-gxReportingAvp5VendorId] [-gxReportingAvp5Type]

コマンドグループ: トラフィック管理

コマンド:

- show tm sessionPolicy
- show tm sessionAction
- show tm global
- show tm sessionParameter [-tmsessionpolicyBindtype] [-tmsessionpolicyCount]

コマンドグループ: URL フィルタリング

コマンド:

- set urlfiltering parameter [-CloudHost] [-SeedDBPath]
- show urlfiltering parameter [-CloudHost] [-SeedDBPath]

コマンドグループ: VPN

コマンド:

- show vpn sessionPolicy
- add vpn sessionAction [-fqdnSpoofedIP]
- set vpn sessionAction [-fqdnSpoofedIP]
- show vpn sessionAction [-feature] [-fqdnSpoofedIP]
- show vpn clientlessAccessPolicy
- bind vpn global [-userDataEncryptionKey]
- unbind vpn global [-userDataEncryptionKey]
- show vpn global [-userDataEncryptionKey]
- set vpn parameter [-fqdnSpoofedIP] [-netmask]
- unset vpn parameter [-fqdnSpoofedIP]
- show vpn parameter [-fqdnSpoofedIP]

Deprecated parameters

コマンドグループ: AppFlow

コマンド:

- add appflow action [-MetricsLog]
- show appflow action [-MetricsLog]

コマンドグループ: LSN

コマンド:

- add lsn transportprofile [-stuntimeout]
- set lsn transportprofile [-stuntimeout]

- show lsn transportprofile [-stuntimeout]

コマンドグループ: Network

コマンド:

- clear rnat

通信サービスプロバイダ向けソリューション

October 7, 2021

情報通信技術（ICT）は、インターネットユーザーをアプリやデータに近づけることです。最新のデータセンターテクノロジにより、ユーザー、アプリ、データをどこにでも配置できるようになりました。ユーザーは、オフィスや自宅から、または空港などの場所からアプリやデータにアクセスできます。アプリとデータは、企業の構内、パブリッククラウド、プライベートクラウド、またはハイブリッドホストのいずれかに配置できます。その結果、生産性の向上だけでなく、所有コストとメンテナンスのコストも削減されました。

サービス提供は、ユーザーのアプリやデータをネットワーク上で運ぶために必要なコアインフラストラクチャを提供します。コアインフラストラクチャは、数百万人の加入者とさまざまなアプリとデータに対応しているため、規模とプロトコルのサポートに対する要件は非常に高くなっています。コアインフラストラクチャは、データプレーンとコントロールプレーンの 2 つの主要なタイプのトラフィックを処理します。これらの各プレーンには、独自のスケールおよびプロトコルサポート要件があります。

データプレーンは、エンドツーエンド、つまりエンドユーザー機器とアプリケーションサーバーの間でユーザーアプリとデータを伝送するコアインフラストラクチャの一部です。アプリやデータにアクセスするユーザーの数は数千万人であるため、スループットと IP アドレスの要件は非常に高くなります。ネットワーク内のすべてのユーザーは、一意に識別できる必要があります。そうしないと、サービスプロバイダーはトラフィックの制御、ネットワーク使用率の監視、ユーザー固有のサービスの提供、および情報のログを正しく行うことができます。今日のクライアントデバイスとアプリケーションサーバーの多くは、IPv6 をネイティブにサポートしています。コアインフラストラクチャは、IPv4 と IPv6 のクライアントとサーバーを組み合わせてサポートするだけでなく、IPv4 と IPv6 の間の相互通信技術も提供する必要があります。最後に、サービスプロバイダは、サービス品質（エンドユーザーエクスペリエンスに直接関係する）とサービスの可用性を中断することなく測定されます。データプレーンは、品質と可用性の両方を同時に提供するのに十分な耐障害性を備える必要があります。

コントロールプレーンインフラストラクチャは、ユーザトラフィックを管理し、ビジネスおよびネットワークオペレーションサービスを維持します。このプレーンで実行される多くのプロトコルの中で最も重要なのは、直径、半径、および SMPP です。直径は、他のいくつかの機能固有のプロトコルが開発されている上で基本プロトコルです。次に例を示します：

- ポリシーと課金実施機能（PCEF）とポリシーおよび課金ルール機能（PCRF）の間の Gx インターフェイス
- オンライン課金システム（OCS）と Cisco Packet Data Network Gateway（PGW）/ポリシーおよび課金実施機能（PCEF）の間の Gy インターフェイス

コントロールプレーントラフィックの量は、ユーザアクティビティに正比例します。コントロールプレーントラフィックを管理するために、サービスプロバイダーは、ロードバランシングやコンテンツスイッチングなど、複数の ADC 機能を使用します。コントロールプレーントラフィックのきめ細かな制御が必要です。これは、データプレーントラフィックの複雑さと同じです。

サービスプロバイダーは、要求の厳しいサービスレベル契約 (SLA) を満たす必要があります。また、規制当局によるコンプライアンスの徹底的な調査を受けています。データおよびコントロールプレーンのトラフィックを管理しながら要件を遵守するには、サービスプロバイダーは、インフラストラクチャを予算内で軽快に、容易にアップグレードでき、柔軟性を維持する必要があります。今日の市場で最も強力で先進的な ADC である Citrix ADC 製品は、サービスプロバイダー環境に自然に適合します。

大規模 **NAT**

October 7, 2021

> 注
>
> この機能は、Citrix ADC Advanced エディションまたは Premium エディションのライセンスで利用できます。

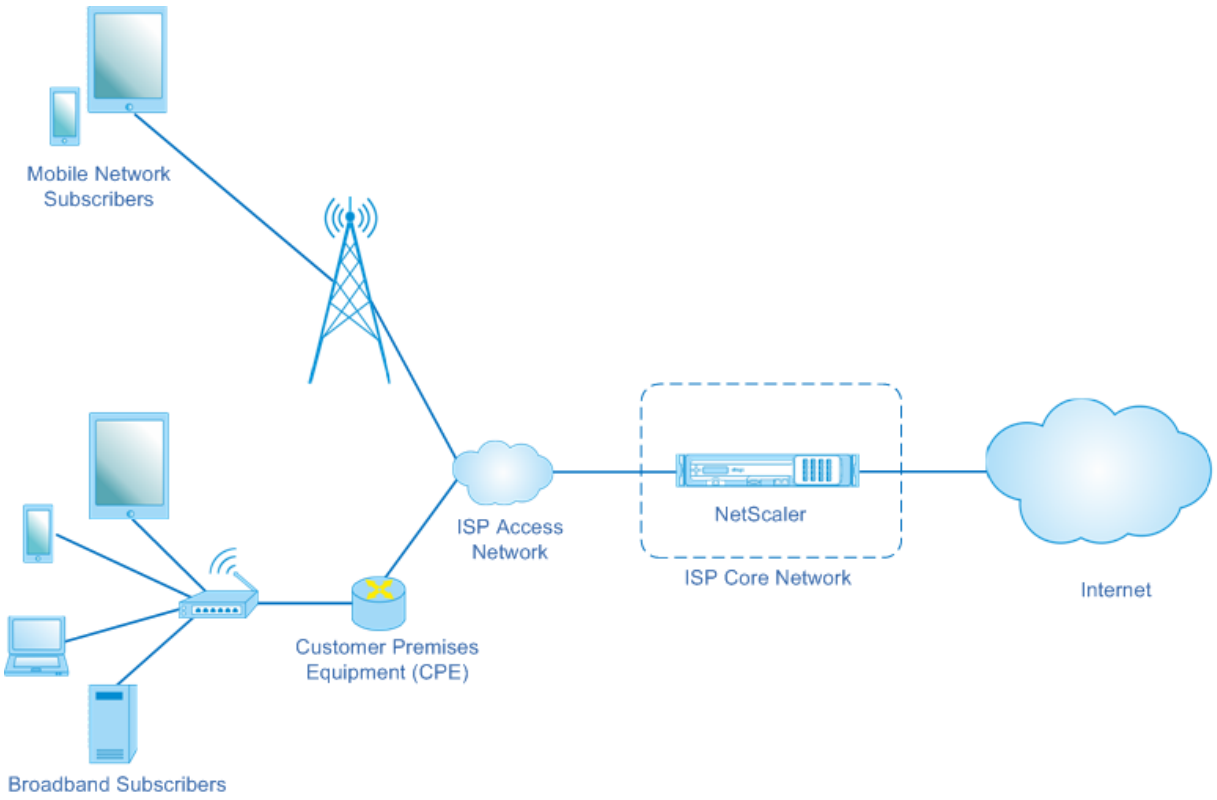
インターネットの驚異的な成長により、パブリック IPv4 アドレスが不足しています。大規模 NAT (LSN/CGNAT) は、この問題に対する解決策を提供し、大規模なインターネットユーザーのプールで少数のパブリック IPv4 アドレスを共有することで、使用可能なパブリック IPv4 アドレスを最大限に活用します。

LSN は、プライベート IPv4 アドレスをパブリック IPv4 アドレスに変換します。これには、多くのプライベート IP アドレスをより少ないパブリック IPv4 アドレスに集約するネットワークアドレスおよびポート変換方式が含まれます。LSN は、大規模な NAT を処理するように設計されています。Citrix ADC LSN 機能は、インターネットサービスプロバイダ（ISP）や通信事業者にとって、多数のユーザー（加入者）と非常に高いスループットで多数の翻訳をサポートするために非常に便利です。

LSN アーキテクチャ

Citrix 製品を使用する ISP の LSN アーキテクチャは、ISP のコアネットワークに展開された Citrix ADC アプライアンスを介してインターネットにアクセスするプライベートアドレス空間のサブスクライバ（インターネットユーザー）で構成されます。加入者は、ISP のアクセスネットワークを介して ISP に接続されています。通常、インターネットを商業的に使用するための加入者は、ISP のアクセスネットワークに直接接続されています。これらのサブスクライバにサービスを提供するには、1 レベルの NAT（NAT44）だけが必要です。

ただし、非商用加入者は、通常、NAT を実装するルータやモデムなどの顧客構内機器（CPE）の背後にいます。これら 2 つのレベルの NAT は、NAT444 モデルを作成します。LSN 機能のために ISP のコアネットワークに Citrix ADC アプライアンスを展開することは、加入者には透過的であり、加入者や CPE の設定を変更する必要はありません。



Large Scale NAT

Citrix ADC アプライアンスは、インターネット宛てのサブスクライバパケットをすべて受信します。アプライアンスは、LSN に使用する事前定義済みの NAT IP アドレスのプールで構成されます。Citrix ADC アプライアンスは、LSN 機能を使用してパケットの送信元 IP アドレス（プライベート）とポートを NAT IP アドレス（パブリック）と NAT ポートに変換し、インターネット上の宛先にパケットを送信します。アプライアンスは、LSN 機能を使用するすべてのアクティブセッションの記録を保持します。これらのセッションを LSN セッションと呼びます。Citrix ADC アプライアンスは、セッションごとにサブスクライバ IP アドレスとポート、NAT IP アドレスとポートのマッピングも維持します。これらのマッピングは LSN マッピングと呼ばれます。Citrix ADC アプライアンスは、LSN セッションと LSN マッピングから、特定のセッションに属する応答パケット（インターネットから受信した）を認識します。アプライアンスは、応答パケットの宛先 IP アドレスとポートを NAT IP アドレス: ポートからサブスクライバ IP アドレス: ポートに変換し、変換されたパケットをサブスクライバに送信します。

Citrix ADC アプライアンスでサポートされる **LSN** 機能

Citrix ADC アプライアンスでサポートされる LSN 機能の一部を以下に示します。

NAT リソース割り当て

Citrix ADC アプライアンスは、事前定義された NAT リソースプールから NAT IP アドレスとポートをサブスクライバに割り当てて、パケットを外部ホスト（インターネット）に送信するために変換します。Citrix ADC アプライアンスは、加入者に対して次のタイプの NAT IP アドレスとポート割り当てをサポートします。

- **Deterministic**。Citrix ADC アプライアンスは、NAT IP アドレスとポートのブロックを各サブスクライバに割り当てます。アプライアンスは、これらのサブスクライバに NAT リソースを順番に割り当てます。これは、開始 NAT IP アドレスの最初のポートブロックを開始サブスクライバ IP アドレスに割り当てます。次の範囲のポートは、次のサブスクライバに割り当てられます。次のサブスクライバに対して十分なポートが NAT アドレスにないまで、次のサブスクライバに割り当てられます。この時点で、次の NAT アドレスの最初のポートブロックがサブスクライバに割り当てられます。

  Citrix ADC アプライアンスは、割り当てられた NAT IP アドレスと加入者のポートブロックを記録します。接続の場合、加入者は、マッピングされた NAT IP アドレスとポートブロックによってのみ識別できます。このため、Citrix ADC アプライアンスは、作成または削除された LSN セッションを記録しません。ポートのブロック全体が使用されている場合、Citrix ADC アプライアンスはサブスクライバーからの新しい接続を切断します。

- **Dynamic**。Citrix ADC アプライアンスは、加入者の接続のために、ランダムな NAT IP アドレスと LSN NAT プールからポートを割り当てます。構成でポートブロック割り当てが有効になっている場合、アプライアンスはサブスクライバが初めて接続を開始するときに、ランダムな NAT IP アドレスとポートブロックを割り当てます。次に、Citrix ADC アプライアンスは、この NAT IP アドレスと割り当てられたブロックのポートの 1 つを、このサブスクライバーからの後続の各接続に割り当てます。ポートのブロック全体が使用されている場合、アプライアンスは新しい接続を開始するときに、サブスクライバに新しいランダムポートブロックを割り当てます。新しいポートブロック内のポートの 1 つが新しい接続に割り当てられます。

IP プーリング

次の NAT リソース割り当てオプションは、既存のセッションにランダム NAT IP アドレスとポートが割り当てられた加入者の後続のセッションで使用できます。

- **Paired**。Citrix ADC アプライアンスは、同じサブスクライバーに関連付けられたすべてのセッションに同じ NAT IP アドレスを割り当てます。そのアドレスに使用できるポートがなくなると、アプライアンスはサブスクライバからの新しい接続をすべてドロップします。このオプションは、同じ送信元 IP アドレスで複数のセッションを作成する必要がある特定のアプリケーション（たとえば、RTP または RTCP プロトコルを使用するピアツーピアアプリケーションなど）を適切に機能させるために必要です。
- **Random**。Citrix ADC アプライアンスは、同じサブスクライバーに関連付けられた異なるセッションに対して、プールからランダムな NAT IP アドレスを割り当てます。

LSN マッピングの再利用

Citrix ADC アプライアンスは、同じ加入者の IP アドレスとポートから発信された新しい接続に対して、既存の LSN マップを再利用できます。Citrix ADC LSN 機能では、次の LSN マッピングの再利用がサポートされています。

1. **Endpoint Independent**。Citrix ADC アプライアンスは、同じ加入者の IP アドレスとポート（X：x）から任意の外部 IP アドレスとポートに送信される後続のパケットに対して、LSN マッピングを再利用します。このタイプの LSN マップ再利用は、VOIP およびピアツーピアアプリケーションを適切に機能させる場合に便利です。

2. **Address dependent**。Citrix ADC アプライアンスは、外部ポートに関係なく、同じサブスクライバ IP アドレスとポート（X：x）から同じ外部 IP アドレス（Y）に送信される後続のパケットに対して、LSN マッピングを再利用します。
3. **Address port dependent**。Citrix ADC アプライアンスは、マッピングがアクティブな状態で、同じ内部 IP アドレスとポート（X：x）から同じ外部 IP アドレスとポート（Y：y）に送信される後続のパケットに対して LSN マッピングを再利用します。

LSN フィルタリング

Citrix ADC アプライアンスは、アクティブな LSN セッションと LSN マッピングに基づいて、外部ホストからのパケットをフィルタリングできます。加入者 IP: ポート（X: x）、NAT IP: ポート（N: n）、および外部ホスト IP: ポート（Y: y）のマッピングを含む LSN マッピングの例を考えてみましょう。Citrix ADC LSN 機能では、次の種類のフィルタリングがサポートされています。

1. **Endpoint Independent**。Citrix ADC アプライアンスは、NAT IP: port（N: n）宛てでないパケットのみをフィルタリングします。NAT IP: port（N: n）は、外部ホストの IP アドレスとポートソース（Z: z）に関係なく、サブスクライバ IP: port（X: x）を表します。Citrix ADC アプライアンスは、X: x 宛てのパケットをすべて転送します。つまり、任意の外部ホストから加入者へのパケットを許可するには、加入者から任意の外部 IP アドレスへのパケットを送信するだけで十分です。このタイプのフィルタリングは、VOIP およびピアツーピアアプリケーションを適切に機能させる場合に便利です。
2. **Address dependent**。Citrix ADC アプライアンスは、NAT IP：ポート（N：n）を宛先としないパケットを除外します。これは、サブスクライバ IP：ポート（X：x）を表します。さらに、加入者が以前 Y: anyport（外部ポートに依存しない）にパケットを送信していない場合、アプライアンスは N: n 宛ての外部ホストの IP アドレスおよびポート（Y: y）からのパケットを除外します。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその特定の外部ホストの IP アドレスにパケットを送信する必要があります。
3. **Address port dependent**。Citrix ADC アプライアンスは、NAT IP：ポート（N：n）を宛先としないパケットを除外します。これは、サブスクライバ IP：ポート（X：x）を表します。さらに、加入者が以前に Y: y にパケットを送信していない場合、アプライアンスは N: n 宛ての外部ホストの IP アドレスおよびポート（Y: y）からのパケットを除外します。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその特定の外部 IP アドレスおよびポートにパケットを送信する必要があります。

クォータ

Citrix ADC アプライアンスは、各サブスクライバの NAT ポートとセッションの数を制限して、サブスクライバ間でリソースを公平に分散させることができます。Citrix ADC アプライアンスは、サブスクライバグループのセッション数を制限して、異なるサブスクライバグループ間でリソースを均等に分散させることもできます。

- ポートクォータ。Citrix ADC アプライアンスは、LSN NAT ポートを、各加入者が指定したプロトコルに対して一度に使用する制限できます。たとえば、各加入者を最大 500 個の TCP NAT ポートに制限できます。加入者の LSN NAT マッピングが制限に達すると、Citrix ADC アプライアンスは、指定されたプロトコルの追加の NAT ポートをその加入者に割り当てません。

- サブスクライバセッション制限。加入者の同時セッション数は、ポートクォータを超える場合があります。Citrix ADC アプライアンスは、指定されたプロトコルの各加入者に許可される LSN セッションを制限できます。LSN セッション数が加入者の制限に達すると、Citrix ADC アプライアンスは、加入者が指定したプロトコルの追加セッションを開くことを許可しません。
- グループ・セッション制限。Citrix ADC アプライアンスは、指定されたプロトコルのサブスクライバーグループに許可される LSN セッションの総数を制限できます。LSN セッションの総数が、指定したプロトコルのグループの制限に達すると、Citrix ADC アプライアンスは、そのグループのサブスクライバが指定したプロトコルの追加セッションを開くことを許可しません。たとえば、グループを最大 10000 の UDP セッションに制限します。このグループの UDP セッションの総数が 10000 に達すると、Citrix ADC アプライアンスは、グループのサブスクライバが追加の UDP セッションを開くことを許可しません。

アプリケーション層ゲートウェイ

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットのペイロードで通信されます。プロトコルのアプリケーション層ゲートウェイは、パケットのペイロードを解析し、プロトコルが LSN 上で動作し続けるために必要な変更を行います。

Citrix ADC アプライアンスは、次のプロトコルに対して ALG をサポートしています。

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

ヘアピンサポート

Citrix ADC アプライアンスは、NAT IP アドレスを使用したサブスクライバまたは内部ホスト間の通信をサポートします。NAT IP アドレスを使用する 2 つの加入者間のこのタイプの通信は、ヘアピンフローと呼ばれます。ヘアピンフローはデフォルトで有効になっており、無効にすることはできません。

LSN を設定する前に考慮すべきポイント

October 7, 2021

Citrix ADC アプライアンスで LSN を構成する前に、次の点を考慮してください。

- RFC 6888、5382、5508、および 4787 で説明されている大規模な NAT のさまざまなコンポーネントについて理解していることを確認してください。

- エンドポイント独立マッピング（EIM）およびエンドポイント独立フィルタリング（EIF）は、デフォルトで無効になっています。VoIP および P2P アプリケーションを適切に機能させるには、これらのオプションを有効にする必要があります。
- **LSN** のロギング：LSN 情報のロギングに関する考慮事項は次のとおりです。
  - Citrix ADC アプライアンスではなく、外部ログサーバーに LSN 情報を記録することをお勧めします。外部サーバーにログオンすると、アプライアンスが大量の LSN ログエントリを (数百万の順) 作成するときの最適なパフォーマンスが促進されます。
  - Citrix は、TCP または NSLOG を介して SYSLOG を使用することをお勧めします。デフォルトでは、SYSLOG は UDP を使用し、NSLOG は TCP だけを使用してログ情報をログサーバに転送します。TCP は、完全なデータを転送するための UDP よりも信頼性が高い。
  - TCP 経由の SYSLOG には、次の制限が適用されます。
    * Syslog over TCP ソリューションは、認証、整合性チェック、およびプライバシーを提供しません。
    * Citrix ADC アプライアンスは、TCP プロトコルを使用して、外部ログサーバーへの SYSLOG メッセージの配信を確認します。
- 高可用性：LSN 用の Citrix ADC アプライアンスの高可用性に関する考慮事項は次のとおりです。
  - すべての LSN セッションを中断することなくシームレスに操作できるように、2 つの Citrix ADC アプライアンスの高可用性展開で LSN 機能を設定することをお勧めします。
  - 高可用性の展開では、以下のことを推奨します。
    * すべての HA 関連通信用の VLAN 専用の SYNC VLAN パラメータを設定します。
    * 多数の LSN マッピングとセッションをステートフルに同期させるため、プライマリノードの対称 RSS キーをセカンダリノードに同期します。
    * フェールオーバー後にすべての VLAN で GARP ブロードキャストがフラッディングされないように、LSN IP アドレスのサブネットを VLAN にバインドします。
  - Citrix ADC アプライアンスの高可用性展開では、ALG 関連のセッションはセカンダリ・アプライアンスにミラーリングされません。
- アプリケーション層ゲートウェイ（**ALG**）：Citrix ADC アプライアンス上の ALG に関連する考慮事項は次のとおりです。
  - SIP ALG では、次の機能はサポートされていません。
    * マルチキャスト IP アドレス
    * 暗号化 SDP
    * TLS 経由の SIP メッセージ
    * SIP メッセージでの FQDN 変換
    * SIP メッセージの認証
    * トラフィックドメイン、管理パーティション、および Citrix ADC クラスタ。
    * マルチパートボディを持つ SIP メッセージ。
  - RTSP ALG では、次の機能はサポートされていません。
    * マルチキャスト RTSP セッション
    * UDP を介した RTSP セッション

        * Citrix ADC トラフィックドメイン、管理パーティション、および Citrix ADC クラスタ
    - Citrix ADC アプライアンスは、IPSec プロトコルの ALG をサポートしていません。
- Citrix ADC アプライアンス上に LSN セッションがあるときに LSN 機能を無効にした場合、これらのセッションは設定されたタイムアウト間隔の間存続します。
- LSN は RNAT よりも優先されます。指定された LSN サブスクライバからのパケットも RNAT ルールに一致する場合、パケットは LSN 設定に従って変換されます。
- LSN セッションのみに関連するパケットの転送は、Citrix ADC アプライアンスのルーティングテーブルに基づいています。
- サブネット IP アドレスとは異なり、加入者の接続の LSN NAT IP アドレスの選択は、宛先 IP アドレスのルーティングエントリに基づいていません。
- 着信パケットの場合、スタティック LSN マッピングは Dynamic LSN マッピングよりも優先されます。
- 発信パケットの場合、LSN アプリケーションプロファイルは、スタティックマッピングよりも優先されます。
- Citrix ADC アプライアンスに多数の LSN セッション（100 万を超える）が存在する場合は、すべてのセッションではなく、選択した LSN セッションを表示することをお勧めします。コマンド・ライン・インターフェイスまたは構成ユーティリティで、選択パラメータを使用して LSN セッション動作を表示します。
- LSN 機能に割り当てられるアクティブメモリの量を減らすには、構成済みのメモリ設定を変更した後、Citrix ADC アプライアンスをウォーム再起動する必要があります。ウォームリスタートを使用しない場合、アクティブメモリの量だけを増やすことができます。

LSN の設定手順

October 7, 2021

Citrix ADC アプライアンスで LSN を構成するには、次の作業を行います。

1. グローバル **LSN** パラメータを設定します。グローバルパラメーターには、LSN 機能用に予約されている Citrix ADC メモリの量と、高可用性セットアップでの LSN セッションの同期が含まれます。
2. **LSN** クライアントエンティティを作成し、加入者をバインドします。LSN クライアントエンティティは、Citrix ADC アプライアンスで LSN を実行するトラフィックを持つサブスクライバのセットです。クライアントエンティティには、サブスクライバを識別するための IPv4 アドレスと拡張 ACL ルールが含まれます。LSN クライアントは、1 つの LSN グループにしかバインドできません。コマンドラインインターフェイスには、LSN クライアントエンティティを作成し、加入者を LSN クライアントエンティティにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。
3. **LSN** プールを作成し、**NAT IP** アドレスをそれにバインドします。NAT IPLSN プールは、LSN を実行するために Citrix ADC アプライアンスが使用する NAT IP アドレスのプールを定義します。プールには、ポートブロック割り当てや NAT タイプ（Deterministic または Dynamic）などのパラメータが割り当てられます。LSN グループにバインドされた LSN プールは、同じグループにバインドされた LSN クライアントエンティティのすべてのサブスクライバに適用されます。同じ NAT タイプ設定の LSN プールと LSN グループだけを 1 つにバインドできます。複数の LSN プールを 1 つの LSN グループにバインドできます。Dynamic NAT で

は、LSN プールを複数の LSN グループにバインドできます。Deterministic NAT の場合、LSN グループにバインドされたプールは他の LSN グループにバインドできません。コマンドラインインターフェイスには、LSN プールを作成し、NAT IP アドレスを LSN プールにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。

4. （オプション）指定されたプロトコルの **LSN** トランスポートプロファイルを作成します。LSN トランスポートプロファイルでは、加入者が特定のプロトコルに対して持つことのできる最大 LSN セッションや最大ポート使用量など、さまざまなタイムアウトと制限を定義します。各プロトコル (TCP、UDP、および ICMP) の LSN トランスポートプロファイルを LSN グループにバインドします。プロファイルは、複数の LSN グループにバインドできます。LSN グループにバインドされたプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのトランスポートプロファイルと呼ばれます。LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルを上書きします。
5. （オプション）指定されたプロトコルの **LSN** アプリケーションプロファイルを作成し、宛先ポートのセットをそれにバインドします。LSN アプリケーションプロファイルは、特定のプロトコルおよび一連の宛先ポートのグループの LSN マッピングおよび LSN フィルタリングコントロールを定義します。一連の宛先ポートでは、各プロトコル（TCP、UDP、および ICMP）の LSN プロファイルを LSN グループにバインドします。プロファイルは、複数の LSN グループにバインドできます。LSN グループにバインドされた LSN アプリケーションプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのアプリケーションプロファイルと呼ばれます。指定された宛先ポートセットを持つ LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートのセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルを上書きします。コマンドラインインターフェイスには、LSN アプリケーションプロファイルを作成し、一連の宛先ポートを LSN アプリケーションプロファイルにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。
6. **LSN** グループを作成し、**LSN** プール、**(**オプション**) LSN** トランスポートプロファイル、**(**オプション**) LSN** アプリケーションプロファイルを **LSN** グループにバインドします。LSN グループは、LSN クライアント、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルで構成されるエンティティです。グループには、ポートブロックサイズや LSN セッションのロギングなどのパラメータが割り当てられます。パラメータ設定は、LSN グループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。同じ NAT タイプ設定の LSN プールと LSN グループだけを 1 つにバインドできます。複数 LSN プールを 1 つの LSN グループにバインドできます。Dynamic NAT では、LSN プールを複数の LSN グループにバインドできます。Deterministic NAT の場合、LSN グループにバインドされたプールは他の LSN グループにバインドできません。LSN グループ 1 つの LSN クライアントエンティティのみにバインドでき、LSN グループにバインドされた LSN クライアントエンティティは他の LSN グループにバインドできません。コマンドラインインターフェイスには、LSN グループを作成し、LSN プール、LSN トランスポー

トプロファイル、LSN アプリケーションプロファイルを LSN グループにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。

次の表に、Citrix ADC アプライアンスで作成できる LSN エンティティとバインディングの最大数を示します。これらの制限は、Citrix ADC アプライアンスで使用可能なメモリにも左右されます。

LSN エンティティとバインディング	上限
LSN クライアント	1024
LSN プール	128
LSN グループ	1024
LSN クライアントにバインドできるサブスクライバネットワーク	64
LSN クライアントにバインドできる拡張 ACL	1024
プール内の NAT IP アドレス	4096
LSN グループにバインドできる LSN プール	8
同じ LSN プールを使用できる LSN グループ	16
LSN グループにバインドできる LSN トランスポートプロファイル	3（TCP、UDP、および ICMP プロトコルごとにそれぞれ 1 つ）
同じ LSN トランスポートプロファイルを使用できる LSN グループ	8
LSN グループにバインドできる LSN アプリケーションプロファイル	64
同じ LSN アプリケーションプロファイルを使用できる LSN グループ	8
LSN アプリケーションプロファイルにバインドできるポート範囲	8

コマンドラインインターフェイスを使用した設定

コマンドラインインターフェイスを使用して **LSN** クライアントを作成するには

コマンドプロンプトで入力します。

```
add lsn client <clientname>

show lsn client
```



```
4 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、ネットワークアドレスまたは **ACL** ルールを **LSN** クライアントにバインドするには

コマンドプロンプトで入力します。

```
1 bind lsn client <clientname> ((-network <ip_addr> [-netmask <netmask>]
     [-td<positive_integer>]) | -aclname <string>)
2
3 show lsn client
4 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プールを作成するには

コマンドプロンプトで入力します。

```
1 add lsn pool <poolname> [-nattype ( DYNAMIC | DETERMINISTIC )] [-
     portblockallocation ( ENABLED | DISABLED )] [-portrealloctimeout <
     secs>] [-maxPortReallocTmq <positive_integer>]
2
3 show lsn pool
4 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IP** アドレス範囲を **LSN** プールにバインドするには

コマンドプロンプトで入力します。

```
1 bind lsn pool <poolname> <lsnip>
2
3 show lsn pool
4 <!--NeedCopy-->
```

注意: LSN プールから LSNIP アドレスを削除するには、unbind lsnpool コマンドを使用します。

コマンドラインインターフェイスを使用して **LSN** トランスポートプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )] [-td <positive_integer>]

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、アプリケーションプロトコルポート範囲を **LSN** アプリケーションプロファイルにバインドするには

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** グループを作成するには

コマンドプロンプトで入力します。


```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
    DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging (
    ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
    sessionSync (ENABLED | DISABLED )] [-snmptraplimit <positive_integer
    >] [-ftp ( ENABLED | DISABLED )]

show lsn group
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プロファイルと **LSN** プールを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -appsprofilename <string>)

show lsn group
<!--NeedCopy-->
```

設定ユーティリティを使用した設定

構成ユーティリティを使用して LSN クライアントを構成し、IPv4 ネットワークアドレスまたは ACL ルールをバインドするには

[システム] > **[**大規模 **NAT]** > [クライアント] に移動し、クライアントを追加し、IPv4 ネットワークアドレスまたは ACL ルールをクライアントにバインドします。

構成ユーティリティを使用して LSN プールを構成し、NAT IP アドレスをバインドするには

System > Large Scale NAT > **Pools** に移動してプールを追加し、NAT IP アドレスまたは NAT IP アドレスの範囲をプールにバインドします。

構成ユーティリティを使用して LSN トランスポートプロファイルを構成するには

1. **System > Large Scale NAT > Profiles** に移動します。
2. 詳細ウィンドウで、[トランスポート] タブをクリックし、トランスポートプロファイルを追加します。

構成ユーティリティを使用して LSN アプリケーションプロファイルを構成するには

1. **System > Large Scale NAT > Profiles** に移動します。
2. 詳細ウィンドウで、[アプリケーション] タブをクリックし、アプリケーションプロファイルを追加します。

構成ユーティリティを使用して LSN グループを構成し、LSN クライアント、プール、トランスポートプロファイル、およびアプリケーションプロファイルをバインドするには

[システム**] > [**大規模 **NAT] > [**グループ] に移動し、グループを追加してから、LSN クライアント、プール、トランスポートプロファイル、およびアプリケーションプロファイルをグループにバインドします。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn client
  - clientname

    LSN クライアントエンティティの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN クライアントが作成された後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn client1” や”lsn client1” など）。

    これは必須の引数です。最大長：127

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- bind lsn client
  - clientname

    LSN クライアントエンティティの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN クライアントが作成された後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn client1” や”lsn client1” など）。

    これは必須の引数です。最大長：127

  - network

    Citrix ADC アプライアンスで大規模 NAT を実行するトラフィックの LSN 加入者または加入者ネットワークの IPv4 アドレス。

  - netmask

    Network パラメータで指定された IPv4 アドレスのサブネットマスク。

    デフォルト値：255.255.255.255

  - td

    この加入者または加入者ネットワーク（network パラメータで指定されたもの）が属するトラフィックドメインの ID。

    ID を指定しない場合、加入者または加入者ネットワークはデフォルトのトラフィックドメインの一部になります。

デフォルト値:0

最小値:0

最大値:4094

– aclname

アクションが ALLOW である設定済みの拡張 ACL の名前。拡張 ACL ルールで指定された条件は、Citrix ADC アプライアンスが大規模な NAT を実行する対象となる LSN サブスクライバーからのトラフィックを識別します。最大長：127

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn pool
  - poolname

    LSN プールの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN プールの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn pool1” または”lsn pool1”）。

    これは必須の引数です。最大長：127

  - nattype

    (LSN グループにバインドされた LSN クライアントエンティティの）加入者の NAT IP アドレスおよびポート割り当てのタイプ（LSN グループにバインドされた LSN プールからの）:

    使用可能なオプションは次のように機能します。

    * **Deterministic**:（LSN グループにバインドされた LSN クライアントの）各加入者に NAT IP アドレスとポートブロックを割り当てます。Citrix ADC アプライアンスは、NAT リソースをこれらのサブスクライバーに順番に割り当てます。Citrix ADC アプライアンスは、開始 NAT IP アドレス上のポートの最初のブロック（LSN グループのポートブロックサイズパラメータによって決定されるブロックサイズ）を、開始サブスクライバ IP アドレスに割り当てます。次の範囲のポートは、次のサブスクライバに割り当てられます。次のサブスクライバに対して十分なポートが NAT アドレスにないまで、次のサブスクライバに割り当てられます。この場合、次の NAT アドレスの最初のポートブロックがサブスクライバに使用されます。各加入者は、Deterministic NAT IP アドレスとポートブロックを受信するようになったため、ロギングを必要とせずに加入者を識別できます。接続の場合、加入者は NAT IP アドレスとポート、および宛先 IP アドレスとポートに基づいてのみ識別できます。

    * **Dynamic**: 加入者接続に、ランダムな NAT IP アドレスと LSN NAT プールからポートを割り当てます。（LSN プールで）ポートブロック割り当てが有効で、（LSN グループで）ポートブロック

サイズが指定されている場合、Citrix ADC アプライアンスは、サブスクライバが初めて接続を開始するときに、ランダムな NAT IP アドレスとポートのブロックを割り当てます。アプライアンスは、この加入者からの異なる接続に、この NAT IP アドレスとポート（割り当てられたポートブロックから）を割り当てます。サブスクライバに割り当てられたポートブロックからすべてのポートが（異なるサブスクライバ接続に対して）割り当てられている場合、アプライアンスはサブスクライバに新しいランダムポートブロックを割り当てます。同じ NAT タイプ設定の LSN プールと LSN グループだけを 1 つにバインドできます。複数 LSN プールを 1 つの LSN グループにバインドできます。

設定可能な値: DYNAMIC, DETERMINISTIC

デフォルト値: DYNAMIC

- portblockallocation

  NAT 割り当てが Dynamic NAT として設定されている場合、NAT IP アドレスの使用可能な NAT ポートプールからランダムな NAT ポートブロックを各サブスクライバに割り当てます。サブスクライバから開始された接続では、Citrix ADC アプライアンスは、サブスクライバに割り当てられた NAT ポートブロックから NAT ポートを割り当てて、LSN セッションを作成します。

  バインドされた LSN グループのポートブロックサイズを設定する必要があります。サブスクライバの場合、サブスクライバに割り当てられたポートブロックからすべてのポートが割り当てられている場合、Citrix ADC アプライアンスはサブスクライバに新しいランダムなポートブロックを割り当てます。

  Deterministic NAT の場合、このパラメータはデフォルトで有効になっており、無効にできません。

  設定可能な値：ENABLED, DISABLED

  デフォルト値：無効

- portrealloctimeout

  LSN NAT ポートの割り当て解除（LSN マッピングが削除された場合）から新しい LSN セッションに再割り当てするまでの待機時間（秒）。このパラメータは、古いマッピングと新しいマッピングとセッション間の衝突を防ぐために必要です。これにより、確立されたすべてのセッションが別のサブスクライバにリダイレクトされるのではなく、確実に中断されます。これは、で使用されるポートには適用されません。

  * Deterministic NAT
  * アドレス依存フィルタリングとアドレスポート依存フィルタリング
  * ポートブロック割り当てによる Dynamic NAT

  この場合、ポートはただちに再割り当てされます。

  デフォルト値:0

  最大値:600

- maxPortReallocTmq

NAT IP アドレスごとにポート再割り当てタイムアウトが適用されるポートの最大数。つまり、NAT IP アドレスごとに再割り当てタイムアウトが適用される、割り当て解除されたポートキューサイズの最大値。

キューサイズがいっぱいになると、割り当て解除された次のポートは、新しい LSN セッション用にただちに再割り当てされます。

デフォルト値:65536

最大値:65536

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- bind lsn pool
  - poolname

    LSN プールの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN プールの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn pool1” または”lsn pool1”）。

    これは必須の引数です。最大長：127

  - lsnip

    LSN の NAT IP アドレスとして使用する IPv4 アドレスまたは IPv4 アドレスの範囲。

    プールが作成されると、これらの IPv4 アドレスは、タイプ LSN の Citrix ADC が所有する IP アドレスとして Citrix ADC アプライアンスに追加されます。LSN プールに関連付けられた LSN IP アドレスは、他の LSN プールと共有できません。このパラメータに指定する IP アドレスは、Citrix ADC アプライアンスに Citrix ADC が所有する IP アドレスとして存在していない必要があります。コマンドラインインターフェイスで、範囲をハイフンで区切ります。たとえば、次のようになります。後でプールから一部またはすべての LSN IP アドレスを削除し、LSN プールに IP アドレスを追加できます。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- lsn トランスポートプロファイルを追加する
  - transportprofilename

    LSN トランスポートプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN トランスポートプロファイルの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場

合は、名前を二重引用符または一重引用符で囲みます（たとえば、「lsn トランスポートプロファイル 1」または「lsn トランスポートプロファイル 1」）。

これは必須の引数です。最大長：127

– transportprotocol

LSN トランスポートプロファイルパラメータを設定するプロトコル。

これは必須の引数です。

指定可能な値：TCP、UDP、ICMP

– sessiontimeout

アイドル状態の LSN セッションのタイムアウト（秒単位）。LSN セッションがこの値を超える時間アイドル状態になると、Citrix ADC アプライアンスはセッションを削除します。

このタイムアウトは、いずれかのエンドポイントから FIN メッセージまたは RST メッセージを受信した場合、TCP LSN セッションには適用されません。

デフォルト値:120

最小値:60

– finrsttimeout

エンドポイントの 1 つから FIN または RST メッセージを受信した後の TCP LSN セッションのタイムアウト（秒単位）。

この値を超えた時間（Citrix ADC アプライアンスが FIN または RST メッセージを受信した後）TCP LSN セッションがアイドル状態になると、Citrix ADC アプライアンスはセッションを削除します。

Citrix ADC アプライアンスの LSN 機能は、TCP LSN セッションの状態情報を保持しないため、このタイムアウトは、両方のエンドポイントが適切に接続を閉じることができるように、FIN または RST の送信、および他方のエンドポイントからの ACK メッセージに対応します。

デフォルト値:30

– portquota

指定されたプロトコルの各サブスクライバが一度に使用する LSN NAT ポートの最大数。たとえば、各サブスクライバは最大 500 個の TCP NAT ポートに制限できます。加入者の LSN NAT マッピングが制限に達すると、Citrix ADC アプライアンスはその加入者に追加の NAT ポートを割り当てません。

デフォルト値:0

最小値:0

最大値:65535

– sessionquota

指定されたプロトコルで各加入者に許可される同時 LSN セッションの最大数。LSN セッション数が加入者の制限に達すると、Citrix ADC アプライアンスは、加入者が追加のセッションを開くことを許可しません。

デフォルト値:0

最小値:0

最大値:65535

- portpreserveparity

  サブスクライバポートとマッピングされた LSN NAT ポート間のポートパリティを有効にします。たとえば、サブスクライバーが奇数番号のポートから接続を開始した場合、Citrix ADC アプライアンスはこの接続に奇数番号の LSN NAT ポートを割り当てます。RTP または RTCP プロトコルを使用するピアツーピアアプリケーションなど、送信元ポートが偶数または奇数の番号になる必要があるプロトコルを適切に機能させるには、このパラメータを設定する必要があります。

  設定可能な値：ENABLED, DISABLED

  デフォルト値：無効

- portpreserverange

  サブスクライバが既知のポート（0 ～1023）から接続を開始する場合は、この接続に既知のポート範囲（0 ～1023）から NAT ポートを割り当てます。たとえば、加入者がポート 80 から接続を開始した場合、Citrix ADC アプライアンスはポート 100 をこの接続の NAT ポートとして割り当てることができます。

  このパラメータは、ポートブロック割り当てのない Dynamic NAT に適用されます。また、割り当てられたポートの範囲に既知のポートが含まれている場合は、Deterministic NAT にも適用されます。

  使用可能なすべての NAT IP アドレスのすべての既知のポートが異なるサブスクライバ接続（LSN セッション）で使用され、サブスクライバが既知のポートから接続を開始すると、Citrix ADC アプライアンスはこの接続を切断します。

  設定可能な値：ENABLED, DISABLED

  デフォルト値：無効

- syncheck

  Citrix ADC アプライアンス上に LSN-NAT セッションが存在しない接続の非 SYN パケットをサイレントにドロップします。

  このパラメーターを無効にすると、Citrix ADC アプライアンスは非 SYN パケットを受け入れ、この接続用に新しい LSN セッションエントリを作成します。

  Citrix ADC アプライアンスがこのようなパケットを受信する理由は次のとおりです。

  * 接続用の LSN セッションは存在していましたが、Citrix ADC アプライアンスはこのセッションを削除しました。これは、LSN セッションが構成されたセッションタイムアウトを超えた時間アイドル状態だったためです。

* このようなパケットは DoS 攻撃の一部である可能性があります。

設定可能な値：ENABLED, DISABLED

デフォルト値:ENABLED

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn appsprofile
  - appsprofilename

    LSN アプリケーションプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN アプリケーションプロファイルの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn application profile1” または’lsn application profile1’）。

    これは必須の引数です。最大長：127
  - transportprotocol

    この LSN アプリケーションプロファイルのパラメータが適用されるプロトコルの名前。

    これは必須の引数です。

    指定可能な値：TCP、UDP、ICMP
  - ippooling

    同じサブスクライバに関連付けられたセッションの NAT IP アドレス割り当てオプション。

    使用可能なオプションは次のように機能します。

    * **Paired**：Citrix ADC アプライアンスは、同じ加入者に関連付けられたすべてのセッションに同じ NAT IP アドレスを割り当てます。NAT IP アドレスのすべてのポートが LSN セッションで使用されている場合（同一または複数のサブスクライバの場合）、Citrix ADC アプライアンスはサブスクライバからの新しい接続を切断します。
    * **Random**：Citrix ADC アプライアンスは、同じサブスクライバに関連付けられた異なるセッションに対して、プールからランダムな NAT IP アドレスを割り当てます。

    このパラメータは、Dynamic NAT 割り当てにだけ適用できます。

    設定可能な値：PAIRED、RANDOM

    デフォルト値: RANDOM
  - mapping

    同じ加入者 IP アドレスおよびポートから発信された後続のパケットに適用する LSN マッピングのタイプ。

加入者 IP: ポート（X: x）、NAT IP: ポート（N: n）、および外部ホスト IP: ポート（Y: y）のマッピングを含む LSN マッピングの例を考えてみましょう。

使用可能なオプションは次のように機能します。

* **ENDPOINT-INDEPENDENT**：同じ加入者 IP アドレスおよびポート（X: x）から任意の外部 IP アドレスおよびポートに送信される後続のパケットに対して、LSN マッピングを再使用します。
* **ADDRESS-DEPENDENT**：外部ポートに関係なく、同じ加入者 IP アドレスおよびポート（X: x）から同じ外部 IP アドレス（Y）に送信される後続のパケットに対して LSN マッピングを再使用します。
* **ADDRESS-PORT-DEPENDENT**：マッピングがアクティブな状態で、同じ内部 IP アドレスおよびポート（X: x）から同じ外部 IP アドレスおよびポート（Y: y）に送信される後続のパケットに対して LSN マッピングを再使用します。

設定可能な値: ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT, ADDRESS-PORT-DEPENDENT

デフォルト値: ADDRESS-PORT-DEPENDENT

– filtering

外部ホストから発信されたパケットに適用するフィルタのタイプ。

加入者 IP: ポート（X: x）、NAT IP: ポート（N: n）、および外部ホスト IP: ポート（Y: y）のマッピングを含む LSN マッピングの例を考えてみましょう。

使用可能なオプションは次のように機能します。

* **ENDPOINT INDEPENDENT**：外部ホストの IP アドレスおよびポート送信元（Z: z）に関係なく、加入者 IP アドレスおよびポート X: x を宛先としないパケットだけをフィルタリングします。Citrix ADC アプライアンスは、X: x 宛てのパケットをすべて転送します。つまり、任意の外部ホストから加入者へのパケットを許可するには、加入者から任意の外部 IP アドレスへのパケットを送信するだけで十分です。
* **ADDRESS DEPENDENT**：加入者 IP アドレスおよびポート X: x を宛先としないパケットをフィルタリングします。さらに、クライアントが Y: anyport（外部ポートに依存しない）にパケットを送信していない場合、アプライアンスは Y: y からのサブスクライバ（X: x）宛てのパケットをフィルタリングします。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその特定の外部ホストの IP アドレスにパケットを送信する必要があります。
* **ADDRESS PORT DEPENDENT**（デフォルト）：加入者の IP アドレスおよびポート（X: x）を宛先としないパケットをフィルタリングします。さらに、サブスクライバが Y: y にパケットを送信していない場合、Citrix ADC アプライアンスは、サブスクライバ宛の Y: y (X: x) からのパケットを除外します。つまり、特定の外部ホストからパケットを受信するには、加入者が最初にその外部 IP アドレスおよびポートにパケットを送信する必要があります。

設定可能な値: ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT, ADDRESS-PORT-DEPENDENT

デフォルト値: ADDRESS-PORT-DEPENDENT

- tcpproxy

  TCP プロキシを有効にします。これにより、Citrix ADC アプライアンスはレイヤー 4 機能を使用して TCP トラフィックを最適化できます。

  設定可能な値：ENABLED, DISABLED

  デフォルト値：無効

- td

  LSN の実行後に Citrix ADC アプライアンスが送信トラフィックを送信するトラフィックドメインの ID。

  ID を指定しない場合、アプライアンスは ID が 0 のデフォルトのトラフィックドメインを介してアウトバウンドトラフィックを送信します。

  デフォルト値:65535

  最大値:65535

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- lsnappsprofile をバインドします
  - appsprofilename

    LSN アプリケーションプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN アプリケーションプロファイルの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn application profile1” または’lsn application profile1’）。

    これは必須の引数です。最大長：127

  - lsnport

    加入者からの着信パケットの宛先ポートと照合するポート番号またはポート番号範囲。宛先ポートが一致すると、LSN アプリケーションプロファイルが LSN セッションに適用されます。ポートの範囲はハイフンで区切ります。たとえば、40-90 と指定します。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- add lsn group

– groupname

LSN グループの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn group1” または”lsn group1”）。

これは必須の引数です。最大長：127

– clientname

LSN グループに関連付けられる LSN クライアントエンティティの名前。LSN グループに関連付けることができる LSN クライアントエンティティは 1 つだけです。LSN グループが作成されると、この関連付けを削除したり、別の LSN クライアントエンティティに置き換えたりすることはできません。

これは必須の引数です。最大長：127

– nattype

加入者の NAT IP アドレスおよびポート割り当て（バインドされた LSN プールからの）のタイプ。

使用可能なオプションは次のように機能します。

* **Deterministic**：（LSN グループにバインドされた LSN クライアントの）各加入者に NAT IP アドレスとポートブロックを割り当てます。Citrix ADC アプライアンスは、NAT リソースをこれらのサブスクライバーに順番に割り当てます。Citrix ADC アプライアンスは、開始 NAT IP アドレス上のポートの最初のブロック（LSN グループのポートブロックサイズパラメータによって決定されるブロックサイズ）を、開始サブスクライバ IP アドレスに割り当てます。次の範囲のポートは、次のサブスクライバに割り当てられます。次のサブスクライバに対して十分なポートが NAT アドレスにないまで、次のサブスクライバに割り当てられます。この場合、次の NAT アドレスの最初のポートブロックがサブスクライバに使用されます。各加入者は、Deterministic NAT IP アドレスとポートブロックを受信するようになったため、ロギングを必要とせずに加入者を識別できます。接続の場合、加入者は NAT IP アドレスとポート、および宛先 IP アドレスとポートに基づいてのみ識別できます。
* **Dynamic**：加入者の接続に、ランダムな NAT IP アドレスと LSN NAT プールからポートを割り当てます。（LSN プールで）ポートブロック割り当てが有効で、（LSN グループで）ポートブロックサイズが指定されている場合、Citrix ADC アプライアンスは、サブスクライバが初めて接続を開始するときに、ランダムな NAT IP アドレスとポートのブロックを割り当てます。アプライアンスは、この加入者からの異なる接続に、この NAT IP アドレスとポート（割り当てられたポートブロックから）を割り当てます。サブスクライバに割り当てられたポートブロックからすべてのポートが（異なるサブスクライバ接続に対して）割り当てられている場合、アプライアンスはサブスクライバに新しいランダムポートブロックを割り当てます。

設定可能な値: DYNAMIC, DETERMINISTIC

デフォルト値: DYNAMIC

– portblocksize

各サブスクライバに割り当てる NAT ポートブロックのサイズ。

Dynamic NAT に対してこのパラメーターを設定するには、バインドされた LSN プールでポートブロック割り当てパラメーターを有効にする必要があります。Deterministic NAT の場合、ポートブロック割り当てパラメータは常に有効であり、無効にできません。

Dynamic NAT では、Citrix ADC アプライアンスは、NAT IP アドレスの使用可能な NAT ポートプールからランダムな NAT ポートブロックを各サブスクライバに割り当てます。サブスクライバの場合、サブスクライバに割り当てられたポートブロックからすべてのポートが割り当てられている場合、アプライアンスはサブスクライバに新しいランダムポートブロックを割り当てます。

– logging

この LSN グループに対して作成または削除されたマッピングエントリとセッションのログ。Citrix ADC アプライアンスは、ロギングとセッションロギングの両方のパラメータが有効になっている場合にのみ、この LSN グループの LSN セッションをログに記録します。

アプライアンスは、既存の syslog フレームワークと監査ログフレームワークを使用して LSN 情報を記録します。関連する NSLOG アクションおよび SYLOG アクションエンティティで LSN パラメータを有効にして、グローバルレベルの LSN ロギングを有効にする必要があります。Logging パラメータを有効にすると、Citrix ADC アプライアンスは、この LSN グループの LSN マッピングおよび LSN セッションに関連するログメッセージを生成します。アプライアンスは、NSLOG アクションエンティティおよび SYSLOG アクションエンティティに関連付けられたサーバにこれらのログメッセージを送信します。

LSN マッピングエントリのログメッセージは、次の情報で構成されます。

* Citrix ADC アプライアンスの NSIP アドレス
* タイム・スタンプ
* エントリの種類 (MAPPING または SESSION)
* LSN マッピングエントリが作成または削除されるかどうか
* 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
* NAT IP アドレスとポート
* プロトコル名
* 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が存在する場合があります。
  - エンドポイントに依存しないマッピングの宛先 IP アドレスとポートがログに記録されない
  - アドレス依存マッピングの宛先 IP アドレス（ポート以外）のみがログに記録されます。
  - 宛先 IP アドレスおよびポートは、アドレス-ポート依存マッピングのログに記録されます。

設定可能な値：ENABLED, DISABLED

デフォルト値：無効

– sessionLogging

LSN グループに対して作成または削除されたログセッション。Citrix ADC アプライアンスは、ロギングとセッションロギングの両方のパラメータが有効になっている場合にのみ、この LSN グループの LSN セッションをログに記録します。

LSN セッションのログメッセージは、次の情報で構成されます。

* Citrix ADC アプライアンスの NSIP アドレス
* タイム・スタンプ
* エントリの種類 (MAPPING または SESSION)
* LSN セッションが作成または削除されるかどうか
* 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
* NAT IP アドレスとポート
* プロトコル名
* 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

設定可能な値：ENABLED, DISABLED

デフォルト値：無効

– sessionSync

高可用性 (HA) 展開では、この LSN グループに関連するすべての LSN セッションの情報をセカンダリノードと同期します。フェールオーバー後、確立された TCP 接続と UDP パケットフローはアクティブに保たれ、セカンダリノード（新しいプライマリ）で再開されます。

この設定を有効にするには、グローバルセッション同期パラメータを有効にする必要があります。

設定可能な値：ENABLED, DISABLED

デフォルト値:ENABLED

– snmptraplimit

LSN グループに対して生成できる SNMP トラップメッセージの最大数。

デフォルト値:100

最小値:0

最大値:10000

– ftp

FTP プロトコルのアプリケーション層ゲートウェイ (ALG) を有効にします。アプリケーション層プロトコルによっては、IP アドレスとプロトコルポート番号は通常、パケットペイロードで通信されます。ALG として動作する場合、アプライアンスはパケットのペイロードを変更し、プロトコルが LSN 上で動作し続けるようにします。

注：Citrix ADC アプライアンスには、ICMP プロトコルと TFTP プロトコル用の ALG も含まれています。ICMP プロトコルの ALG はデフォルトで有効になっており、無効にするプロビジョニングはありません。TFTP プロトコルの ALG は、デフォルトで無効になっています。ALG は、エンドポイント独立マ

ッピング、エンドポイント独立フィルタリング、および宛先ポートを 69（TFTP の場合は既知のポート）として、UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、LSN グループに対して自動的に有効になります。

設定可能な値：ENABLED, DISABLED

デフォルト値:ENABLED

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- bind lsn group
  - groupname

    LSN グループの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn group1” または”lsn group1”）。

    これは必須の引数です。最大長：127

  - poolname

    指定された LSN グループにバインドする LSN プールの名前。同じ NAT タイプ設定の LSN プールと LSN グループだけを 1 つにバインドできます。複数 LSN プールを 1 つの LSN グループにバインドできます。

    Deterministic NAT の場合、LSN グループにバインドされたプールは他の LSN グループにバインドできません。Dynamic NAT では、LSN グループにバインドされたプールを複数の LSN グループにバインドできます。最大長：127

  - transportprofilename

    指定された LSN グループにバインドする LSN トランスポートプロファイルの名前。設定を指定するプロトコルごとにプロファイルをバインドします。

    デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのトランスポートと呼ばれます。

    LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルを上書きします。最大長：127

  - appsprofilename

    指定された LSN グループにバインドする LSN アプリケーションプロファイルの名前。宛先ポートのセットごとに、設定を指定するプロトコルごとにプロファイルをバインドします。

デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのアプリケーションプロファイルと呼ばれます。

指定された宛先ポートセットを持つ LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートのセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルを上書きします。最大長：127

LSN 設定の例

October 7, 2021

次に、コマンドラインインターフェイスを使用して LSN を設定する例を示します。

単一の加入者ネットワーク、単一の **LSN NAT IP** アドレス、およびデフォルト設定を使用して、単純な **LSN** 設定を作成します。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```


LSN サブスクライバを識別するための拡張 **ACL** を持つ **LSN** 設定を作成します。

```
add ns acl LSN-ACL-2 ALLOW -srcIP 192.0.2.10-192.0.2.20

Done

apply acls

Done

add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2  -aclname LSN-ACL-2

Done

add lsn pool LSN-POOL-2

Done

bind lsn pool LSN-POOL-2 203.0.113.5-203.0.113.10

Done

add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2

Done

bind lsn group LSN-GROUP-2 -poolname LSN-POOL-2

Done
<!--NeedCopy-->
```

HTTP プロトコルのエンドポイント独立マッピング（ポート **80**）および **SSH** プロトコルのアドレスポート依存マッピング（ポート **22**）を使用して、**LSN** 設定を作成します。また、**TCP** プロトコルには最大 **1000** 個の **NAT** ポート、**UDP** プロトコルには最大 **100** 個の **NAT** ポートを使用するように各加入者を制限します。**TCP** プロトコルに対して最大 **2000** の同時セッションを持つように、各加入者を制限します。**TCP** プロトコルに対して最大 **30000** の同時セッションを持つようにグループを制限します。

```
add lsn client LSN-CLIENT-3
```



```
2
3  Done
4
5  bind lsn client LSN-CLIENT-3 -network 192.0.3.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-3
10
11 Done
12
13 bind lsn pool LSN-POOL-3 203.0.113.11
14
15 Done
16
17 add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-3
18
19 Done
20
21 bind lsn group LSN-GROUP-3 -poolname LSN-POOL-3
22
23 Done
24
25 add lsn appsprofile LSN-APPS-HTTPPROFILE-3 TCP -mapping ENDPOINT-
      INDEPENDENT
26
27 Done
28
29 bind lsn appsprofile LSN-APPS-HTTPPROFILE-3 80
30
31 Done
32
33 bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-HTTPPROFILE
      -3
34
35 Done
36
37 add lsn appsprofile LSN-APPS-SSHPROFILE-3 TCP -mapping ADDRESS-PORT-
      DEPENDENT
38
39 Done
40
41 bind lsn appsprofile LSN-APPS-SSHPROFILE-3 22
42
43 Done
```



```
44
45 bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-SSHPROFILE
       -3
46
47 Done
48
49 add lsn transportprofile LSN-TRANS-PROFILE-TCP-3 TCP -portquota 1000 -
       sessionquota 2000 -groupSessionLimit 30000
50
51 Done
52
53 bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-TCP
       -3
54
55 Done
56
57 add lsn transportprofile LSN-TRANS-PROFILE-UDP-3 UDP -portquota 100
58
59 Done
60
61 bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-UDP
       -3
62
63 Done
64 <!--NeedCopy-->
```

多数の加入者に対して **LSN** 設定を作成します。

```
1  add lsn client LSN-CLIENT-4
2
3  Done
4
5  bind lsn client LSN-CLIENT-4 -network 192.0.4.0 -netmask 255.255.255.0
6
7  Done
8
9  bind lsn client LSN-CLIENT-4 -network 192.0.5.0 -netmask 255.255.255.0
10
11 Done
12
13 bind lsn client LSN-CLIENT-4 -network 192.0.6.0 -netmask 255.255.255.0
14
15 Done
```



```
16
17 bind lsn client LSN-CLIENT-4 -network 192.0.7.0 -netmask 255.255.255.0
18
19 Done
20
21 bind lsn client LSN-CLIENT-4 -network 192.0.8.0 -netmask 255.255.255.0
22
23 Done
24
25 add lsn pool LSN-POOL-4
26
27 Done
28
29 bind lsn pool LSN-POOL-4 203.0.113.30-203.0.113.40
30
31 Done
32
33 bind lsn pool LSN-POOL-4 203.0.113.45-203.0.113.50
34
35 Done
36
37 bind lsn pool LSN-POOL-4 203.0.113.55-203.0.113.60
38
39 Done
40
41 add lsn group LSN-GROUP-4 -clientname LSN-CLIENT-4
42
43 Done
44
45 bind lsn group LSN-GROUP-4 -poolname LSN-POOL-4
46
47 Done
48
49 add lsn appsprofile LSN-APPS-WELLKNOWNPROFILE-4 TCP -mapping ENDPOINT-
      INDEPENDENT
50
51 Done
52
53 bind lsn appsprofile LSN-APPS-WELLKNOWN-PORTS-PROFILE-4 1- 1023
54
55 Done
56
57 bind lsn group LSN-GROUP-4 -applicationprofilename LSN-APPS-WELLKNOWN-
      PORTS-PROFILE-4
58
```



```
Done
<!--NeedCopy-->
```

複数の **LSN** グループ間で **NAT** リソースを共有する **LSN** 設定を作成します。次の例では、**LSN** プール **LSN-**プール **5** が **LSN** グループ **LSN-**グループ **5** および **LSN-**グループ **6** と共有されています。

```
add lsn client LSN-CLIENT-5

Done

bind lsn client LSN-CLIENT-5 -network 192.0.15.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-5

Done

bind lsn pool LSN-POOL-5 203.0.113.12-203.0.113.14

Done

add lsn group LSN-GROUP-5 -clientname LSN-CLIENT-5

Done

bind lsn group LSN-GROUP-5 -poolname LSN-POOL-5

Done

add lsn client LSN-CLIENT-6

Done

bind lsn client LSN-CLIENT-6 -network 192.0.16.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-6

Done

bind lsn pool LSN-POOL-6 203.0.113.15-203.0.113.18
```



```
38
39 Done
40
41 add lsn group LSN-GROUP-6 -clientname LSN-CLIENT-6
42
43 Done
44
45 bind lsn group LSN-GROUP-6 -poolname LSN-POOL-6
46
47 Done
48
49 bind lsn group LSN-GROUP-6 -poolname LSN-POOL-5
50
51 Done
52 <!--NeedCopy-->
```

Deterministic NAT リソース割り当てを使用して **LSN** 設定を作成します。

```
1  add lsn client LSN-CLIENT-7
2
3  Done
4
5  bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
10
11 Done
12
13 bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
14
15 Done
16
17 add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
      DETERMINISTIC -portblocksize 1024
18
19 Done
20
21 bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
22
23 Done
24 <!--NeedCopy-->
```


同じネットワークアドレスを持つ、各ネットワークが異なるトラフィックドメインに属する、複数の加入者ネットワークを持つ **LSN** 設定を作成します。また、**HTTP** プロトコル（ポート **80**）に関連するアウトバウンドトラフィックを制限し、特定のトラフィックドメイン（**td 5**）を介して送信します。

```
add lsn client LSN-CLIENT-8

Done

bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
    -td 1

Done

bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
    -td 2

Done

bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
    -td 3

Done

add lsn pool LSN-POOL-8

Done

bind lsn pool LSN-POOL-8 203.0.113.80-203.0.113.86

Done

add lsn group LSN-GROUP-8 -clientname LSN-CLIENT-8

Done

bind lsn group LSN-GROUP-8 -poolname LSN-POOL-8

Done

add lsn appsprofile LSN-APPS-HTTP-PROFILE-8 TCP -td 5

Done

```



```
bind lsn appsprofile LSN-APPS-HTTP-PROFILE-8 80

Done

bind lsn group LSN-GROUP-8 -applicationprofilename LSN-APPS-HTTP-
    PROFILE-8

Done
<!--NeedCopy-->
```

特定のプロトコル（**TCP**）の発信トラフィックを制限する **LSN** 設定を作成し、特定のトラフィックドメイン（**td 5**）を介して送信します。エンドポイントに依存しないフィルタリングでは、任意のトラフィックドメインでこのプロトコル（**TCP**）に関連するインバウンドトラフィックを受信します。

```
add lsn client LSN-CLIENT-9

Done

bind lsn client LSN-CLIENT-9 -network 192.0.9.0 -netmask 255.255.255.0
    -td 1

Done

add lsn pool LSN-POOL-9

Done

bind lsn pool LSN-POOL-9 203.0.113.90

Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

add lsn appsprofile LSN-APPS-PROFILE-9 TCP -filtering ENDPOINT-
    INDEPENDENT -td 5

Done
```



```
28
29 bind lsn group LSN-GROUP-9 -approfile LSN-APPS-PROFILE-9
30
31 Done
32 <!--NeedCopy-->
```

特定のトラフィックドメイン（**td 10**）を介して送信する **HTTP**（ポート **80**）トラフィックを制限する **LSN** 設定を作成します。アドレス依存フィルタリングでは、指定されたトラフィックドメイン（**td 10**）で、このプロトコル（**HTTP**）に関連するインバウンドトラフィックを受信します。

```
1  add lsn client LSN-CLIENT-10
2
3  Done
4
5  bind lsn client LSN-CLIENT-10 -network 192.0.10.0 -netmask
       255.255.255.0 -td 1
6
7  Done
8
9  add lsn pool LSN-POOL-10
10
11 Done
12
13 bind lsn pool LSN-POOL-10 203.0.113.100
14
15 Done
16
17 add lsn group LSN-GROUP-10 -clientname LSN-CLIENT-10
18
19 Done
20
21 bind lsn group LSN-GROUP-10 -poolname LSN-POOL-10
22
23 Done
24
25 add lsn appsprofile LSN-APPS-PROFILE-10 TCP -mapping ENDPOINT -
       INDEPENDENT -filtering ADDRESS-DEPENDENT -td 10
26
27 Done
28
29 bind lsn appsprofile LSN-APPS-PROFILE-10 80
30
31 Done
```



```
32
33 bind lsn group LSN-GROUP-10 -approfile LSN-APPS-PROFILE-10
34
35 Done
36 <!--NeedCopy-->
```

スタティック **LSN** マップの設定

October 7, 2021

Citrix ADC アプライアンスは、加入者の IP アドレス：ポートと NAT IP アドレス：ポート間の 1 対 1 の LSN マッピングを手動で作成することができます。スタティック LSN マッピングは、NAT IP: Port に開始された接続が、加入者 IP アドレス:Port にマッピングされるようにする場合に便利です。たとえば、内部ネットワークにある Web サーバーなどです。

コマンドラインインターフェイスを使用して静的 **LSN** マッピングを作成するには

コマンドプロンプトで入力します。

```
1 add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
     <positive_integer>] [<natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd
      <positive_integer>]]
2 -  show lsn static
3 <!--NeedCopy-->
```

構成ユーティリティを使用して静的 **LSN** マッピングを作成するには

[システム] > [大規模 NAT] > [スタティック] に移動し、新しいスタティックマッピングを追加します。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

add lsn static name

LSN スタティックマッピングエントリの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は、CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、”lsn static1” または ‘lsn static1’）。これは必須の引数です。最大長：127

transportprotocol

LSN マッピングエントリのプロトコル。これは必須の引数です。指定可能な値：TCP、UDP、ICMP

subscrIP

LSN マッピングエントリの LSN サブスクライバの IPv4 アドレス。これは必須の引数です。

subscrPort

LSN マッピングエントリの LSN サブスクライバのポート。これは必須の引数です。最大値:65535

td

加入者が属するトラフィックドメインの ID。ID を指定しない場合、加入者はデフォルトのトラフィックドメインの一部であると見なされます。既定値:0, 最小値:0, 最大値:4094

natIP

Citrix ADC アプライアンス上にすでに LSN タイプとして存在し、このマッピングエントリの NAT IP アドレスとして使用される IPv4 アドレス。

NatPort

この LSN マッピングエントリの NAT ポート。

destIP

LSN マッピングエントリの宛先 IP アドレス。

dsttd

この LSN マッピングエントリの宛先 IP アドレスが Citrix ADC アプライアンスから到達可能なトラフィックドメインの ID。ID を指定しない場合、宛先 IP アドレスは、ID が 0 のデフォルトトラフィックドメインを介して到達可能であると見なされます。既定値:0, 最小値:0, 最大値:4094

ワイルドカードポートスタティックマップ

スタティックマッピングエントリは、通常、加入者 IP アドレス: ポートと NAT IP アドレス: ポート間の 1 対 1 の LSN マッピングです。1 対 1 のスタティック LSN マッピングエントリは、加入者の 1 つのポートだけをインターネットに公開します。

場合によっては、加入者のすべてのポート（64K）をインターネットに公開する必要があります（たとえば、内部ネットワークでホストされ、各ポートで異なるサービスを実行しているサーバなど）。インターネットを介してこれらの内部サービスにアクセスできるようにするには、サーバーのすべてのポートをインターネットに公開する必要があります。

この要件を満たす方法の 1 つは、64K の 1 対 1 スタティックマッピングエントリ、各ポートに 1 つのマッピングエントリを追加することです。64K エントリの作成は非常に面倒で大きな作業です。また、この多数の構成エントリは、Citrix ADC アプライアンスのパフォーマンスの問題を引き起こす可能性があります。

もう 1 つの簡単な方法は、スタティックマッピングエントリでワイルドカードポートを使用することです。NAT ポートパラメータとサブスクライバポートパラメータをワイルドカード文字（*）に設定し、プロトコルパラメータを ALL に設定して、サブスクライバのすべてのポートをインターネットに公開するスタティックマッピングエントリを 1 つ作成するだけで済みます。ワイルドカードスタティックマッピングエントリと一致するサブスクライバのインバウンド接続またはアウトバウンドコネクションの場合、サブスクライバのポートは NAT 操作後に変更されません。

インターネットへの加入者が開始する接続がワイルドカード静的マッピングエントリと一致する場合、Citrix ADC アプライアンスは、接続を開始する加入者ポートと同じ番号の NAT ポートを割り当てます。同様に、インターネットホストは、加入者のポートと同じ番号を持つ NAT ポートに接続することによって、加入者のポートに接続されます。

IPv4 サブスクライバのすべてのポートにアクセスできるように Citrix ADC アプライアンスを設定する

IPv4 サブスクライバのすべてのポートへのアクセスを提供するように Citrix ADC アプライアンスを構成するには、次の必須パラメータ設定を使用してワイルドカード静的マップを作成します。

- プロトコル =ALL
- サブスクライバポート = *
- NAT ポート = *

ワイルドカードスタティックマップでは、1 対 1 のスタティックマップとは異なり、NAT IP パラメータの設定は必須です。また、ワイルドカードスタティックマップに割り当てられた NAT IP アドレスは、他の加入者には使用できません。

コマンドラインインターフェイスを使用してワイルドカード静的マップを作成するには

コマンドプロンプトで入力します。

```
1 add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
     positive_integer>] [-destIP <ip_addr> [-dsttd <positive_integer>]]
2
3 show lsn static
4 <!--NeedCopy-->
```


構成例

次のワイルドカードスタティックマップの設定例では、IP アドレスが 192.0.2.10 であるサブスクライバのすべてのポートが、NAT IP 203.0.11.33 を介してアクセス可能になります。

設定例:

```
1 add lsn static NAT44-WILDCARD-STATIC-1 ALL  192.0.2.10 * 203.0.113.33 *
2
3 Done
4 <!--NeedCopy-->
```

アプリケーション層ゲートウェイの設定

October 7, 2021

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットのペイロードで通信されます。プロトコルのアプリケーション層ゲートウェイは、パケットのペイロードを解析し、プロトコルが LSN 上で動作し続けるために必要な変更を行います。

Citrix ADC アプライアンスは、次のプロトコルに対して ALG をサポートしています。

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

FTP、**ICMP**、および **TFTP** プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

LSN 設定の LSN グループの FTP オプションを有効または無効にすることで、LSN 設定の FTP プロトコルの ALG を有効または無効にできます。

ICMP プロトコルの ALG はデフォルトで有効になっており、無効にするプロビジョニングはありません。

TFTP プロトコルの ALG は、デフォルトで無効になっています。エンドポイント独立マッピング、エンドポイント独立フィルタリング、および宛先ポートが 69（TFTP の場合は既知のポート）である UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、LSN 設定に対して TFTP ALG が自動的に有効になります。

FTP ALG の **LSN** 設定例:

次の LSN 設定例では、IP アドレスが 192.0.2.30-192.0.2.100 のサブスクライバに対して FTP ALG が有効になっています。

```
add ns acl LSN-ACL-1 ALLOW -srcIP 192.0.2.30-192.0.2.100

Done

apply acls

Done

add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1  - aclname LSN-ACL

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.10

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -FTP ENABLED

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```

TFTP ALG の **LSN** 設定例:

次の LSN 設定例では、エンドポイントに依存しないマッピングおよびエンドポイントに依存しないフィルタリングが TFTP プロトコル（UDP ポート 69）に対して有効になっています。Citrix ADC アプライアンスは、この LSN 構成に対して TFTP ALG を自動的に有効にします。


```
add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2 -network 198.51.100.0 -netmask
    255.255.255.0

Done

add lsn pool LSN-POOL-2

Done

bind lsn pool LSN-POOL-2 203.0.113.10-203.0.113.11

Done

add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2

Done

bind lsn group LSN-GROUP-2 -poolname pool1 LSN-POOL-2

Done

add lsn appsprofile LSNAPPSPROFILE-TFTP-2 UDP -mapping ENDPOINT-
    INDEPENDENT  - filtering  ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile LSNAPPSPROFILE-TFTP-2 69

Done

bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TFTP
    -2

Done
<!--NeedCopy-->
```


PPTP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

Citrix ADC アプライアンスは、ポイントツーポイントトンネリングプロトコル（PPTP）のアプリケーション層ゲートウェイ（ALG）をサポートしています。

PPTP は、TCP/IP ベースのデータネットワーク間でトンネルを作成することによって、リモートクライアントからエンタープライズサーバーへのデータの安全な転送を可能にするネットワークプロトコルです。PPTP は、PPP パケットを IP パケットにカプセル化し、インターネット経由で送信します。PPTP は、通信する PPTP ネットワークサーバ（PNS）と PPTP アクセスコンセントレータ（PAC）のペアごとにトンネルを確立します。トンネルが設定されると、拡張汎用ルーティングカプセル化（GRE）を使用して PPP パケットを交換します。GRE ヘッダーのコール ID は、特定の PPP パケットが属するセッションを示します。

Citrix ADC アプライアンスは、デフォルトの TCP ポート 1723 に着信する PPTP パケットを認識します。アプライアンスは PPTP 制御パケットを解析し、コール ID を変換し、NAT IP アドレスを割り当てます。クライアントとサーバー間の双方向データ通信の場合、Citrix ADC アプライアンスはサーバーコール ID に基づいて LSN セッションエントリを作成し、クライアントコール ID に基づいて LSN セッションを作成します。次に、アプライアンスは GRE データパケットを解析し、2 つの LSN セッションエントリに基づいてコール ID を変換します。

PPTP プロトコルの場合、Citrix ADC アプライアンスには、アイドル状態の PPTP LSN セッションのタイムアウト設定も含まれます。PPTP LSN セッションがタイムアウト設定を超えた時間アイドル状態の場合、Citrix ADC アプライアンスはセッションを削除します。

制限事項:

Citrix ADC アプライアンスでの PPTP ALG の制限事項は次のとおりです。

- PPTP ALG は、ヘアピン LSN フローではサポートされていません。
- PPTP ALG は、どの RNAT 設定でも動作するようにサポートされていません。
- PPTP ALG は、Citrix ADC クラスタではサポートされていません。

PPTP ALG の設定

Citrix ADC アプライアンスで PPTP ALG を構成するには、次の作業を行います。

- LSN 設定を作成し、その上で PPTP ALG を有効にします。LSN 構成では、LSN グループに PPTP ALG 設定が含まれます。LSN 構成の作成手順については、「LSN の設定手順」を参照してください。
- （任意）アイドル状態の PPTP LSN セッションのグローバルタイムアウトを設定します。

CLI を使用して LSN 設定で PPTP ALG を有効にするには

コマンドプロンプトで入力します。


```
add lsn group <groupname> -clientname <string> [-pptp ( ENABLED |
   DISABLED )]

show lsn group
<!--NeedCopy-->
```

CLI を使用してアイドル状態の **PPTP LSN** セッションのグローバルタイムアウトを設定するには

コマンドプロンプトで入力します。

```
set appAlgParam -pptpGreIdleTimeout <positive_integer>

show appAlgParam
<!--NeedCopy-->
```

例**:**

次の LSN 設定例では、192.0.2.0/24 ネットワークの加入者に対して PPTP ALG が有効になっています。

また、アイドル PPTP LSN セッションタイムアウトは 200 秒に設定されます。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -pptp ENABLED

Done

```



```
21 bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
22
23 Done
24
25 set appAlgParam -pptpGreIdleTimeout 200
26
27 Done
28 <!--NeedCopy-->
```

SIP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

SIP メッセージには SIP ヘッダーと SIP 本文に IP アドレスが含まれているため、大規模な NAT（LSN）とセッション開始プロトコル（SIP）の使用は複雑です。LSN を SIP とともに使用すると、SIP ヘッダーには発信者と受信者の情報が含まれ、デバイスはこの情報を変換して外部ネットワークから隠します。SIP 本体には、セッション記述プロトコル（SDP）情報が含まれています。この情報には、メディアを送信するための IP アドレスとポート番号が含まれます。

SIP ALG は、次の RFC に準拠しています。

- RFC 3261
- RFC 3581
- RFC 4566
- RFC 4475

> 注
>
> SIP ALG は、Citrix ADC スタンドアロンアプライアンス、Citrix ADC 高可用性セットアップ、および Citrix ADC クラスタセットアップでサポートされています。

SIP ALG のしくみ

IP アドレス変換の実行方法は、メッセージのタイプと方向によって異なります。メッセージには、次のいずれかを指定できます。

- インバウンドリクエスト
- アウトバウンド応答
- アウトバウンドリクエスト
- インバウンド応答

送信メッセージの場合、SIP クライアントのプライベート IP アドレスとポート番号は、LSN 構成時に指定された Citrix ADC が所有するパブリック IP アドレスとポート番号（*LSN プールの IP アドレスとポート番号*と呼ばれます）

に置き換えられます。着信メッセージの場合、LSN プール IP アドレスとポート番号は、クライアントのプライベートアドレスに置き換えられます。メッセージにパブリック IP アドレスが含まれている場合、Citrix ADC SIP ALG はそのアドレスを保持します。また、次の項目にピンホールが作成されます。

- プライベートクライアントに代わって LSN プール IP アドレスおよびポート。パブリックネットワークからこの IP アドレスおよびポートに到着するメッセージは SIP メッセージとして扱われます。
- パブリッククライアントに代わってパブリック IP アドレスおよびポート。プライベートネットワークからこの IP アドレスおよびポートに到着するメッセージは SIP メッセージとして扱われます。

SIP メッセージがネットワーク経由で送信されると、SIP アプリケーション層ゲートウェイ（ALG）はメッセージから情報を収集し、次のヘッダー内の IP アドレスを LSN プールの IP アドレスに変換します。

- Via
- Contact
- Route
- Record-Route

次の SIP 要求メッセージの例では、LSN はヘッダーフィールドの IP アドレスを置き換えて、外部ネットワークから非表示にします。

```
1 INVITE adam@10.102.185.156 SIP/2.0 Via: SIP/2.0/UDP 192.170.1.161:62914
      From: eve@10.120.210.3 To: adam@10.102.185.156 Call-ID: a12abcde@10
     .120.210.3 Contact: adam@10.102.185.156 Route: <sip:netscreen@10
     .150.20.3:5060> Record-Route: <sip:netscreen@10.150.20.3:5060>
2 <!--NeedCopy-->
```

SDP 情報を含むメッセージが到着すると、SIP ALG はメッセージから情報を収集し、次のフィールドの IP アドレスを LSN プールの IP アドレスおよびポート番号に変換します。

- c= (接続情報)

  このフィールドは、セッションまたはメディアレベルで表示できます。次の形式で表示されます。

  c=<network-type><address-type><connection-address>

  宛先 IP アドレスがユニキャスト IP アドレスの場合、SIP ALG は m= フィールドで指定された IP アドレスとポート番号を使用してピンホールを作成します。

- m=（メディアアナウンス）

  このフィールドはメディアレベルに表示され、メディアの説明が含まれます。次の形式で表示されます。

  m=<media><port><transport><fmt list>

- a=(information about the media field)

  このフィールドは、セッションレベルまたはメディアレベルで次の形式で表示できます。

a=<attribute>

a=<attribute>:<value>

サンプル SDP セクションの次の抜粋は、リソース割り当て用に変換されるフィールドを示しています。

o=user 2344234 55234434 IN IP4 10.150.20.3

c=IN IP4 10.150.20.3

m=audio 43249 RTP/AVP 0

次の表に、SIP ペイロードの変換方法を示します。

インバウンドリクエスト（パブリックからプライベートへ）	変更後:	なし
	From:	なし
	Call-ID:	なし
	Via:	なし
	Request-URI:	LSN プールの IP アドレスをプライベート IP アドレスに置き換える
	Contact:	なし
	Record-Route	なし
	Route:	なし
アウトバウンド応答（プライベートからパブリックへ）	変更後:	なし
	From:	なし
	Call-ID:	なし
	Via:	なし
	Request-URI:	プライベート IP アドレスを LSN プール IP アドレスに置き換える
	Contact:	プライベート IP アドレスを LSN プール IP アドレスに置き換える
	Record-Route	なし
	Route:	なし
アウトバウンドリクエスト（プライベートからパブリックへ）	変更後:	なし

	From:	なし
	Call-ID:	なし
	Via:	プライベート IP アドレスを LSN プール IP アドレスに置き換える
	Request-URI:	なし
	Contact:	プライベート IP アドレスを LSN プール IP アドレスに置き換える
	Record-Route	なし
	Route:	なし
インバウンド応答（パブリックからプライベートへ）	変更後:	なし
	From:	なし
	Call-ID:	なし
	Via:	LSN プールの IP アドレスをプライベート IP アドレスに置き換える
	Request-URI:	なし
	Contact:	パブリック IP アドレスを保持する（存在する場合）
	Record-Route	なし
	Route:	なし

SIP ALG の制限事項

SIP ALG には、次の制限があります。

- SDP ペイロードだけがサポートされます。
- 以下はサポートされていません:
  - マルチキャスト IP アドレス
  - 暗号化 SDP
  - SIP TLS
  - 完全修飾ドメイン名変換
  - SIP レイヤ認証
  - TD/パーティショニング
  - マルチパートボディ

- IPv6 ネットワーク経由の SIP メッセージ
- 折り畳み

テスト済みの **SIP** クライアントとプロキシサーバ

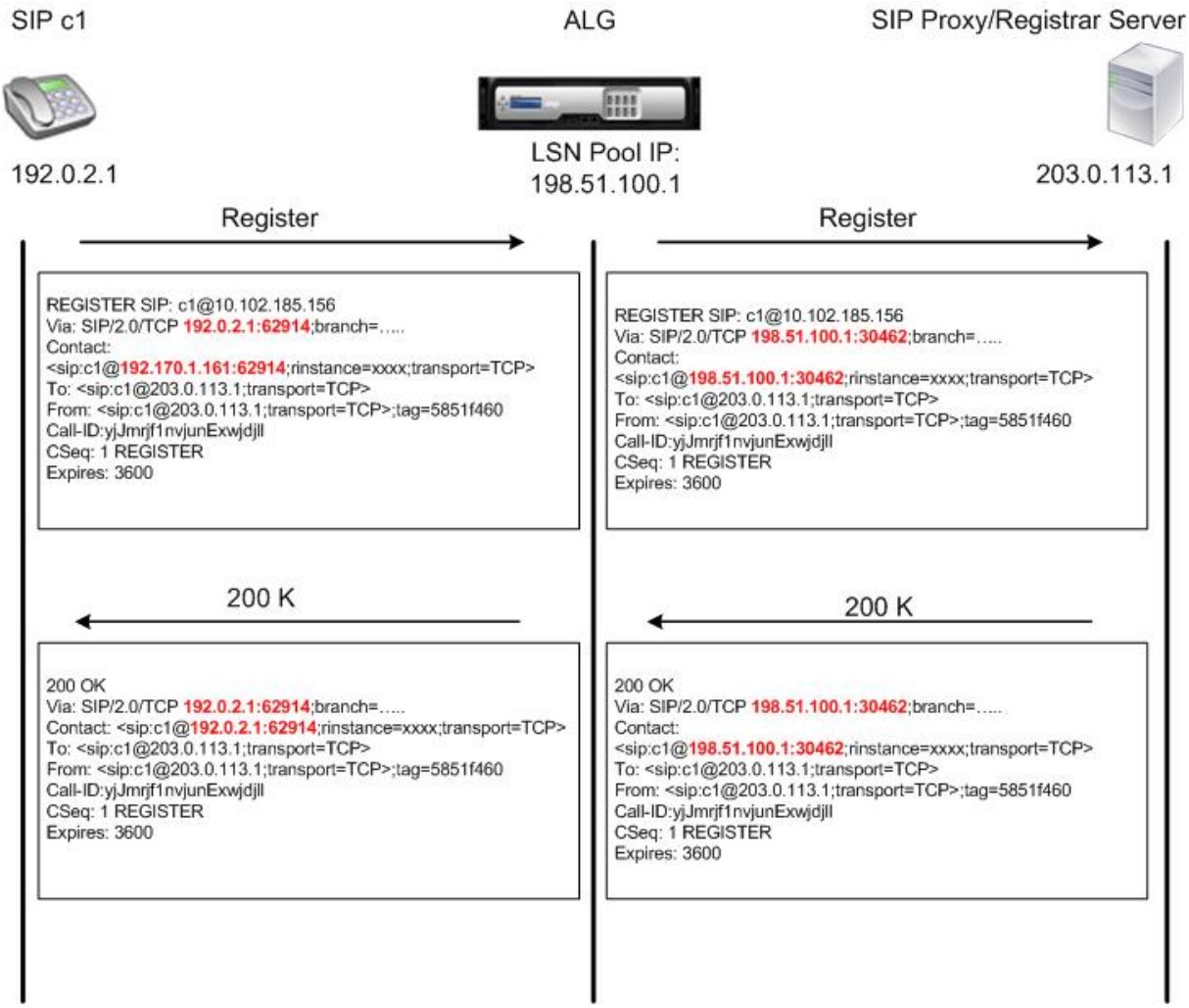
次の SIP クライアントとプロキシサーバは、SIP ALG を使用してテストされています。

- **SIP** クライアント： X-Lite、Zoiper、Ekiga。Avaya
- プロキシサーバー**:** openSIPS

LSN SIP シナリオ：プライベートネットワーク外の **SIP** プロキシ（パブリックネットワーク）

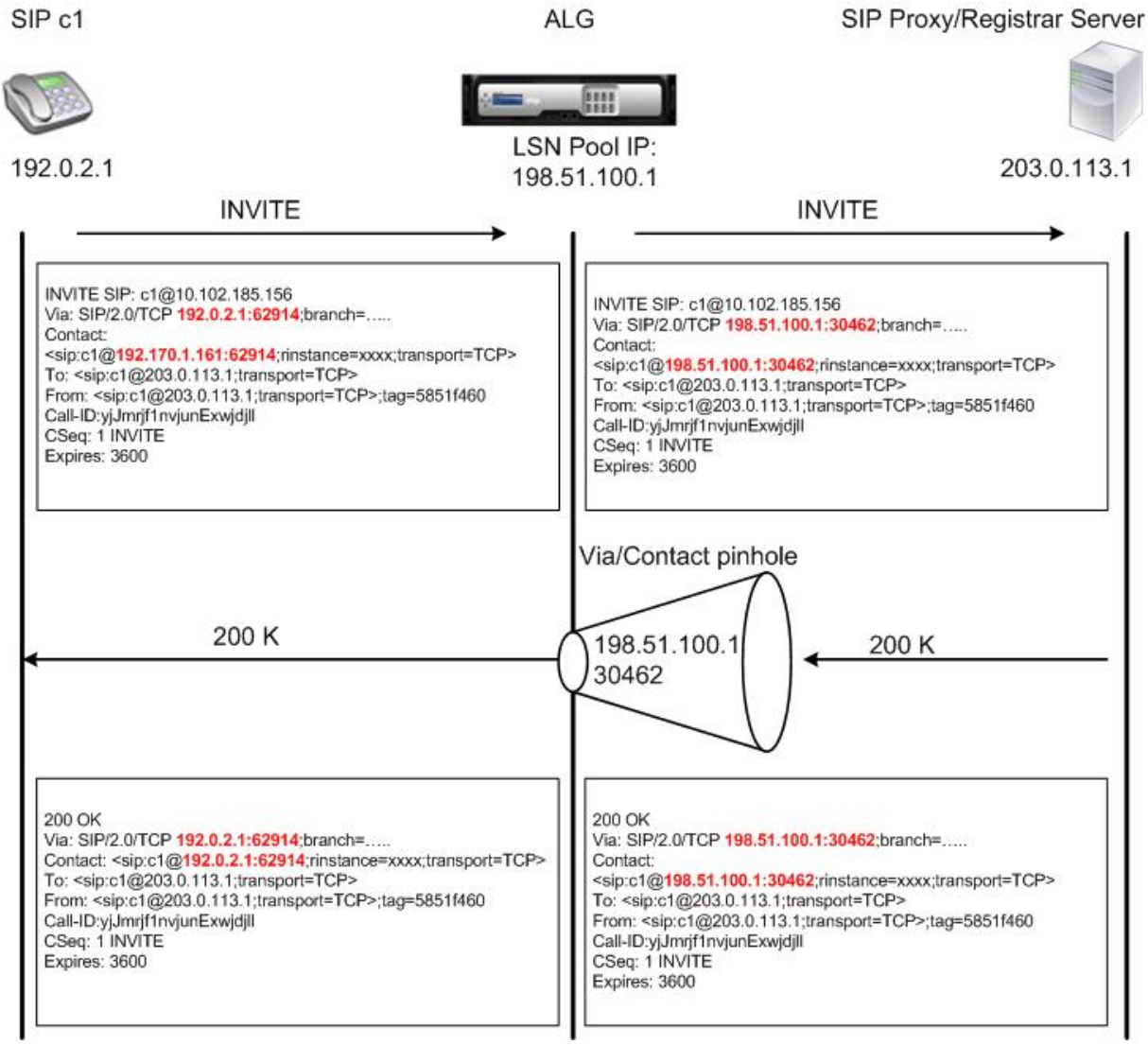
SIP クライアント登録

通常の SIP コールの場合、SIP クライアントは、REGISTER 要求を作成し、SIP レジストラに送信することによって、SIP レジストラに登録する必要があります。Citrix ADC アプライアンスの SIP ALG は要求を代行受信し、要求内の IP アドレスとポート番号を、LSN 構成で提供されている LSN プール IP アドレスとポート番号に置き換え、要求を SIP レジストラに転送します。次に、SIP ALG が Citrix ADC 構成にピンホールを開き、SIP クライアントと SIP レジストラ間のさらなる SIP 通信を可能にします。SIP レジストラは、LSN プールの IP アドレスとポート番号を使用して、SIP クライアントに 200 OK 応答を送信します。Citrix ADC アプライアンスは、この応答をピンホールにキャプチャし、SIP ALG によって SIP ヘッダーが置き換えられ、元の連絡先、経由、および録音ルート SIP フィールドをメッセージに戻します。次に、SIP ALG は SIP クライアントにメッセージを転送します。次の図は、SIP ALG が SIP コール登録フローで LSN を使用する方法を示しています。



発信コール

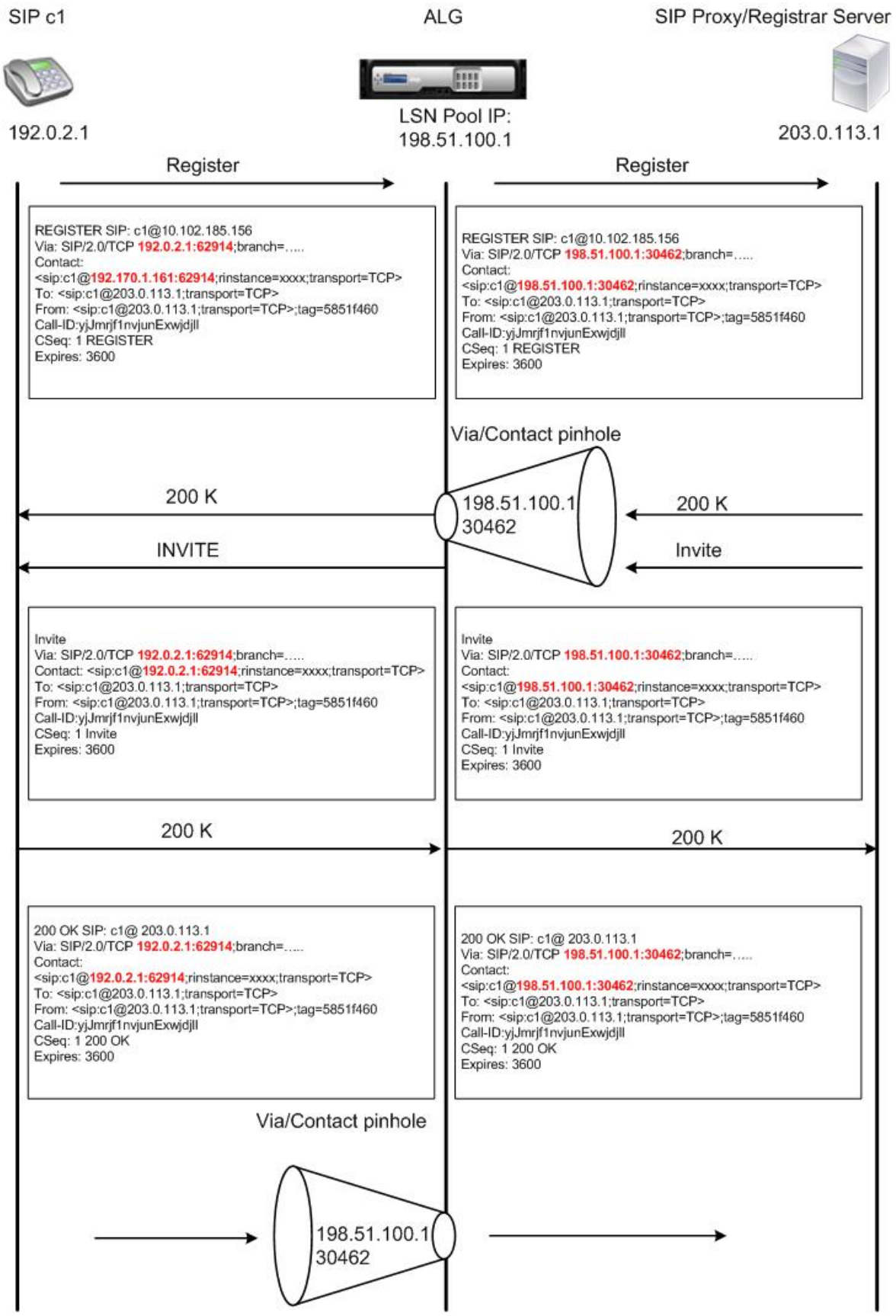
SIP コールは、内部ネットワークから外部ネットワークに送信される SIP INVITE メッセージによって開始されます。SIP ALG は、Via、Contact、Route、および Record-Route SIP ヘッダーフィールドの IP アドレスとポート番号に対して NAT を実行し、LSN プールの IP アドレスとポート番号に置き換えます。LSN は、SIP コールの後続の SIP メッセージ用にこれらのマッピングを保存します。その後、SIP ALG は Citrix ADC 構成で個別のピンホールを開き、SDP ヘッダーと SIP ヘッダーで指定された動的に割り当てられたポート上の Citrix ADC アプライアンスを介して、SIP とメディアを許可します。200 OK メッセージが Citrix ADC に到着すると、作成されたピンホールの 1 つによってそのメッセージがキャプチャされます。SIP ALG は SIP ヘッダーを置き換え、元の Contact、Via、Route、および Record-Route SIP フィールドを復元し、メッセージを内部 SIP クライアントに転送します。



着信コール

SIP 着信コールは、外部クライアントから内部ネットワークへの SIP INVITE メッセージによって開始されます。SIP レジストラは、内部 SIP クライアントが SIP レジストラに登録したときに作成されたピンホールを使用して、内部ネットワーク内の SIP クライアントに INVITE メッセージを転送します。

SIP ALG は、Via、Contact、Route、および Record-Route SIP ヘッダーフィールドの LSN IP アドレスおよびポート番号に対して NAT を実行し、内部 SIP クライアントの IP アドレスおよびポート番号に変換し、要求を SIP クライアントに転送します。内部 SIP クライアントから送信された 200 OK 応答メッセージが Citrix ADC アプライアンスに届くと、SIP ALG は、Via、Contact、Route、およびレコードルート SIP ヘッダーフィールドの IP アドレスとポート番号に対して NAT を実行し、これらを LSN プールの IP アドレスとポート番号に変換し、応答を転送します。メッセージを SIP レジストラに送信し、送信方向にピンホールを開いて、さらの SIP 通信を行います。



コール終了

BYE メッセージは、コールを終了します。デバイスは BYE メッセージを受信すると、他のメッセージの場合と同様に、メッセージ内のヘッダーフィールドを変換します。しかし、BYE メッセージは 200 OK で受信側によって確認応答されなければならないため、ALG は、200 OK の送信時間を許容するために 15 秒間コールティアダダウンを遅延させます。

同じネットワーク内のクライアント間の呼び出し

同じネットワーク内のクライアント A とクライアント B の両方がコールを開始すると、SIP メッセージは外部ネットワークの SIP プロキシを介してルーティングされます。SIP ALG は、クライアント A からの INVITE を通常の発信コールとして処理します。クライアント B は同じネットワーク内にあるため、SIP プロキシは INIVITE を Citrix ADC アプライアンスに送り返します。SIP ALG は、INIVITE メッセージを調べ、クライアント A の NAT IP アドレスが含まれていると判断し、クライアント B にメッセージを送信する前に、そのアドレスをクライアント A のプライベート IP アドレスに置き換えます。クライアント間で通話が確立されると、Citrix ADC はメディア転送に関与しません。クライアント間で。

その他の **LSN SIP** シナリオ：プライベートネットワーク内の **SIP** プロキシ

プライベートネットワーク内で SIP プロキシサーバーをホストする場合は、以下のいずれかを実行することをお勧めします：

- プライベート SIP プロキシのスタティック LSN マッピングを設定します。詳細については、スタティック LSN マップの設定を参照してください。NAT ポートが SIP ALG プロファイルで設定されているポートと同じであることを確認します。
- 非武装地帯（DMZ）内に SIP プロキシサーバを設定します。

図 1：SIP コール登録

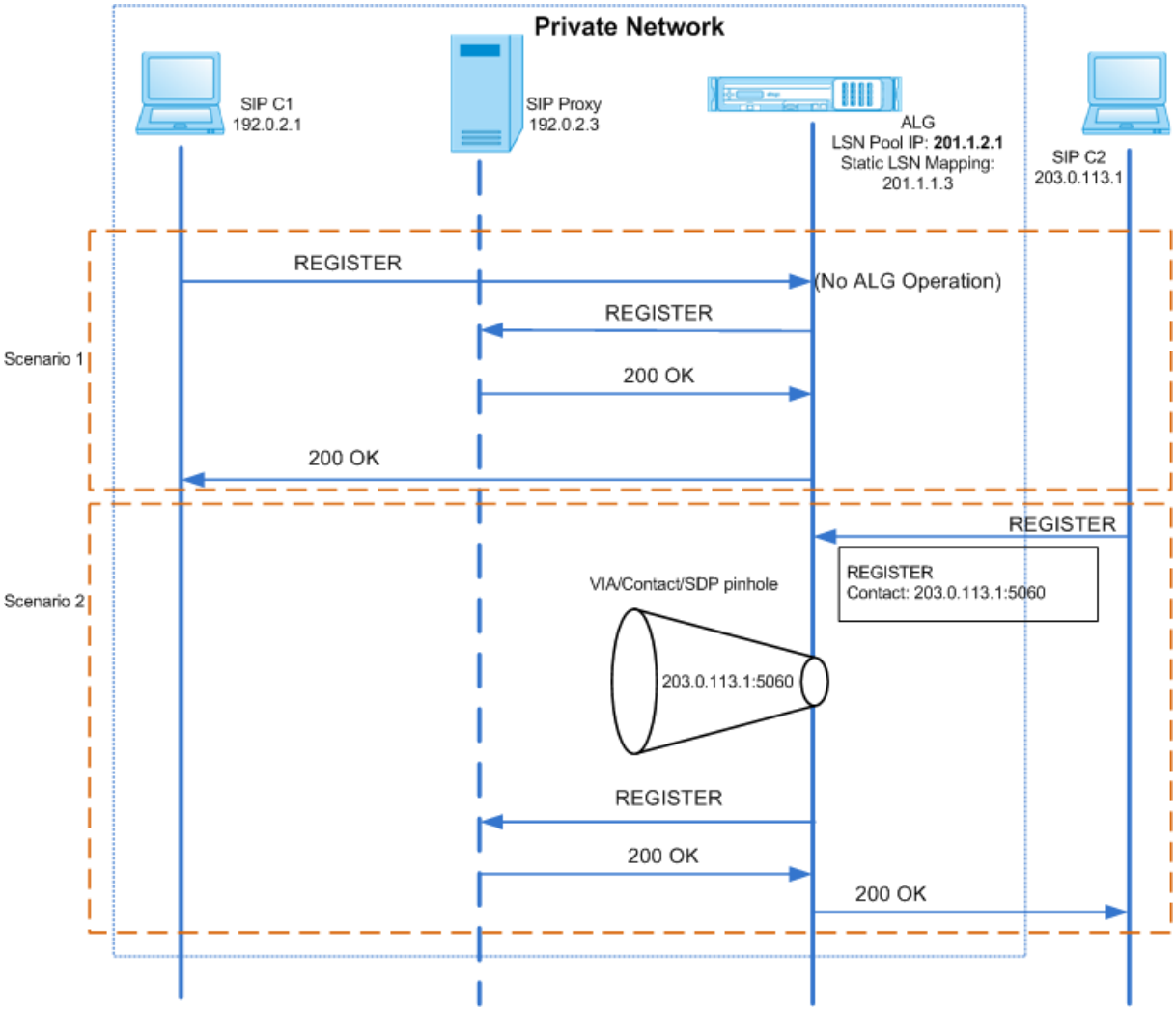


図 2：SIP 着信コールフロー

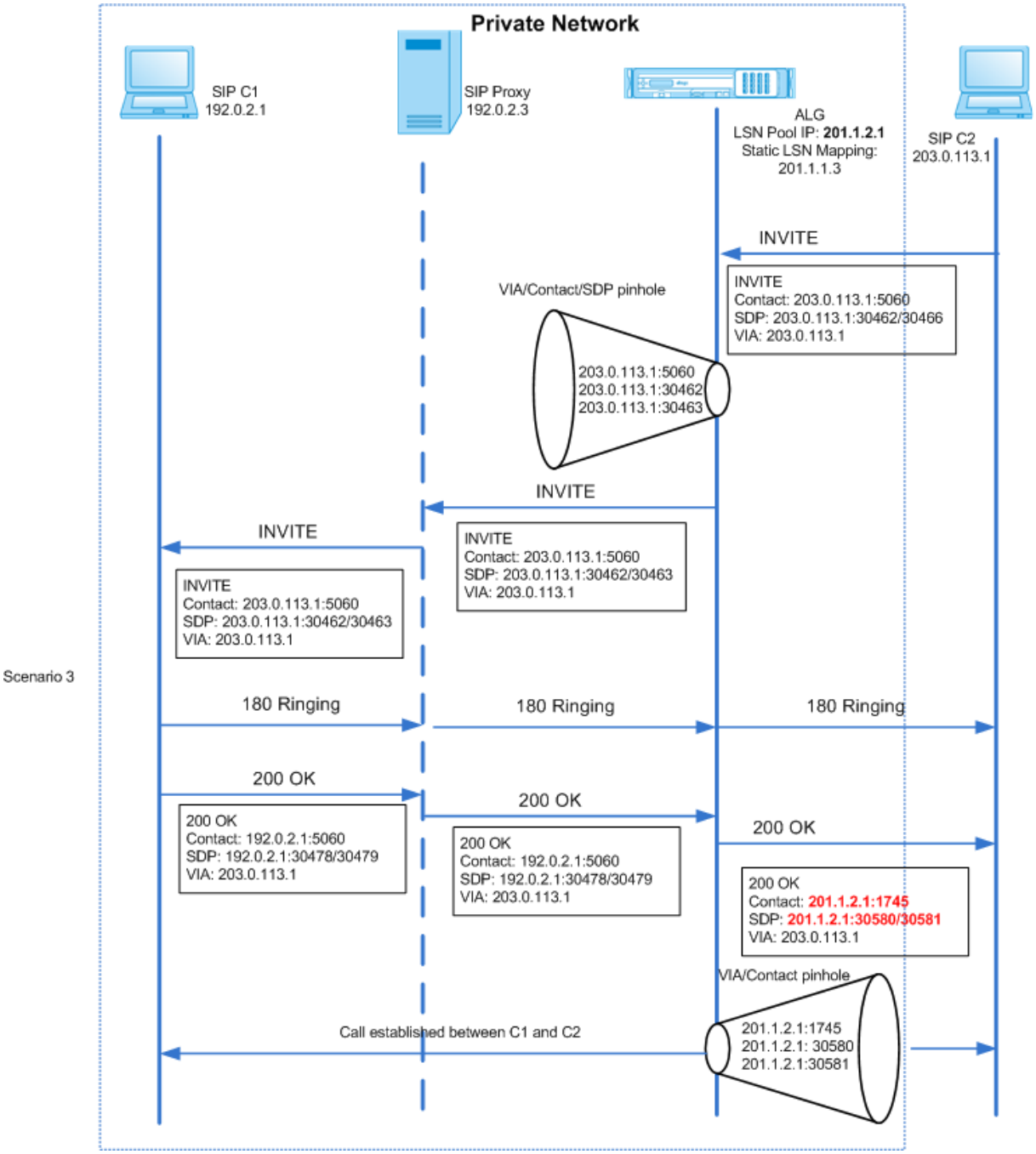


図 1 と 2 は、次のシナリオを示しています。

- シナリオ 1：プライベートネットワーク内の SIP クライアントが、同じネットワーク内の SIP プロキシサーバに登録されます。SIP クライアントと SIP プロキシサーバが同じネットワーク内にあるため、ALG 操作は実行されません。
- シナリオ 2：パブリックネットワーク内の SIP クライアントが、プライベートネットワーク内の SIP プロキシサーバに登録されます。パブリック SIP クライアントからの REGISTER メッセージは、アプライアンス上で構成された静的 LSN マッピングを使用して Citrix ADC アプライアンスに送信され、アプライアンスはさらに

SIP 操作のためのピンホールを作成します。

- シナリオ 3：SIP 着信コールフロー。SIP 着信コールは、外部ネットワークから内部ネットワークへの SIP INVITE メッセージによって開始されます。Citrix ADC アプライアンスは、外部ネットワークにある SIP クライアント C2 から、Citrix ADC アプライアンスで設定された静的 LSN マップを介して、INVITE メッセージを受信します。
  アプライアンスはピンホールを作成し、INVITE メッセージを SIP プロキシに転送します。次に、SIP プロキシは INVITE メッセージを内部ネットワークの SIP クライアント C1 に転送します。その後、SIP クライアント C1 は 180 および 200 の OK メッセージを SIP プロキシに送信します。SIP プロキシは、Citrix ADC アプライアンスを介して SIP クライアント C2 にメッセージを転送します。
  内部 SIP クライアント C1 から送信された 200 OK 応答メッセージが Citrix ADC に到着すると、SIP ALG は Via、Contact、Route、および Record-Route SIP ヘッダーフィールド、および SDP フィールドの IP アドレスとポート番号に対して NAT を実行し、それらを LSN プールの IP アドレスとポート番号に置き換えます。次に、SIP ALG は応答メッセージを SIP クライアント C2 に転送し、発信方向にピンホールを開いて、さらの SIP 通信を行います。

監査ログのサポート

LSN 監査ログ構成で ALG を有効にすると、LSN ログの一部として ALG 情報をログに記録できます。LSN ロギングの詳細については、「LSN のロギングとモニタリング」を参照してください。LSN ログの ALG エントリのログメッセージは、次の情報で構成されます。

- タイム・スタンプ
- SIP メッセージのタイプ（SIP 要求など）
- SIP クライアントの送信元 IP アドレスおよびポート
- SIP プロキシの宛先 IP アドレスおよびポート
- NAT IP アドレスとポート
- SIP メソッド
- シーケンス番号
- SIP クライアントが登録されているかどうか
- 発信者のユーザー名およびドメイン
- 受信者のユーザー名とドメイン

監査ログの例**:**

要求:

```
1 07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 169 0 : Infomsg: "SIP request" - Group: g2
     - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. - Transport
    : TCP - Source_IP: 192.169.1.165 - Source_port: 57952 -
    Destination_IP: 10.102.185.156 - Destination_port: 5060 - Natted_IP:
```



```
        10.102.185.191 - Natted_port: 10313 - Method: REGISTER -
      Sequence_Number: 3060 - Register: YES - Content_Type: -
      Caller_user_name: 156_pvt_1 - Callee_user_name: 156_pvt_1 -
      Caller_domain_name: - Callee_domain_name: -
<!--NeedCopy-->
```

応答:

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 170 0 : Infomsg: "SIP response" - Group:
    g2 - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. -
    Transport: TCP - Response_code 200 - Source_IP: 10.102.185.156 -
    Source_port: 5060 - Destination_IP: 192.169.1.165 - Destination_port
    : 57952 - Natted_IP: 10.102.185.191 - Natted_port: 10313 -
    Sequence_Number: 3060 - Content_Type: - Caller_user_name: 156_pvt_1
    - Callee_user_name: 156_pvt_1 - Caller_domain_name: -
    Callee_domain_name: -
<!--NeedCopy-->
```

SIP ALG の設定

LSN 設定の一部として SIP ALG を設定する必要があります。LSN の設定手順については、「LSN の設定手順」を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加するときに、次のパラメータを設定します。
  - IP プーリング = PAIRED
  - アドレスとポートのマッピング = エンドポイント独在
  - フィルタリング = エンドポイントインデペンデント

重要: SIP ALG が機能するためには、完全なコーンの NAT 設定が必須です。

例:

```
add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT
<!--NeedCopy-->
```

- SIP ALG プロファイルを作成し、送信元ポート範囲または宛先ポート範囲を定義していることを確認します。

例:


```
add lsn sipalgprofile sipalgprofile_tcp -sipsrcportrange 1-65535 -
    sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole
     ENABLED  - sipTransportProtocol TCP
<!--NeedCopy-->
```

- LSN グループの作成中に、SIP ALG = ENABLED を設定します。

例**:**

```
add lsn group g1 -clientname c1 -sipalg ENABLED
<!--NeedCopy-->
```

- SIP ALG プロファイルを LSN グループにバインドします。

SIP ALG 設定の例:

次に、単一加入者ネットワーク、単一の LSN NAT IP アドレス、SIP ALG 固有の設定を使用して、単純な LSN 設定を作成し、SIP ALG を設定する方法の例を示します。

```
add lsn pool p1

Done

bind lsn pool p1 10.102.185.190

Done

add lsn client c1

Done

bind lsn client c1 -network 192.170.1.0 -netmask 255.255.255.0

Done

add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app_udp UDP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

```



```
Done

bind lsn appsprofile app_tcp 1-65535

Done

bind lsn appsprofile app_udp 1-65535

Done

add lsn sipalgprofile sipalgprofile_tcp -sipdstportrange 5060 -
    openViaPinhole ENABLED -openRecordRoutePinhole ENABLED  -
    sipTransportProtocol TCP

Done

add lsn sipalgprofile sipalgprofile_udp -sipdstportrange 5060 -
    openViaPinhole ENABLED -openRecordRoutePinhole ENABLED -
    sipTransportProtocol UDP

Done

add lsn group g1 -clientname c1 -sipalg ENABLED

Done

bind lsn group g1 -poolname p1

Done

bind lsn group g1 -appsprofilename app_tcp

Done

bind lsn group g1 -appsprofilename app_udp

Done

bind lsn group g1 -sipalgprofilename sipalgprofile_tcp

Done

bind lsn group g1 -sipalgprofilename sipalgprofile_udp

Done
```



```
64 <!--NeedCopy-->
```

RTSP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

リアルタイムストリーミングプロトコル (RTSP) は、リアルタイムメディアデータを転送するためのアプリケーションレベルのプロトコルです。RTSP は、エンドポイント間のメディアセッションを確立および制御するために使用され、メディアクライアントとメディアサーバ間のコントロールチャネルプロトコルです。一般的な通信は、クライアントとストリーミングメディアサーバーの間です。

プライベートネットワークからパブリックネットワークにメディアをストリーミングするには、ネットワーク上で IP アドレスとポート番号を変換する必要があります。Citrix ADC の機能には、RTSP 用のアプリケーションレイヤーゲートウェイ（ALG）が含まれています。ALG（大規模 NAT）とともに使用すると、メディアストリームを解析し、プロトコルがネットワーク上で動作し続けるように必要な変更を行うことができます。

IP アドレス変換の実行方法は、メッセージのタイプと方向、およびクライアント/サーバ展開でサポートされるメディアのタイプによって異なります。メッセージは次のように翻訳されます。

- 送信要求：LSN プール IP アドレスと呼ばれる Citrix ADC が所有するパブリック IP アドレスへのプライベート IP アドレス。
- 着信応答：プライベート IP アドレスへの LSN プール IP アドレス。
- インバウンド要求：変換なし。
- 発信応答：LSN プール IP アドレスへのプライベート IP アドレス。

> 注
>
> RTSP ALG は、Citrix ADC スタンドアロンアプライアンス、Citrix ADC 高可用性セットアップ、および Citrix ADC クラスタセットアップでサポートされています。

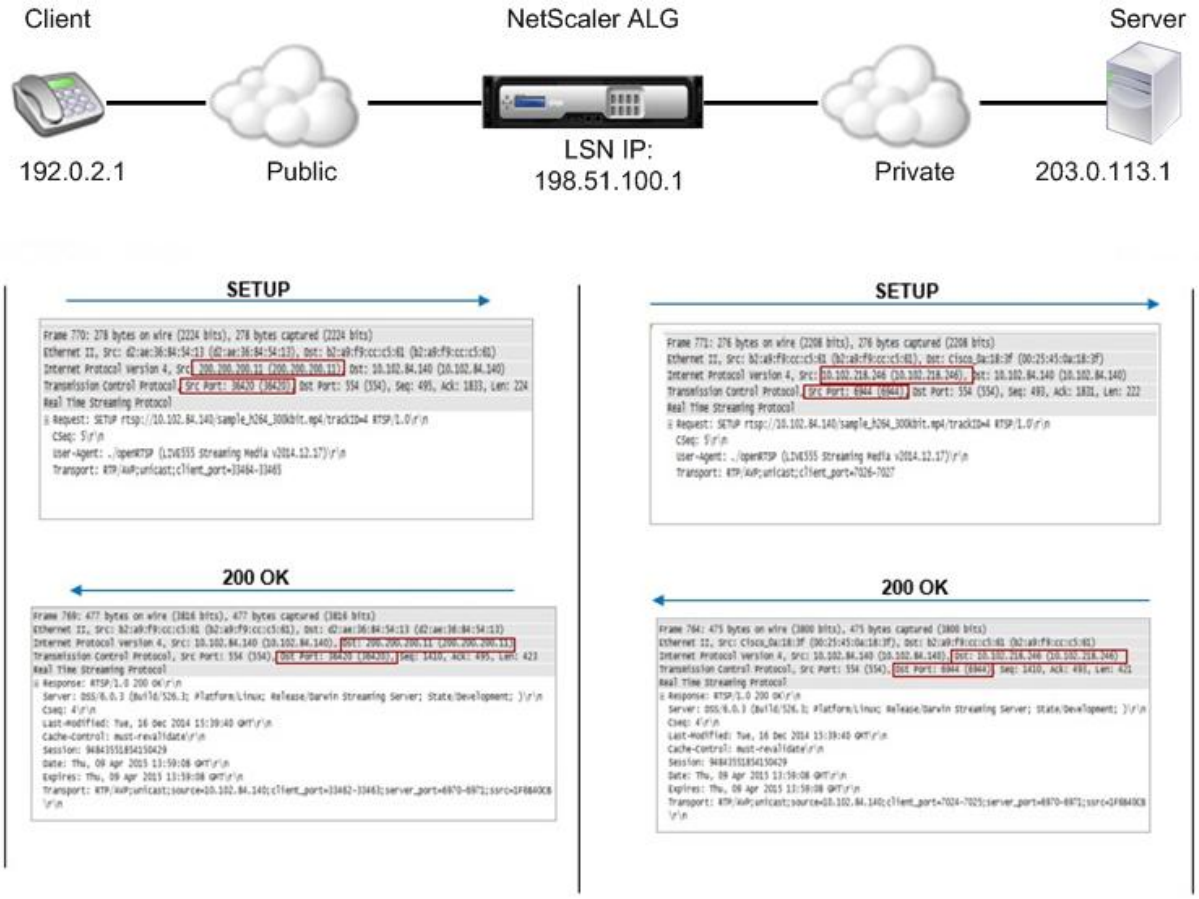
RTSP ALG の制限事項

RTSP ALG は、次の機能をサポートしていません。

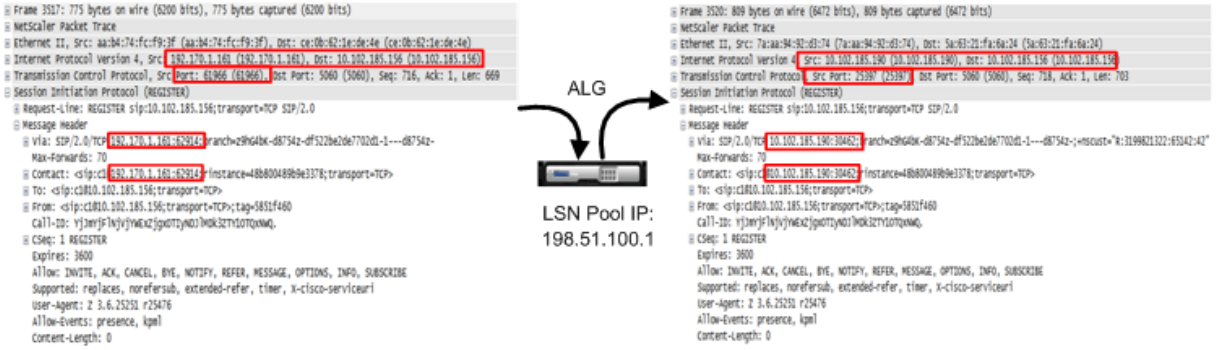
- マルチキャスト RTSP セッション
- UDP を介した RTSP セッション
- TD/admin パーティショニング
- RSTP 認証
- HTTP トンネリング

RTSP および **LSN** のシナリオ

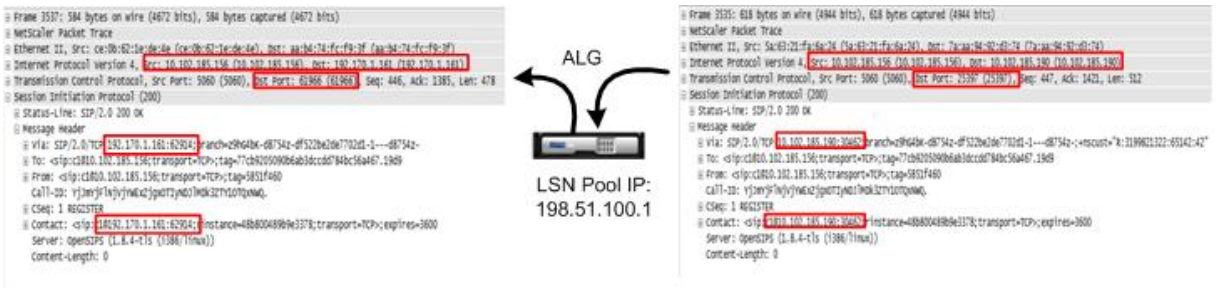
次の図に、RTSP SETUP 要求フローを示します。通常、SETUP 要求は、単一のメディアストリームを転送する方法を指定します。リクエストには、メディアストリームの URL とトランスポート指定子が含まれます。この指定子には、通常、RTP データ（オーディオまたはビデオ）を受信するためのローカルポートと、RTCP データ（メタ情報）を受信するためのローカルポートが含まれます。サーバー応答は、通常、選択されたパラメータを確認し、サーバーの選択されたポートなどの欠落した部分を埋めます。集約再生要求を送信する前に、SETUP コマンドを使用して各メディアストリームを設定する必要があります。



一般的な RTSP 通信では、パブリックネットワークのメディアクライアントがプライベートネットワークのメディアサーバに SETUP 要求を送信します。RSTP ALG は要求を代行受信し、メディアストリーム内でパブリック IP アドレスとポート番号を LSN プール IP アドレスおよび LSN ポート番号に置き換えます。次の図は、送信要求に対してメディアストリーム内の Citrix ADC アプライアンスによって実行される変換を示しています。



プライベートネットワークのメディアサーバは、LSN プールの IP アドレスと LSN ポート番号を使用して、パブリックネットワークのメディアクライアントに 200 OK 応答を送信します。Citrix ADC RTSP ALG は応答を代行受信し、LSN プールの IP アドレスと LSN ポート番号を、メディアクライアントのパブリック IP アドレスとポート番号に置き換えます。次の図は、Citrix ADC アプライアンスが受信応答に対してメディアストリームで実行する変換を示しています。



RTSP ALG の設定

RTSP ALG を LSN 設定の一部として設定します。LSN の設定手順については、「LSN の設定手順」を参照してください。LSN の設定時には、次のことを確認してください。

- LSN プールを追加するときに、**NAT** タイプを DETERMINISTIC または DYNAMIC に設定します。
- LSN アプリケーションプロファイルを追加するときに、次のパラメータを設定します。
  - IP プーリング = PAIRED
  - アドレスとポートのマッピング = エンドポイント独在
  - フィルタリング = エンドポイントインデペンデント
- RTSP ALG プロファイルを作成し、RTSP ALG プロファイルを LSN グループにバインドします。

RTSP ALG 設定の例:

次に、単一加入者ネットワーク、単一の LSN NAT IP アドレス、および RTSP ALG 設定を使用して、単純な LSN 設定を作成する設定例を示します。

```
1 enable ns feature WL SP LB CS LSN
2
```



```
Done

add lsn pool pool1 -nattype DETERMINISTIC

Done

bind lsn pool pool1 10.102.218.246

Done

add lsn client client1

Done

bind lsn client client1 -network 200.200.200.11 -netmask 255.255.255.0

Done

add lsn appsprofile app1 TCP -ippooling PAIRED -mapping ENDPOINT-
   INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app2 UDP -ippooling PAIRED -mapping ENDPOINT-
   INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app1 1-65535

Done

bind lsn appsprofile app2 1-65535

Done

add lsn rtspalgprofile rtspalgprofiledefault -rtspIdleTimeout 1000 -
   rtspportrange 554

Done

add lsn group group1 -clientname client1 -nattype DETERMINISTIC -
   portblocksize 512 -rtspalg ENABLED

Done
```



```
44
45 bind lsn group group1 -poolname pool1
46
47 Done
48
49 bind lsn group group1 -appsprofilename app1
50
51 Done
52
53 bind lsn group group1 -appsprofilename app2
54
55 Done
56
57 bind lsn group group1 -rtspalgprofilename rtspalgprofiledefault
58
59 Done
60 <!--NeedCopy-->
```

IPSec プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

2 つのネットワークデバイス（クライアントとサーバなど）間の通信で IPSec プロトコルが使用されている場合、IKE トラフィック（UDP 経由の）ではポートフィールドが使用されますが、カプセル化セキュリティペイロード（ESP）トラフィックでは使用されません。パス上の NAT デバイスが、同じ宛先の 2 つ以上のクライアントに同じ NAT IP アドレス（ただし、異なるポート）を割り当てた場合、NAT デバイスはポート情報を含まないリターン ESP トラフィックを区別できず、適切にルーティングできません。したがって、NAT デバイスで IPSec ESP トラフィックが失敗します。

NAT-トラバーサル（NAT-T）対応の IPSec エンドポイントは、IKE フェーズ 1 中に中間 NAT デバイスの存在を検出し、後続のすべての IKE および ESP トラフィック（UDP で ESP をカプセル化）に対して UDP ポート 4500 に切り替えます。ピア IPSec エンドポイントで NAT-T がサポートされていない場合、IPSec で保護された ESP トラフィックは UDP カプセル化なしで送信されます。したがって、NAT デバイスで IPSec ESP トラフィックが失敗します。

Citrix ADC アプライアンスは、大規模な NAT 構成用の IPSec アプリケーション層 Gateway（ALG）機能をサポートしています。IPSec ALG は IPSec ESP トラフィックを処理し、セッション情報を維持して、IPSec エンドポイントが NAT-T（ESP トラフィックの UDP カプセル化）をサポートしていない場合にトラフィックに障害が発生しないようにします。

IPSec ALG のしくみ

IPSec ALG は、クライアントとサーバ間の IKE トラフィックを監視し、クライアントとサーバ間の IKE フェーズ 2 メッセージ交換を 1 つだけ許可します。

特定のフローに対して双方向の ESP パケットが受信されると、IPSec ALG はこの特定のフローに対して NAT セッションを作成し、後続の ESP トラフィックがスムーズに流れるようにします。ESP トラフィックは、セキュリティパラメータインデックス（SPI）によって識別されます。SPI は、フローおよび方向ごとに一意です。IPSec ALG は、大規模な NAT を実行するために、送信元ポートと宛先ポートの代わりに ESP SPI を使用します。

ゲートがトラフィックを受信しない場合、ゲートはタイムアウトします。両方のゲートがタイムアウトすると、別の IKE フェーズ 2 交換が許可されます。

IPSec ALG タイムアウト

Citrix ADC アプライアンス上の IPsec ALG には、次の 3 つのタイムアウト・パラメータがあります。

- **ESP** ゲートタイムアウト。クライアントとサーバー間で双方向 ESP トラフィックが交換されない場合、Citrix ADC アプライアンスが、特定のサーバーの特定の NAT IP アドレス上の特定のクライアントの IPSec ALG ゲートをブロックする最大時間。
- **IKE** セッションタイムアウト。Citrix ADC アプライアンスが IKE セッション情報を保持する最大時間。そのセッションに IKE トラフィックがない場合、IKE セッション情報を削除します。
- **ESP** セッションタイムアウト。そのセッションに ESP トラフィックがない場合、Citrix ADC アプライアンスが ESP セッション情報を削除する前に保持する最大時間。

IPSec ALG を設定する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- IPSec プロトコルのさまざまなコンポーネントを理解する必要があります。
- IPSec ALG は、DS-Lite および大規模な NAT64 構成ではサポートされません。
- IPSec ALG は、ヘアピン LSN フローではサポートされません。
- IPSec ALG は、RNAT 設定では機能しません。
- IPsec ALG は、Citrix ADC クラスタではサポートされていません。

構成の手順

Citrix ADC アプライアンスで大規模な NAT44 用の IPSec ALG を構成するには、次の作業を行います。

- **LSN** アプリケーションプロファイルを作成し、**LSN** 構成にバインドします。アプリケーションプロファイルの設定時に、次のパラメータを設定します。
  - Protocol=UDP
  - IP プーリング = PAIRED
  - Port=500

アプリケーションプロファイルを LSN 設定の LSN グループにバインドします。LSN 構成の作成手順については、「LSN の設定手順」を参照してください。

- **IPSec ALG** プロファイルを作成します。IPSec プロファイルには、IKE セッションタイムアウト、ESP セッションタイムアウト、ESP ゲートタイムアウトなど、さまざまな IPSec タイムアウトが含まれます。IPSec ALG プロファイルを LSN グループにバインドするとします。IPSec ALG プロファイルには、次のデフォルト設定があります。
  - IKE セッションタイムアウト = 60 分
  - ESP セッションタイムアウト = 60 分
  - ESP ゲートタイムアウト = 30 秒
- **IPSec ALG** プロファイルを **LSN** 設定にバインドします。IPSec ALG プロファイルを LSN 設定にバインドすると、LSN 設定に対して IPSec ALG が有効になります。IPSec ALG プロファイルパラメータを LSN グループに作成したプロファイルの名前に設定して、IPSec ALG プロファイルを LSN 構成にバインドします。IPSec ALG プロファイルは、複数の LSN グループにバインドできますが、1 つの LSN グループに設定できる IPSec ALG プロファイルは 1 つだけです。

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn appsprofile <appsprofilename> UDP -ippooling PAIRED

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して宛先ポートを **LSN** アプリケーションプロファイルにバインドするには

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを **LSN** グループにバインドするには

コマンドプロンプトで入力します。


```
bind lsn group <groupname> -appsprofilename <string>

show lsn group
<!--NeedCopy-->
```

CLI を使用して **IPSec ALG** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add ipsecalg profile <name> [-ikeSessionTimeout <positive_integer>] [-
    espSessionTimeout <positive_integer>] [-espGateTimeout <
    positive_integer>] [-connfailover ( ENABLED | DISABLED)

show ipsecalg profile <name>
<!--NeedCopy-->
```

CLI を使用して **IPSec ALG** プロファイルを **LSN** 設定にバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -poolname <string> - ipsecAlgProfile <string
    >

show lsn group <name>
<!--NeedCopy-->
```

GUI を使用して **LSN** アプリケーションプロファイルを作成し、**LSN** 構成にバインドするには

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[アプリケーション] タブをクリックし、LSN アプリケーションプロファイルを追加して LSN グループにバインドします。

GUI** を使用して **IPSec ALG** プロファイルを作成するには

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[**IPSEC ALG**] タブをクリックして、IPSec ALG プロファイルを追加します。

GUI** を使用して **IPSec ALG** プロファイルを **LSN** 構成にバインドするには

1. システムに移動 > 大規模 **NAT** > **LSN** グループ、**LSN** グループを開きます。
2. [詳細設定] で、[**+ IPSEC ALG** プロファイル] をクリックして、作成された IPSec ALG プロファイルを LSN グループにバインドします。

構成例

次の大規模な NAT44 設定例では、192.0.2.0/24 ネットワーク内のサブスクライバに対して IPSec ALG が有効になっています。さまざまな IPSec タイムアウト設定を持つ IPSec ALG プロファイル IPSECALGPROFILE-1 が作成され、LSN グループ LSN グループ-1 にバインドされます。

設定例:

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.9

Done

add lsn appsprofile LSN-APPSPROFILE-1 UDP -ippooling PAIRED

Done

bind lsn appsprofile LSN-APPSPROFILE-1 500

Done

add ipsecalg profile  IPSECALGPROFILE-1 -ikeSessionTimeout 45  -
   espSessionTimeout 40  - espGateTimeout 20 -connfailover ENABLED

Done

bind lsn group LSN-GROUP-1 -appsprofilename LSN-APPSPROFILE-1
```



```
30
31 Done
32
33 bind lsn group LSN-GROUP-1 -poolname LSN-POOL-1
34
35 Done
36
37 bind lsn group LSN-GROUP-1 - ipsecAlgProfile IPSECALGPROFILE-1
38
39 Done
40 <!--NeedCopy-->
```

LSN のロギングとモニタリング

October 7, 2021

LSN 情報をログに記録して、問題の診断、トラブルシューティング、法的要件を満たすことができます。LSN 統計カウンタを使用し、現在の LSN セッションを表示することによって、LSN 機能のパフォーマンスを監視できます。

LSN のロギング

LSN 情報のロギングは、法的要件を満たし、任意の時点でトラフィックの送信元を識別するために、ISP が必要とする重要な機能の 1 つです。

Citrix ADC アプライアンスは、LSN マッピングエントリと、LSN グループごとに作成または削除された LSN セッションを記録します。LSN グループのロギングパラメータとセッションロギングパラメータを使用して、LSN グループの LSN 情報のロギングを制御できます。これらはグループレベルのパラメータであり、デフォルトでは無効になっています。Citrix ADC アプライアンスは、ロギングとセッションロギングの両方のパラメータが有効になっている場合にのみ、LSN グループの LSN セッションをログに記録します。

次の表は、ロギングパラメータとセッションロギングパラメータのさまざまな設定に対する LSN グループのロギング動作を示しています。

ログ	セッションロギング	ロギングの動作
有効	有効	LSN マッピングエントリおよび LSN セッションを記録します。
有効	無効	LSN マッピングエントリを記録しますが、LSN セッションは記録しません。

ログ	セッションロギング	ロギングの動作
無効	有効	マッピングエントリも LSN セッションもログに記録しません。

LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）。
- タイム・スタンプ
- エントリの種類 (MAPPING)
- LSN マッピングエントリが作成または削除されたかどうか
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が存在する場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートはログに記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピングのログに記録されます。

LSN セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）。
- タイム・スタンプ
- エントリの種類 (SESSION)
- LSN セッションが作成または削除されるかどうか
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

アプライアンスは、既存の syslog フレームワークと監査ログフレームワークを使用して LSN 情報を記録します。関連する NSLOG アクションおよび SYLOG アクションエンティティで LSN パラメータを有効にして、グローバルレベルの LSN ロギングを有効にする必要があります。Logging パラメータを有効にすると、Citrix ADC アプライアンスは、この LSN グループの LSN マッピングおよび LSN セッションに関連するログメッセージを生成します。アプライアンスは、NSLOG アクションおよび SYSLOG アクションエンティティに関連付けられたサーバにこれらのログメッセージを送信します。

LSN 情報をログに記録するには、次をお勧めします:

- Citrix ADC アプライアンスではなく、外部ログサーバーに LSN 情報を記録します。外部サーバーにログオンすると、アプライアンスが大量の LSN ログエントリを (数百万の順) 作成するときに、最適なパフォーマンスが促進されます。

- TCP または NSLOG 経由での SYSLOG の使用。デフォルトでは、SYSLOG は UDP を使用し、NSLOG は TCP だけを使用してログ情報をログサーバに転送します。TCP は、完全なデータを転送するための UDP よりも信頼性が高い。

  注:

  - Citrix ADC アプライアンスで生成された SYSLOG は、外部ログサーバーに動的に送信されます。
  - SYSLOG over TCP を使用する場合、TCP 接続がダウンしているか、SYSLOG サーバーがビジー状態である場合、Citrix ADC アプライアンスはログをバッファに保存し、接続がアクティブになるとデータを送信します。

ログの構成の詳細については、「監査ログ」を参照してください。

LSN ロギングの設定は、次の作業で構成されます。

- **Citrix ADC** アプライアンスのログ記録の設定このタスクでは、Citrix ADC アプライアンスのさまざまなエンティティとパラメータの作成と設定を行います。
  - **SYSLOG** または **NSLOG** 監査ログ構成を作成します。監査ログ設定の作成には、次の作業が含まれます。
    * NSLOG または SYSLOG 監査アクションを作成し、LSN パラメータを有効にします。監査アクションは、ログサーバーの IP アドレスを指定します。
    * SYSLOG または NSLOG 監査ポリシーを作成し、監査アクションを監査ポリシーにバインドします。監査アクションは、ログサーバーの IP アドレスを指定します。オプションで、外部ログサーバに送信されるログメッセージの転送方法を設定できます。デフォルトでは UDP が選択されており、信頼性の高いトランスポートメカニズムのために TCP としてトランスポート方式を設定できます。監査ポリシーをシステムグローバルにバインドします。
    * SYSLOG または NSLOG 監査ポリシーを作成し、監査アクションを監査ポリシーにバインドします。
    * 監査ポリシーをシステムグローバルにバインドします。
      注: 既存の監査ログ設定では、監査アクションで指定されたサーバーで LSN 情報をログに記録するための LSN パラメータを有効にするだけです。
  - ロギングおよびセッションロギングパラメータを有効にします。LSN グループを追加するか、グループを作成した後で、ロギングパラメータとセッションロギングパラメータを有効にします。Citrix ADC アプライアンスは、これらの LSN グループに関連するログメッセージを生成し、LSN パラメータが有効になっている監査アクションのサーバーに送信します。
- ログサーバの設定。このタスクでは、目的のサーバに SYSLOG パッケージまたは NSLOG パッケージをインストールします。また、このタスクでは、SYSLOG または NSLOG の構成ファイルで Citrix ADC アプライアンスの NSIP アドレスを指定します。NSIP アドレスを指定すると、サーバは Citrix ADC アプライアンスから送信されるログ情報を識別して、ログファイルに保存できます。

ログの構成の詳細については、「監査ログ」を参照してください。

コマンドラインインターフェイスを使用した **SYSLOG** の設定

コマンドラインインターフェイスを使用して **LSN** ログ用の **SYSLOG** サーバーアクションを作成するには

コマンドプロンプトで入力します。

```
add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel>... [-transport (TCP)] [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** ログ用の **SYSLOG** サーバーポリシーを作成するには

コマンドプロンプトで入力します。

```
add audit syslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、**SYSLOG** サーバーポリシーを **LSN** ログ用にシステムグローバルにバインドするには

コマンドプロンプトで入力します。

```
bind system global [<policyName> [-priority <positive_integer>]]
<!--NeedCopy-->
```

構成ユーティリティを使用した **SYSLOG** 構成

構成ユーティリティを使用して **LSN** ログの **SYSLOG** サーバーアクションを構成するには

1. [システム] > [監査] > [**Syslog**] に移動し、[サーバー] タブで新しい監査サーバーを追加するか、既存のサーバーを編集します。
2. LSN ログを有効にするには、[大規模な **NAT** ログ] オプションを選択します。
3. （オプション）TCP を介した SYSLOG を有効にするには、[**TCP** ロギング] オプションを選択します。

構成ユーティリティを使用して **LSN** ログ用の **SYSLOG** サーバーポリシーを構成するには

[システム] > [監査] > [**Syslog**] に移動し、[ポリシー] タブで新しいポリシーを追加するか、既存のポリシーを編集します。

構成ユーティリティを使用して、**SYSLOG** サーバーポリシーを **LSN** ログ用にシステムグローバルにバインドするには

1. [システム] > [監査] > [**Syslog**] に移動します。
2. [ポリシー] タブの [操作] ボックスの一覧で、[グローバルバインド] をクリックして、監査グローバルポリシーをバインドします。

コマンドラインインターフェイスを使用した **NSLOG** の設定

コマンドラインインターフェイスを使用して **LSN** ログ用の **NSLOG** サーバーアクションを作成するには

コマンドプロンプトで入力します。

```
add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> ... [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** ログ用の **NSLOG** サーバーポリシーを作成するには

コマンドプロンプトで入力します。

```
add audit nslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、**NSLOG** サーバポリシーを **LSN** ログ用にシステムグローバルにバインドするには

コマンドプロンプトで入力します。

```
bind system global [<policyName>]
<!--NeedCopy-->
```

構成ユーティリティを使用した **NSLOG** の構成

構成ユーティリティを使用して **LSN** ログ用の **NSLOG** サーバーアクションを構成するには

1. [システム] > [監査] > [**Nslog**] に移動し、[サーバー] タブで新しい監査サーバーを追加するか、既存のサーバーを編集します。
2. LSN ログを有効にするには、[大規模な **NAT** ログ] オプションを選択します。

構成ユーティリティを使用して **LSN** ログ用の **NSLOG** サーバポリシーを構成するには

[システム] > [監査] > [**Nslog**] に移動し、[ポリシー] タブで新しいポリシーを追加するか、既存のポリシーを編集します。

構成ユーティリティを使用して、**NSLOG** サーバポリシーを **LSN** ログ用にシステムグローバルにバインドするには

1. [システム] > [監査] > [**Nslog**] に移動します。
2. [ポリシー] タブの [操作] ボックスの一覧で、[グローバルバインド] をクリックして、監査グローバルポリシーをバインドします。

例

次の構成では、LSN ログを含むログエントリを格納するための 2 つの SYSLOG サーバと 2 つの NSLOG サーバを指定します。LSN ロギングは、LSN グループ LSN-グループ 2 および LSN-グループ 3 に対して設定されます。

Citrix ADC アプライアンスは、これらの LSN グループの LSN マッピングおよび LSN セッションに関連するログメッセージを生成し、指定されたログサーバーに送信します。

```
add audit syslogAction SYS-ACTION-1 198.51.101.10 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done
bind system global SYSLOG-POLICY-1
Done

add audit syslogAction SYS-ACTION-2 198.51.101.20 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-2 ns_true SYS-ACTION-2
Done
bind system global SYSLOG-POLICY-2
Done

add audit nslogAction NSLOG-ACTION-1 198.51.101.30 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-1 ns_true NSLOG-ACTION-1
Done
bind system global NSLOG-POLICY-1
Done
```



```
add audit nslogAction NSLOG-ACTION-2 198.51.101.40 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-2 ns_true NSLOG-ACTION-2
Done
bind system global NSLOG-POLICY-2
Done

add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-2  - logging ENABLED  -
    sessionLogging ENABLED
Done
set lsn group LSN-GROUP-2   - logging ENABLED  - sessionLogging ENABLED
Done
<!--NeedCopy-->
```

次の設定では、TCP を使用して外部 SYSLOG サーバ 192.0.2.10 にログメッセージを送信するための SYSLOG 設定を指定します。

```
add audit syslogAction SYS-ACTION-1 192.0.2.10 -logLevel ALL -transport
    TCP
Done

add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done

bind system global SYSLOG-POLICY-1
Done
<!--NeedCopy-->
```

次の表に、設定されたログサーバに格納されている各タイプの LSN ログエントリの例を示します。これらの LSN ログエントリは、NSIP アドレスが 10.102.37.115 である Citrix ADC アプライアンスによって生成されます。

LSN ログエントリタイプ	サンプルログエントリ
LSN セッションの作成	Local4.Informational 10.102.37.115 08/05/2014:09:59:48 GMT 0-PPE-0 : LSN LSN_SESSION 2581750 : SESSION CREATED Client IP:Port:TD 192.0.2.10: 15136:0, NatIP:NatPort 203.0.113.6: 6234, Destination IP:Port:TD 198.51.100.9: 80:0, Protocol: TCP

LSN ログエントリタイプ	サンプルログエントリ
LSN セッションの削除	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_SESSION 3871790 : SESSION DELETED Client IP:Port:TD 192.0.2.11: 15130:0, NatIP:NatPort 203.0.113.6: 7887, Destination IP:Port:TD 198.51.101.2:80:0, Protocol: TCP
LSN マッピングの作成	Local4.Informational 10.102.37.115 08/05/2014:09:59:47 GMT 0-PPE-0 : LSN LSN_MAPPING 2581580 : EIM CREATED Client IP:Port 192.0.2.15: 14567, NatIP:NatPort 203.0.113.5: 8214, Protocol: TCP
LSN マッピングの削除	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_MAPPING 3871700 : EIM DELETED Client IP:Port 192.0.3.15: 14565, NatIP:NatPort 203.0.113.11: 8217, Protocol: TCP

最小限のロギング

Deterministic LSN 構成とポートブロックを使用した Dynamic LSN 構成により、LSN ログ量が大幅に減少します。この 2 つのタイプの構成では、Citrix ADC アプライアンスは NAT IP アドレスとポートのブロックをサブスクライバに割り当てます。Citrix ADC アプライアンスは、サブスクライバへの割り当て時に、ポートブロックのログメッセージを生成します。Citrix ADC アプライアンスは、NAT IP アドレスとポートブロックが解放されたときにログメッセージも生成します。接続の場合、加入者は、マッピングされた NAT IP アドレスとポートブロックによってのみ識別できます。このため、Citrix ADC アプライアンスは、作成または削除された LSN セッションを記録しません。また、アプライアンスは、セッションに対して作成されたマッピング・エントリーも記録しません。また、マッピング・エントリが削除されることもありません。

Deterministic LSN 設定およびポートブロックを使用した Dynamic LSN 設定の最小ロギング機能は、デフォルトで有効になっており、無効にする規定はありません。つまり、Citrix ADC アプライアンスは、Deterministic LSN 構成とポートブロックを使用した動的 LSN 構成について、最小限のログを自動的に実行します。この機能を無効にするオプションはありません。アプライアンスは、構成されているすべてのログサーバにログメッセージを送信します。

各ポートブロックのログメッセージは、次の情報で構成されます。

- Citrix ADC アプライアンスの NSIP アドレス
- タイム・スタンプ
- エントリのタイプ（DETERMINISTIC 識別型または PORTBLOCK）
- ポート・ブロックが割り当てられているか、解放されているか

- サブスクライバの IP アドレスと割り当てられた NAT IP アドレスとポートブロック
- プロトコル名

Deterministic LSN 設定のための最小限のロギング

IP アドレスが 192.0.17.1、192.0.17.2、192.0.17.3、および 192.0.17.4 の 4 つのサブスクライバに対する単純な Deterministic LSN 設定の例を考えてみましょう。

この LSN 構成では、ポートブロックサイズが 32768 に設定され、LSN NAT IP アドレスプールの IP アドレスは 203.0.113.19 ～203.0.113.23 の範囲にあります。

```
add lsn client LSN-CLIENT-7
Done
bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask
    255.255.255.253
Done
add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
Done
bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
Done
add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
    DETERMINISTIC -portblocksize 32768
Done
bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
Done
<!--NeedCopy-->
```

Citrix ADC アプライアンスは、LSN NAT IP プールから、設定されたポートブロックサイズに基づいて、LSN NAT IP アドレスと各サブスクライバーへのポートのブロックを順番に事前割り当てします。これは、開始 NAT IP アドレス（203.0.113.19）のポートの最初のブロック（1024-33791）を、開始サブスクライバ IP アドレス（192.0.17.1）に割り当てます。次の範囲のポートは、次のサブスクライバに割り当てられます。次のサブスクライバに対して十分なポートが NAT アドレスにないまで、次のサブスクライバに割り当てられます。この時点で、次の NAT IP アドレスの最初のポートブロックがサブスクライバに割り当てられます。アプライアンスは、NAT IP アドレスと、各サブスクライバに割り当てられたポートのブロックを記録します。

Citrix ADC アプライアンスは、これらのサブスクライバーに対して作成または削除された LSN セッションを記録しません。アプライアンスは LSN 設定に関する次のログメッセージを生成します。

```
1) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201453 0 :  Dtrstc ALLOC Client 12.0.0.241,
    NatInfo 50.0.0.2:59904 to 60415
```



```
2) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201454 0 :  Dtrstc ALLOC Client 12.0.0.242,
    NatInfo 50.0.0.2:60416 to 60927
3) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
    NatInfo 50.0.0.2:60928 to 61439
4) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
    NatInfo 50.0.0.2:60928 to 61439
<!--NeedCopy-->
```

LSN 設定を削除すると、割り当てられた NAT IP アドレスとポートのブロックが各加入者から解放されます。アプライアンスは、NAT IP アドレスと各サブスクライバから解放されたポートのブロックを記録します。LSN 設定を削除すると、アプライアンスによってサブスクライバごとに次のログメッセージが生成されます。

```
1) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201706 0 :  Dtrstc FREE Client 12.0.0.238,
    NatInfo 50.0.0.2:58368 to 58879
2) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201707 0 :  Dtrstc FREE Client 12.0.0.239,
    NatInfo 50.0.0.2:58880 to 59391
3) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201708 0 :  Dtrstc FREE Client 12.0.0.240,
    NatInfo 50.0.0.2:59392 to 59903
<!--NeedCopy-->
```

ポートブロックを使用した **Dynamic LSN** 設定の最小ロギング

ネットワーク 192.0.2.0/24 の加入者に対するポートブロックを使用した単純な Dynamic LSN 設定の例を考えてみましょう。この LSN 構成では、ポートブロックサイズが 1024 に設定され、LSN NAT IP アドレスプールが 203.0.113.3-203.0.113.4 の範囲の IP アドレスを持ちます。

```
set lsn parameter -memLimit 4000
Done
add lsn client LSN-CLIENT-1
Done
bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-1
Done
bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
```



```
Done
add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done
bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done
<!--NeedCopy-->
```

Citrix ADC アプライアンスは、サブスクライバが初めてセッションを開始するときに、LSN NAT IP プールから、設定されたポートブロックサイズに基づいて、ランダムな NAT IP アドレスとポートブロックを割り当てます。Citrix ADC は、この加入者に割り当てられた NAT IP アドレスとポートのブロックを記録します。アプライアンスは、このサブスクライバに対して作成または削除された LSN セッションを記録しません。すべてのポートが、加入者の割り当てられたポートブロックから（異なる加入者のセッションに対して）割り当てられている場合、アプライアンスは、追加のセッションのために加入者に新しいランダム NAT IP アドレスとポートブロックを割り当てます。Citrix ADC は、加入者に割り当てられたすべての NAT IP アドレスとポートブロックを記録します。

IP アドレスが 192.0.2.1 の加入者がセッションを開始すると、アプライアンスは次のログメッセージを生成します。ログメッセージには、アプライアンスが NAT IP アドレス 203.0.113.3 とポートブロック 1024-2047 をサブスクライバに割り当てたことが示されます。

```
03/23/2015:00:07:12 GMT  Informational 0-PPE-3 : default LSN
   LSN_PORTBLOCK 106725793 0 :  Portblock ALLOC Client 12.0.2.72,
   NatInfo 203.0.113.3:1024 to 2047, Proto:TCP
<!--NeedCopy-->
```

割り当てられた NAT IP アドレスと割り当てられたポートブロック内のポートの 1 つを使用しているセッションが残っていなければ、割り当てられた NAT IP アドレスとポートのブロックはサブスクライバから解放されます。Citrix ADC は、NAT IP アドレスとポートのブロックがサブスクライバから解放されたことを記録します。割り当てられた NAT IP アドレス（203.0.113.3）および割り当てられたポートブロック（1024-2047）からのポートを使用するセッションが残っていない場合、アプライアンスによって IP アドレス 192.0.2.1 を持つサブスクライバに関する次のログメッセージが生成されます。ログメッセージには、NAT IP アドレスとポートブロックがサブスクライバから解放されたことが示されます。

```
03/23/2015:00:11:09 GMT Informational 0-PPE-3 : default LSN
   LSN_PORTBLOCK 106814342 0 :  Portblock FREE Client 12.0.3.122,
   NatInfo 203.0.113.3: 1024 to 2047, Proto:TC
<!--NeedCopy-->
```


負荷分散 **SYSLOG** サーバ

Citrix ADC アプライアンスは、構成されているすべての外部ログサーバーに SYSLOG イベントとメッセージを送信します。これにより、冗長なメッセージが保存され、システム管理者の監視が困難になります。この問題に対処するため、Citrix ADC アプライアンスは、外部ログサーバー間で SYSLOG メッセージを負荷分散できる負荷分散アルゴリズムを提供し、メンテナンスとパフォーマンスを向上させます。サポートされている負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小接続、最小パケット、監査ログハッシュがあります。

コマンド・ライン・インタフェースを使用した **SYSLOG** サーバのロード・バランシング

サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

```
1 add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
     SYSLOGUDP)> <port>
2 <!--NeedCopy-->
```

負荷分散仮想サーバーを追加し、サービスの種類を SYSLOGTCP または SYSLOGTCP、ロードバランシング方法を AUDITLOGHASH として指定します。

```
1 add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
     <AUDITLOGHASH>]
2 <!--NeedCopy-->
```

負荷分散仮想サーバーにサービスを Bing します。

```
1 Bind lb vserver <name> <serviceName>
2 <!--NeedCopy-->
```

SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つロードバランシングサーバ名を指定します。

```
1 add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
      <logLevel>]
2 <!--NeedCopy-->
```

規則とアクションを指定して SYSLOG ポリシーを追加します。


```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

SYSLOG ポリシーをシステムグローバルにバインドして、ポリシーを有効にします。

```
bind system global <policyName>
<!--NeedCopy-->
```

構成ユーティリティを使用した **SYSLOG** サーバの負荷分散

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   Traffic Management > Services, click Add に移動して SYLOGTCP または SYSLOGUDP をプロトコルとして選択します。

2. 負荷分散仮想サーバーを追加し、サービスの種類を SYSLOGTCP または SYSLOGTCP、ロードバランシング方法を AUDITLOGHASH として指定します。

   [トラフィック管理] > [仮想サーバー] に移動し、[追加] をクリックして、プロトコルとして SYLOGTCP または SYSLOGUDP を選択します。

3. サービスへの負荷分散仮想サーバーにサービスを Bing します。

   負荷分散仮想サーバーにサービスを Bing します。

   [トラフィック管理] > [仮想サーバー] に移動し、仮想サーバーを選択し、ロードバランシング方式で [AUDITLOGHASH] を選択します。

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つロードバランシングサーバ名を指定します。

   [システム] > [監査] に移動し、[サーバー] をクリックし、[サーバー] で [LB Vserver] オプションを選択してサーバーを追加します。

5. 規則とアクションを指定して SYSLOG ポリシーを追加します。

   System > Syslog に移動して Policies をクリックして SYSLOG ポリシーを追加します。

6. SYSLOG ポリシーをシステムグローバルにバインドして、ポリシーを有効にします。

   System > Syslog に移動して SYSLOG ポリシーを選択し、Action、Global Bindings をクリックしてポリシーをシステムグローバルにバインドします。

例**:**

次の構成では、ロードバランシング方式として AUDITLOGHASH を使用して、外部ログサーバ間の SYSLOG メッセージのロードバランシングを指定します。Citrix ADC アプライアンスは、サービス、サービス 1、サービス 2、サービス 3 の間で負荷分散される SYSLOG イベントとメッセージを生成します。


```
add service service1 192.0.2.10 SYSLOGUDP 514
Done

add service service2 192.0.2.11 SYSLOGUDP 514
Done

add service service3 192.0.2.11 SYSLOGUDP 514
Done

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
Done

bind lb vserver lbvserver1 service1
Done

bind lb vserver lbvserver1 service2
Done

bind lb vserver lbvserver1 service3
Done

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
Done

add syslogpolicy syspol1 ns_true sysaction1
Done

bind system global syspol1
Done
<!--NeedCopy-->
```

HTTP ヘッダー情報のロギング

Citrix ADC アプライアンスは、Citrix ADC の LSN 機能を使用している HTTP 接続の要求ヘッダー情報をログに記録できるようになりました。HTTP 要求パケットの次のヘッダー情報をログに記録できます。

- HTTP リクエストの宛先となる URL。
- HTTP リクエストで指定された HTTP メソッド。
- HTTP リクエストで使用される HTTP バージョン。
- HTTP 要求を送信した加入者の IP アドレス。

ISP は、HTTP ヘッダーログを使用して、セット加入者間の HTTP プロトコルに関連する傾向を確認できます。たと

えば、ISP はこの機能を使用して、一連の加入者の中で最も人気のあるウェブサイトを見つけることができます。

HTTP ヘッダーログプロファイルは、ロギングを有効または無効にできる HTTP ヘッダー属性（URL や HTTP メソッドなど）のコレクションです。HTTP ヘッダーログプロファイルは、LSN グループにバインドされます。次に、Citrix ADC アプライアンスは、LSN グループに関連するすべての HTTP 要求のロギング用にバインドされた HTTP ヘッダーログプロファイルで有効になっている HTTP ヘッダー属性をログに記録します。アプライアンスは、構成済みのログサーバにログメッセージを送信します。

HTTP ヘッダーログプロファイルは、複数の LSN グループにバインドできますが、LSN グループには 1 つの HTTP ヘッダーログプロファイルしか設定できません。

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、**HTTP** ヘッダーログプロファイルを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

例

次の LSN 設定の例では、HTTP ヘッダーログプロファイル HTTP-ヘッダー LOG-1 が LSN グループ LSN-GROUP-1 にバインドされています。ログプロファイルでは、すべての HTTP 属性（URL、HTTP 方式、HTTP バージョン、および HOST IP アドレス）がロギング用に有効になっているため、これらの属性はすべて、LSN グループに関連するサブスクライバ（ネットワーク 192.0.2.0/24 内）からの HTTP 要求に対してロギングされます。


```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done

set lsn parameter -memLimit 4000
Done

add lsn client LSN-CLIENT-1
Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done

add lsn pool LSN-POOL-1
Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done

bind lsn group  LSN-GROUP-1  -httphdrlogprofilename HTTP-HEADER-LOG-1
Done
<!--NeedCopy-->
```

Citrix ADC は、LSN 構成例に属する加入者の 1 人が HTTP 要求を送信すると、次の HTTP ヘッダーログメッセージを生成します。

ログメッセージは、IP アドレス 192.0.2.33 を持つクライアントが HTTP メソッド GET および HTTP バージョン 1.1 を使用して、URL example.com に HTTP リクエストを送信していることを示しています。

```
03/19/2015:16:24:04 GMT  Informational 0-PPE-1 : default LSN Message 59
     0 :  ”LSN Client IP:TD 10.102.37.118:0 URL: example.com Host:
    192.0.2.33 Version: HTTP1.1 Method: GET”
<!--NeedCopy-->
```


MSISDN 情報のロギング

モバイルステーション統合サブスクライバディレクトリ番号（MSISDN）は、複数のモバイルネットワークにわたるサブスクライバを一意に識別する電話番号です。MSISDN は、加入者のオペレータを識別する国コードおよび国別宛先コードに関連付けられています。

Citrix ADC アプライアンスは、モバイルネットワークの加入者の LSN ログエントリに MSISDN を含めるように構成できます。LSN ログに MSISDN が存在すると、ポリシーまたは法律に違反したモバイル加入者、または合法的な傍受機関によって情報が要求されるモバイル加入者の迅速かつ正確なバックトレースで管理者が役立ちます。

次の LSN ログエントリの例には、LSN 設定のモバイルサブスクライバからの接続の MSISDN 情報が含まれています。ログエントリには、MSISDN が E 164:5556543210 であるモバイルサブスクライバが、NAT IP: ポート 203.0.113. 3:45195 を介して宛先 IP: ポート 23.0.0. 1:80 に接続されていたことが示されています。

ログエントリタイプ	サンプルログエントリ
LSN セッションの作成	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN マッピングの作成	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN セッションの削除	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP

ログエントリタイプ	サンプルログエントリ
LSN マッピング	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP

LSN ログに **MSISDN** 情報を含めるには、次のタスクを実行します

- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、LSN 設定の LSN ログに MSISDN 情報を含めるかどうかを指定するログサブスクライバ ID パラメータが含まれます。LSN ログプロファイルの作成時に、ログサブスクライバ ID パラメータを有効にします。
- **LSN** ログプロファイルを **LSN** 構成の **LSN** グループにバインドします。作成された LSN ログプロファイル名にログプロファイル名パラメータを設定して、作成された LSN ログプロファイルを LSN 構成の LSN グループにバインドします。大規模な NAT の設定手順については、LSN の設定手順を参照してください。

CLI を使用して **LSN** ログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename -logSubscriberID ( ENABLED |
    DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```


設定例**:**

この LSN 設定の例では、LSN ログプロファイルでログサブスクライバ ID パラメータが有効になっています。プロファイルは LSN グループ LSN-GROUP-9 にバインドされています。MSISDN 情報は、モバイルサブスクライバ（ネットワーク 192.0.2.0/24）からの接続の LSN セッションおよび LSN マッピングログに含まれています。

```
add lsn logprofile  LOG-PROFILE-MSISDN-9   -logSubscriberID ENABLED

Done
add lsn client LSN-CLIENT-9

Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done
add lsn pool LSN-POOL-9

Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done
bind lsn group  LSN-GROUP-9  -logprofilename LOG-PROFILE-MSISDN-9

Done
<!--NeedCopy-->
```

現在の **LSN** セッションの表示

Citrix ADC アプライアンス上の不要な LSN セッションや非効率的な LSN セッションを検出するために、現在の LSN セッションを表示できます。選択パラメータに基づいて、すべてまたは一部の LSN セッションを表示できます。

注：Citrix ADC アプライアンスに 100 万を超える LSN セッションが存在する場合は、選択パラメータを使用してすべてのセッションではなく、選択した LSN セッションを表示することをお勧めします。

コマンドラインインターフェイスを使用した設定

コマンドラインインターフェイスを使用してすべての **LSN** セッションを表示するには

コマンドプロンプトで入力します。

```
show lsn session
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択的な **LSN** セッションを表示するには

コマンドプロンプトで入力します。

```
show lsn session [-clientname <string>] [-network <ip_addr> [-netmask <
    netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
    port>]]
<!--NeedCopy-->
```

例

Citrix ADC 上に存在するすべての LSN セッションを表示するには

```
> show lsn session
        SubscrIP              SubscrPort      SubscrTD              DstIP                    DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10            15136           0             198.51.100.9          80          0       203.0.113.6     6234          TCP     OUT
2.      192.0.2.11            15130           0             198.51.101.2          80          0       203.0.113.6     7887          TCP     OUT
3.      192.0.2.12            16136           0             198.51.100.3          80          0       203.0.113.6     9807          TCP     OUT
4.      192.0.2.13            18148           0             198.51.101.6          80          0       203.0.113.6     4657          TCP     OUT
5.      192.0.2.14            13560           0             198.51.101.7          80          0       203.0.113.7     9341          TCP     OUT
6.      192.0.2.15            14567           0             198.51.100.8          80          0       203.0.113.5     8214          TCP     OUT
7.      192.0.2.15            16890           0             198.51.101.1          80          0       203.0.113.5     8214          TCP     OUT
8.      192.0.2.16            12345           0             198.51.102.9          80          0       203.0.113.5     1678          TCP     OUT
9.      192.0.2.19            19876           0             198.51.103.8          80          0       203.0.113.5     1567          TCP     OUT
10.     192.0.2.20            10989           0             198.51.104.19         80          0       203.0.113.11    1343          TCP     OUT
11.     192.0.3.13            18149           0             198.51.101.61         80          0       203.0.113.11    4653          TCP     OUT
12.     192.0.3.14            13510           0             198.51.101.74         80          0       203.0.113.11    9344          TCP     OUT
13.     192.0.3.15            14565           0             198.51.100.82         80          0       203.0.113.11    8217          TCP     OUT
14.     192.0.3.15            16899           0             198.51.101.12         80          0       203.0.113.11    8219          TCP     OUT
15.     192.0.3.16            12343           0             198.51.102.99         80          0       203.0.113.11    1673          TCP     OUT

Done
```

LSN クライアントエンティティ LSN-CLIENT-2 に関連するすべての LSN セッションを表示するには

```
> show lsn session -clientname LSN-CLIENT-2
        SubscrIP              SubscrPort      SubscrTD              DstIP                    DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10            15136           0             198.51.100.9          80          0       203.0.113.6     6ß234         TCP     OUT
2.      192.0.2.11            15130           0             198.51.101.2          80          0       203.0.113.6     7887          TCP     OUT
3.      192.0.2.12            16136           0             198.51.100.3          80          0       203.0.113.6     9807          TCP     OUT
4.      192.0.2.13            18148           0             198.51.101.6          80          0       203.0.113.6     4657          TCP     OUT
5.      192.0.2.14            13560           0             198.51.101.7          80          0       203.0.113.7     9341          TCP     OUT
6.      192.0.2.15            14567           0             198.51.100.8          80          0       203.0.113.5     8214          TCP     OUT
7.      192.0.2.15            16890           0             198.51.101.1          80          0       203.0.113.5     8214          TCP     OUT
8.      192.0.2.16            12345           0             198.51.102.9          80          0       203.0.113.5     1678          TCP     OUT
9.      192.0.2.19            19876           0             198.51.103.8          80          0       203.0.113.5     1567          TCP     OUT
10.     192.0.2.20            10989           0             198.51.104.19         80          0       203.0.113.11    1343          TCP     OUT

Done
```

NAT IP アドレスとして 203.0.113.5 を使用するすべての LSN セッションを表示するには


```
> show lsn session –natIP 203.0.113.5
SubscrIP                SubscrPort      SubscrTD        DstIP                   DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.15              14567           0               198.51.100.8            80              0       203.0.113.5     8214            TCP     OUT
2.      192.0.2.15              16890           0               198.51.101.1            80              0       203.0.113.5     8214            TCP     OUT
3.      192.0.2.16              12345           0               198.51.102.9            80              0       203.0.113.5     1678            TCP     OUT
4.      192.0.2.19              19876           0               198.51.103.8            80              0       203.0.113.5     1567            TCP     OUT
Done
```

設定ユーティリティを使用した設定

構成ユーティリティを使用してすべての **LSN** セッションまたは選択した **LSN** セッションを表示するには

1. [システム] > [大規模 NAT] > [セッション] に移動し、[NAT44] タブをクリックします。
2. 選択パラメータに基づいて LSN セッションを表示するには、[Search] をクリックします。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- show lsn session
  - clientname

    LSN クライアントエンティティの名前。最大長：127
  - network

    加入者の IP アドレスまたはネットワークアドレス。
  - netmask

    network パラメータで指定された IP アドレスのサブネットマスク。

    デフォルト値：255.255.255.255
  - td

    LSN クライアントエンティティのトラフィックドメイン ID。

    デフォルト値:0

    最小値:0

    最大値:4094
  - natIP

    LSN セッションで使用されるマッピングされた NAT IP アドレス。

LSN 統計情報の表示

LSN 機能に関連する統計情報を表示して、LSN 機能のパフォーマンスを評価したり、問題をトラブルシューティングしたりできます。LSN 機能または特定の LSN グループの統計情報の要約を表示できます。統計カウンタには、Citrix ADC アプライアンスが最後に再起動されてからのイベントが反映されます。Citrix ADC アプライアンスが再起動されると、これらのカウンタはすべて 0 にリセットされます。

コマンドラインインターフェイスを使用してすべての **LSN** 統計情報を表示するには

コマンドプロンプトで入力します。

```
stat lsn
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、指定した **LSN** グループの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat lsn group [<groupname>]
<!--NeedCopy-->
```

例

```
> stat lsn

Large Scale NAT statistics
                                                        Rate(/s)
                                                           Total
LSN TCP Received Packets                                  0
    40
LSN TCP Received Bytes                                    0
    3026
LSN TCP Transmitted Packets                               0
    40
LSN TCP Transmitted Bytes                                 0
    3026
LSN TCP Dropped Packets                                   0
                          0
LSN TCP Current Sessions                                  0
                          0
LSN UDP Received Packets                                  0
                          0
LSN UDP Received Bytes                                    0
                          0
LSN UDP Transmitted Packets                               0
                          0
```



```
14 LSN UDP Transmitted Bytes                              0
                          0
15 LSN UDP Dropped Packets                                0
                          0
16 LSN UDP Current Sessions                               0
                          0
17 LSN ICMP Received Packets                              0
   982
18 LSN ICMP Received Bytes                                0
   96236
19 LSN ICMP Transmitted Packets                           0
                          0
20 LSN ICMP Transmitted Bytes                             0
                          0
21 LSN ICMP Dropped Packets                               0
   982
22 LSN ICMP Current Sessions                              0
                          0
23 LSN Subscribers                                        0
                          1
24
25 Done
26
27 > stat lsn group LSN-GROUP-1
28
29 LSN Group Statistics
30                                                  Rate (/s)
                                                       Total
31 TCP Translated Pkts                                    0
   40
32 TCP Translated Bytes                                   0
   3026
33 TCP Dropped Pkts                                       0
                          0
34 TCP Current Sessions                                   0
                          0
35 UDP Translated Pkts                                    0
                          0
36 UDP Translated Bytes                                   0
                          0
37 UDP Dropped Pkts                                       0
                          0
38 UDP Current Sessions                                   0
                          0
39 ICMP Translated Pkts                                   0
```



```
                          0
ICMP Translated Bytes                                 0
                          0
ICMP Dropped Pkts                                     0
                          0
ICMP Current Sessions                                 0
                          0
Current Subscribers                                   0
                          1

Done
<!--NeedCopy-->
```

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- stat lsn group
  - groupname

    LSN グループの名前。最大長：127
  - detail

    詳細出力（統計情報を含む）を指定します。出力はかなりボリュームがあります。この引数がなければ、出力にはサマリーのみが表示されます。
  - fullValues

    数字と文字列が完全な形式で表示されるように指定します。このオプションを指定しないと、長い文字列が短縮され、大きな数字が省略されます。
  - ntimes

    統計情報を表示する回数（7 秒間隔）。

    デフォルト値:1
  - logFile

    入力として使用するログファイルの名前。
  - clearstats

    静的な/カウンタをクリアする

    設定可能な値:basic、full

コンパクトロギング

LSN 情報のロギングは、ISP が法的要件を満たし、いつでもトラフィックの送信元を特定できるようにするために必要な重要な機能の 1 つです。その結果、ログデータが大量に作成され、ISP はログインフラストラクチャを維持するために多額の投資をする必要があります。

コンパクトログは、イベント名とプロトコル名の短いコードを含む表記変更を使用して、ログサイズを小さくする手法です。たとえば、クライアントの場合は C、作成されたセッションの場合は SC、TCP の場合は T を指定します。コンパクトログでは、ログサイズが平均 40% 削減されます。

次の NAT44 マッピング作成ログエントリの例は、コンパクトロギングの利点を示しています。

Default logging format	02/02/2016:01:1 GMT Informational 0-PPE-2 : default LSN LSN_ADDRPOR 85 0 : A&PDM CREATED Clien-tIP:Port:TD1.1.1. Destina-tionIP:Port:TD2 Protocol: TCP				
Compact logging format	02/02/2016:01:14 GMT Info 0-PE2:default LSN 87 0:A&PDMC	57 1.1.1.1:6500:0	N-8.8.8.9:51066	D-2.2.2.2:80:0	T

構成の手順

コンパクト形式で LSN 情報をロギングするには、次の作業を実行します。

- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、LSN 構成の情報をコンパクト形式で記録するかどうかを指定する Log Compact パラメーターが含まれています。
- **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドします。作成された LSN ログプロファイル名に [ログプロファイル名] パラメーターを設定して、作成された LSN ログプロファイルを LSN 構成の LSN グループにバインドします。この LSN グループのすべてのセッションとマッピングは、コンパクト形式で記録されます。

CLI を使用して **LSN** ログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

設定例:

```
add lsn logprofile  LOG-PROFILE-COMPACT-9 -logCompact ENABLED

Done
add lsn client LSN-CLIENT-9
Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-9
Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4
Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
Done
bind lsn group  LSN-GROUP-9  - logProfileName LOG-PROFILE-COMPACT-9
Done
<!--NeedCopy-->
```


IPFIX ロギング

Citrix ADC アプライアンスは、LSN イベントに関する情報をインターネット・プロトコル・フロー情報エクスポート（IPFIX）形式で IPFIX コレクタで構成されたセットに送信できます。アプライアンスは、既存の AppFlow 機能を使用して、IPFIX 形式の LSN イベントを IPFIX コレクタに送信します。

IPFIX ベースのロギングは、次の大規模な NAT44 関連イベントで使用できます。

- LSN セッションの作成または削除。
- LSN マッピングエントリの作成または削除。
- Deterministic NAT のコンテキストにおけるポートブロックの割り当てまたは割り当て解除。
- Dynamic NAT のコンテキストにおけるポートブロックの割り当てまたは割り当て解除。
- サブスクライバセッションクォータを超えた場合。

IPFIX ロギングを構成する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- Citrix ADC アプライアンスで AppFlow 機能および IPFIX コレクタを構成する必要があります。手順については、「AppFlow 機能の構成」トピックを参照してください。

構成の手順

LSN 情報を IPFIX 形式でロギングするには、次の作業を実行します。

- **AppFlow** 構成で **LSN** ロギングを有効にします。AppFlow 構成の一部として LSN ロギングパラメータを有効にします。
- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、IPFIX 形式のログ情報を有効または無効にする IPFIX パラメータが含まれています。
- **LSN** ログプロファイルを **LSN** 構成の **LSN** グループにバインドします。LSN ログプロファイルを 1 つまたは複数の LSN グループにバインドします。バインドされた LSN グループに関連するイベントは、IPFIX 形式で記録されます。

CLI を使用して **AppFlow** 構成で **LSN** ロギングを有効にするには

コマンドプロンプトで入力します。

```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```


CLI を使用して **LSN** ログプロファイルを作成するにはコマンドプロンプトで

コマンドプロンプトで入力します。

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

GUI を使用して **LSN** ログプロファイルを作成するには

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[ログ] タブをクリックしてログプロファイルを追加します。

GUI を使用して **LSN** ログプロファイルを **LSN** 構成の **LSN** グループにバインドするには

1. [システム] > [大規模 **NAT**] > [**LSN** グループ**]** に移動し、**LSN** グループを開きます。
2. [詳細設定**]** で、 [**+** ログプロファイル] をクリックして、作成したログプロファイルを LSN グループにバインドします。

TCP SYN アイドルタイムアウト

October 7, 2021

SYN アイドル・タイムアウトは、Citrix ADC アプライアンス上で LSN を使用する TCP 接続を確立するためのタイムアウトです。設定したタイムアウト期間内に TCP セッションが確立されない場合、Citrix ADC はそのセッションを削除します。SYN アイドルタイムアウトは、SYN フラッド攻撃に対する保護を提供するのに便利です。LSN 設定では、LSN グループエンティティに SYN アイドルタイムアウト設定が含まれます。

例**:**

次の LSN 設定例では、192.0.2.0/24 ネットワークからの加入者に関連する TCP 接続の SYN アイドルタイムアウトが 30 秒に設定されています。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  - synidletimeout 30

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```

負荷分散設定による **LSN** 設定の上書き

October 7, 2021

デフォルトでは、LSN の設定は、ロードバランシングの設定よりも優先されます。両方の構成に一致するトラフィックの負荷分散構成で大規模ネットワーク (LSN) 構成を上書きするには、Override LSN パラメータを有効にしてネットプロファイルを作成し、このプロファイルを負荷分散構成の仮想サーバーにバインドします。ロードバランシング構成の USNIP または USIP 設定は、LSN 構成の LSN IP アドレスを適用するのではなく、トラフィックに適用されます。

このオプションは、Citrix ADC アプライアンスおよびファイアウォールや最適化デバイスなどの付加価値サービスを含む LSN 展開で便利です。このタイプの展開では、アプライアンス上の LSN 構成がトラフィックに適用される前に、Citrix ADC アプライアンスの入力トラフィックがこれらの付加価値サービスを通過する必要があります。Citrix

ADC アプライアンスが入力トラフィックを付加価値サービスに送信するには、負荷分散構成が作成され、アプライアンスで LSN の上書きが有効になります。負荷分散構成には、負荷分散サービスとして表され、ANY タイプの仮想サーバーにバインドされた付加価値サービスが含まれます。仮想サーバは、付加価値サービスに送信されるトラフィックを識別するためのリッスンポリシーを使用して設定されます。

CLI を使用してネットプロファイルでオーバーライド lsn を有効にするには

ネットプロファイルの追加中にオーバーライド **lsn** を有効にするには、コマンドプロンプトで

```
add netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

ネットプロファイルの追加中にオーバーライド **lsn** を有効にするには、コマンドプロンプトで

```
set netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

GUI を使用してネットプロファイルで lsn の上書きを有効にするには

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. ネットプロファイルの追加または変更時に、**Override LSN** パラメータを設定します。

次の設定例では、ネットプロファイル NETPROFILE-OVERRIDELSN-1 は LSN オプションを上書きし、ロードバランシング仮想サーバー LBVS-1 にバインドされています。

設定例:

```
add netprofile NETPROFILE-OVERRIDELSN-1 -overrideLsn ENABLED

Done

set lb vserver LBVS-1 -netprofile NETPROFILE-OVERRIDELSN-1

Done
<!--NeedCopy-->
```


LSN セッションのクリア

October 7, 2021

不要な LSN セッションや非効率的な LSN セッションを Citrix ADC アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をただちに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、これらの削除されたセッションに関連する後続のパケットもすべてドロップします。Citrix ADC アプライアンスからすべての LSN セッションまたは選択した LSN セッションを削除できます。

コマンドラインインターフェイスを使用してすべての LSN セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session

show lsn session
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択的な LSN セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session [-clientname <string>] [-network <ip_addr> [-netmask
    <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
    port>]]

show lsn session
<!--NeedCopy-->
```

例

Citrix ADC に存在するすべての LSN セッションをクリアする

```
flush lsn session

Done
```



```
4 <!--NeedCopy-->
```

LSN クライアントエンティティ LSN-CLIENT-1 に関連するすべての LSN セッションをクリアします。

```
1 flush lsn session -clientname LSN-CLIENT-1
2
3 Done
4 <!--NeedCopy-->
```

トラフィックドメイン 100 に属する LSN クライアントエンティティ LSN-CLIENT-2 の加入者ネットワーク（192.0.2.0）に関連するすべての LSN セッションをクリアします。

```
1 flush lsn session -clientname LSN-CLIENT-2 -network 192.0.2.0 -
     netmask 255.255.255.0 -td 100
2
3 Done
4 <!--NeedCopy-->
```

構成ユーティリティを使用してすべての **LSN** セッションをクリアするには

[システム] > [大規模 NAT] > [セッション] に移動し、[セッションのフラッシュ] をクリックします。

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

- flush lsn session
  - clientname

    LSN クライアントエンティティの名前。最大長：127
  - network

    加入者の IP アドレスまたはネットワークアドレス。
  - netmask

    network パラメータで指定された IP アドレスのサブネットマスク。

    デフォルト値：255.255.255.255
  - td

    LSN クライアントエンティティのトラフィックドメイン ID。

    デフォルト値:0

最小値:0

最大値:4094

- natIP

  LSN セッションで使用されるマッピングされた NAT IP アドレス。

- NatPort

  LSN セッションで使用されるマッピングされた NAT ポート。

負荷分散 **SYSLOG** サーバ

October 7, 2021

Citrix ADC アプライアンスは、構成されているすべての外部ログサーバーに SYSLOG イベントとメッセージを送信します。これにより、冗長なメッセージが保存され、システム管理者の監視が困難になります。この問題に対処するため、Citrix ADC アプライアンスは、外部ログサーバー間で SYSLOG メッセージを負荷分散できる負荷分散アルゴリズムを提供し、メンテナンスとパフォーマンスを向上させます。サポートされている負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小接続、最小パケット、監査ログハッシュがあります。

コマンド・ライン・インタフェースを使用した SYSLOG サーバのロード・バランシング

コマンドプロンプトで入力します。

サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

```
add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
    SYSLOGUDP)> <port>
<!--NeedCopy-->
```

負荷分散仮想サーバーを追加し、サービスの種類を SYSLOGTCP または SYSLOGTCP、ロードバランシング方法を AUDITLOGHASH として指定します。

```
add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
    <AUDITLOGHASH>]
<!--NeedCopy-->
```

サービスを負荷分散仮想サーバーにバインドします。


```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

1. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つロードバランシングサーバ名を指定します。

```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
<!--NeedCopy-->
```

規則とアクションを指定して SYSLOG ポリシーを追加します。

```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

SYSLOG ポリシーをシステムグローバルにバインドして、ポリシーを有効にします。

```
bind system global <policyName>
<!--NeedCopy-->
```

構成ユーティリティを使用した **SYSLOG** サーバの負荷分散

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   Traffic Management > Services, click Add に移動して SYLOGTCP または SYSLOGUDP をプロトコルとして選択します。

2. 負荷分散仮想サーバーを追加し、サービスの種類を SYSLOGTCP または SYSLOGTCP、ロードバランシング方法を AUDITLOGHASH として指定します。

   [トラフィック管理] > [仮想サーバー] に移動し、[追加] をクリックして、プロトコルとして SYLOGTCP または SYSLOGUDP を選択します。

3. サービスへの負荷分散仮想サーバーにサービスを Bing します。

   負荷分散仮想サーバーにサービスを Bing します。

   [トラフィック管理] > [仮想サーバー] に移動し、仮想サーバーを選択し、ロードバランシング方式で [AUDITLOGHASH] を選択します。

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つロードバランシングサーバ名を指定します。

   [システム] > [監査] に移動し、[サーバー] をクリックし、[サーバー] で [LB Vserver] オプションを選択してサーバーを追加します。

5. 規則とアクションを指定して SYSLOG ポリシーを追加します。

   System > Syslog に移動して Policies をクリックして SYSLOG ポリシーを追加します。

6. SYSLOG ポリシーをシステムグローバルにバインドして、ポリシーを有効にします。

   System > Syslog に移動して SYSLOG ポリシーを選択し、Action、Global Bindings をクリックしてポリシーをシステムグローバルにバインドします。

例**:**

次の構成では、ロードバランシング方式として AUDITLOGHASH を使用して、外部ログサーバ間の SYSLOG メッセージのロードバランシングを指定します。Citrix ADC アプライアンスは、サービス、サービス 1、サービス 2、サービス 3 の間で負荷分散される SYSLOG イベントとメッセージを生成します。

```
add service service1 192.0.2.10 SYSLOGUDP 514

add service service2 192.0.2.11 SYSLOGUDP 514

add service service3 192.0.2.11 SYSLOGUDP 514

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH

bind lb vserver lbvserver1 service1

bind lb vserver lbvserver1 service2

bind lb vserver lbvserver1 service3

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All

add syslogpolicy syspol1 ns_true sysaction1

bind system global syspol1
<!--NeedCopy-->
```

制限事項:

Citrix ADC アプライアンスは、ログサーバー間での SYSLOG メッセージの外部負荷分散仮想サーバーの負荷分散をサポートしていません。

ポート制御プロトコル

October 7, 2021

Citrix ADC アプライアンスは、大規模 NAT（LSN）用のポート制御プロトコル（PCP）をサポートするようになりました。ISP の加入者アプリケーションの多くは、インターネットからアクセスできる必要があります (インターネット経由で監視する IP カメラなど、モノのインターネット (IOT) デバイスなど)。この要件を満たす方法の 1 つは、スタティック大規模 NAT（LSN）マップを作成することです。しかし、非常に多数のサブスクライバの場合、スタティック LSN NAT マップを作成することは実現可能な解決策ではありません。

Port Control Protocol（PCP; ポート制御プロトコル）を使用すると、加入者は、自身または他のサードパーティデバイスに対して特定の LSN NAT マッピングを要求できます。大規模な NAT デバイスは LSN マップを作成し、加入者に送信します。加入者は、インターネット上のリモートデバイスに、加入者に接続できる NAT IP アドレス:NAT ポートを送信します。

通常、アプリケーションは LSN マッピングがタイムアウトしないように、大規模な NAT デバイスに頻繁にキープアライブメッセージを送信します。PCP は、アプリケーションが LSN マッピングのタイムアウト設定を学習できるようにすることで、このようなキープアライブメッセージの頻度を減らすのに役立ちます。これにより、ISP のアクセスネットワークの帯域幅消費とモバイルデバイスのバッテリ消費を削減できます。

PCP はクライアント/サーバーモデルで、UDP トランスポートプロトコル上で動作します。Citrix ADC アプライアンスは、PCP サーバーコンポーネントを実装し、RFC 6887 に準拠しています。

構成の手順

PCP を設定するには、次の作業を実行します。

- （オプション）PCP プロファイルを作成します。PCP プロファイルには、PCP 関連パラメータの設定（たとえば、マッピングおよびピア PCP 要求をリッスンする）が含まれます。PCP プロファイルを PCP サーバーにバインドできます。PCP サーバーにバインドされた PCP プロファイルは、すべての設定を PCP サーバーに適用します。PCP プロファイルは、複数の PCP サーバにバインドできます。デフォルトでは、デフォルトのパラメータ設定を持つ 1 つの PCP プロファイルがすべての PCP サーバーにバインドされます。PCP サーバーにバインドした PCP プロファイルは、そのサーバーのデフォルトの PCP プロファイル設定よりも優先されます。デフォルトの PCP プロファイルには、次のパラメータ設定があります。
  - マッピング: 有効
  - ピア： 有効
  - 最小マップ寿命:120 秒
  - 最大マップ寿命:86400 秒。
  - アナウンス数:10
  - サードパーティ: 無効
- PCP サーバーを作成し、PCP プロファイルをバインドします。Citrix ADC アプライアンス上に PCP サーバーを作成し、PCP 関連の要求とサブスクライバーからのメッセージをリッスンします。サブネット IP（SNIP）

アドレスは、PCP サーバにアクセスするために割り当てる必要があります。デフォルトでは、PCP サーバーはポート 5351 をリッスンします。

- PCP サーバを LSN 設定の LSN グループにバインドします。PCP Server パラメータを設定して、作成した PCP サーバーを指定することで、作成した PCP サーバーを LSN 構成の LSN グループにバインドします。作成された PCP サーバにアクセスできるのは、この LSN グループのサブスクライバだけです。

> 注
>
> 大規模な NAT 設定の PCP サーバは、ACL ルールで識別されるサブスクライバからの要求を処理しません。

CLI を使用して **PCP** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

CLI を使用して **PCP** サーバーを作成するには

コマンドプロンプトで入力します。

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

NAT44 の設定例

次の設定例では、PCP サーバ PCP-SERVER-9 がデフォルトの PCP 設定で LSN グループ LSN-GROUP-9 にバインドされています。PCP-SERVER-9 は、ネットワーク 192.0.2.0/24 の加入者からの PCP 要求を処理します。

設定例:


```
add pcp server PCP-SERVER-9 192.0.3.9

Done

add lsn client LSN-CLIENT-9

Done

bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-9

Done

bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

bind lsn group  LSN-GROUP-9  -pcpServer PCP-SERVER-9

Done
<!--NeedCopy-->
```

クラスタ設定の **LSN44**

October 7, 2021

Citrix ADC クラスターセットアップでは、大規模な NAT44 構成がサポートされます。

ACitrix ADC クラスタは、単一のシステムとして構成および管理される Citrix ADC アプライアンスのグループです。Citrix ADC クラスターは、スケーラビリティと可用性を提供します。クラスターセットアップ内の各 Citrix ADC アプライアンスは、独立した LSN エンティティとして機能し、単一のシステムとして管理されます。

クラスタセットアップの LSN 構成は、スタンドアロンアプライアンスと同じです。ただし、LSN IP アドレスの特定のプールは、一度に 1 つのノードのみによって所有されます。つまり、LSN IP プールエンティティは、特定のノードでスポッティングエンティティとして設定されます。クラスタセットアップのすべてのノードは、特定の LSN IP プールエンティティを持つことができます。LSN セッションに関連するパケットが NAT 操作を実行したのと同じクラスタノードで確実に受信されるようにするには、ポリシーベースのバックプレーン (PBS) ステアリングが構成されています。PBS は、LSN セッションの受信した関連パケットを同じクラスタノードに操縦します。

設定例:

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 -ownerNode 1 203.0.113.3

Done

bind lsn pool LSN-POOL-1 -ownerNode 2 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

add ns acl b1 ALLOW -srcIP = 192.0.2.0-192.0.2.255 -type DFD -dfdhash
    SIP


Done

```



```
apply ns acls -type DFD

Done
<!--NeedCopy-->
```

デュアルスタックライト

October 7, 2021

IPv4 アドレスの不足と、IPv4 よりも IPv6 の利点のために、多くの ISP は、IPv6 インフラストラクチャに移行し始めています。しかし、移行中、ISP は IPv6 とともに IPv4 をサポートし続ける必要があります。これは、パブリックインターネットのほとんどがまだ IPv4 だけを使用し、多くの加入者は IPv6 をサポートしていないためです。

デュアルスタックライト（DS-Lite）は、IPv6 インフラストラクチャを持つ ISP が IPv6 加入者をインターネットに接続するための IPv6 移行ソリューションです。DS-Lite は IPv4-in-IPv6 トンネリングを使用して、加入者の IPv4 パケットを IPv6 アクセスネットワーク上のトンネル経由で ISP に送信します。IPv6 パケットは、加入者の IPv4 パケットを回復するためにカプセル化解除され、NAT アドレスおよびポート変換、およびその他の LSN 関連処理後にインターネットに送信されます。応答パケットは、サブスクライバへの同じパスを通過します。

Citrix ADC アプライアンスは、DS-Lite 展開の AFTR コンポーネントを実装しており、RFC 6333 に準拠しています。

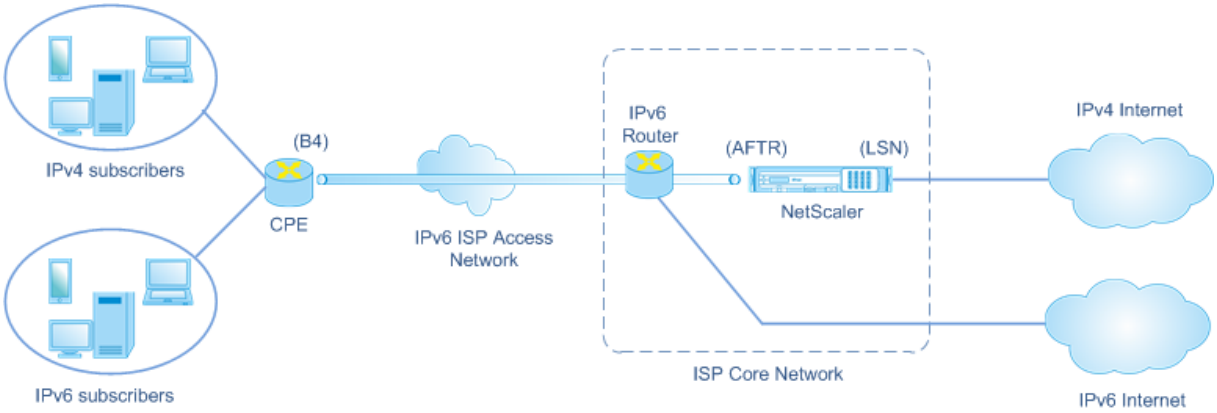
アーキテクチャ

ISP 用のデュアルスタック Lite アーキテクチャは、次のコンポーネントで構成されています。

- ベーシックブリッジングブロードバンド（**B4**）。Basic Bridging ブロードバンド（B4）は、加入者構内にあるデバイスまたはコンポーネントです。通常、B4 は加入者施設内の CPE デバイスのコンポーネントです。IPv4 サブスクライバは、B4 コンポーネントを含む CPE デバイスを介して IPv6 専用の ISP アクセスネットワークに接続されます。B4 の主な機能は、B4 とアドレスファミリ移行ルータ（AFTR）の間の IPv6 トンネルを開始して、トンネル経由で加入者の IPv4 要求パケットまたは応答パケットを送受信することです。B4 には、B4 トンネルエンドポイントアドレスと呼ばれる IPv6 アドレスが含まれます。B4 はこのアドレスを使用して、IPv6 パケットを AFTR に送信し、AFTR からパケットを受信します。
- アドレスファミリ移行ルータ（**AFTR**）。AFTR は、ISP のコアネットワークに存在するデバイスまたはコンポーネントです。AFTR は、B4 デバイスからの IPv6 トンネルを終了します。つまり、IPv6 トンネルは、加入者前提の B4 と ISP コアネットワークの AFTR の間に形成されます。AFTR は、B4 から受信した IPv6 パケットのカプセル化を解除して、加入者の元の IPv4 パケットを回復します。AFTR は、LSN デバイスまたはコンポーネントに IPv4 パケットを送信します。LSN は、NAT アドレスおよびポート変換（NAT 44）およびその他の LSN 関連処理を実行した後、IPv4 パケットを宛先にルーティングします。AFTR には、AFTR トンネルエンドポイントアドレスと呼ばれる IPv6 アドレスが含まれます。AFTR は、このアドレスを使用して、IPv6

パケットを B4 に送信し、B4 から IPv6 パケットを受信します。Citrix ADC アプライアンスは、AFTR コンポーネントを実装します。

- ソフトウェアです。B4 と AFTR の間に作成された IPv6 トンネルは、ソフトウェアと呼ばれます。



Citrix ADC アプライアンスを使用する ISP の DS-Lite アーキテクチャは、ISP のコアネットワークに展開された Citrix ADC アプライアンスを介してインターネットにアクセスするプライベートアドレス空間のサブスクライバで構成されています。IPv4 サブスクライバは、DS-Lite B4 機能を含む CPE デバイスに接続されます。CPE デバイスは、ISP の IPv6 専用アクセスネットワークを介して ISP コアネットワークに接続されています。Citrix ADC アプライアンスには、DS-Lite AFTR 機能および LSN 機能が含まれています。

CPE デバイスに接続された IPv4 サブスクライバには、手動で、または CPE デバイス上で実行されている DHCP サーバを介してプライベート IPv4 アドレスが割り当てられます。CPE デバイスでは、AFTR トンネルエンドポイントアドレスは手動で、または DHCPv6 を介して指定されます。CPE デバイスの設定はベンダー固有であるため、このマニュアルの範囲外です。

IPv4 サブスクライバからインターネット上の場所を宛先とする要求パケットを受信すると、CPE デバイスの B4 コンポーネントは IPv6 パケットに IPv4 パケットをカプセル化し、ISP コアネットワーク内の Citrix ADC アプライアンスに送信します。Citrix ADC アプライアンスの AFTR 機能は、IPv6 パケットのカプセル化を解除して、加入者の元の IPv4 パケットを回復します。Citrix ADC アプライアンスの LSN 機能は、IPv4 パケットの送信元 IP アドレスとポートを、構成済みの NAT プールから選択された NAT IP アドレスと NAT ポートに変換し、そのパケットをインターネット上の宛先に送信します。

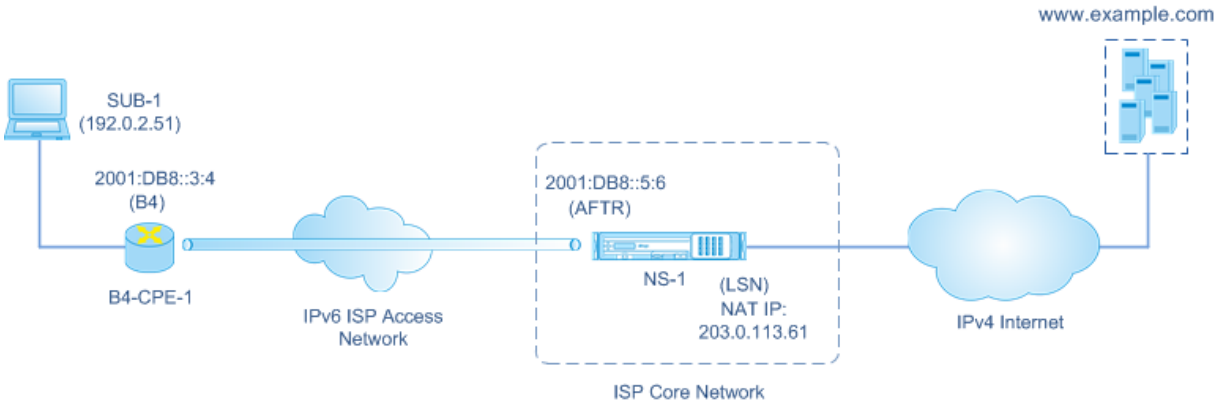
アプライアンスは、AFTR 機能および LSN 機能を使用するすべてのアクティブ・セッションの記録を保持します。これらのセッションを DS-Lite セッションと呼びます。Citrix ADC アプライアンスは、DS-Lite セッションごとに、B4 IPv6 アドレス、サブスクライバ IPv4 アドレスとポート、NAT IPv4 アドレスとポートのマッピングも維持します。これらのマッピングは DS-Lite LSN マッピングと呼ばれます。DS-Lite セッションエントリと DS-Lite LSN マッピングエントリから、Citrix ADC アプライアンスは、（インターネットから受信した）応答パケットを特定の DS-Lite セッションに属するものとして認識します。

Citrix ADC アプライアンスが特定の DS-Lite セッションに属する応答パケットを受信すると、アプライアンスの LSN 機能によって、応答パケットの宛先 IP アドレスとポートが NAT IP アドレスとポートから加入者の IP アドレスとポートに変換され、AFTR 機能によってその結果、パケットが IPv6 パケットに格納され、CPE デバイスに送信されます。CPE デバイスの B4 機能は、IPv6 パケットのカプセル化を解除して IPv4 応答パケットを回復し、IPv4 パ

ケットをサブスクライバに送信します。

例

ISP のコアネットワーク内の Citrix ADC NS-1、加入者前提内の CPE デバイス B4-CPE-1、および単一 IPv4 加入者 SUB-1 で構成される DS-Lite 展開の例を考えてみましょう。B4-CPE-1 は、DS-Lite 機能の B4 機能をサポートします。



次の表に、この例で使用される設定を示します。

エンティティ	Name	詳細
サブスクライバ SUB-1 の IPv4 アドレス		192.2.51
B4 デバイス上のソフトウェアエンドポイントの IPv6 アドレス (B4-CPE-1)		2001:DB8::3:4
AFTR デバイス上のソフトウェア・エンドポイントの IPv6 アドレス (NS-1)		2001:DB8::5:6

Citrix ADC アプライアンス **NS-1** での設定:

エンティティ	Name	詳細
LSN クライアント	LSN-DSLITE-CLIENT-1	Network6 (Identifying traffic from B4 devices) = 2001:DB8::3:0/100
LSN pool	LSN-DSLITE-POOL-1	LSN IPs (NAT IP) = 203.0.113.61 - 203.0.113.70

エンティティ	Name	詳細
IPv6 Profile	LSN-DSLITE-PROFILE-1	Type = DS-LITE; IPv6 address (AFTR IPv6 address) = One of the Citrix ADC owned IPv6 address of type SNIP6 = 2001:DB8::5:6
LSN group	LSN-DSLITE-GROUP-1	LSN client = LSN-DSLITE-CLIENT-1; LSN pool = LSN-DSLITE-POOL-1;IPv6 profile = LSN-DSLITE-PROFILE-1

次に、この例のトラフィックフローを示します。

1. IPv4 サブスクライバ SUB-1 が（`http://www.example.com/`）に要求を送信します。IPv4 パケットには、次のものがあります。
   - 送信元 IP アドレス = 192.0.2.51
   - 送信元ポート = 2552
   - 宛先 IP アドレス = 198.51.100.250
   - 宛先ポート = 80
2. IPv4 要求パケットを受信すると、B4-CPE-1 はパケットを IPv6 パケットのペイロードにカプセル化し、IPv6 パケットを NS-1 に送信します。IPv6 パケットには、次のものがあります。
   - 送信元 IP アドレス = 2001: DB8:: 3:4
   - 宛先 IP アドレス = 2001: DB8:: 5:6
3. NS-1 が IPv6 パケットを受信すると、AFTR モジュールは IPv6 ヘッダーを削除してパケットのカプセル化を解除します。結果のパケットは、SUB-1 の元の IPv4 要求パケットです。
4. NS-1 の LSN モジュールは、パケットの送信元 IP アドレスとポートを、設定された NAT プールから選択された NAT IP アドレスおよび NAT ポートに変換します。変換された IPv4 パケットには、次のものがあります。
   - 送信元 IP アドレス = 203.0.113.61
   - 送信元ポート = 3002
   - 宛先 IP アドレス = 198.51.100.250
   - 宛先ポート = 80
5. LSN モジュールは、この DS Lite セッションの LSN マッピングとセッションエントリも作成します。マッピングには、次の情報が含まれます。
   - IPv6 パケットの送信元 IP アドレス（B4-CPE-1 の IPv6 アドレス）= 2001: DB8:: 3:4

- IPv4 パケットの送信元 IP アドレス（サブ-1 の IPv4 アドレス）= 192.0.2.51
- IPv4 パケットの送信元ポート = 2552
- NAT IP アドレス = 203.0.113.61
- NAT ポート = 3002

6. NS-1 は、結果の IPv4 パケットをインターネット上の宛先に送信します。
7. www.example.com のサーバーは、要求パケットを処理し、応答パケットを送信します。IPv4 応答パケットには、次のものがあります。
   - 送信元 IP アドレス = 198.51.100.250
   - 送信元ポート = 80
   - 宛先 IP アドレス = 203.0.113.61
   - 宛先ポート = 3002
8. IPv4 パケットを受信すると、NS-1 は LSN マッピングとセッションエントリを調べ、IPv4 応答パケットが DS Lite セッションに属していることを検出します。NS-1 の LSN モジュールは、宛先 IP アドレスとポートを変換します。これで、IPv4 パケットには次のものが含まれます。
   - 送信元 IP アドレス = 198.51.100.250
   - 送信元ポート = 80
   - 宛先 IP アドレス = 192.0.2.51
   - 宛先ポート = 2552
9. NS-1 の AFTR モジュールは、IPv4 パケットを IPv6 パケットにカプセル化し、その IPv6 パケットを B4-CPE-1 に送信します。IPv6 パケットには、次のものがあります。
   - 送信元 IP アドレス = 2001: DB8:: 5:6
   - 宛先 IP アドレス = 2001: DB8:: 3:4
10. パケットを受信すると、B4-CPE-1 は IPv6 ヘッダーを削除して IPv6 パケットを切断し、その結果の IPv4 パケットを CL-1 に送信します。

DS-Lite を構成する前に考慮すべきポイント

October 7, 2021

Citrix ADC アプライアンスで DS-Lite を構成する前に、次の点を考慮してください。

1. RFC 6333 で説明されている DS-Lite のさまざまなコンポーネントについて理解しておく必要があります。
2. Citrix ADC アプライアンスの DS-lite 構成では、LSN コマンドセットが使用されます。DS-Lite 構成では、LSN クライアントエンティティは、B4 デバイスからのトラフィックを識別するための IPv6 アドレス、IPv6 ネットワークアドレス、または ACL6 ルールを指定します。DS-Lite 構成には、Citrix ADC アプライアンス上の IPv6 アドレスの AFTR コンポーネントを指定する IPv6 プロファイルも含まれています。Citrix ADC LSN 機能の詳細については、「大規模 NAT」を参照してください。

3. DS-Lite 構成の場合、Citrix ADC アプライアンスは、次のいずれかのプロトコルに属する IPv4 パケットに対して LSN をサポートします。Citrix ADC アプライアンスは、他のプロトコルに属する IPv4 パケットをドロップします。
    - TCP
    - UDP
    - ICMP
4. Citrix ADC アプライアンスは、次の ALGsDS-Lite をサポートしています。
    - ICMP
    - FTP
    - TFTP
    - セッション開始プロトコル（SIP）
    - リアルタイムストリーミングプロトコル (RTSP)

DS-Lite の設定

October 7, 2021

Citrix ADC アプライアンスの DS-lite 構成では、LSN コマンドセットが使用されます。DS-Lite 構成では、LSN クライアントエンティティは、B4 デバイスからのトラフィックを識別するための IPv6 アドレス、IPv6 ネットワークアドレス、または ACL6 ルールを指定します。Citrix ADC LSN 機能の詳細については、「大規模 NAT」を参照してください。DS-Lite 構成には、Citrix ADC アプライアンス上の DS-Lite AFTR コンポーネントの IPv6 アドレス（SNIP6 タイプ）を指定する IPv6 プロファイルも含まれています。

Citrix ADC アプライアンスで DS-Lite を構成するには、次の作業を行います。

- グローバル **LSN** パラメータを設定します。グローバルパラメーターには、LSN 機能用に予約されている Citrix ADC メモリの量と、高可用性セットアップでの LSN セッションの同期が含まれます。
- **B4 CPE** デバイスからのトラフィックを識別するための **LSN** クライアントエンティティを作成します。LSN クライアントエンティティは、DS-Lite B4 デバイスのセットを参照します。クライアントエンティティには、これらの B4 デバイスからのトラフィックを識別するための IPv6 アドレス、IPv6 ネットワークアドレス、または ACL6 ルールが含まれます。LSN クライアントは、1 つの LSN グループにしかバインドできません。コマンドラインインターフェイスには、LSN クライアントエンティティを作成し、加入者を LSN クライアントエンティティにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。
- **LSN** プールを作成し、**NAT IP** アドレスをバインドします。NAT IPLSN プールは、LSN を実行するために Citrix ADC アプライアンスが使用する NAT IP アドレスのプールを定義します。コマンドラインインターフェイスには、LSN プールを作成し、NAT IP アドレスを LSN プールにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。

- **LSN IP6** プロファイルを作成します。LSN IP6 プロファイルは、Citrix ADC アプライアンス上の DS-Lite AFTR コンポーネントの IPv6 アドレスを定義します。IPv6 アドレスは、種類 SNIP6 の Citrix ADC が所有する IPv6 アドレスのいずれかである必要があります。
- （オプション）指定したプロトコルの **LSN** トランスポートプロファイルを作成します。LSN トランスポートプロファイルでは、加入者が特定のプロトコルに対して持つことのできる LSN セッションの最大数やポートの最大使用量など、さまざまなタイムアウトと制限を定義します。各プロトコル (TCP、UDP、および ICMP) の LSN トランスポートプロファイルを LSN グループにバインドします。プロファイルは、複数の LSN グループにバインドできます。LSN グループにバインドされたプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのトランスポートプロファイルと呼ばれます。LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルを上書きします。
- （オプション）指定したプロトコルの **LSN** アプリケーションプロファイルを作成し、宛先ポートのセットをそれにバインドします。LSN アプリケーションプロファイルは、特定のプロトコルおよび一連の宛先ポートのグループの LSN マッピングおよび LSN フィルタリングコントロールを定義します。一連の宛先ポートでは、各プロトコル（TCP、UDP、および ICMP）の LSN プロファイルを LSN グループにバインドします。プロファイルは、複数の LSN グループにバインドできます。LSN グループにバインドされた LSN アプリケーションプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのアプリケーションプロファイルと呼ばれます。指定された宛先ポートセットを持つ LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートのセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルを上書きします。コマンドラインインターフェイスには、LSN アプリケーションプロファイルを作成し、一連の宛先ポートを LSN アプリケーションプロファイルにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。
- **LSN** グループを作成し、**LSN** プール、**LSN IPv6** プロファイル、（オプション）**LSN** トランスポートプロファイル、および（オプション）**LSN** アプリケーションプロファイルを **LSN** グループにバインドします。LSN グループは、LSN クライアント、LSN IPv6 プロファイル、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルで構成されるエンティティです。グループには、ポートブロックサイズや LSN セッションのロギングなどのパラメータが割り当てられます。パラメータ設定は、LSN グループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。LSN グループにバインドできるのは 1 つの LSN IPv6 プロファイルだけです。LSN グループにバインドされた LSN IPv6 プロファイルは他の LSN グループにバインドできません。同じ NAT タイプ設定の LSN プールと LSN グループだけを 1 つにバインドできます。複数 LSN プールを 1 つの LSN グループにバインドできます。LSN グループ 1 つの LSN クライアントエンティティのみにバインドでき、LSN グループにバインドされた LSN クライアントエンティティは他の LSN グループにバインドできません。コマンドラインインターフェイスには、LSN グループを作成し、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルを

LSN グループにバインドするための 2 つのコマンドがあります。構成ユーティリティーは、これら 2 つの操作を 1 つの画面にまとめます。

コマンドラインを使用した設定

コマンドラインインターフェイスを使用して **LSN** クライアントを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IPv6** ネットワークまたは **ACL6** ルールを **LSN** クライアントにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プールを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add lsn pool <poolname> [-nattype ( DYNAMIC )] [-portblockallocation (
    ENABLED | DISABLED )] [-portrealloctimeout <secs>] [-
    maxPortReallocTmq <positive_integer>]

show lsn pool
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IP** アドレス範囲を **LSN** プールにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。


```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

注**:** LSN プールから LSN IP アドレスを削除するには、unbind lsn pool コマンドを使用します。

コマンドラインインターフェイスを使用して **LSN IPv6** プロファイルを構成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add lsn ip6profile <name>  -type DS-Lite  -network6 < ipv6_addr|*s >

show lsn ip6profile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** トランスポートプロファイルを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )] [-td <positive_integer>]

show lsn appsprofile
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して、アプリケーションプロトコルポート範囲を **LSN** アプリケーションプロファイルにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** グループを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC )]
    [-portblocksize <positive_integer>] [-logging (ENABLED | DISABLED )]
     [-sessionLogging ( ENABLED | DISABLED )][-sessionSync ( ENABLED |
    DISABLED )] [-snmptraplimit<positive_integer>] [-ftp ( ENABLED |
    DISABLED )] [-pptp ( ENABLED |DISABLED )] [-sipalg ( ENABLED |
    DISABLED )] [-rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プロトコルプロファイルと **LSN** プールを **LSN** グループにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

構成ユーティリティを使用した構成

構成ユーティリティを使用して **LSN** クライアントを構成し、**IPv6** ネットワークアドレスまたは **ACL6** ルールをバインドするには、次の手順を実行します。

[システム] > **[**大規模 **NAT]** > [クライアント] に移動し、クライアントを追加し、IPv6 ネットワークアドレスまたは ACL6 ルールをクライアントにバインドします。

設定ユーティリティを使用して **LSN** プールを設定し、**NAT IP** アドレスをバインドするには、次の手順を実行します。

[システム] > [大規模 **NAT**] > [プール] に移動し、プールを追加し、NAT IP アドレスまたは NAT IP アドレスの範囲をプールにバインドします。

構成ユーティリティを使用して **LSN IPv6** プロファイルを構成するには、次の手順を実行します。

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[**IPv6**] タブをクリックして、DS-Lite AFTR に IPv6 アドレスを割り当てます。

構成ユーティリティを使用して **LSN** トランスポートプロファイルを構成するには、次の手順を実行します。

1. [システム] > [大規模 **NAT**] > [プロファイル] に移動します。
2. 詳細ウィンドウで、[トランスポート**]** をクリックし、トランスポートプロファイルを追加します。

構成ユーティリティを使用して **LSN** アプリケーションプロファイルを構成するには、次の手順を実行します。

1. [システム] > [大規模 **NAT**] > [プロファイル] に移動します。
2. 詳細ウィンドウで、[アプリケーション**]** をクリックし、アプリケーションプロファイルを追加します。

構成ユーティリティを使用して **LSN** グループを構成し、**LSN** クライアント、**LSN IPv6** プロファイル、プール、トランスポートプロファイル、およびアプリケーションプロファイルをバインドするには、次の手順を実行します。

[システム] > **[**大規模 **NAT]** > [グループ] に移動し、グループを追加してから、LSN クライアント、LSN IPv6 プロファイル、プール、トランスポートプロファイル、およびアプリケーションプロファイルをグループにバインドします。

```
1  > add lsn client LSN-DSLITE-CLIENT-1
2  Done
3  > bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
4  Done
5  > add lsn pool LSN-DSLITE-POOL-1
6  Done
7  > bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
8  Done
9  > add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
       DB8::5:6
10 Done
11 > add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
       portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1
12 Done
13 > add lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
14 Done
```


DS-Lite のロギングとモニタリング

DS-Lite 情報をログに記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。Citrix ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (MAPPING)
- DS-Lite LSN マッピングエントリが作成または削除されたかどうか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が存在する場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートはログに記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピングのログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (SESSION)
- DS-Lite セッションが作成されるか、削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表に、構成済みのログサーバに保存されている各タイプの DS-Lite ログエントリの例を示します。これらのログエントリは、NSIP アドレスが 10.102.37.115.Citrix ADC アプライアンスによって生成されます。DS-Lite 情報をログに記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。Citrix ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ

- エントリの種類 (MAPPING)
- DS-Lite LSN マッピングエントリが作成または削除されたかどうか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が存在する場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートはログに記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピングのログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (SESSION)
- DS-Lite セッションが作成されるか、削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表に、構成済みのログサーバに保存されている各タイプの DS-Lite ログエントリの例を示します。これらのログエントリは、NSIP アドレスが 10.102.37.115 である Citrix ADC アプライアンスによって生成されます。

LSN ログエントリタイプ	サンプルログエントリ
DS-Lite セッションの作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP

DS-Lite セッションの削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN マッピングの作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN マッピングの削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

現在の **DS-Lite** セッションの表示

Citrix ADC アプライアンス上の不要なセッションや非効率的なセッションを検出するために、現在の DS-Lite セッションを表示できます。選択パラメータに基づいて、すべてまたは一部の DS-Lite セッションを表示できます。

コマンドラインインターフェイスを使用した設定

コマンド・ライン・インタフェースを使用してすべての **DS-Lite** セッションを表示するには:

コマンドプロンプトで入力します。

```
1 show lsn session  -nattype DS-Lite
2 <!--NeedCopy-->
```

コマンド・ライン・インタフェースを使用して、選択した **DS-Lite** セッションを表示するには:

コマンドプロンプトで入力します。


```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

例:

次の出力例は、Citrix ADC アプライアンス上に存在するすべての DS-Lite セッションを表示します。

```
show lsn session  -nattype DS-Lite
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1.  2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
    3002 TCP OUT

2.  2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
    52862 TCP OUT

3.  2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
    48116 ICMP OUT

4.  2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
    48305 TCP OUT

Done
<!--NeedCopy-->
```

設定ユーティリティを使用した設定

構成ユーティリティを使用してすべての DS-Lite セッションまたは選択した DS-Lite セッションを表示するには

1. [システム] **>** [** 大規模 **NAT**] > [セッション] に移動し、**[**DS-Lite**]** タブをクリックします。
2. 選択パラメータに基づいて DS-Lite セッションを表示するには、「検索」をクリックします。

DS-Lite セッションのクリア

不要な DS-Lite セッションや非効率的なセッションを Citrix ADC アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をただちに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、これらの削除されたセッションに関連

する後続のパケットもすべてドロップします。Citrix ADC アプライアンスから、すべてまたは選択した DS-Lite セッションを削除できます。

コマンドラインインターフェイスを使用してすべての **DS-Lite** セッションをクリアするには:

コマンドプロンプトで入力します。

```
flush lsn session -nattype DS-Lite
show lsn session -nattype DS-Lite
```

コマンド・ライン・インタフェースを使用して、選択した **DS-Lite** セッションをクリアするには:

コマンドプロンプトで入力します。

```
1 flush lsn session  - nattype DS-Lite [-clientname <string>] [-network <
     ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
     ip_addr> [-natPort <port>]]
2
3 show lsn session  - nattype DS-Lite
4 <!--NeedCopy-->
```

構成ユーティリティを使用してすべての **DS-Lite** セッションまたは選択した **DS-Lite** セッションをクリアするには、次の手順を実行します。

1. システムに移動 > 大規模 **NAT** > セッションをクリックし、[**DS-Lite**] タブをクリックします。
2. [セッションのフラッシュ] をクリックします。

DS-Lite スタティックマップの設定

October 7, 2021

Citrix ADC アプライアンスは、DS-Lite LSN マッピングの手動作成をサポートしています。このマッピングには、以下の情報間のマッピングが含まれます。

- 加入者の IP アドレスとポート、および B4 デバイスまたはコンポーネントの IPv6 アドレス
- NAT IP アドレスとポート

スタティック DS-Lite LSN マッピングは、NAT IP アドレスおよびポートに開始された接続が、指定された B4 デバイス（内部ネットワークにある Web サーバなど）を介して加入者 IP アドレスおよびポートにマッピングされるようにする場合に便利です。

注: この機能は、リリース 11.0 ビルド 64.x 以降でサポートされています。

コマンドラインを使用して **DS-Lite** 静的 **LSN** マッピングを作成するには

コマンドプロンプトで入力します。

```
add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
    <positive_integer>] [-network6 <B4_ADDR>] [<natIP> [<natPort>]] [-
    destIP<ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

パラメータの説明

add lsn static

- name

LSN スタティックマッピングエントリの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。LSN グループの作成後は変更できません。次の要件は、CLI だけに適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、「ds-lite lsn static1」や「ds-lite lsn static1」など）。これは必須の引数です。最大長：127

- transportprotocol

DS-Lite LSN マッピングエントリのプロトコル。

- subscrIP

DS-Lite LSN マッピングエントリのサブスクライバの IPv4 アドレス。

- subscrPort

DS-Lite LSN マッピングエントリのサブスクライバのポート。

- Network6

B4 デバイスまたはコンポーネントの IPv6 アドレス。

- td

B4 デバイスが属するトラフィックドメインの ID。B4 デバイスの IPv6 アドレスは、network6 パラメータで指定されます。ID を指定しない場合、B4 デバイスはデフォルトのトラフィックドメインの一部と見なされます。

- natIP

Citrix ADC アプライアンス上にすでに LSN タイプとして存在し、このマッピングエントリの NAT IP アドレスとして使用される IPv4 アドレス。

- NatPort

この DS-Lite LSN マッピングエントリの NAT ポート。

- destIP

DS-Lite LSN マッピングエントリの宛先 IP アドレス。

- dsttd

この DS-Lite LSN マッピングエントリの宛先 IP アドレスが Citrix ADC アプライアンスから到達可能なトラフィックドメインの ID。ID を指定しない場合、宛先 IP アドレスは、ID が 0 のデフォルトトラフィックドメインを介して到達可能であると見なされます。

構成ユーティリティを使用して **DS-Lite** スタティック **LSN** マッピングを作成するには

[システム] > [大規模 NAT] > [スタティック] に移動し、新しい DS-Lite スタティック LSN マッピングを追加します。

DS-Lite の **Deterministic NAT** 割り当ての設定

October 7, 2021

DS-Lite LSN 展開における Deterministic NAT 割り当ては、Citrix ADC アプライアンスが LSN NAT IP プールから、および指定されたポートブロックサイズ、LSN NAT IP アドレスおよび各サブスクライバ（B4 デバイスの背後にあるサブスクライバ）へのポートのブロックに基づいて、事前に割り当てる NAT リソース割り当ての一種です。

注**:** この機能は、リリース 11.0 ビルド 64.x 以降でサポートされています。

アプライアンスは、これらのサブスクライバに NAT リソースを順番に割り当てます。これは、開始 NAT IP アドレスの最初のポートブロックを開始サブスクライバ IP アドレスに割り当てます。次の範囲のポートは、次のサブスクライバに割り当てられます。次のサブスクライバに対して十分なポートが NAT アドレスにないまで、次のサブスクライバに割り当てられます。この時点で、次の NAT アドレスの最初のポートブロックがサブスクライバに割り当てられます。

Citrix ADC アプライアンスは、割り当てられた NAT IP アドレスと加入者のポートブロックを記録します。接続の場合、加入者は、マッピングされた NAT IP アドレスとポートブロックだけで識別できます。このため、Citrix ADC アプライアンスは、LSN セッションの作成または削除を記録しません。

DS-Lite サブスクライバは、Deterministic ポートブロックを 1 つだけ持つことができます。ポートのブロック全体が使用されている場合、Citrix ADC アプライアンスはサブスクライバーからの新しい接続を切断します。

例**:Deterministic DS-Lite**

この例では、Deterministic DS-Lite 設定には、IP アドレスが 192.0.17.5、192.0.17.6、192.0.17.7、および 192.0.17.8 の 4 つのサブスクライバが含まれています。これらの ipv4 サブスクライバは、IPv6 アドレス 2001: DB8:: 3:4 を

持つ B4 デバイスの背後にあります。この構成では、ポートブロックサイズが 20480 に設定され、LSN NAT IP アドレスプールの IP アドレスは 203.0.113.41-203.0.113.42 の範囲にあります。

Citrix ADC アプライアンスは、LSN NAT IP プールから、設定されたポートブロックサイズに基づいて、LSN NAT IP アドレスと各サブスクライバーへのポートのブロックを順番に事前割り当てします。これは、開始 NAT IP アドレス（203.0.113.41）のポートの最初のブロック（1024-21503）を、開始サブスクライバ IP アドレス（192.0.17.5）に割り当てます。次の範囲のポートは、次のサブスクライバに割り当てられます。次のサブスクライバに対して十分なポートが NAT アドレスにないまで、次のサブスクライバに割り当てられます。この時点で、次の NAT IP アドレスの最初のポートブロックがサブスクライバに割り当てられます。Citrix ADC は、NAT IP アドレスと各サブスクライバーに割り当てられたポートのブロックを記録します。

Citrix ADC アプライアンスは、これらのサブスクライバーに対して作成または削除された LSN セッションを記録しません。

次の表に、この例で各加入者に割り当てられた NAT IP アドレスとポートのブロックを示します。

加入者の IP アドレス	割り当てられた NAT IP アドレス	割り当てたポートのブロック	B4 の IPv6 アドレス
192.0.17.5	113.41	1024 - 21503	2001:DB8::3:4
192.0.17.6	113.41	21504 - 41983	2001:DB8::3:4
192.0.17.7	113.41	41984 - 62463	2001:DB8::3:4
192.0.17.8	113.42	1024 - 21503	2001:DB8::3:4

構成の手順

DS-Lite 設定の一部として Deterministic NAT を設定する必要があります。DS-Lite の設定手順については、DS-Lite の設定を参照してください。

DS-Lite の設定時には、次のことを確認してください。

- LSN プールと LSN グループを追加する場合は、NAT タイプパラメータを [Deterministic] に設定します。
- デフォルト値をそのまま使用できない限り、LSN グループを追加するときに、目的のポートブロックサイズパラメータを設定します。

Deterministic DS-Lite を構成する前に考慮すべきポイント

Deterministic DS-Lite を設定する前に、次の点を考慮してください。

- 各加入者の完全な IP アドレスは、Network パラメータと Netmask パラメータを設定して、別個の add lsn client コマンドで指定する必要があります。（ネットマスクを 255.255.255.255 に設定します）。また、

Network6 パラメータで指定された B4 デバイスの IPv4 アドレスも完全である必要があります（/128 プレフィクス）。つまり、Network パラメータと Network6 パラメータは、それぞれ /32 ビットマスクと /128 プレフィックス以外のアドレスを受け付けません。

- Citrix ADC アプライアンスは、Deterministic DS-Lite 構成で指定されていないが、決定論的な DS-Lite 構成で指定されている B4 デバイスの背後にあるサブスクライバーからの接続を切断します。
- Citrix ADC アプライアンスは、同じ IPv4 アドレスを持つ加入者が異なる B4 デバイスの背後にある場合、異なる加入者として認識します。加入者の IPv4 アドレスと B4 デバイスの組み合わせによって、DS-Lite 設定の LSN クライアントエンティティに一意の加入者が定義されます。

Deterministic DS-Lite 構成の例:

次の構成では、例:Deterministic DS-Lite のセクションに記載されている設定を使用します。

```
add lsn client LSN-DSLITE-CLIENT-10

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.5 -netmask
    255.255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.6 -netmask
    255.255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.7 -netmask
    255.255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.8 -netmask
    255.255.255.255 -network6 2001:DB8::3:4/128

Done
add lsn pool LSN-DSLITE-POOL-10 -nattype DETERMINISTIC

Done
bind lsn pool LSN-DSLITE-POOL-10  203.0.113.41-203.0.113.42

Done
add lsn ip6profile LSN-DSLITE-PROFILE-10 -type DS-Lite -network6 2001:
    DB8::5:6

Done
add lsn group LSN-DSLITE-GROUP-10 -clientname LSN-DSLITE-CLIENT-10 -
    nattype DETERMINISTIC -portblocksize 20480 -ip6profile LSN-DSLITE-
```



```
      PROFILE-10
26
27 Done
28 bind lsn group LSN-DSLITE-GROUP-10 -poolname LSN-DSLITE-POOL-10
29
30 Done
31 <!--NeedCopy-->
```

DS-Lite のアプリケーション層ゲートウェイの設定

October 7, 2021

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットのペイロードで通信されます。プロトコルのアプリケーション層ゲートウェイ（AGL）は、パケットのペイロードを解析し、プロトコルが DS-Lite 上で動作し続けるために必要な変更を行います。

Citrix ADC アプライアンスは、DS-Lite の次のプロトコルで ALG をサポートしています。

- FTP
- ICMP
- TFTP
- SIP
- RTSP

FTP、ICMP、および TFTP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

DS-Lite 設定の FTP プロトコルの ALG を有効または無効にするには、設定の LSN グループの FTP ALG オプションを有効または無効にします。

ICMP プロトコルの ALG はデフォルトで有効になっており、無効にするプロビジョニングはありません。

TFTP プロトコルの ALG は、デフォルトで無効になっています。エンドポイント独立マッピング、エンドポイント独立フィルタリング、および宛先ポートを 69（TFTP の場合は既知のポート）として UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、DS-Lite 設定で TFTP ALG が自動的に有効になります。

SIP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

SIP メッセージには SIP ヘッダーと SIP 本文に IP アドレスが含まれているため、DS-Lite をセッション開始プロトコル（SIP）とともに使用するのは複雑です。LSN を SIP とともに使用すると、SIP ヘッダーには発信者と受信者の情報が含まれ、デバイスはこの情報を変換して外部ネットワークから隠します。SIP 本体には、セッション記述プロトコル（SDP）情報が含まれています。この情報には、メディアを送信するための IP アドレスとポート番号が含まれます。DS-Lite 用の SIP ALG は、RFC 3261、RFC 3581、RFC 4566、および RFC 4475 に準拠しています。

> 注
>
> SIP ALG は、Citrix ADC スタンドアロンアプライアンス、Citrix ADC 高可用性セットアップ、および Citrix ADC クラスタセットアップでサポートされています。

SIP ALG の制限事項

DS-Lite 用の SIP ALG には、次の制限事項があります。

- SDP ペイロードだけがサポートされます。
- 以下はサポートされていません:
  - マルチキャスト IP アドレス
  - 暗号化 SDP
  - SIP TLS
  - 完全修飾ドメイン名変換
  - SIP レイヤ認証
  - 管理パーティション
  - マルチパートボディ
  - 折り畳み

SIP ALG の設定

LSN 設定の一部として SIP ALG を設定する必要があります。LSN の設定手順については、DS-Lite の設定を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加するときに、次のパラメータを設定します。
  - IP プーリング = PAIRED
  - アドレスとポートのマッピング = エンドポイント独在
  - フィルタリング = エンドポイントインデペンデント
- SIP ALG プロファイルを作成し、送信元ポート範囲または宛先ポート範囲を定義していることを確認します。SIP ALG プロファイルを LSN グループにバインドする
- LSN グループで SIP ALG を有効にします。

CLI を使用して **LSN** 設定で **SIP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string>[-sipalg ( ENABLED |
    DISABLED )]

show lsn group<groupname>
<!--NeedCopy-->
```

CLI を使用して **LSN** 設定で **SIP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn sipalgprofile<sipalgprofilename>[-dataSessionIdleTimeout<
    positive_integer>][-sipSessionTimeout<positive_integer>][-
    registrationTimeout<positive_integer>][-sipsrcportrange<port[-port
    ]>][-sipdstportrange<port[-port]>][-openRegisterPinhole ( ENABLED |
    DISABLED )][-openContactPinhole ( ENABLED | DISABLED )][-
    openViaPinhole ( ENABLED | DISABLED )][-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP )[-
    openRoutePinhole ( ENABLED | DISABLED )][-rport ( ENABLED | DISABLED
     )]

show lsn sipalgprofile<sipalgprofilename>
<!--NeedCopy-->
```

構成例

次の DS-Lite 設定の例では、ネットワーク 2001: DB8:: 3:0 /96 内の B4 デバイスからの TCP トラフィックに対して SIP ALG が有効になっています。

```
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/96
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
```



```
8  Done
9  add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
       DB8::5:6
10 Done
11 add lsn appsprofile LSN-DSLITE-APPS-PROFILE-1 TCP -ippooling PAIRED -
       mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
12 Done
13 add lsn sipalgprofile SIPALGPROFILE-1  -sipdstportrange 5060 -
       sipTransportProtocol TCP
14 Done
15 add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
       portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1 -sipalg ENABLED
16 Done
17 bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
18 Done
19 bind lsn group LSN-DSLITE-GROUP-1 -appsprofilename LSN-DSLITE-APPS-
       PROFILE-1
20 Done
21 bind lsn group LSN-DSLITE-GROUP-1 -sipalgprofilename SIPALGPROFILE-1
22 Done
23 <!--NeedCopy-->
```

RTSP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

リアルタイムストリーミングプロトコル (RTSP) は、リアルタイムメディアデータを転送するためのアプリケーションレベルのプロトコルです。RTSP は、エンドポイント間のメディアセッションを確立および制御するために使用され、メディアクライアントとメディアサーバ間のコントロールチャネルプロトコルです。一般的な通信は、クライアントとストリーミングメディアサーバーの間です。

プライベートネットワークからパブリックネットワークにメディアをストリーミングするには、ネットワーク上で IP アドレスとポート番号を変換する必要があります。Citrix ADC の機能には、RTSP 用のアプリケーションレイヤーゲートウェイ（ALG）が含まれています。ALG（大規模 NAT）とともに使用すると、メディアストリームを解析し、プロトコルがネットワーク上で動作し続けるように必要な変更を行うことができます。

IP アドレス変換の実行方法は、メッセージのタイプと方向、およびクライアント/サーバ展開でサポートされるメディアのタイプによって異なります。メッセージは次のように翻訳されます。

- 送信要求：Citrix ADC が所有するパブリック IP アドレスへのプライベート IP アドレス（LSN IP アドレスと呼ばれます）。
- 着信応答：プライベート IP アドレスへの LSN IP アドレス。

- インバウンド要求：変換なし。
- 発信応答：LSN プール IP アドレスへのプライベート IP アドレス。

> 注
>
> RTSP ALG は、Citrix ADC スタンドアロンアプライアンス、Citrix ADC 高可用性セットアップ、および Citrix ADC クラスタセットアップでサポートされています。

RTSP ALG の制限事項

RTSP ALG は、次の機能をサポートしていません。

- マルチキャスト RTSP セッション
- UDP を介した RTSP セッション
- 管理パーティション
- RTSP 認証
- HTTP トンネリング

RTSP ALG の設定

RTSP ALG を LSN 設定の一部として設定します。LSN の設定手順については、DS-Lite の設定を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加するときに、次のパラメータを設定します。
  - IP プーリング = PAIRED
  - アドレスとポートのマッピング = エンドポイント独在
  - フィルタリング = エンドポイントインデペンデント
- LSN グループで RTSP ALG を有効にします。
- RTSP ALG プロファイルを作成し、RTSP ALG プロファイルを LSN グループにバインドします。

CLI を使用して **LSN** 設定の **RTSP ALG** を有効にするには

コマンドプロンプトで入力します。

```
1 add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
     DISABLED )]
2
3 show lsn group <groupname>
4 <!--NeedCopy-->
```


CLI を使用して **LSN** 設定の **RTSP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

RTSP ALG 設定の例

次の DS-Lite 設定例では、ネットワーク 2001: DB8:: 4:0 /96 内の B4 デバイスからの TCP トラフィックに対して RTSP ALG が有効になっています。

RTSP ALG 設定の例:

```
add lsn client LSN-DSLITE-CLIENT-5
Done
bind lsn client LSN-DSLITE-CLIENT-5 -network6 2001:DB8::4:0/96
Done
add lsn pool LSN-DSLITE-POOL-5
Done
bind lsn pool LSN-DSLITE-POOL-5 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-5 -type DS-Lite -network6 2001:
    DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-5 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-5 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-DSLITE-GROUP-5 -clientname LSN-DSLITE-CLIENT-5 -
    portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-5 -rtspalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-5 -poolname LSN-DSLITE-POOL-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -appsprofilename LSN-DSLITE-APPS-
    PROFILE-5
```



```
Done
bind lsn group LSN-DSLITE-GROUP-5 -rtspalgprofilename RTSPALGPROFILE-5
Done
<!--NeedCopy-->
```

DS-Lite のロギングとモニタリング

October 7, 2021

DS-Lite 情報をログに記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。Citrix ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (MAPPING)
- DS-Lite LSN マッピングエントリが作成または削除されたかどうか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が存在する場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートはログに記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピングのログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (SESSION)
- DS-Lite セッションが作成されるか、削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表に、構成済みのログサーバに保存されている各タイプの DS-Lite ログエントリの例を示します。これらのログエントリは、NSIP アドレスが 10.102.37.115.Citrix ADC アプライアンスによって生成されます。DS-Lite 情報をログに記録して、問題の診断やトラブルシューティング、法的要件を満たすことができます。Citrix ADC アプライアンスは、DS-Lite 情報をログに記録するための LSN ロギング機能をすべてサポートしています。DS-Lite ロギングを構成するには、「LSN のロギングとモニタリング」で説明した LSN ロギングを設定する手順を使用します。

DS-Lite LSN マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (MAPPING)
- DS-Lite LSN マッピングエントリが作成または削除されたかどうか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が存在する場合があります。
  - エンドポイントに依存しないマッピングでは、宛先 IP アドレスとポートはログに記録されません。
  - アドレス依存マッピングでは、宛先 IP アドレスのみがログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレス-ポート依存マッピングのログに記録されます。

DS-Lite セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (SESSION)
- DS-Lite セッションが作成されるか、削除されるか
- B4 の IPv6 アドレス
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表に、構成済みのログサーバに保存されている各タイプの DS-Lite ログエントリの例を示します。これらのログエントリは、NSIP アドレスが 10.102.37.115 である Citrix ADC アプライアンスによって生成されます。

LSN ログエントリタイプ	サンプルログエントリ

DS-Lite セッションの作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite セッションの削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN マッピングの作成	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN マッピングの削除	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

現在の **DS-Lite** セッションの表示

Citrix ADC アプライアンス上の不要なセッションや非効率的なセッションを検出するために、現在の DS-Lite セッションを表示できます。選択パラメータに基づいて、すべてまたは一部の DS-Lite セッションを表示できます。

コマンドラインインターフェイスを使用してすべての **DS-Lite** セッションを表示するには

コマンドプロンプトで入力します。


```
show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

コマンド・ライン・インタフェースを使用して選択した **DS-Lite** セッションを表示するには

コマンドプロンプトで入力します。

```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

次の出力例は、Citrix ADC アプライアンス上に存在するすべての DS-Lite セッションを表示します。

> lsn セッションの表示：nattype DS-Lite

```
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1. 2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
    3002 TCP OUT

2. 2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
    52862 TCP OUT

3. 2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
    48116 ICMP OUT

4. 2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
    48305 TCP OUT
Done
<!--NeedCopy-->
```

設定ユーティリティを使用した設定

構成ユーティリティを使用してすべての DS-Lite セッションまたは選択した DS-Lite セッションを表示するには

1. [システム] **>** [****** 大規模 **NAT**] > [セッション] に移動し、**[**DS-Lite**] タブをクリックします。
2. 選択パラメータに基づいて DS-Lite セッションを表示するには、「検索」をクリックします。

DS-Lite セッションのクリア

不要な DS-Lite セッションや非効率的なセッションを Citrix ADC アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をただちに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、これらの削除されたセッションに関連する後続のパケットもすべてドロップします。Citrix ADC アプライアンスから、すべてまたは選択した DS-Lite セッションを削除できます。

コマンドラインインターフェイスを使用してすべての **DS-Lite** セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session  -nattype DS-Lite

show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して選択した **DS-Lite** セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

構成ユーティリティを使用してすべての **DS-Lite** セッションまたは選択した **DS-Lite** セッションをクリアするには

1. システムに移動 > 大規模 **NAT** > セッションをクリックし、[**DS-Lite**] タブをクリックします。
2. [セッションのフラッシュ] をクリックします。

HTTP ヘッダー情報のロギング

Citrix ADC アプライアンスは、DS-Lite 機能を使用している HTTP 接続の要求ヘッダー情報をログに記録できます。HTTP 要求パケットの次のヘッダー情報をログに記録できます。

- HTTP リクエストの宛先となる URL

- HTTP リクエストで指定された HTTP メソッド
- HTTP リクエストで使用される HTTP バージョン
- HTTP 要求を送信したサブスクライバの IPv4 アドレス

ISP は HTTP ヘッダーログを使用して、一連のサブスクライバ間の HTTP プロトコルに関連する傾向を確認できます。たとえば、ISP はこの機能を使用して、一連の加入者の中で最も人気のある Web サイトを見つけることができます。

構成の手順

HTTP ヘッダー情報をログに記録するように Citrix ADC アプライアンスを構成するには、次のタスクを実行します。

- **HTTP** ヘッダーログプロファイルを作成します。HTTP ヘッダーログプロファイルは、ロギングを有効または無効にできる HTTP ヘッダー属性（URL や HTTP メソッドなど）のコレクションです。
- **HTTP** ヘッダーを **DS-Lite LSN** 設定の **LSN** グループにバインドします。HTTP ヘッダーログプロファイル名パラメータを作成された HTTP ヘッダーログプロファイルの名前に設定することにより、HTTP ヘッダーログプロファイルを LSN 設定の LSN グループにバインドします。次に、Citrix ADC アプライアンスは、LSN グループに関連するすべての HTTP 要求の HTTP ヘッダー情報をログに記録します。HTTP ヘッダーログプロファイルは、複数の LSN グループにバインドできますが、LSN グループには 1 つの HTTP ヘッダーログプロファイルしか設定できません。

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、**HTTP** ヘッダーログプロファイルを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
```



```
4 <!--NeedCopy-->
```

構成例

次の DS-Lite LSN 設定では、HTTP ヘッダーログプロファイル HTTP-ヘッダー LOG-1 は LSN グループ LSN-DSLITE-GROUP-1 にバインドされています。ログプロファイルでは、すべての HTTP 属性（URL、HTTP メソッド、HTTP バージョン、および HOST IP アドレス）がロギング用に有効になっているため、B4 デバイスからの HTTP 要求（ネットワーク 2001:DB 8:5001:: /96 内）について、これらの属性はすべてロギングされます。

設定例**:**

```
1  add lsn httphdrlogprofile HTTP-HEADER-LOG-1
2
3  Done
4
5  add lsn client LSN-DSLITE-CLIENT-1
6
7  Done
8
9  bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
10
11 Done
12
13 add lsn pool LSN-DSLITE-POOL-1
14
15 Done
16
17 bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
18
19 Done
20
21 add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
      DB8::5:6
22
23 Done
24
25 add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
      portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1
26
27 Done
28
29 bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
30
```



```
31 Done
32
33 bind lsn group LSN-DSLITE-GROUP-1 -httphdrlogprofilename HTTP-HEADER-
       LOG-1
34
35 Done
36 <!--NeedCopy-->
```

IPFIX ロギング

Citrix ADC アプライアンスは、LSN イベントに関する情報をインターネット・プロトコル・フロー情報エクスポート（IPFIX）形式で IPFIX コレクタで構成されたセットに送信できます。アプライアンスは、既存の AppFlow 機能を使用して、IPFIX 形式の LSN イベントを IPFIX コレクタに送信します。

IPFIX ベースのログ記録は、次の DS_Lite 関連イベントで使用できます。

- LSN セッションの作成または削除。
- LSN マッピングエントリの作成または削除。
- Deterministic NAT のコンテキストにおけるポートブロックの割り当てまたは割り当て解除。
- Dynamic NAT のコンテキストにおけるポートブロックの割り当てまたは割り当て解除。
- サブスクライバセッションクォータを超えた場合。

IPFIX ロギングを構成する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- Citrix ADC アプライアンスで AppFlow 機能および IPFIX コレクタを構成する必要があります。手順については、AppFlow 機能の構成を参照してください。

構成の手順

LSN 情報を IPFIX 形式でロギングするには、次の作業を実行します。

- **AppFlow** 構成で **LSN** ロギングを有効にします。AppFlow 構成の一部として LSN ロギングパラメータを有効にします。
- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、IPFIX 形式のログ情報を有効または無効にする IPFIX パラメータが含まれています。
- **LSN** ログプロファイルを **LSN** 構成の **LSN** グループにバインドします。LSN ログプロファイルを 1 つまたは複数の LSN グループにバインドします。バインドされた LSN グループに関連するイベントは、IPFIX 形式で記録されます。

CLI を使用して **AppFlow** 構成で **LSN** ロギングを有効にするには

コマンドプロンプトで入力します。

```
set appflow param -lsnLogging (ENABLED |DISABLED )

show appflow param
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを作成するにはコマンドプロンプトで

コマンドプロンプトで入力します。

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

GUI を使用して **LSN** ログプロファイルを作成するには

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[ログ] タブをクリックしてログプロファイルを追加します。

GUI を使用して **LSN** ログプロファイルを **LSN** 構成の **LSN** グループにバインドするには

1. [システム] > [大規模 **NAT**] > [**LSN** グループ**]** に移動し、**LSN** グループを開きます。
2. [詳細設定**]** で、 [**+** ログプロファイル] をクリックして、作成したログプロファイルを LSN グループにバインドします。

DS-Lite のポート制御プロトコル

October 7, 2021

Citrix ADC アプライアンスは、大規模 NAT（LSN）用のポート制御プロトコル（PCP）をサポートするようになりました。ISP の加入者アプリケーションの多くは、インターネットからアクセスできる必要があります (インターネット経由で監視する IP カメラなど、モノのインターネット (IOT) デバイスなど)。この要件を満たす方法の 1 つは、スタティック大規模 NAT（LSN）マップを作成することです。しかし、非常に多数のサブスクライバの場合、スタティック LSN NAT マップを作成することは実現可能な解決策ではありません。

Port Control Protocol（PCP; ポート制御プロトコル）を使用すると、加入者は、自身または他のサードパーティデバイスに対して特定の LSN NAT マッピングを要求できます。大規模な NAT デバイスは LSN マップを作成し、加入者に送信します。加入者は、インターネット上のリモートデバイスに、加入者に接続できる NAT IP アドレス:NAT ポートを送信します。

通常、アプリケーションは LSN マッピングがタイムアウトしないように、大規模な NAT デバイスに頻繁にキープアライブメッセージを送信します。PCP は、アプリケーションが LSN マッピングのタイムアウト設定を学習できるようにすることで、このようなキープアライブメッセージの頻度を減らすのに役立ちます。これにより、ISP のアクセスネットワークの帯域幅消費とモバイルデバイスのバッテリ消費を削減できます。

PCP はクライアント/サーバーモデルで、UDP トランスポートプロトコル上で動作します。Citrix ADC アプライアンスは、PCP サーバーコンポーネントを実装し、RFC 6887 に準拠しています。

構成の手順

PCP を設定するには、次の作業を実行します。

- （オプション）PCP プロファイルを作成します。PCP プロファイルには、PCP 関連パラメータの設定（たとえば、マッピングおよびピア PCP 要求をリッスンする）が含まれます。PCP プロファイルを PCP サーバーにバインドできます。PCP サーバーにバインドされた PCP プロファイルは、すべての設定を PCP サーバーに適用します。PCP プロファイルは、複数の PCP サーバにバインドできます。デフォルトでは、デフォルトのパラメータ設定を持つ 1 つの PCP プロファイルがすべての PCP サーバーにバインドされます。PCP サーバーにバインドした PCP プロファイルは、そのサーバーのデフォルトの PCP プロファイル設定よりも優先されます。デフォルトの PCP プロファイルには、次のパラメータ設定があります。
  - マッピング: 有効
  - ピア： 有効
  - 最小マップ寿命:120 秒
  - 最大マップ寿命:86400 秒。
  - アナウンス数:10
  - サードパーティ: 無効
- PCP サーバーを作成し、PCP プロファイルをバインドします。Citrix ADC アプライアンス上に PCP サーバーを作成し、PCP 関連の要求とサブスクライバーからのメッセージをリッスンします。サブネット IP（SNIP）

アドレスは、PCP サーバにアクセスするために割り当てる必要があります。デフォルトでは、PCP サーバーはポート 5351 をリッスンします。

- PCP サーバを LSN 設定の LSN グループにバインドします。PCP Server パラメータを設定して、作成した PCP サーバーを指定することで、作成した PCP サーバーを LSN 構成の LSN グループにバインドします。作成された PCP サーバにアクセスできるのは、この LSN グループのサブスクライバだけです。
  注：大規模な NAT 設定の PCP サーバは、ACL ルールから識別されるサブスクライバからの要求を処理しません。

CLI を使用して **PCP** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

CLI を使用して **PCP** サーバーを作成するには

コマンドプロンプトで入力します。

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

DS-LITE の設定例

次の設定例では、PCP サーバ PCP-SERVER-1 は、PCP-DSLITE-PROFILE-1 からの PCP 設定で、LSN グループ LSN-DSLITE-GROUP-1 にバインドされています。PCP-SERVER-9 は、ネットワーク 2001: DB8:: 3:0 /100 からの B4 デバイスの背後にある IPv4 サブスクライバからの PCP 要求を処理します。

設定例:


```
add pcp profile PCP-DSLITE-PROFILE-1 -minMapLife 300
Done
add pcp server PCP-DSLITE-SERVER-1 192.0.3.10  -pcpProfile PCP-DSLITE-PROFILE-1
Done
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:DB8::5:6
Done
add lsn group LSN-DSLITE-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname  PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

大規模 **NAT64**

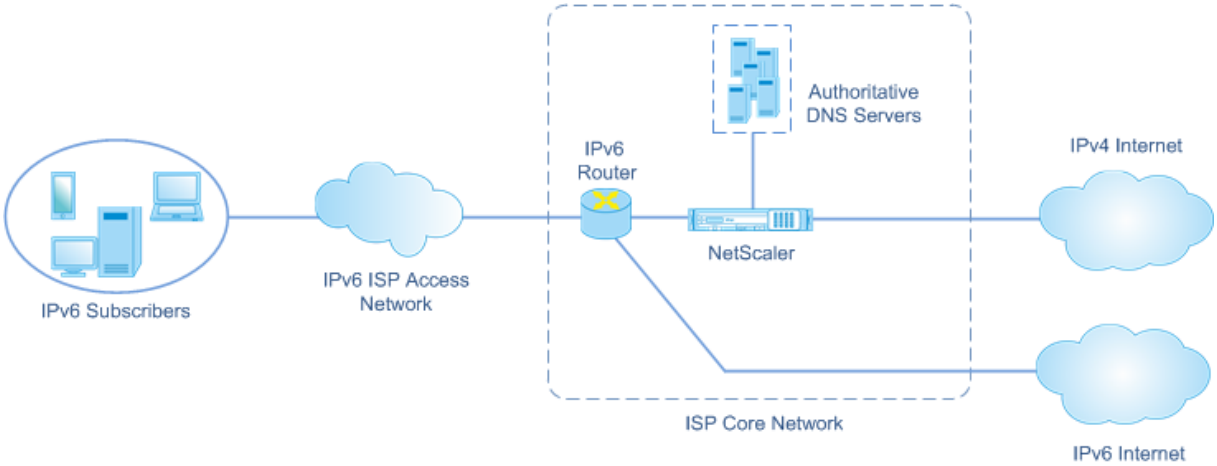
October 7, 2021

IPv4 アドレスの枯渇が間もなく、ISP は IPv6 インフラストラクチャに移行し始めています。しかし、移行中、ISP は IPv6 とともに IPv4 をサポートし続ける必要があります。なぜなら、パブリックインターネットの大半は依然として IPv4 を使用しているからです。大規模な NAT64 は、IPv6 インフラストラクチャを備えた ISP 向けの IPv6 移行ソリューションです。IPv6 のみのサブスクライバを IPv4 インターネットに接続できます。DNS64 は、IPv6 のみのクライアントによる IPv4 のみのドメインの検出を可能にするソリューションです。DNS64 は大規模な NAT64 とともに使用され、IPv6 のみのクライアントと IPv4 のみのサーバー間のシームレスな通信が可能になります。

Citrix ADC アプライアンスは、大規模な NAT64 および DNS64 を実装し、RFC6145、6146、6147、6052、3022、2373、2765、および 2464 に準拠しています。

アーキテクチャ

Citrix ADC アプライアンスを使用する ISP の NAT64 アーキテクチャは、ISP のコアネットワークに展開された Citrix ADC アプライアンスを介して IPv4 インターネットにアクセスする IPv6 サブスクライバで構成されています。IPv6 サブスクライバは、ISP の IPv6 専用アクセスネットワークを介して ISP コアネットワークに接続されます。



Citrix ADC アプライアンスの大規模な NAT64 機能は、Citrix ADC アプライアンス上でセッション情報を維持しながら、IPv6 から IPv4 へのパケット変換を介して IPv6 クライアントと IPv4 サーバー間の通信を可能にします。Citrix ADC DNS64 機能は、IPv6 への IPv4 のみのドメインを表します。サブスクライバは、IPv4 専用ドメインの DNS AAAA レコードを合成し、サブスクライバに送信します。

大規模な NAT64 には、NAT64 プレフィクスと NAT IPv4 プールという 2 つの主要コンポーネントがあります。DNS64 には、NAT64 プレフィクスと同じ値を持つ DNS 64 プレフィクスというメインコンポーネントが 1 つあります。

Citrix ADC DNS64 機能は、IPv6 のみの加入者からインターネット上の IPv4 のみの Web サーバー上でホストされているドメイン名に対する AAAA 要求を受信すると、そのドメイン名の AAAA レコードを合成し、加入者に送信します。AAAA レコードは、DNS64 プレフィクス（NAT64 プレフィクスに設定）とドメイン名の実際の IPv4 アドレスを連結することによって合成されます。

これで、加入者には、目的のドメイン名に対応する IPv6 宛先アドレスが割り当てられます。加入者は、合成された IPv6 アドレスに要求を送信します。IPv6 要求を受信すると、大規模な Citrix ADC NAT64 機能は、IPv6 要求パケットを IPv4 要求パケットに変換します。大規模な NAT64 は、IPv4 要求の宛先アドレスを IPv4 アドレスに設定します。IPv4 アドレスは、IPv6 アドレスから NAT64 プレフィクスを取り除き、IPv6 要求の宛先アドレスから抽出されます。宛先ポートは、IPv6 要求から保持されます。また、大規模な NAT64 は、IPv4 パケットの送信元 IP アドレス: 送信元ポートを、設定された NAT プールから選択された NAT IP アドレス:NAT ポートに設定します。

アプライアンスは、大規模な NAT64 機能を使用するすべてのアクティブセッションの記録を保持します。これらのセッションは、大規模な NAT64 セッションと呼ばれます。アプライアンスは、大規模な NAT64 セッションごとに、サブスクライバ IPv6 アドレスとポート、および NAT IPv4 アドレスとポート間のマッピングも維持します。これらのマッピングは、大規模な NAT64 マッピングと呼ばれます。大規模な NAT64 セッションエントリと大規模な NAT64 マッピングエントリから、Citrix ADC アプライアンスは、（インターネットから受信した）応答パケットを特

定の NAT64 セッションに属するものとして認識します。

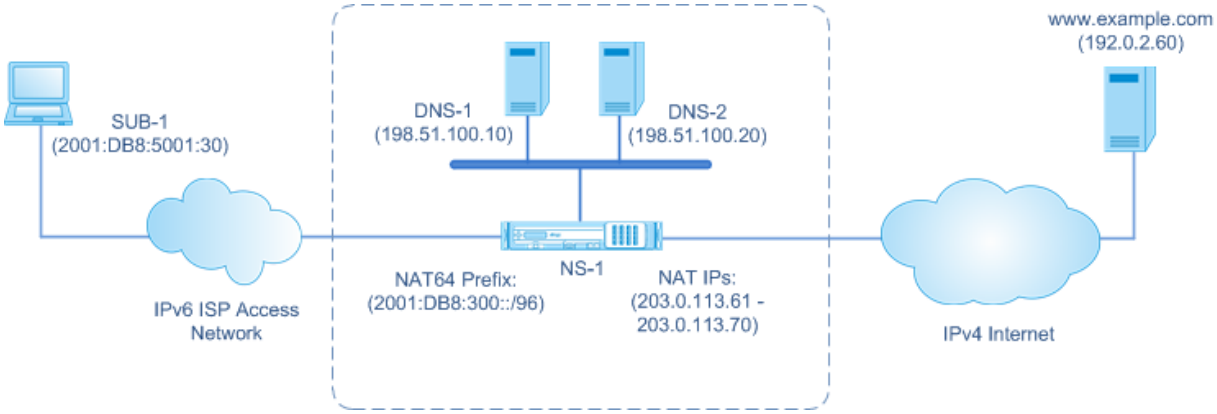
アプライアンスは、特定の NAT64 セッションに属する IPv4 応答パケットを受信すると、NAT64 セッションに格納されている情報を使用して IPv4 パケットを IPv6 パケットに変換し、IPv6 応答パケットをサブスクライバに送信します。

例：**NAT64** および **DNS64** 配置のトラフィックフロー

ISP のコアネットワーク内の Citrix ADC アプライアンス NS-1 と 2 つのローカル DNS サーバー（DNS-1 と DNS-2）、および IPv6 サブスクライバ SUB-1 で構成される大規模な NAT64 および DNS64 展開の例を考えてみましょう。SUB-1 は、ISP の IPv6 アクセス・ネットワークを介して NS-1 に接続されています。NS-1 には、IPv6 サブスクライバ SUB-1 ホストと IPv4 ホスト（内部および外部）間の通信を可能にするための大規模な NAT64 および DNS64 構成が含まれています。

大規模な NAT64 設定には、IPv6 要求を IPv4 要求に変換するための NAT64 プレフィクス（2001:DB 8:300:: /96）と NAT IPv4 プールが含まれます。

DNS64 構成には、DNS ロードバランシング仮想サーバー LBVS-DNS64-1 (2001: DB 8:9999:: 99) と DNS64 プレフィックス (2001: DB 8:300:: /96) が含まれています。LBVS-DNS64-1 は、ISP の加入者に対するローカル DNS サーバ DNS-1 および DNS-2 を表します。NAT64 プレフィクスと同じ値を持つ DNS64 プレフィクスは、DNS サーバ DNS-1 および DNS-2 から受信した DNS A レコードからの DNS AAAA レコードの合成に使用されます。NS-1 は、IPv4 ホストを解決するための DNS 要求に対して、合成された AAAA レコードで SUB-1 に応答します。



DNS64 トラフィックフロー

トラフィックは、IPv6 サブスクライバ SUB-1 と、インターネット上の IPv4 専用の Web サーバ上にあるサイト`www.example.com`の間で次のように流れます。

1. IPv6 サブスクライバ SUB-1 は、`www.example.com`の DNS AAAA 要求を、その指定された DNS サーバ（2001:DB 8:9999:: 99）に送信します。
2. DNS ロードバランシング仮想サーバー LBVS-DNS64-1 (2001: DB 8:9999:: 99) は、Citrix ADC アプライアンス NS1 の AAAA 要求を受信します。LBVS-DNS64-1 の負荷分散アルゴリズムは、DNS サーバー DNS-1 を選択し、AAAA 要求をそれに転送します。

3. www.example.comで使用可能な AAAA レコードがないため、DNS-1 は空のレコードまたはエラーメッセージを返します。
4. LBVS-DNS64-1 で DNS64 オプションが有効になっており、CL1 からの AAAA 要求は DNS64-ポリシー 1 で指定された条件と一致するため、NS1 はwww.example.comの IPv4 アドレスの DNS A 要求を DNS-1 に送信します。
5. DNS-1 は、www.example.comに対して 192.0.2.60 の A レコードで応答します。
6. NS1 上の DNS64 モジュールは、LBVS-DNS64-1 に関連付けられた DNS64 プレフィックス (2001: DB 8:300:: /96) と、www.example.com= 2001: DB 8:300:: 192.0.2.60 の IPv4 アドレス (192.0.2.60) を連結することにより、www.example.comの AAAA レコードを合成します。
7. NS1 は、統合された AAAA レコードを IPv6 クライアント CL1 に送信します。NS1 は、A レコードをメモリにキャッシュします。NS1 は、キャッシュされた A レコードを使用して、後続の AAAA 要求に対して AAAA レコードを合成します。

NAT64 トラフィックフロー

1. IPv6 サブスクライバ SUB-1 は、2001: DB 8:5001:30 www.example.comに要求を送信します。IPv6 パケットには、次のものがあります。
   - 送信元 IP アドレス = 2001: DB 8:5001:30
   - 送信元ポート = 2552
   - 宛先 IP アドレス = 2001:DB8:300::192.0.2.60
   - 宛先ポート = 80
2. IPv6 サブスクライバ SUB-1 は、2001: DB 8:5001:30 www.example.comに要求を送信します。IPv6 パケットには、次のものがあります。
   - 送信元 IP アドレス = 2001: DB 8:5001:30
   - 送信元ポート = 2552
   - 宛先 IP アドレス = 2001:DB8:300::192.0.2.60
   - 宛先ポート = 80
3. NS-1 が IPv6 パケットを受信すると、大規模な NAT64 モジュールは、次のように変換された IPv4 要求パケットを作成します。
   - 送信元 IP アドレス = 設定済みの NAT プールで使用可能な IPv4 アドレスのいずれか（203.0.113.61）
   - 送信元ポート = 割り当てられた NAT IPv4 アドレスで使用可能なポートの 1 つ（3002）
   - 宛先 IP アドレス = IPv6 アドレス (192.0.2.60) から NAT64 プレフィックス (2001: DB 8:300:: /96) を取り除き、IPv6 要求の宛先アドレスから抽出された IPv4 アドレス
   - 宛先ポート = IPv6 要求の宛先ポート (80)
4. 大規模な NAT64 モジュールは、この大規模な NAT64 フローのマッピングおよびセッションエントリも作成します。セッションおよびマッピングのエントリには、次の情報が含まれます。
   - IPv6 パケットの送信元 IP アドレス = 2001: DB 8:5001:30

- IPv6 パケットの送信元ポート = 2552
- NAT IP アドレス = 203.0.113.61
- NAT ポート = 3002
- NS-1 は、結果の IPv4 パケットをインターネット上の宛先に送信します。

5. 要求パケットを受信すると、`www.example.com`サーバはパケットを処理し、応答パケットを NS-1 に送信します。IPv4 応答パケットには、次のものがあります。

   - 送信元 IP アドレス = 192.0.2.60
   - 送信元ポート = 80
   - 宛先 IP アドレス = 203.0.113.61
   - 宛先ポート = 3002

6. IPv4 応答パケットを受信すると、NS-1 は大規模な NAT64 マッピングおよびセッションエントリを調べ、IPv4 応答パケットが大規模な NAT64 セッションに属していることを検出します。大規模な NAT64 モジュールは、変換された IPv6 応答パケットを作成します。

   - Source IP address = 2001:DB8:300::192.0.2.60
   - 送信元ポート = 80
   - 宛先 IP アドレス = 2001: DB 8:5001:30
   - 宛先ポート = 2552

7. NS-1 は、変換された IPv6 応答をクライアント SUB-1 に送信します。

Citrix ADC アプライアンスでサポートされる大規模な **NAT64** 機能

Citrix ADC アプライアンスの大規模 NAT64 は、標準の LSN 機能セットをサポートしています。これらの LSN 機能の詳細については、「LSN アーキテクチャ」を参照してください。

Citrix ADC アプライアンスでサポートされる大規模な NAT64 機能の一部を次に示します。

- ALG。SIP、RTSP、FTP、ICMP、および TFTP プロトコル用のアプリケーション層ゲートウェイ（ALG）のサポート。
- Deterministic /固定 NAT。ロギングを最小限に抑えるために、サブスクライバへのポートブロックの事前割り当てをサポートします。
- マッピング。エンドポイント独立マッピング（EIM)、アドレス依存マッピング（ADM)、およびアドレスポート依存マッピング（APDM）のサポート。
- フィルタリング。エンドポイント独立フィルタリング (EIF)、アドレス依存フィルタリング (ADF)、およびアドレスポート依存フィルタリング (APDF) のサポート。
- クォータ。ポート数、サブスクライバごとのセッション、および LSN グループごとのセッション数に関する設定可能な制限。
- スタティックマッピング。大規模な NAT64 マッピングを手動で定義するサポート。
- ヘアピンの流れ。NAT IP アドレスを使用した加入者または内部ホスト間の通信のサポート。

- 464XLAT 接続。IPv6 サブスクライバホスト上の IPv4 専用アプリケーションと、IPv6 ネットワーク経由のインターネット上の IPv4 ホスト間の通信をサポートします。
- 可変長の NAT64 プレフィクスおよび DNS64 プレフィクス。Citrix ADC アプライアンスは、長さが 32、40、48、56、64、96 の NAT64 および DNS64 プレフィックスの定義をサポートしています。
- 複数の NAT64 プレフィクスおよび DNS64 プレフィクス。Citrix ADC アプライアンスは、複数の NAT64 および DNS64 プレフィックスをサポートしています。
- LSN クライアント。IPv6 プレフィクスおよび拡張 ACL6 ルールを使用した、大規模な NAT64 の加入者の指定または識別をサポートします。
- ロギング。法執行機関の NAT64 セッションのロギングのサポート。さらに、次のロギングもサポートされています。
  - 信頼性の高いシステムログ。より信頼性の高い転送メカニズムを実現するために、TCP 経由で SYSLOG メッセージを外部ログサーバに送信するサポート。
  - ログサーバーの負荷分散。冗長ログメッセージの格納を防止するための外部ログサーバのロードバランシングのサポート。
  - 最小限のロギング。Deterministic LSN 構成またはポートブロックを使用した Dynamic LSN 構成により、大規模な NAT64 ログボリュームが大幅に削減されます。
  - **MSISDN** 情報のロギング。大規模な NAT64 ログにサブスクライバの MSISDN 情報を含めることをサポートし、インターネット上でのサブスクライバアクティビティを識別して追跡します。

大規模な **NAT64** の設定で考慮すべきポイント

October 7, 2021

大規模な NAT64 および DNS64 の設定を開始する前に、次の点を考慮してください。

1. RFC で説明されている大規模な NAT64 のさまざまなコンポーネントについて理解していることを確認してください。
2. Citrix ADC アプライアンスは、大規模 NAT64 に対して以下の ALG のみをサポートします。
   - FTP
   - TFTP
   - ICMP
   - SIP
   - RTSP
3. 2 つの Citrix ADC アプライアンスの高可用性セットアップでは、大規模な NAT64 セッション同期（接続ミラーリング）はサポートされていません。

DNS64 の構成

October 7, 2021

Citrix ADC アプライアンスでステートフル NAT64 構成に必要なエンティティを作成するには、次の手順に従います。

- DNS サービスを追加します。DNS サービスは、Citrix ADC アプライアンスが DNS プロキシサーバーとして機能する DNS サーバーを論理的に表現したものです。サービスのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。
- DNS64 アクションと DNS64 ポリシーを追加し、DNS64 アクションを DNS64 ポリシーにバインドします。DNS64 ポリシーは、関連する DNS64 アクションの設定に従って、DNS64 処理のトラフィックと照合する条件を指定します。DNS64 アクションは、必須の DNS64 プレフィクスと、オプションの除外ルールとマッピングされたルールの設定を指定します。
- DNS 負荷分散仮想サーバーを作成し、DNS サービスと DNS64 ポリシーをバインドします。DNS 負荷分散仮想サーバーは、バインドされた DNS サービスによって表される DNS サーバーの DNS プロキシサーバーとして機能します。仮想サーバに着信するトラフィックは、DNS64 処理用のバインドされた DNS64 ポリシーと照合されます。負荷分散仮想サーバーのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。

> 注
>
> コマンド・ライン・インタフェースには、これら 2 つのタスクに対して個別のコマンドがありますが、GUI はそれらを単一のダイアログ・ボックスにまとめます。

- DNS レコードのキャッシュを有効にします。Citrix ADC アプライアンスのグローバルパラメータを有効にして、DNS プロキシ操作によって取得される DNS レコードをキャッシュします。DNS レコードのキャッシュの有効化の詳細については、「DNS レコードのキャッシュの有効化」を参照してください。

コマンドラインインターフェイスを使用して **DNS** タイプのサービスを作成するには

コマンドプロンプトで入力します。

```
1 add service <name> <IP> <serviceType> <port> …
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS64** アクションを作成するには

コマンドプロンプトで入力します。

```
add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <
    expression>] [-excludeRule <expression>]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS64** ポリシーを作成するには

コマンドプロンプトで入力します。

```
add dns policy64 <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS** 負荷分散仮想サーバーを作成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED
    ) [-bypassAAAA ( YES | NO)] ...
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS** サービスと **DNS64** ポリシーを **DNS** 負荷分散仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName> ...

bind lb vserver <name> -policyName <string> -priority <positive_integer
    > ...
<!--NeedCopy-->
```

設定例:

```
add service SVC-DNS-1 203.0.113.50 DNS 53
Done
add service SVC-DNS-2 203.0.113.60 DNS 53
```



```
4  Done
5  add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
6  Done
7  add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET(2001:
       DB8:5001::/64)" -action DNS64-Action-1
8  Done
9  add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
10 Done
11 bind lb vserver LBVS-DNS64-1 SVC-DNS-1
12 Done
13 bind lb vserver LBVS-DNS64-1 SVC-DNS-2
14 Done
15 bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
16 Done
17 <!--NeedCopy-->
```

大規模 **NAT64** の設定

October 7, 2021

Citrix ADC アプライアンスの大規模な NAT64 構成では、LSN コマンドセットが使用されます。大規模な NAT64 設定では、LSN クライアントエンティティは IPv6 サブスクライバを識別するための IPv6 アドレスまたは IPv6 ネットワークアドレス、または ACL6 ルールを指定します。NAT64 設定には、NAT64 プレフィクスを指定する IPv6 プロファイルも含まれています。

Citrix ADC アプライアンス上で NAT64 を構成するには、次の作業を行います。

- グローバル LSN パラメータを設定します。グローバルパラメーターには、LSN 機能用に予約されている Citrix ADC メモリの量と、高可用性セットアップでの LSN セッションの同期が含まれます。
- IPv6 サブスクライバからのトラフィックを識別するための LSN クライアントエンティティを作成します。LSN クライアントエンティティは、IPv6 サブスクライバのセットを参照します。クライアントエンティティには、これらの加入者からのトラフィックを識別するための IPv6 アドレスまたは IPv6 ネットワークプレフィクス、または ACL6 ルールが含まれます。LSN クライアントは、1 つの LSN グループにしかバインドできません。コマンドラインインターフェイスには、LSN クライアントエンティティを作成し、加入者を LSN クライアントエンティティにバインドするための 2 つのコマンドがあります。GUI では、これら 2 つの操作を 1 つの画面にまとめます。
- LSN プールを作成し、NAT IP アドレスをそれにバインドします。LSN プールは、大規模な NAT64 を実行するために Citrix ADC アプライアンスが使用する NAT IP アドレスのプールを定義します。コマンドラインインターフェイスには、LSN プールを作成し、NAT IP アドレスを LSN プールにバインドするための 2 つのコマンドがあります。GUI では、これら 2 つの操作を 1 つの画面にまとめます。

- LSN IP6 プロファイルを作成します。LSN IP6 プロファイルは、大規模な NAT64 設定用の NAT64 プレフィクスを定義します。
- （オプション）指定されたプロトコルの LSN トランスポートプロファイルを作成します。LSN トランスポートプロファイルでは、特定のプロトコルに対して加入者が持つことのできる、大規模な NAT64 セッションの最大値、ポートの最大使用量など、さまざまなタイムアウトと制限を定義します。各プロトコル (TCP、UDP、および ICMP) の LSN トランスポートプロファイルを LSN グループにバインドします。プロファイルは、複数の LSN グループにバインドできます。LSN グループにバインドされたプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN トランスポートプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのトランスポートプロファイルと呼ばれます。LSN グループにバインドした LSN トランスポートプロファイルは、そのプロトコルのデフォルトの LSN トランスポートプロファイルを上書きします。
- （オプション）指定されたプロトコルの LSN アプリケーションプロファイルを作成し、宛先ポートのセットをそれにバインドします。LSN アプリケーションプロファイルは、特定のプロトコルおよび一連の宛先ポートのグループの LSN マッピングおよび LSN フィルタリングコントロールを定義します。一連の宛先ポートでは、各プロトコル（TCP、UDP、および ICMP）の LSN プロファイルを LSN グループにバインドします。プロファイルは、複数の LSN グループにバインドできます。LSN グループにバインドされた LSN アプリケーションプロファイルは、同じグループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。デフォルトでは、すべての宛先ポートの TCP、UDP、および ICMP プロトコルのデフォルト設定を持つ 1 つの LSN アプリケーションプロファイルは、作成時に LSN グループにバインドされます。このプロファイルは、デフォルトのアプリケーションプロファイルと呼ばれます。指定された宛先ポートセットを持つ LSN アプリケーションプロファイルを LSN グループにバインドすると、バインドされたプロファイルは、その宛先ポートのセットでそのプロトコルのデフォルトの LSN アプリケーションプロファイルを上書きします。コマンドラインインターフェイスには、LSN アプリケーションプロファイルを作成し、一連の宛先ポートを LSN アプリケーションプロファイルにバインドするための 2 つのコマンドがあります。GUI では、これら 2 つの操作を 1 つの画面にまとめます。
- LSN グループを作成し、LSN プール、LSN IPv6 プロファイル、（オプション）LSN トランスポートプロファイル、および（オプション）LSN アプリケーションプロファイルを LSN グループにバインドします。LSN グループは、LSN クライアント、LSN IPv6 プロファイル、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルで構成されるエンティティです。グループには、ポートブロックサイズや LSN セッションのロギングなどのパラメータが割り当てられます。パラメータ設定は、LSN グループにバインドされた LSN クライアントのすべてのサブスクライバに適用されます。LSN グループにバインドできるのは 1 つの LSN IPv6 プロファイルだけです。LSN グループにバインドされた LSN IPv6 プロファイルは他の LSN グループにバインドできません。同じ NAT タイプ設定の LSN プールと LSN グループだけを 1 つにバインドできます。複数 LSN プールを 1 つの LSN グループにバインドできます。LSN グループ 1 つの LSN クライアントエンティティのみにバインドでき、LSN グループにバインドされた LSN クライアントエンティティは他の LSN グループにバインドできません。コマンドラインインターフェイスには、LSN グループを作成し、LSN プール、LSN トランスポートプロファイル、および LSN アプリケーションプロファイルを LSN グループにバインドするための 2 つのコマンドがあります。GUI では、これら 2 つの操作を 1 つの画面

にまとめます。

コマンドラインを使用した設定

コマンドラインインターフェイスを使用して、さまざまな設定を作成できます。以下の手順に従ってください。

コマンドラインインターフェイスを使用して **LSN** クライアントを作成するには

コマンドプロンプトで入力します。

```
add lsn client <clientname

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IPv6** ネットワークまたは **ACL6** ルールを **LSN** クライアントにバインドするには

コマンドプロンプトで入力します。

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)

show lsn client
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LAN** プールを作成するには

コマンドプロンプトで入力します。

```
add lsn pool <poolname>

show lsn pool <poolname>
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **NAT IP** アドレスを **LSN** プールにバインドするには

コマンドプロンプトで入力します。

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

> 注
>
> LSN プールから NAT IP（LSN IP アドレス）アドレスを削除するには、unbind lsn pool コマンドを使用します。

コマンドラインインターフェイスを使用して **LSN IPv6** プロファイルを構成するには

コマンドプロンプトで入力します。

```
add lsn ip6profile <name>  - type NAT64 -natprefix <ipv6_addr|*>

show lsn ip6profile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** トランスポートプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** アプリケーションプロファイルを作成するには

コマンドプロンプトで入力します。


```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )]

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、アプリケーションプロトコルポート範囲を **LSN** アプリケーションプロファイルにバインドするには

コマンドプロンプトで入力します。

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** グループを作成するには

コマンドプロンプトで入力します。

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
    DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging(
    ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
    sessionSync ( ENABLED | DISABLED )] [-snmptraplimit<positive_integer
    >] [-ftp ( ENABLED | DISABLED )] [-sipalg ( ENABLED | DISABLED )] [-
    rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **LSN** プロトコルプロファイルと **LSN** プールを **LSN** グループにバインドするには

コマンドプロンプトで入力します。


```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

大規模な **NAT64** 設定の例

次に、大規模な NAT64 の設定例を示します。

デフォルト設定を使用した単純な大規模な **NAT64** 設定:

```
add lsn client LSN-NAT64-CLIENT-1

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

add lsn pool LSN-NAT64-POOL-1

bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70

add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96

add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1

bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1

<!--NeedCopy-->
```

加入者を識別するための拡張 **ACL6** ルールを使用した単純な大規模な **NAT64** 設定:

```
add ns acl6 LSN-NAT64-ACL-2 ALLOW  - srcIPv6 = 2001:DB8:5002::20 - 2001:
    DB8:5002::200

apply acl6s

add lsn client LSN-NAT64-CLIENT-2

bind lsn client LSN-NAT64-CLIENT-2  - acl6name LSN-NAT64-ACL-2
```



```
8
9  add lsn pool LSN-NAT64-POOL-2
10
11 bind lsn pool LSN-NAT64-POOL-2 203.0.113.5-203.0.113.10
12
13 add lsn ip6profile LSN-NAT64-PROFILE-2 -type NAT64 -natprefix 2001:DB8
       :302::/96
14
15 add lsn group LSN-NAT64-GROUP-2 -clientname LSN-NAT64-CLIENT-2 -
       ip6profile LSN-NAT64-PROFILE-2
16
17 bind lsn group LSN-NAT64-GROUP-2 -poolname LSN-NAT64-POOL-2
18
19 <!--NeedCopy-->
```

Deterministic NAT リソース割り当てを使用する大規模な **NAT64** 設定:

```
1  add lsn client LSN-NAT64-CLIENT-7
2
3  bind lsn client LSN-NAT64-CLIENT-7 -network6 2001:DB8:1002::7/128
4
5  add lsn pool LSN-NAT64-POOL-7 -nattype DETERMINISTIC
6
7  bind lsn pool LSN-NAT64-POOL-7 203.0.113.24-203.0.113.27
8
9  add lsn ip6profile LSN-NAT64-PROFILE-7 -type NAT64 -natprefix 2001:DB8
       :307::/96
10
11 add lsn group LSN-NAT64-GROUP-7 -clientname LSN-NAT64-CLIENT-7 -
       ip6profile LSN-NAT64-PROFILE-7 -nattype DETERMINISTIC -portblocksize
        256
12
13 bind lsn group LSN-NAT64-GROUP-7 -poolname LSN-POOL-7
14
15 <!--NeedCopy-->
```

大規模な **NAT64** のアプリケーション層ゲートウェイの設定

October 7, 2021

一部のアプリケーション層プロトコルでは、IP アドレスとプロトコルポート番号もパケットペイロードで通信され

ます。プロトコルのアプリケーション層ゲートウェイは、パケットのペイロードを解析し、プロトコルが大規模な NAT64 上で動作し続けるように必要な変更を行います。

Citrix ADC アプライアンスは、大規模 NAT64 用に次のプロトコルで ALG をサポートしています。

- FTP
- ICMP
- TFTP
- SIP
- RTSP

FTP、**ICMP**、および **TFTP** プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

大規模な NAT64 設定の FTP プロトコルの ALG を有効または無効にするには、設定の LSN グループの FTP ALG オプションを有効または無効にします。

ICMP プロトコルの ALG はデフォルトで有効になっており、無効にするプロビジョニングはありません。

TFTP プロトコルの ALG は、デフォルトで無効になっています。エンドポイント独立マッピング、エンドポイント独立フィルタリング、および宛先ポートを 69（TFTP の場合は既知のポート）として UDP LSN アプリケーションプロファイルを LSN グループにバインドすると、大規模な NAT64 設定に対して TFTP ALG が自動的に有効になります。

SIP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

セッション開始プロトコル（SIP）で大規模な NAT64 を使用するのは複雑です。これは、SIP メッセージに SIP ヘッダーと SIP 本文に IP アドレスが含まれているためです。LSN を SIP とともに使用すると、SIP ヘッダーには発信者と受信者の情報が含まれ、デバイスはこの情報を変換して外部ネットワークから隠します。SIP 本体には、セッション記述プロトコル（SDP）情報が含まれています。この情報には、メディアを送信するための IP アドレスとポート番号が含まれます。大規模な NAT64 用の SIP ALG は、RFC 3261、RFC 3581、RFC 4566、および RFC 4475 に準拠しています。

> 注
>
> SIP ALG は、Citrix ADC スタンドアロンアプライアンス、Citrix ADC 高可用性セットアップ、および Citrix ADC クラスタセットアップでサポートされています。

SIP ALG の制限事項

大規模な NAT64 用の SIP ALG には、次の制限があります。

- SDP ペイロードだけがサポートされます。
- 以下はサポートされていません:
  - マルチキャスト IP アドレス
  - 暗号化 SDP
  - SIP TLS
  - 完全修飾ドメイン名変換
  - SIP レイヤ認証
  - トラフィックドメイン
  - 管理パーティション
  - マルチパートボディ
  - 折り畳み

SIP ALG の設定

LSN 設定の一部として SIP ALG を設定する必要があります。LSN の設定手順については、「大規模な NAT64 の設定」を参照してください。LSN の設定時には、次のことを確認してください。

- LSN アプリケーションプロファイルを追加するときに、次のパラメータを設定します。
  - IP プーリング = PAIRED
  - アドレスとポートのマッピング = エンドポイント独在
  - フィルタリング = エンドポイントインデペンデント
- SIP ALG プロファイルを作成し、送信元ポート範囲または宛先ポート範囲を定義していることを確認します。SIP ALG プロファイルを LSN グループにバインドします。
- LSN グループで SIP ALG を有効にします。

CLI を使用して LSN 設定で SIP ALG を有効にするには

コマンドプロンプトで入力します。

```
1 add lsn group <groupname> -clientname <string> [-sipalg ( ENABLED |
     DISABLED )]
2
3 show lsn group <groupname>
4 <!--NeedCopy-->
```


CLI を使用して **LSN** 設定で **SIP ALG** を有効にするには

コマンドプロンプトで入力します。

```
add lsn sipalgprofile <sipalgprofilename>[-dataSessionIdleTimeout <
    positive_integer>][-sipSessionTimeout <positive_integer>] [-
    registrationTimeout <positive_integer>] [-sipsrcportrange <port[-
    port]>] [-sipdstportrange <port[-port]>] [-openRegisterPinhole (
    ENABLED | DISABLED )] [-openContactPinhole ( ENABLED | DISABLED )]
    [-openViaPinhole ( ENABLED | DISABLED )] [-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP ) [-
    openRoutePinhole ( ENABLED | DISABLED )] [-rport ( ENABLED |
    DISABLED )]

show lsn sipalgprofile <sipalgprofilename
<!--NeedCopy-->
```

構成例

次の大規模な NAT64 設定の例では、ネットワーク 2001: DB 8:1003:: /96 内のサブスクライバデバイスからの TCP トラフィックに対して SIP ALG が有効になっています。

```
add lsn client LSN-NAT64-CLIENT-9

Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96

Done
add lsn pool LSN-NAT64-POOL-9

Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90

Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
    :309::/96

Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

```



```
18 Done
19 add lsn sipalgprofile SIPALGPROFILE-9  -sipdstportrange 5060  -
      sipTransportProtocol TCP
20
21 Done
22 add lsn group LSN-NAT64-GROUP-9 -clientnameLSN-NAT64-CLIENT-9 -
      ip6profile LSN-NAT64-PROFILE-7 -sipalg ENABLED
23
24 Done
25 bind lsn group LSN-NAT64-GROUP-9 -poolnameLSN-NAT64-POOL-9
26 Done
27 bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
      PROFILE-9
28 Done
29 bind lsn group LSN-NAT64-GROUP-9 -sipalgprofilename SIPALGPROFILE-9
30 Done
31 <!--NeedCopy-->
```

RTSP プロトコルのアプリケーション層ゲートウェイ

October 7, 2021

リアルタイムストリーミングプロトコル (RTSP) は、リアルタイムメディアデータを転送するためのアプリケーションレベルのプロトコルです。RTSP は、エンドポイント間のメディアセッションを確立および制御するために使用され、メディアクライアントとメディアサーバ間のコントロールチャネルプロトコルです。一般的な通信は、クライアントとストリーミングメディアサーバーの間です。

プライベートネットワークからパブリックネットワークにメディアをストリーミングするには、ネットワーク上で IP アドレスとポート番号を変換する必要があります。Citrix ADC の機能には、RTSP 用のアプリケーションレイヤーゲートウェイ（ALG）が含まれています。ALG（大規模 NAT）とともに使用すると、メディアストリームを解析し、プロトコルがネットワーク上で動作し続けるように必要な変更を行うことができます。

IP アドレス変換の実行方法は、メッセージのタイプと方向、およびクライアント/サーバ展開でサポートされるメディアのタイプによって異なります。メッセージは次のように翻訳されます。

- 送信要求：Citrix ADC が所有するパブリック IP アドレスへのプライベート IP アドレス（LSN IP アドレスと呼ばれます）。
- 着信応答：プライベート IP アドレスへの LSN IP アドレス。
- インバウンド要求：変換なし。
- 発信応答：LSN プール IP アドレスへのプライベート IP アドレス。

> 注
>
> RTSP ALG は、Citrix ADC スタンドアロンアプライアンス、Citrix ADC 高可用性セットアップ、および Citrix ADC クラスタセットアップでサポートされています。

RTSP ALG の制限事項

RTSP ALG は、次の機能をサポートしていません。

- マルチキャスト RTSP セッション
- UDP を介した RTSP セッション
- 管理パーティション
- RTSP 認証
- HTTP トンネリング

RTSP ALG の設定

RTSP ALG を LSN 設定の一部として設定します。LSN の設定手順については、「大規模な NAT64 の設定」を参照してください。設定中に、次のことを確認してください。

- LSN アプリケーションプロファイルを追加するときに、次のパラメータを設定します。
  - IP プーリング = PAIRED
  - アドレスとポートのマッピング = エンドポイント独在
  - フィルタリング = エンドポイントインデペンデント
- LSN グループで RTSP ALG を有効にします。
- RTSP ALG プロファイルを作成し、RTSP ALG プロファイルを LSN グループにバインドします。

CLI を使用して LSN 設定の RTSP ALG を有効にするには

コマンドプロンプトで入力します。

```
1 add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
     DISABLED )]
2
3 show lsn group <groupname>
4 <!--NeedCopy-->
```

CLI を使用して LSN 設定の RTSP ALG を有効にするには

コマンドプロンプトで入力します。


```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

RTSP ALG 設定の例

次に、大規模な NAT64 設定の例を示します。RTSP ALG は、ネットワーク 2001: DB 8:1002:: /96 内のサブスクライバデバイスからの TCP トラフィックに対して有効になっています。

```
add lsn client LSN-NAT64-CLIENT-9
Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96
Done
add lsn pool LSN-NAT64-POOL-9
Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90
Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
    :309::/96
Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-9 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-NAT64-GROUP-9 -clientname LSN-NAT64-CLIENT-9 -
    ip6profile LSN-NAT64-PROFILE-7 -rtspalg ENABLED
Done
bind lsn group LSN-NAT64-GROUP-9 -poolname LSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
    PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9  -rtspalgprofilename RTSPALGPROFILE-9
Done
<!--NeedCopy-->
```


スタティック大規模 **NAT64** マップの設定

October 7, 2021

Citrix ADC アプライアンスは、NAT64 マッピングの手動作成をサポートしています。このマッピングには、以下の情報間のマッピングが含まれます。

- 加入者の IP アドレスとポート
- NAT IP アドレスとポート

スタティック大規模な NAT64 マッピングは、NAT IP アドレス:port に対して開始された IPv4 接続が IPv6 変換され、加入者 IP アドレス: ポート（内部ネットワークにある Web サーバなど）にマッピングされていることを確認する場合に便利です。

コマンドラインを使用して大規模な **NAT64** マッピングを作成するには

コマンドプロンプトで入力します。

```
1 add lsn static <name> <transportprotocol> <subscrIP> <subscrPort>  [<
     natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd <positive_integer>]]
2
3 show lsn static
4 <!--NeedCopy-->
```

ワイルドカードポートスタティック大規模 **NAT64** マップ

スタティック大規模 NAT64 マッピングエントリは、通常、サブスクライバ IPv6 アドレス: ポートと NAT IPv4 アドレス: ポートの間の 1 対 1 のマッピングです。1 対 1 のスタティック大規模 NAT64 マッピングエントリは、加入者 IP アドレスの 1 つのポートだけをインターネットに公開します。

場合によっては、加入者 IP アドレスのすべてのポート（64 K-NAT IPv4 アドレスの最大ポート数に制限）をインターネットに公開する必要があります（たとえば、内部ネットワークでホストされ、各ポートで異なるサービスを実行しているサーバ）。インターネットを介してこれらの内部サービスにアクセスできるようにするには、サーバーのすべてのポートをインターネットに公開する必要があります。

この要件を満たす方法の 1 つは、64,000 個の 1 対 1 スタティックマッピングエントリ（各ポートに 1 つのマッピングエントリ）を追加することです。これらのエントリを作成することは非常に面倒で大きな作業です。また、この多数の構成エントリは、Citrix ADC アプライアンスのパフォーマンスの問題を引き起こす可能性があります。

より簡単な方法は、スタティックマッピングエントリでワイルドカードポートを使用することです。NAT ポートパラメータとサブスクライバポートパラメータをワイルドカード文字（*）に設定し、protocol パラメータを ALL に設定

して、すべてのプロトコルのサブスクライバ IP アドレスのすべてのポートをインターネットに公開するスタティックマッピングエントリを 1 つ作成するだけで済みます。

ワイルドカードスタティックマッピングエントリと一致するサブスクライバのインバウンド接続またはアウトバウンドコネクションの場合、サブスクライバのポートは NAT 操作後に変更されません。インターネットへの加入者が開始する接続がワイルドカード静的マッピングエントリと一致する場合、Citrix ADC アプライアンスは、接続を開始する加入者ポートと同じ番号の NAT ポートを割り当てます。同様に、インターネットホストは、加入者のポートと同じ番号を持つ NAT ポートに接続することによって、加入者のポートに接続されます。

サブスクライバ IPv6 アドレスのすべてのポートへのアクセスを提供するように Citrix ADC アプライアンスを構成するには、次の必須パラメータ設定を使用してワイルドカード静的マップを作成します。

- プロトコル =ALL
- サブスクライバポート = *
- NAT ポート = *

ワイルドカードスタティックマップでは、1 対 1 のスタティックマップとは異なり、NAT IP パラメータの設定は必須です。また、ワイルドカードスタティックマップに割り当てられた NAT IP アドレスは、他の加入者には使用できません。

コマンドラインインターフェイスを使用してワイルドカード静的マップを作成するには

コマンドプロンプトで入力します。

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
   positive_integer>] [-destIP <ip_addr>

show lsn static
<!--NeedCopy-->
```

次のワイルドカードスタティックマップの設定例では、IP アドレスが 2001: DB 8:5001:: 3 であるサブスクライバのすべてのポートが、NAT IP 203.0.11.33 を介してアクセス可能になります。

```
add lsn static NAT64-WILDCARD-STATIC-1 ALL 2001:DB8:5001::3 *
   203.0.113.33 *
Done
<!--NeedCopy-->
```

大規模な **NAT64** のロギングとモニタリング

October 7, 2021

大規模な NAT64 情報をログに記録して、問題の診断とトラブルシューティング、法的要件を満たすことができます。統計カウンタを使用して、関連する現在のセッションを表示することで、大規模な NAT64 展開のパフォーマンスを監視できます。

大規模な **NAT64** のロギング

ISP が法的要件を満たし、いつでもトラフィックの送信元を特定するには、大規模な NAT64 情報のロギングが必要です。

大規模な NAT64 マッピングエントリのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）。
- タイムスタンプ。
- エントリタイプ (MAPPING)。
- マッピング・エントリが作成されたか、削除されたか。
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID。
- NAT の IP アドレスとポート。
- プロトコル名。
- 次の条件によっては、宛先 IP アドレス、ポート、およびトラフィックドメイン ID が存在する場合があります。
  - エンドポイントに依存しないマッピングの場合、宛先 IP アドレスおよびポートはログに記録されません。
  - 宛先 IP アドレスのみがアドレス依存マッピング用にログに記録されます。ポートはログに記録されません。
  - 宛先 IP アドレスとポートは、アドレスポートに依存するマッピング用にログに記録されます。

大規模な NAT64 セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- タイム・スタンプ
- エントリの種類 (SESSION)
- セッションが作成されるか、削除されるか
- 加入者の IP アドレス、ポート、およびトラフィックドメイン ID
- NAT IP アドレスとポート
- プロトコル名
- 宛先 IP アドレス、ポート、およびトラフィックドメイン ID

次の表に、構成済みのログサーバに保存されている各タイプの大規模な NAT64 ログエントリの例を示します。ログエントリには、IPv6 アドレスが 2001: db 8:5001:: 9 であるサブスクライバが、宛先 IP: ポート 23.0.0. 1:80 を介して接続されていたことが示されています。2016 年 4 月 7 日の 63:45195（グリニッジ標準時 14:07:57 から 14:10:59 まで）。

ログエントリタイプ	サンプルログエントリ
セッションの作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
マッピングの作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
セッションの削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
マッピングの削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

構成の手順

LSN グループのロギングパラメータとセッションロギングパラメータを設定することで、大規模な NAT64 設定用の大規模な NAT64 情報のロギングを設定できます。これらはグループレベルのパラメータであり、デフォルトでは無効になっています。Citrix ADC アプライアンスは、LSN グループの大規模な NAT64 セッションを記録するのは、ロギングとセッションロギングの両方のパラメータが有効になっている場合のみです。

次の表は、ロギングパラメータとセッションロギングパラメータのさまざまな設定に対する LSN グループのロギング動作を示しています。

ログ	セッションロギング	ロギングの動作
有効	有効	LSN マッピングエントリと LSN セッションを記録します。

ログ	セッションロギング	ロギングの動作
有効	無効	LSN マッピングエントリを記録しますが、LSN セッションは記録しません。
無効	有効	マッピングエントリも LSN セッションもログに記録しません。

CLI を使用して大規模な **NAT64** 情報をログに記録するには

LSN グループの追加時にロギングおよびセッションロギングのパラメータを設定するには、コマンドプロンプトで次のように入力します。

```
add lsn group <groupname> -clientname <string> [-logging (ENABLED|
    DISABLED)] [-sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

既存の LSN グループのロギングパラメータとセッションロギングパラメータを設定するには、コマンドプロンプトで次のように入力します。

```
set lsn group <groupname> [-logging (ENABLED|DISABLED)] [-
    sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

構成例

この大規模な NAT64 設定の例では、LSN グループ LSN-NAT64-GROUP-1 に対してロギングパラメータとセッションロギングパラメータが有効になっています。

Citrix ADC アプライアンスは、加入者からの接続に関する大規模な NAT64 セッションおよびマッピング情報をログに記録します（ネットワーク 2001:DB 8:5001:: /96）。

設定例:


```
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1  -logging ENABLED -sessionLogging
    ENABLED
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
<!--NeedCopy-->
```

大規模な **NAT64** の **MSISDN** 情報のロギング

モバイルステーション統合サブスクライバディレクトリ番号（MSISDN）は、複数のモバイルネットワークにわたるサブスクライバを一意に識別する電話番号です。MSISDN は、加入者のオペレータを識別する国コードおよび国別宛先コードに関連付けられています。

Citrix ADC アプライアンスは、モバイルネットワークの加入者の大規模な NAT64 LSN ログエントリに MSISDN を含めるように構成できます。LSN ログに MSISDN が存在すると、ポリシーまたは法律に違反したモバイル加入者、または合法的な傍受機関によって情報が要求されるモバイル加入者の迅速かつ正確なバックトレースが容易になります。

次の LSN ログエントリの例には、LSN 設定のモバイルサブスクライバからの接続の MSISDN 情報が含まれています。ログエントリには、MSISDN が E 164:5556543210 で、IPv6 アドレスが 2001: デシベル 8:5001:: 9 であるモバイルサブスクライバが、NAT IP を介して宛先 IP: ポート 23.0.0. 1:80 に接続されていたことが示されています。2016 年 4 月 7 日の 14:07:57 GMT から 14: 10:59 までの 63:45195 です。

ログエントリタイプ	サンプルログエントリ
セッションの作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

ログエントリタイプ	サンプルログエントリ
マッピングの作成	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
セッションの削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
マッピングの削除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

構成の手順

LSN ログに MSISDN 情報を含めるには、次のタスクを実行します

- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、LSN 設定の LSN ログに MSISDN 情報を含めるかどうかを指定するログサブスクライバ ID パラメータが含まれます。
- LSN ログプロファイルを LSN 構成の LSN グループにバインドします。作成された LSN ログプロファイル名パラメーターを作成された LSN ログプロファイル名に設定して、LSN 構成の LSN グループにバインドします。MSISDN 情報は、この LSN グループのモバイルサブスクライバに関連するすべての LSN ログに含まれています。

CLI を使用して **LSN** ログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename> -logSubscriberID ( ENABLED |
    DISABLED )

```



```
show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して、**LSN** ログプロファイルを **NAT64 LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

構成例

この NAT64 LSN 設定の例では、LSN ログプロファイル LOG-PROFILE-MSISDN-1 で、ログサブスクライバ ID パラメータが有効になっています。LOG-PROFILE-MSISDN-1 は、LSN グループ LSN-NAT64 グループ-1 にバインドされています。MSISDN 情報は、モバイルサブスクライバ（ネットワーク 2001:DB 8:5001:: /96）からの接続の LSN セッションログおよび LSN マッピングログに含まれます。

```
add lsn logprofile  LOG-PROFILE-MSISDN-1  -logSubscriberID ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -logprofilename  LOG-PROFILE-MSISDN-1
Done
<!--NeedCopy-->
```


大規模な **NAT** のコンパクトロギング

LSN 情報のロギングは、ISP が法的要件を満たし、いつでもトラフィックの送信元を特定できるようにするために必要な重要な機能の 1 つです。その結果、ログデータが大量に作成され、ISP はログインフラストラクチャを維持するために多額の投資をする必要があります。

コンパクトログは、イベント名とプロトコル名の短いコードを含む表記変更を使用して、ログサイズを小さくする手法です。たとえば、クライアントの場合は C、作成されたセッションの場合は SC、TCP の場合は T を指定します。コンパクトログでは、ログサイズが平均 40% 削減されます。

構成の手順

コンパクト形式で LSN 情報をロギングするには、次の作業を実行します。

1. LSN ログプロファイルを作成します。LSN ログプロファイルには、LSN 構成の情報をコンパクト形式で記録するかどうかを指定する Log Compact パラメーターが含まれています。
2. LSN ログプロファイルを LSN 構成の LSN グループにバインドします。作成された LSN ログプロファイル名に [ログプロファイル名] パラメーターを設定して、作成された LSN ログプロファイルを LSN 構成の LSN グループにバインドします。この LSN グループのすべてのセッションとマッピングは、コンパクト形式で記録されます。

CLI を使用して **LSN** ログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

NAT64 の設定例:


```
add lsn logprofile  LOG-PROFILE-COMPACT-1 -logCompact ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
   :300::/96
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
   ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - logProfileName LOG-PROFILE-COMPACT-1
Done
<!--NeedCopy-->
```

HTTP ヘッダー情報のロギング

Citrix ADC アプライアンスは、Citrix ADC の大規模な NAT64 機能を使用している HTTP 接続の要求ヘッダー情報をログに記録できます。HTTP 要求パケットの次のヘッダー情報をログに記録できます。

- HTTP リクエストの宛先となる URL
- HTTP リクエストで指定された HTTP メソッド
- HTTP リクエストで使用される HTTP バージョン
- HTTP 要求を送信したサブスクライバの IPv6 アドレス

ISP は HTTP ヘッダーログを使用して、一連のサブスクライバ間の HTTP プロトコルに関連する傾向を確認できます。たとえば、ISP はこの機能を使用して、一連の加入者の中で最も人気のある Web サイトを見つけることができます。

構成の手順

HTTP ヘッダー情報をログに記録するように Citrix ADC アプライアンスを構成するには、次のタスクを実行します。

- HTTP ヘッダーログプロファイルを作成します。HTTP ヘッダーログプロファイルは、ロギングを有効または無効にできる HTTP ヘッダー属性（URL や HTTP メソッドなど）のコレクションです。

- HTTP ヘッダーを大規模な NAT64 設定の LSN グループにバインドします。HTTP ヘッダーログプロファイル名パラメータを作成された HTTP ヘッダーログプロファイルの名前に設定することにより、HTTP ヘッダーログプロファイルを LSN 設定の LSN グループにバインドします。次に、Citrix ADC アプライアンスは、LSN グループに関連するすべての HTTP 要求の HTTP ヘッダー情報をログに記録します。HTTP ヘッダーログプロファイルは、複数の LSN グループにバインドできますが、LSN グループには 1 つの HTTP ヘッダーログプロファイルしか設定できません。

コマンドラインインターフェイスを使用して **HTTP** ヘッダーログプロファイルを作成するには

コマンドプロンプトで入力します。

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、**HTTP** ヘッダーログプロファイルを **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

構成例

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
```



```
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -httphdrlogprofilename HTTP-HEADER-LOG
    -1
Done
<!--NeedCopy-->
```

現在の大規模な **NAT64** セッションの表示

Citrix ADC アプライアンス上の不要なセッションや非効率的なセッションを検出するために、現在の大規模 NAT64 セッションを表示できます。選択パラメータに基づいて、大規模な NAT64 セッションすべてまたは一部を表示できます。

> 注
>
> Citrix ADC アプライアンス上に 100 万を超える大規模 NAT64 セッションが存在する場合は、選択パラメータを使用して選択した大規模 NAT64 セッションをすべて表示するのではなく、選択パラメータを使用することをお勧めします。

コマンドラインインターフェイスを使用して大規模な **NAT64** セッションをすべて表示するには

コマンドプロンプトで入力します。

```
show lsn session  -nattype NAT64
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、選択的な大規模な **NAT64** セッションを表示するには

コマンドプロンプトで入力します。

```
show lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
     <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```


大規模な **NAT64** 統計情報の表示

大規模な NAT64 モジュールに関連する統計情報を表示し、そのパフォーマンスを評価したり、問題のトラブルシューティングを行うことができます。すべての大規模な NAT64 設定または特定の大規模な NAT64 設定の統計情報のサマリーを表示できます。統計カウンタには、Citrix ADC アプライアンスが最後に再起動されてからのイベントが反映されます。Citrix ADC アプライアンスが再起動されると、これらのカウンタはすべて 0 にリセットされます。

コマンドラインインターフェイスを使用して大規模な **NAT64** の統計情報の合計を表示するには

コマンドプロンプトで入力します。

```
stat lsn nat64
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、指定した大規模な **NAT64** 設定の統計情報を表示するには

コマンドプロンプトで入力します。

```
stat lsn group <groupname>
<!--NeedCopy-->
```

大規模な **NAT64** セッションのクリア

不要なまたは非効率的な大規模 NAT64 セッションを Citrix ADC アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレス、ポート、メモリなど）をただちに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、これらの削除されたセッションに関連する後続のパケットもすべてドロップします。Citrix ADC アプライアンスから、すべてまたは選択した大規模 NAT64 セッションを削除できます。

コマンドラインインターフェイスを使用して大規模な **NAT64** セッションをすべてクリアするには

コマンドプロンプトで入力します。

```
flush lsn session  -nattype NAT64

show lsn session  -nattype NAT64
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して選択的な大規模な **NAT64** セッションをクリアするには

コマンドプロンプトで入力します。

```
flush lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-
   clientname <string>] [-natIP <ip_addr> [-natPort <port>]]

show lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
    <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

設定例:

Citrix ADC アプライアンス上に存在するすべての大規模 NAT64 セッションをクリアする

```
flush lsn session   -nattype NAT64
Done
<!--NeedCopy-->
```

クライアントエンティティ LSN-NAT64-CLIENT-1 に関連する大規模な NAT64 セッションをすべてクリアします。

```
flush lsn session  -nattype NAT64 -clientname LSN-NAT64-CLIENT-1
Done
<!--NeedCopy-->
```

LSN クライアントエンティティ LSN-NAT64 CLIENT-2 の加入者ネットワーク（2001:DB 8:5001:: /96）に関連する大規模な NAT64 セッションをすべてクリアします。

```
flush lsn session  -nattype NAT64  -network6 2001:DB8:5001::/96 -
   clientname LSN-NAT64-CLIENT-2
Done
<!--NeedCopy-->
```

IPFIX ロギング

Citrix ADC アプライアンスは、LSN イベントに関する情報をインターネット・プロトコル・フロー情報エクスポート（IPFIX）形式で IPFIX コレクタで構成されたセットに送信できます。アプライアンスは、既存の AppFlow 機能を使用して、IPFIX 形式の LSN イベントを IPFIX コレクタに送信します。

IPFIX ベースのロギングは、次の NAT64 関連イベントで使用できます。

- LSN セッションの作成または削除。
- LSN マッピングエントリの作成または削除。
- Deterministic NAT のコンテキストにおけるポートブロックの割り当てまたは割り当て解除。
- Dynamic NAT のコンテキストにおけるポートブロックの割り当てまたは割り当て解除。
- サブスクライバセッションクォータを超えた場合。

IPFIX ロギングを構成する前に考慮すべきポイント

IPSec ALG の設定を開始する前に、次の点を考慮してください。

- Citrix ADC アプライアンスで AppFlow 機能および IPFIX コレクタを構成する必要があります。手順については、AppFlow 機能の構成を参照してください。

構成の手順

LSN 情報を IPFIX 形式でロギングするには、次の作業を実行します。

- **AppFlow** 構成で **LSN** ロギングを有効にします。AppFlow 構成の一部として LSN ロギングパラメータを有効にします。
- **LSN** ログプロファイルを作成します。LSN ログプロファイルには、IPFIX 形式のログ情報を有効または無効にする IPFIX パラメータが含まれています。
- **LSN** ログプロファイルを **LSN** 構成の **LSN** グループにバインドします。LSN ログプロファイルを 1 つまたは複数の LSN グループにバインドします。バインドされた LSN グループに関連するイベントは、IPFIX 形式で記録されます。

CLI を使用して **AppFlow** 構成で **LSN** ロギングを有効にするには

コマンドプロンプトで入力します。

```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを作成するにはコマンドプロンプトで

コマンドプロンプトで入力します。

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

```



```
show lsn logprofile
<!--NeedCopy-->
```

CLI を使用して **LSN** ログプロファイルを **LSN** 設定の **LSN** グループにバインドするには

コマンドプロンプトで入力します。

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

GUI を使用して **LSN** ログプロファイルを作成するには

[システム] > [大規模 **NAT**] > [プロファイル] に移動し、[ログ] タブをクリックしてログプロファイルを追加します。

GUI を使用して **LSN** ログプロファイルを **LSN** 構成の **LSN** グループにバインドするには

1. [システム] > [大規模 **NAT**] > [**LSN** グループ**]** に移動し、**LSN** グループを開きます。
2. [詳細設定**]** で、 [**+** ログプロファイル] をクリックして、作成したログプロファイルを LSN グループにバインドします。

大規模な **NAT64** 用ポート制御プロトコル

October 7, 2021

Citrix ADC アプライアンスは、大規模 NAT（LSN）用のポート制御プロトコル（PCP）をサポートするようになりました。ISP の加入者アプリケーションの多くは、インターネットからアクセスできる必要があります (インターネット経由で監視する IP カメラなど、モノのインターネット (IOT) デバイスなど)。この要件を満たす方法の 1 つは、スタティック大規模 NAT（LSN）マップを作成することです。しかし、非常に多数のサブスクライバの場合、スタティック LSN NAT マップを作成することは実現可能な解決策ではありません。

Port Control Protocol（PCP; ポート制御プロトコル）を使用すると、加入者は、自身または他のサードパーティデバイスに対して特定の LSN NAT マッピングを要求できます。大規模な NAT デバイスは LSN マップを作成し、加入者に送信します。加入者は、インターネット上のリモートデバイスに、加入者に接続できる NAT IP アドレス:NAT ポートを送信します。

通常、アプリケーションは LSN マッピングがタイムアウトしないように、大規模な NAT デバイスに頻繁にキープアライブメッセージを送信します。PCP は、アプリケーションが LSN マッピングのタイムアウト設定を学習できるよ

うにすることで、このようなキープアライブメッセージの頻度を減らすのに役立ちます。これにより、ISP のアクセスネットワークの帯域幅消費とモバイルデバイスのバッテリ消費を削減できます。

PCP はクライアント/サーバーモデルで、UDP トランスポートプロトコル上で動作します。Citrix ADC アプライアンスは、PCP サーバーコンポーネントを実装し、RFC 6887 に準拠しています。

構成の手順

PCP を設定するには、次の作業を実行します。

- （オプション）**PCP** プロファイルを作成します。PCP プロファイルには、PCP 関連パラメータの設定（たとえば、マッピングおよびピア PCP 要求をリッスンする）が含まれます。PCP プロファイルを PCP サーバーにバインドできます。PCP サーバーにバインドされた PCP プロファイルは、すべての設定を PCP サーバーに適用します。PCP プロファイルは、複数の PCP サーバにバインドできます。デフォルトでは、デフォルトのパラメータ設定を持つ 1 つの PCP プロファイルがすべての PCP サーバーにバインドされます。PCP サーバーにバインドした PCP プロファイルは、そのサーバーのデフォルトの PCP プロファイル設定よりも優先されます。デフォルトの PCP プロファイルには、次のパラメータ設定があります。
  - マッピング: 有効
  - ピア: 有効
  - 最小マップ寿命:120 秒
  - 最大マップ寿命:86400 秒。
  - アナウンス数:10
  - サードパーティ: 無効
- **PCP** サーバーを作成し、**PCP** プロファイルをバインドします。Citrix ADC アプライアンス上に PCP サーバーを作成し、PCP 関連の要求とサブスクライバーからのメッセージをリッスンします。サブネット IP (SNIP) または (SNIP6) アドレスは、PCP サーバーにアクセスするために割り当てる必要があります。デフォルトでは、PCP サーバーはポート 5351 をリッスンします。
- **PCP** サーバを **LSN** 設定の **LSN** グループにバインドします。PCP Server パラメータを設定して、作成した PCP サーバーを指定することで、作成した PCP サーバーを LSN 構成の LSN グループにバインドします。作成された PCP サーバにアクセスできるのは、この LSN グループのサブスクライバだけです。

> 注
>
> 大規模な NAT 設定の PCP サーバは、ACL ルールで識別されるサブスクライバからの要求を処理しません。

CLI を使用して **PCP** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
```



```
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

CLI を使用して **PCP** サーバーを作成するには

コマンドプロンプトで入力します。

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

NAT64 の設定例

次の設定例では、PCP サーバ PCP-SERVER-1 は、PCP プロファイル-1 からの PCP 設定で、LSN グループ LSN-NAT64-GROUP-1 にバインドされています。PCP-SERVER-1 は、ネットワーク 2001: DB 8:5001:: /96 内の IPv6 サブスクライバからの PCP 要求を処理します。

設定例:

```
add pcp profile PCP-PROFILE-1 -minMapLife 400
Done
add pcp server PCP-SERVER-1 2001:DB8:6001::90  -pcpProfile PCP-PROFILE
    -1
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
```



```
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - pcpServer PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

クラスタ設定の **LSN64**

October 7, 2021

Citrix ADC クラスターセットアップでは、大規模な NAT64 構成がサポートされます。

ACitrix ADC クラスタは、単一のシステムとして構成および管理される Citrix ADC アプライアンスのグループです。Citrix ADC クラスターは、スケーラビリティと可用性を提供します。クラスターセットアップ内の各 Citrix ADC アプライアンスは、独立した LSN エンティティとして機能し、単一のシステムとして管理されます。

クラスタセットアップの LSN 構成は、スタンドアロンアプライアンスと同じです。ただし、LSN IP アドレスの特定のプールは、一度に 1 つのノードのみによって所有されます。つまり、LSN IP プールエンティティは、特定のノードでスポッティングエンティティとして設定されます。クラスタセットアップのすべてのノードは、特定の LSN IP プールエンティティを持つことができます。LSN セッションに関連するパケットが NAT 操作を実行したのと同じクラスタノードで確実に受信されるようにするには、ポリシーベースのバックプレーン (PBS) ステアリングが構成されています。PBS は、LSN セッションの受信した関連パケットを同じクラスタノードに操縦します。

設定例:

```
add lsn client LSN-NAT64-CLIENT-1

Done

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

Done

add lsn pool LSN-NAT64-POOL-1

Done

bind lsn pool LSN-NAT64-POOL-1 -ownerNode 1  203.0.113.61 -
    203.0.113.70
```



```
14
15 Done
16
17 bind lsn pool LSN-NAT64-POOL-1 -ownerNode 2  203.0.113.101 -
      203.0.113.110
18
19 Done
20
21 add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
      :300::/96
22
23 Done
24
25 add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
      ip6profile LSN-NAT64-PROFILE-1
26
27 Done
28
29 bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
30
31 Done
32
33 add ns acl6 NAT64-DFD ALLOW -srcIPv6 = 2001:DB8:5001:: -type DFD -
      dfdhash SIP -dfdprefix 64
34
35 Done
36
37 apply ns acls6 -type DFD
38
39 Done
40 <!--NeedCopy-->
```

変換を使用したアドレスとポートのマッピング

October 7, 2021

変換を使用してアドレスとポートのマッピング（MAP-T）は、IPv4 インターネット_.MAP-T に彼らの IPv4 加入者を接続するための IPv6 インフラストラクチャを持つ ISP のための IPv6 移行ソリューションは、ステートレス IPv4 および IPv6 アドレス変換技術上に構築されています。MAP-T は、カスタマーエッジ（CE）デバイスおよびボーダールータ（ISP コアネットワーク内）で二重変換（IPv4 から IPv6 へ、またはその逆）を実行するメカニズムです。

MAP-T 配置では、CE デバイスはステートフル NAPT44 変換とステートレス NAT46 変換の組み合わせを実装します。CE デバイスは、NAT-IP およびポートブロックを取得し、DHCPv6 またはその他の方法による変換に使用されます。

加入者デバイスからの IPv4 パケットが CE デバイスに到着すると、CE デバイスは NAPT44 を実行し、NAPT44 バインディング情報を保存します。NAT44 変換後、パケットは NAT46 変換の対象となり、ISP のコアネットワークにある BR（ボーダールータ）デバイスに転送されます。BR デバイスは CE デバイスから IPv6 パケットを受信し、IPv6 ヘッダーに埋め込まれた NAT-IP およびポートブロックを抽出して検証し、IPv4 パケットを IPv4 インターネットに転送します。BR はインターネットから IPv4 パケットを受信すると、IPv4 パケットを IPv6 パケットに変換し、IPv6 パケットを CE デバイスに送信します。

MAP-T は BR デバイスではステートレスであるため、BR デバイスがトラフィックに対して NAT を実行する必要はありません。代わりに、NAT 機能は CE デバイスに委任されます。BR デバイスのこの委任およびステートレス機能によって、BR 配置はトラフィックの量に比例して拡張できます。

Citrix ADC アプライアンスは、RFC 7599 で説明されているように、MAP-T ソリューションの BR 機能を実装しています。

MAP-T の設定

Citrix ADC アプライアンスで MAP-T を構成するには、次の作業を行います。

- 既定のマッピングルールを追加する
- 基本的なマッピングルールを追加する
- CE デバイスの IPv4 NAT アドレス範囲を基本マッピングルールにバインドする
- マップドメインを追加し、基本マッピングルールとデフォルトマッピングルールをドメインにバインドする

CLI を使用してデフォルトのマッピングルールを追加するには

コマンドプロンプトで入力します。

```
add MapDmr <name> -BRIpv6Prefix ( <ipv6_addr> | <*> )

show MapDmr <name>
<!--NeedCopy-->
```

CLI を使用して基本的なマッピングルールを追加するには

コマンドプロンプトで入力します。


```
add MapBmr <name> -RuleIpv6Prefix <ipv6_addr> | <*>  [-psidoffset <
    positive_integer>] [-EAbitLength <positive_integer>]  [-psidlength <
    positive_integer>]

show MapBmr <name>
<!--NeedCopy-->
```

CLI を使用して **CE** デバイスの **IPv4 NAT** アドレス範囲を基本マッピングルールにバインドするには

コマンドプロンプトで入力します。

```
bind MapBmr <name> (-network <ip_addr>  [-netmask <netmask>])

show MapBmr <name>
<!--NeedCopy-->
```

CLI を使用してマップドメインを追加するには

コマンドプロンプトで入力します。

```
add MapDomain <name> -MapDmrName <string>

show MapDomain <name>
<!--NeedCopy-->
```

CLI を使用して基本マッピングルールをマップドメインにバインドするには

コマンドプロンプトで入力します。

```
bind MapDomain <name> -MapBmrName <string>

show MapDomain <name>
<!--NeedCopy-->
```


構成例

```
add mapdmr DMR-1 -BRIpv6Prefix 2002:db8::/64

Done

add mapbmr BMR-1 -ruleIpv6Prefix 2002:db8:89ab::/48 -eAbitLength 16 -
    psidlength 8 -psidoffset 6

Done

bind mapbmr BMR-1 -network 192.0.1.0 -netmask 255.255.255.0

Done

add MapDomain MAP-DOMAIN-1 -mapdmrname DMR-1

Done

bind MapDomain MAP-DOMAIN-1 -mapbmrname BMR-1
Done
<!--NeedCopy-->
```

電話会社の加入者管理

October 7, 2021

電話会社ネットワークの加入者数は前例のない速度で増加しており、サービスプロバイダーにとって、加入者の管理は課題となりつつあります。より新しい、より速く、よりスマートなデバイスは、ネットワークおよび加入者管理システムに高い需要を投入しています。各加入者に同じサービス標準を提供することはもはや不可能であり、加入者ごとにトラフィック処理が必要不可欠です。

Citrix ADC アプライアンスは、ポリシーおよび課金ルール機能（PCRF）に保存されている情報に基づいて、加入者をプロファイルするためのインテリジェンスを提供します。モバイルサブスクライバがインターネットに接続すると、パケット Gateway は IP アドレスをサブスクライバに関連付けて、データパケットをアプライアンスに転送します。アプライアンスは、加入者情報を動的に受信するか、静的加入者を設定できます。この情報により、アプライアンスは、コンテンツスイッチング、統合キャッシュ、書き換え、レスポンダなどの豊富なトラフィック管理機能をサブスクライバごとに適用して、トラフィックを管理できます。

サブスクライバを管理するように Citrix ADC アプライアンスを設定する前に、サブスクライバセッションを格納するモジュールにメモリを割り当てる必要があります。ダイナミックサブスクライバの場合は、アプライアンスがセッ

ション情報を受信するインターフェイスを設定する必要があります。スタティックサブスクライバには ID を割り当てる必要があり、それらをポリシーに関連付けることができます。

また、次の操作も実行できます。

- 加入者ポリシーの適用と管理。
- 完全な IPv6 アドレスではなく IPv6 プレフィクスのみを使用して、サブスクライバを一意に識別するようにアプライアンスを設定します。
- ポリシーを使用して、ダイナミックサブスクライバとスタティックサブスクライバの両方の TCP トラフィックを最適化します。これらのポリシーは、異なる TCP プロファイルを異なるタイプのユーザに関連付けます。
- Citrix ADC アプライアンスでアイドル状態のセッションを管理します。
- ログサーバへのロギングを有効にします。
- 削除されたサブスクライバセッションの LSN セッションを削除します。

サブスクライバセッションストアモジュールへのメモリの割り当て

各サブスクライバセッションエントリは、1 KB のメモリを消費します。任意の時点で 500,000 の加入者セッションを保存するには、500 MB のメモリが必要です。この値は、「show extendedmemoryparam」コマンドの出力の一部として表示される最小メモリ要件に追加する必要があります。次の例では、3 つのパケットエンジンと 8 GB メモリを搭載した Citrix ADC VPX インスタンスの出力です。

このアプライアンスで 500,000 加入者セッションを保存するには、設定されたメモリが 2058+500 MB（500,000 x 1 KB = 500 MB）である必要があります。

> 注
>
> 構成されたメモリは、2 MB の倍数でなければならず、最大メモリ使用制限を超えないようにしてください。変更を有効にするには、アプライアンスを再起動する必要があります。

例

```
show extendedmemoryparam
        Extended Memory Global Configuration. This memory is utilized by
            LSN and Subscriber Session Store Modules:
         Active Memory Usage: 0 MBytes
         Configured Memory Limit: 0 MBytes
         Minimum Memory Required: 2058 MBytes
         Maximum Memory Usage Limit: 2606 MBytes
     Done
     set extendedmemoryparam -memLimit 2558
     Done
     show extendedmemoryparam
             Extended Memory Global Configuration. This memory is
                utilized by LSN and Subscriber Session Store Modules:
```



```
12
13              Active Memory Usage: 2558 MBytes
14              Configured Memory Limit: 2558 MBytes
15              Minimum Memory Required: 2058 MBytes
16              Maximum Memory Usage Limit: 2606 MBytes
17      Done
18  <!--NeedCopy-->
```

ダイナミックサブスクライバ用のインターフェイスの設定

Citrix ADC アプライアンスは、次のいずれかのタイプのインターフェイスを介して加入者情報を動的に受信します。

- Gx インターフェイス
- RADIUS インターフェイス
- RADIUS および Gx インターフェイス

> 注
>
> - NetScaler リリース 12.0 ビルド 57.19 以降、クラスタ展開では Gx インターフェイスがサポートされます。詳細については、クラスタトポロジの Gx インターフェイスを参照してください。
> - HA セットアップでは、サブスクライバセッションはセカンダリノードで継続的に同期されます。フェールオーバーが発生しても、サブスクライバ情報はセカンダリノードで引き続き使用できます。

Gx インターフェイス

Gx インターフェイス（3GPP 29.212 で指定）は、Diameter プロトコルに基づく標準インターフェイスで、PCRF と Telco ネットワークの Policy and Charging Enforcement Function（PCEF; ポリシー制御および課金規則）の交換を可能にします。

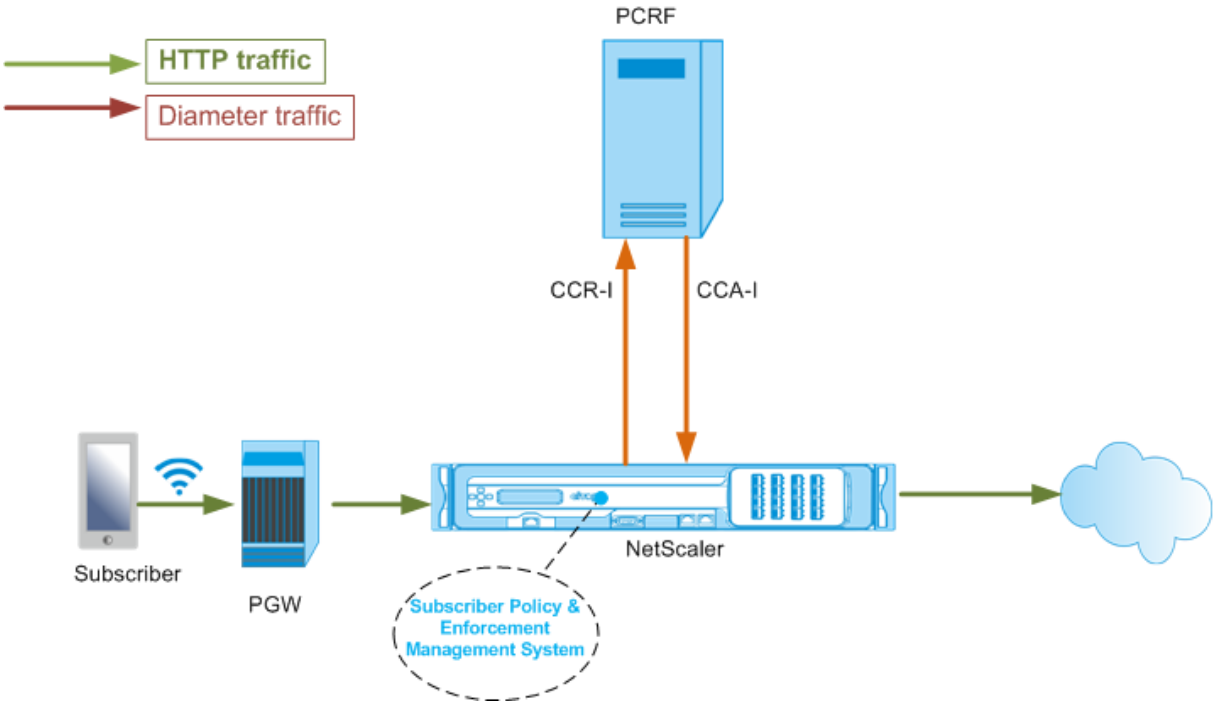
IP-CAN セッションが確立されると、パケット Gateway は MSISDN などのサブスクライバ ID や、サブスクライバに関するフレーム IP アドレス情報を Diameter メッセージとして PCRF に転送します。データパケットがパケット Gateway（PGW）からアプライアンスに到着すると、アプライアンスはサブスクライバ IP アドレスを使用して PCRF を照会し、サブスクライバ情報を取得します。これは、セカンダリ PCEF 機能とも呼ばれます。

Gx インターフェイス経由でアプライアンスが受信した PCC ルールは、サブスクライバセッション中にアプライアンスに保存されます。つまり、PCRF が Re-Auth-Request（RAR; セッション解放原因）メッセージとともにセッション解放要求（RAR）メッセージを送信するか、サブスクライバセッションが CLI または設定ユーティリティーを使用します。既存の加入者にアップデートがある場合、PCRF は RAR メッセージでアップデートを送信します。加入者セッションは、加入者がネットワークにログインしたときに開始され、加入者がログオフすると終了します。

> 注：PCRF サーバーがダウンしている場合、Citrix ADC アプライアンスは保留中または着信 Gx サブスクライバー要求に対してネガティブセッションを作成します。PCRF サーバーが再びバックアップされると、Citrix ADC アプライアンスは、特定のサブスクライバー要求を実行する前にネガティブセッションの有効期限が切れ

> るのを待つことによって、要求のストームを防止します。

次の図は、高レベルのトラフィックフローを示しています。データプレーントラフィックは HTTP であると想定しています。アプライアンスは、Gx インターフェイス経由でクレジット制御要求（CCR）を PCRF サーバに送信し、クレジット制御応答（CCA）で PCC ルールと、オプションで特定の加入者に適用されるその他の情報（無線アクセス技術（RAT）タイプなど）を受信します。PCC 規則には、1 つ以上のポリシー（規則）名およびその他のパラメータが含まれます。アプライアンスはこの情報を使用して、アプライアンスに保存されている定義済みのルールを取得し、トラフィックのフローを誘導します。また、この情報は、加入者セッション中に、加入者ポリシーおよび実施管理システムに保存されます。サブスクライバセッションが終了すると、アプライアンスはサブスクライバに関するすべての情報を破棄します。



次に、Gx インターフェイスを設定するコマンドの例を示します。コマンドは太字で表示されます。

Gx インターフェイスを設定するには、次の作業を実行します

Gx インターフェイスごとに DIAMETER サービスを追加します。次に例を示します:

```
1 add service pcrf-svc1 203.0.113.1 DIAMETER 3868
2
3 add service pcrf-svc2 203.0.113.2 DIAMETER 3868
4 <!--NeedCopy-->
```

アドレス指定できない DIAMETER 負荷分散仮想サーバーを追加し、手順 1 で作成したサービスをこの仮想サーバーにバインドします。複数のサービスでは、特定のセッションが同じ PCRF サーバによって処理されるように、persistenceType と persistAVPno を指定します。次に例を示します:


```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

Citrix ADC 直径のアイデンティティとレルムを設定します。アイデンティティとレルムは、Gx クライアントによって送信される直径メッセージでオリジンホストおよびオリジンレルム AVP として使用されます。次に例を示します:

```
set ns diameter  - identity netscaler.com  - realm com
<!--NeedCopy-->
```

手順 2 で作成した仮想サーバーを PCRF 仮想サーバーとして使用するように Gx インターフェイスを設定します。Gx クライアントから送信される直径メッセージで、宛先レルム AVP として使用する PCRF レルムを指定します。次に例を示します:

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com
<!--NeedCopy-->
```

サブスクライバインターフェイスタイプを GxOnly に設定します。次に例を示します:

```
set subscriber param -interfaceType GxOnly
<!--NeedCopy-->
```

Gx インターフェイスの設定とステータスを表示するには、次のように入力します。

```
show subscriber gxinterface
<!--NeedCopy-->
```

例

```
show subscriber gxinterface
```



```
    Gx Interface parameters:
        PCRF Vserver: vdiam (DOWN)
        Gx Client Identity...:  netscaler1.com
        Gx Client Realm ..........:  com
        PCRF Realm: epc.mnc030.mcc234.3gppnetwork.org
        Hold Packets On Subscriber Absence: YES
        CCR Request Timeout: 4 Seconds
        CCR Request Retry Attempts: 1
        Gx HealthCheck enabled: NO
        Gx HealthCheck TTL : 30 Seconds
        CER Request Timeout: 10 Seconds
        RevalidationTimeout: 30 Seconds
        NegativeTTL: 60 Seconds
        NegativeTTL Limited Success: NO
        Purge SDB on Gx Failure: YES
        ServicePath AVP code: 262099    ServicePath AVP VendorID: 3845
        PCRF Connection State: PCRF is not ready
    Done

<!--NeedCopy-->
```

ARGUMENTS

仮想サーバー

Gx 接続が確立されるロードバランシングまたはコンテンツスイッチング仮想サーバーの名前。仮想サーバーのサービスタイプは、直径または SSL_DIAMETER である必要があります。このパラメータは、service パラメータと相互に排他的です。したがって、Gx インターフェイスでサービスと仮想サーバーの両方を設定することはできません。

サービス

Gx 接続が確立されている PCRF に対応する直径または SSL_DIAMETER サービスの名前。このパラメータは、vserver パラメータと相互に排他的です。したがって、Gx インターフェイスでサービスと仮想サーバーの両方を設定することはできません。

pcrf レルム

メッセージのルーティング先となる PCRF の領域。これは、（直径ノードとして）Citrix ADC Gx クライアントによって宛先レルム AVP で使用されるレルムです。

加入者不在に保留します

サブスクライバセッション情報が PCRF サーバからフェッチされるまでパケットを保持するには、Yes に設定します。No に設定すると、PCRF サーバからサブスクライバセッション情報がフェッチされるまで、デフォルトのサブスクライバプロファイルが適用されます。デフォルトのサブスクライバプロファイルが設定されていない場合、サブスクライバ属性を使用する式に対して UNDEF が生成されます。

要求タイムアウト

Gx CCR 要求が完了するまでの時間（秒単位）。この時間内に要求が完了しない場合、要求は requestRetryAttempts パラメータで指定された回数だけ再送信されます。再送信後も要求が完了しない場合、デフォルトのサブスクライバプロファイルがこのサブスクライバに適用されます。デフォルトのサブスクライバプロファイルが設定されていない場合、サブスクライバ属性を使用する式に対して UNDEF が生成されます。ゼロを指定すると、タイムアウトが無効になります。デフォルト値:10

要求再試行回数

requestTimeout パラメータで指定された値内にリクエストが完了しなかった場合に、リクエストを再送信する必要がある回数を指定します。デフォルト値は 3 です。

ヘルスチェック

Gx ピアのインラインヘルスチェックを有効にするには、[Yes] に設定します。有効にすると、Citrix ADC は DWR パケットを PCRF サーバーに送信します。Gx セッションがアイドル状態になると、healthCheck タイマーが期限切れになり、PCRF サーバがアクティブかどうかをチェックするために DWR パケットが開始されます。デフォルト値: いいえ。
注：このパラメーターは、Citrix ADC 12.1 ビルド 51.xx 以降でサポートされています。

healthCheckTTL

ウォッチドッグ監視用に定義された時間（秒単位）。ヘルスチェック TTL 時間が経過すると、DWR が送信され、PCRF サーバのステータスがチェックされます。CCR、CCA、RAR、または RAA メッセージがあれば、タイマーがリセットされます。
最小値：6 秒。デフォルト値：30 秒。
注：このパラメーターは、Citrix ADC 12.1 ビルド 51.xx 以降でサポートされています。

cerRequestTimeout

機能交換要求の再送信に定義された時間（秒単位）。Citrix ADC は、この構成時間内に PCRF から CEA を受信しない場合、新しい CER メッセージを開始します。
PCRF サーバから応答を受信しない場合、アプライアンスは CER メッセージを 5 回送信しようとします。5 つの

CER メッセージの後でも応答がない場合、アプライアンスは TCP 接続を閉じ、障害を報告します。タイムアウト値が 0 に設定されている場合、アプリケーションヘルスチェック機能は無効になります。
最小値：0 秒。デフォルト値：0 秒。
注：このパラメーターは、Citrix ADC 12.1 ビルド 51.xx 以降でサポートされています。

再検証タイムアウト

セッションでの PCRF アクティビティの後、Gx CCR-U 要求が送信されるまでの時間（秒単位）。RAR メッセージまたは CCA メッセージによって、タイマーがリセットされます。ゼロ値は、アイドルタイムアウトを無効にします。

negativeTTL

サーバがダウンしているか、応答がないか、失敗した応答を受信したために PCRF によって解決されなかったセッションに対して Gx CCR-I 要求が再送信されるまでの時間（秒単位）。PCRF サーバーを常にポーリングする代わりに、負の TTL を使用すると、アプライアンスは未解決のセッションを保持します。ネガティブセッションの場合、アプライアンスはデフォルトのサブスクライバプロファイル（設定されている場合）から属性を継承し、RADIUS アカウンティングメッセージ（受信されている場合）から属性を継承します。ゼロの値は、負のセッションを無効にします。サブスクライバセッションを取得できなかった場合でも、アプライアンスはネガティブセッションをインストールしません。デフォルト値:600

negativeTTLLimitedSuccess

部分的な成功応答コード (2002) のネガティブセッションを作成するには、[Yes] に設定します。「いいえ」に設定すると、通常のセッションが作成されます。デフォルト値：いいえ。

このパラメーターは、Citrix ADC 12.1 ビルド 49.xx 以降でサポートされています。

purgeSDBonGxFailure

Gx インターフェイスに障害が発生したときにサブスクライバデータベースをフラッシュするには、[Yes] に設定します。Gx インターフェイスの障害には、DWR モニタリング（有効の場合）とネットワークヘルスチェック（有効の場合）の両方が含まれます。[Yes] に設定すると、すべての加入者セッションがクリアされます。
デフォルト値：いいえ。
注：このパラメーターは、Citrix ADC 12.1 ビルド 51.xx 以降でサポートされています。

servicePathAVP

PCRF が加入者に適用可能なサービスパスを送信する AVP コード。

servicePathVendorid

PCRF が加入者に適用可能なサービスパスを送信する AVP のベンダー ID。

GUI を使用して **Gx** インターフェイスを設定するには

1. **Traffic Management** > **Subscriber** > **Parameters** に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3. [インタフェースの種類] で、[**GxOnly**] を選択します。
4. 必要なすべてのパラメータの値を指定します。
5. [**OK**] をクリックします。

確立された **Gx** 接続でのトランスポート障害の検出

注：この機能は、Citrix ADC 12.1 ビルド 51.xx 以降でサポートされています。

Citrix ADC アプライアンスは、デバイスウォッチドッグ要求（DWR）およびデバイスウォッチドッグ応答（DWA）メッセージを使用して、確立された Gx 接続を介したトランスポート障害を検出するように構成できます。

Gx セッションが確立されると、事前に定義されたタイマーがトリガーされ、セッションがアイドル状態かどうかを検出します。アイドルタイムタイマーが期限切れになると、DWR メッセージが送信されます。アイドルタイムタイマーは、確立された Gx セッションを介して Citrix ADC アプライアンスがメッセージを受信するたびにリセットされます。DWR メッセージが送信された後、DWA メッセージに基づいてピアの可用性が確認されます。

- DWA を受信すると、ピアの可用性が確認され、ウォッチドッグタイマーがリセットされます。
- DWA が受信されず、ウォッチドッグタイマーが連続して 2 回期限切れになると、セッションはダウンし、ピアが利用できないと見なされます。アプライアンスはセッションを閉じ、Gx ピアとの新しいセッションを確立しようとします。

ウォッチドッグタイマーが応答なしで 2 回期限切れになると、Citrix ADC アプライアンスは Gx 接続に障害があると判断し、接続閉鎖を開始します。接続が閉じられると、他のウォッチドッグ要求は Gx ピアに送信されません。Citrix ADC アプライアンスは、PCRF 要求に対して次に使用可能な Gx セッションを使用します。

CLI を使用して、確立された **Gx** 接続を介したトランスポート障害を検出するには

コマンドプロンプトで入力します。

```
set subscriber gxInterface [-vServer <string>] [-service <string>] [-
    healthCheck ( YES | NO )] [-healthCheckTTL<positive_integer>][-
    cerRequestTimeout <positive_integer>] [-purgeSDBonGxFailure ( YES |
    NO )]
<!--NeedCopy-->
```

例**:**

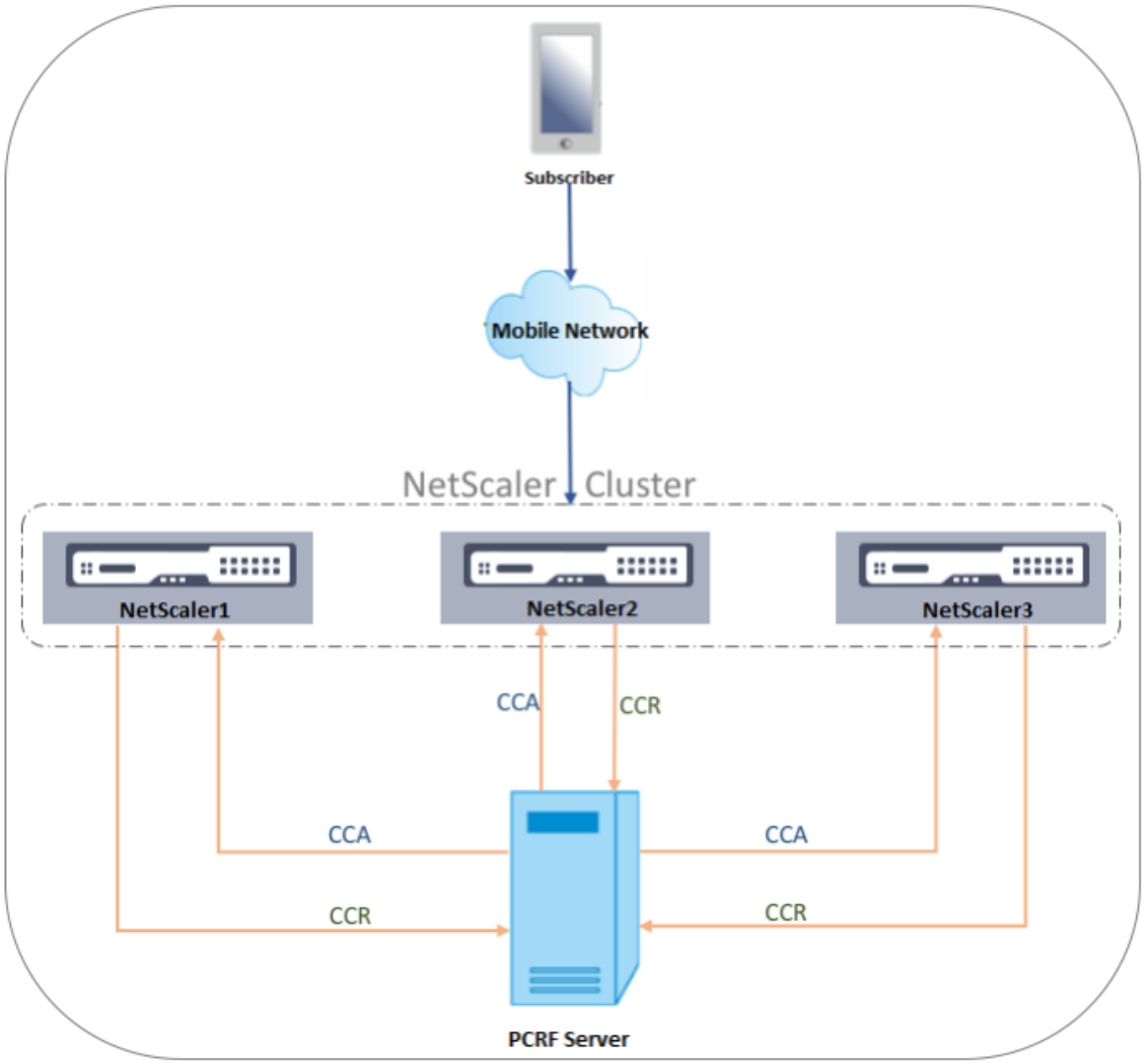

```
set subscriber gxInterface set subscriber gxInterface -vServer vdiam -
    healthCheck YES -healthCheckTTL 31 -cerRequestTimeout 15
    purgeSDBonGxFailure  YES
<!--NeedCopy-->
```

GUI を使用して確立された **Gx** 接続を介したトランスポート障害を検出するには

1. **Traffic Management** > **Subscriber** > **Parameters** に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3. [インターフェイスタイプ] で、[**GxOnly**] を選択します。
4. すべての必須パラメータの値を指定します。
5. [ヘルスチェック] を選択し、[ヘルスチェック **TTL**] および [**CER** 要求タイムアウト] の値を指定します。
6. [**OK**] をクリックします。

クラスタトポロジの **Gx** インターフェイス

Citrix ADC アプライアンスは、クラスタトポロジで Gx インターフェイスをサポートします。



クラスタ内の Citrix ADC ノードは、Gx インターフェイスを介して外部 PCRF サーバーと通信します。ノードがクライアントトラフィックを受信すると、アプライアンスは次の処理を実行します。

- CCR-I 要求を PCRF サーバに送信し、サブスクライバ情報を取得します。
- PCRF サーバは、CCR-A で応答します。
- 次に、Citrix ADC ノードは、受信したサブスクライバ情報をサブスクライバストアに保存し、ルールをクライアントトラフィックに適用します。

各ノードは独立したサブスクライバストアを維持し、サブスクライバセッションは他のノードと同期されません。

Diameter Base Protocol RFC 6733 に従って、各ピアは、直径プロトコルを介して他のピアと通信するために、一意の直径 ID を使用して設定する必要があります。したがって、クラスタ展開では、直径 ID の構成が特定されます。各ノードの直径パラメータ（ID、レルム、サーバクローズ伝播）は、GUI または CLI を使用して個別に設定できます。

ノードがクラスタに追加されると、デフォルトの直径パラメータ（アイデンティティ =netscaler.com、レルム =com、サーバクローズプロパゲーション =NO）が想定されます。ノードを追加したら、各ノードの直径パラメータ

を設定する必要があります。

GUI を使用して直径パラメータを設定するには

1. [システム] > [設定] に移動します。
2. 詳細ウィンドウで、[直径パラメータを変更] をクリックします。
3. [直径パラメータ] ページで、直径パラメータを設定する Citrix ADC ノードを選択し、[構成] をクリックします。
4.「直径パラメータの構成」ページで、選択したノードの直径のアイデンティティ、直径レルム、およびサーバークローズ伝播を構成します。
5. [**OK**] をクリックします。

CLI を使用して直径パラメータを設定するには

コマンドプロンプトで入力します。

```
set ns diameter [-identity <string>] [-ownerNode <positive_integer>]
<!--NeedCopy-->
```

ARGUMENTS

ID

直径識別は、直径ノードを一意に識別するために使用されます。直径の構成を設定する前に、Citrix ADC アプライアンス（直径ノードとして）に一意の直径の識別情報を割り当てる必要があります。

たとえば、ns の直径-アイデンティティーネットスケーラ.com-所有者ノード 1 を設定します。したがって、Citrix ADC システムが直径メッセージにアイデンティティを使用する必要があるときはいつでも、RFC3588 で定義されているように、オリジンホスト AVP として「netscaler.com」を使用します。

最大長：255

OwnerNode

OwnerNode は，直径 ID が設定されているクラスタノードの ID を表します。OwnerNode は、CLIP を介してのみ設定できます。

最小値:0

最大値:31

例**:**

set ns diameter -identity netscaler1.com -ownerNode 1

注:

オーナーノードオプションは、show ns diameter コマンドにも追加されます。

例:

```
1 show diameter -ownerNode <0-31>
2 <!--NeedCopy-->
```

show ns diameter コマンドを実行すると、指定したノードの直径パラメータが表示されます。

クラスター展開用に **Gx** インターフェイスを構成するには

Gx インターフェイスを設定するには、次の作業を行います。

Gx インターフェイスごとに DIAMETER サービスを追加します。

例:

```
1 add service pcrf-svc1 203.0.113.1 DIAMETER 3868
2 add service pcrf-svc2 203.0.113.2 DIAMETER 3868
3 <!--NeedCopy-->
```

DIAMETER 負荷分散仮想サーバーを追加し、手順 1 で作成したサービスをこの仮想サーバーにバインドします。

例:

```
1 add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
     persistAVPno 263
2
3 bind lb vserver vdiam pcrf-svc1
4
5 bind lb vserver vdiam pcrf-svc2
6 <!--NeedCopy-->
```

すべてのクラスタノードで Citrix ADC 直径のアイデンティティとレルムを構成します。アイデンティティとレルムは、Gx クライアントによって送信される直径メッセージでオリジンホストおよびオリジンレルム AVP として使用されます。

例:


```
set ns diameter -identity node0.netscaler.com -realm netscaler.com -
    ownerNode 0

set ns diameter -identity node1.netscaler.com -realm netscaler.com -
    ownerNode 1
<!--NeedCopy-->
```

ステップ 2 で作成した仮想サーバーを PCRF 仮想サーバーとして使用するように Gx インターフェイスを設定し、PCRF レルムも設定します。

例:

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com

Set the subscriber interface type to GxOnly.
<!--NeedCopy-->
```

例:

```
set subscriber param -interfaceType GxOnly
<!--NeedCopy-->
```

Gx インターフェイスの設定とステータスを表示するには、次のように入力します。

```
show subscriber gxinterface
<!--NeedCopy-->
```

RADIUS インターフェイス

RADIUS インターフェイスでは、パケット Gateway は、IP-CAN セッションが確立されると、RADIUS アカウンティング開始メッセージ内のサブスクライバ情報を RADIUS インターフェイス経由でアプライアンスに転送します。RADIUSListener タイプのサービスは、RADIUS アカウンティングメッセージを処理します。RADIUS クライアントの共有シークレットを追加します。共有シークレットが設定されていない場合、RADIUS メッセージはサイレントにドロップされます。次に、RADIUS インターフェイスを設定するコマンドの例を示します。コマンドは太字で表示されます。

RADIUS インターフェイスを設定するには、次の作業を行います。

RADIUS メッセージを受信する SNIP アドレスに RADIUS リスナーサービスを作成します。次に例を示します:


```
add service srad1 192.0.0.206 RADIUSLISTENER 1813
<!--NeedCopy-->
```

このサービスを使用するように加入者の RADIUS インターフェイスを設定します。次に例を示します:

```
set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

サブスクライバインターフェイスタイプを RadiusOnly に設定します。次に例を示します:

```
set subscriber param -interfaceType RadiusOnly
<!--NeedCopy-->
```

サブネットと共有シークレットを指定する RADIUS クライアントを追加します。次に例を示します:

```
add radius client 192.0.2.0/24 -radkey client123
<!--NeedCopy-->
```

0.0.0.0/0 のサブネットは、すべてのクライアントのデフォルトの共有秘密であることを意味します。RADIUS インターフェイスの構成とステータスを表示するには、次のように入力します。

```
show subscriber radiusInterface
<!--NeedCopy-->
```

RADIUS インターフェイスパラメータ:

Radius リスナーサービス:srad1 (UP)

Done

例**:**

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```


ARGUMENTS

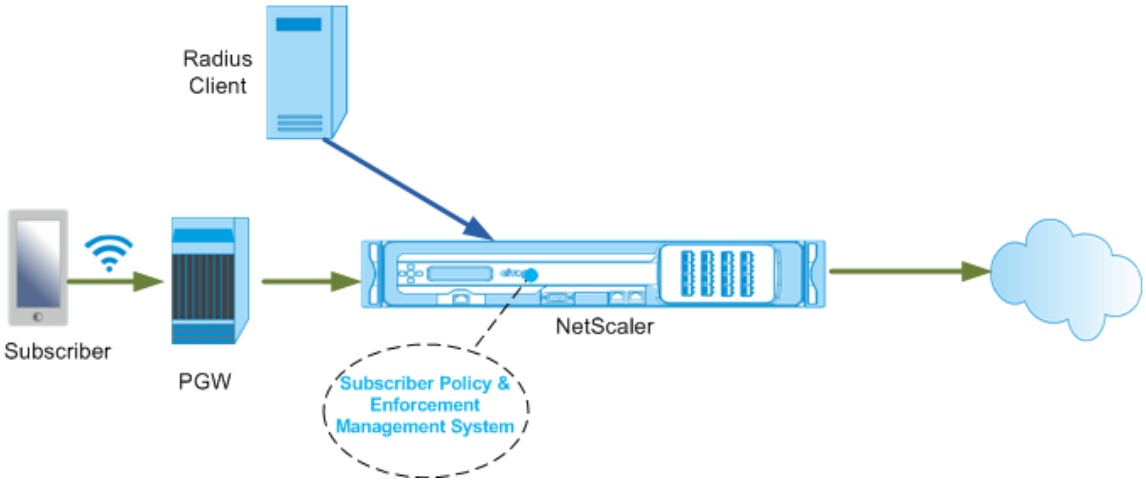
ListeningService

RADIUS アカウンティング要求を処理する RADIUS リスニングサービスの名前。

svrState

RADIUS リスニングサービスの状態。

次の図は、高レベルのトラフィックフローを示しています。



GUI を使用して **RadiusOnly** インターフェイスを設定するには

1. **Traffic Management** > **Subscriber** > **Parameters** に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3. [インタフェースタイプ] で、[**RadiusOnly**] を選択します。
4. 必要なすべてのパラメータの値を指定します。
5. [**OK**] をクリックします。

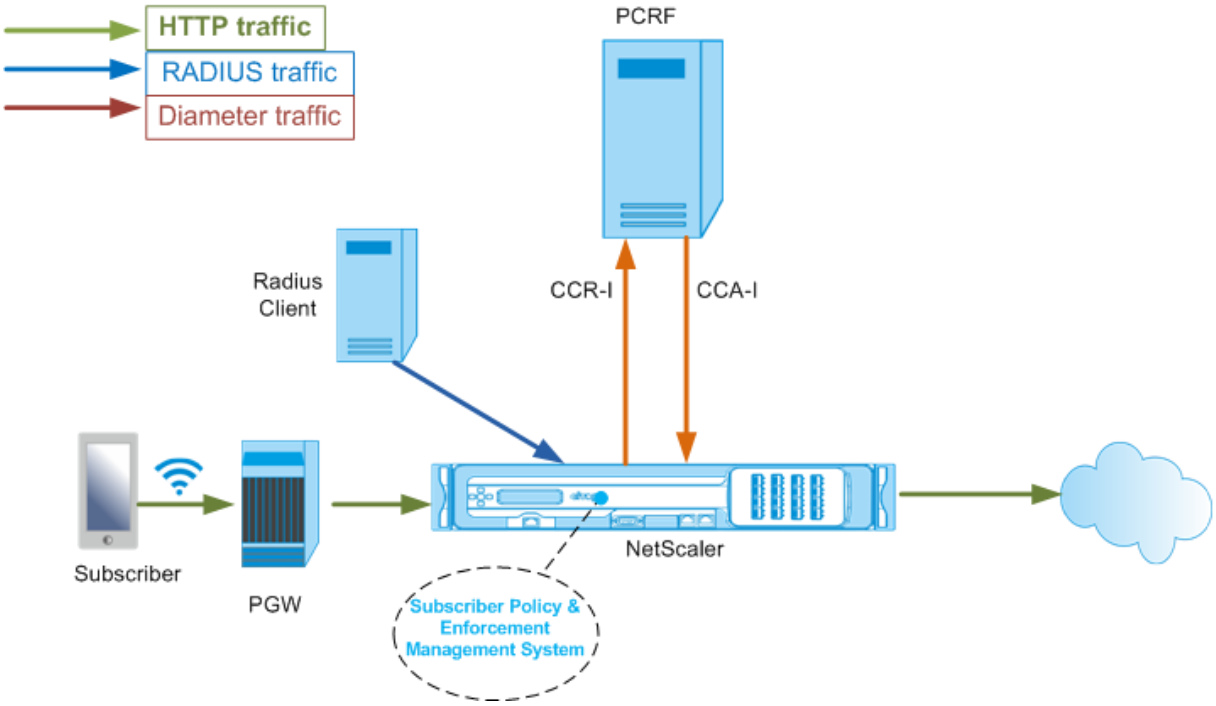
RADIUS および **Gx** インターフェイス

RADIUS および Gx インターフェイスでは、IP-CAN セッションが確立されると、パケット Gateway はサブスクライバ ID（MSISDN など）、サブスクライバに関するフレーム IP アドレス情報を RADIUS インターフェイス経由でアプライアンスに転送します。アプライアンスはこのサブスクライバ ID を使用して、Gx インターフェイス上の PCRF

を照会し、サブスクライバ情報を取得します。これは、プライマリ PCEF 機能と呼ばれます。次に、RADIUS および Gx インターフェイスを設定するコマンドの例を示します。

```
set subscriber param -interfaceType RadiusandGx
add service pcrf-svc 203.0.113.1 DIAMETER 3868
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263
bind lb vserver vdiam pcrf-svc
set subscriber gxInterface -vServer vdiam -pcrfRealm testrealm1.net -
    holdOnSubscriberAbsence YES -revalidationTimeout 60 -negativeTTL 120
add service srad1 192.0.0.206 RADIUSLISTENER 1813    set subscriber
    radiusInterface -listeningService srad1
<!--NeedCopy-->
```

次の図は、高レベルのトラフィックフローを示しています。



GUI を使用して **RadiusAndGx** インターフェイスを設定するには

1. **Traffic Management** > **Subscriber** > **Parameters** に移動します。
2. [サブスクライバーパラメータの構成] をクリックします。
3. 「インタフェース・タイプ」で、「**RadiusAndGx**」を選択します。
4. 必要なすべてのパラメータの値を指定します。
5. [**OK**] をクリックします。

スタティックサブスクライバの設定

コマンドラインまたは構成ユーティリティを使用して、Citrix ADC アプライアンスでサブスクライバを手動で構成できます。スタティックサブスクライバを作成するには、一意のサブスクライバ ID を割り当てて、オプションで各サブスクライバにポリシーを関連付けます。次に、スタティックサブスクライバを設定するコマンドの例を示します。

次の例では、**subscriptionIdvalue** は国際電話番号を指定し、**subscriptionIdType** (この例では E164) は国際電話番号の一般的な形式を指定します。

```
    add subscriber profile 203.0.113.6 -subscriberRules policy1 policy2
        -subscriptionIdType E164 -subscriptionIdvalue 98767543211
    add subscriber profile 2002::a66:e8d3/64 -subscriberRules policy1
       policy3 -subscriptionIdtype E164 -subscriptionIdvalue
       98767543212
    add subscriber profile 203.0.24.2 10 -subscriberRules policy2
       policy3 -subscriptionIdtype E164 -subscriptionIdvalue
       98767543213
<!--NeedCopy-->
```

設定された加入者プロファイルを表示するには、次のように入力します。

> ユーザプロファイルの表示

```
    > show subscriber profile

    1)  Subscriber IP: 203.0.24.2 VLAN:10
        Profile Attributes:
                Active Rules: policy2, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543213
    2)  Subscriber IP: 2002::/64
        Profile Attributes:
                Active Rules: policy1, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543212
    3)  Subscriber IP: 203.0.113.6
        Profile Attributes:
                Active Rules: policy1, policy2
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543211

     Done
<!--NeedCopy-->
```


デフォルトの加入者プロファイル

アプライアンスのサブスクライバセッションストアでサブスクライバ IP アドレスが見つからない場合は、デフォルトのサブスクライバプロファイルが使用されます。次の例では、デフォルトの加入者プロファイルが加入者ルール policy1 とともに追加されています。

```
    > add subscriber profile * -subscriberRules policy1
<!--NeedCopy-->
```

加入者セッションの表示とクリア

スタティックおよびダイナミックサブスクライバセッションをすべて表示するには、次のコマンドを使用します。

> show subscriber sessions

```
    > show subscriber sessions
    1)  Subscriber IP: 2002::/64
        Session Attributes:
                Active Rules: policy1, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543212
    2)  Subscriber IP: *
        Session Attributes:
                Active Rules: policy1
    3)  Subscriber IP: 203.0.24.2 VLAN:10
        Session Attributes:
                Active Rules: policy2, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543213
    4)  Subscriber IP: 203.0.113.6
        Session Attributes:
                Active Rules: policy1, policy2
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543211
    5)  Subscriber IP: 192.168.0.11
        Session Attributes:
                Idle TTL remaining: 361 Seconds
                Active Rules: policy1
                Subscriber Id Type: E164
                Subscriber Id Value: 1234567811
                Service Path: policy1
```



```
27                  AVP(44): 34 44 32 42 42 38 41 43  2D 30 30 30 30 30 30
                        31 31
28                  AVP(257): 00 01 C0 A8 0A 02
29                  PCRF-Host: host.pcrf.com
30                  AVP(280): 74 65 73 74 2E 63 6F 6D
31
32      Done
33  <!--NeedCopy-->
```

単一のセッションまたは完全なセッションストアをクリアするには、次のコマンドを使用します。IP アドレスを指定しない場合、完全な加入者セッションストアがクリアされます。

```
1  clear subscriber sessions <ip>
2  <!--NeedCopy-->
```

加入者ポリシーの適用および管理システム

Citrix ADC アプライアンスは、加入者ポリシーの適用および管理システムのキーとして加入者の IP アドレスを使用します。

加入者式を追加して、加入者ポリシー実施および管理システムで使用可能な加入者情報を読み取ることができます。これらの式は、統合キャッシュ、書き換え、レスポンダー、コンテンツスイッチングなど、Citrix ADC 機能用に構成されたポリシー規則およびアクションで使用できます。

次のコマンドは、加入者ベースの応答者のアクションとポリシーを追加する例です。加入者規則値が「pol1」の場合、ポリシーは true と評価されます。

```
1       add responder action error_msg respondwith '"HTTP/1.1 403 OKrnrn" +
            " You are  not authorized to access Internet"'
2       add responder policy no_internet_access "SUBSCRIBER.RULE_ACTIVE("
            pol1")" error_msg
3   <!--NeedCopy-->
```

次に、加入者ベースの書き換えアクションとポリシーを追加するコマンドの例を示します。アクションは、サブスクライバセッションで AVP（45）の値を使用して HTTP ヘッダー「X-Nokia-MSISDN」を挿入します。

```
1       > add rewrite action AddHDR-act insert_http_header X-Nokia-MSISDN "
            SUBSCRIBER.AVP(45).VALUE"
2       > add rewrite policy AddHDR-pol "HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.
            URL).EQUALS_ANY("patset-test")" AddHDR-act
```



```
<!--NeedCopy-->
```

次の例では、アプライアンスに 2 つのポリシーが設定されています。アプライアンスが加入者情報をチェックし、加入者ルールが cache_enable の場合、キャッシングを実行します。サブスクライバルールが cache_disable の場合、アプライアンスはキャッシュを実行しません。

```
    > add cache policy nocachepol -rule "SUBSCRIBER.RULE_ACTIVE("
       cache_disable")" - action NOCACHE
    > add cache policy cachepol -rule "SUBSCRIBER.RULE_ACTIVE("
       cache_enable")" - action CACHE -storeInGroup cg1
<!--NeedCopy-->
```

「SUBSCRIBER」で始まる式の完全なリストについては、『ポリシー設定ガイド』を参照してください。

> 重要:
>
> Citrix ADC ソフトウェアリリース 12.1 では、加入者インターフェイスが GxOnly に設定されている場合、IPANDVLAN キー検索方式がサポートされています。詳細については、IP アドレスと VLAN ID キールックアップ方式を参照してください。

IPv6 プレフィクスベースのサブスクライバセッション

電話会社のユーザは、完全な IPv6 アドレスではなく IPv6 プレフィクスで識別されます。Citrix ADC アプライアンスは、完全な IPv6 アドレス（/128）の代わりにプレフィックスを使用して、データベース（サブスクライバストア）内のサブスクライバを識別するようになりました。PCRF サーバと通信するために（たとえば、CCR-I メッセージで）、アプライアンスは完全な IPv6 アドレスではなく、フレーム化された IPv6 プレフィクス AVP を使用するようになりました。デフォルトのプレフィクス長は /64 ですが、別の値を使用するようにアプライアンスを設定できます。

コマンドラインを使用して **IPv6** プレフィックスを構成するには

```
set subscriber param [-ipv6PrefixLookupList <positive_integer> ...]
```

以下の最初のコマンド例では、1 つのプレフィックスを設定し、2 番目のコマンド例では、複数のプレフィックスを設定します。

```
    set subscriber param -ipv6PrefixLookupList 64
    set subscriber param -ipv6PrefixLookupList 64 72 96
<!--NeedCopy-->
```


構成ユーティリティを使用して **IPv6** プレフィックスを構成するには

1. **Traffic Management** > **Subscriber** > **Parameters** に移動します。
2. 詳細ペインの [設定] で、[サブスクライバパラメータの設定] をクリックし、[**IPv6** プレフィクスルックアップリスト] で 1 つまたは複数のプレフィクスを指定します。

IP アドレスおよび **VLAN ID** キーの検索方法

Citrix ADC アプライアンスは、加入者ポリシーの適用および管理システムへのキー検索方法として、加入者の IP アドレスを使用します。この方法は、IP アドレスが重複している場合は有効ではありません。このような場合、VLAN ID を追加の加入者検索タイプとして使用できます。IPANDVLAN キールックアップ方式がサポートされるのは、サブスクライバインターフェイスが GxOnly に設定されている場合だけです。IPANDVLAN がルックアップ方式として構成されている場合、Citrix ADC アプライアンスは以下を実行します。

- IPv4 サブスクライバの Gx クエリに発信元 VLAN ID を含めます。
- すべての Gx 応答に Gx VLAN AVP を含めます。ただし、VLAN ID の不一致がある場合、アプライアンスは応答を無視します。

たとえば、アプライアンスが GxSessionId-a: IPv4-b: VLAN-c を含む CCR-I を送信し、応答に GxSessionId-a: IPv4-b: VLAN-d が含まれている場合、応答はドロップされ、デフォルトのサブスクライバエントリが作成されます。

> 注
>
> - インターフェイスタイプ RadiusAndGx および RadiusOnly は、キータイプ IPANDVLAN とともに設定できません。
> - IPv6 アドレスからのトラフィックの場合、Citrix ADC アプライアンスは IP ルックアップ方式を使用します。

CLI を使用して **IP** または **IPANDVLAN** をキールックアップ方式として設定するには

コマンドプロンプトで入力します。

```
set subscriber param [-keytype ( IP | IPANDVLAN )] [-interfaceType <
   interfaceType>]
<!--NeedCopy-->
```

例**:**

```
set subscriber param -keytype IPANDVLAN -interfaceType GxOnly

set subscriber param -keytype IP -interfaceType GxOnly
<!--NeedCopy-->
```


> 注
>
> キータイプパラメータを IP から IPANDVLAN に変更し、逆にすべての加入者データを消去します。

VLAN パラメータ

VLAN パラメータは、次のコマンドにも追加されます。

```
add subscriber profile <ip>@ [-vlan]

set subscriber profile <ip>@ [-vlan] [-subscriptionIdType  <
    subscriptionIdType>]

show subscriber profile [<ip>@] [-vlan]

rm subscriber profile <ip>@ [-vlan <positive_integer>]
<!--NeedCopy-->
```

引数

IP

加入者 IP アドレスを表します。これは必須の引数であり、サブスクライバプロファイルの追加後は変更できません。

vlan

加入者が配置されている VLAN 番号を表します。VLAN 番号は、加入者プロファイルの追加後に変更できません。

最小値:1

最大値:4096

```
add subscriber profile 192.0.2.23 10

set subscriber profile 192.0.2.23 10 -subscriptionIdtype E164

show subscriber profile 192.0.2.23 10

rm subscriber profile 192.0.2.23 10

<!--NeedCopy-->
```


GUI を使用して **IP** または **IPANDVLAN** をキールックアップ方式として設定するには

1. [トラフィック管理] > [加入者] > [パラメータ] に移動します。
2. [サブスクライバパラメータの設定] をクリックします。
3. [キーの種類] で、要件に応じて [**IP**] または [**IPANDVLAN**] を選択します。
4. 構成を完了し、[**OK**] をクリックします。

Telco ネットワークにおける加入者セッションのアイドルセッション管理

Citrix ADC アプライアンスでのサブスクライバセッションのクリーンアップは、RADIUS アカウンティング停止メッセージ、Diameter RAR（セッションリリース）メッセージ、または「サブスクライバセッションのクリア」コマンドなどのコントロールプレーンイベントに基づいています。展開によっては、RADIUS クライアントまたは PCRF サーバからのメッセージがアプライアンスに届かない場合があります。また、大量のトラフィックが発生すると、メッセージが失われる可能性があります。長時間アイドル状態のサブスクライバセッションは、Citrix ADC アプライアンス上のメモリと IP リソースを消費し続けます。アイドルセッション管理機能は、アイドルセッションを識別するための設定可能なタイマーを提供し、指定されたアクションに基づいてこれらのセッションをクリーンアップします。

この加入者からのトラフィックがデータプレーンまたはコントロールプレーンで受信されない場合、セッションはアイドルと見なされます。更新、終了（PCRF に通知してからセッションを削除）、削除（PCRF を通知せずに）アクションを指定できます。このアクションは、アイドルタイムアウトパラメータで指定された時間セッションがアイドル状態になった後にのみ実行されます。

コマンドラインを使用してアイドルセッションタイムアウトおよび関連付けられたアクションを構成するには

```
set subscriber param [-idleTTL <positive_integer>] [-idleAction <
   idleAction>]
<!--NeedCopy-->
```

例:

```
set subscriber param -idleTTL 3600 -idleAction ccrTerminate

set subscriber param -idleTTL 3600 -idleAction ccrUpdate

set subscriber param -idleTTL 3600 -idleAction delete
<!--NeedCopy-->
```

アイドルセッションのタイムアウトを無効にするには、アイドルタイムアウトをゼロに設定します。

set subscriber param –idleTTL 0

構成ユーティリティを使用してアイドルセッションタイムアウトおよび関連するアクションを構成するには

1. **Traffic Management** > **Subscriber** > **Parameters** に移動します。
2. 詳細ペインの [設定] で、[サブスクライバパラメータの構成] をクリックし、[アイドル時間] と [アイドルアクション] を指定します。

サブスクライバセッションイベントロギング

サブスクライバロギングを有効にすると、サブスクライバ固有の RADIUS および Gx コントロールプレーンメッセージを追跡し、履歴データを使用してサブスクライバアクティビティを分析できます。主な属性には、MSISDN とタイムスタンプがあります。次の属性もログに記録されます。

- Session Event (Install, Update, Delete, Error)
- Gx Message Type (CCR-I, CCR-U, CCR-T, RAR)
- Radius Message Type (Start, Stop)
- サブスクライバ IP
- サブスクライバ ID タイプ (MSISDN (E164)、IMSI)
- サブスクライバ ID 値

これらのログを使用すると、IP アドレスと MSISDN (利用可能な場合) でユーザーを追跡できます。

ローカルまたはリモートの Syslog または nslog サーバへのサブスクライバセッションロギングを有効にできます。次に、リモート syslog サーバへのサブスクライバロギングを有効にする例を示します。

```
1      > add syslogAction sysact1 192.0.2.0 -loglevel EMERGENCY ALERT
           CRITICAL ERROR WARNING NOTICE INFORMATIONAL -subscriberlog
           enabled
2  <!--NeedCopy-->
```

これらのログから、セッションが更新、削除、作成（インストール）された時間など、ユーザーに関連するあらゆるアクティビティについて知ることができます。さらに、エラーメッセージも記録されます。

例:

1. 次のログエントリは、RADIUSandGx セッションの作成、セッションの更新、およびセッションの削除の例です。

   > 09/30/2015:16:29:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 147 0 : Session Install, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001
   >
   > 09/30/2015:16:30:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 148 0 : Session Update, GX MsgType: CCR-U, IP: 100.10.1.1, ID: E164 - 30000000001
   >
   > 09/30/2015:17:27:56 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT

> 185 0 : Session Delete, GX MsgType: CCR-T, RADIUS MsgType: Stop, IP: 100.10.1.1, ID: E164 - 30000000001

2. 次のログエントリは、PCRF サーバでサブスクライバが見つからない場合や、アプライアンスが PCRF サーバに接続できない場合など、エラーメッセージの例です。

> 09/30/2015:16:44:15 GMT Error 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 169 0 : Failure Reason: PCRF failure response, GX MsgType: CCR-I, IP: 100.10.1.1
>
> Sep 30 13:03:01 09/30/2015:16:49:08 GMT 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 176 0 : Failure Reason: Unable to connect to PCRF, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001#000#000#000#000#000#000#000#000#000#000#000#000#

Subscriber aware LSN session termination

以前のリリースでは、RADIUS アカウンティング STOP または PCRF-RAR メッセージを受信したときに加入者セッションが削除された場合、または TTL の有効期限やフラッシュなどの他のイベントの結果として加入者セッションが削除された場合、加入者の対応する LSN セッションは、設定された LSN タイムアウト期間の後にのみ削除されます。このタイムアウトが切れるまで開いたままの LSN セッションは、アプライアンス上のリソースを消費し続けます。

リリース 11.1 から、新しいパラメータ（サブスクライブセッション削除）が追加されました。このパラメータが有効で、サブスクライバ情報がサブスクライバデータベースから削除されると、そのサブスクライバに対応する LSN セッションも削除されます。このパラメータが無効の場合、加入者セッションは LSN タイムアウト設定で指定されたとおりにタイムアウトします。

CLI を使用して加入者認識 **LSN** セッション終了を設定するには

コマンドプロンプトで入力します。

```
set lsn parameter -subscrSessionRemoval (          DISABLED )
ENABLED
```

```
1     > set lsn parameter -subscrSessionRemoval ENABLED
2     Done
3     > sh lsn parameter
4                 LSN Global Configuration:
5
6                 Active Memory Usage: 0 MBytes
7                 Configured Memory Limit: 0 MBytes
8                 Maximum Memory Usage Limit: 912 MBytes
9                 Session synchronization: ENABLED
10                Subscriber aware session removal: ENABLED
```



```
<!--NeedCopy-->
```

GUI を使用して加入者認識 **LSN** セッション終了を設定するには

1. [システム] > [大規模 **NAT**] に移動します。
2. [はじめに] で、[**LSN** パラメータの設定] をクリックします。
3. サブスクライバ認識セッションの削除パラメータを設定します。

トラブルシューティング

展開が期待どおりに動作しない場合は、次のコマンドを使用してトラブルシューティングを行います。

- show subscriber gxinterface
  このコマンドの出力には、次のエラーメッセージ（ここに示される応答を示します）を含めることができます。
  - Gx インターフェイスが設定されていない-set subscriber param コマンドを使用して、正しいインターフェイスタイプを設定します。
  - PCRF が設定されていない-GxInterface 上で直径 vServer またはサービスを設定する-set subscriber gx インターフェイスコマンドを使用して、このインターフェイスに Diameter 仮想サーバまたはサービスを割り当てます。
  - PCRF の準備ができていません。詳細については、対応する仮想サーバ/サービスを確認してください。サービスの状態を確認するには、show LB vserver または show service コマンドを使用します。
  - Citrix ADC は、PCRF と Citrix ADC 間の PCRF 機能ネゴシエーションからの CEA を待機しています。これは断続的な状態である可能性があります。問題が解決しない場合は、PCRF サーバーの直径設定を確認してください。
  - サブスクライバセッションを保存するようにメモリが設定されていません。拡張メモリを構成するには、「拡張メモリを設定」を使用してください。拡張メモリを構成するには、set 拡張メモリパラメータを使用します。
- show subscriber radiusinterface
  “Not Configured” の場合は、set subscriber radiusinterface コマンドを使用して、RADIUSListener サービスを指定します。

サブスクライバロギングが有効になっている場合は、ログファイルからより詳細な情報を取得できます。

サブスクライバ認識トラフィックステアリング

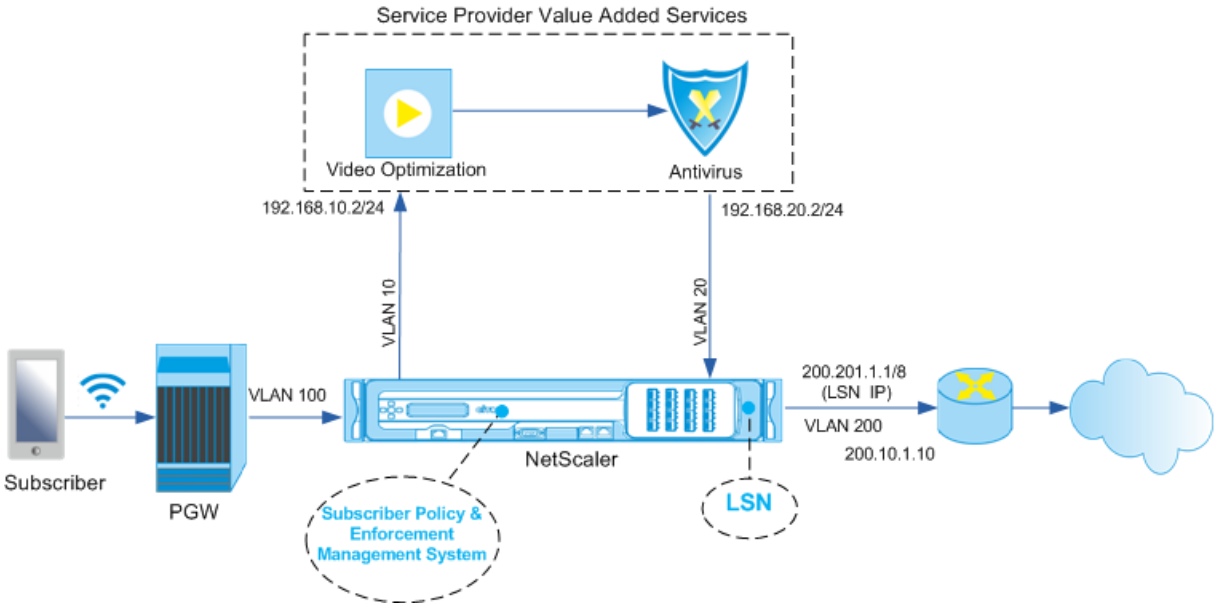
October 7, 2021

トラフィックステアリングは、加入者のトラフィックをあるポイントから別のポイントへ誘導します。サブスクライバがネットワークに接続すると、パケット Gateway は IP アドレスとサブスクライバを関連付け、データパケットを

Citrix ADC アプライアンスに転送します。アプライアンスは、Gx インターフェイスを経由して PCRF サーバと通信し、ポリシー情報を取得します。ポリシー情報に応じて、アプライアンスは次のいずれかのアクションを実行します。

- データパケットを別のサービスセットに転送します (次の図を参照)。
- パケットをドロップします。
- アプライアンスで LSN が設定されている場合は、大規模 NAT（LSN）のみを実行します。

次の図に示す値は、図の後の CLI プロシージャで設定されています。Citrix ADC アプライアンス上のコンテンツスイッチング仮想サーバーは、定義されたルールに応じて付加価値サービスに要求を送信するか、またはスキップし、LSN の実行後にパケットをインターネットに送信します。



CLI を使用して上記のデプロイメントのトラフィックステアリングを設定するには

アプライアンスのサブネット IP（SNIP）アドレスを追加します。

例:

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 100.100.100.1 255.0.0.0 -type snip

add ns ip 200.200.200.1 255.0.0.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
```



```
<!--NeedCopy-->
```

VLAN を追加します。VLAN は、アプライアンスがトラフィックの送信元を識別するのに役立ちます。VLAN をインターフェイスおよびサブネット IP アドレスにバインドします。

例:

```
add vlan 10

add vlan 20

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.1.1.1 255.0.0.0
<!--NeedCopy-->
```

加入者トラフィックがアプライアンスに到着する VLAN を指定します。サブスクライバセッション内のサービスパス名の検索先をアプライアンスに指示するサービスパス AVP を指定します。プライマリ PCEF 機能については、インターフェイスタイプを RadiusAndGx として指定します。

例:

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
    servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

Diameter タイプのサービスと仮想サーバーを構成し、サービスを仮想サーバーにバインドします。次に、PCRF レルムおよびサブスクライバ Gx インターフェイスパラメータを指定します。プライマリ PCEF 機能については、RADIUS リスナーサービスおよび RADIUS インターフェイスを設定します。

例:


```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

VAS と入力 VLAN を関連付けるサービス機能を追加します。サービスパスを追加してチェーンを定義します。つまり、パケットの送信先となる VAS と、その VAS への送信順序を指定します。サービスパス名は、通常 PCRF によって送信されます。ただし、次のいずれかに該当する場合は、デフォルトのサブスクライバプロファイル（*）のサービスパスが適用されます。

- PCRF には加入者情報がありません。
- 加入者情報には、この AVP は含まれません。
- アプライアンスは PCRF を照会できません。たとえば、PCRF を表すサービスは DOWN です。

この名前を含むサービスパス AVP は、グローバル構成の一部としてすでに設定されている必要があります。サービス関数をサービスパスにバインドします。サービスインデックスは、VAS がチェーンに追加される順序を指定します。最大値 (255) はチェーンの開始を示します。

例**:**

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
<!--NeedCopy-->
```

LSN 設定を追加します。つまり、NAT プールを定義し、アプライアンスが LSN を実行する必要があるクライアント

を特定します。

```
add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1
<!--NeedCopy-->
```

アプライアンスはデフォルトで LSN を実行します。LSN を上書きするには、overrideLsn パラメータを有効にしてネットプロファイルを作成し、このプロファイルを付加価値サービス（VAS）用に構成されているすべてのロードバランシング仮想サーバにバインドする必要があります。

例:

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

set lb vserver vs1 -netprofile np1
<!--NeedCopy-->
```

アプライアンスで VAS を設定します。これには、サービスと仮想サーバーを作成し、サービスを仮想サーバーにバインドすることが含まれます。

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

```



```
bind lb vserver vint sint
<!--NeedCopy-->
```

コンテンツスイッチング（CS）設定を追加します。これには、仮想サーバ、ポリシー、および関連するアクションが含まれます。トラフィックは CS 仮想サーバに着信し、適切なロードバランシング仮想サーバにリダイレクトされます。仮想サーバをサービス機能に関連付ける式を定義します。

例**:**

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule  SUBSCRIBER.SERVICEPATH.IS_NEXT(”SF1”) &&
   SYS.VSERVER(”vs1”).STATE.EQ(UP)” -action csact1

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -lbvserver vint
<!--NeedCopy-->
```

GUI を使用してアプライアンスでトラフィックステアリングを設定するには

1. [システム] > [ネットワーク] > [**IP**] に移動し、サブネットの IP アドレスを追加します。
2. システムに移動 > 通信網 > **VLAN** と **VLAN** の追加、VLAN をインターフェースとサブネット IP アドレスにバインドします。
3. [トラフィック管理] > [サービスチェーン] > [サービスパスの入力 **VLAN** の設定] に移動し、入力 VLAN を指定します。
4. [トラフィック管理] > [サブスクライバ**] > [**パラメータ**] > [**サブスクライバパラメータの設定] に移動し、次の項目を指定します。
   - インタフェースタイプ: **[RadiusAndGx]** を指定します。
   - 直径仮想サーバ、PCRF レルム、およびサブスクライバ GX インターフェイスパラメータを設定します。
   - RADIUS インターフェイスパラメータを指定します。
5. 「トラフィック管理」 >「サービス連鎖」 >「サービス 機能」に移動し、付加価値サービスを入力 VLAN に関連付けるサービス機能を追加します。
6. [システム] > [ネットワーク] > [大規模 **NAT**] に移動します。[プール] をクリックしてプールを追加します。[クライアント] をクリックし、クライアントを追加します。[**Groups**] をクリックしてグループを追加し、クライアントを指定します。グループを編集し、プールをこのグループにバインドします。

7. 「システム」>「ネットワーク」>「ネットプロファイル」に移動し、ネットプロファイルを追加します。[**LSN** の上書き] を選択します。必要に応じて、[システム] > [ネットワーク**] > [**設定**] > [**レイヤ **3** パラメータの構成] に移動し、[**LSN** の上書き] が選択されていないことを確認します。
8. トラフィック管理 > 負荷分散 > 仮想サーバーに移動し、アプライアンスの仮想サーバーと付加価値サービスを設定します。サービスとネットプロファイルを仮想サーバーにバインドします。
9. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバー、ポリシー、およびアクションを設定します。ターゲットの負荷分散仮想サーバーを指定します。

GUI を使用してアプライアンスでサービスチェーンを構成するには

1. [システム] > [ネットワーク] > [**IP**] に移動し、サブネットの IP アドレスを追加します。
2. システムに移動 > 通信網 > **VLAN** と **VLAN** の追加、VLAN をインターフェースとサブネット IP アドレスにバインドします。
3. [トラフィック管理] > [サービスチェーン] > [サービスパスの入力 **VLAN** の設定] に移動し、入力 VLAN を指定します。
4. [トラフィック管理] > [サブスクライバ**] > [**パラメータ**] > [**サブスクライバパラメータの設定] に移動し、次の項目を指定します。
   - インタフェースタイプ: **[RadiusAndGx]** を指定します。
   - 直径仮想サーバ、PCRF レルム、およびサブスクライバ GX インターフェイスパラメータを設定します。
   - RADIUS インターフェイスパラメータを指定します。
5. 「トラフィック管理」>「サービス連鎖」>「サービス 機能」に移動し、付加価値サービスを入力 VLAN に関連付けるサービス機能を追加します。
6. [システム] > [ネットワーク] > [大規模 **NAT]** に移動します。[プール] をクリックしてプールを追加します。[クライアント] をクリックし、クライアントを追加します。[**Groups**] をクリックしてグループを追加し、クライアントを指定します。グループを編集し、プールをこのグループにバインドします。
7. 「システム」>「ネットワーク」>「ネットプロファイル」に移動し、ネットプロファイルを追加します。[**LSN** の上書き] を選択します。必要に応じて、[システム] > [ネットワーク**] > [**設定**] > [**レイヤ **3** パラメータの構成] に移動し、[**LSN** の上書き] が選択されていないことを確認します。
8. トラフィック管理 > 負荷分散 > 仮想サーバーに移動し、アプライアンスの仮想サーバーと付加価値サービスを設定します。サービスとネットプロファイルを仮想サーバーにバインドします。
9. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバー、ポリシー、およびアクションを設定します。ターゲットの負荷分散仮想サーバーを指定します。

サブスクライバ認識サービスチェーン

October 7, 2021

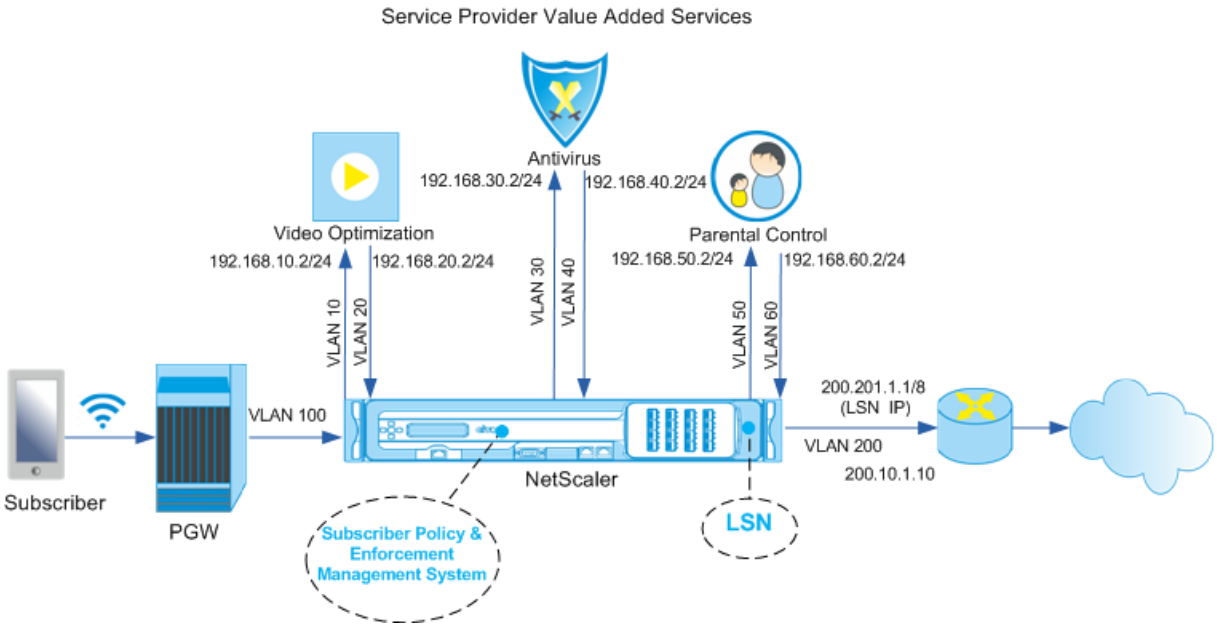
電話会社のネットワークを通過するデータトラフィックが大幅に増加するにつれて、サービスプロバイダーがすべての Value added Services（VAS; 付加価値サービス）を通過するすべてのトラフィックを操縦することはもはや不

可能です。サービスプロバイダーは、VAS の使用を最適化し、トラフィックをインテリジェントに操縦してユーザエクスペリエンスを向上させることができる必要があります。たとえば、ビデオを含まないトラフィックでは、ビデオの最適化は必要ありません。さらに、加入者が 4G ネットワークに接続されている場合、コンテンツを高精細（HD）でストリーミングすることができ、ビデオの最適化が不要になる場合があります。ただし、ビデオの最適化は、3G ネットワーク内のユーザーのエクスペリエンスを向上させます。同様に、キャッシングは、より速く、より良いユーザーエクスペリエンスを提供し、加入者プランに応じて有効にすることができます。VAS のもう 1 つの例は、ペアレンタルコントロールです。両親が未成年の子供に携帯電話を提供する場合、彼らは彼らの子供が訪問するウェブサイト上の制御のいくつかの種類をしたいと思います。

上記以上のことを行うには、サービスプロバイダーが加入者ごとに付加価値サービスを提供できる必要があります。つまり、サービスプロバイダーネットワークのエンティティは、加入者情報を抽出し、この情報に基づいてパケットをインテリジェントに操縦できる必要があります。

サービスチェーンは、加入者からのトラフィックがインターネットに行く前に通過しなければならないサービスのセットを決定します。Citrix ADC は、すべてのトラフィックをすべてのサービスに送信する代わりに、その加入者に定義されたポリシーに基づいて、加入者からのすべての要求を特定のサービスセットにインテリジェントにルーティングします。

次の図は、サービスチェーンに関連するエンティティを示しています。表示される値は、図の後の手順で設定されています。Citrix ADC アプライアンス上のコンテンツスイッチング仮想サーバーは、定義されたルールに応じて付加価値サービスに要求を送信するか、またはスキップし、LSN の実行後にパケットをインターネットに送信します。



CLI を使用して上記のデプロイメントのサービスチェーンを構成するには

アプライアンスのサブネット IP（SNIP）アドレスを追加します。

例**:**


```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.30.1 255.255.255.0 -type snip

add ns ip 192.168.40.1 255.255.255.0 -type snip

add ns ip 192.168.50.1 255.255.255.0 -type snip

add ns ip 192.168.60.1 255.255.255.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
<!--NeedCopy-->
```

VLAN を追加します。VLAN は、アプライアンスがトラフィックの送信元を識別するのに役立ちます。VLAN をインターフェイスおよびサブネット IP アドレスにバインドします。各 VAS の入力 VLAN と出力 VLAN を追加します。

例:

```
add vlan 10

add vlan 20

add vlan 30

add vlan 40

add vlan 50

add vlan 60

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

```



```
bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.30.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.40.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.50.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.60.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.201.1.1 255.0.0.0
<!--NeedCopy-->
```

加入者トラフィックがアプライアンスに到着する VLAN を指定します。サブスクライバセッション内のサービスパス名の検索先をアプライアンスに指示するサービスパス AVP を指定します。プライマリ PCEF 機能については、インターフェイスタイプを RadiusAndGx として指定します。

例:

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
    servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

Diameter タイプのサービスと仮想サーバーを構成し、サービスを仮想サーバーにバインドします。次に、PCRF レルムおよびサブスクライバ Gx インターフェイスパラメータを指定します。プライマリ PCEF 機能については、RADIUS リスナーサービスおよび RADIUS インターフェイスを設定します。

例:

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

```



```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

VAS と入力 VLAN を関連付けるサービス機能を追加します。サービスパスを追加してチェーンを定義します。つまり、パケットの送信先となる VAS と、その VAS への送信順序を指定します。サービスパス名は、通常 PCRF によって送信されます。ただし、次のいずれかに該当する場合は、デフォルトのサブスクライバプロファイル（*）のサービスパスが適用されます。

- PCRF には加入者情報がありません。
- 加入者情報には、この AVP は含まれません。
- アプライアンスは PCRF を照会できません。たとえば、PCRF を表すサービスは DOWN です。

この名前を含むサービスパス AVP は、グローバル構成の一部として事前に設定する必要があります。サービス関数をサービスパスにバインドします。サービスインデックスは、VAS がチェーンに追加される順序を指定します。最大値 (255) はチェーンの開始を示します。

例:

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicefunction SF2 -ingressVLAN 40

add ns servicefunction SF3 -ingressVLAN 60

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

bind ns servicepath pol1 -servicefunction SF2 -index 254

bind ns servicepath pol1 -servicefunction SF3 -index 253

add ns servicepath pol2

bind ns servicepath pol2 -servicefunction SF2 -index 255

add ns servicepath pol3

bind ns servicepath pol3 -servicefunction SF1 -index 255
```



```
22
23 add subscriber profile * -subscriberrules default_path
24 <!--NeedCopy-->
```

LSN 設定を追加します。つまり、NAT プールを定義し、アプライアンスが LSN を実行する必要があるクライアントを特定します。

例:

```
1  add lsn pool pool1
2
3  bind lsn pool pool1 200.201.1.1
4
5  add lsn client client1
6
7  bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0
8
9  add lsn group group1 -clientname client1
10
11 bind lsn group group1 -poolname pool1
12 <!--NeedCopy-->
```

アプライアンスはデフォルトで LSN を実行します。LSN を上書きするには、overrideLsn パラメータを有効にしてネットプロファイルを作成し、付加価値サービス（VAS）用に構成されているすべてのロードバランシング仮想サーバにこのプロファイルをバインドする必要があります。

例:

```
1  add netprofile np1
2
3  set netprofile np1 -overrideLsn ENABLED
4
5  set lb vserver vs1 -netprofile np1
6  <!--NeedCopy-->
```

アプライアンスで VAS を設定します。これには、サービスと仮想サーバーを作成し、サービスを仮想サーバーにバインドすることが含まれます。

例:

```
1  add service vas1 192.168.10.2 ANY 80 -usip YES
```



```
2
3  add service vas2 192.168.30.2 ANY 80 -usip YES
4
5  add service vas3 192.168.50.2 ANY 80 -usip YES
6
7  add service sint 200.10.1.10 ANY 80 -usip YES
8
9  add lb vserver vs1 ANY -m MAC -l2Conn ON
10
11 add lb vserver vs2 ANY -m MAC -l2Conn ON
12
13 add lb vserver vs3 ANY -m MAC -l2Conn ON
14
15 add lb vserver vint ANY -m MAC -l2Conn ON
16
17 bind lb vserver vs1 vas1
18
19 bind lb vserver vs2 vas2
20
21 bind lb vserver vs3 vas3
22
23 bind lb vserver vint sint
24 <!--NeedCopy-->
```

コンテンツスイッチング（CS）設定を追加します。これには、仮想サーバ、ポリシー、および関連するアクションが含まれます。トラフィックは CS 仮想サーバに着信し、適切なロードバランシング仮想サーバにリダイレクトされます。仮想サーバをサービス機能に関連付ける式を定義します。

例**:**

```
1  add cs vserver cs1 ANY * 80 -l2Conn ON
2
3  add cs action csact1 -targetLBVserver vs1
4
5  add cs action csact2 -targetLBVserver vs2
6
7  add cs action csact3 -targetLBVserver vs3
8
9  add cs action csactint -targetLBVserver vint
10
11 add cs policy cspol1 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
       SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1
12
```



```
13 add cs policy cspol2 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF2") &&
      SYS.VSERVER("vs2").STATE.EQ(UP)" -action csact2
14
15 add cs policy cspol3 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF3") &&
      SYS.VSERVER("vs3").STATE.EQ(UP)" -action csact3
16
17 bind cs vserver cs1 -policyName cspol1 -priority 110
18
19 bind cs vserver cs1 -policyName cspol2 -priority 120
20
21 bind cs vserver cs1 -policyName cspol3 -priority 130
22
23 bind cs vserver cs1 -lbvserver vint
24 <!--NeedCopy-->
```

GUI を使用してアプライアンスでサービスチェーンを構成するには

1. [システム] > [ネットワーク] > [**IP**] に移動し、サブネットの IP アドレスを追加します。
2. [システム] > [ネットワーク] > [**VLAN**] に移動し、VLAN を追加し、VLAN をインターフェイスおよびサブネット IP アドレスにバインドします。
3. [トラフィック管理] > [サービスチェーン] > [サービスパスの入力 **VLAN** の設定] に移動し、入力 VLAN を指定します。
4. [トラフィック管理] > [サブスクライバ**] > [**パラメータ**] > [**サブスクライバパラメータの設定] に移動し、次の項目を指定します。
   - インタフェースタイプ: **[RadiusAndGx]** を指定します。
   - 直径仮想サーバ、PCRF レルム、およびサブスクライバ GX インターフェイスパラメータを設定します。
   - RADIUS インターフェイスパラメータを指定します。
5. 「トラフィック管理」>「サービス連鎖」>「サービス 機能」に移動し、付加価値サービスを入力 VLAN に関連付けるサービス機能を追加します。
6. [システム] > [ネットワーク] > [大規模 **NAT]** に移動します。[プール] をクリックしてプールを追加します。[クライアント] をクリックし、クライアントを追加します。[**Groups**] をクリックしてグループを追加し、クライアントを指定します。グループを編集し、プールをこのグループにバインドします。
7. 「システム」>「ネットワーク」>「ネットプロファイル」に移動し、ネットプロファイルを追加します。[**LSN** の上書き] を選択します。必要に応じて、[システム] > [ネットワーク**] > [**設定**] > [**レイヤ **3** パラメータの構成] に移動し、[**LSN** の上書き] が選択されていないことを確認します。
8. トラフィック管理 > 負荷分散 > 仮想サーバーに移動し、アプライアンスの仮想サーバーと付加価値サービスを設定します。サービスとネットプロファイルを仮想サーバーにバインドします。
9. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバー、ポリシー、およびアクションを設定します。ターゲットの負荷分散仮想サーバーを指定します。

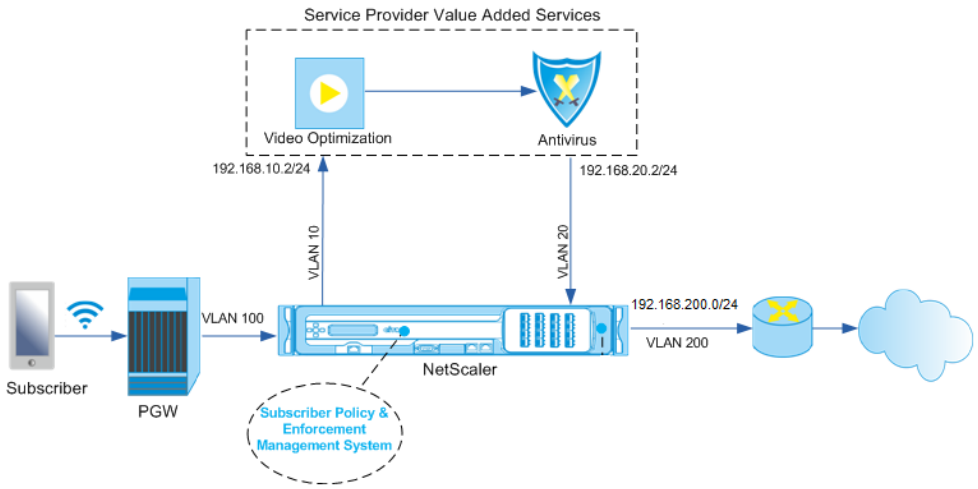
TCP 最適化によるサブスクライバ認識トラフィックステアリング

October 7, 2021

トラフィックステアリングは、加入者のトラフィックをあるポイントから別のポイントへ誘導します。サブスクライバがネットワークに接続すると、パケット Gateway は IP アドレスとサブスクライバを関連付け、データパケットを Citrix ADC アプライアンスに転送します。アプライアンスは、Gx インターフェイスを経由して PCRF サーバと通信し、加入者ポリシー情報を取得します。ポリシー情報に応じて、アプライアンスは次のいずれかのアクションを実行します。

- データパケットを別のサービスセットに転送します (次の図を参照)。
- TCP 最適化だけを実行します。

次の図に示す値は、図の後の CLI プロシージャで設定されています。Citrix ADC アプライアンス上のコンテンツスイッチング仮想サーバーは、付加価値サービスに要求を送信するか、要求をスキップして、定義されたルールに応じて TCP 最適化を実行し、パケットをインターネットに送信します。



> 注
>
> 以下に示す設定のサポートは、リリース 11.1 ビルド 50.10 で導入されました。

CLI を使用して上記の展開のトラフィックステアリングを設定するには、次の手順を実行します。

1. アプライアンスのサブネット IP（SNIP）アドレスを追加します。

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

```



```
add ns ip 192.168.100.1 255.255.255.0 -type snip

add ns ip 192.168.200.1 255.255.255.0 -type snip

add ns ip 10.102.232.236 255.255.255.0  - type snip
<!--NeedCopy-->
```

2. VLAN を追加します。VLAN は、アプライアンスがトラフィックの送信元を識別するのに役立ちます。VLAN をインターフェイスおよびサブネット IP アドレスにバインドします。

```
add vlan 10

add vlan 20

add vlan 100

add vlan 200

add vlan 102

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1
   255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1
   255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 192.168.100.1
   255.255.255.0

bind vlan 200 -ifnum 1/2 -tagged -IPAddress 192.168.200.1
   255.255.255.0

bind vlan 102  - ifnum 1/1  - tagged  - IPAddress 10.102.232.236
   255.255.255.0
<!--NeedCopy-->
```

3. Diameter タイプのサービスと仮想サーバーを構成し、サービスを仮想サーバーにバインドします。加入者 Gx インターフェイスパラメータの PCRF レルムと値を指定します。また、加入者セッション内でアプライアンスがサービスパス名を検索できる場所を示すサービスパス AVP も指定します。プライマリ PCEF 機能については、RADIUS リスナーサービスと RADIUS インターフェイスを設定し、インターフェイスタイプを「RadiusAndGx」と指定します。


```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER
   -persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set extendedmemoryparam -memLimit 2558

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net

set subscriber gxinterface -servicepathAVP 1001 1005 -
   servicepathVendorid 10415

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

4. 次のいずれかに該当する場合に適用されるデフォルトのサブスクライバプロファイル（*）を指定します。

   - PCRF には加入者情報がありません。
   - 加入者情報には、サービスパス AVP は含まれません。
   - アプライアンスは PCRF を照会できません。たとえば、PCRF を表すサービスは DOWN です。

```
add subscriber profile * -subscriberrules default_path
<!--NeedCopy-->
```

5. VAS および TCP 最適化パスの TCP プロファイルをそれぞれ作成します。VAS にステアリングされたトラフィックは、VAS からの送信前または送信後に TCP 最適化を行いません。したがって、VAS プロファイルの TCP モードは TRANSPARENT に設定し、TCPOpt プロファイルの TCP モードは ENDPOINT に設定する必要があります。

   add ns tcpProfile VAS –tcpMode TRANSPARENT

   add ns tcpProfile TCPOpt -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop

ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

6. VAS サーバのロードバランシングを設定します。TCP タイプのアドレス指定不可能な仮想サーバーを作成します。VAS サーバの IP アドレスを使用して TCP サービスを作成し、サービスを仮想サーバにバインドします。仮想サーバおよびサービスは、VAS パス用に作成された透過的 TCP プロファイルを使用します。

```
add service vas1 192.168.10.2 TCP * -usip YES -useproxyport NO -
    TCPB NO -tcpProfileName VAS

add service vas2 192.168.10.3 TCP * -usip YES -useproxyport NO -
    TCPB NO -tcpProfileName VAS

add lb vserver vs1 TCP -m MAC -l2Conn ON  - tcpProfileName VAS

bind lb vserver vs1 vas1

bind lb vserver vs1 vas2
<!--NeedCopy-->
```

7. ロードバランシング仮想サーバを追加して、VAS 出力トラフィックをキャプチャします。この vserver は VAS 出力 VLAN を監視し、トランスペアレント TCP プロファイルを使用します。

```
add lb vserver vsint TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ(20)"
     - Listenpriority 30  - l2Conn ON  - tcpProfileName VAS
<!--NeedCopy-->
```

8. 無線側 VLAN 内のトラフィックをリッスンし、TCP 最適化パス用に作成されたエンドポイント TCP プロファイルを使用する TCP 最適化仮想サーバを追加します。

```
add lb vserver vs-TcpOpt TCP * * -Listenpolicy "client.vlan.id.eq
    (100)"  - Listenpriority 20 -l2Conn ON -tcpProfileName TCPOpt
<!--NeedCopy-->
```

9. コンテンツスイッチング（CS）設定を追加します。これには、仮想サーバ、ポリシー、および関連するアクションが含まれます。CS 仮想サーバは、トラフィックを受信し、定義された CS ポリシーに従って、適切なロードバランシング仮想サーバにリダイレクトします。最高のプライオリティを持つワイヤレス側 VLAN 内のトラフィックをリッスンし、エンドポイント TCP プロファイルを使用する CS TCP 仮想サーバを作成します。加入者規則が「vas」である場合に TRUE と評価される CS ポリシーを作成し、トラフィックを VAS に誘導する

CS アクションを指定します。TCP 最適化仮想サーバーをデフォルトの LB 仮想サーバーにします。ルールが「vas」以外のサブスクライバトラフィックは、デフォルトの LB vserver を通過します。

```
add cs vserver cs1 TCP * * -Listenpolicy ”client.vlan.id.eq(100)” - Listenpriority 10 -l2Conn ON  - tcpProfileName TCPOpt

add cs action csact1 -targetLBVserver vs1

add cs policy cspol1 -rule  SUBSCRIBER.RULE_ACTIVE(”vas”) && SYS.VSERVER(”vs1”).STATE.EQ(UP)” -action csact1

bind cs vserver cs1 -policyName cspol1

bind cs vserver cs1 -lbvserver vs-TcpOpt
<!--NeedCopy-->
```

10. インターネットに静的ルートまたはポリシーベースのルートを追加します。この設定では、ダイナミックルーティングもサポートされます。次の例では、ポリシーベースのルートを使用します。

```
add ns pbr pbr-vlan100-to-vlan200 ALLOW -nextHop 192.168.200.10 -vlan 100 -priority 10

add ns pbr pbr-vlan20-to-vlan200 ALLOW -nextHop 192.168.200.10 -vlan 20 -priority 11

apply ns pbrs
<!--NeedCopy-->
```

注

- CS ポリシーには、サブスクライバ表現に加えて、IP アドレスとポート番号を含めることができます。たとえば、サブスクライバ. ルール _ アクティブ (「バス」) && & (クライアント.TCP.DSTPORT.EQ (80) || クライアント.TCP.DSTPORT.EQ (443) です。また、HTTP ベースの式を含めることもできます。たとえば、HTTP.REQ.HOSTNAME.DOMAIN.EQ (「somedomain.com」) などです。この場合、TCP エンティティ (vserver、サービスなど) を HTTP に置き換えます。TCP プロファイル設定は変わりません。
- IPv6 サブスクライバをサポートする IPv6 構成（アドレス、ルート、PBR）を追加します。Happy Eyeball のクライアントアプリケーションは、VAS と TCP の両方の最適化パスに対してスムーズに動作します。
- VLAN、IP アドレス、PBR、および LB 仮想サーバを VAS（vs1、vs2 など）の前に追加して、複数のサブスクライバフローをサポートします。CS 仮想サーバ「cs1」および LB 仮想サーバ「vsint」のリッス

> ンポリシーを変更して、追加の VLAN を含めます。

ポリシーベースの **TCP** プロファイルの選択

January 31, 2022

サブスクライバー属性に基づいて TCP 最適化を実行するように Citrix ADC アプライアンスを構成できます。たとえば、アプライアンスは、ユーザー機器 (UE) が接続されているネットワークに基づいて、実行時に異なる TCP プロファイルを選択できます。その結果、TCP プロファイルにいくつかのパラメーターを設定し、ポリシーを使用して適切なプロファイルを選択することで、モバイルユーザーの操作性を向上させることができます。

4G ネットワーク経由で接続する加入者用と、他のネットワーク経由で接続するユーザ用に、個別の TCP プロファイルを作成します。無線アクセステクノロジータイプ（RAT タイプ）などの加入者パラメータに基づいて選択されるポリシールールを定義します。次の例では、RAT タイプが EUTRAN の場合、より高速な接続をサポートする TCP プロファイルが選択されます (例 1)。その他すべての RAT タイプの値では、異なる TCP プロファイルが選択されます (例 2)。

無線アクセステクノロジーとそのポリシー設定の詳細については、RFC 29.212を参照してください。

> 注
>
> RAT タイプ AVP（AVP コード 1032）は「列挙」タイプで、UE にサービスを提供する無線アクセステクノロジーを識別するために使用されます。
>
> 値「1004」は、ラットがユートランであることを示します。

例 **1**:

```
add ns tcpProfile tcp2 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
    16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 1000000 -flavor BIC
     - dynamicReceiveBuffering DISABLED -sendBuffsize 1000000 -dsack
    DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 500 -maxburst 15

add appqoe action appact2 -priority HIGH -tcpprofile tcp2

add appqoe policy apppol2 -rule ”SUBSCRIBER.AVP(1032).VALUE.
    GET_UNSIGNED32(0, BIG_ENDIAN).EQ(1004)” -action appact2

bind cs vserver <name>  -policyname apppol2 -priority 20 -type request
<!--NeedCopy-->
```

例 **2**:


```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
    16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 150000 -flavor BIC
    - dynamicReceiveBuffering DISABLED -sendBuffsize 150000 -dsack
    DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 200 -maxburst 15

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "SUBSCRIBER.AVP(1032).VALUE.
    GET_UNSIGNED32(0, BIG_ENDIAN).NE(1004)" -action appact1

bind cs vserver <name>  -policyname apppol1 -priority 10 -type request
<!--NeedCopy-->
```

Diameter、SIP、および SMPP プロトコルに基づくコントロールプレーントラフィックの負荷分散

October 7, 2021

コントロールプレーントラフィックが増加すると、トラフィックがサーバ間で最適に分散されないため、サーバがボトルネックになる可能性があります。したがって、メッセージはロードバランシングする必要があります。Citrix ADC アプライアンスは、直径、SIP、SMPP の負荷分散をサポートします。

SIP

Citrix ADC を使用すると、プロキシサーバーのグループに UDP または TCP（TLS を含む）を介して SIP メッセージを負荷分散できます。Citrix ADC では、Call-ID ベースの永続性と Call-ID ハッシュ負荷分散方式も提供しています。この方式では、特定の SIP セッションのパケットを同じ負荷分散 SIP サーバーに送信します。
Citrix ADC デフォルトの式言語には、セッション開始プロトコル（SIP）接続で動作するいくつかの式が含まれています。これらの式は、要求/応答ベースで動作する SIP プロトコルのポリシーで使用することを意図しています。これらの式は、コンテンツスイッチング、レート制限、レスポンダ、および書き換えポリシーで使用できます。

詳細については、「SIP サーバグループのロードバランシング」を参照してください。

SMPP

個人と銀行、広告主、ディレクトリサービスなどの付加価値サービスプロバイダーの間で、Short Message peer to peer（SMPP; ショートメッセージピアツーピア）プロトコルを使用して、毎日数百万のショートメッセージが交換されます。多くの場合、サーバーが過負荷になり、トラフィックがサーバー間で最適に分散されないため、メッセージの配信が遅れます。

Citrix ADC アプライアンスは、サーバー間でメッセージを最適に配信し、パフォーマンスの低下や停止を防ぎます。Citrix ADC アプライアンスは、以下の操作を行います。

- サーバとクライアントから発信されたメッセージの負荷分散
- メッセージセンターの健全性を監視する
- メッセージセンターのコンテンツスイッチングサポートを提供
- 連結されたメッセージを処理します。

制限：メッセージセンターからのメッセージ ID（59 バイトを超えるもの）はサポートされていません。メッセージセンターから返されたメッセージ ID の長さが 59 バイトを超える場合、補助操作は失敗し、Citrix ADC アプライアンスはエラーメッセージで応答します。

詳細については、SMPP ロードバランシングを参照してください。

Diameter

直径は、50 以上のプロトコル（アプリケーションとも呼ばれます）が構築されたベースプロトコルです。したがって、Telco ネットワークで生成されるトラフィックの直径は高くなります。この直径のトラフィックを最適に維持するために、Citrix ADC アプライアンスは負荷分散とコンテンツスイッチングを行い、リレーエージェントとして機能します。さらに、アプライアンスは、書き換えおよびレスポンダ機能を提供します。アプライアンスは Diameter メッセージのレート制限をサポートしています。

詳細については、「DiametR ロードバランシングの設定」を参照してください。

テレコムサービスプロバイダの負荷分散、キャッシング、ロギングなどの **DNS** インフラストラクチャ/トラフィックサービスの提供

October 7, 2021

通信サービスプロバイダーは、Citrix ADC アプライアンスを DNS プロキシとして機能するように構成できます。DNS プロキシの重要な機能である DNS レコードのキャッシュは、Citrix ADC アプライアンスでデフォルトで有効になっています。これにより、Citrix ADC アプライアンスは、翻訳の繰り返しに迅速に対応できるので、カスタマーエクスペリエンスが向上し、帯域幅も節約できます。は、DNS ネームサーバからの応答をキャッシュします。アプライアンスは DNS クエリーを受信すると、クエリーされたドメインがキャッシュ内でチェックされます。クエリされたドメインのアドレスがキャッシュに存在する場合、Citrix ADC アプライアンスは対応するアドレスをクライアントに返します。それ以外の場合、クエリは DNS ネームサーバーに転送され、アドレスの可用性をチェックし、Citrix ADC アプライアンスに返します。その後、Citrix ADC アプライアンスはアドレスをクライアントに返します。

以前にキャッシュされたドメインに対する要求の場合、Citrix ADC アプライアンスは、構成された DNS サーバーに問い合わせることなく、キャッシュからドメインのアドレスレコードを提供するため、帯域幅を節約します。

11.0 リリース以降では、Citrix ADC は受信した DNS 要求と、クライアントに送信する応答も記録します。通信サービスプロバイダーは、このログを使用して、次のことができます。

- クライアントに対する DNS 応答の監査
- DNS クライアントの監査
- DNS 攻撃の検出と防止
- トラブルシューティング

詳細については、「ドメインネームシステム」を参照してください。

通信サービスプロバイダーのコアネットワーク間で **GSLB** を使用した加入者の負荷分散の提供

October 7, 2021

スケーラビリティ、高可用性、およびパフォーマンスは、サービスプロバイダの導入に不可欠です。多くのサービスプロバイダーがインフラストラクチャを 1 か所または複数の場所に展開していますが、これらの展開には次のような制限があります。

- サイトがパブリックインターネットのすべてまたは一部に接続できなくなった場合、ユーザーや顧客がアクセスできなくなり、ビジネスに大きな影響を与える可能性があります。
- 地理的に離れた場所からサイトにアクセスするユーザーは、大きく変動する遅延が発生する可能性があります。これは、HTTP がコンテンツを転送するために必要な往復の数が多いため、悪化しています。

Citrix ADC アプライアンスのグローバルサーバー負荷分散（GSLB）は、複数の地理的な場所に展開されたサイト間でトラフィックを分散することによって、これらの問題を克服します。GSLB は、インターネット上のさまざまな場所からコンテンツを提供することにより、ネットワーク帯域幅のボトルネックの影響を軽減し、特定のサイトでネットワーク障害が発生した場合に堅牢性を提供します。ユーザーは、リクエスト時に最も近いサイトまたは最も負荷の低いサイトに自動的に移動できるため、ダウンロードの遅延やサービスの中断が長くなる可能性を最小限に抑えることができます。

Citrix ADC アプライアンスのグローバルサーバー負荷分散は、次の目的で使用できます。

- アクティブデータセンターとスタンバイデータセンターで構成されるアクティブ/スタンバイデータセンターの設定を構成することにより、ディザスタリカバリまたは高可用性を実現します。障害イベントの結果としてフェイルオーバーが発生すると、スタンバイ・データ・センターは稼働状態になります。
- 複数のアクティブなデータセンターで構成されるアクティブ-アクティブデータセンターの設定を構成することにより、高可用性と高速性を実現します。クライアント要求は、アクティブなデータセンター間で負荷分散されます。
- 近接設定を構成して、地理的距離またはネットワーク距離が最も近いデータセンターにクライアント要求を転送します。
- 完全 DNS 解決、GSLB は、A、AAAA、CNAME タイプの DNS クエリを処理し、DNS 機能オプションは、MX や PTR などの他のすべてのタイプの DNS クエリを処理することができます。また、再帰的な解決が有効になっている場合、アプライアンスは Citrix ADC アプライアンスで構成されていないドメイン名に対する DNS クエリを転送します。

詳細については、「グローバルサーバーの負荷分散」を参照してください。

キャッシュリダイレクション機能を使用した帯域幅使用率

October 7, 2021

インターネット上の Web トラフィックの量は膨大で、そのトラフィックの大部分が冗長です。複数のクライアントが Web サーバーに同じコンテンツを繰り返し要求し、帯域幅の非効率的な使用につながります。インターネットサービスプロバイダー（ISP）は、Citrix ADC アプライアンスのキャッシュリダイレクト機能を使用して、オリジンサーバーではなくキャッシュサーバーからコンテンツを提供できます。Citrix ADC アプライアンスは、着信要求を分析し、キャッシュ可能なデータに対する要求をキャッシュサーバーに送信し、キャッシュ不可能な要求と動的 HTTP 要求をオリジンサーバーに送信します。Citrix ADC キャッシュリダイレクト機能はポリシーベースです。デフォルトでは、ポリシーに一致するリクエストはオリジンサーバーに送信され、他のすべてのリクエストはキャッシュサーバーに送信されます。コンテンツスイッチングとキャッシュリダイレクトを組み合わせて、選択的なコンテンツをキャッシュし、特定の種類の要求コンテンツに対して特定のキャッシュサーバーからコンテンツを提供できます。

詳細については、「キャッシュリダイレクト」を参照してください。

Citrix ADC TCP の最適化

October 7, 2021

Citrix ADC アプライアンスは、最新の 3.5 および 4G ネットワークに適した高度な TCP チューニングと最適化技術と機能を提供し、ユーザーエクスペリエンスと認識されるダウンロード速度を大幅に向上させます。

このセクションでは、以下に関連する詳細な手順について説明します。

- TCP 最適化のための適切な Citrix ADC T1000 シリーズモデルの選択とモバイルネットワークへの挿入
- TCP 最適化だけでなく、T1 デバイスの適切なレイヤ 2 およびレイヤ 3 設定に関する完全な設定命令

ここでは、次のトピックについて説明します。

- はじめに
- 管理ネットワーク
- ライセンス
- 高可用性
- Gi-LAN 統合
- TCP 最適化の設定
- TCP NILE を使用した TCP パフォーマンスの最適化
- 分析とレポート作成
- リアルタイム統計

- SNMP
- 技術レシピ
- トラブルシューティングのガイドライン
- よくある質問

はじめに

October 7, 2021

ハードウェア

Citrix 社は、2 つの要因に基づいて緩くかもしれない Citrix ADC モデルの広い息を提供します:

- 容量は現在、ローエンド VPX アプライアンスでは数百 Mbps からハイエンド 25000 MPX シリーズアプライアンスの場合は 160Gbps まで
- Telco グレード。Telco データセンター用の T1000 シリーズが利用可能である。

デモ、トライアル、または本番環境のニーズに適したハードウェアの選択については、Citrix の営業担当者またはサポート担当者がサポートします。

このセクションの残りの部分では、Citrix ADC T1200 をリファレンスハードウェアとして使用します。利用可能なインターフェイスの数と表記に関する表面的な違い（注*を参照）、または Citrix ADC VPX 十分に文書化された制限（注*を参照）については、指示が適用されることに注意してくださいほとんど関係なく、選択された Citrix ADC モデルの.

> 注
>
> * たとえば、T1010 モデルには、このドキュメントで使用されている 10/x 表記ではなく、通常 1/1-1/12 とマークされている 12x1GbE しかありません。
>
> **Citrix ADC VPX インスタンスは、通常、LACP 集約をサポートしていません。また、VLAN タグをサポートしていない場合があります。

初期設定

シリアル・コンソール経由

シリアルケーブルを接続したら、次の資格情報を使用して Citrix ADC アプライアンスにログオンできます。

- ユーザー名：nsroot
- パスワード：nsroot

ログインしたら、以下の画面キャプチャに示すように、Citrix ADC アプライアンスの基本的な詳細を設定します。

例**:**


```
set ns config  - IPAddress <ip_addr> -netmask <netmask>

saveconfig

reboot -warm
<!--NeedCopy-->
```

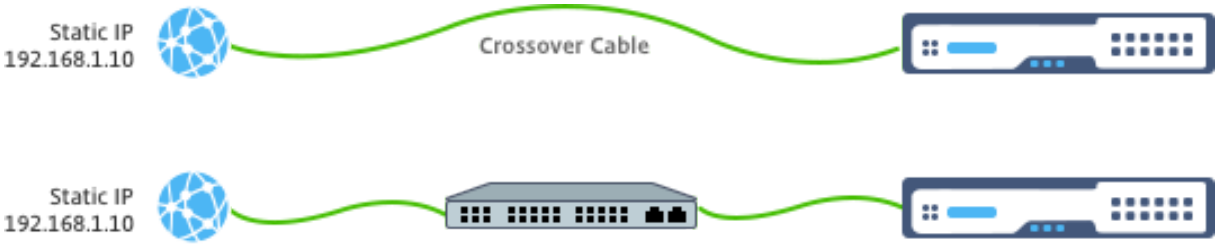
アプライアンスを再起動した後、SSH を使用して T1100 ノードをさらに構成できます。

LOM を経て

Citrix ADC アプライアンスのフロントパネルにあるライトアウト管理（LOM）ポートにより、オペレータはオペレーティングシステムから独立してアプライアンスをリモートで監視および管理できます。オペレータは、LOM ポートを介して Citrix ADC アプライアンスに接続することにより、IP アドレスの変更、電源のオフ/オン、およびコードダンプを実行できます。

LOM ポートのデフォルト IP アドレスは 192.168.1.3 です。

図。LOM モジュールの初期設定



ラップトップに静的 IP を設定し、クロスケーブルを使用して LOM インターフェイスに直接接続するか、LOM インターフェイスと同じブロードキャストドメイン内のスイッチに接続します。

初期設定では、Web ブラウザーでポートのデフォルトアドレスhttp://192.168.1.3を入力し、LOM ポートのデフォルト IP アドレスを変更します。

詳細については、『構成ガイド』を参照してください。

ソフトウェア

モバイルネットワーク向けの Citrix ADC TCP 最適化は常に進化しています。このドキュメントで説明する機能とチューニングには、Citrix ADC Telco のビルドが必要です。次に、Citrix ADC 通信会社のビルドの例を示します。

例**:**


```
show ver

NetScaler NS11.0: Build 64.957.nc, Date: Aug 26 2016, 02:00:23
<!--NeedCopy-->
```

T1000 に適切なビルドリビジョンが付属していない場合は、Citrix ADC カスタマーサポートにお問い合わせください。

> 重要
>
> 両方のアプライアンスに同じソフトウェアイメージが必要です。

SSH クライアント

Citrix ADC アプライアンスは、CLI または HTML5 の GUI を使用して構成できます。ただし、この項では CLI ベースの手順のみについて説明します。

CLI には Citrix ADC シリアルコンソールからアクセスできますが、通常は SSH クライアントを使用してリモート Citrix ADC 構成を行うことをお勧めします。

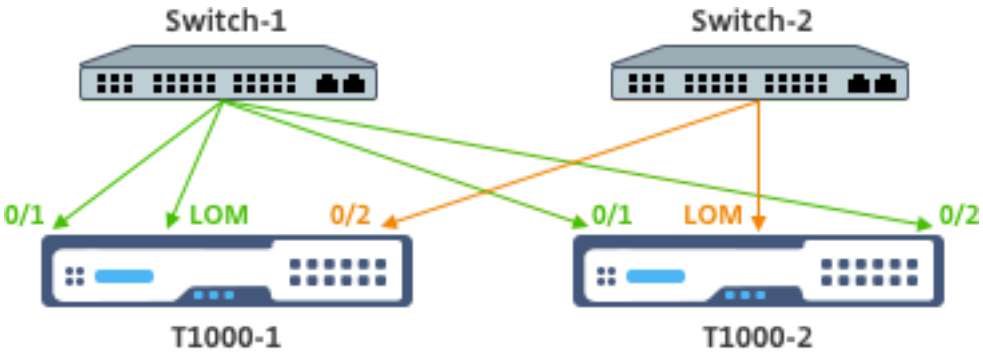
管理ネットワーク

October 7, 2021

接続

ほとんどの Citrix ADC デバイスは、0/1 および 0/2 と表記された冗長 1GbE OAM ポートを提供します。スイッチに障害が発生した場合に冗長性を確保するには、関連するポートを異なるアップストリームスイッチに接続する必要があります。

次の図に、推奨される接続の概要を示します。



Citrix ADC アプライアンスが管理ネットワークに接続されたあと、CLI および GUI への SSH または Web 接続を使用して、その後の構成手順をリモートで実行できます。

ルーティング

add route コマンドを使用して、管理ネットワークに適したルートを設定できます。関連する Gateway は、次に示すように、NSIP サブネット上で到達可能である必要があります。

例:

```
add route <network>  <netmask>  <gateway>
<!--NeedCopy-->
```

ライセンス

October 7, 2021

Citrix ADC アプライアンスに有効なライセンスファイルをインストールする必要があります。ライセンスは、予想される最大 GiLAN スループットと同数の Gbps 以上をサポートする必要があります。

ライセンスファイルは、次の画面キャプチャに示すように、SCP クライアントを介してアプライアンスの /nsconfig/license にコピーする必要があります。

例:

```
shell ls /nsconfig/license/

CNS_V3000_SERVER_PLT_Retail.lic ssl
<!--NeedCopy-->
```

次のスクリーンキャプチャに示すように、ウォームリスタートを実行して、新しいライセンスを適用します。

例:

```
reboot -warm

Are you sure you want to restart NetScaler (Y/N)? [N]:y

Done
<!--NeedCopy-->
```


再起動が完了したら、show license CLI を使用して、ライセンスが正しく適用されていることを確認します。

以下の例では、3 Gbps Premium ライセンスが正常にインストールされています。

例:

```
> show license

              License status:

                                    Web Logging: YES

                                                ...

                                 Model Number ID: 3000

                                    License Type: Premium License

Done

<!--NeedCopy-->
```

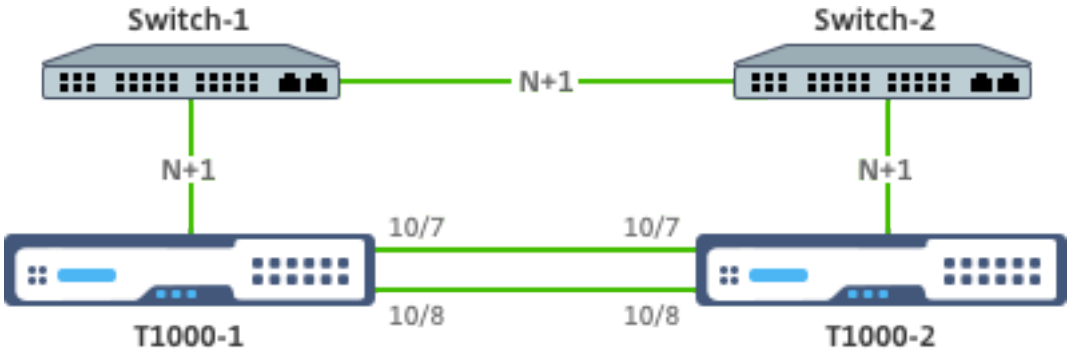
高可用性

October 7, 2021

高可用性（HA）とは、Citrix ADC デバイスペアのアクティブ/スタンバイ動作モードのことです。各デバイスには、専用の管理 IP アドレスがあります。その他のすべての IP アドレスは、ペアのアクティブデバイスによって所有されます。

接続

Citrix ADC HA ペアには複数の接続オプションがありますが、次の図は、最も推奨される接続オプションを示しています。



上の図では、接続で説明されているように、各 T1000 とそれぞれのスイッチ間の N+1 赤色のリンクは N+1 冗長性を意味します。たとえば、45 Gbps の GiLAN N=5 が適切な値であると考えると、各スイッチとそれぞれの T1000 間および 2 つのスイッチ間で 6 x 10 GbE LACP チャネルが使用されます。

OAM ネットワークから HA 通信を分離するために、Citrix ADC ペア間には追加のリンクのペアを推奨します。

Gi-LAN 統合

December 7, 2021

通常、Citrix ADC アプライアンスは、L3 ルーターと同様に、Gi-LAN に個別の L3 インラインノードとして挿入されます。

図:Gi-lan の単純な描写

接続

十分な冗長性を確保するために、アップストリームスイッチへの物理的な Citrix ADC 接続をお勧めします。たとえば、合計（アップリンク + ダウンリンク）24Gbps を処理する Gi-LAN に Citrix ADC アプライアンスが挿入されていると仮定すると、4x10GbE 以上のインターフェイスとの接続を推奨します。これにより、リンク障害が発生した場合に N+1 の冗長性が効果的に提供されます。

アップストリームスイッチの関連ポートは、LACP ポート集約用に設定する必要があります。Citrix ADC 関連する構成の概要を以下に示します。

接続構成**:**


```
set interface 10/1 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/2 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/3 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/4 -tagall ON -lacpMode ACTIVE -lacpKey 1
<!--NeedCopy-->
```

「show interface」コマンドを使用すると、LACP の適切な機能を確認できます。

インターフェイスの表示**:**

```
sh interface LA/1

    1)      Interface LA/1 (802.3ad Link Aggregate) #39

             flags=0x4100c020 <ENABLED, UP, AGGREGATE, UP, HAMON, 802.1
                q>

             MTU=1500, native vlan=1, MAC=02:e0:ed:33:88:b0, uptime 340
                h11m56s

             Requested: media NONE, speed AUTO, duplex NONE, fctl NONE,

             throughput 0

             Actual: throughput 4000

             LLDP Mode: NONE,

             RX: Pkts(918446) Bytes(110087414) Errs(0) Drops(795989)
                Stalls(0)

             TX: Pkts(124113) Bytes(15255532) Errs(0) Drops(0) Stalls
                (0)

             NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
                Muted(0)

             Bandwidth thresholds are not set.

Disable the remaining unused interfaces and turn off the monitor.
```



```
26
27 set interface 10/5  - haMonitor OFF
28 <!--NeedCopy-->
```

コマンド:

```
1 set interface 10/24  - haMonitor OFF
2
3 disable interface 10/5
4
5 disable interface 10/24
6 <!--NeedCopy-->
```

物理インターフェイスの構成は、2 つの Citrix ADC ユニット間で共有されません。したがって、HA ペアを展開する場合は、上記のコマンドを両方の Citrix ADC ノードで実行する必要があります。

HA 設定

他のすべての構成パラメータは、HA ペアの Citrix ADC ノード間で共有されます。したがって、HA 同期は、他の構成コマンドを実行する前に有効にしておく必要があります。基本的な HA 設定には、次の手順が含まれます。

1. まったく同じ Citrix ADC ハードウェア、ソフトウェア、ライセンスの使用: 異なるモデル（T1100 と MPX21550 など）またはファームウェアレベルが異なる同じモデル間では、HA ペアはサポートされません。既存の HA ペアのアップグレード（リリース 11.1 へのアップグレード）に関する適切な手順を参照してください。

2. HA ペアの確立。

例:

```
1 netscaler-1> add HA node 1 <netscaler-2-NSIP>
2
3 netscaler-2> add HA node 1 <netscaler-1-NSIP>
4 <!--NeedCopy-->
```

3. いずれかのノードで次のコマンドを実行して HA ペアの確立を確認します。両方のノードが認識され、一方はプライマリ（アクティブ）で、もう一方はセカンダリ（スタンバイ）になります。

例:

```
1 show HA node
2 <!--NeedCopy-->
```


4. フェイルセーフモードと maxFlips を有効にします。これにより、両方のノードでルートモニタに障害が発生しても、アクティブ/スタンバイステータスが常に切り替わることなく、少なくとも 1 つのノードがアクティブのままになります。

例:

```
set HA node  -failsafe ON

set HA node -maxFlips 3 -maxFlipTime 1200
<!--NeedCopy-->
```

5. 最後に、OAM ネットワークではなく、専用のイントラ Citrix ADC ポートで高可用性同期を実行できるようにします。

例:

```
add vlan 4080 -aliasName syncVlan

set HA node -syncvlan 4080
<!--NeedCopy-->
```

> 注
>
> 上の例のコマンドの VLAN 4080 は、文字どおりに解釈すべきではありません。未使用の VLAN-ID は予約されている可能性があります。

VLAN 設定

物理インターフェイスを適切に設定したら、適切な Gi-LAN VLAN を設定することができます。たとえば、100/101 の VLAN 識別子を持つ入力/出力 VLAN ペアを持つ、かなり単純な Gi-LAN 環境を考えてみましょう。

次のコマンドは、前の手順で作成した LACP チャネルの上に関連する VLAN を設定します。

```
add vlan 100
add vlan 101
bind vlan 100  -ifnum LA/1  -tagged
bind vlan 101  -ifnum LA/1  -tagged
<!--NeedCopy-->
```


IPv4 設定

通常、Citrix ADC アプライアンスでは、VLAN ごとに 1 つの SNIP が必要です。次の例では、このページの冒頭にある Gi-LAN 統合図で概説されているネットワークに /24 サブネットマスクがあることを前提としています。

```
add ns ip 192.168.1.254 255.255.255.0 -vserver DISABLED -mgmtAccess
    DISABLED
add ns ip 192.168.2.254 255.255.255.0 -vserver DISABLED -mgmtAccess
    DISABLED
<!--NeedCopy-->
```

SNIP を設定したら、適切な VLAN に関連付ける必要があります。

```
bind vlan 100 -IPAddress 192.168.1.254 255.255.255.0
bind vlan 101 -IPAddress 192.168.2.254 255.255.255.0
<!--NeedCopy-->
```

IPv4 スタティックルーティング

「管理ネットワーク」セクションで説明した例では、2 つのスタティックルーティングルールしか求めません。

- 入力ルータを経由してクライアントへの 10.0.0.0/8 スタティックルート
- 出力ルータを経由したインターネットへのデフォルトルート

例:

```
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```

IPv4 ポリシーベース (VLAN-VLAN) ルーティング

Citrix ADC アプライアンスでは、静的ルーティングの代わりにポリシーベースのルーティングが可能で、ルーティングの決定は通常、宛先 IP ではなく着信インターフェイスや VLAN に対してキーイングされます。ポリシーベースのルーティングは、クライアントの送信元 IP アドレス範囲が定期的に変更される場合や、パケットの宛先 IP アドレスだけではルーティングの決定に到達できない場合の (つまり、クライアント IP アドレスが重複している場合)、必須の考慮事項です。複数の VLAN にまたがる)

例:


```
add ns pbr fromWirelessToInternet ALLOW  -nextHop 192.168.2.1  -vlan
    100  -priority 10

Done

 add ns pbr fromInternetToWireless ALLOW  -nextHop 192.168.1.1  -vlan
     200  -priority 20

Done

apply ns pbrs
<!--NeedCopy-->
```

IPv6 設定

次のコマンドは、VLAN ごとに IPv6 SNIP を割り当てます。次の例では、図: このページの Gi-LAN の簡単な描写に概説されているネットワークに /64 サブネットマスクがあることを前提としています。

コマンド:

```
add ns ip6 fd00:192:168:1::254/64 -vServer DISABLED  -mgmtAccess
    DISABLED
add ns ip6 fd00:192:168:2::254/64 -vServer DISABLED  -mgmtAccess
    DISABLED
bind vlan 100 -IPAddress fd00:192:168:1::254/64
bind vlan 200 -IPAddress fd00:192:168:2::254/64
<!--NeedCopy-->
```

IPv6 ルーティング

IPv6 アドレッシングが完了すると、IPv6 スタティックルーティングが構成されることがあります。

- fd 00:10:: /64 入力ルーター経由のクライアントへのスタティックルート
- 出力ルータを経由したインターネットへのデフォルトルート

例:

```
add route6 fd00:10::/64 fd00:192:168:1::1
add route6 ::/0 fd00:192:168:2::1
<!--NeedCopy-->
```


または、ポリシーベースルーティングを使用します。

例:

```
add ns pbr6 fromWirelessToInternetv6 ALLOW -vlan 100 -priority 10 -
    nextHop fd00:192:168:2::1

add ns pbr6 fromInternetToWirelessv6 ALLOW -vlan 200 -priority 20 -
    nextHop fd00:192:168:1::1

apply ns pbr6
<!--NeedCopy-->
```

LACP の冗長性とフェールオーバー

HA 構成の場合は、スループットオプションを利用して LACP チャネルの下限しきい値を設定することをお勧めします。たとえば、HA ペアの各 Citrix ADC アプライアンスとアップストリームスイッチの間に 25 Gbps Gi-LAN と 4x10GbE チャネルを設定して、N+1 リンクの冗長性を提供するとします。

例:

```
set interface LA/1 -haMonitor ON -throughput 29000
<!--NeedCopy-->
```

プライマリアプライアンスとアップストリームスイッチの間に二重リンク障害が発生した場合、サポートできる最大 Gi-LAN スループットは 20 Gbps に低下します。上記の例で 29 Gbps の下限しきい値を設定すると、セカンダリアプライアンス（同様のリンク障害が発生していない）に冗長スイッチオーバーイベントが発生し、Gi-LAN トラフィックは影響を受けません。

ルートモニター

LACP の冗長性に加えて、ルートモニタチェックを設定し、HA ペアの設定に関連付けることもできます。ルートモニターチェックは、Citrix ADC アプライアンスとネクストホップルーターの間の障害を検出するのに役立ちます。特に、ルーターが直接接続されておらず、アップストリームスイッチを介して接続されている場合に役立ちます。

セクション 2.5.1 のサンプル GiLAN ごとの一般的な HA ルートモニタの設定を次に示します。

```
add route 192.168.1.0 255.255.255.0 192.168.1.1 -msr ENABLED -monitor
    arp
```



```
add route 192.168.2.0 255.255.255.0 192.168.2.1 -msr ENABLED -monitor
    arp
bind HA node -routeMonitor 192.168.1.0 255.255.255.0
bind HA node -routeMonitor 192.168.2.0 255.255.255.0
<!--NeedCopy-->
```

TCP 最適化構成

October 7, 2021

TCP 最適化を構成する前に、Citrix ADC アプライアンスで次の基本構成設定を適用します。

初期設定:

```
enable ns feature LB IPv6PT
enable ns mode FR L3 USIP MBF Edge USNIP PMTUD
disable ns feature SP
disable ns mode TCPB
set lb parameter -preferDirectRoute NO
set lb parameter -vServerSpecificMac ENABLED
set l4param -l2ConnMethod Vlan
set rsskeytype -rsstype SYMMETRIC
set ns param -useproxyport DISABLED
<!--NeedCopy-->
```

> 注
>
> rsskeytype システムパラメーターを変更する場合は、Citrix ADC アプライアンスを再起動します。

TCP 終端処理

Citrix ADC T1 が TCP 最適化を適用するには、まず着信 TCP トラフィックを終了する必要があります。この目的のために、ワイルドカード TCP vserver を作成し、入力トラフィックを代行受信し、それをインターネットルータに転送するように設定する必要があります。

スタティックまたはダイナミックルーティング環境

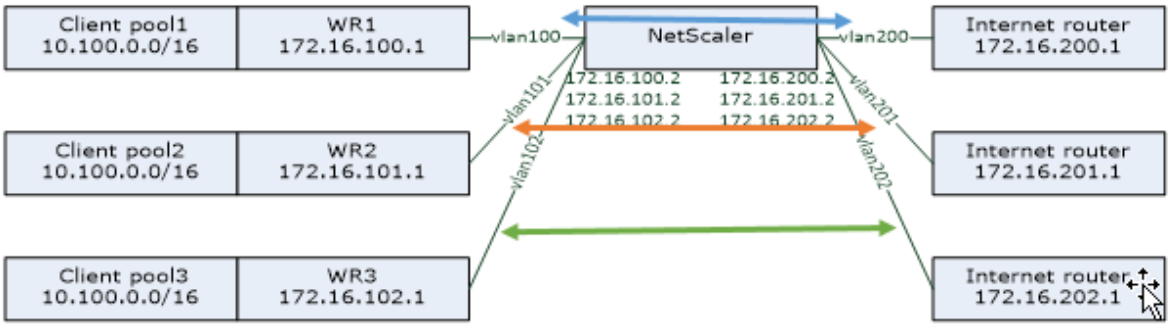
静的または動的ルーティングが設定されている環境では、vserver はルーティングテーブル情報を使用してパケットをインターネットルータに転送できます。デフォルトルートはインターネットルーターを指し、ワイヤレスルーターへのクライアントサブネットのルーティングエントリも配置する必要があります。

例:

```
add lb vserver vsrv-wireless TCP * * -persistenceType NONE -
    Listenpolicy "CLIENT.VLAN.ID.EQ(100) && SYS.VSERVER("vsrv-wireless")
    .STATE.EQ(UP)" -m IP -cltTimeout 9000
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```

VLAN 間（**PBR**）環境

加入者のトラフィックが複数のフローにセグメント化され、着信トラフィックパラメータに基づいて異なるルータに転送する必要があるお客様の環境があります。ポリシーベースルーティング（PBR）を使用すると、VLAN、MAC アドレス、インターフェイス、送信元 IP、送信元ポート、宛先 IP アドレス、宛先ポートなどの着信パケットパラメータに基づいてパケットをルーティングできます。



例:

```
add lb vserver vsrv-wireless TCP * * -m IP -l2Conn ON -listenpolicy "
    CLIENT.VLAN.ID.EQ(100) || CLIENT.VLAN.ID.EQ(101) || CLIENT.VLAN.ID.
    EQ(102)"

add ns pbr pbr-vlan100-to-vlan200 ALLOW -vlan 100 -nexthop 172.16.200.1

add ns pbr pbr-vlan101-to-vlan201 ALLOW -vlan 101 -nexthop 172.16.201.1

add ns pbr pbr-vlan102-to-vlan202 ALLOW -vlan 102 -nexthop 172.16.202.1
<!--NeedCopy-->
```

ポリシーベースルーティングを使用して TCP 最適化トラフィックをルーティングすることは、リリース 11.1 50.10 で追加された新機能です。以前のリリースでは、VLAN ごとに複数の「モード MAC」仮想サーバエンティティを持つ

ことは、マルチ VLAN 環境の代替ソリューションです。各 vserver には、特定のフローのインターネットルーターを表すバインドされたサービスがあります。

例:

```
add server internet_router_1 172.16.200.1

add server internet_router_2 172.16.201.1

add server internet_router_3 172.16.202.1

add service svc-internet-1 internet_router_1 TCP * -usip YES -
   useproxyport NO

add service svc-internet-2 internet_router_2 TCP * -usip YES -
   useproxyport NO

add service svc-internet-3 internet_router_3 TCP * -usip YES -
   useproxyport NO

bind service svc-internet-1 -monitorName arp

bind service svc-internet-2 -monitorName arp

bind service svc-internet-3 -monitorName arp

add lb vserver vsrv-wireless-1 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (100) && SYS.VSERVER("vsrv-wireless-1").STATE.EQ(UP)" -m MAC -l2Conn
    ON

add lb vserver vsrv-wireless-2 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (101) && SYS.VSERVER("vsrv-wireless-2").STATE.EQ(UP)" -m MAC -l2Conn
    ON

add lb vserver vsrv-wireless-3 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (102) && SYS.VSERVER("vsrv-wireless-3").STATE.EQ(UP)" -m MAC -l2Conn
    ON

bind lb vserver vsrv-wireless-1 svc-internet-1

bind lb vserver vsrv-wireless-2 svc-internet-2

bind lb vserver vsrv-wireless-3 svc-internet-3
<!--NeedCopy-->
```


> 注:
>
> vserver モードは MAC であり、以前の例とは対照的に、モード IP です。これは、vserver にバインドされたサービス（複数可）を持っているときに宛先 IP 情報を保持するために必要です。また、追加の PBR 設定では、最適化されていないトラフィックをルーティングする必要があります。

TCP の最適化

すぐに使用できる Citrix ADC TCP 終端は、TCP パススルー機能用に構成されています。TCP パススルーとは、本質的に、Citrix ADC T1 がクライアント/サーバー TCP ストリームを透過的に傍受することがありますが、クライアント/サーバーバッファーを個別に保持したり、最適化手法を適用したりしないことを意味します。

TCP 最適化を有効にするには、nstcpprofile という名前の TCP プロファイルを使用して、TCP 構成を指定します。TCP 構成は、サービスレベルまたは仮想サーバーレベルで提供されていない場合に使用され、次のように変更する必要があります。

コマンド:

```
1 add ns tcpProfile nstcpprofile -WS ENABLED -SACK ENABLED -WSVal 8 -mss
      1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
      bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
      ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
      spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
      fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT
2 <!--NeedCopy-->
```

> 注:
>
> 明示的に作成され、vserver および service にバインドされたプロファイルが存在しない場合、プロファイルの nstcp_default_profile はデフォルトでバインドされます。
>
> 複数の TCP プロファイル要件がある場合は、追加の TCP プロファイルを作成し、適切な仮想サーバに関連付けることができます。

コマンド:

```
1 add ns tcpProfile custom_profile -WS ENABLED -SACK ENABLED -WSVal 8 -
      mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
      bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
      ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
      spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
      fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT
2
```



```
3 set lb vserver vsrv-wireless -tcpProfileName custom_profile
4 <!--NeedCopy-->
```

> 注:
>
> vserver-m MAC および service を使用するデプロイメントでは、同じプロファイルをサービスに関連付ける必要があります。

```
1 set service svc-internet -tcpProfileName custom_profile
2 <!--NeedCopy-->
```

TCP 最適化機能

Citrix ADC アプライアンスの関連する TCP 最適化機能のほとんどは、対応する TCP プロファイルを介して公開されます。TCP プロファイルの作成時に考慮すべき典型的な CLI パラメータは次のとおりです。

1. ウィンドウスケーリング **(WS)**: TCP ウィンドウスケーリングにより、TCP 受信ウィンドウサイズを 65535 バイト以上に増やすことができます。これは、全体的に、特に高帯域幅と長い遅延ネットワークで TCP のパフォーマンスを向上させるのに役立ちます。これは、遅延を低減し、TCP 上の応答時間を向上させるのに役立ちます。
2. 選択的確認応答（**SACK**）: TCP SACK は、全体的なスループット容量を低減する複数のパケット損失の問題に対処します。選択的確認応答で受信機は、唯一の失われたセグメントを再送信する送信者を可能にする、正常に受信されたすべてのセグメントについて送信者に通知することができます。この手法は、T1 が全体的なスループットを改善し、接続待ち時間を短縮するのに役立ちます。
3. ウィンドウスケール係数 **(WSVal)**: 新しいウィンドウサイズの計算に使用される係数。NS によってアドバタイズされたウィンドウがバッファサイズと少なくとも等しくなるようにするには、この値を大きく設定する必要があります。
4. 最大セグメントサイズ（**MSS**）: 単一の TCP セグメントの MSS。この値は、中間ルータおよびエンドクライアントの MTU 設定によって異なります。値 1460 は、1500 の MTU に対応します。
5. **maxBurst**: バーストで許可される TCP セグメントの最大数。
6. 初期輻輳ウィンドウサイズ **(initialCwnd)**: TCP 初期輻輳ウィンドウサイズは、トランザクションの開始時に未処理のバイト数を決定します。これにより、T1 は、ワイヤ上の輻輳を気にすることなく、これらの多くのバイトを送信できます。
7. 最大 **OOO** パケットキューサイズ（**oooQSize**）: TCP は、TCP 通信で OO パケットを維持するために、アウトオブオーダーキューを維持します。この設定は、パケットをランタイムメモリに保持する必要がある限り、キューサイズが長い場合、システムメモリに影響します。したがって、これは、ネットワークおよびアプリケーションの特性の種類に基づいて最適化されたレベルに保つ必要があります。
8. 最小 **RTO (minRTO)**: TCP 再送信タイムアウトは、内部実装ロジックに基づいて、受信した ACK ごとに計算されます。デフォルトの再送信タイムアウトは 1 秒で開始され、この設定で調整できます。これらのパケッ

トの 2 回目の再送信の場合、RTO は N*2 によって計算され、次に N*4... N*8... は、最後の再送信試行まで続きます。

9. **buffersize/sendBuffSize**: これらは、T1 がサーバーから受信できるデータの最大量を参照し、クライアントに送信せずに内部的にバッファします。これらは、基礎となる伝送チャネルの帯域幅遅延積（Bandwidth Delay Product）よりも大きい（少なくとも 2 倍）値に設定する必要があります。
10. フレーバー: これは TCP 輻輳制御アルゴリズムを指します。有効な値は、デフォルト、BIC、CUBIC、Westwood、Nile です。
11. 動的受信バッファリング: 受信バッファは、メモリとネットワークの条件に基づいて動的に調整することができます。後者は TCP プロファイルで指定され、通常は 2* BDP などの基準に基づいて接続のために、サーバーから先読んで、固定サイズのバッファをいっぱいにするのではなく、クライアントのダウンロードパイプをいっぱいに保つために必要なだけバッファをいっぱいにします。Citrix ADC T1 は、クライアントのネットワーク状態を監視し、サーバーから先読みする量を見積もります。
12. **Keep-Alive**（**KA**）: 定期的な TCP キープアライブ（KA）プローブを送信して、ピアがまだアップしているかどうかを確認します。
13. rstWindowAttenuate: なりすまし攻撃から TCP を防御します。シーケンス番号が無効な場合、修正 ACK で応答します。
14. **rstMaxAck**: ウィンドウ外にある RST の受け入れを有効または無効にします。最大の ACK シーケンス番号がエコーされます。
15. **spoofSynDrop**: スプーフィングから保護するために、無効な SYN パケットをドロップします。
16. 明示的な輻輳通知（**ecn**）: これは、データの送信者にネットワークの輻輳状態の通知を送信し、データの輻輳やデータ破損のための是正措置を取ります。
17. フォワード **RTO-Recovery**: スプリアス再送信の場合、輻輳制御の設定は元の状態に戻ります。
18. **TCP** 最大輻輳ウィンドウ（**maxcwnd**）: ユーザが設定可能な TCP 最大輻輳ウィンドウサイズ。
19. **Forward ACK**（**FACK;** 転送確認応答）: ネットワークで未処理のデータバイトの合計数を明示的に測定し、送信者（T1 またはクライアント）が再送信タイムアウト時にネットワークに注入されるデータ量を制御できるようにすることで、TCP 輻輳を回避します。
20. **tcpmode**: 特定のプロファイルの TCP 最適化モード。透過とエンドポイント-2 つの TCP 最適化モードがあります。

    - [エンドポイント] このモードでは、アプライアンスはクライアントとサーバーの接続を別々に管理します。
    - 透明。透過モードでは、クライアントは仮想サーバを介在させることなく、サーバに直接アクセスする必要があります。クライアントはサーバーにアクセスできる必要があるので、サーバー IP アドレスはパブリックにする必要があります。次の図に示す例では、NetScaler アプライアンスがクライアントとサーバーの間に配置されています。そのため、トラフィックはアプライアンスを経由する必要があります。

アイドル状態の接続をサイレントにドロップする

電話会社ネットワークでは、Citrix ADC アプライアンスの TCP 接続のほぼ 50%がアイドル状態になり、アプライアンスは RST パケットを送信してそれらを閉じます。無線チャネルを介して送信されたパケットは、これらのチャネルを不必要にアクティブ化し、メッセージが大量に発生し、その結果、アプライアンスがサービス拒否メ

ッセージを大量に生成します。既定の TCP プロファイルには、ドロップの半分閉じられた接続タイムアウトと DropEstConnOnTimeout パラメーターが含まれるようになりました。これらのパラメーターはデフォルトで無効になっています。両方を有効にした場合、ハーフクローズ接続でも確立された接続でも、接続がタイムアウトしたときに RST パケットがクライアントに送信されることはありません。アプライアンスは接続を切断するだけです。

```
set ns tcpProfile nstcpprofile  -DropHalfClosedConnOnTimeout ENABLED
set ns tcpProfile nstcpprofile  -DropEstConnOnTimeout ENABLED
<!--NeedCopy-->
```

分析とレポート作成

October 7, 2021

TCP 速度レポートは、TCP のダウンロードおよびアップロードのパフォーマンスの尺度として、TCP 接続統計を抽出する Citrix ADC 機能であり、Citrix Application Delivery Management（ADM）の TCP Insight レポートで使用されます 。これを実現するために、Citrix ADC は各 TCP 接続を監視し、アイドルタイムアウトベースでパケットバーストを特定し、識別された最大バーストに関する主要なメトリック（バイト数、再送信されたバイト数、継続時間など）を報告します。TCP 速度レポート機能はデフォルトで有効になっており、TCP および HTTP vServer の両方をサポートし、Appflow/ULFD レポートインフラストラクチャによって異なります。

リアルタイム統計

October 7, 2021

stat コマンドは、TCP 最適化が適切に適用されていることを確認するために使用できます。

コマンド:

```
> stat lb vserver vsrv-wireless
Virtual Server Summary
                     vsrvIP  port     Protocol        State   Health
                        actSvcs
vsrv...eless              *     0          TCP           UP      100
           1

            inactSvcs
vsrv...eless       0
```



```
Virtual Server Statistics
                                                Rate (/s)
                                                   Total
Vserver hits                                           0
    10
Requests                                               0
                        0
Responses                                              0
                        0
Request bytes                                          0
    1580
Response bytes                                         0
    532594360
Total Packets rcvd                                     0
    216463
Total Packets sent                                     0
    369898
Current client connections                            --
                        0
Current Client Est connections                        --
                        0
Current server connections                            --
                        0
Requests in surge queue                               --
                        0
Requests in vserver's surgeQ                          --
                        0
Requests in service's surgeQs                         --
                        0
Spill Over Threshold                                  --
                        0
Spill Over Hits                                       --
                        0
Labeled Connection                                    --
                        0
Push Labeled Connection                               --
                        0
Deferred Request                                       0
                        0
Invalid Request/Response                              --
                        0
Invalid Request/Response Dropped                      --
                        0
Bound Service(s) Summary
                          IP  port         Type        State     Hits
```



```
                                    Hits/s
32 svc-internet     192.168.2.2      0          TCP           UP       10
         0/s
33
34                  Req     Req/s      Rsp    Rsp/s Throughp ClntConn
                       SurgeQ
35 svc-internet       0       0/s        0       0/s         0        0
            0
36                   SvrConn    ReuseP  MaxConn ActvTran  SvrTTFB     Load
37 svc-internet         0         0        0        0         0        0
```

Total カウンタは、運用システムの場合は常に増加します。また、Rate カウンタは 0 以外の値にする必要があります。

> 注
>
> 前述の出力は、動作中でもアイドル状態の実習システムからのものであり、ゼロレートを説明しています。

SNMP

October 7, 2021

SNMP エージェントは、リモートデバイス（SNMP マネージャ）からシステム固有の情報を照会できます。クエリに基づいて、エージェントは、管理情報ベース（MIB）で要求されたデータの等価オブジェクト識別子（OID）を検索し、SNMP マネージャに情報を送信します。次に、Telco の配置で最も有用な SNMP OID を示します。

メモリ

- **resMemUsage (1.3.6.1.4.1.5951.4.1.1.41.2)**

Citrix ADC のメモリ使用率の割合。

パケットエンジン **CPU**

- **resCpuUsage (1.3.6.1.4.1.5951.4.1.1.41.1)**

CPU 使用率の割合。

- **nsCPUTable (1.3.6.1.4.1.5951.4.1.1.41.6)**

この表には、Citrix ADC の各 CPU に関する情報が含まれています。

インデックス作成日:nsCPUname

- **nsCPUname (1.3.6.1.4.1.5951.4.1.1.41.6.1.1)**

CPU の名前。

- **nsCPUusage (1.3.6.1.4.1.5951.4.1.1.41.6.1.2)**

CPU 使用率の割合。

スループット

- **allNicTotRxMbits (1.3.6.1.4.1.5951.4.1.1.71.1)**

Citrix ADC アプライアンスが受信したメガビット数。

- **allNicTotTxMbits (1.3.6.1.4.1.5951.4.1.1.71.2)**

Citrix ADC アプライアンスによって送信されるメガビットの数。

- **ipTotRxPkts (1.3.6.1.4.1.5951.4.1.1.43.25)**

受信した IP パケット。

- **ipTotRxMbits (1.3.6.1.4.1.5951.4.1.1.43.27)**

受信した IP データのメガバイト数。

- **ipTotTxPkts (1.3.6.1.4.1.5951.4.1.1.43.28)**

送信された IP パケット。

- **ipTotTxMbits (1.3.6.1.4.1.5951.4.1.1.43.30)**

転送される IP データのメガバイト数。

接続

アクティブな接続**:**

- **tcpActiveServerConn (1.3.6.1.4.1.5951.4.1.1.46.8)**

現在要求に応答しているサーバーへの接続。

合計接続数**:**

- **tcpCurServerConn (1.3.6.1.4.1.5951.4.1.1.46.1)**

サーバー接続 ([開始]、[確立]、[終了] 状態の接続を含む)。

- **tcpCurClientConn (1.3.6.1.4.1.5951.4.1.1.46.2)**

クライアント接続（「開始」、「確立」、および「終了」状態の接続を含む）。

注**:** SYN-Cookie のため、これにはクライアントが「オープン」状態に含まれません。

- **tcpTotZombieCltConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.26)**

クライアントがしばらくアイドル状態になっているためにフラッシュされるクライアント接続。

- **tcpTotZombieSvrConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.27)**

キューにクライアント要求がないためにフラッシュされるサーバー接続。

エラー

- **tcpErrSynGiveUp (1.3.6.1.4.1.5951.4.1.1.46.37)**

Citrix ADC でタイムアウトした接続の確立を試みます。

- **tcpErrRetransmitGiveUp (1.3.6.1.4.1.5951.4.1.1.46.60)**

その接続でパケットを 7 回再送信した後、Citrix ADC が接続を終了する回数。再送信は、受信側が、パケットを確認応答しない場合に発生します。

- **ifInDiscards (1.3.6.1.2.1.2.2.1.13)**

上位層プロトコルに配信されないようにエラーが検出されなかった場合でも、廃棄するように選択された着信パケットの数。このようなパケットを廃棄する原因の 1 つとして、バッファ領域を解放することが考えられます。

- **ifOutDiscards (1.3.6.1.2.1.2.2.1.19)**

送信を防止するためにエラーが検出されなかったにもかかわらず、廃棄するように選択された発信パケットの数。このようなパケットを廃棄する原因の 1 つとして、バッファ領域を解放することが考えられます。

- **ifErrTxOverflow (1.3.6.1.4.1.5951.4.1.1.54.1.36)**

Citrix ADC アプライアンスが起動されてから、またはインターフェイス統計情報がクリアされてから、指定されたインターフェイスでの送信中にオーバーフローキューを通過したパケット数。これは、輻輳しているポートでのみ増加します。

最適化された/バイパス接続

- **tcpOptimizationEnabled (1.3.6.1.4.1.5951.4.1.1.46.131)**

TCP 最適化で有効にされた接続の総数。

- **tcpOptimizationBypassed (1.3.6.1.4.1.5951.4.1.1.46.132)**

TCP 最適化をバイパスした接続の総数。

技術レシピ

October 7, 2021

Citrix ADC T1 モデルは、高度な機能と強力なポリシー構成言語を提供し、実行時に複雑な決定の評価を可能にします。

T1000 機能およびポリシー構成ガイドでロック解除される可能性のあるすべての機能を評価することはできませんが、技術受領書では、通信事業者によってもたらされるさまざまな要件の実装を検討します。「レシピ」をそのまま再利用するか、環境に適応してください。

ユーザーごとの接続制限

Citrix ADC T1 モデルは、一意のサブスクライバ IP あたりの接続数を制限するように設定できます。以下の設定では、IP あたりの N 個の同時接続（CLIENT.IP.SRC）が許可されます。設定されたしきい値を超える接続を試行するたびに、T1 は RST を送信します。ユーザーあたり最大 2 つの同時接続の場合:

コマンド:

```
add stream selector streamSel_usrlimit CLIENT.IP.SRC
add ns limitIdentifier limitId_usrlimit -threshold 2 -mode CONNECTION -
    selectorName streamSel_usrlimit
add responder policy respPol_usrlimit "SYS.CHECK_LIMIT("
    limitId_usrlimit")" RESET
bind lb vserver vsrv-wireless -policyName respPol_usrlimit -priority 1
    -gotoPriorityExpression END
<!--NeedCopy-->
```

仮想サーバのスムーズな挿入/削除

多くのオペレータは、Citrix ADC T1 モデルが TCP 最適化のためにインラインでアクティブ化されている場合、またはメンテナンス目的で無効化されている場合に、TCP 接続の中断について懸念しています。vserver の導入時に既存の接続が切断されないようにするには、TCP 最適化のために vserver を設定またはアクティブにする前に、次の設定を適用する必要があります。

コマンド:

```
add ns acl acl-ingress ALLOW  - vlan 100
add forwardingSession fwd-ingress  - aclname acl-ingress
apply ns acls
<!--NeedCopy-->
```

転送セッションは、ルーティング（スタティック、ダイナミック、または PBR）上で有効であり、ルーティングされるトラフィックのセッションエントリを作成します（L3 モード）。既存の接続は、対応するセッションのために転送セッションによって処理され、vserver が導入されると、新しい TCP 接続のみのキャプチャが開始されます。

ACL は、メモリを消費する不要なトラフィックのセッションを作成しないように、vserver などの特定のポートだけをキャプチャするように設定できます。別のオプションは、vserver のアクティベーション後に特定の設定を削除することです。

メンテナンスのために、vserver を無効にし、その状態は OUT OF SERVICE と表示されます。この場合、vserver はデフォルトですべての接続をただちに終了します。vserver が既存の接続を処理し、新しい接続を受け入れないようにするには、次の設定を適用する必要があります。

コマンド:

```
set lb vserver vsrv-wireless  -downStateFlush DISABLED
<!--NeedCopy-->
```

新しい接続はルーティングテーブルを通過し、転送セッションのために対応するセッションエントリが作成されます。

ポリシーベースの **TCP** プロファイリング

ポリシーベースの TCP プロファイル選択により、オペレータは、異なるトラフィックドメイン（3G または 4G）から着信するクライアントに対して TCP プロファイルを動的に設定できます。一部の QoS メトリックは、これらのトラフィックドメインで異なります。パフォーマンスを向上させるには、TCP パラメータの一部を動的に変更する必要があります。3G と 4G からのクライアントが同じ vserver にヒットし、同じ TCP プロファイルを使用する場合を考えてみましょう。これは、クライアントのパフォーマンスに悪影響を及ぼします。AppQoE 機能は、これらのクライアントを分類し、vserver 上の TCP プロファイルを動的に変更することができます。

例:

```
enable feature AppQoE

add ns tcpProfile nstcpprofile1 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 4000000 -flavor BIC -KA ENABLED -
    sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -tcpmode
    ENDPOINT

add ns tcpProfile nstcpprofile2 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 15 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 128000 -flavor BIC -KA ENABLED -
    sendBuffsize 6000000 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -frto ENABLED -maxcwnd 64000 -fack ENABLED -tcpmode ENDPOINT

add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1
```



```
8
9  add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2
10
11 add appqoe policy appqoe_4G -rule ”CLIENT.VLAN.ID.EQ(100)” -action
       action_1
12
13 add appqoe policy appqoe_3G -rule ”CLIENT.VLAN.ID.EQ(200)” -action
       action_2
14
15 bind lb vserver vsrv-wireless -policyName appqoe_4G -priority 100
16
17 bind lb vserver vsrv-wireless -policyName appqoe_3G -priority 110
18 <!--NeedCopy-->
```

Citrix ADC T1 モデルは、Gx または Radius または Radius と Gx インターフェイスを介して加入者情報を動的に受信し、加入者ごとに異なる TCP プロファイルを適用することができます。

コマンド:

```
1  add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1
2
3  add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2
4
5  add appqoe policy appqoe_4G -rule ”SUBSCRIBER.RULE_ACTIVE(”3G”)” -
       action action_1
6
7  add appqoe policy appqoe_3G -rule ”SUBSCRIBER.RULE_ACTIVE(”4G”)” -
       action action_2
8  <!--NeedCopy-->
```

Citrix ADC T1 モデルとオペレータコントロールプレーンネットワークとの統合については、Telco サブスクライバ管理を参照してください。

スケーラビリティ

October 7, 2021

TCP の最適化はリソースを大量に消費するため、単一の Citrix ADC アプライアンス（ハイエンドアプライアンス）であっても、高い GiLAN スループットを維持できない場合があります。ネットワークの容量を拡張するには、Citrix ADC アプライアンスを N+1 クラスタ構成で展開します。クラスタ展開では、Citrix ADC アプライアンスは単一のシ

ステムイメージとして連携して動作します。クライアントトラフィックは、外部スイッチデバイスの助けを借りて、クラスタノード間で分散されます。

トポロジ

図 1 は、4 つの T1300-40G ノードで構成されるクラスタの例です。

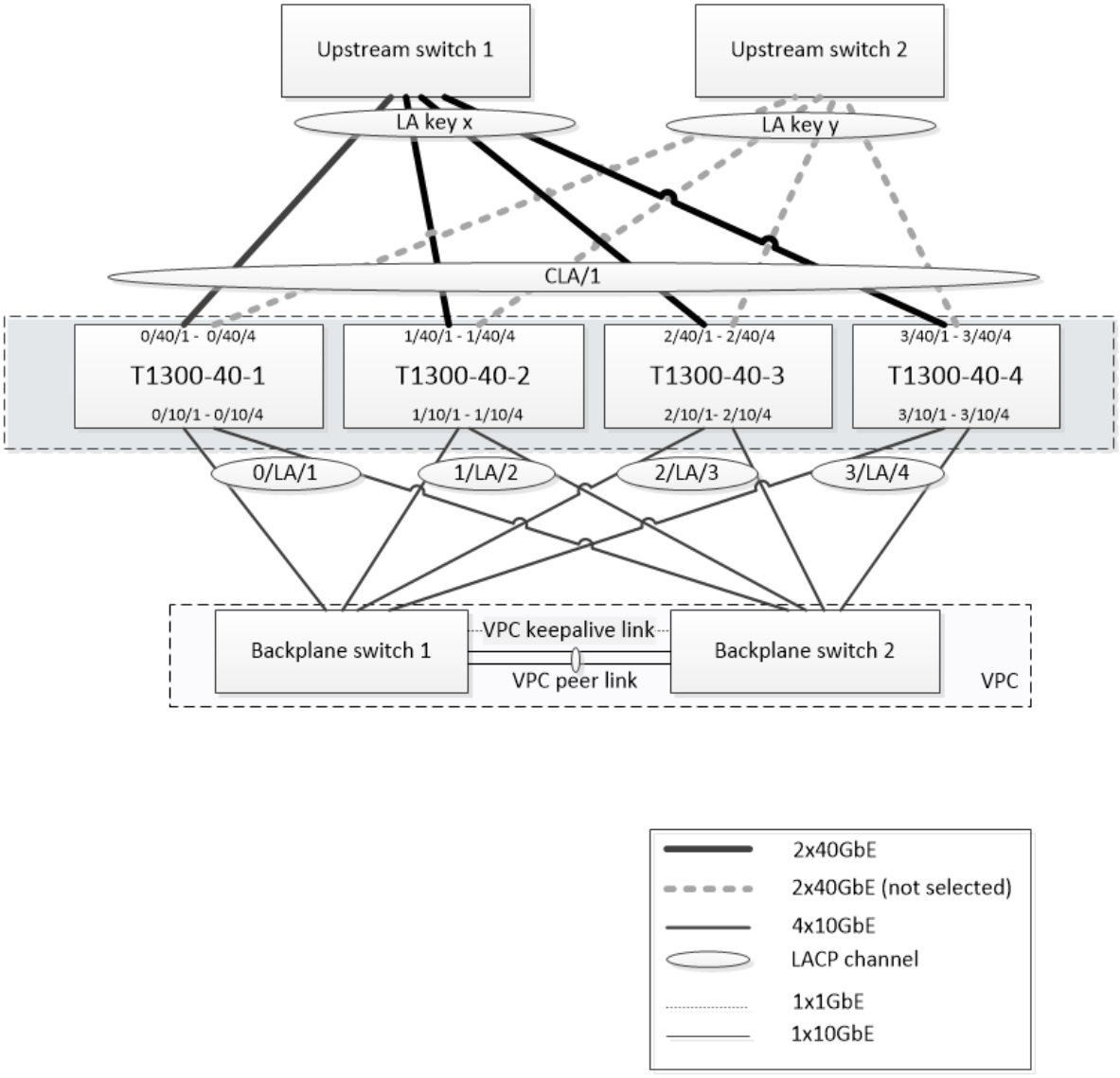


図 1 に示すセットアップには、次のプロパティがあります。

1. すべてのクラスターノードは、同じネットワーク (L2 クラスターとも呼ばれます) に属します。
2. データプレーンとバックプレーントラフィックは、異なるスイッチによって処理されます。
3. GiLAN のスループットが 200 Gbps で、T1300-40G アプライアンスが 80 Gbps のスループットを維持できると仮定すると、3 つの T1300-40G アプライアンスが必要です。単一クラスタノードで障害が発生した場合に冗長性を確保するために、合計で 4 つのアプライアンスを導入しています。

4. 各ノードは最大 67 Gbps のトラフィック（通常の動作条件では 50 Gbps、単一クラスタノードに障害が発生した場合は 67 Gbps）を受信するため、アップストリームスイッチへの 2 x 40 Gbps の接続が必要です。スイッチに障害が発生した場合に冗長性を確保するために、2 つのアップストリームスイッチを配置し、接続数を 2 倍にします。
5. クラスタリンクアグリゲーション（CLAG）は、クラスタノード間でトラフィックを分散するために使用されます。1 つの CLAG は、クライアントトラフィックとサーバトラフィックの両方を処理します。CLAG でリンク冗長性が有効になっているため、一度に 1 つの「サブチャネル」だけが選択され、トラフィックが処理されます。一部のリンクに障害が発生した場合、またはスループットが指定されたしきい値を下回った場合、もう一方のサブチャネルが選択されます。
6. アップストリームスイッチは、対称ポートチャネルロードバランシング（たとえば、Cisco IOS 7.0 (8) N1 (1) の source-dest-ip-only アルゴリズム）を実行して、転送および逆方向のトラフィックフローが同じクラスタノードで処理されるようにします。このプロパティは、パケットの並べ替えがなくなり、TCP のパフォーマンスが低下するため、望ましいです。
7. データトラフィックの 50% がバックプレーンに操縦されることが予想されます。つまり、各ノードは他のクラスタノードまで最大 34 Gbps（通常の動作条件では 25 Gbps、単一クラスタノードで障害が発生した場合は 34 Gbps）です。したがって、各ノードには、バックプレーンスイッチへの少なくとも 4x10G 接続が必要です。スイッチに障害が発生した場合に冗長性を確保するために、バックプレーンスイッチをいくつか導入し、接続数を 2 倍にします。リンク冗長性は現在、バックプレーンではサポートされていないため、スイッチレベルの冗長性を実現するには Cisco VPC または同等のテクノロジーが必要です。
8. ステアドパケットの MTU サイズは 1578 バイトであるため、バックプレーンスイッチは 1500 バイトを超える MTU をサポートする必要があります。

注：図 1 に示す設計は、T1120 および T1310 アプライアンスにも適用可能です。T1310 の場合、我々はバックプレーン接続に 40GbE インターフェイスを使用します, それは 10GbE ポートを欠いているので、.

注：このドキュメントでは例として Cisco VPC を使用していますが、他社製スイッチを使用する場合は、Juniper の MLAG など、代替の同等のソリューションを使用できます。

注**:** CLAG ではなく ECMP などの他のトポロジーも可能ですが、この特定のユースケースでは現在サポートされていません。

Citrix ADC T1000 クラスタでの **TCP** 最適化の構成

物理的なインストール、物理的な接続、ソフトウェアのインストール、およびライセンスが完了したら、実際のクラスタ構成を続行できます。以下に説明する構成は、図 1 に示すクラスタに適用されます。

注：クラスター構成の詳細については、「Citrix ADC クラスターの設定」を参照してください。

図 1 の 4 つの T1300 ノードには、次の NSIP アドレスがあるとします。

NSIP アドレスを持つ **4** つの **T1300** ノード：

```
1 T1300-40-1: 10.102.29.60
```



```
T1300-40-2: 10.102.29.70
T1300-40-3: 10.102.29.80
T1300-40-4: 10.102.29.90
```

クラスタは、クラスタ IP (CLIP) アドレスを使用して管理されます。このアドレスは 10.78.16.61 であると想定されます。

クラスタのセットアップ

図 1 に示すクラスタの構成を開始するには、クラスタに追加する最初のアプライアンス (T1300-40-1 など) にログオンし、次の操作を行います。

1. コマンドプロンプトで、次のコマンドを入力します。

   コマンド:

```
> add cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE
> add ns ip 10.102.29.61 255.255.255.255 -type clip
> enable cluster instance 1
> save ns config
> reboot  - warm
```

2. アプライアンスが再起動したら、クラスタ IP（CLIP）アドレスに接続し、残りのノードをクラスタに追加します。

   コマンド:

```
> add cluster node 1 10.102.29.70 -state ACTIVE
> add cluster node 2 10.102.29.80 -state ACTIVE
> add cluster node 3 10.102.29.90  - state ACTIVE
> save ns config
```

3. 新しく追加された各ノードの NSIP アドレスに接続し、クラスタに参加します。

   コマンド:

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot  - warm
```


4. ノードが再起動したら、バックプレーン構成に進みます。クラスタ IP アドレスで次のコマンドを入力して、各クラスタノードのバックプレーンリンク用の LACP チャネルを作成します。

   コマンド:

```
> set interface 0/10/[1-8]  -lacpkey 1  -lacpmode ACTIVE
> set interface 1/10/[1-8]  -lacpkey 2  -lacpmode ACTIVE
> set interface 2/10/[1-8]  -lacpkey 3  -lacpmode ACTIVE
> set interface 3/10/[1-8]  -lacpkey 4  -lacpmode ACTIVE
```

5. 同様に、バックプレーンスイッチでダイナミック LA と VPC を設定します。バックプレーンスイッチインターフェイスの MTU が 1578 バイト以上であることを確認します。
6. チャネルが動作していることを確認します。

   コマンド:

```
> show channel 0/LA/1
> show channel 1/LA/2
> show channel 2/LA/3
> show channel 3/LA/4
```

7. クラスタノードバックプレーンインターフェイスを設定します。

   コマンド:

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/LA/3
> set cluster node 3  -backplane 3/LA/4
```

8. クラスタのステータスを確認し、クラスタが動作していることを確認します。

```
> show cluster instance
> show cluster node
```

クラスターの設定の詳細については、「Citrix ADC クラスターの設定」を参照してください。

クラスタノード間でのトラフィックの分散

Citrix ADC クラスターを形成したら、クラスタリンクアグリゲーション (CLAG) を展開して、クラスタノード間でトラフィックを分散します。単一の CLAG リンクは、クライアントとサーバの両方のトラフィックを処理します。

クラスタ IP アドレスで次のコマンドを実行して、図 1 に示すクラスタリンク集約 (CLAG) グループを作成します。

コマンド:

```
> set interface 0/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 3/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
```

外部スイッチでダイナミックリンク集約を設定します。

次に、次のようにリンク冗長性を有効にします。

コード**:**

```
> set channel CLA/1 -linkRedundancy ON -lrMinThroughput 240000
```

最後に、次のように入力してチャネルのステータスを確認します。

コマンド:

```
> show channel CLA/1
```

チャネルは UP で、実際のスループットは 320000 である必要があります。

クラスタリンク集約の詳細については、次のトピックを参照してください。

- 動的クラスタリンク集約
- LACP を使用したクラスタ内のリンク冗長性。

MAC ベースフォワーディング（MBF）を使用するため、次のようにリンクセットを設定し、CLAG グループにバインドします。

コマンド:

```
> add linkset LS/1
> bind linkset LS/1 -ifnum CLA/1
```


リンクセットの詳細については、以下のトピックを参照してください。

- リンクセットの設定
- リンクセットでのクラスター LA チャネルの使用

VLAN および IP アドレスの設定

ストライプ IP 構成を使用します。これは、IP アドレスがすべてのノードでアクティブであることを意味します（デフォルト設定）。このトピックの詳細については、「ストライプ、部分的にストライプ、およびスポッティングされた構成」を参照してください。

1. 入力および出力 SNIP を追加します。

   コマンド：

```
> add ns ip 172.16.30.254 255.255.255.0  - type SNIP
> add ns ip 172.16.31.254 255.255.255.0  - type SNIP
> add ns ip6 fd00:172:16:30::254/112  - type SNIP
> add ns ip6 fd00:172:16:31::254/112  - type SNIP
```

2. 対応する入力 VLAN と出力 VLAN を追加します。

   コマンド：

```
> add vlan 30 -aliasName wireless
> add vlan 31 -aliasName internet
```

3. IP およびリンクセットを使用して VLAN をバインドします。

   コマンド：

```
> bind vlan 31 -ifnum LS/1 -tagged
> bind vlan 30 -ifnum LS/1 -tagged
> bind vlan 30 -IPAddress 172.16.30.254 255.255.255.0
> bind vlan 31 -IPAddress 172.16.31.254 255.255.255.0
> bind vlan 30 -IPAddress fd00:172:16:30::254/112
> bind vlan 31 -IPAddress fd00:172:16:31::254/112
```

必要に応じて、さらに多くの入力 VLAN と出力 VLAN を追加できます。

TCP 最適化の設定

この時点で、クラスタ固有のコマンドをすべて適用しました。設定を完了するには、TCP 最適化の設定で説明されている手順に従います。

ダイナミックルーティングの設定

Citrix ADC クラスタは、お客様のネットワークの動的ルーティング環境に統合できます。BGP ルーティングプロトコルを使用したダイナミックルーティング設定の例を次に示します（OSPF もサポートされています）。

1. CLIP アドレスから、入力 IP アドレスおよび出力 IP アドレスで BGP およびダイナミックルーティングを有効にします。

   コマンド：

```
> enable ns feature bgp
> set ns ip 172.16.30.254  -dynamicRouting ENABLED
> set ns ip 172.16.31.254  -dynamicRouting ENABLED
```

2. vtysh を開き、出力側に BGP を設定します。

   コード**:**

```
> shell
root@ns# vtysh
ns# configure terminal
ns(config)# router bgp 65531
ns(config-router)# network 10.0.0.0/24
ns(config-router)# neighbor 172.16.31.100 remote-as 65530
ns(config-router)# neighbor 172.16.31.100 update-source
    172.16.31.254
ns(config-router)# exit
ns(config)# ns route-install propagate
ns(config)# ns route-install default
ns(config)# ns route-install bgp
ns(config)# exit
```

3. デフォルトルートを Citrix ADC クラスタにアドバタイズするように、出力側 BGP ピアを設定します。次に例を示します：

   コマンド：


```
router bgp 65530
 bgp router-id 172.16.31.100
 network 0.0.0.0/0
 neighbor 172.16.31.254 remote-as 65531
```

4. 入力側を設定するには、同様の手順に従います。
5. vtysh から、次のように入力して、構成がすべてのクラスタ・ノードに伝播されていることを確認します。

   コマンド：

```
ns# show running-config
```

6. 最後に、各クラスタノードの NSIP アドレスにログオンし、BGP ピアからアドバタイズされたルートを確認します。

   コマンド：

```
> show route | grep BGP
```

TCP Nile を使用した TCP パフォーマンスの最適化

October 7, 2021

TCP は、次の最適化手法と輻輳制御戦略（またはアルゴリズム）を使用して、データ伝送におけるネットワークの輻輳を回避します。

輻輳制御方針

Transmission Control Protocol (TCP) は、インターネット接続の確立と管理、送信エラーの処理、およびクライアントデバイスとの Web アプリケーションのスムーズな接続のために長い間使用されてきました。しかし、パケット損失はネットワークの輻輳だけに依存せず、輻輳が必ずしもパケット損失を引き起こすわけではないため、ネットワークトラフィックの制御が困難になっています。したがって、輻輳を測定するには、TCP アルゴリズムはパケット損失と帯域幅の両方に焦点を当てる必要があります。

NILE アルゴリズム

Citrix システムズは、新しい輻輳制御アルゴリズム、NILE、LTE、LTE などの高速ネットワーク用に設計された TCP 最適化アルゴリズムを開発しました。Nile は、フェーディング、ランダムまたは輻輳損失、リンク層の再送信、およ

びキャリア集約によって生じる固有の課題に対処します。

NILE アルゴリズム:

- ラウンドトリップ時間測定に基づいてキュー遅延推定を行います。
- 測定されたキュー遅延に反比例する輻輳ウィンドウ増加関数を使用します。この方法では、標準の TCP 方式よりもネットワーク輻輳ポイントに近づくのが遅くなり、輻輳時のパケット損失が減少します。
- 推定キュー遅延を使用して、ネットワーク上のランダム損失と輻輳ベースの損失を区別できます。

通信サービスプロバイダーは、TCP インフラストラクチャで NILE アルゴリズムを使用して、次の操作を実行できます。

- モバイルおよび長距離ネットワークの最適化：NILE アルゴリズムは、標準の TCP よりも高いスループットを達成します。この機能は、モバイルネットワークや長距離ネットワークで特に重要です。
- アプリケーション認識遅延を低減し、加入者エクスペリエンスを向上させる：Nile アルゴリズムは、パケット損失情報を使用して送信ウィンドウサイズを増減するかどうかを決定し、キューイング遅延情報を使用して増分または減少のサイズを決定します。この送信ウィンドウサイズの動的な設定により、ネットワーク上のアプリケーション遅延が減少します。

コマンドラインインターフェイスを使用して **NILE** サポートを設定するには

コマンドプロンプトで、次のように入力します。

```
set ns tcpProfile <name> [-flavor NILE]
<!--NeedCopy-->
```

設定ユーティリティを使用した **NILE** サポートの設定

1. [システム] > [プロファイル] > [**TCP** プロファイル] に移動し、[**TCP** プロファイル] をクリックします。
2. [**TCP** フレーバー] ドロップダウンリストから、[**NILE**] を選択します。

例**:**

```
set ns tcpProfile tcpprofile1 -flavor NILE
<!--NeedCopy-->
```

比例レート回復（**PRR**）アルゴリズム

TCP 高速回復メカニズムは、パケット損失に起因する Web 遅延を低減します。新しい比例速度回復（PRR）アルゴリズムは、損失回復中に TCP データを評価する高速回復アルゴリズムです。輻輳制御アルゴリズムによって選択され

たターゲットウィンドウに適した分数を使用して、Rate-Halving の後にパターン化されます。これにより、ウィンドウの調整が最小限に抑えられ、リカバリ終了時の実際のウィンドウ・サイズは Slow-Start しきい値（ssthresh）に近い値になります。

TCP ファスト・オープン（**TFO**）

TCP ファストオープン (TFO) は、TCP の最初のハンドシェイク中に、クライアントとサーバー間で迅速かつ安全なデータ交換を可能にする TCP メカニズムです。この機能は、Citrix ADC アプライアンスの仮想サーバーにバインドされた TCP プロファイルで TCP オプションとして使用できます。TFO は、Citrix ADC アプライアンスが生成する TCP 高速オープンクッキー（セキュリティクッキー）を使用して、仮想サーバーへの TFO 接続を開始するクライアントを検証および認証します。TFO メカニズムを使用すると、1 回のフルラウンドトリップに必要な時間だけアプリケーションのネットワーク遅延を減らすことができます。これにより、短い TCP 転送で発生する遅延が大幅に削減されます。

TFO の仕組み

クライアントが TFO 接続を確立しようとすると、それ自体を認証する最初の SYN セグメントで TCP ファストオープン Cookie が含まれます。認証が成功すると、Citrix ADC アプライアンスの仮想サーバーがスリーウェイハンドシェイクの最終 ACK セグメントを受信していなくても、SYN-ACK セグメントにデータを含めることができます。これにより、通常の TCP 接続と比較して最大 1 回のラウンドトリップが節約されます。この場合、データを交換する前に 3 ウェイハンドシェイクが必要になります。

クライアントとバックエンドサーバーは、次の手順を実行して TFO 接続を確立し、初期 TCP ハンドシェイク中にデータを安全に交換します。

1. クライアントが自身を認証するための TCP ファスト・オープン Cookie を持っていない場合、SYN パケット内のファスト・オープン Cookie 要求を Citrix ADC アプライアンス上の仮想サーバーに送信します。
2. 仮想サーバーにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、アプライアンスはクッキーを生成し（シークレットキーでクライアントの IP アドレスを暗号化して）、生成された高速オープン Cookie を TCP オプションフィールドに含む SYN-ACK でクライアントに応答します。
3. クライアントは、アプライアンス上の同じ仮想サーバーへの将来の TFO 接続用に Cookie をキャッシュします。
4. クライアントは、同じ仮想サーバーへの TFO 接続を確立しようとすると、HTTP データと共に (TCP オプションとして) キャッシュされた高速オープン Cookie を含む SYN を送信します。
5. Citrix ADC アプライアンスはクッキーを検証し、認証に成功すると、サーバーは SYN パケット内のデータを受け入れ、SYN-ACK、TFO クッキー、HTTP レスポンスでイベントを確認します。

注：クライアント認証が失敗した場合、サーバーはデータを削除し、セッションのタイムアウトを示す SYN のみでイベントを確認します。

1. サーバー側では、サービスにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、Citrix ADC アプライアンスは、接続しようとしているサービスに TCP 高速オープン Cookie が存在するかどうかを決定します。

2. TCP ファスト・オープン・クッキーが存在しない場合、アプライアンスは SYN パケットでクッキー要求を送信します。
3. バックエンドサーバーが Cookie を送信すると、アプライアンスは Cookie をサーバー情報キャッシュに保存します。
4. アプライアンスが所定の宛先 IP ペアのクッキーをすでに持っている場合、古いクッキーを新しいものと置き換えます。
5. 仮想サーバが同じ SNIP アドレスを使用して同じバックエンドサーバに再接続しようとしたときに、サーバ情報キャッシュで cookie が使用可能な場合、アプライアンスは SYN パケット内のデータを cookie と結合し、バックエンドサーバに送信します。
6. バックエンドサーバーは、データと SYN の両方でイベントを確認します。

注：サーバーが SYN セグメントのみでイベントを確認した場合、Citrix ADC アプライアンスは、元のパケットから SYN セグメントと TCP オプションを削除した後、直ちにデータパケットを再送信します。

TCP ファストオープンの設定

TCP ファストオープン（TFO）機能を使用するには、関連する TCP プロファイルで TCP ファストオープンオプションを有効にし、TFO Cookie タイムアウトパラメータにそのプロファイルのセキュリティ要件に適した値を設定します。

コマンドラインを使用して **TFO** を有効または無効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存のプロファイルで TFO を有効または無効にします。

注**:** デフォルト値は DISABLED です。

```
add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
unset tcpprofile <TCP Profile Name> - tcpFastOpen
<!--NeedCopy-->
```

例：

add tcpprofile Profile1 – tcpFastOpen
Set tcpprofile Profile1 – tcpFastOpen Enabled
unset tcpprofile Profile1 – tcpFastOpen

コマンドラインインターフェイスを使用して **TCP** ファストオープン **cookie** タイムアウト値を設定するには

コマンドプロンプトで入力します。


```
set tcpparam  - tcpfastOpenCookieTimeout <Timeout Value>
<!--NeedCopy-->
```

例**:**

```
set tcpprofile  - tcpfastOpenCookieTimeout 30secs
<!--NeedCopy-->
```

GUI を使用して **TCP** ファストオープンを構成するには

1. [設定] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. [**TCP** プロファイルの設定] ページで、[**TCP** 高速オープン] チェックボックスを選択します。
3. [**OK**]、[完了] の順にクリックします。

GUI を使用して **TCP** ファストクッキーのタイムアウト値を設定するには

[設定**]** > [システム**] > [**設定**] > [TCP** パラメータの変更] に移動し、**[TCP** パラメータの構成**]** ページに移動して、TCP ファストオープン Cookie タイムアウト値を設定します。

TCP Hystart

新しい TCP プロファイルパラメータ hystart を使用すると、ハイスタートアルゴリズムが有効になります。ハイスタートアルゴリズムは、終了するセーフポイント（ssthresh）を動的に決定するスロースタートアルゴリズムです。これにより、大量のパケット損失を伴わずに、輻輳回避への移行が可能になります。この新しいパラメータは、デフォルトでは無効になっています。

輻輳が検出されると、Hystart は輻輳回避フェーズに入ります。これを有効にすると、パケット損失が大きい高速ネットワークのスループットが向上します。このアルゴリズムは、トランザクションの処理中に最大帯域幅に近い状態を維持するのに役立ちます。したがって、スループットを向上させることができます。

TCP ハイスタートの設定

ハイスタート機能を使用するには、関連する TCP プロファイルで [キュービックハイスタート] オプションを有効にします。

コマンドラインインターフェイス（**CLI**）を使用して **Hystart** を構成するには

コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存の TCP プロファイルで Hystart を有効または無効にします。


```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

例:

```
add tcpprofile Profile1 - tcpFastOpen
Set tcpprofile Profile1 - tcpFastOpen Enabled
unset tcpprofile Profile1 - tcpFastOpen
<!--NeedCopy-->
```

GUI を使用して **Hystart** サポートを構成するには

1. [設定] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. [**TCP** プロファイルの設定] ページで、[キュービックハイスタート] チェックボックスをオンにします。
3. [**OK**]、[完了] の順にクリックします。

最適化テクニック

TCP は、フロー制御を最適化するために、次の最適化手法と方法を使用します。

ポリシーベースの **TCP** プロファイルの選択

今日のネットワークトラフィックは、かつてないほど多様で帯域幅が集中しています。トラフィックが増加すると、サービス品質（QoS）が TCP パフォーマンスに与える影響が大きくなります。QoS を強化するために、ネットワークトラフィックのクラスごとに異なる TCP プロファイルを使用して AppQoE ポリシーを構成できるようになりました。AppQoE ポリシーは、仮想サーバーのトラフィックを分類して、3G、4G、LAN、WAN などの特定のタイプのトラフィックに最適化された TCP プロファイルを関連付けます。

この機能を使用するには、TCP プロファイルごとにポリシーアクションを作成し、AppQoE ポリシーにアクションを関連付け、負荷分散仮想サーバーにポリシーをバインドします。

ポリシーベースの **TCP** プロファイル選択の設定

ポリシーベースの TCP プロファイル選択の設定は、次の作業で構成されます。

- AppQoE を有効にします。TCP プロファイル機能を設定する前に、AppQoE 機能を有効にする必要があります。

- AppQoE アクションを追加しています。AppQoE 機能を有効にした後、TCP プロファイルを使用して AppQoE アクションを設定します。
- AppQoE ベースの TCP プロファイル選択の設定。異なるクラスのトラフィックに対して TCP プロファイル選択を実装するには、Citrix ADC アプライアンスが接続を区別し、正しい AppQoE アクションを各ポリシーにバインドできる AppQoE ポリシーを構成する必要があります。
- 仮想サーバーへの AppQoE ポリシーのバインド。AppQoE ポリシーを構成したら、1 つまたは複数の負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーにポリシーをバインドする必要があります。

コマンドラインインターフェイスを使用した構成

コマンドラインインターフェイスを使用して AppQoE を有効にするには:

コマンドプロンプトで次のコマンドを入力して、機能を有効にし、有効になっていることを確認します。

```
enable ns feature appqoe

show ns feature
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **AppQoE** アクションの作成中に **TCP** プロファイルをバインドするには

コマンドプロンプトで、tcpprofiletobind オプションを指定して、次の AppQoE アクションコマンドを入力します。

TCP プロファイルのバインド:

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS |
    NS ) [<CustomFile>] [-altContentSvcName <string>] [-altContentPath <
    string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth
     <positive_integer>] [-priqDepth <positive_integer>] [-
    dosTrigExpression <expression>] [-dosAction ( SimpleResponse |
    HICResponse )] [-tcpprofiletobind <string>]

show appqoe action
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **AppQoE** ポリシーを構成するには

コマンドプロンプトで入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **AppQoE** ポリシーを負荷分散、キャッシュリダイレクト、コンテンツスイッチング仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <
    priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <
    priority>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <
    priority>
<!--NeedCopy-->
```

例:

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
    ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500 -
    slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
    sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack ENABLED
    -tcpmode ENDPOINT

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule ”client.ip.src.eq(10.102.71.31)” -
    action appact1

bind lb vserver lb2 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST

bind cs vserver cs1 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```


GUI を使用したポリシーベースの **TCP** プロファイリングの設定

GUI を使用して AppQoE を有効にするには

1. [システム] > [設定**]** に移動します。
2. 詳細ペインで、[高度な機能の構成] をクリックします。
3. [高度な機能の構成] ダイアログボックスで、[**AppQoE**] チェックボックスをオンにします。
4. [**OK**] をクリックします。

GUI を使用して **AppQoE** ポリシーを構成するには

1.「アプリ エキスパート」>「**AppQoE**」**>**「アクション」に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
3. 新しいアクションを作成するには、[追加] をクリックします。
4. 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
5.「**AppQoE** アクションの作成」または「**AppQoE** アクションの設定」画面で、パラメーターの値を入力または選択します。ダイアログ・ボックスの内容は、「AppQoE アクションを構成するためのパラメータ」で説明されているパラメータに対応しています（アスタリスクは必須パラメータを示します）。
   a) 名前—name
   b) アクション・タイプ-respondWith
   c) プライオリティ—priority
   d) ポリシーキューの深さ：polqDepth
   e) キューの深さ：priqDepth
   f) DOS アクション：dosAction
6. ［作成］をクリックします。

GUI を使用して **AppQoE** ポリシーをバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択して [編集] をクリックします。
2. [ポリシー] セクションで、[+] をクリックして AppQoE ポリシーをバインドします。
3. [ポリシー] スライダで、次の操作を行います。
   a) ドロップダウンリストから [AppQoE] としてポリシータイプを選択します。
   b) ドロップダウンリストからトラフィックタイプを選択します。
4. [ポリシーのバインド] セクションで、次の操作を行います。
   a) [新規] をクリックして、新しい AppQoE ポリシーを作成します。
   b) [既存のポリシー] をクリックして、ドロップダウンリストから AppQoE ポリシーを選択します。
5. バインドの優先順位を設定し、[仮想サーバにポリシーに バインド] をクリックします。
6. ［完了］をクリックします。

SACK ブロック生成

1 つのデータウィンドウで複数のパケットが失われると、TCP のパフォーマンスが低下します。このようなシナリオでは、選択的確認応答（SACK）メカニズムと選択的繰り返し再送信ポリシーを組み合わせると、この制限が克服されます。着信順不同パケットごとに、SACK ブロックを生成する必要があります。

順序外パケットが再構成キューブロックに収まる場合は、ブロックにパケット情報を挿入し、完全なブロック情報を SACK-0 に設定します。順序外パケットが再構成ブロックに収まらない場合は、パケットを SACK-0 として送信し、以前の SACK ブロックを繰り返します。順序外パケットが重複していて、パケット情報が SACK-0 に設定されている場合は、ブロックを D-SACK します。

注：パケットは、受信確認されたパケット、またはすでに受信された順序どおりのパケットである場合、D-SACK と見なされます。

クライアントの再作成

Citrix ADC アプライアンスは、SACK ベースのリカバリ中にクライアントの再接続を処理できます。

PCB 上のエンドポイントをマーキングするためのメモリチェックは、使用可能な合計メモリを考慮していません

Citrix ADC アプライアンスでは、メモリ使用量のしきい値が、使用可能なメモリの合計を使用する代わりに 75% に設定されている場合、新しい TCP 接続は TCP 最適化をバイパスします。

SACK ブロックが欠落しているため、不必要な再送信

非エンドポイントモードでは、DUPACKS を送信するときに、順序どおりのパケットが少ない場合に SACK ブロックが欠落すると、サーバからの追加の再送信がトリガーされます。

過負荷のために最適化をバイパスした接続数の **SNMP**

過負荷が原因で TCP 最適化をバイパスした接続数を追跡するために、Citrix ADC アプライアンスに次の SNMP ID が追加されました。

1. 1.3.6.1.4.1.5951.4.1.1.46.13 (tcpOptimizationEnabled). TCP 最適化で有効にされた接続の合計数を追跡します。
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed). 接続の総数を追跡するには、TCP 最適化をバイパスします。

動的受信バッファ

TCP パフォーマンスを最大化するために、Citrix ADC アプライアンスは TCP 受信バッファサイズを動的に調整できるようになりました。

トラブルシューティングのガイドライン

October 7, 2021

テクニカルサポート

すべてのトラブルシューティングとエスカレーションのクエリには、最新の Citrix ADC テクニカルサポートバンドルが必要です。バンドルには、現在の構成、インストールされているファームウェアのバージョン、ログファイル、未処理のコアなどが含まれます。

例:

```
show techsupport

showtechsupport data collector tool - $Revision: #5 $!
...
<!--NeedCopy-->
```

すべてのデータは、

```
...
Archiving all the data into "/var/tmp/support/collector_P_192
   .168.121.117_18Jun2015_09_53.tar.gz" ....
Created a symbolic link for the archive with /var/tmp/support/support.
   tgz
/var/tmp/support/support.tgz  ---- points to ---> /var/tmp/support/
   collector_P_192.168.121.117_18Jun2015_09_53.tar.gz
<!--NeedCopy-->
```

テクニカルサポートバンドルが生成されると、SCP を使用してコピーされることがあります。

トレース

Citrix ADC TCP 最適化の問題では、通常、Citrix ADC トレースを適切にトラブルシューティングする必要があります。同様の条件、すなわち、同じセル上、同じ時間帯に、同じユーザー機器とアプリケーションを使用して、トレースをキャプチャしようとする必要があることに注意してください。

トレースのキャプチャには、nstrace の起動コマンドと nstrace の停止コマンドを使用できます。

- トレース上で無関係な不要なパケットをキャプチャしないようにするため、適切なフィルタを使用することを強くお勧めします。たとえば、スタート nstrace-filter 'IP == 10.20.30.40 'を使用して、IP アドレス

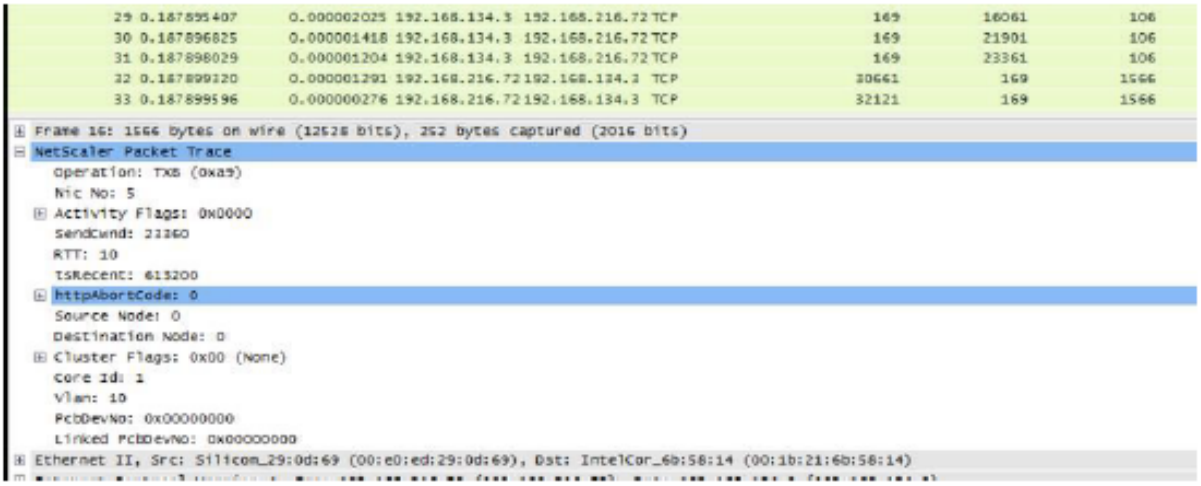
10.20.30.40 との間で送受信されるパケットだけをキャプチャします。これは、ユーザー機器の IP アドレスです。

- -tcpdump オプションは使用しないでください。これは、デバッグに必要な nstrace ヘッダーを取り除くためです。

トレース分析

Citrix ADC トレースがキャプチャされると、Wireshark 1.12 以降でトレースが表示されることがあります。キャプチャされたトレースに、次の画面キャプチャに示すように、適切な Citrix ADC パケットトレースヘッダーが含まれていることを確認します。

追加のデバッグヘッダーは、次の図にも表示されます。



接続テーブル

問題が TCP 最適化に関連し、再現できるか、または進行中の場合は、プライマリ T1 ノードから問題が発生したときに接続テーブルも取得することをお勧めします。

テーブルを取得するには、BSD シェルに切り替えて次のコマンドを実行する必要があります。

```
1 shell
```



```
...

nscli -U 127.0.0.1:nsroot:nsroot show connectiontable -detail full link
    > /var/tmp/contable.log
<!--NeedCopy-->
```

> 注
>
> コマンドはより長い時間実行され、その時点で管理 CPU に負荷がかかることがありますが（接続テーブルのエントリ数によって異なります）、サービスには影響しません。

よくある質問

October 7, 2021

タイムアウト

> 重要
>
> *nsapimgr* ノブを使用する前に、Citrix カスタマーサポートに問い合わせてください。

Citrix ADC T1 仮想サーバーおよびサービスで設定できるさまざまなアイドル接続タイムアウトを次に示します。vserver またはサービスレベルでクライアントまたはサーバー接続に対して設定されたアイドルタイムアウトは、TCP ESTABLISHED 状態の接続に対してのみ適用可能で、アイドル状態です。

- 負荷分散仮想サーバー cltTimeout パラメータは、クライアントから負荷分散仮想サーバーへの接続が、アプライアンスが接続を閉じるまでにアイドル状態である必要がある時間を秒単位で指定します。
- Service svrTimeout パラメータは、アプライアンスからサービスまたはサーバーへの接続が、アプライアンスが接続を閉じるまでにアイドル状態である必要がある時間を秒単位で指定します。
- Service cltTimeout パラメータは、アプライアンスが接続を閉じる前に、クライアントからサービスへの接続がアイドル状態である必要がある時間を秒単位で指定します。

サービスが負荷分散仮想サーバーにバインドされると、負荷分散仮想サーバーの cltTimeout が優先され、サービスの cltTimeout は無視されます。

負荷分散仮想サーバーにバインドされたサービスがない場合、グローバルアイドルタイムアウト（tcpServer）がサーバー側の接続に使用されます。次のように設定できます。

コマンド:

```
set ns timeout  - tcpServer 9000
<!--NeedCopy-->
```


他の状態の接続では、タイムアウト値が異なります。

- ハーフオープン接続のアイドルタイムアウト:120 秒 (ハードコードされた値)
- TIME_WAIT 接続のアイドルタイムアウト:40 秒 (ハードコードされた値)
- 閉じる接続のアイドルタイムアウトの半分。デフォルトでは 10 秒であり、スニペットを使用して 1〜600 の間に設定することができます

コマンド:

```
1  set ns timeout -halfclose 10
2  <!--NeedCopy-->
```

ハーフクローズタイムアウトがトリガーされると、接続はゾンビ状態になります。ゾンビタイムアウトが期限切れになると、ゾンビクリーンアップが起動し、T1 はデフォルトで特定の接続に対してクライアント側とサーバー側の両方で RST を送信します。

- ゾンビタイムアウト: 非アクティブな TCP 接続をクリーンアップするためにゾンビクリーンアッププロセスを実行する必要がある間隔。デフォルトのタイムアウト値は 120 で、1 〜600 の間で設定できます。

コマンド:

```
1  set ns timeout -zombie 120
2  <!--NeedCopy-->
```

最大セグメントサイズテーブル

Citrix ADC T1 アプライアンスは、システムメモリスタックでハーフオープン接続を維持するのではなく、SYN Cookie を使用して SYN フラッド攻撃を防御します。アプライアンスは、TCP 接続を要求する各クライアントに Cookie を送信しますが、ハーフオープン接続の状態は維持されません。代わりに、アプライアンスは、最終的な ACK パケットを受信したときにのみ、または HTTP トラフィックの場合は HTTP 要求を受信したときにのみ、接続にシステムメモリを割り当てます。これにより、SYN 攻撃が防止され、正規のクライアントとの通常の TCP 通信が中断されずに継続されます。特定の機能は、無効にするオプションなしでデフォルトで有効になっています。

ただし、標準の SYN クッキーは 8 つの最大セグメントサイズ (MSS) 値のみを使用するように接続を制限するため、注意事項があります。接続 MMS が事前定義された値と一致しない場合、クライアント側とサーバー側の両方に対して次の利用可能な低い値を取得します。

事前定義された TCP 最大セグメントサイズ (MSS) の値は次のとおりです。新しい nsapimgr ノブを使用して構成できます。

1460	1440	1330	1220	956	536	384	128

新しい MSS テーブル:

- ジャンボフレームのサポートを含める必要はありません。デフォルトでは、ジャンボフレーム用の MSS テーブルでは 8 つの値が予約されていますが、標準イーサネットサイズのフレームだけを含むようにテーブル設定を変更できます。
- 16 個の値を持つ必要があります
- 降順の値を持つ必要があります
- 最後の値として 128 を含める必要があります

新しい MSS テーブルが有効な場合は、テーブルが格納され、SYN-cookie ローテーション時に古い値がスイッチアウトされます。それ以外の場合、新しいテーブルはエラーを返します。変更が新しい接続に適用されます。既存の接続では、接続が期限切れになるか終了するまで、古い MSS テーブルが保持されます。

Citrix ADC アプライアンスの現在の MSS テーブルを表示するには、次のコマンドを入力します。

コマンド:

```
>shell

#nsapimgr -d mss_table
```

例:

```
#nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1440,1330,1212,956,536,384,128
      }


Done.
```

mss テーブルを変更するには、次のコマンドを入力します。

コマンド:


```
>shell

#nsapimgr -s mss_table=<16 comma seperated values>
```

例:

```
#nsapimgr -ys mss_table
   =9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128


# nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128
     }


Done.
```

標準イーサネットサイズの値を使用する例を次に示します。

例:

```
#nsapimgr -ys mss_table
   =1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128


# nsapimgr -d mss_table

MSS table

{
 1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128
     }


Done.
```

Citrix ADC アプライアンスの再起動後もこの変更を永続的にするには、「/nsconfig/rc.netscaler」ファイルにコ

マンド##`nsapimgr -ys mss_table=<16 comma seperated values>` を含めます。「rc.netscaler」ファイルが存在しない場合は、「/nsconfig」フォルダの下にファイルを作成し、コマンドを追加します。

メモリ過負荷保護

Citrix ADC パケット処理エンジン（PPE）は、その 1 つの PPE によって使用されているメモリが指定されたハイウォーターマーク値を超える場合、TCP 最適化からの接続のバイパスを開始します。PPE メモリ使用率が～2.6GB を超えると、最適化から 新しい接続をバイパスし始めます。既存の接続（以前に最適化のために許可された接続）は、引き続き最適化を取得しています。このウォーターマーク値は意図的に選択されているため、チューニングにはお勧めできません。

> 注
>
> ウォーターマークの値を変更する正当な理由があると思われる場合は、カスタマーサポートにお問い合わせください。

Happy Eyeballs クライアントのサポート

Citrix ADC アプライアンスが、状態が不明な宛先に対する SYN を受信した場合、アプライアンスはまずサーバーの到達可能性をチェックし、次にクライアントの確認応答を行います。このプロービングメカニズムにより、デュアル IP スタックを持つクライアントは、デュアルスタックのインターネットサーバの到達可能性を検出できます。クライアントが IPv6 アクセスと IPv4 アクセスの両方が使用可能であることを検出すると、より迅速に応答するサーバーへの接続を確立し、もう一方のアクセスをリセットします。Citrix ADC アプライアンスの接続がリセットを受信すると、対応するサーバー側の接続がリセットされます。

注：この機能には、Citrix ADC アプライアンスで無効/有効にするユーザー設定可能な TCP 設定はありません。

Happy Eyeball のサポートの詳細については、RFC 6555 を参照してください。

Citrix ADC ビデオ最適化

October 7, 2021

Citrix ADC アプライアンスは、モバイルネットワークを介したビデオトラフィックの ABR ビデオトラフィックを最適化するための最適化技術と機能を提供します。これにより、ユーザーエクスペリエンスが向上し、ネットワーク帯域幅の消費量が削減されます。

ここでは、次のトピックについて説明します。

- はじめに
- ライセンス
- TCP 経由のビデオ最適化の設定
- UDP によるビデオ最適化の設定

はじめに

October 7, 2021

メディアファイルは、モバイルネットワークを介してトラフィック量が増加しており、より高速なネットワーキング技術への移行により、暗号化されたビデオトラフィックの量が劇的に増加しています。従来のメディア配信技術 (プログレッシブダウンロード) は、高い伝送速度で許容可能なエクスペリエンス (QoE) を提供できません。これにより、アダプティブビットレート（ABR）プロトコルの導入がなされました。これは、利用可能なネットワーク帯域幅にストリーミングビットレートを調整し、ビデオを受信するハンドセットの能力に合わせてストリーミング品質を制限することができます。しかし、ABR プロトコルは、インターネット上で行うようにモバイルネットワークでも同様に動作しません。したがって、モバイル事業者は ABR トラフィックを最適化する必要があります。

Citrix ADC アプライアンスは、着信ビデオトラフィックを検出し、ABR ビデオを選択的に最適化する独自の機能を備えています。

Citrix ADC ビデオ最適化の仕組み

Citrix ADC アプライアンスは、TCP を介した暗号化された ABR トラフィック（Facebook のビデオトラフィックを含む）を特定し、QUIC を介した YouTube ABR トラフィックを最適化できます。アプライアンスには次の機能があります。

1. HTTP 経由でプログレッシブダウンロード (PD) 動画を検出します。
2. 検出し、HTTP 経由で ABR 動画を最適化します。
3. 検出し、HTTPS 経由で ABR 動画を最適化します。
4. 検出し、QUIC 上の YouTube の ABR 動画を最適化.

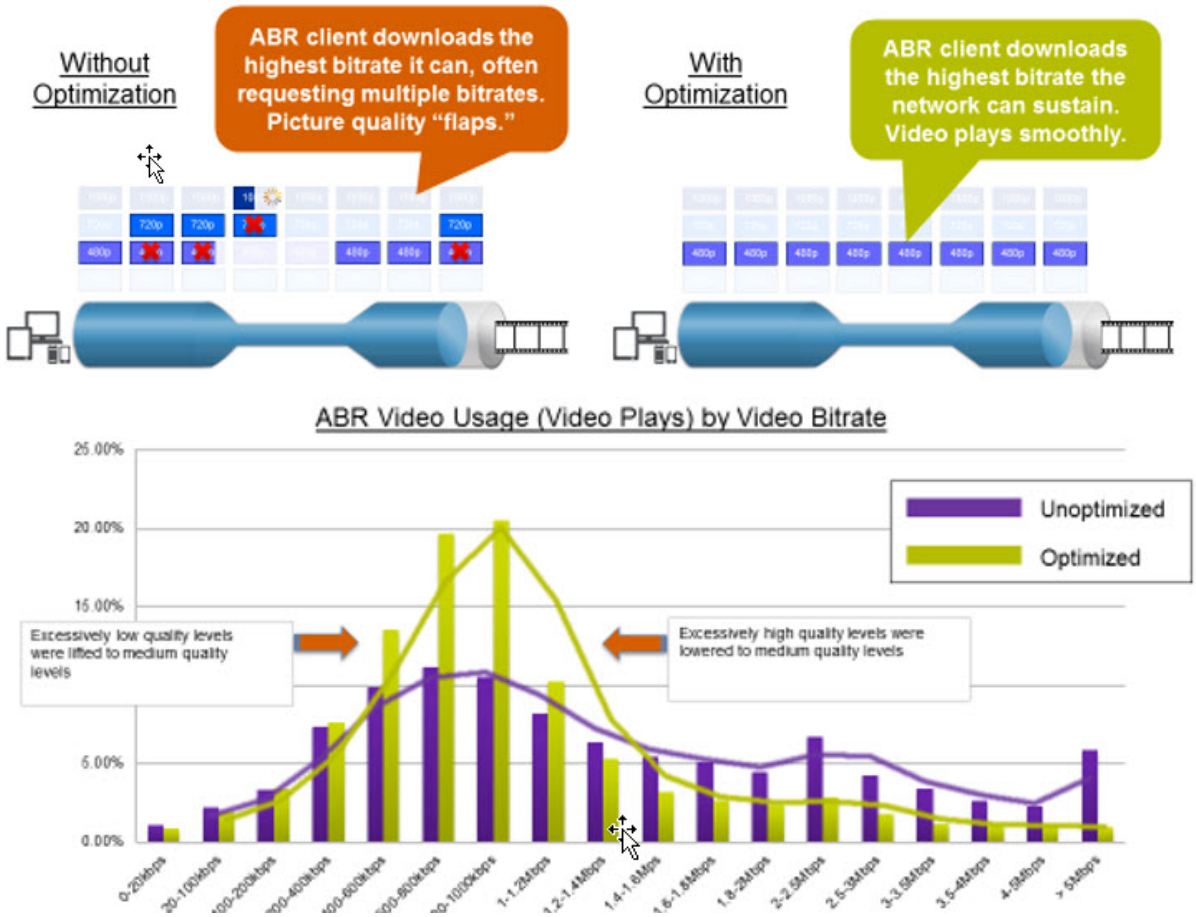
また、アプライアンスは、TCP および QUIC プロトコルを介したビデオトラフィックを検出するために、以下のサポートドメインを使用します。

- TCP 経由で暗号化されていない ABR ビデオ。Appliance は、標準準拠のビデオストリーミング Web サイトをすべて検出します。アプライアンスは、応答ビデオペイロードヘッダー、URL、および HTTP ヘッダーを検査して ABR セッションを検出します。
- TCP 経由で暗号化された ABR ビデオ。アプライアンスは、ドメイン、SSL ヘッダー、トラフィックパターンに基づく汎用的なヒューリスティックアルゴリズムを使用して、ABR セッションを検出します。これを使用して、アプライアンスは 95% の精度で、上位のビデオ Web サイトを検出するための組み込みサポートを備えており、新しいビデオタイプのサポートを追加していきます。Citrix ADC には、ネットワークカバレッジを確保するために、地域または国の上位の暗号化 ABR サイトに対する追加の検証を提供するプログラムもあります。
- QUIC 経由で暗号化された ABR 動画。アプライアンスは、YouTube などの QUIC ベースのビデオプロバイダの ABR セッションを検出します。検出アルゴリズムは、QUIC ヘッダーとドメインを活用するヒューリスティックに基づいています。Citrix ADC は、QUIC を使用して新しいビデオサイトのサポートを引き続き追加します。

長所

ABR ビデオトラフィックを最適化すると、次の利点があります。

- ピーク時間における混雑時にネットワークを管理する。
- 動画再生の一貫性を向上させ動画の再生速度低下を抑える。
- 新しい動画サービスオファリング（Binge-on 動画サービスなど）を有効にする。
- 顧客が持続可能で最適な動画品質を選択できるようにする。
- サブスクライバーに一貫性のあるユーザーエクスペリエンスを提供する。



TCP を介したビデオの最適化

TCP 経由の ABR トラフィックの Citrix ADC 最適化は、次のように動作します。

1. アプライアンスが TCP 経由で受信する HTTP または HTTPS トラフィックは、対応する負荷分散仮想サーバーに送信されます。
2. 仮想サーバにバインドされた組み込みの検出ポリシーと、他の独自の検出アルゴリズムを組み合わせることで、トラフィックが評価されます。
3. ポリシーでは、組み込みのビデオ検出シグネチャのセットを使用して、ビデオタイプを検出します。トラフィックに一致するポリシーでは、ビデオタイプを次のいずれかとして分類するアクションが適用されます。

a) クリアテキスト PD
b) クリアテキスト ABR
c) 暗号化された ABR
d) その他

4. 同じ仮想サーバにバインドされた最適化ポリシーは、トラフィックを評価し、トラフィックに適用する最適化ビットレートを決定します。
5. 最適化ビットレートは、トラフィックがクリアテキストの ABR または暗号化された ABR の場合に適用されます。

モバイルサービスプロバイダは、2G、3G、および 4G モバイルトラフィックのダウンロード速度を設定することで、エクスペリエンスの品質（QoE）を向上させることができます。これにより、ビデオの開始時間やバッファリングイベントが短縮されます。最適化によって、ビデオセッションで消費されるネットワーク帯域幅の量を減らすこともできます。

最適化技術には、ダイナミックバースト制御とランダムサンプリングが含まれます。

ダイナミックバースト制御

Citrix ADC ABR の最適化は、変化するネットワーク条件に動的に適応します。設定済みのペーシングレートの 1.3 倍の初期バーストレートを 15 秒間設定できます。初期バーストレートは、複数のセッションが同じ TCP 接続または TCP 接続のグループを使用している場合でも、最適化された ABR ビデオセッションの開始に適用されます。

アプライアンスは、ネットワークでサポートされているビットレートが設定されたペーシングレートを下回った場合のリカバリバーストもサポートします。たとえば、有効ビットレートが 7 秒目に低下し、最初のバーストの 15 秒目に回復した場合、アプライアンスは次のバーストサイクル中に損失を回復します。これにより、アプライアンスはすべての加入者のネットワーク帯域幅を動的に最適化し、画質がピクセルごとに一貫した状態に保たれます。

注：初期バースト中に回復バーストが発生した場合、ペーシングビットレートは最大回復バーストレートと初期バーストレートを超えないようにしてください（初期バースト係数の上に回復バースト係数を追加しないでください）。そうしないと、メディアプレーヤーが高画質モードに移行するほど高速になる可能性があります。ただし、必要に応じて、初期バースト（Initial Burst）の継続時間を延長して、未使用の帯域幅を補正できます。

ランダムサンプリング

ビデオの最適化による削減額を見積もるために、Citrix ADC アプライアンスはランダムサンプリングを実装しています。この方法では、アプライアンスは、検出されたビデオトラフィックの構成可能な割合をランダムに選択します（ランダムサンプリングパラメータは 0 ～100 の範囲の整数値であるため、1% 未満は不可能です）。これらのランダムに選択され、最適化されていないトランザクション（およびセッション）は参照グループになり、トランザクションログで識別されます（バイトサイズやタイマーフィールドなど、他の特性とともに）。最適化されたセッションの特性もログに記録され、レポートエンジンは最適化グループと参照グループの統計を比較して、最適化による節約（ABR 最適化による節減を含む）を見積もります。

UDP によるビデオの最適化

Google は、QUIC と呼ばれる新しいトランスポートプロトコルを導入しました。Google の QUIC プロトコルは、TCP+TLS+HTTP/2 と非常によく似ており、UDP の上に実装されています。Citrix ADC は、QUIC プロトコルでストリーミングされた YouTube の ABR 動画を検出し、ABR 動画の最適化を TCP 経由の ABR と同様の方法で適用できます。

ライセンス

October 7, 2021

ビデオ最適化機能は、基本的な CBM ライセンスと CBM Premium ライセンスを購入すると Telco プラットフォームで動作します。その他の Citrix ADC プラットフォームでは、この機能は CNS Premium ライセンスの購入と連動します。ビデオ最適化機能を設定する前に、アプライアンスに適切なライセンスが必要です。

Telco プラットフォームのライセンスサポート：

- **CBM_TXXX_SERVER_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **CNS_WEBF_SSERVER_Retail.lic**

ここで、XXX はスループットです（例：Citrix ADC T1000）。

その他の Citrix ADC プラットフォームのライセンスサポート：

- **CNS_XXX_SERVER_PLT_Retail.lic**

ここで、XXX はスループットです。

プレミアムライセンスファイルをアップロードするには、以下の手順に従います。

1. Citrix ADC アプライアンスに有効なライセンスファイルをインストールする必要があります。ライセンスは、予想される最大 GiLAN スループットと同数の Gbps 以上をサポートする必要があります。

   ライセンスファイルは、次の画面キャプチャに示すように、SCP クライアントを介してアプライアンスの /nsconfig/license にコピーする必要があります。

```
1 > shell ls /nsconfig/license/
2 CNS_V3000_SERVER_PLT_Retail.lic ssl
3 <!--NeedCopy-->
```

2. 次のスクリーンキャプチャに示すように、ウォームリスタートを実行して、新しいライセンスを適用します。


```
> reboot -warm
Are you sure you want to restart NetScaler (Y/N)? [N]:y
 Done
<!--NeedCopy-->
```

3. 再起動が完了したら、show license CLI を使用して、ライセンスが正しく適用されていることを確認します。

以下の例では、Premium エディションの Premium ライセンスが正常にインストールされています。

```
    > show license

                License status:

                                        Video Optimization: YES

                                                    ...

                                    Model Number ID: 110050

                                       License Type: Premium License
<!--NeedCopy-->
```

TCP 経由のビデオ最適化の設定

October 7, 2021

TCP 経由でビデオトラフィックを最適化するには、まずビデオ最適化機能を有効にします。次に、アプライアンスは組み込みの検出ポリシーをアクティブにして、着信ビデオトラフィックを検出し、ビデオの種類を特定します。各ビデオタイプのユーザ設定可能な最適化ポリシーは、トラフィックの最適化に必要な最適化ビットレートを指定します。

CLI を使用した TCP 経由のビデオ最適化の設定

Citrix ADC アプライアンスでビデオの最適化を構成するには、次のタスクを実行します。

1. ビデオ最適化機能を有効にします。
2. HTTP および HTTPS トラフィック用の仮想サーバーを追加します。
3. すべての組み込み検出ポリシーを、HTTP トラフィック用の負荷分散仮想サーバーにバインドします。
4. 組み込みのすべての検出ポリシーを、HTTPS トラフィック用の SSL ブリッジロードバランシング仮想サーバーにバインドします。
5. HTTP トラフィックと HTTPS トラフィックに必要な最適化ポリシーを追加します。

6. 最適化ポリシーを HTTP トラフィックの負荷分散仮想サーバーにバインドします。
7. HTTPS トラフィック用に SSL ブリッジロードバランシング仮想サーバに最適化ポリシーをバインドします。

ビデオの最適化の有効化

Citrix ADC アプライアンスでビデオトラフィックの検出、最適化、およびレポートを実行する場合は、ビデオ最適化機能を有効にして、最適化をオンに設定する必要があります。この機能を有効にすると、組み込みの検出ポリシーを使用して着信ビデオトラフィックを識別し、最適化ポリシーを設定して暗号化された ABR トラフィックを最適化できます。ABR ビデオトラフィックを最適化するには、ダウンロードビットレート（ペーシングレートとも呼ばれます）を設定する必要があります。

ロードバランシング機能も有効にする必要があります。また、HTTPS トラフィックにビデオ最適化を使用する場合は、SSL 機能を有効にする必要があります。

ビデオ最適化機能を有効にするには

コマンドプロンプトで、次のコマンドを入力します。

```
enable ns feature VideoOptimization
<!--NeedCopy-->
```

> 注
>
> ビデオ最適化のパフォーマンスとビデオインサイトレポートを監視する場合は、AppFlow 機能を有効にしてから、Citrix Application Delivery Management（ADM）のビデオ分析機能にアクセスする必要があります。詳細については、Video Insight ドキュメントを参照してください。

HTTP および **HTTPS** ビデオトラフィック用の仮想サーバの作成

Citrix ADC アプライアンスは、さまざまな種類の着信ビデオトラフィックを検出して最適化するために、異なる仮想サーバーを使用します。アプライアンスは、TCP トラフィック用に次のタイプの仮想サーバーをサポートします。

- **HTTP** ロードバランシング仮想サーバー。HTTP ビデオトラフィックを検出するために、アプライアンスは HTTP ロードバランシング仮想サーバーを使用します。アプライアンスがクライアントから受信する HTTP ビデオ要求を管理します。
- **SSL** ブリッジ負荷分散仮想サーバー。暗号化されたビデオトラフィックを検出するには、アプライアンスで SSL ブリッジ仮想サーバーを設定する必要があります。

HTTP ビデオトラフィックを検出するための **HTTP** ロードバランシング仮想サーバーを追加するには

コマンドプロンプトで、次のように入力します。


```
add lb vserver <name> HTTP * 80 -persistenceType NONE
<!--NeedCopy-->
```

例:

```
add lb vserver ProxyVserver-HTTP HTTP * 80 -persistenceType NONE -
    cltTimeout 120
<!--NeedCopy-->
```

HTTPS ビデオトラフィックを検出するための **SSL Bridge** 仮想サーバを追加するには

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name> SSL_BRIDGE * 443 -persistenceType NONE
<!--NeedCopy-->
```

例:

```
add lb vserver ProxyVserver-SSL SSL_BRIDGE * 443 -persistenceType NONE
    -cltTimeout 180
<!--NeedCopy-->
```

HTTP ロードバランシング仮想サーバーへの組み込み検出ポリシーのバインド

HTTP 接続を介してビデオトラフィックを検出するには、組み込みのすべての検出ポリシーをロードバランシング仮想サーバーにバインドする必要があります。ポリシーは、ポリシータイプに応じて、要求時間または応答時間処理のいずれかにバインドする必要があります。

> 注:
>
> ns_videoopt_http_body_detectionビデオ最適化ポリシーは CONNECTHTTP リクエストメソッドをサポートしていません。

さまざまなビデオタイプの検出ポリシーを **HTTP** ロードバランシング仮想サーバにバインドするには

コマンドプロンプトで、種類ごとに適切なコマンドを入力します。使用可能なコマンドは次のとおりです。


```
bind lb vserver <name> -policyName ns_videoopt_http_abr_netflix -
    priority <integer> -type (REQUEST | RESPONSE)

bind lb vserver <name>   -policyName ns_videoopt_http_abr_netflix2 -
    priority <integer> -type (REQUEST | RESPONSE)

bind lb vserver <name>   -policyName ns_videoopt_http_abr_youtube -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube2  -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>   -policyName ns_videoopt_http_pd_youtube3 -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>  -policyName ns_videoopt_http_abr_generic -
    priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

例：

```
bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_netflix -priority 400  type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_netflix2 -priority 500 -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_youtube -priority 600  -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_pd_youtube -priority 800  -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_pd_youtube2 -priority 900 -type RESPONSE

bind lb vserver  ProxyVserver-HTTP   -policyName
    ns_videoopt_http_pd_youtube3 -priority 1000 -type REQUEST

bind lb vserver  ProxyVserver-HTTP   -policyName
```



```
    ns_videoopt_http_abr_generic -priority 1100 -type RESPONSE
<!--NeedCopy-->
```

HTTP 本文コンテンツ検出ポリシーのロードバランシング仮想サーバーのバインド

HTTP 経由でビデオトラフィックを検出するには、本文のコンテンツ検出ポリシーをロードバランシング仮想サーバーにバインドする必要があります。次のコマンドを使用できます。

```
bind lb vserver <name> -policyName ns_videoopt_http_body_detection -
    priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

例:

```
bind lb vserver ProxyVserver-HTTP -policyName
    ns_videoopt_http_body_detection -priority 1500 -type REQUEST
<!--NeedCopy-->
```

SSL ブリッジロードバランシング仮想サーバへの組み込み検出ポリシーのバインド

HTTPS 接続を介してビデオトラフィックを検出するには、組み込みの検出ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドする必要があります。

検出ポリシーを **SSL** ブリッジロードバランシング仮想サーバにバインドするには

コマンドプロンプトで、種類ごとに適切なコマンドを入力します。使用可能なコマンドは次のとおりです。

```
bind lb vserver <name> -policyName ns_videoopt_https_abr_netflix -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_youtube -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_generic -
    priority <positive_integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```


例:

```
bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_netflix -priority 120 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_youtube -priority 140 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_generic -priority 150 -type REQUEST
<!--NeedCopy-->
```

ABR トラフィックのペーシングのための最適化ポリシーの追加

ABR トラフィックを最適化するには、最適化ポリシーと関連するアクションを設定する必要があります。次に、検出ポリシーをバインドしたのと同じ負荷分散仮想サーバーにポリシーをバインドします。ポリシーを作成するときにアクションを含めることができるように、ポリシーごとにアクションを最初に作成します。

最適化アクションを追加するには

コマンドプロンプトで入力します。

```
add videooptimization pacingaction <action Name> -rate <integer> [-
    comment <string>]
<!--NeedCopy-->
```

rate パラメータは、トラフィックを送信するレート（ペーシングレート）を Kbps 単位で指定します。

例:

```
add videooptimization pacingaction MyOptAct2000 -rate 2000
<!--NeedCopy-->
```

最適化ポリシーを追加するには

コマンドプロンプトで入力します。


```
add videooptimization pacingpolicy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

例:

```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
    MyOptAct2000
<!--NeedCopy-->
```

HTTP 負荷分散仮想サーバーへの最適化ポリシーのバインド

HTTP 接続で ABR ビデオトラフィックを最適化するには、検出ポリシーがバインドされているロードバランシング仮想サーバに最適化ポリシーをバインドする必要があります。

最適化ポリシーを負荷分散仮想サーバーにバインドするには

コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name> -policyName <policy_name> -priority <
    positive_integer> -type  (REQUEST | RESPONSE)
<!--NeedCopy-->
```

例:

```
bind lb vserver ProxyVserver-HTTP -policyName myOptPolicy2000 -priority
     3400 -type  REQUEST
<!--NeedCopy-->
```

SSL ブリッジ仮想サーバーへの最適化ポリシーのバインド

HTTPS 接続で ABR ビデオトラフィックを最適化するには、組み込み検出ポリシーがバインドされている SSL Bridge 仮想サーバに最適化ポリシーをバインドする必要があります。

最適化ポリシーを **SSL Bridge** 仮想サーバーにバインドして、暗号化されたトラフィックをペーシングするには

コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name> -policyName <policy_name> -priority <
    positive_integer> -type  (REQUEST |RESPONSE)
<!--NeedCopy-->
```

例:

```
bind lb vserver ProxyVserver-SSL -policyName myOptPolicy2000 -priority
    3400 -type  REQUEST
<!--NeedCopy-->
```

ビデオの最適化ペーシングパラメータの設定

CLI では、ランダムサンプリング率などのビデオ最適化ペーシングパラメータを設定できます。

ランダムサンプリングのパーセンテージを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set videooptimization parameter  - RandomSamplingPercentage <realNumber>
<!--NeedCopy-->
```

ここで、実数は 0.0 から 100.0 までの値です。

例:

```
set videooptimization parameter -RandomSamplingPercentage 50
<!--NeedCopy-->
```

GUI を使用した TCP 経由のビデオ最適化の設定

GUI を使用すると、次のことが可能になります。

- ビデオ最適化機能を有効にします。
- HTTP ロードバランシング仮想サーバを作成します。

- SSL ブリッジ負荷分散仮想サーバーを作成します。
- 組み込みの検出ポリシーを HTTP 負荷分散仮想サーバーにバインドします。
- 組み込みの検出ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドします。
- 最適化ポリシーを作成します。
- 最適化アクションを作成します。
- 最適化ペーシングパラメータを設定します。
- 最適化ポリシーをバインドして、HTTP トラフィック用の仮想サーバのロードバランシングを行います。
- HTTPS トラフィック用の SSL ブリッジ負荷分散仮想サーバーに最適化ポリシーをバインドします。

ビデオ最適化機能を有効にするには

1. ナビゲーションペインで、[システム] を展開し、[設定**]** をクリックします。
2. [設定] ページで、[拡張機能の構成] リンクをクリックします。
3. [高度な機能の設定] ページで、[ビデオの最適化] チェックボックスをオンにします。
4. [**OK**] をクリックし、[**Close**] をクリックします。

HTTP トラフィック用の負荷分散仮想サーバーを作成するには

1. Citrix ADC アプライアンスにサインインし、「トラフィック管理」>「負荷分散」>「仮想サーバー」ページに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。
   a) **Name**：負荷分散仮想サーバの名前。
   b) プロトコル。プロトコルの種類を HTTP として選択
   c) **IP** アドレスタイプ。IP アドレスの種類:IPv4 または IPv6。
   d) **IP** アドレス。仮想サーバに割り当てられた IPv4 アドレスまたは IPv6 アドレス。
   e) ポート。仮想サーバのポート番号。
4. [**OK]** をクリックして、他のオプションのパラメータの設定を続行します。詳細は、「仮想サーバーの作成」を参照してください。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTPS トラフィック用の負荷分散仮想サーバーを作成するには

1. Citrix ADC アプライアンスにサインインし、「トラフィック管理」>「負荷分散」>「仮想サーバー」ページに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。
   a) **Name**：負荷分散仮想サーバの名前。
   b) プロトコル。SSL ブリッジとしてプロトコルタイプを選択します。
   c) **IP** アドレスタイプ。IP アドレスの種類:IPv4 または IPv6。

d) **IP** アドレス。仮想サーバに割り当てられた IPv4 アドレスまたは IPv6 アドレス。

e) ポート。仮想サーバのポート番号。

4. [**OK**] をクリックして、他のオプションのパラメータの設定を続行します。詳細については、仮想サーバーの作成を参照してください。
5. [作成] をクリックし、[閉じる] をクリックします。

組み込みの検出ポリシーを負荷分散仮想サーバーにバインドするには

1. Citrix ADC アプライアンスにサインインし、「トラフィック管理」>「負荷分散」>「仮想サーバー」画面に移動します。
2. 詳細ペインで、負荷分散仮想サーバーを選択し、[**Edit**] をクリックします。
   a) [詳細設定] セクションで、[ポリシー] をクリックします。
   b) [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
   c) [ポリシー] セクションで、次のパラメータを設定します。
   d) ポリシーを選択します。ドロップダウンリストからビデオ最適化検出ポリシーを選択します。
   e) タイプを選択します。ポリシーの種類を [要求] として選択します。
   f) [続行] をクリックします。
3. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

組み込みの検出ポリシーを **SSL** ブリッジ負荷分散仮想サーバーにバインドするには

1. Citrix ADC アプライアンスにログオンし、「トラフィック管理」>「負荷分散」>「仮想サーバー」画面に移動します。
2. 詳細ペインで、SSL ブリッジロードバランシング仮想サーバを選択し、[**Edit**] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. [ポリシー] セクションで、次のパラメータを設定します。
   a) ポリシーを選択します。ドロップダウンリストから、ビデオ最適化検出ポリシーを選択します。
   b) タイプを選択します。ポリシーの種類を [要求] として選択します。
6. [続行] をクリックします。
7. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

ビデオ最適化アクションを作成するには

1. Citrix ADC アプライアンスにログオンし、「構成」>「最適化」>「ビデオの 最適化」>「ペーシング」>「アクション」の順に選択します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [ビデオの最適化ペーシングアクションの作成] ページで、次のパラメータを設定します。
   a) **Name**: 最適化アクションの名前。

b) **ABR** 最適化レート（**Kbps**）。ABR ビデオトラフィックを送信するペーシングレート。ABR 最適化のデフォルトレートは 1000 Kbps です。最小値は 1 で、最大値は 2147483647 です。
c) コメント。アクションの簡単な説明。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

ビデオ最適化ポリシーを作成するには

1. Citrix ADC アプライアンスにログオンし、「構成」>「最適化」>「ビデオの 最適化」>「ペーシング」>「ポリシー」の順に選択します。
2. 詳細ペインで、［**Add**］をクリックします。
3. [ビデオ最適化ペーシングポリシーの作成] ページで、次のパラメータを設定します。
   a) **Name**：最適化ポリシーの名前
   b) 式。ポリシーを実装するカスタム正規表現式。
   c) 操作。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
   d) **UNDEF** アクション。着信要求が最適化ポリシーと一致しない場合、未定義のイベント。
   e) コメント。ポリシーの簡単な説明。
   f) アクションをログに記録します。目的のログメッセージを作成する監査ログアクションを選択します。
4. [作成] をクリックし、[閉じる**]** をクリックします。

ビデオの最適化ペーシングパラメータを設定するには

1. Citrix ADC アプライアンスにログオンし、［構成］>［最適化］>［ビデオの最適化］に移動します。
2. [ビデオの最適化] ページで、[ビデオの最適化設定の変更**]** リンクをクリックします。
3. [ビデオの最適化設定**]** ページで、次のパラメータを設定します。
   a) ランダムサンプリングのパーセンテージ **(%)**。ランダムサンプリング用に選択されたパケットのパーセンテージ。
4. [**OK]** をクリックして [閉じる] をクリックします。

ビデオ最適化ポリシーを **HTTP** ロードバランシング仮想サーバにバインドするには

1. Citrix ADC アプライアンスにログオンし、［構成］>［最適化］>［ビデオの最適化］に移動します。
2. [ビデオの最適化] ページで、[ビデオの最適化ペーシングポリシーマネージャ] リンクをクリックします。
3. 次のパラメータを設定します。
   a) バインドポイント。要求または応答の処理中に最適化ポリシーを適用するポイント。
   b) **[**接続タイプ**]**。要求または応答としての接続タイプ。
   c) 仮想サーバ。ポリシーをバインドする負荷分散仮想サーバー。
   d)［続行］をクリックします。
4. バインドポイント **(Bind Point**) セクションで、次のいずれかの操作を行います。
   a) リストからポリシーを選択します。
   b)「バインドの追加」をクリックして、「ポリシーバインディング」スライダーにアクセスします。

i. 既存のポリシーを選択するか、新しいポリシーを追加します。
ii. バインドの詳細を入力し、「バインド」をクリックします。

5. [閉じる] をクリックします。

ビデオ最適化ポリシーを **SSL** ブリッジ負荷分散仮想サーバーにバインドするには

1. 構成への Citrix ADC アプライアンスとナビゲートにログオンします > 最適化 > ビデオの 最適化。
2. [ビデオの最適化] ページで、[ビデオの最適化ペーシングポリシーマネージャ] リンクをクリックします。
3. [ビデオ最適化ポリシーマネージャ] ページで、次のパラメータを設定します。
   a) バインドポイント。要求/応答処理中に最適化ポリシーを適用するポイント。
   b) [接続タイプ]。要求または応答としての接続タイプ。
   c) 仮想サーバ。ポリシーをバインドする SSL ブリッジロードバランシング仮想サーバ。
4. [続行] をクリックします。
5. バインドポイント **(Bind Point**) セクションで、次のいずれかの操作を行います。
   a) リストからポリシーバインディングを選択します。
   b)「バインドの追加」をクリックして、「ポリシーバインディング」スライダーにアクセスします。
      i. 既存のポリシーを選択するか、新しいポリシーを追加します。
      ii. バインドの詳細を入力し、「バインド」をクリックします。
6. [閉じる] をクリックします。

UDP によるビデオ最適化の設定

October 7, 2021

UDP 経由で QUIC ABR ビデオトラフィックを最適化するには、まずビデオ最適化機能を有効にします。設定が完了すると、アプライアンスは QUIC ベースの ABR ビデオトラフィックを検出し、アプライアンスに設定されている最適化ビットレートを適用します。

CLI を使用した **QUIC** のビデオ最適化の設定

UDP を介した QUIC ビデオトラフィックのビデオ最適化を設定するには、次のタスクを実行する必要があります。

1. ビデオの最適化を有効にします。
2. QUIC サービスを作成します。
3. QUIC 負荷分散仮想サーバーを作成します。
4. QUIC Web サービスをロード・バランシング仮想サーバーにバインドします。
5. QUIC ベースの UDP トラフィックをペーシングするためのビデオ最適化ポリシーを作成します。
6. 最適化ポリシーを QUIC ベースの負荷分散仮想サーバーにバインドします。

QUIC トラフィックのビデオ最適化の有効化

Citrix ADC アプライアンスでビデオトラフィックの検出、最適化、およびレポートを実行する場合は、ビデオ最適化機能を有効にして、最適化をオンに設定する必要があります。

> 注
>
> QUIC トラフィックにビデオ最適化を使用する場合は、負荷分散機能および AppFlow 機能を有効にする必要があります。

ビデオの最適化を有効にするには

コマンドプロンプトで、次のコマンドを入力します。

```
enable ns feature VideoOptimization
<!--NeedCopy-->
```

QUIC トラフィック用のサービスの作成

Citrix ADC アプライアンスは、負荷分散仮想サーバーに QUIC サービスを使用して、静的ルーティングモードで出力ルーターに接続します。

> 注
>
> 現在、ダイナミックルーティングはサポートされていません。

QUIC ビデオトラフィック用の負荷分散 **Web** サービスを作成するには

コマンドプロンプトで入力します。

```
add service <name> <router-IP> <serviceType> <port> -usip yes -
   useproxyport  [yes | no]
<!--NeedCopy-->
```

例:

```
add service svc-quic 10.102.29.200 QUIC 443  - usip yes  - useproxyport
   no

where IP address is the internet router address.
<!--NeedCopy-->
```


QUIC トラフィック用の負荷分散仮想サーバーの作成

Citrix ADC アプライアンスは、UDP を介した QUIC ビデオトラフィックを検出して最適化するために、負荷分散仮想サーバーを使用します。

QUIC ビデオトラフィック用の負荷分散仮想サーバーを作成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <ip> <port> -m MAC
<!--NeedCopy-->
```

例**:**

```
add lb vserver vs-quic QUIC * 443 -persistenceType NONE -m MAC -
    cltTimeout 120
<!--NeedCopy-->
```

負荷分散仮想サーバーへの **QUIC Web** サービスのバインド

QUIC トラフィック用の Web サービスと負荷分散仮想サーバーを作成したら、サービスを仮想サーバーにバインドする必要があります。

QUIC ビデオトラフィック用に **Web** サービスをロードバランシング仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver vs-quic svc-quic
<!--NeedCopy-->
```


QUIC ベースの **UDP** トラフィックに対するビデオ最適化ポリシーの作成

QUIC ベースの UDP トラフィックを最適化するには、最適化ペーシングポリシーとそのアクションを設定する必要があります。次に、QUIC ベースの負荷分散仮想サーバーにポリシーをバインドする必要があります。ポリシーごとに、アクションを最初に作成し、ポリシーに関連付けることができます。

最適化アクションを追加するには

コマンドプロンプトで入力します。

```
add videooptimization pacingaction <action Name> -rate <integer> [-
    comment <string>]
<!--NeedCopy-->
```

ここで、**rate** パラメータは、トラフィックを送信するレート（ペーシングレート）を Kbps 単位で指定します。

例**:**

```
set videooptimization parameter -QUICPacingRate 1000
<!--NeedCopy-->
```

ここで、1000 は、希望するペーシングレート（キロビット/秒）を表します。

最適化ポリシーを追加するには

コマンドプロンプトで入力します。

```
add videooptimization pacingpolicy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

例**:**

```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
    MyOptAct2000
<!--NeedCopy-->
```


QUIC 負荷分散仮想サーバーへの最適化ポリシーのバインド

UDP 接続で QUIC ビデオトラフィックを最適化するには、最適化ポリシーを QUIC ロードバランシング仮想サーバにバインドする必要があります。

最適化ポリシーを **QUIC** ロードバランシング仮想サーバーにバインドするには

コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name> -policyName <policy_name> -priority <
    positive_integer> -type (REQUEST)
<!--NeedCopy-->
```

> 注
>
> ペーシングポリシーは、要求時にのみ QUIC ロードバランシング仮想サーバーにバインドする必要があります。

例**:**

```
bind lb vserver vs-quic -policyName myOptPolicy2000 -priority 3400 -
    type  REQUEST
<!--NeedCopy-->
```

GUI を使用した **QUIC** のビデオ最適化の設定

GUI を使用してアプライアンスの機能を設定するには、次のタスクを実行する必要があります。

1. ビデオの最適化を有効にする
2. QUIC サーバーを構成する
3. QUIC サービスの設定
4. QUIC 負荷分散仮想サーバーの構成
5. QUIC Web サービスを負荷分散仮想サーバーにバインドする
6. 最適化ポリシーを作成します。
7. 最適化アクションを作成します。
8. 最適化ペーシングパラメータを設定します。
9. QUIC トラフィック用の仮想サーバーの負荷分散に最適化ポリシーをバインドします。

ビデオの最適化を有効にするには

1. Citrix ADC アプライアンスにログオンし、［システム］>［設定］に移動します。
2. 詳細ページで、[拡張機能の構成] リンクを選択します。

3. [高度な機能の設定] ページで、[ビデオの最適化] チェックボックスをオンにします。

QUIC サーバーを作成するには

1. Citrix ADC アプライアンスにログオンし、「トラフィック管理」>「負荷分散」>「サーバー」画面に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [サーバーの作成] ページで、次のパラメータを設定します。
   a) Name：QUIC サーバーの名前。
   b) IP アドレス。QUIC サーバーの IP アドレス
   c) トラフィックドメイン。サーバのドメイン名。
   d) 作成後に有効にします。サーバの初期状態。
   e) コメント。サーバーに関する簡単な情報。
4. [作成] をクリックします。

QUIC サービスを作成するには

1. Citrix ADC アプライアンスにログオンし、「トラフィック管理」>「負荷分散」>「サービス」画面に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [負荷分散サービス] ページで、次のパラメータを設定します。
   a) サービス名。QUIC サービスの名前。
   b) **IP** アドレス。QUIC サービスに割り当てられた IP アドレス。
   c) プロトコル。プロトコルを QUIC として選択します。
   d) ポート。Web サービスのポート番号。
4. [**OK**] をクリックして続行します。その後、その他のオプションのパラメータを設定できます。詳細については、「サービスの設定」を参照してください。
5. オプションのパラメータを設定したら、[**OK**] をクリックして [閉じる] をクリックします。

負荷分散仮想サーバーを作成するには

1. Citrix ADC アプライアンスにログオンし、「トラフィック管理」>「負荷分散」>「仮想サーバー」画面に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [負荷分散仮想サーバー] ページで、次のパラメータを設定します。
   a) **Name**：負荷分散仮想サーバの名前。
   b) プロトコル。QUIC 要求を送信するためにサービスが使用するプロトコル。
   c) IP アドレスタイプ。IP アドレスの種類:IPv4 または IPv6。
   d) **IP** アドレス。仮想サーバに割り当てられた IP 4 または IP6 IP アドレス。
   e) ポート。仮想サーバのポート番号。
4. [OK] をクリックして、他のオプションのパラメータの設定を続行します。詳細については、仮想サーバーの作成を参照してください。

負荷分散仮想サーバーを **QUIC** サービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. [サービスとサービスグループ] をクリックして、[負荷分散仮想サーバーサービスバインディング] 画面にアクセスします。
3. QUIC ベースの Web サービスを選択し、[バインド] をクリックします。
4. [完了] をクリックします。

負荷分散仮想サーバーを **QUIC** サービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. [サービスとサービスグループ] をクリックして、[負荷分散仮想サーバーサービスバインディング] 画面にアクセスします。
3. QUIC ベースの Web サービスを選択し、[バインド] をクリックします。
4. [完了] をクリックします。

QUIC トラフィックのビデオ最適化アクションを作成するには

1. Citrix ADC アプライアンスにログオンし、「構成」>「最適化」>「ビデオの 最適化」>「ペーシング」>「アクション」の順に選択します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [ビデオの最適化ペーシングアクションの作成] ページで、次のパラメータを設定します。
   a) **Name**：最適化アクションの名前。
   b) **ABR** 最適化レート（**Kbps**）。ABR ビデオトラフィックを送信するペーシングレート。ABR 最適化のデフォルトレートは 1000 Kbps です。最小値は 1 で、最大値は 2147483647 です。
   c) コメント。アクションの簡単な説明。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

QUIC トラフィックのビデオ最適化ポリシーを作成するには

1. Citrix ADC アプライアンスにログオンし、「構成」>「最適化」>「ビデオの 最適化」>「ペーシング」>「ポリシー」の順に選択します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [ビデオ最適化ペーシングポリシーの作成] ページで、次のパラメータを設定します。
   a) Name：最適化ポリシーの名前
   b) 式。ポリシーを実装するカスタム正規表現式。
   c) 操作。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
   d) UNDEF アクション。着信要求が最適化ポリシーと一致しない場合、未定義のイベント。
   e) コメント。ポリシーの簡単な説明。
   f) ログアクション。目的のログメッセージを作成する監査ログアクションを選択します。
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

ビデオ最適化ポリシーを **QUIC** 負荷分散仮想サーバーにバインドするには

1. 構成への Citrix ADC アプライアンスとナビゲートにログオンします > 最適化 > ビデオの 最適化。
2. [ビデオの最適化] ページで、[ビデオの最適化ペーシングポリシーマネージャ] リンクをクリックします。
3. [ビデオ最適化ポリシーマネージャ] ページで、次のパラメータを設定します。
   a) バインドポイント。要求処理中に最適化ポリシーを適用するポイント。注：ペーシングポリシーは、要求時にのみ QUIC 負荷分散仮想サーバーにバインドする必要があります。
   b) [接続タイプ]。要求または応答としての接続タイプ。
   c) 仮想サーバ。ポリシーをバインドする負荷分散仮想サーバー。
4. [続行] をクリックします。
5. バインドポイント **(Bind Point**) セクションで、次のいずれかの操作を行います。
   a) リストからポリシーを選択します。
   b)「バインドの追加」をクリックして、「ポリシーバインディング」スライダーにアクセスします。
      i. 既存のポリシーを選択するか、新しいポリシーを追加します。
      ii. バインドの詳細を入力し、「バインド」をクリックします。
6. [閉じる] をクリックします。

Citrix ADC の URL フィルタリング

October 7, 2021

URL フィルタリングは、URL に含まれる情報を使用して Web サイトをポリシーベースで制御できるようにします。この機能は、ネットワーク管理者がモバイルネットワーク上の悪意のある Web サイトへのユーザーアクセスを監視および制御するのに役立ちます。

管理者は、URL 分類機能または URL リスト機能を使用して URL フィルタリングポリシーを構成できます。

URL リスト。アプライアンスにインポートされた URL セット内の URL へのアクセスをブロックすることにより、ブラックリストに登録された Web サイトおよび Web ページへのアクセスを制御します。

URL 分類。事前に定義されたカテゴリのリストに基づいてトラフィックをフィルタリングすることにより、Web サイトや Web ページへのアクセスを制御します。

URL リスト

October 7, 2021

URL リスト機能を使用すると、カスタマイズされた URL リスト（最大 100 万エントリ）へのアクセスを制御できます。この機能は、仮想サーバーにバインドされた URL フィルタリングポリシーを適用して Web サイトをフィルタリングします。

管理者として、URL リストを Citrix ADC アプライアンスにインポートする必要があります。このインポートされたリストは、*URL* セットと呼ばれるポリシーデータセットとして内部的に格納されます。次に、アプライアンスは、着信 URL 要求に一意の高速 URL 照合アルゴリズムを適用します。着信 URL 要求がセット内のエントリと一致する場合、アプライアンスは関連付けられたポリシーアクションを適用してアクセスを制御します。

URL リストのタイプ

URL セット内の各エントリには、URL と、必要に応じてそのメタデータ (URL カテゴリ、カテゴリグループ、またはその他の関連データ) を含めることができます。メタデータを含む URL の場合、アプライアンスはメタデータを評価するポリシー式を使用します。詳細については、「URL セット」を参照してください。

カスタム **URL** リスト。最大 1,000,000 個の URL エントリのカスタマイズされた URL セットを作成し、それをテキストファイルとしてアプライアンスにインポートできます。リストには、メタデータの有無にかかわらず URL を含めることができます（URL カテゴリのようなものです）。Citrix ADC プラットフォームは、メタデータが存在するかどうかを自動的に検出します。また、インポートされたリストを安全に保存することもできます。詳細については、「URL セット」を参照してください。

URL リストをホストし、Citrix ADC アプライアンスを構成して、手動で操作しなくてもリストを定期的に更新できます。URL リストが更新されると、アプライアンスはポリシー式を使用して各着信 URL を評価し、許可、ブロック、リダイレクト、ユーザーに通知などのアクションを適用することで、メタデータとカテゴリを自動的に検出できます。

URL リストポリシー式

次の表に、着信トラフィックの評価に使用できる基本的な式を示します。アプライアンスに URL リストをインポートすると、*URL* セットと呼ばれます。

式	操作
`<URL expression>.URLSET_MATCHES_ANY (<URLSET>)`	URL が URL セット内のエントリと完全に一致する場合に TRUE と評価されます。
`<URL expression>. GET_URLSET_METADATA(<URLSET>)`	GET_URLSET_METADATA () 式は、URL が URL セット内の任意のパターンに完全に一致する場合に、関連するメタデータを返します。一致しない場合は、空の文字列が返されます。
`<URL expression>.GET_ URLSET_METADATA(<URLSET>).EQ(< METADATA>)`	一致するメタデータが<METADATA>と等しい場合は TRUE と評価されます。

式	操作
<URLexpression>.GET_URLSET_METADATA (<URLSET>).TYPECAST_LIST_T(‘,’).GET (0).EQ(<CATEGORY>)	一致するメタデータがカテゴリの先頭にある場合は TRUE と評価されます。このパターンは、メタデータ内の個別のフィールドをエンコードするために使用できますが、1st フィールドのみに一致します。
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	ホストパラメータと URL パラメータを結合します。このパラメータは、照合のため に<URL expression>として使用できます。

URL リストポリシーアクション

URL リストに一致する URL に対する最も一般的な強制アクションは、アクセスを制限することです。目的の URL リスト一致式と強制アクションを使用して URL リストポリシーを作成します。ポリシーグループの使用状況は、着信トラフィックタイプ（HTTP または HTTPS）とアプライアンスで設定された仮想サーバーによって異なります。HTTP トラフィックにはレスポンダーポリシーを使用するか、HTTPS トラフィックにはビデオ最適化ポリシーを使用できます。ポリシー内の式に一致する URL に適用するアクションを指定します。次の表に、使用可能なアクションを示します。

アクションの種類	ポリシー	説明
ALLOW	レスポンダー	リクエストにターゲット URL へのアクセスを許可します。
REDIRECT	レスポンダー	ターゲットとして指定された URL にリクエストをリダイレクトします。
DENY	レスポンダー	要求を拒否します。
RESET	レスポンダー、ビデオ最適化	接続をリセットします。
DROP	レスポンダー、ビデオ最適化	接続を切断します。

前提条件

URL リスト機能を設定するには、次のサーバが設定されていることを確認してください。

DNS 要求用の DNS サーバー

ホスト名の URL から URL セットをインポートする場合は、DNS サーバを設定する必要があります。

コマンドプロンプトで入力します。

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>) [-state (
    ENABLED | DISABLED )] [-type <type>] [-dnsProfileName <string>]
<!--NeedCopy-->
```

例:

```
add dns nameServer 10.140.50.5
<!--NeedCopy-->
```

カスタム URL リストのインポート

URL セットをインポートするには、URL セットのトピックを参照してください。

HTTP トラフィックの URL リストの設定

Citrix ADC アプライアンスは、HTTP トラフィックと HTTPS トラフィックをサポートします。HTTP トラフィック用のロードバランシング仮想サーバを設定し、URL リストポリシーをサーバにバインドするには、次の手順を実行します。

- URL リストアクションを追加します。
- URL リストポリシーを追加します。
- HTTP トラフィック用の HTTP ロードバランシング仮想サーバーの追加
- HTTP トラフィック用の HTTP ロードバランシング仮想サーバへの URL リストポリシーのバインド

URL リストアクションを追加するには

コマンドプロンプトで、次のように入力します。

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
    string>]
<!--NeedCopy-->
```


HTTP トラフィック用の **HTTP** ロードバランシング仮想サーバーを追加するには

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
   imeout <secs>]
<!--NeedCopy-->
```

例:

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout
   120
<!--NeedCopy-->
```

URL リストポリシーを **HTTP** ロードバランシング仮想サーバにバインドするには

コマンドプロンプトで、次のように入力します。

```
bind lb vserver <vServerName> -policyName <string> [-priority <
   positive_integer>]
<!--NeedCopy-->
```

HTTPS トラフィックの **URL** リストの設定

Citrix ADC アプライアンスは、HTTP トラフィックと HTTPS トラフィックをサポートします。HTTPS トラフィック用の SSL ブリッジロードバランシング仮想サーバを設定し、URL リストポリシーをサーバにバインドするには、次の手順を実行します。

- URL リストアクションを追加します。
- URL リストポリシーを追加します。
- HTTP トラフィック用の SSL ブリッジ負荷分散仮想サーバーの追加
- HTTP トラフィック用の SSL ブリッジロードバランシング仮想サーバへの URL リストポリシーのバインド

HTTPS トラフィックの **URL** リストポリシーを追加するには

コマンドプロンプトで入力します。


```
add videooptimization detectionpolicy <name> -rule <expression> -action
     <string> [-undefAction <string>] [-comment <string>] [-logAction <
    string>]
<!--NeedCopy-->
```

SSL ブリッジ負荷分散仮想サーバーを追加するには

コマンドプロンプトで次のように入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
    imeout <secs>]
<!--NeedCopy-->
```

例**:**

```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -
    cltTimeout 180
<!--NeedCopy-->
```

CLI を使用して **SSL** ブリッジロードバランシングで **URL** リストポリシーをバインドするには

コマンドプロンプトで次のように入力します。

```
bind lb vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

GUI を使用した **URL** リストの構成

GUI を使用すると、次のことが可能になります。

- URL リストをインポートします。
- URL リストを追加します。
- URL リストアクションを設定します。
- HTTP トラフィックの URL リストポリシーを設定します。
- HTTP トラフィック用の HTTP ロードバランシング仮想サーバを追加します。
- HTTPS トラフィック用の SSL ブリッジ負荷分散仮想サーバーを追加します。

- URL リストポリシーを HTTP ロードバランシング仮想サーバにバインドします。
- URL リストポリシーを SSL ブリッジロードバランシング仮想サーバにバインドします。

URL リストを読み込むには

1. ナビゲーションペインで、「**AppExpert** 」>「**URL** セット」の順に展開します。
2. 詳細ウィンドウで、[インポート] をクリックします。
3. [**URL** セットの構成**]** ページで、次のパラメータを設定します。
   a) **Name**：URL セットの名前。
   b) **URL**。URL セットにアクセスする場所の Web アドレス。
   c) 上書き。以前にインポートした URL セットを上書きします。
   d) 区切り文字。CSV ファイルレコードを区切る文字シーケンス。
   e) 行セパレータ。CSV ファイルで使用される行区切り文字。「/n」のように、1 文字の値を使用できます。
   f) **[**間隔**]**。URL セットが更新される直近の 15 分に切り捨てられた間隔（秒単位）。
   g) プライベートセット。URL セットのエクスポートを防止するオプション
   h) カナリアの **URL**。URL セットの内容が機密に保たれるかどうかをテストするための内部 URL。URL の最大長は 2047 文字です。
4. [作成] をクリックし、[閉じる] をクリックします。

URL リストを追加するには

1. ナビゲーションペインで、「**AppExpert** 」>「**URL** セット」の順に展開します。
2. 詳細ペインで、［**Add**］をクリックします。
3. [**URL** セットの作成**]** ページで、次のパラメータを設定します。
   a) **Name**：インポート時に指定された URL セットの名前。
   b) コメント。URL セットに関する簡単な説明。
4. ［作成］をクリックします。

URL リストアクションを設定するには

1. Citrix ADC アプライアンスにログオンし、［構成］タブページに移動します。
2. メニューペインで、[**AppExpert**] > [レスポンダー] > [アクション] に移動します。
3. 詳細ペインで、［**Add**］をクリックします。
4. [レスポンダーアクションの作成**]** ページで、次のパラメータを設定します。
   a) **Name**：URL リストポリシーアクションの名前。
   b) タイプ。アクションタイプを選択します。
   c) 式。式エディタを使用して、ポリシー式を作成します。
   d) コメント。ポリシーアクションに関する簡単な説明。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

URL リストポリシーを設定するには

1. ナビゲーションペインで、[**AppExpert**] > [レスポンダー] > [ポリシー] の順に展開します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [レスポンダーポリシーの作成] ページで、次のパラメーターを設定します。
   a) **Name**：URL リストポリシーアクションの名前。
   b) 操作。ポリシーに関連付ける [URL List] アクションを選択します。
   c) ログアクション。ログアクションを選択します。
   d) **AppFlow**。AppFlow アクションを選択します。
   e) 式。式エディタを使用して、ポリシー式を作成します。
   f) コメント。ポリシーに関する簡単な説明。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTP 負荷分散仮想サーバーを追加するには

1.「トラフィック管理」>「負荷分散」>「仮想サーバー」ページに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。
   a) **Name**：負荷分散仮想サーバの名前。
   b) プロトコル。プロトコルの種類を HTTP として選択します。
   c) **IP** アドレスタイプ。IP アドレス可能なタイプ。
   d) **IP** アドレス。仮想サーバに割り当てられた IP 4 または IP6 IP アドレス。
   e) ポート。仮想サーバのポート番号。
4. [**OK]** をクリックして、他のオプションのパラメータの設定を続行します。詳細は、「仮想サーバーの作成」を参照してください。

URL リストポリシーを **HTTP** ロードバランシング仮想サーバにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ペインで、負荷分散仮想サーバーを選択し、[**Edit**] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. [ポリシー] セクションで、次のパラメータを設定します。
   a) ポリシーを選択します。ドロップダウンリストから URL 分類ポリシーを選択します。
   b) タイプを選択します。ポリシーの種類を [要求] として選択します。
6. [続行] をクリックします。
7. [ポリシー] ページで、リストから [URL リスト] ポリシーを選択し、[選択] をクリックします。
8.「ポリシー」スライダーで、「バインドして 閉じる」をクリックします。

HTTPS トラフィックの **URL** リストポリシーを追加するには

1. Citrix ADC アプライアンスにログオンし、「構成」>「最適化」>「ビデオの 最適化」>「検出」の順に選択します。
2. [検出] ページで、[ビデオ最適化検出ポリシー] リンクをクリックします。
3. [ビデオ最適化検出ポリシー] ページで、[追加] をクリックします。
4. [ビデオ最適化検出ポリシーの作成] ページで、次のパラメータを設定します。
   a) **Name**：最適化ポリシーの名前
   b) 式。カスタム式を使用してポリシーを設定します。
   c) 操作。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
   d) **UNDEF** アクション。着信要求が最適化ポリシーと一致しない場合、未定義のイベント。
   e) コメント。ポリシーの簡単な説明。
   f) ログアクション。ログ・メッセージに対して実行するアクションを指定する監査ログ・アクションを選択します。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTPS トラフィック用の **SSL** ブリッジ負荷分散仮想サーバーを追加するには

1.「トラフィック管理」>「負荷分散」>「仮想サーバー」ページに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [負荷分散仮想サーバー] 画面で、次のパラメータを設定します。
   a) **Name**：負荷分散仮想サーバの名前。
   b) プロトコル。SSL ブリッジとしてプロトコルタイプを選択します。
   c) **IP** アドレスタイプ。IP アドレスの種類:IPv4 または IPv6。
   d) **IP** アドレス。仮想サーバに割り当てられた IPv4 または IP6vIP アドレス。
   e) ポート。仮想サーバのポート番号。
4. [**OK]** をクリックして、他のオプションのパラメータの設定を続行します。詳細については、「仮想サーバーの作成」を参照してください。

URL リストポリシーを **SSL** ブリッジロードバランシング仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ペインで、SSL ブリッジロードバランシング仮想サーバを選択し、[**Edit**] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. 次のパラメータを設定します。
   a) ポリシーを選択します。ドロップダウンリストから [ビデオ検出ポリシー] を選択します。
   b) タイプを選択します。ポリシーの種類を [要求] として選択します。
6. [続行] をクリックします。
7. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

監査ログ・メッセージの構成

監査ログを使用すると、URL リストプロセスのどのフェーズでも条件や状況を確認できます。Citrix ADC アプライアンスが受信 URL を受信すると、レスポンダーポリシーに URL セットの詳細ポリシー式がある場合、監査ログ機能によって URL セットの情報が収集され、監査ログで許可されるターゲットのログメッセージとして詳細が保存されます。

ログメッセージには、次の情報が含まれています。

1. タイムスタンプ。
2. ログメッセージのタイプ。
3. 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)。
4. URL セット名、ポリシーアクション、URL などのログメッセージ情報。

URL リスト機能の監査ログを設定するには、次のタスクを実行する必要があります。

1. 監査ログを有効化。
2. 監査ログメッセージの作成アクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ」を参照してください。

URL リストのセマンティクス

次の表に、URL の一致パターンと、URL リスト内の URL と受信リクエスト URL の照合方法を示します。たとえば、www.example.com/bar というパターンは、www.example.com/bar にある 1 ページのみに一致します。URL が 'www.example.com/bar' で始まるすべてのページを一致させるには、URL の末尾にアスタリスク (*) を追加します。

説明	URL パターン	一致しました	一致しません
サブドメインの一致	ドメイン.com	domain.com; www.domain.com; sub.one.domain.com	yourdomain.com; wwwdomain.com
URL の一致、正確なパス	ドメイン.com/example/bar/index.html	ドメイン.com/example/bar/index.html; www.domain.com/example/bar/index.html; s.domain.com/example/bar/index.html	wwwdomaincom/example/bar/index.html; domain.com/example/bar/index.html/
URL の一致、正確なパス	ドメイン.com/example/bar/in	ドメイン.com/example/bar/in htmlkey=value; www.domain.com/exar s.domain.com/example	wwwdomaincom/example/bar/index do- main.com/example/bar/index.html/

説明	URL パターン	一致しました	一致しません
URL マッチング、サブパスマッチング	ドメイン.com/example/bar/	domain.com/example/bar/; www.domain.com/example/bar/index.html; domain.com/example/bar/index.html/one.jpg	domain.com/example/bar/index.html; ンデックス.html

URL の分類

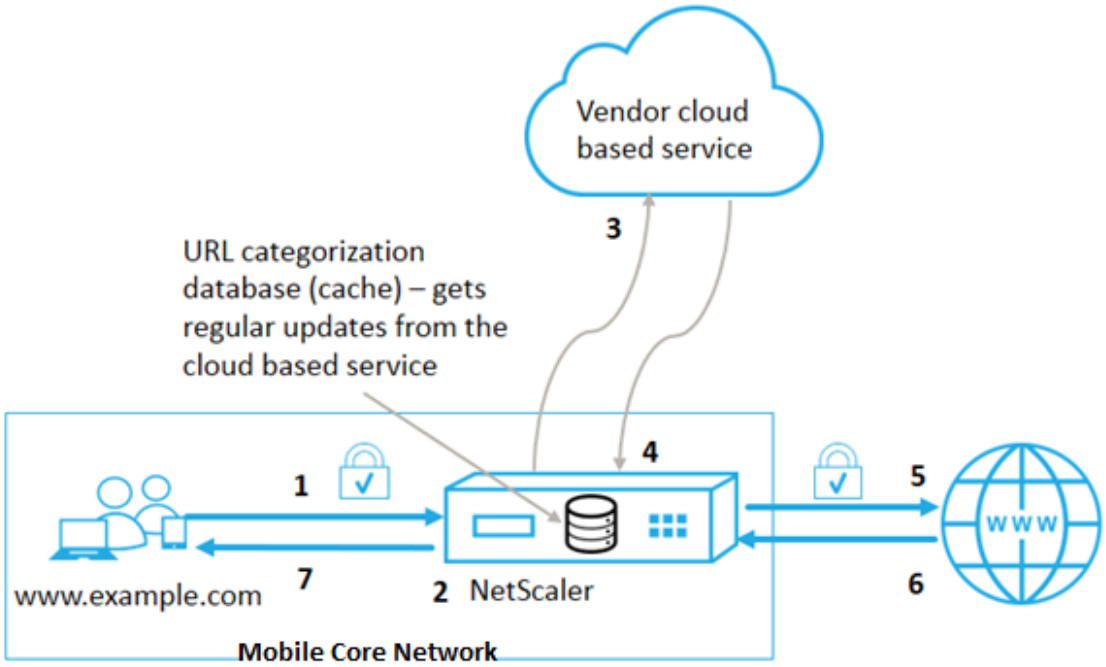
October 7, 2021

URL の分類は、特定の Web サイトおよび Web サイトのカテゴリへのユーザーアクセスを制限します。NetSTARと共同でサブスクライブされたサービスとして、この機能を使用すると、企業のお客様は、商用分類データベースを使用して Web トラフィックをフィルタリングできます。このNetSTAR データベースには、ソーシャルネットワーキング、ギャンブル、アダルトコンテンツ、新しいメディア、ショッピングなど、さまざまなカテゴリに分類された URL が多数あります。分類に加えて、各 URL には、サイトの履歴リスクプロファイルに基づいて最新のレピュテーションスコアがあります。NetSTARデータを使用して、カテゴリ、カテゴリグループ（テロ、違法薬物など）、またはサイトレピュテーションスコアに基づいて高度なポリシーを設定することにより、トラフィックをフィルタリングできます。

たとえば、マルウェアに感染したサイトなど、危険なサイトへのアクセスをブロックしたり、アダルトコンテンツやエンターテイメントストリーミングメディアへのアクセスを選択的に制限したりできます。

URL 分類の仕組み

次の図は、Citrix ADC URL フィルタリングサービスを商用 URL 分類データベースおよびクラウドサービスと統合して、頻繁な更新を行う方法を示しています。



コンポーネントは次のように相互作用します。

1. クライアントは、インターネットにバインドされた URL 要求を送信します。
2. Citrix ADC ポリシーは、URL 分類データベースから取得した分類の詳細（カテゴリ、カテゴリグループ、サイト評価スコアなど）に基づいてリクエストを評価しようとします。データベースがカテゴリの詳細を返す場合、プロセスは手順 5 にジャンプします。
3. データベースが分類の詳細を返さない場合、要求は URL 分類ベンダーが管理するクラウドベースの検索サービスに送信されます。ただし、アプライアンスは応答を待機しません。代わりに、URL を [未分類] としてマークし、手順 5 に進みます。ただし、引き続きクラウドクエリのフィードバックを監視し、それを使用してキャッシュを更新し、今後の要求がクラウド検索の恩恵を受けることができるようにします。
4. Citrix ADC アプライアンスは、クラウドベースのサービスから URL カテゴリの詳細（カテゴリ、カテゴリグループ、レピュテーションスコア）を受け取り、クラウドキャッシュに格納します。
5. ポリシーで URL が許可されている場合、リクエストはオリジンサーバーに送信されます。それ以外の場合、アプライアンスはリクエストをドロップまたはリダイレクトするか、カスタム HTML ページを使用して応答します。
6. オリジンサーバーは、要求されたデータを Citrix ADC アプライアンスに応答します。
7. アプライアンスは応答をクライアントに送信します。

URL フィルタリング機能を使用して、政府が発行した安全なインターネット使用規制に違反するサイトを検出し、これらのサイトをブロックするポリシーを実装できます。成人向けコンテンツ、ストリーミングメディア、ソーシャルネットワーキングをホストしているサイトで、子供にとって安全でないと判断されたり、違法であると認められるサイト。

前提条件

この機能は、基本的な CBM ライセンスと CBM Premium ライセンスを購入すると Telco プラットフォームで動作します。その他の Citrix ADC プラットフォームでは、CNS Premium ライセンスの購入と連動します。

注: アプライアンスには、Basic CBM ライセンスと CBM Premium ライセンスに加えて、URL Threat Intelligence ライセンスと 1 年または 3 年間のサブスクリプションサービスが必要です。機能を有効にして設定する前に、次のライセンスをインストールする必要があります。

Telco プラットフォームのライセンスサポート:

- **CBM_TXXX_SERVER_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **CNS_WEBF_SSERVER_Retail.lic**

ここで、XXX はスループットです（例: Citrix ADC T1000）。

その他の Citrix ADC プラットフォームのライセンスサポート:

- **CNS_XXX_SERVER_PLT_Retail.lic**

ここで、XXX はスループットです。

URL 分類ポリシー式

次の表に、着信 URL を識別するためのさまざまな URL 分類ポリシー式を示し、設定済みのアクションを適用します。

式	操作
<text>. URL_CATEGORIZE (<min_reputation>, <max_reputation>)	URL_CATEGORY オブジェクトを返します。レピュテーションスコアは 1 から 4 までの数字です。オブジェクトを取得するには、すべてのレピュテーションスコアは<min reputation>に 0.0 を使用します。<max_reputation>。もしあれば <min_reputation> が 0 より大きい場合、返されるオブジェクトにはレピュテーションが以下のカテゴリが含まれていません <min_reputation>。もしあれば <max_reputation> が 0 より大きい場合、返されるオブジェクトにはレピュテーションが次のカテゴリが含まれていません。<max_reputation>。カテゴリがタイムリーに解決に失敗した場合は、undef 値が返されます。
<url_category>。CATEGORY	このオブジェクトのカテゴリ文字列を返します。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「未分類」になります。

式	操作
<url_category>。GROUP	オブジェクトのカテゴリグループを識別する文字列を返します。これは、カテゴリのより高いレベルのグループです。これは、URL カテゴリに関する詳細情報を必要としない操作に役立ちます。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「未分類」になります。
<url_category>。REPUTATION	レピュテーションスコアを 1 ～4 の数値として返します。4 は最もリスクの高いレピュテーションを示します。カテゴリが「未分類」の場合、評価値は 2 です。

ポリシー表現の例

ポリシー	ポリシー表現
「検索エンジン」カテゴリにある URL に対するリクエストを選択するポリシー	add responder policy p1 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY.EQ(“Search Engine”)
アダルトカテゴリグループ内の URL に対するリクエストを選択するためのポリシー	add responder policy p1 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO GROUP.EQ(“Adult”)’
評価スコアが 4 の検索エンジンの URL に対するリクエストを選択するポリシー。	add responder policy p2 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY.EQ(“Search Engine”)’
検索エンジンとショッピング URL のリクエストを選択するポリシー	add policy patset good_categories; bind policy good_categories “Search Engine”; bind policy good_categories “Shopping”; add responder policy p3 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY .EQUALS_ANY(“good_categories”)
評価スコアが 4 の検索エンジンの URL に対するリクエストを選択するポリシー。	add responder policy p5 ‘CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(4,0). CATEGORY.EQ(“Search Engine”)

URL 分類ポリシーのアクション

URL フィルタリングポリシーは、トラフィックを評価して、特定のカテゴリに属する要求を識別します。次の表に、URL フィルタリングポリシーに割り当てることができるアクションを示します。

ポリシー・アクション	ポリシーグループ	説明
ALLOW	レスポンダー	受信要求がターゲット URL にアクセスすることを許可する
REDIRECT	レスポンダー	着信要求をターゲットとして指定された URL にリダイレクトします。
DENY	レスポンダー	着信要求を拒否します。
RESET	レスポンダー、ビデオ最適化	接続をリセットします。
DROP	レスポンダー、ビデオ最適化	接続をドロップします。

> 注
>
> 暗号化されたトラフィックの場合、VideoOptimization ポリシーには URL フィルタリングアクションを実装するアクションが含まれます。

URL 分類の設定

URL の分類を設定するには、まず URL フィルタリング機能を有効にします。次に、HTTP および HTTPS トラフィック用のキャッシュメモリの制限、分類ポリシー、および仮想サーバを設定する必要があります。CLI を使用した URL 分類の設定

Citrix ADC アプライアンスで CLI 構成の URL 分類を使用するには、次の手順を実行します。

- URL 分類を設定します。
  - URL フィルタリング機能を有効にします。
  - キャッシュメモリを制限するように共有メモリを構成します。
  - URL 分類パラメータを設定します。
- HTTP トラフィックの URL 分類を設定します。
  - URL 分類アクションを追加します。
  - URL 分類ポリシーを追加します。
  - HTTP トラフィック用の負荷分散仮想サーバーを追加します。
  - URL 分類ポリシーをロードバランシング仮想サーバにバインドします。
- HTTPS トラフィックの URL 分類を設定します。
  - URL 分類ポリシーを追加します。
  - SSL ブリッジ負荷分散仮想サーバーを追加します。

- URL 分類ポリシーをロードバランシング仮想サーバにバインドします。

URL 分類の設定

この機能を設定するには、URL 分類機能を有効にし、フィルタリングパラメータを設定し、共有メモリ制限を設定する必要があります。

URL フィルタリング機能を有効にするには

コマンドプロンプトで入力します。

```
enable ns feature URLFiltering VideoOptimization Responder IC SSL AppFlow
```

共有メモリ制限を構成するには

コマンドプロンプトで入力します。

```
1 set cache parameter [-memLimit <megaBytes>]
2 <!--NeedCopy-->
```

ここで、MemLimit はキャッシュのメモリ制限です。

例**:**

```
set cache parameter -memLimit 10
```

URL 分類パラメータを設定するには

コマンドプロンプトで入力します。

```
1 set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>]
     [-TimeOfDayToUpdateDB <HH:MM>]
2 <!--NeedCopy-->
```

***** 例:

```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
03:00
```


HTTP トラフィックの URL 分類の設定

HTTP トラフィックの URL 分類機能を設定するには、ロードバランシング仮想サーバを設定し、URL 分類ポリシーを追加して、ポリシーを仮想サーバにバインドする必要があります。これにより、仮想サーバは HTTP トラフィックを受信し、ポリシー評価に基づいてフィルタリングアクションを割り当てます。

HTTP トラフィックの URL 分類アクションを追加するには

コマンドプロンプトで入力します。

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment <string
>] [-responseStatusCode <positive_integer>] [-reasonPhrase <string>]
```

例:

```
add responder action act_url_categorize respondwith "\"HTTP/1.1 200 OK\r\n\
r\n\" + HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).CATEGORY
 + \"\n\""
```

HTTP トラフィックの URL 分類ポリシーを追加するには

コマンドプロンプトで入力します。

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment <
string>] [-logAction <string>] [-appflowAction <string>]
```

例:

```
add responder policy pol_url_categorize_http "HTTP.REQ.HOSTNAME.APPEND(
HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"Adult\")|| HTTP.REQ.HOSTNAME.
APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"Gambling\")"RESET
```

HTTP 負荷分散仮想サーバーを追加するには

HTTP トラフィック用の仮想サーバーが構成されていない場合は、コマンドプロンプトで次のように入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-clt Timeout
<secs>]
```

例:

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout 120
```

負荷分散仮想サーバーで URL 分類ポリシーをバインドするには

コマンドプロンプトで入力します。


```
bind lb vserver <name> -policyName <string> [-priority <positive_integer>]
```

例:

```
bind lb vserver vsrv-HTTP -policyName pol_url_categorize_http -priority 10
-gotoPriorityExpression END -type REQUEST
```

HTTPS トラフィックの URL 分類の設定

HTTPS トラフィックの URL 分類機能を設定するには、SSL ブリッジロードバランシング仮想サーバを設定し、URL 分類ポリシーを追加して、ポリシーを SSL ブリッジ仮想サーバにバインドする必要があります。これにより、サーバは HTTPS トラフィックを受信し、ポリシー評価に基づいてフィルタリングアクションを割り当てます。

HTTPS トラフィックの URL 分類ポリシーを追加するには

コマンドプロンプトで入力します。

```
add videooptimization detectionpolicy <name> -rule <expression> -action <
string> [-undefAction <string>] [-comment <string>] [-logAction <string>]
```

例:

```
add videooptimization detectionpolicy pol_url_categorize_https_block_adult -
rule ”CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(0,0).CATEGORY.EQ(”Adult”)' -
action RESET
```

SSL ブリッジ負荷分散仮想サーバーを追加するには

コマンドプロンプトで入力します。

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT imeout
<secs>]
```

例:

```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -cltTimeout
 180
```

分類ポリシーを SSL ブリッジ仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> -policyName <string> [-priority <positive_integer>]
```

例:

```
bind lb vserver vsrv-HTTPS -policyName pol_url_categorize_https_block_adult
 -priority 20 -type REQUEST
```


GUI を使用した URL 分類の設定

GUI を使用すると、次のことが可能になります。

- URL 分類機能を有効にします。
- HTTP トラフィックの URL 分類アクションを追加します。
- HTTP トラフィックの URL 分類ポリシーを追加します。
- HTTPS トラフィックの URL 分類ポリシーを追加します。
- HTTP トラフィック用の負荷分散仮想サーバーを追加します。
- HTTPS トラフィック用の SSL ブリッジロードバランシング仮想サーバを追加します。
- URL 分類ポリシーをロードバランシング仮想サーバにバインドします。
- URL 分類ポリシーを SSL ブリッジ負荷分散仮想サーバーにバインドします。
- 共有メモリ制限を設定します。
- URL 分類パラメータを設定します。

URL 分類を有効にするには

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] をクリックします。
2. [設定] ページで、[拡張機能の構成] リンクをクリックします。
3. [高度な機能の構成] ページで、[**URL** フィルタリング] チェックボックスをオンにします。
4. [**OK]** をクリックして [閉じる] をクリックします。

URL 分類アクションを追加するには

1. ナビゲーションペインで、[**AppExpert**] > [レスポンダー] > [アクション] の順に展開します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [レスポンダーアクションの作成**]** ページで、次のパラメータを設定します。
   a) **Name**：URL 分類ポリシーアクションの名前。
   b) タイプ。アクションタイプを選択します。
   c) 式。式エディタを使用して、ポリシー式を作成します。
   d) コメント。ポリシー・アクションの簡単な説明。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTP トラフィックの URL 分類ポリシーを追加するには

1. ナビゲーションペインで、[**AppExpert**] > [レスポンダー] > [ポリシー] の順に展開します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. [レスポンダーポリシーの作成] ページで、次のパラメーターを設定します。
   a) **Name**：URL 分類ポリシーアクションの名前。
   b) 操作。ポリシーに関連付ける URL 分類アクションを選択します。
   c) ログアクション。ログアクションを選択します。

d) **AppFlow**。AppFlow アクションを選択します。
e) 式。式エディタを使用して、ポリシー式を作成します。
f) コメント。ポリシーアクションに関する簡単な説明。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTPS トラフィックの分類ポリシーを追加するには

1. Citrix ADC アプライアンスにログオンし、「構成」>「最適化」>「ビデオの 最適化」>「検出」の順に選択します。
2. [検出] ページで、[ビデオ最適化検出ポリシー] リンクをクリックします。
3. [ビデオ最適化検出ポリシー] ページで、[追加] をクリックします。
4. [ビデオ最適化検出ポリシーの作成] ページで、次のパラメータを設定します。
   a) **Name**：最適化ポリシーの名前
   b) 式。カスタム式を使用してポリシーを設定します。
   c) 操作。着信ビデオトラフィックを処理するポリシーに関連付けられた最適化アクション。
   d) **UNDEF** アクション。着信要求が最適化ポリシーと一致しない場合、未定義のイベント。
   e) コメント。ポリシーに関する簡単な説明。
   f) アクションをログに記録します。ログ・メッセージに対して実行するアクションを指定する監査ログ・アクションを選択します。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

HTTP トラフィック用の負荷分散仮想サーバーを追加するには

1. 「トラフィック管理」>「負荷分散」>「仮想サーバー」ページに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [ロードバランシング仮想サーバー] ページで、次のパラメータを設定します。
   a) **Name**：負荷分散仮想サーバの名前。
   b) プロトコル。プロトコルの種類を HTTP として選択します。
   c) **IP** アドレスタイプ。IPv4 または IPv6 です。
   d) **IP** アドレス。IPv4 または IPv6、仮想サーバーに割り当てられた VIP アドレス。
   e) ポート。仮想サーバのポート番号。
4. [**OK]** をクリックして、他のオプションのパラメータの設定を続行します。
5. **[**作成**]** して **[**閉じる**]** をクリックします。

SSL ブリッジ負荷分散仮想サーバーを追加するには

1. 「トラフィック管理」>「負荷分散」>「仮想サーバー」ページに移動します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. [負荷分散仮想サーバー] ページで、次のパラメータを設定します。
   a) **Name**：負荷分散仮想サーバの名前。

b) プロトコル。SSL ブリッジとしてプロトコルタイプを選択します。
c) **IP** アドレスタイプ。IP アドレス可能なタイプ。
d) **IP** アドレス。仮想サーバに割り当てられた IP 4 または IP6 IP アドレス。
e) ポート。仮想サーバのポート番号。

4. [**OK]** を選択して、その他のオプションパラメータの設定を続行します。
5. [作成] をクリックし、[閉じる] をクリックします。

URL 分類ポリシーを **HTTP** 負荷分散仮想サーバーにバインドするには

1. 「トラフィック管理」>「負荷分散」>「仮想サーバー」ページに移動します。
2. 詳細ペインで、負荷分散仮想サーバーを選択し、[**Edit**] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. 次のパラメータを設定します。
   a) **[**ポリシー**]** を選択します。ドロップダウンリストから [URL 分類ポリシー] を選択します。
   b) タイプを選択します。ポリシーの種類を [要求] として選択します。
6. [続行] をクリックします。
7. リストから URL 分類ポリシーを選択し、[**Close**] をクリックします。

分類ポリシーを **SSL** ブリッジ負荷分散仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] 画面に移動します。
2. 詳細ペインで、SSL ブリッジロードバランシング仮想サーバを選択し、[**Edit**] をクリックします。
3. [詳細設定] セクションで、[ポリシー] をクリックします。
4. [ポリシー] セクションで、[**+**] アイコンをクリックして [ポリシー] スライダにアクセスします。
5. [ポリシー] セクションで、次のパラメータを設定します。
   a) **[**ポリシー**]** を選択します。ドロップダウンリストから [ビデオ検出ポリシー] を選択します。
   b) タイプを選択します。ポリシーの種類を [要求] として選択します。
6. [続行] をクリックします。
7. リストからビデオ検出ポリシーを選択し、[閉じる] をクリックします。

共有メモリ制限を構成するには

1. アプライアンスにサインオンし、[最適化] > [統合キャッシュ] に移動します。
2. 詳細ウィンドウで、[キャッシュ設定の変更] リンクをクリックします。
3. [キャッシュのグローバル設定**]** ページで、次のパラメータを設定します。
   a) メモリ使用制限 **(MB)**。
   b) アクティブなメモリ使用量の制限。
   c) ヘッダー経由。
   d) キャッシュされるポストボディの最大長さ

e) グローバル未定義の結果アクション
f) **HA** オブジェクトの持続の有効化
g) キャッシュされたオブジェクトが永続的であることの確認
h) プリフェッチ

4. [**OK]** をクリックして [閉じる] をクリックします。

URL 分類パラメータを設定するには

1. アプライアンスにサインインし、[セキュリティ] に移動します。
2. 詳細ウィンドウで、[**URL** フィルタ設定の変更] リンクをクリックします。
3. 「**URL** フィルタ・パラメータの設定」ページで、次のパラメータを設定します。
   a) DB 更新間隔 (時間)。URL データベース更新間のフィルタリング時間。最小値:0、最大値:720。
   b) DB を更新する時刻。URL データベースを更新するためのフィルタリング時間。
4. [**OK**] をクリックして閉じます。

監査ログ・メッセージの構成

Citrix ADC アプライアンスが着信 URL を受信すると、レスポンダーポリシーに URL フィルタリング式がある場合、監査ログ機能は分類情報を収集し、構成されているすべてのターゲット監査ログサーバーにログメッセージとして表示します。情報がログに記録されます。

- 送信元 IP アドレス（要求を行ったクライアントの IP アドレス)。
- 宛先 IP アドレス（要求されたサーバの IP アドレス)。
- スキーマ、ホスト、およびドメイン名を含む要求された URL（`http://www.example.com`)。
- URL フィルタリングフレームワークが返す URL カテゴリ。
- URL フィルタリングフレームワークが返した URL カテゴリグループ。
- URL フィルタリングフレームワークが返した URL レピュテーション番号。
- URL 分類ポリシーによって実行された監査ログアクション。

URL リスト機能の監査ログを設定するには、次の作業を完了する必要があります。

1. 監査ログを有効化。
2. 監査ログメッセージの作成アクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ 」トピックを参照してください。

SYSLOG メッセージを使用した障害エラーの保存

URL フィルタリングプロセスのどの段階でも、システムレベルの障害が発生した場合、Citrix ADC アプライアンスは監査ログメカニズムを使用してログを ns.log ファイルに保存します。エラーは、SYSLOG 形式のテキストメッセージとして保存されるため、管理者は後でイベントの発生を時系列で表示できます。これらのログは、アーカイブのために外部 SYSLOG サーバにも送信されます。詳細については、CTX229399 を参照してください。

たとえば、URL フィルタリング SDK を初期化するときにエラーが発生した場合、エラーメッセージは次のメッセージ形式で格納されます。

Oct 3 15:43:40 <local0.err> ns URLFiltering[1349]: NetStar SDK の初期化中にエラーが発生しました (SDK error=-1)。(ステータス =1)。

Citrix ADC アプライアンスは、4 つの異なる障害カテゴリにエラーメッセージを格納します。

- ダウンロードに失敗しました。分類データベースをダウンロードしようとしたときにエラーが発生した場合。
- 統合の失敗。既存の分類データベースに更新を統合するときにエラーが発生した場合。
- 初期化に失敗しました。URL 分類機能の初期化時にエラーが発生した場合は、分類パラメータを設定するか、分類サービスを終了します。
- 取得に失敗しました。アプライアンスがリクエストの分類の詳細を取得するときにエラーが発生した場合。

URL レピュテーションスコア

URL 分類機能は、ポリシーベースの制御を提供し、ブラックリスト URL を制限します。URL カテゴリ、レピュテーションスコア、URL カテゴリとレピュテーションスコアに基づいて Web サイトへのアクセスを制御できます。ネットワーク管理者は、危険性が高い Web サイトにアクセスするユーザーを監視する場合、URL レピュテーションスコアにバインドされたレスポンダーポリシーを使用して、そのような危険な Web サイトをブロックできます。

着信 URL 要求を受信すると、アプライアンスは URL 分類データベースからカテゴリとレピュテーションスコアを取得します。データベースから返されたレピュテーションスコアに基づいて、アプライアンスはウェブサイトにレピュテーションレーティングを割り当てます。値の範囲は 1 ～4 です。4 は、次の表に示すように、最もリスクのある Web サイトのタイプです。

URL レピュテーション評価	評価コメント
1	清潔なサイト。
2	不明なサイトです。
3	潜在的に危険な、または危険なサイトに所属している。
4	悪意のあるサイト。

よくある質問

October 7, 2021

このセクションでは、次の **Citrix ADC** 機能に関する **FAQ** について説明します

- 管理パーティション

- AppFlow
- Call Home
- クラスタリング
- 接続管理
- コンテンツスイッチ
- デバッグ
- ハードウェア
- 高可用性
- 統合キャッシング
- インストール、アップグレード、ダウングレード
- 負荷分散
- NetScaler GUI
- SSL

管理パーティション

October 7, 2021

パーティションの **Citrix ADC** 構成ファイルはどこで入手できますか**?**

デフォルト・パーティションの構成ファイル（*ns.conf*）は、*/nsconfig* ディレクトリにあります。管理パーティションの場合、ファイルは **/nsconfig/partitions/ <partitionName> ディレクトリにあります。

パーティション分割された **Citrix ADC** アプライアンスで統合キャッシュを構成するにはどうすればよいですか**?**

> 注
>
> 管理パーティションの統合キャッシュは、NetScaler11.0 以降でサポートされています。

パーティション分割された Citrix ADC で統合キャッシュ（IC）を構成するには、デフォルトパーティションに IC メモリを定義した後、スーパーユーザーは各管理パーティションに IC メモリを構成して、すべての管理パーティションに割り当てられる IC メモリの合計がデフォルトパーティションに定義されている IC メモリを超えないようにすることができます。admin パーティション用に構成されていないメモリは、デフォルトパーティションで引き続き使用できます。

たとえば、2 つの管理パーティションを持つ Citrix ADC アプライアンスのデフォルトパーティションに 10GB の IC メモリが割り当てられており、2 つの管理パーティションの IC メモリ割り当ては次のようになります。

- パーティション 1：4 GB
- パーティション 2：3 GB

次に、デフォルトのパーティションには 10-(4 + 3) = 3 GB の IC メモリが使用可能です。

> 注
>
> すべての IC メモリが admin パーティションによって使用されている場合、デフォルトパーティションには IC メモリを使用できません。

管理パーティションの **L2** と **L3** パラメータのスコープは何ですか**?**

> 注
>
> - NetScaler11.0 以降に適用されます。
> - デフォルト以外のパーティションで ARP を機能させるには、「setl2param」コマンドで「proxyArp」パラメータを有効にする必要があります。

パーティション分割された Citrix ADC アプライアンスでは、L2 および L3 パラメーターの更新範囲は次のとおりです。

- 「set L2Param」コマンドを使用して設定された L2 パラメータの場合、次のパラメータはデフォルトパーティションからのみ更新でき、その値はすべての管理パーティションに適用されます。

  最大ブリッジ衝突、bdg 設定、garpOnVridIntf、ガープ応答、プロキシ Arp、フェイルオーバー時にインターフェイスをリセット、トラフィックはスキップされます。

  他の L2 パラメータは、特定の管理パーティションで更新でき、それらの値はそれらのパーティションに対してローカルです。

- 「set L3Param」コマンドを使用して設定された L3 パラメータの場合、特定の管理パーティション内のすべてのパラメータを更新でき、それらの値はそれらのパーティションに対してローカルです。同様に、デフォルトパーティションで更新される値は、デフォルトパーティションにのみ適用されます。

管理パーティションで動的ルーティングを有効にする方法は**?**

> 注
>
> 管理パーティションでの動的ルーティングは、NetScaler11.0 以降でサポートされています。

ダイナミックルーティング（OSPF、RIP、BGP、ISIS、BGP+）がデフォルトパーティションでデフォルトで有効になっている場合、管理パーティションでは次のコマンドを使用して有効にする必要があります。

```
> set L3Param -dynamicRouting ENABLED
```

> 注
>
> 最大 63 のパーティションで動的ルーティングを実行できます（62 の管理パーティションと 1 つのデフォルトパーティション）。

管理パーティションで動的ルーティングを有効にすると、仮想ルーター（VR）が作成されます。

- 各 VR は独自の vlan0 を保持しており、vlan0_ と表示されます <partition-name>。

- ZebOS に公開されているバインドされていないすべての IP アドレスは vlan0 にバインドされます。
- デフォルト VR（デフォルトパーティションの）には、設定されているすべての VR が表示されます。
- デフォルト VR には、これらの VR にバインドされている VLAN が表示されます（デフォルト VLAN を除く）。

パーティションのログはどこで確認できますか?

Citrix ADC ログはパーティション固有ではありません。すべてのパーティションのログエントリは、*/var/log/* ディレクトリに格納する必要があります。

管理パーティションの監査ログを取得するにはどうすればよいですか

パーティション分割された Citrix ADC では、特定のパーティションに対して特定のログサーバーを使用することはできません。デフォルトパーティションで定義されたサーバは、すべての管理パーティションに適用されます。したがって、特定のパーティションの監査ログを表示するには、「show audit messages」コマンドを使用する必要があります。

> 注
>
> 管理パーティションのユーザーはシェルにアクセスできないため、ログファイルにアクセスできません。

管理パーティションの **Web** ログを取得するにはどうすればよいですか?

管理パーティションの Web ログは、次のように取得できます。

- **NetScaler 11.0** 以降のバージョンの場合

  Web ロギング機能は、Web ロギングを必要とする各パーティションで有効にする必要があります。Citrix ADC Web ログ（NSWL）クライアントを使用して、Citrix ADC は、ユーザーが関連付けられているすべてのパーティションの Web ログを取得します。

- **NetScaler 11.0** より前のバージョンの場合

  Web ログは、`nsroot` および他のスーパーユーザーのみが取得できます。また、デフォルトのパーティションで Web ログが有効になっていても、Citrix ADC Web Logging（NSWL）クライアントはすべてのパーティションの Web ログをフェッチします。

各ログエントリのパーティションを表示するには、%P オプションを含めるようにログフォーマットをカスタマイズします。その後、ログをフィルタリングして、特定のパーティションのログを表示できます。

管理パーティションのトレースを取得するにはどうすればよいですか?

管理パーティションのトレースは、次のようにして取得できます。

- **NetScaler 11.0** 以降のバージョンの場合

パーティション化された Citrix ADC アプライアンスでは、`nstrace` 操作は個々の管理パーティションで実行できます。トレースファイルは、*/var/partitions/ <partitionName> /nstrace/* ディレクトリに格納されます。

注： GUI を使用して管理パーティションのトレースを取得することはできません。CLI を使用する必要があります。

- **NetScaler 11.0** より前のバージョンの場合

  `nstrace` 操作は、デフォルトのパーティションでのみ実行できます。したがって、パケットキャプチャは Citrix ADC システム全体で使用できます。パーティション固有のパケットキャプチャを取得するには、VLAN-ID ベースのフィルタを使用します。

admin パーティションに固有のテクニカルサポートバンドルを入手するにはどうしたらいいですか

特定のパーティションのテクニカルサポートバンドルを取得するには、デフォルトのパーティションから次のコマンドを実行する必要があります。

```
> show techsupport -scope partition -partitionname <string>
```

注**:** このコマンドは、システム固有の情報も提供します。

AppFlow

October 7, 2021

- **AppFlow** をサポートしている **Citrix ADC** ビルドはどれですか**?**

  AppFlow は、nCore ビルドでバージョン 9.3 以降を実行している Citrix ADC アプライアンスでサポートされています。

- **AppFlow** がデータ送信に使用するフォーマットは何ですか**?**

  AppFlow は、RFC 5101 で定義されたオープンなインターネット技術標準化委員会（IETF）標準であるインターネットプロトコルフロー情報 eXport（IPFIX）形式で情報を送信します。IPFIX（Cisco 社製 NetFlow の標準化バージョン）は、ネットワークフロー情報を監視するために幅広く使用されています。

- **AppFlow** レコードには何が含まれていますか**?**

  AppFlow レコードには、フローの開始と終了のタイムスタンプ、パケットカウント、バイトカウントなど、標準の NetFlow または IPFIX 情報が含まれます。AppFlow レコードには、アプリケーションレベルの情報（HTTP URL、HTTP リクエストメソッドとレスポンスステータスコード、サーバーのレスポンス時間、待機時間など）も含まれます。IPFIX フローレコードは、フローレコードを送信する前に送信する必要があるテンプレートに基づいています。

- **NetScaler** バージョン **9.3** ビルド **48.6 Cl** にアップグレードした後、**GUI** から仮想サーバーを開こうとすると、「**AppFlow** 機能は **Citrix ADC Ncore** でのみ使用可能」というエラーメッセージが表示されるのはなぜですか**?**

  AppFlow は、nCore アプライアンスでのみサポートされています。仮想サーバー構成タブを開いたら、[**AppFlow**] チェックボックスをオフにします。

- **AppFlow** レコードのトランザクション **ID** には何が含まれていますか**?**

  トランザクション ID は、アプリケーションレベルのトランザクションを識別する符号なしの 32 ビット数値です。HTTP の場合、トランザクションは要求と応答のペアに対応します。この要求と応答のペアに対応するすべてのフローレコードは、同じトランザクション ID を持ちます。一般的なトランザクションには 4 つのフローレコードがあります。Citrix ADC が単独で応答を生成する場合（統合キャッシュまたはセキュリティポリシーによって提供される）、トランザクションには 2 つのフローレコードしか存在しない可能性があります。

- **AppFlow** アクションとは何ですか**?**

  AppFlow アクションは、関連付けられた AppFlow ポリシーが一致した場合にフローレコードが送信されるコレクタのセットです。

- **AppFlow** アクションがヒットしたことを確認するために、**Citrix ADC** アプライアンスでどのようなコマンドを実行できますか**?**

  AppFlow を表示アクション。次に例を示します:

```
> show appflow action
1) Name: aFL-act-collector-1
  Collectors: collector-1
  Hits: 0
  Action Reference Count: 2
 2) Name: apfl-act-collector-2-and-3
  Collectors: collector-2, collecter-3
  Hits: 0
  Action Reference Count: 1
 3) Name: apfl-act-collector-1-and-3
  Collectors: collector-1, collecter-3
  Hits: 0
  Action Reference Count: 1
<!--NeedCopy-->
```

- **AppFlow** コレクターとは何ですか**?**

コレクタは、Citrix ADC アプライアンスによって生成されたフローレコードを受信します。フローレコードを送信できるようにするには、少なくとも 1 つのコレクタを指定する必要があります。最大 4 つまで指定できます。未使用のコレクターを削除できます。

- **AppFlow** を使用するには、どのような **Citrix ADC** バージョンが必要ですか**?**

  NetScaler バージョン 9.3.49.5 以降を使用し、AppFlow は nCore ビルドでのみ使用できることに注意してください。

- **AppFlow** はどのようなトランスポートプロトコルを使用しますか**?**

  AppFlow は、トランスポートプロトコルとして UDP を使用します。

- ネットワークにファイアウォールがある場合、どのポートを開く必要がありますか。

  ポート 4739 です。これは、AppFlow コレクタが IPFIX メッセージをリッスンするために使用するデフォルトの UDP ポートです。ユーザーがデフォルトポートを変更した場合、そのポートをファイアウォールで開く必要があります。

- **AppFlow** が使用するデフォルトのポートを変更するにはどうすればよいですか**?**

  add appflowCollector コマンドを使用して AppFlow コレクタを追加する場合、使用するポートを指定できます。

```
1  > add appflowCollector coll1 -IPAddress 10.102.29.251 -port 8000
2   Done
3  <!--NeedCopy-->
```

- クライアントトラフィックのみを設定するとどうなりますか**?**

  Citrix ADC は、クライアント側のトラフィックに対してのみ AppFlow レコードを生成します。

- 一度にいくつのコレクタを設定できますか。

  Citrix ADC アプライアンスでは、一度に最大 4 つの AppFlow コレクタを構成できます。Citrix ADC アプライアンスで構成できるコレクタの最大数は 4 台であることに注意してください。

Call Home

October 7, 2021

- **Citrix ADC** アプライアンスの **Call Home** とは何ですか。

  Call Home は、Citrix ADC アプライアンスの重要なイベントを監視し、通知します。Call Home を有効にすると、エラー通知プロセスを自動化できます。Citrix サポートが問題のトラブルシューティングを行う前に、Citrix サポートに電話をかけたり、サービスリクエストを送信したり、システムデータをアップロードしたりするだけでなく、問題が発生する前に問題を特定して解決することもできます。

- **Citrix ADC** アプライアンスで **Call Home** がデフォルトで有効になっていますか**?**

はい、Call Home はデフォルトでアプライアンスで有効になっています。Call Home がデフォルトで無効になっている古いバージョンから最新のソフトウェアにアップグレードすると、アップグレードプロセスによって機能が自動的に有効になります。後で無効にすると、更新後のすべてのアップグレードで更新された設定が記憶されます。詳細については、「Call Home」を参照してください。

- **Call Home** が機能するための前提条件を教えてください。

  インターネット接続へのアクセス。

  注：Citrix ADC アプライアンスにインターネット接続がない場合は、Citrix ADC がシステムログを生成して Citrix テクニカルサポートサーバー（CIS）にアップロードできるプロキシサーバーを構成できます。

- **Call Home** を使用するメリットは何ですか。

  - ハードウェアおよびソフトウェアのエラー状態を監視します。
  - ネットワークに影響を与える重要なイベントの発生について通知します。
  - パフォーマンスデータとシステムログを Citrix に送信して、以下を行います。
    * 製品の品質を分析し、改善します。
    * プロアクティブな問題の特定と迅速な問題解決のために、リアルタイムのトラブルシューティング情報を提供します。

- **Call Home** をサポートしている **Citrix ADC** ソフトウェアのリリースはどれですか**?**

  Citrix ADC リリース 10.0 以降のバージョンです。

- **Call Home** をサポートしている **Citrix ADC** プラットフォームモデルは何ですか**?**

  Call Home 機能は、すべての Citrix ADC プラットフォームとすべてのアプライアンスモデル（MPX、VPX、SDX）でデフォルトで有効になっています。

  - Citrix ADC の MPX: すべての MPX モデル.
  - Citrix ADC VPX: すべての VPX モデル. また、外部または中央のライセンスプールからライセンスを取得する VPX アプライアンスでもサポートされます。ただし、この機能は標準 VPX アプライアンスの場合と同じままです。
  - Citrix ADC SDX：ディスクドライブを監視し、エラーや障害がないか SSL チップを割り当てます。ただし、VPX インスタンスは電源ユニット（PSU）にアクセスできないため、ステータスは監視されません。SDX プラットフォームでは、Call Home を個々のインスタンスで直接設定することも、SVM を介して設定することもできます。

- エラー状態を通知するために、**Call Home** の **SNMP** アラームを設定する必要がありますか。

  いいえ。SNMP と Call Home のアップロードは互いに独立しているため、エラー状態を監視するために Call Home の SNMP を設定する必要はありません。エラー状態が発生するたびに通知を受けるには、Call Home アップロードが発生するたびに SNMP アラートを生成するように CALLHOME-UPLOAD-EVENT SNMP アラームを設定できます。SNMP アラートは、重要なイベントの発生についてローカル管理者に通知します。

- テクニカルサポートに連絡する方法を教えてください。

すべての重要なハードウェア関連イベントについて、Call Home は Citrix へのサービスリクエストを自動的に作成します。その他のエラーについては、システムログを確認した後、Citrix テクニカルサポートチームに連絡して、詳細な調査のためのサービスリクエストを開くことができます。詳細は http://support.citrix.com/article/CTX200021を参照してください。

- **Call Home** は **Citrix ADC** アプライアンスでどのようなエラー状態を監視しますか**?**

  Call Home は、Citrix ADC アプライアンスで次のイベントの監視をサポートします。

  - コンパクトフラッシュドライブのエラー
  - ハードディスクドライブのエラー
  - 電源装置障害
  - SSL カードの障害
  - ウォームリスタート
  - メモリ異常
  - レート制限のドロップ

- **Call Home** 用に別のライセンスが必要ですか。

  いいえ、Call Home では別のライセンスは必要ありません。すべての Citrix ADC プラットフォームライセンスで有効にすることができます。

- **Call Home** が **Citrix** サポートサーバーに送信するデータと、送信頻度はどれくらいですか**?**

  Call Home は、2 種類のデータを収集し、CIS に送信します。これには、次の種類のアカウントがあります。

  - 基本的なシステム情報（Citrix ADC バージョン、展開モード（スタンドアロン、HA、クラスタ）、ハードウェアの詳細など）。このメッセージは、Call Home 登録時および定期的なハートビートの一部として送信されます。ハートビートは 30 日に 1 回送信されますが、この間隔は 1 ～30 日の範囲で設定できます。ただし、頻繁なアップロードはあまり役に立たないため、5 日未満の値は推奨されません。
  - エラー状態がある場合の `show tech support bundle` の短縮版。アプライアンスが最後に起動されてから、特定のエラー状態が最初に発生したときに送信されます。つまり、同じエラー状態が再発生しても、前回の発生後にアプライアンスが再起動されない限り、別のアップロードはトリガーされません。

- **Call Home** は、プロキシサーバを介してシステムログを生成およびアップロードできますか。

  はい。Citrix ADC アプライアンスに直接インターネット接続がない場合は、プロキシサーバーを構成し、システムログを Citrix テクニカルサポートサーバー（CIS）にアップロードできます。

- **Call Home** データを **CIS** に送信する前に確認できますか。

  申し訳ございませんが、CIS に送信される前に Call Home データを確認することはできません。Call Home は、Citrix サポートチームに連絡する際に提供するデータに加えて、その他のデータを収集しません。

- **Call Home** のアップロードの安全性とプライベート性はどれくらいですか。

  Call Home は、次の方法でデータのセキュリティとプライバシーを提供します。

- セキュアな SSL/TLS チャネルを使用して、Citrix サーバーにデータを転送します。
- アップロードされたデータは、権限のある担当者のみがレビューし、第三者と共有することはありません。

クラスタリング

October 7, 2021

クラスタリングに関する FAQ については、ここをクリックしてください 。

接続管理

October 7, 2021

- 管理者接続とは何ですか**?**

  管理者接続では、NSIP アドレスへの接続を確立し、管理者は Citrix ADC アプライアンスを構成および監視できます。

- 管理者接続にはどのような種類がありますか。

  管理接続には、次の 2 種類があります。

  - SSH 接続 — 管理者ユーザーは SSH クライアントを使用して NSIP アドレスを介してログオンします。
  - NITRO API 接続 — 管理者ユーザーは、NITRO API を使用して、Citrix ADC アプライアンスへのログオンプロセスを自動化します。

> 注
>
> 管理者ユーザーは、ブラウザを使用して NSIP アドレスに接続することで、GUI からログオンしてログオンすることもできます。GUI は内部的に NITRO API 接続を開きます。したがって、GUI セッションは NITRO API 接続と同等であり、NITRO API に関連する FAQ は GUI に適用されます。

- **Citrix ADC** アプライアンスで許可される管理接続の数はいくつですか**?**

  アプライアンスは、最大 20 の同時管理接続を許可します。

- 管理者ログオンにはどのログイン資格情報が必要ですか**?**

  管理者ログオンには、ユーザー名とパスワードが必要です。

  注：パスワードの代わりに認証キーを使用できます。

- **Citrix ADC** アプライアンスでサポートされる外部認証方法はどれですか**?**

  アプライアンスは、次の外部認証方式をサポートしています。

  - RADIUS
  - LDAP
  - TACACS

- クライアントとは**?**

  クライアントとは、管理者ユーザーが管理者接続を開くために使用するデバイス (ラップトップまたはデスクトップ) です。

- セッショントークンとは何ですか**?**

  セッショントークンは、Citrix ADC アプライアンスが NITRO API ログオン要求を送信するクライアントに発行する一意の識別子です。

  - API クライアントは、セッショントークンが期限切れでない場合、新しい TCP 接続での後続の API 要求に再利用できます。
  - GUI クライアントは内部的に NITRO API 接続を開き、GUI セッション中はセッショントークンをアクティブに保ちます。

- **Citrix ADC** アプライアンスのアクティブセッションとは何ですか**?**

  CLI セッションは、セッションの有効期限が切れず、Citrix ADC アプライアンスとのオープンな SSH 接続がある場合、アクティブと見なされます。

  Citrix ADC アプライアンスでセッショントークンのタイムアウトが期限切れになっていない場合、NITRO API セッションはアクティブと見なされます。

- **Citrix ADC** は同時接続制限をどのように適用しますか**?**

  Citrix ADC アプライアンスは、管理接続要求（SSH または NITRO API）を受信するたびに、開いている管理接続の数をチェックします。数が 20 未満の場合、新しい接続が開かれます。

- **Citrix ADC** アプライアンスの管理接続数を反映するカウンタはどれですか**?**

  接続カウンタ（nsconfigd_cur_clients）は、アクティブな接続の数を表します。このカウンタは、クライアントがアプライアンスへの新しい接続を開くと増分され、接続が閉じられると減少します。

- **Citrix ADC** アプライアンス上のアクティブなトークンの数を反映するカウンタはどれですか**?**

  config_cur_tokens カウンタは、Citrix ADC アプライアンス上のアクティブなトークンの数を反映します。

- **Citrix ADC** アプライアンスは接続時のエラーをどのように処理しますか**?**

  Citrix ADC アプライアンスは、接続でエラーが発生すると、クライアント（CLI、API、GUI）接続をただちに閉じます。

- 管理アドレスへの接続の **CLI** または **GUI** セッションは、**admin** 接続制限にカウントされますか**?**

  はい。すべての CLI および GUI 接続は TCP ベースの接続であり、管理アドレスへの TCP 接続はすべて、管理接続制限に対してカウントされます。

- **NITRO** セッションは管理者接続制限に対してカウントされますか**?**

  Citrix ADC アプライアンスによって発行されたセッショントークンを使用してオープンな TCP 接続がある場合、NITRO セッションは管理者接続制限に対してカウントされます。

- **Citrix ADC** アプライアンスでの **API**、**GUI**、**CLI** セッションのデフォルトのタイムアウト期間はどれくらいですか**?**

  次の表に、Citrix ADC アプライアンスでの API、GUI、および CLI セッションのデフォルトのタイムアウト期間を示します。

Citrix ADC のリリース	CLI のデフォルトのタイムアウト期間（分）	API のデフォルトのタイムアウト期間 (分)	GUI のデフォルトのタイムアウト期間（分）
NetScaler 9.3	なし	30 分間	30 分間
NetScaler 10.1	なし	30 分間	30 分間
NetScaler 10.5 以降	15 分間	30 分間	15 分間

- **Citrix ADC** アプライアンスで **CLI** セッションのタイムアウトを設定するにはどうすればよいですか**?**

  CLI セッションタイムアウトを設定するには、CLI プロンプトで次のコマンドを実行します。

  ```
  set cli mode -timeout \<xx seconds>
  ```

- **NITRO API** を使用する場合、デフォルトのタイムアウト期間をどのように上書きしますか**?**

  ログインオブジェクトの「timeout」フィールドにタイムアウト期間を設定することで、NITRO API のデフォルトのタイムアウト期間を上書きできます。セッションタイムアウトがゼロに設定されている場合、セッショントークンのタイムアウトは無限になります。

  注：タイムアウトしないセッションは、管理接続カウントに対してカウントされ続けるため、無限タイムアウトはお勧めできません。

- 管理セッションの作成後に **Citrix ADC** アプライアンスからユーザーアカウントを削除するとどうなりますか**?**

  内部システムユーザーの場合、Citrix ADC アプライアンスは既存の CLI または NITRO API セッションを閉じます。

  外部システムユーザの場合、セッションは有効期限が切れるまでアクティブなままです。

- **NITRO API** クライアントは、単一のセッショントークンを使用して、**Citrix ADC** アプライアンスで複数の管理接続を開くことができますか**?**

  はい。このような各接続は、管理者の接続制限に対してカウントされます。

- **SNIP** アドレスに対して管理アクセスが有効になっている場合、そのアドレスへの管理者接続は、管理者接続数の制限に対してカウントされますか**?**

はい、管理アドレス（SNIP）への管理者接続は、Citrix ADC 管理者接続制限に対してカウントされます。

- 最大接続数に達した後、**Citrix ADC** 管理者が **Citrix ADC** アプライアンスにログオンすることはできますか**?**

  はい。最大接続制限に達すると、もう 1 つの管理者接続が許可されます。

- **NITRO API** エンドポイントは、アプライアンスの **Citrix ADC** 上で複数の管理接続を開くことができますか**?**

  はい。NITRO API エンドポイントは、複数の管理接続を開いて、Citrix ADC アプライアンスの同時管理接続制限を使い果たすことができます。このような状況では、追加の SSH/CLI 接続が許可され、管理者は古い API セッションの強制終了や、既存の API セッションのセッションタイムアウト時間を短縮できます。

- 同じクライアントで **Citrix ADC** アプライアンスで複数の **API** セッションを開くことはできますか**?**

  はい。クライアントは繰り返しログオンすることで、複数の API セッションを開くことができます。たとえば、クライアントは再起動後に再びログオンすることがあります。
  注：クライアントのログオンの繰り返しは、Citrix ADC アプライアンスの管理者接続制限に対してカウントされます。

- **API** クライアントは **API** セッショントークンの制限全体を使用できますか**?**

  はい。API クライアントは、以前に発行されたトークンを使用せずに繰り返しログオンすることによって提供される API セッショントークンの制限全体を使用できます。

  注：クライアントのセッションタイムアウトがゼロの場合、トークンは永久に有効です。新しいセッショントークンを使用して繰り返しログオンすると、API セッショントークンの制限にカウントされます。

- **CLI** セッションは **API** セッショントークンの制限に対してカウントされますか**?**

  いいえ。CLI セッションは API セッショントークンの制限に対してカウントされません。

- 管理者ユーザーは **telnet** を使用して **CLI** セッションを開くことができますか**?**

  いいえ。CLI セッションを開くことができるのは SSH クライアントだけです。

- さまざまな **Citrix ADC** リリースに適用される接続制限と **API** セッション制限とは何ですか**?**

  次の表は、さまざまな Citrix ADC リリースに適用可能な最大同時管理接続とアクティブな API セッション制限の一覧です。

Citrix ADC のリリース	9.3	10.1 (130.x 以前)	10.1 (130.10 以前)	10.1 (130.10 以降)
同時管理接続の最大数	20	20	20	20
アクティブな API セッションの最大数 *	1000	20	1000	1000

注:

- API セッションは、タイムアウトしていない場合、アクティブと見なされます。たとえば、500 個の API セッションが作成され、100 個の有効期限が切れている場合、400 個の API セッションがアクティブになります。
- API セッションでは、Citrix ADC アプライアンスへの TCP 接続を開く必要はありません。

コンテンツスイッチ

October 7, 2021

- **Citrix ADC** 以外の負荷分散アプライアンスをネットワークにインストールしました。ただし、**Citrix ADC** アプライアンスのコンテンツスイッチング機能を使用して、クライアント要求を負荷分散アプライアンスに転送したいと考えています。**Citrix ADC** アプライアンスのコンテンツスイッチング機能を、**Citrix ADC** 以外の負荷分散アプライアンスで使用することはできますか**?**

  はい。Citrix ADC アプライアンスのコンテンツスイッチング機能を、Citrix ADC アプライアンスまたは非 Citrix ADC 負荷分散アプライアンスの負荷分散機能とともに使用できます。ただし、Citrix ADC 以外の負荷分散アプライアンスを使用する場合は、Citrix ADC アプライアンスに負荷分散仮想サーバーを作成し、Citrix ADC 以外の負荷分散アプライアンスにサービスとしてバインドしてください。

- コンテンツスイッチング仮想サーバーと負荷分散仮想サーバーはどのように異なりますか**?**

  コンテンツスイッチング仮想サーバーは、クライアント要求を他の仮想サーバーに送信することしかできません。サーバーと通信しません。

  負荷分散仮想サーバは、サーバ間でクライアントの負荷を分散し、サーバと通信します。サーバの可用性を監視し、異なるロードバランシングアルゴリズムを適用してトラフィックの負荷を分散するために使用できます。

  コンテンツスイッチングは、仮想サーバーの負荷分散によって、特定の種類のコンテンツに対するクライアント要求をターゲットサーバーに送信するために使用される方法です。クライアント要求を処理するのに最適なサーバーにクライアント要求を転送できます。これにより、サーバー上のクライアント要求を処理するためのオーバーヘッドが削減されます。

- **Citrix ADC** アプライアンスのコンテンツスイッチング機能を実装して、クライアントの要求を転送します。コンテンツスイッチング機能を使用して送信できるクライアント要求の種類を教えてください**?**

  コンテンツスイッチング機能を使用すると、HTTP、HTTPS、FTP、TCP、セキュア TCP、および RTSP クライアント要求のみを転送できます。HTTPS クライアント要求を送信するには、アプライアンスで SSL オフロード機能を設定する必要があります。

- **Citrix ADC** アプライアンスでコンテンツスイッチングルールを作成したい。コンテンツスイッチングルールを作成できるクライアント要求のさまざまな要素は何ですか**?**

  コンテンツスイッチングルールは、クライアントリクエストの次の要素とその値に基づいて作成できます。

  - URL
  - URL トークン
  - HTTP バージョン
  - HTTP ヘッダー
  - クライアントの送信元 IP アドレス
  - クライアントバージョン
  - 宛先 TCP ポート

- **Citrix ADC** アプライアンスのコンテンツスイッチング機能は、ネットワークのパフォーマンスを向上させるのに役立つことを理解しています。これは正しいですか**?**

  はい。あなたは、クライアントがあなたにそれらを処理するのに最適なサーバーを指示することができます。その結果、サーバー上のクライアント要求を処理するためのオーバーヘッドが削減されます。

- クライアント要求に対するサイト管理性と応答時間を向上させるために、**Citrix ADC** アプライアンスのどの機能を **Citrix ADC** アプライアンスに構成する必要がありますか**?**

  Citrix ADC アプライアンスのコンテンツスイッチング機能を構成して、サイト管理機能とクライアント要求に対する応答時間を向上させることができます。この機能を使用すると、同じドメイン名と IP アドレス内にコンテンツグループを作成できます。このアプローチは、ユーザーが参照できる異なるドメイン名や IP アドレスにコンテンツを明示的に分割する一般的なアプローチとは異なり、柔軟です。

  ウェブサイトをさまざまなドメイン名と IP アドレスに分割する複数のパーティションにより、ブラウザはウェブページのコンテンツをレンダリングおよび取得するときに検出されたドメインごとに個別の接続を強制的に作成します。これらの追加の WAN 接続により、Web ページの応答時間が低下します。

- 私は **Web** サーバーファームでウェブサイトをホストしました。**Citrix ADC** コンテンツスイッチング機能では、このようなセットアップにはどのようなメリットがありますか**?**

  コンテンツスイッチング機能には、Web サーバーファームに基づくサイト内の Citrix ADC アプライアンスで次のような利点があります。

  - 同じドメインと IP アドレス内にコンテンツグループを作成して、サイトコンテンツを管理します。
  - 同じドメインおよび IP アドレス内のコンテンツグループを使用して、クライアント要求に対する応答時間を短縮します。
  - ドメイン間での完全なコンテンツレプリケーションの必要性を回避します。
  - アプリケーション固有のコンテンツ・パーティショニングを有効にします。たとえば、要求に応じて、動的コンテンツのみを処理するサーバーまたは静的コンテンツのみを処理するサーバーにクライアント要求を転送できます。
  - 同じサーバー上の複数のドメインのマルチホームをサポートし、同じ IP アドレスを使用します。
  - サーバーへの接続を再利用します。

- **Citrix ADC** アプライアンスにコンテンツスイッチング機能を実装したい。私は、各要求のさまざまなパラメータを評価した後、クライアント要求をさまざまなサーバーに指示したいと思います。コンテンツスイッチング機能を設定するときに、このセットアップを実装するにはどのようなアプローチに従うべきですか**?**

ポリシー式を使用して、コンテンツスイッチング機能のポリシーを作成できます。式は、演算子を使用してクライアント要求の修飾子をオペランドと比較することによって評価される条件です。クライアント要求の次のパラメータを使用して、式を作成できます。

- メソッド-HTTP リクエストメソッド。
- **URL**-HTTP ヘッダー内の URL。
- **URL** トークン-URL 内の特別なトークン。
- バージョン-HTTP リクエストのバージョン。
- **URL** クエリ-URL クエリ LEN、URL LEN、および HTTP ヘッダーが含まれています。
- **SOURCEIP**-クライアントの IP アドレス。

式を作成するために使用できる演算子の完全なリストを次に示します。

- == (等しい)
- != (等しくない)
- EXISTS
- NOT EXISTS
- CONTAINS
- NOT CONTAINS
- GT (より大きい)
- LT (より小さい)

また、一連の式を論理的に集約するさまざまなルールを作成することもできます。複数の式を組み合わせてルールを作成できます。式を組み合わせるには、&& (AND) と (OR) 演算子を使用します。括弧を使用して、ネストされた複雑な規則を作成することもできます。

- 私は、同じコンテンツスイッチング仮想サーバーの **URL** ベースのポリシーと一緒にルールベースのポリシーを構成したいと思います。同じコンテンツスイッチ仮想サーバに対して両方のタイプのポリシーを作成することは可能ですか**?**

  はい。同じコンテンツスイッチ仮想サーバーに対して、両方のタイプのポリシーを作成できます。ただし、必ず優先順位を割り当てて、ポリシーに適切な優先順位を設定してください。

- 私は、**URL** の接頭辞と接尾辞とともにドメイン名を評価し、それに応じてクライアント要求を指示するコンテンツスイッチポリシーを作成したいと思います。どのタイプのコンテンツスイッチングポリシーを作成すればよいですか**?**

  ドメインと完全な URL ポリシーを作成できます。このタイプのポリシーを評価すると、クライアント要求内の完全なドメイン名および URL が構成済みのものと一致する場合、Citrix ADC アプライアンスはコンテンツグループを選択します。クライアント要求は、設定されたドメイン名と一致し、URL のプレフィクスとサフィ

ックスが設定されている場合は完全に一致する必要があります。

- 私は、**URL** の部分的な接頭辞と接尾辞とともにドメイン名を評価し、それに応じてクライアント要求を指示するコンテンツスイッチポリシーを作成したいと思います。どのタイプのコンテンツスイッチングポリシーを作成すればよいですか**?**

  コンテンツスイッチ仮想サーバーのドメインおよびワイルドカード URL ポリシーを作成できます。このタイプのポリシーが評価されると、Citrix ADC アプライアンスはリクエストが完全なドメイン名と一致し、URL プレフィックスと部分的に一致する場合、コンテンツグループを選択します。

- ワイルドカード **URL** ポリシーとは**?**

  ワイルドカードを使用すると、Citrix ADC アプライアンスで設定した URL へのクライアントリクエストの部分 URL を評価できます。ワイルドカードは、次の種類の URL ベースのポリシーで使用できます。

  - プレフィックスのみ。たとえば、/sports/* 式は、/sports URL の下で使用可能なすべての URL に一致します。同様に、/sports * 式は、接頭辞が /sports であるすべての URL に一致します。
  - 接尾辞のみ。たとえば、/*.jsp 式は、ファイル名拡張子が jsp のすべての URL に一致します。
  - 接頭辞と接尾辞。たとえば、/sports/ *.jsp 式は、/sports/ URL の下にある jsp ファイル名拡張子を持つすべての URL に一致します。同様に、/sports *.jsp 式は、接頭辞 /sports * とファイル名拡張子が jsp のすべての URL に一致します。

- ドメインと規則のポリシーとは**?**

  ドメインおよび規則ポリシーを作成する場合、クライアント要求は、完全なドメインおよび Citrix ADC アプライアンスで構成されたルールと一致する必要があります。

- ポリシーの評価に使用するデフォルトの優先順位はどれですか**?**

  デフォルトでは、ルールベースのポリシーが最初に評価されます。

- 一部のコンテンツがすべてのクライアント要求で同じ場合、ポリシーの評価にはどのような優先順位を使用すればよいですか**?**

  一部のコンテンツがすべてのユーザーで同じで、クライアント属性に基づいて異なるコンテンツを提供する必要がある場合は、ポリシー評価に URL ベースの優先順位を使用できます。

- コンテンツスイッチングでは、どのようなポリシー式構文がサポートされていますか**?**

  コンテンツスイッチングでは、次の 2 種類のポリシー表現がサポートされます。

  - クラシック構文**-** コンテンツスイッチングのクラシック構文は、キーワード REQ で始まり、デフォルトの構文よりも高度なものです。クラシックポリシーはアクションにバインドできません。したがって、ターゲット負荷分散仮想サーバーは、コンテンツスイッチング仮想サーバーをバインドした後にのみ追加できます。
  - デフォルト構文**:** デフォルトの構文は、通常 HTTP というキーワードで始まり、構成が簡単です。ターゲットの負荷分散仮想サーバーのアクションは、Default Syntax ポリシーにバインドできます。このポリシーは、複数のコンテンツスイッチング仮想サーバーで使用できます。

- 単一のコンテンツスイッチポリシーを複数の仮想サーバーにバインドできますか**?**

  はい。アクションを定義したポリシーを使用して、1 つのコンテンツスイッチポリシーを複数の仮想サーバーにバインドできます。ターゲット負荷分散仮想サーバーがコンテンツスイッチングポリシーで指定されなくなったため、アクションを使用するコンテンツスイッチングポリシーは、複数のコンテンツスイッチング仮想サーバーにバインドできます。1 つのポリシーを複数のコンテンツスイッチング仮想サーバーにバインドする機能により、コンテンツスイッチング構成のサイズをさらに縮小できます。

  詳細については、次のナレッジセンター記事および Citrix のドキュメントトピックを参照してください。

  - CTX122918- Citrix ADC アプライアンス上の 2 つのコンテンツスイッチング仮想サーバーに同じコンテンツスイッチングポリシーをバインドする方法を参照してください。
  - CTX122736- ポリシーラベルを使用して同じ詳細ポリシーを複数のコンテンツスイッチング仮想サーバにバインドする方法を参照してください。
  - 基本的なコンテンツスイッチングの設定。

- 従来の式を使用してアクションベースのポリシーを作成できますか**?**

  いいえ。現在のところ、Citrix ADC では、アクションで古典的な構文式を使用するポリシーはサポートされていません。ポリシーをバインドするときに、アクションで定義するのではなく、ターゲットの負荷分散仮想サーバーを追加する必要があります。

デバッグ

October 7, 2021

- 操作が実行されたインターフェイス（**CLI**、**GUI**、**API**）をどのように判断できますか**?**

  Citrix ADC は、操作が実行されるインターフェイスを追跡します。この情報は、syslog（GUI で、[設定] > [システム] > [監査] > [監査メッセージ] > [Syslog メッセージ] の順に選択）または ns.log ファイル（/var/log/ ディレクトリにあります）で表示できます。

  たとえば、API を介して実行される操作には「API CMD_EXECUTED」というフラグが付けられます。

ハードウェア

October 7, 2021

MPX ハードウェアに関する FAQ については、ここをクリックしてください 。

高可用性

October 7, 2021

- **HA** 構成のノード間で **HA** 関連情報を交換するために使用するさまざまなポートは何ですか。

  HA 構成では、両方のノードは次のポートを使用して HA 関連情報を交換します。

  - ハートビートパケットを交換するための UDP ポート 3003
  - 同期およびコマンド伝播用のポート 3010

- **INC** モードまたは非 **INC** モードの **HA** 構成で同期または伝播されない構成は何ですか**?**

  次のコマンドで実装された構成は、セカンダリノードに伝達も同期もされません。

  - すべてのノード固有の HA 構成コマンド。たとえば、`add ha node`、`set ha node`、`bind ha node`などです。
  - インターフェイス関連のすべての構成コマンド。たとえば、インターフェイスを設定し、インターフェイスを設定解除します。
  - チャネル関連のすべての構成コマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。

  INC モードの HA 設定の詳細については、「異なるサブネットでのハイアベイラビリティノードの設定」を参照してください。

- **INC** モードの **HA** 構成で同期または伝播されない構成は何ですか**?**

  次の設定は、同期も伝播もされません。各ノードには独自のノードがあります。

  - MIP
  - SNIP
  - VLAN
  - ルート（LLB ルートを除く）
  - ルートモニタ
  - RNAT ルール（NAT IP として VIP を使用するすべての RNAT ルールを除く）
  - ダイナミックルーティング設定。

- 同期をトリガーする条件は何ですか。

  同期は、次のいずれかの条件によってトリガーされます。

  - セカンダリが受信したプライマリノードのインカネーション番号が、セカンダリノードのインカネーション番号と一致しません。
    注：高可用性構成の両方のノードは、ノード構成ファイル内の構成 の数をカウントインカネーション番号と呼ばれるカウンタを維持します。各ノードは、ハートビートメッセージ内の別のノードにインカネーション番号を送信します。次のコマンドでは、インカネーション番号は増分されません。
    * すべての HA 設定関連コマンド。たとえば、`add ha node`、`set ha node`、`bind ha node`などです。

        * すべてのインターフェイス関連コマンド。たとえば、インターフェイスを設定し、インターフェイスを設定解除します。
        * すべてのチャネル関連コマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。
    - 再起動後、セカンダリノードが起動します。
    - プライマリノードは、フェールオーバー後にセカンダリになります。

- セカンダリノードに追加された構成はプライマリで同期されますか?

  いいえ。セカンダリノードに追加された構成は、プライマリノードと同期されません。

- 両方のノードが **HA** 構成でプライマリであると主張する理由は何ですか**?**

  最も可能性の高い理由は、プライマリノードとセカンダリノードは両方とも正常ですが、セカンダリはプライマリからハートビートパケットを受信しないことです。問題は、ノード間のネットワークにある可能性があります。

- 異なるシステムクロック設定で **2** つのノードを展開すると、**HA** 構成で問題が発生しますか**?**

  2 つのノードのシステムクロックの設定が異なると、次の問題が発生する可能性があります。

    - ログファイルエントリのタイムスタンプが一致しません。この状況では、問題のログエントリを分析することが困難になります。
    - フェイルオーバー後、負荷分散のためのあらゆる種類のクッキーベースの永続性に問題がある可能性があります。時間が大きく異なると、Cookie が予想よりも早く期限切れになり、永続セッションが終了する可能性があります。
    - 同様の考慮事項は、ノードの時間に関連する決定にも適用されます。

- ***force HA sync*** コマンドが失敗する条件は何ですか?

  強制同期は、次のいずれかの状況で失敗します。

    - 同期が既に進行中の場合は、同期を強制します。
    - セカンダリノードは無効です。
    - HA 同期は、現在のセカンダリノードで無効になっています。
    - 現在のプライマリノードで HA 伝播が無効になり、プライマリからの同期が強制されます。

- ***sync HA files*** コマンドが失敗する条件は何ですか?

  セカンダリノードが無効になっている場合、設定ファイルの同期は失敗します。

- **HA** 構成で、セカンダリノードがプライマリとして引き継ぐ場合、元のプライマリがオンラインに戻ると、セカンダリステータスに戻りますか。

  いいえ。セカンダリノードがプライマリとして引き継ぐと、元のプライマリノードが再びオンラインに戻っても、プライマリノードはプライマリのままになります。ノードのプライマリとセカンダリのステータスを交換するには、*force failover* コマンドを実行します。

- ***forcefailover*** コマンドが失敗する条件は何ですか。

次の状況では、強制フェールオーバーを実行できません。

- セカンダリノードは無効です。
- セカンダリノードは、セカンダリノードを維持するように構成されています。
- プライマリノードはプライマリノードとして構成されています。
- ピアノードの状態が不明です。

統合キャッシング

October 7, 2021

コンテンツグループ

- **DEFAULT** コンテンツグループと他のコンテンツグループとの違い**?**

  DEFAULT コンテンツグループの動作は、他のグループと同じです。DEFAULT コンテンツグループを特別なものにする唯一の属性は、オブジェクトがキャッシュされていて、コンテンツグループが作成されていない場合です。オブジェクトは DEFAULT グループにキャッシュされます。

- コンテンツ・グループ・レベルの「キャッシュ制御」オプションとは何ですか**?**

  あなたは、任意のキャッシュ制御ヘッダーをブラウザに送信することができます。コンテンツグループレベルのオプション-cacheControl があり、ブラウザへの応答に挿入するキャッシュコントロールヘッダーを指定できます。

- コンテンツグループレベルの「**Minhit**」オプションとは何ですか**?**

  Minhitは、オブジェクトがキャッシュされる前のキャッシュポリシーの選択の最小数を指定する整数値です。この値は、コンテンツグループレベルで設定できます。次に、CLI からこの値を設定する構文を示します。

  ```
  add/set cache contentGroup \<Content_Group_Name> [-minHits \<Integer>]
  ```

- 期限切れの最後のバイトオプションの使用は何ですか**?**

  expireAtLastByte オプションを使用すると、統合キャッシュは、ダウンロード時にオブジェクトを期限切れにすることができます。未処理のリクエストであるリクエストのみがキャッシュから提供されます。新しいリクエストはすべてサーバーに送信されます。この設定は、株価の場合と同様に、オブジェクトが頻繁に変更される場合に便利です。この有効期限メカニズムは、Flash Cache 機能とともに機能します。expireAtLastByte オプションを設定するには、CLI から次のコマンドを実行します。

  ```
  add cache contentGroup \<Group_Name> -expireAtLastByte YES
  ```

キャッシュポリシー

- キャッシュポリシーとは何ですか**?**

ポリシーは、キャッシュ可能なトランザクションとキャッシュできないトランザクションを決定します。また、ポリシーは標準の HTTP キャッシング動作を追加またはオーバーライドします。ポリシーは、要求または応答の特性に応じて、CACHE や NOCACHE などのアクションを決定します。応答がポリシールールに一致する場合、応答内のオブジェクトがポリシーで構成されたコンテンツグループに追加されます。コンテンツグループを設定していない場合、オブジェクトは DEFAULT コンテンツグループに追加されます。

- ポリシーヒットとは何ですか**?**

  選択は、要求または応答がキャッシュポリシーに一致する場合に発生します。

- ミスとは何ですか**?**

  要求または応答がどのキャッシュポリシーにも一致しない場合、ミスが発生します。また、要求または応答がキャッシュポリシーと一致するが、RFC の動作のオーバーライドによってオブジェクトがキャッシュに格納されない場合にも発生します。

- **Citrix ADC** アプライアンスの統合キャッシュ機能を設定しました。次のポリシーを追加すると、エラーメッセージが表示されます。コマンドにエラーはありますか**?**

  ```
  add cache policy image_caching -rule exp1 | ns_ext_not_jpeg -action
  cache

  \> ERROR: No such command
  ```

  上記のコマンドでは、式は引用符で囲まれている必要があります。引用符を使用しない場合、演算子はパイプ演算子と見なされます。

メモリ要件

- **Citrix ADC** アプライアンスで実行して、キャッシュに割り当てられたメモリをチェックできるコマンドは何ですか**?**

  Citrix ADC アプライアンスでキャッシュに割り当てられたメモリを表示するには、CLI から次のコマンドのいずれかを実行します。

  - `show cache parameter`

    出力で、メモリ使用制限パラメータの値を確認します。これは、キャッシュに割り当てられた最大メモリです。

  - `show cache \<Content_Group_Name>`

    出力で、個々のコンテンツグループに使用および割り当てられているメモリを示す [メモリ使用量] および [メモリ使用量] パラメータの値を確認します。

- 私の **Citrix ADC** アプライアンスには、**2 GB** のメモリがあります。キャッシュの推奨メモリ制限はありますか**?**

Citrix ADC アプライアンスのどのモデルでも、メモリの半分をキャッシュに割り当てることができます。ただし Citrix 内部メモリの依存関係があるため、メモリの半分以下を割り当てることをお勧めします。次のコマンドを実行して、1GB のメモリをキャッシュに割り当てることができます。

```
set cache parameter -memLimit 1024
```

- 個々のコンテンツグループにメモリを割り当てることは可能ですか**?**

  はい。set cache パラメータ —memlimit を実行して統合キャッシュにメモリをグローバル <Integer> に割り当てる場合でも、set cache <Content_Group_Name> —memLimit <Integer> コマンドを実行すると、個々のコンテンツグループにメモリを割り当てることができます。コンテンツグループ (組み合わせ) に割り当てることができる最大メモリは、統合キャッシュに割り当てたメモリを超えることはできません。

- 統合キャッシュと **TCP** バッファの間のメモリの依存関係は何ですか**?**

  Citrix ADC アプライアンスに 2GB のメモリがある場合、アプライアンスは約 800MB から 900MB のメモリを予約し、残りは FreeBSD オペレーティングシステムに割り当てられます。したがって、最大 512 MB のメモリを統合キャッシュに割り当てることができ、残りは TCP バッファに割り当てられます。

- 統合キャッシュにグローバルメモリを割り当てないと、キャッシュプロセスに影響しますか**?**

  統合キャッシュにメモリを割り当てない場合、すべての要求がサーバーに送信されます。メモリを統合キャッシュに割り当てたことを確認するには、show cache parameter コマンドを実行します。グローバルメモリが 0 の場合、実際にはオブジェクトはキャッシュされないため、最初に設定する必要があります。

検証コマンド

- キャッシュ統計情報を表示するためのオプションは何ですか**?**

  次のいずれかのオプションを使用して、キャッシュの統計を表示できます。

  - `stat cache`

    キャッシュ統計情報のサマリーを表示します。

  - `stat cache -detail`

    キャッシュ統計情報の完全な詳細を表示します。

- キャッシュされたコンテンツを表示するためのオプションは何ですか**?**

  キャッシュされたコンテンツを表示するには、`show cache object` コマンドを実行します。

- キャッシュに格納されているオブジェクトの特性を表示するために実行できるコマンドは何ですか**?**

  キャッシュに格納されているオブジェクトが GET //10.102.12.16:80/index.html などの場合は、アプライアンスの CLI から次のコマンドを実行して、オブジェクトの詳細を表示できます。

  ```
  show cache object -url '/index.html'-host 10.102.3.96 -port 80
  ```


- キャッシュ内のパラメータ化されたオブジェクトを表示するために、グループ名をパラメータとして指定することは必須ですか**?**

  はい。パラメータ化されたオブジェクトをキャッシュに表示するには、グループ名をパラメータとして指定する必要があります。たとえば、同じ規則で次のポリシーを追加したとします。

```
add cache policy p2 -rule ns_url_path_cgibin -action CACHE -
    storeInGroup g1
add cache policy p1 -rule ns_url_path_cgibin -action CACHE -
    storeInGroup g2
<!--NeedCopy-->
```

  この場合、複数の要求に対して、ポリシー p1 が評価されると、その選択カウンターが増分され、ポリシーはオブジェクトを選択パラメーターを持つ g1 グループに格納します。したがって、キャッシュからオブジェクトを表示するのには、次のコマンドを実行する必要があります。

```
show cache object -url ”/cgi-bin/setCookie.pl”-host 10.102.18.152
groupName g1
```

  同様に、複数の要求の別のセットの場合、ポリシー p2 が評価されると、その選択カウンターが増分され、ポリシーはオブジェクトを選択パラメーターを持たない g2 グループに格納します。したがって、キャッシュからオブジェクトを表示するのには、次のコマンドを実行する必要があります。

```
show cache object -url ”/cgi-bin/setCookie2.pl”-host 10.102.18.152
```

- **nscachemgr** コマンドの出力に空白のエントリがいくつかあることに気付きました。これらのエントリは何ですか**?**

  nscachemgrコマンドの次のサンプル出力について考えてみましょう。この出力の空白のエントリは、参考のために太字で強調表示されます。

```
root@ns# /netscaler/nscachemgr -a
//10.102.3.89:80/image8.png
//10.102.3.97:80/staticdynamic.html
//10.102.3.97:80/
//10.102.3.89:80/image1.png
//10.102.3.89:80/file5.html
//10.102.3.96:80/
//10.102.3.97:80/bg_logo_segue.png
//10.102.3.89:80/file500.html
//10.102.3.92:80/
//10.102.3.96:80/cgi-bin/rfc/ccProxyReval.pl
Total URLs in IC = 10
<!--NeedCopy-->
```


出力内の空白のエントリは、GET/HTTP/1.1 のデフォルトのキャッシングプロパティによるものです。

オブジェクトのフラッシュ

- 選択オブジェクトをキャッシュからフラッシュするにはどうすればよいですか**?**

  完全な URL でオブジェクトを一意に識別できます。このようなオブジェクトをフラッシュするには、次のタスクのいずれかを実行できます。

  - キャッシュ記憶
  - コンテンツグループのフラッシュ
  - 特定のオブジェクトをフラッシュします

  特定のオブジェクトをフラッシュするには、クエリパラメータを指定する必要があります。オブジェクトをフラッシュするには、invalParam パラメータを指定します。このパラメータは、クエリにのみ適用されます。

- キャッシュ構成の変更により、キャッシュのフラッシュがトリガーされますか**?**

  はい。キャッシュ構成に変更すると、すべての SET キャッシュコマンドは本質的に適切なコンテンツグループをフラッシュします。

- サーバー上のオブジェクトを更新しました。キャッシュされたオブジェクトをフラッシュする必要がありますか**?**

  はい。サーバー上のオブジェクトを更新するときは、キャッシュされたオブジェクト、または少なくとも関連するオブジェクトとコンテンツグループをフラッシュする必要があります。統合キャッシュは、サーバーへの更新の影響を受けません。有効期限が切れるまで、キャッシュされたオブジェクトを提供し続けます。

フラッシュキャッシュ

- **Citrix ADC** アプライアンスのフラッシュキャッシュ機能とは何ですか**?**

  Flash クラウドの現象は、多くのクライアントが同じコンテンツにアクセスするときに発生します。その結果、サーバへのトラフィックが急激に急増します。フラッシュキャッシュ機能により、Citrix ADC アプライアンスは、サーバーに 1 つの要求のみを送信することにより、このような状況でのパフォーマンスを向上させることができます。他のすべての要求はアプライアンスのキューに入れられ、単一の応答が要求に提供されます。次のいずれかのコマンドを使用して、高速キャッシュ機能を有効にできます。

  - `add cache contentGroup \<Group_Name> -flashCache YES`
  - `set cache contentGroup \<Group_Name> -flashCache YES`

- フラッシュキャッシュクライアントにはどのような制限がありますか**?**

  Flash Cache クライアントの数は、Citrix ADC アプライアンス上のリソースの可用性によって異なります。

デフォルトの動作

- **Citrix ADC** アプライアンスは、有効期限が切れた時点でオブジェクトをプロアクティブに受信しますか**?**

  Citrix ADC アプライアンスは、有効期限が切れるとプロアクティブにオブジェクトを受信することはありません。これは、負のオブジェクトに対しても当てはまります。有効期限後の最初のアクセスは、サーバーへの要求をトリガーします。

- 統合キャッシュは、応答の受信を開始する前にサービスを提供するためにクライアントをキューに追加しますか**?**

  はい。統合キャッシュは、応答の受信を開始する前にサービスを提供するためにクライアントをキューに追加します。

- キャッシュ構成のパラメータを使用してキャッシュされたオブジェクトを検証するためのデフォルト値は何ですか**?**

  HOSTNAME_AND_IP がデフォルト値です。

- **Citrix ADC** アプライアンスはログファイルにログエントリを作成しますか**?**

  はい。Citrix ADC アプライアンスは、ログファイルにログエントリを作成します。

- 圧縮されたオブジェクトはキャッシュに格納されていますか**?**

  はい。圧縮されたオブジェクトはキャッシュに格納されます。

他の機能との相互運用性

- 現在キャッシュに保存されており、**SSL VPN** 経由でアクセスされているオブジェクトはどうなりますか?

  キャッシュに格納され、定期的にアクセスされるオブジェクトは、キャッシュとして提供されます。SSL VPN 経由でアクセスしたときに選択します。

- **SSL VPN** を介してアクセスし、後で通常の接続を介してアクセスすると、キャッシュに格納されたオブジェクトはどうなりますか**?**

  SSL VPN アクセスを介して格納されたオブジェクトは、通常の接続を介してアクセスされるときに SELECT として提供されます。

- **Web** ロギングを使用する場合、キャッシュから提供される応答を示すエントリとサーバーによって提供されるエントリをどのように区別しますか?

  統合キャッシュから処理される応答の場合、サーバログフィールドには IC の値が含まれます。サーバーから送信された応答の場合、サーバーログフィールドにはサーバーによって送信された値が含まれます。次に、統合キャッシュトランザクションのログエントリの例を示します。

  ```
  ”10.102.1.52 - ”Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 4.0; .NET
   CLR 1.0.3705)””GET /”200 0 ”IC”10.102.1.45”
  ```


クライアント要求とともに、ログに記録される応答は、クライアントに送信される応答であり、必ずしもサーバによって送信された応答ではありません。

> 注
>
> Web ロギングを使用する場合、統合キャッシュからの応答には、サーバーログフィールドに値 IC が含まれます。サーバーログフィールドは、NSWL クライアントに存在します。“%o1” フォーマット指定子。

その他

- 再失効と失効を設定するとどういう意味ですか**?**

  relexpiryとabsexpiryを設定すると、ヘッダーに表示される内容に関係なく、ヘッダーがオーバーライドされます。別の有効期限設定とコンテンツグループレベルを構成できます。relexpiryでは、ヘッダーの有効期限は、Citrix ADC がオブジェクトを受信した時刻に基づきます。absexpiryでは、有効期限は Citrix ADC で構成された時間に基づいています。Relexpiryは秒単位で設定されます。Absexpiry は時刻です。

- **weakpos** とヒューリスティックを設定するとどういう意味ですか**?**

  weakposとヒューリスティックはフォールバック値のようなものです。有効期限ヘッダーがある場合、最後に変更されたヘッダーが存在する場合にのみ考慮されます。Citrix ADC アプライアンスは、最後に変更されたヘッダーとヒューリスティックパラメーターに基づいて有効期限を設定します。ヒューリスティックな有効期限の計算では、最後に変更されたヘッダーをチェックして有効期限を決定します。オブジェクトが最後に変更されてからの期間の一部は、有効期限として使用されます。長期間変更されないままであり、有効期限が長くなる可能性が高いオブジェクトのヒューリスティック。–heurExpiryParam は、この計算で使用するパーセンテージ値を指定します。それ以外の場合、アプライアンスはweakpos値を使用します。

- 動的キャッシュを設定する前に考慮すべきことは何ですか**?**

  名前と値の形式で完全な URL クエリがないパラメーターがある場合、またはアプライアンスが Cookie ヘッダーまたは POST 本文でパラメーターを受信する場合は、動的キャッシュの構成を検討してください。動的キャッシュを構成するには、hitParams パラメーターを構成する必要があります。

- パラメータ名で **16** 進エンコーディングはどのようにサポートされていますか**?**

  Citrix ADC アプライアンスでは、%HEXHEX エンコーディングはパラメータ名でサポートされています。hitParams または invalParams に指定する名前で、を含む名前を指定できます%HEXHEX 名前のエンコーディング。たとえば、名前、name%65, および n %61m%65 同等です。

- **hitParam** パラメータを選択するプロセスは何ですか**?**

  POST リクエストの HTTP ヘッダーの次の抜粋を検討してください。

```
POST /data2html.asp?param1=value1&param2=&param3&param4=value4
HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
```



```
4  application/vnd.ms-powerpoint, application/vnd.ms-excel,
5  application/msword, application/x-shockwave-flash, \*/\*
6  Referer: http://10.102.3.97/forms.html
7  Accept-Language: en-us
8  Content-Type: application/x-www-form-urlencoded
9  Accept-Encoding: gzip, deflate
10 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
11 Host: 10.102.3.97
12 Content-Length: 153
13 Connection: Keep-Alive
14 Cache-Control: no-cache
15 Cookie: ASPSESSIONIDQGQGGRNY=NNLLKDADEENOAFLCCDGFGDMO
16 S1=This+text+is+only+text%2C+not+more+and+not+less%2C+%0D%0Ajust+
      text+to+be+itself%2C+namely+%22Text%22+to+be+posted+as+text
      +%28what+else...%29&B1=Submit
17 <!--NeedCopy-->
```

前のリクエストでは、要件に応じて hitParams として、参照用に太字で強調表示されている S1 と B1 を使用できます。また、ASPSESSIONIDQGQGGRNY コンテンツグループで-matchCookies YES を使用する場合は、これらのパラメーターを hitParams として使用することもできます。

- 応答がキャッシュ可能でない場合、キューに入れられたクライアントはどうなりますか**?**

  応答がキャッシュ可能でない場合、キュー内のすべてのクライアントは、最初のクライアントが受信するのと同じ応答を受信します。

- 同じコンテンツグループで毎回ポーリング（**PET**）機能およびフラッシュキャッシュ機能を有効にできますか**?**

  いいえ。同じコンテンツグループで PET とフラッシュキャッシュを有効にすることはできません。統合キャッシュは、フラッシュキャッシュコンテンツグループに対して AutoPET 機能を実行しません。PET 機能は、サーバーに問い合わせることなく、統合キャッシュが保存されたオブジェクトを提供しないことを保証します。コンテンツグループに対して PET を明示的に設定できます。

- キューに入れられたクライアントのログエントリはいつ作成されますか**?**

  アプライアンスが応答ヘッダーを受信するとすぐに、キューに入れられたクライアントのログエントリが作成されます。ログエントリは、応答ヘッダーによってオブジェクトがキャッシュ不可にならない場合にのみ作成されます。

- キャッシュ構成のパラメータを使用してキャッシュされたオブジェクトを検証する **DNS**、ホスト名、および **HOSTNAME_AND_IP** 値の意味は何ですか**?**

  意味は次のとおりです。

  - `set cache parameter -verifyUsing HOSTNAME`

    このコマンドは、宛先 IP アドレスを無視します。

- `set cache parameter -verifyUsing HOSTNAME_AND_IP`

  このコマンドは宛先 IP アドレスと一致します。

- `set cache parameter -verifyUsing DNS`

  このコマンドは DNS サーバーを使用します。

- **weakNegRelExpiry** を **600** に設定しました。これは **10** 分です。**404** 応答がキャッシュされていないことに気づきました。理由は何ですか**?**

  これは完全に構成に依存します。デフォルトでは、404 応答が 10 分間キャッシュされます。404 応答すべてをサーバーからフェッチする場合は、–weakNegRelExpiry0 を指定します。–weakNegRelExpiry を、より高いまたはより低いなどの目的の値に微調整して、404 応答を適切にキャッシュすることができます。肯定応答用に–absExpiry を構成した場合、望ましい結果が得られない可能性があります。

- ユーザーが **Mozilla Firefox** ブラウザを使用してサイトにアクセスすると、更新されたコンテンツが配信されます。ただし、ユーザーが **Microsoft Internet Explorer** ブラウザーを使用してサイトにアクセスすると、古いコンテンツが提供されます。理由は何でしょうか**?**

  Microsoft Internet Explorer ブラウザーは、Citrix ADC 統合キャッシュではなくローカルキャッシュからコンテンツを取得している可能性があります。その理由は、Microsoft InternetExplorer ブラウザーが応答の有効期限関連のヘッダーを尊重していないことが原因である可能性があります。

  この問題を解決するには、Internet Explorer のローカルキャッシュを無効にして、オフラインコンテンツをクリアします。オフラインコンテンツをクリアした後、ブラウザは更新されたコンテンツを表示する必要があります。

- ヒット数がゼロの場合はどうなりますか**?**

  サーバー時間と NS 時間が同期しているかどうかを確認します。また、weakPosrelexpiry 制限セットは、次のように NS とサーバー間の時間差に耐える必要があります。

```
root@ns180# date
Tue May 15 18:53:52 IST 2012
<!--NeedCopy-->
```

- ポリシーがヒットしたのに、キャッシュされていないのはなぜですか**?**

  メモリが統合キャッシュに割り当てられていること、および割り当てがゼロより大きいことを確認してください。

- キャッシュカウンタをゼロにすることは可能ですか**?**

  キャッシュカウンターをゼロに設定するためのコマンドラインまたは GUI オプションはなく、キャッシュをフラッシュすることもできません。ボックスを再起動すると、これらのカウンターが自動的にゼロに設定されます。

インストール、アップグレード、およびダウングレード

October 7, 2021

インストールとアップグレード

特定の **Citrix ADC** リリースビルドパッケージをダウンロードするには**?**

特定の Citrix ADC リリースビルドパッケージをダウンロードする方法については、「Citrix ADC リリースパッケージのダウンロード」を参照してください。

Citrix ADC アプライアンスのシステムソフトウェアをアップグレードするには**?**

Citrix ADC アプライアンスのシステムソフトウェアのアップグレードについては、Citrix ADC スタンドアロンアプライアンスのアップグレードを参照してください。

Citrix ADC リリースビルドのリリースノートはどこにありますか**?**

Citrix ADC リリースビルドのリリースノートドキュメントには、リリースビルドの次のリストがあります。

- 強化された機能
- 解決された問題
- 既知の問題

Citrix ADC リリースビルドのリリースノートドキュメントは、次の場所にあります。

- Citrix ADC ファームウェアまたは仮想アプライアンスは、特定のリリースビルドのダウンロードページをダウンロードします 。
- Citrix ドキュメントサイトの Citrix ADC リリースノートページ

Citrix ADC アプライアンスのセキュリティ更新プログラムはどこにありますか**?**

Citrix セキュリティチームは、関連するすべての Citrix 製品の Common Vulnerabilities and Exposures（CVE）に関するセキュリティ情報を定期的にリリースしています。この情報は、Citrix セキュリティ情報で確認できます。または、Citrix サポートサイトで特定の CVE を検索することもできます。

Citrix ADC リリースで利用可能な **zebos.conf** ファイルの使用方法について教えてください**?**

Citrix ADC アプライアンスは、ルーティングスイートとして ZebOS を使用します。Citrix ADC リリースで使用可能な zebos.conf ファイルは、ZebOS の構成ファイルです。

Citrix ADC アプライアンスの **SSH** ポート（**22**）を他のポートに変更したい。アプライアンスの **SSH** ポートを変更することは可能ですか**?**

はい。Citrix ADC アプライアンスの SSH ポートを変更するには、sshd_config のファイル /nsconfig ディレクトリ。ファイルがに存在しない場合 /nsconfig ディレクトリからコピーします /etc ディレクトリ。

の中に sshd_config ファイル、ポート 22 のエントリをポートに編集します < Number>, どこ < 番号 > はターゲットポート番号です。アプライアンスを再起動して変更を有効にしたくない場合は、kill コマンドを使用して `sshd` プロセスを終了してから、プロセスを再起動します。

フラッシュディレクトリが **Citrix ADC** アプライアンスにありません。フラッシュディレクトリをマウントするには、どのような手順に従わなければなりませんか

フラッシュディレクトリをマウントするには、次の手順を実行します。

1. Citrix ADC アプライアンスをシングルユーザーモードで起動します。

   アプライアンスが起動すると、次のメッセージが表示されます。

   [[Enter]] を選択してすぐに起動するか、コマンドプロンプトのその他のキーを選択します。[カーネルを 10 秒で起動中]...” スペースを選択すると、次のプロンプトが表示される必要があります。

   「?」と入力します。コマンドのリストについては、’help’ より詳細なヘルプを参照してください。

2. FreeBSD をシングルユーザーモードで起動するには、以下のコマンドを入力します。

   boot –s

   アプライアンスの起動後、次のメッセージが表示されます。

   シェルのフルパス名を入力または /bin/sh の RETURN を入力:

3. [Enter] を押して、# プロンプトを表示します。

4. 次のコマンドを実行して、フラッシュディレクトリをマウントします。

```
mount /dev/ad0s1a /flash

Note: If the preceding command displays an error message about
   permissions, run the following command to check the disk for
   consistency:

fsck /dev/ad0s1a

Run the mount command again to mount the flash directory.
```

5. アプライアンスを再起動します。

6. シェルプロンプトから次のコマンドを実行して、flash ディレクトリがマウントされていることを確認します。

```
df -kh
```

パスワードを入力せずに **Citrix ADC** アプライアンスにログオンします。アプライアンスに **SSH** を設定して許可することは可能ですか**?**

はい。Citrix ADC アプライアンスで SSH を構成して、パスワードなしでログオンできます。ただし、ユーザー名を入力する必要があります。パスワードなしでログインするように SSH を設定するには、次の手順を実行します。

1. 次のコマンドを実行して、公開キーと秘密キーを生成します。

```
\# ssh-keygen -t rsa
```

2. 次のコマンドを実行して、ログオン先のリモートホストの.ssh ディレクトリに id_rsa.pub ファイルをコピーします。

```
\# scp id_dsa.pub \<user>@\<remote_host>/.ssh/id_dsa.pub
```

3. リモートホストにログオンします。
4. .ssh ディレクトリに移動します。
5. 次のコマンドを実行して、クライアントの公開キーを既知の公開キーに追加します。

```
\# cat id_dsa.pub >> authorized_keys2

\# chmod 640 authorized_keys2

\# rm id_dsa.pub
```

Citrix ADC アプライアンスの **BIOS** をリセットする手順は何ですか**?** どのような状況で **BIOS** をリセットする必要がありますか

Citrix ADC アプライアンスの BIOS をリセットするには、次の手順に従います。

1. シリアル・ポート経由でアプライアンスに接続します。

2. アプライアンスを起動し、起動プロセスが開始したら Delete キーを押します。

   POST プロセス中に Delete キーを押すと、アプライアンスの BIOS 設定が表示されます。

3. BIOS 設定の終了ページを有効にします。
4. [最適な既定値をロード] オプションを選択します。[最適設定をロード] メッセージボックスが表示されます。
5. [OK] を選択します。
6. さまざまなタブで BIOS 設定を次のように変更します。

   タブ

7. BIOS 設定の終了ページを有効にします。
8. [変更を保存して終了] を選択します。
9. [OK] を選択して確定します。
10. アプライアンスが正常に起動し、アプライアンスの起動後にシリアルコンソールに出力が表示されることを確認します。

    シリアルコンソールが応答しない場合は、BIOS をリセットする必要があります。これは通常、アプライアンスをアップグレードし、シリアルコンソールが無効になった後に発生します。ただし、telnet または SSH ユーティリティを使用してアプライアンスにアクセスすることはできます。

Citrix ADC アプライアンスを工場出荷時のデフォルトにリセットする必要があります。どのような手順に従わなければなりませんか

Citrix ADC アプライアンスを工場出荷時のデフォルトにリセットするには、Citrix ADC アプリケーション環境と FreeBSD 環境の 2 つの環境をリセットする必要があります。
アプライアンスの Citrix ADC アプリケーション環境を工場出荷時のデフォルトにリセットするには、次の手順に従います。

1. アプライアンスの /nsconfig/ns.conf のバックアップを作成します。
2. /nsconfig/ns.conf ファイルを削除します。
3. アプライアンスを再起動します。アプライアンスの FreeBSD 環境を工場出荷時のデフォルトにリセットするには、以下を実行します。
   a) 新しい Citrix ADC コードイメージをアプライアンスにインストールします。これにより、いくつかの FreeBSD レベルの設定ファイルがデフォルト値で上書きされます。
   b) アプライアンスに追加されたすべてのユーザーとグループ、つまりデフォルトユーザーを除くすべてのユーザーおよびグループを削除します。
   c) /etc/resolv.conf ファイルを削除します。
   d) /etc/hosts ファイルに追加したエントリを削除します。
   e) /etc/rc.netscaler ファイルが存在する場合は、それを削除します。
   f) /etc/nsperm_group_suser ファイルを開き、すべての IOCTL エントリがコメントエントリであることを確認します。

g) /etc/rc.conf ファイルを開き、syslogd_enable=NO エントリが syslogd_enable=YES に変更されていないことを確認します。
h) /etc/syslog.conf ファイルを開き、ファイルに追加のエントリがないことを確認します。
i) /var/nslog ファイル、/var/nstrace ファイル、および /var/crash ファイルの内容を削除します。
j) アプライアンスで syslog プロセスが有効になっていて、アプライアンスがログファイルをローカルに作成する場合は、/etc/syslog.conf ファイルにリストされているログファイルの内容を削除します。ファイルは /var/log ディレクトリに作成されます。たとえば、syslog プロセスが /var/log/events ファイルにシステムイベントを書き込み、/var/log/sslvpnevents ファイルに`sslvpn` イベントにアクセスする場合は、これらのファイルを削除します。

アプライアンスは、コンソールに「**Jun 21 12:20:18 ns /flash/ns-10.0-47.15: [1/2]dc0: NIC** ハング状態 **#663: TX 10000/10000**、**RX 0**、**HF 0**」というメッセージに似たメッセージを表示します。このメッセージの意味は何ですか**?**

このメッセージは、次のコンポーネントで構成されています（例としてここに示されています）。

- #663: この状態がアプライアンスで発生した回数。
- TX 10000/10000：アプライアンスが送信を試みたパケット数および送信パケット数。この例のように、両方の数値が同じ場合、NIC はアプライアンスが送信を試みたすべてのパケットを送信しました。
- RX 0：受信したパケットの数。この例では、パケットは受信されませんでした。
- HF0: NIC によって報告されたハードウェアの問題の数。この例では、NIC はハードウェアの問題を報告していません。

アプライアンスがパケットを受信しない場合、ネットワーク上ではパケットを受信しない可能性が高いため、ハング状態が報告されます。ただし、アプライアンスがインターフェイスに接続されている場合は、このエラーメッセージを無視できます。

アプライアンスで **Citrix ADC** リリースをアップグレードした後も、アプライアンスには以前のリリース**/**ビルドが表示されます。理由は何でしょうか**?**

アプライアンスは、/flash/boot/loader.conf ファイルからソフトウェアのバージョン番号を表示します。現在の Citrix ADC リリースのカーネルエントリがそのファイルに存在しない場合、アプライアンスはそのエントリーが使用可能であった最新の Citrix ADC リリースバージョンを表示します。
この問題を解決するには、次の手順を実行します：

1. カーネルファイルが /nsconfig ディレクトリに存在することを確認します。
2. /flash/boot/loader.conf ファイルでカーネルのエントリを確認します。

   （カーネルのエントリを期待できます release/build インストールしたファイルから欠落していること。）
3. vi エディタなどのテキストエディタで loader.conf ファイルを開き、新しいリリース/ビルドのカーネルエントリを更新します。

4. ファイルを保存して閉じます。
5. /flash/boot/loader.conf.local ファイルに対して手順 2. ～4. を繰り返します。
6. ns.conf ファイルのリリース/ビルドエントリを更新します。
7. アプライアンスを再起動します。

アプライアンスの **Citrix ADC** リリースをアップグレードしたため、アプライアンスの前面パネルの **LCD** ディスプレイにアウトアウトアウトアウトアウトアウトアウト（**out of Service**）メッセージが表示されるか、何も表示されません。この問題を解決するにはどうしたらいいですか**?**

アプライアンスのシェルプロンプトから次のコマンドを実行します。

```
/netscaler/nslcd  - k
```

Citrix ADC のリリース/ビルドをアップグレードしました。ただし、アップグレードプロセスの後、アプライアンスの起動に失敗します。アプライアンスのソフトウェアを以前のリリース/ビルドにダウングレードできますか**?**

はい。kernel.old カーネルファイルを使用してアプライアンスを起動できます。アプライアンスを再起動するとき、アプライアンスコンソールに PressF1 メッセージが表示されたら F1 キーを押します。kernel.old と 入力し、**Enter** キーを押します。

アプライアンスで **Citrix ADC** リリースをアップグレードした後、カーネルファイルを誤って削除しました **/flash** ディレクトリ。その結果、アプライアンスを起動できません。この状況でアプライアンスを起動する方法はありますか**?**

はい。次のように、`kernel.GENERIC` カーネルファイルを使用してアプライアンスを起動できます。

1. アプライアンスを再起動するとき、アプライアンスコンソールに PressF1 メッセージが表示されたら F1 キーを押します。
2.「カーネル」と入力します。一般的な [Enter] を押します。
3. root ユーザーとしてログインします。
4. Citrix ADC リリースを再インストールします。
5. アプライアンスを再起動します。

アプライアンスソフトウェアをアップグレードした後、アプライアンスにログオンできず、次のメッセージが表示されます。パスワード回復手順を使用してこの問題を解決しようとしましたが、成功しませんでした。私は間違って何かをしましたか**?**

```
```
login: nsroot
```

 
```

```
Password:
connect: No such file or directory
nsnet_connect: No such file or directory
Login incorrect
<!--NeedCopy--> ```
```

パスワード回復手順を使用してこの問題を解決することはできません。Citrix ADC リリース 12.1 以降では、起動手順中に実行される Imgrd デーモンに基づく新しいライセンスシステムが使用されます。このデーモンが正常に動作するには、/nsconfig/rc.conf ファイルで設定された Citrix ADC アプライアンスのホスト名を、ネーム・サーバによって NSIP アドレスに解決する必要があります。または、/nsconfig ディレクトリに hosts ファイルを作成し、ファイル <Host_Name> に 127.0.0.1 エントリを追加することもできます。

また、ライセンスファイルをにコピーしたことを確認してください /nsconfig/license/ ディレクトリ。

高可用性ペアのアップグレード中に、次のメッセージが繰り返し表示されます。理由は何でしょうか**?**

ns sshd[5035]: エラー: ユーザー名またはパスワードが無効です
このエラーメッセージは、高可用性ペアリングに関係するアプライアンスに、異なる Citrix ADC リリースまたは同じリリースの異なるビルドがインストールされている場合に表示されます。1 つのアプライアンスをアップグレードまたはダウングレードして他のアプライアンスをアップグレードまたはダウングレードしなかった場合、アプライアンスに異なるバージョンをインストールできます。
Citrix ADC アプライアンス上の NSIP アドレスのネットマスクを変更したい。停止を発生させずに実行できますか?
Citrix ADC IP のネットマスクを変更すると、短時間停止することがあります。セカンダリアプライアンスのネットマスクを変更してから、高可用性のペアリングを解除してください。アプライアンスの機能を確認します。すべてが期待どおりに機能する場合は、高可用性ペアを再構築します。
アプライアンスのネットマスクを変更するには、CLI プロンプトから ‘config ns‘ コマンドを実行し、メニューの 2 番目のオプションを選択します。
Citrix ADC アプライアンスの高可用性ペアを構成しました。ソフトウェアリリースをプレビューリリースから最終リリースにアップグレードした後、アプライアンス構成の一部が欠落していることに気付きました。失われた構成を取得できますか?
構成を復元するには、次の手順を使用します。
1. プライマリアプライアンスの Citrix ADC コマンドラインにログオンします。
2. 次のコマンドを実行します:
save config
shell
\#cp /nsconfig/ns.conf /nsconfig/ns.conf.bkup
The ns.conf.bkup file is a backup for the running configuration.
3. 両方のアプライアンスのソフトウェアを最終リリースにアップグレードします。
4. プライマリアプライアンスの Citrix ADC コマンドラインにログオンします。
プライマリアプライアンスとセカンダリアプライアンスのビルドは別個にできますか

プライマリアプライアンスとセカンダリアプライアンスの両方で同じバージョンとビルド番号を使用することをお勧めします。
高可用性（HA）ペアの両方のアプライアンスを同時にアップグレードできますか?
いいえ。HA ペアでは、最初にセカンダリノードをアップグレードしてから、プライマリノードをアップグレードします。
詳細については、「[ハイアベイラビリティペアのアップグレード](/ja-jp/citrix-adc/13/upgrade-downgrade-citrix-adc-appliance/upgrade-downgrade-HA-pair.html)」を参照してください。
Citrix は Amazon Web Services クラウドでのファームウェアアップグレードをサポートしていますか
はい。
Citrix ADC インスタンスを SDX バージョンとは別にアップグレードできますか?
Citrix ADC アプライアンスのアップグレード時に、SDX バージョンをアップグレードする必要はありません。ただし、一部の機能が動作しない場合があります。
FTP サーバーを使用して Citrix ADC アプライアンスをアップグレードできますか?
いいえ。まず、Citrix ダウンロードサイトからファームウェアをダウンロードし、ローカルコンピューターに保存してから、アプライアンスをアップグレードする必要があります。
GSLB 構成で Citrix ADC アプライアンスをアップグレードする手順は、GSLB に関与していないアプライアンスのアップグレードとは異なりますか?
いいえ。アップグレード手順は、基本的なアップグレード手順と似ています。唯一の違いは、スタンドアロンまたは HA アプライアンスを異なるサイトで段階的にアップグレードできることです。
ダウングレード
Citrix ADC アプライアンスに最新の Citrix ADC リリースがインストールされています。ただし、ソフトウェアリリースをダウングレードします。私はそうできますか?
いいえ。ソフトウェアリリースをダウングレードしようとすると、新しいリリースの ns.conf ファイルが以前のリリースと互換性がなく、アプライアンスが工場出荷時の設定に戻る可能性があるため、アプライアンスが期待どおりに動作しない可能性があります。
Citrix ADC リリースをダウングレードするときは、指示に従いました。ただし、アプライアンスは次のメッセージを表示します。Citrix ADC アプライアンスでのロールバック手順はどのように実行されますか?
root@LBCOL03B# ./installns
installns version (10.0-47.7) kernel (ns-10.0-47.7.gz)
Note:
Installation may pause for up to 3 minutes while data is written to the flash.
Caution:
Do not interrupt the installation process.
Doing so may cause the system to become unusable.
Installation will proceed in 5 seconds, CTRL-C to abort
No Valid Citrix ADC Version Detected
root@LBCOL03B#
ロールバック手順は、基本的なアップグレード手順に似ています。ロールバックするターゲットビルドを選択し、ダウングレードを実行します。別のリリースにロールバックする前に、現在の構成ファイルのコピーを作成するこ

とをお勧めします。リリースからダウングレードするには、[Citrix ADC スタンドアロンアプライアンスのダウングレードを参照してください](/ja-jp/citrix-adc/13/upgrade-downgrade-citrix-adc-appliance/downgrade-standalone-appliance.html)。

負荷分散

October 7, 2021

- **Citrix ADC** アプライアンスで作成できるさまざまな負荷分散ポリシーは何ですか**?**

  Citrix ADC アプライアンスでは、次の種類の負荷分散ポリシーを作成できます。

  - 最小接続数
  - ラウンドロビン
  - 応答時間の最短短短縮
  - 最小帯域幅
  - 最小パケット
  - URL ハッシュ
  - ドメイン名のハッシュ化
  - 送信元 IP アドレスのハッシュ
  - 宛先 IP アドレスのハッシュ
  - 送信元 IP：宛先 IP ハッシュ
  - トークン
  - LRTM

- **Citrix ADC** アプライアンスを使用して負荷分散を実装することで、**Web** ファームのセキュリティを実現できますか**?**

  はい。Citrix ADC アプライアンスを使用して負荷分散を実装することで、Web ファームのセキュリティを実現できます。Citrix ADC アプライアンスでは、負荷分散機能の次のオプションを実装できます。

  - IP アドレスの非表示: セキュリティ上の理由と IP アドレスの保存のために、実際のサーバーをプライベート IP アドレス空間にインストールできます。Citrix ADC アプライアンスはサーバーに代わって要求を受け入れるため、このプロセスはエンドユーザーに対して透過的です。アドレス隠蔽モードでは、アプライアンスによって 2 つのネットワークが完全に隔離されます。したがって、クライアントは、そのサービス用のアプライアンス上の異なる VIP を介して、FTP や Telnet サーバーなどのプライベートサブネットで実行されているサービスにアクセスできます。
  - ポートマッピング: セキュリティ上の理由から、実際の TCP サービスを非標準ポートでホストできるようにします。このプロセスは、Citrix ADC アプライアンスが標準のアドバタイズ IP アドレスとポート番号でサーバーに代わって要求を受け入れるため、エンドユーザーには透過的です。

- **Citrix ADC** アプライアンスの負荷分散に使用できるさまざまなデバイスは何ですか**?**

Citrix ADC アプライアンスを使用して、次のデバイスを負荷分散できます。

- サーバーファーム
- キャッシュまたはリバースプロキシ
- ファイアウォールデバイス
- 侵入検知システム
- SSL オフロードデバイス
- 圧縮装置
- コンテンツ検査サーバー

- なぜ私は、ウェブサイトの負荷分散機能を実装する必要がありますか**?**

  Web サイトの負荷分散機能を実装すると、次の利点があります。

  - 応答時間の短縮: Web サイトの負荷分散機能を実装する場合、大きな利点の 1 つは、ロード時に期待できるブーストです。2 つ以上のサーバーが Web トラフィックの負荷を共有している場合、各サーバーのトラフィック負荷は 1 台のサーバーだけよりも少なくなります。これは、クライアント要求を満たすために利用可能なリソースがさらに存在することを意味します。これにより、ウェブサイトがより高速になります。
  - 冗長性: 負荷分散機能を実装すると、少しの冗長性が導入されます。たとえば、ウェブサイトが 3 つのサーバー間でバランスが取れていて、そのうちの 1 つがまったく応答しない場合、他の 2 つは実行し続けることができ、ウェブサイトの訪問者もダウンタイムに気付かない。負荷分散ソリューションは、利用できないバックエンドサーバーへのトラフィックの送信を直ちに停止します。

- リンクロードバランシング **(LLB)** の **Mac** ベースフォワーディング **(MBF)** オプションを無効にする必要があるのはなぜですか**?**

  - MBF オプションを有効にすると、Citrix ADC アプライアンスは、クライアントからの着信トラフィックと同じクライアントへの発信トラフィックが同じアップストリームルーターを通過すると見なします。ただし、LLB 機能では、リターントラフィックに最適パスを選択する必要があります。
  - MBF オプションを有効にすると、着信クライアントトラフィックを転送したルータを介して発信トラフィックを送信することによって、このトポロジ設計が中断されます。

- **Citrix ADC** アプライアンスで使用できるさまざまな永続性タイプは何ですか**?**

  Citrix ADC アプライアンスは、次の永続性タイプをサポートしています。

  - 接続元 IP
  - クッキーインサート
  - SSL セッション ID
  - URL パッシブ
  - Custom Server ID
  - 規則
  - DESTIP

GUI

January 31, 2022

- **Firefox** を使用して **2** つの **Citrix ADC** 構成を比較すると、ブラウザがフリーズするようです。

  Firefox は最終的に設定の違いを表示しますが、1000 を超える違いがあると、処理にかなりの時間がかかります。Chrome を使用すると、応答が速くなります。

- **MAC Safari** ブラウザを使用して **Citrix ADC** をアップグレードしています。アップグレードウィザードで、**[Browse]** ボタンをクリックしてアプライアンスからビルドファイルを選択すると、ダイアログボックスにファイルまたはフォルダが表示されません。また、ルートフォルダに戻ると、ダイアログボックスに最上位フォルダが表示されますが、参照できません。どうしたらいいですか**?**

  Safari ブラウザで「設定」アイコンをクリックし、「環境設定」**>**「セキュリティ」**>**「**Web** サイト設定の管理」**>**「**Java**」に移動します。[他の **Web** サイトにアクセスしたとき] 設定の値を [安全でないモードで実行する] に変更します。

- **GUI** にアクセスする前に何をすべきですか**?**

  新しいバージョンの Citrix ADC ソフトウェアにアクセスする前に:

  - Cookie を含むブラウザのキャッシュをクリアします。
  - ブラウザのシークレットモードで GUI にアクセスします。
  - 他のブラウザで GUI にアクセスします。
  - 設定の [ソフトウェアアクセラレーションを使用する] オプションをオフにして、ブラウザを再起動します。
  - **chrome:** 拡張機能にアクセスし、[有効] チェックボックスをオフにして、Chrome ブラウザを再起動します。

- **HTTP** または **HTTPS** を使用して **GUI** にアクセスするにはどのポートを開くべきですか**?**

  Citrix ADC MPX、VPX、および CPX アプライアンスの HTTP および HTTPS 管理サービス（GUI）のデフォルトのポート番号を以下に示します。

  - Citrix ADC MPX および VPX アプライアンス：80（HTTP）および 443（HTTPS）アプライアンス
  - Citrix ADC CPX アプライアンス：9080（HTTP）および 9443（HTTPS）

  また、ポート 80 および 443 以外の HTTP および HTTPS 管理サービス (GUI) 用のポートを構成することもできます。詳細については、「HTTP および HTTPS 管理ポートの設定」を参照してください。

- どのブラウザで、異なるオペレーティングシステムと互換性のある **GUI** がありますか**?**

  次の表に、NetScaler GUI バージョン 12.0、12.1、および 13.0 と互換性のあるブラウザーを示します。

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 7 およびそれ以降	Internet Explorer	11、Edge、およびそれ以降

オペレーティングシステム	**Web** ブラウザー	バージョン
Windows 7 およびそれ以降	Mozilla Firefox	45 およびそれ以降
Windows 7 およびそれ以降	Chrome	60 およびそれ以降
MAC	Mozilla Firefox	45 およびそれ以降
MAC	Safari	10.1.1 およびそれ以降

SSL

October 7, 2021

SSL に関するよくある質問については、ここをクリックしてください 。

アプリケーショントラフィックの認証、承認、監査

October 7, 2021

多くの企業は、Web サイトへのアクセスを有効なユーザーのみに制限し、各ユーザーに許可されるアクセスのレベルを制御しています。認証、承認、監査機能を使用すると、サイト管理者は Citrix ADC アプライアンスでアクセス制御を管理できます。これらの制御をアプリケーションごとに個別に管理する必要はありません。アプライアンスで認証を行うと、アプライアンスによって保護されている同じドメイン内のすべての Web サイト間でこの情報を共有することもできます。

認証、認可、および監査を使用するには、認証プロセスを処理する認証仮想サーバと、認証を必要とする Web アプリケーションへのトラフィックを処理するトラフィック管理仮想サーバを設定する必要があります。また、各仮想サーバーに FQDN を割り当てるように DNS を構成します。仮想サーバーを構成したら、Citrix ADC アプライアンスを介して認証される各ユーザーのユーザーアカウントを構成します。オプションで、グループを作成し、ユーザーアカウントをグループに割り当てます。ユーザー・アカウントとグループを作成したら、アプライアンスにユーザーの認証方法、ユーザーがアクセスを許可するリソース、ユーザー・セッションのログ記録方法を指示するポリシーを設定します。ポリシーを有効にするには、各ポリシーをグローバルに、特定の仮想サーバ、または適切なユーザーアカウントまたはグループにバインドします。ポリシーを設定したら、セッション設定を構成し、セッションポリシーをトラフィック管理仮想サーバにバインドして、ユーザセッションをカスタマイズします。最後に、イントラネットでクライアント証明書を使用する場合は、クライアント証明書の構成を設定します。

分散環境で認証、承認、および監査がどのように機能するかを理解するために、従業員がオフィス、自宅、および出張時にアクセスするイントラネットを持つ組織を検討してください。イントラネット上のコンテンツは機密情報であり、安全なアクセスが必要です。イントラネットにアクセスするすべてのユーザーは、有効なユーザー名とパスワードを持っている必要があります。これらの要件を満たすために、ADC は次のことを行います。

- ユーザーがログインせずにイントラネットにアクセスした場合、ユーザーをログインページにリダイレクトします。
- ユーザーの資格情報を収集して認証サーバーに配信し、ライトウェイトディレクトリアクセスプロトコル（LDAP）を介してアクセスできるディレクトリにキャッシュします。詳細については、LDAP ディレクトリ内の属性の決定を参照してください。
- ユーザーの要求をアプリケーションサーバーに配信する前に、ユーザーが特定のイントラネットコンテンツにアクセスする権限を持っていることを確認します。
- セッションタイムアウトを維持します。このタイムアウトに達すると、ユーザーはイントラネットへのアクセスを回復するために再度認証する必要があります。（タイムアウトを設定できます）。
- 無効なログイン試行を含め、ユーザーのアクセスを監査ログに記録します。

サポートされている認証タイプ

- ローカル
- LDAP
- RADIUS
- SAML
- TACACS+
- クライアント証明書認証（スマートカード認証を含む）
- Web
- 高度な認証
- フォームベース認証
- 401 ベースの認証
- ネイティブ OTP
- プッシュ通知
- E メール OTP
- reCaptcha

Citrix Gateway は、RSA SecurID、Gemalto Protiva、および SafeWord もサポートしています。RADIUS サーバーを使用して、これらのタイプの認証を構成します。

認証、承認、および監査を構成する前に、Citrix ADC アプライアンスで負荷分散、コンテンツスイッチング、および SSL を構成する方法に精通して理解している必要があります。

許可なしの認証

承認は、ユーザーがアプライアンスにログオンするときにアクセスできるネットワークリソースを指定します。承認のデフォルト設定は、すべてのネットワークリソースへのアクセスを拒否することです。デフォルトのグローバル設定を使用し、承認ポリシーを作成して、ユーザーがアクセスできるネットワークリソースを定義することをお勧めします。

許可ポリシーと式を使用して、アプライアンスで許可を構成します。許可ポリシーを作成したら、アプライアンスで構成したユーザーまたはグループにポリシーをバインドできます。

許可なしで認証のみを使用するようにアプライアンスを構成できます。許可なしで認証を設定すると、アプライアンスはグループ許可チェックを実行しません。ユーザーまたはグループに構成するポリシーは、ユーザーに割り当てられます。

認証、承認、監査の有効化

認証、承認、および監査機能を使用するには、それを有効にする必要があります。認証、承認、および監査機能を有効にする前に、認証、承認、および監査エンティティ（認証およびトラフィック管理仮想サーバーなど）を構成できますが、エンティティは機能が有効になるまで機能しません。

CLI を使用して認証、承認、および監査を有効にするには

コマンドプロンプトで次のコマンドを入力して、認証、承認、および監査を有効にし、構成を確認します。

```
enable ns feature AAA
<!--NeedCopy-->
```

GUI を使用して認証、承認、および監査を有効にするには

1. **System > Settings** に移動します。
2. 詳細ウィンドウの [モードと機能] で、[基本機能の変更] をクリックします。
3. [基本機能の構成] ダイアログボックスで、[認証、承認、監査] チェックボックスをオンにします。
4. [**OK**] をクリックします。

認証の無効化

デプロイメントで認証が必要ない場合は、認証を無効にすることができます。認証を必要としない仮想サーバーごとに認証を無効にすることができます。

> 重要:
>
> 重要： Citrix では、認証を慎重に無効にすることをお勧めします。外部認証サーバーを使用していない場合は、ローカルユーザーとグループを作成して、アプライアンスがユーザーを認証できるようにします。認証を無効にすると、アプライアンスへの接続を制御および監視する認証、許可、およびアカウンティング機能の使用が停止します。ユーザーがアプライアンスに接続するために Web アドレスを入力すると、ログオンページは表示されません。

認証を無効にするには

1. **Configuration > Citrix Gateway > Virtual Servers** に移動します。
2. 詳細ペインで、仮想サーバーをクリックし、[開く] をクリックします。
3. **[基本設定]** ページで、[認証を有効にする] チェックボックスをオフにします。

認証、承認、監査の仕組み

October 7, 2021

認証、承認、および監査は、適切な資格情報を持つすべてのクライアントが、インターネット上のどこからでも保護されたアプリケーションサーバーに安全に接続できるようにすることで、分散インターネット環境のセキュリティを提供します。この機能には、認証、認可、監査の 3 つのセキュリティ機能が組み込まれています。認証により、Citrix ADC は、ローカルまたはサードパーティの認証サーバーを使用してクライアントの資格情報を検証し、承認されたユーザーのみが保護されたサーバーにアクセスできるようにします。認証により、ADC は、各ユーザーがアクセスできる保護されたサーバー上のコンテンツを確認することができます。監査により、ADC は保護されたサーバーでの各ユーザーのアクティビティを記録できます。

分散環境で認証、承認、および監査がどのように機能するかを理解するために、従業員がオフィス、自宅、および出張時にアクセスするイントラネットを持つ組織を検討してください。イントラネット上のコンテンツは機密情報であり、安全なアクセスが必要です。イントラネットにアクセスするすべてのユーザーは、有効なユーザー名とパスワードを持っている必要があります。これらの要件を満たすために、ADC は次のことを行います。

- ユーザーがログインせずにイントラネットにアクセスした場合、ユーザーをログインページにリダイレクトします。
- ユーザーの資格情報を収集し、認証サーバーに配信し、LDAP 経由でアクセス可能なディレクトリにキャッシュします。詳細については、LDAP ディレクトリ内の属性の決定を参照してください。
- ユーザーの要求をアプリケーションサーバーに配信する前に、ユーザーが特定のイントラネットコンテンツにアクセスする権限を持っていることを確認します。
- セッションタイムアウトを維持します。このタイムアウトに達すると、ユーザーはイントラネットへのアクセスを回復するために再度認証する必要があります。(タイムアウトを設定できます)。
- 無効なログイン試行を含め、ユーザーのアクセスを監査ログに記録します。

認証承認と監査ポリシーを構成します

ユーザーとグループを設定したら、次に認証ポリシー、承認ポリシー、および監査ポリシーを構成して、イントラネットへのアクセスを許可するユーザー、各ユーザーまたはグループがアクセスできるリソース、認証、承認、および監査の詳細レベルを定義します。は監査ログに保持されます。認証ポリシーは、ユーザーがログオンしようとしたときに適用される認証の種類を定義します。外部認証を使用する場合、ポリシーによって外部認証サーバも指定されま

す。承認ポリシーは、ユーザーとグループがログオンした後にアクセスできるネットワークリソースを指定します。監査ポリシーは、監査ログの種類と場所を定義します。

ポリシーを有効にするには、各ポリシーをバインドする必要があります。認証ポリシーは、認証仮想サーバー、承認ポリシーを 1 つ以上のユーザーアカウントまたはグループにバインドし、監査ポリシーをグローバルに、1 つ以上のユーザーアカウントまたはグループの両方にバインドします。

ポリシーをバインドするときは、そのポリシーに優先度を割り当てます。優先順位によって、定義したポリシーが評価される順序が決まります。プライオリティは任意の正の整数に設定できます。Citrix ADC オペレーティングシステムでは、ポリシーの優先順位は逆の順序で機能します。値が大きいほど、優先順位は低くなります。たとえば、プライオリティ 10、100、1000 の 3 つのポリシーがある場合、プライオリティ 10 が割り当てられたポリシーが最初に実行され、次にポリシーにプライオリティ 100 が割り当てられ、最後に 1000 というプライオリティが割り当てられます。認証、認可、および監査機能では、リクエストが一致する各タイプのポリシーのうち最初のもののみが実装され、リクエストが一致する可能性のあるタイプの追加のポリシーは実装されません。したがって、ポリシーの優先順位は、意図した結果を得るために重要です。

ポリシーをバインドするときに、各ポリシー間に 50 または 100 の間隔で優先度を設定することで、任意の順序で他のポリシーを追加できる十分な余地を残すことができます。その後、既存のポリシーの優先順位を再割り当てすることなく、いつでもポリシーを追加できます。

Citrix ADC アプライアンスでのバインドポリシーの詳細については、Citrix ADC 製品のドキュメントを参照してください。

を構成します **No_Auth** 特定のトラフィックをバイパスするポリシー

トラフィック管理仮想サーバで 401 ベースの認証が有効になっている場合に、認証からの特定のトラフィックをバイパスするように No_Auth ポリシーを設定できるようになりました。このようなトラフィックの場合、バインドする必要があります “No_Auth” ポリシー。

CLI を使用して特定のトラフィックをバイパスするように **No_Auth** ポリシーを設定するには

コマンドプロンプトで入力します。

```
add authentication policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

例:

```
add authentication policy ldap -rule ldapAct1 -action No_Auth
<!--NeedCopy-->
```


認証、承認、および監査構成の基本コンポーネント

October 7, 2021

認証、承認、および監査の構成の基本的なコンポーネントは次のとおりです。

- 認証仮想サーバー -すべての認証要求は、トラフィック管理仮想サーバー（負荷分散またはコンテンツスイッチング）によって認証仮想サーバーにリダイレクトされます。この仮想サーバは、関連付けられた認証ポリシーを処理し、それに応じてアプリケーションへのアクセスを提供します。詳細については、「認証仮想サーバー」を参照してください。
- 認証プロファイル -認証プロファイルは、認証仮想サーバー、認証ホスト、認証ドメイン、および認証レベルを指定します。

1 つ以上の認証プロファイルを作成して、さまざまな認証設定を指定し、要件に基づいてこれらの認証プロファイルを関連するトラフィック管理サーバーにバインドできます。詳細については、認証プロファイルを参照してください。

- 認証ポリシー -ユーザーが Citrix ADC または Citrix Gateway アプライアンスにログオンすると、ユーザーは作成したポリシーに従って認証されます。認証ポリシーは、式とアクションで構成されます。認証ポリシーでは、Citrix ADC 式が使用されます。詳細については、「認証ポリシー」を参照してください。
- 承認ポリシー -承認ポリシーを構成するときに、内部ネットワーク内のネットワークリソースへのアクセスを許可または拒否するように設定できます。詳細については、承認ポリシーを参照してください。
- ユーザーとグループ： -認証、承認、および監査の基本設定を構成した後、ユーザーとグループを作成します。まず、Citrix ADC アプライアンスを介して認証を行うユーザーごとにユーザーアカウントを作成します。Citrix ADC アプライアンス自体によって制御されるローカル認証を使用している場合は、ローカルユーザーアカウントを作成し、それらの各アカウントにパスワードを割り当てます。詳細については、「ユーザーとグループ」を参照してください。

認証仮想サーバー

January 31, 2022

トラフィック管理仮想サーバー（負荷分散またはコンテンツスイッチング）は、すべての認証要求を認証仮想サーバーにリダイレクトします。この仮想サーバは、関連付けられた認証ポリシーを処理し、それに応じてアプリケーションへのアクセスを提供します。

注：トラフィック管理ポリシーを認証、承認、および監査仮想サーバーにバインドすることはできません。

認証仮想サーバーをセットアップする

認証仮想サーバーのセットアップに関連する手順は次のとおりです。

1. 認証、認可、および監査機能を有効にします。

```
enable ns feature AAA
<!--NeedCopy-->
```

2. 認証仮想サーバを設定します。タイプは SSL である必要があり、SSL 証明書とキーのペアを仮想サーバーにバインドするようにしてください。

```
add authentication vserver <name> SSL <ipaddress> <port>

bind ssl certkey <auth-vserver-name> <certkey>
<!--NeedCopy-->
```

3. 認証仮想サーバーのドメインの FQDN を指定します。

```
set authentication vserver <name> -authenticationDomain <FQDN>
<!--NeedCopy-->
```

4. 認証仮想サーバを、関連するトラフィック管理仮想サーバに関連付けます。

   注意事項:

   - ドメインセッション cookie が正しく機能するためには、トラフィック管理仮想サーバの FQDN が、認証仮想サーバの FQDN と同じドメインにある必要があります。トラフィック管理仮想サーバで、次の手順を実行します。
     - 認証を有効にします。
     - トラフィック管理仮想サーバの認証ホストとして、認証仮想サーバの FQDN を指定します。
     - [（オプション）] トラフィック管理仮想サーバーの認証ドメインを指定します。
     - 認証ドメインを構成しない場合、アプライアンスは、ホスト名部分のない認証仮想サーバーの FQDN で構成される FQDN を割り当てます。例えば、場合には、認証仮想サーバのドメイン名は、認証ドメインとして **tm.xyz.bar.com**、アプライアンス割り当て **xyz.bar.com** です。
       * 負荷分散の場合:

```
set lb vserver <name> -authentication ON -
    authenticationhost <FQDN> [-authenticationdomain <
    authdomain>]
<!--NeedCopy-->
```

       * コンテンツスイッチングの場合:


```
set cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

- 認証ドメインにドメイン全体の Cookie を設定する必要がある場合は、負荷分散仮想サーバーで認証プロファイルを有効にする必要があります。

5. 両方の仮想サーバが稼動していて、正しく設定されていることを確認します。

```
show authentication vserver <name>
<!--NeedCopy-->
```

GUI を使用して認証仮想サーバーをセットアップするには

1. 認証、認可、および監査機能を有効にします。

   [システム] **>** [設定] に移動し、[基本機能の構成] をクリックして 、認証、承認、監査を有効にします。

2. 認証仮想サーバを設定します。

   Security > AAA - Application Traffic > Virtual Servers に移動して、必要に応じて構成します。

3. 認証用にトラフィック管理仮想サーバを設定します。

   - 負荷分散の場合:

     Traffic Management > Load Balancing > Virtual Servers に移動して必要に応じて仮想サーバーを構成します。

   - コンテンツスイッチングの場合:

     Traffic Management > Content Switching > Virtual Servers に移動して、必要に応じて仮想サーバーを構成します。

4. - 認証の設定を確認します。

     [セキュリティ] **> [AAA-**アプリケーショントラフィック] **>** [仮想サーバ] に移動し、関連する認証仮想サーバの詳細を確認します。

認証仮想サーバーを構成します

認証、認可、および監査を構成するには、まず認証トラフィックを処理するように認証仮想サーバーを構成します。
次に、SSL 証明書とキーのペアを仮想サーバーにバインドして、SSL 接続を処理できるようにします。
SSL の構成および証明書とキーのペアの作成の詳細については、「SSL 証明書」を参照してください。

CLI を使用して認証仮想サーバーを構成します

認証仮想サーバーを構成して構成を確認するには、コマンドプロンプトで次のコマンドを同じ順序で入力します。

```
dd authentication vserver <name> ssl <ipaddress>

show authentication vserver <name>

bind ssl certkey <certkeyName>

show authentication vserver <name>

set authentication vserver <name>

show authentication vserver <name>
<!--NeedCopy-->
```

例**:**

```
add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
    Primary Vserver On Down : DISABLED Authentication : ON Current AAA
    Users: 0 Done

bind ssl certkey Auth-Vserver-2 Auth-Cert-1 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: UP Client Idle Timeout
    : 180 sec Down state flush: DISABLED Disable Primary Vserver On Down
     : DISABLED Authentication : ON Current AAA Users: 0 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
    Primary Vserver On Down : DISABLED Authentication : ON Current AAA
    Users: 0 Done
<!--NeedCopy-->
```


> 注:
>
> 認証ドメインパラメーターは非推奨です。ドメイン全体のクッキーを設定するには、認証プロファイルを使用します。

GUI を使用して認証仮想サーバーを構成する

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** 仮想サーバに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しい認証仮想サーバーを作成するには、[追加] をクリックします。
   - 既存の認証仮想サーバーを変更するには、仮想サーバーを選択し、[編集] をクリックします。[基本設定] 領域を展開した状態で、[構成] ダイアログが開きます。
3. パラメータの値を次のように指定します (アスタリスクは必須パラメータを示します)。
   - 名前 *—name (以前に作成した仮想サーバーでは変更できません)
   - IP アドレス Type*—IP 認証仮想サーバーのアドレスタイプ
   - [IP Address*]: 認証仮想サーバの IP アドレス
   - [Port*]: 仮想サーバが接続を受け付ける TCP ポート。
   - ログインタイムアウトの失敗-failedLoginTimeout（ログインが失敗するまでの秒数。ユーザーはログインプロセスを再開する必要があります。）
   - 最大ログイン試行回数: maxLoginAttempts（ユーザーがロックアウトされるまでに許可されるログイン試行回数）

   > 注:
   >
   > 認証仮想サーバーは SSL プロトコルとポート 443 のみを使用するため、これらのオプションはグレー表示されます。言及されていないオプションは無視できます。
4. [続行] をクリックして、[証明書] 領域を表示します。
5. [証明書**]** 領域で、この仮想サーバーで使用する SSL 証明書を構成します。
   - CA 証明書を構成するには、[CA 証明書] の右側にある矢印をクリックして [CA Cert Key] ダイアログボックスを表示し、この仮想サーバーにバインドする証明書を選択して [保存] をクリックします。
   - サーバー証明書を構成するには、サーバー証明書の右側にある矢印をクリックし、CA 証明書の場合と同じプロセスに従います。
6. [続行] をクリックして、[高度な認証ポリシー] 領域を表示します。
7. 高度な認証ポリシーを仮想サーバーにバインドする場合は、行の右側にある矢印をクリックして [認証ポリシー] ダイアログボックスを表示し、サーバーにバインドするポリシーを選択し、優先順位を設定して [**OK**] をクリックします。
8. [続行] をクリックして、[基本認証ポリシー**]** 領域を表示します。

9. 基本認証ポリシーを作成して仮想サーバにバインドする場合は、プラス記号をクリックして [**Policies**] ダイアログ・ボックスを表示し、プロンプトに従ってポリシーを構成し、この仮想サーバにバインドします。
10. [続行] をクリックして、401 ベースの仮想サーバー領域を表示します。
11. [401 ベースの仮想サーバー] 領域で、この仮想サーバーにバインドする負荷分散またはコンテンツスイッチ仮想サーバーを構成します。
    - 負荷分散仮想サーバーをバインドするには、負荷分散仮想サーバーの右側にある矢印をクリックして [負荷分散仮想サーバー] ダイアログボックスを表示し、プロンプトに従います。
    - コンテンツスイッチング仮想サーバーをバインドするには、コンテンツスイッチング仮想サーバーの右側にある矢印をクリックして [コンテンツスイッチング仮想サーバー] ダイアログボックスを表示し、LB 仮想サーバーをバインドするのと同じプロセスに従います。
12. グループを作成または設定する場合は、[グループ] 領域で矢印をクリックして [グループ] ダイアログボックスを表示し、プロンプトに従います。
13. 設定を確認し、完了したら、[完了] をクリックします。ダイアログボックスが閉じます。新しい認証仮想サーバを作成した場合は、そのサーバが [**Configuration**] ウィンドウのリストに表示されます。

トラフィック管理仮想サーバー

認証仮想サーバーを作成および構成したら、次にトラフィック管理仮想サーバーを作成または構成し、認証仮想サーバーをそれに関連付けます。トラフィック管理仮想サーバーには、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーを使用できます。
いずれかのタイプの仮想サーバーの作成と構成の詳細については、「*Citrix* トラフィック管理ガイド」の「トラフィック管理」を参照してください。

> 注:
>
> ドメインセッション Cookie が正しく機能するには、トラフィック管理仮想サーバーの FQDN が認証仮想サーバーの FQDN と同じドメインにある必要があります。

認証を有効にし、認証サーバの FQDN をトラフィック管理仮想サーバに割り当てることで、認証、認可、および監査用にトラフィック管理仮想サーバを設定します。現在、トラフィック管理仮想サーバーで認証ドメインを構成することもできます。このオプションを構成しない場合、Citrix ADC アプライアンスは、トラフィック管理仮想サーバーに、ホスト名部分のない認証仮想サーバーの FQDN で構成される FQDN を割り当てます。たとえば、認証仮想サーバーのドメイン名が tm.xyz.bar.com の場合、アプライアンスは xyz.bar.com を割り当てます。認証ドメインとして。

CLI を使用してトラフィック管理仮想サーバーを構成するには

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。


```
set lb vserver <name>  -authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]
show lb vserver <name>
set cs vserver <name>  -authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]
show cs vserver <name>
<!--NeedCopy-->
```

例**:**

```
set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost mywiki.index.com Done

show lb vserver vs-cont-sw vs-cont-sw (0.0.0.0:0) - TCP Type: ADDRESS State: DOWN Last state change was at Wed Aug 19 10:03:15 2009 (+410 ms) Time since last state change: 5 days, 20:00:40.290 Effective State: DOWN Client Idle Timeout: 9000 sec Down state flush: ENABLED Disable Primary Vserver On Down : DISABLED No. of Bound Services : 0 (Total) 0 (Active) Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Connection Failover: DISABLED Authentication: ON Host: mywiki.index.com
Done
<!--NeedCopy-->
```

GUI を使用してトラフィック管理仮想サーバーを構成するには

1. ナビゲーションペインで、次のいずれかの操作を行います。
   - **Traffic Management > Load Balancing > Virtual Servers** に移動します。
   - [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動します。
   - 詳細ウィンドウで、認証を有効にする仮想サーバーを選択し、[編集] をクリックします。
   - 「ドメイン」テキスト・ボックスに、認証ドメインを入力します。
   - 右側の [詳細設定] メニューで、[認証] を選択します。
   - [フォームベース認証] または [**401** ベース認証] のいずれかを選択し、認証情報を入力します。
     - フォームベース認証の場合は、認証 FQDN（認証サーバーの完全修飾ドメイン名）、認証仮想サーバー（認証仮想サーバーの IP アドレス）、および認証プロファイル（認証に使用するプロファイル）を入力します。
     - 401 ベースの認証の場合は、認証仮想サーバーと認証プロファイルのみを入力します。

- [**OK**] をクリックします。仮想サーバーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

認証、承認、監査のためのシンプルなログインプロトコルのサポート

認証、認可、および監査トラフィック管理仮想サーバと認証、認可、監査仮想サーバ間のログインプロトコルは、クエリパラメータを使用して暗号化されたデータを送信するのではなく、内部メカニズムを使用するように簡略化されます。この機能を使用すると、リクエストの再生が防止されます。

DNS を構成する

認証プロセスで使用されるドメインセッション Cookie が正しく機能するには、認証とトラフィック管理の両方の仮想サーバーを同じドメイン内の FQDN に割り当てるように DNS を構成する必要があります。DNS アドレスレコードを構成する方法については、「ドメインネームシステム」を参照してください。

認証仮想サーバーを確認する

認証およびトラフィック管理仮想サーバーを構成した後、ユーザーアカウントを作成する前に、両方の仮想サーバーが正しく構成され、UP 状態にあることを確認する必要があります。

CLI を使用して noAuth 認証を構成します

コマンドプロンプトで、次のコマンドを入力します。

```
show authentication vserver <name>
<!--NeedCopy-->
```

例:

```
show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
Done
<!--NeedCopy-->
```


GUI を使用して **noAuth** 認証を構成します

1. **Security > Citrix ADC AAA - Application Traffic > Virtual Servers** に移動します。
   注：Citrix Gateway から、**Citrix Gateway > Virtual Servers** に移動します。
2. **AAA Virtual Servers** ペインの情報を確認して、設定が正しく、認証仮想サーバがトラフィックを受け入れていることを確認します。特定の仮想サーバーを選択して、詳細情報を詳細ウィンドウに表示できます。

承認ポリシー

October 7, 2021

承認ポリシーを構成するときに、内部ネットワーク内のネットワークリソースへのアクセスを許可または拒否するように設定できます。たとえば、ユーザーが 10.3.3.0 ネットワークにアクセスできるようにするには、次の式を使用します。

CLIENT.IP.DST.IN_SUBNET(10.3.0.0/16)

承認ポリシーはユーザーとグループに適用されます。ユーザーが認証された後、Citrix Gateway は、RADIUS、LDAP、または TACACS + サーバーのいずれかからユーザーのグループ情報を取得することにより、グループ認証チェックを実行します。ユーザーがグループ情報を利用できる場合、Citrix Gateway はグループに許可されているネットワークリソースを確認します。

ユーザーがアクセスできるリソースを制御するには、承認ポリシーを作成する必要があります。許可ポリシーを作成する必要がない場合は、デフォルトのグローバル許可を構成できます。

承認ポリシー内でファイルパスへのアクセスを拒否する式を作成する場合は、サブディレクトリパスのみを使用でき、ルートディレクトリは使用できません。たとえば、fs.path を使用します “\\dir1\\dir2” fs.path の代わりに “\\rootdir\\dir1\\dir2”. この例で 2 番目のバージョンを使用すると、ポリシーは失敗します。

承認ポリシーを構成した後、それをユーザーまたはグループにバインドします。

デフォルトでは、承認ポリシーは、最初に仮想サーバーにバインドするポリシーに対して検証され、次にグローバルにバインドされるポリシーに対して検証されます。ポリシーをグローバルにバインドし、ユーザー、グループ、または仮想サーバーにバインドするポリシーよりもグローバルポリシーを優先する場合は、ポリシーの優先順位番号を変更できます。優先順位番号はゼロから始まります。優先順位の数値が小さいほど、ポリシーの優先順位が高くなります。

たとえば、グローバルポリシーの優先度が 1 で、ユーザーの優先度が 2 の場合、グローバル認証ポリシーが最初に適用されます。

> 重要：
>
> - 従来の承認ポリシーは、TCP トラフィックにのみ適用されます。
> - 高度な承認ポリシーは、すべてのタイプのトラフィックに適用できます (TCP/UDP/ICMP/DNS).
>   - To apply policy on UDP/ICMP/DNS traffic, policies must be bound at type UDP_REQUEST,

> ICMP_REQUEST, and DNS_REQUEST respectively.
>
> - While binding, if “type” is not explicitly mentioned or “type” is set to REQUEST, the behavior does not change from earlier builds, that is these policies are applied only to TCP traffic.
> - The policies bound at UDP_REQUEST do not apply for DNS traffic. For DNS, policies must be explicitly bound to DNS_REQUEST TCP_DNS is similar to other TCP requests.

高度な承認ポリシーの詳細については、「https://support.citrix.com/article/CTX232237」を参照してください。

承認ポリシーを構成してバインドする

GUI を使用して許可ポリシーを構成します

1. **Citrix Gateway >** ［ポリシー］ **>** ［認証］ に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [名前] に、ポリシーの名前を入力します。
4. **[**アクション**]** で、**[**許可**]** または **[** 拒否**]** を選択します。
5. 式では、式エディタをクリックします。
6. 式の構成を開始するには、[選択] をクリックして、必要な要素を選択します。
7. 式が完成したら、[完了] をクリックします。
8. ［作成］ をクリックします。

GUI を使用して、承認ポリシーをユーザーにバインドします

1. **Citrix Gateway >** ユーザー管理に移動します。
2. **AAA** ユーザーをクリックします。
3. 詳細ペインで、ユーザーを選択し、[編集] をクリックします。
4. [詳細設定] で、[承認ポリシー] をクリックします。
5. [ポリシーのバインド] ページで、ポリシーを選択するか、ポリシーを作成します。
6. [優先度] で、優先度番号を設定します。
7. [タイプ] で、リクエストタイプを選択し、[**OK**] をクリックします。

GUI を使用して、承認ポリシーをグループにバインドします

1. **Citrix Gateway > User Administration** に移動します。
2. **AAA** グループをクリックします。
3. 詳細ペインで、グループを選択し、[編集] をクリックします。
4. [詳細設定] で、[承認ポリシー] をクリックします。
5. [ポリシーのバインド] ページで、ポリシーを選択するか、ポリシーを作成します。
6. [優先度] で、優先度番号を設定します。

7. [タイプ] で、リクエストタイプを選択し、[**OK**] をクリックします。

承認は、ユーザーが Citrix Gateway にログオンするときにアクセスできるネットワークリソースを指定します。承認のデフォルト設定は、すべてのネットワークリソースへのアクセスを拒否することです。デフォルトのグローバル設定を使用し、承認ポリシーを作成して、ユーザーがアクセスできるネットワークリソースを定義することをお勧めします。

承認ポリシーと式を使用して、Citrix Gateway で承認を構成します。許可ポリシーを作成したら、アプライアンスで構成したユーザーまたはグループにポリシーをバインドできます。

デフォルトのグローバル認証

ユーザーが内部ネットワーク上でアクセスできるリソースを定義するために、デフォルトのグローバル認証を構成できます。内部ネットワーク上のネットワークリソースへのアクセスをグローバルに許可または拒否することにより、グローバル認証を構成します。

作成したグローバル承認アクションは、直接またはグループを介して、承認ポリシーがまだ関連付けられていないすべてのユーザーに適用されます。ユーザーまたはグループの承認ポリシーは、常にグローバル承認アクションを上書きします。デフォルトの承認アクションが [拒否] に設定されている場合は、すべてのユーザーまたはグループに承認ポリシーを適用して、それらのユーザーまたはグループがネットワークリソースにアクセスできるようにする必要があります。この要件は、セキュリティの向上に役立ちます。

デフォルトのグローバル認証を設定するには:

1. ナビゲーションペインにある、[Configuration] タブの構成ユーティリティで、[Citrix Gateway] を展開し、[Global Settings] をクリックします。
2. 詳細ペインの [設定] で、[グローバル設定の変更] をクリックします。
3. [セキュリティ] タブの [デフォルトの承認アクション] の横にある [許可] または [拒否] を選択し、[OK] をクリックします。

認証プロファイル

December 7, 2021

複数のトラフィック管理仮想サーバーで同じ認証設定を使用する場合は、認証仮想サーバー、認証ホスト、認証ドメイン、および認証レベルを指定する認証プロファイルを作成できます。

この認証プロファイルは、関連するトラフィック管理仮想サーバに関連付けることができます。

認証プロファイルを構成する

CLI を使用して認証プロファイルを構成します

- 認証プロファイルを作成し、必要なパラメータを設定します。

たとえば、「authVS」という名前の認証仮想サーバーを使用してプロファイルを作成します。

```
add authentication authnProfile authProfile1 -authnVsName authVS
    -authenticationHost authnVS.example.com -authenticationDomain
    example.com -authenticationLevel
<!--NeedCopy-->
```

> 注:
>
> 認証の重みまたはレベルは、トラフィックがバインドされている仮想サーバーによって異なります。特定のレベルのトラフィック管理仮想サーバーに対して認証によって作成されたセッションを使用して、より高いレベルのトラフィック管理仮想サーバーにアクセスすることはできません。

- 認証プロファイルを関連するトラフィック管理仮想サーバにバインドします。

  たとえば、authProfile1 を「vserver1」という名前の負荷分散仮想サーバーにバインドします。

```
set lb vserver vserver1 -authnProfile authProfile1
<!--NeedCopy-->
```

GUI を使用して認証プロファイルを構成する

[設定] タブで、[セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**認証プロファイル] に移動し、必要に応じて認証プロファイルを設定します。

> 注:
>
> - Citrix Gateway ウィザードを使用して認証プロファイルを作成することもできます。プロファイルには、認証ポリシーのすべての設定が含まれています。認証ポリシーを作成するときにプロファイルを構成します。
> - Citrix Gateway Wizard を使用すると、選択した認証タイプを使用して認証を構成できます。ウィザードの実行後に他の認証ポリシーを構成する場合は、構成ユーティリティを使用できます。Citrix Gateway ウィザードの詳細については、「Citrix Gateway ウィザードを使用した設定の構成」を参照してください。

認証ポリシー

February 21, 2022

ユーザーが Citrix ADC または Citrix Gateway アプライアンスにログオンすると、作成したポリシーに従って認証されます。認証ポリシーは式とアクションで構成されます。認証ポリシーでは、Citrix ADC 式が使用されます。

認証アクションと認証ポリシーを作成したら、それを認証仮想サーバーにバインドし、優先度を割り当てます。バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。プライマリポリシーは、セカンダリポリシーよりも先に評価されます。両方のタイプのポリシーを使用する設定では、通常、プライマリポリシーはより限定的なポリシーですが、セカンダリポリシーは通常、より一般的なポリシーです。これは、より具体的な基準を満たさないユーザーアカウントの認証を処理することを目的としています。ポリシーは、認証タイプを定義します。単一の認証ポリシーは、単純な認証のニーズに使用でき、通常はグローバルレベルでバインドされます。デフォルトの認証タイプ (ローカル) を使用することもできます。ローカル認証を設定する場合は、アプライアンス上でユーザとグループも設定する必要があります。

複数の認証ポリシーを設定し、それらをバインドして、詳細な認証手順と仮想サーバーを作成できます。たとえば、複数のポリシーを設定して、カスケード認証と 2 要素認証を設定できます。また、認証ポリシーの優先度を設定して、アプライアンスがユーザクレデンシャルをチェックするサーバと順序を決定することもできます。認証ポリシーには、式とアクションが含まれます。たとえば、式を True 値に設定した場合、ユーザーのログオン時にアクションによってユーザーログオンが true と評価され、ユーザーはネットワークリソースにアクセスできます。

認証ポリシーを作成したら、ポリシーをグローバルレベルまたは仮想サーバにバインドします。少なくとも 1 つの認証ポリシーを仮想サーバーにバインドすると、グローバル認証の種類が仮想サーバーにバインドされたポリシーよりも優先されない限り、ユーザーが仮想サーバーにログオンするときに、グローバルレベルにバインドした認証ポリシーは使用されません。

ユーザーがアプライアンスにログオンすると、認証は次の順序で評価されます。

- 仮想サーバは、バインドされた認証ポリシーがあるかどうかがチェックされます。
- 認証ポリシーが仮想サーバにバインドされていない場合、アプライアンスはグローバル認証ポリシーをチェックします。
- 認証ポリシーが仮想サーバまたはグローバルにバインドされていない場合、ユーザはデフォルトの認証タイプで認証されます。

LDAP および RADIUS 認証ポリシーを構成し、2 要素認証用にポリシーをグローバルにバインドする場合は、設定ユーティリティでポリシーを選択し、ポリシーがプライマリ認証タイプかセカンダリ認証タイプかを選択できます。グループ抽出ポリシーを構成することもできます。

> 注:
>
> Citrix ADC または Citrix Gateway アプライアンスは、認証用に UTF-8 文字のみをエンコードし、ISO-8859-1 文字を使用するサーバーとは互換性がありません。

認証ポリシーを作成する

GUI を使用して認証ポリシーを作成する

1. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認証] に移動し、作成するポリシー

のタイプを選択します。
Citrix Gateway の場合は、**Citrix Gateway >** ポリシー **>** 認証に移動します。

2. 詳細ウィンドウの [ポリシー] タブで、次のいずれかの操作を行います。
   - 新しいポリシーを作成するには、[追加（**Add**）] をクリックします。
   - 既存のポリシーを変更するには、アクションを選択し、[**Edit**] をクリックします。
3. [認証ポリシーの作成] または [認証ポリシーの構成] ダイアログで、パラメーターの値を入力または選択します。
   - **Name**： ポリシー名（設定済みのアクションでは変更不可）
   - 認証タイプ — `authtype`
   - サーバ — `authVsName`
   - **Expression** — rule (式を入力するには、まず [式] ウィンドウの下にある左端のドロップダウンリストで式のタイプを選択し、次に [式] テキスト領域に直接式を入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、ドロップダウンを使用します)。をその中にリストし、式を作成します。)
4. [作成] または [**OK**] をクリックします。作成したポリシーが [Policies] ページに表示されます。
5. [サーバー] タブをクリックし、詳細ウィンドウで次のいずれかの操作を行います。
   - 既存のサーバを使用するには、そのサーバを選択し、をクリックします。
   - サーバを作成するには、[Add] をクリックし、指示に従います。
6. このポリシーをセカンダリ認証ポリシーとして指定する場合は、[認証] タブで [セカンダリ] をクリックします。このポリシーをプライマリ認証ポリシーとして指定する場合は、この手順をスキップしてください。
7. [ポリシーの挿入] をクリックします。
8. 認証仮想サーバにバインドするポリシーをドロップダウンリストから選択します。
9. 左側の [**Priority**] 列で、デフォルトの優先度を変更して、ポリシーが適切な順序で評価されるようにします。
10. [**OK**] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

GUI を使用して認証ポリシーを変更する

認証サーバの IP アドレスや式など、設定済みの認証ポリシーおよびプロファイルを変更できます。

1. 構成ユーティリティの［構成］タブで、［**Citrix Gateway**］ **>** ［ポリシー］ **>** ［認証］の順に展開します。
   注:[セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認証] からポリシーを設定し、変更するポリシーのタイプを選択することもできます。
2. ナビゲーションペインの [認証] で、認証の種類を選択します。
3. 詳細ウィンドウの [サーバー] タブで、サーバーを選択し、[開く] をクリックします。

GUI を使用して認証ポリシーを削除する

ネットワークから認証サーバーを変更または削除した場合は、対応する認証ポリシーを Citrix Gateway から削除します。

1. 構成ユーティリティの［構成］タブで、［**Citrix Gateway**］**>**［ポリシー］**>**［認証］の順に展開します。
   注:ADC から設定するには、[セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認証] に移動し、削除するポリシーのタイプを選択します。
2. ナビゲーションペインの [認証] で、認証の種類を選択します。
3. 詳細ペインの [ポリシー] タブで、ポリシーを選択し、[削除] をクリックします。

CLI を使用して認証ポリシーを作成する

コマンドプロンプトで、次のコマンドを入力します。

```
add authentication negotiatePolicy <name> <rule> <reqAction>

show authentication localPolicy <name>

bind authentication vserver <name> -policy <policyname> [-priority <
    priority>][-secondary]]

show authentication vserver <name>
<!--NeedCopy-->
```

例**:**

```
add authentication localPolicy Authn-Pol-1 ns_true
Done

show authentication localPolicy
1)      Name: Authn-Pol-1       Rule: ns_true          Request action:
    LOCAL   Done

bind authentication vserver Auth-Vserver-2 -policy Authn-Pol-1
Done

show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT State: UP Client
    Idle
Timeout: 180 sec Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
```



```
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
1)  Primary authentication policy name:  Authn-Pol-1 Priority: 0
Done
<!--NeedCopy-->
```

CLI を使用して認証ポリシーを変更する

コマンドプロンプトで次のコマンドを入力して、既存の認証ポリシーを変更します。

```
set authentication localPolicy <name> <rule> [-reqaction <action>]
<!--NeedCopy-->
```

例

```
set authentication localPolicy Authn-Pol-1 'ns_true'
<!--NeedCopy-->
```

CLI を使用して認証ポリシーを削除する

コマンドプロンプトで次のコマンドを入力して、認証ポリシーを削除します。

```
rm authentication localPolicy <name>
<!--NeedCopy-->
```

例

```
rm authentication localPolicy Authn-Pol-1
<!--NeedCopy-->
```

認証ポリシーをバインドする

認証ポリシーを設定したら、ポリシーをグローバルにバインドするか、仮想サーバにバインドします。いずれかの構成ユーティリティを使用して、認証ポリシーをバインドできます。

構成ユーティリティを使用して認証ポリシーをグローバルにバインドするには、次の手順に従います。

1. 構成ユーティリティの［構成］タブで、［**Citrix Gateway**］**>**［ポリシー］**>**［認証］の順に展開します。
   注:ADC から設定するために、**Security > AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証をナビゲートして下さい
2. 認証タイプをクリックします。
3. 詳細ウィンドウの [ポリシー] タブでサーバーをクリックし、[操作] の [グローバルバインディング] をクリックします。
4. [プライマリ] または [セカンダリ] タブの [詳細] で、[ポリシーの挿入
5. ［ポリシー名］でポリシーを選択し、［**OK**］をクリックします。

   注：ポリシーを選択すると、Citrix Gateway によって式が自動的に真値に設定されます。

構成ユーティリティを使用してグローバル認証ポリシーをバインド解除するには、次の手順を実行します。

1. 構成ユーティリティの［構成］タブで、［**Citrix Gateway**］**>**［ポリシー］**>**［認証］の順に展開します。
   注:ADC から設定するために、**Security > AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証をナビゲートして下さい
2. [ポリシー] タブの [操作] で、[グローバルバインディング] をクリックします。
3. [認証ポリシーをグローバルにバインド/バインド解除] ダイアログボックスの [プライマリ] タブまたは [セカンダリ] タブの [ポリシー名] でポリシーを選択し、[ポリシーのバインド解除] をクリックし、[**OK**] をクリックします。

認証アクションを追加する

CLI を使用して認証アクションを追加する

LOCAL 認証を使用しない場合は、明示的な認証アクションを追加する必要があります。コマンドプロンプトで、次のコマンドを入力します。

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812 -authtimeout 15 -tacacsSecret ”minotaur” -
    authorization OFF -accounting ON -auditFailedCmds OFF -
    defaultAuthenticationGroup ”users”
<!--NeedCopy-->
```


CLI を使用して認証アクションを設定する

既存の認証アクションを構成するには、コマンドプロンプトで次のコマンドを入力します。

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
set authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812 -authtimeout 15 -tacacsSecret ”minotaur” -
    authorization OFF -accounting ON -auditFailedCmds OFF -
    defaultAuthenticationGroup ”users”
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証アクションを削除する

既存の RADIUS アクションを削除するには、コマンドプロンプトで次のコマンドを入力します。

```
rm authentication radiusAction <name>
<!--NeedCopy-->
```

例

```
rm authentication tacacsaction Authn-Act-1
<!--NeedCopy-->
```

NoAuth 認証

Citrix ADC アプライアンスは NoAuth 認証機能をサポートしており、ユーザーがこのポリシーを実行したときに、noAuthAction コマンドで DefaultAuthenticationGroup パラメーターを構成できます。管理者は、ユーザーのグループ内にこのグループが存在するかどうかをチェックして、NoAuth ポリシーによるユーザーのナビゲーションを判断できます。

コマンドラインインターフェイスを使用して **NoAuth** 認証を設定するには

コマンドプロンプトで次を入力します。

```
add authentication noAuthAction <name> [-defaultAuthenticationGroup <
    string>]
<!--NeedCopy-->
```

例

```
add authentication noAuthAction noauthact  -defaultAuthenticationGroup
    mynoauthgroup
<!--NeedCopy-->
```

デフォルトのグローバル認証タイプ

Citrix Gateway をインストールして Citrix Gateway ウィザードを実行すると、ウィザード内で認証を構成しました。この認証ポリシーは、Citrix Gateway グローバルレベルに自動的にバインドされます。Citrix Gateway ウィザードで構成する認証の種類は、デフォルトの認証の種類です。デフォルトの認証の種類を変更するには、Citrix Gateway ウィザードを再度実行するか、構成ユーティリティでグローバル認証設定を変更します。

他の認証の種類を追加する必要がある場合は、Citrix Gateway で認証ポリシーを構成し、構成ユーティリティを使用してポリシーを Citrix Gateway にバインドできます。認証をグローバルに設定する場合は、認証のタイプを定義し、設定を構成し、認証できる最大ユーザ数を設定します。

ポリシーを設定してバインドしたら、優先度を設定して、優先する認証タイプを定義できます。たとえば、LDAP および RADIUS 認証ポリシーを設定します。LDAP ポリシーのプライオリティ番号が 10 で、RADIUS ポリシーのプライオリティ番号が 15 の場合、各ポリシーをバインドする場所に関係なく、LDAP ポリシーが優先されます。これをカスケード認証と呼びます。

ログオンページは、Citrix Gateway のメモリ内キャッシュまたは Citrix Gateway で実行されている HTTP サーバーから配信するように選択できます。インメモリキャッシュからログオンページを配信することを選択した場合、Citrix Gateway からのログオンページの配信は、HTTP サーバーからのログオンページの配信よりも高速です。インメモリキャッシュからログオンページを配信するように選択すると、多数のユーザーが同時にログオンする場合の待ち時間が短縮されます。キャッシュからのログオンページの配信は、グローバル認証ポリシーの一部としてのみ構成できます。

また、認証用の特定の IP アドレスであるネットワークアドレス変換（NAT）IP アドレスを設定することもできます。この IP アドレスは認証用に一意であり、Citrix Gateway のサブネット、マッピング、または仮想 IP アドレスではありません。この設定はオプションです。

> 注:
>
> - Citrix Gateway ウィザードを使用して SAML 認証を構成することはできません。
> - クイック構成ウィザードを使用して、LDAP、RADIUS、およびクライアント証明書認証を構成できます。ウィザードを実行すると、Citrix Gateway で構成された既存の LDAP サーバーまたは RADIUS サーバーから選択できます。LDAP または RADIUS の設定を構成することもできます。2 要素認証を使用する場合は、プライマリ認証タイプとして LDAP を使用することをお勧めします。

デフォルトのグローバル認証タイプを構成する

1. GUI の［構成］タブのナビゲーションペインで［**Citrix Gateway**］を展開し、［グローバル設定］をクリックします。
2. 詳細ウィンドウの [設定] で、[認証設定の変更] をクリックします。
3. [最大ユーザー数] に、この認証の種類を使用して認証できるユーザーの数を入力します。
4. [**NAT IP** アドレス] に、認証用の一意の IP アドレスを入力します。
5. [静的キャッシュを有効にする**]** を選択すると、ログオンページが高速に配信されます。
6. 認証が失敗した場合にユーザーにメッセージを表示するには、**[**拡張認証フィードバックを有効にする] を選択します。ユーザーが受け取るメッセージには、パスワードエラー、アカウントが無効またはロックされている、またはユーザーが見つからない、などが含まれます。
7. [既定の認証タイプ] で、認証タイプを選択します。
8. 使用する認証タイプの設定を構成し、「**OK**」をクリックします。

ユーザーの現在のログイン試行回数の取得をサポート

Citrix ADC アプライアンスには、ユーザーの現在のログイン試行回数の値を新しい式`aaa.user.login_attempts`で取得するオプションがあります。式は 1 つの引数 (ユーザー名) を取るか、引数を取らないかのいずれかを取ります。引数がない場合、エクスプレッションは`aaa_session`または`aaa_info`からユーザ名を取得します。

`aaa.user.login_attempts`式を認証ポリシーとともに使用して、さらに処理することができます。

CLI を使用してユーザ **1** 人あたりのログイン試行回数を設定するには

コマンドプロンプトで入力します。

```
add expression er aaa.user.login_attempts
```

ユーザーおよびグループ

October 7, 2021

認証、承認、および監査の基本設定を構成したら、ユーザーとグループを作成します。最初に、Citrix ADC アプライアンスを介して認証する各ユーザーのユーザーアカウントを作成します。Citrix ADC アプライアンス自体によって制御されるローカル認証を使用している場合は、ローカルユーザーアカウントを作成し、それらの各アカウントにパスワードを割り当てます。

また、外部認証サーバーを使用している場合は、Citrix ADC アプライアンスにユーザーアカウントを作成します。ただし、この場合、各ユーザーアカウントは外部認証サーバー上のそのユーザーのアカウントと正確に一致する必要があります。また、Citrix ADC で作成したユーザーアカウントにはパスワードを割り当てないでください。外部認証サーバは、外部認証サーバで認証するユーザのパスワードを管理します。

外部認証サーバーを使用している場合は、一時的なユーザー（訪問者など）のログインを許可するが、認証サーバー上にそれらのユーザーのエントリを作成したくない場合など、Citrix ADC アプライアンスにローカルユーザーアカウントを作成できます。すべてのユーザーアカウントにローカル認証を使用する場合と同様に、各ローカルユーザーアカウントにパスワードを割り当てます。

各ユーザーアカウントは、認証と承認のポリシーにバインドする必要があります。このタスクを簡略化するために、1 つ以上のグループを作成し、それらにユーザーアカウントを割り当てることができます。その後、個々のユーザーアカウントではなくグループにポリシーをバインドできます。

グループでポリシーを構成する

グループを構成した後、[グループ] ダイアログボックスを使用して、ユーザーアクセスを指定するポリシーと設定を適用できます。ローカル認証を使用している場合は、ユーザーを作成し、Citrix Gateway で構成されているグループに追加します。その後、ユーザーはそのグループの設定を継承します。

[グループ] ダイアログボックスで、ユーザーのグループに対して次のポリシーまたは設定を構成できます。

- ユーザー
- 承認ポリシー
- 監査ポリシー
- セッション・ポリシー
- トラフィックポリシー
- ブックマーク
- イントラネットアプリケーション
- イントラネット IP アドレス

構成では、複数のグループに属するユーザーがいる場合があります。さらに、各グループには、異なるパラメータが設定された 1 つ以上のバインドされたセッションポリシーがある場合があります。複数のグループに属するユーザーは、ユーザーが属するすべてのグループに割り当てられたセッションポリシーを継承します。どのセッションポリシー評価が他よりも優先されるようにするには、セッションポリシーの優先度を設定する必要があります。

たとえば、ホームページ www.homepage1.com で構成されたセッションポリシーにバインドされた group1 があります。Group2 は、ホームページ www.homepage2.com で構成されたセッションポリシーにバインドされています。これらのポリシーが優先順位番号なしまたは同じ優先順位番号を持つそれぞれのグループにバインドされてい

る場合、両方のグループに属するユーザーに表示されるホームページは、どちらのポリシーが最初に処理されるかによって異なります。ホームページ www.homepage1.com とのセッションポリシーに低い優先順位を設定することで、両方のグループに属するユーザーがホームページ www.homepage1.com を確実に受信できるようになります。

セッションポリシーに優先順位番号が割り当てられていないか、同じ優先順位番号がある場合、優先順位は次の順序で評価されます。

- ユーザー
- グループ
- 仮想サーバ
- グローバル

ポリシーが優先順位番号なしで同じレベルにバインドされている場合、またはポリシーの優先順位番号が同じである場合、評価の順序はポリシーバインドの順序に従います。最初にレベルにバインドされたポリシーは、後でバインドされたポリシーよりも優先されます。

ユーザーが複数のグループにバインドされており、各グループに IIP がバインドされている場合、ユーザーはバインドされたグループのいずれかから無料の IP を取得できます。

ユーザーとグループを作成する

GUI を使用して、ローカルユーザーの認証、承認、および監査を構成します

1. **Security > AAA - Application Traffic > Users** に移動します
   Citrix Gateway から、**Citrix Gateway> User Administration** を展開し、**AAA Users** をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいユーザーアカウントを作成するには、[追加] をクリックします。
   - 既存のユーザーアカウントを変更するには、ユーザーアカウントを選択し、[開く] をクリックします。
3. **[AAA User**] ダイアログボックスの [**User Name**] テキストボックスに、ユーザの名前を入力します。
4. ローカル認証されたユーザーアカウントを作成する場合は、[外部認証**]** チェックボックスをオフにして、ユーザーがログオンに使用するローカルパスワードを入力します。
5. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。ステータスバーに、ユーザーが正常に構成されたことを示すメッセージが表示されます。

認証、承認、および監査のローカルグループを構成し、構成ユーティリティを使用してそれらにユーザーを追加します

1. **Security > AAA - Application Traffic > Groups** に移動します
   Citrix Gateway から、**Citrix Gateway> User Administration** を展開し、**AAA Groups** をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。

- 新しいグループを作成するには、[追加] をクリックします。
- 既存のグループを変更するには、グループを選択し、[編集] をクリックします。

3. 新しいグループを作成する場合は、[**Create AAA Group**] ダイアログボックスの [**Group Name**] テキストボックスに、グループの名前を入力します。
4. 右側の [詳細] 領域で、[**AAA Users**] をクリックします。
   - グループにユーザーを追加するには、ユーザーを選択し、[追加] をクリックします。
   - グループからユーザーを削除するには、ユーザーを選択し、[削除] をクリックします。
   - 新しいユーザーアカウントを作成してグループに追加するには、[プラス] アイコンをクリックし、「構成ユーティリティを使用してローカルユーザーの認証、承認、および監査を構成するには」の手順に従います。
5. [作成] または [**OK]** をクリックします。作成したグループが [**AAA Groups]** ページに表示されます。

GUI を使用してグループを削除する

Citrix Gateway からユーザーグループを削除することもできます。

1. **Security > AAA - Application Traffic > Groups** に移動します
   Citrix Gateway から、**Citrix Gateway> User Administration** を展開し、**AAA Groups** をクリックします。
   詳細ペインでグループを選択し、[削除] をクリックします。

CLI を使用して、ローカルユーザーの認証、承認、および監査を構成します

コマンドプロンプトで、次のコマンドを入力します。

```
add aaa group <groupname>

bind aaa group <groupname> -username <username>
<!--NeedCopy-->
```

例**:**

```
add aaa group group-2

bind aaa group group-2 -username user-2
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して、認証、承認、および監査グループからユーザーを削除します

コマンドプロンプトで、グループにバインドされているユーザーアカウントごとに、次のコマンドを 1 回入力して、グループからユーザーをバインド解除します。

```
unbind aaa group <groupname> -username <username><!--NeedCopy-->
```

```
**例:**

<!--NeedCopy-->
```

unbind aaa group group-hr -username user-hr-1

```
### コマンドラインインターフェイスを使用して、認証、承認、および監査グループを削除します

まず、グループからすべてのユーザーを削除します。次に、コマンドプロンプトで次のコマンドを入力して、Citrix ADC AAAグループを削除し、構成を確認します。

<!--NeedCopy-->
```

rm aaa group

```
**例:**

<!--NeedCopy-->
```

rm aaa group group-hr

```
> **注:**
>
>ドメインなしでユーザー名がすでに追加されている場合、ドメインを使用してユーザー名を追加することはできません。ドメインのあるユーザー名が最初に追加され、次にドメインのない同じユーザー名が追加された場合、Citrix ADCアプライアンスはユーザー名をユーザーリストに追加します。

```



```
次の例は、ドメインなしで同じユーザー名を追加した場合、ドメインを使用したユーザー名の追加が許可されないことを示しています。

<!--NeedCopy-->
```

```
add aaa user u47985
Done
show aaa users
1) UserName: u47985
Done
add aaa user u47985@domain.com
ERROR: User already exists
‘‘‘
```

次の例は、ドメインのあるユーザー名が最初に追加され、その後にドメインのない同じユーザー名が追加され、Citrix ADC アプライアンスがユーザーリストに追加されます。

```
> add aaa user u47985@domain.com
Done
> add aaa user u47985
Done
> sh aaa user
1)   UserName: u47985@domain.com
2)   UserName: u47985
```

‘‘‘

認証方法

October 7, 2021

Citrix ADC アプライアンスは、ローカルユーザーアカウントを使用するか、外部認証サーバーを使用してユーザーを認証できます。アプライアンスは、次の認証タイプをサポートしています。

- ローカル：外部認証サーバーを参照せずに、パスワードを使用して Citrix ADC アプライアンスを認証します。ユーザーデータは、Citrix ADC アプライアンスにローカルに保存されます。
- **RADIUS**：外部 RADIUS サーバーへの認証。
- **LDAP**：外部 LDAP 認証サーバーに対して認証します。
- **TACACS**：外部ターミナルアクセスコントローラアクセス制御システム（TACACS）認証サーバーを認証します。

- **CERT**：外部認証サーバーを参照せずに、クライアント証明書を使用して Citrix ADC アプライアンスを認証します。
- **NEGOTIATE**：Kerberos 認証サーバーに対して認証します。Kerberos 認証でエラーが発生した場合、Citrix ADC は NTLM 認証を使用します。
- **SAML**： Security Assertion Markup Language（SAML）をサポートするサーバーに対して認証します。
- **SAML IDP**：セキュリティアサーションマークアップ言語（SAML）ID プロバイダー（IdP）として機能するように Citrix ADC を構成します。
- **WEB**： Web サーバーを認証し、Web サーバーが HTTP 要求で必要とする資格情報を提供し、Web サーバーの応答を分析して、ユーザー認証が成功したことを確認します。
- ネイティブ **OTP**： Citrix ADC アプライアンスは、サードパーティのサーバーを使用せずにワンタイムパスワード（OTP）をサポートします。
- プッシュ通知：Citrix Gateway は OTP のプッシュ通知をサポートしています。ユーザーは、Citrix Gateway にログインするために、登録されたデバイスで受信した OTP を手動で入力する必要はありません。管理者は、プッシュ通知サービスを使用してユーザーの登録デバイスにログイン通知が送信されるように Citrix Gateway を設定できます。
- 電子メール **OTP**：電子メール OTP 方式では、登録された電子メールアドレスに送信されるワンタイムパスワード（OTP）を使用して認証できます。任意のサービスで認証しようとすると、サーバーはユーザーの登録された電子メールアドレスに OTP を送信します。
- **reCaptcha** 認証**-Citrix Gateway** は、reCaptcha の構成を簡素化する新しいファーストクラスアクション「captchaAction」をサポートしています。reCaptcha はファーストクラスのアクションであるため、独自の要因になる可能性があります。nFactor フローのどこにでも reCaptcha を注入することができます。
- **nFactor** 認証：多要素認証は、アクセスを取得するためにユーザーに複数の ID 証明を提供するように要求することにより、アプリケーションのセキュリティを強化します。Citrix ADC アプライアンスは、多要素認証を構成するための拡張性と柔軟なアプローチを提供します。このアプローチを nFactor 認証と呼びます。
- **OAuth** 認証： OAuth 認証は、Google、Facebook、Twitter などのアプリケーションでホストされているサービスに対してユーザーを承認および認証します。

nFactor 認証

December 7, 2021

> 重要
>
> - nFactor 認証は、NetScaler 11.0 ビルド 62.x 以降でサポートされています。
> - nFactor 認証を Citrix ADC で機能させるには、アドバンスライセンスまたはプレミアムライセンスが必

> 要です。
> - リリース 13.0 ビルド 67.x 以降、nFactor 認証は標準ライセンスでのみサポートされます。Gateway/VPN 仮想サーバー。Citrix Gateway での nFactor 認証の詳細については、「ゲートウェイ認証の nFactor」を参照してください。
> - nFactor 認証は Linux クライアントではサポートされていません。

多要素認証は、アクセス権を付与するために複数の ID をユーザーに要求することで、アプリケーションのセキュリティを強化します。Citrix ADC アプライアンスは、多要素認証を構成するための拡張性と柔軟なアプローチを提供します。このアプローチは *nFactor* 認証と呼ばれます。

nFactor 認証のしくみ

各認証ファクタは、次のタスクを実行します。

- ユーザーから認証情報を収集します。Citrix ADC でサポートされている認証メカニズムには、LDAP、RADIUS、SAML アサーション、クライアント証明書、OAuth OpenID Connect、Kerberos などがあります。
- 指定されたクレデンシャルを評価して、認証が成功したか、失敗したか、またはグループ抽出、属性抽出などのアクションを実行するかを決定します。
- 評価結果に基づいて、アクセスが許可されるか、拒否されるか、次の要素が選択されます。
- 評価する次の要因がなくなるまで、これらの手順を繰り返します。

nFactor 認証を使用すると、次のことができます。

- 任意の数の認証要素を設定します。
- 前のファクタの実行結果に基づいて、次のファクタの選択を行います。
- ログインインターフェイスをカスタマイズします。たとえば、ラベル名、エラーメッセージ、およびヘルプテキストをカスタマイズできます。
- 認証を行わずにユーザーグループ情報を抽出します。
- 認証ファクタのパススルーを設定します。つまり、その要素には明示的なログイン操作は必要ありません。
- 異なるタイプの認証が適用される順序を設定します。Citrix ADC アプライアンスでサポートされている認証メカニズムは、nFactor 認証設定の任意の要素として構成できます。これらの要因は、設定されている順序で実行されます。
- Citrix ADC を構成して、認証が失敗したときに実行する必要のある認証係数に進みます。そのためには、まったく同じ条件で、次に高い優先順位で、アクションを「NO_AUTH」に設定して、別の認証ポリシーを設定します。次の要素を設定する必要があります。次の要素では、適用する代替認証メカニズムを指定する必要があります。

nFactor 認証用の **Citrix Gateway** ログイン情報の暗号化

nFactor 認証を使用した Citrix Gateway では、認証プロセス中にクライアント（ブラウザまたは SSO アプリ）から送信されたログイン要求フィールドを暗号化できます。暗号化されたログイン要求フィールドは、ユーザーの機密データが開示されないように保護するための追加のセキュリティレイヤーを提供します。

互換性のあるブラウザー

次の表に、ログイン暗号化をサポートするバージョンの詳細とブラウザの一覧を示します。

Web ブラウザー	バージョン
Chrome	78 以降
Firefox	69 以降
Internet Explorer	11
Edge	42 以降
Safari	11.0 以降
オペラ	66

互換性のあるクライアント

次のセクションでは、Citrix Gateway ログイン情報の暗号化をサポートするバージョンの詳細とともに、クライアントの一覧を示します。

- Mac の Citrix Workspace アプリは、OS バージョンが 10.14.x 以降の場合にのみ暗号化をサポートします。
- Mac の Citrix SSO アプリは、OS バージョンが 10.14.x 以降の場合にのみ暗号化をサポートします。
- Windows SSO アプリには、互換性に関する制限はありません。
- Windows クライアント用 Citrix Workspace アプリでのパスワード暗号化は、Internet Explorer 11 のバージョンでのみサポートされています。

CLI を使用してログイン暗号化を有効にするには

コマンドプロンプトで入力します。

```
set aaa parameter [-loginEncryption (ENABLED | DISABLED)]
```

> 注
>
> LoginEncryption パラメータは、デフォルトで無効になっています。有効にする必要があります。

GUI を使用してログイン暗号化を有効にするには

1. [セキュリティ**] > [AAA —** アプリケーショントラフィック] に移動し、[認証設定**]** セクションで **[**** 認証 **AAA** 設定の変更 **] をクリックします。
2. **[AAA** パラメータの設定**]** ページで、[ログイン暗号化] オプションまでスクロールダウンし、有効にします。

nFactor の概念、エンティティ、および用語

March 8, 2022

このトピックでは、nFactor 認証に関係する主要なエンティティのいくつかとその重要性について説明します。

ログインスキーマ

nFactor は、ユーザーインターフェイスである 'view' を、ランタイム処理である 'model' と切り離します。nFactor のビューはログインスキーマによって定義されます。ログインスキーマは、ユーザーに表示される内容を定義し、ユーザーからデータを抽出する方法を指定するエンティティです。

ビューを定義するために、ログインスキーマはログオンフォームを定義するディスク上のファイルを指します。このファイルは、「Citrix 共通形式プロトコル」の仕様に準拠している必要があります。このファイルは基本的に、ログオンフォームの XML 定義です。

ログインスキーマには、XML ファイルに加えて、ユーザーのログイン要求からユーザー名とパスワードを収集するための高度なポリシー式が含まれています。これらの式はオプションであり、user からのユーザー名とパスワードが予想されるフォーム変数名で届いた場合は省略できます。

ログインスキーマは、現在の資格情報セットをデフォルトの SingleSignon 資格情報として使用する必要があるかどうかも定義します。

ログインスキーマを作成するには、次の CLI コマンドを実行します。

```
1    add authentication loginSchema <name> -authenticationSchema <string>
         [-userExpression <string>] [-passwdExpression <string>] [-
        userCredentialIndex <positive_integer>] [-passwordCredentialIndex
         <positive_integer>] [-authenticationStrength <positive_integer>]
         [-SSOCredentials ( YES | NO )]
2 <!--NeedCopy-->
```

> 注:
>
> **ssoCredentials** は、現在の要素認証情報がデフォルトの SSO 認証情報であるかどうかを示します。デフォルト値は NO です。
>
> nFactor 認証設定では、デフォルトで SSO に最終要素の認証情報が使用されます。**ssoCredentials** 構成を使用すると、現在の要素の認証情報を使用できます。この構成が異なるファクタで設定される場合、この構成が設定された最後のファクタが優先されます。

各パラメータの詳細については、「https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/13/authentication/authentication-loginSchema/#add-authentication-loginS」を参照してください。

ポリシーラベル

ポリシーラベルはポリシーの集まりです。これは、Citrix ADC のポリシーインフラストラクチャとは異質ではない構成要素です。ポリシーラベルは認証要素を定義します。つまり、ユーザーからの認証情報が満たされているかどうかを判断するのに必要なすべてのポリシーが含まれています。ポリシーラベル内のポリシーはすべて同種であると見なすことができます。認証用のポリシーラベルは、書き換えなど、異なるタイプのポリシーを使用できません。言い換えると、ポリシーラベル内のすべてのポリシーは、ほとんどの場合、ユーザーからの同じパスワード/クレデンシャルを検証します。PolicyLabel のポリシーの結果は、論理 OR 条件に従います。したがって、最初のポリシーで指定された認証が成功すると、そのポリシーに続く他のポリシーはスキップされます。

ポリシーラベルは、次の CLI コマンドを実行して作成できます。

```
1 add authentication policy label mylabel  - loginSchema <>
2 <!--NeedCopy-->
```

ポリシーラベルは、ログインスキーマをプロパティとして使用します。ログインスキーマは、そのポリシーラベルのビューを定義します。ログインスキーマが指定されていない場合、暗黙的なログインスキーマ LSCHEMA_INT がそのポリシーラベルに関連付けられます。ログインスキーマは、ポリシーラベルがパススルーになるかどうかを決定します。

仮想サーバラベル

Citrix ADC の高度なポリシーインフラストラクチャでは、仮想サーバーも暗黙的なポリシーラベルです。これは、仮想サーバを複数のポリシーにバインドできるためです。ただし、仮想サーバはクライアントトラフィックのエントリポイントであり、異なるタイプのポリシーを取ることができるため、仮想サーバは特殊です。各ポリシーは、仮想サーバ内で独自のラベルの下に配置されます。したがって、仮想サーバはラベルの集合体です。

次の因子

ポリシーが仮想サーバまたはポリシーラベルにバインドされる場合は常に、次の要素で指定できます。次の要素は、特定の認証が成功した場合に何をする必要があるかを決定します。次の要素がない場合、そのユーザーの認証プロセスは終了です。

仮想サーバまたはポリシーラベルにバインドされたポリシーごとに、次の要素が異なる場合があります。これにより、すべてのポリシーが成功したときにユーザー認証の新しいパスを定義できる、究極の柔軟性が得られます。管理者はこの事実を利用して、特定のポリシーを満たしていないユーザーに対して巧妙なフォールバックファクターを作成できます。

認証なしポリシー

nFactor は NO_AUTHN と呼ばれる特別なビルトインポリシーを導入しています。NO_AUTHN ポリシーは常に認証結果として成功を返す。No-auth ポリシーを作成するには、次の CLI コマンドを実行します。

```
add authentication policy noauthpolicy  - rule <> -action NO_AUTHN
<!--NeedCopy-->
```

コマンドに従って、no-authentication ポリシーには任意の高度なポリシー式を指定できるルールが適用されます。認証結果は常に NO_AUTHN から成功する。

no-authポリシー自体は価値を付加するものではないようです。ただし、パススルーポリシーラベルとともに使用すると、ユーザ認証フローを促進する論理的な決定を柔軟に行うことができます。NO_AUTHN ポリシーとパススルーファクターは、nFactor の柔軟性に新たな次元を提供します。

注**:** 以降のセクションで、no-auth およびパススルーの使用方法を示す例を確認してください。

パススルー・ファクタ/ラベル

ユーザが仮想サーバで（第 1 要素の）認証に合格すると、以降の認証はポリシーラベルまたはユーザ定義（2 次）要素で行われます。
すべてのポリシーラベル/ファクタはログインスキーマエンティティに関連付けられ、そのファクタのビューが表示されます。これにより、ユーザーが特定の要素に到達するまでの経路に基づいてビューをカスタマイズできます。

ログインスキーマを明示的に指していない特殊な種類のポリシーラベルがあります。特殊なポリシーラベルは、実際にはビューの XML ファイルを指していないログインスキーマを指しています。これらのポリシーラベル/ファクターは「パススルー」ファクターと呼ばれます。

パススルー係数は、次の CLI コマンドを実行して作成できます。

例 1:

```
add authentication policylabel example1
<!--NeedCopy-->
```

例 2:

```
add loginschema passthrough_schema  - authenticationSchema noschema

add authentication policylabel example2  - loginschema
    passthrough_schema
<!--NeedCopy-->
```


パススルーファクターは、認証、承認、および監査サブシステムがユーザーに戻ってそのファクターのクレデンシャルセットを取得してはならないことを意味します。その代わり、認証、承認、および監査は、すでに取得した認証情報を使用して続行するためのヒントになります。これは、ユーザーの介入が望ましくない場合に便利です。例:

- ユーザに 2 つのパスワードフィールドが表示された場合、1 つ目のファクタの後に 2 番目のファクタにユーザの介入は必要ありません。
- あるタイプ (証明書など) の認証が行われ、管理者がそのユーザーのグループを抽出する必要がある場合。

パススルー係数を`NO_AUTH`ポリシーとともに使用して、条件付きジャンプを行うことができます。

nFactor 認証フロー

認証は常に nFactor の仮想サーバで開始されます。仮想サーバーは、ユーザーの第 1 の要素を定義します。ユーザーに最初に表示されるフォームは、仮想サーバーによって提供されます。ユーザーに表示されるログオンフォームは、ログインスキーマポリシーを使用して仮想サーバーでカスタマイズできます。ログインスキーマポリシーがない場合、ユーザー名とパスワードのフィールドが 1 つだけユーザーに表示されます。

カスタマイズされたフォームで複数のパスワードフィールドをユーザーに表示する必要がある場合は、ログインスキーマポリシーを使用する必要があります。構成された規則 (イントラネットユーザーと外部ユーザー、サービスプロバイダー A とサービスプロバイダー B など) に基づいて異なるフォームを表示できます。

ユーザー資格情報が投稿されると、最初の要素である認証仮想サーバーで認証が開始されます。認証仮想サーバには複数のポリシーを設定できるため、各ポリシーが順番に評価されます。任意の時点で、認証ポリシーが成功すると、そのポリシーに対して指定された次の要素が使用されます。次の要素がない場合、認証プロセスは終了します。次の因子が存在する場合、その因子が通過因子か標準因子かがチェックされます。パススルーの場合、その要素に関する認証ポリシーはユーザーの介入なしに評価されます。それ以外の場合は、そのファクタに関連付けられたログインスキーマがユーザに表示されます。

パススルーファクターと非認証ポリシーを使用して論理的な決定を下す例

管理者はグループに基づいて NextFactor を決定したいと考えています。

```
1 add authentication policylabel group check
2
3 add authentication policy admin group  -rule http.req.user.is_member_of
    ("Administrators")  -action NO_AUTHN
4
5 add authentication policy nonadmins  -rule true  -action NO_AUTHN
6
7 bind authentication policy label group check  -policy admingroup  -pri
    1  -nextFactor factor-for-admin
8
```



```
bind authentication policy label groupcheck -policy nonadmins -pri 10
     -nextfactor factor-for-others

add authentication policy first_factor_policy -rule <> -action <>

bind authentication vserver <> -policy first_factor_policy -priority
    10 -nextFactor groupcheck
<!--NeedCopy-->
```

nFactor 認証の設定

September 27, 2022

nFactor 設定を使用して、複数の認証要素を設定できます。nFactor 構成は、Citrix ADC アドバンスドエディションおよびプレミアムエディションでのみサポートされます。

nFactor を設定するメソッド

nFactor 認証は、次のいずれかの方法で設定できます。

- **nFactor** ビジュアライザー**:** nFactor ビジュアライザーを使用すると、単一のペインでファクターまたはポリシーラベルを簡単にリンクでき、同じペインでファクターのリンクを変更することもできます。ビジュアライザーを使用して nFactor フローを作成し、そのフローを認証、承認、および監査仮想サーバーにバインドできます。nFactor Visualizer の詳細と、ビジュアライザーを使用した nFactor の設定例については、nFactor ビジュアライザーの簡単な設定を参照してください。
- **Citrix ADC GUI**: 詳細については、**nFactor** 構成に関連する構成要素のセクションを参照してください。
- **Citrix ADC CLI**: Citrix ADC CLI を使用した nFactor 構成のサンプルスニペットについては、Citrix ADC CLI を使用した nFactor 構成のサンプルスニペットを参照してください。

> 重要: このトピックでは、Citrix ADC GUI を使用して nFactor を構成する方法の詳細について説明します。

nFactor の設定に関係する設定要素

nFactor の設定には、次の要素が含まれます。詳細な手順については、このトピックの該当するセクションを参照してください。

構成要素	実行するタスク
AAA 仮想サーバ	AAA 仮想サーバを作成する
	ポータルテーマを AAA 仮想サーバにバインドする

構成要素	実行するタスク
	クライアント証明書認証を有効にする
ログインスキーマ	ログインスキーマプロファイルの設定
	ログインスキーマポリシーの作成とバインド
高度な認証ポリシー	高度な認証ポリシーの作成
	第 1 要素の詳細認証ポリシーを Citrix ADC AAA 仮想サーバーにバインドする
	抽出された LDAP グループを使用して、次の認証要素を選択する
認証ポリシーラベル	認証ポリシーラベルの作成
	認証ポリシーラベルのバインド
Citrix Gateway の **nFactor**	Citrix ADC AAA 仮想サーバーと Citrix Gateway 仮想サーバーをリンクするための認証プロファイルを作成する
	Citrix Gateway の SSL パラメーターと CA 証明書を構成する
	StoreFront への nFactor シングルサインオン用の Citrix Gateway トラフィックポリシーを構成する

nFactor のしくみ

ユーザーが認証、承認、監査、または Citrix Gateway 仮想サーバーに接続すると、発生する一連のイベントは次のとおりです。

1. フォームベース認証を使用すると、認証、承認、および監査仮想サーバーにバインドされたログインスキーマが表示されます。
2. 認証、承認、および監査仮想サーバーにバインドされた高度な認証ポリシーが評価されます。
    - 高度な認証ポリシーが成功し、次の要素 (認証ポリシーラベル) が構成されると、次の要素が評価されます。Next Factor が設定されていない場合、認証は完了し、成功します。
    - 高度な認証ポリシーが失敗し、[式に移動] が [次へ] に設定されている場合、次にバインドされた高度な認証ポリシーが評価されます。いずれの高度な認証ポリシーも成功しない場合、認証は失敗します。
3. ネクストファクタ認証ポリシーラベルにログインスキーマがバインドされている場合は、ユーザーに表示されます。
4. 次の要素認証ポリシーラベルにバインドされた高度な認証ポリシーが評価されます。

- 詳細認証ポリシーが成功し、次の要素 (認証ポリシーラベル) が構成されると、次の要素が評価されます。
- Next Factor が設定されていない場合、認証は完了し、成功します。

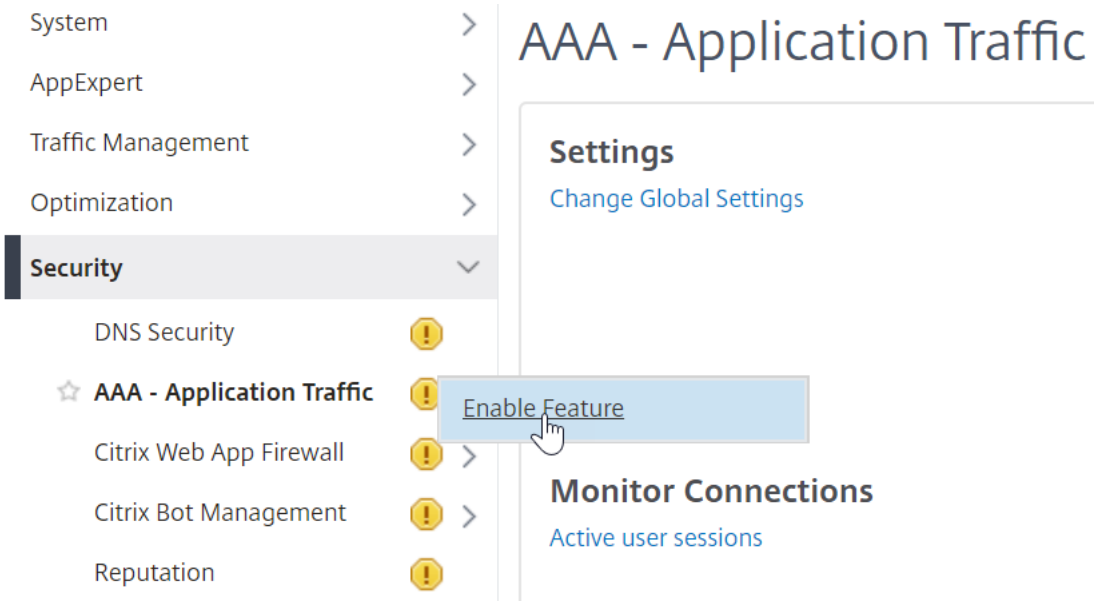
5. 詳細認証ポリシーが失敗し、[式に移行] が [次へ] の場合、次にバインドされた高度な認証ポリシーが評価されます。
6. ポリシーが成功すると、認証は失敗します。

仮想サーバの認証、承認、監査

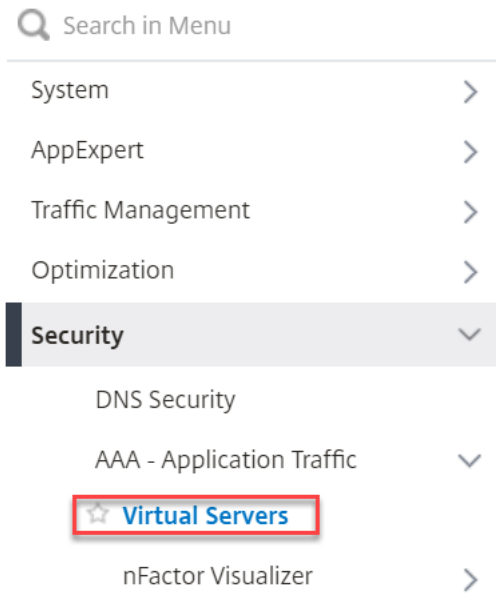
Citrix Gateway で nFactor を使用するには、まず認証、承認、および監査仮想サーバーで nFactor を構成します。その後、認証、承認、および監査仮想サーバーを Citrix Gateway 仮想サーバーにリンクします。

認証、承認、監査用仮想サーバーの作成

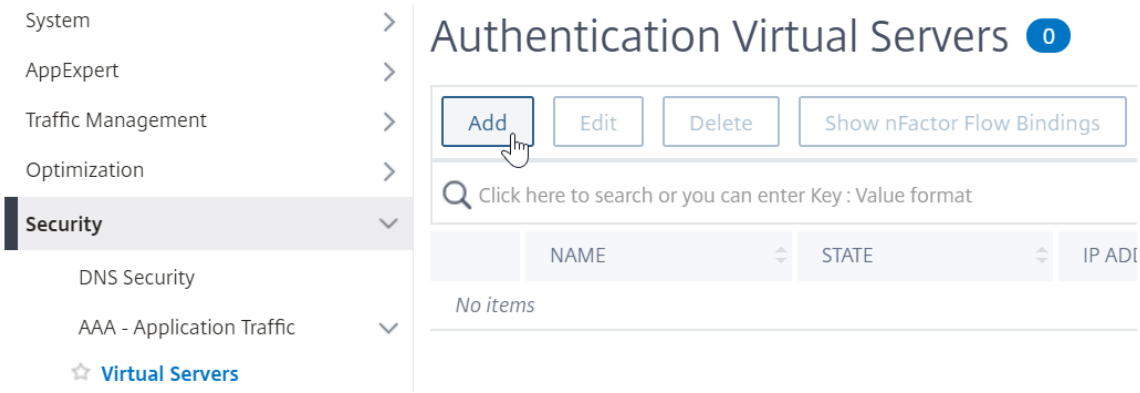
1. 認証、認可、および監査機能がまだ有効になっていない場合は、[セキュリティ] **>** **[AAA —** アプリケーショントラフィック] に移動し、右クリックしてこの機能を有効にします。



2. 設定 **>** セキュリティ **> AAA-**アプリケーショントラフィック **>** 仮想サーバにナビゲートして下さい。

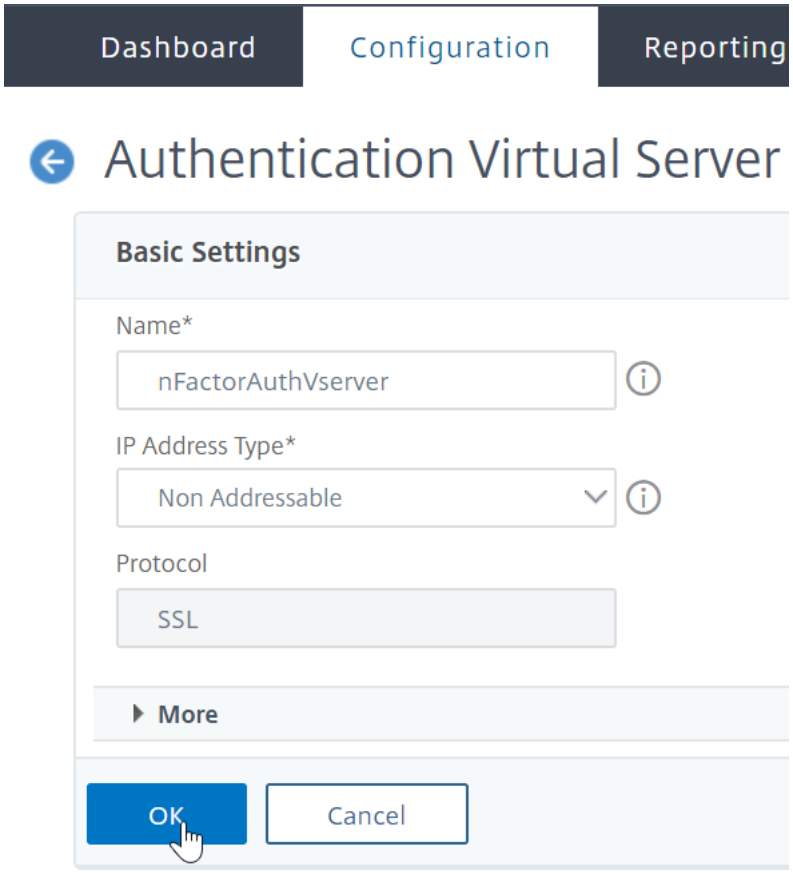


3. [追加] をクリックして、認証仮想サーバーを作成します。

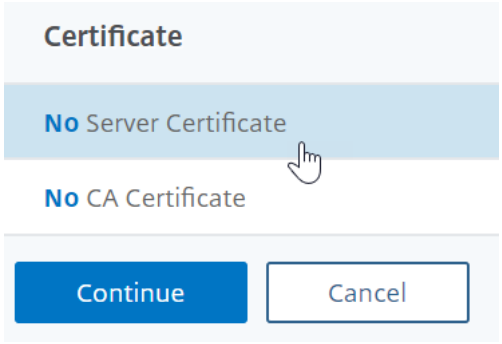


4. 次の情報を入力し、「**OK**」をクリックします。

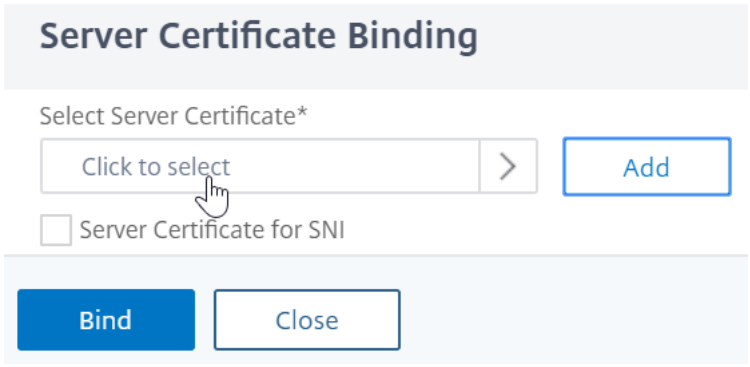
パラメーター名	パラメータの説明
名前	認証、承認、および監査仮想サーバーの名前。
IP アドレスタイプ	この仮想サーバーを Citrix Gateway でのみ使用する場合は、[IP アドレスの種類] を [アドレス指定不可] に変更します。



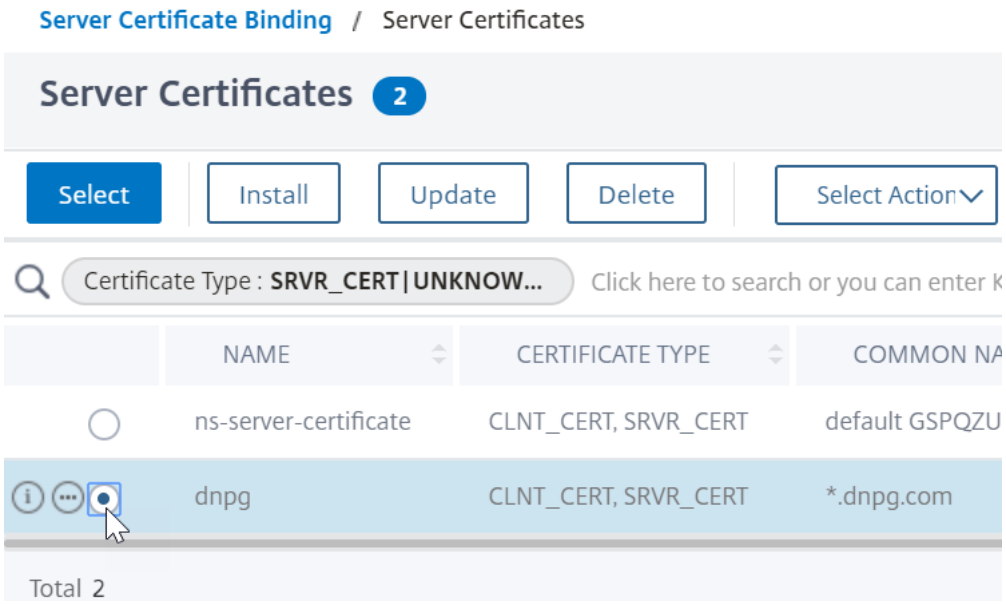
5. [証明書] で、[サーバー証明書なし] を選択します。



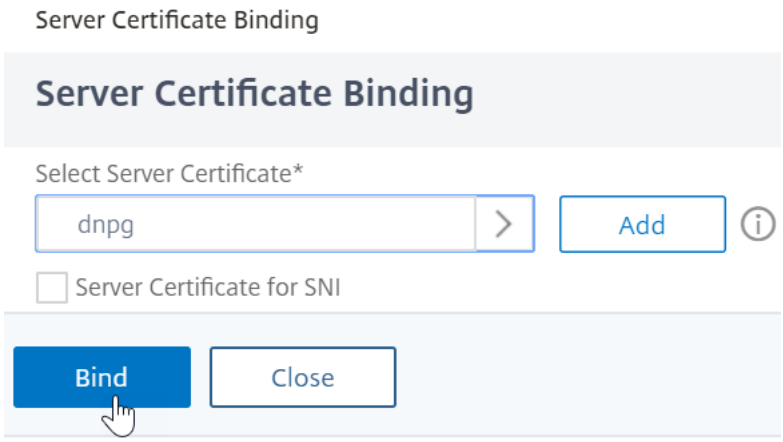
6. [クリックして選択] というテキストをクリックし、サーバー証明書を選択します。



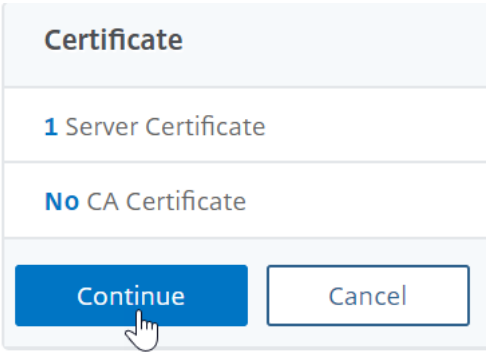
7. 認証、承認、および監査用の仮想サーバーの証明書の横にあるラジオボタンをクリックし、[選択] をクリックします。このサーバーには直接アクセスできないため、選択した証明書は関係ありません。



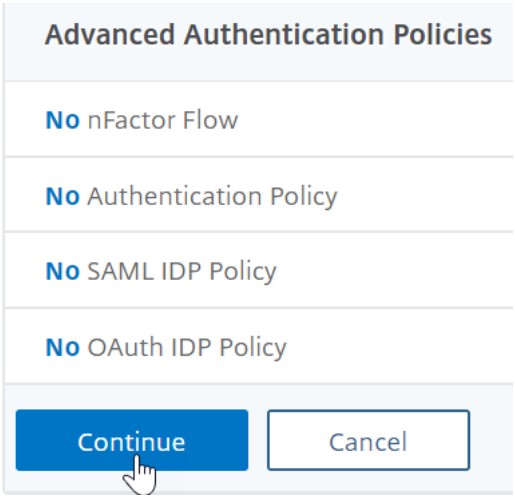
8. ［バインド］をクリックします。



9. [続行] をクリックして、[証明書] セクションを閉じます。

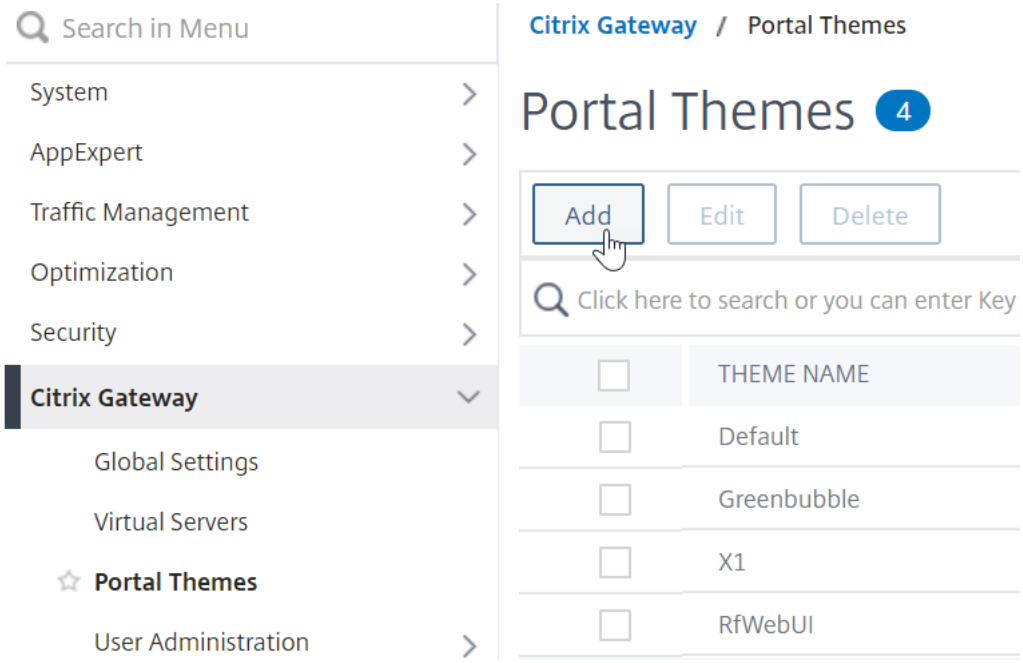


10. ［続行］をクリックします。

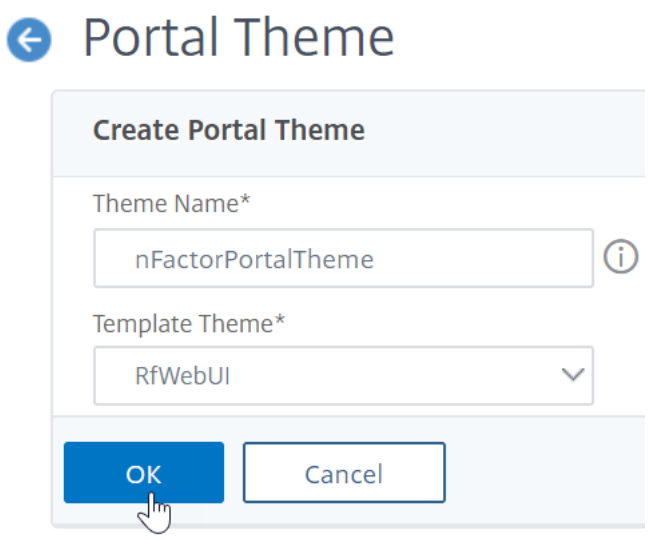


ポータル・テーマを認証、承認、監査用仮想サーバーにバインドする

1. [**Citrix Gateway**] > [ポータルのテーマ] に移動し、テーマを追加します。Citrix Gateway でテーマを作成し、後で認証、承認、および監査仮想サーバーにバインドします。

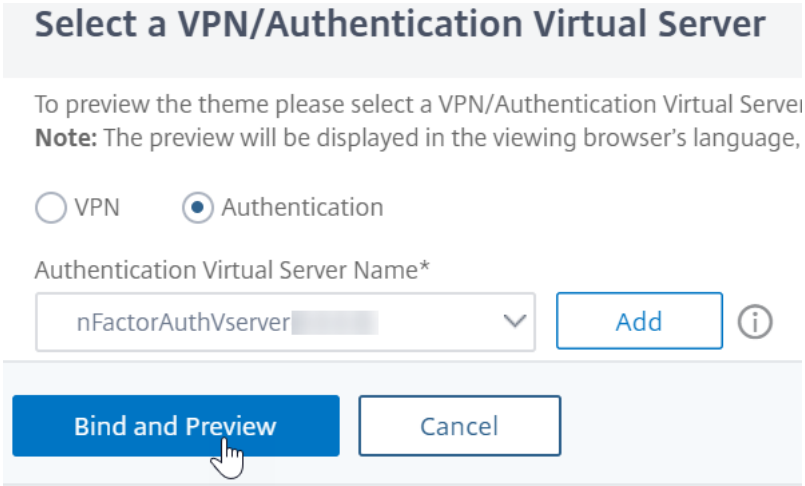


2. rfWebUI テンプレートテーマに基づいてテーマを作成します。



3. 必要に応じてテーマを調整した後、ポータル・テーマ編集ページの上部で、[クリックしてバインドして構成済みテーマを表示] をクリックします。

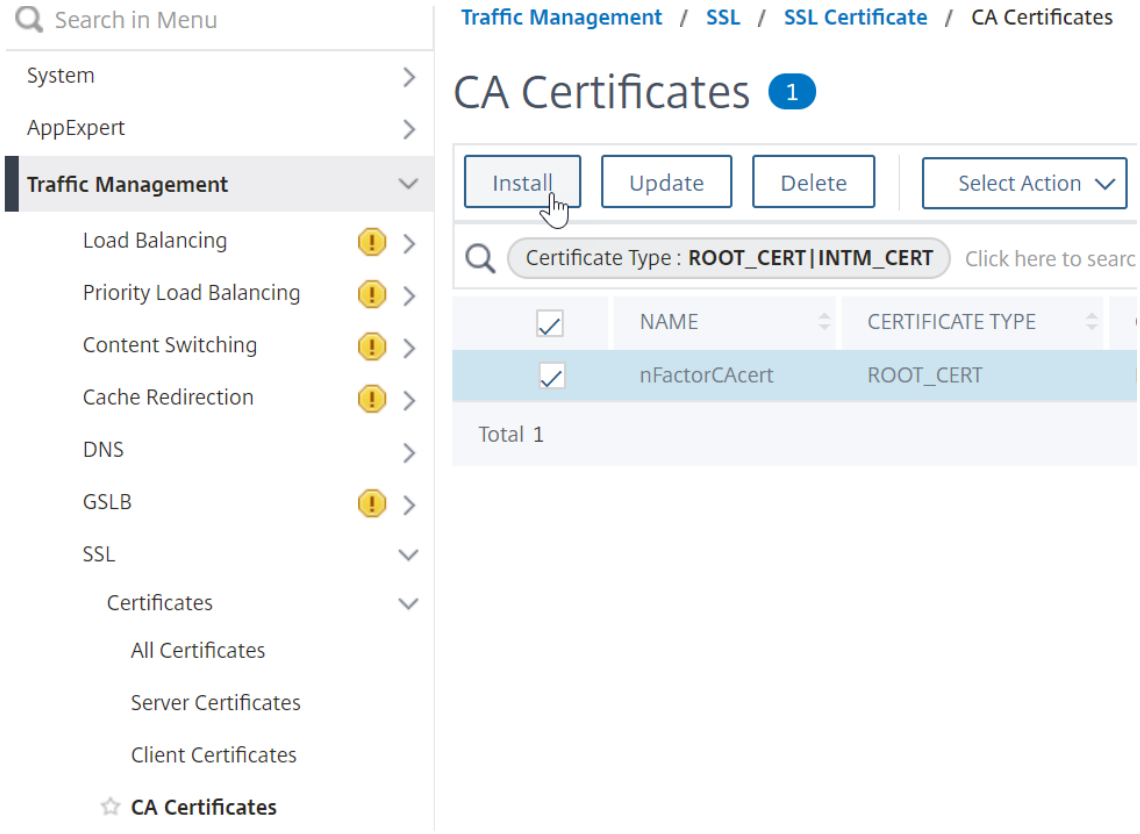
4. 選択を [認証] に変更します。[認証仮想サーバ名] ドロップダウンメニューから、認証、承認、および監査仮想サーバを選択し、[バインドとプレビュー] をクリックしてプレビューウィンドウを閉じます。

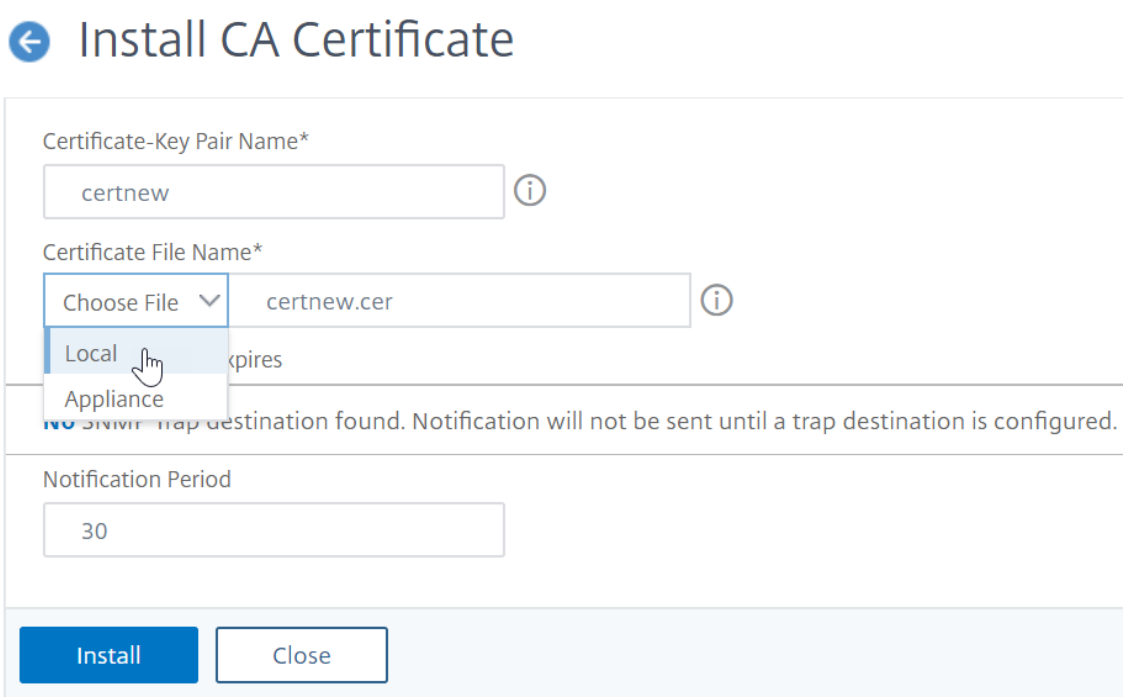


クライアント証明書認証を有効にする

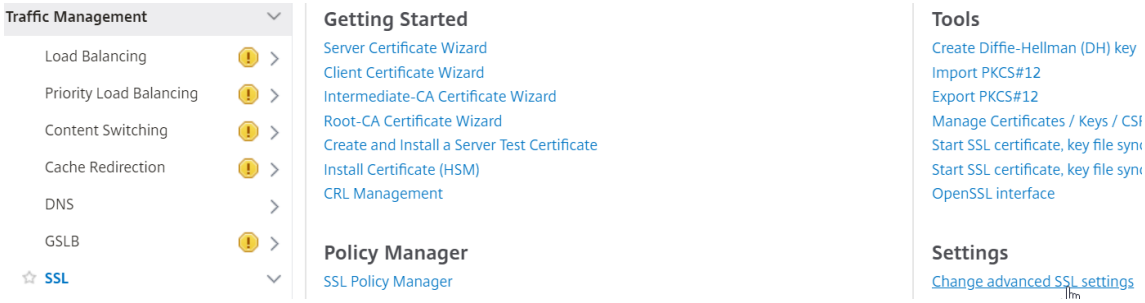
認証要素の 1 つがクライアント証明書である場合は、認証、承認、および監査仮想サーバーで SSL 構成を実行する必要があります。

1. [トラフィック管理] **> [SSL] > [**証明書] **> [CA** 証明書] に移動し、クライアント証明書の発行者のルート証明書をインストールします。ルート証明書にはキーファイルがありません。





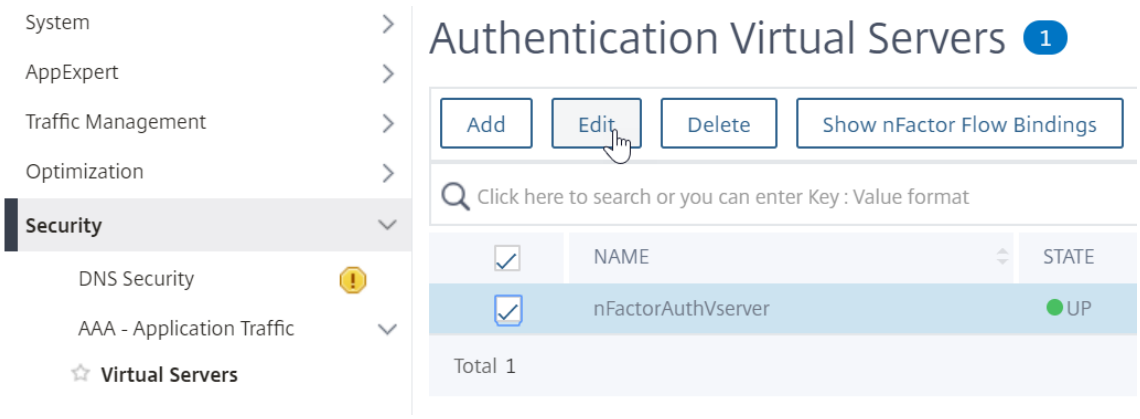
2. [トラフィック管理] **> [SSL] > [SSL** の詳細設定の変更] に移動します。



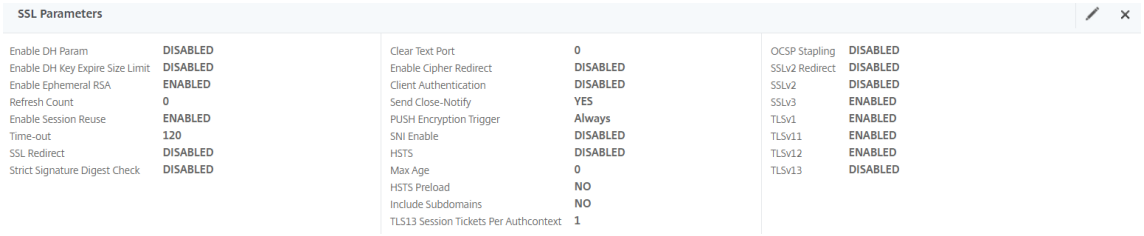
   a) 下にスクロールして、[デフォルトプロファイル] が [有効] になっているかどうかを確認します。「はい」の場合は、SSL プロファイルを使用してクライアント証明書認証を有効にする必要があります。それ以外の場合は、[SSL パラメーター] セクションの認証、承認、および監査仮想サーバーでクライアント証明書認証を直接有効にできます。

3. デフォルトの SSL プロファイルが有効になっていない場合は、次の手順を実行します。

   a) [セキュリティ] **> [AAA-**アプリケーション] **> [**仮想サーバ] に移動し、既存の認証、認可、および監査仮想サーバを編集します。

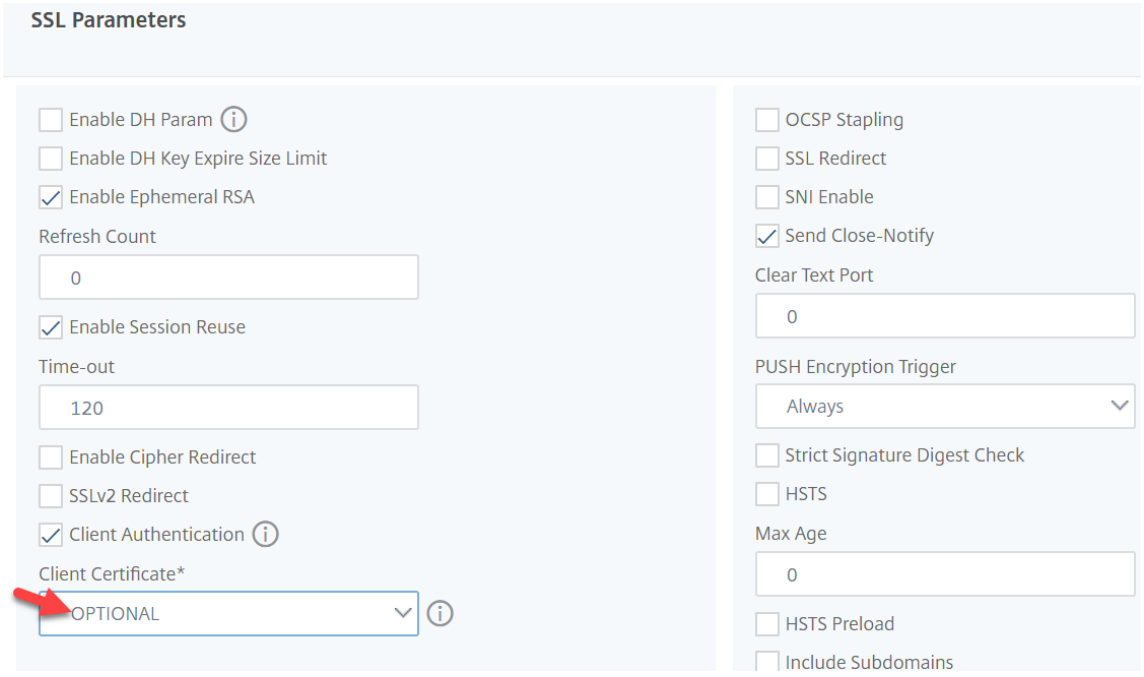


a) 左側の [**SSL** パラメータ] セクションで、鉛筆アイコンをクリックします。



a) [クライアント認証] の横にあるチェックボックスをオンにします。

b) [クライアント証明書] ドロップダウンメニューで [オプション] が選択されていることを確認し、[**OK**] をクリックします。



4. デフォルト SSL プロファイルが有効な場合は、クライアント認証を有効にして SSL プロファイルを作成します。

a) 左側のメニューで、[システム] を展開し、[プロファイル] をクリックします。

b) 右上の [SSL プロファイル] タブに切り替えます。

c) ns_default_ssl_profile_frontend プロファイルを右クリックし、[追加] をクリックします。これにより、デフォルトプロファイルから設定がコピーされます。

d) プロファイルに名前を付けます。このプロファイルの目的は、クライアント証明書を有効にすることです。

e) 下にスクロールして、[クライアント認証] チェックボックスを見つけます。ボックスにチェックを入れます。

f) [クライアント証明書] ドロップダウンメニューを [オプション] に変更します。

g) デフォルトの SSL プロファイルをコピーしても、SSL 暗号はコピーされません。やり直す必要があります。

h) SSL プロファイルの作成が完了したら、[完了] をクリックします。

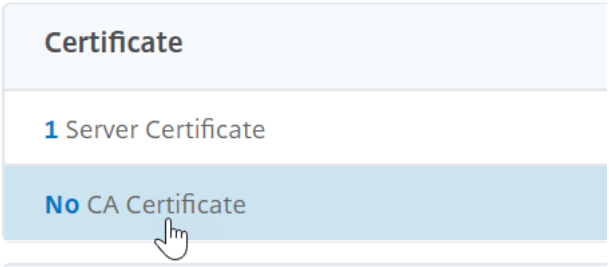
i) [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動し、認証、認可、および監査仮想サーバを編集します。

j) [SSL プロファイル] セクションまでスクロールし、鉛筆をクリックします。

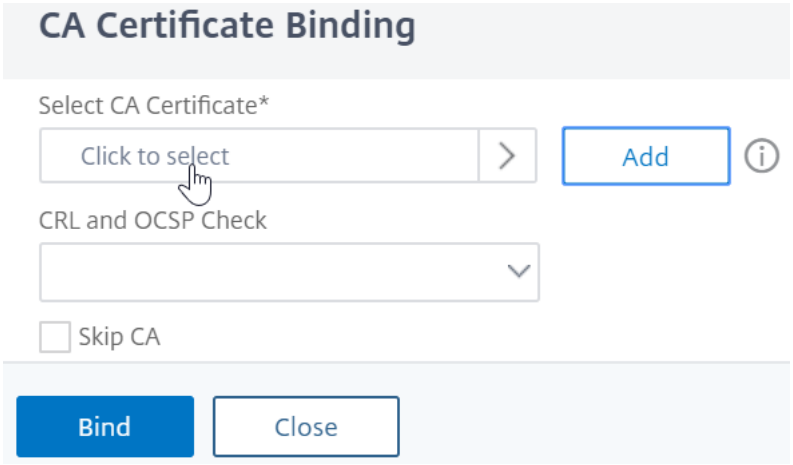
k) [SSL プロファイル] ドロップダウンメニューを、クライアント証明書が有効になっているプロファイルに変更します。[OK] をクリックします。

l) CA 証明書をバインドする手順が表示されるまで、この記事を下にスクロールします。

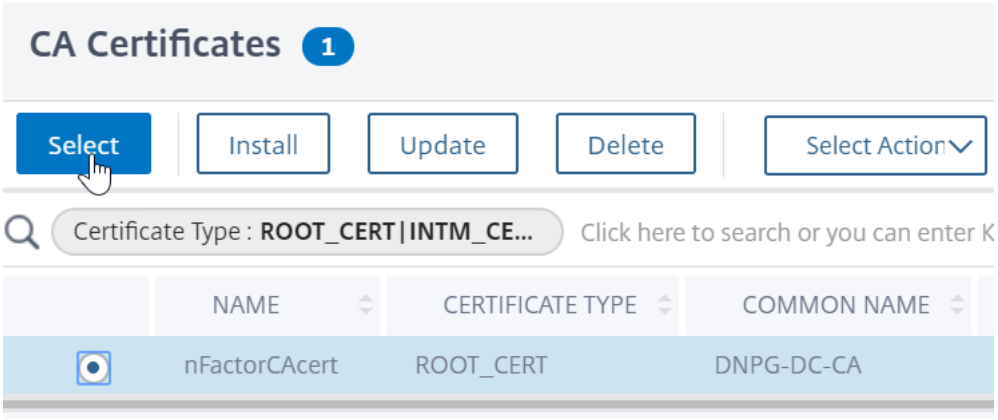
5. 左側の [証明書] セクションで、[**CA** 証明書なし] と表示されている場所をクリックします。



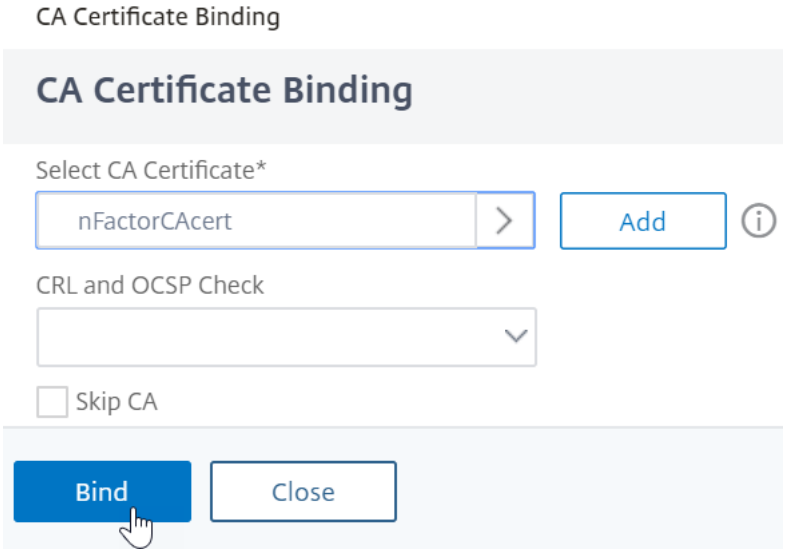
6. テキストをクリックし、クリックして選択します。



7. クライアント証明書の発行者のルート証明書の隣にあるオプションボタンをクリックし、[選択（**Select**）] をクリックします。



8. ［バインド］をクリックします。



ログインスキーマ **XML** ファイル

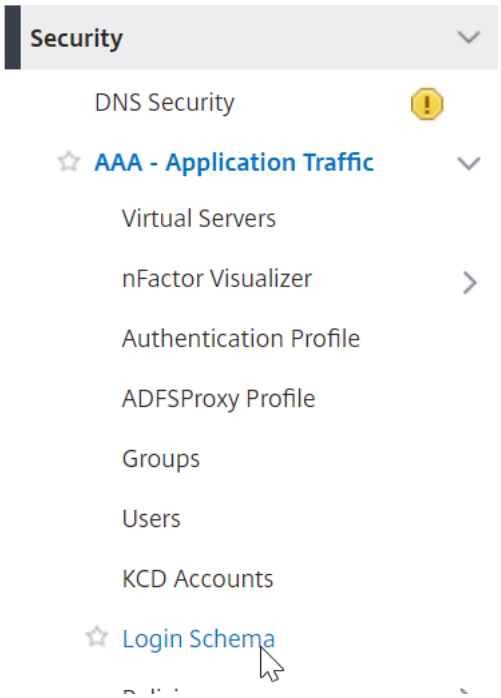
ログインスキーマは、フォームベース認証ログオンページの構造を提供する XML ファイルです。

nFactor は、複数の認証要素が連鎖していることを意味します。各ファクタは、異なるログインスキーマページ/ファイルを持つことができます。一部の認証シナリオでは、ユーザーに複数のログオン画面を表示することができます。

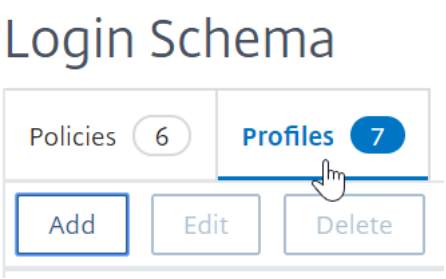
ログインスキーマプロファイルの設定

ログインスキーマプロファイルを設定するには、次の手順を実行します。

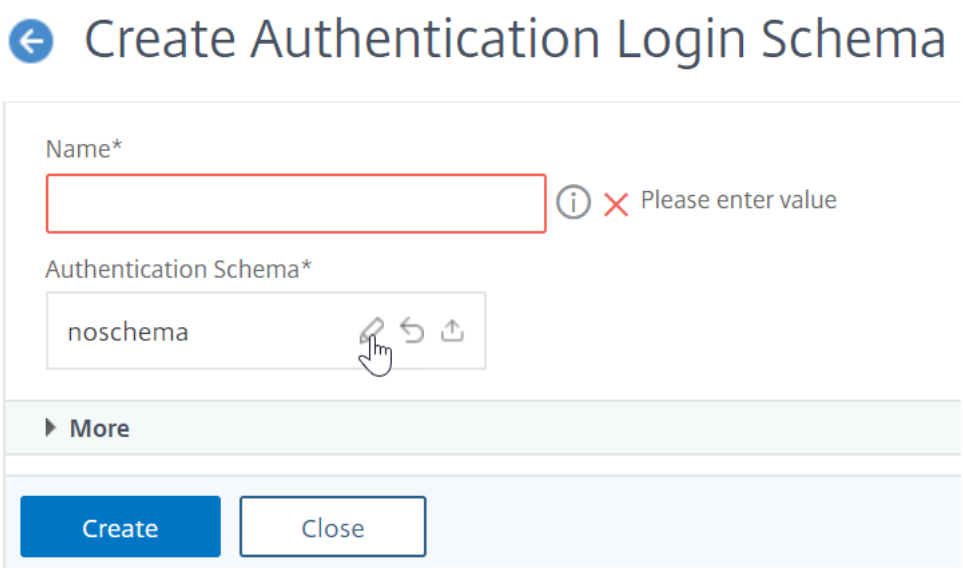
1. nFactor の設計に基づいて、ログインスキーマの.XML ファイルを作成または編集します。
2. ［セキュリティ］ **>** ［**AAA-**アプリケーショントラフィック］ **>** ［ログインスキーマ］に移動します。



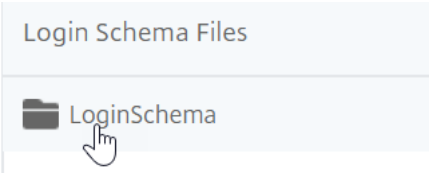
3. 右側の [プロファイル] タブに切り替えて、[追加] をクリックします。



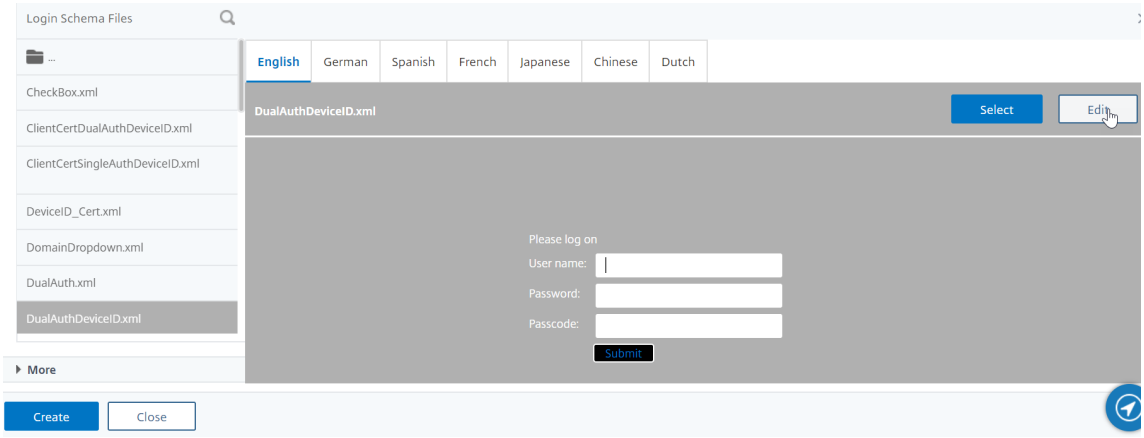
4. [認証スキーマ] フィールドで、鉛筆アイコンをクリックします。



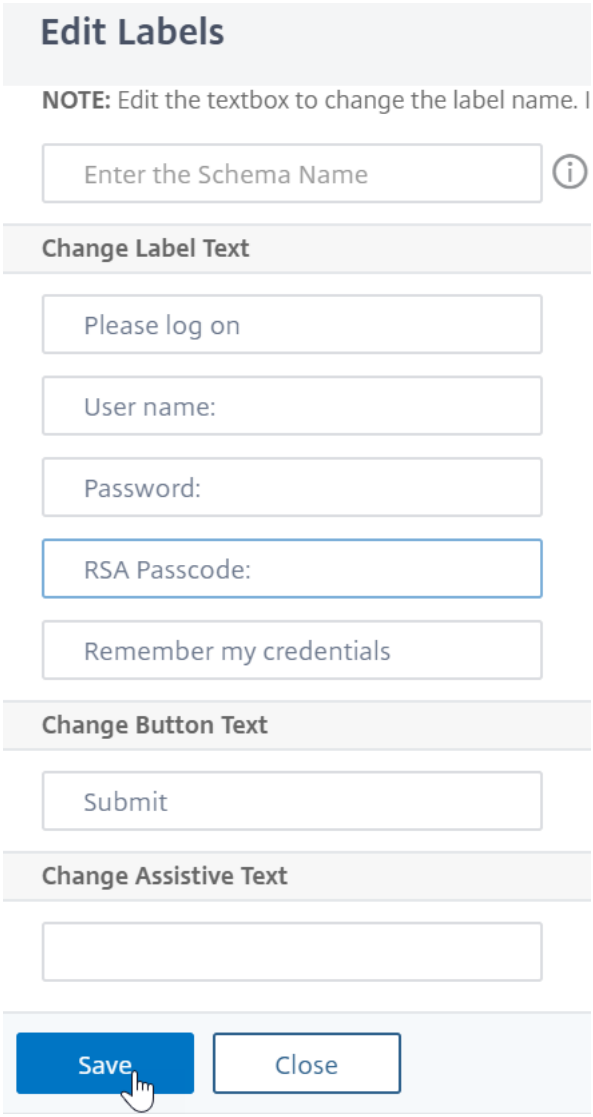
5. LoginSchema フォルダをクリックして、フォルダ内のファイルを表示します。



6. ファイルの 1 つを選択します。右側にプレビューが表示されます。ラベルは、右上の [編集] ボタンをクリックして変更できます。

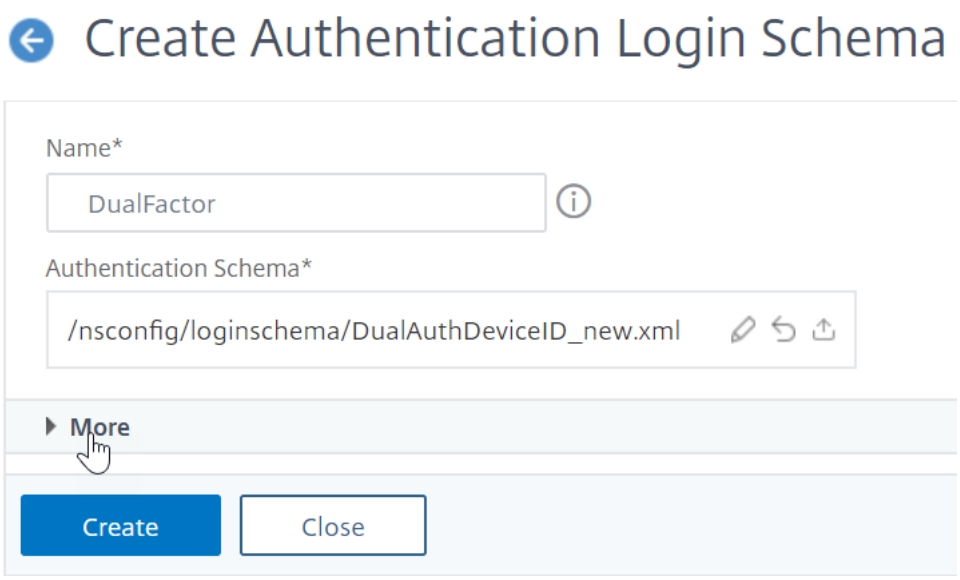


7. 変更を保存すると、/nsConfig/loginSchema の下に新しいファイルが作成されます。



8. 右上の [選択] をクリックします。

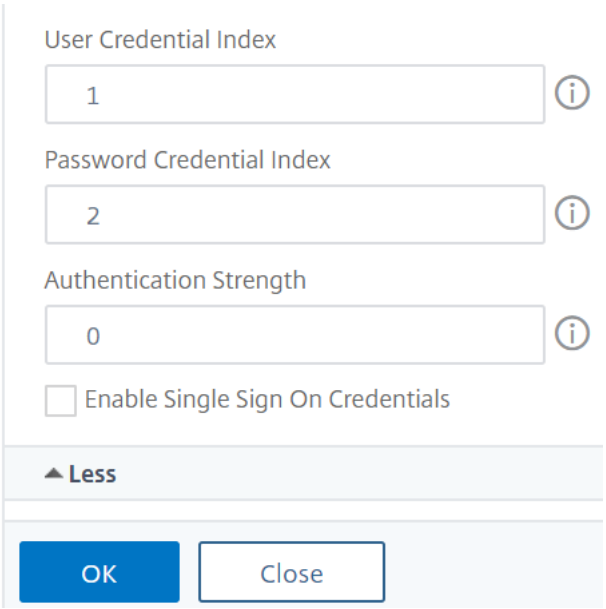
9. ログインスキーマに名前を付けて、[**More**] をクリックします。



10. StoreFront などのバックエンドサービスへのシングルサインオン（SSO）のログインスキーマに入力したユーザー名とパスワードを使用します。

    ログインスキーマに入力された資格情報をシングルサインオンの認証情報として使用するには、次のいずれかの方法を使用します。

    - [認証ログインスキーマの作成] ページの下部にある [詳細] をクリックし、[シングルサインオン資格情報の有効化] を選択します。
    - [認証ログインスキーマの作成] ページの下部にある [詳細] をクリックし、ユーザクレデンシャルインデックスとパスワードクレデンシャルインデックスに一意の値を入力します。これらの値は 1 ～16 の範囲です。後で AAA.USER.ATTRIBUTE (#) 式を使用して、トラフィックポリシー/プロファイルでこれらのインデックス値を参照します。



11. **OK** をクリックして、ログインスキーマプロファイルを作成します。

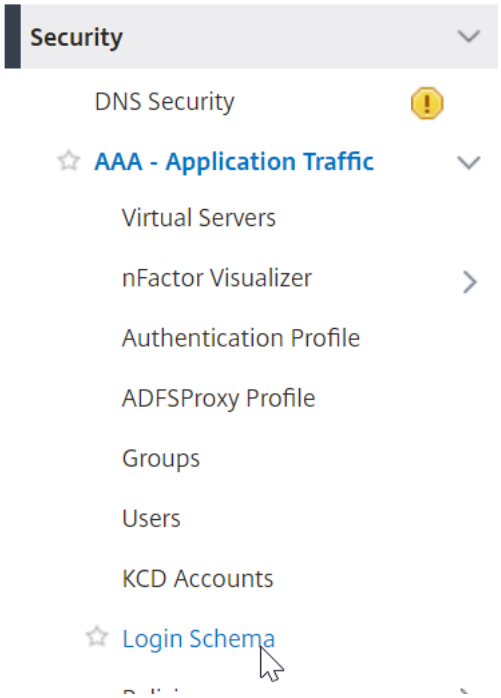
> 注**:** ログインスキーマファイル (.xml) を後で編集する場合、変更を反映させるには、ログインスキーマプロファイルを編集して、ログインスキーマ (.xml) ファイルを再度選択する必要があります。

ログインスキーマポリシーの作成とバインド

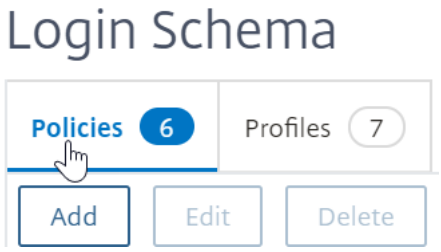
ログインスキーマプロファイルを認証、承認、および監査仮想サーバーにバインドするには、最初にログインスキーマポリシーを作成する必要があります。後で詳述するように、ログインスキーマプロファイルを認証ポリシーラベルにバインドする場合は、ログインスキーマポリシーは必要ありません。

ログインスキーマポリシーを作成してバインドするには、次の手順を実行します。

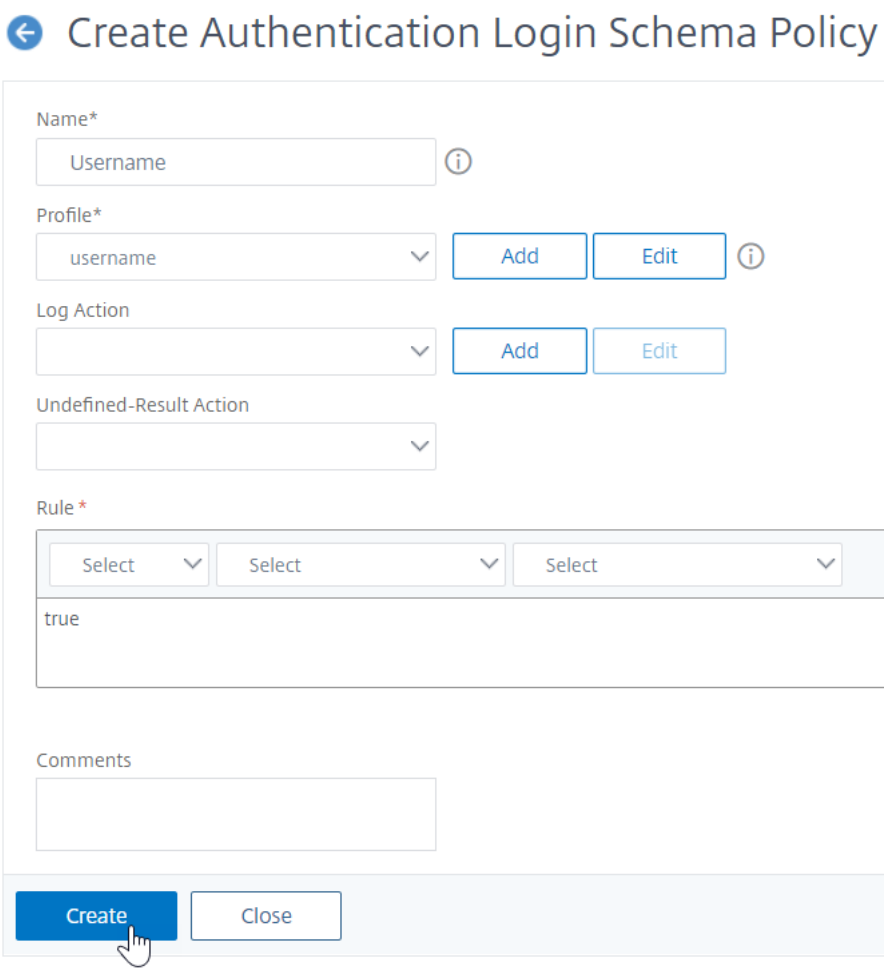
1. [セキュリティ] **>** [**AAA-**アプリケーショントラフィック] **>** [ログインスキーマ] に移動します。



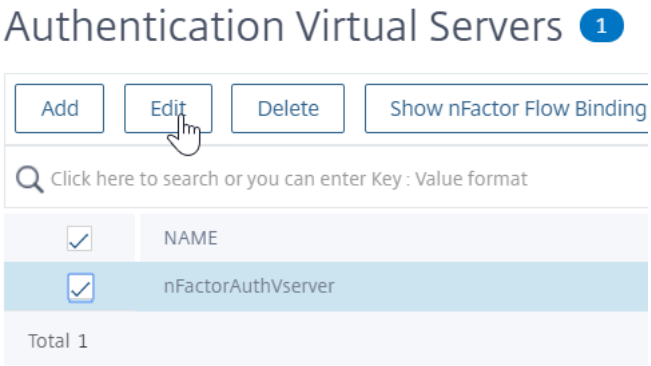
2. [**Policies**] タブで [**Add**] をクリックします。



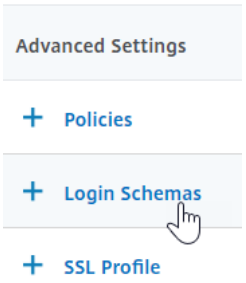
3. [**Profile**] ドロップダウンメニューを使用して、すでに作成したログインスキーマプロファイルを選択します。
4. [**Rule**] ボックスに高度なポリシー式を入力し、[**Create**] をクリックします。



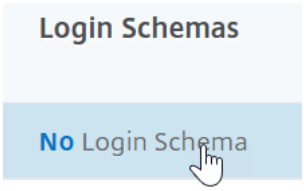
5. 左側で、[セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動し、既存の認証、認可、および監査仮想サーバを編集します。



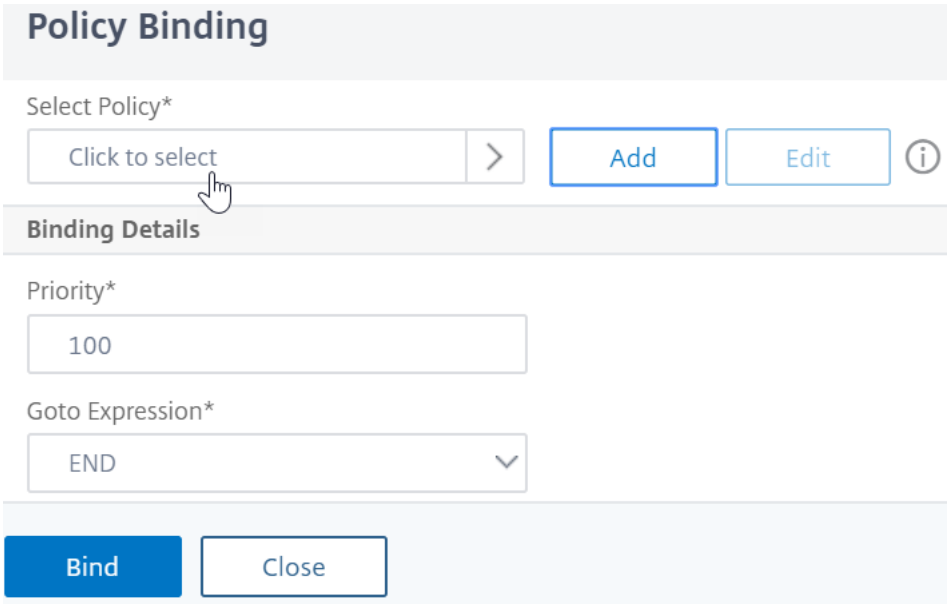
6. [詳細設定] 列で、[ログインスキーマ] をクリックします。



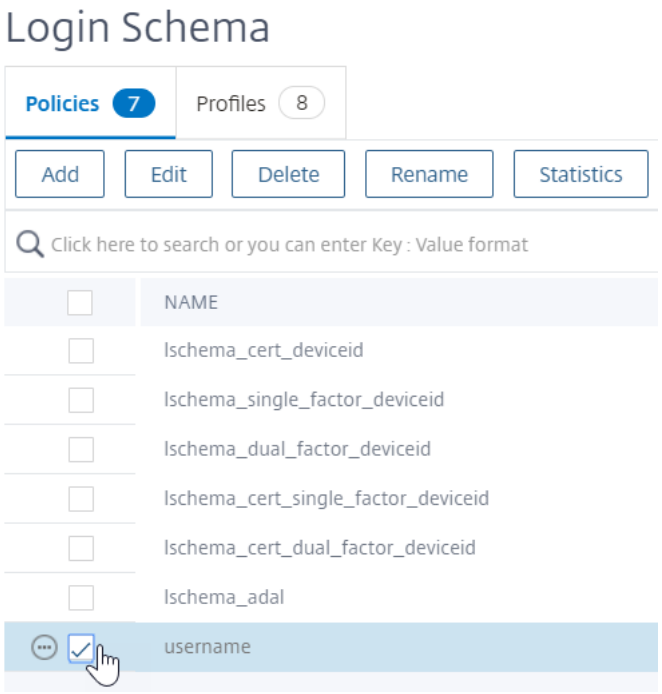
7. [ログインスキーマ] セクションで、[ログインスキーマなし] というテキストをクリックします。



8. テキストをクリックし、クリックして選択します。



9. ログインスキーマポリシーの横にあるオプションボタンをクリックし、[**Select**] をクリックします。このリストには、ログインスキーマポリシーのみが表示されます。ログインスキーマプロファイル (ポリシーなし) は表示されません。



10. ［バインド］をクリックします。

高度な認証ポリシー

認証ポリシーは、ポリシー式とポリシーアクションの組み合わせです。式が true の場合は、認証アクションを評価します。

高度な認証ポリシーの作成

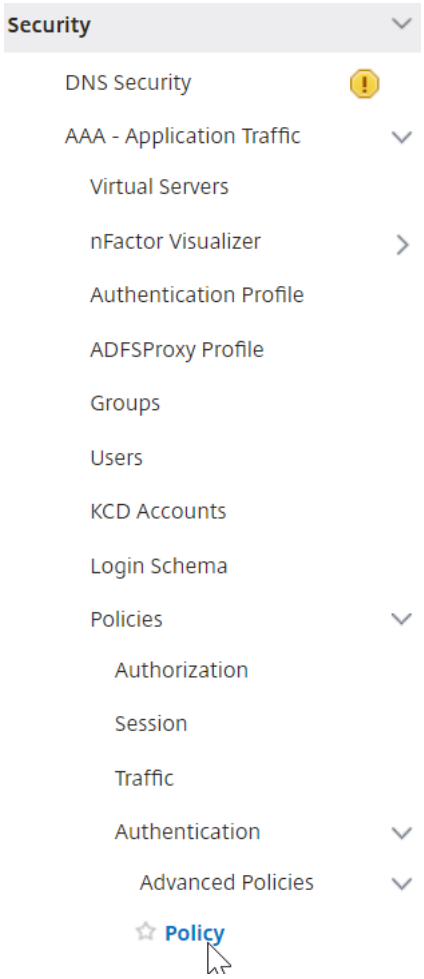
認証ポリシーは、ポリシー式とポリシーアクションの組み合わせです。式が true の場合は、認証アクションを評価します。

認証アクション/サーバー (LDAP、RADIUS、CERT、SAML など) が必要です。
高度な認証ポリシーを作成する場合、認証アクション/サーバーを作成するためのプラス (Add) アイコンが表示されます。

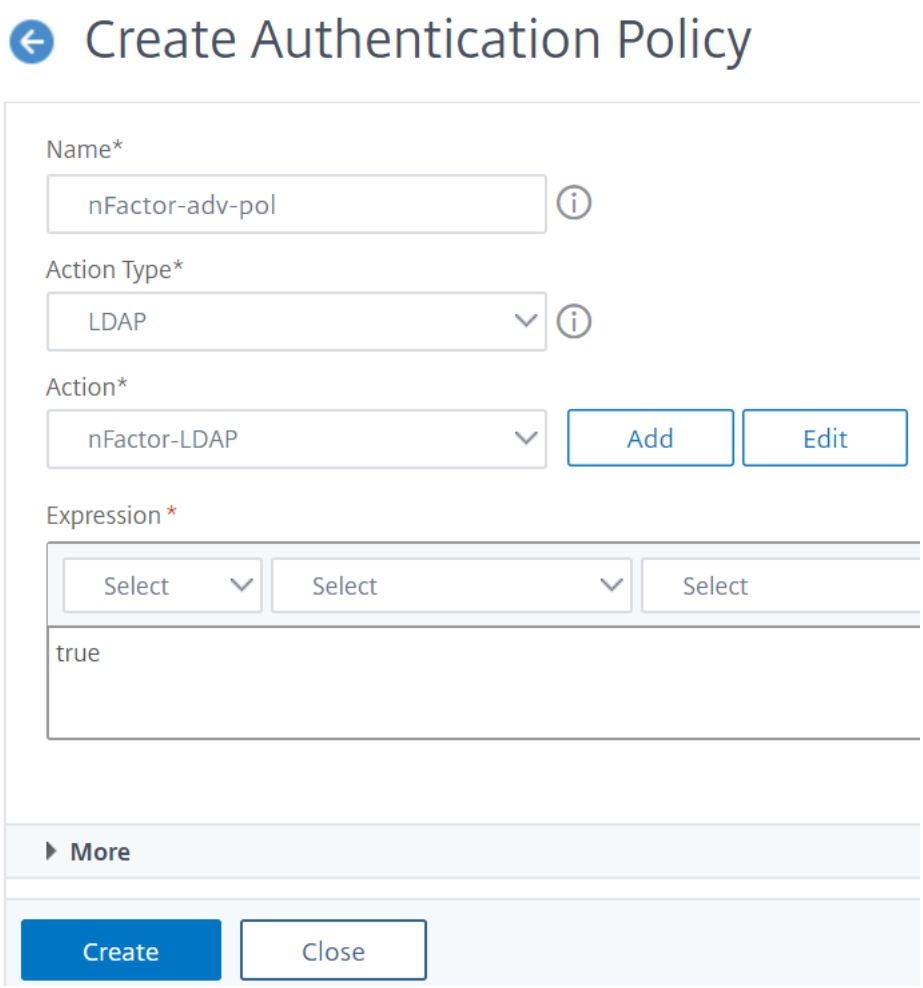
または、高度な認証ポリシーを作成する前に認証アクション (サーバー) を作成することもできます。認証サーバーは、[認証] **>** [ダッシュボード] の下にあります。右側の [追加] をクリックし、[サーバーの種類] を選択します。これらの認証サーバの作成手順については、ここでは詳しく説明しません。認証 — NetScaler 12/Citrix ADC 12.1 の手順を参照してください。

高度な認証ポリシーを作成するには、次の手順を実行します。

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 高度なポリシー **>** ポリシーへのナビゲート



2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。
3. [認証ポリシーの作成] または [認証ポリシーの構成] ダイアログボックスで、パラメータの値を入力または選択します。



- 名前 -ポリシー名。以前に設定したポリシーでは変更できません。
- アクションタイプ -ポリシータイプ：証明書、ネゴシエート、LDAP、RADIUS、SAML、SAMLIDP、TACACS、または WEBAUTH。
- アクション -ポリシーに関連付ける認証アクション (プロファイル)。既存の認証アクションを選択するか、プラス記号をクリックして適切なタイプのアクションを作成できます。
- ログアクション -ポリシーに関連付ける監査アクション。既存の監査アクションを選択するか、プラス記号をクリックしてアクションを作成できます。
  アクションが設定されていません。アクションを作成するには、[**Add**] をクリックして手順を完了します。
- 式 -指定したアクションを適用する接続を選択するルール。ルールは、シンプル (「true」はすべてのトラフィックを選択) または複雑にすることができます。式を入力するには、まず [式] ウィンドウの下にある左端のドロップダウンリストで式の種類を選択し、次に式テキスト領域に式を直接入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、その中のドロップダウンリストを使用して式。)
- [**Comment**]：この認証ポリシーが適用されるトラフィックのタイプを説明するコメントを入力できます。オプションです。

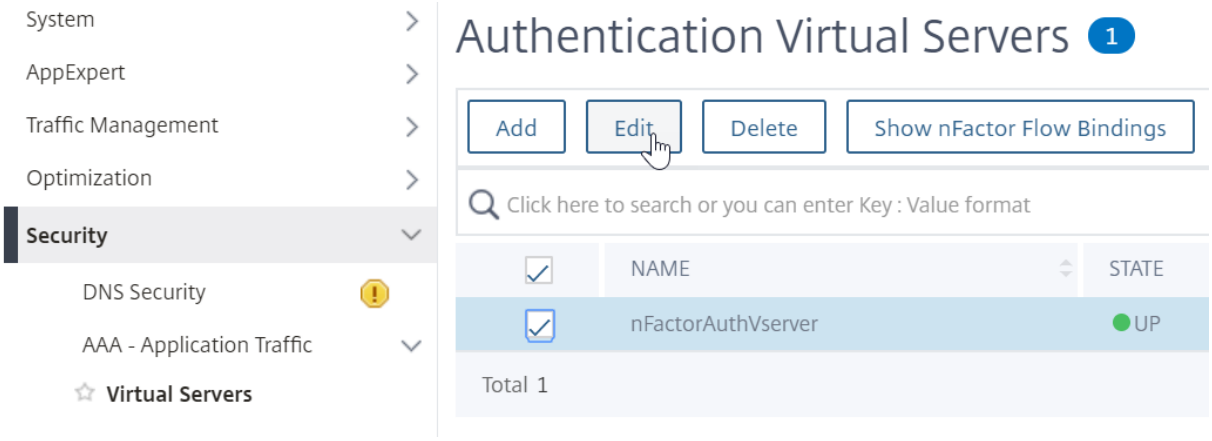
4. [作成] をクリックし、[閉じる] をクリックします。ポリシーを作成した場合、そのポリシーは [認証ポリシーおよびサーバ] ページに表示されます。

nFactor の設計に基づいて、必要に応じて追加の高度な認証ポリシーを作成します。

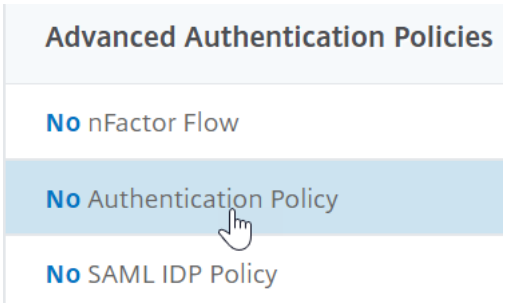
第 **1** 要素の高度な認証ポリシーを認証、承認、および監査にバインドする

最初の要素である認証、承認、および監査仮想サーバーに対して、高度な認証ポリシーを直接バインドできます。次の要素については、高度な認証ポリシーを認証ポリシーラベルにバインドする必要があります。

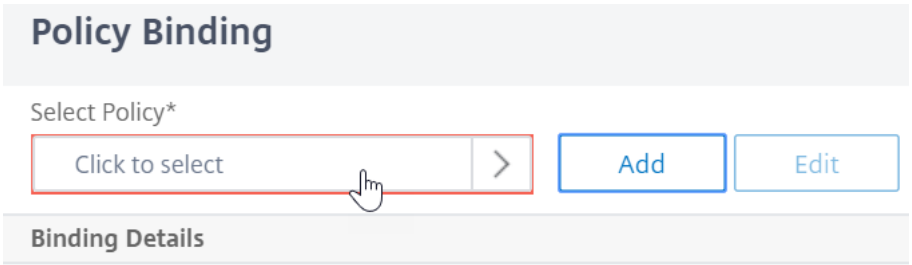
1. セキュリティ **> AAA-**アプリケーショントラフィック **>** 仮想サーバに移動します。既存の仮想サーバーを編集します。



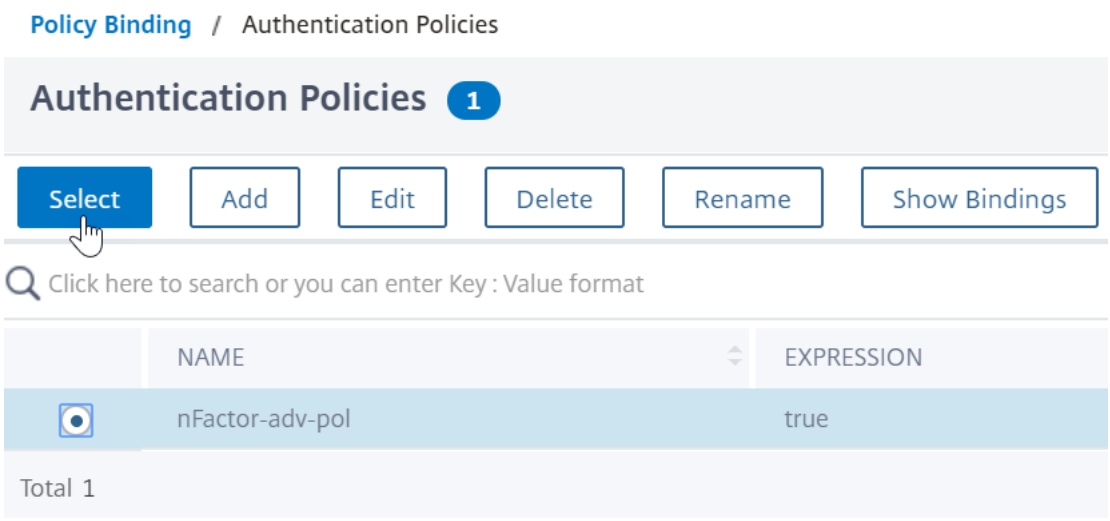
1. 左側の [高度な認証ポリシー] セクションで、[認証ポリシーなし] をクリックします。



2. **[**ポリシーの選択] で、[クリックして選択**]** というテキストをクリックします。

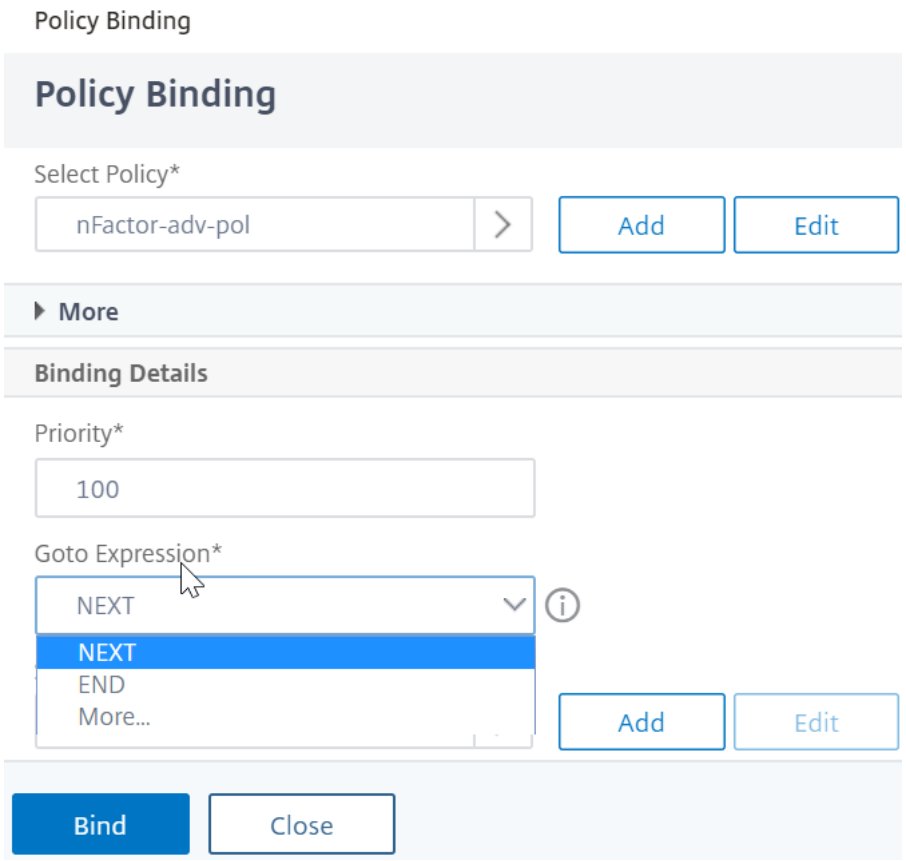


3. [高度な認証ポリシー**]** の横にあるオプションボタンをクリックし、[選択] をクリックします。

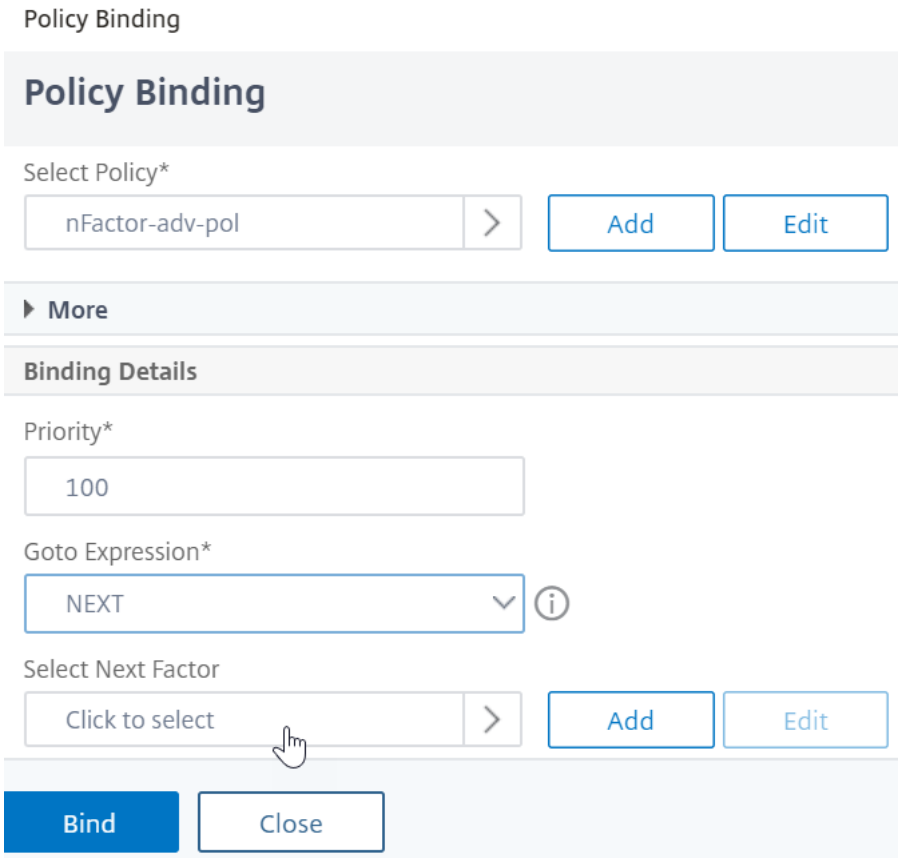


4. [バインドの詳細] セクションの [**Goto** 式] は、この高度な認証ポリシーが失敗した場合に次に何が起こるかを決定します。

   - **Goto Expression** が **NEXT** に設定されている場合、この認証、承認、および監査仮想サーバーにバインドされた次の高度な認証ポリシーが評価されます。
   - **Goto Expression** が **END** に設定されている場合、またはこの認証、承認、および監査仮想サーバーにバインドされた高度な認証ポリシーがこれ以上存在しない場合、認証は完了し、失敗としてマークされます。



5. [次の要素を選択] で、認証ポリシーラベルをポイントできるかを選択できます。次の要素は、高度な認証ポリシーが成功した場合にのみ評価されます。最後に、[バインド] をクリックします。



抽出された **LDAP** グループを使用して、次の認証要素を選択する

抽出された LDAP グループを使用して、実際に LDAP による認証を行わずに、次の認証要素を選択できます。

1. LDAP サーバーまたは LDAP アクションを作成または編集するときは、[認証] チェックボックスをオフにします。
2. [その他の設定] で、[グループ属性] と **[**** サブ属性名 ****]** で適切な値を選択します。

ポリシーラベルを認証する

高度な認証ポリシーを認証、承認、および監査仮想サーバーにバインドし、次の要素を選択した場合、次の要素は高度な認証ポリシーが成功した場合にのみ評価されます。次に評価される要素は、認証ポリシーラベルです。

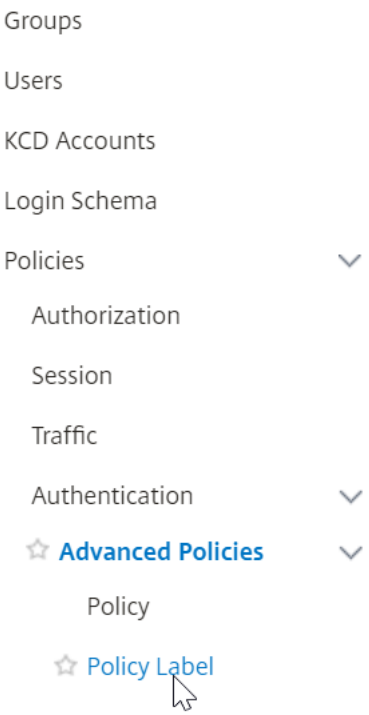
認証ポリシーラベルは、特定の要素に対する認証ポリシーのコレクションを指定します。各ポリシーラベルは 1 つの要素に対応します。また、ユーザーに提示する必要があるログインフォームも指定します。認証ポリシーラベルは、認証ポリシーまたは別の認証ポリシーラベルの次の要素としてバインドする必要があります。

注: すべての要素にログインスキーマは必要ありません。ログインスキーマプロファイルは、ログインスキーマを認証ポリシーラベルにバインドする場合にのみ必要です。

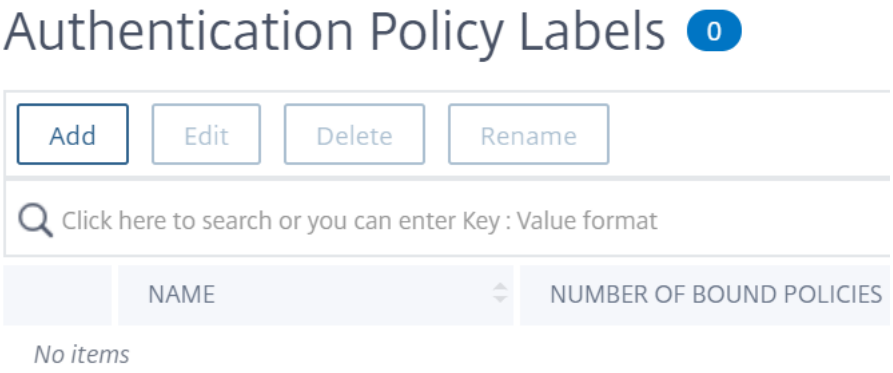
認証ポリシーラベルを作成する

ポリシーラベルは、特定のファクタの認証ポリシーを指定します。各ポリシーラベルは 1 つの要素に対応します。ポリシーラベルは、ユーザーに提示する必要があるログインフォームを指定します。ポリシーラベルは、認証ポリシーまたは別の認証ポリシーラベルの次の要素としてバインドする必要があります。通常、ポリシーラベルには、特定の認証メカニズムの認証ポリシーが含まれます。ただし、異なる認証メカニズムの認証ポリシーを持つポリシーラベルを使用することもできます。

1. セキュリティ **> AAA** ―アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 高度ポリシー **>** ポリシーラベルにナビゲートして下さい。



2. [追加] をクリックします。

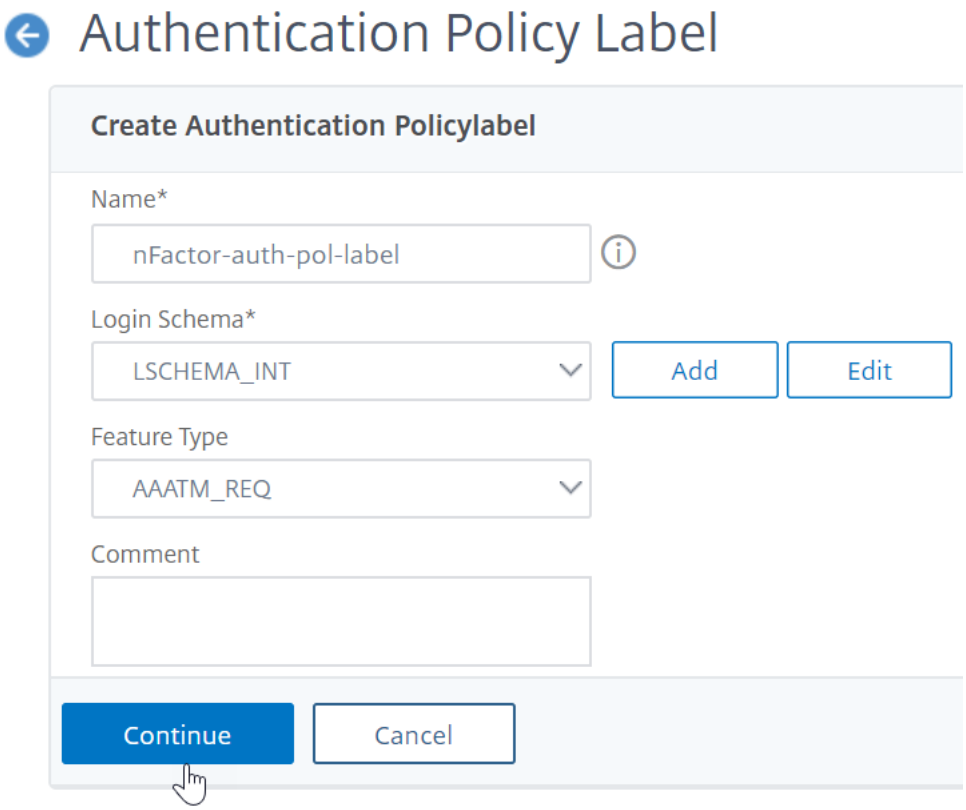


3. 次のフィールドに入力して、認証ポリシーラベルを作成します。

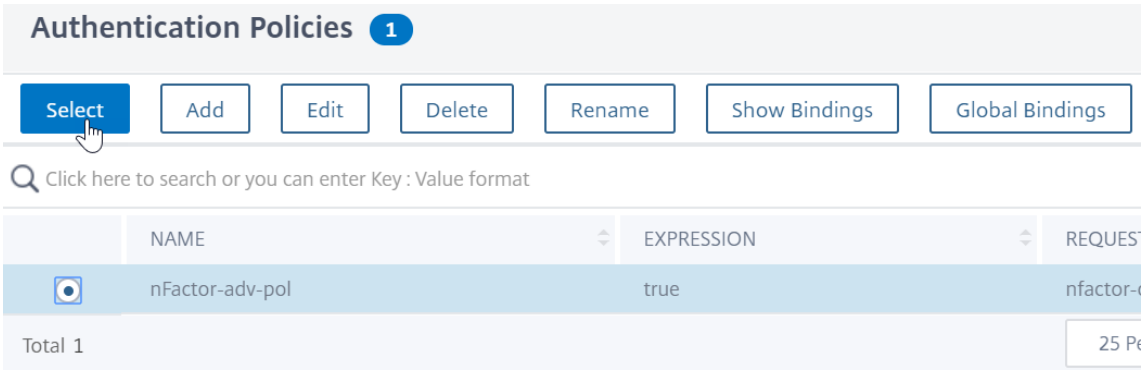
   a) 新しい認証ポリシーラベルの名前を入力します。

b) 認証ポリシーラベルに関連付けられたログインスキーマを選択します。ユーザーに対して何も表示しない場合は、スキーマなし (LSCHEMA_INT) に設定されたログインスキーマプロファイルを選択できます。

c) [続行] をクリックします。



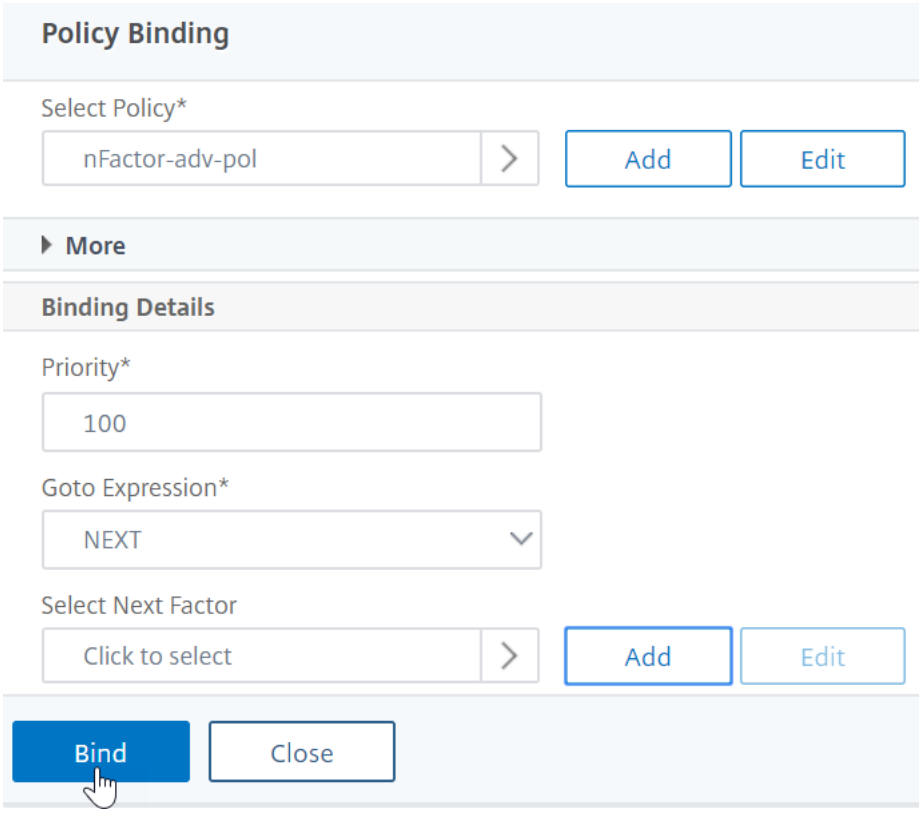
4. [ポリシーバインド] セクションで、[クリックして選択] と表示されている場所をクリックします。
5. この要素を評価する認証ポリシーを選択します。



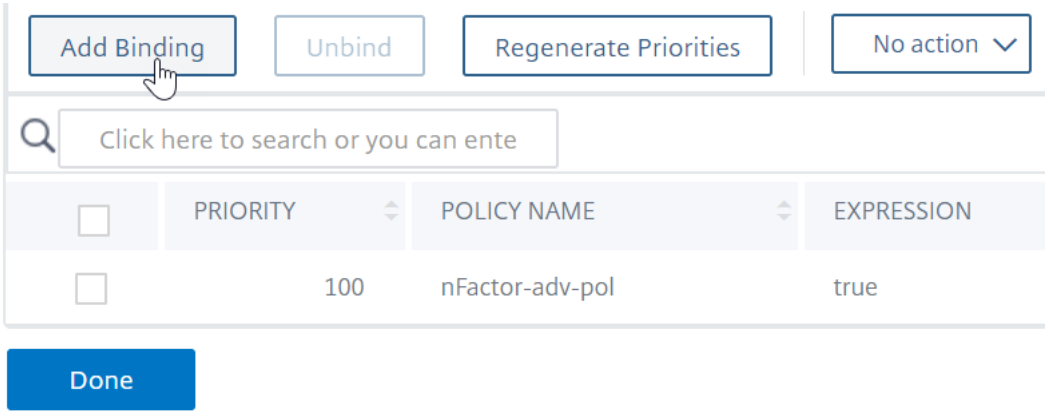
6. 次のフィールドに入力します。

a) ポリシーバインディングの優先度を入力します。

b) [式に移行] で、より高度な認証ポリシーをこの要素にバインドする場合は [次へ] を選択するか、[終了] を選択します。



7. 別の要素を追加する場合は、**[**次の要素の選択] で、次の認証ポリシーラベル (次の要素) をクリックして選択し、バインドします。
   次の要素を選択せず、この高度な認証ポリシーが成功すると、認証は成功し、完了します。
8. [バインド] をクリックします。
9. [**Add Binding**] をクリックすると、このポリシーラベル (ファクタ) に高度な認証ポリシーを追加できます。完了したら **[**完了] をクリックします。



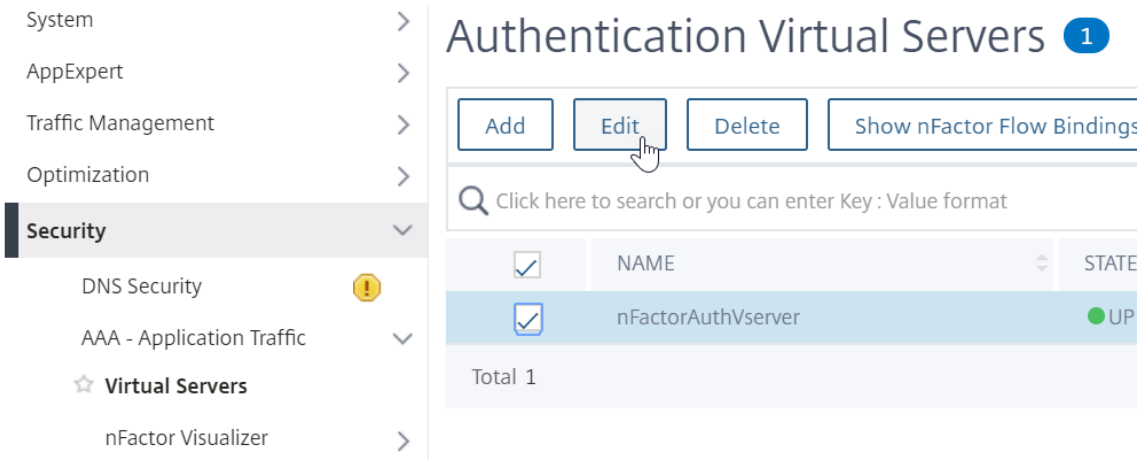
認証ポリシーラベルのバインド

ポリシーラベルを作成したら、既存の高度な認証ポリシーバインディングにポリシーラベルをバインドして、要素を連鎖させます。

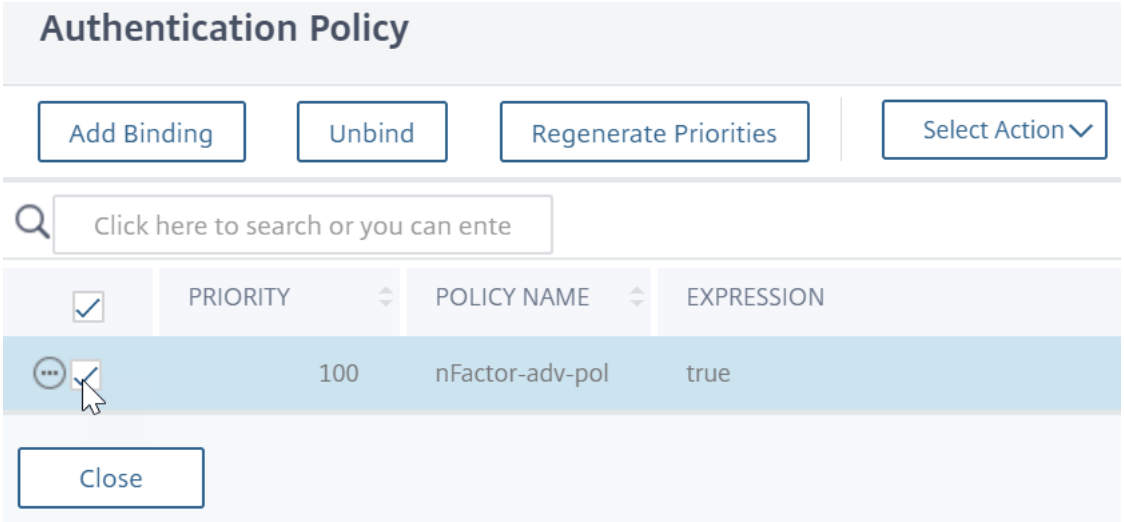
高度な認証ポリシーがバインドされている既存の認証、承認、および監査仮想サーバーを編集する場合、または別のポリシーラベルを編集して次の要素を含める場合は、次の要素を選択できます。

高度な認証ポリシーが既にバインドされている既存の認証、承認、および監査仮想サーバーを編集するには

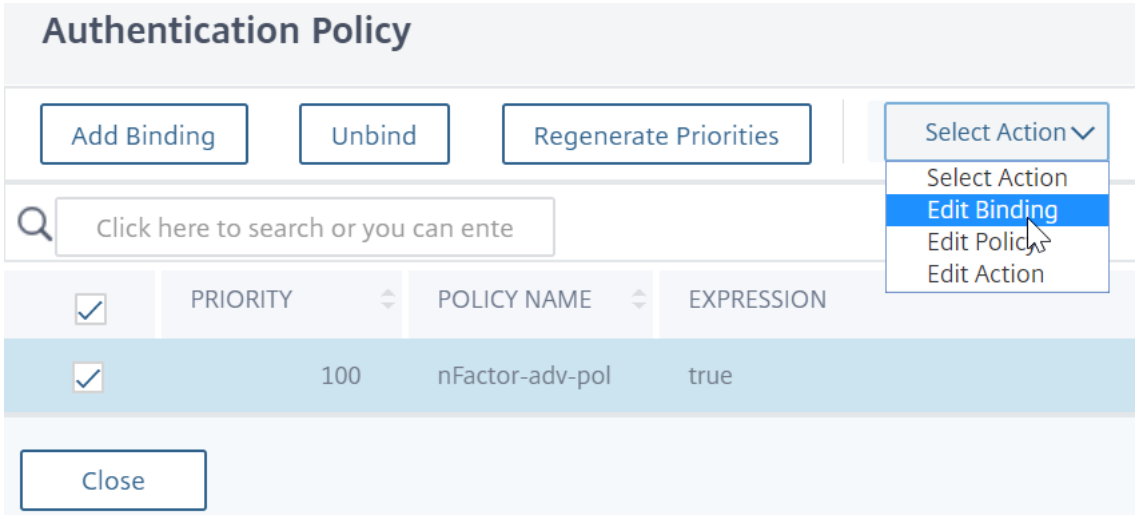
1. セキュリティ **> AAA —** アプリケーショントラフィック **>** 仮想サーバに移動します。仮想サーバを選択し、[編集（Edit）] をクリックします。



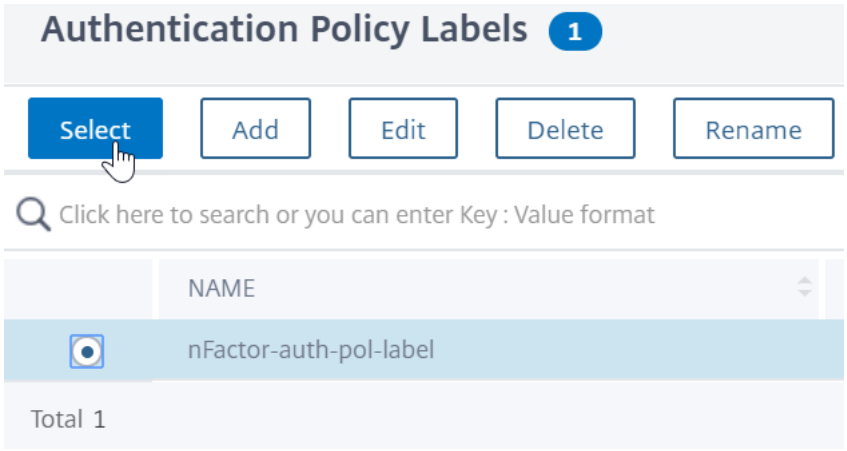
2. 左側の [高度な認証ポリシー] セクションで、既存の認証ポリシーバインドをクリックします。



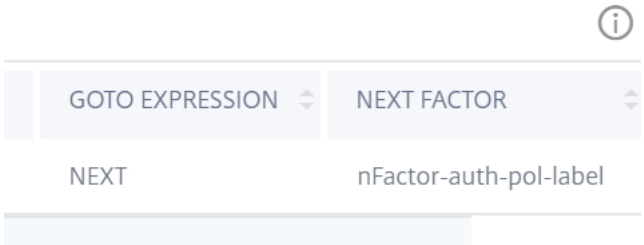
3.「アクションを選択」で、「バインドの編集」をクリックします。



4. **[**次の要素の選択] で既存の認証ポリシーラベル (次の要素) をクリックし、選択します。

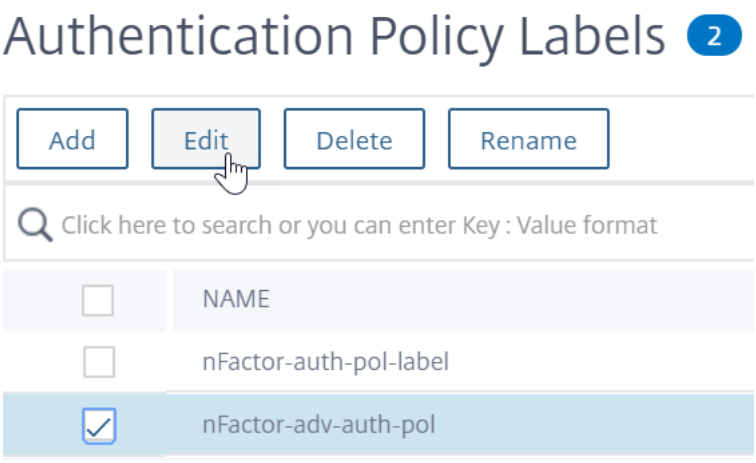


5. ［バインド］をクリックします。右端で次の要因を表示できる

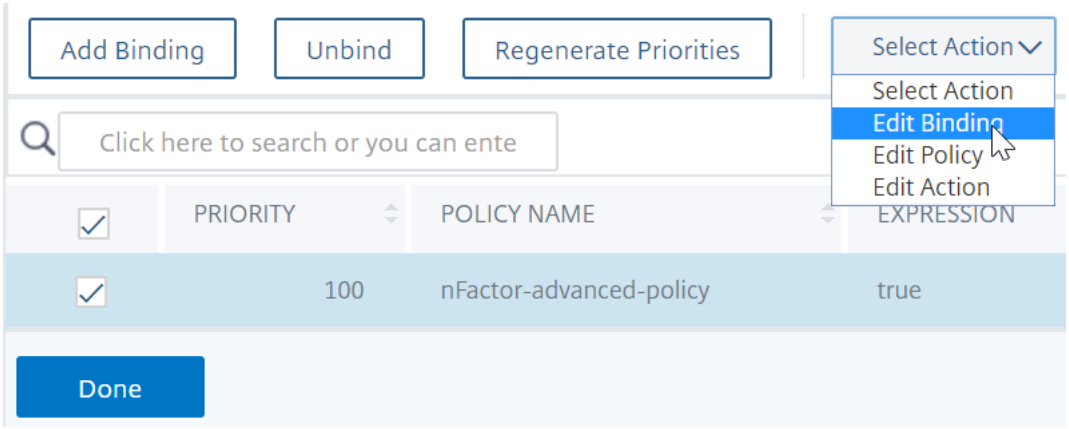


ポリシーラベルネクストファクタを別のポリシーラベルに追加するには

1. セキュリティ **> AAA** ―アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 高度ポリシー **>** ポリシーラベルにナビゲートして下さい。別のポリシーラベルを選択し、[**Edit]** をクリックします。

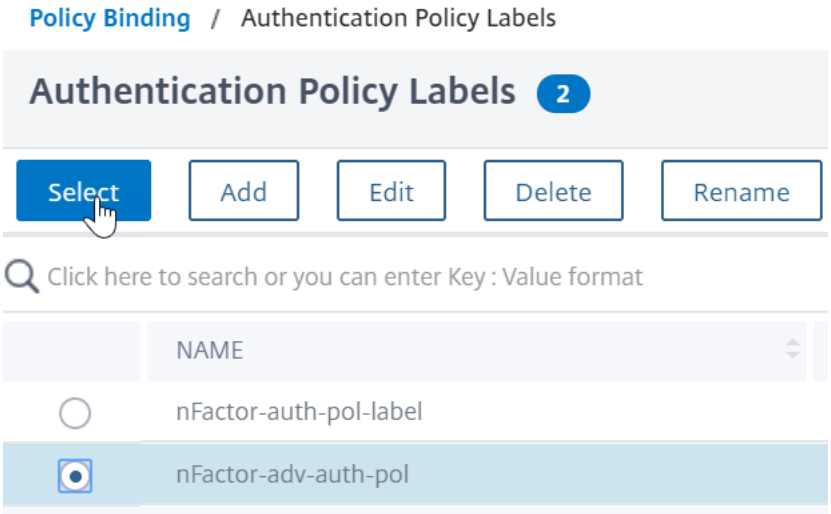


2. 「アクションを選択」で、「バインドの編集」をクリックします。

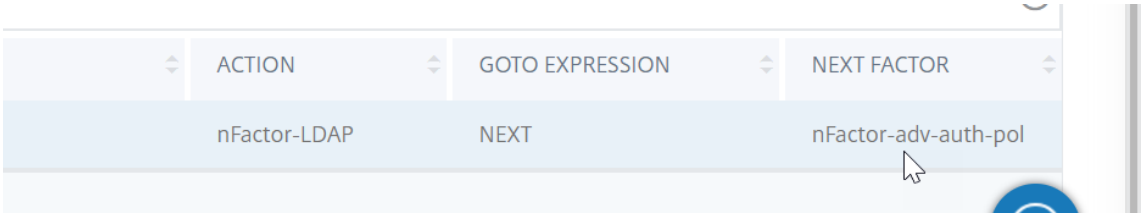


3. 「バインドの詳細」**>**「次のファクタを選択」で、次のファクタをクリックして選択します。

4. 次のファクタのポリシーラベルを選択し、[**Select**] ボタンをクリックします。



5. バインドをクリックします。次の要因が右側に表示されます。

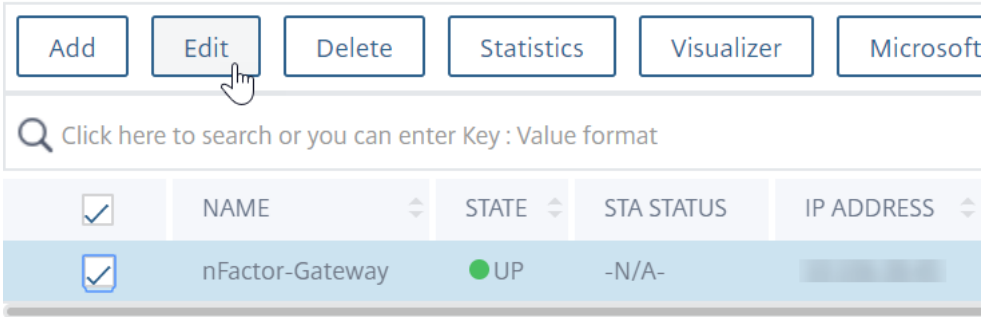


Citrix Gateway の nFactor

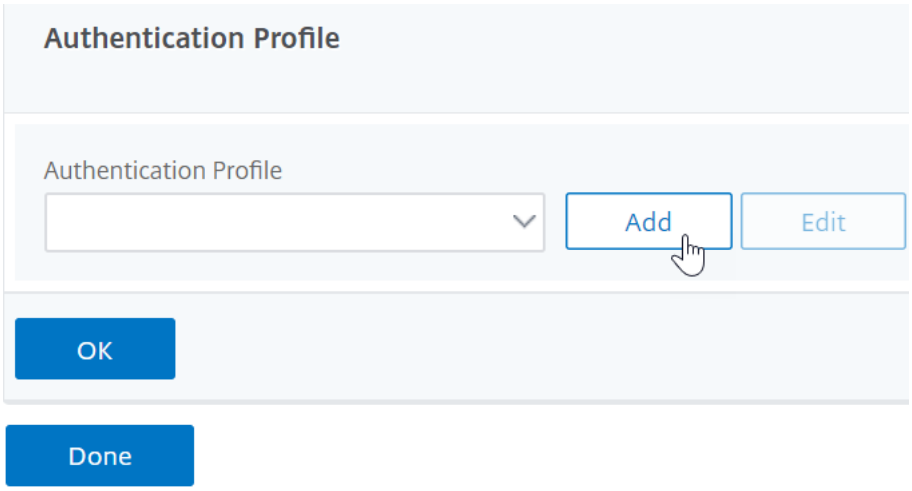
Citrix Gateway で nFactor を有効にするには、認証プロファイルを認証、承認、および監査仮想サーバーにリンクする必要があります。

認証、承認、監査仮想サーバーを **Citrix Gateway** 仮想サーバーにリンクする認証プロファイルを作成する

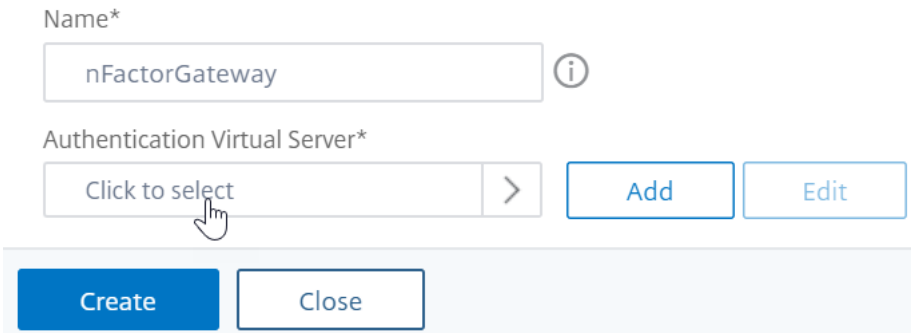
1. [**Citrix Gateway] > [**仮想サーバー] に移動し、編集する既存のゲートウェイ仮想サーバーを選択します。



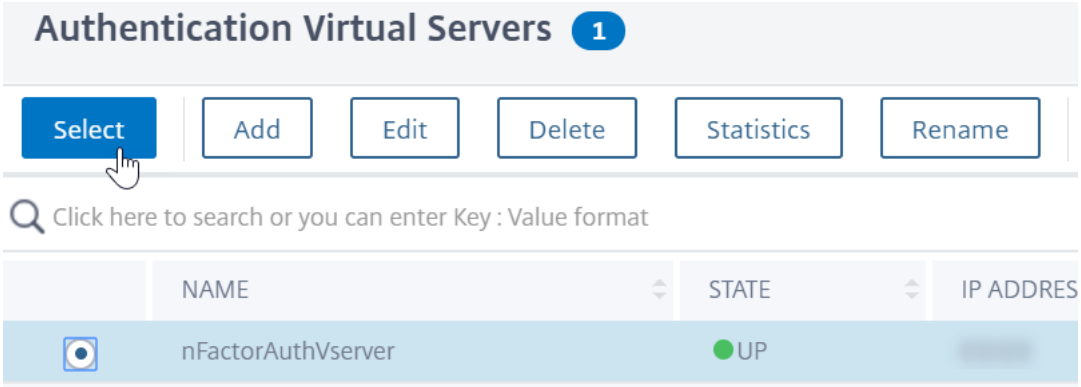
2. [詳細設定] で、[認証プロファイル] をクリックします。
3. [認証プロファイル] の [追加] をクリックします



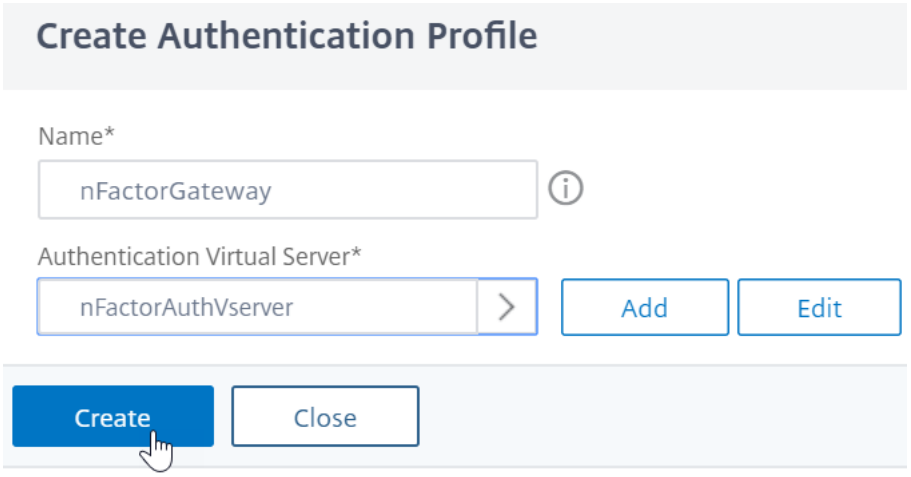
4. 認証プロファイルの名前を入力し、[クリックして選択] と表示されている場所をクリックします。



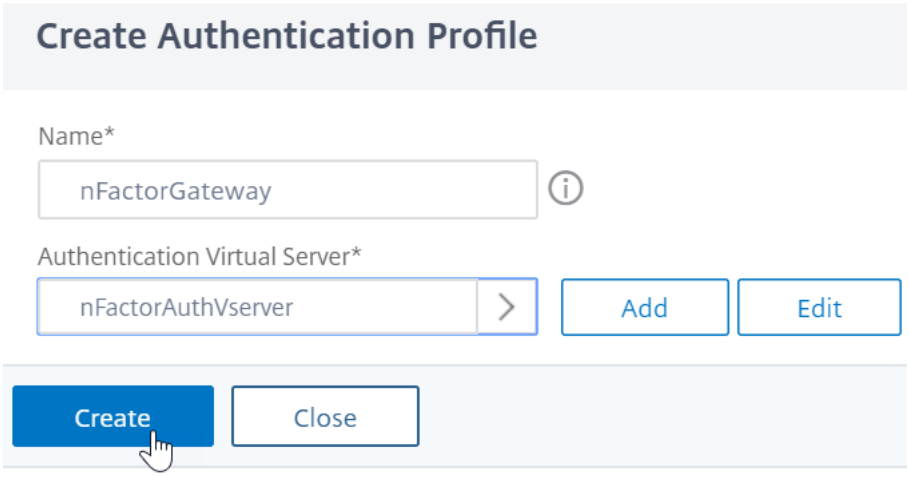
5. [認証仮想サーバー] で、ログインスキーマ、高度な認証ポリシー、および認証ポリシーのラベルが構成されている既存のサーバーを選択します。認証仮想サーバーを作成することもできます。認証、承認、および監査仮想サーバーには IP アドレスは必要ありません。［選択］をクリックします。



6. ［**Create**］をクリックします。



7. ［**OK**］をクリックして、［認証プロファイル］セクションを閉じます。



注**:** ファクタの 1 つをクライアント証明書として設定した場合は、SSL パラメータと CA 証明書を設定する必要があります。

認証、承認、および監査仮想サーバーへの認証プロファイルのリンクが完了し、Citrix Gateway を参照すると、nFactor 認証画面を表示できます。

SSL パラメータと **CA** 証明書の設定

認証要素の 1 つが証明書の場合は、Citrix Gateway 仮想サーバーで SSL 構成を実行する必要があります。

1. [トラフィック管理**] > [SSL] > [**証明書**] > [CA** 証明書] に移動し、クライアント証明書の発行者のルート証明書をインストールします。認証局の証明書にはキーファイルは必要ありません。

   デフォルトの SSL プロファイルが有効な場合は、クライアント認証が有効になっている SSL プロファイルが既に作成されています。

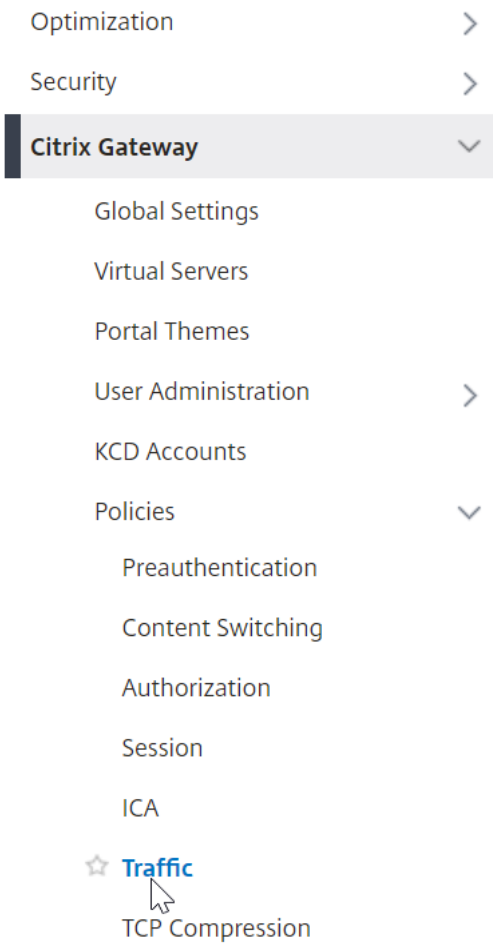
2. [**Citrix Gateway**] **>** [仮想サーバー] に移動し、nFactor が有効になっている既存の Citrix Gateway 仮想サーバーを編集します。
   - デフォルトの SSL プロファイルが有効になっている場合は、[編集（Edit）] アイコンをクリックします。
   - [SSL プロファイル] リストで、[クライアント認証] が有効で [オプション] に設定されている SSL プロファイルを選択します。
   - デフォルトの SSL プロファイルが有効でない場合は、[編集（Edit）] アイコンをクリックします。
   - [クライアント認証] チェックボックスをオンにします。
   - [クライアント証明書] が [任意] に設定されていることを確認します。
3. [OK] をクリックします。
4. [証明書] セクションで、[**CA** 証明書なし**]** をクリックします。
5. [CA 証明書の選択] で、クライアント証明書の発行元のルート証明書をクリックして選択します。
6. [バインド] をクリックします。

注: クライアント証明書を発行した中間 CA 証明書もバインドする必要がある場合があります。

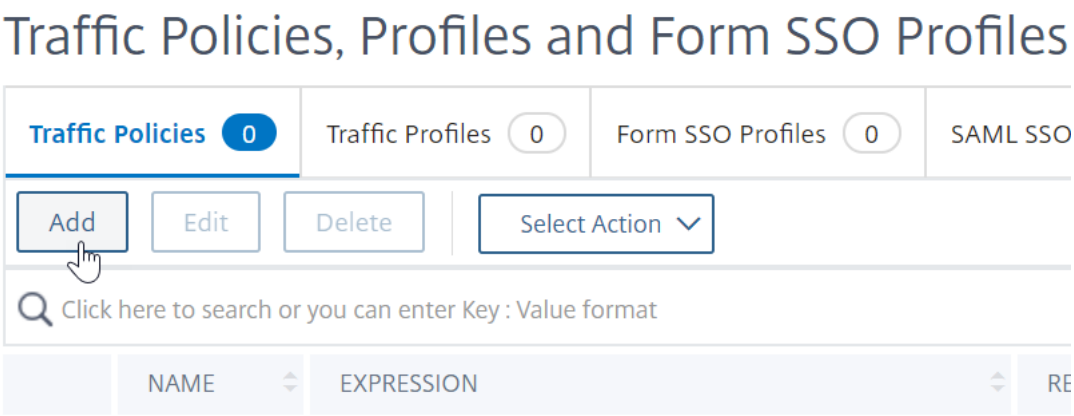
StoreFront への **nFactor** シングルサインオン用の **Citrix Gateway** トラフィックポリシーを構成する

StoreFront へのシングルサインオンでは、nFactor はデフォルトで最後に入力したパスワードを使用します。LDAP が最後に入力されたパスワードでない場合は、トラフィックポリシー/プロファイルを作成して、デフォルトの nFactor 動作を上書きする必要があります。

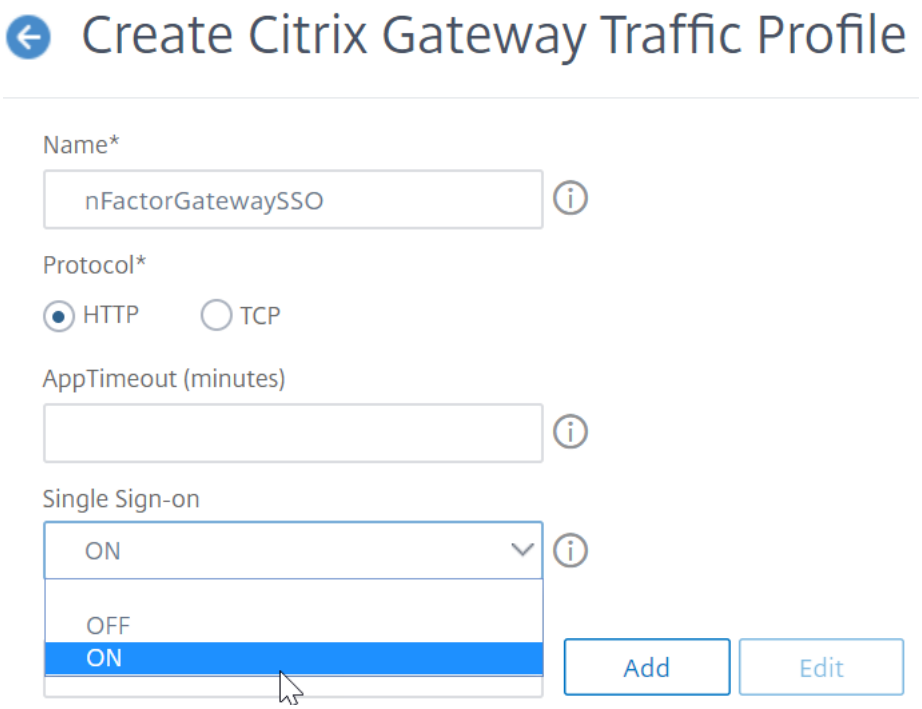
1. **Citrix Gateway >** ポリシー **>** トラフィックに移動します。



2. [トラフィックプロファイル] タブで、[追加] をクリックします。

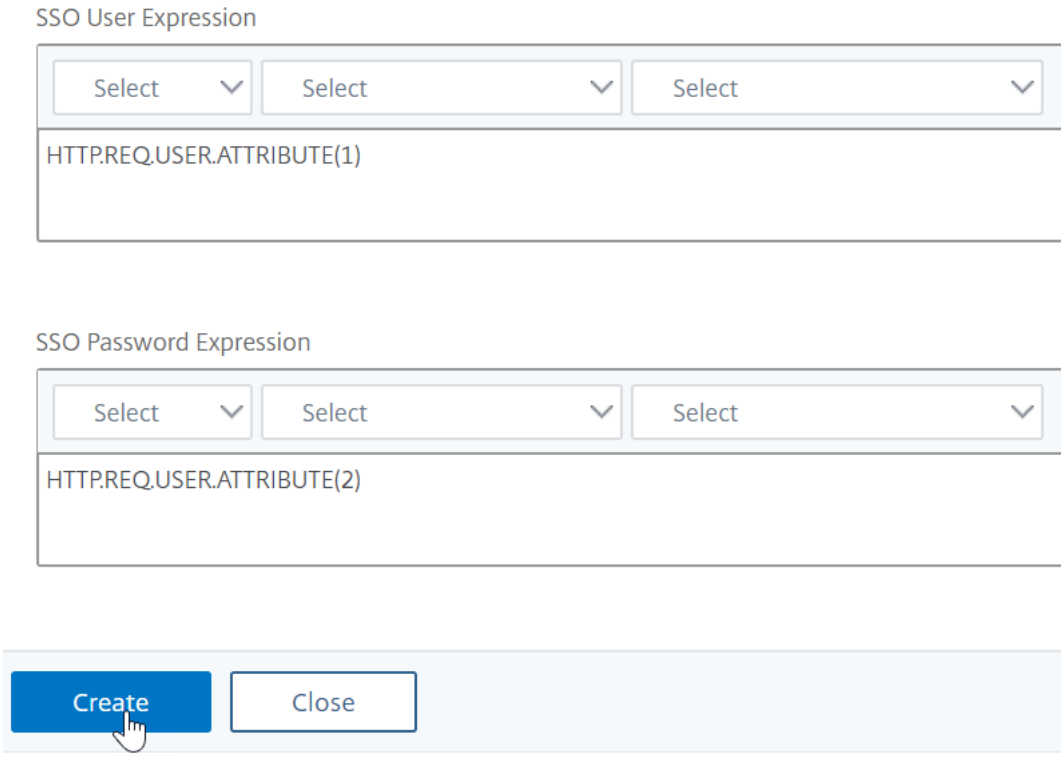


3. トラフィックプロファイルの名前を入力します。**HTTP** プロトコルを選択します。
[シングルサインオン] で、[オン] を選択します。



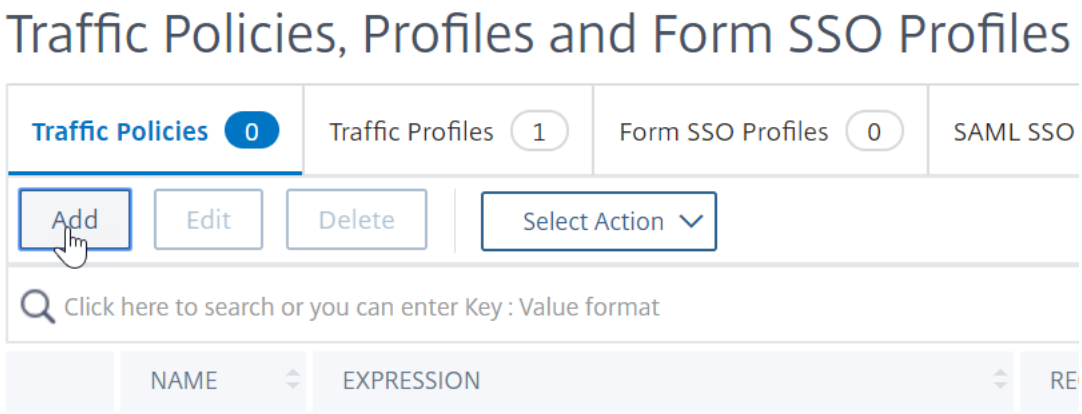
4. [**SSO** 式] で、ログインスキーマで指定されたインデックスと一致する AAA.USER.ATTRIBUTE (#) 式を入力し、[作成] をクリックします。

> 注
> AAA.USER 式は、非推奨の HTTP.REQ.USER 式を置き換えるように実装されるようになりました。



5. [トラフィックポリシー] タブをクリックし、[追加] をクリックします。

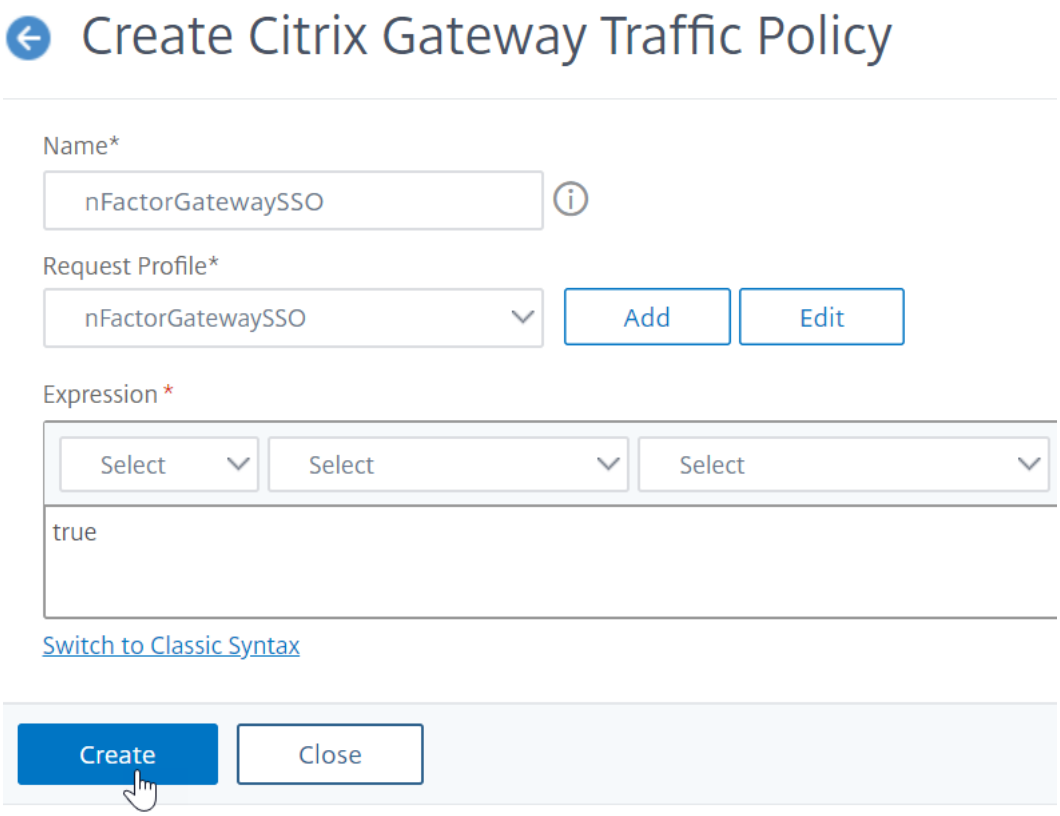
   ポリシーの名前を入力します。
   前の手順で作成したトラフィックプロファイルを選択します。
   [式] に、高度な式 (true など) を入力します。
   「作成」をクリックします。



6. [**Citrix** ゲートウェイ] **>** [**Citrix Gateway** 仮想サーバー] に移動します。

   - 既存の仮想サーバを選択し、[編集] をクリックします。
   - [ポリシー] セクションで、[**+**] 記号をクリックします。
   - **[**ポリシーの選択] で [トラフィック] を選択します
   - 「タイプの選択」で、「要求」を選択します。

- 作成したトラフィックポリシーを選択し、[**Bind**] をクリックします。



Citrix ADC CLI を使用した **nFactor** 構成のサンプルスニペット

nFactor 認証の段階的な設定を理解するために、最初の要素が LDAP 認証で、2 番目の要素が RADIUS 認証である 2 要素認証の展開について考えてみましょう。

このサンプル展開では、ユーザーは単一のログインフォームを使用して両方の要素にログインする必要があります。したがって、2 つのパスワードを受け入れる単一のログインフォームを定義します。最初のパスワードは LDAP 認証に使用され、もう 1 つは RADIUS 認証に使用されます。
実行される設定は次のとおりです。

1. 認証用の負荷分散仮想サーバーの構成

   add lb vserver lbvs89 HTTP 1.136.19.55 80 -AuthenticationHost auth56.aaatm.com -Authentication ON

2. 認証仮想サーバを設定します。

   add authentication vserver auth56 SSL 10.106.30.223 443 -AuthenticationDomain aaatm.com

3. ログインフォームのログインスキーマを設定し、ログインスキーマポリシーにバインドします。

   add authentication loginSchema login1 -authenticationSchema login-2passwd.xml -userCredentialIndex 1 -passwordCredentialIndex 2

注:

StoreFront などのバックエンドサービスへのシングルサインオン（SSO）には、ログインスキーマに入力したユーザー名とパスワードのいずれかを使用します。AAA.USER.ATTRIBUTE (#) 式を使用して、トラフィックアクションでこれらのインデックス値を参照できます。値は 1 から 16 の範囲で指定できます。

または、次のコマンドを使用して、ログインスキーマに入力された資格情報をシングルサインオンの認証情報として使用できます。

```
add authentication loginSchema login1 -authenticationSchema login
    -2passwd.xml -SSOCredentials YES

 add authentication loginSchemaPolicy login1 -rule true -action
     login1
<!--NeedCopy-->
```

4. パススルーのログインスキーマを構成し、ポリシーラベルにバインドします。

```
add authentication loginSchema login2 -authenticationSchema
    noschema

add authentication policylabel label1 -loginSchema login2
<!--NeedCopy-->
```

5. LDAP ポリシーと RADIUS ポリシーを設定します。

```
add authentication ldapAction ldapAct1 -serverIP 10.17.103.28 -
    ldapBase "dc=aaatm, dc=com" -ldapBindDn administrator@aaatm.com
     -ldapBindDnPassword 81
    qw1b99ui971mn1289op1abc12542389b1f6c111n0d98e1d78ae90c8545901 -
    encrypted -encryptmethod ENCMTHD_3 -ldapLoginName
    samAccountName -groupAttrName memberOf -subAttributeName CN

add authentication Policy ldap -rule true -action ldapAct1

add authentication radiusAction radius -serverIP 10.101.14.3 -
    radKey
    n231d9a8cao8671or4a9ace940d8623babca0f092gfv4n5598ngc40b18876hj32
     -encrypted -encryptmethod ENCMTHD_3 -radNASip ENABLED -
    radNASid NS28.50 -radAttributeType 11 -ipAttributeType 8

```



```
7 add authentication Policy radius -rule true -action radius
8 <!--NeedCopy-->
```

6. ログインスキーマポリシーを認証仮想サーバーにバインドします。

```
1 bind authentication vserver auth56 -policy login1 -priority 1 -
     gotoPriorityExpression END
2 <!--NeedCopy-->
```

7. LDAP ポリシー (第 1 要素) を認証仮想サーバーにバインドします。

```
1 bind authentication vserver auth56 -policy ldap -priority 1 -
     nextFactor label1 -gotoPriorityExpression next
2 <!--NeedCopy-->
```

8. RADIUS ポリシー（第 2 要素）を認証ポリシーラベルにバインドします。

```
1 bind authentication policylabel label1 -policyName radius -
     priority 2 -gotoPriorityExpression end
2 <!--NeedCopy-->
```

シンプルな構成のための **nFactor** ビジュアライザー

January 31, 2022

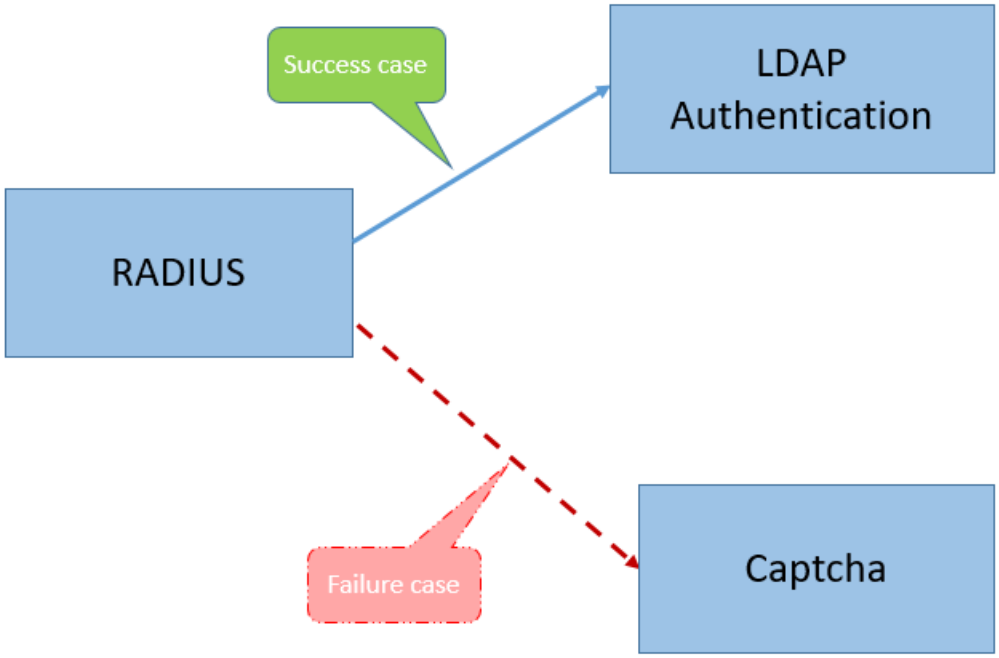
Citrix ADC リリース 13.0 ビルド 36.27 以降、GUI を介した nFactor 構成は、nFactorVisualizer を使用することで簡素化されています。nFactor ビジュアライザーは、管理者が各要因の追跡を失うことなく、複数の要因を追加するのに役立ちます。フローに構築された因子のグループが 1 か所に表示されます。管理者は、認証の成功パスと失敗パスを別々に追加できます。フローを作成した後、管理者は nFactor フローを認証仮想サーバーにバインドする必要があります。

> 注
>
> - 管理者が nFactor フローで作成したすべての要素は、将来の使用のために保持されます。
> - Citrix ADC 機能リリース 13.0 ビルド 64.35 以降では、nFactor ビジュアライザーを使用して、決定ブロックを使用して nFactor フローを開始できます。

以前は、nFactor の設定は煩雑で、管理者は多くのページにアクセスして設定する必要がありました。変更が必要な場合、管理者は毎回構成済みのセクションを再確認する必要がありました。また、完全な構成を 1 か所で表示するオプションもありませんでした。

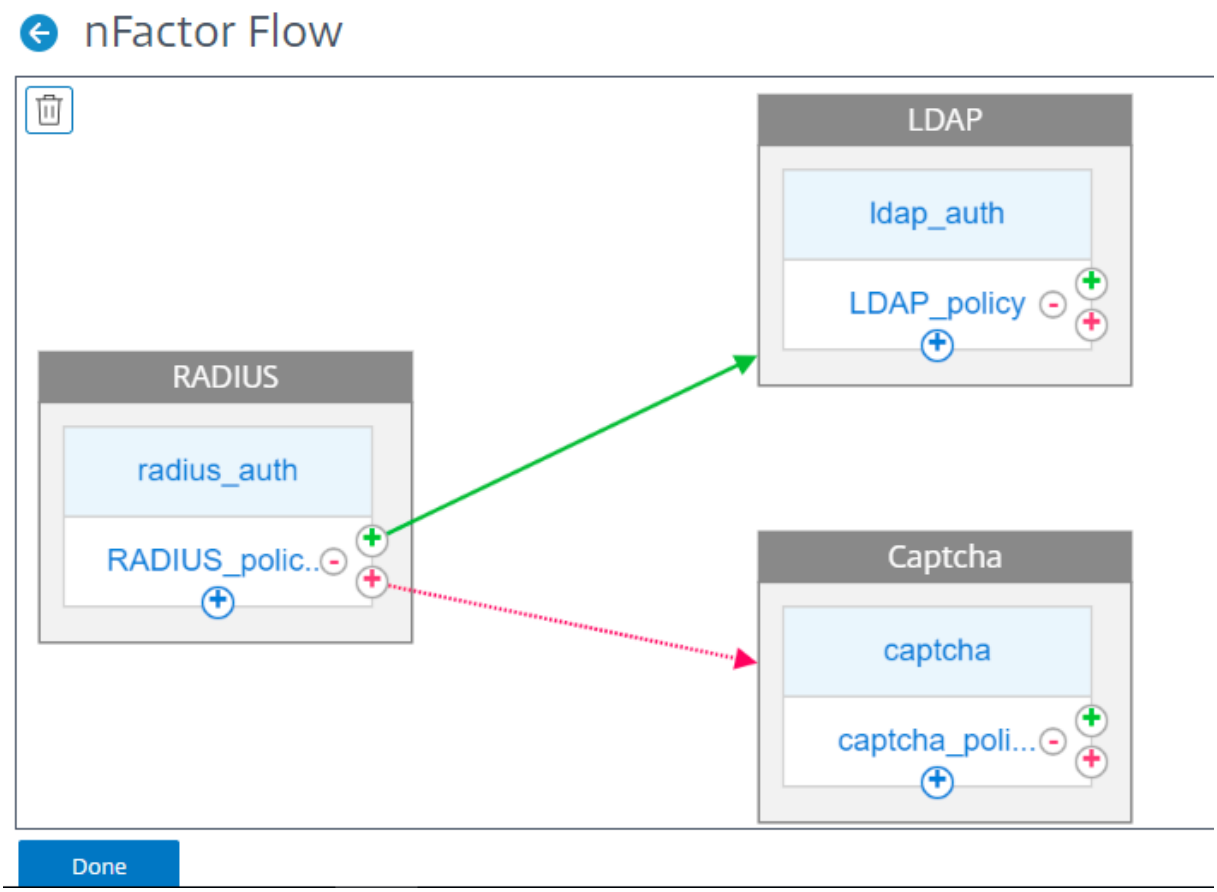
ユースケース **1**：**RADIUS** に続いて **LDAP** 認証、それ以外の場合は **nFactor** ビジュアライザを介してキャプチャにフォールバック

第 1 レベルの認証として RADIUS 認証を実現し、続いて LDAP 認証を実行します。RADIUS が失敗した場合、認証はキャプチャにフォールバックする必要があります。



このユースケースを実現するには、nFactor ビジュアライザーを使用します。ビジュアライザーには、このフローと関連項目を追加するために使用できるさまざまなコントロールが用意されています。

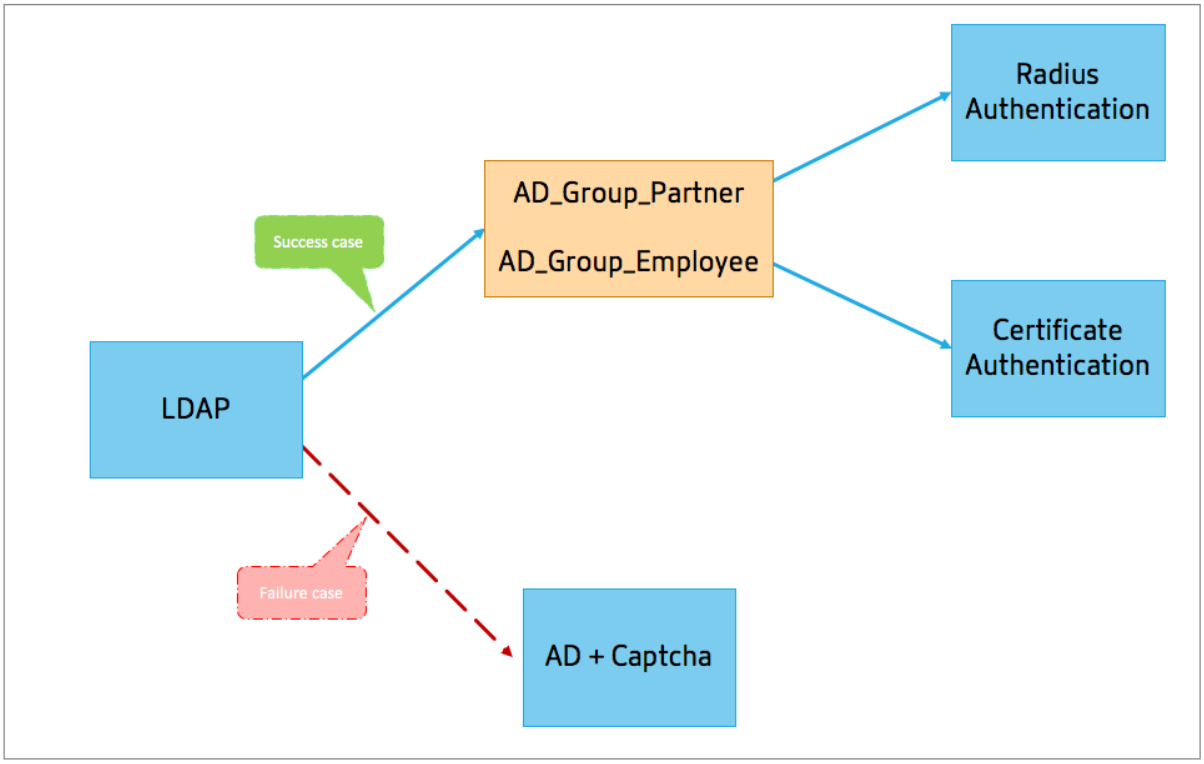
次の図は、ビジュアライザを使用して前述のユースケース用に作成された nFactor フローを示しています。



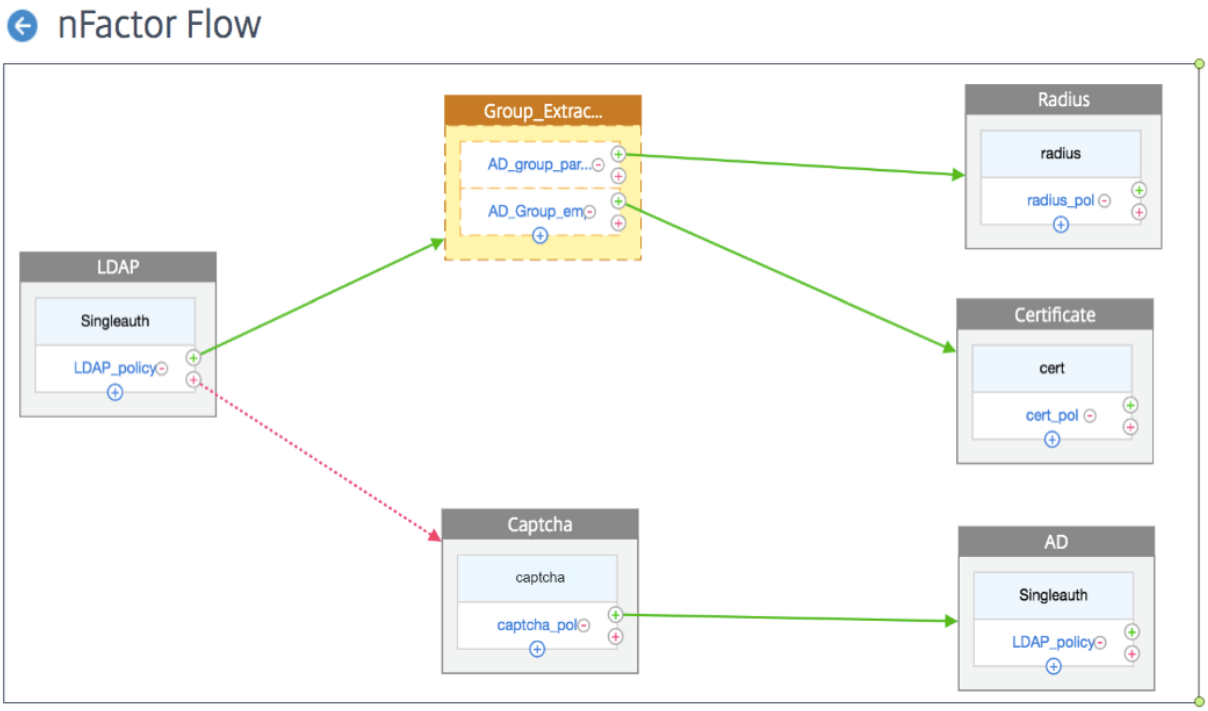
- 半径。RADIUS を最初の要素として設定します。ログインスキーマとポリシーを追加します。この例では、radius_auth と RADIUS_policy が追加されるログインスキーマおよびポリシーです。のために RADIUS_Policy, 成功事例には別の要素を追加できます。この例では、成功の場合に LDAP ファクターブロックが追加されています。失敗した場合は、キャプチャ係数を追加できます。
- **LDAP**。LDAP 認証は、2 番目の要素として設定します。ログインスキーマとポリシーを追加します。この例では、ldap_auth と LDAP_policy が追加されるログインスキーマとポリシーです。
- **Captcha**。RADIUS ポリシー障害の場合は、キャプチャ係数を作成します。この例では、captcha と captcha_policy が追加されるログインスキーマとポリシーです。

ユースケース **2**: **LDAP** の後に、**nFactor** ビジュアライザによる **LDAP** グループメンバーシップに基づくキャプチャによる **RADIUS/**証明書認証

第 1 レベルの認証として RADIUS 認証を実現し、続いて LDAP 認証を実行します。RADIUS が失敗した場合、認証はキャプチャにフォールバックする必要があります。



次の図は、ビジュアライザを使用して前述のユースケース用に作成された nFactor フローを示しています。



- **LDAP**。最初の要素として LDAP を設定します。ログインスキーマとポリシーを追加します。この例では、SingleAuth と LDAP_Policy が追加されるログインスキーマとポリシーです。のために LDAP_Policy, 成功事例には別の要素を追加できます。この例では、成功ケースの決定ブロックが追加されています。失敗した場

合は、キャプチャの後に AD 係数を追加できます。

- グループ抽出 **LDAP**。LDAP 成功ケースに追加される決定ブロックです。デシジョンブロックは、ポリシー規則に基づいてユーザーを分岐する分岐アウト係数として使用されます。ビジュアライザーでは、デシジョンブロックに対して NO_AUTHN ポリシーのみを構成できます。

  この例では、グループ _ 抽出 _LDAP がデシジョンブロックです。この決定ブロックに 2 つのポリシー（`AD_Group_Partner and AD_Group_Employee`）を追加します。ユースケースで説明されているように、AD_Group_Partner ポリシーを介してルーティングされるすべての要求は RADIUS 認証を使用します。したがって、このポリシーの成功事例を、RADIUS ファクタである次のファクタに接続します。同様に、AD_Group_Employee ポリシーを介してルーティングされるすべての要求は、証明書認証を使用します。したがって、このポリシーの成功事例を、証明の認証係数である次の要素に接続します。

  - 半径。のために AD_Group_Partner ポリシーが成功した場合は、RADIUS 認証要素を作成します。
  - 証明書。のために AD_Group_Employee ポリシーが成功した場合は、証明書認証要素を作成します。

- **Captcha**。LDAP ポリシーの障害の場合は、次の 2 つの要素、キャプチャと AD ファクタを作成します。

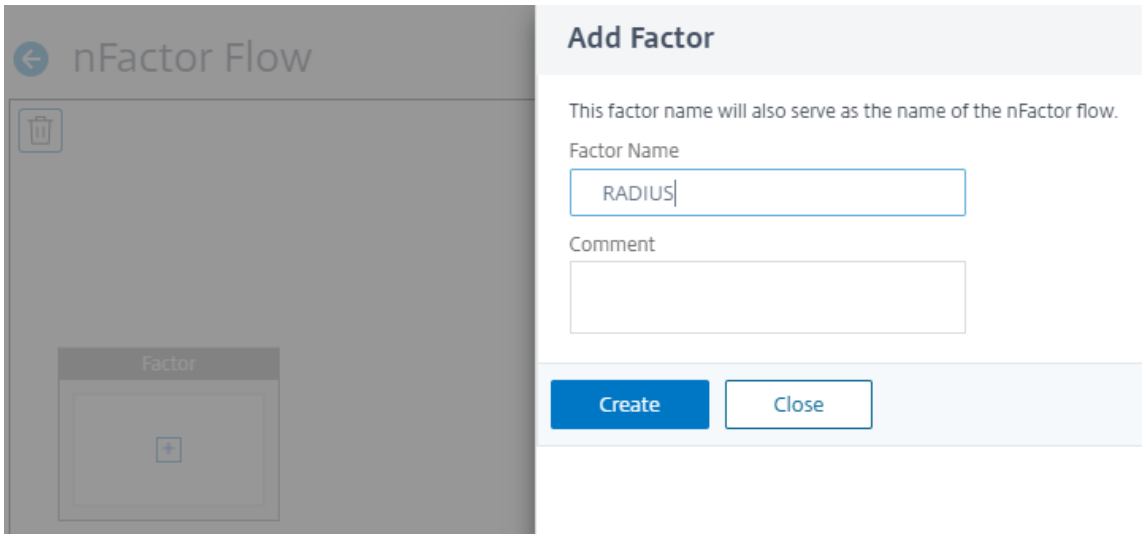
> 注
>
> - 最初に分岐するユースケースがある場合は、2 つのフローを作成して別々にバインドするか、最初のフローを分岐として 1 つのフローを作成して、仮想サーバーにバインドすることができます。
> - 複数のブロックがあり、nFactor Flow 画面でフロー全体を表示するには、ビジュアライザーをクリックして、フローを左端にドラッグします。
> - nFactorFlows ページのみを使用して nFactorFlows を変更することをお勧めします。

nFactor ビジュアライザーを使用して **nFactor** を構成するには

> 注**:**
>
> 次の nFactor 構成は、ユースケース 1 シナリオ構成の実現に役立つ簡単な例です。

1. **[**セキュリティ**] > [AAA] — [**アプリケーショントラフィック**] > [nFactor** ビジュアライザー**] > [nFactor** フロー] に移動します。
2. [追加] をクリックします。
3. [**nFactor** フロー] ページで、[**+**] をクリックして、フローの最初のファクタを追加します。最初のファクターは、この nFactor フローの識別子としても機能します。

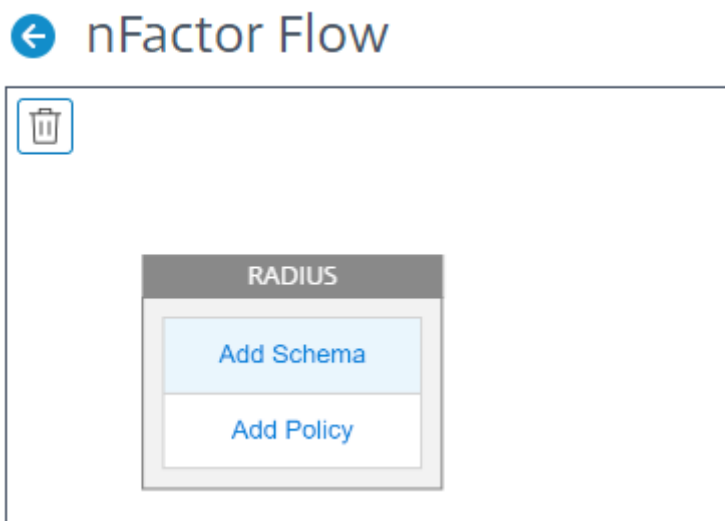
4. 因子名を入力し、「作成」をクリックします。



係数名が nFactor Flow ページの係数ブロックに表示されます。

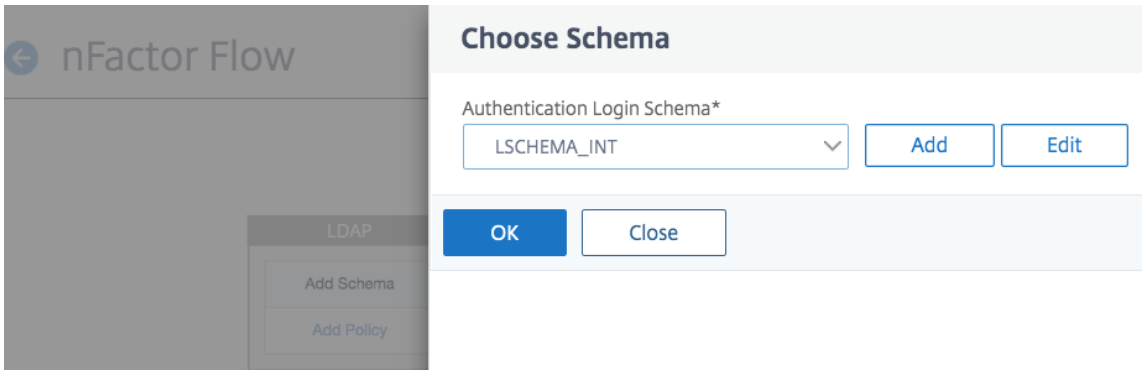
> 注
>
> : `__root`や`__<flow_name>`などのポリシーラベル名をサフィックスに使用せず、`_db_`をプレフィックスに使用しないでください。これは、nFactor フローで作成されるファクター名として使用されます。

5. RADIUS ファクタを作成したら、[スキーマの追加] と [ポリシーの追加] を作成する必要があります。

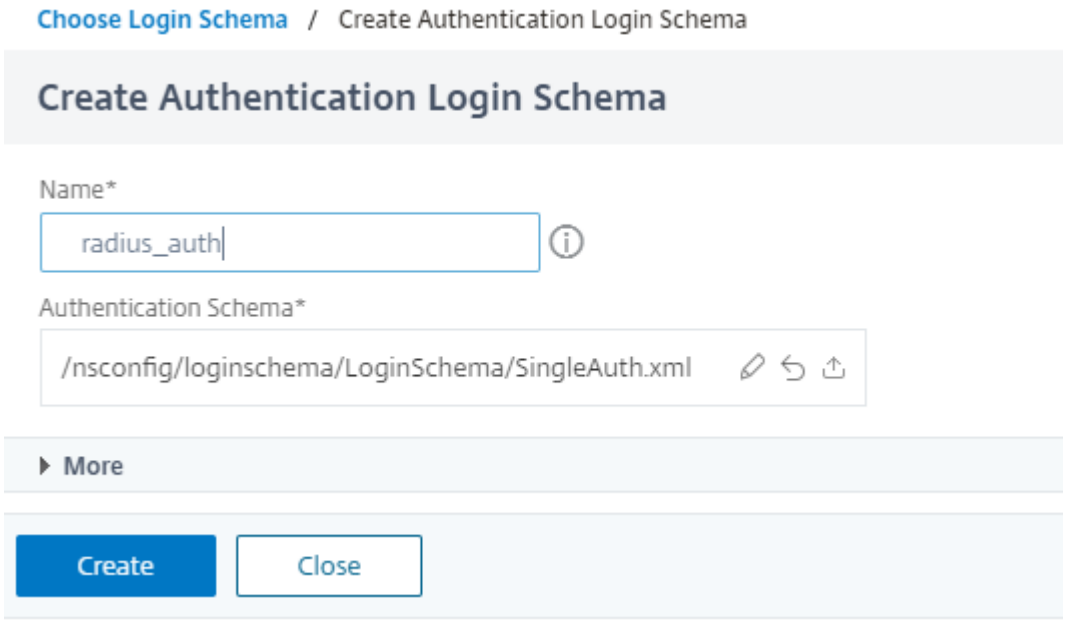


> 注
>
> 詳細については、nFactor の概念、エンティティ、用語を参照してください。

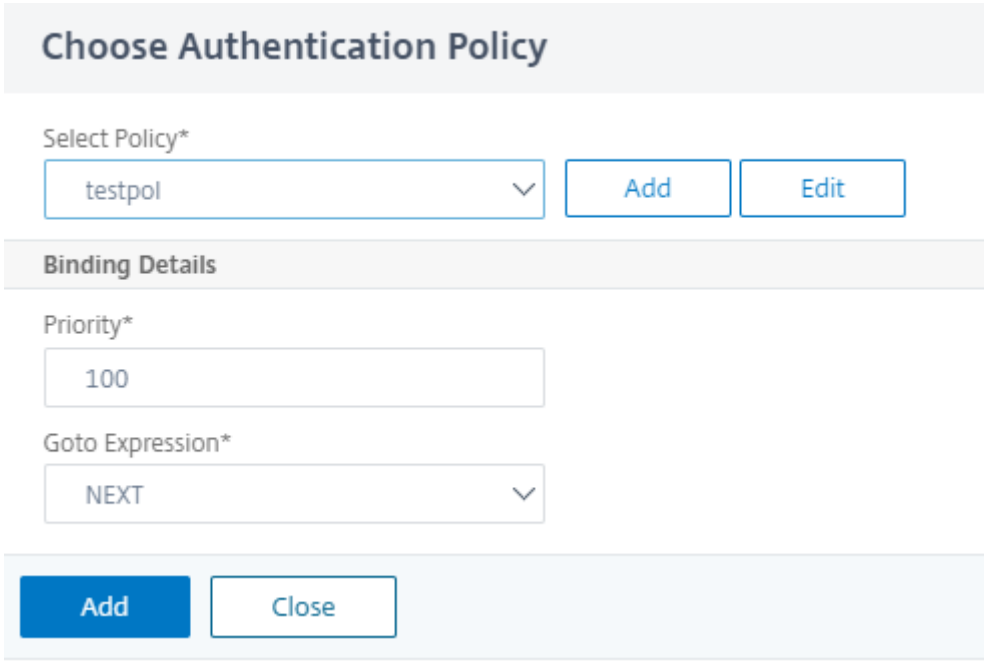
6. [スキーマの追加] をクリックします。新しいログインスキーマを追加するか、[認証ログインスキーマ] リストから既存のログインスキーマを選択します。



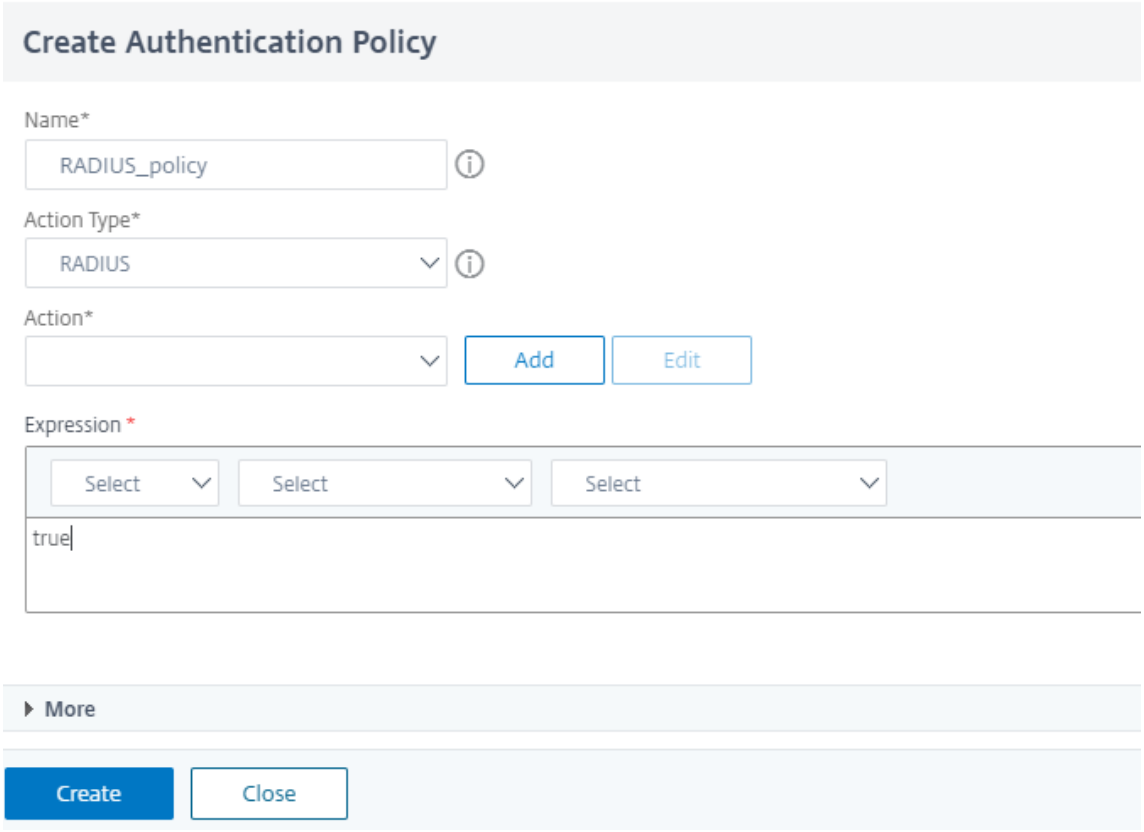
7. ログイン・スキーマを作成するには、「追加」をクリックし、「認証ログイン・スキーマの作成」ページでスキーマの名前を入力します。「編集」（鉛筆アイコン）をクリックして、リストから「ログインスキーマファイル」を選択します。



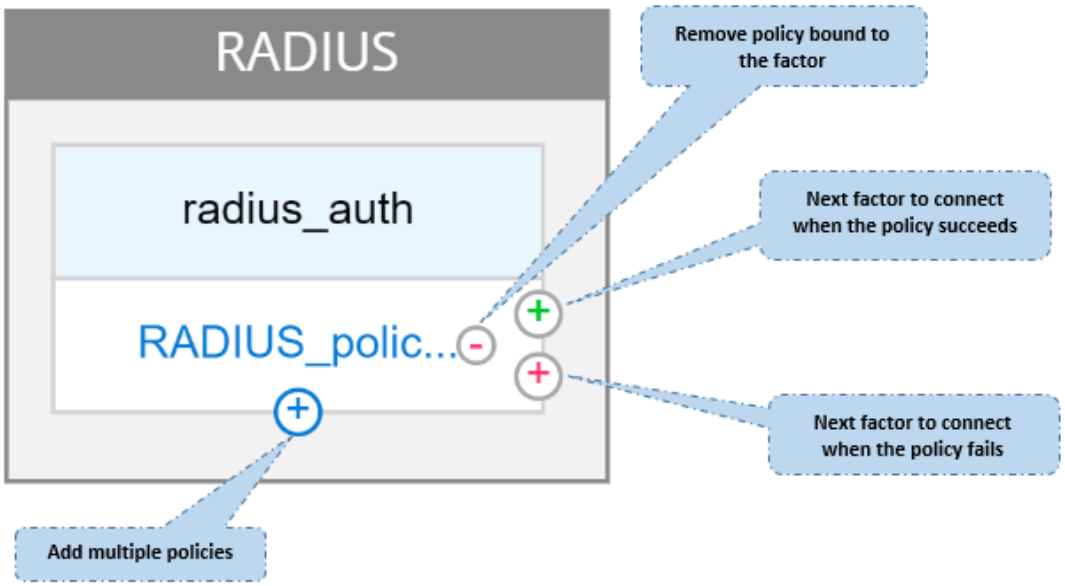
8. [ポリシーの追加] をクリックします。認証ポリシーを作成するか、既存の認証ポリシーを選択できます。



9. 新しいポリシーを作成するには、[追加] をクリックし、[認証ポリシーの作成] ページでポリシーの名前を入力し、[作成] をクリックします。



10. ファクタにログインスキーマとポリシーを追加すると、次の図に示すように、ログインスキーマとポリシーがビジュアライザのファクタに表示されます。任意の要素について、複数のポリシーを追加し、各ポリシーの成功と失敗の次の要素を定義できます。要素の一部であるポリシーを削除することもできます。



11. フローを作成したら、nFactor フローを認証仮想サーバにバインドできます。

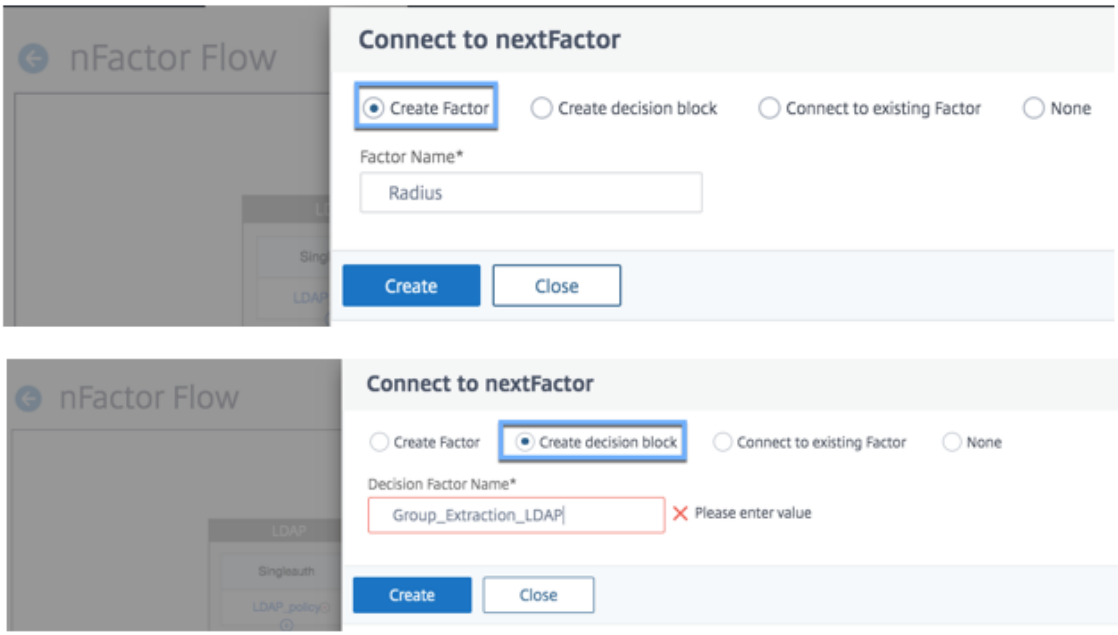
次の要因の追加

次の要素を追加するには、要件に従って次のオプションのいずれかを選択できます。

- **[係数の作成]:** 係数を作成します。フローで作成される各ファクターは、そのフローに対して排他的です。
- 決定ブロックを作成します。分岐アウト係数として機能するデシジョンブロックを作成します。デシジョンブロックにログインスキーマを追加することはできません。ビジュアライザーでは、デシジョンブロックに対して NO_AUTHN ポリシーのみを構成できます。

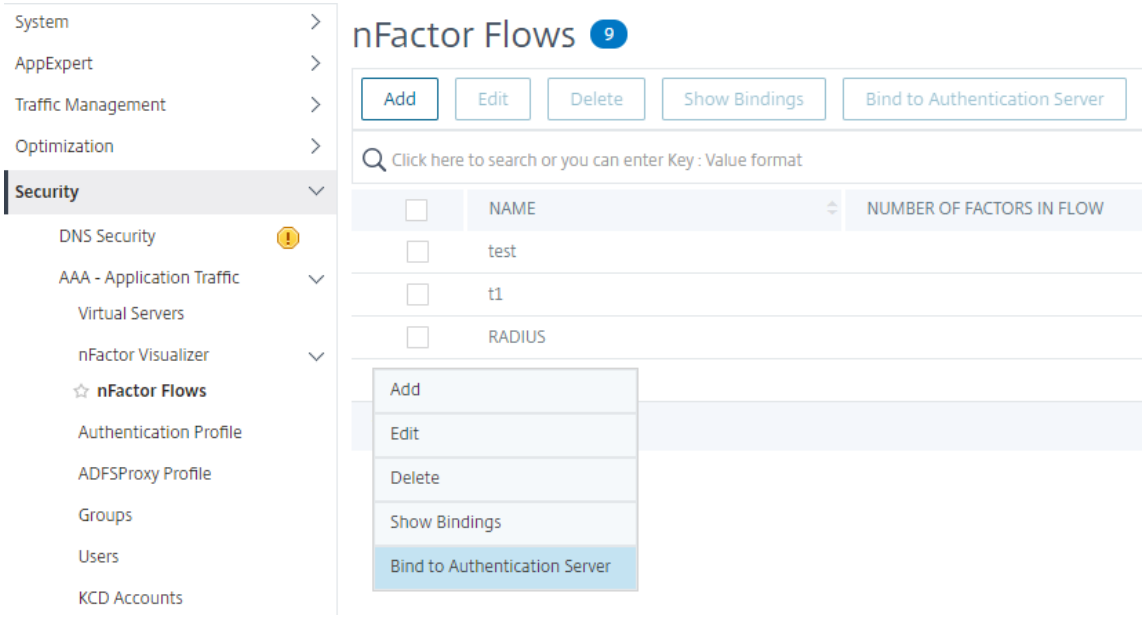
  > 注
  >
  > 決定ブロックは、Citrix ADC GUI を介してのみ追加または編集できます。CLI コマンドから決定ブロックを構成するオプションはありません。

- 既存の係数に接続します。次の因子として既存の因子を選択します。既存のリストに表示されるすべての要素は、そのフロー専用に作成されます。
- なし。既存の接続を削除します。



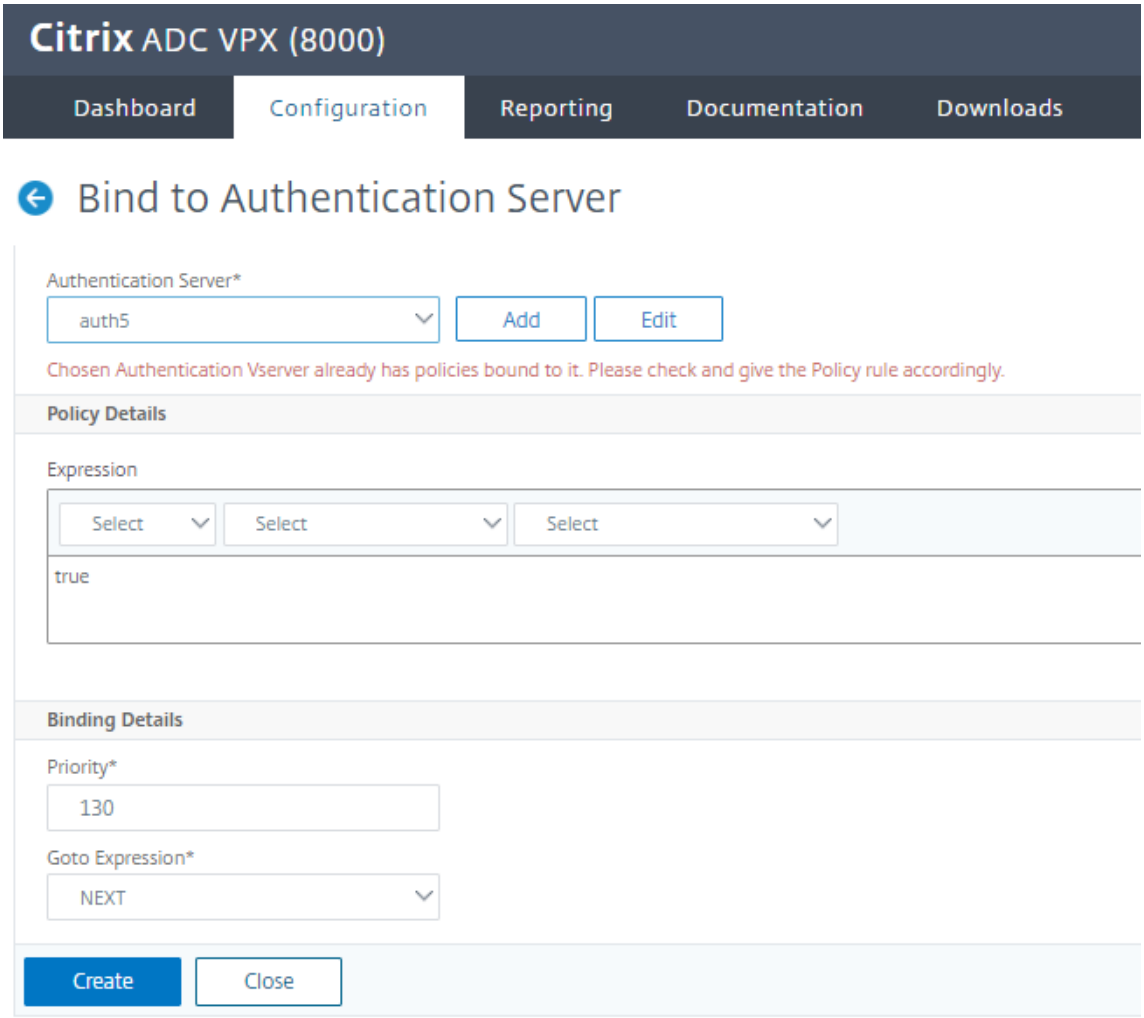
nFactor フローを認証サーバにバインドするには

1. [**nFactor** フロー] ページで、認証仮想サーバーにバインドする nFactor フローを選択します。
2. ハンバーガーアイコンをクリックして [認証サーバーにバインド] オプションを選択するか、詳細ペインで [認証サーバーにバインド] をクリックします。

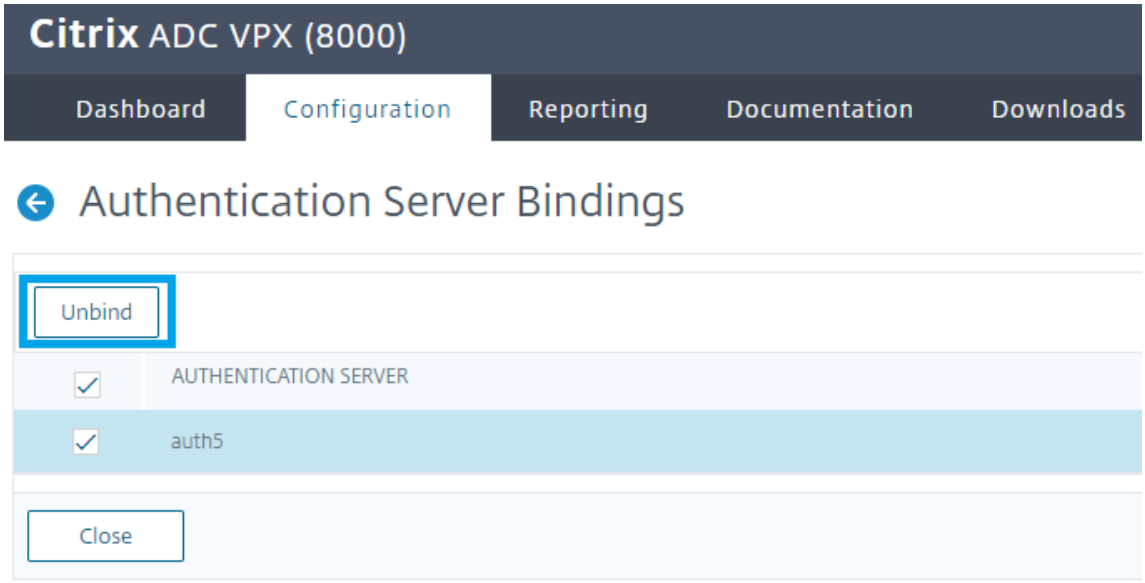


3. [認証サーバにバインド] ページでは、次の操作を実行できます。

   - 認証仮想サーバーを追加するには、[追加] をクリックします。
   - リストから既存の認証サーバーを選択するには、[認証サーバー] フィールドをクリックします。



4. ハンバーガーのアイコンから「バインディングを表示」をクリックして、バインディングを表示します。
5. 特定の nFactor フローから認証サーバをバインド解除するには、次の手順を実行します。
   - **nFactor** がページをフローに、ハンバーガーのアイコンから バインドの表示] をクリックします。
   - [認証サーバのバインド] ページで、バインドを解除する認証サーバを選択し、[バインド解除] をクリックします。[閉じる] をクリックします。



nFactor 認証の詳細については、次のトピックを参照してください。

- 概念: 多要素 (nFactor) 認証。
- ワークフロー: nFactor 認証の仕組み。
- 構成:nFactor 認証の設定。

nFactor ビジュアライザーの機能強化

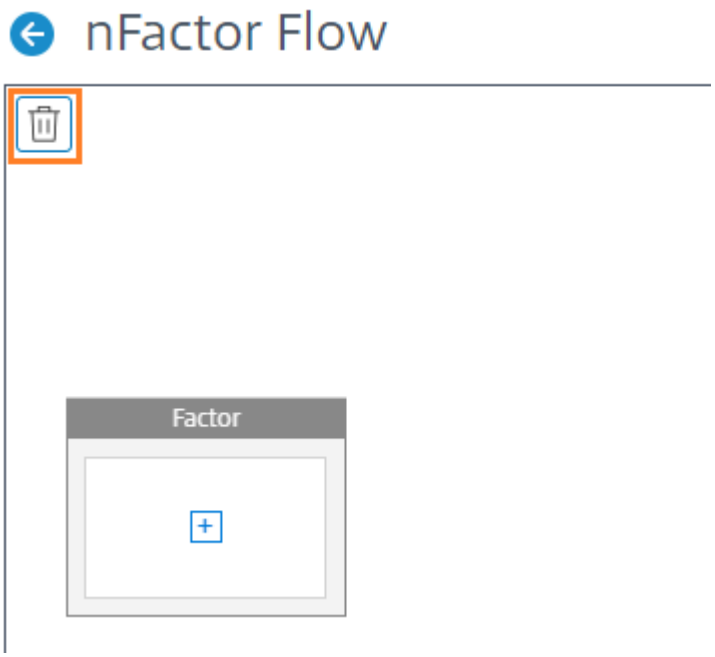
Citrix ADC リリース 13.0 ビルド 41.20 以降では、nFactor ビジュアライザーで次の機能が強化されています。

- 管理者は、作成した要素をゴミ箱アイコンに移動できます。
- 仮想サーバーの認証ページで nFactor フローを表示します。

ごみ箱アイコン。管理者は、接続のないノードのみを削除できます。ただし、要素をごみ箱に移動した場合、基になるポリシーまたは要素に対して作成されたスキーマは削除されません。

ごみ箱アイコンを表示するには、

1. **[セキュリティ] > [AAA] — [アプリケーショントラフィック] > [nFactor** ビジュアライザー**] > [nFactor** フロー] に移動します。

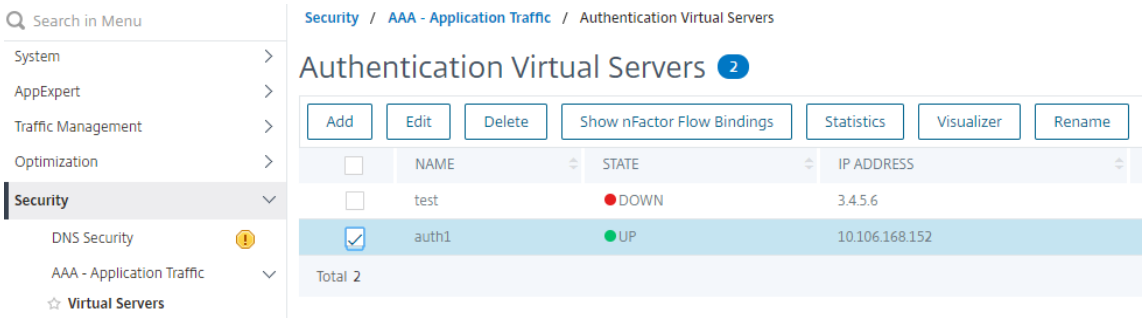


2. ファクタを削除するには、ファクタブロックをクリックしてゴミ箱にドラッグします。

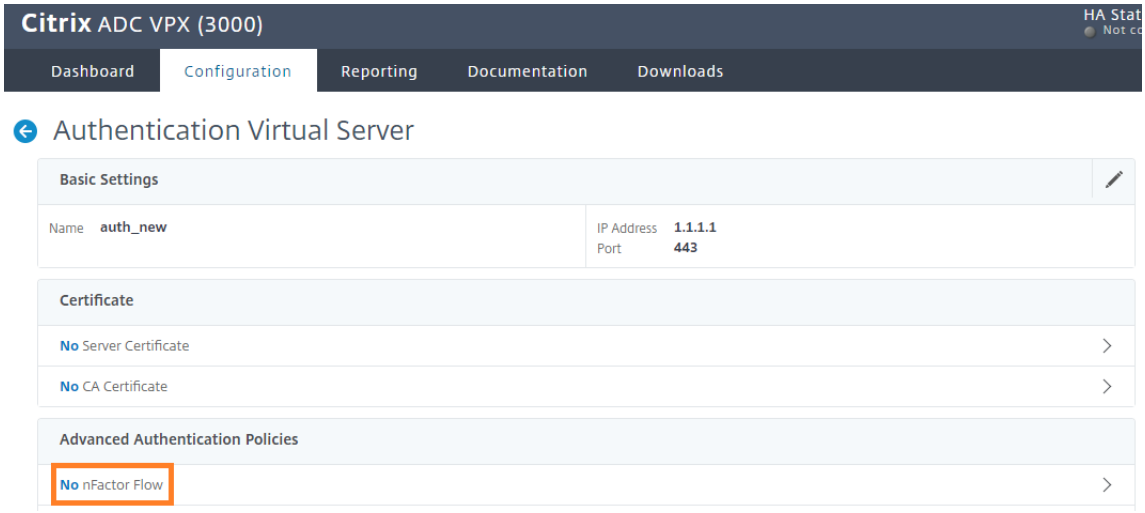
認証仮想サーバからの **nFactor** フローを表示します。管理者は、作成された nFactor フローを Authentication VirtualServer ページから表示することもできます。

仮想サーバーの認証ページから nFactor フローを表示するには、

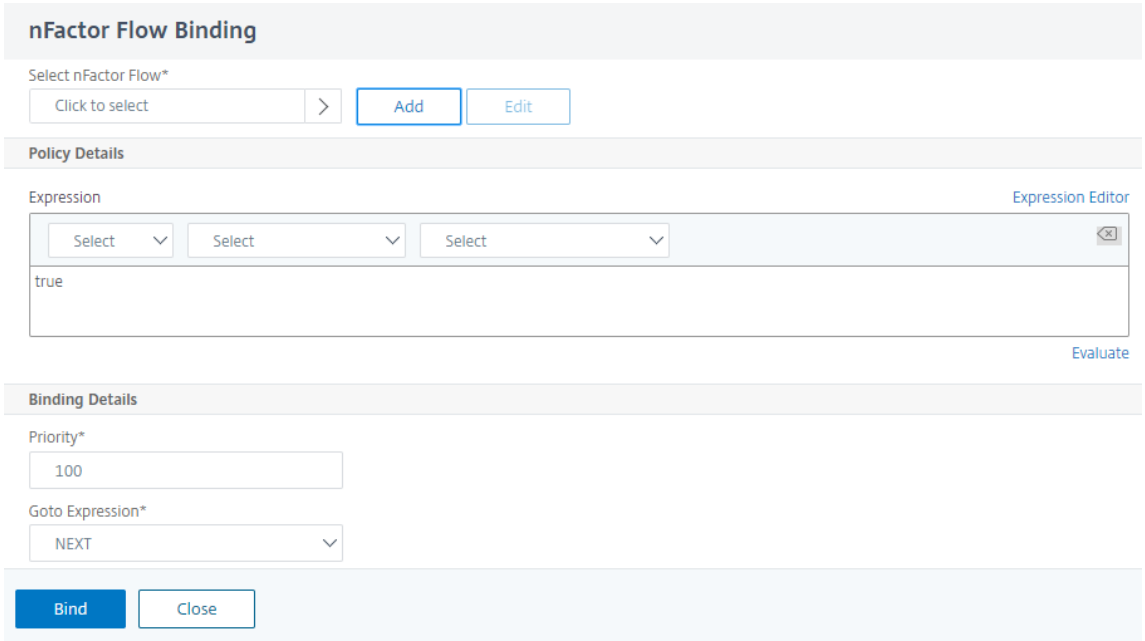
1. **Security > AAA – Application Traffic > Virtual Servers** に移動します。[仮想サーバーの認証] ページでは、次の手順を実行します。
   - 認証仮想サーバーを追加するには、[追加] をクリックします。
   - 既存の認証仮想サーバーを編集するには、詳細ウィンドウ領域で [編集] をクリックします。



2. [認証仮想サーバー] ページでは、[高度な認証ポリシー] の **nFactor Flow** オプションを表示できます。



3. 仮想サーバーに nFactor フローがバインドされていない場合は、[高度な認証ポリシー] セクションの [**nFactor** フローなし] をクリックして、新しい nFactor フローを追加するか、リストから既存の nFactor フローを選択します。



nFactor 拡張性

April 8, 2022

nFactor 認証フレームワークは、カスタマイズを追加できる柔軟性を提供し、ログオンインターフェイスをより直感的にし、リッチなユーザーエクスペリエンスを実現します。カスタムログインラベル、カスタムログイン認証情報、UI 表示のカスタマイズなどを追加できます。

nFactor を使用すると、各ファクターに独自のログオン画面を持つことができます。各ログオン画面には、以前の要素からの情報や、他の要素では見えない情報を表示できます。たとえば、最後の要素は、ユーザーが指示を読み、[続行] をクリックする情報ページです。

nFactor 以前は、カスタムログインページには制限があり、カスタマイズとサポートが必要でした。tmindex.html を置き換えたり、書き換えルールを適用したりして、その動作の一部を変更することができました。しかし、基盤となる機能を実現することは不可能でした。

このトピックでは、次の nFactor 関連のカスタマイズについて詳しく説明します。

- ログインラベルをカスタマイズする
- UI をカスタマイズして画像を表示する
- Citrix ADC nFactor ログオンフォームのカスタマイズ

仮定

nFactor、シェルコマンド、XML、およびテキストエディタに精通している。

前提条件

- このトピックで説明するカスタマイズは、Citrix ADC で rfWeb UI テーマ（またはテーマベース）が構成されている場合にのみ可能です。
- 認証ポリシーは、認証、承認、および監査仮想サーバーにバインドする必要があります。バインドしないと、フローが意図したとおりに機能しません。
- nFactor に関連する次のアイテムがあります
  - XML スキーマ
  - JavaScript
  - 認証アクション
  - 認証仮想サーバー
  - Citrix ADC バージョン 11.1 以降

ログオンラベルをカスタマイズする

ログオンラベルをカスタマイズするには、次のものが必要です。

- ログオンページの外観を記述する XML スキーマ。
- レンダリングプロセスの変更に使用される JavaScript を含む script.js ファイル。

> 注:
>
> script.js ファイルはディレクトリ/var/netscaler/logon/themes/<custom_theme>/にあります。

機能

JavaScript は XML ファイルを解析し、<Requirements> タグ内の各項目をレンダリングします。各要素は HTML フォームの 1 行に対応します。たとえば、ログインフィールドは行、パスワードフィールドは別の行、ログオンボタンも行です。新しい行を導入するには、StoreFront SDK を使用して XML スキーマファイルで新しい行を指定する必要があります。StoreFront SDK では、XML スキーマのあるログオンページで<Requirement>タグを使用し、そのタグ上の要素を定義できます。これらの要素により、JavaScript を使用して、必要な HTML 要素が何であれ、その空間に導入することができます。この場合、HTML 形式のテキストを含む行が作成されます。

使用できる XML は、次のとおりです。

```
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<!--NeedCopy-->
```

<Requirement>: ログオンページで提供される領域。クレデンシャルはスペースを埋め、他の部分はエンジンを正しい情報にルーティングします。この場合は、nsg-custom-credと入力します。これはプレーンテキストとして定義され、ラベルはその本文に対して定義されます。
要件 XML は JavaScript コードとペアになっており、必要な結果が得られます。

```
// Custom Label Handler for Self Service Links
CTXS.ExtensionAPI.addCustomAuthLabelHandler({

getLabelTypeName: function () {
  return ”nsg-custom-label”;  }
 ,
getLabelTypeMarkup: function (requirements) {

return $(”< Enter your HTML code here>”);
 }
 ,
// Instruction to parse the label as if it was a standard type
parseAsType: function () {

return ”plain”;
```



```
 }

 }
 );
//Custom Credential Handler for Self Service Links
CTXS.ExtensionAPI.addCustomCredentialHandler({

getCredentialTypeName: function () {
  return "nsg-custom-cred";  }
 ,
getCredentialTypeMarkup: function (requirements) {

return $("<div/>");
 }
 ,
 }
 );
<!--NeedCopy-->
```

> 重要:
>
> HTML コードを追加するときは、戻り値が HTML タグで始まることを確認してください。

XML の部分はログオンページに何を表示するかを示し、JavaScript コードは実際のテキストを提供します。認証情報ハンドラがスペースを開き、ラベルがスペースを埋めます。これで、すべての認証トラフィックが書き換えや応答側から見えなくなるため、ページのルックアンドフィールを変更できます。
ログインラベルをカスタマイズする設定

1. rfWeb に基づいてテーマを作成してバインドします。

   ```
   add vpn portaltheme RfWebUI_MOD -basetheme RfWebUI

   bind vpn vserver TESTAAA  - portaltheme RfWebUI_MOD
   <!--NeedCopy-->
   ```

   テーマに基づいたファイルのパスは、/var/NetScaler/logon/themes/rfwebui_MOD というディレクトリにあります。

2. script.js ファイルの最後に次のスニペットを追加します。

   > 注:
   >
   > 前の行を正しいファイルに含めなかったり、JavaScript 関数を含めなかったりすると、XML が読み込まれなくなります。このエラーは、ブラウザの開発者コンソールで「Undefined Type nsg-custom

> -cred」というテキストでのみ表示されます。

```
// Custom Label Handler for Self Service Links
CTXS.ExtensionAPI.addCustomAuthLabelHandler({

getLabelTypeName: function () {
  return "nsg-custom-label";  }
 ,
getLabelTypeMarkup: function (requirements) {

return $("<a href="https://identity.test.com/identity/faces/
   register" style="font-size: 16px;" style="text-align: center;">
   Self Registration</a><br><a href="https://identity.test.com/
   identity/faces/forgotpassword" style="font-size: 16px;" style="
   text-align: center;">Forgot Password</a><br><a href="https://
   identity.test.com/identity/faces/forgotuserlogin" style="font-
   size: 16px;" style="text-align: center;">Forgot User Login</a
   >");
 }
 ,
// Instruction to parse the label as if it was a standard type
parseAsType: function () {

return "plain";
 }

 }
 );
//Custom Credential Handler for Self Service Links
CTXS.ExtensionAPI.addCustomCredentialHandler({

getCredentialTypeName: function () {
  return "nsg-custom-cred";  }
 ,
getCredentialTypeMarkup: function (requirements) {

return $("<div/>");
 }
 ,
 }
 );
<!--NeedCopy-->
```


> 重要:
>
> HTML コードを追加するときは、戻り値が HTML タグで始まることを確認してください。

この例で使用されるログインスキーマ

```
<?xml version="1.0" encoding="utf-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response/1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext/>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/Citrix/Authentication/ExplicitForms/CancelAuthenticate</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement>
<Credential>
<ID>login</ID>
<SaveID>Username</SaveID>
<Type>username</Type>
</Credential>
<Label>
<Text>User name</Text>
<Type>plain</Type>
</Label>
<Input>
<AssistiveText>Please supply either domain\username or user@fully.qualified.domain</AssistiveText>
<Text>
<Secret>false</Secret>
<ReadOnly>false</ReadOnly>
<InitialValue></InitialValue>
<Constraint>.+</Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<ID>passwd</ID>
<SaveID>Password</SaveID>
<Type>password</Type>
```



```
</Credential>
<Label>
<Text>Password:</Text>
<Type>plain</Type>
</Label>
<Input>
<Text>
<Secret>true</Secret>
<ReadOnly>false</ReadOnly>
<InitialValue/>
<Constraint>.+</Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<Requirement>
<Credential>
<ID>loginBtn</ID>
<Type>none</Type>
</Credential>
<Label>
<Type>none</Type>
</Label>
<Input>
<Button>Please Log On</Button>
</Input>
</Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

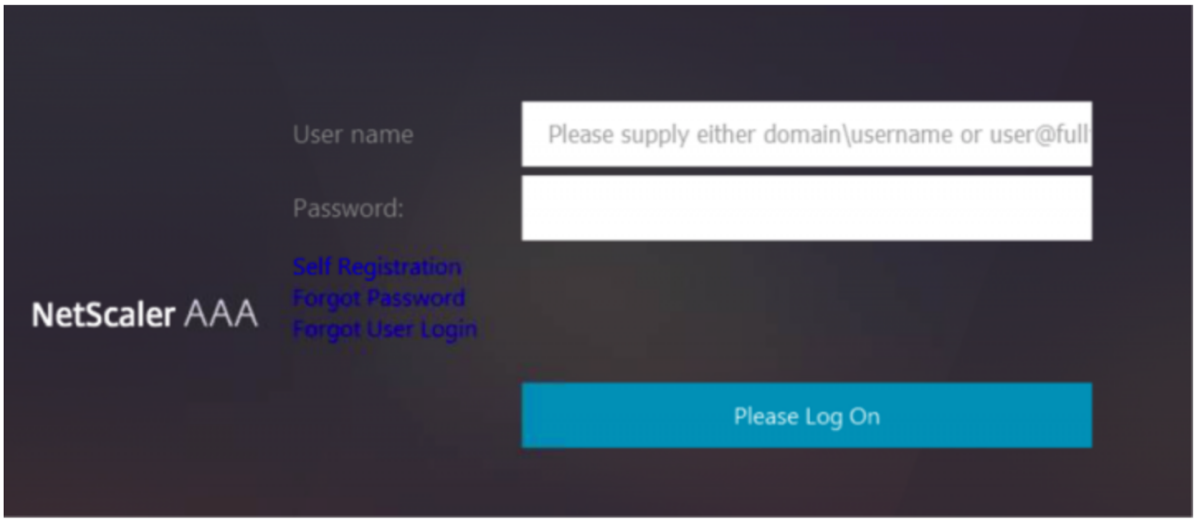
次のコマンドを実行して、カスタムスキーマを config に読み込みます。

```
add authentication loginSchema custom -authenticationSchema custom.xml
```



```
2
3 add authentication loginSchemaPolicy custom -rule true -action custom
4
5 bind authentication vserver AAATEST -policy custom -priority 100 -
     gotoPriorityExpression END
6 <!--NeedCopy-->
```

次の図は、この構成でレンダリングされるログインページを示しています。



UI をカスタマイズして画像を表示する

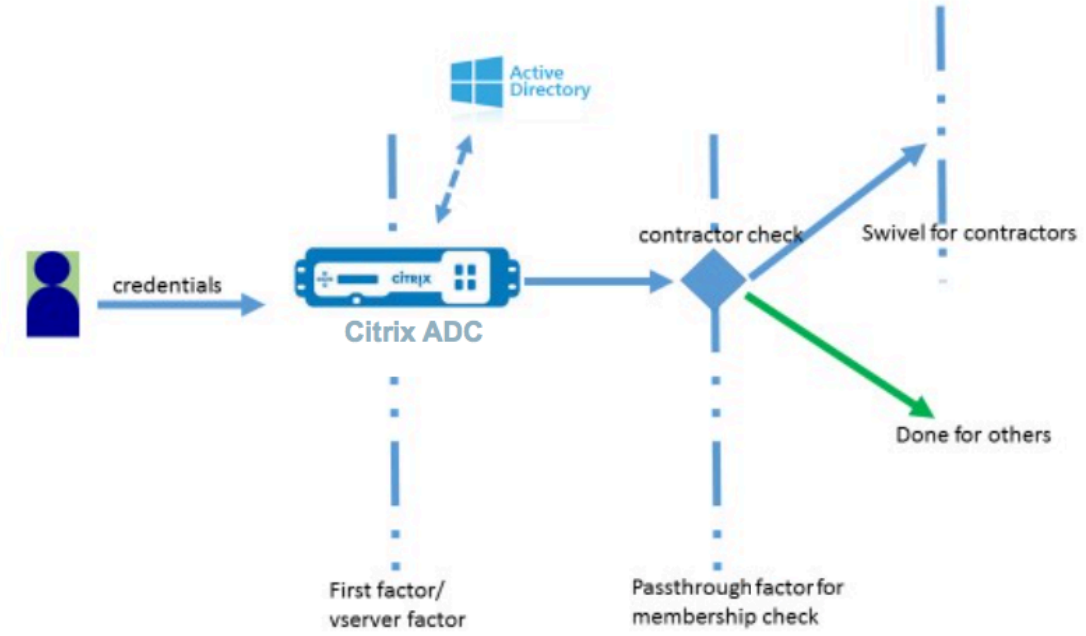
nFactor では、ログインスキーマファイルを使用して表示をカスタマイズできます。組み込みのログインスキーマファイルで提供されるもの以外に、さらにカスタマイズが必要な場合があります。たとえば、UI にハイパーリンクを表示したり、カスタムロジックを記述したりします。これは、ログインスキーマ拡張と対応する JavaScript ファイルで構成される「カスタム認証情報」を使用して実現できます。

ログインスキーマファイルは/nsconfig/loginschema/LoginSchemaディレクトリにあります。

イメージを表示するための UI のカスタマイズでは、「Citrix ADC-Swivel」統合の展開フローが例として使用されます。

このフローには 2 つの要因があります。

- 第 1 要素: ユーザーの AD 資格情報を確認します。
- 第 2 要因: グループメンバーシップに基づいてユーザーログオンを要求します。



このフローでは、すべてのユーザーが第 1 の要素を通過します。2 番目の要素の前に、一部のユーザーを「旋回」係数から除外できるかどうかを確認する疑似要素があります。ユーザーが「回転」係数を必要とする場合、コードを入力するための画像とテキストボックスが表示されます。

解決策

UI をカスタマイズして画像を表示するソリューションには、次の 2 つの部分があります。

- ログインスキーマ拡張。
- ログインスキーマ拡張を処理するカスタムスクリプト。

ログインスキーマ拡張

フォームのレンダリングを制御するために、カスタム 'id'/'credential' がログインスキーマに注入されます。これは、既存のスキーマを再利用し、要件に従って変更することで実現できます。

この例では、テキストフィールド (/nsconfig/loginschema/LoginSchema/OnlyPassword.xml など) が 1 つしかないログインスキーマが考慮されます。
次のスニペットがログインスキーマに追加されます。

```
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><
   Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
```



```
<!--NeedCopy-->
```

このスニペットでは、認証情報の「タイプ」として「swivel_cred」が指定されています。これは組み込みの「認証情報」として認識されないため、UI はこのタイプのハンドラを探し、存在する場合はそれを呼び出します。
この資格情報には、Citrix ADC が動的に入力する式である初期値が送信されます。この例では、スイベルサーバーにユーザー名を通知するために使用されるユーザー名です。常に必要というわけではないかもしれませんし、他のデータで補強することもできます。これらの詳細は、必要に応じて追加する必要があります。

カスタム認証情報を処理する **Javascript**

UI はカスタム認証情報を検出すると、ハンドラーを探します。すべてのカスタム・ハンドラーは、デフォルトのポータル・テーマ用に /var/netscaler/logon/LogonPoint/custom/script.js に記述されます。

カスタムポータルテーマの場合、script.js はディレクトリ/var/netscaler/logon/themes/<custom_theme>/にあります。

カスタム認証情報のマークアップをレンダリングするために、次のスクリプトが追加されました。

```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by the server
    getCredentialTypeName: function () {
  return "swivel_cred";  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $("<div></div>");
        var image = $("<img/>");
        var username = requirements.input.text.initialValue; //Get the secret from the response
        image.attr({

            "style" : "width:200px;height:200px;",
            "id" : "qrcodeimg",
            "src" : "https://myswivelserver.citrix.com:8443/pinsafe/SCImage?username=" + username
         }
 );
        div.append(image);
        return div;
     }
```



```
23
24  }
25  );
26 <!--NeedCopy-->
```

このスニペットは、「wivel_cred」のマークアップを処理するためのものです。強調表示された認証情報名は、ログインスキーマ拡張で以前に指定された「タイプ」と一致する必要があります。
マークアップを生成するには、ソースがスイベルサーバーを指すイメージを追加する必要があります。これが完了すると、UI は指定した場所からイメージをロードします。このログインスキーマにもテキストボックスがあるため、UI はそのテキストボックスをレンダリングします。

> 注:
>
> 管理者は、イメージ要素の「スタイル」を変更して、イメージのサイズを変更できます。現在、200x200 ピクセルに設定されています。

UI をカスタマイズして画像を表示するための設定

nFactor の構成は、ボトムアップで構築する方が適切です。これは、前の因子に 'nextFactor' を指定しようとすると、後続の因子の名前が必要になるため、最後の因子が先になります。

旋回係数構成:

```
1 add loginschema swivel_image  -authenticationSchema /nsconfig/
     loginschema/SwivelImage.xml
2
3 add authentication policylabel SwivelFactor  -loginSchema swivel_image
4
5 bind authentication policylabel SwivelFactor  -policy <policy-to-check-
     swivel-image> -priority 10
6 <!--NeedCopy-->
```

> 注:
>
> この例で使用されているログインスキーマから SwivelImage.xml をダウンロードします。

グループチェック設定の疑似係数:

```
1 add authentication policylabel GroupCheckFactor
2
3 add authentication policy contractors_auth_policy  -rule  'http.req.
     user.is_member_of( "contractors" )'  -action NO_AUTHN
4
```



```
add authentication policy not_contractors _auth_policy - rule true -
    action NO_AUTHN

bind authentication policylabel GroupCheckFactor - policy
    contractors_auth_policy - pri 10 - nextFactor SwivelFactor

bind authentication policylabel GroupCheckFactor - policy
    not_contractors_auth_policy - pri 20
<!--NeedCopy-->
```

Active Directory ログインの第 **1** 要因**:**

```
add ldapAction <>

add authentication policy user_login_auth_policy - rule true - action
    <>

bind authentication vserver <> -policy user_login_auth_policy - pri 10
    - nextFactor GroupCheckFactor
<!--NeedCopy-->
```

この構成では、暗黙的/疑似的要素の 3 つの要素が指定されています。

この例で使用されるログインスキーマ

次に、スイベル認証情報とテキストボックスを持つスキーマの例を示します。

> 注**:**
>
> Web ブラウザ用にデータをコピーする場合、引用符の表示が異なる場合があります。ファイルに保存する前に、メモ帳などのエディターでデータをコピーします。

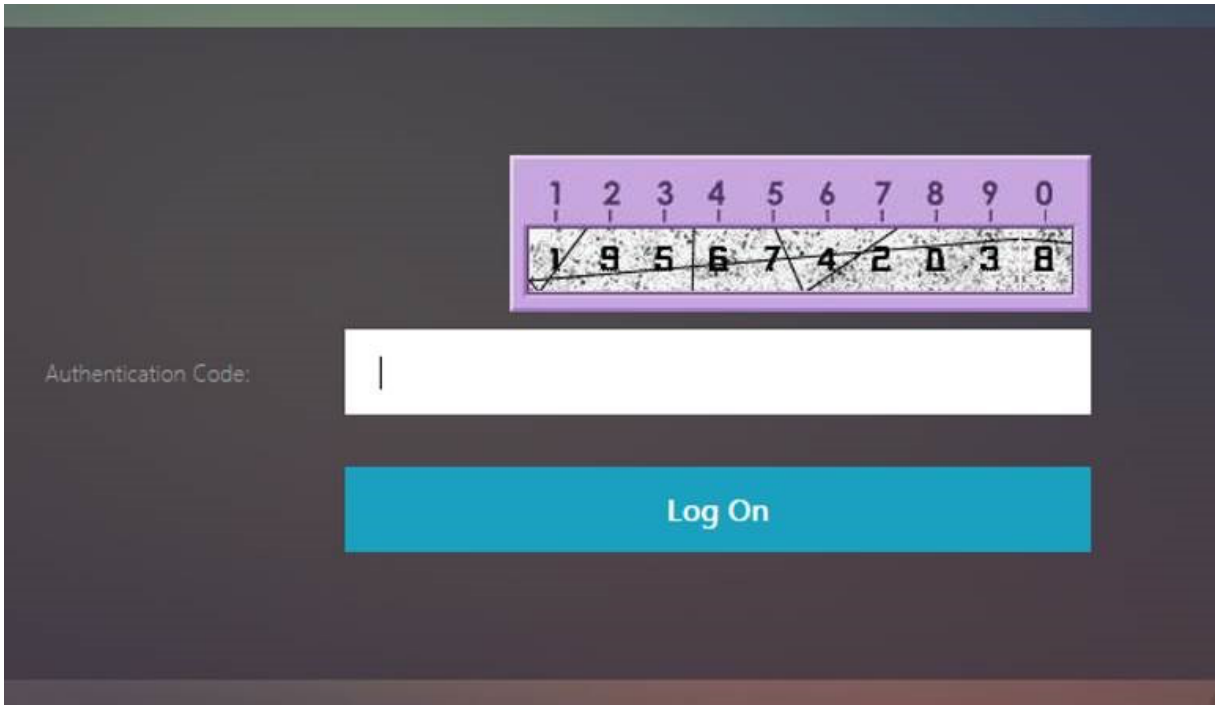
```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response
    /1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
```



```
<Requirements>
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><
   Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password
   </SaveID><Type>password</Type></Credential><Label><Text>Password:</
   Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><
   ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint
   >.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>
   Hello ${
 http.req.user.name }
 , Please enter passcode from above image.</Text><Type>confirmation</
    Type></Label><Input /></Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
   </Type></Credential><Label><Text>Remember my password</Text><Type>
   plain</Type></Label><Input><CheckBox><InitialValue>false</
   InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
   ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
   Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

出力

設定が完了すると、次のイメージが表示されます。



> 注:
>
> 画像の高さと配置は JavaScript で変更できます。

Citrix ADC nFactor ログオンフォームをカスタマイズしてフィールドを表示または非表示にする

Citrix Gateway の rfWeb UI では、さまざまなカスタマイズが可能です。この機能を nFactor 認証フレームワークと組み合わせると、既存のワークフローを損なうことなく複雑なフローを構成できます。

この例では、[Logon Type] リストから OAuth と LDAP の 2 つの認証オプションを使用できます。フォームを初めて読み込むと、ユーザー名とパスワードのフィールド (LDAP が先に表示されます) が表示されます。OAuth を選択した場合、OAuth は認証をサードパーティサーバーにオフロードすることを暗示するため、すべてのフィールドが非表示になります。これにより、管理者はユーザーの利便性に応じて直感的なワークフローを構成できます。

> 注:
>
> - [Logon Type] リストの値は、スクリプトファイルに簡単に変更を加えるだけで変更できます。
> - このセクションでは、フローの UI 部分のみについて説明します。認証の実行時の処理については、この記事では説明しません。認証設定については、nFactor のドキュメントを参照することをお勧めします。

nFactor ログオンフォームをカスタマイズする方法

nFactor ログオンフォームのカスタマイズは 2 つの部分に分類できます

- 適切なログインスキーマを UI に送信する
- ログインスキーマとユーザー選択を解釈するハンドラーの作成

正しいログインスキーマを **UI** に送信する

この例では、単純なクレーム/要件がログインスキーマで送信されます。

このため、SingleAuth.xml ファイルは変更されます。SingleAuth.xml は Citrix ADC ファームウェアに同梱されており、/nsconfig/loginschema/LoginSchema ディレクトリにあります。

ログインスキーマを送信する手順**:**

1. SSH 経由でログインし、シェルにドロップします (「shell」と入力します)。
2. SingleAuth.xml を別のファイルにコピーして変更します。

   > 注:
   >
   > 宛先フォルダーは、デフォルトの Citrix ADC ログインスキーマフォルダーとは異なります。
   >
   > cp /nsconfig/loginschema/LoginSchema/SingleAuth.xml /nsconfig/loginschema/SingleAuthDynamic.xml

3. SingleAuthDynamic.xml に次のクレームを追加します。

```
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></Label></Requirement>
<!--NeedCopy-->
```

4. このログインスキーマを送信して最初のフォームを読み込むように Citrix ADC を構成します。

```
add loginschema single_auth_dynamic -authenticationSchema SingleAuthDynamic.xml

add loginschemaPolicy single_auth_dynamic -rule true -action single_auth_dynamic

bind authentication vserver aaa_nfactor -policy single_auth_dynamic -pri 10
<!--NeedCopy-->
```

フォームを読み込み、ユーザーイベントを処理するためのスクリプト変更

管理者がログオンフォームの表示をカスタマイズできるように JavaScript を変更できます。この例では、LDAP が選択されている場合はユーザー名とパスワードのフィールドが表示され、OAuth を選択すると非表示になります。また、管理者はパスワードのみを非表示にすることもできます。

管理者は、「/var/NetScaler/ログオン/ログオンポイント/カスタム」ディレクトリにある「script.js」に次のスニペットを追加する必要があります。

> 注:
>
> このディレクトリはグローバルディレクトリなので、ポータル・テーマを作成し、”/var/netscaler/logon/themes/<THEME_NAME>”でそのフォルダー内の「script.js」ファイルを編集します。

```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by the
       server
    getCredentialTypeName: function () {
  return ”nsg_dropdown”;  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $(”<div></div>”);
        var select = $(”<select name='nsg_dropdown'></select>”).attr(”
           id”, ”nsg_dropdown”);

        var rsa = $(”<option></option>”).attr(”selected”, ”selected”).
           text(”LDAP”).val(”LDAP”);
        var OAuthID = $(”<option></option>”).text(”OAuth”).val(”OAuth”)
           ;
        select.append(rsa, OAuthID);

        select.change(function(e) {

            var value = $(this).val();
            var ldapPwd = $($(”.credentialform”).find(”.
               CredentialTypepassword”)[0]);
            var ldapUname = $($(”.credentialform”).find(”.
               CredentialTypeusername”));
            if(value == ”OAuth”) {

                if (ldapPwd.length)
                        ldapPwd.hide();
                if (ldapUname.length)
                        ldapUname.hide();
             }
  else if(value == ”LDAP”) {

```



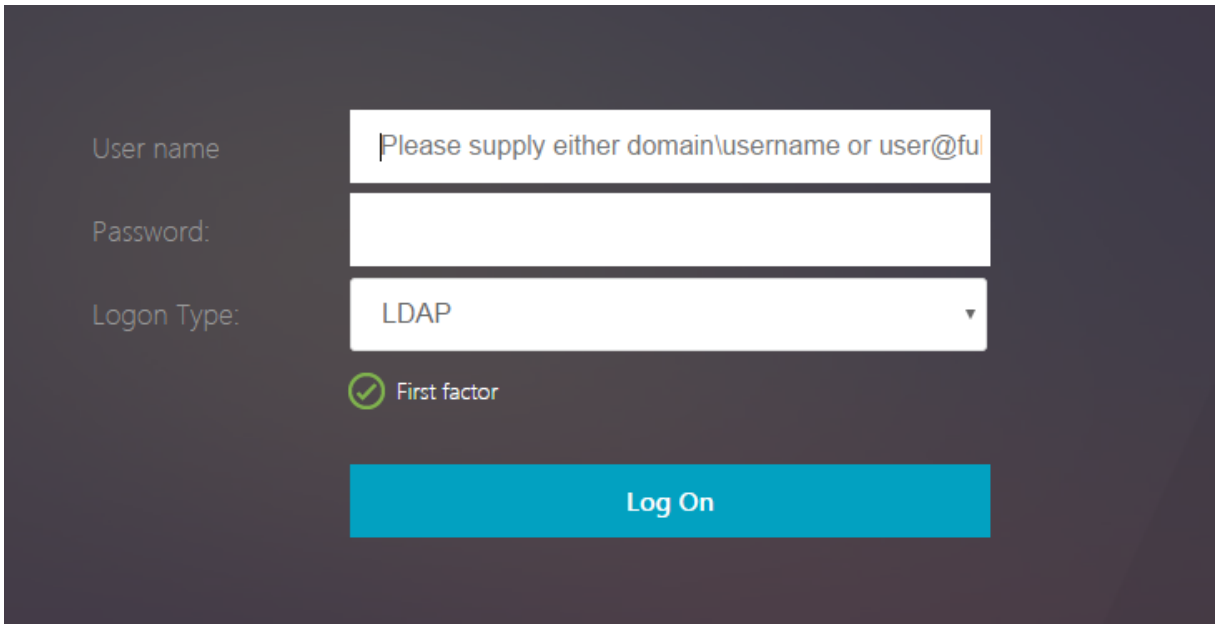
```
                if (ldapPwd.length)
                        ldapPwd.show();
                if (ldapUname.length)
                        ldapUname.show();
            }

          }
);
       div.append(select);
        return div;
     }

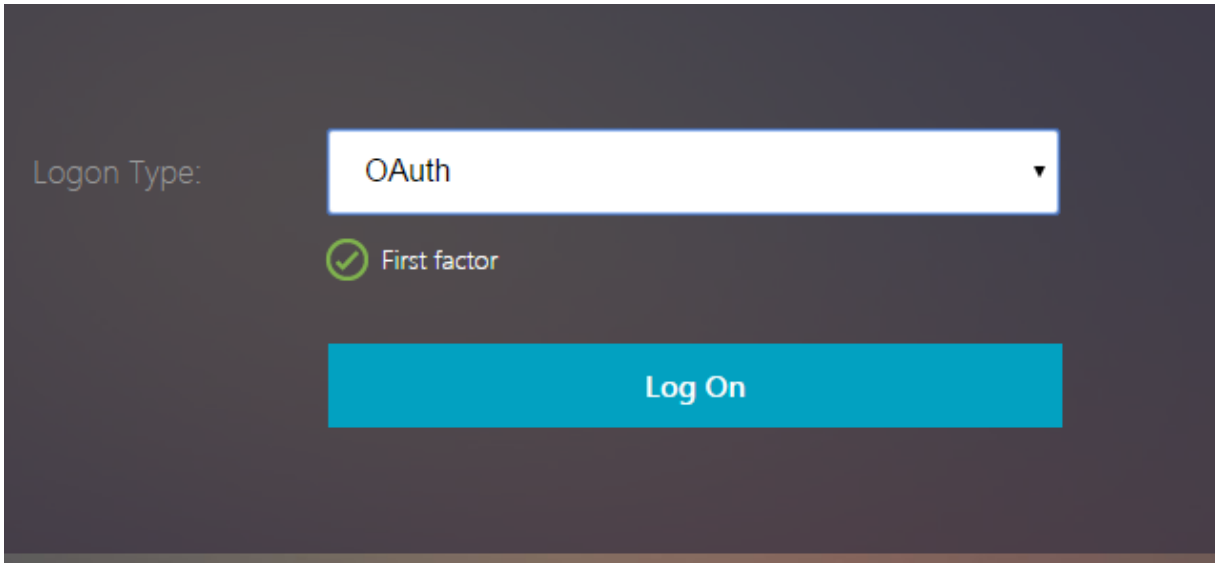
}
);
<!--NeedCopy-->
```

エンドユーザーエクスペリエンス

エンドユーザーが初めてログオンページを読み込むと、次の画面が表示されます。



［ログオンの種類］で［**OAuth**］が選択されている場合、［ユーザー名］および［パスワード］フィールドは非表示になります。



LDAP を選択すると、ユーザー名とパスワードが表示されます。これにより、ユーザーの選択に基づいてログオンページを動的に読み込むことができます。

この例で使用されるログインスキーマ

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response/1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</SaveID><Type>username</Type></Credential><Label><Text>User name</Text><Type>plain</Type></Label><Input><AssistiveText>Please supply either domain\username or user@fully.qualified.domain</AssistiveText><Text><Secret>false</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password</SaveID><Type>password</Type></Credential><Label><Text>Password:</Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
```



```
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type
    ></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></
    Label></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>
    First factor</Text><Type>confirmation</Type></Label><Input /></
    Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
    </Type></Credential><Label><Text>Remember my password</Text><Type>
    plain</Type></Label><Input><CheckBox><InitialValue>false</
    InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
    ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
    Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

> 注:
>
> nFactor 関連のさまざまなトピックの詳細については、「nFactor 認証」を参照してください。

nFactor を使用してクッキーを設定する

February 21, 2022

nFactor カスタムラベルを適用し、認証フローの要素として Cookie を設定できます。カスタムラベルを使用すると、JavaScript を使用してログインスキーマを操作できます。

Cookie を要素として設定するために、スキーマなしのログインで実行される情報をユーザーに表示する必要はありません。代わりに、ユーザーのブラウザと対話して、必要なデータを格納するようにログインスキーマに指示する必要があります。ログインスキーマは、ページが読み込まれたときに Cookie を設定するために必要です。クッキーはカスタムラベルと JavaScript コードで設定されます。

Cookie を設定する要素を実装するには、cookie.xml という名前の XML ファイルを作成し、スキーマを /nsconfig/loginschema/ ディレクトリに以下の内容で保存します。

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response
    /1">
<Status>success</Status>
<Result>more-info</Result>
```



```
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>

<Requirement>
<Credential><ID>nsg_cookie</ID><Type>nsg_cookie</Type></Credential>
<Label><Text>Logon Type:</Text><Type>Plain</Type></Label>
</Requirement>

<Requirement>
<Credential><ID>loginBtn</ID><Type>none</Type></Credential>
<Label><Type>none</Type></Label><Input><Button>Log On</Button></Input>
</Requirement>

</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

この XML では

- カスタムラベル nsg_cookie は、Cookie を作成してフォームを送信し、フォームボタンを送信するために使用されます。
- RFWebUI_Custom は、RFWebUI テーマに基づいた新しいポータルテーマです。

nFactor を使用してクッキーを設定する手順

1. RFWebUI テーマに基づいてポータルテーマを作成します。

   ```
   add vpn portaltheme RfWebUI_custom -basetheme RfWebUI
   <!--NeedCopy-->
   ```

   このコマンドは、/var/netscaler/logon/themes/rfwebUI_CUSTOM にこのテーマのフォルダーを作成します。

2. /var/netscaler/logon/themes/RfWebUI_custom/script.js ファイルを編集し、次のスクリプトを追加します。

   ```
   CTXS.ExtensionAPI.addCustomCredentialHandler({
   ```



```
2
3     // The name of the credential, must match the type returned by
          the server
4     getCredentialTypeName: function () {
5    return "nsg_cookie";  }
6   ,
7     // Generate HTML for the custom credential
8     getCredentialTypeMarkup: function (requirements) {
9
10         var div = $("<div></div>");
11         $(document).ready(function() {
12
13         //Set cookie valid for 1000 days
14         var exdays = 1000;
15         var d = new Date();
16         d.setTime(d.getTime() + (exdays\*24\*60\*60\*1000));
17         var expires = "expires="+ d.toUTCString();
18         document.cookie = "NSC_COOKIE_NAME=CookieValue;" + expires
               + ";path=/";
19
20         //Submit form
21         document.getElementById('loginBtn').click();
22          }
23   );
24         return div;
25      }
26
27   }
28   );
29  <!--NeedCopy-->
```

このコードは次の処理を実行します。

- ブラウザがページの読み込みを完了するのを待ちます。
- NSC_COOKIE_NAME という名前のクッキーを cookieValue という値で設定します。1000 日間有効です。
- フォームを自動送信します。

クッキーが作成され、ユーザーはページを操作する必要がありません。

3. ログインスキーマを作成して、set cookie ファクターを表すポリシーラベルにバインドします。

```
1  add authentication loginSchema Cookie_LS -authenticationSchema "/
       nsconfig/loginschema/cookie.xml"
```



```
<!--NeedCopy-->
```

4. NO_AUTHN 認証ポリシーを作成して、設定された Cookie 係数を表すポリシーラベルにバインドします。

```
add authentication Policy NO_AUTHN_POL -rule TRUE -action NO_AUTHN
<!--NeedCopy-->
```

このポリシーは常に true と評価され、ユーザは次の要素に移動するか、認証フローを完了します。

5. ポータルテーマ rfwebUI_custom を Citrix Gateway 仮想サーバーまたは Citrix ADC AAA 仮想サーバーにバインドします。

nFactor 認証を使用したサンプルデプロイ

July 28, 2022

以下は、nFactor 認証を使用するサンプルデプロイメントです。

- 2 つのパスワードを前もって取得し、次の要素でパススルーします。Read
- グループの抽出に続いて、グループメンバーシップに基づく証明書または LDAP 認証。読み取り
- SAML の後に、SAML 中に抽出された属性に基づく LDAP または証明書認証が続きます。Read
- SAML を第 1 要素とし、次にグループを抽出し、抽出されたグループに基づいて LDAP または証明書認証を行います。読み取り
- 証明書からユーザー名を事前に入力しています。Read
- 401 対応トラフィック管理仮想サーバーの証明書認証とそれに続くグループ抽出。読み取り
- ユーザ名と 2 つのパスワード、グループ抽出の 3 番目の要因。Read
- 証明書は同じカスケードで LDAP にフォールバックします。証明書と LDAP 認証の両方に 1 つの仮想サーバーを使用します。読み取り
- 第 1 ファクタは LDAP で、第 2 ファクタは WebAuth です。読み取り
- ドメインは最初の要素でドロップダウンし、次にグループに基づいて異なるポリシー評価を行います。読み取り

すべての方法記事

October 7, 2021

認証、承認、監査の「How to articles」は、シンプルで関連性が高く、実装が容易な記事です。これらの記事には、LDAP 認証や多要素認証など、一般的な認証、承認、監査機能の一部に関する情報が含まれています。Citrix ADC に

よる認証の構成とトラブルシューティングに関する一般的な記事の一部については、Citrix ADC 認証：どうすればよいですか？ を参照してください。

エンドポイント分析

nFactor 認証の要素として事前認証エンドポイント分析スキャンを設定する

Citrix ADC nFactor 認証の要素として認証後のエンドポイント分析スキャンを構成する

nFactor 認証の要素として、事前認証と事後認証の EPA スキャンを構成します

nFactor 認証の要素として定期的なエンドポイント分析スキャンを設定する

第 **1** 因子と第 **2** 因子の構成の組み合わせ

最初の要素で WebAuth を使用し、2 番目の要素でパスワードを変更して LDAP を使用する Citrix Gateway の nFactor を構成する

nFactor 認証での SAML 属性の抽出に基づく LDAP 認証または証明書認証の後に SAML を構成します

Citrix ADC nFactor 認証で証明書認証を第 1 要素として構成し、LDAP を第 2 要素として構成します

Citrix ADC nFactor 認証で、1 つのログインスキーマと 1 つのパススルースキーマを使用して 2 要素認証を構成する

nFactor 認証による 3 番目の要素のグループ抽出によるユーザー名と 2 つのパスワードの構成

最初のファクタのドメインドロップダウン、ユーザー名、およびパスワードフィールドを設定し、次のファクタのグループに基づくポリシー評価を設定します。

最初の要素で電子メール ID（またはユーザー名）入力ベースのグループ抽出を構成して、次の要素認証フローを決定します。

最初の要素でユーザー入力用のドメインドロップダウンリストを設定し、次の要素認証フローを決定します。

認証要素としての **EULA**

Citrix ADC nFactor システムで認証係数として EULA を構成する

証明書からユーザー名を事前入力します

Citrix ADC nFactor 認証で証明書からプリフィルのユーザー名を構成する

ステップアップ認証

ステップアップ認証など、ログインサイトの要件が異なるアプリケーション用に nFactor を構成します

SAML 認証

October 7, 2021

セキュリティアサーションマークアップ言語 (SAML) は、シングルサインオン機能を提供する XML ベースの認証メカニズムであり、OASIS セキュリティサービス技術委員会によって定義されています。

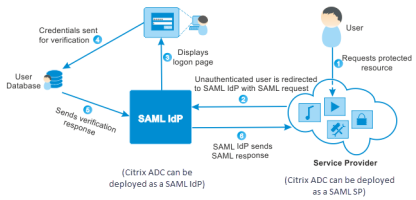
> 注
>
> NetScaler 12.0 ビルド 51.x 以降、多要素（nFactor）認証を使用する SAML サービスプロバイダー（SP）として使用される Citrix ADC アプライアンスは、ログインページのユーザー名フィールドに事前入力されるようになりました。アプライアンスは、SAML 認証要求の一部として名前 ID 属性を送信し、Citrix ADC SAML アイデンティティプロバイダー（IdP）から名前 ID 属性値を取得し、ユーザー名フィールドに事前入力します。

SAML 認証を使用する理由

サービスプロバイダー (LargeProvider) が顧客 (BigCompany) の多数のアプリケーションをホストするシナリオを考えてみましょう。BigCompany には、これらのアプリケーションにシームレスにアクセスする必要があるユーザーがいます。従来のセットアップでは、LargeProvider は BigCompany のユーザーのデータベースを維持する必要があります。これは、次の利害関係者のそれぞれにいくつかの懸念を提起します。

- LargeProvider は、ユーザーデータのセキュリティを確保する必要があります。
- BigCompany は、ユーザーを検証し、独自のデータベースだけでなく、LargeProvider が管理するユーザーデータベースにもユーザーデータを最新の状態に保つ必要があります。たとえば、BigCompany データベースから削除されたユーザーは、LargeProvider データベースからも削除する必要があります。
- ユーザーは、ホストされる各アプリケーションに個別にログオンする必要があります。

SAML 認証メカニズムは、別のアプローチを提供します。次の配置図は、SAML がどのように機能するかを示しています（SP が開始するフロー）。



従来の認証メカニズムによって提起された懸念事項は、次のように解決されます。

- LargeProvider は、BigCompany ユーザー用のデータベースを維持する必要はありません。LargeProvider は、アイデンティティ管理から解放され、より良いサービスを提供することに集中することができます。
- BigCompany は、LargeProvider ユーザーデータベースが独自のユーザーデータベースと同期していることを確認する負担を負いません。
- ユーザーは、LargeProvider でホストされている 1 つのアプリケーションに一度ログオンし、そこでホストされている他のアプリケーションに自動的にログオンできます。

Citrix ADC アプライアンスは、SAML サービスプロバイダー（SP）および SAML アイデンティティプロバイダー（IdP）として展開できます。関連するトピックを読んで、Citrix ADC アプライアンスで実行する必要がある構成を理解してください。

SAML サービスプロバイダーとして構成された Citrix ADC アプライアンスは、オーディエンス制限チェックを強制できるようになりました。対象者制限条件は、SAML の返信側が指定された対象者の少なくとも 1 つのメンバーである場合のみ、「有効」と評価されます。

SAML アサーション内の属性をグループ属性として解析するように、Citrix ADC アプライアンスを構成できます。これらをグループ属性として解析すると、アプライアンスはグループにポリシーをバインドできます。

SAML SP としての Citrix ADC

October 7, 2021

SAML サービスプロバイダー (SP) は、サービスプロバイダーによってデプロイされた SAML エンティティです。ユーザーが保護されたアプリケーションにアクセスしようとすると、SP はクライアント要求を評価します。クライアントが認証されていない（有効な NSC_TMAA または NSC_TMAS クッキーがない）場合、SP は要求を SAML アイデンティティ・プロバイダ（IdP）にリダイレクトします。

また、SP は IdP から受信した SAML アサーションも検証します。

Citrix ADC アプライアンスが SP として構成されている場合、すべてのユーザー要求は、関連する SAML アクションに関連付けられたトラフィック管理仮想サーバー（負荷分散またはコンテンツスイッチング）によって受信されます。

Citrix ADC アプライアンスは、ログアウト時の POST およびリダイレクトバインディングもサポートしています。

> 注
>
> Citrix ADC アプライアンスは、SAML IdP がアプライアンスまたは任意の外部 SAML IdP で構成されるデプロイメントで、SAML SP として使用できます。

SAML SP として使用する場合、Citrix ADC アプライアンスは以下を実行します。

- SAML トークンからユーザー情報 (属性) を抽出できます。この情報は、Citrix ADC アプライアンス上で構成されているポリシーで使用できます。たとえば、GroupMember 属性と emailaddress 属性を抽出する場合は、SAMLAction で **Attribute2** パラメーターを GroupMember として指定し、**Attribute3** パラメーターをメールアドレスとして指定します。

  > 注
  >
  > ユーザー名、パスワード、ログアウト URL などのデフォルト属性は、暗黙的に解析され、セッションに格納されるため、属性 1 ～16 で抽出しないでください。

- 着信 SAML アサーションから最大 127 バイトの属性名を抽出できます。以前の制限は 63 バイトでした。
- ポスト、リダイレクト、アーティファクトのバインディングをサポートします。

> 注
>
> 膨張またはデコード後のアサーションが 10K を超える場合、大量のデータにはリダイレクトバインディングを使用しないでください。

- アサーションを復号化できます。
- SAML アサーションから複数値の属性を抽出できます。これらの属性は、次のようなネストされた XML タグで送信されます。

  <AttributeValue> <AttributeValue>Value1</AttributeValue>
  <AttributeValue>Value2</AttributeValue>
  </AttributeValue>

  > 注
  >
  > Citrix ADC 13.0 ビルド 63.x 以降では、SAML 属性の個々の最大長が増加し、最大 40k バイトが許可されるようになりました。すべての属性のサイズは 40k バイトを超えてはなりません。

  Citrix ADC アプライアンスは、以前の XML で示されている場合、Value1 のみを抽出する古いファームウェアとは異なり、Value1 と Value2 の両方を特定の属性の値として抽出できます。
- SAML アサーションの有効性を指定できます。

  Citrix ADC SAML IdP とピア SAML SP のシステム時刻が同期していない場合、いずれかの当事者によってメッセージが無効になることがあります。このようなケースを回避するために、アサーションが有効な期間を設定できるようになりました。

  この期間は「スキュー時間」と呼ばれ、メッセージを受信する分数を指定します。スキュー時間は、SAML SP および SAML IdP で構成できます。
- 認証要求の 'ForceAuth' という余分な属性を外部の IdP (ID プロバイダ) に送信できます。デフォルトでは、ForceAuthn は「False」に設定されています。既存の認証コンテキストにもかかわらず認証を強制するように IdP を提案するには、「True」に設定できます。また、アーティファクトバインディングで構成されている場合、Citrix ADC SP はクエリパラメータで認証要求を行います。

コマンドラインインターフェイスを使用して **Citrix ADC** アプライアンスを **SAML SP** として構成するには

1. SAML SP アクションを構成します。

   例

   次のコマンドは、認証されていないユーザー要求をリダイレクトする SAML アクションを追加します。

   ```
   add authentication samlAction SamlSPAct1 -samlIdPCertName nssp -samlSigningCertNa
    nssp -samlRedirectUrl https://auth1.example.com -relaystateRule "AAA.
   LOGIN.RELAYSTATE.EQ(\"https://lb.example1.com/\")"
   ```


注意事項

- SamLAction コマンドので-samlIdPCertName提供される証明書は、シグニチャの検証を成功させるには、IdP の対応する証明書と一致する必要があります。
- SAML は RSA 証明書のみをサポートします。HSM、FIPS などの他の証明書はサポートされていません。
- 式の末尾に「/」を付けた完全なドメイン名を付けることをお勧めします。
- 管理者は、samlAction コマンドで **RelaysStateRule** の式を設定する必要があります。式には、ユーザーが認証仮想サーバーにリダイレクトされる前に接続する公開ドメインのリストが含まれている必要があります。たとえば、式には、この SAML アクションを認証に使用するフロントエンド仮想サーバー (VPN、LB、または CS) のドメインが含まれている必要があります。

コマンドの詳細については、「https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction」および「https://support.citrix.com/article/CTX316577」を参照してください。

2. SAML ポリシーを設定します。

   例

   次のコマンドは、以前に定義された SAML アクションをすべてのトラフィックに適用する SAML ポリシーを定義します。

   ```
   add authentication policy SamlSPPol1 -rule true -action SamlSPAct1
   ```

3. SAML ポリシーを認証仮想サーバーにバインドします。

   例

   次のコマンドは、SAML ポリシーを「av_saml」という名前の認証仮想サーバーにバインドします。

   ```
   bind authentication vserver av_saml -policy SamlSPPol1
   ```

4. 認証仮想サーバを適切なトラフィック管理仮想サーバにバインドします。

   例

   次のコマンドは、「lb1_ssl」という名前の負荷分散仮想サーバーを追加し、「av_saml」という名前の認証仮想サーバーを負荷分散仮想サーバーに関連付けます。

   ```
   add lb vserver lb1_ssl SSL 10.217.28.224 443 -persistenceType NONE -
   cltTimeout 180 -AuthenticationHost auth1.example.com -Authentication ON
    -authnVsName av_saml
   ```

   コマンドの詳細については、https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlActionを参照してください。

GUI を使用して Citrix ADC アプライアンスを SAML SP として構成するには

1. [セキュリティ] **>** [**AAA** ポリシー] **>** [認証] **>** [基本ポリシー] **>** [**SAML**] に移動します。

2. [サーバ] タブを選択し、[追加] をクリックし、次のパラメータの値を入力して、[作成] をクリックします。

   パラメータの説明:

   Name: サーバーの名前

   リダイレクト URL-ユーザーが認証する URL。一部の IdP には、SAML セットアップでなければ到達できない特別な URL があります。

   シングルログアウト URL-Citrix ADC がクライアントを IdP に戻してサインアウトプロセスを完了するタイミングを認識できるように指定する URL。この単純な展開では使用しません。

   SAML バインド: クライアントを SP から IdP に移動するために使用されるメソッド。これは IdP でクライアントがどのように接続するかを理解するために、IdP で同じである必要があります。
   Citrix ADC が SP として機能する場合、POST、REDIRECT、ARTIFACT バインディングがサポートされます。

   ログアウトバインド-リダイレクト

   IDP 証明書名-SAML 署名証明書の下にある IdPCert 証明書 (Base64)。

   User Field-IdP の SAML 認証フォームのセクション。必要な場合に抽出する SP のユーザー名が含まれています。

   署名証明書名-Citrix ADC が IdP への認証要求に署名するために使用する SAML SP 証明書（秘密キー付き）を選択します。IdP が認証要求署名を検証できるように、同じ証明書（秘密キーなし）を IdP にインポートする必要があります。このフィールドは、ほとんどの IdP では必要ありません。

   発行者名-識別子。SP と IdP の両方で指定される一意の ID。サービスプロバイダーを相互に識別するのに役立ちます。

   署名されていないアサーションを拒否する-IdP のアサーションに署名する必要がある場合に指定できるオプション。アサーションのみを署名（ON）するか、IdP からのアサーションとレスポンスの両方を署名する必要がある（STRICT）ことを保証できます。

   Audience-IdP によって送信されたアサーションが適用されるオーディエンス。これは通常、Service-Provider を表すエンティティ名または URL です。

   署名アルゴリズム-RSA-SHA256

   ダイジェストメソッド-SHA256

   デフォルト認証グループ-抽出されたグループに加えて認証が成功したときに選択されるデフォルトグループ。

   グループ名フィールド-ユーザーグループを含むアサーション内のタグの名前。

   スキュー時間（分）-このオプションは、Citrix ADC ServiceProvider が着信アサーションで許可するクロックスキューを分単位で指定します。

3. 同様に、対応する SAML ポリシーを作成し、それを認証仮想サーバーにバインドします。

[セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバー] に移動し、SAML ポリシーを認証仮想サーバーに関連付けます。

4. 認証サーバを適切なトラフィック管理仮想サーバに関連付けます。

   [トラフィック管理**] > [**負荷分散]（または [コンテンツスイッチング]）> [仮想サーバー] に移動し、仮想サーバーを選択し、認証仮想サーバーをそれに関連付けます。

SAML IdP としての **Citrix ADC**

October 7, 2021

SAML IdP (ID プロバイダー) は、顧客ネットワークにデプロイされる SAML エンティティです。IdP は SAML SP から要求を受信し、ユーザーをログオンページにリダイレクトします。このページでは、認証情報を入力する必要があります。IdP は、Active Directory（LDAP などの外部認証サーバー）を使用してこれらの資格情報を認証してから、SP に送信される SAML アサーションを生成します。

SP はトークンを検証し、要求された保護されたアプリケーションへのアクセス権をユーザーに付与します。

Citrix ADC アプライアンスが IdP として構成されている場合、すべての要求は、関連する SAML IdP プロファイルに関連付けられた認証仮想サーバーによって受信されます。

> 注
>
> Citrix ADC アプライアンスは、SAML SP がアプライアンスまたは外部 SAML SP のいずれかに構成されている展開で IdP として使用できます。

SAML IdP として使用する場合、Citrix ADC アプライアンスは以下を実行します。

- 従来のログオンでサポートされているすべての認証方法をサポートします。
- アサーションにデジタル署名します。
- 一要素認証と 2 要素認証をサポートします。SAML をセカンダリ認証メカニズムとして設定しないでください。
- SAML SP の公開キーを使用してアサーションを暗号化できます。これは、アサーションに機密情報が含まれている場合に推奨されます。
- SAML SP からのデジタル署名された要求のみを受け入れるように構成できます。
- ネゴシエート、NTLM、および証明書の 401 ベースの認証メカニズムを使用して SAML IdP にログオンできます。
- NameId 属性に加えて 16 個の属性を送信するように設定できます。属性は、適切な認証サーバから抽出する必要があります。それぞれについて、SAML IdP プロファイルで名前、式、形式、およびフレンドリ名を指定できます。

- Citrix ADC アプライアンスが複数の SAML SP の SAML IdP として構成されている場合、ユーザーは毎回明示的に認証することなく、異なる SP 上のアプリケーションにアクセスできます。Citrix ADC アプライアンスは、最初の認証用にセッション Cookie を作成し、それ以降のすべてのリクエストでこの Cookie を認証に使用します。
- SAML アサーションで複数値属性を送信できます。
- ポストおよびリダイレクトバインディングをサポートします。アーティファクトバインディングのサポートは、Citrix ADC リリース 13.0 ビルド 36.27 で導入されました。
- SAML アサーションの有効性を指定できます。

  Citrix ADC SAML IdP とピア SAML SP のシステム時刻が同期していない場合、いずれかの当事者によってメッセージが無効になることがあります。このようなケースを回避するために、アサーションが有効な期間を設定できるようになりました。

  この期間は「スキュー時間」と呼ばれ、メッセージを受け付ける必要がある分数を指定します。スキュー時間は、SAML SP および SAML IdP で構成できます。
- IdP で事前構成されている、または IdP によって信頼されている SAML SP にのみアサーションを提供するように構成できます。この構成では、SAML IdP には、関連する SAML SP のサービスプロバイダー ID（または発行者名）が必要です。

  > 注
  >
  > 先に進む前に、LDAP 認証サーバにリンクされている認証仮想サーバがあることを確認してください。

コマンドラインインターフェイスを使用して **Citrix ADC** アプライアンスを **SAML IdP** として構成するには

1. SAML IdP プロファイルを設定します。

   例

   SiteMinder を SP として使用して、Citrix ADC アプライアンスを IdP として追加する。

   ```
   add authentication samlIdPProfile samlIDPProf1 -samlSPCertName siteminder
   -cert -encryptAssertion ON -samlIdPCertName ns-cert -assertionConsumerServiceURL
    http://sm-proxy.nsi-test.com:8080/affwebservices/public/saml2assertionconsumer
    -rejectUnsignedRequests ON -signatureAlg RSA-SHA256 -digestMethod
   SHA256 -acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re###^https://
   example2\.com/cgi/samlauth$##)
   ```

   注意事項

   - SAML IdP プロファイルで、この IdP に適用可能なサービスプロバイダー URL のリストの式をとる **acsurlRule** を設定します。この式は、使用している SP によって異なります。Citrix ADC が SP として構成されている場合、ACS URL は`https://<SP-domain_name>/cgi/samlauth`になります。一致するために、式に完全な URL を設定することをお勧めします。

- SAML は RSA 証明書のみをサポートします。HSM、FIPS などの他の証明書はサポートされていません。
- 文字列の末尾にドル記号「$」とともに「^」記号 (例:^https) でドメインの開始を指定する必要があります (例:samlauth$)。

コマンドの詳細については、「https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction」および「https://support.citrix.com/article/CTX316577」を参照してください。

2. SAML 認証ポリシーを設定し、SAML IdP プロファイルをポリシーのアクションとして関連付けます。

```
add authentication samlIdPPolicy samlIDPPol1 -rule true -action samlIDPProf1
```

3. 認証仮想サーバにポリシーをバインドします。

```
bind authentication vserver saml-auth-vserver -policy samlIDPPol1 -
priority 100
```

コマンドの詳細については、https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlIdPProfileを参照してください。

GUI を使用して **Citrix ADC** アプライアンスを **SAML IdP** として構成するには

1. [セキュリティ] **>** **[AAA** ポリシー**]** **>** **[**認証**]** **>** **[**高度なポリシー**]** **>** **[SAML IdP**] に移動します。
2. [サーバ] タブを選択し、[追加] をクリックし、次のパラメータの値を入力して、[作成] をクリックします。

   パラメータの説明:

   アサーションコンシューマサービス URL-認証されたユーザーがリダイレクトされる URL。

   IdP 証明書名-認証ページに使用される証明書とキーのペア。

   SP 証明書名-サービスプロバイダの証明書このシナリオでは、キーは必要ありません。

   Sign Assertion- クライアントをサービスプロバイダにリダイレクトするときに、アサーションと応答に署名するオプション。

   発行者名-識別子。SP と IdP の両方で指定される一意の ID。サービスプロバイダーを相互に識別するのに役立ちます。

   サービスプロバイダ ID：SP と IdP の両方で指定される一意の ID。サービスプロバイダーを相互に識別するのに役立ちます。これは何でもかまいません。以下で指定する URL である必要はありませんが、SP プロファイルと IdP プロファイルの両方で同じである必要があります。

   署名されていないリクエストを拒否する：SP 証明書で署名されたアサーションのみが受け入れられるように指定できるオプション。

署名アルゴリズム-IdP と SP の間のアサーションの署名と検証に使用されるアルゴリズム。これは IdP プロファイルと SP プロファイルの両方で同じである必要があります。

Digest Method-IdP と SP の間のアサーションの整合性を検証するために使用されるアルゴリズム。これは、IdP と SP プロファイルの両方で同じである必要があります。

SAML バインド：SP プロファイルで説明されているように、SP と IdP の両方で同じである必要があります。

3. SAML IdP ポリシーを認証仮想サーバーに関連付けます。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバー] に移動し、SAML IdP ポリシーを認証仮想サーバーに関連付けます。

SAML シングルサインオンの設定

May 9, 2022

サービスプロバイダーでホストされているアプリケーション間でシングルサインオン機能を提供するには、SAML SP で SAML シングルサインオンを構成します。

コマンドラインインターフェイスを使用した SAML シングルサインオンの設定

1. SAML SSO プロファイルを設定します。

   例

   次のコマンドでは、SharePoint ポータルからの Web リンクを持つ負荷分散仮想サーバーの例を示します 。Nssp.example.com は、SharePoint サーバーの負荷分散を行うトラフィック管理の仮想サーバーです。

```
1  add tm samlSSOProfile tm-saml-sso -samlSigningCertName nssp -
       assertionConsumerServiceURL "https://nssp2.example.com/cgi/
       samlauth" -relaystateRule "\\"https://nssp2.example.com/
       samlsso.html\\"" -sendPassword ON -samlIssuerName nssp.example
       .com
2  <!--NeedCopy-->
```

2. SAML SSO プロファイルをトラフィックアクションに関連付けます。

   例

   次のコマンドは、SSO を有効にし、上記で作成した SAML SSO プロファイルをトラフィックアクションにバインドします。


```
add tm trafficAction html_act -SSO ON -samlSSOProfile tm-saml-sso
<!--NeedCopy-->
```

3. アクションを実行する必要があるタイミングを指定するトラフィックポリシーを設定します。

   例

   次のコマンドは、トラフィックアクションをトラフィックポリシーに関連付けます。

```
add tm trafficPolicy html_pol "HTTP.REQ.URL.CONTAINS(\"abc.html\")" html_act
<!--NeedCopy-->
```

4. 以前に作成したトラフィックポリシーをトラフィック管理仮想サーバ（負荷分散またはコンテンツスイッチング）にバインドします。また、トラフィックポリシーをグローバルに関連付けることもできます。

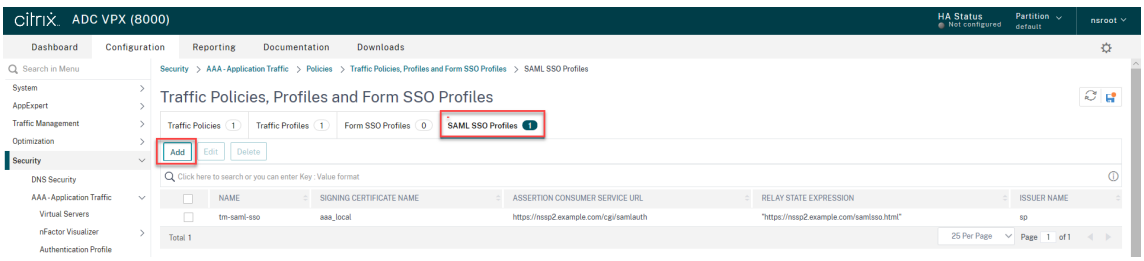
   > 注
   >
   > このトラフィック管理仮想サーバーは、SAML アクションに関連付けられた関連する認証仮想サーバーに関連付ける必要があります。

```
bind lb vserver lb1_ssl -policyName html_pol -priority 100 -gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

GUI を使用した SAML シングルサインオンの設定

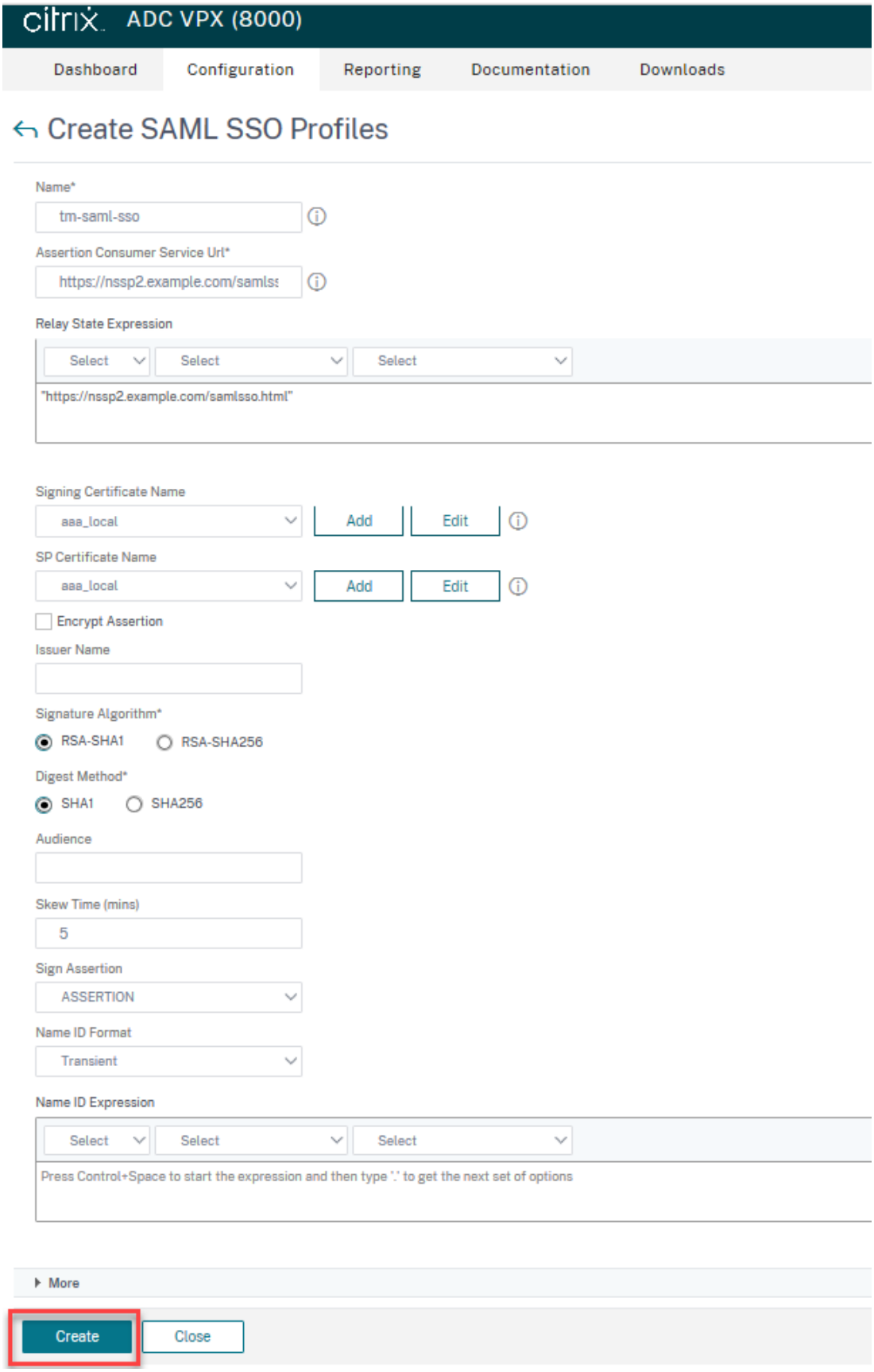
SAML シングルサインオンを構成するには、SAML SSO プロファイル、トラフィックプロファイル、およびトラフィックポリシーを定義し、トラフィックポリシーをトラフィック管理仮想サーバーまたは Citrix ADC アプライアンスにグローバルにバインドする必要があります。

1. セキュリティ **>** **AAA** アプリケーショントラフィック **>** ポリシー **>** トラフィック **>** **SAML SSO** プロファイルに移動し、[追加] をクリックします。

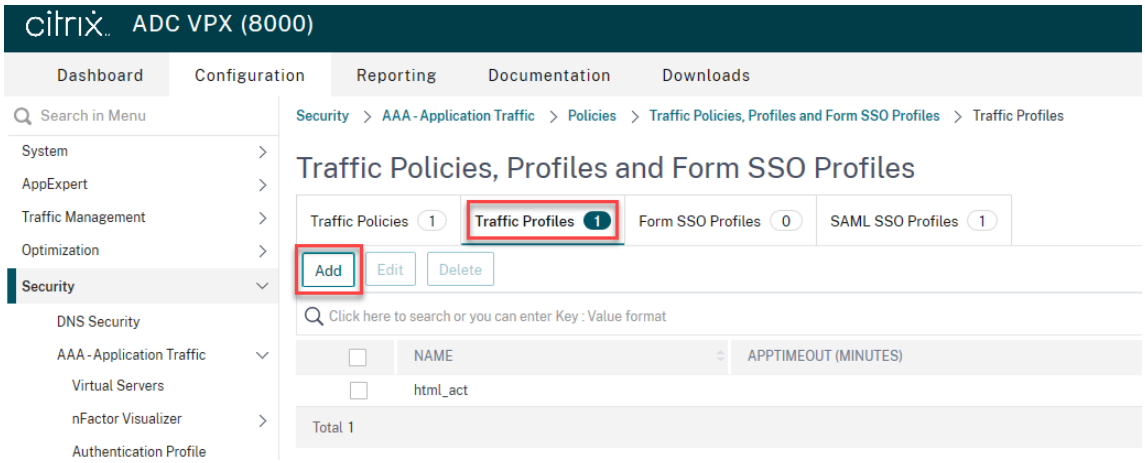


2. [**SAML SSO** プロファイルの作成] ページで、次のフィールドに値を入力し、[作成] をクリックします。

- Name-SAML SSO プロファイルの名前
- アサーションコンシューマサービス URL-アサーションが送信される URL
- 署名証明書名-アサーションの署名に使用される SSL 証明書の名前
- SP 証明書名：アサーションが暗号化されるピア/受信側の SSL 証明書の名前
- 発行者名-Citrix ADC から IdP に送信される Citrix ADC を一意に識別するためのリクエストで使用される名前
- 署名アルゴリズム-SAML トランザクションの署名/検証に使用するアルゴリズム
- Digest Method-SAML トランザクションのダイジェストを計算/検証するために使用されるアルゴリズム
- Audience-IdP によって送信されたアサーションが適用可能なオーディエンス。これは通常、サービスプロバイダを表すエンティティ名または URL です。
- スキュー時間 (分)-アサーションが有効になる現在の時刻の両側の分数
- アサーションの署名-Citrix ADC IDP がアサーションを送信するときにアサーションの一部に署名するオプション。ユーザーの選択に基づいて、[アサーション] または [応答]、または [両方] または [なし] のいずれかに署名できます。
- 名前 ID 形式-アサーションで送信される名前識別子の形式
- 名前 ID 式-アサーションで送信される NameIdentifier を取得するために評価される式

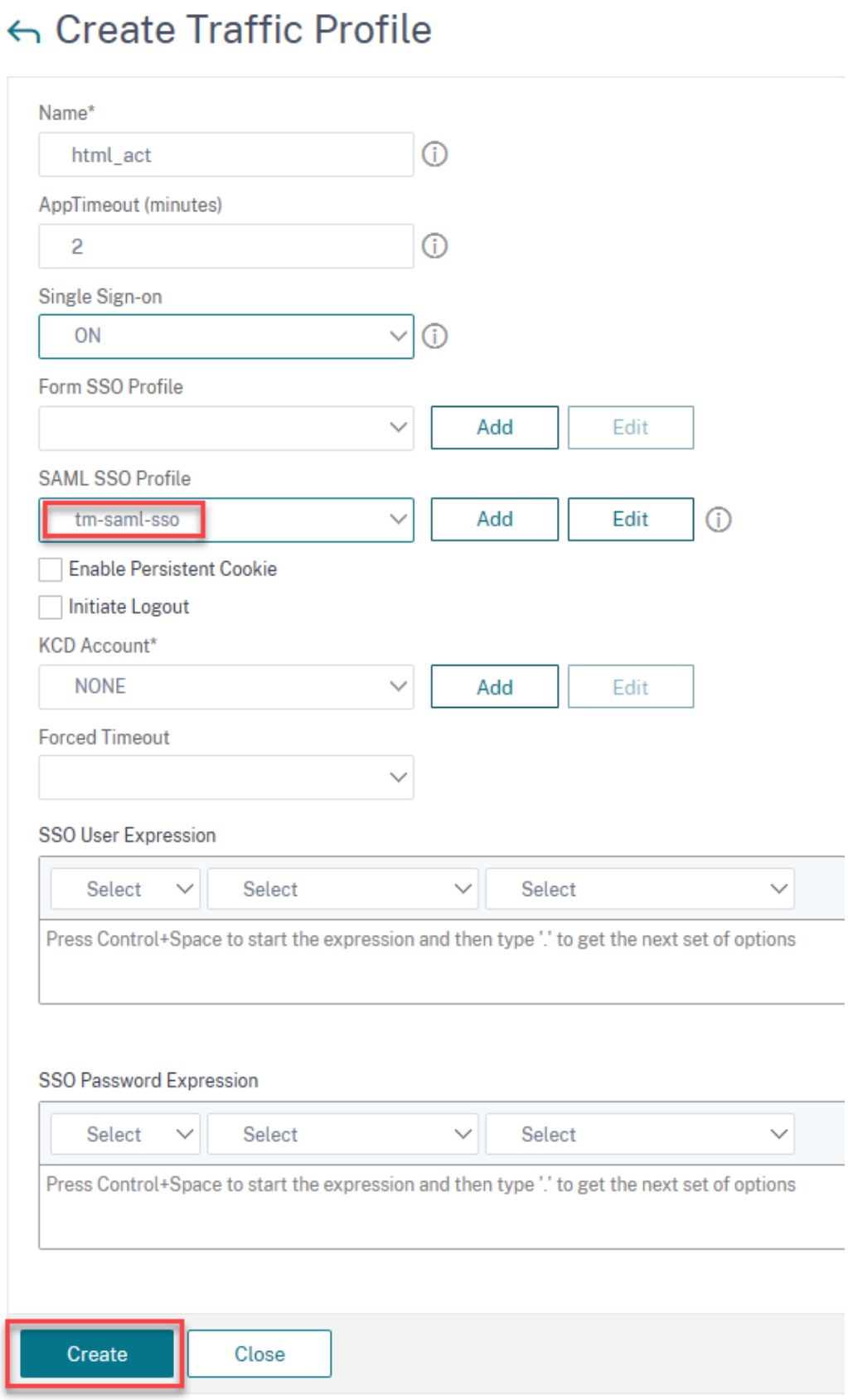


3. セキュリティ **>** **AAA** アプリケーショントラフィック **>** ポリシー **>** トラフィック **>** トラフィックプロファイルに移動し、[追加] をクリックします。

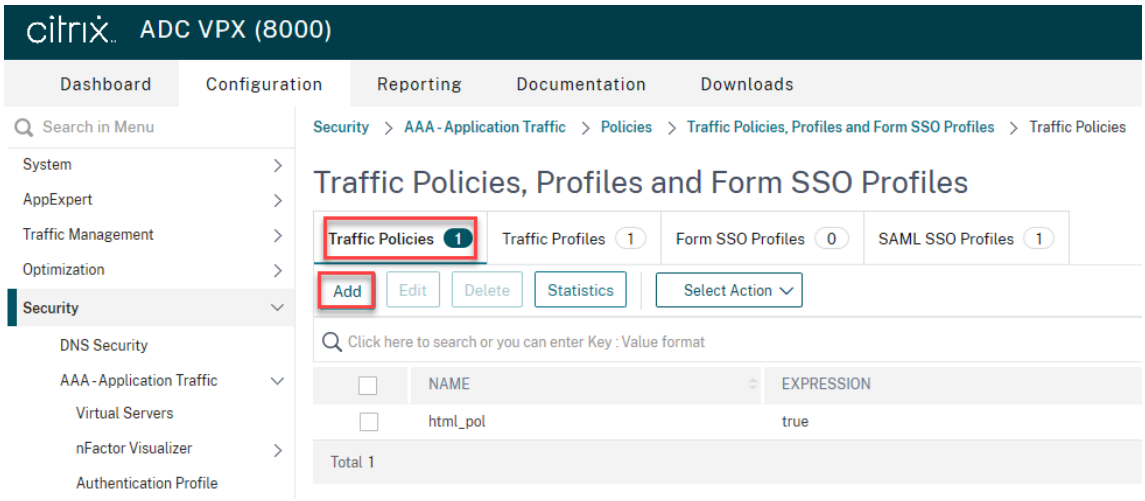


4. [トラフィックプロファイルの作成] ページで、次のフィールドに値を入力し、[作成] をクリックします。

   - Name-トラフィックアクションの名前。
   - AppTimeout (分)-接続が閉じられるまでのユーザーの非アクティブ時間の間隔（分単位）。
   - シングルサインオン-[オン] を選択します。
   - SAML SSO プロファイル-作成した SAML SSO プロファイルを選択します。
   - KCD アカウント-Kerberos 制約付き委任アカウント名
   - SSO ユーザー式-SingleSignon のユーザー名を取得するために評価される式
   - SSO パスワード式-SingleSignon のパスワードを取得するために評価される式

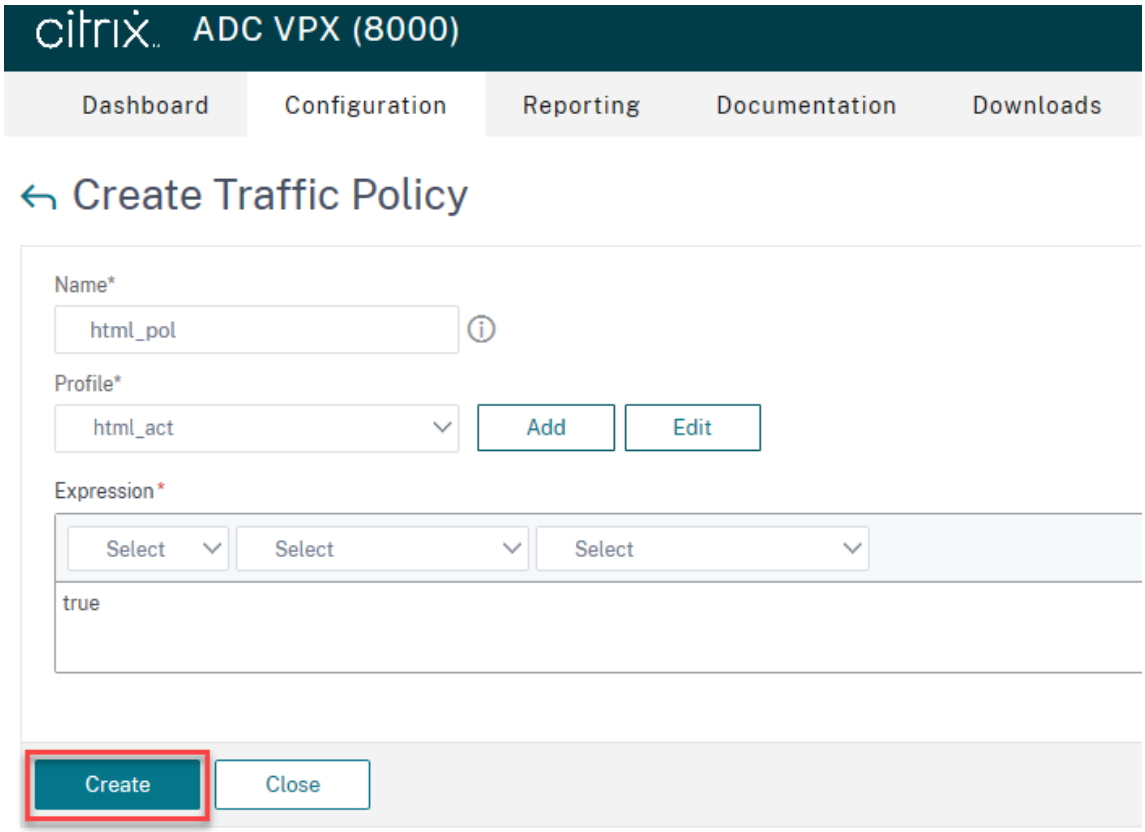


5. セキュリティ **>** **AAA** アプリケーショントラフィック **>** ポリシー **>** トラフィック **>** トラフィックポリシーに移動し、[追加] をクリックします。

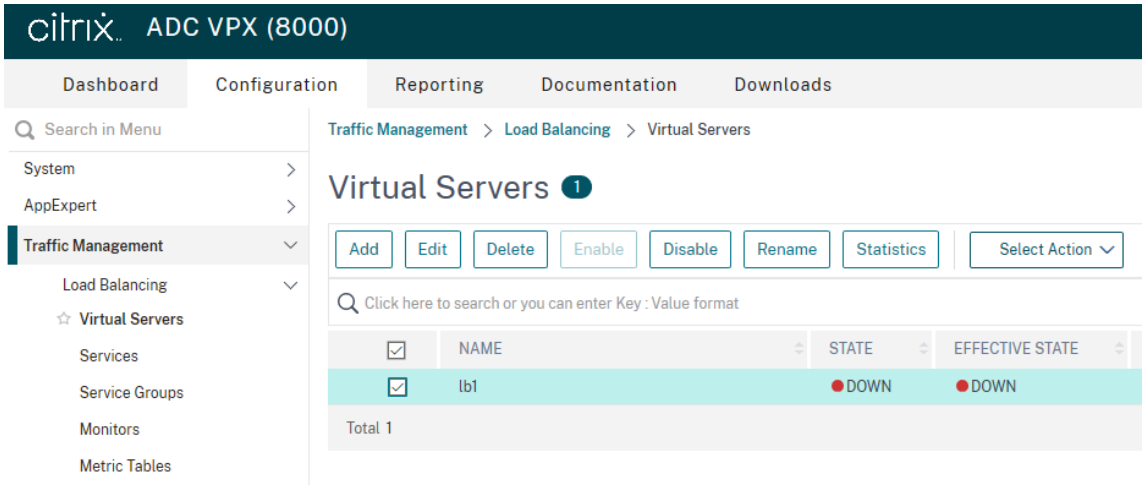


6. [トラフィックポリシーの作成] ページで、次の値を入力し、[作成] をクリックします。

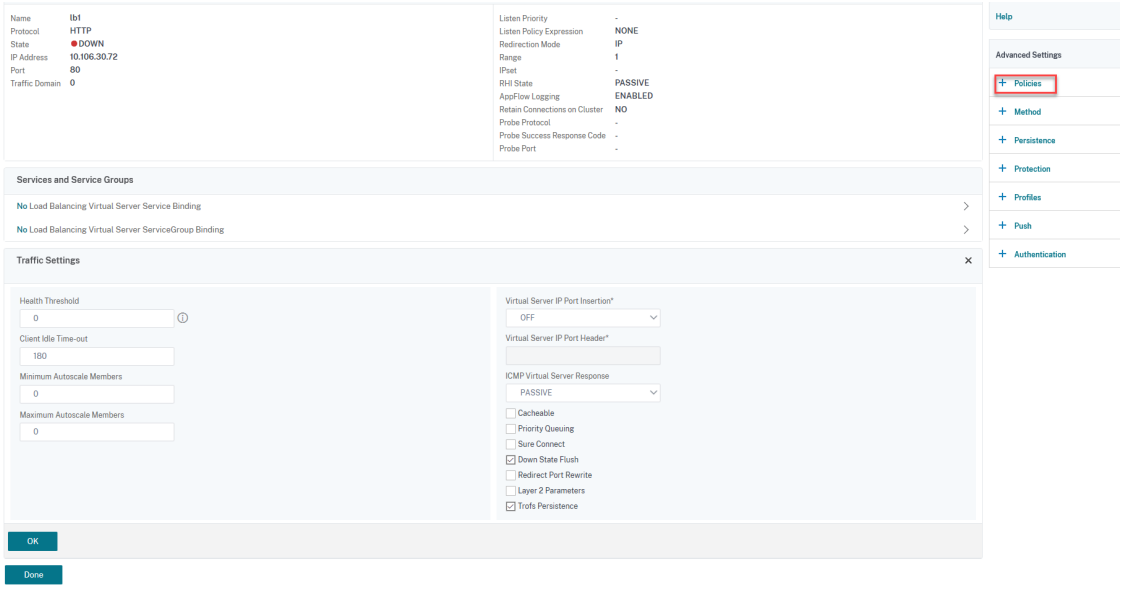
   - Name — 作成するトラフィックポリシーの名前
   - Profile — 作成したトラフィックプロファイルを選択します。
   - Expression — ポリシーが特定のリクエストに応答するために使用するデフォルトの構文式。たとえば、true です。



7. トラフィックポリシーをトラフィック管理仮想サーバーにバインドするには、仮想サーバーを選択します。



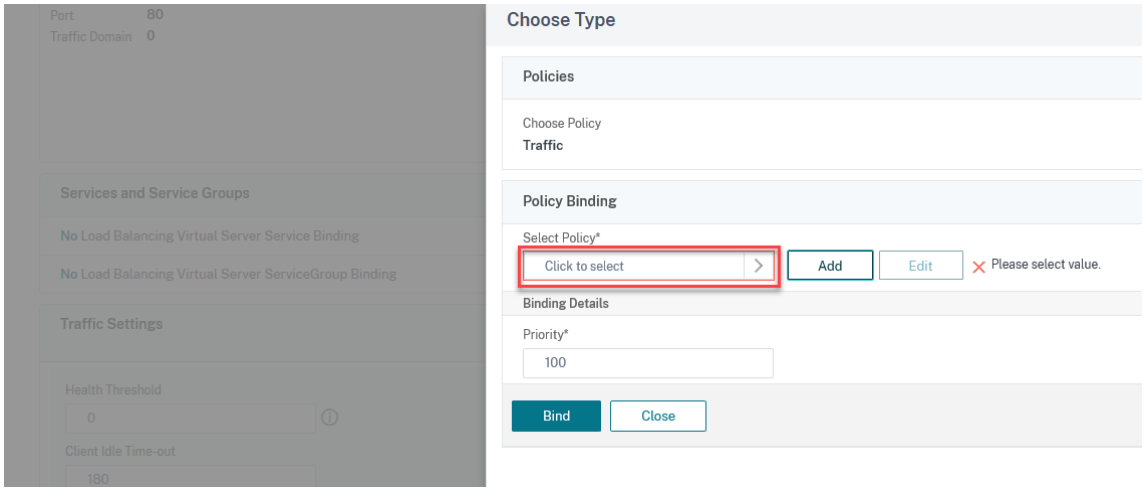
8. [ポリシー] をクリックします。



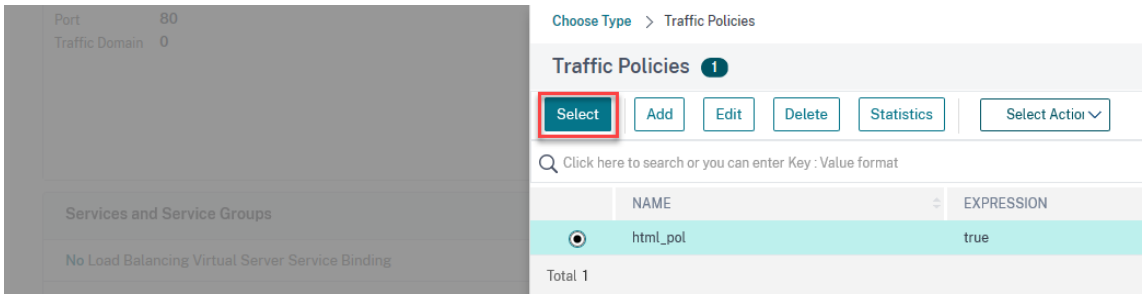
9. [ポリシーの選択] フィールドで [トラフィック] を選択し、[タイプの選択] フィールドで **[** 要求] を選択し ** 、[続行] をクリックします。

   ![クリックしてポリシーを追加 (/en-us/citrix-adc/media/saml-9.png)

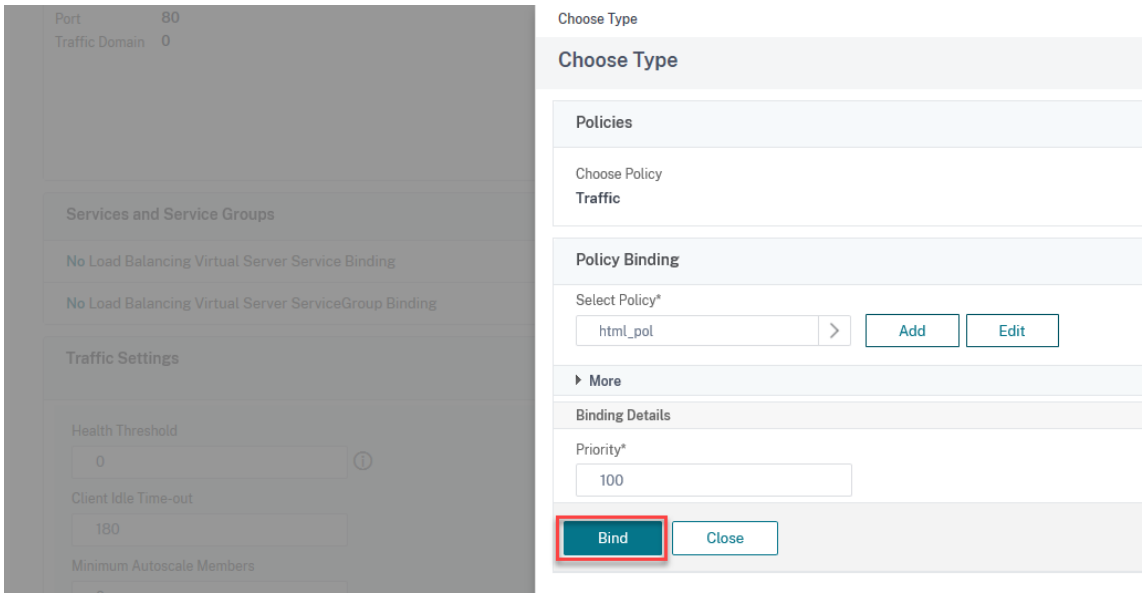
10. **[**ポリシーの選択（Select Policy）] フィールドで、作成したトラフィックをクリックして選択します。



11. [選択] をクリックします。



12. [バインド] をクリックして、トラフィックポリシーを仮想サーバーにバインドします。



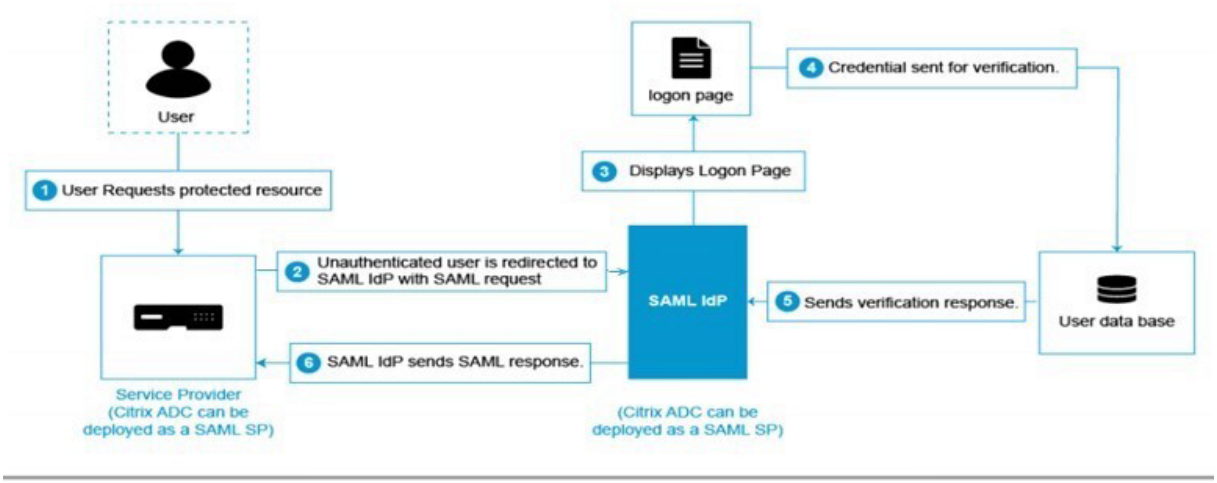
Azure AD を SAML IdP として構成し、Citrix ADC を SAML SP として構成する

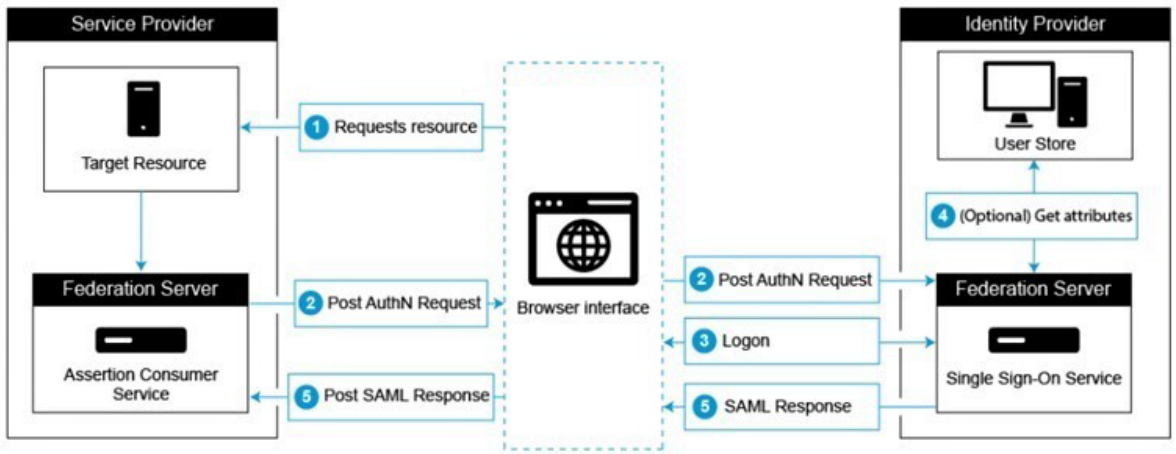
October 7, 2022

SAML サービスプロバイダ (SP) は、サービスプロバイダーによって展開される SAML エンティティです。ユーザーが保護されたアプリケーションにアクセスしようとすると、SP はクライアント要求を評価します。クライアントが認証されていない（有効な NSC_TMAA または NSC_TMAS Cookie がない）場合、SP は要求を SAML アイデンティティプロバイダ（IdP）にリダイレクトします。また、SP は IdP から受信した SAML アサーションも検証します。

SAML IdP (ID プロバイダー) は、顧客ネットワークにデプロイされる SAML エンティティです。IdP は SAML SP からリクエストを受信し、認証情報を入力する必要があるログオンページにユーザーをリダイレクトします。IdP は、これらの認証情報をユーザディレクトリ（LDAP などの外部認証サーバ）で認証し、SP に送信される SAML アサーションを生成します。SP はトークンを検証し、要求された保護されたアプリケーションへのアクセス権をユーザーに付与します。

次の図は、SAML 認証メカニズムを示しています。

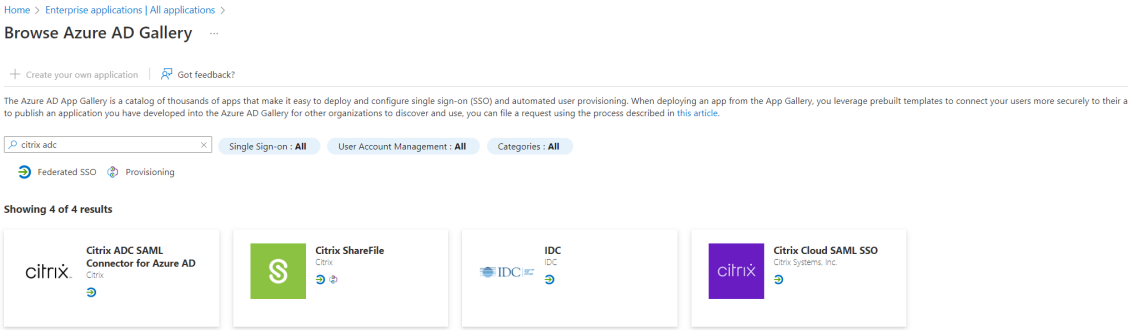




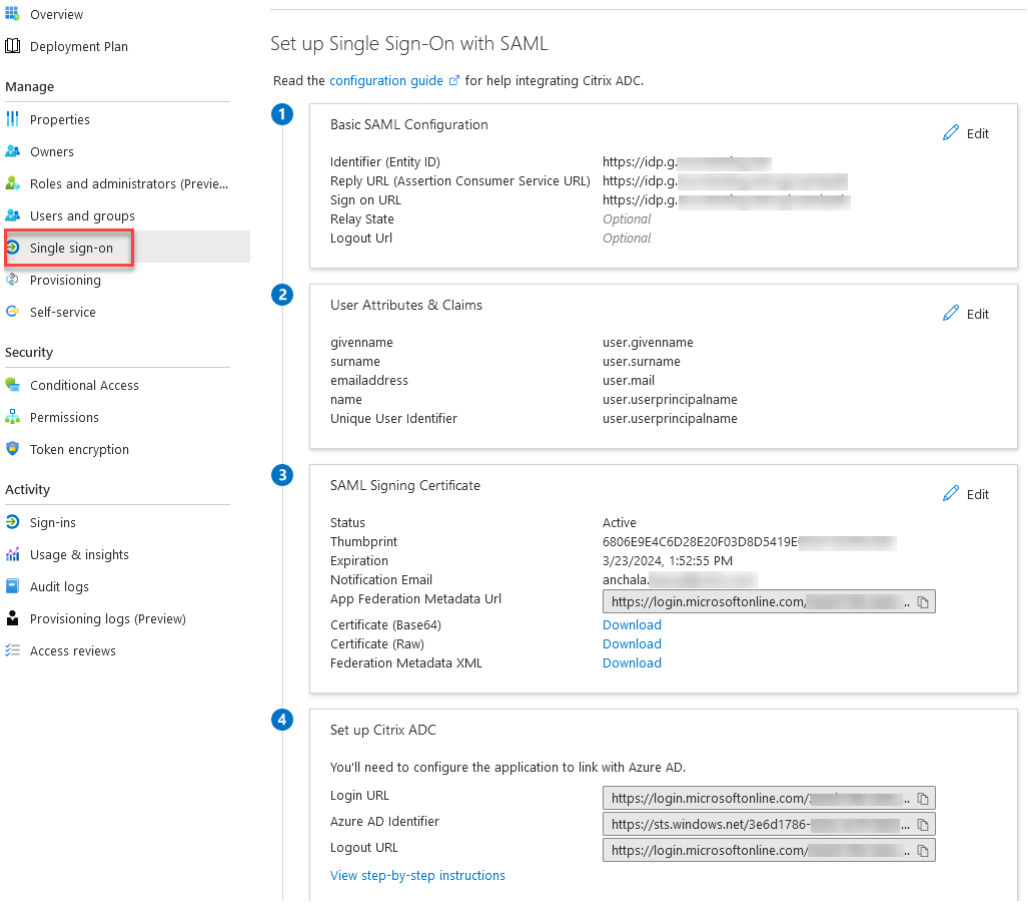
Azure AD サイド構成

シングルサインオン設定を構成します。

1. Azure ポータルで、[**Azure Active Directory**] をクリックします。
2. ナビゲーションペインの「管理」セクションで、「エンタープライズアプリケーション」をクリックします。Azure AD テナント内のアプリケーションのランダムなサンプルが表示されます。
3. 検索バーに「**Azure AD** 用 **Citrix ADC SAML** コネクタ」と入力します。



4. [管理] セクションで、[シングルサインオン] を選択します。
5. [**SAML**] を選択して、シングルサインオンを構成します。[**SAML** でシングルサインオンを設定する**-**プレビュー] ページが表示されます。ここで、Azure は SAML IdP として機能しています。
6. Citrix **ADC** を **SAML SP** として構成するときに、**SAML** 署名証明書の下に存在する証明書（Base64）を samlidPCertName として使用するダウンロードします。



7. 基本的な **SAML** オプションを設定します。

   識別子 **(**エンティティ **ID)** -一部のアプリで必須です。シングルサインオンが設定されているアプリケーションを一意に識別します。Azure AD は、SAML トークンのオーディエンスパラメータとして識別子をアプリケーションに送信します。アプリケーションは、それを検証することが期待されます。この値は、アプリケーションによって提供されるすべての SAML メタデータにもエンティティ ID として表示されます。

   返信 **URL** -必須。アプリケーションが SAML トークンの受信を期待する場所を指定します。応答 URL は、アサーションコンシューマサービス（ACS）URL とも呼ばれます。

   サインオン **URL** -ユーザーがこの URL を開くと、サービスプロバイダーは Azure AD にリダイレクトしてユーザーを認証し、サインインします。

   Relay State： 認証の完了後にユーザーをリダイレクトする場所をアプリケーションに指定します。

Citrix ADC 側の構成

1. [セキュリティ **] > [AAA** ポリシー **] > [**認証 **] > [**基本ポリシー **] > [SAML]** に移動します。
2. [サーバ] タブを選択し、[追加] をクリックし、次のパラメータの値を入力して、[作成] をクリックします。

   パラメータの説明:

太字のパラメーターの値は、Azure 側の構成から取得する必要があります。

Name: サーバーの名前

リダイレクト **URL** -Azure AD の「Citrix ADC セットアップ」セクションに以前使用したログイン URL を入力します。`https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2`

シングルログアウト URL- `https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2`

SAML バインディング-ポスト

ログアウトバインド-リダイレクト

IDP 証明書名 -SAML 署名証明書の下にある IdPCert 証明書 (Base64)。

ユーザーフィールド -UserPrincipalName。Azure IdP の「ユーザー属性とクレーム」セクションから取得されます。

署名証明書名 -Azure AD では必要ありません。Citrix ADC が IdP への認証要求に署名するために使用する SAML SP 証明書（秘密キー付き）を選択します。IdP が認証要求署名を検証できるように、同じ証明書（秘密キーなし）を IdP にインポートする必要があります。このフィールドは、ほとんどの IdP では必要ありません。

issuerName -識別子。`https://idp.g.nssvctesting.net`

符号なしアサーションを拒否する-オン

Audience-IdP によって送信されたアサーションが適用されるオーディエンス。これは通常、サービスプロバイダーを表すエンティティ名または URL です。

署名アルゴリズム-RSA-SHA256

ダイジェストメソッド-SHA256

デフォルト認証グループ-抽出されたグループに加えて、認証が成功した場合に選択されるデフォルトグループ。

グループ名フィールド-ユーザーグループを含むアサーション内のタグの名前。

スキュー時間 (分)-このオプションは、Citrix ADC ServiceProvider が受信アサーションで許容するクロックスキューを分単位で指定します。

2 つの要素- オフ

要求された認証コンテキスト-正確

認証クラスタイプ-なし

サムプリントを送信-オフ

ユーザー名を強制する-オン

認証を強制する-オフ

SAML レスポンスを保存-オフ

同様に、対応する SAML ポリシーを作成し、認証仮想サーバーにバインドします。

注:

- Azure AD では、SAML リクエストのサブジェクト ID フィールドは想定していません。
- Citrix ADC がサブジェクト ID フィールドを送信しないようにするには、Citrix ADC CLI で次のコマンドを入力します。
  `nsapimgr_wr.sh -ys call="ns_saml_dont_send_subject"`
  このコマンドは nFactor 認証ワークフローにのみ適用されます。

SAML でサポートされる機能の追加

March 8, 2022

SAML では次の機能がサポートされています。

SAML SP および IdP 構成のメタデータの読み取りと生成のサポート

Citrix ADC アプライアンスは、SAML サービスプロバイダー（SP）と ID プロバイダー（IdP）の両方の構成エンティティの手段としてメタデータファイルをサポートするようになりました。メタデータファイルは、エンティティの設定を記述する構造化された XML ファイルです。SP と IdP のメタデータファイルは別々です。デプロイメントに基づき、1 つの SP または IdP エンティティに複数のメタデータファイルがある場合もあります。
管理者は、Citrix ADC で（SAML SP および IdP）メタデータファイルをエクスポートおよびインポートできます。SAML SP および IdP のメタデータのエクスポートとインポートの機能については、次のセクションで説明します。

SAML SP のメタデータエクスポート

Citrix ADC が SAML SP として構成され、SAML IdP が Citrix ADC SP 構成を含むメタデータをインポートしたい場合を考えてみましょう。Citrix ADC アプライアンスが、SAML SP 構成を指定する「samlAction」属性ですでに構成されているとします。

ユーザーまたは管理者からメタデータをエクスポートするには、以下のように Citrix Gateway または認証仮想サーバーにクエリを実行します。

```
1 https://vserver.company.com/metadata/samlsp/<action-name>
```


SAML SP のメタデータのインポート

現在、Citrix ADC アプライアンスでの SAML アクション構成にはさまざまなパラメーターが必要です。これらは管理者が手動で指定します。ただし、異なる SAML システムとの相互運用に関しては、管理者は命名法に気付かないことがよくあります。IdP のメタデータが利用可能であれば、「samlAction」エンティティの大部分の設定を回避できます。実際、IdP メタデータファイルを指定すると、IdP 固有の設定全体が省略されることがあります。'samlAction 'エンティティは、メタデータファイルから設定を読み取るための追加パラメータを取るようになりました。

Citrix ADC アプライアンスにメタデータをインポートすると、メタデータには使用する署名アルゴリズムは含まれず、エンドポイントの詳細が含まれます。メタデータは、メタデータそのものを検証するために使用できる特定のアルゴリズムで署名できます。アルゴリズムは 'samlAction' エンティティには保存されません。

したがって、'samlAction' エンティティで指定する内容は、データを送信するときに使用されます。受信データには、Citrix ADC アプライアンスが処理する異なるアルゴリズムを含めることができます。
最大サイズは 64 K バイトのメタデータをインポートできます。

コマンドラインインターフェイスを使用してメタデータファイルをフェッチする。

```
set samlAction <name> [-metadataUrl <url> [-metadataRefreshInterval <
    int>] https://idp.citrix.com/samlidp/metadata.xml
```

> 注
>
> metadataRefreshInterval パラメータは、指定したメタデータ URL からメタデータ情報をフェッチする間隔（分）です。デフォルト値は 36000 です。

Metadata import for SAML IdP

「samlidpProfile」パラメータは、SP に固有の設定全体を読み取るために、新しい引数を取ります。SAML IdP の設定は、SP 固有のプロパティを SP メタデータファイルに置き換えることで簡略化できます。このファイルは HTTP 経由で照会されます。

コマンドラインインターフェイスを使用してメタデータファイルから読み込むには:

```
set samlIdPProfile <name> [-metadataUrl <url>] [-
    metadataRefreshInterval <int>]
```

Name-value attribute support for SAML authentication

SAML 認証属性に一意の名前と値を設定できるようになりました。名前は SAML アクションパラメータで設定され、値は名前のクエリによって取得されます。name 属性値を指定すると、管理者は属性名に関連付けられた属性値を簡単に検索できます。また、管理者は属性を値だけで覚える必要がなくなりました。

> 重要
>
> - samlAction コマンドでは、合計サイズが 2048 バイト未満で、カンマで区切って最大 64 個の属性を設定できます。
> - 属性リストを使用することをお勧めします。「属性 1 から属性 16」を使用すると、抽出された属性のサイズが大きい場合にセッションが失敗します。

CLI を使用して名前と値の属性を設定するには

コマンドプロンプトで入力します。

```
add authentication samlAction <name> [-Attributes <string>]
```

例**:**

```
add authentication samlAction samlAct1 -attributes  “mail,sn,
   userprincipalName”
```

SAML IdP に対するアサーションコンシューマーサービス **URL** のサポート

SAML ID プロバイダ（IdP）として構成された Citrix ADC アプライアンスは、SAML サービスプロバイダ（SP）要求を処理するためのアサーションコンシューマサービス（ACS）インデックス作成をサポートするようになりました。SAML IdP は、ACS インデックス設定を SP メタデータからインポートするか、ACS インデックス情報を手動で入力できるようにします。
次の表は、Citrix ADC アプライアンスが SAML SP または SAML IdP として使用される展開に固有の記事の一覧です。

次の表は、Citrix ADC アプライアンスが SAML SP または SAML IdP として使用される展開に固有の記事の一覧です。

SAML SP	SAML ID プロバイダー	情報リンク
Citrix ADC	Microsoft Azure AD	Citrix サポート
Okta	Citrix ADC	Citrix サポート
AWS	Citrix ADC	Citrix サポート

他の特定のデプロイメントに関するいくつかの情報:

- FIPS デバイス上の SAML SP としての NetScaler
- NetScaler を SAML IdP としてシングルサインオン用に Office365 を構成する

認証メカニズムに対する **WebView** クレデンシャルタイプのサポート

Citrix ADC アプライアンスの認証で AuthV3 プロトコルをサポートできるようになりました。AuthV3 プロトコルの WebView 資格情報タイプは、すべてのタイプの認証メカニズム (SAML や OAuth を含む) をサポートします。WebView 資格情報の種類は AuthV3 の一部で、Citrix Receiver およびブラウザによって Web アプリケーションに実装されます。

次の例では、Citrix Gateway および Citrix Receiver を介した WebView イベントのフローについて説明します。

1. Citrix Receiver は、AuthV3 プロトコルのサポートについて Citrix Gateway
2. Citrix ADC アプライアンスは肯定的に応答し、特定の開始 URL を提案します。
3. その後、Citrix Receiver は特定のエンドポイント（URL）に接続します。
4. Citrix Gateway は、WebView を起動するための応答をクライアントに送信します。
5. Citrix Receiver が WebView を起動し、Citrix ADC アプライアンスに初期要求を送信します。
6. Citrix ADC アプライアンスは URI をブラウザのログインエンドポイントにリダイレクトします。
7. 認証が完了すると、Citrix ADC アプライアンスは WebView に完了応答を送信します。
8. WebView が終了し、セッション確立のために AuthV3 プロトコルを続行するように Citrix Receiver に制御が戻されます。

SAML SP でのセッションインデックスサイズの増加

SAML サービスプロバイダ (SP) の SessionIndex サイズが 96 バイトに増加しました。以前は、sessionIndex のデフォルトの最大サイズは 63 バイトでした。

> 注
>
> NetScaler 13.0 ビルド 36.x で導入されたサポート

SAML SP のカスタム認証クラスリファレンスのサポート

SAML action コマンドで、カスタム認証クラス参照属性を設定できます。カスタム認証クラス参照属性を使用すると、適切な SAML タグ内のクラス名をカスタマイズできます。カスタム認証クラス参照属性と名前空間は、SAML SP 認証リクエストの一部として SAML IdP に送信されます。

以前は、SAML action コマンドを使用すると、authnctxClassRef 属性に定義されている定義済みクラスのセットのみを設定できました。

> 重要
>
> customAuthnctxClassRef 属性を設定する際には、次の点を確認してください。

- クラス名には、英数字、または適切な XML タグを持つ有効な URL を含める必要があります。

- 複数のカスタムクラスを設定する必要がある場合は、各クラスをカンマで区切る必要があります。

CLI を使用して **customAuthnCtxClassRef** 属性を設定するには

コマンドプロンプトで入力します。

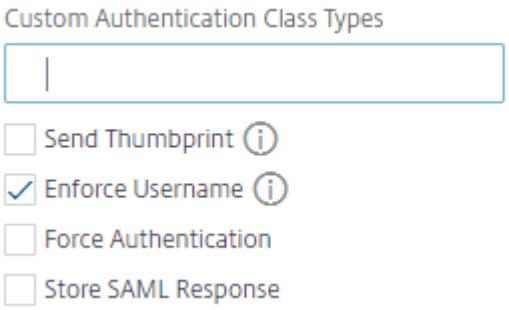
- add authentication samlAction <name> [-customAuthnCtxClassRef <string>]
- set authentication samlAction <name> [-customAuthnCtxClassRef <string>]

例:

- `add authentication samlAction samlact1 -customAuthnCtxClassRef http://www.class1.com/LoA1,http://www.class2.com/LoA2`
- `set authentication samlAction samlact2 -customAuthnCtxClassRef http://www.class3.com/LoA1,http://www.class4.com/LoA2`

GUI を使用して **customAuthnCtxClassRef** 属性を設定するには

1. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **>** **SAML** に移動します。
2. [SAML] ページで [サーバー] タブを選択し、[追加] をクリックします。
3. **[**認証 **SAML** サーバーの作成] ページで、SAML アクションの名前を入力します。
4. 下にスクロールして、[カスタム認証クラスタイプ**]** セクションのクラスタイプを設定します。



SAML IdP でのアーティファクトバインディングのサポート

SAML ID プロバイダー（IdP）として構成された Citrix ADC アプライアンスは、アーティファクトバインディングをサポートします。Artifact バインディングは SAML IdP のセキュリティを強化し、悪意のあるユーザによるアサーションの検査を制限します。

SAML IdP に対するアサーションコンシューマーサービス **URL** のサポート

SAML ID プロバイダ（IdP）として構成された Citrix ADC アプライアンスは、SAML サービスプロバイダ（SP）要求を処理するためのアサーションコンシューマサービス（ACS）インデックス作成をサポートするようになりました。SAML IdP は、ACS インデックス設定を SP メタデータからインポートするか、ACS インデックス情報を手動で入力できるようにします。

FIPS オフロードサポート

SAML サービスプロバイダーとして使用される Citrix ADC MPX FIPS アプライアンスで、暗号化されたアサーションがサポートされるようになりました。また、SAML サービスプロバイダーまたは SAML ID プロバイダーとして機能する Citrix ADC MPX FIPS アプライアンスを、FIPS ハードウェアで SHA2 アルゴリズムを使用するように構成できるようになりました。

> 注
>
> FIPS モードでは、RSA-V1_5 アルゴリズムのみがキートランスポートアルゴリズムとしてサポートされます。

コマンドラインインターフェイスを使用して **FIPS** オフロードサポートを構成します。

1. SSL FIPS を追加する

   ssl fipsKey fipsKey を追加する

2. CSR を作成し、CA サーバで使用して証明書を生成します。その後、**/nsconfig/ssl** に証明書をコピーできます。このファイルが *fips3cert.cer* であると仮定します。

   ```
   add ssl certKey fips-cert -cert fips3cert.cer -fipsKey fips-key<!--
   NeedCopy-->
   ```

3. SAML SP モジュールの SAML アクションでこの証明書を指定する

   ```
   set samlAction <name> -samlSigningCertName fips-cert<!--NeedCopy-->
   ```

4. SAML IdP モジュールの SAMLidpProfile の証明書を使用する

   ```
   set samlidpprofile fipstest -samlIdpCertName fips-cert<!--NeedCopy-->
   ```

一般的な **SAML** 用語

SAML の一般的な用語をいくつか挙げます。

- アサーション： SAML アサーションは、ユーザの認証後に ID プロバイダからサービスプロバイダに返される XML ドキュメントです。アサーションには、SAML 標準で定義されている特定の構造があります。
- アサーションのタイプ**:** アサーションのタイプは次のとおりです。
  - 認証-ユーザーは特定の時間に、特定の手段で認証されます。
  - Authorization-指定されたリソースへのアクセスを許可または拒否されたユーザー

  - Attributes-ユーザーは指定された属性に関連付けられます。

- アサーションコンシューマサービス **(ACS):** SAML アサーションの受信と解析を担当するサービスプロバイダのエンドポイント (URL)
- オーディエンス制限： SAML アサーション内の値で、アサーションの対象となるユーザー（および対象者のみ）を指定します。「オーディエンス」はサービスプロバイダーで、通常は URL ですが、技術的には任意のデータ文字列としてフォーマットできます。
- **ID** プロバイダ **(IdP):** SAML の観点では、ID プロバイダは、サービスプロバイダからの要求に応じて、ユーザの ID を検証するエンティティです。

  ID プロバイダーは、ユーザーの ID を維持および認証する責任を負います。
- サービスプロバイダ **(SP):** SAML に関しては、サービスプロバイダ (SP) はユーザーにサービスを提供し、ユーザーが SAML を使用してサインインできるようにします。ユーザーがサインインしようとすると、SP は ID プロバイダ (IdP) に SAML 認証要求を送信します。
- **SAML** バインディング**:** SAML リクエスタとレスポンダは、メッセージを交換して通信します。これらのメッセージを転送するメカニズムを SAML バインディングと呼びます。
- **HTTP Artifact:** SAML プロトコルでサポートされるバインディングオプションの 1 つ。HTTP Artifact は、SAML リクエスタとレスポンダが HTTP User-Agent を使用していて、技術的またはセキュリティ上の理由からメッセージ全体を転送したくないシナリオで役立ちます。代わりに、SAML Artifact が送信されます。SAML Artifact は、完全な情報の一意の ID です。IdP は Artifact を使用して完全な情報を取得できます。Artifact issuer は、Artifact が保留されている間、状態を維持する必要があります。Artifact 解決サービス (ARS) を設定する必要があります。

  HTTP Artifact はアーティファクトをクエリパラメータとして送信します。
- **HTTP POST:** SAML プロトコルでサポートされているバインディングオプションの 1 つ。

  HTTP POST は、メッセージコンテンツを POST パラメータとしてペイロードで送信します。
- **HTTP** リダイレクト**:** SAML プロトコルでサポートされるバインディングオプションの 1 つ。

  HTTP リダイレクトを使用すると、サービスプロバイダはログインが発生する ID プロバイダにユーザをリダイレクトし、ID プロバイダはユーザをサービスプロバイダにリダイレクトして戻します。HTTP リダイレクトには、User-Agent（ブラウザ）による介入が必要です。

  HTTP リダイレクトはメッセージコンテンツを URL で送信します。このため、レスポンスのサイズは通常、ほとんどのブラウザで許可されている URL の長さを超えるため、SAML レスポンスには使用できません。

  注： Citrix ADC アプライアンスは、ログアウト中の POST およびリダイレクトバインディングをサポートします。
- メタデータ: メタデータは、SP と IdP の相互通信方法を知るための設定データで、XML 標準になります。

SAML 認証に関連する Citrix その他の有用な記事

SAML 認証に関する次の記事が役に立つかもしれません。

- https://support.citrix.com/article/CTX277558
- https://support.citrix.com/article/CTX259127
- https://support.citrix.com/article/CTX228135
- https://support.citrix.com/article/CTX221631
- https://support.citrix.com/article/CTX138988

OAuth 認証

October 7, 2021

認証、承認、および監査のトラフィック管理機能は、OAuth および OpenID Connect（OIDC）認証をサポートします。Google、Facebook、Twitter などのアプリケーションでホストされているサービスにユーザーを認証し、認証します。

注意事項

- ソリューションが機能するには、Citrix ADC Advanced Edition 以降が必要です。
- アプライアンスが OIDC を使用して OAuthIdP として機能するには、Citrix ADC アプライアンスがバージョン 12.1 以降である必要があります。
- Citrix ADC アプライアンスの OAuth は、「OpenID コネクト 2.0」に準拠しているすべての SAML IdP に対して認定されています。

Citrix ADC アプライアンスは、SAML および OIDC を使用して、サービスプロバイダー（SP）または ID プロバイダー（IdP）として動作するように構成できます。以前は、IdP として構成された Citrix ADC アプライアンスは SAML プロトコルのみをサポートしていました。Citrix ADC 12.1 バージョン以降、Citrix ADC は OIDC もサポートします。

OIDC は OAuth の拡張機能です authorization/delegation. Citrix ADC アプライアンスは、他の認証メカニズムと同じクラスの OAuth および OIDC プロトコルをサポートします。OIDC は、ユーザー情報を収集できないトークンのみを取得する OAuth とは対照的に、承認サーバーからユーザー情報を取得する方法を提供するため、OAuth へのアドオンです。

認証メカニズムにより、OpenID トークンのインライン検証が容易になります。Citrix ADC アプライアンスは、証明書を取得し、トークンの署名を検証するように構成できます。

OAuth および OIDC メカニズムを使用する主な利点は、ユーザー情報がホストされているアプリケーションに送信されないことです。したがって、個人情報の盗難のリスクが大幅に軽減されます。

認証、承認、および監査用に構成された Citrix ADC アプライアンスは、HMACHS256 アルゴリズムを使用して署名された着信トークンを受け入れるようになりました。さらに、SAML ID プロバイダー (IdP) の公開キーは、URL エンドポイントから学習するのではなく、ファイルから読み取られます。

Citrix ADC 実装では、アプリケーションは、認証、承認、および監査トラフィック管理仮想サーバーによってアクセスされます。したがって、OAuth を設定するには、OAuth ポリシーを設定する必要があります。OAuth ポリシーは、認証、認可、および監査トラフィック管理仮想サーバーに関連付ける必要があります。

OpenIDConnect プロトコルを構成します

Citrix ADC アプライアンスは、OIDC プロトコルを使用して ID プロバイダーとして構成できるようになりました。OIDC プロトコルは、Citrix ADC アプライアンスの ID 提供機能を強化します。これで、シングルサインオンでエンタープライズ全体でホストされているアプリケーションにアクセスできます。OIDC は、ユーザーパスワードを転送しないことでセキュリティを強化しますが、特定の有効期間を持つトークンを処理します。OIDC は、アプリやサービスなどの非ブラウザークライアントと統合するようにも設計されています。したがって、多くの実装で OIDC が広く採用されています。

OpenID コネクトをサポートすることの利点

- OIDC は、ユーザーが組織全体で単一の ID を持っているため、複数の認証パスワードを維持するオーバーヘッドを排除します。
- OIDC は、パスワードが ID プロバイダーとのみ共有され、アクセスするアプリケーションとは共有されないため、パスワードに堅牢なセキュリティを提供します。
- OIDC は、さまざまなシステムとの相互運用性が非常に高いため、ホストされているアプリケーションが OpenID を簡単に受け入れることができます。
- OIDC は、ネイティブクライアントがサーバーと簡単に統合できるようにするシンプルなプロトコルです。

GUI を使用して **OpenIDConnect** プロトコルを使用して **Citrix ADC** アプライアンスを **IdP** として構成するには

1. **[**設定**] > [**セキュリティ**] > [AAA** アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**高度なポリシー**] > [OAuth IdP**] に移動します。
2. **[**プロファイル**]** をクリックし、**[**追加**]** をクリックします。

   「認証 **OAuth IDP** プロファイルの作成」画面で、次のパラメータの値を設定し、「作成」をクリックします。

   - **[Name**]：認証プロファイルの名前。
   - クライアント **ID**： SP を識別する一意の文字列。
   - クライアント・シークレット：SP を識別する一意のシークレット。
   - リダイレクト **URL**： コード/トークンを投稿する必要がある SP 上のエンドポイント。
   - 発行者名 — IdP を識別する文字列。
   - 対象者 — IdP によって送信されるトークンのターゲット受信者。これは、受信者が確認することがあります。
   - **Skew Time** — トークンが有効である時間。
   - デフォルトの認証グループ – ポリシーの評価を簡素化し、ポリシーのカスタマイズを支援するために、このプロファイルのセッションに追加されるグループ。

3. **[**ポリシー**]** をクリックし、**[**追加**]** をクリックします。
4. 「認証 **OAuth IDP** ポリシーの作成」画面で、次のパラメータの値を設定し、「作成」をクリックします。
   - **Name —** 認証ポリシーの名前。
   - アクション –以前に作成されたプロファイルの名前。
   - ログアクション– リクエストがこのポリシーに一致する場合に使用するメッセージログアクションの名前。提出必須ではありません。
   - 未定義の結果アクション：ポリシー評価の結果が未定義（UNDEF）の場合に実行するアクション。必須フィールドではありません。
   - 式 –ポリシーが特定の要求に応答するために使用するデフォルトの構文式。たとえば、true です。
   - **[Comments]**： ポリシーに関するコメント。

OAuthIdP ポリシーと **LDAP** ポリシーを認証仮想サーバーにバインドする

1. **[**設定**] > [**セキュリティ**] > [AAA** アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**高度なポリシー**] > [**アクション**] > [LDAP]** に移動します。
2. [**LDAP** アクション] 画面で、[追加] をクリックします。
3. 「認証 **LDAP** サーバーの作成」画面で、次のパラメーターの値を設定し、「作成」をクリックします。
   - 名前 –LDAP アクションの名前
   - サーバー名**/**サーバー **IP —** LDAP サーバーの FQDN または IP を指定します。
   - [セキュリティタイプ**]**、**[**ポート**]**、**[**サーバタイプ**]**、**[**タイムアウト] に適切な値を選択します。
   - [認証**]** がオンになっていることを確認します。
   - ベース **DN**： LDAP 検索を開始するベース。たとえば、dc=aaa、dc= ローカルです。
   - 管理者バインド **DN**：LDAP サーバーへのバインドのユーザー名。たとえば、admin@aaa.local です。
   - 管理者パスワード**/**パスワードの確認**:LDAP** をバインドするためのパスワード
   - [接続のテスト] をクリックして、設定をテストします。
   - サーバーのログオン名属性**:**「**sAM** アカウント名」を選択します。
   - その他のフィールドは必須ではないため、必要に応じて設定できます。
4. **[**設定**] > [**セキュリティ**] > [AAA** アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**高度なポリシー**] > [**ポリシー**]** に移動します。
5. [認証ポリシー] 画面で、[追加] をクリックします。
6. **[**認証ポリシーの作成**]** ページで、次のパラメータの値を設定し、**[**作成**]** をクリックします。
   - 「名前」**—** LDAP 認証ポリシーの名前。
   - アクションタイプ**— [LDAP]** を選択します。
   - 「アクション」**—** LDAP アクションを選択します。
   - **Expression —** ポリシーが特定の要求に応答するために使用するデフォルトの構文式。たとえば、true** です。

CLI を使用して **OpenIDConnect** プロトコルを使用して **Citrix ADC** アプライアンスを **IdP** として構成するには

コマンドプロンプトで、次のコマンドを入力します。

- add authentication OAuthIDPProfile <name> [-clientID <string>][-clientSecret][-redirectURL <URL>][-issuer <string>][-audience <string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]<!--NeedCopy-->
- add authentication OAuthIdPPolicy <name> -rule <expression> [-action <string> [-undefAction <string>] [-comment <string>][-logAction <string >]<!--NeedCopy-->
- add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -ldapBase ”dc=aaa,dc=local”<!--NeedCopy-->
- ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -ldapLoginName sAMAccountName<!--NeedCopy-->
- add authentication policy aaa-ldap-adv-pol -rule **true** -action aaa-ldap-act<!--NeedCopy-->
- bind authentication vserver auth_vs -policy <ldap_policy_name> -priority 100 -gotoPriorityExpression NEXT<!--NeedCopy-->
- bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -priority 5 -gotoPriorityExpression END<!--NeedCopy-->
- bind vpn global -certkey <><!--NeedCopy-->

> 注
>
> 複数のキーをバインドできます。バインドされた証明書のパブリック部分は、jwks_uri query (https://gw/oauth/idp/certs)への応答として送信されます。

OAuth SP としての Citrix ADC

April 25, 2022

認証、承認、および監査トラフィック管理機能は、Google、Facebook、Twitter などのアプリケーションでホストされているアプリケーションに対してユーザーを認証するための OAuth 認証をサポートします。

注意事項

- ソリューションが機能するには、Citrix ADC Advanced Edition 以上が必要です。
- Citrix ADC アプライアンス上の OAuth は、「OpenID コネクト 2.0」に準拠しているすべての SAML IdP に対して認定されています。

> 重要:
>
> セッションの有効期限が切れると、コンテンツが多い Web サイトが複数の認証要求を送信すると、Citrix ADC アプライアンスが CSRF エラーで応答することがあります。回避策として、OAuth ポリシーを設定するときに、メインエントリポイントであるホスト名とパスの両方に対してポリシーが設定されていることを確認することをお勧めします。

GUI を使用して **OAuth** を設定する

1. OAuth アクションとポリシーを設定します。

   [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**ポリシー] **>** **[**認証] **>** **[**高度なポリシー] > [ポリシー] に移動し、アクションタイプとして OAuth を使用してポリシーを作成し、必要な OAuth アクションをポリシーに関連付けます。

2. OAuth ポリシーを認証仮想サーバーに関連付けます。

   [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**仮想サーバ] に移動し、OAuth ポリシーを認証仮想サーバに関連付けます。

> 注:
>
> 属性 (1 ～16) は OAuth レスポンスで抽出できます。現在、これらの属性は評価されません。これらは将来の参照のために追加されます。

CLI を使用して **OAuth** を設定する

1. OAuth アクションを定義します。

```
1 add authentication OAuthAction <name> -authorizationEndpoint <URL>
      -tokenEndpoint <URL> [-idtokenDecryptEndpoint <URL>] -clientID
      <string> -clientSecret <string> [-defaultAuthenticationGroup <
     string>][-tenantID <string>][-GraphEndpoint <string>][-
     refreshInterval <positive_integer>] [-CertEndpoint <string>][-
     audience <string>][-userNameField <string>][-skewTime <mins>][-
     issuer <string>][-Attribute1 <string>][-Attribute2 <string>][-
     Attribute3 <string>]
2 <!--NeedCopy-->
```

2. アクションを高度な認証ポリシーに関連付けます。

```
1 add authentication Policy <name> -rule <expression> -action <
     string>
```



```
<!--NeedCopy-->
```

例:

```
add authentication oauthAction a -authorizationEndpoint https://
    example.com/ -tokenEndpoint https://example.com/ -clientiD sadf
     -clientsecret df
<!--NeedCopy-->
```

認証 OAuthAction パラメータの詳細については、「認証 OAuthAction」を参照してください。

> 注:
>
> CertendPoint が指定されると、Citrix ADC アプライアンスは構成された頻度でそのエンドポイントをポーリングしてキーを学習します。

ローカルファイルを読み取り、そのファイルからキーを解析するように Citrix ADC を構成するために、次のような新しい構成オプションが導入されました。

```
set authentication OAuthAction <> -CertFilePath <path to local file
    with jwks>
<!--NeedCopy-->
```

OAuth 機能は、証明書利用者側（RP）側および Citrix Gateway および Citrix ADC の IdP 側からのトークン API で次の機能をサポートするようになりました。

- PKCE（コード交換のための証明キー）のサポート
- client_assertion のサポート

OAuth 認証に対する名前と値の属性のサポート

OAuth 認証属性に一意の名前と値を設定できるようになりました。名前は OAuth アクションパラメーターで「Attributes」として設定され、名前はクエリーによって取得されます。抽出された属性は、認証、認可、および監査セッションに保存されます。管理者は、選択した属性名の指定方法に基づいて、`http.req.user.attribute("attribute name")`または`http.req.user.attribute(1)`を使用して、これらの属性をクエリできます。

属性の名前を指定することで、管理者はその属性名に関連付けられている属性値を簡単に検索できます。また、管理者は「attribute1 to attribute16」を番号だけで覚えておく必要がなくなりました。

> 重要
>
> OAuth コマンドでは、最大 64 個の属性をカンマで区切って設定でき、合計サイズは 1024 バイト未満です。
>
> 注
>
> 「属性 1 から属性 16」の合計値サイズと「属性」で指定された属性の値が 10 KB 以下であれば、セッションの失敗を回避できます。

CLI を使用して名前と値の属性を設定するには

コマンドプロンプトで入力します。

```
add authentication OAuthAction <name> [-Attributes <string>]

set authentication OAuthAction <name> [-Attributes <string>]
<!--NeedCopy-->
```

例:

```
add authentication OAuthAction a1  - attributes ”email,company”  -
    attribute1 email

set authentication OAuthAction oAuthAct1 -attributes ”mail,sn,
    userprincipalName”
<!--NeedCopy-->
```

OAuth IdP としての Citrix ADC

July 15, 2022

Citrix ADC アプライアンスは、OpenID-Connect（OIDC）プロトコルを使用して、アイデンティティプロバイダとして構成できるようになりました。OIDC プロトコルは、Citrix ADC アプライアンスのアイデンティティ提供機能を強化します。OIDC は、ユーザーパスワードを転送せず、特定の有効期間を持つトークンを使用することで、セキュリティを強化するため、シングルサインオンでエンタープライズワイドホストアプリケーションにアクセスできるようになりました。OpenID は、アプリやサービスなどのブラウザ以外のクライアントと統合するように設計されています。したがって、OIDC プロトコルは多くの実装で広く採用されています。

> 注:
>
> アプライアンスが OIDC プロトコルを使用して OAuth IdP として機能するには、Citrix ADC がバージョン 12.1 以降である必要があります。

Citrix ADC を OAuth IdP として持つ利点

- ユーザーが組織全体で 1 つの ID を持つため、複数の認証パスワードを維持するオーバーヘッドを排除します。
- パスワードは ID プロバイダでのみ共有され、アクセスするアプリケーションとは共有されないため、パスワードの堅牢なセキュリティを提供します。
- さまざまなシステムとの膨大な相互運用性を提供し、ホストされたアプリケーションが OpenID を受け付けやすくなります。

> 注
>
> ソリューションが機能するには、Citrix ADC Advanced Edition 以上が必要です。

GUI を使用して Citrix ADC アプライアンスを OAuth IdP として構成するには

1. [構成] **> [**セキュリティ**] > [AAA** アプリケーショントラフィック**] > [**ポリシー**] > [**認証**] > [**高度なポリシー**] > [OAuth IdP]** に移動します。
2. [プロファイル] をクリックし、[追加**]** をクリックします。

   [認証 **OAuth IDP** プロファイルの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。

   - **Name**： 認証プロファイルの名前。文字、数字、またはアンダースコア文字（_）で始まり、文字、数字、ハイフン（-)、ピリオド（.）ポンド（#)、スペース（)、アットマーク（@)、等号（=)、コロン（:)、およびアンダースコア文字のみを含める必要があります。プロファイルの作成後は変更できません。
   - クライアント **ID**： SP を識別する一意の文字列。承認サーバーは、この ID を使用してクライアントの構成を推測します。最大文字数：127。
   - **Client Secret** - ユーザーと承認サーバーによって確立されたシークレット文字列。最大長:239
   - リダイレクト **URL** — コード/トークンのポスト先となる SP 上のエンドポイント。
   - [**Issuer Name**]：トークンを受け付けるサーバーの ID。最大長:127
   - オーディエンス — IdP によって送信されるトークンのターゲット受信者。これは受信者がチェックするかもしれません。
   - スキュー時間：このオプションは、Citrix ADC が受信トークンで許可するクロックスキューを分単位で指定します。たとえば、SkewTime が 10 の場合、トークンは (現在の時刻-10) 分から (現在時刻 + 10) 分、つまり 20 分まで有効です。デフォルト値:5。
   - デフォルト認証グループ：nFactor フローで使用できる IdP によってこのプロファイルが選択されたときに、セッション内部グループリストに追加されるグループ。これは、次の認証ポリシーの式

(AAA.USER.IS_MEMBER_OF (「xxx」)) で使用できます。
証明書利用者に関連する nFactor フローを特定します。最大長：63

A group added to the session for this profile to simplify policy evaluation and help in customizing policies. This is the default group that is chosen when the authentication succeeds in addition to the extracted groups. Maximum Length: 63.

3. **[**ポリシー**]** をクリックし、**[**追加**]** をクリックします。
4. [認証 **OAuth IDP** ポリシーの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。
   - **[** Name]：認証ポリシーの名前。
   - **Action** - 以前に作成されたプロファイルの名前。
   - **Log Action** - 要求がこのポリシーに一致する場合に使用するメッセージログアクションの名前。必須の提出ではありません。
   - **Undefined-Result** アクション：ポリシー評価の結果が未定義（UNDEF）である場合に実行するアクション。必須フィールドではありません。
   - **Expression** - ポリシーが特定の要求に応答するために使用するデフォルトの構文式。たとえば、true です。
   - コメント — ポリシーに関するコメント。

OAuthIdP ポリシーと **LDAP** ポリシーを認証仮想サーバーにバインドする

1. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **> LDAP** に移動します。
2. [**LDAP** アクション] 画面で、[追加] をクリックします。
3. [認証 **LDAP** サーバーの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。
   - **Name —** LDAP アクションの名前
   - サーバ名**/**サーバ **IP —** LDAP サーバの FQDN または IP を指定します。
   - [セキュリティタイプ**]**、**[**ポート**]**、**[**サーバータイプ**]**、**[**タイムアウト] の適切な値を選択する
   - [認証**]** がオンになっていることを確認します
   - ベース **DN**： LDAP 検索を開始する基本。例えば、dc=aaa、dc=local。
   - 管理者バインド **DN:** LDAP サーバーへのバインドのユーザー名。たとえば、admin@aaa.local。
   - 管理者パスワード**/**パスワードの確認**:LDAP** をバインドするためのパスワード
   - [接続のテスト] をクリックして、設定をテストします。
   - サーバーログオン名属性**:**「**samAccountName**」を選択します。
   - その他のフィールドは必須ではないため、必要に応じて設定できます。
4. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** ポリシーにナビゲートして下さい。
5. [認証ポリシー] 画面で、[追加] をクリックします。

6. [認証ポリシーの作成] ページで、次のパラメータの値を設定し、[作成] をクリックします。

   - **Name —** LDAP 認証ポリシーの名前。
   - [アクションタイプ] — [**LDAP**] を選択します。
   - [アクション] — LDAP アクションを選択します。
   - **Expression —** ポリシーが特定のリクエストに応答するために使用するデフォルトの構文式。たとえば、true** です。

OAuth 機能は、証明書利用者側（RP）側および Citrix Gateway および Citrix ADC の IdP 側からのトークン API で次の機能をサポートするようになりました。

- PKCE（コード交換のための証明キー）のサポート
- client_assertion のサポート

CLI を使用して **OIDC** プロトコルを使用して **Citrix ADC** アプライアンスを **IdP** として構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]

add authentication OAuthIdPPolicy <name> -rule <expression> [-action <
    string> [-undefAction <string>] [-comment <string>][-logAction <
    string>]

add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -
    ldapBase ”dc=aaa,dc=local”

ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -
    ldapLoginName sAMAccountName

add authentication policy aaa-ldap-adv-pol -rule true -action aaa-ldap-
    act

bind authentication vserver auth_vs -policy <ldap_policy_name> -
    priority 100 -gotoPriorityExpression NEXT

bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -
    priority 5 -gotoPriorityExpression END

bind vpn global  - certkey <>
<!--NeedCopy-->
```


> 注
>
> - 複数のキーをバインドできます。バインドされた証明書のパブリック部分は、jwks_uri query (https://gw/oauth/idp/certs)への応答として送信されます。
> - Citrix ADC 13.0—85.19 リリース以降、OAuth IdP イントロスペクティブエンドポイントはこのプロパティactive: **true**をサポートします

OIDC プロトコルでの暗号化されたトークンのサポート

OIDC メカニズムを備えた Citrix ADC アプライアンスは、暗号化されたトークンの署名付きトークンの送信をサポートするようになりました。Citrix ADC アプライアンスは、JSON Web 暗号化仕様を使用して暗号化されたトークンを計算し、暗号化されたトークンのコンパクトなシリアル化のみをサポートします。OpenID トークンを暗号化するには、Citrix ADC アプライアンスに証明書利用者（RP）の公開キーが必要です。公開鍵は、依存パーティの既知の構成エンドポイントをポーリングすることによって動的に取得されます。

新しい「relyingPartyMetadataURL」オプションが「認証 OAuthidpProfile」プロフィールに導入されました。

CLI を使用して証明書利用者のエンドポイントを設定するには

コマンドプロンプトで入力します。

“‘set authentication OAuthIDPProfile [-relyingPartyMetadataURL] [-refreshInterval] [-status < >]

```
- **relyingPartyMetadataURL** : Citrix ADC IdPが構成されている証明書利用者に関する詳細を取得できるエンドポイント。メタデータ応答には、RP公開キーの jwks_uri のエンドポイントを含める必要があります。

- **refreshInterval** -証明書を分単位で更新するために、このエンドポイントをポーリングする必要があるレートを定義します。

- **status** -ポーリング操作のステータスを反映します。Citrix ADCアプライアンスが公開鍵を正常に取得すると、ステータスは完了します。

 **例**

 ```
 set authentication OAuthIDPProfile sample_profile -
    relyingPartyMetadataURL https://rp.customer.com/metadata -
    refreshInterval 50 -status < >
 <!--NeedCopy-->
```

エンドポイントを構成すると、Citrix ADC アプライアンスは最初に証明書利用者の既知のエンドポイントをポーリングして構成を読み込みます。現在、Citrix ADC アプライアンスは「jwks_uri」エンドポイントのみを処理します。


```

- 応答に「jwks_uri」が存在しない場合、プロファイルのステータスは完了しません。
- 応答に「jwks_uri」が存在する場合、Citrix ADC はそのエンドポイントをポーリングして、依存パーティの公開鍵を読み取ります。

> **注:** トークンの暗号化では、RSAES-OAEP および AES GCM 暗号化タイプのアルゴリズムのみがサポートされます。

### OpenID コネクトでのカスタム属性のサポート

OpenID 証明書利用者は、ユーザープロファイルの作成または承認の決定を行うために、トークンにユーザー名またはユーザープリンシパル名 (UPN) 以上のものを要求する場合があります。ほとんどの場合、ユーザーグループはユーザーに認可ポリシーを適用する必要があります。場合によっては、ユーザーアカウントのプロビジョニングに、姓や姓などの詳細が必要になることがあります。

IdP として構成された Citrix ADC アプライアンスは、式を使用して OIDCid_token に追加の属性を送信するために使用できます。高度なポリシー式は、要件に従ってカスタム属性を送信するために使用されます。Citrix IdP は、属性に対応する式を評価し、最終トークンを計算します。

Citrix ADC アプライアンスは出力データを自動的に JSONify します。たとえば、数値 (SSN など) やブール値 (true または false) は引用符で囲まれません。グループなどの複数値を持つ属性は、配列マーカー (「[」と「]」) 内に配置されます。複合型属性は自動的に計算されず、要件に従ってこれらの複合値の PI 式を設定できます。

#### CLI を使用して証明書利用者のエンドポイントを設定するには

コマンドプロンプトで入力します。

```
1 set oauthidpprofile <name> -attributes <AAA-custom-attribute-pattern>
2 <!--NeedCopy-->
```

<AAA-custom-attribute-pattern>は、次のように記述できます。

Attribute1=PI-Expression@@@attribute2=PI-Expression@@@

「attribute1」,「attribute2」は、id_token に挿入される属性の名前を表すリテラル文字列です。

> **注:** 最大 2,000 バイトの属性を設定できます。

例: `set oauthidpprofile sample_1 -attributes q{ myname=http.req.user.name@@@ssn ="123456789"@@@jit="false"@@@groups=http.req.user.groups }`

- 先行する PI 式は、属性に対して使用される値を表す高度なポリシー式です。PI 式は、「' ハードコードされた文字列」' などの文字列リテラルを送信するために使用できます。文字列リテラルは、一重引用符で囲まれた二重引用符、または開始パターンとパターンの周りを二重引用符で囲みます（前述のように、開始パターンは“q{“です）。属性の値が文字列リテラルでない場合、式は実行時に評価され、その値はトークンで送信されます。実行時の値が空の場合、対応する属性は ID トークンに追加されません。

- 例で定義されているように、「false」は属性「jit」のリテラル文字列です。また、`ssn`には参照用にハードコードされた値があります。グループと`myname`は文字列を生成する PI 式です。

Citrix Gateway でのアクティブ/アクティブ GSLB 展開のサポート

OIDC プロトコルを使用してアイデンティティプロバイダ（IdP）として構成された Citrix Gateway は、アクティブ/アクティブ GSLB 展開をサポートできます。Citrix Gateway IdP でのアクティブ/アクティブ GSLB 展開では、複数の地理的ロケーションで受信したユーザーログイン要求を負荷分散できます。

> 重要
>
> セキュリティを強化するために、CA 証明書を SSL サービスにバインドし、SSL サービスで証明書検証を有効にすることをお勧めします。

GSLB セットアップの設定の詳細については、GSLB のセットアップと設定の例を参照してください。

Citrix ADC アプライアンスを使用した API 認証

October 7, 2021

最新のアプリケーションがクライアントとやり取りする方法には、パラダイムシフトがあります。従来、ブラウザクライアントはサービスにアクセスするために使用されていました。アプリケーションは、通常、セッション cookie を設定してユーザーコンテキストを追跡します。最新の分散アプリケーションでは、マイクロサービス間でユーザーセッションを維持することが難しくなっています。このため、ほとんどのアプリケーションアクセスは API ベースになっています。

これらの分散サービスと通信するクライアントも進化してきました。ほとんどのクライアントは、認証サーバーと呼ばれる信頼されたエンティティからトークンを取得し、ユーザーの身元とアクセスを証明します。これらのクライアントは、各アクセス要求でアプリケーションにトークンを提示します。したがって、Citrix ADC のような従来のプロキシデバイスは、これらのクライアントをサポートするために進化する必要があります。Citrix ADC アプライアンスは、管理者がこのようなトラフィックを処理する方法を提供します。Citrix ADC は、公開サービス宛てのすべてのトラフィックをフロントエンドに API ゲートウェイとして展開できます。API ゲートウェイは、従来型（ハイブリッドマルチクラウドまたは HMC）またはクラウドネイティブ環境にデプロイできます。API Gateway は、認証、認可、レート制限、ルーティング、キャッシュ、SSL オフロード、アプリケーションファイアウォールなどのいくつかのサービスを提供するために、すべてのインバウンドトラフィックを終了します。したがって、インフラストラクチャでは重要なコンポーネントになります。

トークンの種類

API アクセス中に交換されるトークンは、主に OAuth/OpenID Connect (OIDC) プロトコルに準拠します。「委任アクセス」にのみ使用されるアクセストークンは、OAuth プロトコルに準拠しますが、OIDC に準拠する ID トークンは、ユーザー情報も持ちます。

アクセストークンは、通常、不透明なまたはランダムなデータのブロブです。ただし、JWT（Json Web Token）標準に準拠した署名トークンであることがあります。ID トークンは常に署名された JWT です。

OAuth を使用した API アクセス

Citrix ADC アプライアンスでの OAuth 認証タイプは、OAuth プロトコルと OIDC プロトコルの両方を処理するために使用できます。OIDC は、OAuth プロトコルの拡張です。

Citrix ADC アプライアンス上の OAuthAction は、ブラウザなどの対話型クライアントやクライアントアプリなどのネイティブクライアントを処理するために使用できます。対話型クライアントは、OIDC プロトコルを使用してログインするために ID プロバイダーにリダイレクトされます。ネイティブクライアントは帯域外トークンを取得し、Citrix ADC アプライアンスでこれらのトークンを提示してアクセスすることができます。

> 注:
>
> エンドポイントから取得したアクセストークンは、後続のリクエストのためにキャッシュできるため、API のパフォーマンスが向上します。

コマンドラインインターフェイスを使用してトークンキャッシュサポートを構成するには、コマンドプロンプトで次のコマンドを入力します。

```
set aaaparameter  - apITokenCache <ENABLED>

<!--NeedCopy-->
```

次のセクションでは、ネイティブクライアントによって実行される API アクセスメソッドについて説明します。

API アクセス用の仮想サーバー

API アクセス用に Citrix ADC アプライアンスを展開するには、401 認証を使用してトラフィック管理（TM）仮想サーバーを展開します。これは、認証（認証、認可、監査）仮想サーバーに関連付けられ、認証ポリシーとセッションポリシーを保持します。設定スニペットに続いて、そのような仮想サーバーを 1 つ作成します。

```
Add lb vserver lb-api-access SSL <IP> 443 -authn401 On -AuthnVsName
    auth-api-access

Bind ssl vserver lb-api-access -certkeyName <ssl-cert-entity>

Add authentication vserver auth-api-access SSL
<!--NeedCopy-->
```


> 注:
>
> 構成を完了するには、サービスを TM vserver にバインドし、認証ポリシー（以下に説明する OAuthAction を使用）を認証仮想サーバーにバインドする必要があります。

仮想サーバーを作成した後、対応するポリシーとともに OAuthAction を追加する必要があります。OAuth アクションには、トークンの種類やその他のセキュリティメカニズムに応じて、他にもいくつかのオプションがあります。

ID トークンの OAuth 設定

ID トークンは常に署名された JWT です。つまり、ヘッダー、ペイロード、および署名を運びます。これらは自己完結型のトークンであるため、Citrix ADC アプライアンスはこれらのトークンをローカルで検証できます。これらのトークンを検証するには、アプライアンスは、これらのトークンの署名に使用される対応する秘密キーの公開鍵を知っている必要があります。

以下は、「certEndpoint」と一緒に特定の必須引数を持つ OAuthAction の例です。

```
1 Add authentication OAuthAction oauth-api-access -clientid <your-client-
    id> -clientsecret <your-client-secret> -authorizationEndpoint <URL
    to which users would be redirected for login> -tokenEndpoint <
    endpoint at which tokens could be obtained> -certEndpoint <uri at
    which public keys of IdP are published>
2 <!--NeedCopy-->
```

各項目の意味は次のとおりです。

- クライアント **ID**： SP を識別する一意の文字列。認可サーバは、この ID を使用してクライアント設定を推測します。最大長さ:127。
- **Client Secret** — ユーザーと認可サーバーによって確立されたシークレット文字列。最大長さ:239。
- **AuthorizationEndpoint** -ユーザーが通常ログインする URL（対話型クライアントを使用する場合）。
- **TokenEndpoint** -トークン/コードを取得/交換する認可サーバ上の URL
- **CertendPoint** -承認サーバーがトークンの署名に使用する公開キーを発行する URL。認可サーバーは複数の鍵を公開し、そのうちの 1 つを選択してトークンに署名できます。

> 注： Client ID/Client Secret/authorizationEndpoint/TokenEndpoint は、API アクセスのオプションのパラメータです。ただし、アクションエンティティはさまざまな目的に再利用できるため、これらのパラメータに値を指定することをお勧めします。

前述の構成では、「certEndpointpoint」は、ID トークンの検証に不可欠です。このエンドポイントには、トークンの署名に使用される証明書の公開キーが含まれています。これらの公開鍵は、JWK (Json Web Keys) 仕様に対応している必要があります。

Citrix ADC アプライアンスで CertendPoint を構成すると、公開キーを最新の状態に保つために、エンドポイントを定期的にポーリングします（デフォルト間隔は 1 日で、構成でカスタマイズできます）。公開鍵が利用可能になった後、ADC は着信 ID トークンのローカル検証を実行できます。

不透明なアクセストークンの **OAuth** 設定

不透明なトークンは、Citrix ADC アプライアンスではローカルで検証できません。これらは、認証サーバー上で検証する必要があります。Citrix ADC アプライアンスは、これらのトークンを検証するために、OAuth 仕様に記載されている「イントロスペクションプロトコル」を使用します。OAuth 設定では、不透明なトークンを検証するための新しいオプション IntroSpecturl が提供されています。

```
set oauthAction oauth-api-acccess -introspectURL <uri of the
    Authorization Server for introspection>

<!--NeedCopy-->
```

イントロスペクション API の形式は、次のようにhttps://tools.ietf.org/html/rfc7662##section-2.1の仕様に準拠しています:

```
POST /introspect HTTP/1.1
Host: server.example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
token=mF_9.B5f-4.1JqM&token_type_hint=access_token

<!--NeedCopy-->
```

認証仮想サーバーへのポリシーのバインド

OAuthAction が作成されると、それを呼び出すための対応するポリシーを作成する必要があります。

```
add authentication policy oauth-api-access -rule <> -action <oauth-api-
    access><!--NeedCopy-->
```

bind authentication vserver auth-api-access -policy oauth-api-access -pri 100


```
## Citrix ADCアプライアンスの追加のセキュリティ設定

トークン検証には、トークンの有効期間チェックが含まれます。許容時間外のトークンは拒否されます。以下は、追加のセキュリティのための追加設定です。これらのいくつかは、常に設定することをお勧めします。

**対象ユーザー**: OAuth アクションは、トークンの意図した受信者で構成できます。すべてのトークンは、この設定された URL と照合されます。Citrix ADCアプライアンスには、オーディエンスフィールドが実際にアプライアンスのパターンセットを指す追加機能があります。このパターンセットを使用して、管理者はオーディエンスに複数の URL を設定できます。

<!--NeedCopy-->
```

add policy patset oauth_audiences

bind patset oauth_audiences https://app1.company.com

bind patset oauth_audiences https://app2.company.com

bind patset oauth_audiences httpsL//app1.company.com/path1

set oAuthAccess oauth-api-access -audience oauth_audiences

```
前の例では、1 つのパターンセットに複数のオーディエンスが指定されています。したがって、着信トークンは、パターンセット内に設定された URL のいずれかが含まれている場合にのみ許可されます。

**発行者**: トークンが受け入れられるサーバーの ID。 最大長さ:127。OAuthアクションでトークンの発行者を設定することをお勧めします。これにより、誤った認可サーバによって発行されたトークンが許可されないようになります。

**SkewTime**: Citrix ADCアプライアンスが受信トークンに対して許可するクロックスキューを分単位で指定します。たとえば、SkewTimeが10の場合、トークンは（現在の時間-10）分から（現在の時間+10）分まで、つまり20分の範囲で有効です。デフォルト値: 5

**AllowedAlgorithms**: このオプションを使用すると、管理者は受信トークン内の特定のアルゴリズムを制限できます。デフォルトでは、サポートされているすべてのメソッドが許可されています。ただし、このオプションを使用して制御することができます。

次の構成では、RS256 および RS512 を使用するトークンだけが許可されます。

```



```
<!--NeedCopy-->
```

set oAuthAction oauth-api-access -allowedAlgorithms RS256 RS512

```
上記の設定後、RS256とRS512を使用するトークンだけが許可されます。

## 認証からの特定のトラフィックのバイパス

多くの場合、クライアントがパブリックにアクセス可能な検出APIがいくつかあります。これらのAPIは、通常、サービス自体の構成と機能を明らかにします。管理者は、次のように「認証なし」ポリシーを使用して、これらのメタデータURLからの認証をバイパスするようにCitrix ADCアプライアンスを構成できます。

<!--NeedCopy-->
```

add authentication policy auth-bypass-policy -rule <> -action NO_AUTHN

bind authentication vserver auth-api-access -policy auth-bypass-policy -pri 110

```
NO_AUTHNは、ルールが一致したときに認証が完了する暗黙のアクションです。APIアクセスの範囲を超えて、NO_AUTHNアクションの他の用途があります。
<!--NeedCopy-->
```

LDAP 認証

April 7, 2022

他の種類の認証ポリシーと同様に、ライトウェイトディレクトリアクセスプロトコル (LDAP) 認証ポリシーは式とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバーにバインドし、プライオリティを割り当てます。バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。LDAP では、標準の認証機能に加えて、他の Active Directory (AD) サーバーで、ローカルに存在しないユーザーのユーザーアカウントを検索できます。この機能は、紹介サポートまたは紹介追跡と呼ばれます。

通常、認証時に認証サーバーの IP アドレスを使用するように Citrix ADC を構成します。LDAP 認証サーバーでは、IP アドレスの代わりに LDAP サーバーの FQDN を使用してユーザーを認証するように ADC を設定することもできます。FQDN を使用すると、認証サーバが複数の IP アドレスのいずれかにあっても、常に 1 つの FQDN を使用する環境で、より複雑な認証、認可、および監査の設定を簡素化できます。IP アドレスではなくサーバーの FQDN を使用して認証を構成するには、認証アクションの作成時を除き、通常の構成プロセスに従います。アクションを作成す

るときは、ServerIP パラメータの代わりに **ServerName** パラメータを使用し、**IP** アドレスの代わりにサーバの FQDN を使用します。

LDAP サーバの IP または FQDN を使用してユーザを認証するように ADC を設定する前に、IP アドレスではなく FQDN に対して認証するように認証、承認、および監査を設定すると、認証プロセスに追加の手順が追加されることを考慮してください。ADC はユーザーを認証するたびに、FQDN を解決する必要があります。非常に多くのユーザーが同時に認証を試みると、DNS ルックアップの結果として認証プロセスが遅くなる可能性があります。

LDAP リフェラルサポートはデフォルトで無効になっており、グローバルに有効にすることはできません。LDAP アクションごとに明示的に有効にする必要があります。AD サーバーが、参照 (GC) サーバーで使用されているものと同じ`binddn credentials`を受け入れることを確認します。紹介サポートを有効にするには、紹介をフォローするように LDAP アクションを設定し、フォローする紹介の最大数を指定します。

紹介サポートが有効になっていて、Citrix ADC が要求に対する LDAP_REFERRAL 応答を受信すると、認証、承認、および監査は、参照に含まれる Active Directory (AD) サーバーへの参照に従い、そのサーバーで更新を実行します。まず、認証、承認、および監査は DNS 内の参照サーバーを検索し、そのサーバーに接続します。紹介ポリシーで SSL/TLS が必要な場合は、SSL/TLS 経由で接続します。次に、前のサーバで使用していた`binddn credentials`を使用して新しいサーバにバインドし、参照を生成したオペレーションを実行します。この機能はユーザーには透過的です。

LDAP 接続のポート番号は次のとおりです。

- 389 (セキュリティで保護されていない LDAP 接続の場合) (プレーンテキスト LDAP の場合)
- 636 セキュアな LDAP 接続 (SSL LDAP の場合)
- 3268 (Microsoft のセキュリティで保護されていない LDAP 接続の場合) (プレーンテキストのグローバルカタログサーバー用)
- 3269 (Microsoft セキュリティで保護された LDAP 接続の場合) (SSL グローバルカタログサーバー用)

次の表に、LDAP サーバのユーザ属性フィールドの例を示します。

LDAP サーバ	ユーザー属性	大文字と小文字を区別
Microsoft Active Directory サーバー	sAMAccountName	いいえ
Novell eDirectory	ou	はい
IBM Directory Server	uid	はい
Lotus Domino	CN	はい
Sun ONE ディレクトリ (旧 iPlanet)	uid か cn	はい

次の表に、ベース DN の例を示します。

LDAP サーバ	ベース DN
Microsoft Active Directory サーバー	DC=citrix、DC= ローカル
Novell eDirectory	ou=users、ou=dev
IBM Directory Server	cn=users
Lotus Domino	OU=City、O=Citrix、C=US
Sun ONE ディレクトリ (旧 iPlanet)	OU=People、dc=citrix、dc=com

次の表に、バインド DN の例を示します。

LDAP サーバ	バインド DN
Microsoft Active Directory サーバー	cn=Administrator、cn=Users、DC=citrix、DC=local
Novell eDirectory	cn=admin、o=citrix
IBM Directory Server	LDAP_dn
Lotus Domino	cn=Notes Administrator、O=Citrix、C=US
Sun ONE ディレクトリ (旧 iPlanet)	uid=admin,ou=Administrators, ou=TopologyManagement,o=NetscapeRoot

一般的な認証ポリシーの設定の詳細については、「認証ポリシー」を参照してください。ポリシールールで使用される Citrix ADC 式の詳細については、「ポリシーと式」を参照してください。

CLI を使用して LDAP 認証サーバーを作成するには

コマンドプロンプトで、次のコマンドを入力します。

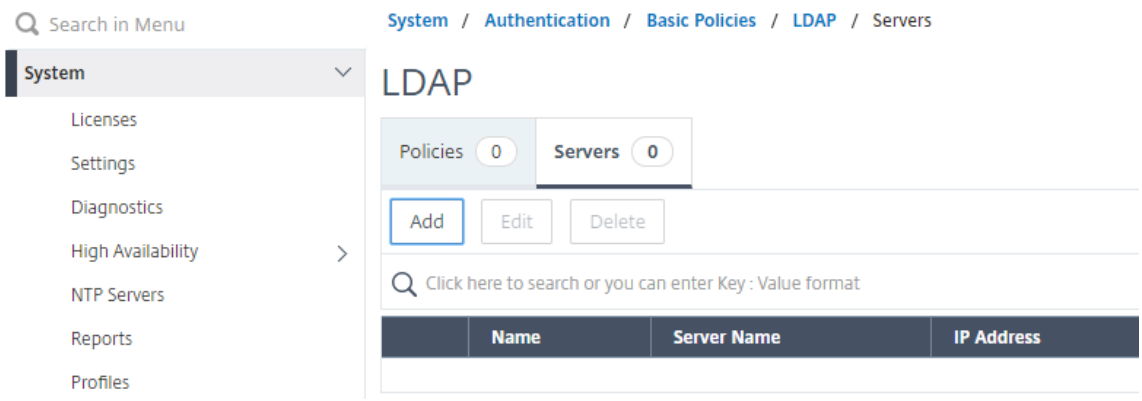
```
add authentication ldapAction <name> {
 -serverIP }
  <ip_addr|ipv6_addr|> | {
 -serverName <string> }
  }
```

例


```
add authentication ldapAction ldap_server -serverip 1.1.1.1 -serverName
    ldap_test
```

GUI を使用して **LDAP** 認証サーバーを作成するには

1. システム **>** 認証 **>** 基本ポリシー **> LDAP >** サーバ **>** 追加に移動します。



2. **[**認証 **LDAP** サーバの作成] ページで、LDAP サーバのパラメータを設定します。
3. [**Create**] をクリックします。

CLI を使用して認証ポリシーを有効にするには

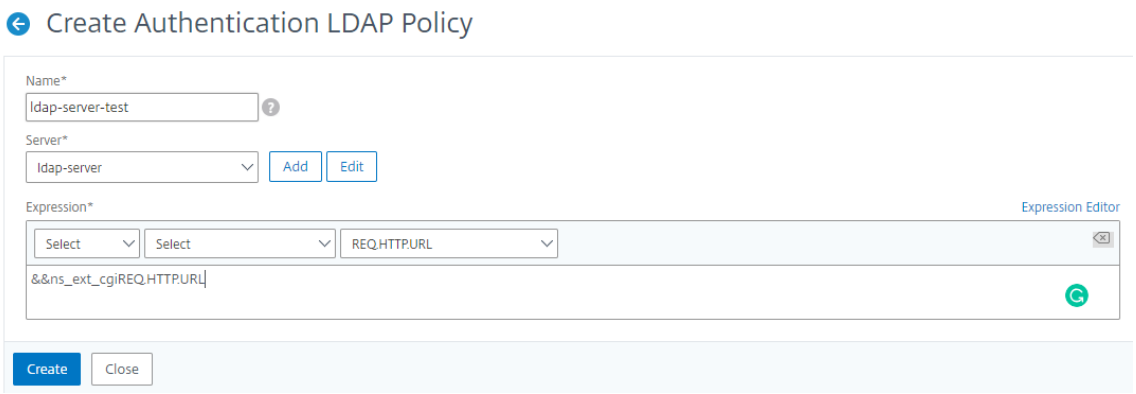
```
add authentication ldappolicy <name> <rule> [<reqAction>]
```

例:

```
add authentication ldappolicy ldap-service-policy ns_true ldap_Server
```

GUI を使用して **LDAP** 認証ポリシーを作成するには

1. システム **>** 認証 **>** 基本ポリシー **> LDAP >** ポリシー **>** 追加に移動します。
2. **[**認証 **LDAP** ポリシーの作成] ページで、LDAP ポリシーのパラメータを設定します。



3. [**Create**] をクリックします。

> 注
>
> LDAP サーバ/ポリシーは、[セキュリティ] タブで設定できます。セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 基本ポリシー **> LDAP >** サーバ/ポリシーに移動します。

CLI を使用して **LDAP** 参照サポートを有効にするには

コマンドプロンプトで、次のコマンドを入力します。

```
set authentication ldapAction <name> -followReferrals ON
set authentication ldapAction <name> -maxLDAPReferrals <integer>
<!--NeedCopy-->
```

例

```
set authentication ldapAction ldapAction-1 -followReferrals ON
set authentication ldapAction ldapAction-1 -maxLDAPReferrals 2
<!--NeedCopy-->
```

LDAP ユーザに対するキーベース認証のサポート

キーベース認証では、SSH を使用して LDAP サーバー内のユーザーオブジェクトに格納されている公開キーのリストを取得できるようになりました。ロールベース認証（RBA）プロセス中の Citrix ADC アプライアンスは、LDAP サーバーから公開 SSH キーを抽出する必要があります。取得した公開キーは SSH と互換性があり、RBA メソッドを使用してログインできる必要があります。

「認証の追加 ldapAction」コマンドと「認証の設定 ldapAction」コマンドに新しい属性「sshPublicKey」が導入されました。この属性を使用することで、次のようなメリットが得られます。

- 取得した公開キーを格納でき、LDAP アクションはこの属性を使用して LDAP サーバーから SSH キー情報を取得します。
- 最大 24 KB の属性名を抽出できます。

> 注
>
> LDAP などの外部認証サーバは、SSH キー情報の取得にのみ使用されます。認証目的では使用されません。

次に、SSH を介したイベントのフローの例を示します。

- SSH デーモンは、パスワードフィールドを空にして AAA_AUTHENTICATE 要求を認証、承認、および監査デーモンポートに送信します。
- LDAP が SSH 公開キーを格納するように設定されている場合、認証、承認、および監査は、他の属性とともに「sshPublicKey」属性で応答します。
- SSH デーモンは、これらのキーをクライアントキーで検証します。
- SSH デーモンはリクエストペイロードでユーザー名を渡し、認証、承認、監査は汎用キーとともにこのユーザーに固有のキーを返します。

sshPublicKey 属性を構成するには、コマンドプロンプトで次のコマンドを入力します。

- add オペレーションでは、`ldapAction` コマンドの設定中に「sshPublicKey」属性を追加できます。

```
add authentication ldapAction <name> {
-serverIP <ip_addr|ipv6_addr|*> | {
-serverName <string> }
 }
 [-serverPort <port>] … [-Attribute1 <string>] … [-Attribute16
    <string>][-sshPublicKey <string>][-authentication off]
<!--NeedCopy-->
```

- set オペレーションでは、既に追加された LDAPaction コマンドに「sshPublicKey」属性を設定できます。

```
set authentication ldapAction <name> [-sshPublicKey <string>][-
   authentication off]
<!--NeedCopy-->
```

LDAP 認証に対する名前と値の属性のサポート

LDAP 認証の属性に、一意の名前と値を設定できるようになりました。名前は LDAP アクションパラメータで設定され、名前はクエリーによって取得されます。この機能を使用することで、Citrix ADC アプライアンス管理者は次の利点を実現できます。

- 属性を (値だけでなく) 名前で記憶することで、管理者の労力を最小化

- 名前に関連付けられた属性値をクエリーするように検索機能を拡張します。
- 複数の属性を抽出するオプションを提供します。

Citrix ADC アプライアンスのコマンドプロンプトでこの機能を構成するには、次のように入力します。

```
add authentication ldapAction <name> [-Attribute1 <string>]
<!--NeedCopy-->
```

例

```
add authentication ldapAction ldapAct1 attribute1 mail
<!--NeedCopy-->
```

エンドツーエンドの **LDAP** 認証の検証のサポート

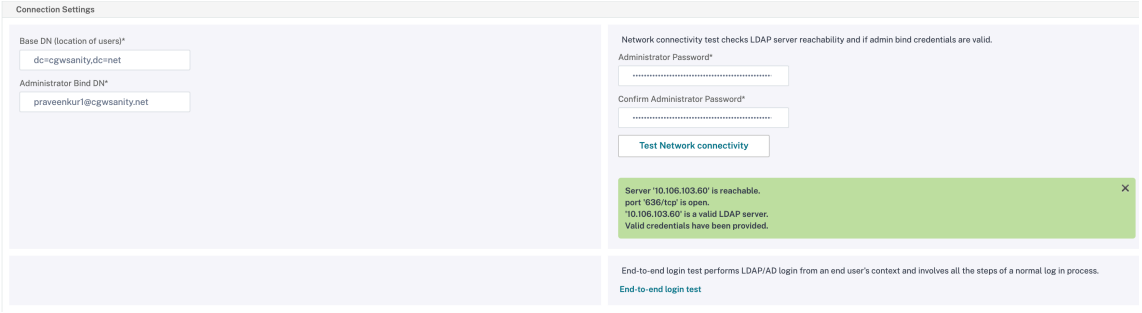
Citrix ADC アプライアンスは、GUI を介してエンドツーエンドの LDAP 認証を検証できるようになりました。この機能を検証するために、GUI に新しい「テスト」ボタンが導入されました。Citrix ADC アプライアンス管理者は、この機能を使用して次の利点を実現できます。

- フロー全体（パケットエンジン-Citrix ADC AAA デーモン-外部サーバー）を統合して、より優れた分析を実現
- 個々のシナリオに関連する問題の検証とトラブルシューティングにかかる時間を短縮

GUI を使用して LDAP エンドツーエンド認証のテスト結果を設定および表示するには、2 つのオプションがあります。

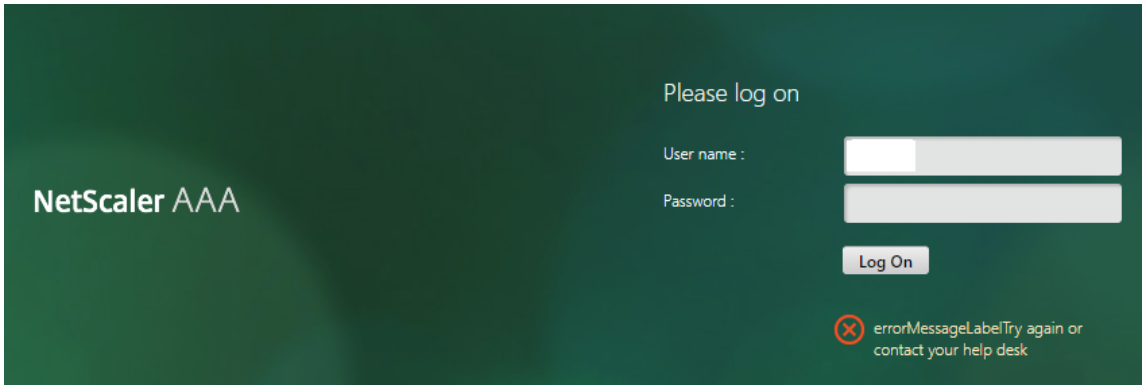
[システムから**]** オプション

1. [システム**] > [**認証**] > [**基本ポリシー**] > [LDAP]** に移動し、 [サーバー] タブをクリックします。
2. 使用可能な **LDAP** アクションをリストから選択します。
3. **[**認証 **LDAP** サーバーの構成] ページで、**[**接続設定**]** セクションまで下にスクロールします。
4. [ネットワーク接続のテスト] をクリックして、LDAP サーバ接続を確認します。LDAP サーバへの接続が成功したことを示すポップアップメッセージを、TCP ポートの詳細と有効なクレデンシャルの認証情報とともに表示できます。



5. エンドツーエンドの LDAP 認証を表示するには、[エンドツーエンドログインテスト] リンクをクリックします。
6. [エンドツーエンドログインテスト] ページで、[テスト] をクリックします。
   - 認証ページで、有効な認証情報を入力してログインします。成功画面が表示されます。

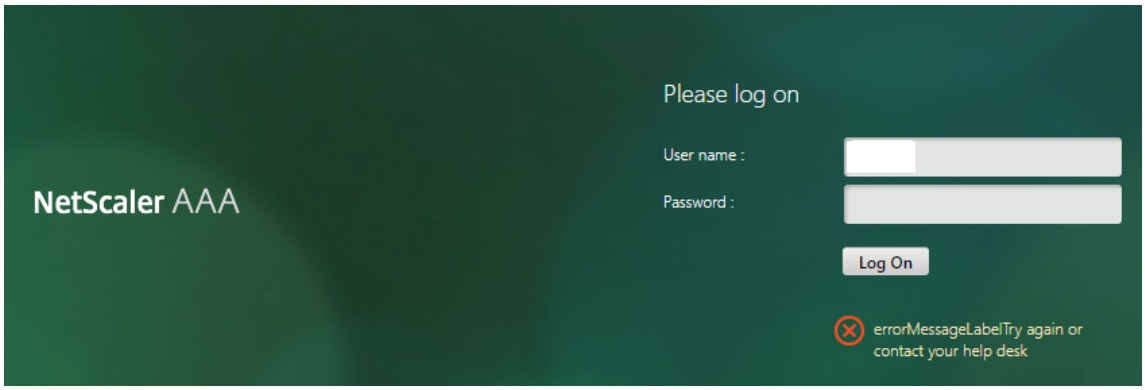
   - 認証に失敗すると、エラー画面が表示されます。



[認証] オプションから

1. [認証] **>** **[**ダッシュボード] に移動し、使用可能な LDAP アクションをリストから選択します。
2. **[**認証 **LDAP** サーバーの構成] ページの [接続設定**]** セクションには、2 つのオプションがあります。
3. LDAP サーバ接続を確認するには、[**LDAP** 到達可能性のテスト] タブをクリックします。LDAP サーバへの接続が成功したことを示すポップアップメッセージを、TCP ポートの詳細と有効なクレデンシャルの認証情報とともに表示できます。
4. エンドツーエンドの LDAP 認証ステータスを表示するには、[エンドユーザー接続のテスト] リンクをクリックします。
5. [エンドユーザー接続のテスト] ページで、[テスト] をクリックします。
   - 認証ページで、有効な認証情報を入力してログインします。成功画面が表示されます。

- 認証に失敗すると、エラー画面が表示されます。



LDAP 認証のための 14 日間のパスワード有効期限通知

Citrix ADC アプライアンスは、LDAP ベースの認証で 14 日間のパスワード有効期限切れ通知をサポートするようになりました。この機能を使用すると、管理者はパスワードの有効期限のしきい値を日単位でエンドユーザに通知できます。14 日間のパスワード期限切れ通知は、セルフサービスパスワードリセット (SSPR) の前兆です。

> （注）
>
> パスワード期限切れ通知の最大値またはしきい値（日数）は 255 日です。

パスワード期限切れ通知のメリット

- ユーザーは自分でパスワードをリセットでき、管理者はパスワードの有効期限を日単位でエンドユーザーに柔軟に通知できます。
- エンドユーザーがパスワードの有効期限の追跡に依存する必要がなくなります。
- 有効期限が切れる前にパスワードを変更するようユーザに（日数に基づいて）VPN ポータルページへの通知を送信します。

> 注
>
> この機能は LDAP ベースの認証方式にのみ適用され、RADIUS または TACACS には適用されません。

14 日間のパスワード通知について

Citrix ADC アプライアンスは、LDAP 認証サーバーから 2 つの属性（`Max-Pwd-Age and Pwd-Last-Set`）を取得します。

- **Max-Pwd-Age**。この属性は、パスワードが有効になるまでの最大時間を 100 ナノ秒間隔で示します。この値は、パスワードが設定されてからパスワードの有効期限が切れるまでに 100 ナノ秒の間隔を表す大きな整数として格納されます。
- **Pwd-Last-Set.** この属性は、アカウントのパスワードが最後に変更された日時を決定します。

Citrix ADC アプライアンスは、LDAP 認証サーバーから 2 つの属性をフェッチすることにより、特定のユーザーのパスワードが期限切れになるまでの残り時間を決定します。この情報は、認証サーバでユーザクレデンシャルが検証され、通知がユーザに返送されるときに収集されます。

新しいパラメータ「pwdExpiryNotification」が`set aaa parameter`コマンドに導入されました。このパラメーターを使用することで、管理者はパスワードの有効期限の残り日数を追跡できます。これで、Citrix ADC アプライアンスは、パスワードの有効期限に関するエンドユーザーへの通知を開始できます。

> 注
>
> 現在、この機能は、LDAP 実装の Microsoft AD サーバーを持つ認証サーバーでのみ動作します。OpenLDAP ベースのサーバーのサポートは後から対象となります。

14 日間のパスワード期限切れ通知を設定するイベントのフローの例を次に示します。

1. 管理者は、Citrix ADC アプライアンスを使用して、パスワードの有効期限（14 日間）を設定します。
2. ユーザーは HTTP または HTTPS リクエストを送信して、バックエンドサーバー上のリソースにアクセスします。
3. Citrix ADC アプライアンスは、アクセスを提供する前に、LDAP 認証サーバーで構成されているものを使用してユーザー資格情報を検証します。
4. Citrix ADC アプライアンスは、このクエリとともに、2 つの属性の詳細を取得するための要求を転送します（`Max-Pwd-Age and Pwd-Last-Set`）。
5. パスワードの有効期限が切れるまでの残り時間に基づいて、有効期限の通知が表示されます。
6. その後、ユーザは適切なアクションを実行してパスワードを更新します。

コマンドラインインターフェイスを使用して **14** 日間の有効期限通知を構成するには

> 注
>
> 14 日間の有効期限の通知は、ICA プロキシではなく、クライアントレス VPN とフル VPN のユースケースに対して構成できます。

コマンドプロンプトで、次のコマンドを入力します。

```
1 set aaa parameter  - pwdExpiryNotificationDays <positive_integer>
```



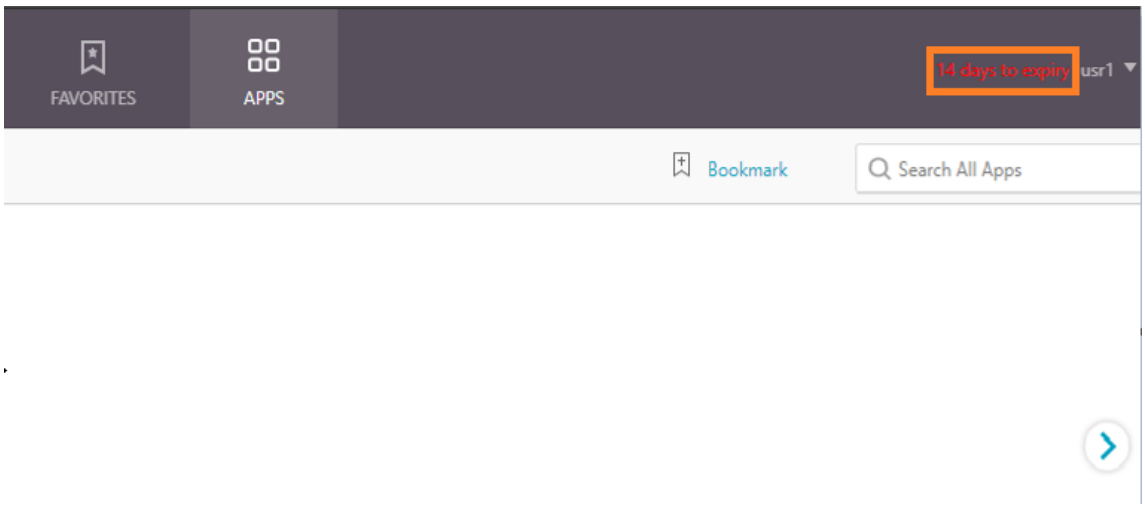
```
2
3 show aaa parameter
4 <!--NeedCopy-->
```

例

```
1      > set aaa parameter -pwdExpiryNotificationDays 14
2      Done
3      > show aaa parameter                              Configured AAA
          parameters  EnableStaticPageCaching: YES
          EnableEnhancedAuthFeedback: NO  DefaultAuthType: LOCAL
          MaxAAAUsers:           Unlimited
                                                 AAAD nat ip: None
                     EnableSessionStickiness : NO  aaaSessionLoglevel :
          INFORMATIONAL               AAAD Log Level : INFORMATIONAL
                           Dynamic address: OFF
4         GUI mode: ON
5         Max Saml Deflate Size: 1024             Password Expiry
             Notification Days: 14
6 <!--NeedCopy-->
```

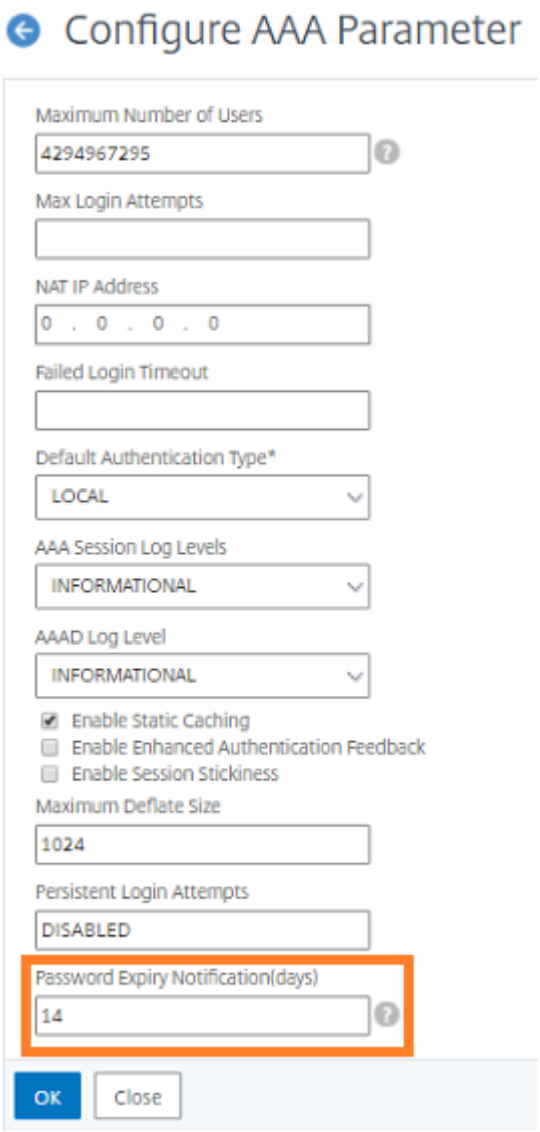
GUI を使用して **14** 日間の有効期限の通知を構成するには

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** 認証設定に移動します。
2. [認証 **AAA** 設定の変更] をクリックします。
3. **[AAA** パラメータの設定**]** ページで、[パスワード有効期限通知 **(**日**)]** フィールドに日数を指定します。



4. **OK** をクリックします。

VPN ポータルページの右上隅に通知が表示されます。



管理目的で **Citrix ADC** アプライアンスでの **LDAP** 認証を構成する

December 7, 2021

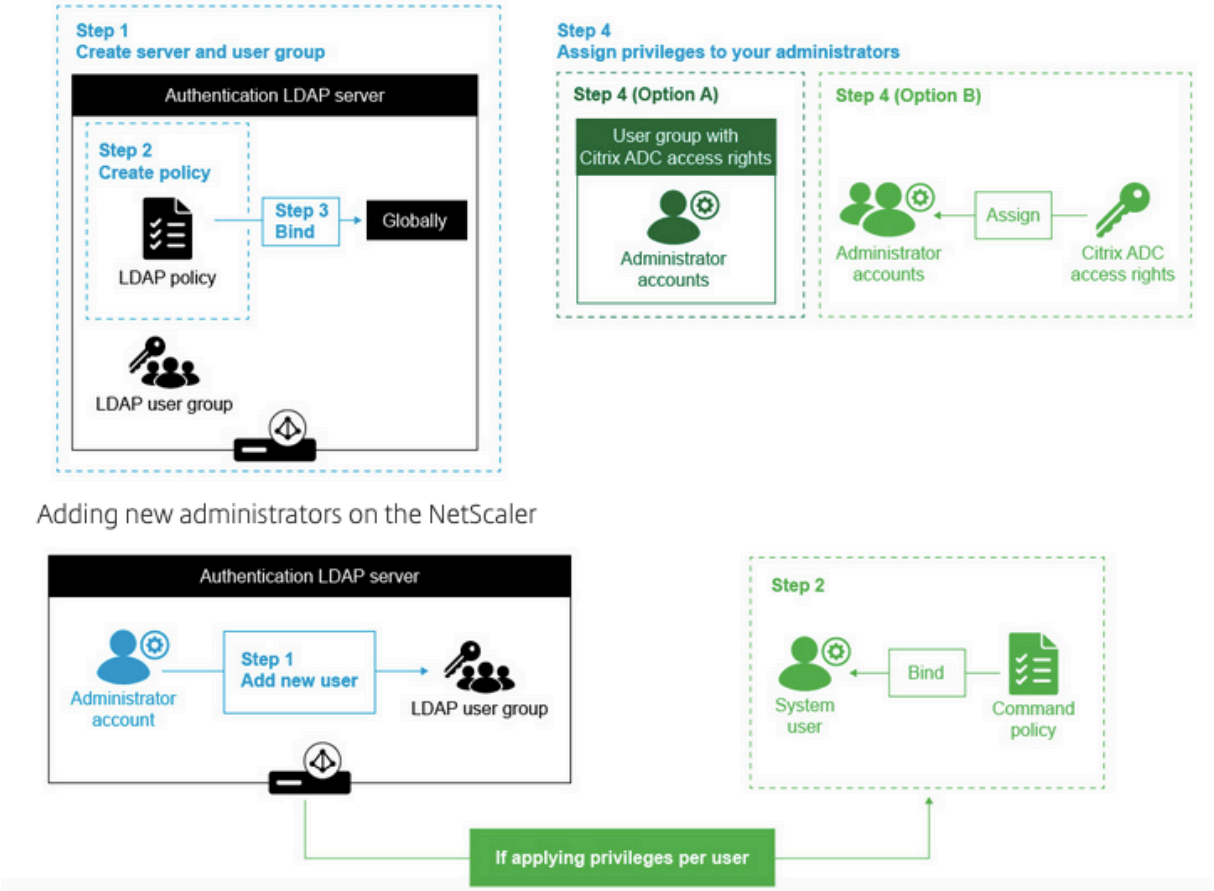
管理目的（スーパーユーザー、読み取り専用、ネットワーク権限など）で、Active Directory 資格情報（ユーザー名とパスワード）を使用して、Citrix ADC アプライアンスへのユーザーログオンを構成できます。

前提条件

- Windows Active Directory ドメインコントローラーサーバー

- NetScaler 管理者専用のドメイングループ
- Citrix Gateway 10.1 以降のバージョン

次の図は、Citrix ADC アプライアンスでの LDAP 認証を示しています。

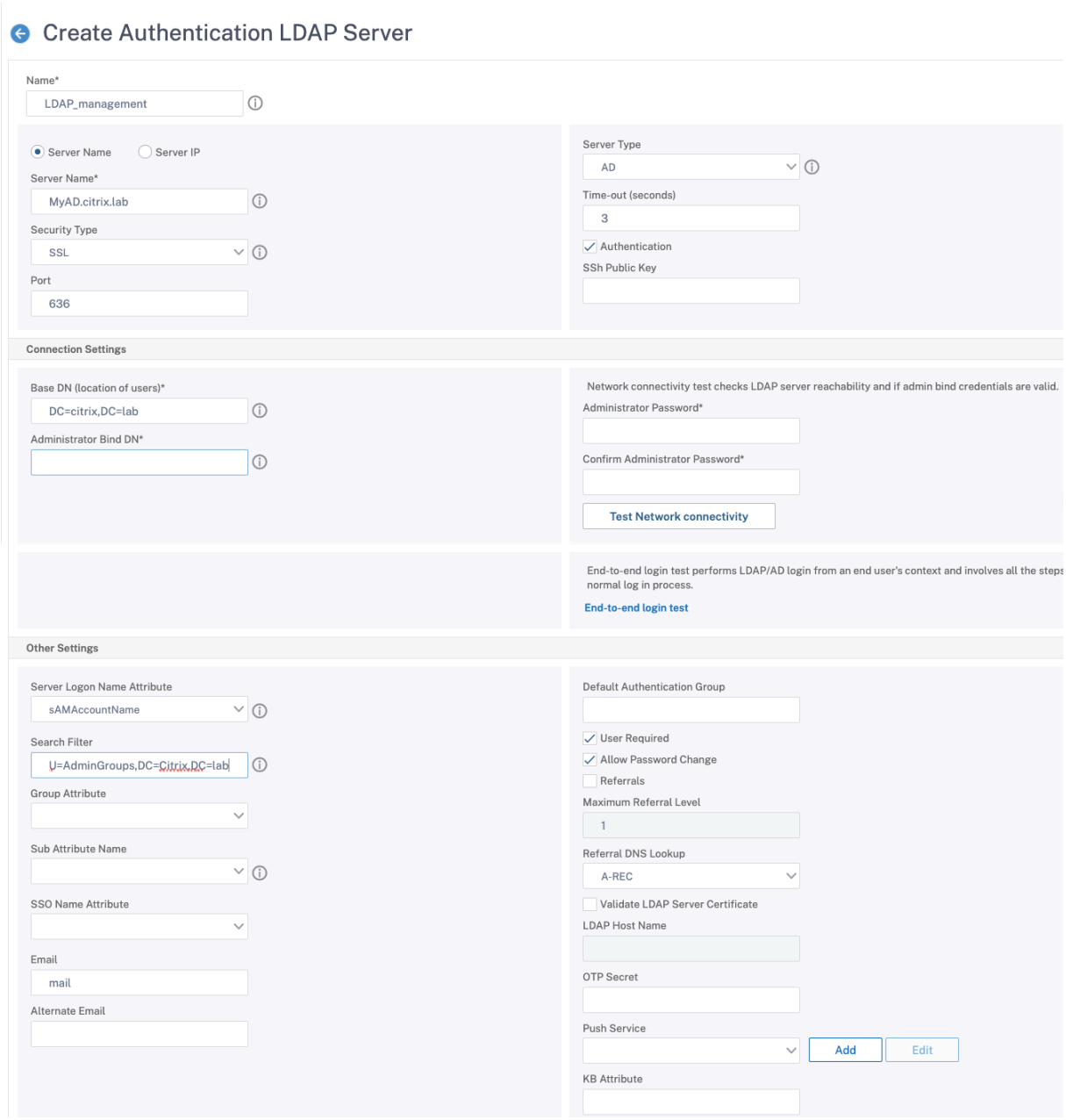


高レベルの設定手順

1. LDAP サーバーを作成する
2. LDAP ポリシーを作成する
3. LDAP ポリシーをバインドする
4. 次のいずれかの方法で、管理者に権限を割り当てます。
   - グループに対する権限の適用
   - ユーザーごとに権限を個別に適用する

認証 **LDAP** サーバーを作成する

1. [**System**] > [**Authentication**] > [**LDAP**] の順に選択します。
2. [サーバー] タブをクリックし、[追加] をクリックします。
3. 設定を完了し、[**Create**] をクリックします。



> 注:
>
> この例では、検索フィルターを設定してユーザーグループメンバーシップの認証をフィルタリングすることにより、アクセスは Citrix ADC アプライアンスに制限されます。この例で使用される値は、-&(memberOf=CN=NSG_admin、OU=AdminGroups、DC=Citrix、DC=lab) です。

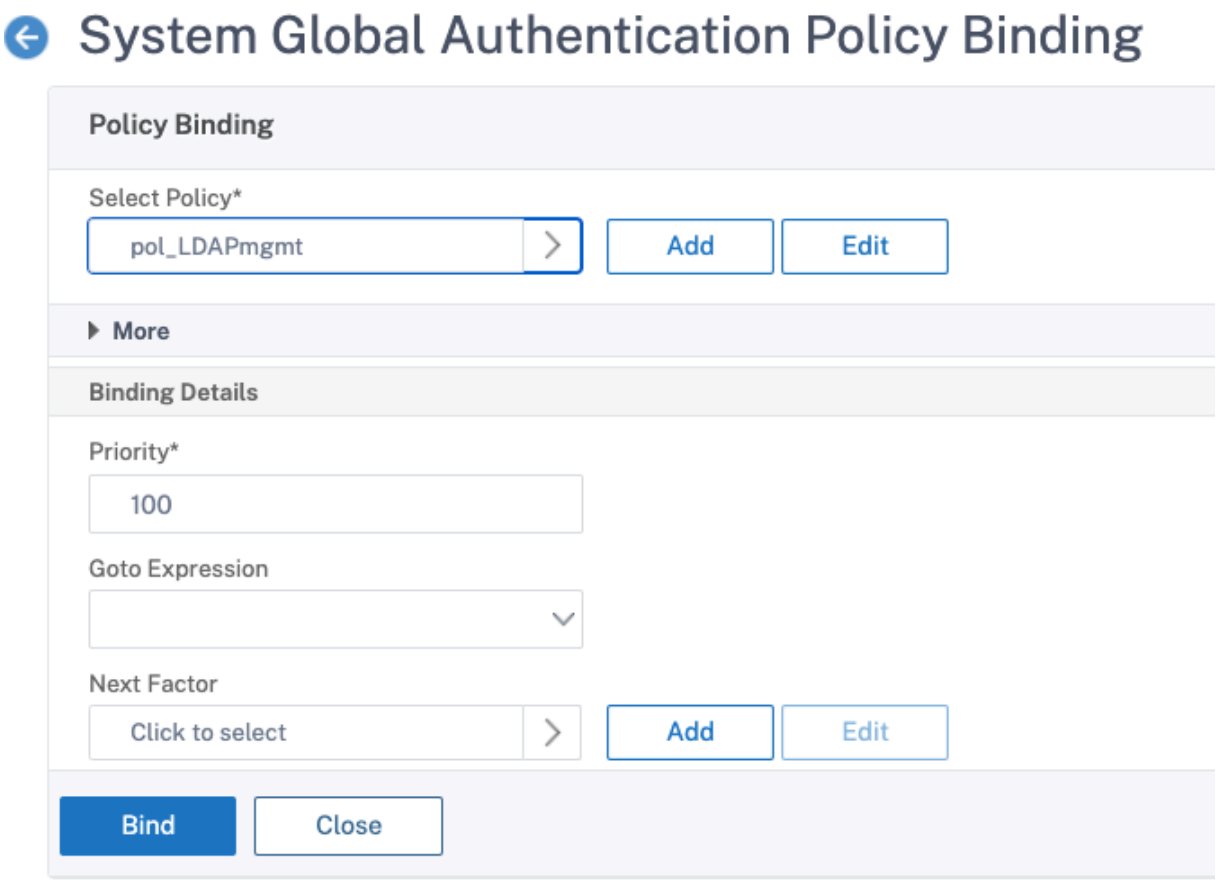
LDAP ポリシーを作成する

1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. [追加] をクリックします。

3. ポリシーの名前を入力し、前の手順で作成したサーバを選択します。
4. [式] テキストフィールドに適切な式を入力し、[作成] をクリックします。

LDAP ポリシーをグローバルにバインドする

1. [システム] > [認証] > [詳細ポリシー] > [ポリシー] に移動します。
2. [認証ポリシー] ページで、[グローバルバインディング] をクリックします。
3. 作成したポリシー (この例では pol_ldapMgmt) を選択します。
4. 優先順位を適宜選択してください（数字が小さいほど優先度が高くなります)。
5. [バインド]、[完了] の順にクリックします。[グローバル境界] 列に緑色のチェックマークが表示されます。



管理者に権限を割り当てる

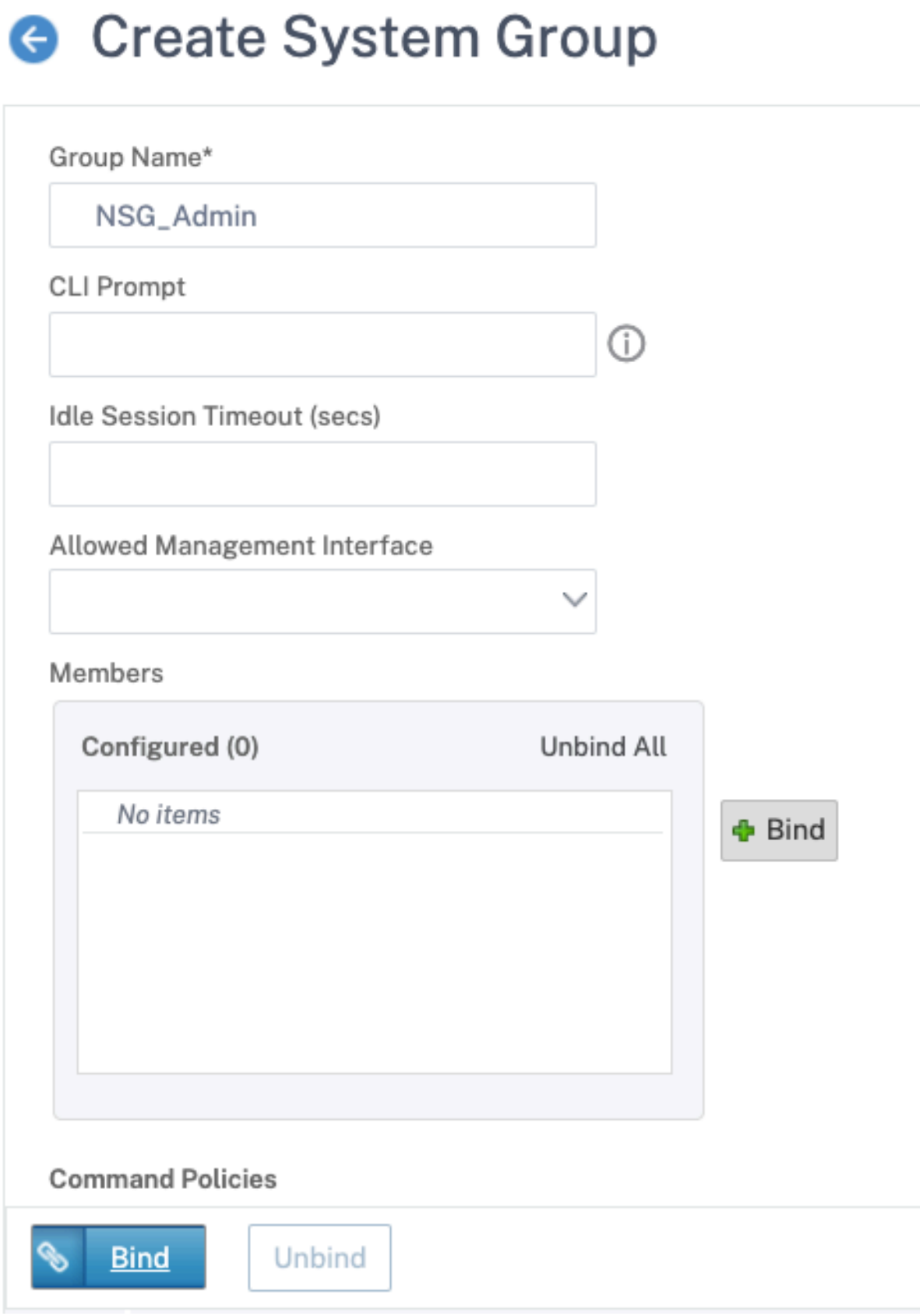
次の 2 つのオプションのいずれかを選択できます。

- グループに権限を適用する： Citrix ADC アプライアンスにグループを追加し、このグループのメンバーである各ユーザーに同じアクセス権を割り当てます。
- ユーザーごとに権限を個別に適用する**:** 各ユーザー管理者アカウントを作成し、それぞれに権限を割り当てます。

グループに対する権限の適用

グループに権限を適用すると、検索フィルターで構成された Active Directory グループ（この例では NSG_Admin）のメンバーであるユーザーは、Citrix ADC 管理インターフェイスに接続し、スーパーユーザーコマンドポリシーを持つことができます。

1. [**System**] **>** [**User Administration**] **>** [**Groups**] の順に選択します。
2. 要件に従って詳細を入力し、[**Create**] をクリックします。



ユーザーが属する Active Directory グループと、ログイン時にアカウントに関連付ける必要があるコマンドポリシーレベルを定義しました。検索フィルタで設定した LDAP グループに、新しい管理者ユーザを追加できます。

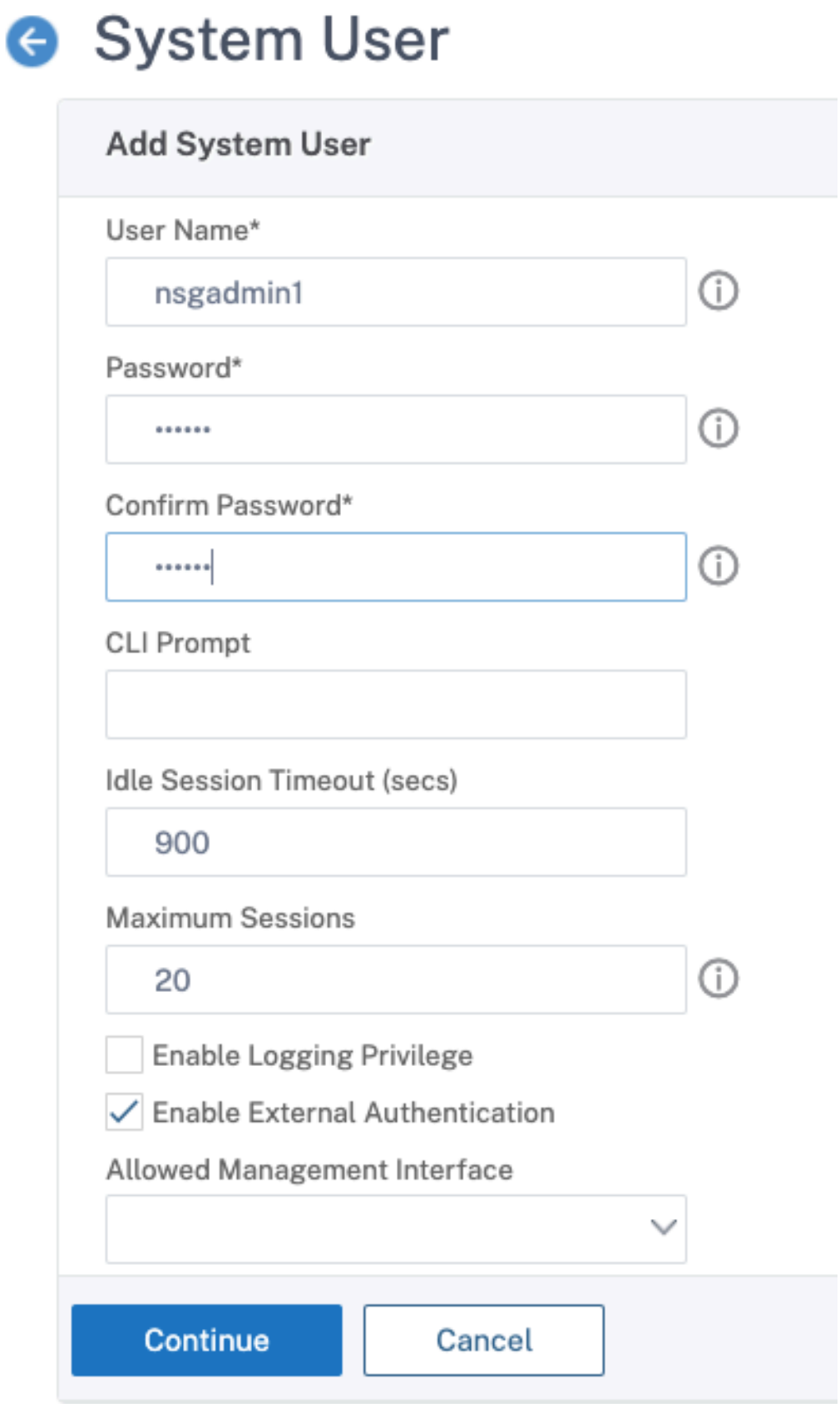
> 注**:**
>
> グループ名は Active Directory レコードと一致する必要があります。

ユーザーごとに権限を個別に適用する

このシナリオでは、検索フィルターで構成された Active Directory グループ（この例では、NSG_Admin）のメンバーであるユーザーは、Citrix ADC 管理インターフェイスに接続できますが、Citrix ADC アプライアンスで特定のユーザーを作成してコマンドポリシーをバインドするまで、権限を持ちません。

1. ［**System**］**>** ［**User Administration**］**>** ［**Users**］の順に選択します。
2. ［追加］をクリックします。
3. 要件に従って詳細を入力します。

   注**:** [外部認証を有効にする] を必ず選択してください。



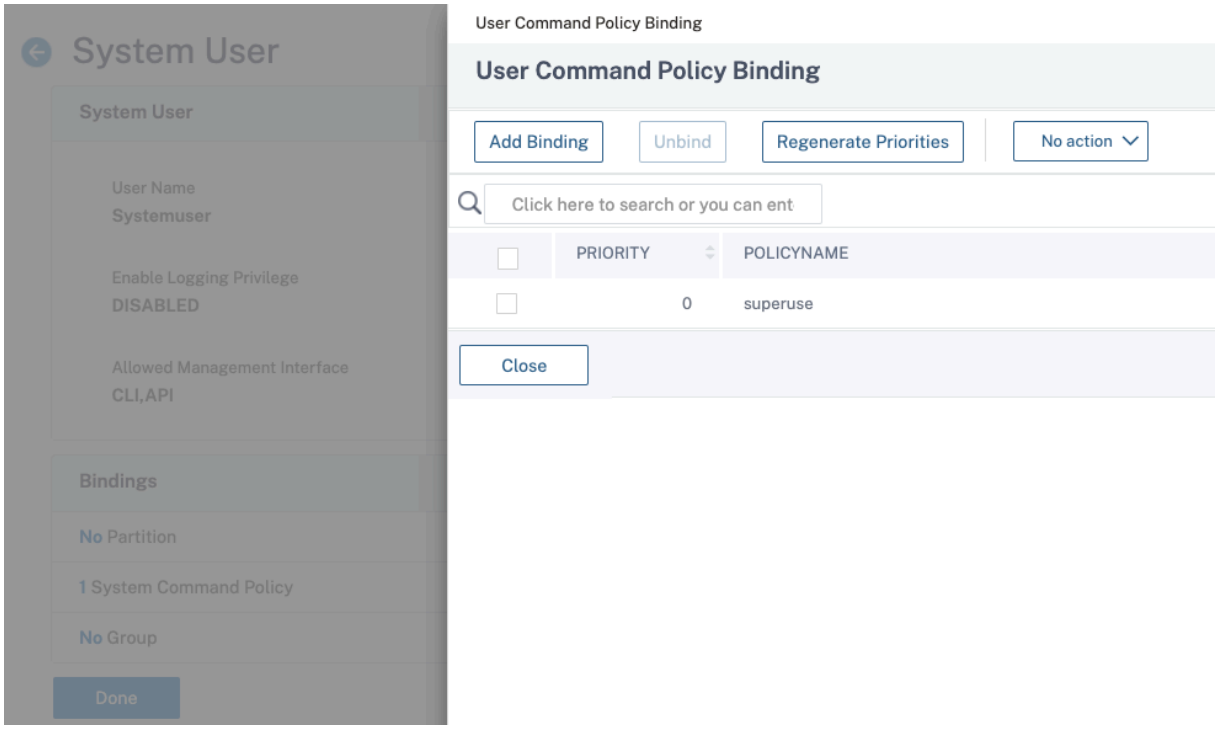
1. ［続行］をクリックします。

ログイン時にアカウントに関連付ける必要がある Active Directory ユーザーとコマンドポリシーレベルを定義しました。

> 注:
>
> - ユーザー名は、既存のユーザーの Active Directory レコードと一致する必要があります。
> - 外部認証のためにユーザーを Citrix ADC に追加するときに、外部認証が利用できない場合はパスワードを入力する必要があります。外部認証を正しく機能させるには、内部パスワードをユーザーアカウントの LDAP パスワードと同じにしないでください。

コマンドポリシーをユーザーに追加する

1. ［**System**］**>** ［**User Administration**］**>** ［**Users**］の順に選択します。
2. 作成したユーザーを選択し、[**Edit]** をクリックします。
3. [バインディング] で、[システムコマンドポリシー] をクリックします。
4. ユーザーに適用する正しいコマンドポリシーを選択します。
5. [バインド] をクリックして [閉じる] をクリックします。



管理者を追加するには

- 検索フィルタに設定した LDAP グループに管理者ユーザを追加します。
- Citrix ADC でシステムユーザーを作成し、正しいコマンドポリシーを割り当てます。

CLI を使用して管理目的で Citrix ADC アプライアンスで LDAP 認証を構成するには

Citrix ADC アプライアンス CLI でスーパーユーザー権限を持つグループのログオンを構成するには、次のコマンドをリファレンスとして使用します。

1. LDAP サーバーを作成する

```
add authentication ldapAction LDAP_mgmt -serverIP myAD.citrix.lab
    -serverPort 636 -ldapBase "DC=citrix,DC=lab" -ldapBindDn
    readonly@citrix.lab -ldapBindDnPassword -ldapLoginName
    sAMAccountName -searchFilter "&(memberof=CN=NSG_Admin,OU=
    AdminGroups,DC=citrix,DC=lab)" -groupAttrName memberOf
<!--NeedCopy-->
```

2. 作成および LDAP ポリシー

```
add authentication policy pol_LDAPmgmt -rule true -action
    LDAP_mgmt
<!--NeedCopy-->
```

3. LDAP ポリシーのバインド

```
bind system global pol_LDAPmgmt -priority 110
<!--NeedCopy-->
```

4. 管理者に権限を割り当てる

   - グループに権限を適用するには

```
add system group NSG_Admin
bind system group NSG_Admin -policyName superuser 100
<!--NeedCopy-->
```

   - ユーザーごとに個別に特権を適用するには

```
add system user admyoa
bind system user admyoa superuser 100
<!--NeedCopy-->
```

RADIUS 認証

October 7, 2021

他の種類の認証ポリシーと同様に、リモート認証ダイヤルインユーザーサービス (RADIUS) 認証ポリシーは、式とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバにバインドし、プライオリティを割り当てます。また、バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。ただし、RADIUS 認証ポリシーを設定するには、次に説明する特定の特別な要件があります。

通常、認証時に認証サーバーの IP アドレスを使用するように Citrix ADC を構成します。RADIUS 認証サーバを使用すると、IP アドレスではなく RADIUS サーバの FQDN を使用してユーザを認証するように ADC を設定できるようになりました。FQDN を使用すると、認証サーバが複数の IP アドレスのいずれかに存在し、常に 1 つの FQDN を使用する環境において、より複雑な認証、承認、および監査の構成を簡素化できます。IP アドレスの代わりにサーバーの FQDN を使用して認証を構成するには、認証アクションを作成する場合を除き、通常の構成プロセスに従います。アクションを作成するときは、**serverIP** パラメータを **serverName** パラメータに置き換えます。

ユーザーの認証に RADIUS サーバーの IP または FQDN を使用するよう Citrix ADC を構成するかを決定する前に、IP アドレスの代わりに FQDN に対して認証を行うように認証、承認、および監査を構成すると、認証プロセスに余分な手順が追加されることを検討してください。ADC は、ユーザを認証するたびに、FQDN を解決する必要があります。多数のユーザーが同時に認証を試みると、結果として生じる DNS ルックアップによって認証プロセスが遅くなる可能性があります。

> 注
>
> これらの手順は、すでに RADIUS プロトコルに精通しており、選択した RADIUS 認証サーバをすでに設定していることを前提としています。

コマンドラインインターフェイスを使用して **RADIUS** サーバーの認証アクションを追加するには

RADIUS サーバに対して認証を行う場合は、明示的な認証アクションを追加する必要があります。これを行うには、コマンドプロンプトで次のコマンドを入力します。

```
1 add authentication radiusAction <name> [-serverip <IP> | -serverName] <
     FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
2  -radKey  }
3   [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
       <positive_integer>][-radAttributeType <positive_integer>][-
       radGroupsPrefix <string>] [-radGroupSeparator <string>][-
       passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
       ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
       pwdVendorID <positive_integer> [-pwdAttributeType <
       positive_integer>]] [-defaultAuthenticationGroup <string>] [-
       callingstationid ( ENABLED | DISABLED )]
```



```
4
5 <!--NeedCopy-->
```

次の例では、**Authn-Act-1** という名前の RADIUS 認証アクションを追加します。このアクションには、サーバ IP **10.218.24.65**、サーバポート **1812**、認証タイムアウト **15** 分、RADIUS キー **WareTheLorax**、NAS IP が無効、NAS ID が含まれます。**NAS1**.

```
1 add authentication radiusaction Authn-Act-1 -serverip 10.218.24.65 -
     serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
     DISABLED -radNASid NAS1
2 Done
3
4 <!--NeedCopy-->
```

次の例では、同じ RADIUS 認証アクションを追加しますが、IP の代わりにサーバ FQDN **rad01.example.com** を使用します。

```
1 add authentication radiusaction Authn-Act-1 -serverName rad01.example.
     com -serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
     DISABLED -radNASid NAS1
2 Done
3
4 <!--NeedCopy-->
```

コマンドラインを使用して外部 **RADIUS** サーバーの認証アクションを構成するには

既存の RADIUS アクションを構成するには、コマンドプロンプトで次のコマンドを入力します。

```
1 set authentication radiusAction <name> [-serverip <IP> | -serverName] <
     FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
2  -radKey  }
3   [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
       <positive_integer>][-radAttributeType <positive_integer>][-
       radGroupsPrefix <string>] [-radGroupSeparator <string>][-
       passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
       ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
       pwdVendorID <positive_integer> [-pwdAttributeType <
       positive_integer>]] [-defaultAuthenticationGroup <string>] [-
       callingstationid ( ENABLED | DISABLED )]
```



```
4
5 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して外部 **RADIUS** サーバーの認証アクションを削除するには

既存の RADIUS アクションを削除するには、コマンドプロンプトで次のコマンドを入力します。

```
1 rm authentication radiusAction <name>
2
3 <!--NeedCopy-->
```

例

```
1 rm authentication radiusaction Authn-Act-1
2 Done
3
4 <!--NeedCopy-->
```

構成ユーティリティを使用して **RADIUS** サーバを構成するには

> 注
>
> 構成ユーティリティでは、「アクション」ではなく「サーバー」という用語が使用されますが、同じタスクを指します。

1. [セキュリティ] **> [AAA-**アプリケーショントラフィック] **> [**ポリシー] **> [**認証] **> [Radius]** に移動します。
2. 詳細ウィンドウの [サーバー] タブで、次のいずれかの操作を行います。
   - 新しい RADIUS サーバを作成するには、[**Add**] をクリックします。
   - 既存の RADIUS サーバを変更するには、サーバを選択し、[**Edit**] をクリックします。
3. [認証 **RADIUS** サーバーの作成] または [認証 **RADIUS** サーバーの構成] ダイアログで、パラメータの値を入力または選択します。[発信ステーション **ID** を送信] の下に表示されるパラメータを入力するには、[詳細] を展開します。
   - 名前 *—radiusActionName (以前に設定されたアクションでは変更できません)
   - 認証タイプ *: authtype（RADIUS に設定され、変更不可）
   - サーバ名/ IP アドレス *: サーバ名またはサーバ IP のいずれかを選択します。
     - サーバー名 *—serverName <FQDN>

- IP アドレス *: serverIp <IP> サーバに IPv6 IP アドレスが割り当てられている場合は、[IPv6] チェックボックスをオンにします。

- ポート *: serverPort
- タイムアウト (秒) *—authTimeout
- シークレットキー *: radKey（RADIUS 共有シークレット）
- 確認シークレットキー *: RADIUS 共有シークレットをもう一度入力します。(コマンドラインに相当するものはありません)。
- 発信側ステーション ID の送信: callingstationid
- グループベンダー識別子: radVendorID
- グループ属性タイプ-radAttributeType
- IP アドレスベンダー識別子: ipVendorID
- pwd ベンダー ID—pwdVendorID
- パスワードエンコーディング: passEncoding
- デフォルト認証グループ: defaultAuthenticationGroup
- NAS ID—radNASid
- NAS の IP アドレス抽出を有効にする: radNASip
- グループプレフィクス-radGroupsPrefix
- グループ・セパレーター—radGroupSeparator
- IP アドレス属性タイプ: ipAttributeType
- パスワード属性タイプ: pwdAttributeType
- アカウンティング: accounting

4. [作成] または [**OK]** をクリックします。作成したポリシーが [Servers] ページに表示されます。

RADIUS 属性 **66** を通過するサポート（トンネル**-**クライアントエンドポイント）

Citrix ADC アプライアンスは、RADIUS 認証中に RADIUS 属性 66（トンネル-クライアントエンドポイント）のパススルーを許可するようになりました。この機能を適用することで、クライアントの IP アドレスが第 2 要素認証によって受信され、リスクベースの認証の決定が委託されます。

「add authentication radiusAction」コマンドと「set radiusParams」コマンドの両方に、新しい属性「tunnelEndpointClientIP」が導入されました。

この機能を使用するには、**Citrix ADC** アプライアンスのコマンドプロンプトで次のように入力します。


```
add authentication radiusAction <name> {
 -serverIP <ip_addr|ipv6_addr|*> | {
 -serverName <string> }
  }
  [-serverPort <port>] … [-tunnelEndpointClientIP (ENABLED|DISABLED)]

set radiusParams {
 -serverIP <ip_addr|ipv6_addr|*> |{
 -serverName <string> }
  }
  [-serverPort<port>] … [-tunnelEndpointClientIP(ENABLED|DISABLED)]

<!--NeedCopy-->
```

例

```
add authentication radiusAction radius -severIP 1.217.22.20 -serverName
     FQDN -serverPort 1812 -tunnelEndpointClientIp ENABLED

set radiusParams -serverIp 1.217.22.20 -serverName FQDN1 -serverPort
    1812 -tunnelEndpointClientIP ENABLED

<!--NeedCopy-->
```

エンドツーエンドの **RADIUS** 認証の検証のサポート

Citrix ADC アプライアンスは、GUI を使用してエンドツーエンドの RADIUS 認証を検証できるようになりました。この機能を検証するために、GUI に新しい「テスト」ボタンが導入されました。Citrix ADC アプライアンス管理者は、この機能を利用して、次のメリットを得ることができます。

- 完全なフロー（パケットエンジン-aaa デーモン-外部サーバ）を統合し、より優れた分析を提供します。
- 個々のシナリオに関連する問題の検証とトラブルシューティングにかかる時間を短縮

GUI を使用して RADIUS エンドツーエンド認証のテスト結果を設定および表示するには、2 つのオプションがあります。

システム・オプションから

1. **[システム] > [認証] > [基本ポリシー] > [RADIUS]** に移動し、**[サーバー]** タブをクリックします。
2. リストから使用可能な **RADIUS** アクションを選択します。
3. [認証 **RADIUS** サーバーの構成] ページで、[接続の設定] セクションに 2 つのオプションがあります。

4. RADIUS サーバの接続を確認するには、[**RADIUS** 到達可能性のテスト] タブをクリックします。
5. エンドツーエンド RADIUS 認証を表示するには、[エンドユーザ接続のテスト] リンクをクリックします。

「認証から」オプション

1. [認証**] > [**ダッシュボード] に移動し、リストから使用可能な RADIUS アクションを選択します。
2. [認証 **RADIUS** サーバーの構成] ページで、[接続の設定**]** セクションに 2 つのオプションがあります。
3. RADIUS サーバの接続を確認するには、[**RADIUS** 到達可能性のテスト] タブをクリックします。
4. エンドツーエンド RADIUS 認証ステータスを表示するには、[エンドユーザ接続のテスト] リンクをクリックします。

TACACS 認証

October 7, 2021

TACACS 認証ポリシーは、外部の Terminal Access Controller Access-Control System（TACACS）認証サーバーに対して認証を行います。
ユーザーが TACACS サーバーへの認証を行った後、Citrix ADC は後続のすべての認証のために同じ TACACS サーバーに接続します。プライマリ TACACS サーバが使用できない場合、この機能により、ADC が最初の TACACS サーバがタイムアウトするのを待つ間遅延が回避されます。これは、2 番目の TACACS サーバに認可要求を再送信する前に発生します。

> 注:
>
> TACACS 許可サーバーは、ストリング長が 255 文字を超えるコマンドをサポートしていません。
>
> 回避策： TACACS 認証サーバーの代わりにローカル認証を使用します。

TACACS サーバーを介して認証する場合、認証、許可、およびトラフィック管理ログの監査では、TACACS コマンドのみが正常に実行されます。これにより、実行を許可されていないユーザーが入力した TACACS コマンドがログに表示されなくなります。

NetScaler 12.0 ビルド 57.x 以降、ターミナルアクセスコントローラアクセス制御システム（TACACS）は、TACACS 要求の送信中に認証、承認、監査デーモンをブロックしていません。LDAP および RADIUS 認証が要求を続行できるようになります。TACACS サーバが TACACS 要求を確認すると、TACACS 認証要求が再開されます。

> 重要:
>
> - 「clear ns config」コマンドを実行するときは、TACACS 関連の設定を変更しないことをお勧めします。
> - 詳細ポリシーの「clear ns config」コマンドで「rbaConfig」パラメータが NO に設定されている場合、詳細ポリシーに関連する TACACS 関連の設定がクリアされ、再適用されます。

TACACS 認証に対する名前/値属性のサポート

TACACS 認証属性に、一意の名前と値を設定できるようになりました。名前は TACACS アクションパラメータで設定され、値は名前のクエリによって取得されます。name 属性値を指定することで、管理者は属性名に関連付けられた属性値を簡単に検索できます。また、管理者は属性値を単独で覚えておく必要がなくなりました。

> 重要
>
> - tacacsAction コマンドでは、合計サイズが 2048 バイト未満のカンマで区切られた最大 64 個の属性を設定できます。

CLI を使用して名前/値属性を設定するには

コマンドプロンプトで入力します。

```
add authentication tacacsAction <name> [-Attributes <string>]
<!--NeedCopy-->
```

例:

```
add authentication tacacsAction tacacsAct1 -attributes  “mail,sn,
   userprincipalName”
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証アクションを追加するには

LOCAL 認証を使用しない場合は、明示的な認証アクションを追加する必要があります。コマンドプロンプトで、次のコマンドを入力します。

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <
   port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
   serverport 1812                    -authtimeout 15 -tacacsSecret ”
   minotaur” -authorization OFF -accounting ON -auditFailedCmds OFF -
   defaultAuthenticationGroup ”users”
```



```
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証アクションを構成するには

既存の認証操作を構成するには、コマンドプロンプトで次のコマンドを入力します。

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

例

```
    > set authentication tacacsaction Authn-Act-1 -serverip
        10.218.24.65 -serverport 1812                  -authtimeout 15
        -tacacsSecret "minotaur" -authorization OFF -accounting ON -
        auditFailedCmds OFF -defaultAuthenticationGroup "users"    Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証操作を削除するには

既存の RADIUS アクションを削除するには、コマンドプロンプトで次のコマンドを入力します。

```
rm authentication radiusAction <name>
<!--NeedCopy-->
```

例

```
rm authentication tacacsaction Authn-Act-1
<!--NeedCopy-->
```

クライアント証明書認証

May 9, 2022

オンラインバンキングの Web サイトや従業員の個人情報を含む Web サイトなど、機密性の高いコンテンツを含む Web サイトでは、認証のためにクライアント証明書が必要になる場合があります。クライアント側の証明書属性に基づいてユーザを認証するように認証、承認、および監査を構成するには、まずトラフィック管理仮想サーバでクライアント認証を有効にし、ルート証明書を認証仮想サーバにバインドします。次に、2 つのオプションのいずれかを実装します。認証仮想サーバーのデフォルトの認証タイプを CERT として構成するか、クライアント証明書に基づいてユーザーを認証するために Citrix ADC が実行する必要があることを定義する証明書アクションを作成できます。いずれの場合も、認証サーバーは CRL をサポートしている必要があります。クライアント証明書の **subjectCN** フィールドまたは別の指定されたフィールドからユーザー名を抽出するように ADC を構成します。

認証ポリシーが構成されておらず、グローバルカスケードが構成されていない認証仮想サーバーにユーザーがログオンしようとすると、証明書の指定されたフィールドからユーザー名情報が抽出されます。必須フィールドが抽出されると、認証は成功します。SSL ハンドシェイク中にユーザーが有効な証明書を提供しなかった場合、またはユーザー名の抽出が失敗した場合、認証は失敗します。クライアント証明書を検証した後、ADC はユーザーにログオンページを表示します。

次の手順は、機能する認証、承認、および監査構成がすでに作成されていることを前提としており、クライアント証明書を使用して認証を有効にする方法のみを説明しています。また、これらの手順では、ルート証明書とクライアント証明書を取得し、ADC の /nsconfig/ssl ディレクトリに配置したことを前提としています。

クライアント証明書認証の構成

コマンドラインインターフェイスを使用して認証、承認、および監査クライアント証明書パラメーターを構成するには

コマンドプロンプトで、次のコマンドを表示されている順序で入力して、証明書を構成し、構成を確認します。

```
1  add ssl certKey <certkeyName> -cert <certFile> -key <keyFile> -password
       -inform <inform> -expiryMonitor <expiryMonitor> -notificationPeriod
       <notificationPeriod>
2
3  bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
      positive_integer>]
4
5  show ssl certKey [<certkeyName>]
6
7  set aaa parameter -defaultAuthType CERT
8
9  show aaa parameter
10
11 set aaa certParams -userNameField ”Subject:CN”
12
13 show aaa certParams
14 <!--NeedCopy-->
```


構成ユーティリティを使用して認証、承認、および監査クライアント証明書パラメーターを構成するには

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** 仮想サーバに移動します。
2. 詳細ウィンドウで、クライアント証明書認証を処理するように構成する仮想サーバーを選択し、[編集**]** をクリックします。
3. ［構成］ページの［証明書］で、右矢印（>）をクリックして［CA 証明書キー］インストールダイアログを開きます。
4. [**CA** 証明書キー] ダイアログボックスで、[挿入**]** をクリックします。
5. [**CA** 証明書キー**-SSL** 証明書] ダイアログボックスで、[インストール**]** をクリックします。
6. [証明書のインストール（Install Certificate）] ダイアログボックスで、次のパラメータを設定します。これらの名前は、次に示すように CLI パラメータ名に対応します。
   - 証明書とキーのペア名 *—certKeyname
   - 証明書ファイル名 — certFile
   - キーファイル名 — keyFile
   - 証明書形式—inform
   - パスワード-password
   - 証明書バンドル — bundle
   - 有効期限が切れると通知—expiryMonitor
   - 通知期間 — notificationPeriod
7. [インストール] をクリックし、[閉じる] をクリックします。
8. [**CA** 証明書キー] ダイアログボックスの [証明書] リストで、ルート証明書を選択します。
9. ［保存］をクリックします。
10. [**Back**] をクリックして、メインの設定画面に戻ります。
11. セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 証明書に移動します。
12. 詳細ウィンドウで、クライアント証明書認証を処理するように構成するポリシーを選択し、[編集] をクリックします。
13. [認証証明書ポリシーの構成] ダイアログの [サーバー] ドロップダウンリストで、クライアント証明書認証を処理するように構成した仮想サーバーを選択します。
14. **OK** をクリックします。ステータスバーに、構成が正常に完了したことを示すメッセージが表示されます。

高度なポリシーを使用したクライアント証明書認証

以下は、高度なポリシーを使用して Citrix ADC でクライアント証明書認証を構成する手順です。

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** 仮想サーバに移動します。
2. 詳細ウィンドウで、クライアント証明書認証を処理するように構成する仮想サーバーを選択し、[編集] をクリックします。

> 注:
>
> 仮想サーバーの有効な CA 証明書とサーバー証明書をインポートした場合は、手順 **3** から手順 **10** までスキップできます。

3. ［構成］ページの［証明書］で、［****** ］をクリックして［****CA** 証明書キーインストール］ダイアログボックスを開きます。
4. [**CA** 証明書キー] ダイアログボックスで、[挿入] をクリックします。
5. [**CA** 証明書キー**-SSL** 証明書] ダイアログボックスで、[インストール] をクリックします。
6. [証明書のインストール（Install Certificate）] ダイアログボックスで、次のパラメータを設定します。これらの名前は、次に示すように CLI パラメータ名に対応します。
   - 証明書とキーのペア名—certKeyName
   - 証明書ファイル名 — certFile
   - キーファイル名 — keyFile
   - 証明書形式—inform
   - パスワード-password
   - 証明書バンドル —bundle
   - 有効期限が切れると通知—expiryMonitor
   - 通知期間 — notificationPeriod
7. [インストール] をクリックし、[閉じる] をクリックします。
8. [**CA Cert Key**] ダイアログボックスの [証明書] リストで、ルート証明書を選択します。
9. ［保存］をクリックします。
10. [**Back**] をクリックして、メインの設定画面に戻ります。
11. セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシーに移動し、ポリシーを選択します。
12. 詳細ウィンドウで、次のいずれかの操作を行います。
    - 新しいポリシーを作成するには、[追加（**Add**）] をクリックします。
    - 既存のポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。
13. [認証ポリシーの作成] または [認証ポリシーの構成] ダイアログボックスで、パラメータの値を入力または選択します。
    - Name -ポリシー名。以前に設定したポリシーでは変更できません。
    - アクションタイプ-証明書を選択
    - Action： ポリシーに関連付ける認証アクション（プロファイル）。既存の認証アクションを選択するか、プラス記号をクリックして適切なタイプの新しいアクションを作成できます。
    - Log Action -ポリシーに関連付ける監査アクション。既存の監査アクションを選択するか、プラス記号をクリックして新しいアクションを作成できます。

- 式 -指定したアクションを適用する接続を選択するルール。ルールは、シンプル (「true」はすべてのトラフィックを選択) または複雑にすることができます。式を入力するには、まず [式] ウィンドウの下にある左端のドロップダウンリストで式の種類を選択し、次に式テキスト領域に式を直接入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、その中のドロップダウンリストを使用して式。)
- [Comment]: この認証ポリシーが適用されるトラフィックのタイプを説明するコメントを入力できます。オプションです。

14. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。ポリシーを作成した場合、そのポリシーは [認証ポリシーおよびサーバ] ページに表示されます。

クライアント証明書パススルー

Citrix ADC は、ユーザー認証にクライアント証明書を必要とする保護されたアプリケーションにクライアント証明書を渡すように構成できるようになりました。ADC は最初にユーザーを認証し、次にクライアント証明書を要求に挿入してアプリケーションに送信します。この機能は、適切な SSL ポリシーを追加することによって構成されます。

ユーザーがクライアント証明書を提示したときのこの機能の正確な動作は、VPN 仮想サーバーの設定によって異なります。

- VPN 仮想サーバーがクライアント証明書を受け入れるように構成されているが、それを必要としない場合、ADC は証明書を要求に挿入し、保護されたアプリケーションに要求を転送します。
- VPN 仮想サーバーでクライアント証明書認証が無効になっている場合、ADC は認証プロトコルを再ネゴシエーションし、ユーザーを再認証してから、クライアント証明書をヘッダーに挿入し、要求を保護されたアプリケーションに転送します。
- VPN 仮想サーバーがクライアント証明書認証を要求するように構成されている場合、ADC はクライアント証明書を使用してユーザーを認証し、ヘッダーに証明書を挿入し、要求を保護されたアプリケーションに転送します。

いずれの場合も、クライアント証明書のパススルーを次のように構成します。

コマンドラインインターフェイスを使用してクライアント証明書パススルーを作成および構成する

コマンドプロンプトで、次のコマンドを入力します。

```
add vpn vserver <name> SSL <IP> 443
<!--NeedCopy-->
```

name には、仮想サーバーの名前を置き換えます。名前には、1 ～ 127 の ASCII 文字が文字またはアンダースコア (_) で始まり、英字、数字、アンダースコア (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、ハイフン (-) のみを含む必要があります。<IP>の場合は、仮想サーバに割り当てられた IP アドレスを置き換えます。


```
set ssl vserver <name> -clientAuth ENABLED -clientCert <clientcert>
<!--NeedCopy-->
```

<name>では、作成した仮想サーバーの名前を置き換えます。<clientCert>には、次の値のいずれかを代入します。

- [disabled]：VPN 仮想サーバでのクライアント証明書認証を無効にします。
- [必須（mandatory）]：認証にクライアント証明書を要求するように VPN 仮想サーバを設定します。
- optional：クライアント証明書認証を許可するが、要求しないように VPN 仮想サーバを設定します。

```
bind vpn vserver <name> -policy local
<!--NeedCopy-->
```

<name>で、作成した VPN 仮想サーバーの名前を置き換えます。

```
bind vpn vserver <name> -policy cert
<!--NeedCopy-->
```

<name>の場合は、作成した VPN 仮想サーバーの名前を置き換えます。

```
bind ssl vserver <name> -certkeyName <certkeyname>
<!--NeedCopy-->
```

<name>では、作成した仮想サーバーの名前を置き換えます。<certkeyName>では、クライアント証明書キーを代用します。

```
bind ssl vserver <name> -certkeyName <cacertkeyname> -CA -ocspCheck
    Optional
<!--NeedCopy-->
```

<name>では、作成した仮想サーバーの名前を置き換えます。<cacertkeyName>の場合は、CA 証明書キーを代用します。

```
add ssl action <actname> -clientCert ENABLED -certHeader CLIENT-CERT
<!--NeedCopy-->
```


<actname>では、SSL アクションの名前を置き換えます。

```
add ssl policy <polname> -rule true -action <actname>
<!--NeedCopy-->
```

<polname>では、新しい SSL ポリシーの名前を置き換えます。<actname>では、作成した SSL アクションの名前を置き換えます。

```
bind ssl vserver <name> -policyName <polname> -priority 10
<!--NeedCopy-->
```

<name>で、VPN 仮想サーバーの名前を置き換えます。

例

```
add vpn vserver vs-certpassthru SSL 10.121.250.75 443
set ssl vserver vs-certpassthru -clientAuth ENABLED -clientCert
    optional
bind vpn vserver vs-certpassthru -policy local
bind vpn vserver vs-certpassthru -policy cert
bind ssl vserver vs-certpassthru -certkeyName mycertKey
bind ssl vserver vs-certpassthru -certkeyName mycertKey -CA -ocspCheck
    Optional
add ssl action act-certpassthru -clientCert ENABLED -certHeader CLIENT-
    CERT
add ssl policy pol-certpassthru -rule true -action act-certpassthru
bind ssl vserver vs-certpassthru -policyName pol-certpassthru -priority
     10
<!--NeedCopy-->
```

認証のネゴシエーション

October 7, 2021

他の種類の認証ポリシーと同様に、Negotiate 認証ポリシーは式とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバにバインドし、プライオリティを割り当てます。また、バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。

標準の認証機能に加えて、Negotiate Action コマンドでは、手動で情報を入力する代わりに、キータブファイルからユーザー情報を抽出できるようになりました。キータブに複数の SPN がある場合、認証、承認、および監査によって正しい SPN が選択されます。この機能は、コマンドラインまたは構成ユーティリティを使用して構成できます。

> 注
>
> これらの手順は、すでに LDAP プロトコルに精通しており、選択した LDAP 認証サーバをすでに設定していることを前提としています。

コマンドラインインターフェイスを使用して **keytab** ファイルからユーザー情報を抽出するように認証、承認、および監査を構成するには

コマンドプロンプトで、適切なコマンドを入力します。

```
add authentication negotiateAction <name> {
 -domain <string> }
  {
 -domainUser <string> }
  {
 -domainUserPasswd  }
  [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
     <string>]

set authentication negotiateAction <name> {
 -domain <string> }
  {
 -domainUser <string> }
  {
 -domainUserPasswd }
  [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
     <string>]
<!--NeedCopy-->
```

Parameter description

- 名前 -交渉アクションの名前を使用します。
- ドメイン -サービスプリンシパルのドメイン名があること
  、Citrix の ADC を represnts。
- **domainUser** - Citrix ADC プリンシパルにマップされているアカウントのユーザー名。これは、keytab ファイルが利用できない場合にドメインとパスワードとともに指定できます。ユーザー名が keytab ファイルと一緒に指定されている場合、その keytab ファイルでこのユーザーの資格情報が検索されます。最大長：127
- **domainUserPasswd** - Citrix ADC プリンシパルにマップされているアカウントのパスワード。

- **defaultAuthenticationGroup-** これは、抽出されたグループに加えて、認証が成功したときに選択されるデフォルトのグループです。最大長：63
- **keytab** -Citrix ADC に提示された Kerberos チケットを復号化するために使用される keytab ファイルへのパス。キータブが利用できない場合は、domain/username/password ネゴシエーションアクション設定で指定できます。最大長：127
- **NTLMPath-** サーバーの FQDN を含む、NTLM 認証が有効になっているサイトへのパス。これは、クライアントが NTLM にフォールバックするときに使用されます。最大長：127

設定ユーティリティを使用して **keytab** ファイルからユーザー情報を抽出するための認証、承認、および監査を構成するには

> 注
>
> 構成ユーティリティでは、「アクション」ではなく「サーバー」という用語が使用されますが、同じタスクを指します。

1. **[**セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**認証**] > [**高度なポリシー**] > [**アクション**] > [SAML**] に移動します。
2. 詳細ウィンドウの [サーバー] タブで、次のいずれかの操作を行います。
   - 新しい [ネゴシエート] アクションを作成する場合は、[追加] をクリックします。
   - 既存の [ネゴシエート] アクションを変更する場合は、データウィンドウでアクションを選択し、[編集] をクリックします。
3. 新しい [ネゴシエート] アクションを作成する場合は、[名前] テキストボックスに新しいアクションの名前を入力します。名前の長さは 1 ～127 文字で、大文字、小文字、数字、ハイフン (-) およびアンダースコア (_) を使用できます。既存の「ネゴシエート」アクションを変更する場合は、この手順をスキップします。名前は読み取り専用です。変更できません。
4. [ネゴシエート] で、[キータブファイルを使用] チェックボックスがオンになっていない場合は、オンにします。
5. 「キータブファイルのパス」テキストボックスに、使用するキータブファイルのフルパスとファイル名を入力します。
6. [既定の認証グループ] テキストボックスに、このユーザーの既定として設定する認証グループを入力します。
7. [作成**]** または **[OK]** をクリックして変更を保存します。

Kerberos 認証に高度な暗号化を使用する場合の注意点

> - キータブを使用する場合の設定例：認証 negotiateAction を追加 neg_act_aes256 -キータブ “/nsconfig/krb/lbvs_aes256.keytab”
> - **keytab** に複数の暗号化タイプがある場合は、次のコマンドを使用します。このコマンドは、ドメインユーザーパラメータを追加でキャプチャします。addauthenticationnegotiateAction neg_act_keytab_all -キータブ “/nsconfig/krb/lbvs_all.keytab” -domainUser

> “HTTP/lbvs.aaa.local”
> - ユーザー資格情報を使用する場合は、次のコマンドを使用します。**addauthentication** negotiateAction neg_act_user -domain AAA.LOCAL -domainUser “HTTP/lbvs.aaa.local” -domainUserPasswd <password>
> - 正しい **domainUser** 情報が提供されていることを確認してください。AD でユーザーログオン名を探すことができます。

Web 認証

October 7, 2021

認証、認可、および監査で、Web サーバが HTTP 要求で要求するクレデンシャルを提供し、Web サーバの応答を分析して、ユーザ認証が成功したかどうかを判断できるようになりました。他の種類の認証ポリシーと同様に、Web 認証ポリシーは式とアクションで構成されます。認証ポリシーを作成したら、それを認証仮想サーバにバインドし、プライオリティを割り当てます。また、バインドするときは、プライマリポリシーまたはセカンダリポリシーとして指定します。

特定の Web サーバで Web ベース認証を設定するには、まず Web 認証アクションを作成します。Web サーバへの認証では厳格な形式が使用されないため、Web サーバが必要とする情報と、アクションの作成時にどの形式を使用するかを正確に指定する必要があります。これを行うには、Citrix ADC アプライアンスのデフォルト構文で、次の項目を含む式を作成します。

- サーバ **IP**：認証 Web サーバの IP アドレス。
- サーバポート：認証 Web サーバのポート。
- 認証規則：Citrix ADC アプライアンスのデフォルト構文の式。Web サーバーが想定する形式でユーザーの資格情報を格納します。
- **[Scheme]**：HTTP（暗号化されていない Web 認証の場合）または HTTPS（暗号化された Web 認証の場合）。
- 成功規則：Citrix ADC アプライアンスのデフォルトの構文で、ユーザーが正常に認証されたことを示す Web サーバーの応答文字列と一致します。

他のすべてのパラメータについては、add authentication action コマンドの通常の規則に従います。

次に、そのアクションに関連付けられたポリシーを作成します。ポリシーは LDAP ポリシーに似ており、LDAP ポリシーと同様に Citrix ADC アプライアンスの構文を使用します。

> 注
>
> これらの手順は、認証する Web サーバの認証要件をすでに理解しており、Web 認証サーバをすでに設定していることを前提としています。

コマンドラインインターフェイスを使用して **Web** 認証アクションを構成するには

コマンドラインで Web 認証アクションを作成するには、コマンドラインで次のコマンドを入力します。

```
1 add authentication webAuthAction <name> -serverIP <ip_addr|ipv6_addr
    |\*> -serverPort <port|\*> [-fullReqExpr <string>] -scheme ( http |
    https ) -successRule <expression> [-defaultAuthenticationGroup <
    string>][-Attribute1 <string>][-Attribute2 <string>] [-Attribute3 <
    string>][-Attribute4 <string>] [-Attribute5 <string>][-Attribute6 <
    string>] [-Attribute7 <string>][-Attribute8 <string>] [-Attribute9 <
    string>][-Attribute10 <string>] [-Attribute11 <string>][-Attribute12
     <string>] [-Attribute13 <string>][-Attribute14 <string>] [-
    Attribute15 <string>][-Attribute16 <string>]
2 <!--NeedCopy-->
```

例

```
1 add policy expression post_data ""username=" + http.REQ.BODY(1000).
    SET_TEXT_MODE(IGNORECASE).AFTER_STR("login=").BEFORE_STR("&") + "&
    passwort=" + http.REQ.BODY(1000).SET_TEXT_MODE(IGNORECASE).AFTER_STR
    ("passwd=")"
2
3 add policy expression length_post_data "("username= " + http.REQ.BODY
    (1000).SET_TEXT_MODE(IGNORECASE).AFTER_STR("login=").BEFORE_STR("&")
     + "passwort=" + http.REQ.BODY(1000).SET_TEXT_MODE(IGNORECASE).
    AFTER_STR("passwd=")).length"
4
5 add authentication webAuthAction webAuth_POST -serverIP 10.106.187.54 -
    serverPort 80 -fullReqExpr q{
6  "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." + http.req
    .version.major + "\r\nAccept:\*/\*\r\nHost: 10.106.187.54\r\
    nReferer: http://10.106.187.54/MyPHP/auth.php\r\nAccept-Language:
    en-US\r\nUser-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT
    6.1; Trident/5.0)\r\nContent-Type: application/x-www-form-
    urlencoded\r\n" + "Content-Length: " + length_post_data + "\r\
    nConnection: Keep-Alive\r\n\r\n" + post_data }
7   -scheme http -successRule "http.res.status.eq(200)"
8 <!--NeedCopy-->
```

構成ユーティリティを使用して **Web** 認証アクションを構成するには

> 注
>
> 構成ユーティリティでは、「アクション」ではなく「サーバー」という用語が使用されますが、同じタスクを指します。

1. [セキュリティ] **> [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [LDAP]** に移動します。
2. 詳細ウィンドウの [サーバー] タブで、次のいずれかの操作を行います。
   - 新しい Web 認証アクションを作成する場合は、[**Add**] をクリックします。
   - 既存の Web 認証アクションを変更する場合は、データペインでアクションを選択し、[編集] をクリックします。
3. 新しい Web 認証アクションを作成する場合は、[認証 **Web** サーバーの作成] ダイアログボックスの [名前] ボックスに、新しい Web 認証アクションの名前を入力します。名前の長さは 1 ～127 文字で、大文字、小文字、数字、ハイフン (-) およびアンダースコア (_) を使用できます。既存の Web 認証アクションを変更する場合は、この手順をスキップします。名前は読み取り専用です。変更できません。
4. [**Web** サーバーの **IP** アドレス] テキストボックスに、認証 Web サーバーの IPv4 または IPv6 IP アドレスを入力します。アドレスが IPv6 IP アドレスの場合は、最初に [IPv6] チェックボックスをオンにします。
5. 「ポート」テキストボックスに、Web サーバーが接続を受け入れるポート番号を入力します。
6. [プロトコル] ドロップダウンリストで [**HTTP**] または [**HTTPS**] を選択します。
7. [HTTP 要求式] テキスト領域で、認証 Web サーバーで想定される正確な形式で、ユーザーの資格情報を含む Web サーバー要求を作成する PCRE-形式の正規表現を入力します。
8. [認証を検証する式] テキスト領域に、Citrix ADC アプライアンスのデフォルトの構文式を入力します。この式には、ユーザー認証が成功したことを示す Web サーバー応答の情報を記述します。
9. 一般的な認証アクションのマニュアルで説明されているように、残りのフィールドに入力します。
10. [**OK**] をクリックします。

Web 認証を使用した **SMS** 二要素認証

June 1, 2022

Citrix ADC をサードパーティの SMS プロバイダーと統合して、追加の認証レイヤーを提供できるようになりました。

Citrix ADC アプライアンスは、認証の第 2 要素としてユーザーのモバイルで OTP を送信するように構成できます。アプライアンスは、AD ログインの成功後に OTP に入るためのログオンフォームをユーザーに提示します。SMS OTP 認証が正常に検証されて初めて、要求されたリソースがユーザーに提示されます。

SMS OTP 認証を実現するために、Citrix ADC アプライアンスはバックエンドの次の要素に依存しています。

1. LDAP 認証を使用してユーザーを認証し、ユーザーの携帯電話番号を抽出します。

2. OTP を作成し、NS 変数に格納します。変数の設定と使用。
3. LDAP から抽出した携帯電話番号に WebAuth 認証方式で OTP を送信します。
4. OTP を検証します。

前提条件

OTP ストアを構成する

管理者は、次の CLI コマンドを使用して、SMS 認証に使用される OTP を保存するデータベース/ストアを設定します。

```
1 add ns variable otp_store -type "map(text(65),text(6),100000)" -
     ifValueTooBig undef -ifNoValue undef -expires 5
2 <!--NeedCopy-->
```

ユーザーセッションごとにランダム **OTP** を生成

次のコマンドを使用して、ユーザーセッションごとに 6 桁のランダム OTP を生成し、OTP ストアに保存します。

```
1 add ns assignment generate_otp -variable "$otp_store[AAA.USER.SESSIONID
     ]" -set ("000000" + SYS.RANDOM.MUL(1000000).
     TYPECAST_UNSIGNED_LONG_AT.TYPECAST_TEXT_T).SUFFIX(6)
2 <!--NeedCopy-->
```

Citrix ADC で SMS OTP 認証を構成する

- SMS 二要素認証機能を構成する前に、Citrix ADC アプライアンスで認証を有効にした第 1 要素として LDAP 認証を構成する必要があります。LDAP 認証を設定する手順については、「構成ユーティリティを使用して LDAP 認証を構成するには」を参照してください。
- LDAP を設定し、SMS OTP 認証に使用する携帯電話番号を抽出します。

第 **1** 要素構成のサンプル

```
1 add authentication ldapAction ldap_action -serverIP 1.1.1.1 -serverPort
      3268 -authTimeout 30 -ldapBase "dc=nsi-test,dc=com" -ldapBindDn
     Administrator@nsi-test.com -ldapBindDnPassword freebsd -
     ldapLoginName samaccountname -groupAttrName memberOf -
     ssoNameAttribute samaccountname  -Attribute1 mobile -email mail
```



```
2
3 add authentication Policy ldap_policy -rule true -action ldap_action
4 <!--NeedCopy-->
```

注

携帯電話番号は AAA.USER.ATTRIBUTE (1) を使用して抽出でき、バックエンドサーバーに送信するときに含めることができます。

第 2 要素構成のサンプル

次のサンプル設定を使用して、エンドユーザーに送信される OTP が生成されます。

```
1 add authentication Policy set_otp -rule true -action generate_otp
2
3 add authentication policylabel set_otp -loginSchema LSCHEMA_INT
4
5 add authentication policy cascade_noauth -rule true -action NO_AUTHN
6
7 bind authentication policylabel set_otp -policyName set_otp -priority 1
      -gotoPriorityExpression NEXT
8 <!--NeedCopy-->
```

第 3 要素構成の例

次のサンプル構成を使用して、第 2 要素構成で生成された OTP が Web 認証方法を使用してエンドユーザーに送信されます。Web 認証の詳細については、Web 認証を参照してください。

- SMS サーバーが GET メソッドを介して API を公開する場合の Web 認証設定の例。

```
1  add policy expression otp_exp_get ""method=sendMessage&send_to="
      + AAA.USER.ATTRIBUTE(1) + "&msg=OTP is " + $otp_store[AAA.USER
      .SESSIONID] + "for login into secure access  gateway. Valid
      till EXPIRE_TIME. Do not share the OTP with anyone for
      security reasons.&userid=######&password=###=1.0""
2
3  add authentication webAuthAction webAuth_Get -serverIP
      10.106.168.210 -serverPort 8080 -fullReqExpr q{
4  "GET /GatewayAPI/rest?" + otp_exp_get + "HTTP/" + http.req.
      version.major + "." + http.req.version.minor.sub(1) + "\r\
      nAccept:\*/\*\r\nHost: <FQDN>\r\n" }
```



```
 -successRule "http.res.status.eq(200)" -scheme http
<!--NeedCopy-->
```

- SMS サーバーが GET メソッドを介して API を公開する場合の Web 認証設定の例。

```
add policy expression otp_exp_post ""Message: OTP is " +
    $otp_store[AAA.USER.SESSIONID] + "for login into secure access
      gateway. Valid till EXPIRE_TIME. Do not share the OTP with
    anyone for security reasons&Mobile:" + AAA.USER.ATTRIBUTE(1)"

add authentication webAuthAction webAuth_POST -serverIP
    10.106.168.210 -serverPort 8080 -fullReqExpr q{
"POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." +
    http.req.version.major + "\r\nAccept:\*/\*\r\nHost:
    10.106.168.210 \r\nContent-Length: 10\r\n\r\n" + otp_exp_post
    }
 -scheme http -successRule true
<!--NeedCopy-->
```

```
add authentication webAuthAction webAuth_Get -serverIP
    10.106.168.210 -serverPort 8080 -fullReqExpr q{
"GET /GatewayAPI/rest?" + otp_exp_get + "HTTP/" + http.req.
    version.major + "." + http.req.version.minor.sub(1) + "\r\
    nAccept:/\r\nHost: <FQDN>\r\n" }
 -successRule "http.res.status.eq(200)" -scheme http

add policy expression otp_exp_post "$otp_store[AAA.USER.SESSIONID
    ]"
<!--NeedCopy-->
```

- 最後に、OTP を送信します。

```
add authentication Policy wpp -rule true -action webAuth_POST

add authentication policylabel send_otp -loginSchema LSCHEMA_INT
bind authentication policylabel send_otp -policyName wpp -
    priority 1 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```


第 **4** の要素構成の例

次のサンプル設定を使用して、エンドユーザーに送信された OTP を検証します。

この構成では、ポリシールールを使用して、エンドユーザーに送信される OTP に対して OTP を検証します。

```
add authentication Policy otp_verify -rule ”AAA.LOGIN.PASSWORD.EQ(
    $otp_store[AAA.USER.SESSIONID])” -action NO_AUTHN

add authentication policylabel otp_verify -loginSchema onlyPassword

bind authentication policylabel otp_verify -policyName otp_verify -
    priority 1 -gotoPriorityExpression NEXT

<!--NeedCopy-->
```

次のコマンドを使用して、OnlyPassword ログインスキーマを追加します。

```
add authentication loginSchema onlypassword -authenticationschema /
    nsconfig/loginschema/LoginSchema/OnlyPassword.xml”
<!--NeedCopy-->
```

SMS OTP 認証を成功させるためのすべての要素をリンクする

次の CLI コマンドを使用して、すべての要素をリンクします。

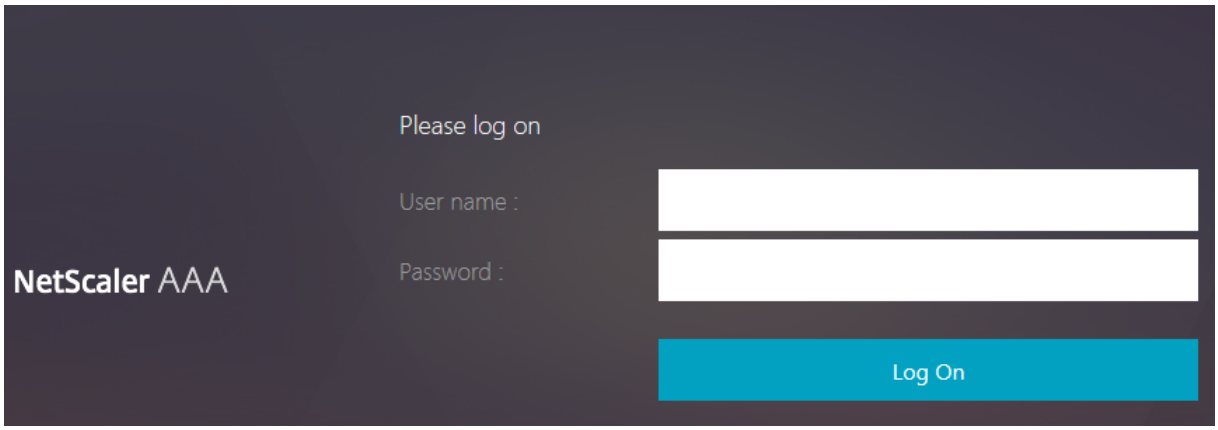
```
bind authentication policylabel send_otp -policyName wpp -priority 1 -
    gotoPriorityExpression NEXT -nextFactor otp_verify
<!--NeedCopy-->
```

> 注:
>
> カスケード認証のポリシーが追加され、エンドユーザーの信頼性が高く継続的な認証が可能になります。現在の要因が失敗した場合、ユーザーエクスペリエンスに影響がないように次の要素が評価されます。

フォームベース認証

October 7, 2021

フォームベースの認証では、ログオンフォームがエンドユーザーに提示されます。このタイプの認証フォームは、多要素（nFactor）認証とクラシック認証の両方をサポートしています。



フォームベースの認証が機能するには、次のことを確認してください。

- 負荷分散仮想サーバーでは、認証が **ON** になっている必要があります。
- 「authenticationHost」パラメータを指定する必要があります。このパラメータには、ユーザーが認証のためにリダイレクトされる必要があります。これを設定するためのコマンドは次のとおりです：

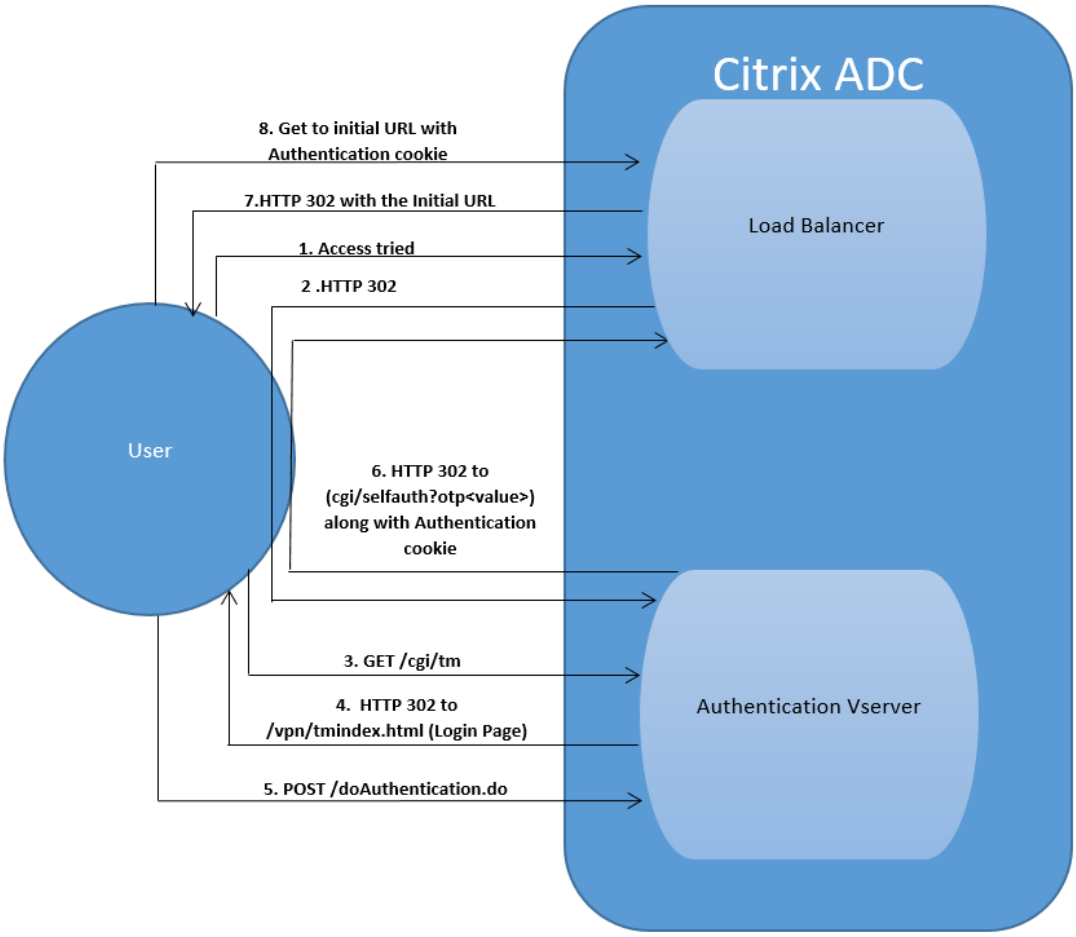
```
1  set lb vs lb1 -authentication on  - authenticationhost aaavs-ip/
       fqdn
```

- フォームベース認証は、HTML をサポートするブラウザと互換性があります

次の手順では、フォームベースの認証がどのように機能するかを説明します。

1. クライアント（ブラウザ）は、TM（ロードバランシング/CS）仮想サーバー上の URL に対する GET 要求を送信します。
2. TM 仮想サーバーは、クライアントが認証されていないと判断し、HTTP 302 応答をクライアントに送信します。応答には、クライアントが認証仮想サーバーに/cgi/tm の GET 要求を発行する隠しスクリプトが含まれています。
3. クライアントは、ターゲット URL を含む GET /cgi/tm を認証仮想サーバに送信します。
4. 認証仮想サーバは、ログインページへのリダイレクトを送信します。
5. ユーザーは、POST /doAuthentication.do を使用して認証仮想サーバーに資格情報を送信します。認証は、認証仮想サーバーによって行われます。
6. 資格情報が正しい場合、認証仮想サーバーは、負荷分散サーバーの cgi/selfauth url に HTTP 302 応答をワンタイムトークン（OTP）で送信します。
7. 負荷分散サーバーは HTTP 302 をクライアントに送信します。

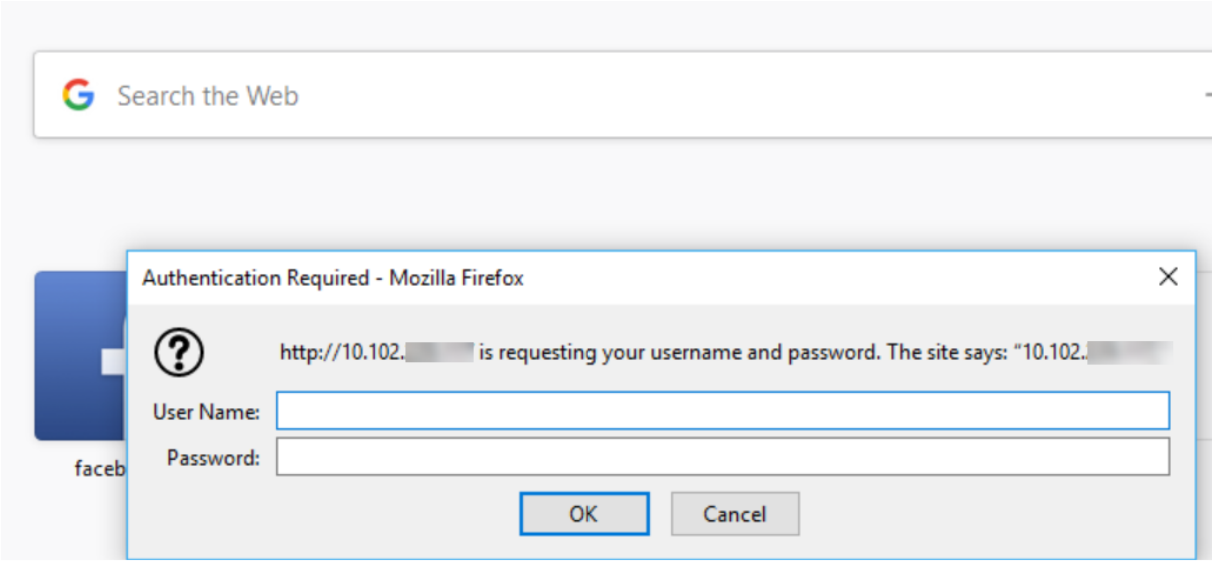
8. クライアントは、32 バイトの cookie とともに、最初の URL のターゲット URL の GET 要求を送信します。



401 ベースの認証

May 9, 2022

401 ベースの認証では、Citrix ADC アプライアンスはエンドユーザーにポップアップダイアログボックスを表示します。



フォームベースの AAA-TM はリダイレクトメッセージで動作します。ただし、リダイレクトをサポートしていないアプリケーションもあります。このようなアプリケーションでは、401 認証対応の AAA-TM が使用されます。

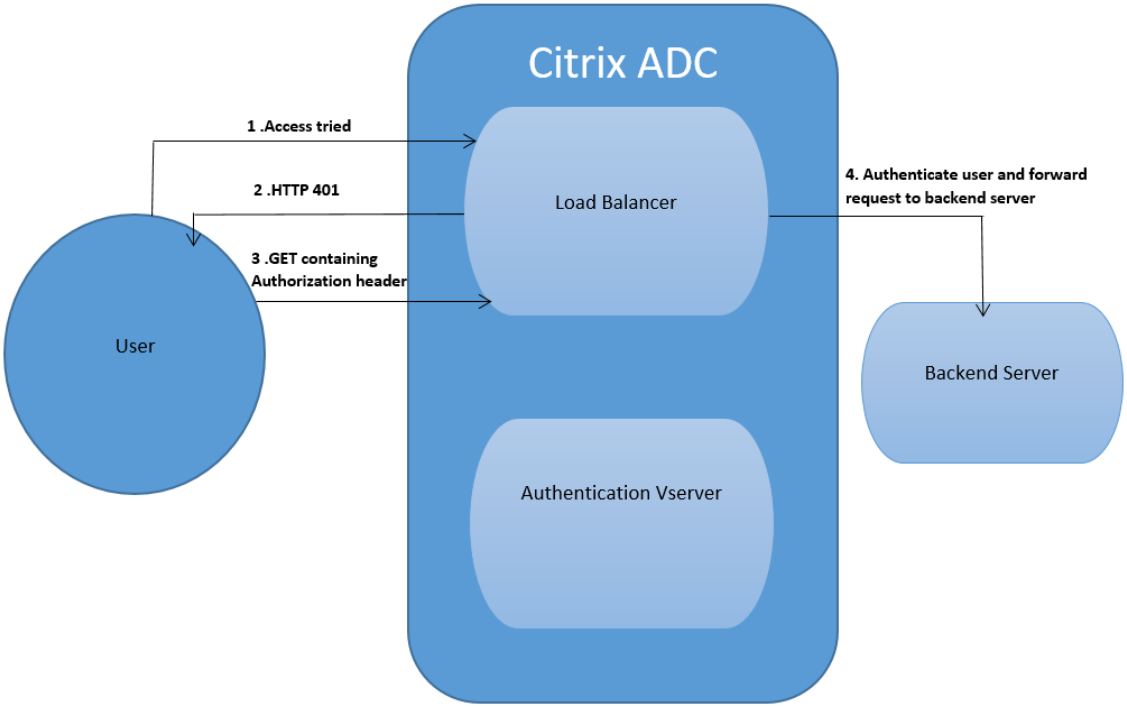
401 認証対応 AAA-TM が機能するには、次の点を確認してください。

- 負荷分散仮想サーバーの「AuthNvsName」パラメーター値は、ユーザーの認証に使用する認証仮想サーバーの名前である必要があります。
- 'authn401' パラメータを有効にする必要があります。これを設定するためのコマンドは次のとおりです:

```
1  set lb vs lb1 -authn401 on -authnvsName <aaavs-name>
```

次の手順では、401 認証の動作について説明します。

1. ユーザーが負荷分散仮想サーバーを使用して特定の URL にアクセスしようとしました。
2. 負荷分散仮想サーバーは、アクセスに認証が必要であることを示す 401 HTTP 応答をユーザーに送信します。
3. ユーザーは、認証ヘッダーで資格情報を負荷分散仮想サーバーに送信します。
4. 負荷分散仮想サーバーはユーザーを認証し、そのユーザーをバックエンドサーバーに接続します。



> 重要:
>
> 401 認証がオンになっている負荷分散仮想サーバーでは、同じユーザーに対して短時間で複数の認証および承認セッションが作成されることがあります。これにより、メモリが急増する可能性があります。この場合、Citrix ADC アプライアンスに次の構成を適用して、エンドクライアントアプリケーションをデバッグおよび識別できます。

```
>set syslogparams -userDefinedAuditlog yes
>
>add audit messageaction 401_log_act InFORMATIONAL '"LB-401 accessed: User: <" + AAA.USER.NAME + "> SessionID <"+ AAA.USER.SESSIONID + "> Client :<" + CLIENT.IP.SRC + "> accessed URL: <" + HTTP.REQ.URL + ">"'
>
>add rewritepolicy rewrite_401_log true NOREWRITE -logAction 401_log_act
>
>bind lb vserver <lb_name> -policyName rewrite_401_log -priority 100 -type reqUEST

<!--NeedCopy-->
```


nFactor 認証用の再キャプチャ設定

October 7, 2021

Citrix Gateway は、新しいファーストクラスのアクション「captchaAction」をサポートし、再キャプチャの構成を簡素化します。reCaptcha はファーストクラスのアクションであるため、独自の要因になる可能性があります。nFactor フローのどこにでも reCaptcha を注入することができます。

以前は、RfWeb UI の変更を含むカスタム WebAuth ポリシーを記述する必要がありました。キャプチャアクションの導入により、JavaScript を変更する必要はありません。

> 重要
>
> reCaptcha がスキーマ内のユーザー名またはパスワードのフィールドと一緒に使用されている場合、reCaptcha が満たされるまで送信ボタンは無効になります。

再キャプチャ設定

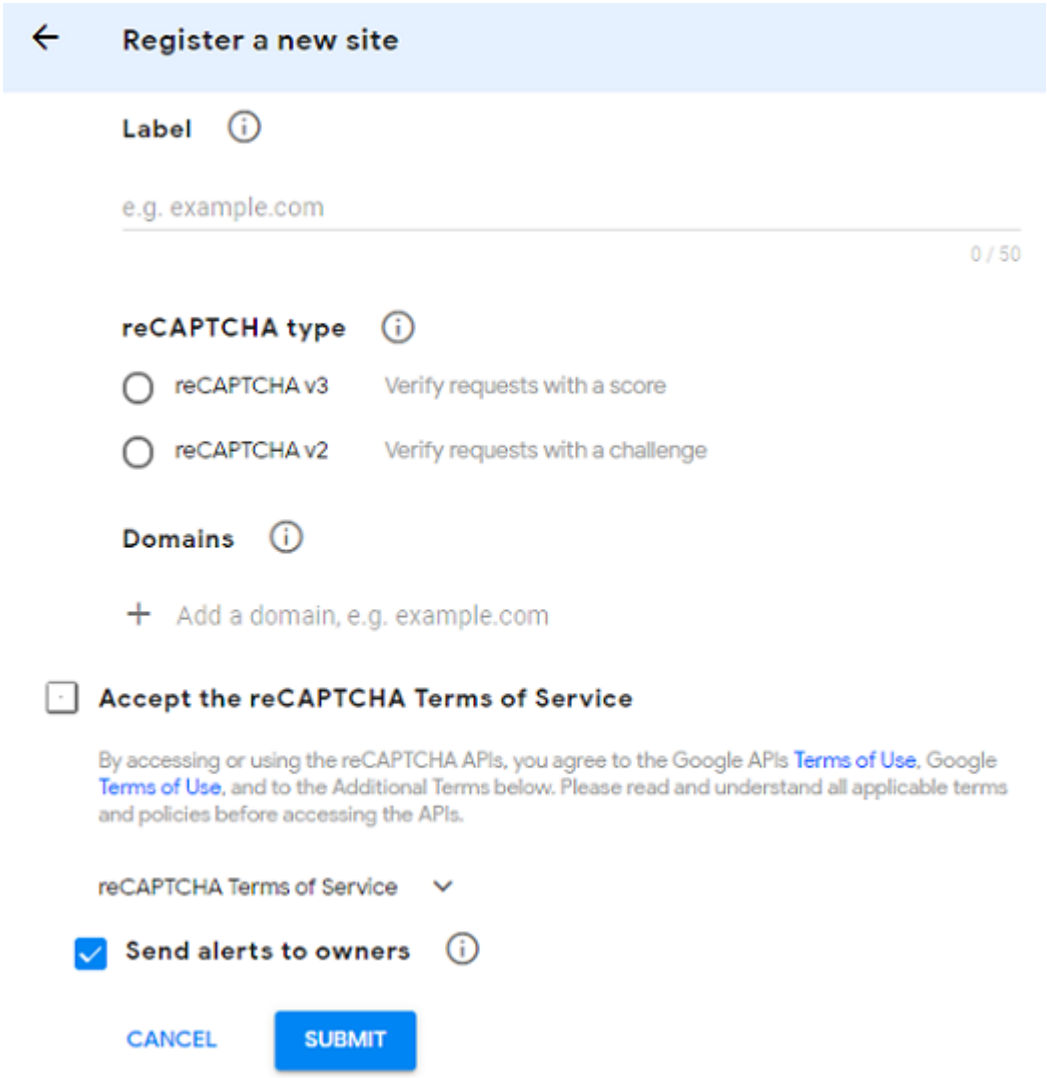
reCaptcha 構成には、2 つの部分が含まれます。

1. 再キャプチャを登録するための Google の構成。
2. ログインフローの一部として再キャプチャを使用する Citrix ADC アプライアンスの構成。

Google で再キャプチャ設定

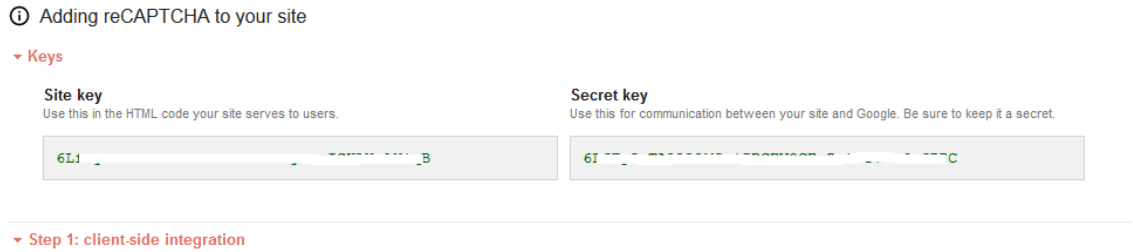
https://www.google.com/recaptcha/adminで reCaptcha のドメインを登録します。

1. このページに移動すると、次の画面が表示されます。



> 注
>
> 再 CAPTCHA v2 のみを使用してください。目に見えない reCAPTCHA はまだベータ版です。

2. ドメインを登録すると、「サイトキー」と「秘密キー」が表示されます。



> 注
>
> セキュリティ上の理由から、「SiteKey」と「SecretKey」はグレー表示になっています。「SecretKey」

> は安全に保管する必要があります。

Citrix ADC アプライアンスの構成を再キャプチャする

Citrix ADC アプライアンスの再キャプチャ構成は、3 つの部分に分けることができます。

- 再キャプチャ画面を表示する
- Google サーバーに再キャプチャ応答を投稿する
- LDAP 構成は、ユーザーログオンの 2 番目の要素です (オプション)

再キャプチャ画面を表示する

ログインフォームのカスタマイズは、SingleAuthCaptcha.xml Loginschema を介して行われます。このカスタマイズは、認証仮想サーバーで指定され、ログインフォームをレンダリングするために UI に送信されます。組み込みの Loginschema である SingleAuthCaptcha.xml は、Citrix ADC アプライアンス上の/nsconfig/loginschema/Loginschema ディレクトリにあります。

> 重要
>
> - ユースケースと異なるスキーマに基づいて、既存のスキーマを変更できます。たとえば、reCaptcha 係数（ユーザー名またはパスワードなし）または reCaptcha との二重認証のみが必要な場合。
> - カスタム変更を実行した場合、またはファイルの名前を変更した場合は、すべての loginSchema を/nsconfig/loginschema ディレクトリから親ディレクトリ/nsconfig/loginschema にコピーすることをお勧めします。

CLI を使用して再キャプチャの表示を設定するには

- `add authentication loginSchema singleauthcaptcha -authenticationSchema /nsconfig/loginschema/SingleAuthCaptcha.xml`
- `add authentication loginSchemaPolicy singleauthcaptcha -rule true -action singleauthcaptcha`
- `add authentication vserver auth SSL <IP> <Port>`
- `add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-key-file>`
- `bind ssl vserver auth -certkey vserver-cert`
- `bind authentication vserver auth -policy singleauthcaptcha -priority 5 -gotoPriorityExpression END`

Google サーバーに再キャプチャ応答を投稿する

あなたは、ユーザーに表示されなければならない reCaptcha を設定した後、管理者は、ブラウザからの reCaptcha 応答を確認するために、Google サーバーに構成を追加ポスト。

ブラウザから再キャプチャ応答を確認するには

- add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>
- add authentication policy myrecaptcha -rule **true** -action myrecaptcha
- bind authentication vserver auth -policy myrecaptcha -priority 1

AD 認証が必要な場合は、次のコマンドが必要です。それ以外の場合は、この手順を無視できます。

- add authentication ldapAction ldap-**new** -serverIP x.x.x.x -serverPort 636 -ldapBase ”cn=users,dc=aaatm,dc=com”-ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup
- add authenticationpolicy ldap-**new** -rule **true** -action ldap-**new**

LDAP 構成は、ユーザーログオンの **2** 番目の要素です **(**オプション**)**

LDAP 認証は reCaptcha の後に行われ、2 番目の要素に追加します。

- add authentication policylabel second-factor
- bind authentication policylabel second-factor -policy ldap-**new** -priority 10
- bind authentication vserver auth -policy myrecaptcha -priority 1 -nextFactor second-factor

管理者は、負荷分散仮想サーバーと Citrix Gateway アプライアンスのどちらを使用してアクセスするかに応じて、適切な仮想サーバーを追加する必要があります。ロードバランシング仮想サーバが必要な場合は、管理者が次のコマンドを設定する必要があります。

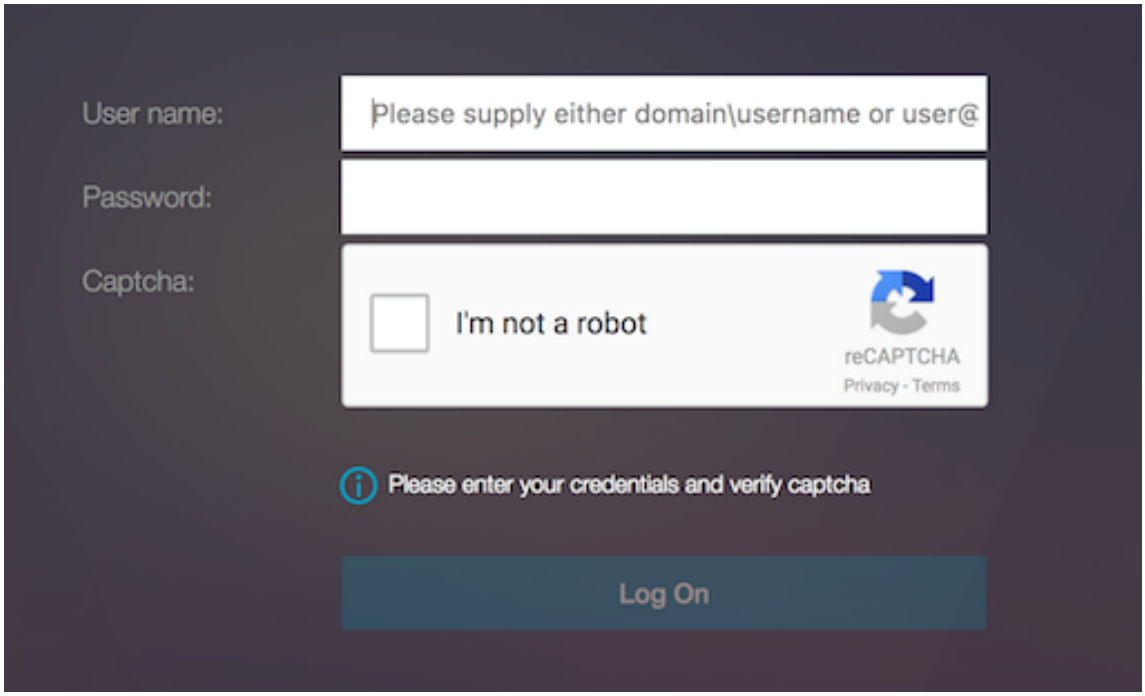
- add lb vserver lbtest HTTP <IP> <Port> -authentication ON -authenticationHost nssp.aaatm.com

  nssp.aaatm.com — 認証仮想サーバーに解決します。

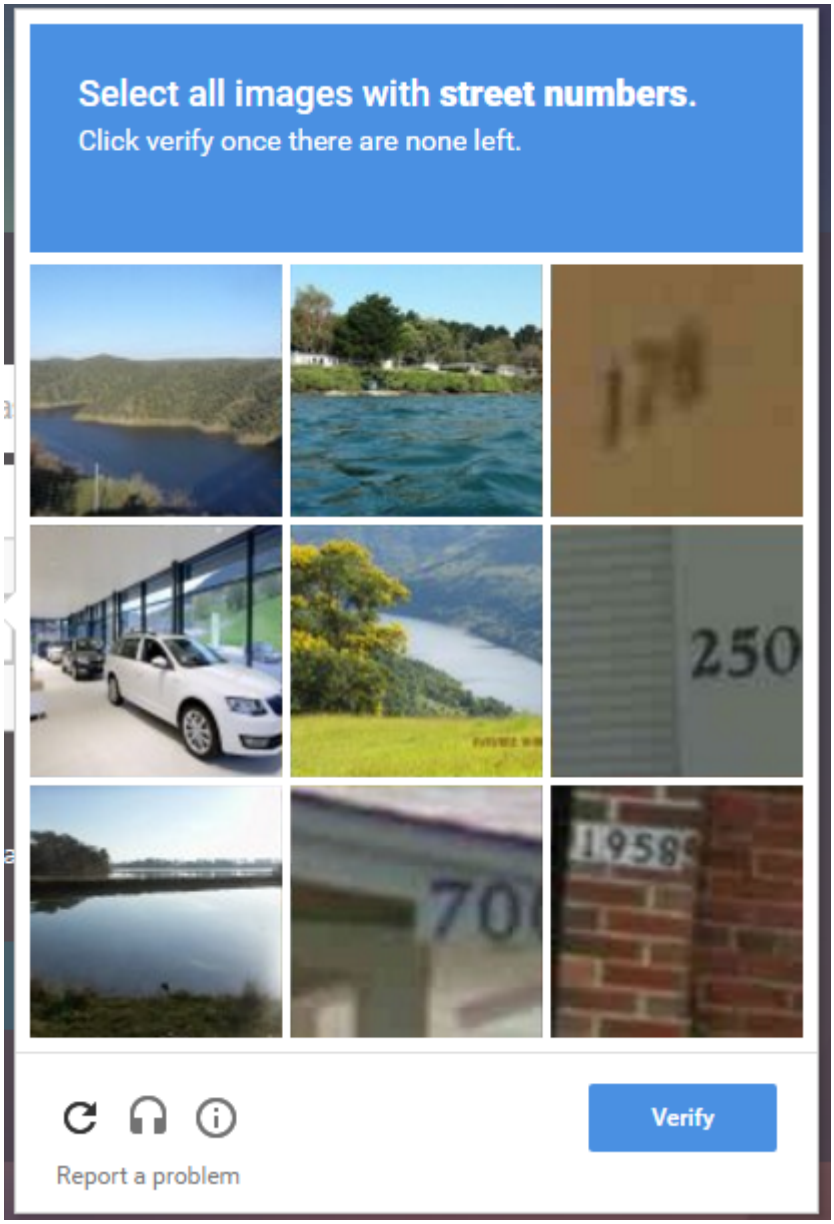
再キャプチャのユーザー検証

前のセクションで説明したすべての手順を設定したら、以下に示す UI のスクリーンショットを確認する必要があります。

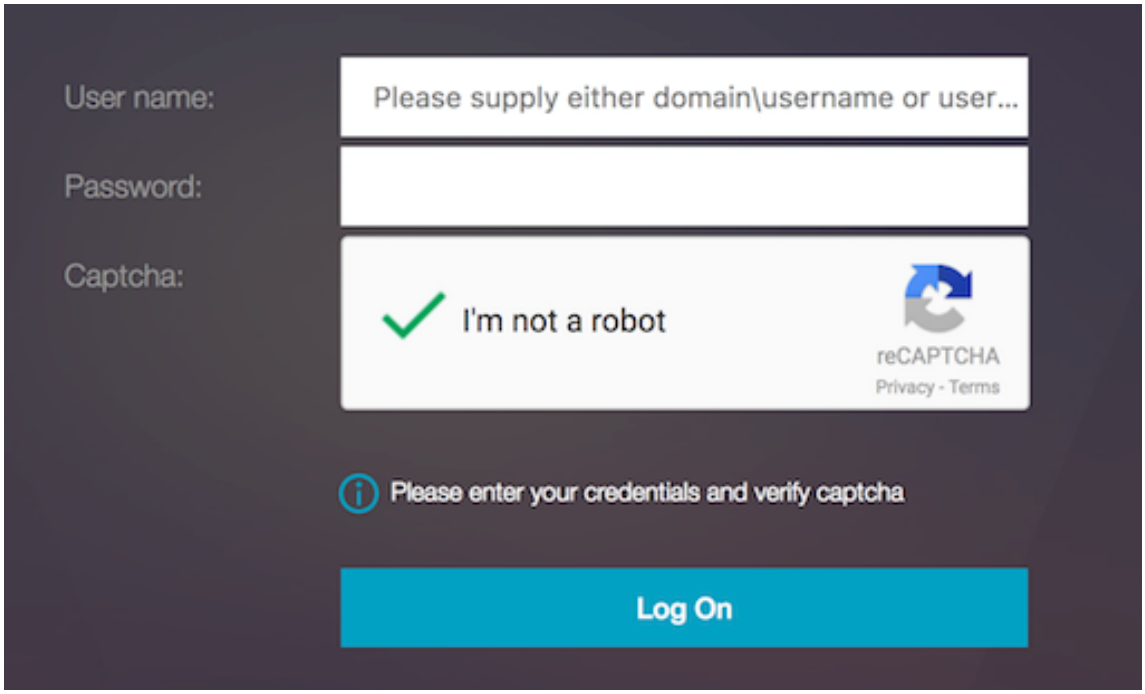
1. 認証仮想サーバーがログインページを読み込むと、ログオン画面が表示されます。ログオンは、再キャプチャが完了するまで無効になります。



2. [私はロボットではありません] オプションを選択します。再キャプチャウィジェットが表示されます。



3. 完了ページが表示される前に、一連の reCaptcha イメージ間を移動します。
4. AD 資格情報を入力し、[ロボットではありません] チェックボックスをオンにして、[ログオン] をクリックします。認証が成功すると、目的のリソースにリダイレクトされます。



メモ

- reCaptcha が AD 認証で使用されている場合、reCaptcha が完了するまで、資格情報の送信ボタンは無効になります。
- reCaptcha は、独自の要因で発生します。したがって、AD のような後続の検証は、reCaptcha の 'nextfactor' で発生する必要があります。

認証に対するネイティブ **OTP** サポート

July 15, 2022

Citrix ADC は、サードパーティのサーバーを使用せずにワンタイムパスワード（OTP）をサポートします。ワンタイムパスワードは、生成される番号またはパスコードがランダムであるため、セキュリティで保護されたサーバーを認証するための非常に安全なオプションです。以前は、乱数を生成する特定のデバイスを持つ RSA などの専門企業が、OTP を提供していました。

この機能は、設備コストと運用コストの削減に加えて、Citrix ADC アプライアンスの構成全体を維持することで、管理者の管理を強化します。

> 注:
>
> サードパーティのサーバーは不要になったため、Citrix ADC 管理者はユーザーデバイスを管理および検証するためのインターフェイスを構成する必要があります。

OTP ソリューションを使用するには、ユーザーを Citrix ADC 仮想サーバーに登録する必要があります。登録は一意のデバイスごとに一度だけ必要で、特定の環境に制限することができます。登録ユーザーの設定と検証は、追加の認

証ポリシーの設定と似ています。

ネイティブ **OTP** をサポートすることの利点

- Active Directory に加えて、認証サーバー上に追加のインフラストラクチャを用意する必要がなくなるため、運用コストが削減されます。
- 構成を Citrix ADC アプライアンスにのみ統合し、管理者に優れた制御を提供します。
- クライアントが期待する数値を生成するために、追加の認証サーバーへのクライアントの依存を排除します。

ネイティブ **OTP** ワークフロー

ネイティブ OTP ソリューションは 2 つのプロセスであり、ワークフローは次のように分類されます。

- デバイス登録
- エンドユーザログイン

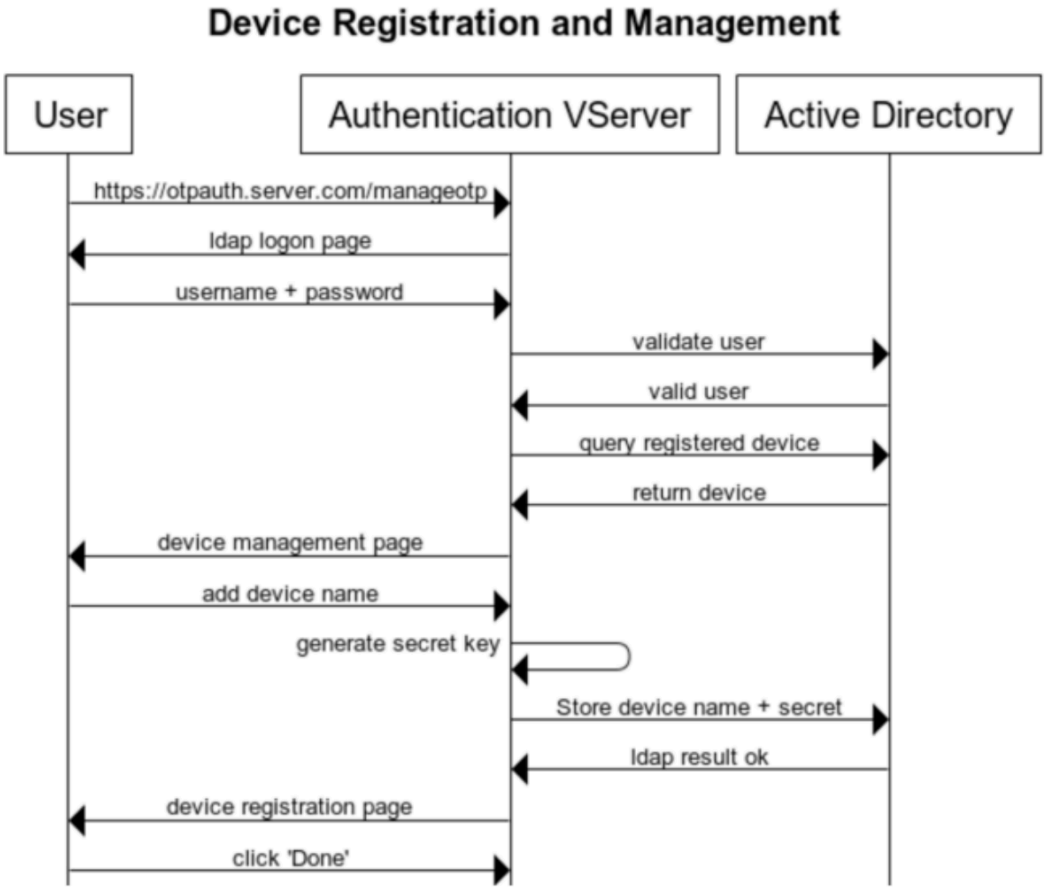
> 重要:
>
> サードパーティのソリューションを使用している場合、または Citrix ADC アプライアンス以外の他のデバイスを管理している場合は、登録プロセスをスキップできます。追加する最後の文字列は、Citrix ADC で指定された形式である必要があります。

次の図は、OTP を受信する新しいデバイスを登録するためのデバイス登録フローを示しています。

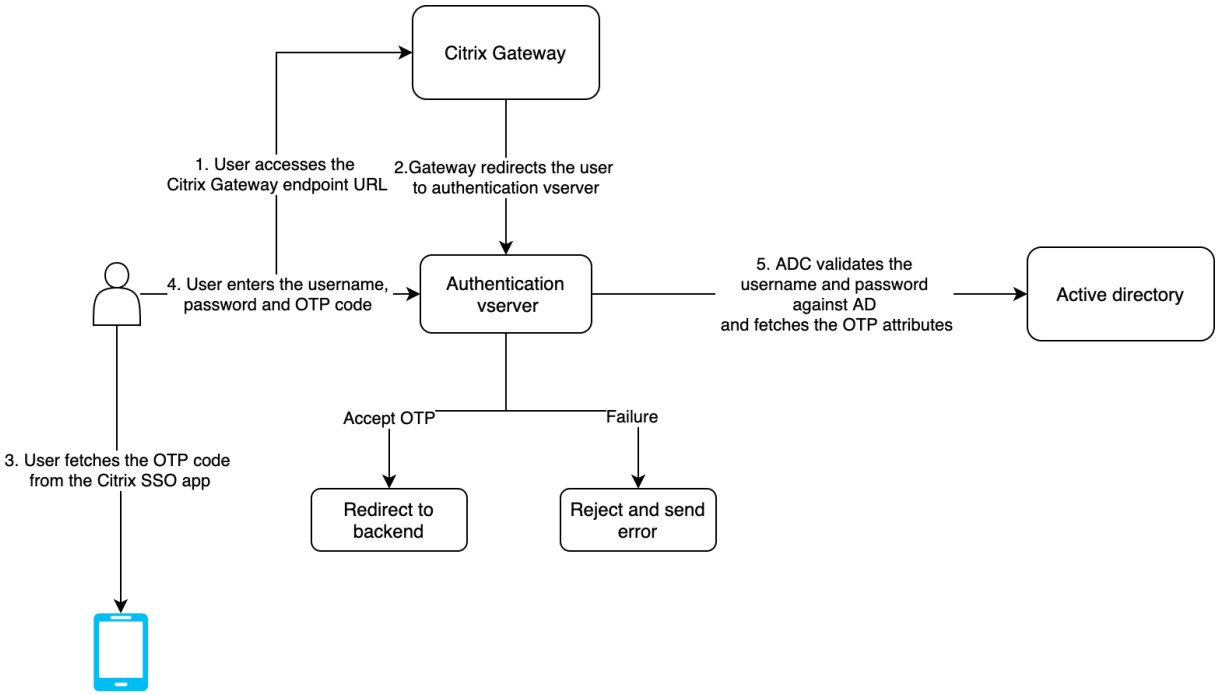
注: デバイスの登録は、任意の数の要素を使用して実行できます。デバイス登録プロセスを説明するために、1 つの要素（前の図で指定）を例として使用します。

次の図は、登録済みデバイスを介した OTP の検証を示しています。

次の図は、デバイスの登録と管理フローを示しています。



次の図は、ネイティブ OTP 機能のエンドユーザフローを示しています。



前提条件

ネイティブ OTP 機能を使用するには、次の前提条件が満たされていることを確認してください。

- Citrix ADC 機能のリリースバージョンは 12.0 ビルド 51.24 以降です。
- アドバンスドエディションまたはプレミアムエディションのライセンスが Citrix Gateway にインストールされている。
- Citrix ADC は管理 IP で構成されており、管理コンソールにはブラウザとコマンドラインの両方を使用してアクセスできます。
- Citrix ADC は、ユーザーを認証するための認証、承認、監査仮想サーバーで構成されています。詳細については、「認証仮想サーバー」を参照してください。
- Citrix ADC アプライアンスは Unified Gateway で構成され、認証、承認、監査プロファイルが Gateway 仮想サーバーに割り当てられます。
- ネイティブ OTP ソリューションは、nFactor 認証フローに制限されています。ソリューションを構成するには、高度なポリシーが必要です。詳細については、ネイティブ OTPを参照してください。

また、Active Directory については、次の点を確認してください。

- 属性の最小長は 256 文字です。
- 属性タイプは、ユーザパラメータなどの 'DirectoryString' でなければなりません。これらの属性は文字列値を保持できます。
- デバイス名が英語以外の文字である場合、属性文字列タイプは Unicode である必要があります。
- Citrix ADC LDAP 管理者は、選択した AD 属性への書き込みアクセス権を持っている必要があります。
- Citrix ADC アプライアンスとクライアントマシンは、共通のネットワークタイムサーバーと同期する必要があります。

GUI を使用したネイティブ **OTP** の設定

ネイティブ OTP 登録は、単要素認証だけではありません。次のセクションでは、シングルファクタ認証とセカンドファクタ認証の設定について説明します。

第 **1** ファクタのログインスキーマの作成

1. セキュリティ **AAA >** アプリケーショントラフィック **>** ログインスキーマに移動します。
2. [プロファイル] に移動し、[追加] をクリックします。
3. [認証ログインスキーマの作成] ページで、[名前] フィールドに *lschema_single_auth_manage_otp* と入力し、**noschema** の横にある [編集] をクリックします。
4. [ログインスキーマ] フォルダをクリックします。
5. 下にスクロールして **SingleAuth.xml** を選択し、[選択] をクリックします。
6. [**Create**] をクリックします。

7. [ポリシー] をクリックし、[追加] をクリックします。
8. **[**認証ログインスキーマポリシーの作成] 画面で、次の値を入力します。

   名前**:** lpol_single_auth_manage_otp_by_url

   プロファイル**:** リストから lschema_single_auth_manage_otp を選択します。

   ルール: HTTP.REQ.COOKIE.VALUE(“NSC_TASS”).EQ(“manageotp“)

認証、承認、および監査仮想サーバーの構成

1. [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**認証仮想サーバ] に移動します。既存の仮想サーバーを編集する場合にクリックします。詳細については、「認証仮想サーバー」を参照してください。
2. 右側のペインの [詳細設定**]** の [ログインスキーマ] の横にある [**+**] アイコンをクリックします。
3. [ログインスキーマなし] を選択します。
4. 矢印をクリックし、**lpol_single_auth_manage_otp_by_url** ポリシーを選択し、**[**選択] をクリックして、[バインド] をクリックします。
5. 上にスクロールし、[高度な認証ポリシー**]** の下の **[**認証ポリシー**]** を **1** つ選択します。
6. **nFactor** ポリシーを右クリックし、[バインディングの編集**]** を選択します。すでに設定されている nFactor ポリシーを右クリックするか、nFactor を参照して作成して [バインドの編集] を選択します。
7. [次のファクターを選択**]** の下の矢印をクリックして既存の構成を選択するか、[追加] をクリックして新しいファクターを作成します。
8. [認証 **PolicyLabel** の作成] 画面で次のように入力し、[続行] をクリックします。

   名前: manage_otp_flow_label

   ログインスキーマ: Lschema_Int

9. [認証 **PolicyLabel**] 画面で、[追加] をクリックしてポリシーを作成します。

   ```
   Create a policy for a normal LDAP server.
   ```

10. [認証ポリシーの作成] 画面で、次のように入力します。

    名前:auth_pol_ldap_native_otp

11. [アクションタイプ] リストを使用して、[アクションタイプ] を [**LDAP**] として選択します。
12. [アクション] フィールドで、[追加] をクリックしてアクションを作成します。

    ```
    Create the first LDAP action with authentication enabled to be used for
     single factor.
    ```

13. [認証 **LDAP** サーバーの作成] ページで、[サーバー **IP**] ラジオボタンを選択し、[認証] の横にあるチェックボックスをオフにし、次の値を入力して [接続のテスト] を選択します。次に、設定例を示します。

名前:ldap_native_otp

IP アドレス: 192.168.xx.xx

ベース **DN**: DC = トレーニング、DC = ラボ

管理者:Administrator@training.lab

パスワード: xxxxx

```
Create a policy for OTP .
```

14. [認証ポリシーの作成] 画面で、次のように入力します。

    名前: auth_pol_ldap_otp_action

15. [アクションタイプ] リストを使用して、[アクションタイプ] を [**LDAP**] として選択します。

16. [アクション] フィールドで、[追加] をクリックしてアクションを作成します。

    ```
    Create the second LDAP action to set OTP authenticator with OTP secret
    configuration and authentication unchecked.
    ```

17. [認証 **LDAP** サーバーの作成] ページで、[サーバー **IP**] ラジオボタンを選択し、[認証] の横にあるチェックボックスをオフにし、次の値を入力して [接続のテスト] を選択します。次に、設定例を示します。

    名前: ldap_otp_action

    IP アドレス: 192.168.xx.xx

    ベース **DN**: DC = トレーニング、DC = ラボ

    管理者:Administrator@training.lab

    パスワード: xxxxx

18. [その他の設定**]** セクションまでスクロールします。ドロップダウンメニューを使用して、次のオプションを選択します。
    サーバーログオン名属性として「新規」と入力し、**userprincipalname** と入力します。

19. ドロップダウンメニューを使用して [**SSO** 名属性] を [新規] として選択し、**userprincipalname** と入力します。

20. [**OTP** シークレット] フィールドに「UserParameters」と入力し、[詳細] をクリックします。

21. 次の属性を入力します。

    属性 **1** =
    メール属性 **2** = objectGUID
    属性 **3** = immutableID

22. 「**OK**」をクリックします。

23. [認証ポリシーの作成] ページで、[式] を **true** に設定し、[作成] をクリックします。

24. [認証ポリシーラベルの作成] ページで、[バインド] をクリックし、[完了] をクリックします。
25. [ポリシーのバインド] ページで、[バインド] をクリックします。
26. [認証ポリシー] ページで、[閉じる] をクリックし、[完了] をクリックします。

    ```
    Create OTP for OTP verification.
    ```

27. [認証ポリシーの作成] 画面で、次のように入力します。

    名前:auth_pol_ldap_otp_verify

28. [アクションタイプ] リストを使用して、[アクションタイプ] を [**LDAP**] として選択します。
29. [アクション] フィールドで、[追加] をクリックしてアクションを作成します。

    ```
    Create the third LDAP action to verify OTP.
    ```

30. [認証 **LDAP** サーバーの作成] ページで、[サーバー **IP**] ラジオボタンを選択し、[認証] の横にあるチェックボックスをオフにし、次の値を入力して [接続のテスト] を選択します。次に、設定例を示します。

    名前:ldap_verify_otp

    IP アドレス: 192.168.xx.xx

    ベース **DN**: DC = トレーニング、DC = ラボ

    管理者:Administrator@training.lab

    パスワード: xxxxx

31. [その他の設定] セクションまでスクロールします。ドロップダウンメニューを使用して、次のオプションを選択します。
    サーバーログオン名属性として「新規」と入力し、**userprincipalname** と入力します。
32. ドロップダウンメニューを使用して [**SSO** 名属性] を [新規] として選択し、**userprincipalname** と入力します。
33. [**OTP** シークレット] フィールドに「UserParameters」と入力し、[詳細] をクリックします。
34. 次の属性を入力します。

    属性 **1** =
    メール属性 **2** = objectGUID
    属性 **3** = immutableID

35. 「**OK**」をクリックします。
36. [認証ポリシーの作成] ページで、[式] を **true** に設定し、[作成] をクリックします。
37. [認証ポリシーラベルの作成] ページで、[バインド] をクリックし、[完了] をクリックします。
38. [ポリシーのバインド] ページで、[バインド] をクリックします。
39. [認証ポリシー] ページで、[閉じる] をクリックし、[完了] をクリックします。

通常の LDAP サーバーの高度な認証ポリシーがまだ作成されていない可能性があります。
[アクションタイプ] を LDAP に変更します。
通常の LDAP サーバーを選択します。これは、認証が有効になっているサーバーです。
式に true を入力します。これは、クラシック構文の代わりにデフォルト構文を使用します。
[作成] をクリックします。

> 注:
>
> 認証仮想サーバーは、rfWebUI ポータルテーマにバインドする必要があります。サーバー証明書をサーバーにバインドします。サーバー IP '1.2.3.5' には、後で使用するために対応する FQDN (otpauth.server.com) が必要です。

第 **2** 要素 **OTP** のログインスキーマの作成

1. [セキュリティ**] > [AAA** アプリケーショントラフィック**] > [**仮想サーバ] に移動します。編集する仮想サーバーを選択します。
2. 下にスクロールして、[**1** つのログインスキーマ] を選択します。
3. [バインドを追加**]** をクリックします。
4. [ポリシーバインド] セクションで、[追加] をクリックしてポリシーを追加します。
5. [認証ログインスキーマポリシーの作成] ページで、[名前] を OTP と入力し、[追加] をクリックしてプロファイルを作成します。
6. [認証ログインスキーマの作成] ページで、[名前] を OTP と入力し、**noschema** の横にある鉛筆アイコンをクリックします。
7. **LoginSchema** フォルダをクリックし 、**DualAuthManageOTP.xml** を選択し 、**[**選択] をクリックします。
8. [詳細] をクリックし、下にスクロールします。
9. [パスワードクレデンシャルインデックス] フィールドに、1 と入力します。これにより、nFactor はユーザーのパスワードを Citrix ADC AAA Attribute #1 に保存します。この属性は、後でトラフィックポリシーで使用して、StoreFront にシングルサインオンできます。これを行わないと、Citrix Gateway はパスコードを使用して StoreFront への認証を試行しますが、これは機能しません。
10. [**Create**] をクリックします。
11. [ルール] セクションで、**True** と入力します。[**Create**] をクリックします。
12. [バインド] をクリックします。
13. 認証の 2 つの要素に注目してください。[閉じる] をクリックし、[完了] をクリックします。

シングルサインオンのトラフィックポリシー

1. **Citrix Gateway >** ポリシー **>** トラフィックに移動します。
2. [トラフィックプロファイル] タブで、[追加] をクリックします。
3. OTP のトラフィックプロファイルの名前を入力します。

4. 下にスクロールして、[SSO パスワード式] ボックスに次のように入力し、[作成] をクリックします。ここでは、第 2 要素 OTP に指定されたログインスキーマのパスワード属性を使用します。

   ```
   http.REQ.USER.ATTRIBUTE(1)
   ```

5. [トラフィックポリシー] タブで、[追加] をクリックします。
6. [**Name**] フィールドに、トラフィックポリシーの名前を入力します。
7. [リクエストプロファイル（Request Profile）] フィールドで、作成したトラフィックプロファイルを選択します。
8. [式] ボックスに **True** と入力します。Citrix Gateway 仮想サーバーでフル VPN が許可されている場合は、式を次のように変更します。

   ```
   http.req.method.eq(post)||http.req.method.eq(get)&& false
   ```

9. [**Create**] をクリックします。

OTP を管理するためのコンテンツスイッチングポリシーを構成する

Unified Gateway を使用している場合は、次の構成が必要です。

1. **Traffic Management > Content Switching > Policies** に移動します。コンテンツスイッチングポリシーを選択し、右クリックして [編集] を選択します。
2. 式を編集して次の OR ステートメントを評価し、「**OK**」をクリックします。

   ```
   is_vpn_url\|\|HTTP.REQ.URL.CONTAINS( “manageotp” )
   ```

CLI を使用したネイティブ **OTP** の設定

OTP デバイス管理ページを設定するには、次の情報が必要です。

- 認証仮想サーバーに割り当てられた IP
- 割り当てられた IP に対応する FQDN
- 認証仮想サーバーのサーバー証明書

> 注**:**
>
> ネイティブ OTP は Web ベースのソリューションのみです。

OTP デバイスの登録および管理ページを設定するには

認証仮想サーバーを作成する


```
1 ```
2 add authentication vserver authvs SSL 1.2.3.5 443
3 bind authentication vserver authvs -portaltheme RFWebUI
4 bind ssl vserver authvs -certkeyname otpauthcert
5 <!--NeedCopy--> ```
```

> 注:
>
> 認証仮想サーバーは rfWebUI ポータルテーマにバインドする必要があります。サーバー証明書をサーバーにバインドします。サーバー IP '1.2.3.5' には、後で使用するために対応する FQDN (otpauth.server.com) が必要です。

LDAP ログオンアクションを作成するには

```
1 add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
      -  serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
      ldapBindDnPassword <PASSWO> -ldapLoginName <USER FORMAT>
```

例:

```
1 add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4 -
      serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
      administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
      ldapLoginName userprincipalname
```

LDAP ログオンの認証ポリシーを追加するには

```
1 add authentication Policy auth_pol_ldap_logon -rule true -action
      ldap_logon_action
```

Loginschema を使用して **UI** を表示するには

ログオン時にユーザー名フィールドとパスワードフィールドをユーザーに表示する

```
1 add authentication loginSchema lschema_single_auth_manage_otp -
      authenticationSchema "/nsconfig/loginschema/LoginSchema/
      SingleAuthManageOTP.xml"
```


デバイスの登録と管理ページを表示する

デバイスの登録画面と管理画面を表示するには、URL またはホスト名の 2 つの方法をお勧めします。

- **URL** を使う

  URL に '/manageotp' が含まれている場合

  - add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_url -rule ”http.req.cookie.value(”NSC_TASS”).contains(”manageotp”)”-action lschema_single_auth_manage_otp
  - bind authentication vserver authvs -policy lpol_single_auth_manage_otp_by_url -priority 10 -gotoPriorityExpression END

- ホスト名を使う

  ホスト名が「alt.server.com」の場合

  - add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_host -rule ”http.req.header(”host”).eq(”alt.server.com”)”-action lschema_single_auth_manage_otp
  - bind authentication vserver authvs -policy lpol_single_auth_manage_otp_by_host -priority 20 -gotoPriorityExpression END

CLI を使用してユーザログインページを設定するには

[ユーザーログオン（User Logon）] ページを設定するには、次の情報が必要です。

- 負荷分散仮想サーバーの IP
- 負荷分散仮想サーバーの対応する FQDN
- 負荷分散仮想サーバーのサーバー証明書

```
bind ssl vserver lbvs_https -certkeyname lbvs_server_cert
<!--NeedCopy-->
```

負荷分散におけるバックエンドサービスは、次のように表されます。

```
```
add service iis_backendsso_server_com 1.2.3.210 HTTP 80
bind lb vserver lbvs_https iis_backendsso_server_com
<!--NeedCopy--> ```
```

 
```

OTP パスコード検証アクションを作成するには

```
add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
    -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
    authentication DISABLED -OTPSecret <LDAP ATTRIBUTE>
<!--NeedCopy-->
```

例:

```
add authentication ldapAction ldap_otp_action -serverIP 1.2.3.4 -
    serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName userprincipalname -authentication DISABLED -OTPSecret
    userParameters
<!--NeedCopy-->
```

> 重要:
>
> LDAP ログオンと OTP アクションの違いは、認証を無効にして新しいパラメータOTPSecretを導入する必要があることです。AD 属性値は使用しないでください。

OTP パスコード検証の認証ポリシーを追加するには

```
add authentication Policy auth_pol_otp_validation -rule true -action
    ldap_otp_action
```

LoginSchema を介して **2** 要素認証を提示するには

2 要素認証用の UI を追加します。

```
add authentication loginSchema lscheme_dual_factor -
    authenticationSchema "/nsconfig/loginschema/LoginSchema/DualAuth.xml
    "
add authentication loginSchemaPolicy lpol_dual_factor -rule true -
    action lscheme_dual_factor
```


ポリシーラベルを介してパスコード検証係数を作成するには

次の要素の管理 OTP フローポリシーラベルを作成します（最初の要素は LDAP ログオンです）

```
1 add authentication loginSchema lschema_noschema -authenticationSchema
    noschema
2 add authentication policylabel manage_otp_flow_label -loginSchema
    lschema_noschema
```

OTP ポリシーをポリシー・ラベルにバインドするには

```
1 bind authentication policylabel manage_otp_flow_label -policyName
    auth_pol_otp_validation -priority 10 -gotoPriorityExpression NEXT
```

UI フローをバインドするには

LDAP ログオンに続いて、認証仮想サーバーを使用した OTP 検証をバインドします。

```
1 bind authentication vserver authvs -policy auth_pol_ldap_logon -
    priority 10 -nextFactor  manage_otp_flow_label -
    gotoPriorityExpression NEXT
2 bind authentication vserver authvs -policy lpol_dual_factor -priority
    30 -gotoPriorityExpression END
```

Citrix ADC でデバイスを登録する

1. /manageotp サフィックスを付けて、Citrix ADC FQDN（最初のパブリック向きの IP）に移動します。たとえば、https://otpauth.server.com/manageotp。ユーザーの資格情報を使用してログインします。
2. [**+**] アイコンをクリックしてデバイスを追加します。
3. デバイス名を入力して **Go** を押します。画面にバーコードが表示されます。
4. [セットアップの開始] をクリックし、[バーコードのスキャン] をクリックします。
5. デバイスのカメラを QR コードの上に置きます。オプションでコードを入力できます。

> 注:
>
> 表示された QR コードは 3 分間有効です。

6. スキャンが成功すると、ログインに使用できる 6 桁の時刻依存コードが表示されます。
7. テストするには、QR 画面で [完了] をクリックし、右側の緑色のチェックマークをクリックします。
8. ドロップダウンメニューからデバイスを選択し、Google Authenticator のコード（赤ではなく青である必要があります）を入力し、[**Go**] をクリックします。
9. ページの右上隅にあるドロップダウンメニューを使用して、必ずログアウトしてください。

OTP を使用して **Citrix ADC** にログインします

1. 最初に公開されている URL に移動し、Google Authenticator から OTP を入力してログオンします。
2. Citrix ADC スプラッシュページに認証します。

OTP シークレットデータを暗号化形式で保存

October 7, 2021

Citrix ADC リリース 13.0 ビルド 41.20 以降、OTP シークレットデータはプレーンテキストではなく暗号化された形式で保存できます。

以前は、Citrix ADC アプライアンスは、OTP シークレットをプレーンテキストとして AD に保存していました。OTP シークレットをプレーンテキストで保存すると、悪意のある攻撃者や管理者が他のユーザーの共有シークレットを表示してデータを悪用する可能性があるため、セキュリティ上の脅威が生じます。

暗号化パラメーターは、AD での OTP シークレットの暗号化を有効にします。Citrix ADC バージョン 13.0 ビルド 41.20 に新しいデバイスを登録し、暗号化パラメーターを有効にすると、OTP シークレットはデフォルトで暗号化された形式で保存されます。ただし、暗号化パラメータが無効になっている場合、OTP シークレットはプレーンテキスト形式で保存されます。

13.0 ビルド 41.20 より前に登録されたデバイスの場合は、ベストプラクティスとして以下を実行する必要があります。

1. 13.0 の Citrix ADC アプライアンスを 13.0 ビルド 41.20 にアップグレードします。
2. アプライアンスで暗号化パラメータを有効にします。
3. OTP シークレット移行ツールを使用して、OTP シークレットデータをプレーンテキスト形式から暗号化形式に移行します。

OTP シークレット移行ツールの詳細については、「OTP 暗号化ツール」を参照してください。

> 重要:
>
> 管理者として、以下の条件を満たしていることを確認してください。
>
> - セルフサービスパスワードリセット機能の一部として KBA を使用していない場合は、OTP シークレット

を暗号化するように新しい証明書を構成する必要があります。

- To bind the certificate to VPN global, you can use the following command:

  ```
  bind vpn global -userDataEncryptionKey <certificate name>
  ```

- すでに証明書を使用して KBA を暗号化している場合は、同じ証明書を使用して OTP シークレットを暗号化できます。

CLI を使用して OTP 暗号化データを有効にするには

コマンドプロンプトで入力します。

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

例

```
set aaa otpparameter -encryption ON
```

GUI を使用して OTP 暗号化を構成するには

1. [セキュリティ] > [AAA — アプリケーショントラフィック] に移動し、[認証 ** 設定] セクションの [認証 AAA OTP パラメータの変更 **] をクリックします。
2. [AAA OTP パラメータの設定] ページで、[OTP シークレット暗号化] を選択します。
3. [OK] をクリックします。

OTP 通知を受信するためのエンドユーザデバイスの数の設定

管理者は、エンドユーザが OTP 通知または認証を受信するために登録できるデバイスの数を設定できるようになりました。

CLI を使用して OTP のデバイス数を設定するには

コマンドプロンプトで入力します。

```
set aaa otpparameter [-maxOTPDevices <positive_integer>]
```

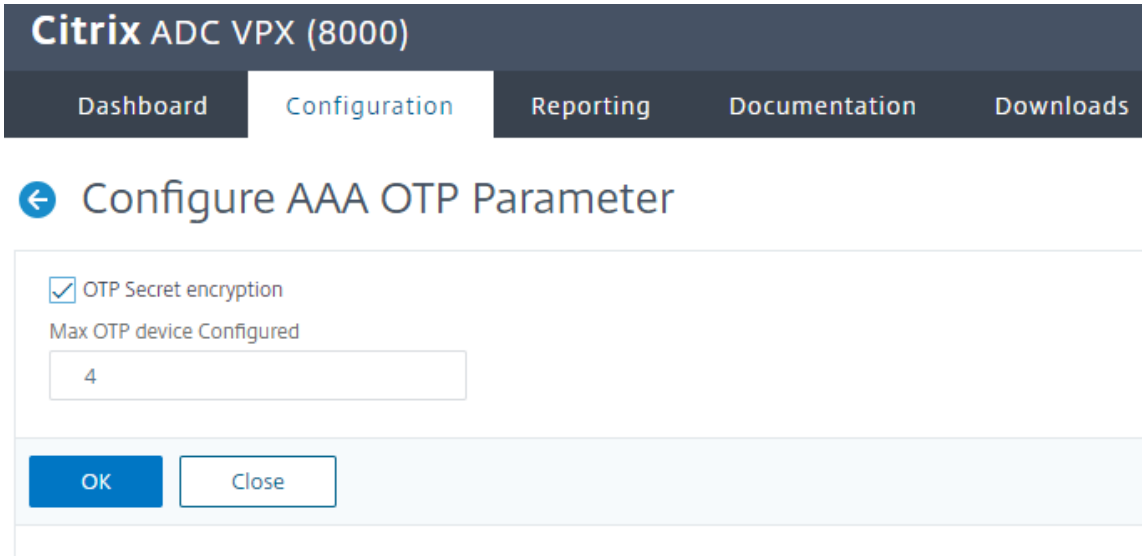
例

```
set aaa otpparameter -maxOTPDevices 4
```

GUI を使用してデバイス数を構成するには

1. [セキュリティ] > [AAA — アプリケーショントラフィック] に移動し、[認証設定] セクションの [認証 AAA OTP パラメータの変更] をクリックします。
2. [AAA OTP パラメータの設定] ページで、[設定済み OTP デバイスの最大数] の値を入力します。

3. [**OK**] をクリックします。



OTP 暗号化ツール

June 1, 2022

Citrix ADC リリース 13.0 ビルド 41.20 以降、OTP シークレットデータは、セキュリティを強化するためにプレーンテキストではなく暗号化された形式で保存されます。OTP シークレットは暗号化された形式で自動的に保存され、手動で操作する必要はありません。

以前は、Citrix ADC アプライアンスは OTP シークレットをプレーンテキストとしてアクティブディレクトリに保存していました。OTP シークレットをプレーンテキスト形式で保存すると、悪意のある攻撃者や管理者が他のユーザーの共有シークレットを表示してデータを悪用する可能性があるため、セキュリティ上の脅威となりました。

OTP 暗号化ツールには次の利点があります。

- 古い形式 (プレーンテキスト) を使用している古いデバイスがあっても、データが失われることはありません。
- 古いバージョンの Citrix Gateway との下位互換性がサポートされているため、既存のデバイスと新しいデバイスの統合とサポートに役立ちます。
- OTP 暗号化ツールは、管理者が一度にすべてのユーザーのすべての OTP 秘密データを移行するのに役立ちます。

> 注:
>
> OTP 暗号化ツールは、KBA 登録または電子メール登録データを暗号化または復号化しません。

OTP 暗号化ツールの使用

OTP 暗号化ツールは、次の目的で使用できます。

- 暗号化。OTP シークレットを暗号化された形式で保存します。このツールは、Citrix ADC に登録されたデバイスの OTP データを抽出し、プレーンテキスト形式の OTP データを暗号化形式に変換します。
- 復号化。OTP シークレットをプレーンテキスト形式に戻します。
- 証明書を更新します。管理者は、いつでも証明書を新しい証明書に更新できます。管理者は、ツールを使用して新しい証明書を入力し、すべてのエントリを新しい証明書データで更新できます。証明書パスは、絶対パスまたは相対パスのいずれかである必要があります。

> 重要
>
> - OTP 暗号化ツールを使用するには、Citrix ADC アプライアンスで暗号化パラメータを有効にする必要があります。
> - ビルド 41.20 より前に Citrix ADC に登録されているデバイスの場合は、次の手順を実行する必要があります。
>   - Upgrade the 13.0 Citrix ADC appliance to 13.0 build 41.20.
>   - Enable the encryption parameter on the appliance.
>   - Use the OTP Secret migration tool to migrate OTP secret data from plain text format to encrypted format.

プレーンテキスト形式の **OTP** シークレットデータ

例:

```
##@devicename=<16 or more bytes>&tag=<64bytes>&,
```

ご覧のとおり、古い形式の開始パターンは常に「#@」で、終了パターンは常に「&」です。”devicename=”および終了パターンの間のすべてのデータは、ユーザー OTP データを構成します。

暗号化された形式の **OTP** シークレットデータ

OTP データの新しい暗号化形式は、次の形式になります。

例:

```
{

      ”otpdata” : {

       “devices” : {

                        “device1” :  “value1” ,
                        “device2” :  “value2” , …
                    }

```



```
11                }
12
13        }
14
15  <!--NeedCopy-->
```

ここで、value1 は、KID + IV + 暗号データの base64 エンコードされた値です。

暗号データは次のように構成されています。

```
1      {
2
3        secret:<16-byte secret>,
4        tag : <64-byte tag value>
5        alg: <algorithm used> (not mandatory, default is sha1, specify
             the algorithm only if it is not default)
6       }
7
8   <!--NeedCopy-->
```

- 「デバイス」では、それぞれの名前に対して価値があります。値は base64encode (kid) .base64encode (IV) .base64encode (暗号データ) です。
- 標準 AES アルゴリズムでは、IV は常に暗号データの最初の 16 バイトまたは 32 バイトとして送信されます。同じモデルに従うことができます。
- キーは同じですが、IV はデバイスごとに異なります。

OTP 暗号化ツールのセットアップ

OTP 暗号化ツールはディレクトリ\var\netscaler\otptoolにあります。Citrix ADC ソースからコードをダウンロードし、必要な AD 資格情報を使用してツールを実行する必要があります。

- OTP 暗号化ツールを使用するための前提条件は次のとおりです。
  - このツールを実行している環境に python 3.5 以降のバージョンをインストールします。
  - pip3 以降のバージョンをインストールします。
- 次のコマンドを実行します:
  - **pip requirements.txt** をインストールします。要件を自動的にインストールします。
  - **python main.py**。OTP 暗号化ツールを起動します。OTP シークレットデータの移行の必要性に応じて、必要な引数を指定する必要があります。
- このツールは、\var\netscaler\otptoolのシェルプロンプトから見つけることができます。
- 必要な AD 認証情報を使用してツールを実行します。

OTP 暗号化ツールインターフェイス

次の図に、OTP 暗号化ツールのインターフェイスの例を示します。インターフェイスには、暗号化/復号化/証明書のアップグレード用に定義する必要のあるすべての引数が含まれています。また、各引数の簡単な説明も取り込まれます。

オペレーション引数

暗号化、復号化、または証明書のアップグレードに OTP 暗号化ツールを使用するには、OPERATION 引数を定義する必要があります。

次の表は、OTP 暗号化ツールおよび対応する OPERATION 引数の値を使用できるシナリオの一部をまとめたものです。

シナリオ	演算引数の値とその他の引数
プレーンテキスト OTP シークレットを同じ属性で暗号化された形式に変換する	OPERATION 引数の値に 0 を入力し、ソース属性とターゲット属性に同じ値を指定します。例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute unixhomedirectory -operation 0 -cert_path aaatm_wild_all.cert`
プレーンテキスト OTP シークレットを別の属性で暗号化された形式に変換する	OPERATION 引数の値に 0 を入力し、ソース属性とターゲット属性に対応する値を指定します。例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 0 -cert_path aaatm_wild_all.cert`

シナリオ	演算引数の値とその他の引数
暗号化されたエントリをプレーンテキストに変換し直す	OPERATION 引数の値に 1 を入力し、ソース属性とターゲット属性に対応する値を指定します。例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 1 -cert_path aaatm_wild_all.cert`
証明書を新しい証明書に更新する	OPERATION 引数の値に 2 を入力し、対応する引数に以前の証明書と新しい証明書の詳細をすべて入力します。例: `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -operation 2 -cert_path aaatm_wild_all.cert -new_cert_path aaatm_wild_all_new.cert`

CERT_PATH 引数

CERT_PATH 引数は、Citrix ADC でデータの暗号化に使用される証明書を含むファイルです。ユーザーは、暗号化、** 復号化、** および更新証明書の 3 つの操作すべてに対してこの引数を指定する必要があります。

CERT_PATH 引数ファイルには、証明書と関連する秘密キーの両方を PEM または CERT 形式で含める必要があります (pfx はサポートされていません)。

たとえば、certificate.cert ファイルと certificate.key ファイルが証明書ファイルとその秘密鍵に対応している場合、UNIX ライクなシステムでは、次のコマンドで cert_path フラグの値として使用できるファイル`certkey.merged`を作成します。

```
$ cat certificate.cert certificate.key > certkey.merged
$
<!--NeedCopy-->
```


> 証明書に関する注意点
>
> - ユーザーは、ユーザーデータの暗号化のために Citrix ADC アプライアンスでグローバルにバインドされているのと同じ証明書を提供する必要があります。
> - 証明書には、Base64 でエンコードされたパブリック証明書とそれに対応する RSA 秘密鍵を同じファイルに含める必要があります。
> - 証明書の形式は PEM または CERT のいずれかである必要があります。証明書は X509 形式に準拠している必要があります。
> - パスワードで保護された証明書形式および*.pfx* ファイルは、このツールでは受け入れられません。ユーザーは、ツールに証明書を提供する前に、PFX 証明書を*.cert* に変換する必要があります。

検索フィルタ引数

SEARCH_FILTER 引数は、AD ドメインまたはユーザーをフィルタリングするために使用されます。この検索フィルターの形式は、Citrix ADC アプライアンスの LDAP アクションコマンドで使用される LDAP 検索フィルター形式と同じです。

Citrix ADC アプライアンスでの暗号化オプションの有効化

プレーンテキスト形式を暗号化するには、Citrix ADC アプライアンスで暗号化オプションを有効にする必要があります。

CLI を使用して OTP 暗号化データを有効にするには、コマンドプロンプトで次のように入力します。

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

例:

```
set aaa otpparameter -encryption ON
```

OTP 暗号化ツールの使用例

OTP 暗号化ツールは、次のユースケースに使用できます。

Citrix ADC アプライアンスバージョン 13.0 ビルド 41.20 に新しいデバイスを登録する

新しいデバイスを Citrix ADC アプライアンスバージョン 13.0 ビルド 41.x に登録し、暗号化オプションが有効になっている場合、OTP データは暗号化された形式で保存されます。手動による介入を避けることができます。

暗号化オプションが有効になっていない場合、OTP データはプレーンテキスト形式で保存されます。

13.0 build 41.20 より前に登録されたデバイスの **OTP** データを移行する

Citrix ADC アプライアンスに 13.0 ビルド 41.20 より前に登録されたデバイスの OTP シークレットデータを暗号化するには、次の手順を実行する必要があります。

- 変換ツールを使用して、OTP データをプレーンテキスト形式から暗号化形式に移行します。
- Citrix ADC アプライアンスで「暗号化」パラメーターを有効にします。
  - CLI を使用して暗号化オプションを有効にするには、次の手順を実行します。
    * `set aaa otpparameter -encryption ON`
  - GUI を使用して暗号化オプションを有効にするには、次の手順を実行します。
    * [セキュリティ**] > [AAA-**アプリケーショントラフィック] に移動し、[****** 認証設定**]** セクションで **[**認証 AAA OTP パラメータの変更 ******]
    * [**AAA OTP** パラメータの設定] ページで、[**OTP** シークレット暗号化] を選択し、[**OK**] をクリックします。
  - 有効な AD 資格情報を使用してログインします。
  - 必要な場合は、追加のデバイスを登録します (オプション)。

暗号化されたデータを古い証明書から新しい証明書に移行する

管理者が証明書を新しい証明書に更新する場合、ツールには新しい証明書データエントリを更新するオプションが用意されています。

CLI を使用して証明書を新しい証明書に更新するには

コマンドプロンプトで入力します。

例:

```
python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local
-search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -
target_attribute userparameters -operation 2 -cert_path aaatm_wild_all.cert
 -new_cert_path aaatm_wild_all_new.cert
```

> 注
>
> - 証明書には秘密鍵と公開鍵の両方が含まれている必要があります。
> - 現在、この機能は OTP に対してのみ提供されています。

アプライアンスを暗号化して **13.0 build 41.20** にアップグレードした後に登録されたデバイスの再暗号化または新しい証明書への移行

管理者は、証明書ですでに暗号化されているデバイスでこのツールを使用し、その証明書を新しい証明書で更新できます。

暗号化されたデータをプレーンテキスト形式に変換し直す

管理者は OTP シークレットを復号化して、元のプレーンテキスト形式に戻すことができます。OTP 暗号化ツールは、暗号化された形式の OTP シークレットをすべてのユーザーにスキャンし、復号化された形式に変換します。

CLI を使用して証明書を新しい証明書に更新するには

コマンドプロンプトで入力します。

例:

```
python3 main.py -Host 192.0.2.1 - Port 636 -username ldapbind_user@aaa
    .local -search_base cn=users,dc=aaa,dc=local -source_attribute
    unixhomedirectory -target_attribute userparameters -operation 1
<!--NeedCopy-->
```

トラブルシューティング

このツールは、次のログファイルを生成します。

- **app.log**。すべての主要な実行ステップと、エラー、警告、および失敗に関する情報を記録します。
- **unmodified_users.txt**。プレーンテキスト形式から暗号化形式にアップグレードされなかったユーザ DN の一覧が含まれます。これらのログは、形式のエラーとして生成されるか、他の理由が原因である可能性があります。

OTP のプッシュ通知

May 9, 2022

Citrix Gateway は、OTP のプッシュ通知をサポートしています。ユーザーは、Citrix Gateway にログインするために、登録済みデバイスで受信した OTP を手動で入力する必要はありません。管理者は、プッシュ通知サービスを使用してログイン通知がユーザーの登録済みデバイスに送信されるように Citrix Gateway を構成できます。ユーザーが通知を受け取ったら、通知の［許可］をタップするだけで、Citrix Gateway にログインできます。ゲートウェイは、ユーザーからの確認応答を受信すると、リクエストのソースを特定し、そのブラウザ接続に応答を送信します。

タイムアウト期間（30 秒）内に通知応答が受信されない場合、ユーザーは Citrix Gateway ログインページにリダイレクトされます。その後、ユーザは OTP を手動で入力するか、または [通知を再送する（**Resend Notification**）] をクリックして、登録されたデバイスで通知を再度受信できます。

管理者は、プッシュ通知用に作成されたログインスキーマを使用して、プッシュ通知認証をデフォルトの認証として設定できます。

> 重要:
>
> プッシュ通知機能は、Citrix ADC Premium エディションライセンスで利用できます。

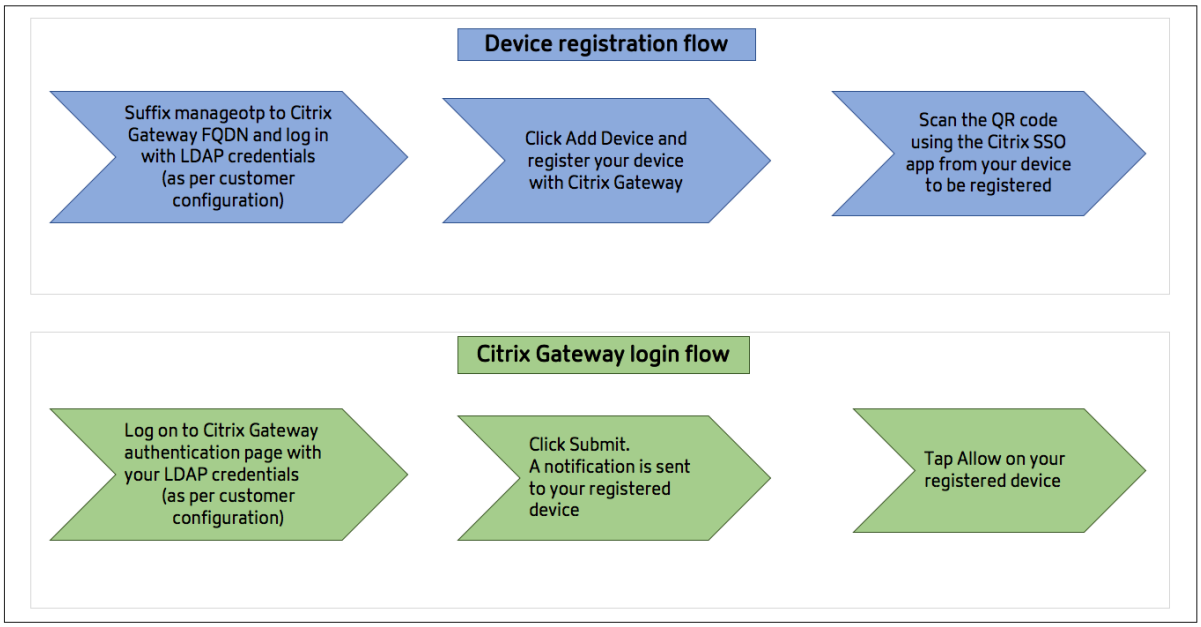
プッシュ通知の利点

- プッシュ通知は、より安全な多要素認証メカニズムを提供します。Citrix Gateway への認証は、ユーザーがログイン試行を承認するまで成功しません。
- プッシュ通知は管理と使用が簡単です。ユーザーは、管理者の支援を必要としない Citrix SSO モバイルアプリをダウンロードしてインストールする必要があります。
- ユーザーはコードをコピーしたり覚えたりする必要はありません。認証を受けるには、デバイスをタップするだけで済みます。
- ユーザーは複数のデバイスを登録できます。

プッシュ通知の動作

プッシュ通知ワークフローは、次の 2 つのカテゴリに分類できます。

- デバイス登録
- エンドユーザログイン



プッシュ通知を使用するための前提条件

- Citrix Cloud のオンボーディングプロセスを完了します。
    1. Citrix Cloud 企業アカウントを作成するか、既存のアカウントに参加します。詳細なプロセスと手順については、「Citrix Cloud へのサインアップ」を参照してください。

2. https://citrix.cloud.comにログインし、顧客を選択します。
3. メニューから [**ID** とアクセス管理] を選択し、[**API** アクセス] タブに移動して、アカウントのクライアントを作成します。
4. ID、シークレット、および顧客 ID をコピーします。ID とシークレットは、Citrix ADC プッシュサービスをそれぞれ「clientID」と「ClientSecret」として構成するために必要です。

> 重要:
>
> - 同じ API 認証情報を複数のデータセンターで使用できます。
> - オンプレミスの Citrix ADC アプライアンスは、サーバーアドレス mfa.cloud.com および trust.citrixworkspacesapi.net を解決でき、アプライアンスからアクセスできる必要があります。これは、ポート 443 を介してこれらのサーバーのファイアウォールまたは IP アドレスブロックがないことを保証するためです。

- iOS デバイス用と Android デバイス用に、それぞれ App Store と Play Store から Citrix SSO モバイルアプリをダウンロードします。プッシュ通知は、iOS 2.3.5 から Android のビルド 1.1.13 からサポートされています。
- Active Directory について、次のことを確認します。
  - 属性の最小長は 256 文字以上にする必要があります。
  - 属性タイプは、ユーザパラメータなどの 'DirectoryString' でなければなりません。これらの属性は文字列値を保持できます。
  - デバイス名が英語以外の文字である場合、属性文字列タイプは Unicode である必要があります。
  - Citrix ADC LDAP 管理者は、選択した AD 属性への書き込みアクセス権を持っている必要があります。
  - Citrix ADC とクライアントマシンは、共通のネットワークタイムサーバーと同期する必要があります。

プッシュ通知の設定

プッシュ通知機能を使用するために完了する必要がある大まかな手順は次のとおりです。

- Citrix Gateway 管理者は、ユーザーを管理および検証するようにインターフェイスを構成する必要があります。
  1. プッシュサービスを設定します。
  2. OTP 管理とエンドユーザーログイン用に Citrix Gateway を構成します。

     ユーザーは、Citrix Gateway にログインするために、デバイスをゲートウェイに登録する必要があります。
  3. デバイスを Citrix Gateway に登録します。
  4. Citrix Gateway にログインします。

プッシュサービスを作成する

1. セキュリティ > **AAA** アプリケーショントラフィック > ポリシー > 認証 > 高度なポリシー > アクション > プッシュサービスに移動し、追加をクリックします。
2. [名前] に、プッシュサービスの名前を入力します。
3. [クライアント **ID**] に、クラウド内の Citrix Push サーバーと通信するための証明書利用者の一意の ID を入力します。
4. [クライアントシークレット] に、クラウド内の Citrix Push サーバーと通信するための証明書利用者の一意のシークレットを入力します。
5. [顧客 **ID**] に、クライアント ID とクライアントシークレットのペアの作成に使用される、クラウド内の顧客 ID またはアカウントの名前を入力します。

> 重要
>
> プッシュサービスには、TLS 1.2 バージョンが必要です。詳細については、TLS 1.2 の設定の詳細を参照してください。

OTP 管理とエンドユーザーログインのための **Citrix Gateway** の構成

OTP 管理とエンドユーザログインについては、次の手順を実行します。

- OTP 管理用のログインスキーマの作成
- 認証、承認、および監査仮想サーバーの構成
- VPN または負荷分散仮想サーバーを構成する
- ポリシーラベルの設定
- エンド・ユーザー・ログイン用のログイン・スキーマの作成

設定の詳細については、ネイティブ OTP サポートを参照してください。

重要: プッシュ通知の場合、管理者は以下を明示的に設定する必要があります。

- プッシュサービスを作成します。
- OTP 管理用のログインスキーマを作成する際には、必要に応じて SingleAuthManageOTP.xml ログインスキーマまたは同等のものを選択します。
- エンドユーザーログイン用のログインスキーマを作成する際には、必要に応じて DualAuthOrPush.xml ログインスキーマまたは同等のものを選択します。

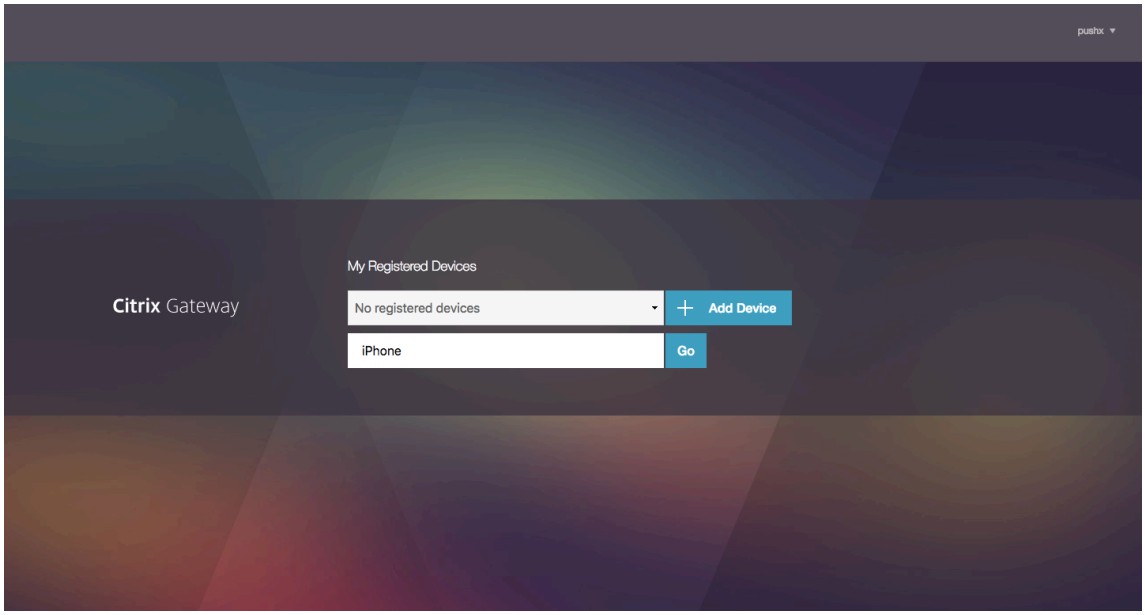
デバイスを **Citrix Gateway** に登録する

プッシュ通知機能を使用するには、デバイスを Citrix Gateway に登録する必要があります。

1. Web ブラウザーで、Citrix Gateway FQDN を参照し、**FQDN** のサフィックスに**/manageotp** を付けます。

   これにより、認証ページが読み込まれます。
   例: https://gateway.company.com/manageotp

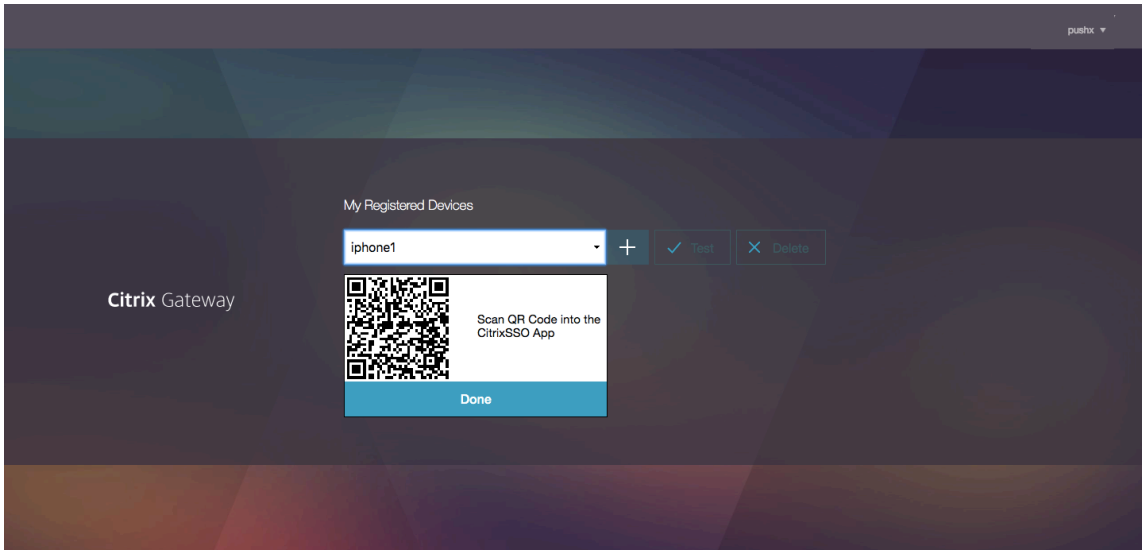
2. 必要に応じて、LDAP 認証情報または適切な 2 要素認証メカニズムを使用してログインします。



3. ［デバイスを追加］をクリックします。

4. デバイスの名前を入力し、［実行］をクリックします。

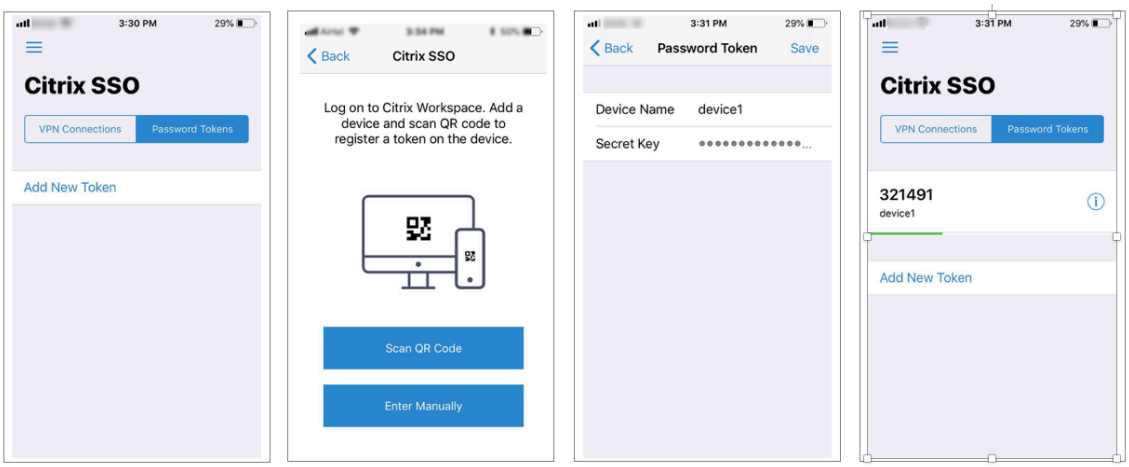
   Citrix Gateway のブラウザーページに QR コードが表示されます。



5. 登録するデバイスから Citrix SSO アプリを使用してこの QR コードをスキャンします。

   Citrix SSO は QR コードを検証し、プッシュ通知のためにゲートウェイに登録します。登録プロセスでエラーがなければ、トークンはパスワードトークンページに正常に追加されます。

   > 重要:
   >
   > QR コードに記載されているシークレットキーを手動で入力すると、ログインに失敗します。



6. 追加/管理するデバイスがない場合は、ページの右上隅にあるリストを使用してログアウトします。

ワンタイムパスワード認証をテストする

1. OTP をテストするには、リストからデバイスをクリックし、[テスト] をクリックします。
2. デバイスで受信した OTP を入力し、[**Go**] をクリックします。

   OTP 検証に成功したことを示すメッセージが表示されます。
3. ページの右上隅にあるリストを使用してログアウトします。

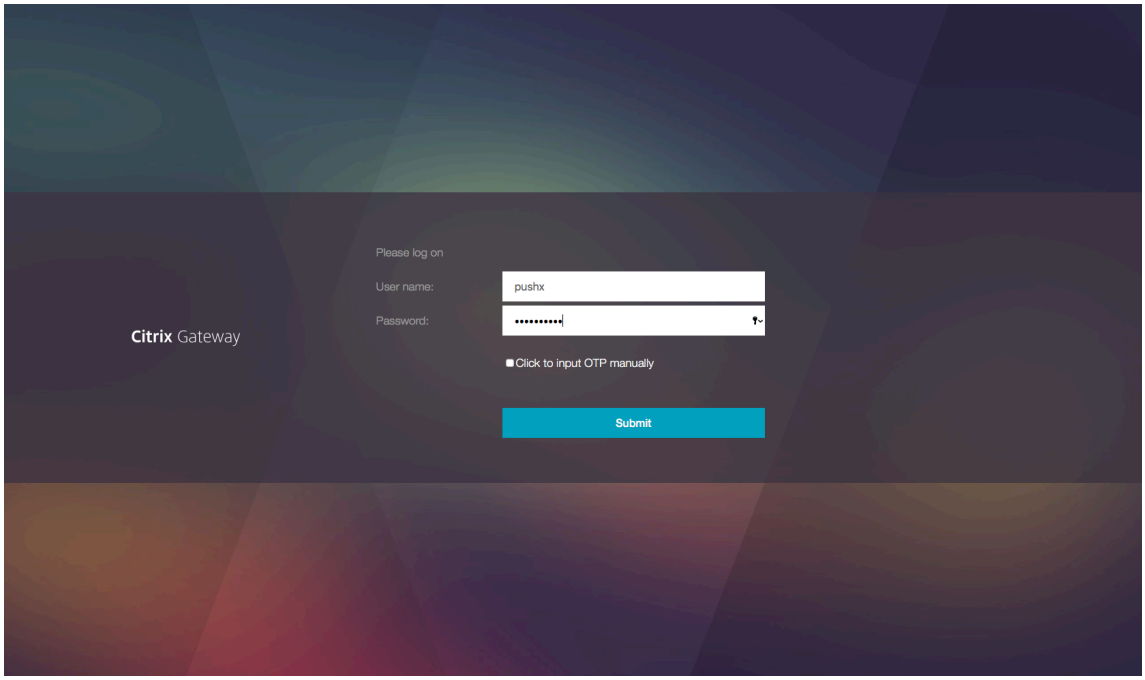
注：OTP 管理ポータルは、認証のテスト、登録済みデバイスの削除、または追加デバイスの登録にいつでも使用できます。

Citrix Gateway にログインする

デバイスを Citrix Gateway に登録すると、ユーザーはプッシュ通知機能を認証に使用できます。

1. Citrix Gateway 認証ページに移動します（例：https://gateway.company.com）。

   ログインスキーマの設定に応じて、LDAP 認証情報のみを入力するように求められます。

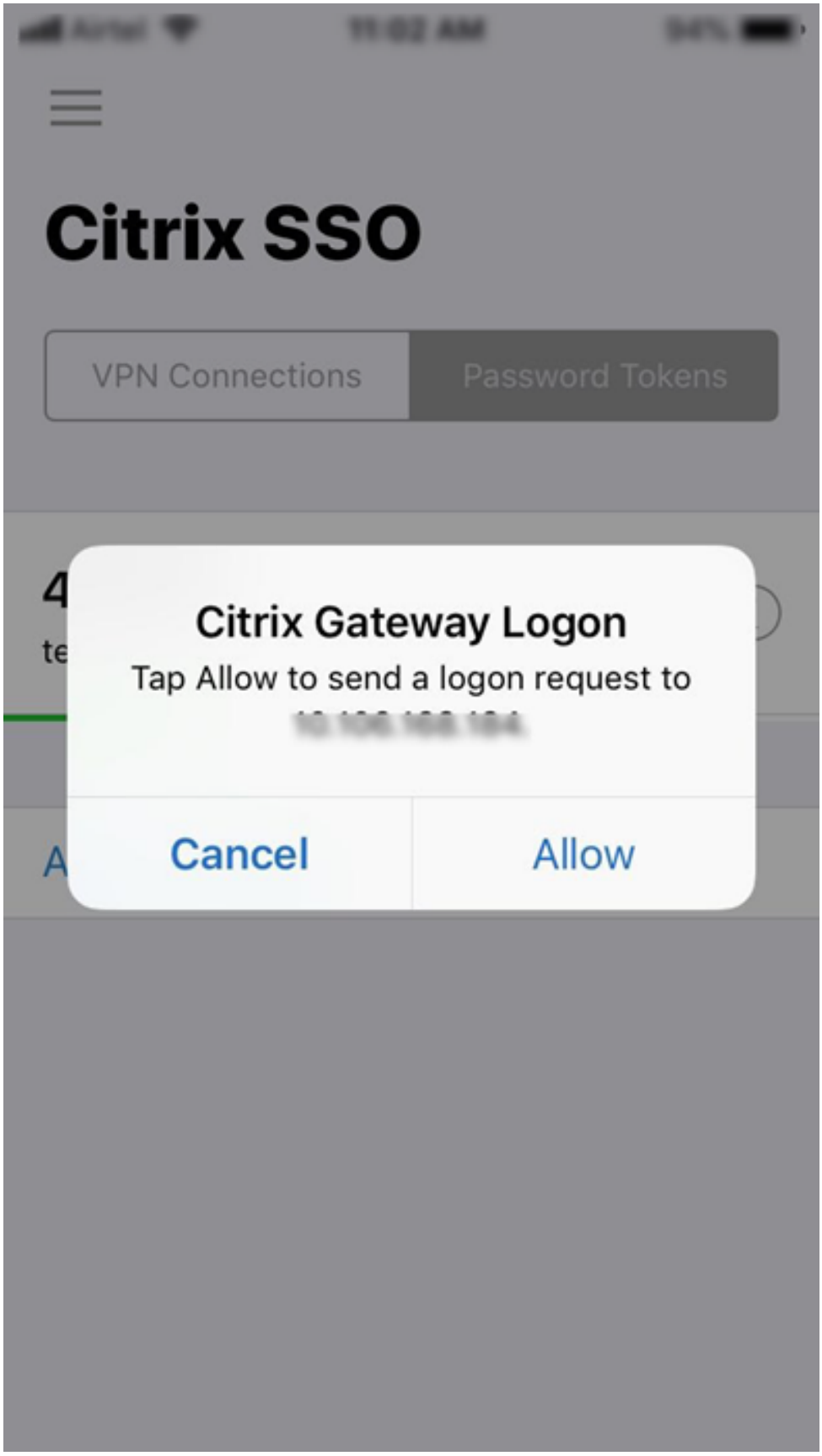


2. LDAP ユーザ名とパスワードを入力し、[送信（**Submit**）] を選択します。

   登録済みのデバイスに通知が送信されます。

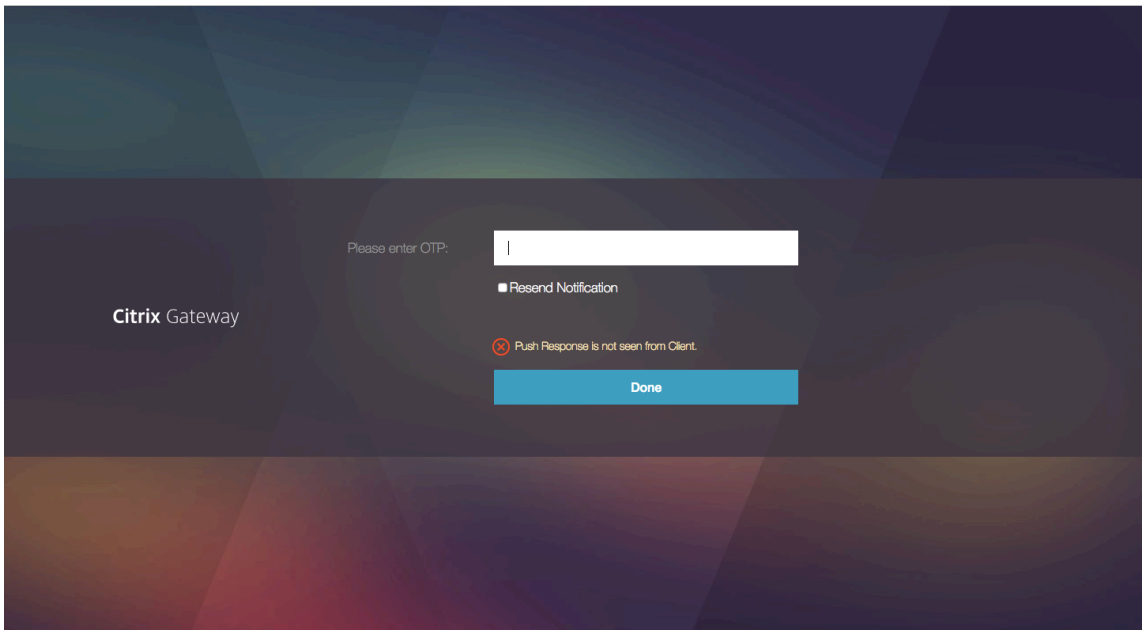
   注意**:** OTP を手動で入力する場合は、「クリックして **OTP** を手動で入力する」を選択し、「**TOTP** 」フィールドに OTP を入力する必要があります。

3. 登録済みのデバイスで Citrix SSO アプリを開き、［許可］をタップします。



注:

- iOS デバイスでは、認証の追加要素として Touch-ID/Face-ID/パスコードの入力を求められます。
- 認証サーバは、設定されたタイムアウト期間が終了するまで、プッシュサーバ通知応答を待ちます。タイムアウト後、Citrix Gateway はログインページを表示します。その後、ユーザは OTP を手動で入力するか、または [通知を再送する（**Resend Notification**）] をクリックして、登録されたデバイスで通知を再度受信できます。選択したオプションに基づいて、ゲートウェイは入力した OTP を検証するか、登録済みデバイスで通知を再送信します。



- ログイン失敗に関する通知は登録済みデバイスに送信されません。

障害状態

- デバイスの登録は、次の場合に失敗することがあります。
  - サーバー証明書は、エンドユーザーデバイスによって信頼されていない可能性があります。
  - OTP への登録に使用された Citrix Gateway に、クライアントからアクセスできません。
- 通知は、次の場合に失敗することがあります。
  - ユーザーデバイスがインターネットに接続されていない
  - ユーザーデバイス上の通知はブロックされます
  - ユーザーがデバイス上の通知を承認しない

このような場合、認証サーバは、設定されたタイムアウト期間が経過するまで待機します。タイムアウト後、Citrix Gateway はログインページを表示し、手動で OTP を入力するか、登録済みのデバイスで通知を再送信するかを選択できます。選択したオプションに基づいて、さらに検証が行われます。

障害ログ

OTP プッシュサービスに到達できない場合の予想されるログを次に示します。

- ユーザーデバイスがインターネットに接続されていないときのプッシュ通知の失敗-プッシュ: プッシュサービスの “client name” へのプッシュリクエストの準備に失敗しました。
- デバイス登録失敗ログ -プッシュ:「client name」のクラウドにプッシュリクエストを送信するためのデバイスが登録されていません。
- ユーザーがプッシュを受け入れない場合-プッシュ:「user name」でクライアントからの応答は表示されません。再試行オプションをチェックします。

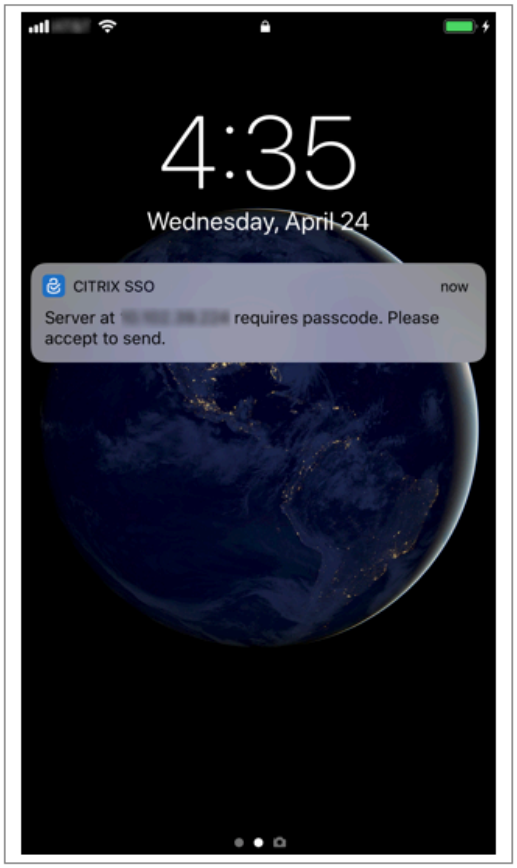
iOS での **Citrix SSO** アプリケーションの動作 **—** 注意すべきポイント

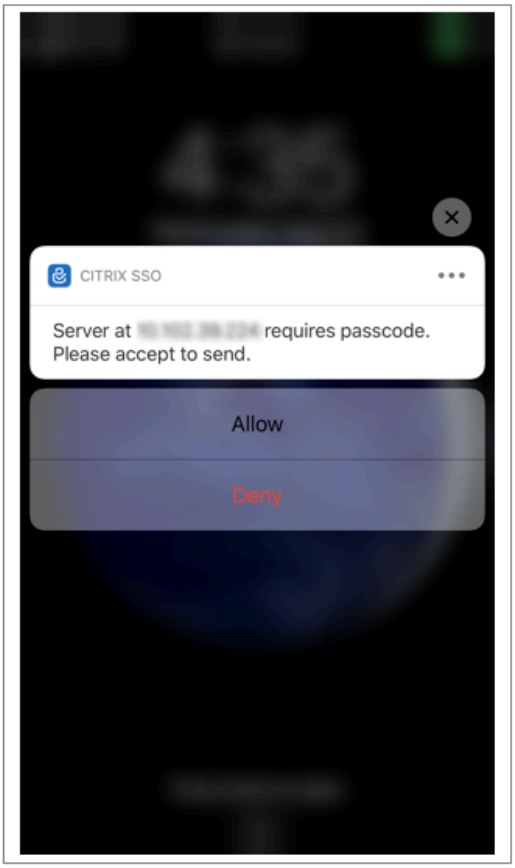
通知ショートカット

Citrix SSO iOS アプリには、ユーザーエクスペリエンスを向上させるための実用的な通知のサポートが含まれています。iOS デバイスで通知を受信し、デバイスがロックされているか、Citrix SSO アプリがフォアグラウンドになっていない場合、ユーザーは通知に組み込まれているショートカットを使用して、ログイン要求を承認または拒否できます。

通知ショートカットにアクセスするには、デバイスのハードウェアに応じて、ユーザーは通知を強制的にタッチ（3D タッチ）するか、長押しする必要があります。ショートカットを許可する］アクションを選択すると、Citrix ADC にログイン要求が送信されます。認証、承認、および監査仮想サーバーで認証ポリシーがどのように構成されているかに応じて、

- ログイン要求は、アプリをフォアグラウンドで起動したり、デバイスのロックを解除したりすることなく、バックグラウンドで送信される可能性があります。
- アプリは追加の要素として Touch-ID/Face-ID/Passcode の入力を求める場合があります。その場合、アプリはフォアグラウンドで起動されます。



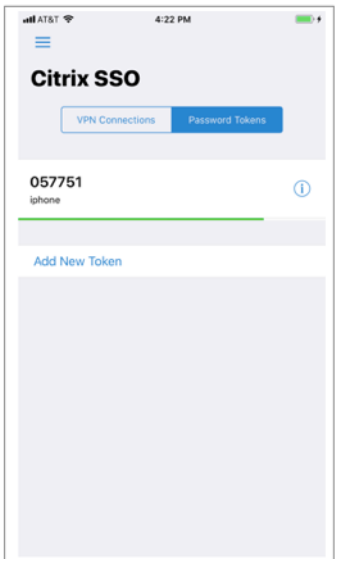


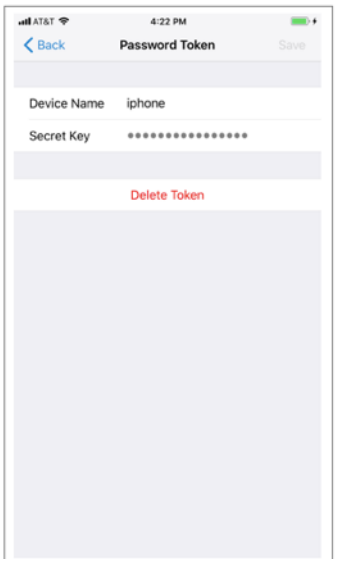
Citrix SSO からパスワードトークンを削除する

1. Citrix SSO アプリでプッシュ用に登録されたパスワードトークンを削除するには、ユーザーは次の手順を実行する必要があります。
2. ゲートウェイの iOS/Android デバイスを登録解除 (削除) します。デバイスから登録を削除するための QR コードが表示されます。
3. Citrix SSO アプリを開き、削除するパスワードトークンの情報ボタンをタップします。
4. [トークンを削除] をタップし、QR コードをスキャンします。

注意:

- QR コードが有効な場合、トークンは Citrix SSO アプリから正常に削除されます。
- デバイスがすでにゲートウェイから削除されている場合、ユーザーは QR コードをスキャンしなくても、[強制削除] をタップしてパスワードトークンを削除できます。強制的に削除すると、デバイスが Citrix Gateway から削除されていない場合、デバイスが引き続き通知を受信する可能性があります。





電子メール **OTP** 認証

March 8, 2022

電子メール OTP は、Citrix ADC 12.1 ビルド 51.x で導入されました。電子メール OTP 方式では、登録された電子メールアドレスに送信されるワンタイムパスワード (OTP) を使用して認証できます。いずれかのサービスで認証を試みると、サーバーは登録されているユーザーのメールアドレスに OTP を送信します。

電子メール OTP 機能を使用するには、まず代替電子メール ID を登録する必要があります。アカウントがロックアウトされた場合や AD パスワードを忘れた場合は、プライマリ電子メール ID にアクセスできないため、OTP をそのメール ID に送信できるように、代わりの電子メール ID の登録が必要です。

AD 属性の一部として代替電子メール ID をすでに指定している場合は、電子メール ID を登録せずに電子メール OTP 検証を使用できます。電子メールアドレスセクションで代替電子メール ID を指定する代わりに、電子メールアクションで同じ属性を参照できます。

前提条件

電子メール OTP 機能を設定する前に、次の前提条件を確認してください。

- Citrix ADC 機能リリース 12.1 ビルド 51.28 以降
- 電子メール OTP 機能は nFactor 認証フローでのみ使用できます
  - 詳細については、「https://support.citrix.com/pages/citrix-adc-authentication-how#nfactor」を参照してください。
  - AAA-TM、Citrix Gateway（ブラウザ、ネイティブプラグイン、Receiver）でサポートされています。

アクティブディレクトリ設定

- サポートされているバージョンは 2016/2012 および 2008 年の Active Directory ドメイン機能レベルです
- Citrix ADC LDAPBind ユーザー名には、ユーザーの AD パスへの書き込みアクセス権が必要です

メールサーバ

- 電子メール OTP ソリューションを機能させるには、SMTP サーバーでログインベースの認証が有効になっていることを確認します。Citrix ADC は、電子メール OTP が機能するために AUTH LOGIN ベースの認証のみをサポートします。
- AUTH LOGIN ベースの認証が有効になっていることを確認するには、SMTP サーバーで次のコマンドを入力します。ログインベースの認証が有効な場合、AUTH LOGIN というテキストが出力に太字で表示されます。

```
root@ns# telnet <IP address of the SMTP server><Port number of the server>
ehlo
root@ns# telnet 10.106.3.
Trying 10.106.
Connected to 10.106.
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22 Nov
2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
For information on how to enable login based authentication, see
https://support.microfocus.com/kb/doc.php?id=7020367
```

制限事項

- この機能は、認証バックエンドが LDAP の場合にのみサポートされます。
- すでに登録されている代替電子メール ID は表示されません。
- KBA 登録ページの代替電子メール ID のみ更新できません。
- 電子メール OTP 認証を認証フローの最初の要素にすることはできません。これは、堅牢な認証を実現するための仕様です。
- 代替電子メール ID と KBA の両方が同じ認証アクションを使用して設定されている場合、属性は両方で同じである必要があります。
- ネイティブプラグインと Receiver では、登録はブラウザ経由でのみサポートされます。

Active Directory 構成

- 電子メール OTP は、Active Directory 属性をユーザーデータストレージとして使用します。
- 代替電子メール ID を登録すると、電子メール ID が Citrix ADC アプライアンスに送信され、アプライアンスは AD ユーザーオブジェクト内の構成済みの KB 属性に電子メール ID を保存します。
- 代替電子メール ID は暗号化され、設定された AD 属性に保存されます。

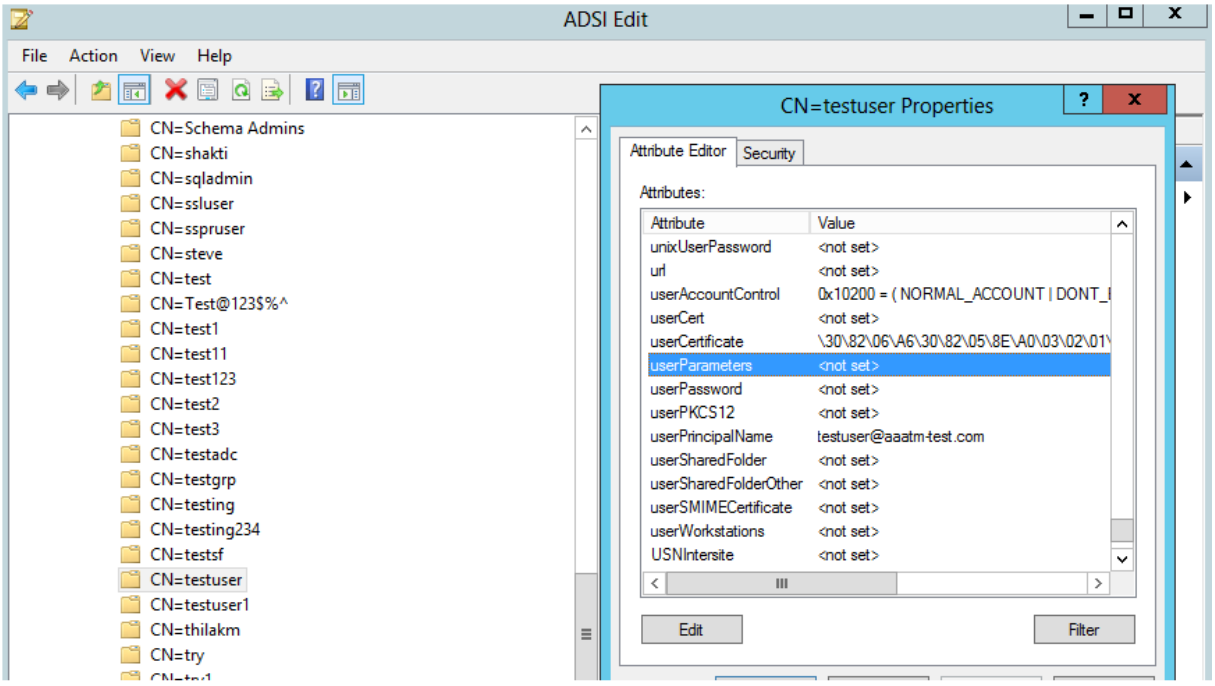
AD 属性を設定する場合は、次の点を考慮してください。

- サポートされる属性名の長さは 128 文字以上にする必要があります。
- 属性タイプは 'directoryString' でなければなりません。
- ネイティブ OTP と電子メール OTP 登録データに同じ AD 属性を使用できます。
- LDAP 管理者には、選択した AD 属性に対する書き込みアクセス権が必要です。

既存の属性の使用

この例で使用されている属性は`Userparameters`です。これは AD ユーザ内の既存の属性なので、AD 自体に変更を加える必要はありません。ただし、その属性が使用されていないことを確認する必要があります。

属性が使用されないようにするには、[**ADSI**] に移動して [user] を選択し、そのユーザーを右クリックし、属性リストまでスクロールダウンします。**UserParameters** の属性値が設定されていないことを確認する必要があります。これは、その属性が現在使用されていないことを示します。



電子メール **OTP** の設定

電子メール OTP ソリューションは、次の 2 つの部分で構成されています。

- メール登録
- メール検証

メール **ID** 登録

KBA 登録スキーマが正常に作成されたら、CLI を使用して次の設定を行います。

1. ポータルテーマと証明書を VPN Global にバインドします。

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> 注:
>
> AD 属性に格納されているユーザーデータ (KB Q&A および登録済みの代替メール ID) を暗号化するには、証明書のバインドを先行する必要があります。

2. LDAP 認証ポリシーを作成します。

```
add authentication ldapAction ldap -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL
add authentication Policy ldap -rule true -action ldap
<!--NeedCopy-->
```

3. 電子メール登録用の LDAP 認証ポリシーを作成します。

```
add authentication ldapAction ldap_email_registration -serverIP
    10.102.2.2 -serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -
    ldapBindDn administrator@aaatm-test.com -ldapBindDnPassword
    freebsd -ldapLoginName samAccountName -secType SSL -KBAttribute
     userParameters -alternateEmailAttr userParameters
add authentication Policy ldap_email_registration -rule true -
    action ldap_email_registration
<!--NeedCopy-->
```

4. E メール登録ログインスキーマとポリシーラベルを作成します。

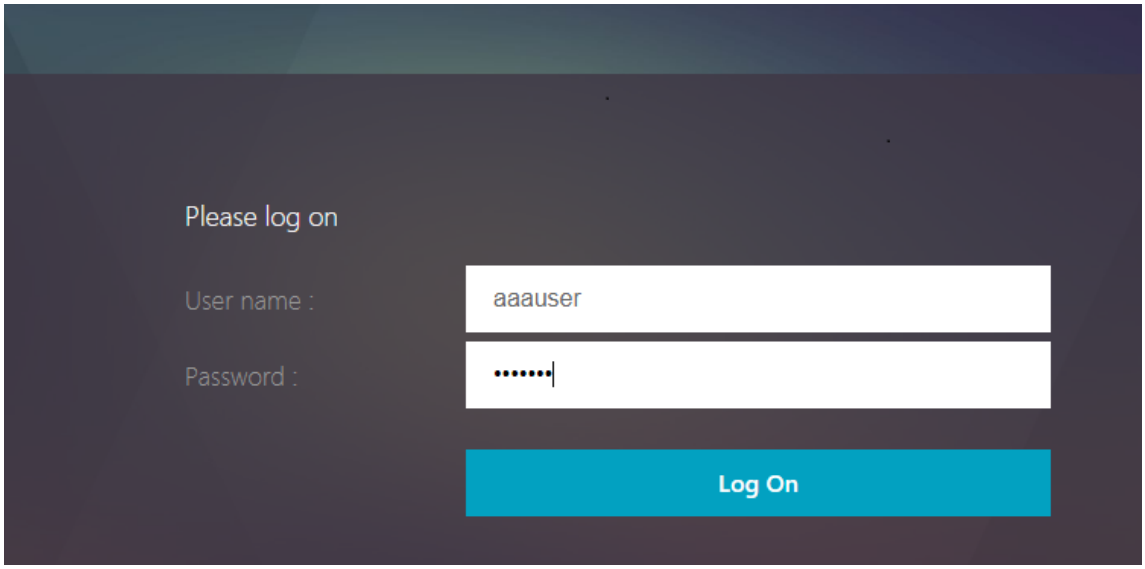

```
add authentication loginSchema onlyEmailRegistration -
    authenticationSchema /nsconfig/loginschema/LoginSchema/
    AltEmailRegister.xml
add authentication policylabel email_Registration_factor -
    loginSchema onlyEmailRegistration
bind authentication policylabel email_Registration_factor -
    policyName ldap_email_registration -priority 1 -
    gotoPriorityExpression NEXT
<!--NeedCopy-->
```

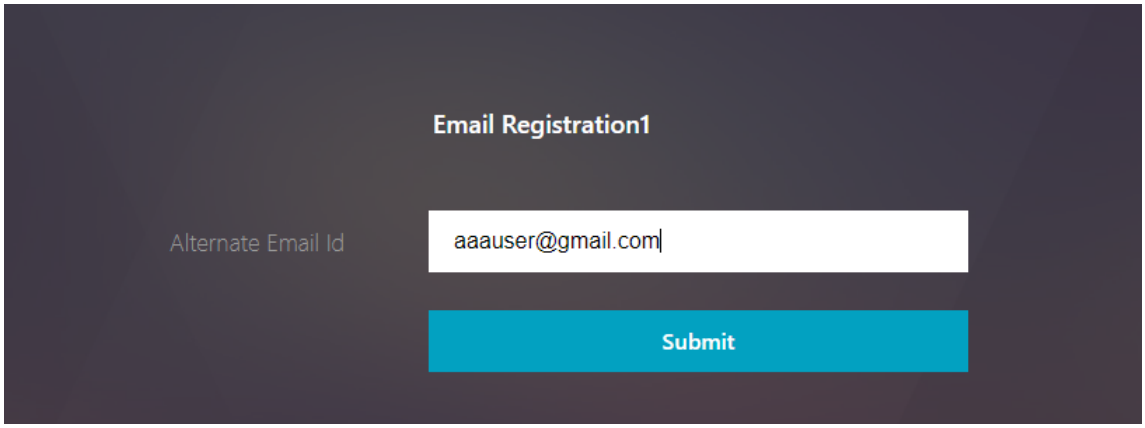
5. 認証ポリシーを認証仮想サーバーにバインドします。

```
bind authentication vserver authvs  - policy ldap -priority 1 -
    nextFactor email_Registration_factor -gotoPriorityExpression
    NEXT
<!--NeedCopy-->
```

6. 前のセクションで説明した手順をすべて設定したら、次の GUI 画面が表示されます。URL（たとえば、https://lb1.server.com/）を使用してアクセスすると、LDAP ログオン資格情報だけを必要とする初期ログインページが表示され、その後に代替の電子メール登録ページが表示されます。

   注**:** ドメインhttps://lb1.server.com/は、ゲートウェイまたは認証仮想サーバのいずれかに属することができます。





> 注:
>
> - KBA 登録と電子メール ID 登録の両方に同じ認証スキーマを使用できます。
> - KBA 登録の設定時に、[電子メール登録] セクションの [代替電子メールの登録] を選択して、代替電子メール ID を登録できます。

メール検証

E メール検証を行うには、次の手順を実行します。

1. ポータルテーマと証明書を VPN Global にバインドする

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> 注:
>
> AD 属性に保存されているユーザーデータ (KB Q&A および登録済みの代替電子メール ID) を復号化するには、証明書のバインドを先行する必要があります。

2. LDAP 認証ポリシーを作成します。電子メール OTP 検証にはユーザーの電子メール ID または代替電子メール ID が必要なため、LDAP は電子メール検証係数の前の要素である必要があります。

```
add authentication ldapAction ldap1 -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -   ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL -KBAttribute
    userParameters -alternateEmailAttr userParameters
add authentication Policy ldap1 -rule true -action ldap1
<!--NeedCopy-->
```


3. E メール認証ポリシーを作成します。

```
1 add authentication emailAction email -userName sqladmin@aaa.com -
      password freebsd-encrypted -encryptmethod ENCMTHD_3 -serverURL
      "smtps://10.2.3.3:25" -content "OTP is $code" -
      defaultAuthenticationGroup emailgrp -emailAddress "aaa.user.
      attribute("alternate_mail")"
2 add authentication Policy email -rule true  - action email
3 <!--NeedCopy-->
```

前述のコマンドでは、メールアドレスは KBA 登録時に指定された代替電子メール ID ユーザーです。

4. 電子メール OTP 検証ポリシーラベルを作成します。

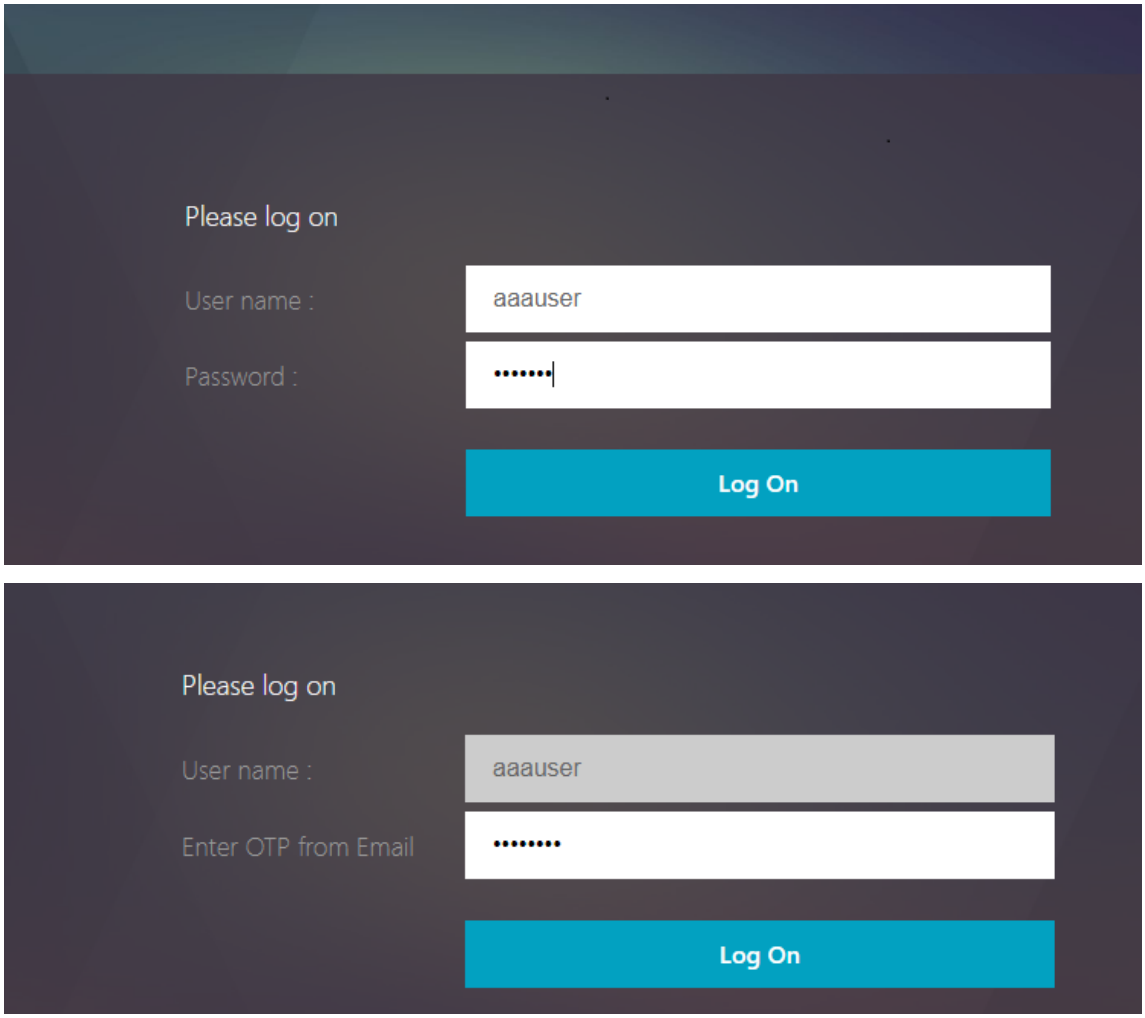
```
1 add authentication policylabel email_Validation_factor
2 bind authentication policylabel email_Validation_factor -
      policyName email -priority 1 -gotoPriorityExpression NEXT
3 <!--NeedCopy-->
```

5. 認証ポリシーを認証仮想サーバーにバインドします。

```
1 bind authentication vserver authvs  -  policy ldap1 -priority 1 -
      nextFactor email_Validation_factor -gotoPriorityExpression NEXT
2 <!--NeedCopy-->
```

6. 前のセクションで説明した手順をすべて設定したら、電子メール OTP 検証用の次の GUI 画面が表示されます。URL（たとえば、https://lb1.server.com/）を使用してにアクセスすると、LDAP ログオン資格情報だけを必要とする初期ログインページが表示され、その後に [電子メール OTP 検証] ページが表示されます。

> **注:**
>
> LDAP ポリシーでは、AD 属性からユーザの電子メール ID を照会できるように`alternateEmailAttr`を設定することが重要です。



トラブルシューティング

ログを分析する前に、以下のようにログレベルを debug に設定することをお勧めします。

```
set syslogparams -loglevel DEBUG
<!--NeedCopy-->
```

登録-成功シナリオ

次のエントリは、ユーザ登録が成功したことを示します。

```
”ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1”
```



```
Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
    0-PPE-1 : default SSLVPN Message 1588 0 :  ”
    ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
    eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
    ==.oKmvOalaOJ3a9z7BcGCSegNPMw==”

<!--NeedCopy-->
```

登録-失敗シナリオ

ユーザーのログインページに、「リクエストを完了できません」というエラーメッセージが表示されます。これは、ユーザデータを暗号化するために VPN グローバルにバインドされる証明書キーがないことを示します。

```
Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:4 6 GMT
    0-PPE-1 : default SSLVPN Message 696 0 :  ”Encrypt UserData: No
    Encryption cert is bound to vpn global”
Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:46 GMT  0-
    PPE-1 : default SSLVPN Message 697 0 :  ”KBA Register: Alternate
    email id Encrypted blob length is ZERO aaauser”
<!--NeedCopy-->
```

メール検証 ー 成功シナリオ

次のエントリは、電子メール OTP 検証が成功したことを示します。

```
”NFactor: Successfully completed email auth, nextfactor is pwd_reset”
<!--NeedCopy-->
```

E メール検証 ー 失敗シナリオ

ユーザーのログインページに、「リクエストを完了できません」というエラーメッセージが表示されます。これは、電子メールサーバーでログインベースの認証が有効になっていないため、同じ認証を有効にする必要があることを示します。

```
” /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp
    [100]: void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID:
    8]SMTP Configuration is Secure..
```



```
2  /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[108]:
       void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID: 8]
       First login succeeded
3  Wed Mar  4 17:16:28 2020
4  /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/naaad.c[697]: main
       0-0: timer 2 firing...
5  /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[127]:
       void ThreadWorker_SendMailJob(SMTPJob*) 0-0: [POCO-ERROR][JobID: 8]
       Poco SMTP Mail Dispatch  Failed. SMTP TYPE:1, SMTPException:
       Exception occurs. SMTP Exception: The mail service does not support
       LOGIN authentication: 250-smtprelay.citrix.com Hello [10.9.154.239]
6  250-SIZE 62914560
7  250-PIPELINING
8  250-DSN
9  250-ENHANCEDSTATUSCODES
10 250-8BITMIME
11 250-BINARYMIME
12 250 CHUNKING
13 <!--NeedCopy-->
```

nFactor 認証用の再キャプチャ設定

October 7, 2021

Citrix Gateway は、新しいファーストクラスのアクション「captchaAction」をサポートし、再キャプチャの構成を簡素化します。reCaptcha はファーストクラスのアクションであるため、独自の要因になる可能性があります。nFactor フローのどこにでも reCaptcha を注入することができます。

以前は、RfWeb UI の変更を含むカスタム WebAuth ポリシーを記述する必要がありました。キャプチャアクションの導入により、JavaScript を変更する必要はありません。

> 重要
>
> reCaptcha がスキーマ内のユーザー名またはパスワードのフィールドと一緒に使用されている場合、reCaptcha が満たされるまで送信ボタンは無効になります。

再キャプチャ設定

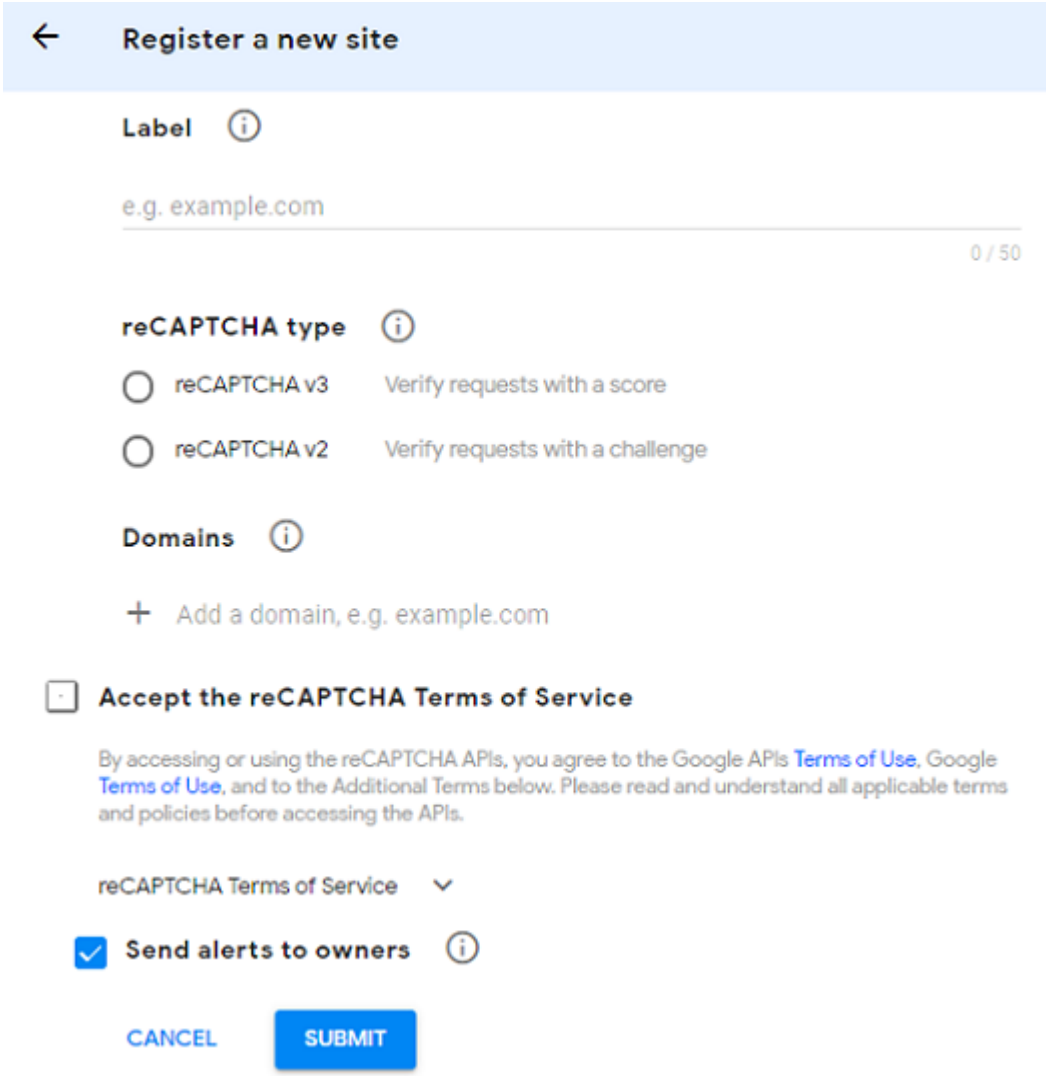
reCaptcha 構成には、2 つの部分が含まれます。

1. 再キャプチャを登録するための Google の構成。
2. ログインフローの一部として再キャプチャを使用する Citrix ADC アプライアンスの構成。

Google で再キャプチャ設定

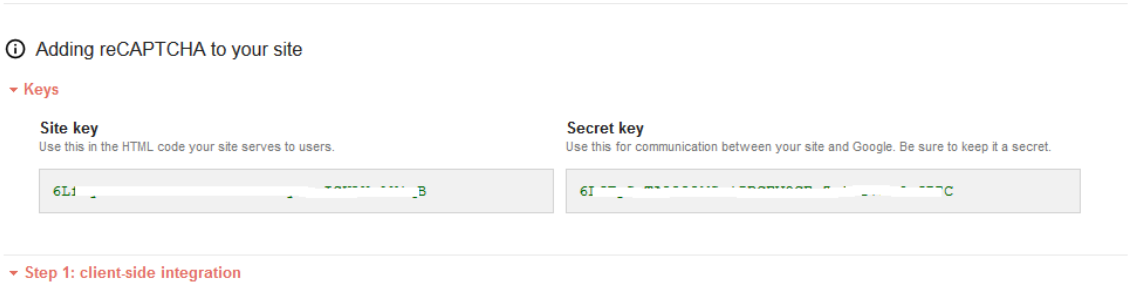
https://www.google.com/recaptcha/adminで reCaptcha のドメインを登録します。

1. このページに移動すると、次の画面が表示されます。



> 注
>
> 再 CAPTCHA v2 のみを使用してください。目に見えない reCAPTCHA はまだベータ版です。

2. ドメインを登録すると、「サイトキー」と「秘密キー」が表示されます。



> 注
>
> セキュリティ上の理由から、「SiteKey」と「SecretKey」はグレー表示になっています。「SecretKey」は安全に保管する必要があります。

Citrix ADC アプライアンスの構成を再キャプチャする

Citrix ADC アプライアンスの再キャプチャ構成は、3 つの部分に分けることができます。

- 再キャプチャ画面を表示する
- Google サーバーに再キャプチャ応答を投稿する
- LDAP 構成は、ユーザーログオンの 2 番目の要素です (オプション)

再キャプチャ画面を表示する

ログインフォームのカスタマイズは、SingleAuthCaptcha.xml Loginschema を介して行われます。このカスタマイズは、認証仮想サーバーで指定され、ログインフォームをレンダリングするために UI に送信されます。組み込みの Loginschema である SingleAuthCaptcha.xml は、Citrix ADC アプライアンス上の/nsconfig/loginschema/Loginschema ディレクトリにあります。

> 重要
>
> - ユースケースと異なるスキーマに基づいて、既存のスキーマを変更できます。たとえば、reCaptcha 係数（ユーザー名またはパスワードなし）または reCaptcha との二重認証のみが必要な場合。
> - カスタム変更を実行した場合、またはファイルの名前を変更した場合は、すべての loginSchema を/nsconfig/loginschema ディレクトリから親ディレクトリ/nsconfig/loginschema にコピーすることをお勧めします。

CLI を使用して再キャプチャの表示を設定するには

- `add authentication loginSchema singleauthcaptcha -authenticationSchema /nsconfig/loginschema/SingleAuthCaptcha.xml`
- `add authentication loginSchemaPolicy singleauthcaptcha -rule true -action singleauthcaptcha`
- `add authentication vserver auth SSL <IP> <Port>`

- add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-key-file>
- bind ssl vserver auth -certkey vserver-cert
- bind authentication vserver auth -policy singleauthcaptcha -priority 5 -gotoPriorityExpression END

Google サーバーに再キャプチャ応答を投稿する

あなたは、ユーザーに表示されなければならない reCaptcha を設定した後、管理者は、ブラウザからの reCaptcha 応答を確認するために、Google サーバーに構成を追加ポスト。

ブラウザから再キャプチャ応答を確認するには

- add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>
- add authentication policy myrecaptcha -rule **true** -action myrecaptcha
- bind authentication vserver auth -policy myrecaptcha -priority 1

AD 認証が必要な場合は、次のコマンドが必要です。それ以外の場合は、この手順を無視できます。

- add authentication ldapAction ldap-**new** -serverIP x.x.x.x -serverPort 636 -ldapBase ”cn=users,dc=aaatm,dc=com”-ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup
- add authenticationpolicy ldap-**new** -rule **true** -action ldap-**new**

LDAP 構成は、ユーザーログオンの **2** 番目の要素です **(**オプション**)**

LDAP 認証は reCaptcha の後に行われ、2 番目の要素に追加します。

- add authentication policylabel second-factor
- bind authentication policylabel second-factor -policy ldap-**new** -priority 10
- bind authentication vserver auth -policy myrecaptcha -priority 1 -nextFactor second-factor

管理者は、負荷分散仮想サーバーと Citrix Gateway アプライアンスのどちらを使用してアクセスするかに応じて、適切な仮想サーバーを追加する必要があります。ロードバランシング仮想サーバが必要な場合は、管理者が次のコマンドを設定する必要があります。

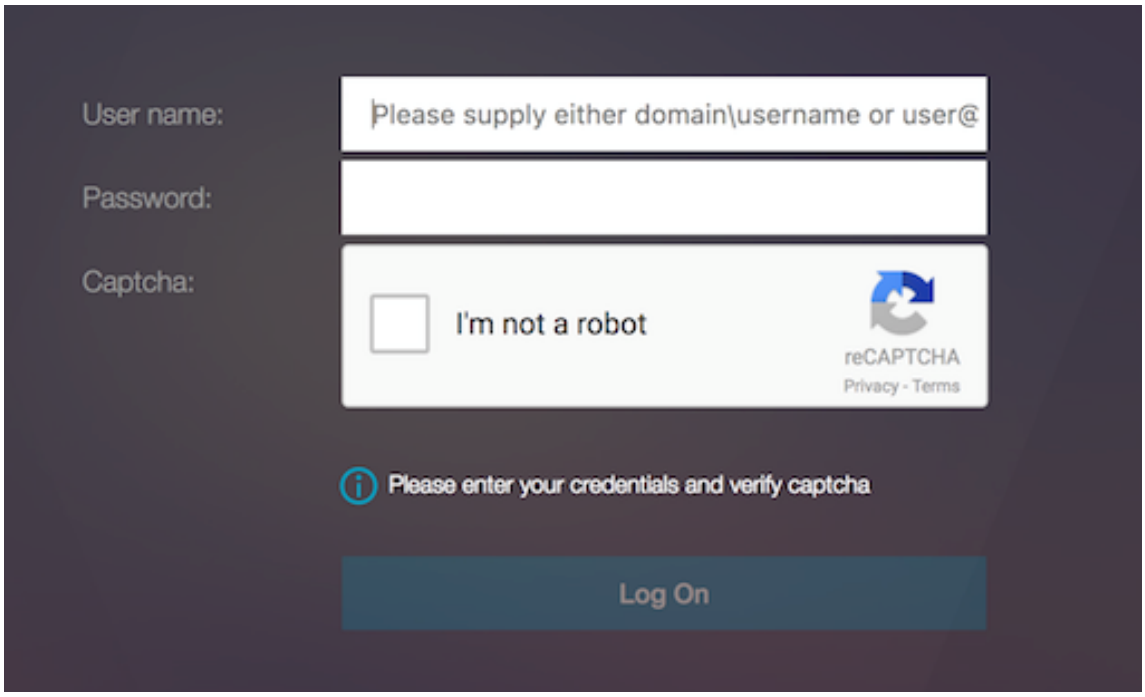
- add lb vserver lbtest HTTP <IP> <Port> -authentication ON -authenticationHost nssp.aaatm.com

nssp.aaatm.com — 認証仮想サーバーに解決します。

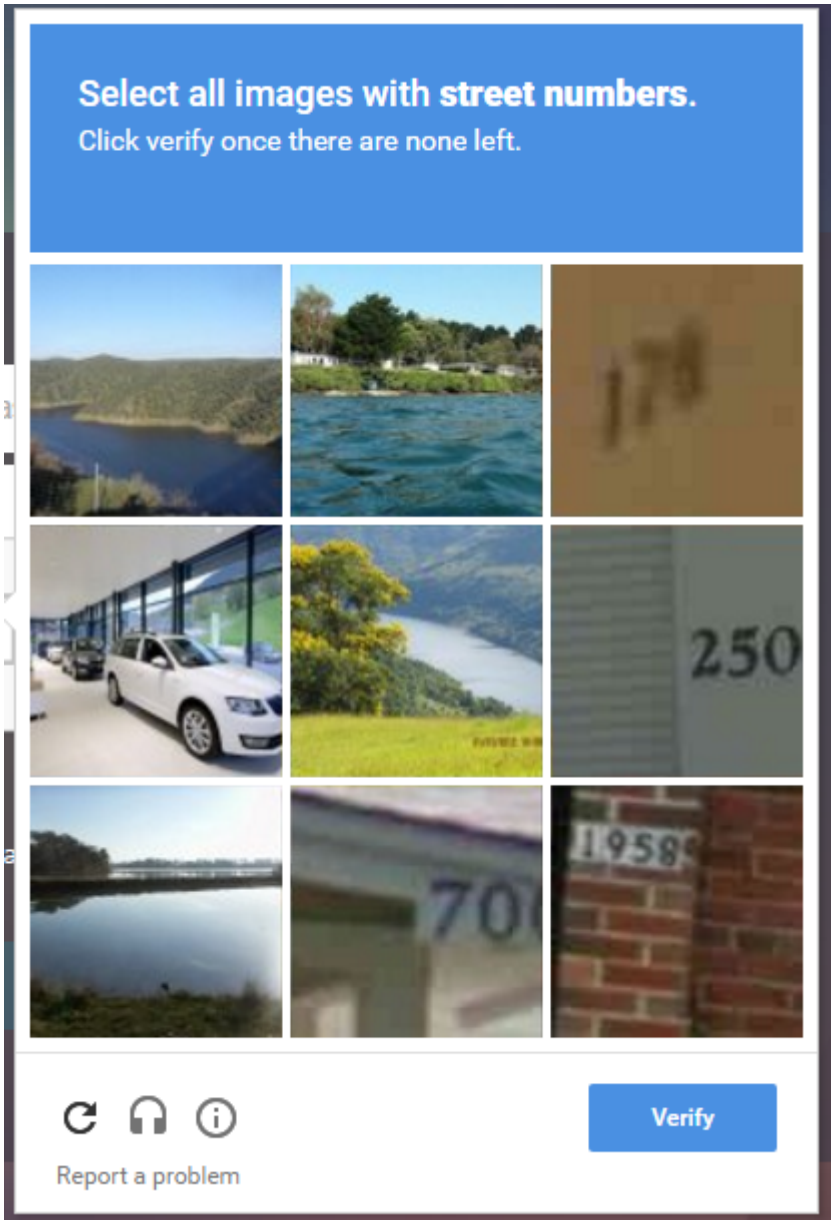
再キャプチャのユーザー検証

前のセクションで説明したすべての手順を設定したら、以下に示す UI のスクリーンショットを確認する必要があります。

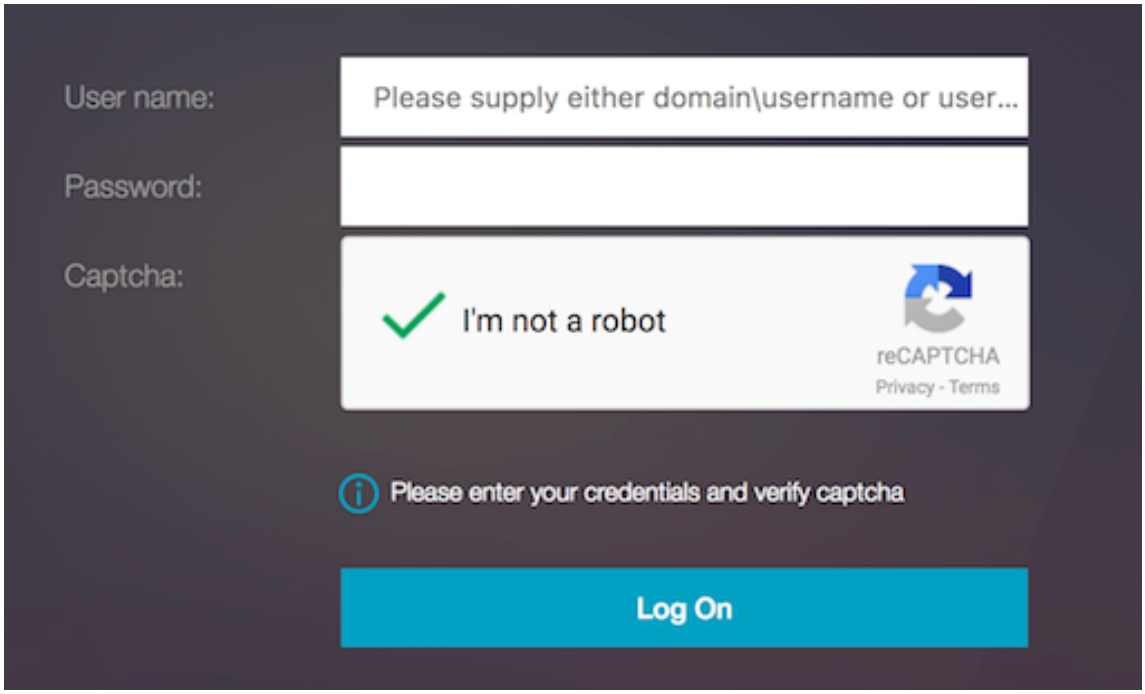
1. 認証仮想サーバーがログインページを読み込むと、ログオン画面が表示されます。ログオンは、再キャプチャが完了するまで無効になります。



2. [私はロボットではありません] オプションを選択します。再キャプチャウィジェットが表示されます。



3. 完了ページが表示される前に、一連の reCaptcha イメージ間を移動します。
4. AD 資格情報を入力し、[ロボットではありません] チェックボックスをオンにして、[ログオン] をクリックします。認証が成功すると、目的のリソースにリダイレクトされます。



メモ

- reCaptcha が AD 認証で使用されている場合、reCaptcha が完了するまで、資格情報の送信ボタンは無効になります。
- reCaptcha は、独自の要因で発生します。したがって、AD のような後続の検証は、reCaptcha の 'nextfactor' で発生する必要があります。

一般的に使用されるプロトコルの認証、承認、および監査の構成

January 31, 2022

Citrix ADC アプライアンスを認証、承認、監査用に構成するには、Citrix ADC アプライアンスおよびクライアントのブラウザで特定の設定が必要です。設定は、認証、認可、および監査に使用されるプロトコルによって異なります。

Kerberos 認証用の Citrix ADC アプライアンスの構成の詳細については、「Kerberos/NTLM による認証、承認、および監査の処理」を参照してください。

Kerberos /NTLM による認証、承認、監査の処理

October 7, 2021

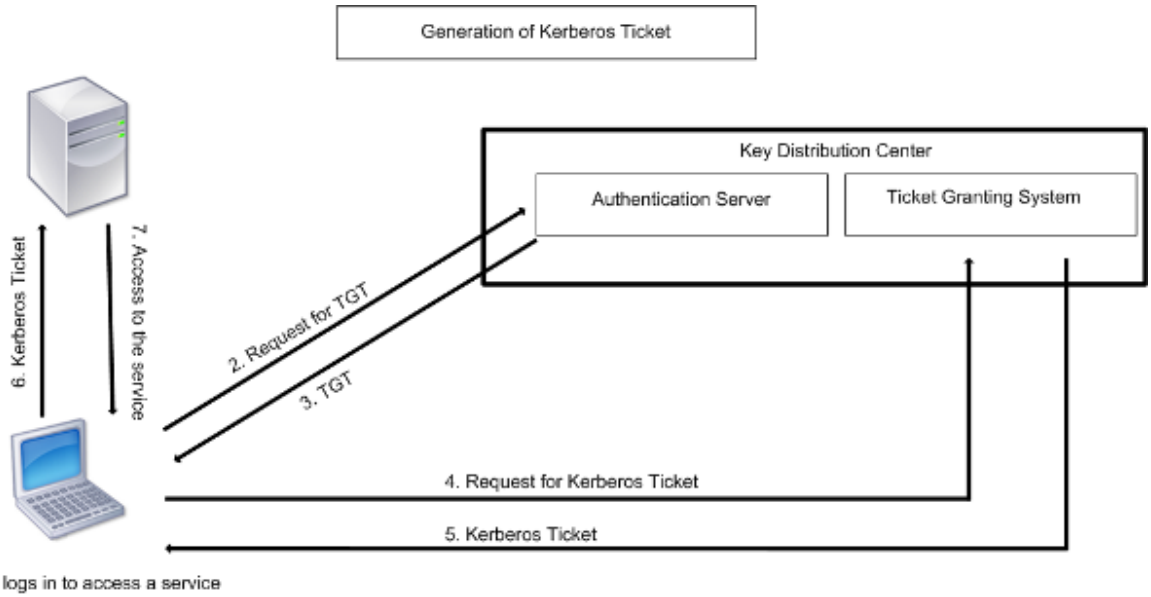
コンピューターネットワーク認証プロトコルである Kerberos は、インターネットを介した安全な通信を提供します。主にクライアント/サーバーアプリケーション用に設計されており、クライアントとサーバーがそれぞれ互いの信

頼性を保証するための相互認証を提供します。Kerberos は、キー配布センター (KDC) と呼ばれる信頼できるサードパーティを使用します。KDC は、ユーザーを認証する認証サーバー (AS) と、チケット交付サーバー (TGS) で構成されます。

ネットワーク上の各エンティティ (クライアントまたはサーバー) には、自身と KDC のみが認識する秘密鍵があります。このキーの知識は、エンティティの真正性を意味します。ネットワーク上の 2 つのエンティティ間の通信のために、KDC は Kerberos チケットまたはサービスチケットと呼ばれるセッションキーを生成します。クライアントは、特定のサーバのクレデンシャルを AS に要求します。クライアントは、チケット交付チケット（TGT）と呼ばれるチケットを受け取ります。次に、クライアントは、AS から受信した TGT を使用して TGS に接続し、その ID を証明し、サービスを要求します。クライアントがサービスに適格である場合、TGS はクライアントに Kerberos チケットを発行します。次に、クライアントは Kerberos チケットを使用して、サービスをホストしているサーバー (サービスサーバーと呼ばれる) と通信し、サービスの受信が許可されていることを証明します。Kerberos チケットには、設定可能な有効期間があります。クライアントは AS で自身を認証するのは 1 回だけです。物理サーバに複数回接続すると、AS チケットが再利用されます。

次の図は、Kerberos プロトコルの基本的な機能を示しています。

図 1: **Kerberos** の機能



Kerberos 認証には、次の利点があります。

- 認証の高速化。物理サーバーがクライアントから Kerberos チケットを取得すると、サーバーはクライアントを直接認証するのに十分な情報を持ちます。クライアント認証のためにドメイン Controller に接続する必要がないため、認証プロセスが高速になります。
- 相互認証。KDC がクライアントに Kerberos チケットを発行し、クライアントがチケットを使用してサービスにアクセスする場合、認証されたサーバーのみが Kerberos チケットを復号化できます。Citrix ADC アプライアンス上の仮想サーバーが Kerberos チケットを復号化できる場合は、仮想サーバーとクライアントの両方が認証されていると結論付けることができます。したがって、サーバーの認証は、クライアントの認証とともに行われます。

- Windows と Kerberos をサポートする他のオペレーティング・システムの間のシングル・サインオン。

Kerberos 認証には、次のような欠点があります。

- Kerberos には厳しい時間要件があります。認証が失敗しないように、関連するホストのクロックを Kerberos サーバのクロックと同期させる必要があります。ネットワークタイムプロトコルデーモンを使用してホストクロックの同期を維持することで、この欠点を緩和できます。Kerberos チケットには可用性期間があり、これを設定できます。
- Kerberos では、中央サーバを継続的に使用できるようにする必要があります。Kerberos サーバーがダウンすると、誰もログオンできません。複数の Kerberos サーバーとフォールバック認証メカニズムを使用することで、このリスクを軽減できます。
- すべての認証は一元化された KDC によって制御されるため、ローカルワークステーションのユーザーのパスワードが盗まれるなど、このインフラストラクチャに侵入すると、攻撃者が任意のユーザーを偽装する可能性があります。信頼するデスクトップマシンまたはラップトップのみを使用するか、ハードウェアトークンを使用して事前認証を適用することで、このリスクをある程度軽減できます。

Kerberos 認証を使用するには、Citrix ADC アプライアンスおよび各クライアントで認証を構成する必要があります。

認証、承認、監査での **Kerberos** 認証の最適化

Citrix ADC アプライアンスは、Kerberos 認証中にシステムパフォーマンスを最適化し、改善するようになりました。認証、承認、および監査デーモンは、同じユーザーに対する未処理の Kerberos 要求を記憶し、キー配布センター (KDC) への負荷を回避します。これにより、要求の重複を回避できます。

Citrix ADC がクライアント認証に **Kerberos** を実装する方法

October 7, 2021

> 重要
>
> Kerberos/NTLM 認証は、NetScaler 9.3 nCore リリース以降でのみサポートされており、トラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。

Citrix ADC は、Kerberos 認証に関連するコンポーネントを次のように処理します。

キー配布センター **(KDC)**

Windows 2000 サーバーまたはそれ以降のバージョンでは、ドメインコントローラーと KDC は Windows サーバーの一部です。Windows Server が起動し、実行している場合は、ドメインコントローラと KDC が構成されていることを示します。KDC は、Active Directory サーバーでもあります。

> 注
>
> すべての Kerberos の相互作用は、Windows Kerberos ドメインコントローラーで検証されます。

認証サービスとプロトコルのネゴシエーション

Citrix ADC アプライアンスは、認証、承認、監査トラフィック管理認証仮想サーバーでの Kerberos 認証をサポートします。Kerberos 認証が失敗した場合、Citrix ADC は NTLM 認証を使用します。

既定では、Windows 2000 Server 以降のバージョンの Windows Server では、認証、承認、および監査に Kerberos が使用されます。認証タイプとして NEGOTIATE を使用して認証ポリシーを作成すると、Citrix ADC は認証、承認、および監査に Kerberos プロトコルを使用しようとします。クライアントのブラウザが Kerberos チケットを受信できない場合、Citrix ADC は NTLM 認証を使用します。このプロセスは、ネゴシエーションと呼ばれます。

クライアントは、次のいずれかに該当する場合、Kerberos チケットの受信に失敗することがあります。

- Kerberos はクライアントでサポートされていません。
- Kerberos がクライアントで有効になっていません。
- クライアントが KDC 以外のドメインに存在する。
- KDC 上のアクセスディレクトリは、クライアントからアクセスできません。

Kerberos/NTLM 認証の場合、Citrix ADC は、Citrix ADC アプライアンスにローカルに存在するデータを使用しません。

承認

トラフィック管理仮想サーバは、負荷分散仮想サーバまたはコンテンツスイッチング仮想サーバです。

監査

Citrix ADC アプライアンスは、次の監査ログによる Kerberos 認証の監査をサポートします。

- トラフィック管理エンドユーザアクティビティの完全な監査証跡
- SYSLOG と高パフォーマンス TCP ロギング
- システム管理者の完全な監査証跡
- すべてのシステムイベント
- スクリプト可能なログ形式

サポートされる環境

Kerberos 認証では、Citrix ADC 上の特定の環境は必要ありません。クライアント（ブラウザ）が Kerberos 認証をサポートする必要があります。

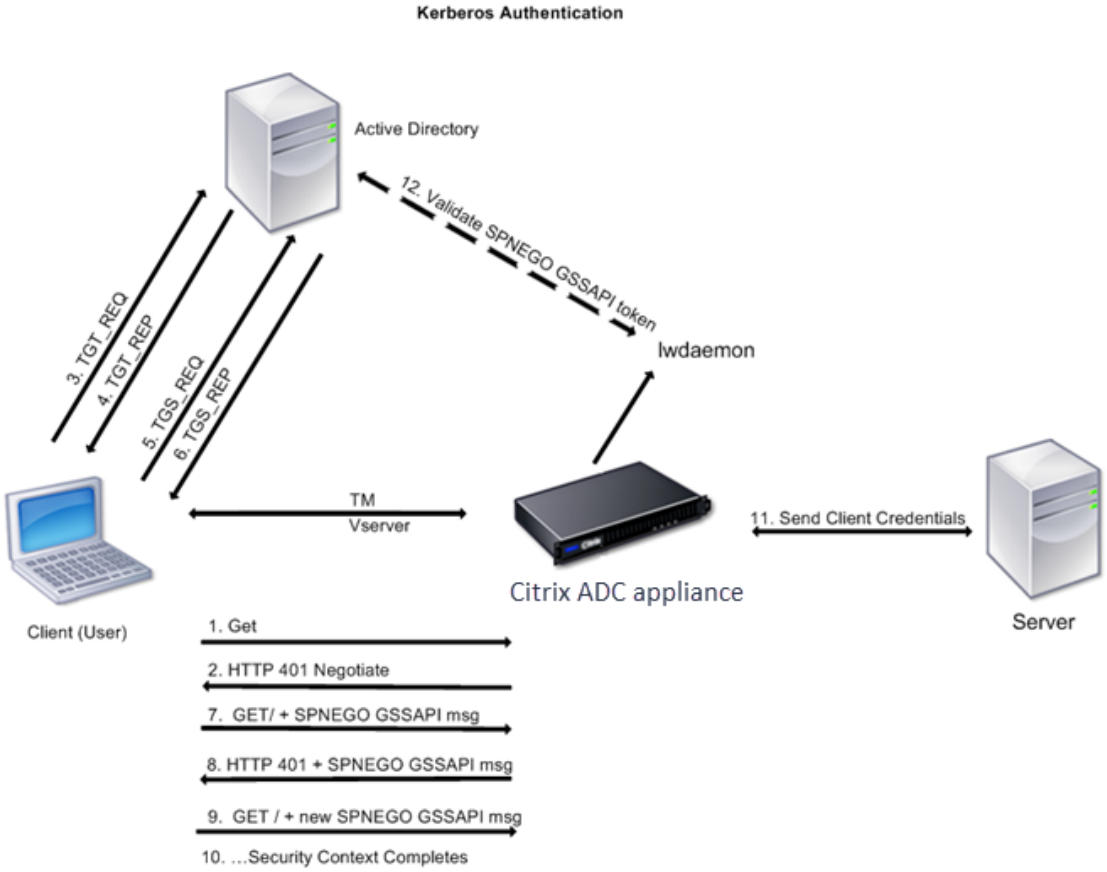
高可用性

高可用性セットアップでは、アクティブな Citrix ADC のみがドメインに参加します。フェイルオーバーの場合、Citrix ADC lwagent デーモンはセカンダリ Citrix ADC アプライアンスをドメインに参加させます。この機能に特定の設定は必要ありません。

Kerberos 認証プロセス

次の図は、Citrix ADC 環境における Kerberos 認証の一般的なプロセスを示しています。

図 1: Citrix ADC での Kerberos 認証プロセス



Kerberos 認証は、次の段階で行われます。

クライアントが **KDC** に対して自身を認証する

1. Citrix ADC アプライアンスは、クライアントから要求を受信します。
2. Citrix ADC アプライアンスのトラフィック管理（負荷分散またはコンテンツスイッチング）仮想サーバーは、クライアントにチャレンジを送信します。
3. このチャレンジに対応するために、クライアントは Kerberos チケットを取得します。

- クライアントは、KDC の認証サーバにチケット認可チケット (TGT) の要求を送信し、TGT を受信します。(図「Kerberos 認証プロセス」の 3、4 を参照してください)。
- クライアントは TGT を KDC のチケット交付サーバーに送信し、Kerberos チケットを受け取ります。(図「Kerberos 認証プロセス」の 5、6 を参照してください)。

> 注
>
> クライアントがライフタイムの期限が切れていない Kerberos チケットをすでに持っている場合、上記の認証プロセスは不要です。さらに、SPNEGO をサポートする Web サービス、.NET、または J2EE などのクライアントは、ターゲットサーバーの Kerberos チケットを取得し、SPNEGO トークンを作成し、HTTP 要求を送信するときに HTTP ヘッダーにトークンを挿入します。クライアント認証プロセスは行われません。

クライアントがサービスを要求します。

1. クライアントは、SPNEGO トークンと HTTP 要求を含む Kerberos チケットを Citrix ADC 上のトラフィック管理仮想サーバーに送信します。SPNEGO トークンに必要な GSSAPI データがあります。
2. Citrix ADC アプライアンスは、クライアントと Citrix ADC の間にセキュリティコンテキストを確立します。Citrix ADC が Kerberos チケットで提供されたデータを受け付けることができない場合、クライアントは別のチケットを取得するように求められます。このサイクルは、GSSAPI データが受け入れられ、セキュリティコンテキストが確立されるまで繰り返されます。Citrix ADC 上のトラフィック管理仮想サーバーは、クライアントと物理サーバー間の HTTP プロキシとして機能します。

Citrix ADC アプライアンスが認証を完了します。

1. セキュリティコンテキストが完了すると、トラフィック管理仮想サーバーは SPNEGO トークンを検証します。
2. 有効な SPNEGO トークンから、仮想サーバーはユーザー ID と GSS 資格情報を抽出し、認証デーモンに渡します。
3. 認証が成功すると、Kerberos 認証が完了します。

Citrix ADC アプライアンスでの **Kerberos** 認証の構成

October 7, 2021

このトピックでは、CLI と GUI を使用して Citrix ADC アプライアンスで Kerberos 認証を構成するための詳細な手順について説明します。

CLI での **Kerberos** 認証の設定

1. 認証、承認、および監査機能を有効にして、アプライアンスでトラフィックの認証を確実にします。

   ns-cli-prompt> **enable ns feature** AAA

2. キータブファイルを Citrix ADC アプライアンスに追加します。Kerberos 認証中にクライアントから受信したシークレットを復号化するには、キータブファイルが必要です。単一のキータブファイルには、Citrix ADC

アプライアンス上のトラフィック管理仮想サーバーにバインドされているすべてのサービスの認証の詳細が含まれています。

まず、Active Directory サーバーでキータブファイルを生成してから、Citrix ADC アプライアンスに転送します。

- Active Directory サーバーにログオンし、Kerberos 認証用のユーザーを追加します。たとえば、「Kerb-SVC-Account」という名前のユーザーを追加するには、次のようにします。

  ネットユーザー **Kerb-SVC** アカウント無料です！ **@ #456 /**追加

  > 注
  >
  > [ユーザーのプロパティ] セクションで、[次のログオン時にパスワードを変更する] オプションが選択されておらず、[パスワードの有効期限] オプションが選択されていることを確認します。

- 上記のユーザーに HTTP サービスをマッピングし、キータブファイルをエクスポートします。たとえば、Active Directory サーバーで次のコマンドを実行します。

  ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM /pass freebsd!@#456 /mapuser newacp\dummy /ptype KRB5_NT_PRINCIPAL

  > 注
  >
  > 複数のサービスに認証が必要な場合は、複数のサービスをマッピングできます。さらに多くのサービスをマッピングする場合は、すべてのサービスに対して上記のコマンドを繰り返します。出力ファイルには同じ名前を指定することも、別の名前を指定することもできます。

- unix **ftp** コマンドまたは任意のファイル転送ユーティリティを使用して、キータブファイルを Citrix ADC アプライアンスに転送します。

3. Citrix ADC アプライアンスは、完全修飾ドメイン名（FQDN）からドメイン Controller IP アドレスを取得する必要があります。したがって、Citrix ADC に DNS サーバーを構成することをお勧めします。

   ns-cli-prompt> **add dns nameserver** <ip-address>

   > 注
   >
   > または、静的なホストエントリを追加するか、その他の方法を使用して、Citrix ADC アプライアンスがドメイン Controller FQDN 名を IP アドレスに解決できるようにします。

4. 認証アクションを設定し、認証ポリシーに関連付けます。

   - ネゴシエートアクションを設定します。

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

注: ドメインユーザおよびドメイン名の設定については、クライアントに移動し、次の例に示すように klist コマンドを使用します。

クライアント: ユーザー名 @ AAA.LOCAL

サーバー:http/onprem_idp.AAA.local @ AAA.LOCAL

add authentication negotiateAction <name> -domain <AAA.LOCAL> -domainUser <HTTP/onprem_idp.aaa.local>

- ネゴシエートポリシーを設定し、ネゴシエートアクションをこのポリシーに関連付けます。

  ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. 認証仮想サーバを作成し、ネゴシエートポリシーをそのサーバに関連付けます。

   - 認証仮想サーバーを作成します。

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - ネゴシエートポリシーを認証仮想サーバにバインドします。

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. 認証仮想サーバをトラフィック管理（ロードバランシングまたはコンテンツスイッチング）仮想サーバに関連付けます。

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の設定を行うことができます。

7. 次の手順を実行して、設定を確認します。

   - FQDN を使用して、トラフィック管理仮想サーバにアクセスします。たとえば、サンプル
   - CLI でセッションの詳細を表示します。

     ns-cli-prompt> **show aaa session**

GUI での Kerberos 認証の設定

1. 認証、認可、および監査機能を有効にします。

   [システム] > [設定] に移動し、[基本機能の構成] をクリックして、認証、承認、監査機能を有効にします。

2. 上記の CLI 手順のステップ 2 の説明に従って、キータブファイルを追加します。

3. DNS サーバーを追加します。

   [トラフィック管理] **> [DNS] > [**ネームサーバー] に移動し、DNS サーバーの IP アドレスを指定します。

4. [ネゴシエート] アクションとポリシーを設定します。

   [セキュリティ] **>** **[AAA-**アプリケーショントラフィック**] >** **[**ポリシー**] >** **[**認証**] >** **[**高度なポリシー**] >** **[**ポリシー] に移動し、アクションタイプとして [ネゴシエート] を使用してポリシーを作成します。[**ADD**] をクリックして新しい認証ネゴシエートサーバを作成するか、[**Edit**] をクリックして既存の詳細を設定します。

5. ネゴシエートポリシーを認証仮想サーバにバインドします。

   [セキュリティ] **>** **[AAA-**アプリケーショントラフィック**] >** **[**仮想サーバ] に移動し、ネゴシエートポリシーを認証仮想サーバに関連付けます。

6. 認証仮想サーバをトラフィック管理（ロードバランシングまたはコンテンツスイッチング）仮想サーバに関連付けます。

   [トラフィック管理**] >** **[**ロードバランシング**] >** **[**仮想サーバー] に移動し、関連する認証設定を指定します。

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の設定を行うことができます。

7. 上記の CLI 手順のステップ 7 で詳述した設定を確認します。

クライアントで **Kerberos** 認証を構成する

October 7, 2021

認証に Kerberos を使用するには、ブラウザで Kerberos サポートを構成する必要があります。Kerberos 準拠の任意のブラウザを使用できます。Internet Explorer および Mozilla Firefox で Kerberos サポートを設定する手順は次のとおりです。その他のブラウザについては、ブラウザのドキュメントを参照してください。

Kerberos 認証用に **Internet Explorer** を構成するには

1. [ツール**]** メニューの **[**インターネットオプション**]** を選択します。
2. [セキュリティ] タブで、[ローカルイントラネット] をクリックし、[サイト] をクリックします。
3. [ローカルイントラネット] ダイアログボックスで、[イントラネットネットワークを自動的に検出する] オプションが選択されていることを確認し、[詳細設定] をクリックします。
4. [ローカルイントラネット] ダイアログボックスで、Citrix ADC アプライアンス上のトラフィック管理仮想サーバーのドメインの Web サイトを追加します。指定したサイトはローカルイントラネットサイトになります。
5. [閉じる**]** または **[OK]** をクリックしてダイアログボックスを閉じます。

Kerberos 認証用に **Mozilla Firefox** を設定するには

1. コンピューターに Kerberos が正しく構成されていることを確認します。
2. URL バーに about: config と入力します。

3. フィルタテキストボックスに「network.negotiate」と入力します。
4. 追加するドメインに、ネットワークを変更します。
5. 追加するドメインに、ネットワークを変更します。

> 注： Windows を実行している場合は、フィルターテキストボックスに sspi と入力し、network.auth.use-sspi オプションを False に変更する必要もあります。

物理サーバーから **Kerberos** 認証をオフロードします

January 31, 2022

Citrix ADC アプライアンスは、認証タスクをサーバーからオフロードできます。Citrix ADC は、クライアントからの要求を認証する物理サーバーではなく、すべてのクライアント要求を認証してから、バインドされた物理サーバーに転送します。ユーザー認証は、Active Directory トークンに基づいています。

Citrix ADC と物理サーバー間の認証は行われず、認証オフロードはエンドユーザーに対して透過的です。Windows コンピューターへの初回ログオン後、エンドユーザーは、ポップアップまたはログオンページに追加の認証情報を入力する必要はありません。

現在の Citrix ADC アプライアンスのリリースでは、Kerberos 認証は、トラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。Kerberos 認証は、Citrix Gateway アドバンスエディションアプライアンスでの SSL VPN または Citrix ADC アプライアンスの管理ではサポートされません。

Kerberos 認証では、Citrix ADC アプライアンスおよびクライアントブラウザでの構成が必要です。

Citrix ADC アプライアンスで **Kerberos** 認証を構成するには

1. Active Directory にユーザー・アカウントを作成します。ユーザー・アカウントを作成するときは、「ユーザー・プロパティ」セクションで次のオプションを確認します。
   - [次回ログオン時にパスワードを変更する] オプションを選択していないことを確認します。
   - [パスワードの有効期限切れ] オプションを必ず選択してください。
2. AD サーバーで、CLI コマンドプロンプトで次のように入力します。
   - ktpass-princ HTTP/ kerberos.crete.lab.net@crete.lab.net-ptype KRB5_NT_PRINCIPAL-マップユーザー-マップポップセット-<password> アウト C: kerbtabfile.txt kerbuser@crete.lab.net

   > 注
   >
   > 上記のコマンドを 1 行で入力してください。kerbtabfile.txt ファイル：上記のコマンドの出力は、C に書き込まれます。
3. セキュアコピー（SCP）クライアントを使用して、kerbtabfile.txt ファイルを Citrix ADC アプライアンスの /etc ディレクトリにアップロードします。

4. 次のコマンドを実行して、Citrix ADC アプライアンスに DNS サーバーを追加します。

   - DNS ネームサーバーを追加する 1.2.3.4

   Citrix ADC アプライアンスは、DNS サーバーがないと Kerberos 要求を処理できません。Microsoft Windows ドメインで使用されているものと同じ DNS サーバーを使用してください。

5. Citrix ADC コマンドラインインターフェイスに切り替えます。
6. 次のコマンドを実行して、Kerberos 認証サーバーを作成します。

   - 認証の追加ネゴシエーションアクション Kerberos サーバードメイン「crete.lab.net」-ドメインユーザーユーザーユーザーユーザーユーザーパスワード <password>-keytab /var/mykcd.keytab

   > 注
   >
   > キータブが使用できない場合は、ドメイン、ドメインユーザー、および-domainUserPasswd のパラメーターを指定できます。

7. 次のコマンドを実行して、ネゴシエーション・ポリシーを作成します。

   - `add authentication negotiatePolicy Kerberos-Policy ”REQ.IP.DESTIP == 192.168.17.200”KerberosServer<!--NeedCopy-->`

8. 次のコマンドを実行して、認証仮想サーバーを作成します。

   - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`

9. 次のコマンドを実行して、Kerberos ポリシーを認証仮想サーバーにバインドします。

   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`

10. 次のコマンドを実行して、SSL 証明書を認証仮想サーバーにバインドします。GUI の Citrix ADC アプライアンスからインストールできるテスト証明書のいずれかを使用できます。次のコマンドを実行して、ServerTestCert サンプル証明書を使用します。

    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`

11. IP アドレス 192.168.17.200 を使用して、HTTP ロードバランシング仮想サーバーを作成します。

    NetScaler 9.3 リリースが 9.3.47.8 より古い場合は、コマンドラインインターフェイスから仮想サーバーを作成してください。

12. 次のコマンドを実行して、認証仮想サーバーを構成します。

    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`

13. Web ブラウザのアドレスバーにホスト名 Example を入力します。

    Kerberos 認証がブラウザで設定されていないため、Web ブラウザに認証ダイアログボックスが表示されます。

    > 注
    >
    > Kerberos 認証には、クライアント上で特定の設定が必要です。クライアントがホスト名を解決できることを確認します。これにより、Web ブラウザが HTTP 仮想サーバに接続されます。

14. クライアントコンピュータの Web ブラウザで Kerberos を構成します。

    - Internet Explorer での構成については、「Kerberos 認証用の Internet Explorer の構成」を参照してください。
    - Mozilla Firefox での設定については、Internet Explorer の Kerberos 認証の設定を参照してください。

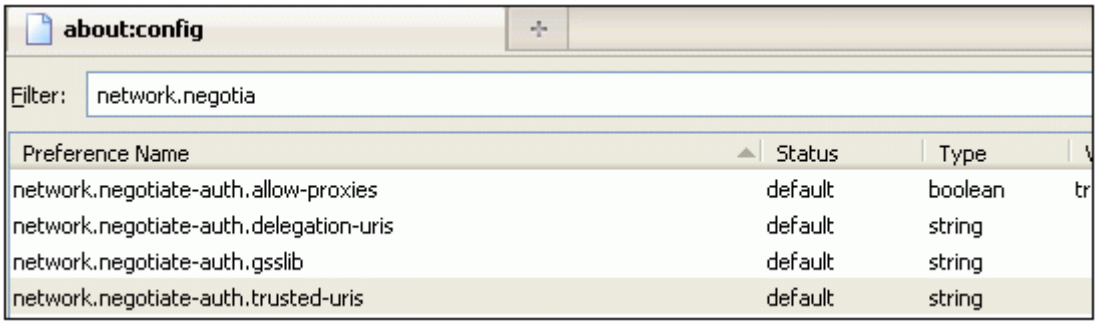
15. 認証なしでバックエンド物理サーバにアクセスできるかどうかを確認します。

Kerberos 認証用に **Internet Explorer** を構成するには

1. [ツール] メニューの [インターネットオプション] を選択します。
2. [セキュリティ] タブをアクティブにします。
3. [セキュリティ設定の変更を表示するゾーンの選択] セクションで、[ローカルイントラネット] を選択します。
4. [サイト] をクリックします。
5. [詳細設定] をクリックします。
6. URL、例] を指定し、[追加] をクリックします。
7. **Internet Explorer** を再起動します。

Kerberos 認証用に **Mozilla Firefox** を設定するには

1. ブラウザのアドレスバーに about: config と入力します。
2. 警告免責事項をクリックします。
3. [フィルタ] ボックスに「ネットワーク」と入力します。
4. [ネットワーク] をダブルクリックします。[ネゴシエート認証. 信頼された **uris]** をクリックします。画面の例を以下に示します。



5. [文字列値の入力] ダイアログボックスで、www.crete.lab.net と指定します。
6. Firefox を再起動します。

シングルサインオンのタイプ

May 9, 2022

Citrix ADC 認証、承認、および監査機能は、次のシングルサインオンの種類をサポートしています。

- **Citrix ADC kerberos** シングルサインオン： Citrix ADC アプライアンスは、Kerberos 5 プロトコルを使用したシングルサインオン（SSO）をサポートするようになりました。ユーザーは、プロキシであるアプリケーション Delivery Controller（ADC）にログオンし、保護されたリソースへのアクセスを提供します。詳細については、Citrix ADC kerberos シングルサインオンを参照してください。
- 基本認証、ダイジェスト認証、および **NTLM** 認証用の **SSO**：Citrix ADC および Citrix Gateway のシングルサインオン（SSO）構成は、グローバルレベルおよびトラフィックレベルごとに有効にできます。デフォルトでは、SSO 設定は OFF で、管理者はトラフィックごとまたはグローバルに SSO を有効にできます。セキュリティの観点から、管理者は SSO をグローバルにオフにし、トラフィックごとに有効にすることをお勧めします。この機能拡張は、特定のタイプの SSO メソッドをグローバルに無効にすることにより、SSO 構成をより安全にすることです。詳細については、ベーシック、ダイジェスト、NTLM 認証の SSOを参照してください。

Citrix ADC kerberos シングルサインオン

October 7, 2021

Citrix ADC アプライアンスは、Kerberos 5 プロトコルを使用したシングルサインオン（SSO）をサポートするようになりました。ユーザーは、プロキシであるアプリケーション Delivery Controller（ADC）にログオンし、保護されたリソースへのアクセスを提供します。

Citrix ADC Kerberos SSO 実装では、基本認証、NTLM 認証、またはフォームベースの認証に依存する SSO メソッドのユーザーのパスワードが必要です。Kerberos SSO ではユーザーのパスワードは必要ありません。ただし、

Kerberos SSO が失敗し、Citrix ADC アプライアンスがユーザーのパスワードを持っている場合は、パスワードを使用して NTLM SSO を試行します。

ユーザーのパスワードが使用可能で、KCD アカウントがレルムで構成され、委任されたユーザー情報が存在しない場合、Citrix AD Kerberos SSO エンジンはユーザーを偽装して、承認されたリソースへのアクセスを取得します。偽装は、拘束されていない委任とも呼ばれます。

Citrix ADC Kerberos SSO エンジンは、委任されたアカウントを使用して、ユーザーに代わって保護されたリソースへのアクセスを取得するように構成することもできます。この構成では、委任されたユーザーの資格情報、キータブ、または委任されたユーザー証明書と、一致する CA 証明書が必要です。委任されたアカウントを使用する構成は、制約付き委任と呼ばれます。

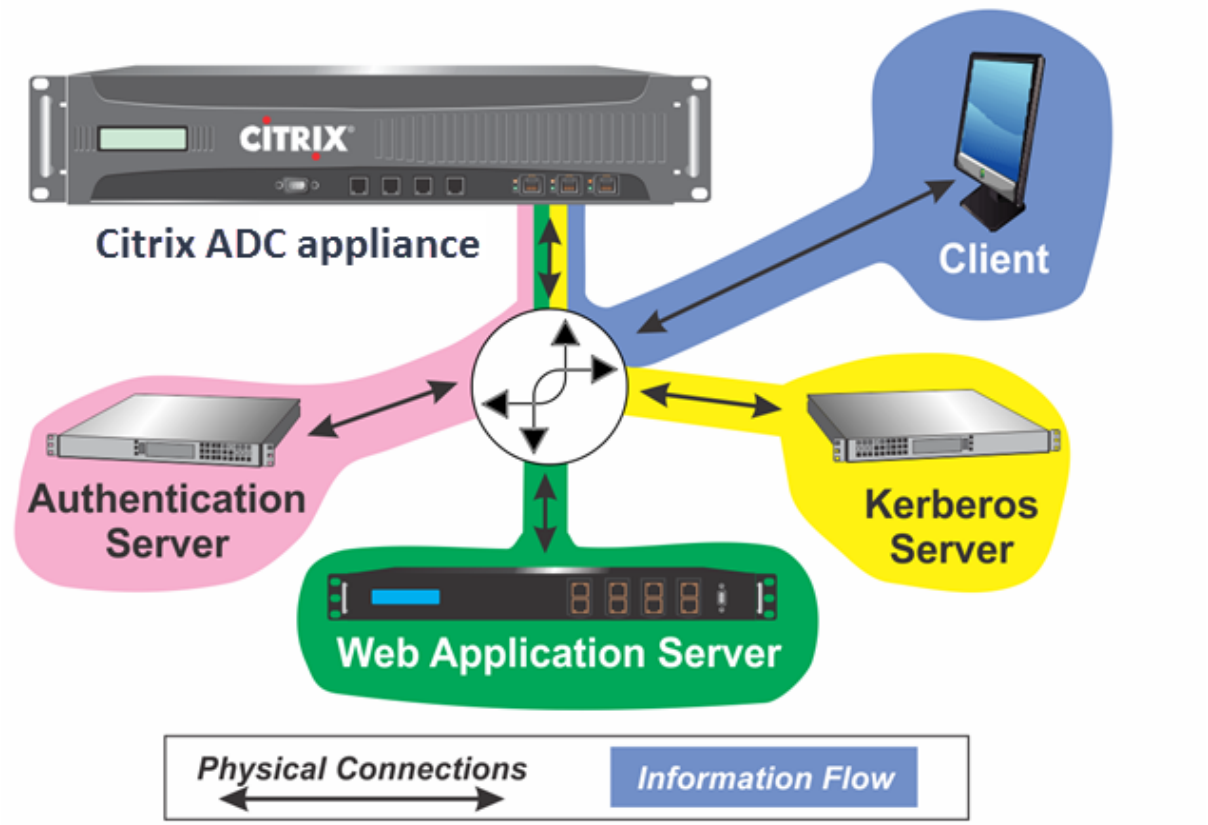
Citrix ADC Kerberos SSO の概要

October 7, 2021

Citrix ADC Kerberos SSO 機能を使用するには、まず Kerberos またはサポートされているサードパーティ認証サーバーを使用して認証します。認証されると、ユーザーは保護された Web アプリケーションへのアクセスを要求します。Web サーバーは、ユーザーがその Web アプリケーションへのアクセスを許可されていることを証明する要求で応答します。ユーザーのブラウザーは Kerberos サーバーに接続し、ユーザーがそのリソースにアクセスする権限があることを確認し、証明を提供するサービスチケットをユーザーのブラウザーに提供します。ブラウザは、サービスチケットが添付された状態でユーザーの要求を Web アプリケーションサーバーに再送信します。Web アプリケーションサーバーはサービスチケットを検証し、ユーザーがアプリケーションにアクセスできるようにします。

認証、認可、および監査トラフィック管理は、次の図に示すように、このプロセスを実装します。この図は、LDAP 認証と Kerberos 認証を使用するセキュアなネットワーク上で、Citrix ADC アプライアンスを介した情報のフローと、認証、承認、監査トラフィック管理を示しています。他のタイプの認証を使用する認証、認可、および監査トラフィック管理環境では、基本的に同じ情報フローがありますが、詳細が異なる場合があります。

図 1: LDAP と Kerberos を使用するセキュアなネットワーク



Kerberos 環境での認証と認可を使用した認証、認可、および監査トラフィック管理では、次のアクションを実行する必要があります。

1. クライアントは、Citrix ADC アプライアンス上のトラフィック管理仮想サーバーにリソース要求を送信します。
2. トラフィック管理仮想サーバは、要求を認証仮想サーバに渡します。この仮想サーバはクライアントを認証し、その要求をトラフィック管理仮想サーバに戻します。
3. トラフィック管理仮想サーバは、クライアントの要求を Web アプリケーションサーバに送信します。
4. Web アプリケーションサーバーは、Kerberos 認証を要求する 401 Unauthorized メッセージでトラフィック管理仮想サーバーに応答し、クライアントが Kerberos をサポートしていない場合は NTLM 認証にフォールバックします。
5. トラフィック管理仮想サーバは Kerberos SSO デーモンに接続します。
6. Kerberos SSO デーモンは Kerberos サーバーに接続し、チケット許可チケット (TGT) を取得します。これにより、保護されたアプリケーションへのアクセスを許可するサービスチケットを要求できます。
7. Kerberos SSO デーモンは、ユーザのサービスチケットを取得し、そのチケットをトラフィック管理仮想サーバに送信します。
8. トラフィック管理仮想サーバーは、チケットをユーザーの最初の要求にアタッチし、変更された要求を Web アプリケーション・サーバーに送信します。
9. Web アプリケーションサーバーは 200 OK メッセージで応答します。

これらのステップはクライアントに対して透過的です。クライアントは要求を送信し、要求されたリソースを受信す

るだけです。

認証方法を使用した **Citrix ADC Kerberos SSO** の統合

すべての認証、承認、および監査トラフィック管理認証メカニズムは、Citrix ADC Kerberos SSO をサポートしています。認証、承認、監査のトラフィック管理では、Kerberos、CAC（スマートカード）、SAML 認証メカニズムを使用した Kerberos SSO メカニズムがサポートされ、Citrix ADC アプライアンスに対する任意の形式のクライアント認証がサポートされます。また、クライアントが HTTP 基本認証またはフォームベースの認証を使用して Citrix ADC アプライアンスにログオンする場合、HTTP ベーシック、HTTP ダイジェスト、フォームベース、および NTLM（バージョン 1 および 2）の SSO メカニズムもサポートされます。

次の表に、サポートされている各クライアント側の認証方法と、そのクライアント側の認証方法でサポートされているサーバー側の認証方法を示します。

表 1. サポートされている認証方法

	基本/ダイジェスト/NTLM	Kerberos 制約付き委任	ユーザーなりすまし
CAC（スマートカード）: SSL/T LS レイヤ		☒	☒
フォームベース（LDAP/RADIUS/ACACS）	☒	☒	☒
HTTP 基本（LDAP/RADIUS/ACACS）	☒	☒	☒
kerberos		☒	
NT LM v1/v2		☒	☒
SAML		☒	
SAML 2 要素	☒	☒	☒
証明書の 2 ファクタ	☒	☒	☒

Citrix ADC SSO を設定する

January 31, 2022

Citrix ADC SSO は、偽装または委任のいずれかの方法で動作するように構成できます。偽装による SSO は、委任に

よる SSO よりも簡単な設定であるため、通常は、設定によって許可される場合に適しています。偽装によって Citrix ADC SSO を構成するには、ユーザーのユーザー名とパスワードが必要です。

委任によって Citrix ADC SSO を構成するには、ユーザーのユーザー名とパスワード、ユーザー名と暗号化されたパスワードを含むキータブ構成、委任されたユーザー証明書と一致する CA 証明書のいずれかの形式で委任されたユーザーの資格情報が必要です。

Citrix ADC SSO を構成するための前提条件

Citrix ADC SSO を構成する前に、Citrix ADC アプライアンスを完全に構成して、Web アプリケーションサーバーへのトラフィックと認証を管理する必要があります。したがって、これらの Web アプリケーションサーバーに対して、負荷分散またはコンテンツの切り替えを構成し、認証、承認、および監査を構成する必要があります。アプライアンス、LDAP サーバー、および Kerberos サーバー間のルーティングも確認する必要があります。

ネットワークがこの方法でまだ設定されていない場合は、次の設定作業を実行します。

- Web アプリケーションサーバーごとに、サーバーとサービスを構成します。
- Web アプリケーションサーバーとの間で送受信されるトラフィックを処理するように、トラフィック管理仮想サーバーを設定します。

次に、Citrix ADC コマンドラインからこれらの各タスクを実行するための簡単な手順と例を示します。詳細については、「認証仮想サーバーのセットアップ」を参照してください。

CLI を使用してサーバーとサービスを作成するには

Citrix ADC SSO がサービスの TGS（サービス・チケット）を取得するには、Citrix ADC アプライアンスのサーバー・エンティティに割り当てられた FQDN が Web アプリケーション・サーバーの FQDN と一致するか、サーバー・エンティティ名が Web アプリケーション・サーバーの NetBios 名と一致する必要があります。次のいずれかの方法を使用できます。

- Web アプリケーションサーバーの FQDN を指定して、Citrix ADC サーバーエンティティを構成します。
- Web アプリケーションサーバーの IP アドレスを指定して Citrix ADC サーバーエンティティを構成し、Web アプリケーションサーバーの NetBios 名と同じ名前をサーバーエンティティに割り当てます。

コマンドプロンプトで、次のコマンドを入力します。

```
1 -  add server name <serverFQDN>
2
3 -  add service name serverName serviceType port
4 <!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **serverName**。このサーバーを参照するために使用する Citrix ADC アプライアンスの名前。

- **serverFQDN**。サーバーの FQDN。サーバーにドメインが割り当てられていない場合は、サーバーの IP アドレスを使用し、サーバーエンティティ名が Web アプリケーションサーバーの NetBios 名と一致していることを確認します。
- **serviceName**。このサービスを参照するために使用する Citrix ADC アプライアンスの名前。
- **type**。サービスによって使用されるプロトコル（HTTP または MSSQLSVC）。
- **port**。サービスがリッスンするポート。HTTP サービスは通常、ポート 80 でリッスンします。セキュリティで保護された HTTPS サービスは、通常、ポート 443 でリッスンします。

例**:**

次の例では、Web アプリケーションサーバー was1.example.com の Citrix ADC アプライアンスにサーバーエントリとサービスエントリを追加します。1 番目の例では、Web アプリケーションサーバーの FQDN を使用し、2 番目の例では IP アドレスを使用します。

Web アプリケーションサーバー FQDN was1.example.com を使用してサーバーとサービスを追加するには、次のコマンドを入力します。

```
add server was1 was1.example.com
add service was1service was1 HTTP 80
<!--NeedCopy-->
```

Web アプリケーションサーバーの IP と NetBios 名を使用してサーバーとサービスを追加するには、Web アプリケーションサーバーの IP が 10.237.64.87 で、NetBios 名が WAS1 である場合は、次のコマンドを入力します。

```
add server WAS1 10.237.64.87
add service was1service WAS1 HTTP 8
<!--NeedCopy-->
```

CLI を使用してトラフィック管理仮想サーバーを作成するには

トラフィック管理仮想サーバは、クライアントと Web アプリケーションサーバ間のトラフィックを管理します。トラフィック管理サーバーとして、負荷分散またはコンテンツスイッチング仮想サーバーを使用できます。SSO の設定は、どちらのタイプでも同じです。

負荷分散仮想サーバーを作成するには、コマンドプロンプトで次のコマンドを入力します。

```
add lb vserver <vserverName> <type> <IP> <port>
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **vserverName**：この仮想サーバーを参照するために使用する Citrix ADC アプライアンスの名前。
- **type**：サービスによって使用されるプロトコル（HTTP または MSSQLSVC）。
- [**IP**]：仮想サーバに割り当てられた IP アドレス。これは通常、LAN 上の IANA で予約された、非パブリック IP アドレスです。
- **port**：サービスがリッスンするポート。HTTP サービスは通常、ポート 80 でリッスンします。セキュリティで保護された HTTPS サービスは、通常、ポート 443 でリッスンします。

例**:**

tmvserver1 という負荷分散仮想サーバーを、ポート 80 の HTTP トラフィックを管理する構成に追加し、LAN IP アドレス 10.217.28.20 を割り当ててから、負荷分散仮想サーバーを wasservice1 サービスにバインドするには、次のコマンドを入力します。

```
1 add lb vserver tmvserver1 HTTP 10.217.28.20 80
2 bind lb vserver tmvserv1 wasservice1
3 <!--NeedCopy-->
```

CLI を使用して認証仮想サーバーを作成するには

認証仮想サーバは、クライアントと認証（LDAP）サーバ間の認証トラフィックを管理します。認証仮想サーバーを作成するには、コマンドプロンプトで次のコマンドを入力します。

```
1 add authentication vserver <authvserverName> SSL <IP> 443
2 <!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **authvserverName**： この認証仮想サーバーを参照するために使用する Citrix ADC アプライアンスの名前。文字、数字、またはアンダースコア文字 (_) で始まり、文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等しい (=)、コロン (:)、およびアンダースコア文字のみを含める必要があります。rename authentication vserver コマンドを使用して、認証仮想サーバを追加した後に変更できます。
- [**IP**]：認証仮想サーバに割り当てられた IP アドレス。トラフィック管理仮想サーバと同様に、このアドレスは通常、LAN 上の IANA 予約済みで非パブリック IP になります。
- **domain**：仮想サーバに割り当てられたドメイン。これは通常、ネットワークのドメインです。認証仮想サーバーを構成するときは、すべての大文字でドメインを入力することが通例ですが、必須ではありません。

例**:**

authver1 という名前の認証仮想サーバーを構成に追加し、LAN IP 10.217.28.21 とドメイン EXAMPLE.COM を割り当てるには、次のコマンドを入力します。


```
add authentication vserver authvserver1 SSL 10.217.28.21 443
<!--NeedCopy-->
```

認証プロファイルを使用するようにトラフィック管理仮想サーバを構成するには

認証仮想サーバーは、単一のドメインまたは複数のドメインの認証を処理するように構成できます。複数のドメインの認証をサポートするように構成されている場合は、認証プロファイルを作成し、その認証プロファイルを使用するようにトラフィック管理仮想サーバーを構成して、Citrix ADC SSO のドメインも指定する必要があります。

> 注
>
> トラフィック管理仮想サーバは、負荷分散（lb）またはコンテンツスイッチング（cs）仮想サーバのいずれかになります。次の手順では、負荷分散仮想サーバーを使用していることを前提としています。コンテンツスイッチング仮想サーバを設定するには、単に
> set cs vserver を set lb vserver に置き換えます。それ以外の場合は手順は同じです。

認証プロファイルを作成し、トラフィック管理仮想サーバーで認証プロファイルを構成するには、次のコマンドを入力します。

```
-  add authentication authnProfile <authnProfileName> {
 -authvserverName <string> }
  {
 -authenticationHost <string> }
  {
 -authenticationDomain <string> }

-  set lb vserver <vserverName> -authnProfile <authnprofileName>
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **authnprofileName**：認証プロファイルの名前。文字、数字、またはアンダースコア文字（_）で始まり、英数字またはハイフン（-）、ピリオド（.）ポンド（#）、スペース（）、アットマーク（@）、等しい（=）、コロン（:）、およびアンダースコア文字で構成されている必要があります。
- **authvserverName**：このプロファイルが認証に使用する認証仮想サーバの名前。
- **authenticationHost**：認証仮想サーバのホスト名。
- **authenticationDomain**：Citrix ADC SSO が認証を処理するドメイン。認証仮想サーバーが複数のドメインに対して認証を実行する場合、Citrix ADC アプライアンスがトラフィック管理仮想サーバーの Cookie を設定するときに正しいドメインが含まれるようにする必要があります。

例**:**

example.com ドメインの認証用に authnProfile1 という名前の認証プロファイルを作成し、認証プロファイル authnProfile1 を使用するように負荷分散仮想サーバー vserver1 を構成するには、次のコマンドを入力します。

```
1 add authentication authnProfile authnProfile1 -authnvsName
     authvsesrver1
2       -authenticationHost authvsesrver1 -authenticationDomain example.
           com
3 set lb vserver vserver1 -authnProfile authnProfile1
4 <!--NeedCopy-->
```

シングルサインオンを構成する

July 15, 2022

なりすましによる認証を行うように Citrix ADC シングルサインオン（SSO）を構成することは、SSO よりも委任による認証を行うように構成するよりも簡単であるため、構成で許可されている場合に適しています。KCD アカウントを作成するとします。ユーザーのパスワードを使用できます。

ユーザーのパスワードがない場合は、委任によって認証されるように Citrix ADC SSO を構成できます。偽装によって認証するように SSO を構成するよりも複雑ですが、委任方法を使用すると、すべての状況で Citrix ADC アプライアンスがユーザーの資格情報を使用できないという柔軟性が得られます。

偽装または委任の場合は、Web アプリケーションサーバーでも統合認証を有効にする必要があります。

Web アプリケーションサーバーで統合認証を有効にする

Kerberos SSO が管理する各 Web アプリケーションサーバーで Citrix ADC Kerberos SSO を設定するには、そのサーバーの構成インターフェイスを使用して、認証を要求するようにサーバーを構成します。Kerberos をサポートしていないクライアントの場合は NTLM にフォールバックして、Kerberos (ネゴシエート) 認証を優先的に選択します。

次に、認証を要求するように Microsoft インターネットインフォメーションサーバー (IIS) を構成する手順を示します。Web アプリケーションサーバで IIS 以外のソフトウェアを使用している場合は、その Web サーバソフトウェアのマニュアルを参照して手順を確認してください。

統合認証を使用するように **Microsoft IIS** を構成するには

1. IIS サーバーにログオンし、インターネットインフォメーションサービスマネージャーを開きます。
2. 統合認証を有効にする Web サイトを選択します。IISM が管理するすべての IIS Web サーバーに対して統合認証を有効にするには、既定の Web サイトの認証設定を構成します。個々のサービス (Exchange、Exadmin、

ExchWeb、Public など) に対して統合認証を有効にするには、これらの認証設定をサービスごとに個別に構成します。

3. 既定の Web サイトまたは個々のサービスの [プロパティ] ダイアログボックスを開き、[ディレクトリセキュリティ] タブをクリックします。
4. [****** 認証とアクセス制御**]** の横にある **[**** 編集**]** を選択します。
5. 匿名アクセスを無効にしてください。
6. 統合 Windows 認証を有効にします (のみ)。統合 Windows 認証を有効にすると、Web サーバのプロトコルネゴシエーションが Negotiate、NTLM に自動的に設定される必要があります。これは、Kerberos 非対応デバイスの NTLM にフォールバックする Kerberos 認証を指定します。このオプションが自動的に選択されない場合は、プロトコルネゴシエーションを [Negotiate, NTLM] に手動で設定します。

偽装による **SSO** の設定

Citrix ADC SSO の KCD アカウントは、偽装によって構成できます。この構成では、Citrix ADC アプライアンスは、ユーザーが認証サーバーに対して認証されると、ユーザーのユーザー名とパスワードを取得し、その資格情報を使用してユーザーを偽装してチケット許可チケット（TGT）を取得します。ユーザー名が UPN 形式の場合、アプライアンスはユーザーのレルムを UPN から取得します。それ以外の場合は、初期認証時に使用された SSO ドメインまたはセッションプロファイルからユーザーの名前とレルムを抽出して取得します。

> 注
>
> domain なしでユーザー名が既に追加されている場合、domain でユーザー名を追加することはできません。ドメインのあるユーザー名が最初に追加され、次にドメインのない同じユーザー名が追加された場合、Citrix ADC アプライアンスはユーザー名をユーザーリストに追加します。

KCD アカウントを設定するときは、ユーザーがアクセスしているサービスのレルムに realm パラメーターを設定する必要があります。Citrix ADC アプライアンスでの認証またはセッションプロファイルからユーザーのレルムを取得できない場合は、同じレルムがユーザーのレルムとしても使用されます。

パスワードを使用して偽装して **SSO** の **KCD** アカウントを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add aaa kcdaccount <accountname> -realmStr <realm>

<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- アカウント名。KCD アカウント名。
- レルム。Citrix ADC SSO に割り当てられたドメイン。

例

kcdccount1 という名前の KCD アカウントを追加し、kcdvserver.keytab という名前のキータブを使用するには、次のコマンドを入力します。

```
add aaa kcdAccount kcdaccount1 -keytab kcdvserver.keytab

<!--NeedCopy-->
```

Citrix ADC GUI を使用した Kerberos 偽装の構成については、Citrix サポートを参照してください。

委任による **SSO** の構成

委任による SSO を設定するには、次のタスクを実行する必要があります。

- 委任されたユーザー証明書による委任を構成する場合は、一致する CA 証明書を Citrix ADC アプライアンスにインストールし、Citrix ADC 構成に追加します。
- アプライアンスで KCD アカウントを作成します。アプライアンスはこのアカウントを使用して、保護対象アプリケーションのサービスチケットを取得します。
- Active Directory サーバを設定します。

> 注:
>
> NetScaler アプライアンスでの KCD アカウントの作成と構成について詳しくは、次のトピックを参照してください。
>
> - Kerberos/NTLM による認証、承認、監査の処理
> - Citrix ADC がクライアント認証用に Kerberos を実装する方法
> - Citrix ADC アプライアンスでの Kerberos 認証の構成

Citrix ADC アプライアンスへのクライアント **CA** 証明書のインストール

クライアント証明書を使用して Citrix ADC SSO を構成する場合は、クライアント証明書ドメインに一致する CA 証明書（クライアント CA 証明書）を Citrix ADC アプライアンスにコピーしてから、CA 証明書をインストールする必要があります。クライアント CA 証明書をコピーするには、選択したファイル転送プログラムを使用して証明書と秘密キーファイルを Citrix ADC アプライアンスに転送し、ファイルを/nsconfig/ssl に保存します。

Citrix ADC アプライアンスにクライアント **CA** 証明書をインストールするには

コマンドプロンプトで、次のコマンドを入力します。


```
add ssl certKey <certkeyName> -cert <cert> [(-key <key> [-password]) |
    -fipsKey <fipsKey>][-inform ( DER | PEM )][-expiryMonitor ( ENABLED
    | DISABLED | UNSET ) [-notificationPeriod <positive_integer>]] [-
    bundle ( YES | NO )]

<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **certKeyName**。クライアント CA 証明書の名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、1～31 文字で構成されている必要があります。使用できる文字は、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等号 (=)、ハイフン (-) です。証明書とキーのペアの作成後は変更できません。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「my cert」や「my cert」など)。
- 証明書。証明書とキーのペアを形成するために使用される X509 証明書ファイルのフルパス名とファイル名。証明書ファイルは、Citrix ADC アプライアンスの/nsconfig/ssl/ ディレクトリに保存する必要があります。
- キー。X509 証明書ファイルへの秘密キーを含むファイルのフルパス名とファイル名。キーファイルは、Citrix ADC アプライアンスの/nsconfig/ssl/ ディレクトリに保存する必要があります。
- パスワード。秘密鍵を指定した場合、秘密鍵の暗号化に使用されるパスフレーズ。暗号化された秘密キーを PEM 形式でロードするには、このオプションを使用します。
- **fipsKey**。FIPS アプライアンスのハードウェアセキュリティモジュール (HSM) 内で作成された FIPS キー、または HSM にインポートされたキーの名前。

  > 注
  >
  > キーと FipsKey のどちらかを指定できますが、両方を指定することはできません。

- **inform**. 証明書と秘密キーファイルの形式 (PEM または DER)。
- **passplain**. 秘密鍵の暗号化に使用されるパスフレーズ。暗号化された秘密キーを PEM 形式で追加する場合に必要です。
- **expiryMonitor**. 証明書の有効期限が近づいたときにアラートを発行するように Citrix ADC アプライアンスを構成します。有効な値: 有効、無効、未設定。
- **notificationPeriod**.`expiryMonitor`が ENABLED の場合、証明書が期限切れになる前にアラートを発行するまでの日数。
- **bundle**. サーバー証明書をファイル内の発行者の証明書にリンクした後、証明書チェーンを 1 つのファイルとして解析します。可能な値: はい、いいえ。

例

次の例では、指定された委任ユーザー証明書 customer-cert.pem をキー customer-key.pem とともに Citrix ADC 構成に追加し、パスワード、証明書形式、有効期限モニター、および通知期間を設定します。

委任されたユーザー証明書を追加するには、次のコマンドを入力します。

```
add ssl certKey customer -cert ”/nsconfig/ssl/customer-cert.pem”
-key ”/nsconfig/ssl/customer-key.pem” -password ”dontUseDefaultPWs!”
-inform PEM -expiryMonitor ENABLED [-notificationPeriod 14]

<!--NeedCopy-->
```

KCD アカウントの作成

Citrix ADC SSO を委任で構成する場合は、ユーザーのログオン名とパスワードを使用するか、ユーザーのログオン名とキータブを使用するか、ユーザーのクライアント証明書を使用するように、KCD アカウントを構成できます。ユーザー名とパスワードを使用して SSO を構成すると、Citrix ADC アプライアンスは委任されたユーザーアカウントを使用してチケット交付チケット（TGT）を取得し、TGT を使用して各ユーザーが要求する特定のサービスのサービスチケットを取得します。キータブファイルを使用して SSO を構成すると、Citrix ADC アプライアンスは委任されたユーザーアカウントとキータブ情報を使用します。委任ユーザー証明書を使用して SSO を構成すると、Citrix ADC アプライアンスは委任ユーザー証明書を使用します。

> 注:
>
> クロスレルムの場合、委任されたユーザーの servicePrincipalName はhost/<name>の形式である必要があります。この形式でない場合は、委任されたユーザーの servicePrincipalName <servicePrincipalName>をhost/<service-account-samaccountname>に変更します。委任されたユーザーアカウントの属性は、ドメインコントローラーで確認できます。変更する 1 つの方法は、委任されたユーザーのlogonName属性を変更することです。

パスワードを使用して委任して **SSO** の **KCD** アカウントを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add aaa kcdAccount <kcdAccount> {
 -realmStr <string> }
  {
 -delegatedUser <string> }
  {
 -kcdPassword  }
  [-userRealm <string>]
[-enterpriseRealm <string>] [-serviceSPN <string>]
<!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **kcdAccount** -KCD アカウントの名前。これは必須の議論です。最大長：31
- **realmStr** -ケルベロスの領域。最大長：255
- **delegatedUser** -Kerberos 制約付き委任を実行できるユーザー名。委任されたユーザー名は、ドメインコントローラーの ServicePrincipalName から派生します。クロスレルムの場合、委任されたユーザーの ServicePrincipalName はhost/<name>の形式である必要があります。最大長:255
- **kcdPassword** -委任されたユーザーのパスワード。最大長：31
- **userRealm** -ユーザーのレルム。最大長：255
- **EnterpriseRealm** -ユーザーのエンタープライズレルム。これは、KDC がプリンシパル名ではなくエンタープライズユーザー名を要求する特定の KDC 展開でのみ指定されます。最大長：255
- **serviceSPN** -サービス SPN。指定すると、Kerberos チケットの取得に使用されます。指定しない場合、Citrix ADC はサービス FQDN を使用して SPN を構築します。最大長：255

例 **(UPN** 形式**):**

パスワード 1、レルム EXAMPLE.COM を使用して、委任されたユーザーアカウントを UPN 形式（root）で指定して、kcdaccount1 という名前の KCD アカウントを Citrix ADC アプライアンス構成に追加するには、次のコマンドを入力します。

```
add aaa kcdaccount kcdaccount1 -delegatedUser root
-kcdPassword password1 -realmStr EXAMPLE.COM

<!--NeedCopy-->
```

例 **(SPN** 形式**):**

パスワード 1、レルム EXAMPLE.COM で、委任されたユーザーアカウントを SPN 形式で指定して、kcdaccount1 という名前の KCD アカウントを Citrix ADC アプライアンス構成に追加するには、次のコマンドを入力します。

```
add aaa kcdAccount kcdaccount1 -realmStr EXAMPLE.COM
-delegatedUser "host/kcdvserver.example.com" -kcdPassword password1

<!--NeedCopy-->
```

キータブを使用した委任による **SSO** の **KCD** アカウントの作成

認証に keytab ファイルを使用する場合は、まず keytab を作成します。キータブファイルは、AD サーバーにログオンしてktpassユーティリティを使用して手動で作成することも、Citrix ADC 構成ユーティリティを使用してバッチスクリプトを作成し、そのスクリプトを AD サーバーで実行してキータブファイルを生成することもできます。次に、FTP または別のファイル転送プログラムを使用してキータブファイルを Citrix ADC アプライアンスに転送し、/nsconfig/krb ディレクトリに配置します。最後に、委任によって Citrix ADC SSO の KCD アカウントを構成し、キータブファイルのパスとファイル名を Citrix ADC アプライアンスに指定します。

> 注:
>
> クロスレルムで Keytab ファイルを KCD アカウントの一部として取得する場合は、更新された委任ユーザー名に対して次のコマンドを使用します。
>
> ドメインコントローラーで、更新された Keytab ファイルを作成します。
>
> ```
> ktpass /princ <servicePrincipalName-with-prefix<host/>Of-delegateUser>@<DC REALM in uppercase> /ptype KRB5_NT_PRINCIPAL /mapuser <DC REALM in uppercase>\<sAMAccountName> /pass <delegatedUserPassword> -out filepathfor.keytab
> ```
>
> この`filepathfor.keytab`ファイルは Citrix ADC アプライアンスに配置でき、ADC KCD アカウントの Keytab 構成の一部として使用できます。

keytab ファイルを手動で作成するには

AD Server コマンドラインにログオンし、コマンドプロンプトで次のコマンドを入力します。

```
1 ktpass princ <SPN> ptype KRB5_NT_PRINCIPAL mapuser <DOMAIN><username>
     pass <password> -out <File_Path>
2 <!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- **SPN**。KCD サービスアカウントのサービスプリンシパル名。
- ドメイン。Active Directory サーバのドメイン。
- **username**）を使用します。KSA アカウントのユーザー名。
- パスワード。KSA アカウントのパスワード。
- 道。keytab ファイルが生成された後に保存されるディレクトリのフルパス名。

Citrix ADC 構成ユーティリティを使用して、**keytab** ファイルを生成するスクリプトを作成するには

1. セキュリティ **> AAA-**アプリケーショントラフィックに移動します。
2. データウィンドウの [**Kerberos** 制約付き委任] で、[バッチファイル] をクリックして Keytab を生成します。
3. 「**KCD (Kerberos** 制約付き委任**) Keytab** スクリプトの生成」ダイアログ・ボックスで、次のパラメータを設定します。
   - ドメインユーザー名。KSA アカウントのユーザー名。
   - ドメインパスワード。KSA アカウントのパスワード。
   - サービスプリンシパル KSA のサービスプリンシパル名。
   - 出力ファイル名。キータブファイルを AD サーバーに保存するフルパスとファイル名。
4. **[**ドメインユーザーアカウントの作成] チェックボックスをオフにします。

5. [スクリプトを生成] をクリックします。
6. Active Directory サーバにログオンし、コマンドラインウィンドウを開きます。
7. [生成されたスクリプト] ウィンドウからスクリプトをコピーし、Active Directory サーバのコマンドラインウィンドウに直接貼り付けます。キータブが生成され、「出力ファイル名」**(Output File Name)** として指定したファイル名のディレクトリに格納されます。
8. 選択したファイル転送ユーティリティを使用して、キータブファイルを Active Directory y サーバーから Citrix ADC アプライアンスにコピーし、/nsconfig/krb ディレクトリに配置します。

KCD アカウントを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
1 add aaa kcdaccount <accountname>  -keytab <keytab>
2 <!--NeedCopy-->
```

例

kcdccount1 という名前の KCD アカウントを追加し、kcdvserver.keytab という名前のキータブを使用するには、次のコマンドを入力します。

```
1 add aaa kcdaccount kcdaccount1  -keytab kcdvserver.keytab
2 <!--NeedCopy-->
```

委任されたユーザー証明書を使用して委任によって **SSO** の **KCD** アカウントを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
1 add aaa kcdaccount <accountname> -realmStr <realm> -delegatedUser <
     user_nameSPN> -usercert <cert> -cacert <cacert>
2 <!--NeedCopy-->
```

変数の場合は、次の値を置き換えます。

- アカウント名。KCD アカウントの名前。
- **realmStr**。KCD アカウントのレルム。通常は SSO が設定されているドメインです。
- 委任されたユーザー。委任されたユーザー名 (SPN 形式)。
- **usercert**。Citrix ADC アプライアンス上の委任ユーザー証明書ファイルのフルパスと名前。委任されたユーザー証明書には、クライアント証明書と秘密キーの両方が含まれていて、PEM 形式である必要があります。ス

マートカード認証を使用する場合は、秘密キーとともに証明書をインポートできるように、スマートカード証明書テンプレートを作成する必要があります。

- **cacert**。Citrix ADC アプライアンス上の CA 証明書ファイルへのフルパスと名前。

例

kcdccount1 という名前の KCD アカウントを追加し、kcdvserver.keytab という名前のキータブを使用するには、次のコマンドを入力します。

```
add aaa kcdaccount kcdaccount1 -realmStr EXAMPLE.COM
     -delegatedUser "host/kcdvserver.example.com" -usercert /certs/
        usercert
     -cacert /cacerts/cacert
<!--NeedCopy-->
```

Citrix ADC SSO 用の **Active Directory** セットアップ

委任によって SSO を構成する場合は、Citrix ADC アプライアンスに KcdAccount を作成することに加えて、一致する Kerberos サービスアカウント（KSA）を LDAP Active Directory サーバー上に作成し、そのサーバーを SSO 用に構成する必要があります。KSA を作成するには、Active Directory サーバでアカウント作成プロセスを使用します。Active Directory サーバで SSO を設定するには、KSA のプロパティウィンドウを開きます。[委任] タブで、[指定したサービスへの委任に対してのみこのユーザーを信頼する] と [任意の認証プロトコルを使用する] オプションを有効にします。(Kerberos only オプションは、プロトコルの移行や制約付き委任を有効にしないため、機能しません)。最後に、Citrix ADC SSO が管理するサービスを追加します。

> 注:
>
> [KSA アカウントのプロパティ] ダイアログボックスに [委任] タブが表示されない場合は、説明に従って KSA を構成する前に、Microsoft
> setspn コマンドラインツールを使用して、タブが表示されるように Active Directory サーバーを構成する必要があります。

Kerberos サービスアカウントの委任を構成するには

1. 作成した Kerberos サービスアカウントの [LDAP アカウント構成] ダイアログボックスで、[委任] タブをクリックします。
2. [指定したサービスへの委任に対してのみ、このユーザーを信頼する] を選択します。
3. [指定したサービスへの委任に対してのみこのユーザーを信頼する] で、[任意の認証プロトコルを使用する] を選択します。
4. [このアカウントが委任された認証情報を提示できるサービス] で、[追加] をクリックします。

5. [サービスの追加] ダイアログボックスで、[ユーザー] または [コンピューター] をクリックし、サービスアカウントに割り当てるリソースをホストするサーバーを選択して、[**OK**] をクリックします。

   > 注:
   >
   > - 制約付き委任では、Kerberos が他のドメインと信頼関係を持っている場合でも、アカウントに割り当てられたドメイン以外のドメインでホストされているサービスはサポートされません。
   > - 新しいユーザが Active Directory に作成された場合は、次のコマンドを使用して`setspn`を作成します。`setspn -A host/kcdvserver.example.com example\kcdtest`

6. [サービスの追加] ダイアログボックスの [利用可能なサービス] リストに戻り、サービスアカウントに割り当てられているサービスを選択します。Citrix ADC SSO は HTTP および MSSQLSVC サービスをサポートします。

7. 「**OK**」をクリックします。

KCD が子ドメインをサポートできるようにするための設定変更

KCD アカウントが-`delegatedUser`で`samAccountName`に設定されている場合、子ドメインからサービスにアクセスするユーザーに対して KCD は機能しません。この場合、Citrix ADC アプライアンスと Active Directory の構成を変更できます。

- AD のサービスアカウント <`service-account-samaccountname`> (KCD アカウントで delegateUser として設定されている) のログオン名を`host/<service-account-samaccountname>.<completeUSERDNSDOMAIN>`形式 (例: `host/svc_act.child.parent.com`) で変更します。

  サービスアカウントは手動で、または`ktpass`コマンドを使用して変更できます。`ktpass`は、サービスアカウントを自動的に更新します。

  ```
  ktpass /princ host/svc_act.child.parent.com@CHILD.PARENT.COM /ptype KRB5_NT_PRINCIPAL /mapuser CHILD\sv_act /pass serviceaccountpassword -out filepathfor.keytab
  ```

- Citrix ADC アプライアンスの KCD アカウントで DelegatedUser を変更します。
- KCD アカウントの-delegatedUser パラメータを次のように変更します。`host/svc_act.child.parent.com`

KCD アカウントの設定に高度な暗号化を使用する場合の注意点

> - **keytab** を使用する場合の設定例**:** `add kcdaccount lbvs_keytab_aes256 -keytab "/nsconfig/krb/kcd2_aes256.keytab"`
> - **keytab** に複数の暗号化タイプがある場合は、次のコマンドを使用します。このコマンドは、ドメインユーザーパラメーターもキャプチャします。`add kcdaccount lbvs_keytab_aes256 -keytab "/nsconfig/krb/kcd2_aes256.keytab"-domainUser "HTTP/lbvs.aaa.local"`
> - ユーザクレデンシャルを使用する場合は、次のコマンドを使用します。 `add kcdaccount`


```
kslb2_user -realmStr AAA.LOCAL -delegatedUser lbvs -kcdPassword <
password>
```

- 正しい **domainUser** 情報が提供されていることを確認します。AD でユーザーログオン名を検索できます。

KCD キータブスクリプトを生成します

October 7, 2021

「KCD キータブスクリプト」ダイアログボックスでは、キータブスクリプトが生成されます。キータブスクリプトは、Citrix ADC で KCD を構成するために必要なキータブファイルを生成します。

構成ユーティリティを使用して **KCD** キータブスクリプトを生成するには

1. [セキュリティ] **> [AAA-**アプリケーショントラフィック**]** に移動します。
2. 詳細ペインの [**Kerberos** 制約付き委任] で、[バッチファイル] をクリックして、キータブを生成します。
3. 「KCD (Kerberos 制約付き委任) キータブスクリプトの生成」ダイアログボックスで、以下の説明に従ってフィールドに入力します。
   - ドメインユーザー名**:** ドメインユーザーの名前。
   - ドメインパスワード**:** ドメインユーザーのパスワード。
   - サービス主体**:** サービス主体。
   - 出力ファイル名**:** KCD スクリプトファイルのファイル名。
   - **[**ドメインユーザーアカウントの作成**]:** 指定したドメインユーザーアカウントを作成するには、このチェックボックスをオンにします。
4. [スクリプトの生成] をクリックして、スクリプトを生成します。スクリプトが生成され、**[**スクリプト ****** を生成**]** ボタンの下にある **[**生成されたスクリプト] テキストボックスに表示されます。**
5. スクリプトをコピーし、AD ドメイン Controller にファイルとして保存します。このスクリプトをドメイン Controller で実行してキータブファイルを生成し、キータブファイルを Citrix ADC アプライアンスの /nsconfig/krb/ ディレクトリにコピーする必要があります。
6. 「**OK**」をクリックします。

基本認証、ダイジェスト認証、および **NTLM** 認証用の **SSO**

October 7, 2021

Citrix ADC および Citrix Gateway のシングルサインオン（SSO）構成は、グローバルレベルおよびトラフィックレベルごとに有効にできます。デフォルトでは、SSO 構成は オフであり、管理者はトラフィックごとまたはグローバルに SSO を有効にできます。セキュリティの観点から、Citrix は、管理者が SSO をグローバルに オフにし、トラフィ

ックごとに有効にすることをお勧めします。この機能拡張は、特定のタイプの SSO メソッドをグローバルに不名誉にすることにより、SSO 構成をより安全にすることです。

> 注:
>
> Citrix ADC 機能リリース 13.0 ビルド 64.35 以降では、次の SSO タイプがグローバルに不渡りになっています。
>
> - 基本認証
> - ダイジェストアクセス認証
> - NTLM2 キーのネゴシエーションまたはサインのネゴシエーションなしの NTLM

影響を受けない **SSO** タイプ

次の SSO タイプは、この拡張機能の影響を受けません。

- Kerberos 認証
- SAML 認証
- フォームベースの認証
- OAuth ベアラ認証
- NTLM2 キーまたはサインのネゴシエートを伴う NTLM

影響を受ける **SSO** 構成

以下は、影響を受ける（不名誉な）SSO 構成です。

グローバル構成

```
set tmsessionparam -SSO ON
set vpnparameter -SSO ON
add tmsessionaction tm_act -SSO ON
add vpn sessionaction tm_act -SSO ON
Per traffic configurations
add vpn trafficaction tf_act http -SSO ON
add tm trafficaction tf_act -SSO ON
<!--NeedCopy-->
```

トラフィック構成ごと

```
add vpn trafficaction tf_act http -SSO ON
add tm trafficaction tf_act -SSO ON
<!--NeedCopy-->
```


> あなたはできる enable/disable SSO 全体であり、個々の SSO タイプを変更することはできません。

適用されるセキュリティ対策

セキュリティ対策の一環として、セキュリティに敏感な SSO タイプはグローバル構成で不名誉になりますが、トラフィックアクション構成でのみ許可されます。
したがって、バックエンドサーバーが NTLM2 キーのネゴシエーションまたはサインのネゴシエーションなしで Basic、Digest、または NTLM を予期している場合、管理者は次の構成を介してのみ SSO を許可できます。

トラフィックアクション

```
add vpn trafficaction tf_act  http -SSO ON
add tm trafficaction tf_act -SSO ON
<!--NeedCopy-->
```

交通政策

```
add tm trafficpolicy <name> <rule> tf_act
add vpn trafficpolicy <name> <rule> tf-act
<!--NeedCopy-->
```

> 管理者は、信頼できるバックエンドサーバーに対してのみ SSO が有効になっていることを確認するために、トラフィックポリシーに対して適切なルールを構成する必要があります。

AAA-TM

グローバル構成に基づくシナリオ:

```
set tmsessionparam -SSO ON
<!--NeedCopy-->
```

回避策:

```
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol true tf_act
<!--NeedCopy-->
```


次のトラフィックポリシーを、**SSO** が必要なすべての **LB** 仮想サーバーにバインドします。

```
bind lb vserver <LB VS Name> -policy tf_pol -priority 65345
<!--NeedCopy-->
```

セッションポリシー構成に基づくシナリオ：

```
add tmsessionaction tm_act -SSO ON
add tmsession policy <name> <rule> tm_act
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol <same rule as session Policy> tf_act
<!--NeedCopy-->
```

注意点：

- Citrix ADC AAA user/group 前のセッションのポリシーは、トラフィックポリシーに置き換える必要があります。
- 次のポリシーを、前のセッションポリシーの負荷分散仮想サーバーにバインドします。

```
bind lb vserver [LB VS Name] -policy tf_pol -priority 65345
<!--NeedCopy-->
```

- 他の優先度のトラフィックポリシーが設定されている場合、上記のコマンドは適切に機能しません。

次のセクションでは、トラフィックに関連付けられた複数のトラフィックポリシーとの競合に基づくシナリオについて説明します。

特定の TM トラフィックには、1 つの TM トラフィックポリシーのみが適用されます。SSO 機能の変更はグローバルに設定されているため、優先度の高い（SSO 構成が不要な）TM トラフィックポリシーがすでに適用されている場合は、優先度の低い追加の TM トラフィックポリシーを適用できない場合があります。次のセクションでは、このようなケースを確実に処理する方法について説明します。

優先度の高い次の **3** つのトラフィックポリシーが負荷分散（**LB**）仮想サーバーに適用されることを考慮してください。

```
add tm trafficaction tf_act1 <Addition config>
add tm trafficaction tf_act2 <Addition config>
add tm trafficaction tf_act3 <Addition config>

add tm trafficpolicy tf_pol1 <rule1> tf_act1
```



```
6  add tm trafficpolicy tf_pol2 <rule2> tf_act2
7  add tm trafficpolicy tf_pol3 <rule3> tf_act3
8
9  bind lb vserver <LB VS Name> -policy tf_pol1 -priority 100
10 bind lb vserver <LB VS Name> -policy tf_pol2 -priority 200
11 bind lb vserver <LB VS Name> -policy tf_pol3 -priority 300
12 <!--NeedCopy-->
```

エラーが発生しやすい方法-グローバル **SSO** 構成を解決するには、次の構成を追加します。

```
1  add tm trafficaction tf_act_default -SSO ON
2  add tm trafficpolicy tf_pol_default true tf_act_default
3
4  bind lb vserver <LB VS Name> -policy tf_pol_default -priority 65345
5  <!--NeedCopy-->
```

注**:** 上記の変更により、<tf_pol1/tf_pol2/tf_pol3> トラフィックポリシーとしてヒットするトラフィックの SSO が壊れる可能性があります。<tf_pol_default> は適用されません。

正しい方法-これを軽減するには、対応するトラフィックアクションごとに **SSO** プロパティを個別に適用する必要があります。

たとえば、前述のシナリオでは、tf_pol1/tf_pol3 に到達するトラフィックに対して SSO が発生するには、次の設定を <tf_pol_default>。

```
1  add tm trafficaction tf_act1 <Addition config> -SSO ON
2  add tm trafficaction tf_act3 <Addition config> -SSO ON
3  <!--NeedCopy-->
```

Citrix Gateway のケース

グローバル構成に基づくシナリオ:

```
1  set vpnparameter -SSO ON
2  <!--NeedCopy-->
```

回避策:


```
add vpn trafficaction vpn_tf_act http  -SSO ON
add vpn trafficpolicy vpn_tf_pol  true vpn_tf_act
bind the following traffic policy to all VPN virtual server where SSO
   is expected:
bind vpn vserver vpn_vs -policy vpn_tf_pol -priority 65345
<!--NeedCopy-->
```

セッションポリシー構成に基づくシナリオ:

```
add vpn sessionaction vpn_sess_act -SSO ON
add vpnsession policy <name> <rule> vpn_sess_act
<!--NeedCopy-->
```

注意事項**:**

- Citrix ADC AAA user/group 前のセッションのポリシーは、トラフィックポリシーに置き換える必要があります。
- 次のポリシーを、前のセッションポリシー `bind lb virtual server [LB VS Name] -policy tf_pol -priority` 65345の LB 仮想サーバーにバインドします。
- 他の優先度のトラフィックポリシーが設定されている場合、上記のコマンドは適切に機能しません。次のセクションでは、トラフィックに関連付けられた複数のトラフィックポリシーとの競合に基づくシナリオについて説明します。

トラフィックに関連付けられた複数のトラフィックポリシーとの競合に基づく機能シナリオ:

特定の Citrix Gateway トラフィックには、1 つの VPN トラフィックポリシーのみが適用されます。SSO 機能のグローバル設定が変更されたため、必要な SSO 構成がない優先度の高い VPN トラフィックポリシーが他にある場合、優先度の低い追加の VPN トラフィックポリシーを適用できない場合があります。

次のセクションでは、このようなケースを確実に処理する方法について説明します。

VPN 仮想サーバーに適用される優先度の高い 3 つのトラフィックポリシーがあると考えてください。

```
add vpn trafficaction tf_act1 <Addition config>
add vpn trafficaction tf_act2 <Addition config>
add vpn trafficaction tf_act3 <Addition config>

add vpn trafficpolicy tf_pol1 <rule1> tf_act1
add vpn trafficpolicy tf_pol2 <rule2> tf_act2
add vpn trafficpolicy tf_pol3 <rule3> tf_act3

bind vpn vserver <VPN VS Name> -policy tf_pol1 -priority 100
```



```
bind vpn vserver <VPN VS Name> -policy tf_pol2 -priority 200
bind vpn vserver <VPN VS Name> -policy tf_pol3 -priority 300
<!--NeedCopy-->
```

エラーが発生しやすい方法：グローバル SSO 構成を解決するには、次の構成を追加します。

```
add vpn trafficaction tf_act_default -SSO ON
add vpn trafficpolicy tf_pol_default true tf_act_default

bind vpn vserver <VPN VS Name> -policy tf_pol_default -priority 65345
<!--NeedCopy-->
```

注**:** 上記の変更により、<tf_pol1/tf_pol2/tf_pol3> トラフィックポリシーとしてヒットするトラフィックの SSO が壊れる可能性があります。<tf_pol_default> は適用されません。

正しい方法：これを軽減するには、対応するトラフィックアクションごとに SSO プロパティを個別に適用する必要があります。

たとえば、前述のシナリオでは、tf_pol1/tf_pol3 に到達するトラフィックに対して SSO が発生するには、次の設定を <tf_pol_default>。

```
add vpn trafficaction tf_act1 [Additional config] -SSO ON

add vpn trafficaction tf_act3 [Additional config] -SSO ON
<!--NeedCopy-->
```

Citrix Gateway および認証サーバーで生成されたレスポンスの書き換え

October 7, 2021

書き換えとは、Citrix ADC アプライアンスが処理する要求または応答の情報を書き換えることです。書き換えは、ウェブサイトの実際の設定に関する不必要な詳細を公開することなく、要求されたコンテンツへのアクセスを提供するのに役立ちます。書き換えの概念の詳細については、「書き換え」を参照してください。

Citrix ADC リリースビルド 13.0-76.29 から、書き換えポリシーのサポートが Citrix Gateway 仮想サーバーおよび認証仮想サーバー生成応答に拡張されました。

> 注:

> Citrix Gateway 仮想サーバーおよび認証仮想サーバーで生成されたレスポンスの書き換えポリシーをサポートするために、バインドタイプ **AAA_Response** が導入されます。

書き換えを使用する例

書き換えを使用して、オンプレミスの Citrix ADC で利用可能なリソースを Citrix Cloud 展開と共有できます。これは、CORS オリジンリソース共有を実装することで安全に実現できます。書き換えは、CORS ヘッダーを実装するために以下のように使用することができる。

構成例

```
add rewrite action cors_header_action insert_http_header access-control
    -allow-credentials \”true\”

add rewrite policy cors_header_pol true cors_header_action

add rewrite action non_cors_header_action insert_http_header X-Frame-
    Options \'\”DENY\”\'

add rewrite policy non_cors_header_pol true non_cors_header_action

bind authentication vserver av_cors -policy cors_header_pol -priority
    100 -type AAA_RESPONSE

bind vpn vserver av_cors -policy cors_header_pol -priority 100 -type
    AAA_RESPONSE
```

Citrix Gateway および認証仮想サーバーで生成された応答に対するコンテンツセキュリティポリシーの応答ヘッダーのサポート

July 15, 2022

Citrix ADC リリースビルド 13.0～76.29 以降、コンテンツセキュリティポリシー（CSP）応答ヘッダーは、Citrix Gateway および認証仮想サーバーで生成された応答でサポートされています。

Content-Security-Policy（CSP）レスポンスヘッダーは、クロスサイトスクリプティング（CSS）攻撃を回避するためにブラウザが使用するポリシーの組み合わせです。
HTTP CSP レスポンスヘッダーを使用すると、Web サイト管理者は、ユーザーエージェントが特定のページに対してロードできるリソースを制御できます。いくつかの例外を除いて、ポリシーには主にサーバーオリジンとスクリプトエンドポイントの指定が含まれます。これにより、クロスサイトスクリプティング攻撃から保護できます。

CSP ヘッダーは、ブラウザーがページをレンダリングする方法を変更し、CSS を含むさまざまなクロスサイトインジェクションから保護するように設計されています。ウェブサイトの適切な操作を妨げないように、ヘッダー値を正しく設定することが重要です。たとえば、ヘッダーがインライン JavaScript の実行を妨げるように設定されている場合、Web サイトはそのページでインライン JavaScript を使用してはいけません。

CSP レスポンスヘッダーの利点は次のとおりです。

- CSP 応答ヘッダーの主な機能は、CSS 攻撃を防ぐことです。
- サーバーでは、コンテンツのロード元となるドメインを制限することに加えて、どのプロトコルを使用できるかを指定できます。たとえば（理想的にはセキュリティの観点から）、サーバーはすべてのコンテンツを HTTPS を使用してロードする必要があることを指定できます。
- CSP は、「tmindex.html」や「homepage.html」などのファイルを保護することで、クロスサイトスクリプティング攻撃から Citrix ADC を保護するのに役立ちます。ファイル “tmindex.html” は認証に関連しており、ファイル “homepage.html” は公開されたアプリ/リンクに関連しています。

Citrix Gateway のコンテンツセキュリティポリシーヘッダーと認証仮想サーバーが生成する応答の構成

CSP ヘッダーを有効にするには、CSP HTTP ヘッダーを返すように Web サーバーを設定する必要があります。

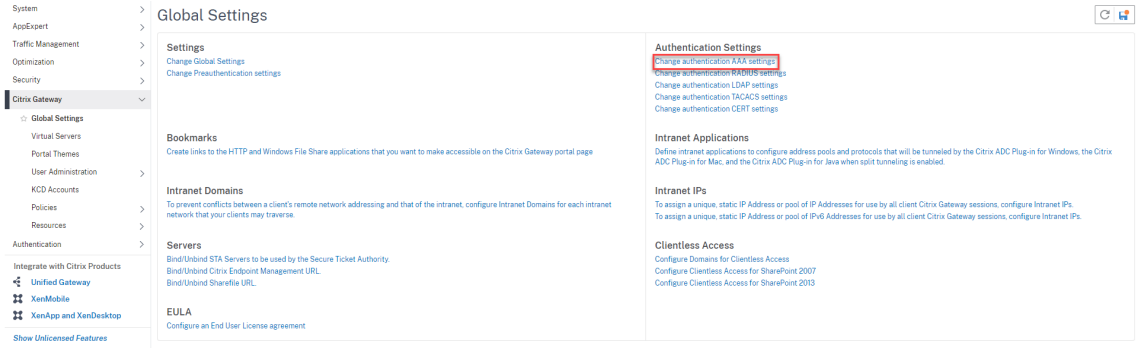
> 注意事項
>
> - デフォルトでは、CSP ヘッダーは無効になっています。
> - デフォルトの CSP ポリシーを有効または無効にする場合は、次のコマンドを実行することをお勧めします。`Flush cache contentgroup loginstaticobjects`
> - /logon/LogonPoint/index.html の CSP を変更するには、ディレクトリ `/var/netscaler/logon` の下にあるログオンディレクトリに対応するセクションで、必要に応じて「ヘッダーセット Content-Security-Policy」の値を変更します。

CLI を使用して認証仮想サーバーおよび Citrix Gateway が生成する応答用に CSP を構成するには、コマンドプロンプトで次のコマンドを入力します。

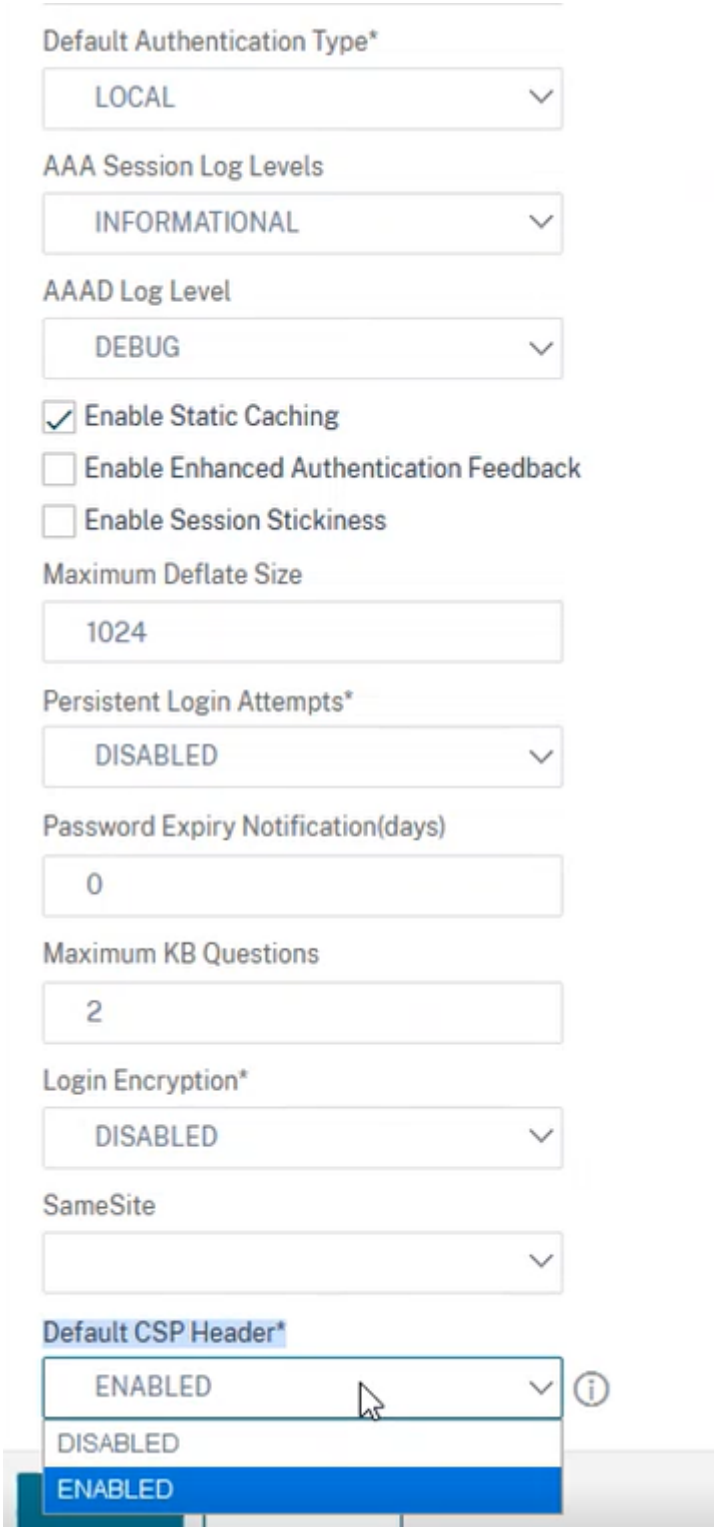
```
1 set aaa parameter -defaultCSPHeader <ENABLE/DISABLE>
```

GUI を使用して、Citrix Gateway と認証仮想サーバーが生成する応答用に CSP を構成します。

1. **Citrix Gateway >** グローバル設定に移動し、［認証設定］の下の［認証 **AAA** 設定の変更］をクリックします。



2. [**AAA** パラメータの設定] ページで、[デフォルト **CSP** ヘッダーで有効] フィールドを選択します。



Content-Security-Policy ヘッダーのカスタマイズの例

次に、CSP ヘッダーのカスタマイズで、次の 2 つの指定されたソースからのイメージとスクリプトのみをそれぞれ含める例を示します。`https://company.fqdn.com, https://example.com.`。

設定例

```
add rewrite action modify_csp insert_http_header Content-Security-
    Policy "\"default-src 'self'; script-src 'self' https://company.fqdn
    .com 'unsafe-inline' 'unsafe-eval'; connect-src 'self'; img-src http
    ://localhost:* https://example.com 'self' data: http: https:; style-
    src 'self' 'unsafe-inline'; font-src 'self'; frame-src 'self'; child
    -src 'self' com.citrix.agmacepa://* citrixng://* com.citrix.
    nsgclient://*; form-action 'self'; object-src 'self'; report-uri /
    nscsp_violation/report_uri\""

add rewrite policy add_csp true modify_csp

bind authentication vserver auth1 -policy add_csp -priority 1 -
    gotoPriorityExpression NEXT -type AAA_RESPONSE
```

セルフサービスパスワードリセット

September 27, 2022

セルフサービスパスワードリセットは、Web ベースのパスワード管理ソリューションです。これは、Citrix ADC アプライアンスと Citrix Gateway の認証、承認、および監査機能の両方で使用できます。これにより、ユーザーはパスワードの変更に対する管理者の支援に依存する必要がなくなります。

セルフサービスパスワードリセットにより、エンドユーザは次のシナリオでパスワードを安全にリセットまたは作成できます。

- ユーザーがパスワードを忘れた。
- ユーザーはログオンできません。

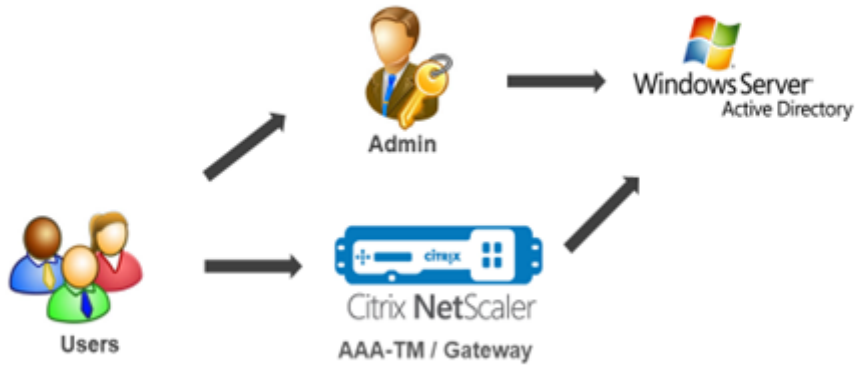
これまでは、エンドユーザーが AD パスワードを忘れた場合、エンドユーザーは AD 管理者に連絡してパスワードをリセットする必要がありました。セルフサービスパスワードリセット機能により、エンドユーザーは管理者の介入なしにパスワードをリセットできます。

セルフサービスパスワードリセットを使用する利点には、次のようなものがあります。

- パスワードの自動変更メカニズムにより生産性が向上し、ユーザーがパスワードのリセットを待つ時間がなくなります。

- パスワードの自動変更メカニズムにより、管理者は他の重要なタスクに集中できます。

次の図は、パスワードをリセットするセルフサービスパスワードリセットフローを示しています。



セルフサービスパスワードリセットを使用するには、Citrix 認証、承認、監査、または Citrix Gateway 仮想サーバーのいずれかにユーザーが登録されている必要があります。

セルフサービスパスワードリセットには、次の機能があります。

- 新規ユーザーの自己登録。新規ユーザーとして自己登録できます。
- ナレッジベースの質問を構成します。管理者は、ユーザに対して一連の質問を設定できます。
- 代替電子メール **ID** 登録。登録時に別の電子メール ID を指定する必要があります。ユーザーがプライマリ電子メール ID のパスワードを忘れたため、OTP は代替電子メール ID に送信されます。

  > 注:
  >
  > バージョン 12.1 ビルド 51.xx から、代替電子メール ID の登録をスタンドアロンとして行うことができます。代替の電子メール ID 登録のみを行うために、新しいログインスキーマ **AltEmailRegister.xml** が導入されました。以前は、代替電子メール ID の登録は、KBA 登録の実行中にのみ実行できました。

- 忘れたパスワードをリセットします。ユーザーはナレッジベースの質問に答えることでパスワードをリセットできます。管理者は、質問を設定して保存できます。

セルフサービスパスワードリセットでは、次の 2 つの新しい認証メカニズムが提供されます。

- 知識ベースの質問と回答。ナレッジベースの質問と回答のスキーマを選択する前に、Citrix 認証、承認、監査、または Citrix Gateway に登録する必要があります。
- 電子メール **OTP** 認証。OTP は、ユーザーがセルフサービスパスワードリセット登録時に登録した代替電子メール ID に送信されます。

> 注
>
> これらの認証メカニズムは、セルフサービスパスワードリセットのユースケースや、既存の認証メカニズムと同様の認証目的に使用できます。

前提条件

セルフサービスパスワードリセットを構成する前に、次の前提条件を確認してください。

- Citrix ADC 機能リリース 12.1、ビルド 50.28。
- サポートされているバージョンは 2016、2012、および 2008 の AD ドメイン機能レベルです。
- Citrix ADC にバインドされた LDAPBind ユーザー名には、ユーザーの AD パスへの書き込みアクセス権が必要です。

  > 注
  >
  > セルフサービスパスワードリセットは、nFactor 認証フローでのみサポートされます。詳細については、Citrix ADC による nFactor 認証を参照してください。

制限事項

セルフサービスパスワードのリセットには、次のような制限があります。

- セルフサービスパスワードリセットは、LDAPS でサポートされています。セルフサービスパスワードリセットは、認証バックエンドが LDAP (LDAP プロトコル) の場合にのみ使用できます。
- 登録済みの代替電子メール ID がユーザーに表示されることはありません。
- ナレッジベースの質疑応答、電子メール OTP の認証と登録は、認証フローの最初の要素にはなりません。
- ネイティブプラグインおよび Receiver では、登録はブラウザ経由でのみサポートされます。
- セルフサービスパスワードのリセットに使用される証明書の最小サイズは 1024 バイトで、x.509 標準に従う必要があります。
- セルフサービスパスワードリセットでは RSA 証明書のみがサポートされています。

アクティブディレクトリ設定

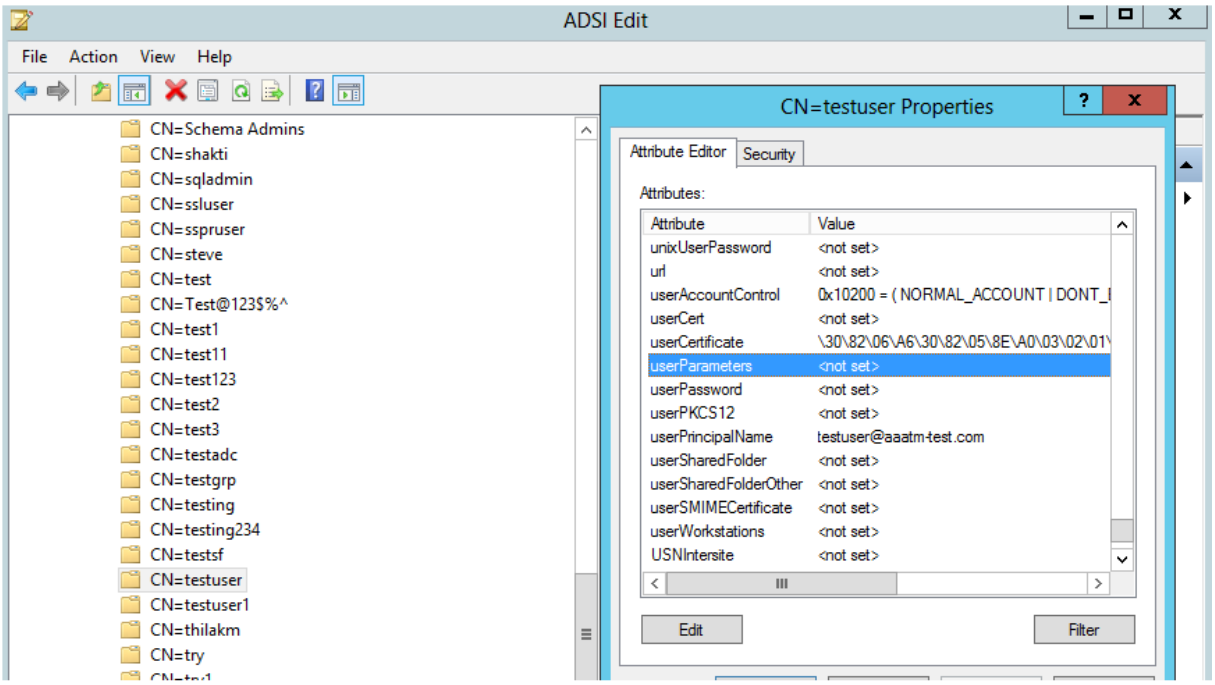
Citrix ADC ナレッジベースの質問と回答、および電子メールの OTP では、AD 属性を使用してユーザーデータを保存します。質問と回答を代替電子メール ID とともに保存するように AD 属性を設定する必要があります。Citrix ADC アプライアンスは、AD ユーザーオブジェクトの構成済みの KB 属性にこれを保存します。AD 属性を設定する場合は、次の点を考慮してください。

- 属性の長さは 128 文字以上にする必要があります。
- AD 属性は最大 32K の値をサポートする必要があります。
- 属性タイプは 'DirectoryString' でなければなりません。
- 1 つの AD 属性を、ナレッジベースの質問と回答、および代替電子メール ID に使用できます。

- ネイティブ OTP およびナレッジベースの質問と回答、または代替電子メール ID の登録には、単一の AD 属性を使用できません。
- Citrix ADC LDAP 管理者は、選択した AD 属性への書き込みアクセス権を持っている必要があります。

既存の AD 属性を使用することもできます。ただし、使用する予定の属性が他のケースでは使用されないようにしてください。たとえば、userParameters は AD ユーザー内の既存の属性であり、これを使用できます。この属性を確認するには、次の手順を実行します。

1. [**ADSI] > [**ユーザーの選択**]** に移動します。
2. 右クリックして、属性リストまでスクロールダウンします。
3. **cn=TestUser** プロパティウィンドウペインで、**UserParameters** 属性が設定されていないことを確認できます。



セルフサービスパスワードリセット登録

Citrix ADC アプライアンスにセルフサービスパスワードリセットソリューションを実装するには、次の手順を実行する必要があります。

- セルフサービスパスワードリセット (ナレッジベースの質問と回答/電子メール ID) 登録
- ユーザーログオンページ (パスワードのリセット用。ナレッジベースの質問と回答、電子メールの OTP 検証、最終的なパスワードリセット係数が含まれます)。

定義済みの質問カタログのセットが JSON ファイルとして提供されます。管理者は、Citrix ADC GUI を使用して質問を選択し、セルフサービスパスワードリセット登録ログインスキーマを作成できます。次のオプションのいずれかを選択できます。

- システム定義の質問を最大 4 つ選択します。

- ユーザーが 2 つの質問と回答をカスタマイズできるオプションを提供します。

CLI からデフォルトのナレッジベースの質問 **JSON** ファイルを表示するには

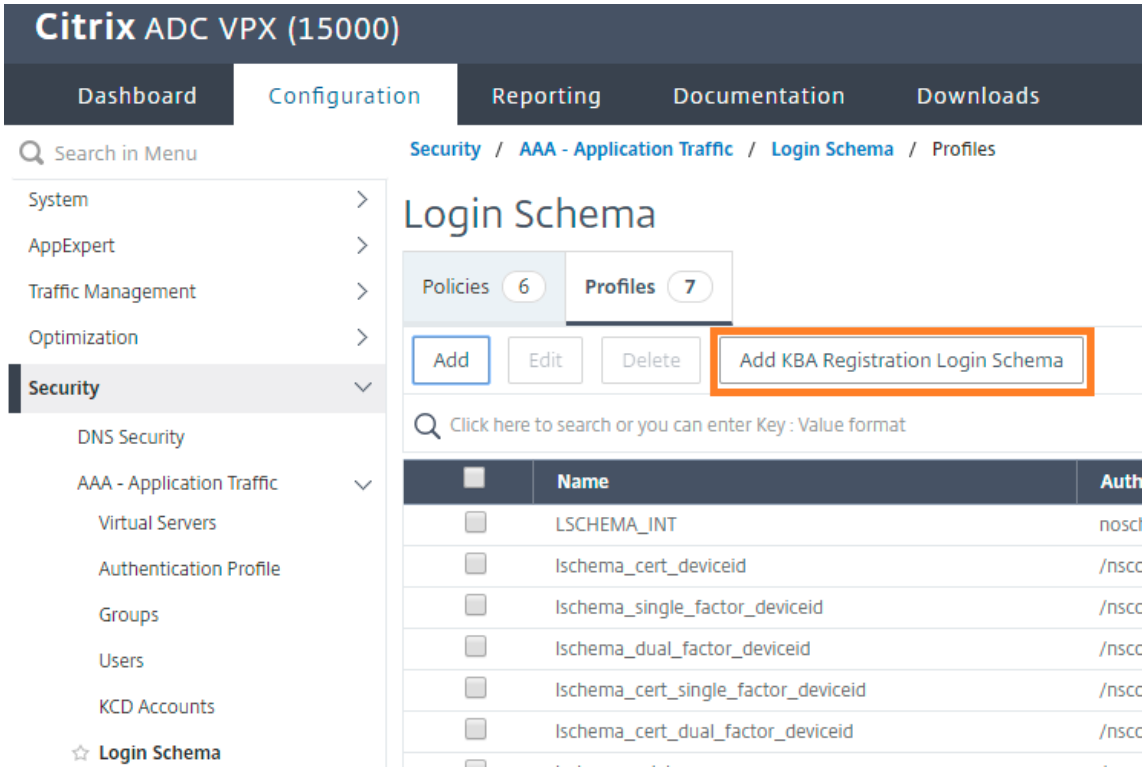
```
root@ns# cd /nsconfig/loginschema/LoginSchema/
root@ns# cat KBQuestions.json
[
    {"question":"What is the last name of the teacher who gave you your first failing
grade?"},
    {"question":"What is the name of your favourite childhood friend?"},
    {"question":"Where were you when you first heard about 9/11?"},
    {"question":"What is the name of a college you applied to but didn't attend?"},
    {"question":"What was the last name of your third grade teacher?"},
    {"question":"What was the name of your first stuffed animal?"},
    {"question":"What is the name of the teacher who gave you your first A?"},
    {"question":"What is the name of the city where you got lost?"},
    {"question":"In what city or town did your mother and father meet?"},
    {"question":"What was your most hated food as a child?"},
    {"question":"What was your most favourite food as a child?"},
    {"question":"What is your favourite website?"},
    {"question":"What is your most disliked website?"},
    {"question":"What is your dream job?"},
    {"question":"Why did the chicken cross the road?"},
    {"question":"Name your first boss."},
    {"question":"What is the name of your favorite school teacher?"},
    {"question":"What is the name of your favorite actor or actress?"},
    {"question":"What is the title of your favorite movie?"},
    {"question":"In what city or town did you spend most of your youth?"}
]
```

注

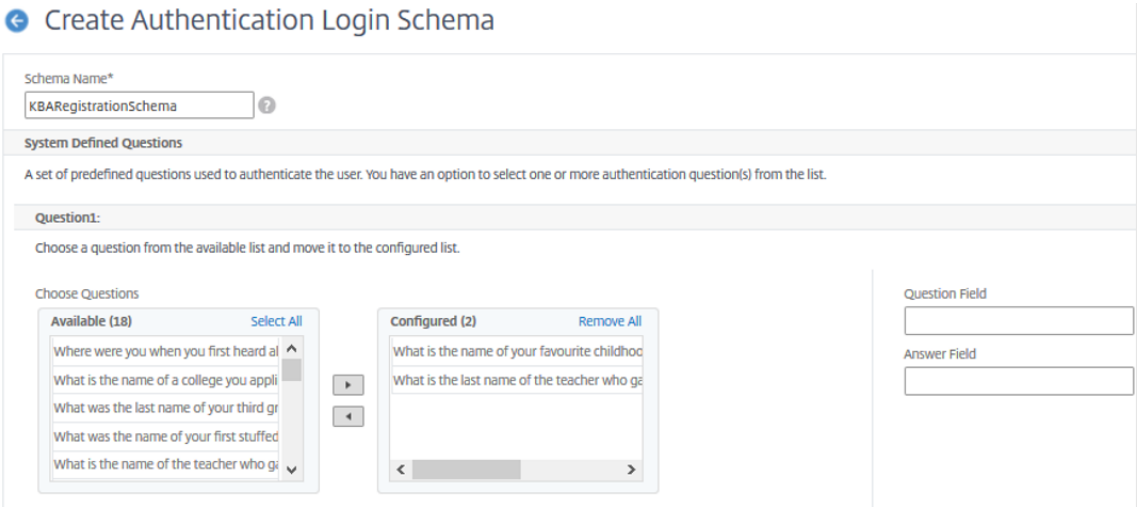
- Citrix Gateway には、デフォルトでシステム定義の質問のセットが含まれています。管理者は「KbQuestions.json」ファイルを編集して、選択した質問を含めることができます。
- システム定義の質問は英語でのみ表示され、これらの質問では言語ローカリゼーションのサポートは利用できません。

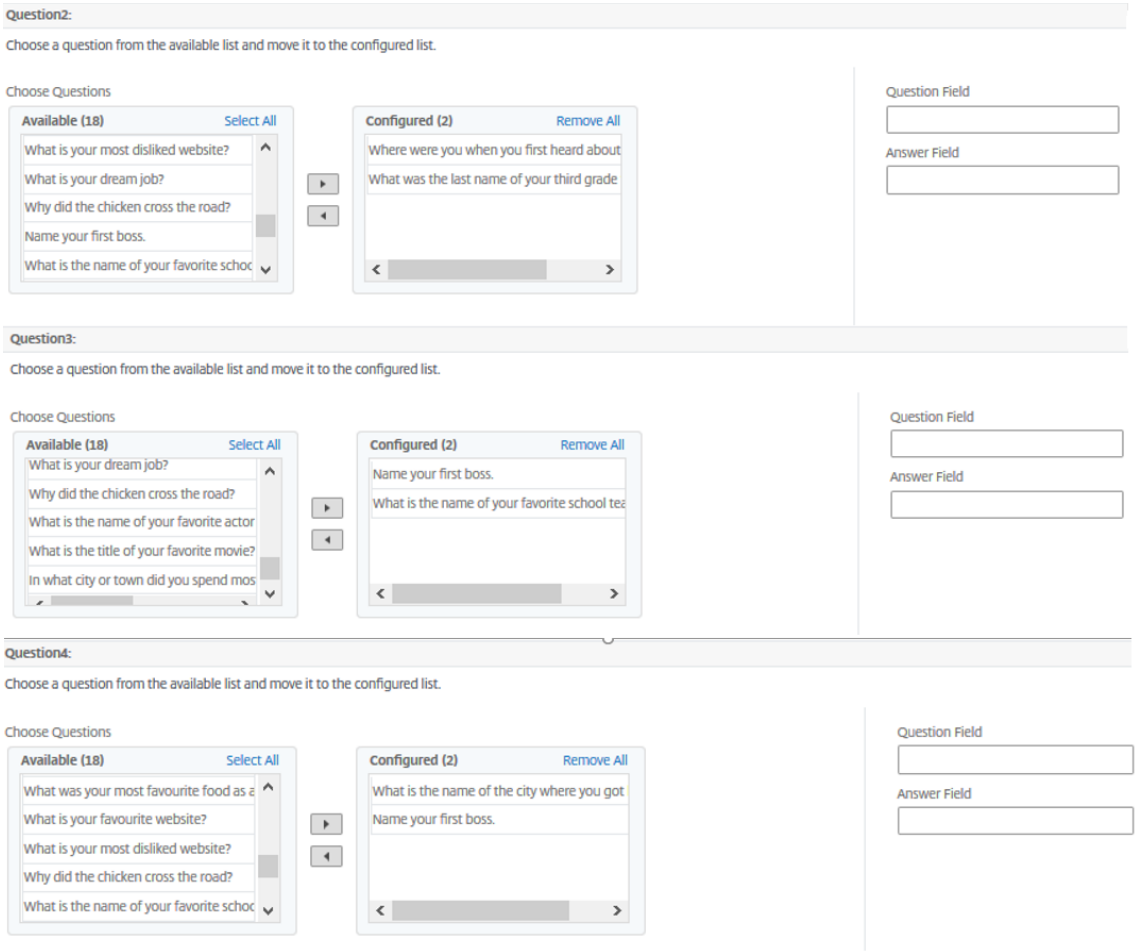
GUI を使用してナレッジベースの質問と回答の登録ログインスキーマを完了するには

1. セキュリティ **> AAA ―** アプリケーショントラフィック **>** ログインスキーマに移動します。

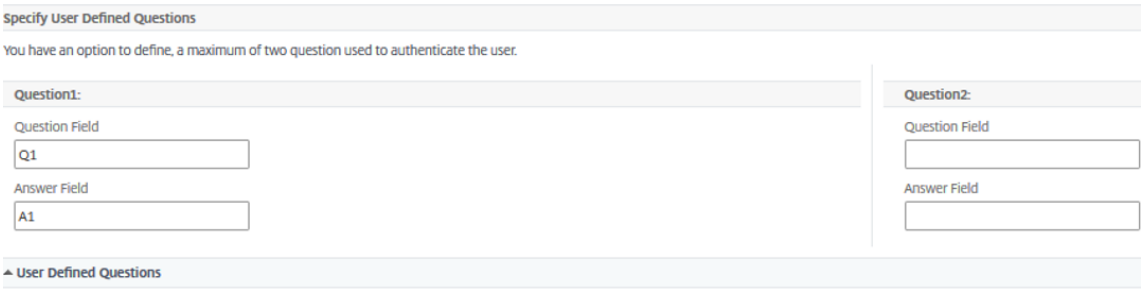


2. [ログインスキーマ] ページで、[プロファイル] をクリックします。
3. [**KBA** 登録ログインスキーマの追加] をクリックします。
4. **[**認証ログインスキーマの作成] ページで、[スキーマ名**]** フィールドに名前を指定します。

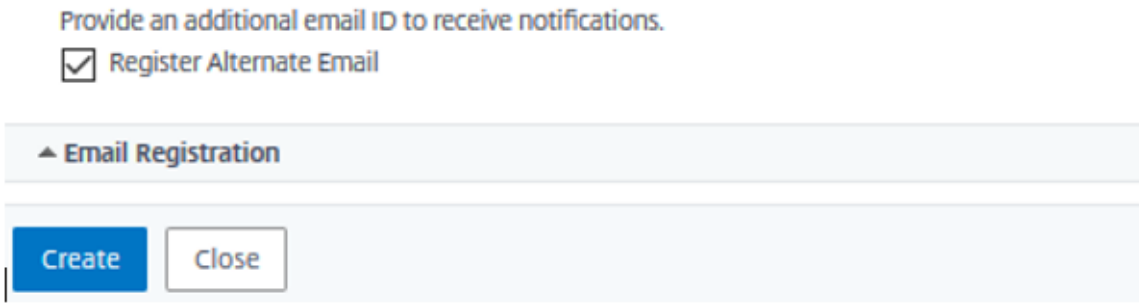




5. 任意の質問を選択し、[設定済み] リストに移動します。
6. [ユーザ定義の質問（**User Defined** Questions）] セクションでは、Q1 および A1 フィールドに質問と回答を入力できます。



7. [メール登録] セクションで、[代替メールの登録] オプションをオンにします。OTP を受け取るには、ユーザー登録ログオンページから代替電子メール **ID** を登録します。



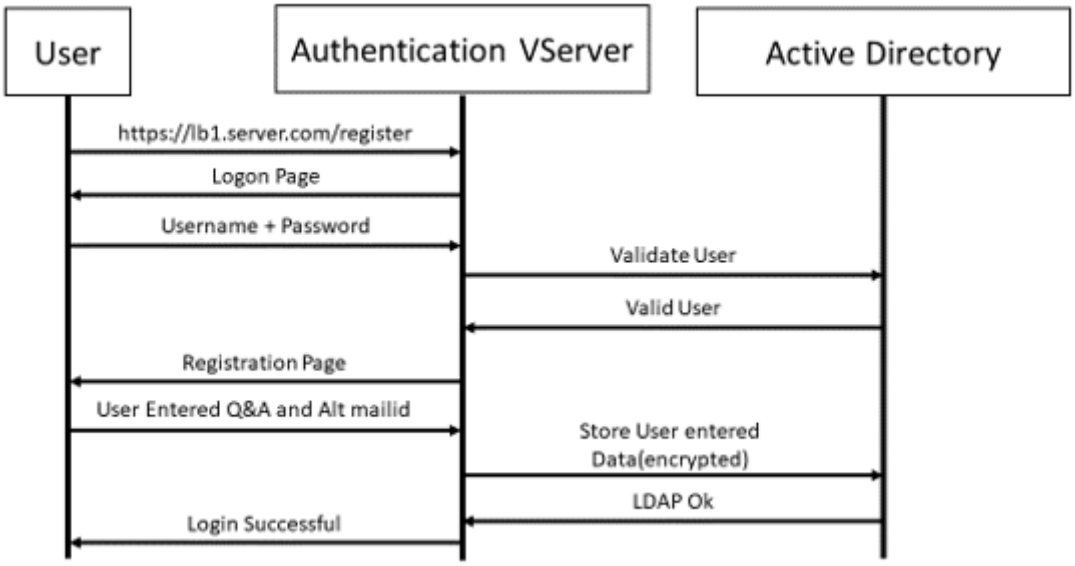
8. ［**Create**］をクリックします。ログインスキーマが生成されると、登録プロセス中に設定されたすべての質問がエンドユーザに表示されます。

CLI を使用してユーザー登録と管理のワークフローを作成する

設定を開始する前に、次のことが必要です。

- 認証仮想サーバに割り当てられた IP アドレス
- 割り当てられた IP アドレスに対応する FQDN
- 認証仮想サーバーのサーバー証明書

デバイスの登録と管理ページを設定するには、認証仮想サーバーが必要です。次の図に、ユーザー登録を示します。



認証仮想サーバーを作成するには

1. 認証仮想サーバを設定します。SSL タイプで、認証仮想サーバーをポータル・テーマにバインドする必要があります。


```
> add authentication vserver <vServerName> SSL <ipaddress> <port>
> bind authentication vserver <vServerName> [-portaltheme<string>]
```

2. SSL 仮想サーバ証明書とキーのペアをバインドします。

```
> bind ssl vserver <vServerName> certkeyName <string>
```

例:

```
> add authentication vserver authvs SSL 1.2.3.4 443
> bind authentication vserver authvs -portaltheme RFWebUI
> bind ssl vserver authvs -certkeyname c1
```

LDAP ログオンアクションを作成するには

```
    > add authentication ldapAction <name> {
-serverIP <ipaddr|ipv6_addr|> [-serverPort <port>] [-ldapBase <BASE> ]
     [-ldapBindDn <AD USER>] [-ldapBindDnPassword <PASSWORD>] [-
    ldapLoginName <USER FORMAT>]
```

> 注
>
> 第 1 要素として任意の認証ポリシーを設定できます。

例:

```
    > add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4
      -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
     ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
     PASSWORD -ldapLoginName samAccountName -serverport 636 -sectype
     SSL -KBAttribute userParameters
```

LDAP ログオン用の認証ポリシーを作成するには

```
    > add authentication policy <name> <rule> [<reqAction]
```


例:

```
1      > add authentication policy ldap_logon -rule true -action
          ldap_logon_action
```

ナレッジベースの質問と回答の登録アクションを作成するには

ldapActionに 2 つの新しいパラメータが導入されました。KBA 認証 (登録と検証) 用のKBAttributeおよびユーザーの代替電子メール ID の登録用のalternateEmailAttr。

```
1      > add authentication ldapAction <name> {
2   -serverIP  <ipaddr|ipv6_addr|> [-serverPort <port>] [-ldapBase <BASE>
       ] [-ldapBindDn <AD USER>] [-ldapBindDnPassword <PASSWORD>] [-
       ldapLoginName <USER FORMAT>] [-KBAttribute <LDAP ATTRIBUTE>] [-
       alternateEmailAttr <LDAP ATTRIBUTE>]
```

例:

```
1      > add authentication ldapAction ldap1 -serverIP 1.2.3.4 -sectype
          ssl -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
          ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
          PASSWORD -ldapLoginName samAccountName -KBAttribute
          userParameters -alternateEmailAttr userParameters
```

ユーザー登録と管理画面を表示する

「KBARegistrationSchema.xml」ログインスキーマは、エンドユーザーにユーザー登録ページを表示するために使用されます。次の CLI を使用して、ログインスキーマを表示します。

```
1  > add authentication loginSchema <name> -authenticationSchema <string>
```

例:

```
1  > add authentication loginSchema kba_register -authenticationSchema /
       nsconfig/loginschema/LoginSchema/KBARegistrationSchema.xml
```

ユーザー登録と管理画面を表示するには、URL または LDAP 属性の 2 つの方法があります。

URL を使う

URL パスに '/register' (https://lb1.server.com/registerなど) が含まれている場合、URL を使用してユーザー登録ページが表示されます。

登録ポリシーを作成してバインドするには

```
> add authentication policylabel user_registration -loginSchema
   kba_register
> add authentication policy ldap1 -rule true -action ldap1
> bind authentication policylabel user_registration -policy ldap1 -
   priority 1
```

URL に **'/register'** が含まれている場合に、認証ポリシーを認証、承認、監査仮想サーバーにバインドするには

```
> add authentication policy ldap_logon -rule "http.req.cookie.value(\"
   NSC_TASS\").contains(\"register\")" -action ldap_logon
> bind authentication vserver authvs -policy ldap_logon -nextfactor
   user_registration -priority 1
```

証明書を **VPN** グローバルにバインドするには

```
bind vpn global -userDataEncryptionKey c1
```

> 注
>
> - AD 属性に保存されているユーザーデータ (KB Q&A および登録済みの代替電子メール ID) を暗号化するには、証明書をバインドする必要があります。
> - 証明書の有効期限が切れた場合は、新しい証明書をバインドして、登録をもう一度実行する必要があります。

属性の使用

認証ポリシーを認証、承認、および監査仮想サーバーにバインドして、ユーザーがすでに登録されているかどうかを確認できます。このフローでは、ナレッジベースの質問と回答の登録係数より前の前述のポリシーは、KBA 属性が設定された LDAP である必要があります。これは、AD ユーザーが登録されているか、AD 属性を使用していないかを確認するためです。

> 重要
>
> ルール”AAA.USER.ATTRIBUTE(“kba_registered”).EQ(“0”)” は、新しいユーザにナレッジベースの質問と回答、および代替の電子メールへの登録を強制します。

ユーザーがまだ登録されていないかどうかを確認する認証ポリシーを作成するには

```
> add authentication policy switch_to_kba_register -rule ”AAA.USER.ATTRIBUTE(\”kba_registered\”).EQ(\”0\”)” -action NO_AUTHN
> add authentication policy first_time_login_forced_kba_registration -rule true -action ldap1
```

登録ポリシーラベルを作成し、**LDAP** 登録ポリシーにバインドするには

```
> add authentication policylabel auth_or_switch_register -loginSchema LSCHEMA_INT
> add authentication policylabel kba_registration -loginSchema kba_register

> bind authentication policylabel auth_or_switch_register -policy switch_to_kba_register -priority 1 -nextFactor kba_registration
> bind authentication policylabel kba_registration -policy first_time_login_forced_kba_registration -priority 1
```

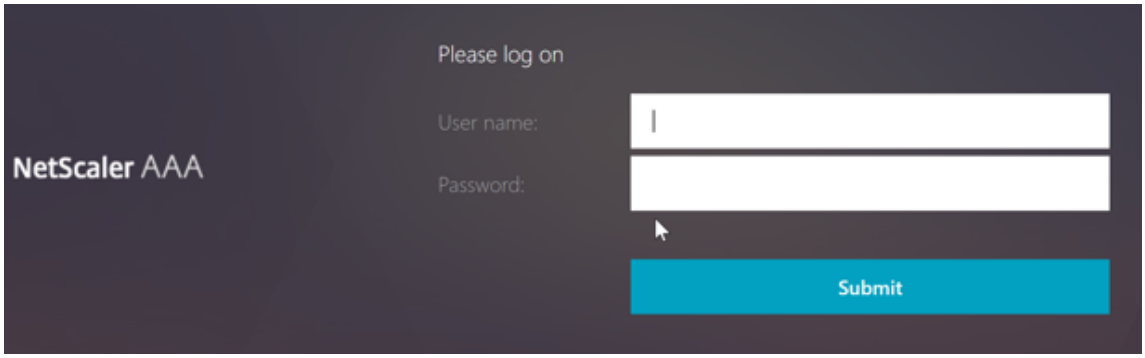
認証ポリシーを認証、承認、監査仮想サーバーにバインドするには

```
bind authentication vserver authvs -policy ldap_logon -nextfactor auth_or_switch_register -priority 2
```

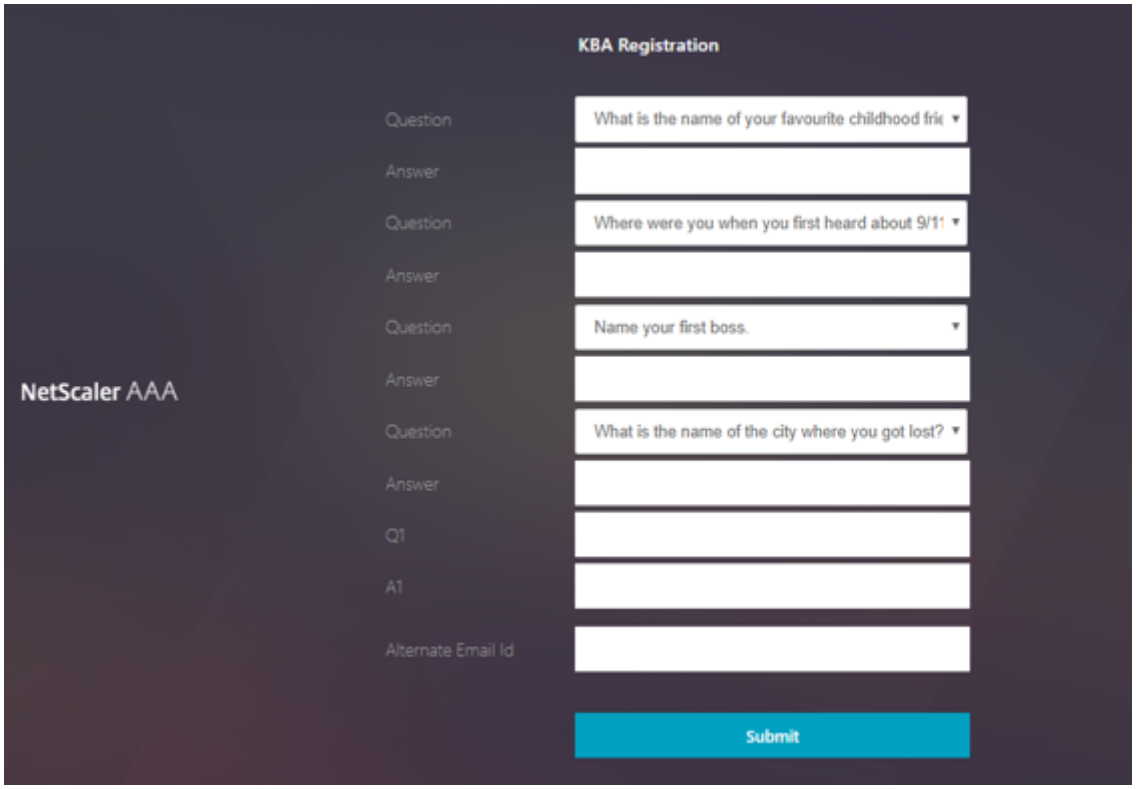
ユーザー登録と管理の検証

前のセクションで説明した手順をすべて設定したら、次の UI 画面が表示されます。

1. ロードバランサー仮想サーバの URL を入力します（例:https://lb1.server.com）。ログオン画面が表示されます。



2. ユーザ名とパスワードを入力します。[**Submit**] をクリックします。[ユーザー登録] 画面が表示されます。



3. ドロップダウンリストから希望する質問を選択し、[**Answer**] を入力します。
4. [**Submit**] をクリックします。ユーザー登録成功画面が表示されます。

ユーザーログオンの設定ページ

この例では、管理者は最初の要素が LDAP ログオン (エンドユーザーがパスワードを忘れた) であると想定しています。その後、ユーザーはナレッジベースの質問と回答の登録と電子メール ID OTP 検証に従い、最後にセルフサービスパスワードリセットを使用してパスワードをリセットします。

セルフサービスパスワードリセットには、どの認証メカニズムでも使用できます。Citrix では、強力なプライバシーを確保し、不正なユーザーパスワードのリセットを避けるために、ナレッジベースの質問と回答、または OTP のメール、あるいはその両方を行うことをお勧めします。

ユーザーログオンページの設定を開始する前に、次のことが必要です。

- ロードバランサ仮想サーバの IP
- ロードバランサ仮想サーバに対応する FQDN
- ロードバランサーのサーバー証明書

CLI を使用してロードバランサ仮想サーバを作成する

内部 Web サイトにアクセスするには、バックエンドサービスの前にある LB 仮想サーバーを作成し、認証ロジックを認証仮想サーバーに委任する必要があります。

```
> add lb vserver lb1 SSL 1.2.3.162 443 -persistenceType NONE -
   cltTimeout 180 -AuthenticationHost otpauth.server.com -
   Authentication ON -authnVsName authvs

> bind ssl vserver lb1 -certkeyname c1
```

負荷分散でバックエンドサービスを表すには、次のようにします。

```
> add service iis_backendsso_server_com 1.2.3.4 HTTP 80

> bind lb vserver lb1 iis_backendsso_server_com
```

最初のポリシーとして認証を無効にした **LDAP** アクションの作成

```
> add authentication ldapAction ldap3 -serverIP 1.2.3.4 -serverPort 636
    -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
   administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
   ldapLoginName samAccountName -authentication disabled

> add authentication policy ldap3 -rule aaa.LOGIN.VALUE("passwdreset").
   EQ("1") -action ldap3
```

ナレッジベースの質問と回答の検証アクションを作成する

セルフサービスパスワードリセットフローでナレッジベースの質問と回答を検証するには、認証を無効にして LDAP サーバを設定する必要があります。


```
> add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP
    > -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
    KBAttribute  <LDAP ATTRIBUTE> - alternateEmailAttr <LDAP ATTRIBUTE>
    -authentication DISABLED
```

例:

```
> add authentication ldapAction ldap2 -serverIP 1.2.3.4 -serverPort 636
     -ldapBase ”OU=Users,DC=server,DC=com” -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName samAccountName -KBAttribute userParameters -
    alternateEmailAttr userParameters -authentication disabled
```

CLI を使用してナレッジベースの質問と回答を検証するための認証ポリシーを作成するには

```
add authentication policy kba_validation -rule true -action ldap2
```

E メール検証アクションを作成する

セルフサービスパスワードリセット登録の一環として、ユーザーの電子メール ID または代替電子メール ID が必要なため、LDAP は電子メール検証要素に先行する必要があります。

> 注:
>
> 電子メール OTP ソリューションを機能させるには、SMTP サーバーでログインベースの認証が有効になっていることを確認してください。

ログインベースの認証が有効になっていることを確認するには、SMTP サーバーで次のコマンドを入力します。ログインベースの認証が有効になっている場合、**AUTH LOGIN** というテキストが出力に太字で表示されます。

```
root@ns# telnet <IP address of the SMTP server><Port number of the
    server>
ehlo
```

例:


```
root@ns# telnet 10.106.3.66 25
Trying 10.106.3.66...
Connected to 10.106.3.66.
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22
    Nov 2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.41.151]
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
```

ログインベースの認証を有効にする方法については、https://support.microfocus.com/kb/doc.php?id=7020367を参照してください。

CLI を使用して電子メールアクションを設定するには

```
add authentication emailAction emailact -userName sender@example.com -
    password <Password> -serverURL "smtps://smtp.example.com:25" -
    content "OTP is $code"
```

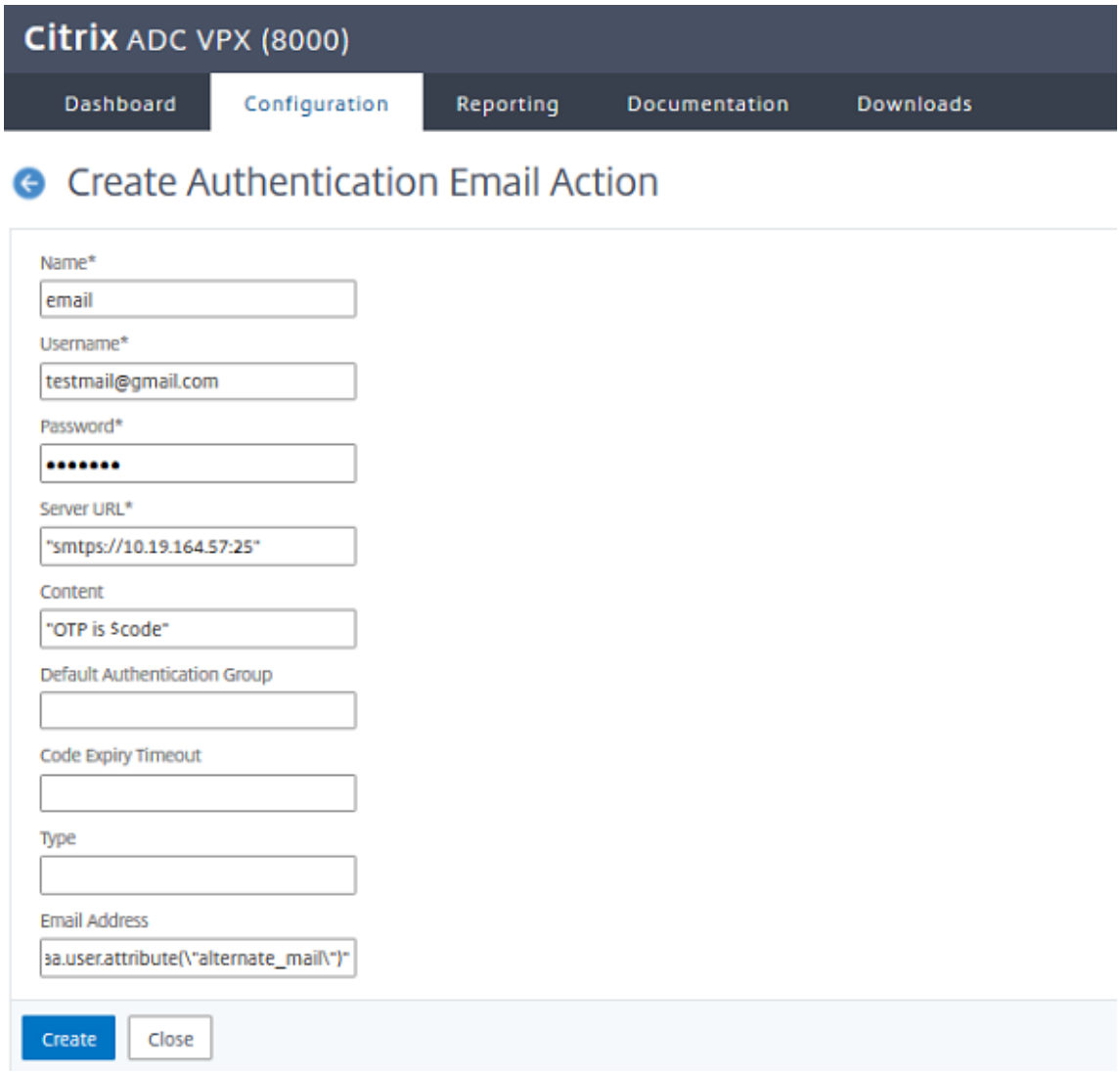
例:

```
add authentication emailAction email -userName testmail@gmail.com -
    password 298
    a34b1a1b7626cd5902bbb416d04076e5ac4f357532e949db94c0534832670 -
    encrypted -encryptmethod ENCMTHD_3 -serverURL "smtps
    ://10.19.164.57:25" -content "OTP is $code" -emailAddress "aaa.user.
    attribute(\"alternate_mail\")"
```


> 注
>
> 構成の「EmailAddress」パラメータは PI 式です。そのため、セッションのデフォルトのユーザー電子メール ID、または既に登録済みの代替電子メール ID のいずれかを使用するように設定されます。

GUI を使用して電子メール **ID** を設定するには

1. [セキュリティ] **>** **[AAA-**アプリケーショントラフィック] **>** **[**ポリシー] **>** **[**認証] **>** **[**詳細ポリシー] **>** **[**アクション] [追加] をクリックします。
2. [認証メールアクションの作成] ページで詳細を入力し、[作成] をクリックします。



CLI を使用して **E** メール検証用の認証ポリシーを作成するには

```
add authentication policy email_validation -rule true -action email
```


パスワードリセット係数の認証ポリシーを作成するには

```
add authentication policy ldap_pwd -rule true -action ldap_logon_action
```

ログインスキーマによる **UI** の提示

パスワードをリセットするためのセルフサービスパスワードリセット用の LoginSchema は 3 つあります。次の CLI コマンドを使用して、3 つのログインスキーマを表示します。

```
root@ns# cd  /nsconfig/loginschema/LoginSchema/
root@ns# ls -ltr | grep -i password
-r--r--r--  1 nobody  wheel  2088 Nov 13 08:38
   SingleAuthPasswordResetRem.xml
-r--r--r--  1 nobody  wheel  1541 Nov 13 08:38
   OnlyUsernamePasswordReset.xml
-r--r--r--  1 nobody  wheel  1391 Nov 13 08:38 OnlyPassword.xml
```

CLI を使用して **1** つの認証パスワードリセットを作成するには

```
> add authentication loginSchema lschema_password_reset -
   authenticationSchema ”/nsconfig/loginschema/LoginSchema/
   SingleAuthPasswordResetRem.xml”

> add authentication loginSchemaPolicy lpol_password_reset -rule true -
   action lschema_password_reset
```

ポリシーラベルを使用して、ナレッジベースの質問と回答と **E** メールによる **OTP** 検証ファクタを作成

最初の要素が LDAP ログオンの場合、次のコマンドを使用して、次の要素に対する知識ベースの質問と回答および電子メールの OTP ポリシーラベルを作成できます。

```
> add authentication loginSchema lschema_noschema -authenticationSchema
    noschema

> add authentication policylabel kba_validation -loginSchema
   lschema_noschema

```



```
5 > add authentication policylabel email_validation -loginSchema
    lschema_noschema
```

ポリシーラベルによるパスワードリセット係数の作成

次のコマンドを使用して、ポリシーラベルからパスワードリセット係数を作成できます。

```
1 > add authentication loginSchema lschema_noschema -authenticationSchema
     noschema
2
3 > add authentication policylabel password_reset -loginSchema
    lschema_noschema
4
5 > bind authentication policylabel password_reset -policyName ldap_pwd -
    priority 10 -gotoPriorityExpression NEXT
```

次のコマンドを使用して、ナレッジベースの質問と回答と電子メールポリシーを以前に作成したポリシーにバインドします。

```
1 > bind authentication policylabel email_validation -policyName
    email_validation -nextfactor password_reset -priority 10 -
    gotoPriorityExpression NEXT
2
3 > bind authentication policylabel kba_validation -policyName
    kba_validation -nextfactor email_validation -priority 10 -
    gotoPriorityExpression NEXT
```

フローをバインドする

LDAP Logon の認証ポリシーの下で LDAP ログオンフローが作成されている必要があります。このフローでは、最初の LDAP ログオンページに表示される [パスワードを忘れた場合] リンクをクリックし、次に KBA 検証、OTP 検証、最後にパスワードリセットページの順にクリックします。

```
1 bind authentication vserver authvs -policy ldap3 -nextfactor
    kba_validation -priority 10 -gotoPriorityExpression NEXT
```

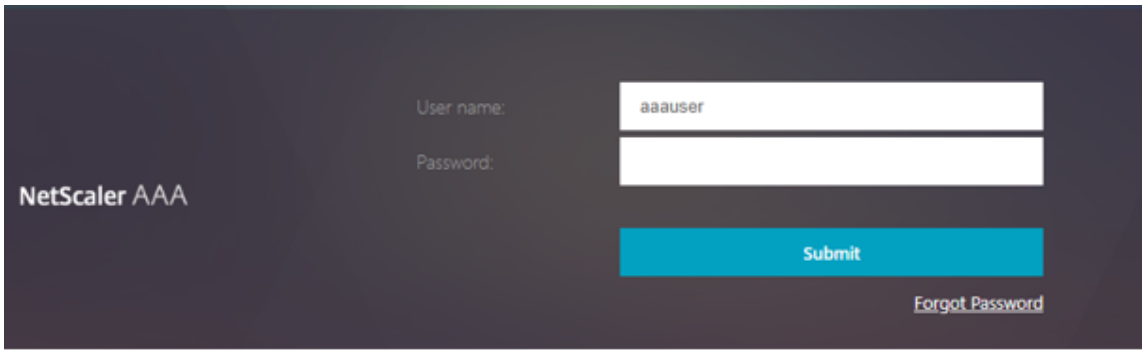

すべての **UI** フローをバインドするには

```
bind authentication vserver authvs -policy lpol_password_reset -
    priority 20 -gotoPriorityExpression END
```

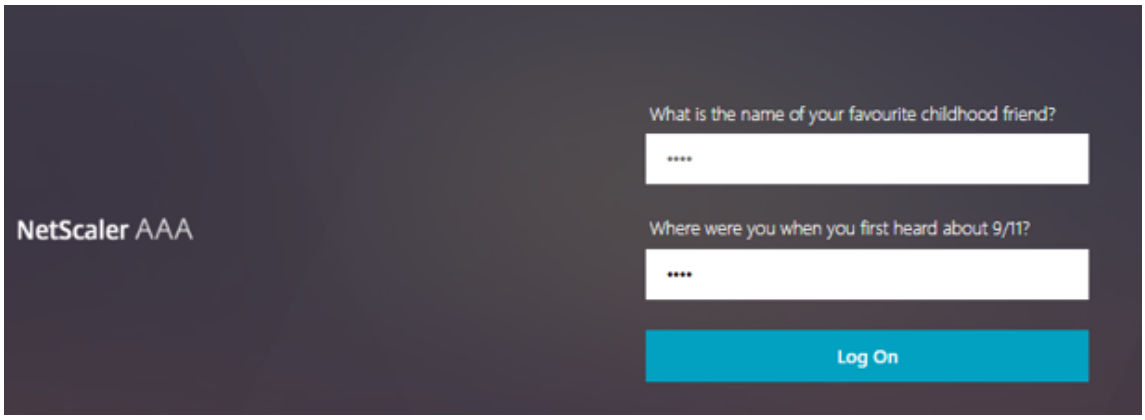
パスワードをリセットするユーザーログオンワークフロー

ユーザーがパスワードをリセットする必要がある場合のユーザーログオンワークフローを次に示します。

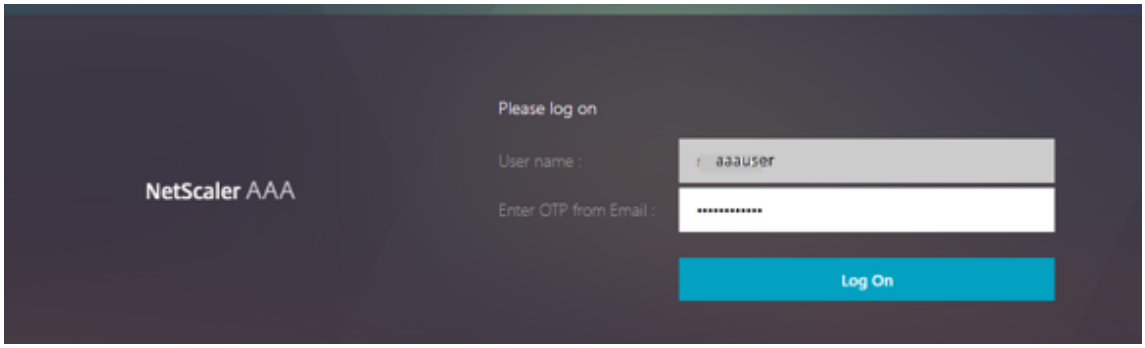
1. ロードバランサー仮想サーバの URL を入力します（例:https://lb1.server.com）。ログオン画面が表示されます。



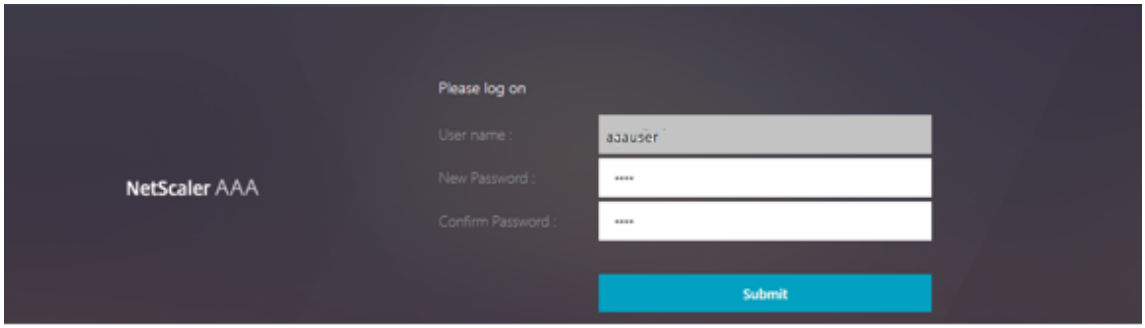
2. [パスワードを忘れた場合] をクリックします。検証画面には、AD ユーザーに対して登録された最大 6 つの質問と回答のうち 2 つの質問が表示されます。



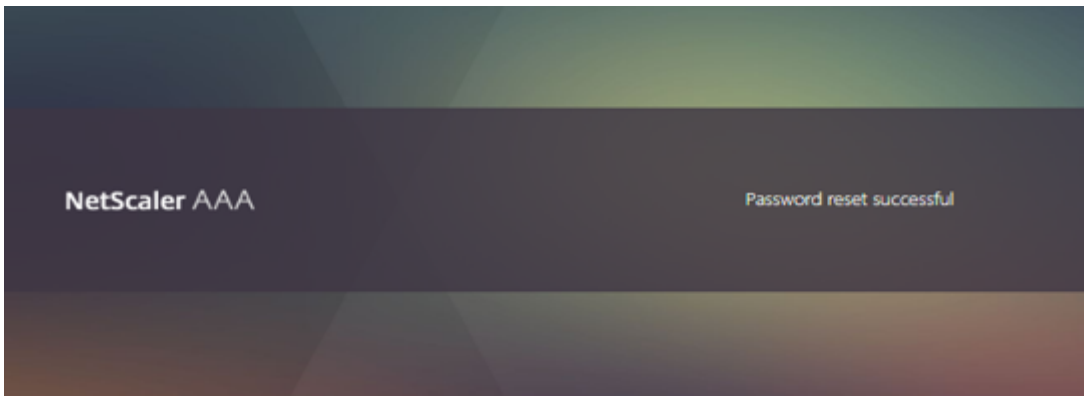
3. 質問に答えて、[ログオン] をクリックします。登録済みの代替電子メール ID で受信した OTP を入力する必要がある電子メール OTP 検証画面が表示されます。



4. 電子メール OTP を入力します。電子メール OTP の検証に成功すると、パスワードのリセットページが表示されます。



5. 新しいパスワードを入力し、新しいパスワードを確認します。[**Submit**] をクリックします。パスワードのリセットに成功すると、[パスワードリセット成功] 画面が表示されます。



これで、リセットパスワードを使用してログオンできます。

トラブルシューティング

Citrix には、セルフサービスパスワードリセットの使用中に発生する可能性のある基本的な問題のトラブルシューティングを行うオプションがあります。次のセクションは、特定の領域で発生する可能性がある問題のトラブルシューティングに役立ちます。

NS ログ

ログを分析する前に、次のコマンドを使用してログレベルを debug に設定することをお勧めします。

```
> set syslogparams -loglevel DEBUG
```

登録

次のメッセージは、ユーザー登録が成功したことを示します。

```
”ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1”
Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
    0-PPE-1 : default SSLVPN Message 1588 0 :  ”
    ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
    eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
    ==.oKmvOalaOJ3a9z7BcGCSegNPMw==”
```

知識ベースの質問と回答の検証

次のメッセージは、ナレッジベースの質問と回答の検証に成功したことを示します。

```
”NFactor: Successfully completed KBA Validation, nextfactor is email”
```

電子メール **ID** の検証

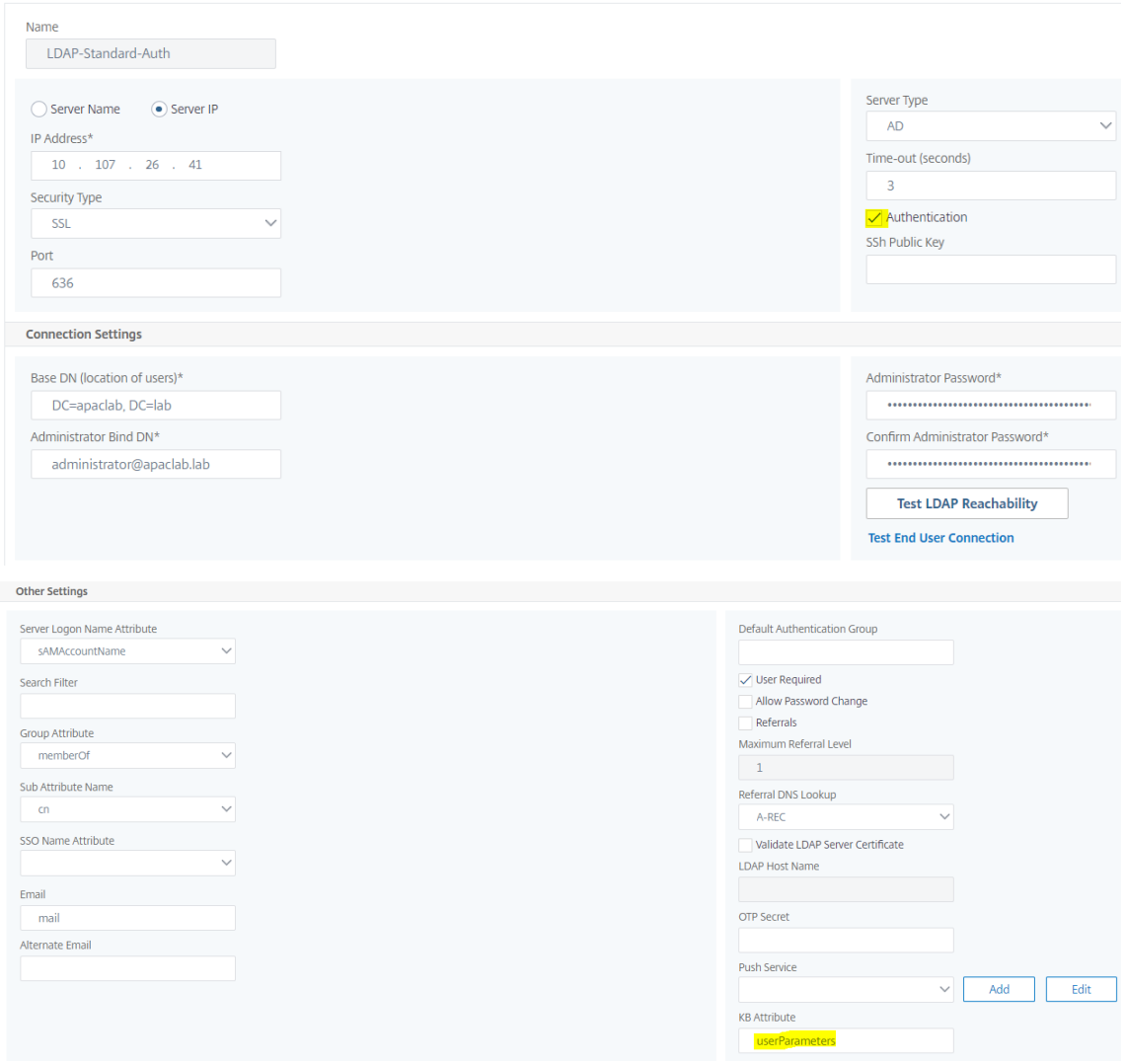
次のメッセージは、パスワードのリセットが成功したことを示します。

```
”NFactor: Successfully completed email auth, nextfactor is pwd_reset”
```

nFactor ビジュアライザーを使用した **SSPR** の設定

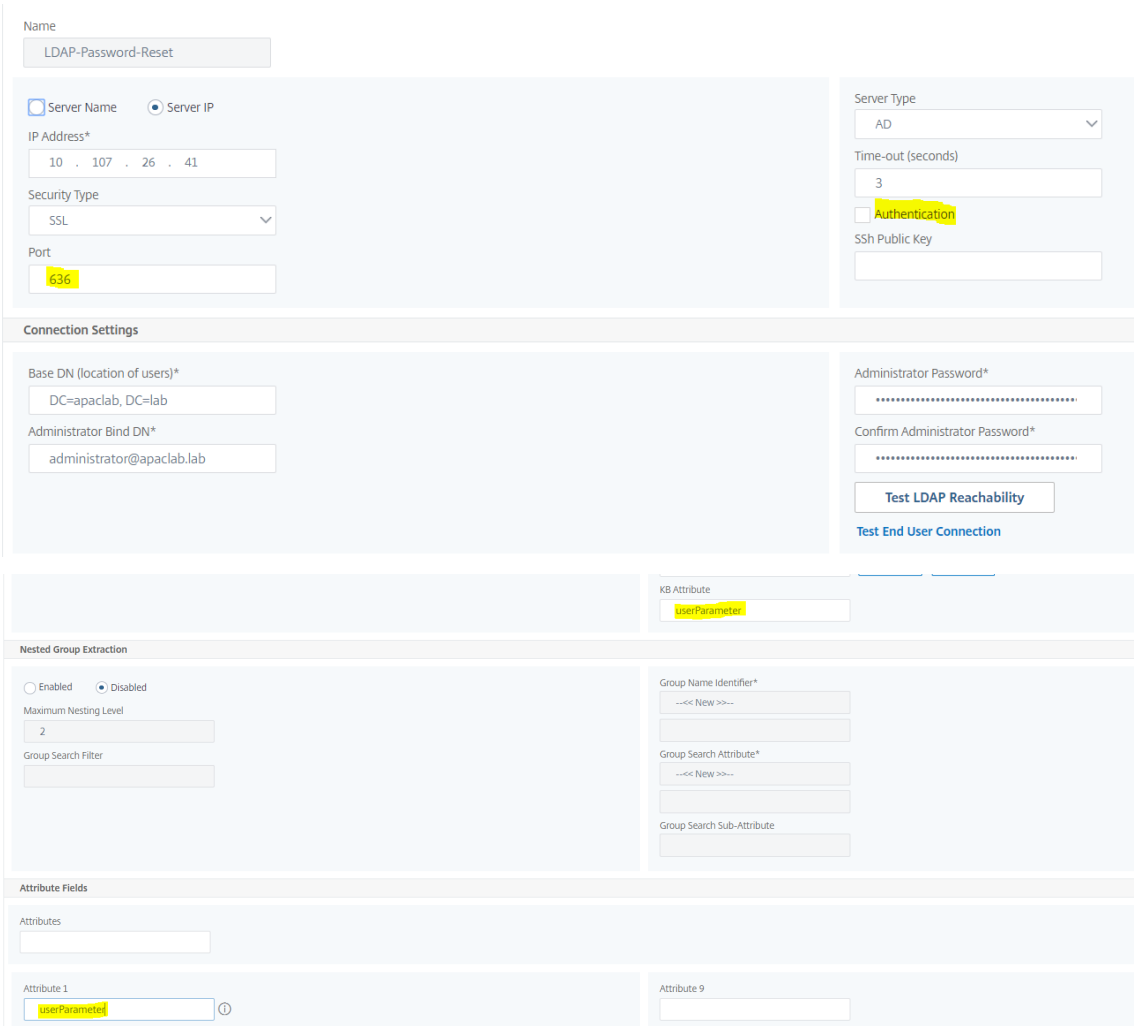
SSPR の設定を開始する前に、次の LDAP サーバを追加する必要があります。

1. ユーザ認証に対して認証が有効になっていて、AD 属性が指定されている標準 LDAP サーバ。

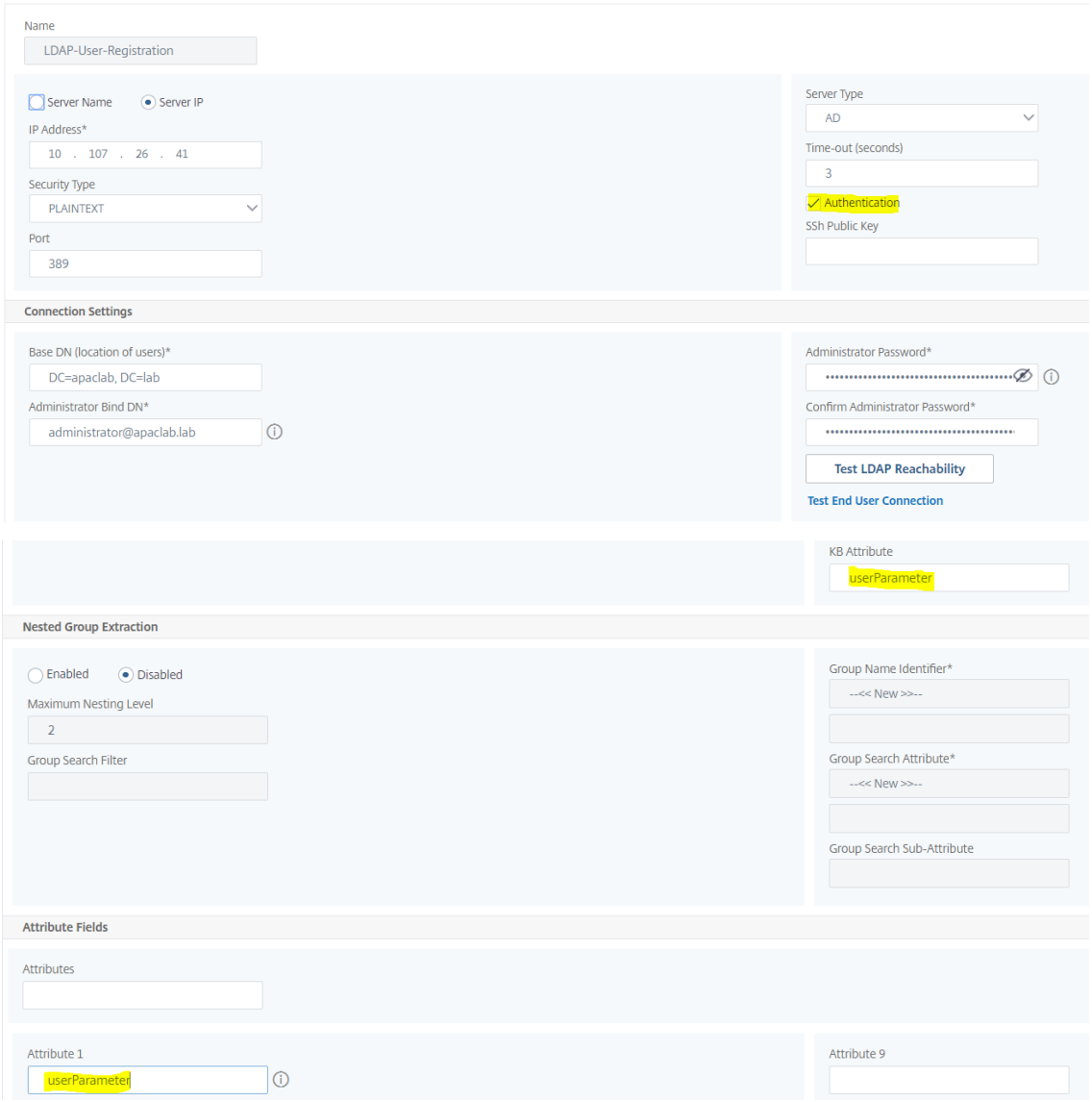


2. 認証なしでユーザパラメータを抽出するための LDAP サーバ。

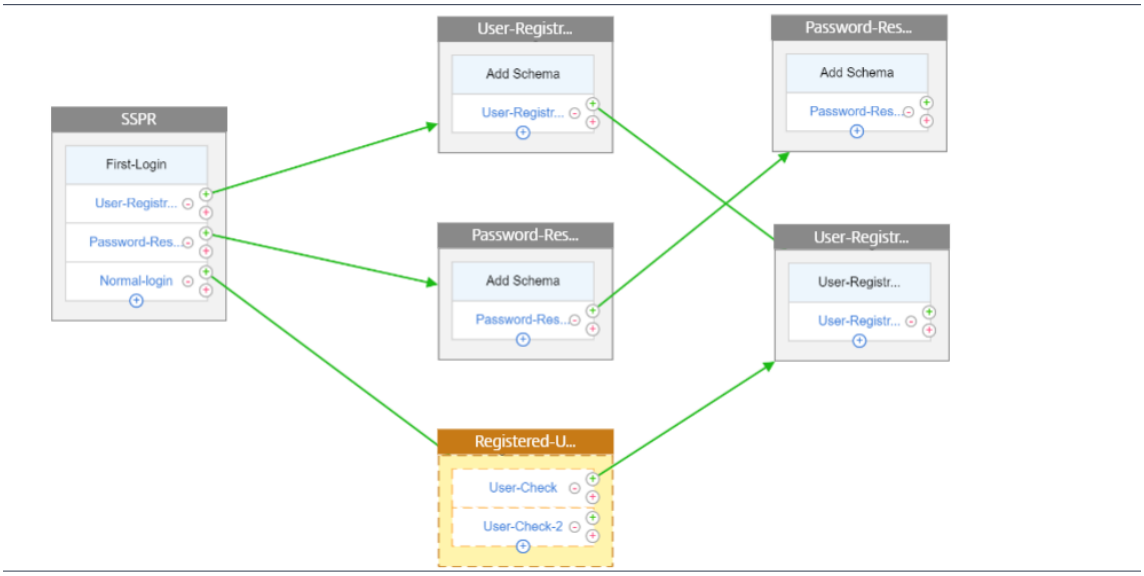
3. 認証なしの SSL でパスワードをリセットするための LDAP サーバ。また、ユーザー詳細の保存に使用する AD 属性は、このサーバーで定義する必要があります。



4. 認証が有効で、AD 属性が指定された、ユーザ登録用の LDAP サーバ



5. 次の図に、フロー全体を示します。



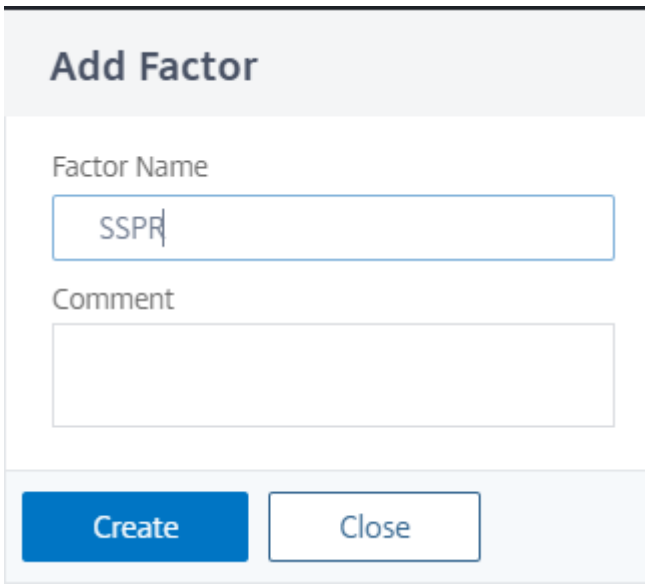
6. 次の CLI コマンドを使用して、証明書をグローバルにバインドします。

```
1 bind vpn global -userDataEncryptionKey Wildcard
```

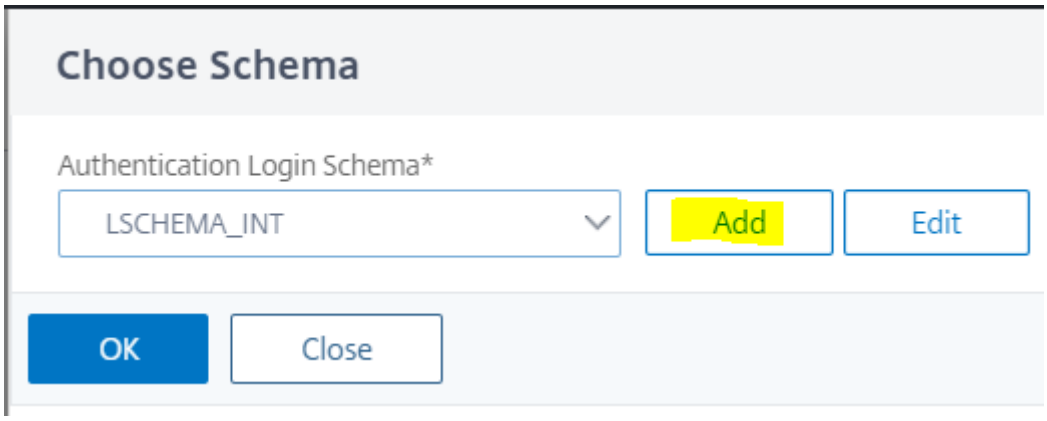
LDAP サーバが追加されたので、ビジュアライザを使用して **nFactor** の設定を続行します

1. [セキュリティ] **> [AAA] >** [アプリケーショントラフィック] **> [nFactor Visualizer] > [nFactor** フロー] に移動し、[追加] をクリックして、ボックス内の + アイコンをクリックします。

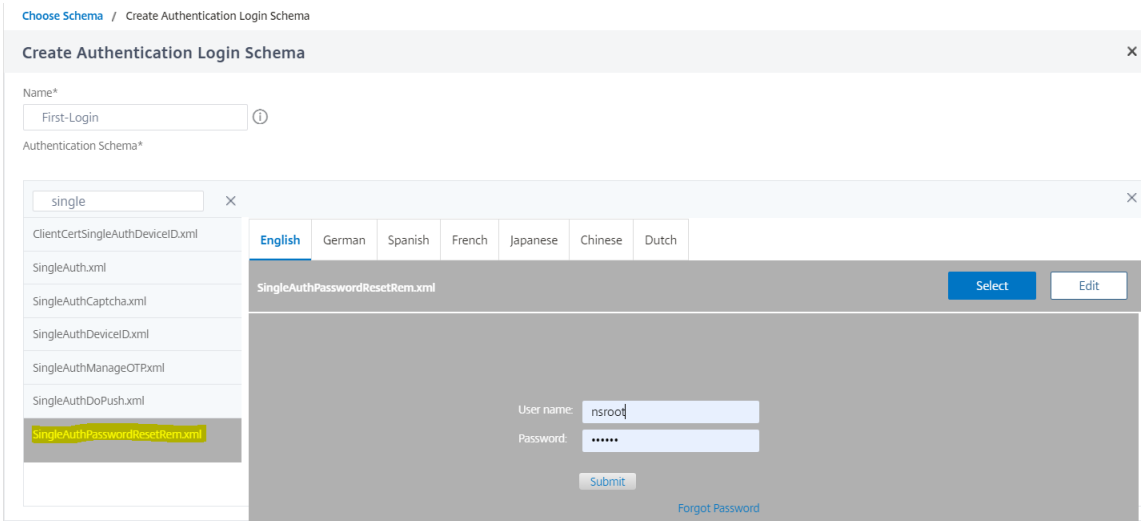
2. フローに名前を付けます。



3. 既定のスキーマとして機能する **[**スキーマの追加] をクリックします。[ログインスキーマ] ページで [追加] をクリックします。



4. スキーマに名前を付けたら、スキーマを選択します。右上隅の「選択」(**Select**) をクリックして、スキーマを選択します。



5. 「作成」をクリックして「**OK**」をクリックします。

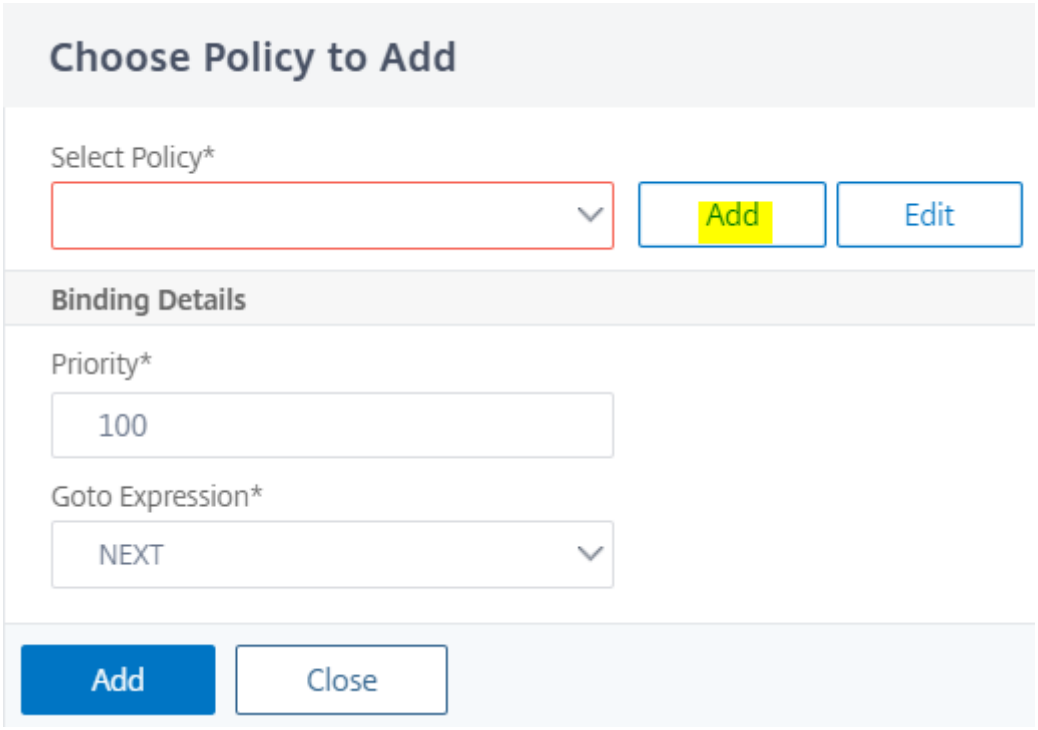
デフォルトのスキーマを追加したら、次の 3 つのフローを設定する必要があります。

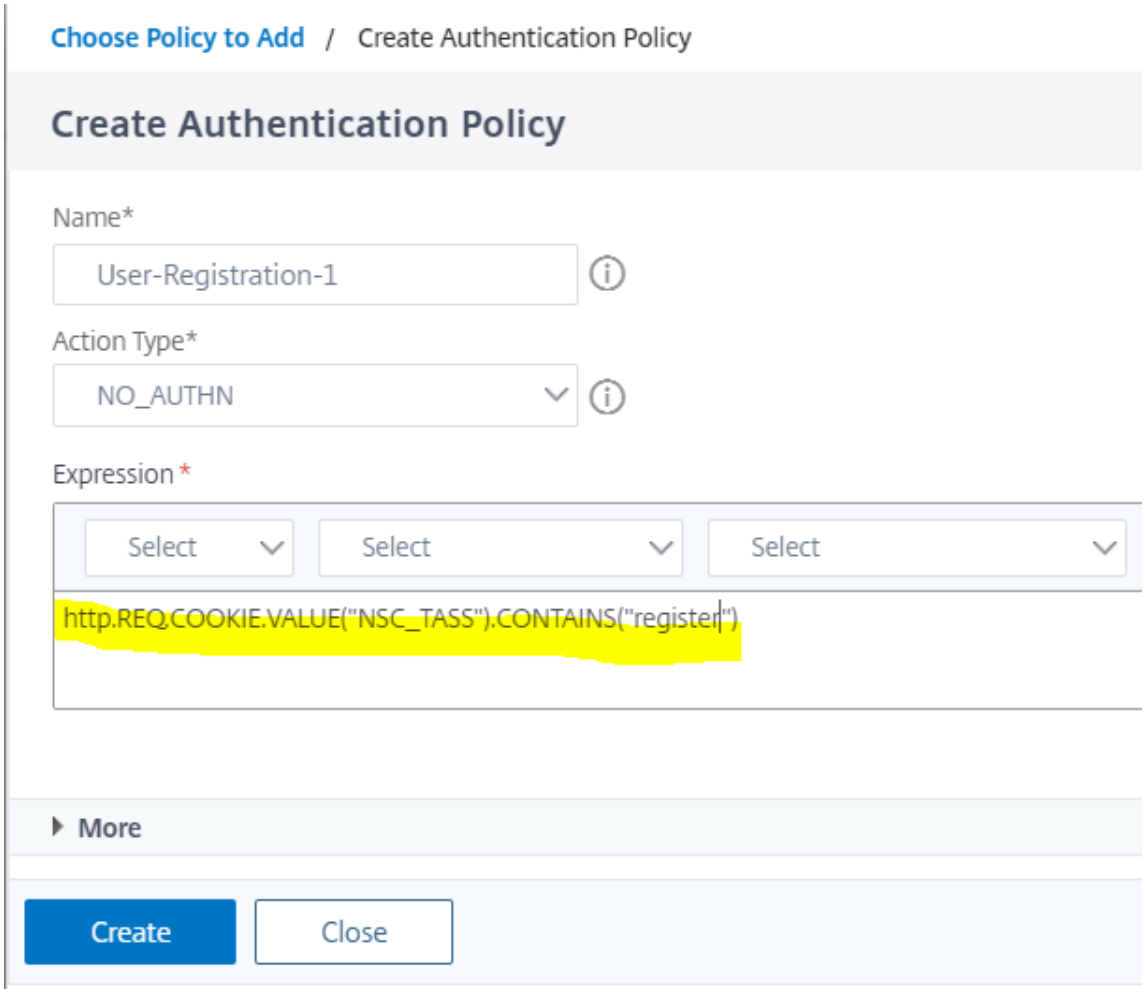
- ユーザー登録: 明示的なユーザー登録の場合
- パスワードリセット: パスワードリセット用
- 通常ログイン **+** 登録済みユーザーチェック: ユーザーが登録され、正しいパスワードを入力した場合、そのユーザーはログインしています。ユーザーが登録されていない場合は、登録ページに移動します。

ユーザー登録

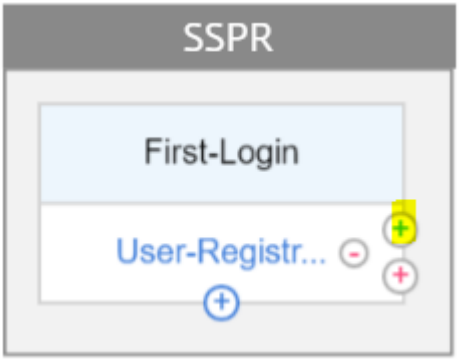
スキーマを追加した後、残ったところから続けてみましょう。

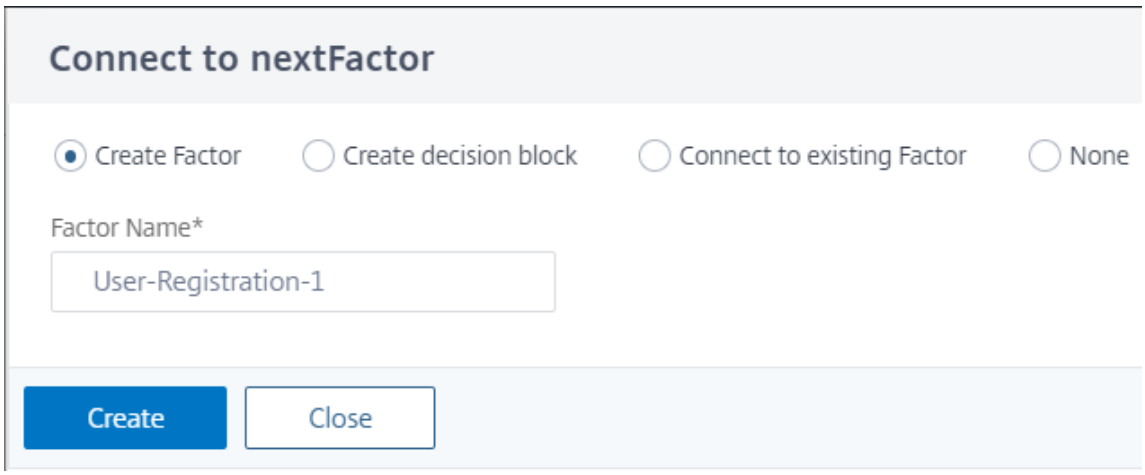
1. [**Add Policy**] をクリックすると、ユーザーが明示的に登録しようとしているかどうかがチェックされます。



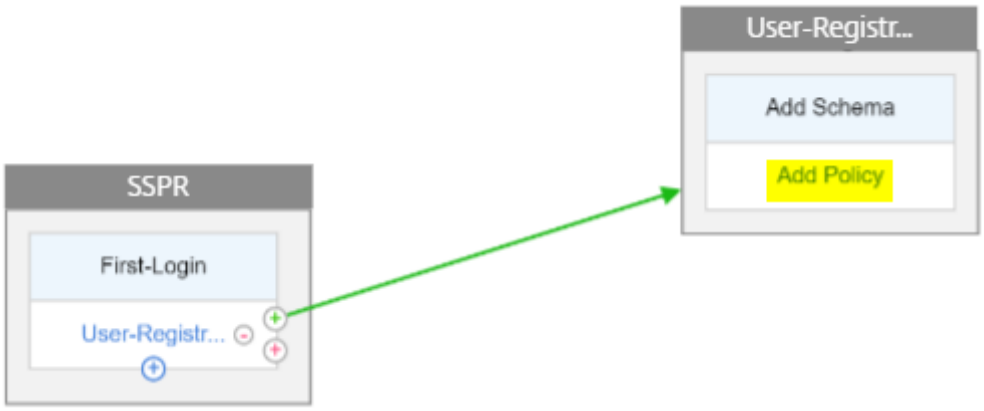


2. [作成] をクリックし、[追加] をクリックします。
3. 強調表示された緑色の「+」アイコンをクリックして、次の認証要素をユーザー登録フローに追加します。

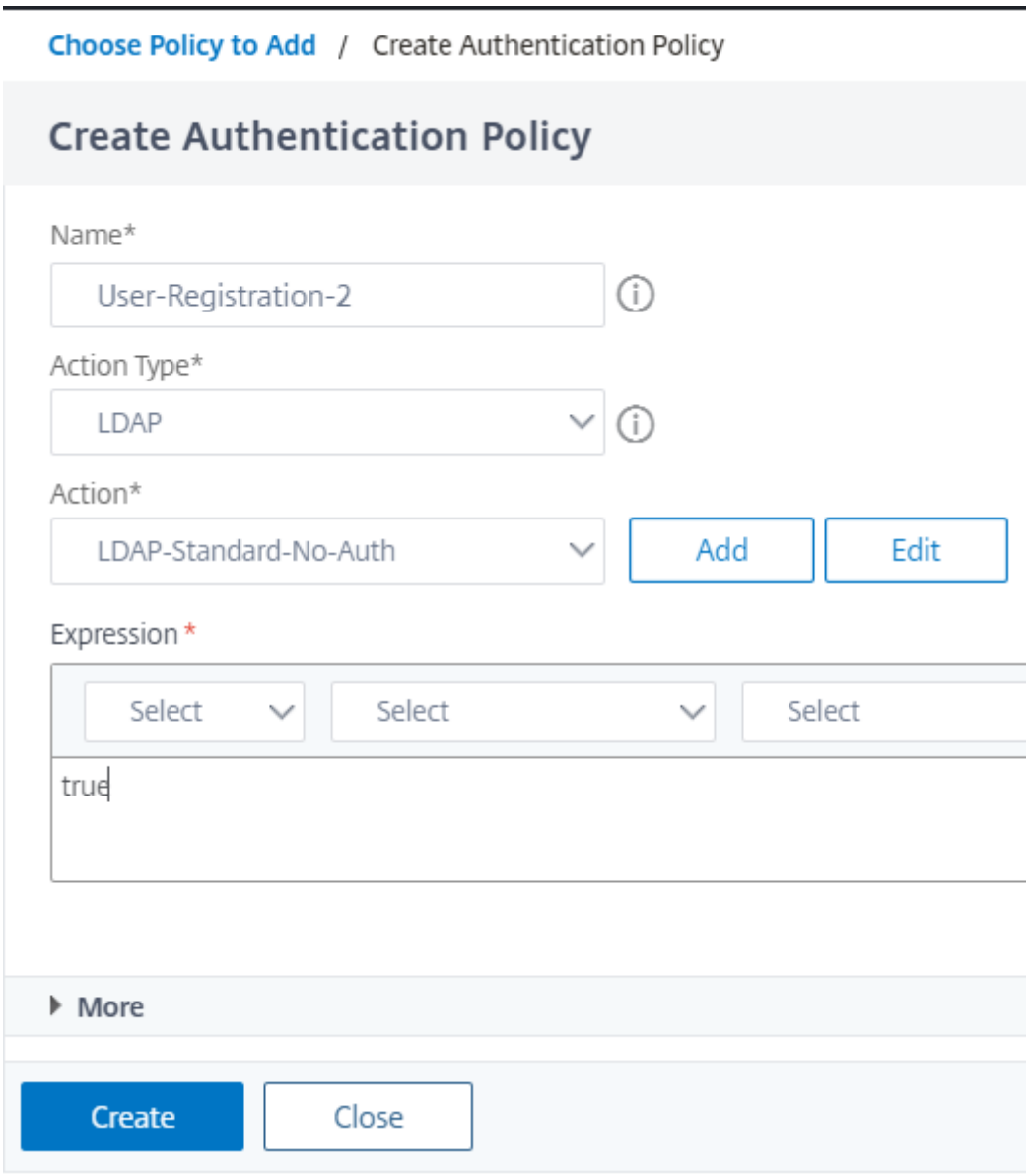




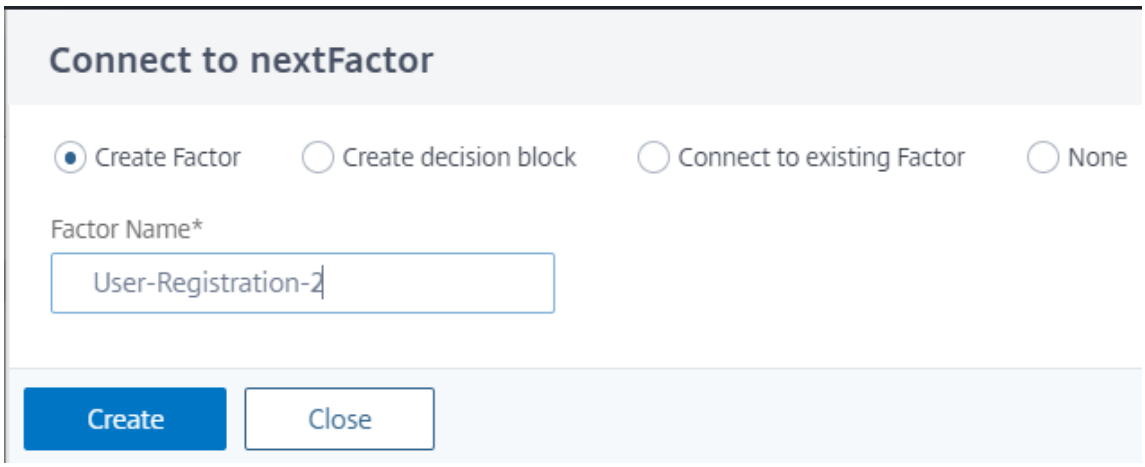
4. [**Create**] をクリックします。
5. [ユーザー登録 1 係数のポリシーの追加] をクリックします。



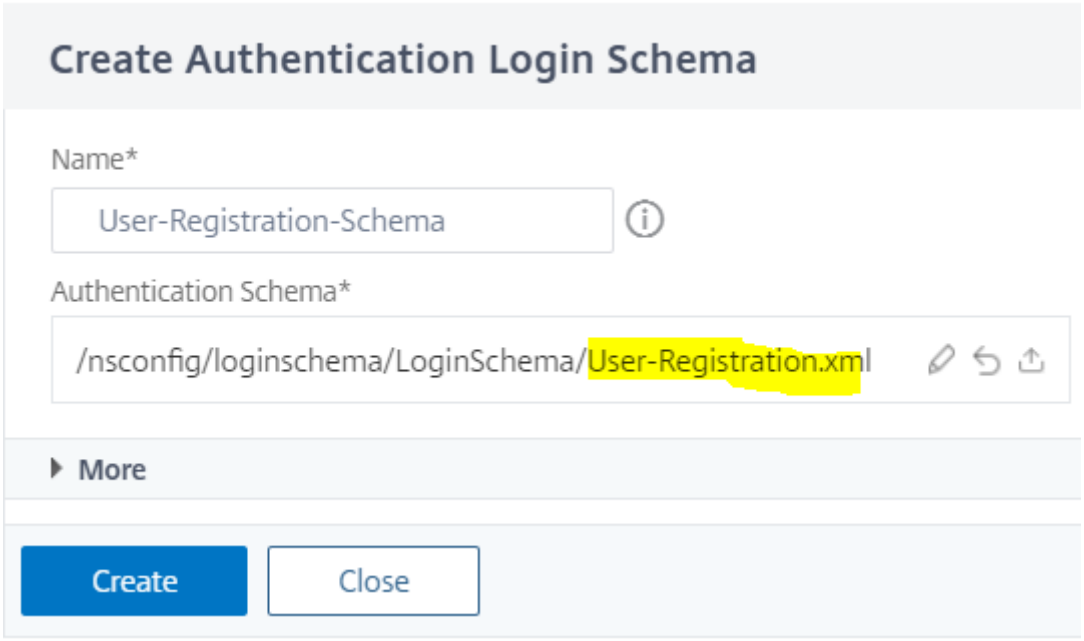
6. 認証ポリシーを作成します。このポリシーは、登録ページにリダイレクトする前に、ユーザー情報を抽出して検証します。



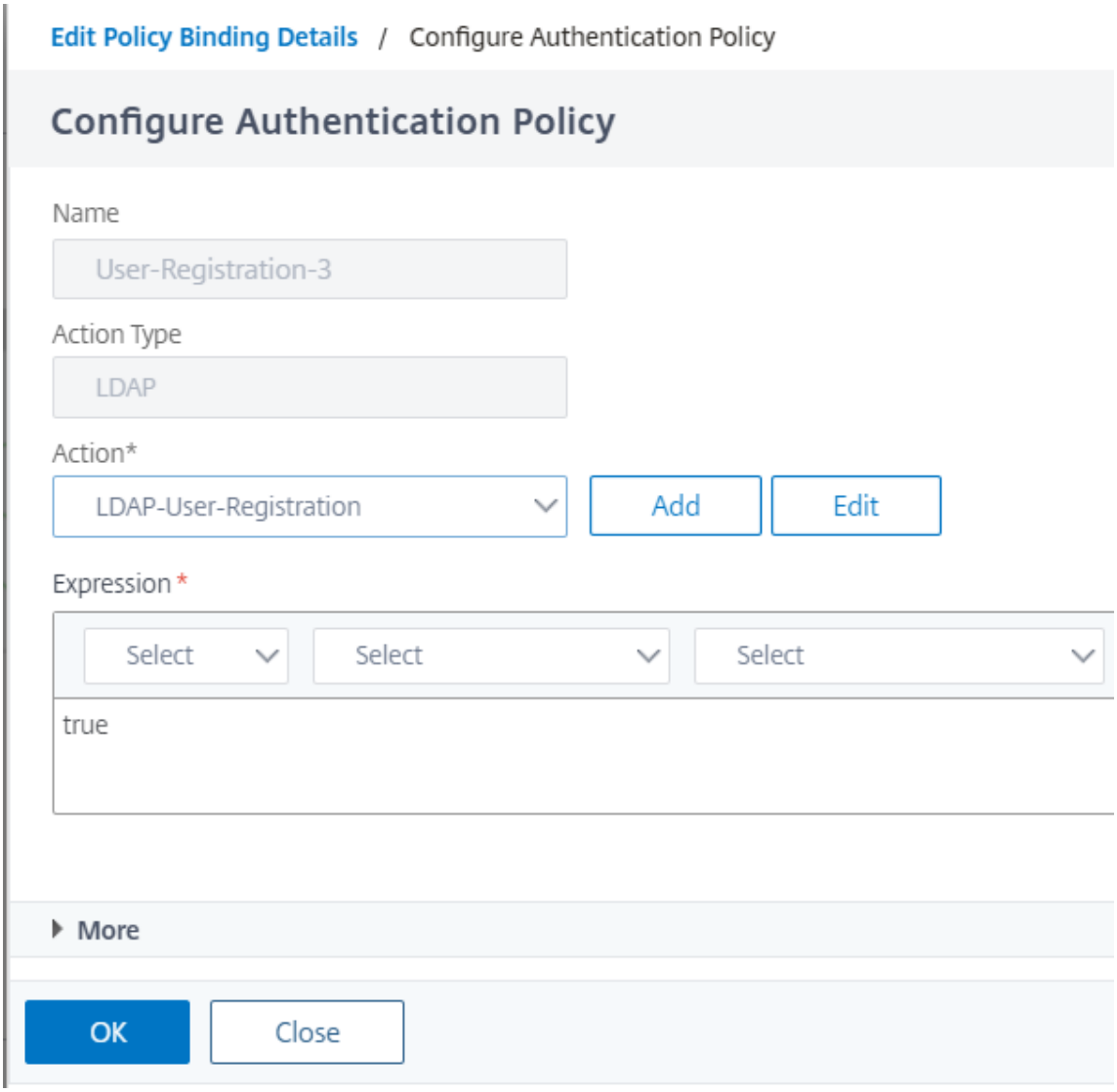
7. [作成] をクリックし、[追加] をクリックします。
8. 緑色の「+」アイコンをクリックしてユーザー登録用の別の要素を作成し、「作成」をクリックします。[スキーマの追加] をクリックします。



9. 次のスキーマを作成します。



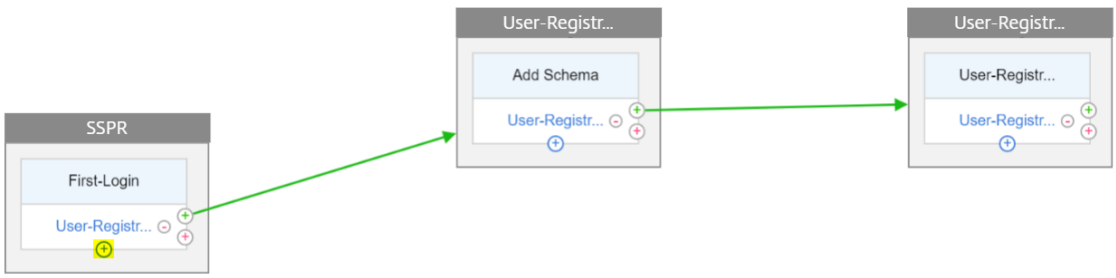
10. [**Add Policy**] をクリックし、次の認証ポリシーを作成します。



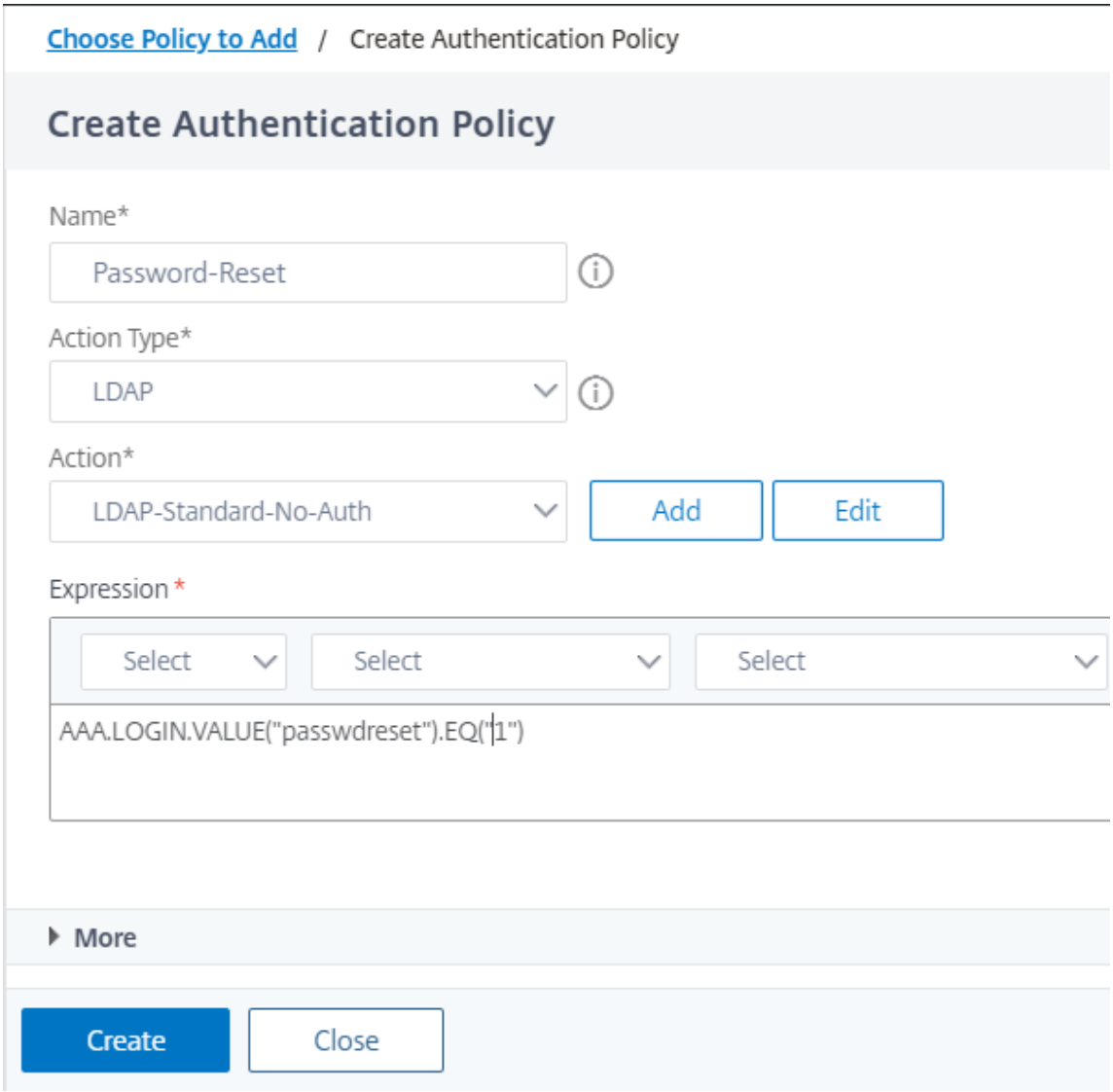
11. [作成] をクリックし、[追加] をクリックします。

パスワードリセット

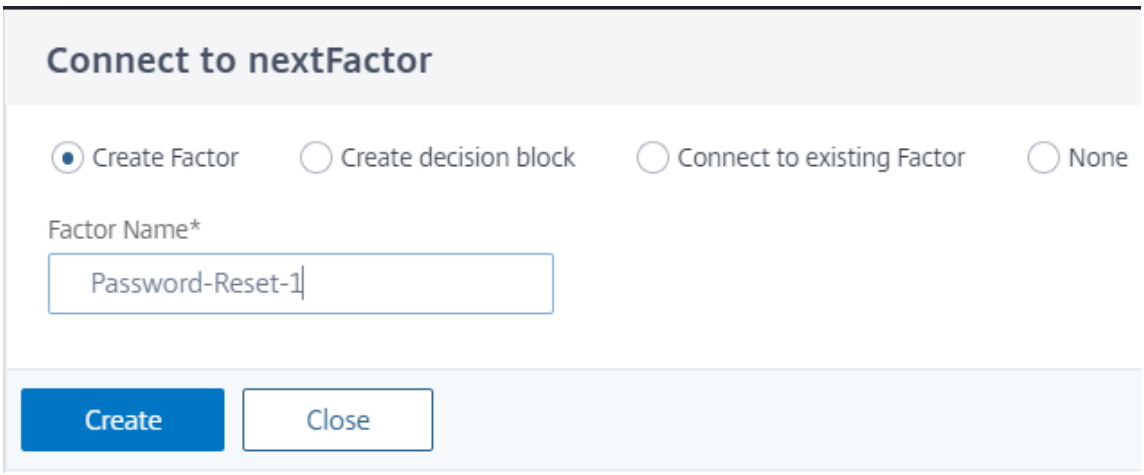
1. 青色の「+」アイコンをクリックして、親 SSPR ファクターに別のポリシー (パスワードリセットフロー) を追加します。



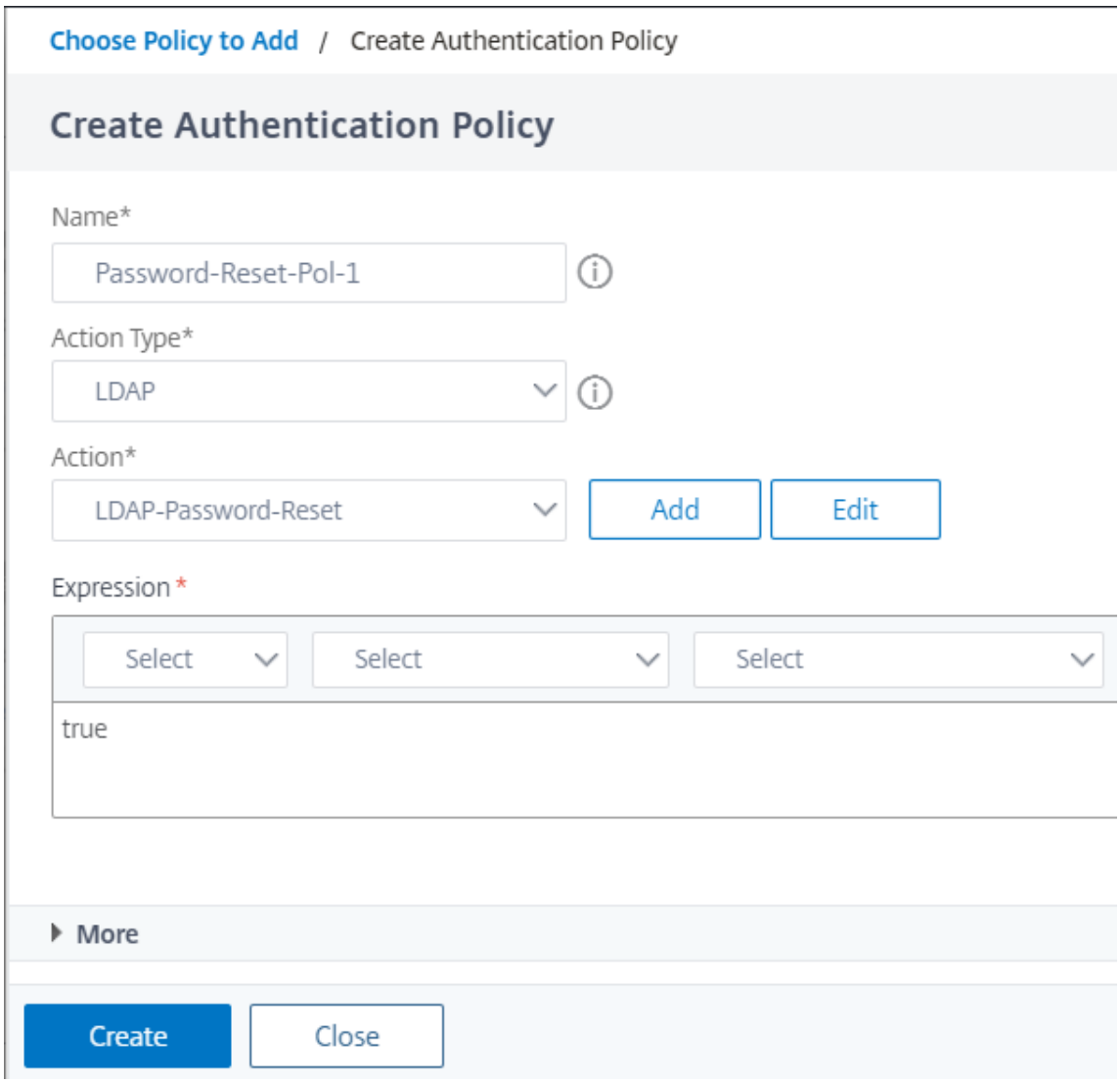
2. [**Add**] をクリックし、認証ポリシーを作成します。このポリシーは、ユーザーがログインページで [パスワードを忘れた場合] をクリックするとトリガーされます。



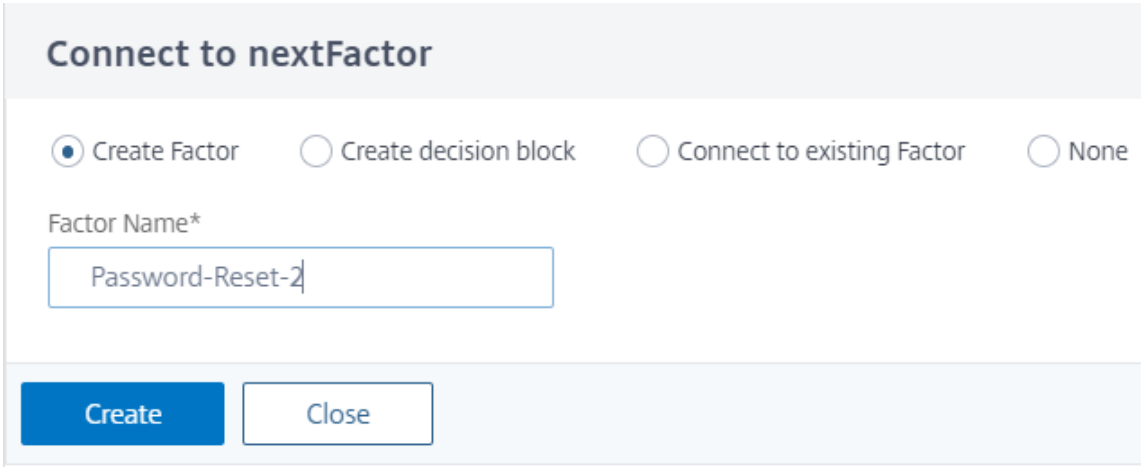
3. [作成] をクリックし、[追加] をクリックします。
4. パスワードリセット認証ポリシーの緑色の「+」アイコンをクリックして、別の要素を追加します。



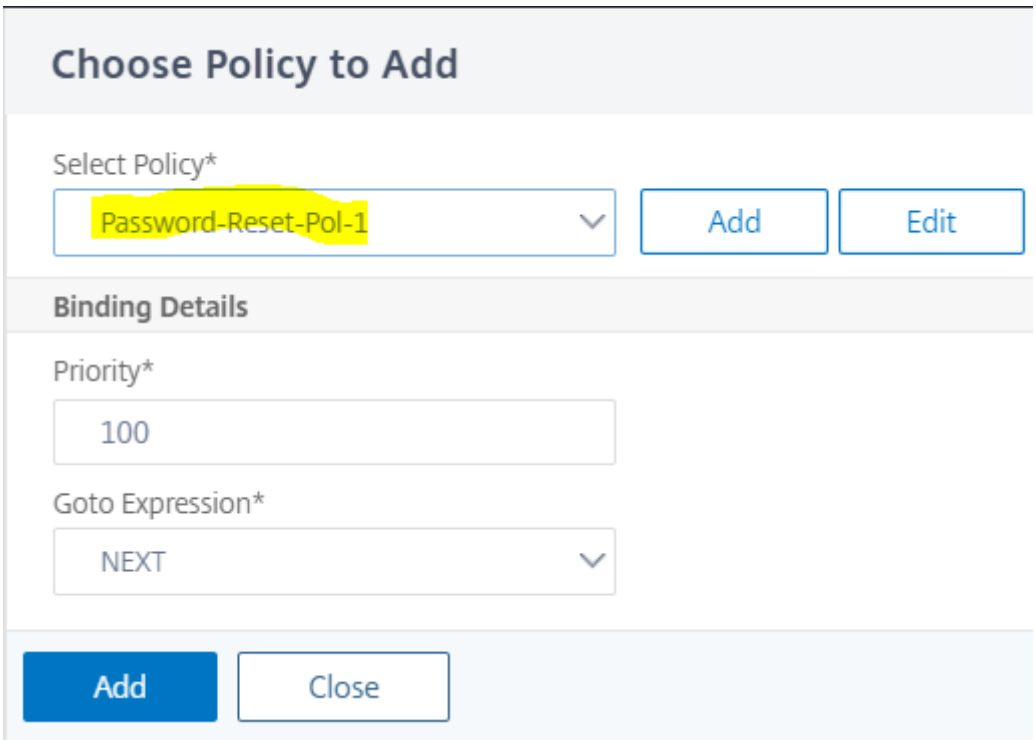
5. [**Create**] をクリックします。
6. [**Add policy**] をクリックして、以前に作成したファクターの認証ポリシーを作成します。この係数は、ユーザーを検証するためのものです。



7. [作成] をクリックし、[追加] をクリックします。
8. 緑色の「+」アイコンをクリックして、パスワード係数フローに別の要素を追加します。これにより、パスワードをリセットするための回答が検証されます。[**Create**] をクリックします。

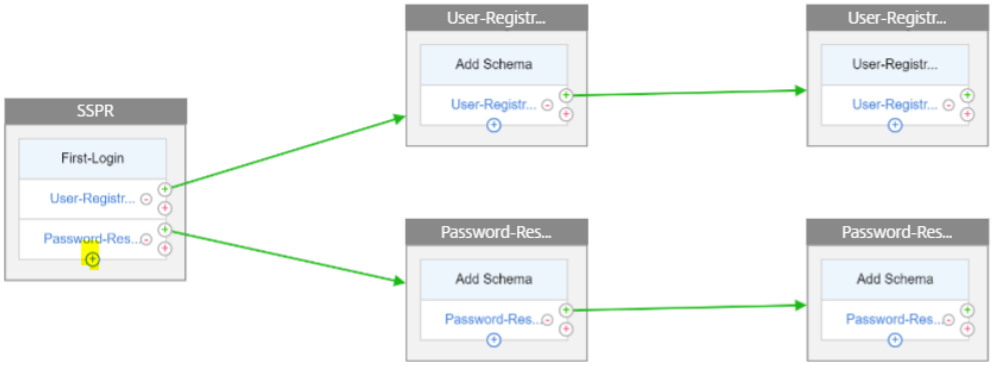


9. [**Add Policy**] をクリックして、ファクタの認証ポリシーを追加します。
10. 前に作成したものと同じ認証ポリシーをドロップダウンメニューから選択し、[**Add**] をクリックします。

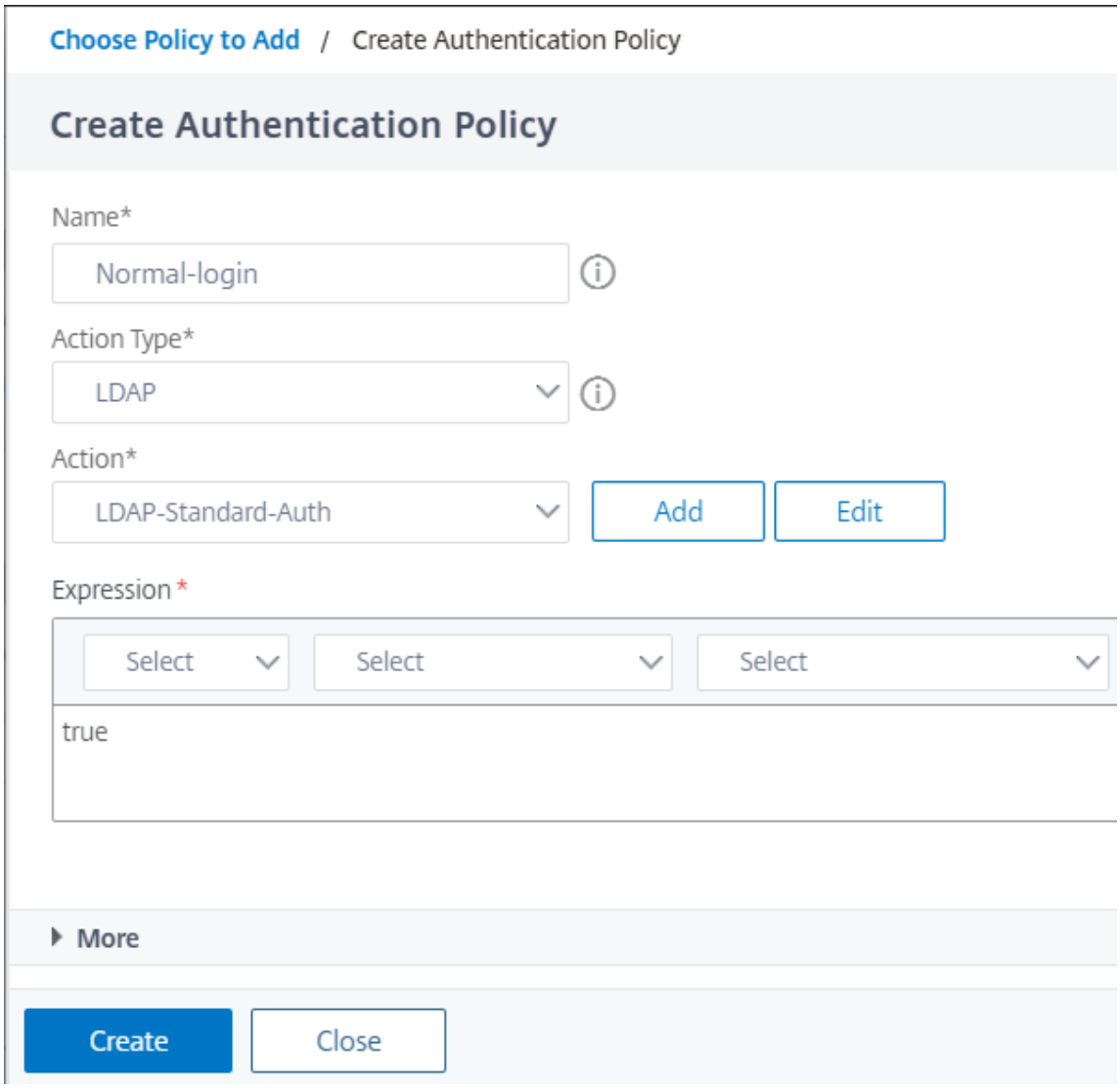


通常ログイン＋登録ユーザーチェック

1. 青い「+」アイコンをクリックして、親 SSPR ファクターに別の認証ポリシー (通常のログインフロー) を追加します。

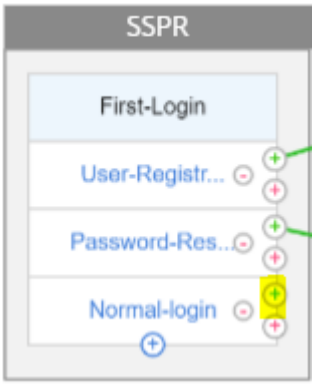


2. [**Add**] をクリックして、通常のユーザログイン用の認証ポリシーを作成します。

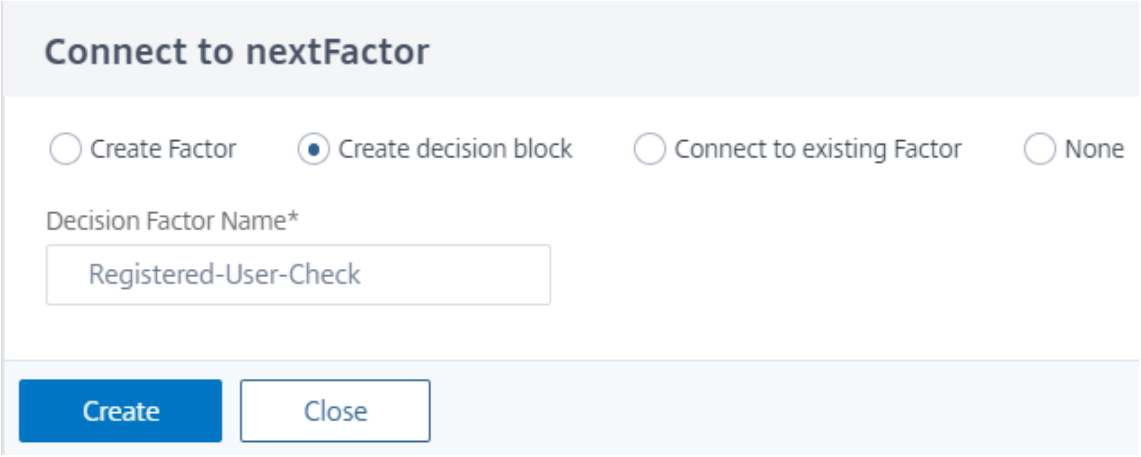


3. [作成] をクリックし、[追加] をクリックします。
4. 以前に作成したポリシーの緑色の「+」アイコンをクリックして、決定ブロックという別の要素を追加します。

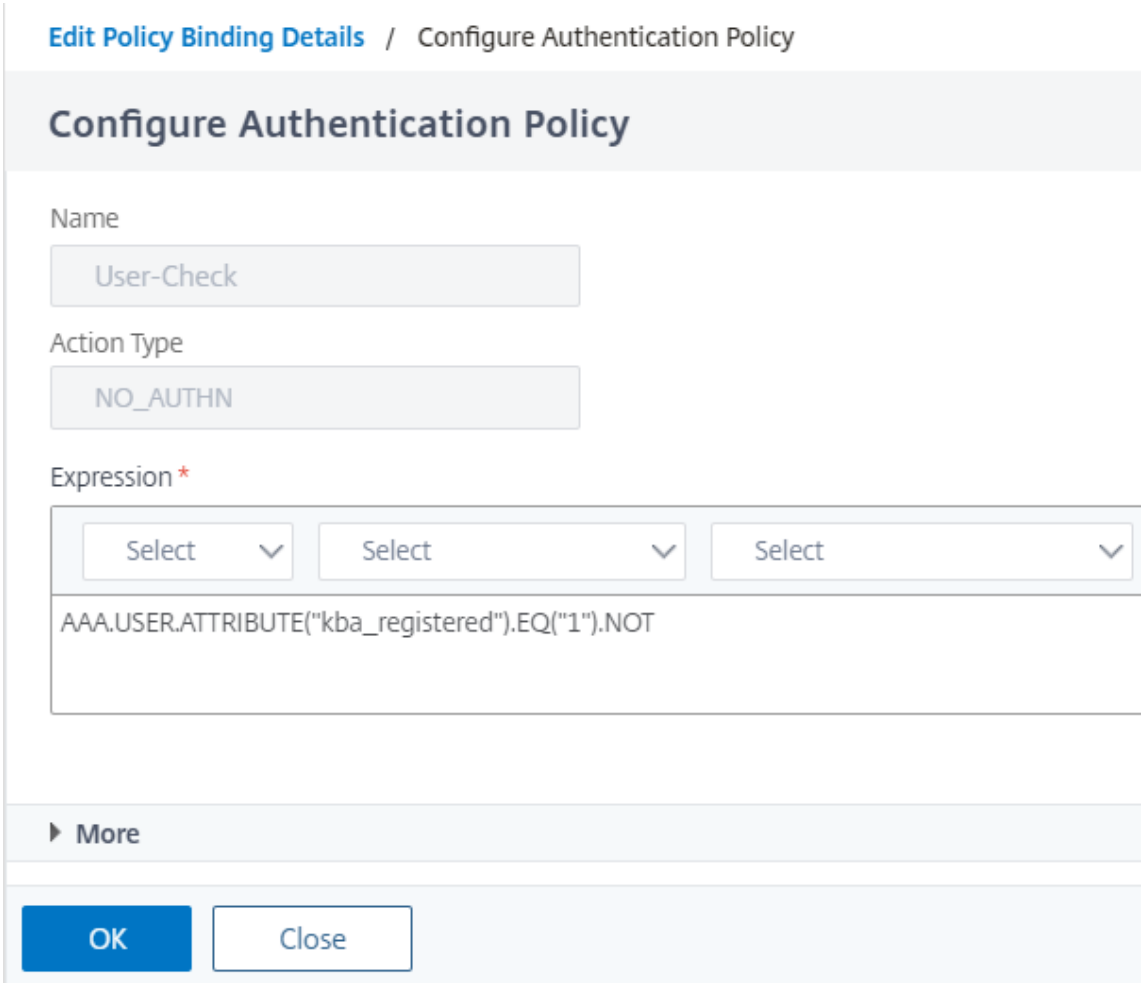
［**Create**］をクリックします。



5. ［**Create**］をクリックします。

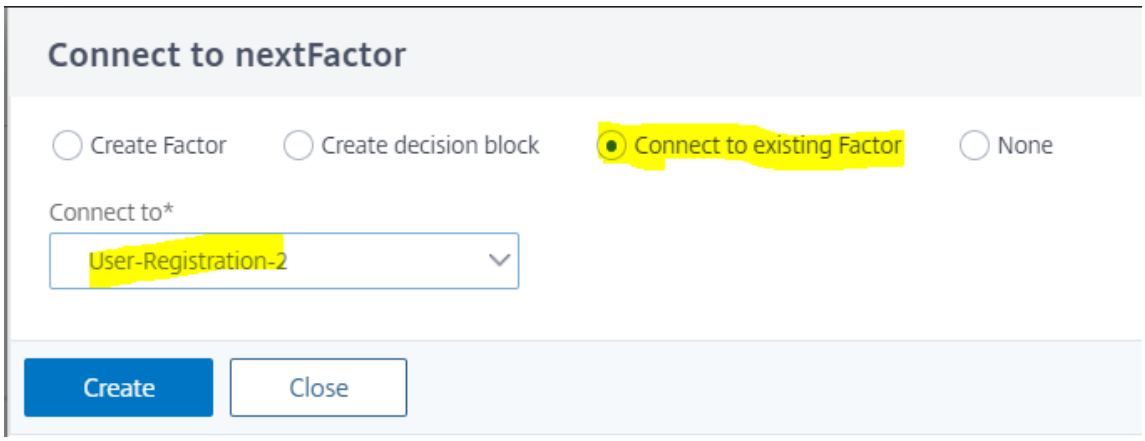


6. [**Add Policy**] をクリックして、この決定要因の認証ポリシーを作成します。

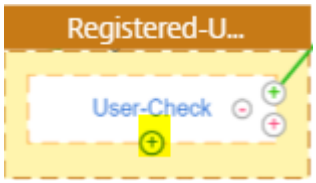


7. [作成] をクリックして [追加] をクリックします。これは、ユーザーが登録されているかどうかをチェックします。
8. 緑色の「+」アイコンをクリックして、登録ポリシーをユーザに指し示します。

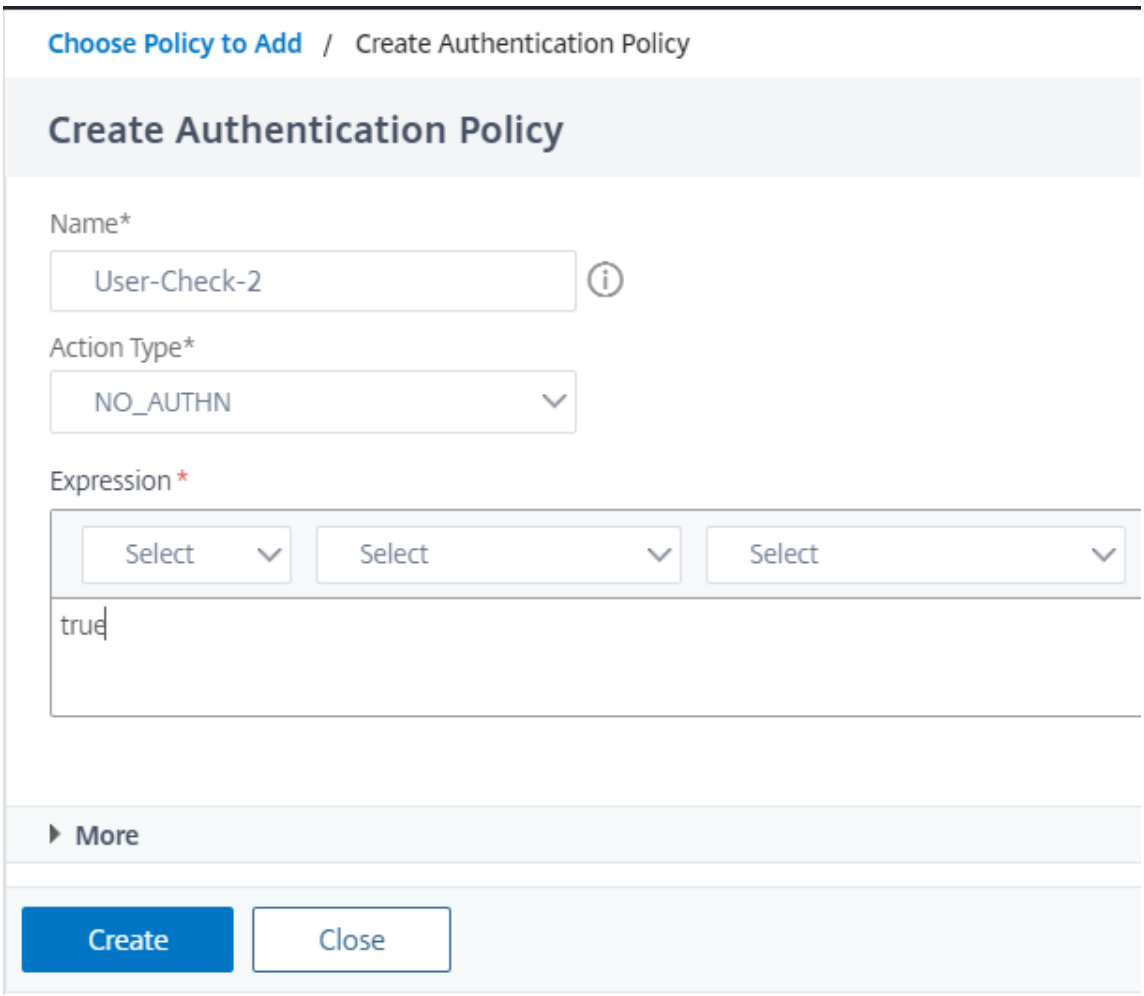
9. ドロップダウンメニューから登録係数を選択し、[**Create**] をクリックします。



10. 青い「+」アイコンをクリックして、別のポリシーを Decision Block に追加します。このポリシーは、登録ユーザーが認証を終了するためのものです。



11. [**Add Policy**] をクリックして、認証ポリシーを作成します。



12. [作成] をクリックし、[追加] をクリックします。

認証中のポーリング

October 7, 2021

Citrix ADC リリースビルド 13.0.79.64 から、多要素認証中に Citrix ADC アプライアンスをポーリングメカニズム用に構成できます。

Citrix ADC アプライアンスでポーリングを構成すると、エンドポイント（Web ブラウザやアプリなど）は、構成された間隔で認証中にアプライアンスをポーリング（プローブ）して、送信された認証要求のステータスを取得できます。

Citrix ADC アプライアンスでの認証中にエンドポイントが TCP 接続をドロップしたときに認証を処理するようにポーリングを構成できます。

注意事項

- ポーリング設定は、LDAP、RADIUS、および TACACS 認証方式でサポートされています。
- クライアントは、第 2 要素以降から認証要求をプローブできます。

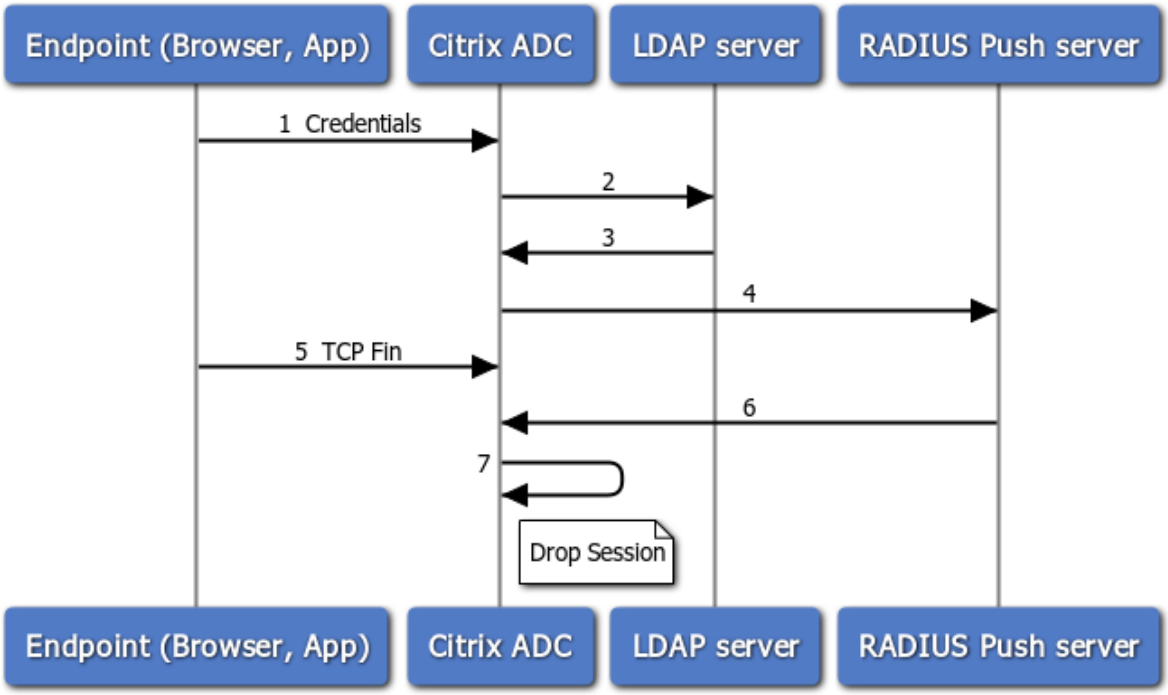
ポーリングを設定する理由

認証中にアプリケーション（ログインアプリと認証アプリなど）を切り替えると、エンドポイントが Citrix ADC アプライアンスとの接続を失い、認証フローが中断することがあります。ポーリングを設定すると、この認証のブレークを回避できます。

ポーリングメカニズムを理解する

次に、ポーリングが設定されていない認証中のイベントのフローの例を示します。

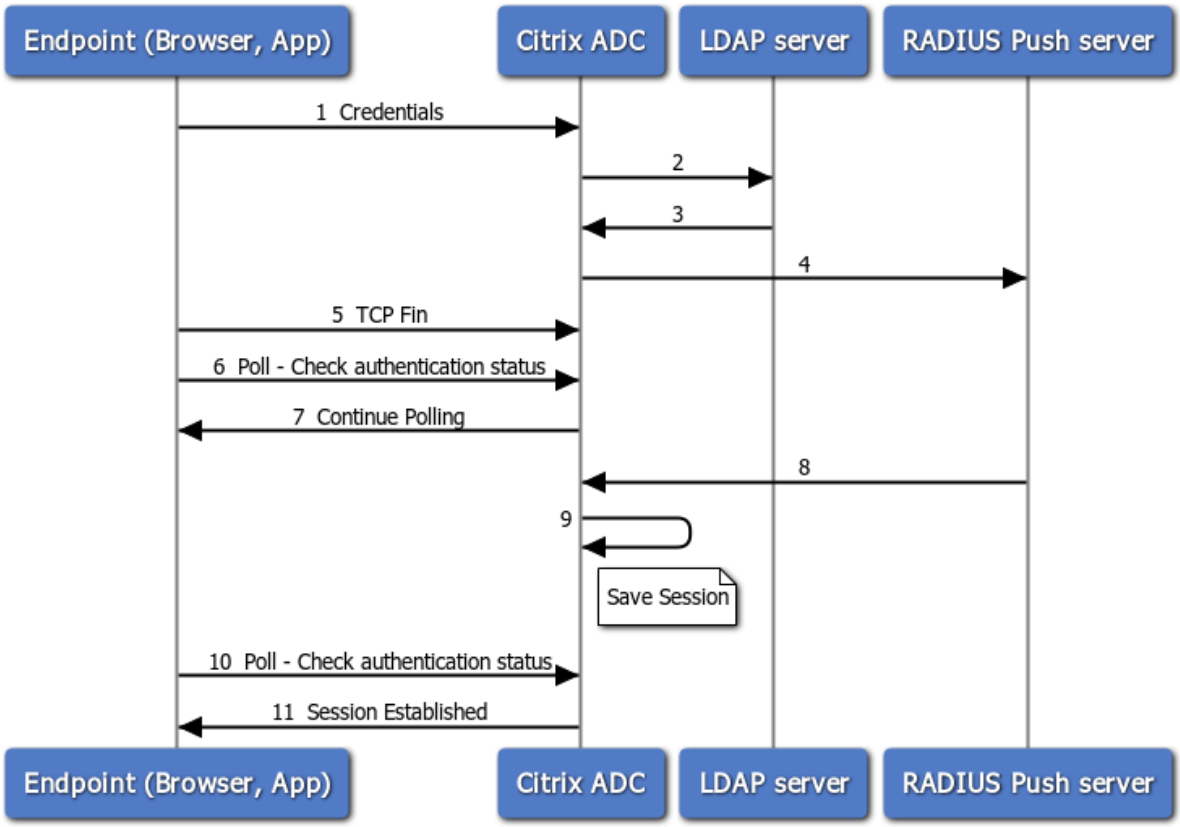
ポーリングメカニズムを使用すると、Citrix ADC アプライアンスは、まれにエンドポイントで TCP 接続がリセットされた場合に、認証プロセスを再起動することなく、エンドポイントでの継続的な認証を再開できます。



1. エンドポイント (アプリまたは Web ブラウザ) は、認証情報を使用して認証します。
2. ユーザー名とパスワードは、既存の第 1 要素ディレクトリ (LDAP/Active Directory) に対して検証されます。
3. 正しいクレデンシャルが指定されている場合、認証は次の要素に移行します。
4. この時点で、Citrix ADC アプライアンスは RADIUS プッシュサーバーに要求を送信します。
5. Citrix ADC アプライアンスが RADIUS サーバーからの応答を待っている間、エンドポイントは TCP 接続をドロップします。

6. Citrix ADC は、RADIUS プッシュサーバーから応答を受信します。
7. クライアント TCP 接続が見つからない場合、Citrix ADC アプライアンスはセッションをドロップし、ログインに失敗します。

次に、ポーリングが設定された認証時のイベントのフローの例を示します。



1. エンドポイント (アプリまたは Web ブラウザ) は、認証情報を使用して認証します。
2. ユーザー名とパスワードは、既存の第 1 要素ディレクトリ (LDAP/Active Directory) に対して検証されます。
3. 正しいクレデンシャルが指定されている場合、認証は次の要素に移行します。
4. この時点で、Citrix ADC アプライアンスは RADIUS プッシュサーバーに要求を送信します。
5. Citrix ADC アプライアンスが RADIUS サーバーからの応答を待っている間、エンドポイントは TCP 接続をドロップします。
6. エンドポイントは Citrix ADC アプライアンスにポーリング（プローブ）を送信し、認証ステータスを確認します。
7. Citrix ADC アプライアンスは RADIUS サーバーから受信しないため、エンドポイントにポーリングの継続を要求します。
8. Citrix ADC アプライアンスは、RADIUS プッシュサーバーから応答を受信します。
9. クライアント TCP 接続が見つからない場合、ADC はセッション状態を保存します。
10. エンドポイントは再びポーリングして、認証ステータスを確認します。
11. Citrix ADC アプライアンスがセッションを確立し、ログインは成功します。

CLI を使用したポーリングの設定

次に、CLI 設定の例を示します。

第 1 要素を構成する

```
add authentication ldapAction ldap-new -serverIP 10.106.40.65 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword 2
    f63d3659103464a4fad0ade65e2ccfd4e8440e36ddff941d29796af03e01139 -
    encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -
    groupAttrName memberof -subAttributeName CN -secType SSL -
    alternateEmailAttr userParameters

add authentication Policy ldap-new -rule true -action ldap-new

bind authentication vserver avs -policy ldap-new -priority 1 -
    nextFactor rad_factor
<!--NeedCopy-->
```

第 2 要素を構成する

```
add authentication radiusAction rad1 -serverIP 10.102.229.120 -radKey 1
    b1613760143ce2371961e9a9eb5392c86a4954a62397f29a01b5d12b42ce232 -
    encrypted -encryptmethod ENCMTHD_3

add authentication Policy rad -rule true -action rad1
<!--NeedCopy-->
```

Poll.xml ログインスキーマを構成する

```
add authentication loginSchema polling_schema -authenticationSchema
    LoginSchema/Poll.xml

add authentication policylabel rad_factor -loginSchema polling_schema

bind authentication policylabel rad_factor -policyName rad -priority 1
    -gotoPriorityExpression NEXT
<!--NeedCopy-->
```


GUI を使用したポーリングの設定

GUI を使用した多要素認証の設定手順の詳細については、nFactor 認証の設定を参照してください。

次に、第 2 要素以降のポーリング用に Citrix ADC を構成するために必要な高レベル手順の例を示します。

1. LDAP など、認証の最初の要素を作成します。
2. RADIUS など、認証の 2 番目の要素を作成します。
3. Citrix ADC（/NSCONFIG/loginSchema/loginSchema/）に存在する **Poll.xml** を 2 番目の要素のログインスキーマとして追加します。

セッションとトラフィックの管理

October 7, 2021

セッションの設定

認証、承認、および監査プロファイルを構成したら、セッション設定を構成してユーザーセッションをカスタマイズします。セッション設定は次のとおりです。

- セッションのタイムアウト。

  ユーザーが自動的に切断され、イントラネットにアクセスするために再度認証が必要になるまでの期間を制御します。

- デフォルトの認可設定。

  Citrix ADC アプライアンスで、特定の認証ポリシーがないコンテンツへのアクセスをデフォルトで許可するか拒否するかを決定します。

- シングルサインオンの設定。

  Citrix ADC アプライアンスがユーザーを認証後にすべての Web アプリケーションに自動的にログオンするか、ユーザーを Web アプリケーションログオンページに渡して各アプリケーションを認証するかを決定します。

- クレデンシャルインデックスの設定。

  Citrix ADC アプライアンスがシングルサインオンにプライマリ認証資格情報とセカンダリ認証資格情報のどちらを使用するかを決定します。

セッション設定を構成するには、2 つの方法のいずれかを使用します。ユーザーアカウントまたはグループごとに異なる設定が必要な場合は、カスタムセッション設定を構成するユーザーアカウントまたはグループごとにプロファイルを作成します。また、特定のプロファイルを適用する接続を選択するポリシーを作成し、そのポリシーをユーザーまたはグループにバインドします。また、プロファイルを適用するトラフィックを処理する認証仮想サーバにポリシーをバインドすることもできます。

すべてのセッションに同じ設定を使用する場合、または特定のプロファイルとポリシーが設定されていないセッションのデフォルト設定をカスタマイズする場合は、単にグローバルセッション設定を構成できます。

セッションプロファイル

ユーザーセッションをカスタマイズするには、まずセッションプロファイルを作成します。セッションプロファイルを使用すると、任意のセッションパラメータのグローバル設定を上書きできます。

> 注
>
> 「セッションプロファイル」と「セッションアクション」という用語は同じことを意味します。

コマンドラインインターフェイスを使用してセッションプロファイルを作成するには

コマンドプロンプトで次のコマンドを入力して、セッションプロファイルを作成し、構成を確認します。

```
add tm sessionAction <name> [-sessTimeout <mins>] [-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
     )] [-persistentCookieValidity <minutes>]

show tm sessionAction <name>
<!--NeedCopy-->
```

例

```
> add tm sessionAction session-profile -sessTimeout 30 -
    defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを変更するには

コマンド・プロンプトで次のコマンドを入力して、セッション・プロファイルを変更し、構成を確認します。


```
set tm sessionAction <name> [-sessTimeout <mins>] [-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
     )] [-persistentCookieValidity <minutes>]

show tm sessionAction
<!--NeedCopy-->
```

例

```
> set tm sessionAction session-profile -sessTimeout 30 -
    defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを削除するには

コマンドプロンプトで次のコマンドを入力して、セッションプロファイルを削除します。

```
rm tm sessionAction <name>
<!--NeedCopy-->
```

構成ユーティリティを使用してセッションプロファイルを構成するには

1. セキュリティ **> AAA-**アプリケーショントラフィック **>** セッションに移動します。
2. セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** セッションに移動します。
3. 詳細ウィンドウで、[プロファイル] タブをクリックします。
4. [プロファイル] タブで、次のいずれかの操作を行います。
   - 新しいセッションプロファイルを作成するには、[追加] をクリックします。
   - 既存のセッションプロファイルを変更するには、プロファイルを選択し、[編集] をクリックします。
5. [TM セッションプロファイルの作成] または [TM セッションプロファイルの設定] ダイアログで、パラメータの値を入力または選択します。

- Name*：actionname（以前に設定されたセッションアクションでは変更できません）
- セッションタイムアウト：sesstimeout
- Web アプリケーションへのシングルサインオン—sso
- デフォルトの認可アクション：defaultAuthorizationAction
- 資格情報インデックス—ssocredential
- シングルサインオンドメイン：ssoDomain
- HTTP のみのクッキー—httpOnlyCookie
- パーシステント Cookie の有効化-persistentCookie
- 永続的クッキーの有効性-persistentCookieValidity

6. [作成] または [**OK]** をクリックします。作成したセッションプロファイルが [セッションポリシーおよびプロファイル] ペインに表示されます。

セッション・ポリシー

1 つ以上のセッションプロファイルを作成したら、セッションポリシーを作成し、ポリシーをグローバルにバインドするか、認証仮想サーバーにバインドして有効にします。

コマンドラインインターフェイスを使用してセッションポリシーを作成するには

コマンドプロンプトで次のコマンドを入力して、セッションポリシーを作成し、構成を確認します。

```
-  add tm sessionPolicy <name> <rule> <action>
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

例

```
> add tm sessionPolicy session-pol ”URL == /\*.png” session-profile
Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/\*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーを変更するには

コマンド・プロンプトで次のコマンドを入力して、セッション・ポリシーを変更し、構成を確認します。


```
-  set tm sessionPolicy <name> [-rule <expression>] [-action <action>]
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

例

```
> set tm sessionPolicy session-pol ”URL == /\*.png” session-profile
 Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/\*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーをグローバルにバインドするには

コマンドプロンプトで次のコマンドを入力して、セッションポリシーをグローバルにバインドし、構成を確認します。

```
bind tm global -policyName <policyname> [-priority <priority>]
<!--NeedCopy-->
```

例

```
> bind tm global -policyName session-pol
 Done

> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.png'
        Action: session-profile
        Policy is bound to following entities
        1) TM GLOBAL    PRIORITY : 0
 Done

<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーを認証仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、セッションポリシーを認証仮想にバインドし、構成を確認します。


```
bind authentication vserver <name> -policy <policyname> [-priority <
    priority>]
<!--NeedCopy-->
```

例

```
bind authentication vserver auth-vserver-1 -policyName Session-Pol-1 -
    priority 1000
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して認証仮想サーバーからセッションポリシーをバインド解除するには

コマンドプロンプトで次のコマンドを入力して、認証仮想サーバーからセッションポリシーをバインド解除し、構成を確認します。

```
unbind authentication vserver <name> -policy <policyname>
<!--NeedCopy-->
```

例

```
unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してグローバルにバインドされたセッションポリシーをバインド解除するには

コマンドプロンプトで次のコマンドを入力して、グローバルにバインドされたセッションポリシーのバインドを解除します。

```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```


例

```
unbind tm global -policyName Session-Pol-1
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションポリシーを削除するには

まず、グローバルからセッションポリシーをバインド解除し、コマンドプロンプトで次のコマンドを入力してセッションポリシーを削除し、構成を確認します。

```
rm tm sessionPolicy <name>
<!--NeedCopy-->
```

例

```
rm tm sessionPolicy Session-Pol-1
Done

<!--NeedCopy-->
```

構成ユーティリティを使用してセッションポリシーを構成およびバインドするには

1. **Security > AAA - Application Traffic > Session** に移動します。
2. **Security > AAA - Application Traffic > Policies > Session** に移動します。
3. 詳細ウィンドウの [ポリシー] タブで、次のいずれかの操作を行います。
   - 新しいセッションポリシーを作成するには、[追加] をクリックします。
   - 既存のセッションポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。
4. パラメータの値を入力するか選択し、セッションポリシーの設定やセッションポリシーの作成] ダイアログボックスで。
   - Name*: policyname（以前に設定されたセッションポリシーでは変更できません）
   - 要求プロファイル *—actionname
   - 式 *—rule (式を入力するには、まず [式] テキスト領域の下の左端のドロップダウンリストで式のタイプを選択し、式テキスト領域に直接式を入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、ダウンリストを使用して式を作成します)。
5. [作成] または [**OK]** をクリックします。作成したポリシーが、[セッションポリシーおよび プロファイル] ページの詳細ペインに表示されます。

6. セッションポリシーをグローバルにバインドするには、詳細ペインで、[**Action**] ドロップダウンリストから [**Global Bindings**] を選択し、ダイアログに入力します。
   - グローバルにバインドするセッションポリシーの名前を選択します。
   - [**OK**] をクリックします。
7. セッションポリシーを認証仮想サーバにバインドするには、ナビゲーションペインで [**Virtual Servers**] をクリックし、そのポリシーをポリシーリストに追加します。
   - 詳細ウィンドウで、仮想サーバーを選択し、[編集] をクリックします。
   - 詳細領域の右側にある **[**詳細選択] で、[ポリシー] をクリックします。
   - ポリシーを選択するか、プラスアイコンをクリックしてポリシーを追加します。
   - 左側の **[Priority]** 列で、デフォルトの優先度を変更し、ポリシーが適切な順序で評価されるようにします。
   - [**OK**] をクリックします。
     ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

グローバルセッション設定

セッションプロファイルおよびポリシーを作成する代わりに、グローバルセッション設定を構成できます。これらの設定は、セッション構成を上書きする明示的なポリシーがない場合に、セッション構成を制御します。

コマンドラインインターフェイスを使用してセッション設定を構成するには

コマンドプロンプトで次のコマンドを入力して、グローバルセッション設定を構成し、構成を確認します。

```
set tm sessionParameter [-sessTimeout <mins>][-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )][-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )][-persistentCookie ( ENABLED | DISABLED
    )] [-persistentCookieValidity <minutes>]
<!--NeedCopy-->
```

例

```
> set tm sessionParameter -sessTimeout 30
  Done
> set tm sessionParameter -defaultAuthorizationAction DENY
  Done
> set tm sessionParameter -SSO ON
  Done
> set tm sessionParameter -ssoCredential PRIMARY
  Done
```



```
<!--NeedCopy-->
```

構成ユーティリティを使用してセッション設定を構成するには

1. [セキュリティ**] > [AAA-**アプリケーショントラフィック] に移動します。
2. 詳細ウィンドウの [設定] で、[グローバル設定の変更] をクリックします。
3. [グローバルセッション設定] ダイアログで、パラメータの値を入力または選択します。
    - セッション・タイムアウト：sessTimeout
    - デフォルトの認可アクション：defaultAuthorizationAction
    - Web アプリケーションへのシングル・サインオン：sso
    - クレデンシャルインデックス-ssoCredential
    - シングルサインオンドメイン：ssoDomain
    - HTTP のみのクッキー—httpOnlyCookie
    - パーシステント Cookie の有効化-persistentCookie
    - 永続的クッキーの有効性 (分)-persistentCookieValidity
    - ホームページ-ホームページ
4. [**OK**] をクリックします。

トラフィックの設定

保護されたアプリケーションにフォームベースまたは SAML シングルサインオン (SSO) を使用する場合は、トラフィック設定でこの機能を構成します。SSO を使用すると、ユーザーは 1 回ログオンして、保護されたすべてのアプリケーションにアクセスできます。各アプリケーションにアクセスするために個別にログオンする必要はありません。

フォームベースの SSO を使用すると、一般的なポップアップウィンドウではなく、独自のデザインの Web フォームをサインオン方法として使用できます。したがって、ログオンフォームに、ユーザーに表示してほしい会社のロゴやその他の情報を入力できます。SAML SSO を使用すると、1 つの Citrix ADC アプライアンスまたは仮想アプライアンスのインスタンスを構成して、最初のアプライアンスで認証されたユーザーの代わりに別の Citrix ADC アプライアンスに対して認証を行うことができます。

いずれかのタイプの SSO を設定するには、まずフォームまたは SAML SSO プロファイルを作成します。次に、トラフィックプロファイルを作成し、作成した SSO プロファイルにリンクします。次に、ポリシーを作成し、トラフィックプロファイルにリンクします。最後に、ポリシーをグローバルにバインドするか、認証仮想サーバーにバインドして、設定を有効にします。

トラフィックプロファイル

少なくとも 1 つのフォームまたは SAML sso プロファイルを作成したら、次にトラフィックプロファイルを作成する必要があります。

> 注:
>
> この機能では、「プロファイル」と「アクション」という用語は同じことを意味します。

コマンドラインインターフェイスを使用してトラフィックプロファイルを作成するには

コマンドプロンプトで入力します。

```
add tm trafficAction <name> [-appTimeout <mins>][-SSO ( ON | OFF ) [-
    formSSOAction <string>]][-persistentCookie ( ENABLED | DISABLED )][-
    InitiateLogout ( ON | OFF )]
<!--NeedCopy-->
```

例

```
add tm trafficAction Traffic-Prof-1  - appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを変更するには

コマンドプロンプトで入力します。

```
set tm trafficAction <name> [-appTimeout <mins>] [-SSO ( ON | OFF ) [-
    formSSOAction <string>]] [-persistentCookie ( ENABLED | DISABLED )]
    [-InitiateLogout ( ON | OFF )]
<!--NeedCopy-->
```

例

```
set tm trafficAction Traffic-Prof-1  - appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセッションプロファイルを削除するには

コマンドプロンプトで入力します。


```
rm tm trafficAction <name>
<!--NeedCopy-->
```

例

```
rm tm trafficAction Traffic-Prof-1
<!--NeedCopy-->
```

構成ユーティリティを使用してトラフィックプロファイルを構成するには

1. [セキュリティ] **>** **[AAA-**アプリケーショントラフィック**]** **>** **[**トラフィック**]** に移動します。
2. セキュリティ **> AAA-**アプリケーショントラフィック **>** ポリシー **>** トラフィックに移動します。
3. 詳細ウィンドウで、[プロファイル] タブをクリックします。
4. [プロファイル] タブで、次のいずれかの操作を行います。
   - 新しいトラフィックプロファイルを作成するには、[追加] をクリックします。
   - 既存のトラフィックプロファイルを変更するには、プロファイルを選択し、[**Edit**] をクリックします。
5. [トラフィックプロファイルの作成] または [トラフィックプロファイルの設定] ダイアログボックスで、パラメータの値を指定します。
   - Name*—name（以前に設定したセッションアクションでは変更できません）
   - アプリケーションタイムアウト-appTimeout
   - シングル・サインオン：SSO
   - フォーム SSO アクション — formSSOAction
   - SAML SSO アクション-samlSSOAction
   - パーシステント Cookie の有効化-persistentCookie
   - ログアウトの開始：InitiateLogout
6. [作成**]** または [**OK]** をクリックします。作成したトラフィックプロファイルは、必要に応じて [トラフィックポリシー]、[プロファイル]、[フォーム SSO プロファイル] または [SAML SSO プロファイル] ペインに表示されます。

AAA.USER および **AAA.LOGIN** 式のサポート

AAA.USER 式は、既存の HTTP.REQ.USER 式を置き換えるために実装されました。AAA.USER 式は、Secure Web Gateway（SWG）やロールベースアクセス（RBA）メカニズムなどの非 HTTP トラフィックの処理に適用できます。AAA.USER 式は、HTTP.REQ.USER 式と同等です。

式は、さまざまなアクションまたはプロファイル構成で使用できます。

コマンドプロンプトで入力します。


```
add tm trafficAction <name> [SSO (ON|OFF)] [-userExpression <string>]

add tm trafficAction <name> [SSO (ON|OFF)] [-passwdExpression <string>]

<!--NeedCopy-->
```

例

```
add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.NAME"

add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.PASSWD"

add tm trafficPolicy tm_pol true tm_act

bind lb vserver lb1 -policyName tm_pol -priority 2
<!--NeedCopy-->
```

> 注:
>
> HTTP.REQ.USER 式を使用すると、警告メッセージ「HTTP.REQ.USER は廃止されました。代わりに AAA.USER を使用する」とコマンドプロンプトが表示されます。

- **AAA.LOGIN Expression.** LOGIN 式は、ログイン要求とも呼ばれるプレログインを表します。ログイン要求は、Citrix Gateway、SAML IdP、または OAuth 認証から行うことができます。Citrix ADC は、ポリシー構成から必要な属性を抽象化します。AAA.LOGIN 式には、次の条件に基づいて取得できる属性が含まれています。
  - **AAA.LOGIN.USERNAME.** ユーザー名（見つかった場合）は、現在のログイン要求から取得されます。ログイン以外の要求（認証、承認、および監査によって決定される）に適用された同じ式は、空の文字列になります。
  - **AAA.LOGIN.PASSWORD.** ユーザーパスワード（見つかった場合）は、現在のログイン要求から取得されます。パスワードが見つからない場合、式は空の文字列になります。
  - **AAA.LOGIN.PASSWORD2.** 2 番目のパスワード（見つかった場合）は、ログイン要求から取得されます。
  - **AAA.LOGIN.DOMAIN.** ドメイン情報は、ログイン要求からフェッチされます。
- **AAA.USER.ATTRIBUTE (「#」).** この式は、ユーザー属性を格納するために使用されます。ここで # は、整数 (1 から 16) または文字列値のいずれかになります。これらのインデックス値は、式 AAA.USER.ATTRIBUTE (「#」) を使用して使用できます。認証、承認、および監査モジュールはユーザーセッション属性を検索し、`AAA.USER.ATTRIBUTE("##")` はその特定の属性についてハッシュテーブルにクエリを実行します。たとえば、`Attributes("samaccountname")` が設定されている場合、`AAA.USER.ATTRIBUTE("samaccountname")` はハッシュマップを照会し、`samaccountname` に対応する値を取得します。

トラフィックポリシー

1 つ以上のフォーム SSO およびトラフィックプロファイルを作成したら、トラフィックポリシーを作成し、グローバルまたはトラフィック管理仮想サーバにポリシーをバインドして有効にします。

コマンドラインインターフェイスを使用してトラフィックポリシーを作成するには

コマンドプロンプトで入力します。

```
add tm trafficPolicy <name> <rule> <action>
<!--NeedCopy-->
```

例

```
add tm trafficPolicy Traffic-Pol-1 ”HTTP.REQ.HEADER(”Cookie”).CONTAINS(
   ”login=true”)” Traffic-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーを変更するには

コマンドプロンプトで入力します。

```
set tm trafficPolicy <name> <rule> <action>
<!--NeedCopy-->
```

例

```
set tm trafficPolicy Traffic-Pol-1 ”HTTP.REQ.HEADER(”Cookie”).CONTAINS(
   ”login=true”)” Traffic-Prof-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーをグローバルにバインドするには

コマンドプロンプトで入力します。


```
bind tm global -policyName <string> [-priority <priority>]
<!--NeedCopy-->
```

例

```
bind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーを負荷分散またはコンテンツスイッチング仮想サーバーにバインドするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
bind lb vserver <name> -policy <policyName> [-priority <priority>]

bind cs vserver <name> -policy <policyName> [-priority <priority>]
<!--NeedCopy-->
```

例

```
bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -
   priority 1000
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してグローバルにバインドされたトラフィックポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```

例


```
unbind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、負荷分散またはコンテンツスイッチング仮想サーバーからトラフィックポリシーをバインド解除するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
unbind lb vserver <name> -policy <policyname>

unbind cs vserver <name> -policy <policyname>
<!--NeedCopy-->
```

例

```
unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してトラフィックポリシーを削除するには

最初に global からセッションポリシーをバインド解除し、コマンドプロンプトで次のように入力します。

```
rm tm trafficPolicy <name>
<!--NeedCopy-->
```

例

```
rm tm trafficPolicy Traffic-Pol-1
<!--NeedCopy-->
```

構成ユーティリティを使用してトラフィックポリシーを構成およびバインドするには

1. [セキュリティ] **>** **[AAA-**アプリケーショントラフィック**]** **>** **[**トラフィック**]** に移動します。
2. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** ポリシー **>** トラフィックに移動します。
3. 詳細ウィンドウで、次のいずれかの操作を行います。

- 新しいセッションポリシーを作成するには、[追加] をクリックします。
- 既存のセッションポリシーを変更するには、ポリシーを選択し、[編集] をクリックします。

4. [トラフィックポリシーの作成] または [トラフィックポリシーの設定] ダイアログで、パラメータの値を指定します。
   - Name*: policyName（以前に設定されたセッションポリシーでは変更できません）
   - プロファイル *—actionName
   - 式-規則 (式を入力するには、まず [式] テキスト領域の下の左端のドロップダウンリストで式のタイプを選択し、式テキスト領域に直接式を入力するか、[追加] をクリックして [式の追加] ダイアログボックスを開き、その中のドロップダウンリストを使用して式を構築します)。
5. [作成] または [**OK**] をクリックします。作成したポリシーが、[セッションポリシーおよび プロファイル] ページの詳細ペインに表示されます。

フォーム **SSO** プロファイル

フォームベースの SSO を有効にして構成するには、最初に SSO プロファイルを作成します。

> 注
>
> - フォームが Javascript を含むようにカスタマイズされている場合、フォームベースのシングルサインオンは機能しません。
> - この機能では、「プロファイル」と「アクション」という用語は同じことを意味します。

コマンドラインインターフェイスを使用してフォーム **SSO** プロファイルを作成するには

コマンドプロンプトで入力します。

```
add tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]

show tm formSSOAction [<name>]
<!--NeedCopy-->
```

例

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
```



```
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してフォーム **SSO** を変更するには

コマンドプロンプトで入力します。

```
set tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]
<!--NeedCopy-->
```

例

```
set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してフォーム **SSO** プロファイルを削除するには

コマンドプロンプトで入力します。

```
rm tm formSSOAction <name>
<!--NeedCopy-->
```

例

```
rm tm sessionAction SSO-Prof-1
<!--NeedCopy-->
```


構成ユーティリティを使用してフォーム **SSO** プロファイルを構成するには

1. **Security > AAA - Application Traffic > Policies > Traffic** に移動します。
2. 詳細ウィンドウで、[フォーム **SSO** プロファイル] タブをクリックします。
3. [フォーム SSO プロファイル] タブで、次のいずれかの操作を行います。
   - 新しいフォーム SSO プロファイルを作成するには、[追加] をクリックします。
   - 既存のフォーム SSO プロファイルを変更するには、プロファイルを選択し、[編集] をクリックします。
4. 作成フォーム **SSO** プロファイルまたは 構成フォーム **SSO** プロファイル] ダイアログボックスで、パラメータの値を指定します。
   - Name*—name（以前に設定したセッションアクションでは変更できません）
   - アクション URL * — actionURL
   - ユーザー名フィールド *: userField
   - パスワードフィールド *—passField
   - 式 *—ssoSuccessRule
   - 名前値のペア-nameValuePair
   - 応答サイズ: responsesize
   - 抽出-nvtype
   - 送信方法-submitMethod
5. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。作成したフォーム SSO プロファイルが、[トラフィックポリシー]、プロファイル、および [フォーム **SSO** プロファイル] ペインに表示されます。

SAML SSO プロファイル

SAML ベースの SSO を有効にして設定するには、最初に SAML SSO プロファイルを作成します。

コマンドラインインターフェイスを使用して **SAML SSO** プロファイルを作成するには

コマンドプロンプトで入力します。

```
1 add tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
2 <!--NeedCopy-->
```

例

```
1 add tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
```



```
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SAML SSO** を変更するには

コマンドプロンプトで入力します。

```
set tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
<!--NeedCopy-->
```

例

```
set tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName ”Example,
    Inc.”  -assertionConsumerServiceURL ”https://service.example.com” -
    relaystateRule ”true”  -sendPassword ”ON” -samlIssuerName ”Example,
    Inc.”
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SAML SSO** プロファイルを削除するには

コマンドプロンプトで入力します。

```
rm tm samlSSOProfile <name>
<!--NeedCopy-->
```

例

```
rm tm sessionAction saml-SSO-Prof-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **SAML SSO** プロファイルを構成するには

1. セキュリティ **>** **AAA-**アプリケーショントラフィック **>** ポリシー **>** トラフィックに移動します。
2. 詳細ペインで、[**SAML SSO** プロファイル] タブをクリックします。
3. [**SAML SSO** プロファイル] タブで、次のいずれかの操作を行います。

- 新しい SAML SSO プロファイルを作成するには、[追加] をクリックします。
- 既存の SAML SSO プロファイルを変更するには、プロファイルを選択し、[**OpenEdit**] をクリックします。

4. [**SAML SSO** プロファイルの作成] ダイアログボックスまたは [**SAML SSO** プロファイルの構成] ダイアログボックスで、次のパラメータを設定します。
   - Name*
   - 署名証明書名 *
   - ACS URL*
   - リレー状態ルール *
   - パスワードを送信
   - 発行者名
5. [作成] または [**OK**] をクリックし、[閉じる] をクリックします。作成した SAML SSO プロファイルが、[トラフィックポリシー、プロファイル、および SAML SSO プロファイル] ペインに表示されます。

OWA 2010 のセッションタイムアウト

非アクティブの指定された期間後にタイムアウトに OWA 2010 接続を強制できるようになりました。OWA は、タイムアウトを防ぐために、繰り返しキープアライブ要求をサーバーに送信します。接続を開いたままにしておくと、シングルサインオンが妨げられることがあります。

コマンドラインインターフェイスを使用して、**OWA2010** を指定された期間後に強制的にタイムアウトさせるには

コマンドプロンプトで、次のコマンドを入力します。

```
add tm trafficAction <actname> [-forcedTimeout <forcedTimeout> -
    forcedTimeoutVal <mins>]
<!--NeedCopy-->
```

<actname> では、トラフィックポリシーの名前に置き換えます。<mins> には、強制タイムアウトを開始するまでの時間（分単位）を代入します。<forcedTimeout> には、次のいずれかの値を置き換えます。

–START —タイマーがまだ開始されていない場合、強制タイムアウトのためにタイマーを開始します。実行中のタイマーが存在する場合、効果はありません。
–STOP —実行中のタイマーを停止します。実行中のタイマーが見つからない場合、は効果がありません。
–RESET —実行中のタイマーを再起動します。実行中のタイマーが見つからない場合は、START オプションが使用されたかのようにタイマーを開始します。

```
add tm trafficPolicy <polname> <rule> <actname>
<!--NeedCopy-->
```


<polname> では、トラフィックポリシーの名前に置き換えます。<rule> の場合は、Citrix ADC デフォルト構文でルールを置き換えます。

```
bind lb vserver <vservername>  - policyName <name> -priority <number>
<!--NeedCopy-->
```

<vservername> には、認証、認可、および監査トラフィック管理仮想サーバの名前を置き換えます。<priority> には、ポリシーのプライオリティを指定する整数を代入します。

例

```
add tm trafficAction act-owa2010timeout -forcedTimeout RESET -
    forcedTimeoutVal 10
add tm trafficPolicy pol-owa2010timeout true act-owa2010timeout
bind lb vserver vs-owa2010 -policyName pol-owa2010timeout -priority 10
<!--NeedCopy-->
```

Citrix Gateway のレート制限

March 8, 2022

Citrix Gateway のレート制限機能を使用すると、Citrix Gateway アプライアンス上の特定のネットワークエンティティまたは仮想エンティティの最大負荷を定義できます。Citrix Gateway アプライアンスはすべての非認証トラフィックを消費するため、アプライアンスは多くの場合、高いレートでプロセス要求にさらされます。レート制限機能を使用すると、エンティティに関連付けられたトラフィックのレートを監視し、トラフィックレートに基づいてリアルタイムで予防措置を実行するように Citrix Gateway アプライアンスを設定できます。Citrix ADC アプライアンスでのレート制限の仕組みの詳細については、「レート制限」を参照してください。

Citrix ADC には、予期しない速度でバックエンドサーバーを保護するレート制限機能があります。Citrix ADC の機能は Citrix Gateway が処理する認証されていないトラフィックを処理しなかったため、Citrix Gateway には独自のレート制限機能が必要でした。これは、Citrix Gateway アプライアンスが公開されているさまざまなソースからの予期しない要求率をチェックするために必要です。たとえば、認証されていない/ログイン/制御要求や、エンドユーザーまたはデバイスの検証のために公開される特定の API。

レート制限の一般的なユースケース

- URL からの 1 秒あたりのリクエスト数を制限します。

- リクエストがレート制限を超えた場合、特定のホストからのリクエストで受信した Cookie に基づいて接続をドロップします。
- 同じホスト（特定のサブネットマスク）から到着し、同じ宛先 IP アドレスを持つ HTTP 要求の数を制限します。

Citrix Gateway のレート制限を構成する

前提条件

設定済みの認証仮想サーバー。

注意事項

- 設定手順では、サンプルの制限 ID が設定されます。ストリームセレクタ、モードなどのサポートされているすべてのパラメータで同じように設定することができます。レート制限機能の完全な説明については、レート制限を参照してください。
- このポリシーは、次のように VPN 仮想サーバーにバインドすることもできます。次のコマンドを使用してポリシーをバインドするには、設定された VPN 仮想サーバーが必要です。

```
1 bind vpn vserver  -policy denylogin  - pri 1  - type aaa_request
2 <!--NeedCopy-->
```

- AAA_REQUEST は、レスポンダーポリシー用に新しく導入されたバインドポイントです。このバインドポイントで構成されたポリシーは、指定された仮想サーバーのすべての着信要求に適用されます。ポリシーは、他の処理の前に最初に非認証/制御トラフィックに対して処理されます。
- ポリシーを Citrix Gateway 仮想サーバーにバインドすると、認証されていない要求を含む、Citrix Gateway によって消費されるすべてのトラフィックの AAA_REQUEST バインドポイントでレート制限が有効になります。
- ポリシーを認証仮想サーバレートにバインドすると、認証仮想サーバにヒットする非認証/制御要求が制限されます。

コマンドラインインターフェイスを使用してレート制限を構成するには、コマンドプロンプトで次のコマンドを入力します。

```
1 add limitIdentifier <limitIdentifier name> -threshold <positive_integer
     > -timeslice <positive_integer> -mode <mode type>
2 <!--NeedCopy-->
```


例:

```
add limitIdentifier limit_one_login -threshold 10 -timeslice 4294967290
    -mode REQUEST_RATE
<!--NeedCopy-->
```

```
add responderaction denylogin respondwith  ‘ “HTTP/1.1 200 OK\r\n\r\n”
    +  “Request is denied due to unusual rate” ’
<!--NeedCopy-->
```

```
add responder policy denylogin 'sys.check_limit(”limit_one_login”)'
   denylogin
<!--NeedCopy-->
```

```
bind authentication vserver <vserver name> -policy denylogin  -pri 1  -
   type aaa_request
<!--NeedCopy-->
```

例:

```
bind authentication vserver authvserver -policy denylogin  -pri 1  -
   type aaa_request
<!--NeedCopy-->
```

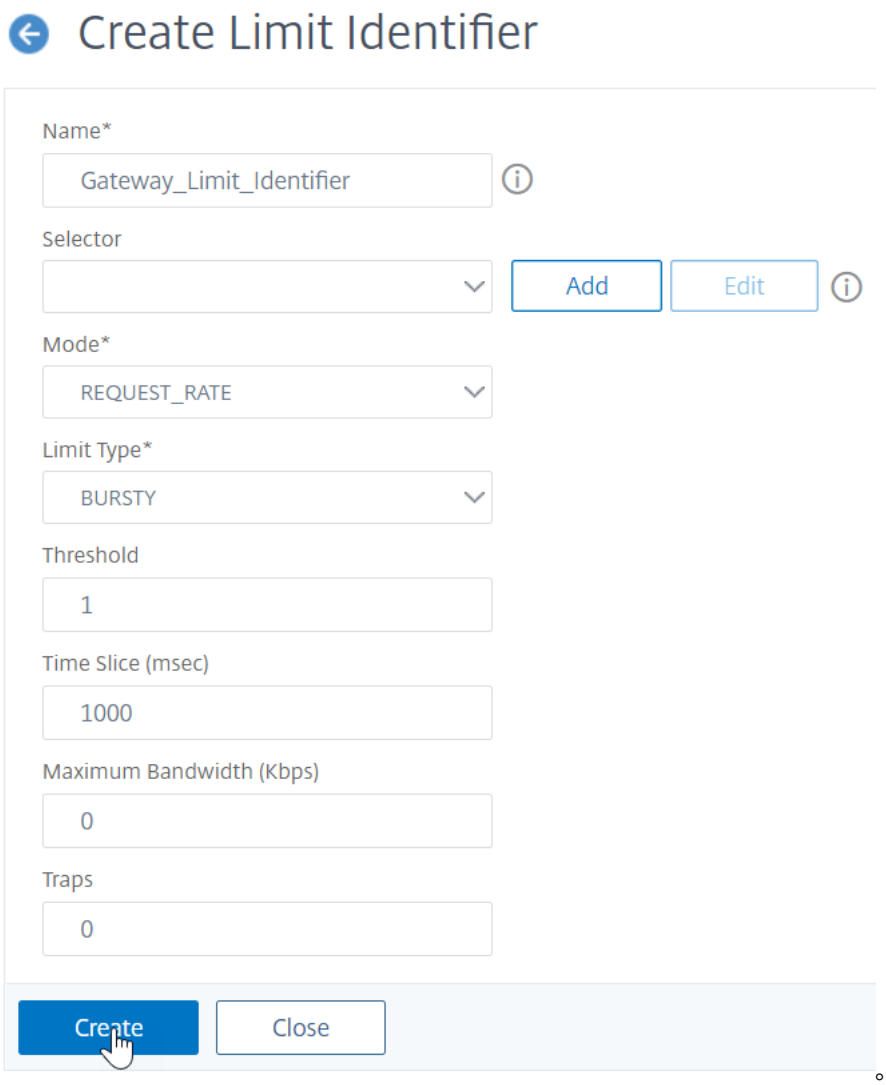
パラメータの説明

- **limitIdentifier**: レート制限識別子の名前。ASCII 文字またはアンダースコア (_) 文字で始まり、ASCII 英数字またはアンダースコア文字のみで構成する必要があります。予約語は使用できません。これは必須の議論です。最大長: 31
- **threshold-** 要求 (モードが REQUEST_RATE として設定されている) がタイムスライスごとに追跡される場合に、指定されたタイムスライスで許可される要求の最大数。接続 (モードが CONNECTION として設定されている) が追跡される場合、通過する接続の合計数になります。デフォルト値:1 最小値:1 最大値:4294967295

- **timeSlice-** 10 の倍数で指定された時間間隔（ミリ秒単位）。この間は、要求がしきい値を超えるかどうかをチェックするために追跡されます。引数は、モードが REQUEST_RATE に設定されている場合にのみ必要です。デフォルト値:1000 最小値:10 最大値:4294967295
- **mode-** 追跡するトラフィックのタイプを定義します。
  - REQUEST_RATE-リクエスト/タイムスライスを追跡します。
  - CONNECTION-アクティブなトランザクションを追跡します。

Citrix ADC GUI を使用してレート制限を構成するには:

1. 「**AppExpert**」**>**「レート制限」**>**「制限識別子」の順に選択し、「追加」をクリックして、CLI セクションで指定されている関連の詳細を指定します。



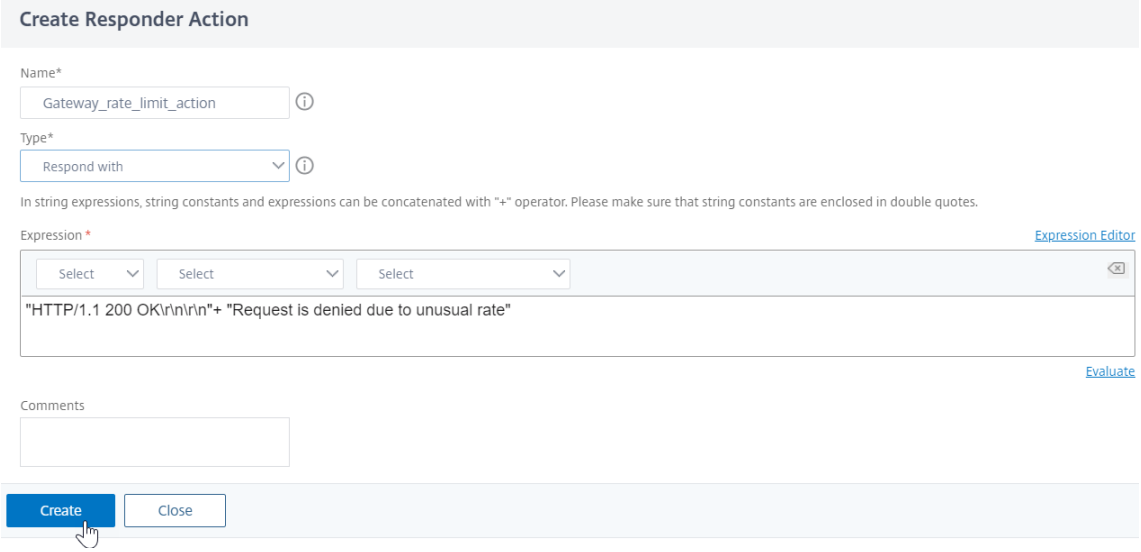
。

2. [**AppExpert] > [**レスポンダー**] > [**ポリシー**]** に移動します。[レスポンダーポリシー] ページで、[追加] をクリックします。
3. **[**レスポンダーポリシーの作成] ページで、制限識別子を持つレスポンダーアクションを含むレスポンダーポリ

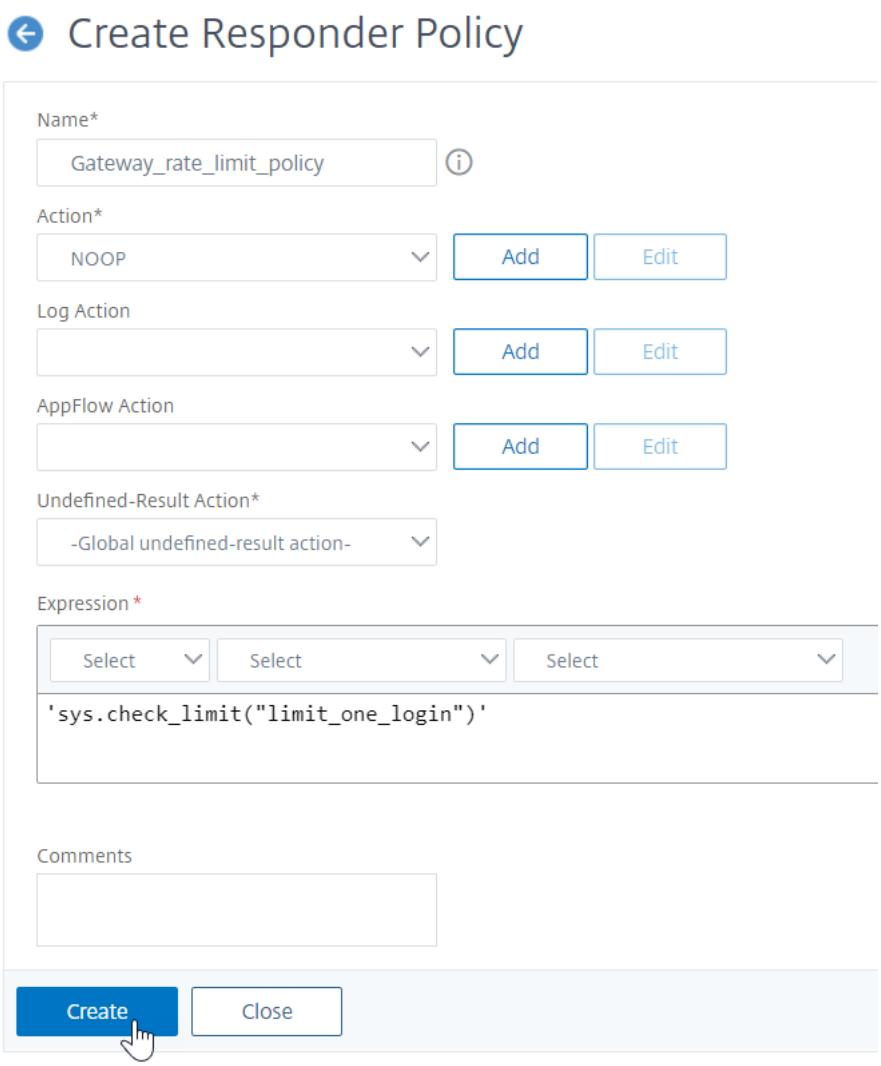
シーを作成します。

4. レスポンダーアクションを作成するには、**[**アクション**]** の横にある **[**** 追加] をクリックし、レスポンダーアクションの名前を入力します。******

5. ドロップダウンメニューからタイプとして応答を選択し、次の式”HTTP/1.1 200 OK\ r\ n\ r\ n” +「要求は異常なレートのために拒否されました」を指定して、「作成」をクリックします。

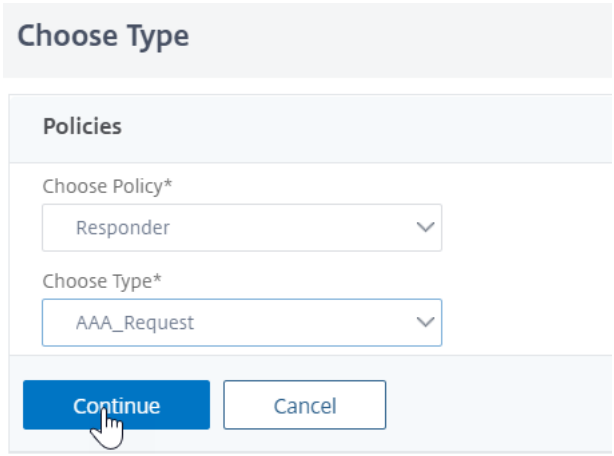


6. レスポンダーポリシーを作成するには、[レスポンダーポリシーの作成] ページで、レスポンダーポリシーの名前を入力し、次の式「sys.check_limit (「limit_one_login」)」を指定して、[作成] をクリックします。



7. レスポンダーポリシーを認証仮想サーバーにバインドします。

   - [セキュリティ] **> [AAA** アプリケーショントラフィック**] > [**仮想サーバー] に移動します。
   - 仮想サーバを選択します。
   - ポリシーを追加します。
   - サーバーにバインドするレスポンダーポリシーを選択し、優先度を設定します。
   - タイプとして **AAA-REQUEST** を選択し、[続行] をクリックします。



> 注: VPN 仮想サーバの AAA_REQUEST バインドポイントでレート制限を有効にすることもできます。

Citrix Gateway にレート制限を適用するための一般的なユースケースの構成

次に、一般的なユースケースを設定するコマンドの例を示します。

- URL からの 1 秒あたりのリクエスト数を制限します。

```
1 add stream selector ipStreamSelector http.req.url “client.ip.src
    ”
2
3 add ns limitIdentifier ipLimitIdentifier -threshold 4 -
    timeslice 1000 -mode request_rate -limitType smooth -
    selectorName ip StreamSelector
4
5 add responder policy ipLimitResponderPolicy “http.req.url.
    contains(\” myasp.asp\” ) && sys.check_limit(\”
    ipLimitIdentifier\” )” myWebSiteRedirectAction
6
7 bind authentication virtual server authvserver -policy denylogin
    -pri 1 -type aaa_request
8 <!--NeedCopy-->
```

- リクエストがレート制限を超えた場合、www.yourcompany.com からのリクエストで受け取ったクッキーに基づいて接続をドロップします。

```
1 add stream selector cacheStreamSelector “http.req.cookie.value(\
    ” mycookie\” )” “client.ip.src.subnet(24)”
2
```



```
add ns limitIdentifier myLimitIdentifier -Threshold 2 -
    timeSlice 3000 -selectorName reqCookieStreamSelector

add responder action sendRedirectURL redirect `"http://www.
    mycompany.com"` + http.req.url' -bypassSafetyCheck Yes

add responder policy rateLimitCookiePolicy

 "http.req.url.contains(\www.yourcompany.com) && sys.check_limit
    (\" myLimitIdentifier\" )"  sendRedirectUrl

<!--NeedCopy-->
```

- 同じホスト（サブネットマスクが 32）から着信し、同じ宛先 IP アドレスを持つ HTTP 要求の数を制限します。

```
add stream selector ipv6_sel  "CLIENT.IPv6.src.subnet(32)" CLIENT
    .IPv6.dst

add ns limitIdentifier ipv6_id -imeSlice 20000 -selectorName
    ipv6_sel

add lb vserver ipv6_vip HTTP 3ffe:: 209 80 -persistenceType NONE
     -cltTime

add responder action redirect_page redirect  "\" `http://
    redirectpage.com/\" "`

add responder policy ipv6_resp_pol  "SYS.CHECK_LIMIT(\" ipv6_id\
    " )"  redirect_page

bind responder global ipv6_resp_pol 5 END -type DEFAULT
<!--NeedCopy-->
```

アプリケーションリソースへのユーザーアクセスの許可

October 7, 2021

認証されたユーザーがアプリケーション内でアクセスできるリソースを制御できます。

これを行うには、認可ポリシーを各ユーザーに関連付けて、個別に、またはポリシーをユーザーのグループに関連付けます。認可ポリシーでは、次の項目を指定する必要があります。

- **Rule.** アクセスを許可する必要があるリソース。これは、基本式または高度な式を使用して指定できます。
- アクション。リソースへのアクセスを許可または拒否する必要があるかどうか。

デフォルトでは、アプリケーション内のすべてのリソースへのアクセスは、すべてのユーザーに対して 拒否されます。ただし、このデフォルトの認可アクションを [すべてのユーザーにアクセスを 許可する] に変更できます（セッションプロファイルでセッションパラメータを設定するか、グローバルセッションパラメータを設定することによって）。

> 警告
>
> 最適なセキュリティを確保するために、デフォルトの認証操作を「拒否」から「許可」に変更しないことをお勧めします。代わりに、特定のリソースにアクセスする必要があるユーザーに対して、特定の承認ポリシーを作成することをお勧めします。

CLI を使用して認可を設定するには

1. 認可ポリシーを設定します。

   ns-cli-prompt> **add authorization policy <name> <rule> <action>**

2. ポリシーを適切なユーザーまたはグループに関連付けます。

   - ポリシーを特定のユーザーにバインドします。

     ns-cli-prompt> **bind aaa user <username> -policy** <policyname>

   - ポリシーを特定のグループにバインドします。

     ns-cli-prompt> **bind aaa group <groupName> -policy** <policyname>

GUI を使用して承認を構成するには ([構成] タブ)

1. 認可ポリシーを作成します。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**認可] に移動し、[追加] をクリックして、必要に応じてポリシーを定義します。

2. ポリシーを適切なユーザーまたはグループに関連付けます。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ユーザまたは グループ] に移動し、関連するユーザまたはグループを編集して認可ポリシーに関連付けます。

認証設定の例

次に、一部のアプリケーションリソースへのユーザーアクセスを許可する設定例を示します。これらは CLI コマンドであることに注意してください。GUI を使用して同様の設定を行うことができますが、式を引用符 (“) で囲むことはできません。

- `add authorization policy authzpol1 ”HTTP.REQ.URL.SUFFIX.EQ(\”gif\”)” ALLOW<!--NeedCopy-->`

- `bind aaa user user1 -policy authzpol1<!--NeedCopy-->`
- `add authorization policy authzpol2 ”HTTP.REQ.URL.SUFFIX.EQ(\”png\”)” DENY<!--NeedCopy-->`
- `bind aaa group group1 -policy authzpol2<!--NeedCopy-->`

認証されたセッションを監査する

October 7, 2021

Citrix ADC アプライアンスは、認証されたセッションでトリガーされたすべてのイベントのログを保持するように構成できます。この情報を使用して、状態およびステータス情報を監査し、ユーザーの履歴を時系列で確認できます。

これを行うには、以下を指定する監査ポリシーを定義します。

- ログのタイプ。ログは、リモート（syslog）に保存することも、Citrix ADC アプライアンス（nslog）にローカルに保存することもできます。
- **Rule.** ログが格納される条件。
- アクション。ログサーバの詳細、およびログエントリを作成するためのその他の詳細。

この監査ポリシーは、ユーザーレベル、グループレベル、認証、承認、監査仮想サーバー、グローバルシステムレベルなど、さまざまなレベルで構成できます。ユーザレベルで設定されたポリシーは、最も高いプライオリティを持ちます。

> 注
>
> このトピックでは、syslog を使用する手順について詳しく説明します。nslog を使用するために必要な変更を加えます。

CLI を使用して syslog 監査を設定するには

1. 関連するログ設定で監査サーバーを構成します。

   ns-cli-prompt> **add audit syslogAction <name> <serverIP> ...**

2. 監査サーバーを関連付けて、監査ポリシーを設定します。

   ns-cli-prompt> **add audit syslogPolicy <name> <rule> <action>**

3. 監査ポリシーを次のいずれかのエンティティに関連付けます。

   - ポリシーを特定のユーザーにバインドします。

     ns-cli-prompt> **bind aaa user <userName>-policy** <policyname> ...

   - ポリシーを特定のグループにバインドします。

     ns-cli-prompt> **bind aaa group <groupName>-policy** <policyname> ...

- 認証、承認、および監査仮想サーバーにポリシーをバインドします。

  ns-cli-prompt> **bind authentication vserver <name> -policy** <policyname> …

- ポリシーを Citrix ADC アプライアンスにグローバルにバインドします。

  ns-cli-prompt> **bind tm global -policyName** <policyname> …

GUI を使用して **syslog** 監査を設定するには **([**設定**]** タブ**)**

1. 監査サーバとポリシーを設定します。

   [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**監査**] > [Syslog**] に移動し、関連するタブでサーバとポリシーを設定します。

2. ポリシーを次のいずれかに関連付けます。

   - ポリシーを特定のユーザーにバインドします。

     [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ユーザ**]** に移動し、認可ポリシーを関連するユーザに関連付けます。

   - ポリシーを特定のグループにバインドします。

     [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**グループ**]** に移動し、認可ポリシーを関連するグループに関連付けます。

   - 認証、承認、および監査仮想サーバーにポリシーをバインドします。

     [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**仮想サーバ] に移動し、認可ポリシーを関連する仮想サーバに関連付けます。

   - ポリシーを Citrix ADC アプライアンスにグローバルにバインドします。

     [セキュリティ**] > [AAA-**アプリケーショントラフィック**] > [**ポリシー**] > [**監査**] > [Syslog**] または [**Nslog**] に移動し、承認ポリシーを選択し、[アクション**] > [**グローバルバインド] をクリックしてポリシーをグローバルにバインドします。

Active Directory フェデレーションサービスプロキシとしての **Citrix ADC**

January 31, 2022

Active Directory フェデレーションサービス (ADFS) は、エンタープライズデータセンター外のリソースに対する Active Directory 認証クライアントのシングルサインオン (SSO) エクスペリエンスを可能にする Microsoft のサービスです。ADFS サーバーファームを使用すると、内部ユーザーは外部クラウドでホストされるサービスにアクセスできます。しかし、外部ユーザーが混在する瞬間に、外部ユーザーにリモートで接続し、フェデレーション ID を介してクラウドベースのサービスにアクセスする方法を与える必要があります。ほとんどの企業は、ADFS サーバーを

DMZ に公開したままにしておくことを望んでいません。したがって、ADFS プロキシは、リモートユーザーの接続とアプリケーションアクセスで重要な役割を果たします。

10 年以上にわたり、Citrix ADC アプライアンスは、リモートユーザー接続とアプリケーションアクセスにおいて同様の役割を果たしています。Citrix ADC アプライアンスは、次のサービスを有効にする新しい ADFS 実装をサポートするための ADFS プロキシとして使用される推奨ソリューションになります。

- 安全な接続。
- フェデレーション ID の認証と処理。

SAML IdP としての Citrix ADC の詳細については、「SAML IdP としての Citrix ADC」を参照してください。

ADFS プロキシの利点

- DMZ の設置面積を削減し、ほとんどの企業のニーズに応えます。
- エンドユーザーに SSO エクスペリエンスを提供します。
- 事前認証のための豊富な方法をサポートし、多要素認証を可能にします。
- アクティブクライアントとパッシブクライアントの両方をサポートします。

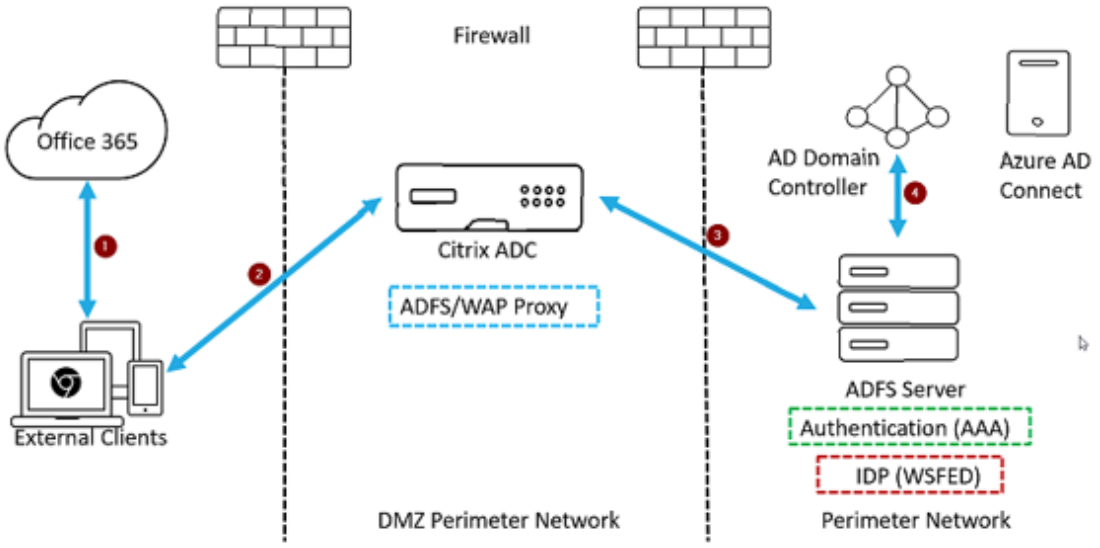
Citrix ADC を ADFS プロキシとして使用するための前提条件

Citrix ADC アプライアンスを ADFS プロキシとして構成する前に、次の前提条件が満たされていることを確認します。

- 12.1 以降のビルドを搭載した Citrix ADC アプライアンス。
- ドメイン ADFS サーバー。
- ドメイン SSL 証明書。
- コンテンツスイッチング仮想サーバーの仮想 IP。
- Citrix ADC アプライアンスで、負荷分散、SSL オフロード、コンテンツスイッチング、書き換え、認証、承認、監査のトラフィック管理機能を有効にします。

Citrix ADC アプライアンスを ADFS プロキシとして構成する

このユースケースを実現するには、DMZ ゾーンで ADFS プロキシとして Citrix ADC を構成します。ADFS サーバーは、バックエンドの AD ドメイン Controller と共に構成されます。



1. Microsoft Office365 にアクセスするためのクライアント要求は、ADFS プロキシとして展開された Citrix ADC にリダイレクトされます。
2. ユーザーの資格情報は ADFS サーバーに渡されます。
3. ADFS サーバーは、ドメインのオンプレミス AD で資格情報を認証します。
4. ADFS サーバーは、AD との資格情報の検証に成功すると、セッションを確立するために Microsoft Office365 に渡されるトークンを生成します。

以下に、ADFS プロキシとして構成する前に Citrix ADC アプライアンスを構成する手順の概要を示します。

Citrix ADC コマンドプロンプトで、次のコマンドを入力します。

1. バックエンドの SSL プロファイルを作成し、SSL プロファイルで SNI を有効にします。SSLv3/TLS1 を無効にします。

   ```
   add ssl profile <new SSL profile> -sniEnable ENABLED -ssl3 DISABLED -
   tls1 DISABLED -commonName <FQDN of ADFS>
   ```

2. サービスの SSLv3/TLS1 を無効にします。

   ```
   set ssl service <adfs service name> -sslProfile <SSL profile created in
    the above step>
   ```

3. バックエンドサーバハンドシェイクの SNI 拡張を有効にします。

   - `set vpn parameter -backendServerSni ENABLED`
   - `set ssl parameter -denySSLReneg NONSECURE`

CLI を使用して、**Citrix ADC** アプライアンスを **ADFS** プロキシとして構成する

次の項は、設定手順を完了するための要件に基づいて分類されています。

ADFS サービスを構成するには

1. ADFS サーバー用の Citrix ADC で ADFS サービスを構成します。

   ```
   add service <Domain_ADFS_Service> <ADFS Server IP> SSL 443 -gslb NONE -
   maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF
   -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
   ```

   例

   add service CTXTEST_ADFS_Service 1.1.1.1 SSL 443 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

2. コンテンツスイッチ仮想サーバーの FQDN を構成し、SNI を有効にします。

   ```
   set ssl service <Domain_ADFS_Service> -SNIEnable ENABLED -commonName <
   sts.domain.com>
   ```

   例

   set ssl service CTXTEST_ADFS_Service -SNIEnable ENABLED -commonName sts.ctxtest.com

ADFS 負荷分散仮想サーバーを構成するには

> 重要
>
> セキュリティで保護されたトラフィックには、ドメイン SSL 証明書 (SSL_CERT) が必要です。

1. ADFS 負荷分散仮想サーバーを構成します。

   ```
   add lb vserver <Domain_ADFS_LBVS> SSL <IP_address> -persistenceType
   NONE -cltTimeout 180
   ```

   例

   ```
   add lb vserver CTXTEST_ADFS_LBVS SSL 192.168.1.0 -persistenceType NONE
   -cltTimeout 180
   ```

2. ADFS 負荷分散仮想サーバーを ADFS サービスにバインドします。

   ```
   bind lb vserver <Domain_ADFS_LBVS> <Domain_ADFS_Service>
   ```

   例

   ```
   bind lb vserver CTXTEST_ADFS_LBVS CTXTEST_ADFS_Service
   ```

3. SSL 仮想サーバ証明書とキーのペアをバインドします。

   ```
   bind ssl vserver <Domain_ADFS_LBVS> -certkeyName <SSL_CERT>
   ```

   例

   ```
   bind ssl vserver CTXTEST_ADFS_LBVS -certkeyName ctxtest_newcert_2019
   ```


ドメインのコンテンツスイッチング仮想サーバーを構成するには

> 注:
>
> コンテンツスイッチング仮想サーバーには、パブリック IP にナットされている無料の仮想 IP (2.2.2.2 など) が 1 つ必要です。外部トラフィックと内部トラフィックの両方で到達可能である必要があります。

1. 無料 VIP でコンテンツスイッチング仮想サーバーを作成します。

   ```
   add cs vserver <Domain_CSVS> SSL <FREE VIP> 443 -cltTimeout 180 -
   persistenceType NONE
   ```

   例

   ```
   add cs vserver CTXTEST_CSVS SSL 2.2.2.2 443 -cltTimeout 180 -persistenceType
    NONE
   ```

2. コンテンツスイッチ仮想サーバーを負荷分散仮想サーバーにバインドします。

   ```
   bind cs vserver <Domain_CSVS> -lbvserver <Domain_ADFS_LBVS>
   ```

   例

   - `bind cs vserver CTXTEST_CSVS -lbvserver CTXTEST_ADFS_LBVS`
   - `set ssl vserver CTXTEST_CSVS -sessReuse DISABLED`

3. SSL 仮想サーバ証明書とキーのペアをバインドします。

   ```
   bind ssl vserver <Domain_CSVS> -certkeyName <SSL_CERT>
   ```

   例

   ```
   bind ssl vserver CTXTEST_CSVS -certkeyName ctxtest_newcert_2019
   ```

サポートされているプロトコル

Microsoft が提供するプロトコルは、Citrix ADC アプライアンスとの統合において重要な役割を果たします。ADFS プロキシとしての Citrix ADC は、次のプロトコルをサポートしています。

- **WS** フェデレーション。詳細については、「Web サービスフェデレーションプロトコル」を参照してください。
- **ADFSPIP**。詳細については、「Active Directory フェデレーションサービスプロキシ統合プロトコルコンプライアンス」を参照してください。

> 注
>
> Citrix ADC アプライアンスは、ADFS プロキシとして展開されている場合、デバイス証明書認証をサポートしません。

Web サービスフェデレーションプロトコル

May 9, 2022

Web サービスフェデレーション (WS-Federation) は、2 つのドメイン間に信頼関係がある場合に、ある信頼ドメインのセキュリティトークンサービス (STS) が別の信頼ドメインの STS に認証情報を提供できるようにするアイデンティティプロトコルです。

WS-フェデレーションの利点

WS フェデレーションはアクティブクライアントとパッシブクライアントの両方をサポートしますが、SAML IdP はパッシブクライアントのみをサポートします

- アクティブクライアントは、Outlook などの Microsoft ネイティブクライアントと Office クライアント (Word、PowerPoint、Excel、および OneNote) です。
- パッシブクライアントは、Google Chrome、Mozilla Firefox、インターネットエクスプローラーなどのブラウザベースのクライアントです。

Citrix ADC を WS フェデレーションとして使用するための前提条件

Citrix ADC アプライアンスを ADFS プロキシとして構成する前に、以下を確認してください。

- Active Directory
- ドメイン SSL 証明書。
- ADFS サーバー上の Citrix ADC SSL 証明書と ADFS トークン署名証明書は同じである必要があります。

> 重要
>
> SAML IdP は WS フェデレーションプロトコルを処理できるようになりました。したがって、WS-Federation IdP を設定するには、実際に SAML IdP を設定する必要があります。WS-Federation について明示的に言及しているユーザーインターフェイスは表示されません。

ADFS プロキシおよび WS-Federation IdP として構成されている場合に Citrix ADC でサポートされる機能

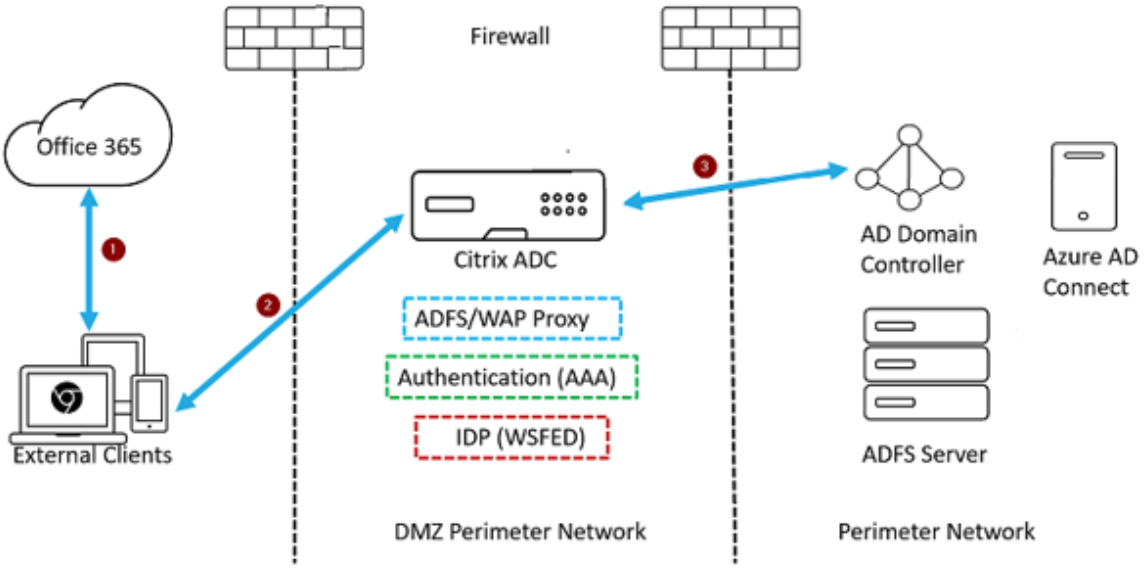
次の表に、ADFS プロキシおよび WS-Federation IdP として構成された場合に Citrix ADC アプライアンスでサポートされる機能を示します。

機能	Citrix ADC アプライアンスを ADFS プロキシとして構成する	WS-フェデレーション IdP としての Citrix ADC	ADFSPIP としての Citrix ADC
負荷分散	はい	はい	はい

機能	Citrix ADC アプライアンスを ADFS プロキシとして構成する	WS-フェデレーション IdP としての Citrix ADC	ADFSPIP としての Citrix ADC
SSL 終了	はい	はい	はい
レート制限	はい	はい	はい
統合 (DMZ サーバの設置面積を削減し、パブリック IP を節約)	はい	はい	はい
Web アプリケーションファイアウォール（WAF）	はい	はい	はい
Citrix ADC アプライアンスへの認証オフロード	はい	はい (アクティブクライアントとパッシブクライアント)	はい
シングルサインオン（SSO）	はい	はい (アクティブクライアントとパッシブクライアント)	はい
多要素（nFactor）認証	いいえ	はい (アクティブクライアントとパッシブクライアント)	はい
Azure 多要素認証	いいえ	はい (アクティブクライアントとパッシブクライアント)	はい
ADFS サーバーファームは避けられる	いいえ	はい	はい

Citrix ADC アプライアンスを **WS** フェデレーション **IdP** として構成する

DMZ ゾーンで Citrix ADC を WS フェデレーション IdP（SAML IdP）として構成します。ADFS サーバーは、バックエンドの AD ドメインコントローラーとともに構成されます。



1. Microsoft Office365 へのクライアント要求は、Citrix ADC アプライアンスにリダイレクトされます。
2. ユーザーは、多要素認証の資格情報を入力します。
3. Citrix ADC は AD で資格情報を検証し、Citrix ADC アプライアンスでネイティブにトークンを生成します。資格情報は、アクセスのために Office365 に渡されます。

> 注
>
> WS-Federation IdP のサポートは、F5 Networks ロードバランサーと比較した場合、Citrix ADC アプライアンスを介してネイティブに行われます。

CLI を使用して Citrix ADC アプライアンスを WS-フェデレーション IdP（SAML IdP）として構成する

以下のセクションは、構成手順を完了するための要件に基づいて分類されています。

LDAP 認証を設定してポリシーを追加するには

> 重要
>
> ドメインユーザーの場合、会社の電子メールアドレスを使用して Citrix ADC アプライアンスにログオンするには、以下を構成する必要があります。

- Citrix ADC アプライアンスで LDAP 認証サーバーとポリシーを構成します。
- 認証、承認、および監査用の仮想 IP アドレスにバインドします (既存の LDAP 設定の使用もサポートされています)。

```
1 add authentication ldapAction <Domain_LDAP_Action> -serverIP <Active
    Directory IP> -serverPort 636 -ldapBase ”cn=Users,dc=domain,dc=com”
```



```
    -ldapBindDn "cn=administrator,cn=Users,dc=domain,dc=com" -
    ldapBindDnPassword <administrator password> -encrypted -
    encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName
     memberOf -subAttributeName cn -secType SSL -ssoNameAttribute
    UserPrincipalName -followReferrals ON -Attribute1 mail -Attribute2
    objectGUID
2
3 add authentication Policy <Domain_LDAP_Policy> -rule true -action <
    Domain_LDAP_Action>
4 <!--NeedCopy-->
```

例

```
1 add authentication ldapAction CTXTEST_LDAP_Action -serverIP 3.3.3.3 -
    serverPort 636 -ldapBase "cn=Users,dc=ctxtest,dc=com" -ldapBindDn "
    cn=administrator,cn=Users,dc=ctxtest,dc=com" -ldapBindDnPassword
    xxxxxxxxxx -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName
    sAMAccountName -groupAttrName memberOf -subAttributeName cn -secType
     SSL -ssoNameAttribute UserPrincipalName -followReferrals ON -
    Attribute1 mail -Attribute2 objectGUID
2
3 add authentication Policy CTXTEST_LDAP_Policy -rule true -action
    CTXTEST_LDAP_Action
4 <!--NeedCopy-->
```

Citrix ADC を **WS** フェデレーション **IdP** または **SAML IdP** として構成するには

トークン生成のための WS-Federation IdP (SAML IdP) アクションとポリシーを作成します。後で認証、承認、および監査仮想サーバーにバインドします。

```
1 add authentication samlIdPProfile <Domain_SAMLIDP_Profile> -
    samlIdPCertName <SSL_CERT> -assertionConsumerServiceURL "https://
    login.microsoftonline.com/login.srf" -samlIssuerName <Issuer Name
    for Office 365 in ADFS Server> -rejectUnsignedRequests OFF -audience
     urn:federation:MicrosoftOnline -NameIDFormat persistent -NameIDExpr
     "HTTP.REQ.USER.ATTRIBUTE(2).B64ENCODE" -Attribute1 IDPEmail -
    Attribute1Expr "HTTP.REQ.USER.ATTRIBUTE(1)"
2
3 add authentication samlIdPPolicy <Domain_SAMLIDP_Policy> -rule "HTTP.
    REQ.HEADER("referer").CONTAINS("microsoft") || true" -action <
    Domain_SAMLIDP_Profile>
```



```
<!--NeedCopy-->
```

例

```
add authentication samlIdPProfile CTXTEST_SAMLIDP_Profile -
    samlIdPCertName ctxtest_newcert_2019 -assertionConsumerServiceURL "
    https://login.microsoftonline.com/login.srf" -samlIssuerName "http
    ://ctxtest.com/adfs/services/trust/" -rejectUnsignedRequests OFF -
    audience urn:federation:MicrosoftOnline -NameIDFormat persistent -
    NameIDExpr "HTTP.REQ.USER.ATTRIBUTE(2).B64ENCODE" -Attribute1
    IDPEmail -Attribute1Expr "HTTP.REQ.USER.ATTRIBUTE(1)"

add authentication samlIdPPolicy CTXTEST_SAMLIDP_Policy -rule "HTTP.REQ
    .HEADER("referer").CONTAINS("microsoft") || true" -action
    CTXTEST_SAMLIDP_Profile
<!--NeedCopy-->
```

企業の資格情報を使用して **Office365** にログオンする従業員を認証するように、認証、承認、および監査仮想サーバーを構成するには

```
add authentication vserver <Domain_AAA_VS> SSL <IP_address>`
<!--NeedCopy-->
```

例

```
add authentication vserver CTXTEST_AAA_VS SSL 192.168.1.0

bind authentication vserver CTXTEST_AAA_VS -portaltheme RfWebUI
<!--NeedCopy-->
```

認証仮想サーバーとポリシーをバインドするには

```
bind authentication vserver <Domain_AAA_VS> -policy <
    Domain_SAMLIDP_Policy> -priority 100 -gotoPriorityExpression NEXT

bind authentication vserver <Domain_AAA_VS> -policy <Domain_LDAP_Policy
    > -priority 100 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```


例

```
bind authentication vserver CTXTEST_AAA_VS -policy
    CTXTEST_SAMLIDP_Policy -priority 100 -gotoPriorityExpression NEXT

bind authentication vserver CTXTEST_AAA_VS -policy CTXTEST_LDAP_Policy
    -priority 100 -gotoPriorityExpression NEXT

bind ssl vserver CTXTEST_AAA_VS -certkeyName ctxtest_newcert_2019
<!--NeedCopy-->
```

コンテンツスイッチングを設定するには

```
add cs action <Domain_CS_Action> -targetVserver <Domain_AAA_VS>

add cs policy <Domain_CS_Policy> -rule "is_vpn_url || http.req.url.
    contains("/adfs/ls") || http.req.url.contains("/adfs/services/trust"
    ) || -action <Domain_CS_Action>
<!--NeedCopy-->
```

例

```
add cs action CTXTEST_CS_Action -targetVserver CTXTEST_AAA_VS

add cs policy CTXTEST_CS_Policy -rule "is_vpn_url || http.req.url.
    contains("/adfs/ls") || http.req.url.contains("/adfs/services/trust"
    ) || -action CTXTEST_CS_Action
<!--NeedCopy-->
```

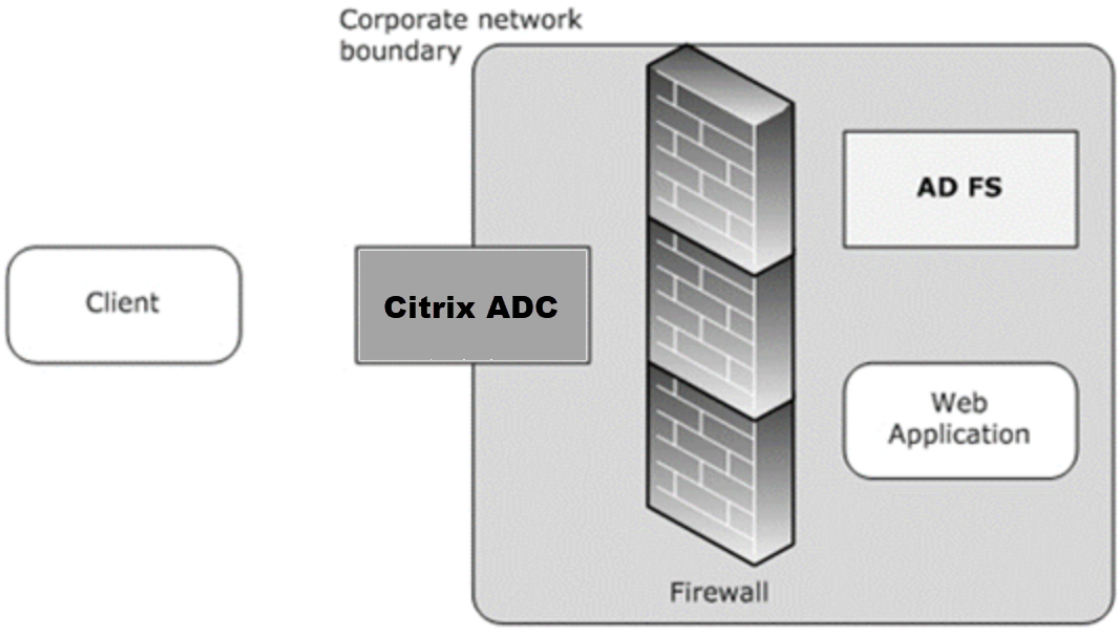
コンテンツスイッチ仮想サーバーをポリシーにバインドするには

```
bind cs vserver CTXTEST_CSVS -policyName CTXTEST_CS_Policy -priority
    100
<!--NeedCopy-->
```


Active Directory フェデレーションサービスプロキシ統合プロトコルコンプライアンス

March 8, 2022

Web アプリケーションプロキシの代わりにサードパーティプロキシを使用する場合は、ADFS および WAP 統合ルールを指定する MS-ADFSPIP プロトコルをサポートする必要があります。ADFSPIP は Active Directory フェデレーションサービスを認証およびアプリケーションプロキシと統合し、企業ネットワークの境界の外側にあるクライアントが、企業ネットワークの境界内にあるサービスにアクセスできるようにします。



前提条件

プロキシサーバーと ADFS ファームの間に信頼を正常に確立するには、Citrix ADC アプライアンスで次の構成を確認します。

- バックエンドの SSL プロファイルを作成し、SSL プロファイルで SNI を有効にします。SSLv3/TLS1 を無効にします。コマンドプロンプトで、次のコマンドを入力します。

```
add ssl profile <new SSL profile> -sniEnable ENABLED -ssl3
   DISABLED -tls1 DISABLED -commonName <FQDN of ADFS>
<!--NeedCopy-->
```


- サービスの SSLv3/TLS1 を無効にします。コマンドプロンプトで、次のコマンドを入力します。

```
set ssl service <adfs service name> -sslProfile
   ns_default_ssl_profile_backend
<!--NeedCopy-->
```

- バックエンドサーバーハンドシェイクの SNI 拡張を有効にします。コマンドプロンプトで、次のコマンドを入力します。

```
set vpn parameter  - backendServerSni ENABLED

set ssl parameter -denySSLReneg NONSECURE
<!--NeedCopy-->
```

> 重要
>
> 認証を ADFS サーバーにオフロードする必要があるホームレルム検出（HRD）シナリオでは、Citrix ADC アプライアンスで認証と SSO の両方を無効にすることをお勧めします。

認証メカニズム

認証のイベントフローの概要を次に示します。

1. **ADFS** サーバーとの信頼の確立 -Citrix ADC サーバーは、クライアント証明書を登録して ADFS サーバーとの信頼を確立します。信頼が確立されると、Citrix ADC アプライアンスは再起動後にユーザーの介入なしに信頼を再確立します。

   証明書の有効期限が切れたら、ADFS プロキシプロファイルを削除して再度追加して、信頼を再確立する必要があります。

2. 公開エンドポイント -Citrix ADC アプライアンスは、信頼の確立後、ADFS サーバー上の公開エンドポイントのリストを自動的にフェッチします。これらの公開されたエンドポイントは、ADFS サーバーに転送される要求をフィルタリングします。

3. クライアント要求にヘッダーを挿入する -Citrix ADC アプライアンスがクライアント要求をトンネリングすると、ADFSPIP に関連する HTTP ヘッダーがパケットに追加され、ADFS サーバーに送信されます。ADFS サーバーでは、これらのヘッダー値に基づいてアクセス制御を実装できます。次のヘッダーがサポートされています。

   - X-MS プロキシ
   - X-MS-エンドポイント-絶対パス
   - X-MS 転送クライアント IP

- X-MS プロキシ
- X-MS-ターゲット-ロール
- X-MS-ADFS-プロキシ-クライアント-IP

4. エンドユーザートラフィックの管理 — エンドユーザートラフィックは、目的のリソースに安全にルーティングされます。

> 注
>
> Citrix ADC アプライアンスはフォームベース認証を使用します。

ADFS サーバーをサポートするように Citrix ADC を構成する

前提条件

- Context Switching（CS; コンテキストスイッチング）サーバをフロントエンドとして設定し、CS の背後に認証、許可、および監査サーバコマンドプロンプトで入力します。

```
add cs vserver <cs vserver name> SSL 10.220.xxx.xx 443
-cltTimeout 180 -AuthenticationHost <adfs server hostname> -
    Authentication OFF -persistenceType NONE
<!--NeedCopy-->
```

```
add cs action <action name1> -targetLBVserver <lb vserver name>
<!--NeedCopy-->
```

```
add cs action <action name2> -targetLBVserver <lb vserver name>
<!--NeedCopy-->
```

```
add cs policy <policy name1> -rule " http.req.url.contains("/adfs
    /services/trust") || http.req.url.contains("federationmetadata
    /2007-06/federationmetadata.xml")" -action <action name1>
<!--NeedCopy-->
```

```
add cs policy <policy name2> -rule "HTTP.REQ.URL.CONTAINS("/adfs/
    ls")" -action <action name2>
<!--NeedCopy-->
```



```
bind cs vserver <cs vserver name> -policyName <policy name1> -
   priority 100
<!--NeedCopy-->
```

```
bind cs vserver <cs vserver name> -policyName <policy name2> -
   priority 110
<!--NeedCopy-->
```

```
bind cs vserver <cs vserver name> -lbvserver <lb vserver name>
<!--NeedCopy-->
```

- ADFS サービスを追加します。コマンドプロンプトで入力します。

```
add service <adfs service name> <adfs server ip> SSL 443
<!--NeedCopy-->
```

```
set ssl service <adfs service name> -sslProfile
   ns_default_ssl_profile_backend
<!--NeedCopy-->
```

- 負荷分散された仮想サーバを追加します。コマンドプロンプトで入力します。

```
add lb vserver <lb vserver name> SSL 0.0.0.0 0
<!--NeedCopy-->
```

```
set ssl vserver <lb vserver name> -sslProfile
   ns_default_ssl_profile_frontend
<!--NeedCopy-->
```

- 負荷分散されたサーバーにサービスをバインドします。コマンドプロンプトで入力します。


```
bind lb vserver <lb vserver name> <adfs service name>
<!--NeedCopy-->
```

Citrix ADC が **ADFS** サーバーと連携するように構成するには、次の手順を実行する必要があります。

1. ADFS プロキシプロファイルで使用する SSL CertKey プロファイルキーを作成する
2. ADFS プロキシプロファイルを作成する
3. ADFS プロキシプロファイルを LB 仮想サーバに関連付けます。

ADFS プロキシプロファイルで使用するプライベートキーを含む **SSL** 証明書を作成する

コマンドプロンプトで入力します。

```
add ssl certkey <certkeyname>  - cert <certificate path> -key <
   keypath>
<!--NeedCopy-->
```

注：証明書ファイルとキーファイルが Citrix ADC アプライアンスに存在する必要があります。
CLI を使用して ADFS プロキシプロファイルを作成する

コマンドプロンプトで入力します。

```
add authentication adfsProxyProfile  <profile name> -serverUrl <https:
   //<server FQDN or IP address>/> -username <adfs admin user name> -
   password <password for admin user> -certKeyName <name of the CertKey
    profile created above>
<!--NeedCopy-->
```

どこ;

[Profile name]：作成する AFDS プロキシプロファイルの名前

serverURL — プロトコルとポートを含む ADFS サービスの完全修飾ドメイン名。例：https://adfs.citrix.com

Username — ADFS サーバに存在する管理者アカウントのユーザ名

Password — ユーザー名として使用する管理者アカウントのパスワード

certKeyName — 以前に作成された SSL CertKey プロファイルの名前

CLI を使用して **ADFS** プロキシプロファイルを負荷分散仮想サーバーに関連付ける

ADFS 展開では、2 つの負荷分散仮想サーバーが使用されます。1 つはクライアントトラフィック用で、もう 1 つはメタデータ交換用です。ADFS プロキシプロファイルは、ADFS サーバーのフロントエンドである負荷分散仮想サーバーに関連付ける必要があります。

コマンドプロンプトで入力します。

```
set lb vserver <adfs-proxy-lb> -adfsProxyProfile <name of the ADFS
   proxy profile>
<!--NeedCopy-->
```

ADFSPIP の信頼更新サポート

有効期限が近づいているか、既存の証明書が有効でない場合は、既存の証明書の信頼を更新できます。証明書の信頼の更新は、Citrix ADC アプライアンスと ADFS サーバーの間で信頼が確立された場合にのみ実行されます。証明書の信頼性を更新するには、新しい証明書を提供する必要があります。

> 重要
>
> 新しい証明書の信頼更新には、手動による介入が必要です。

次の例は、証明書信頼の更新に関係する手順を示しています。

1. Citrix ADC アプライアンスは、信頼の更新のために、POST 要求で古い（シリアル化された TrustCertificate）証明書と新しい証明書（SerializedReplacementCertificate）の両方を ADFS サーバーに送信します。
2. 信頼が正常に更新されると、ADFS サーバーは 200 OK 成功を返して応答します。
3. 信頼の更新が成功した場合、Citrix ADC アプライアンスは状態を「ESTABLISHED_RENEW_SUCCESS」として更新します。信頼の更新に失敗すると、状態は「ESTABLISHED_RENEW_FAILED」として更新され、Citrix ADC アプライアンスは古い証明書を使用し続けます。

> 注
>
> 証明書キーがすでに一部の ADFS プロキシプロファイルにバインドされている場合は、証明書キーを更新できません。

CLI を使用して証明書の信頼更新を設定するには

コマンドプロンプトで入力します。

```
set authentication adfsProxyProfile <name> [-CertKeyName <string>]
<!--NeedCopy-->
```


例:

```
set authentication adfsProxyProfile adfs_2 -CertKeyName ca_cert1
<!--NeedCopy-->
```

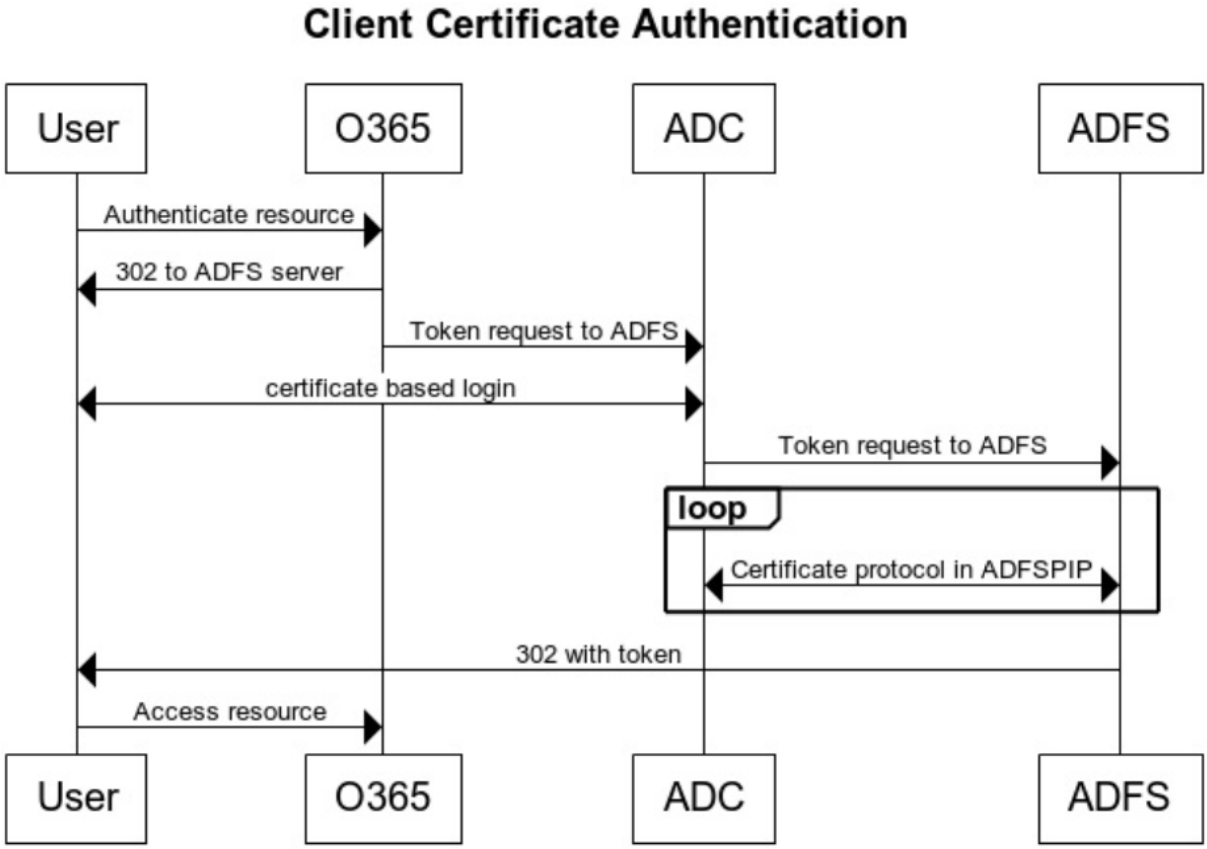
ADFS サーバーでのクライアント証明書ベースの認証

Microsoft は、Windows Server 2016 以降、プロキシサーバーを介して ADFS にアクセスするときにユーザーを認証する新しい方法を導入しました。これで、エンドユーザーは証明書を使用してログインできるようになり、パスワードの使用を回避できます。

エンドユーザーは、特に社内にいない場合に、プロキシ経由で ADFS にアクセスすることがよくあります。したがって、ADFS プロキシサーバーは ADFSPIP プロトコルによるクライアント証明書認証をサポートする必要があります。

Citrix ADC アプライアンスを使用して ADFS を負荷分散する場合、ADFS サーバーで証明書ベースの認証をサポートするには、ユーザーは証明書を使用して Citrix ADC アプライアンスにもログインする必要があります。これにより、Citrix ADC はユーザー証明書を ADFS に渡して、ADFS サーバーに SSO を提供できます。

次の図は、クライアント証明書認証フローを示しています。



クライアント証明書を使用して **ADFS** サーバの **SSO** を設定する

クライアント証明書を使用して ADFS サーバーの SSO を構成するには、まず Citrix ADC アプライアンスでクライアント証明書認証を構成する必要があります。次に、証明書認証ポリシーを認証、承認、および監査仮想サーバーにバインドする必要があります。

また、次の手順を実行する必要があります。

- ポート 49443 を持つ追加のコンテキストスイッチング仮想サーバを設定し、このコンテキストスイッチング仮想サーバが、前に作成したすべてのポートに対して開いている同じ負荷分散仮想サーバをポイントする必要があります。
- 認証のために、Citrix ADC アプライアンスでポート 49443 を開く必要があります。
- コンテキストスイッチングポリシーは、先に作成したポート 443 が開いているのと同じ負荷分散仮想サーバにバインドする必要があります。
- 前に作成したものと同じ SSL サービスを、負荷分散仮想サーバーにバインドする必要があります。
- バックエンドの SSL プロファイルを既に作成している場合は、そのプロファイルを使用する必要があります。

コマンドプロンプトで次を入力します。

```
add cs vserver <name> <serviceType> <port>

bind cs vserver <name> (-lbvserver <string> | -vServer <string> | [-
    targetLBVserver <string>]

set ssl vserver <vServerName [-sslProfile <string>]

bind ssl vserver <vServerName -certkeyName <string>

add authentication certAction <action name>

add authentication Policy <policy name> -rule <expression> -action <
    action name>

add authentication policylable <label Name>

bind authentication policylabel <label Name> -policyName <name of the
    policy> -priority<integer>

<!--NeedCopy-->
```

例:


```
add cs vserver srv123_adfsproxy_csvs_tls SSL $VIP_1 49443

bind cs vserver srv123_adfsproxy_csvs_tls -lbvserver
    srv123_adfs_lbvserver

set ssl vserver srv123_adfsproxy_csvs_tls -sslProfile
    ns_default_ssl_profile_frontend

bind ssl vserver srv123_adfsproxy_csvs_tls -certkeyName
    srv123_wildcardcert

add authentication certAction adfsproxy-cert

add authentication Policy cert1 -rule TRUE -action adfsproxy-cert

add authentication policylable certfactor

bind authentication policylabel certfactor  - policyName cert1  -
    priority 100

<!--NeedCopy-->
```

Citrix ADC アプライアンスでのクライアント証明書の構成について詳しくは、「高度なポリシーを使用したクライアント証明書認証の構成」を参照してください。

オンプレミス **Citrix Gateway** を **ID** プロバイダーとして **Citrix Cloud** に使用

October 7, 2021

Citrix Cloud では、オンプレミスの Citrix Gateway を ID プロバイダーとして使用してワークスペースにサインインする利用者が認証されるようにできます。

Citrix Gateway 認証を使用すると、以下のことを実行できます:

- 引き続き、既存の Citrix Gateway でユーザーを認証するため、Citrix Workspace 経由でオンプレミスの Virtual Apps and Desktops のリソースにアクセスできます。
- Citrix Workspace では、Citrix Gateway の認証、承認、および監査機能を使用します。
- パススルー認証、スマートカード、セキュアトークン、条件付きアクセスポリシー、フェデレーション、その他多くの機能を使用しながら、ユーザーに必要なリソースへの Citrix Workspace 経由のアクセスを提供できます。

Citrix Gateway 認証は、次の製品バージョンでの使用がサポートされています:

- Citrix Gateway 13.0 41.20 Advanced Edition 以降
- Citrix Gateway 12.1 54.13 Advanced Edition 以降

前提条件

- クラウドコネクタ-Citrix Cloud Connector ソフトウェアをインストールするには、少なくとも 2 台のサーバーが必要です。
- Active Directory -必要なチェックを実行します。
- Citrix Gateway の要件
  - クラシックポリシーの非推奨のため、オンプレミスゲートウェイで高度なポリシーを使用します。
  - Citrix Workspace へのサブスクライバを認証するためにゲートウェイを構成する場合、ゲートウェイは OpenID Connect プロバイダーとして機能します。Citrix Cloud と Gateway 間のメッセージは OIDC プロトコルに準拠し、デジタル署名トークンが含まれます。したがって、これらのトークンに署名するための証明書を構成する必要があります。
  - クロック同期-ゲートウェイは NTP 時刻に同期する必要があります。

詳しくは、「前提条件」を参照してください。

オンプレミスの **Citrix Gateway** で **OAuthIdP** ポリシーを作成します

> 重要:
>
> **Citrix Cloud >** ［アイデンティティおよびアクセス管理］ **>** ［認証］タブで、クライアント **ID**、シークレット、およびリダイレクト **URL** を生成しておく必要があります。詳細については、「オンプレミスの Citrix Gateway を Citrix Cloud に接続する」を参照してください。

OAuth IdP 認証ポリシーの作成には、次のタスクが含まれます。

1. OAuth IdP プロファイルを作成します。
2. OAuth IdP ポリシーを追加します。
3. OAuth IdP ポリシーを認証仮想サーバーにバインドします。
4. 証明書をグローバルにバインドします。

CLI を使用した **OAuth IdP** プロファイルの作成

コマンドプロンプトで次を入力します。

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]
```



```
2
3 add authentication OAuthIdPPolicy <name> -rule <expression> [-action <
    string> [-undefAction <string>] [-comment <string>][-logAction <
    string>]
4
5 add authentication ldapAction <name> -serverIP <IP> -ldapBase ”dc=aaa,
    dc=local”
6
7 ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -
    ldapLoginName sAMAccountName
8
9 add authentication policy <name> -rule <expression> -action <string>
10
11 bind authentication vserver auth_vs -policy <ldap_policy_name> -
    priority <integer> -gotoPriorityExpression NEXT
12
13 bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -
    priority <integer> -gotoPriorityExpression END
14
15 bind vpn global  - certkey <>
16 <!--NeedCopy-->
```

GUI を使用した **OAuth IdP** プロファイルの作成

1. セキュリティ **> AAA —**アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 高度なポリシー **> OAuth IDP** に移動します。

   ![Oauth-IDP-navigation](/en-us/citrix-adc/media/oauth-navigation-to-idp.png)

2. **OAuth IDP** ページで、[プロファイル] タブを選択し、[追加] をクリックします。
3. OAuth IdP プロファイルを設定します。

   > 注**:**
   >
   > - **Citrix Cloud >** ［アイデンティティとアクセス管理］ **>** ［認証］ タブからクライアント **ID**、シークレット、およびリダイレクト **URL** の値をコピーして貼り付けて、Citrix Cloud への接続を確立します。
   > - 発行者名の例にゲートウェイ URL を正しく入力します。https://GatewayFQDN.com
   > - また、クライアント ID を [オーディエンス] フィールドにもコピーして貼り付けます。
   > - パスワードを送信: シングルサインオンサポートの場合、このオプションを有効にします。デフォルトでは、このオプションは無効になっています。

4. [認証 **OAuth IDP** プロファイルの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。

- **Name**： 認証プロファイルの名前。文字、数字、またはアンダースコア文字（_）で始まり、文字、数字、ハイフン（-)、ピリオド（.）ポンド（#)、スペース ()、アットマーク（@)、等号（=)、コロン（:)、およびアンダースコア文字のみを含める必要があります。プロファイルの作成後は変更できません。
- クライアント **ID**： SP を識別する一意の文字列。認可サーバは、この ID を使用してクライアント設定を推測します。最大長:127。
- **Client Secret** — ユーザーと認可サーバーによって確立されたシークレット文字列。最大長:239
- リダイレクト **URL** — コード/トークンのポスト先となる SP 上のエンドポイント。
- 発行者名 — トークンが受け入れられるサーバーのアイデンティティ。最大長:127。例: https://GatewayFQDN.com
- オーディエンス — IdP によって送信されるトークンのターゲット受信者。これは受信者がチェックするかもしれません。
- スキュー時間: このオプションは、Citrix ADC が受信トークンで許可するクロックスキューを分単位で指定します。たとえば、SkewTime が 10 の場合、トークンは (現在の時刻-10) 分から (現在時刻 + 10) 分、つまり 20 分まで有効です。デフォルト値:5。
- デフォルト認証グループ: nFactor フローで使用できる IdP によってこのプロファイルが選択されたときに、セッション内部グループリストに追加されるグループ。証明書パーティ関連の nFactor フローを識別するための認証ポリシーの式（AAA.USER.IS_MEMBER_OF (「xxx」)）で使用できます。最大長:63

ポリシー評価を簡素化し、ポリシーのカスタマイズに役立つように、このプロファイルのセッションにグループが追加されます。これは、抽出されたグループに加えて認証が成功したときに選択されるデフォルトグループです。最大長:63。

![Oauth-IDP-profile-parameters](/en-us/citrix-adc/media/oauth-idp-profile.png)

5. **[**ポリシー**]** をクリックし、**[**追加**]** をクリックします。
6. [認証 **OAuth IDP** ポリシーの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。
   - **Name** — 認証ポリシーの名前。
   - **Action** — 以前に作成されたプロファイルの名前。
   - **Log Action** — リクエストがこのポリシーに一致する場合に使用するメッセージログアクションの名前。必須の提出ではありません。
   - **Undefined-Result** アクション: ポリシー評価の結果が未定義（UNDEF）である場合に実行するアクション。必須フィールドではありません。
   - **Expression** — ポリシーが特定の要求に応答するために使用するデフォルトの構文式。たとえば、true です。
   - コメント — ポリシーに関するコメント。

   ![Oauth-IDP-policy](/en-us/citrix-adc/media/oauth-idp-policy.png)

> 注:
>
> **SendPassword** がオン（デフォルトではオフ）に設定されている場合、ユーザー資格情報は暗号化され、安全なチャネルを介して Citrix Cloud に渡されます。セキュリティで保護されたチャネルを介してユーザー資格情報を渡すと、起動時に Citrix Virtual Apps and Desktops への SSO を有効にできます。

OAuthIdP ポリシーと **LDAP** ポリシーを認証仮想サーバーにバインドする

1. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** アクション **> LDAP** に移動します。
2. [**LDAP** アクション] 画面で、[追加] をクリックします。
3. [認証 **LDAP** サーバーの作成] 画面で、次のパラメータの値を設定し、[作成] をクリックします。
   - **Name —** LDAP アクションの名前
   - サーバ名**/**サーバ **IP —** LDAP サーバの FQDN または IP を指定します。
   - [セキュリティタイプ**]**、**[**ポート**]**、**[**サーバータイプ**]**、**[**タイムアウト] の適切な値を選択する
   - [認証**]** がオンになっていることを確認します
   - ベース **DN**： LDAP 検索を開始する基本。たとえば、`dc=aaa,dc=local`などです。
   - 管理者バインド **DN:** LDAP サーバーへのバインドのユーザー名。たとえば、admin@aaa.local。
   - 管理者パスワード**/**パスワードの確認**:LDAP** をバインドするためのパスワード
   - [接続のテスト] をクリックして、設定をテストします。
   - サーバーログオン名属性**:**「**samAccountName**」を選択します。
   - その他のフィールドは必須ではないため、必要に応じて設定できます。
4. 設定 **>** セキュリティ **> AAA** アプリケーショントラフィック **>** ポリシー **>** 認証 **>** 詳細ポリシー **>** ポリシーにナビゲートして下さい。
5. [認証ポリシー] 画面で、[追加] をクリックします。
6. [認証ポリシーの作成] ページで、次のパラメータの値を設定し、[作成**]** をクリックします。
   - **Name —** LDAP 認証ポリシーの名前。
   - [アクションタイプ**] —**[**LDAP]** を選択します。
   - [アクション**] —** LDAP アクションを選択します。
   - **Expression —** ポリシーが特定の要求に応答するために使用するデフォルトの構文式。たとえば、true** です。

Citrix Gateway でのアクティブ**/**アクティブ **GSLB** 展開のサポート

OIDC プロトコルを使用してアイデンティティプロバイダ（IdP）として構成された Citrix Gateway は、アクティブ/アクティブ GSLB 展開をサポートできます。Citrix Gateway IdP でのアクティブ/アクティブ GSLB 展開では、複数の地理的ロケーションで受信したユーザーログイン要求を負荷分散できます。

> 重要
>
> セキュリティを強化するために、CA 証明書を SSL サービスにバインドし、SSL サービスで証明書検証を有効にすることをお勧めします。

GSLB セットアップの設定の詳細については、GSLB のセットアップと設定の例を参照してください。

SameSite クッキー属性の構成サポート

May 9, 2022

SameSite 属性は、Cookie をクロスサイトコンテキストで使用するか、同じサイトコンテキストに対してのみ使用できるかをブラウザに示します。また、アプリケーションがクロスサイトコンテキストでアクセスされることを意図している場合は、HTTPS 接続を介してのみアクセスできます。詳細については、RFC6265 を参照してください。

2020 年 2 月まで、SameSite 属性は Citrix ADC で明示的に設定されていませんでした。ブラウザはデフォルト値 (None) を使用しました。SameSite 属性を設定しなくても、Citrix Gateway、認証、承認、監査の展開には影響しませんでした。

Google Chrome 80 などの特定のブラウザのアップグレードでは、Cookie のデフォルトのクロスドメイン動作に変更があります。SameSite 属性は、次のいずれかの値に設定できます。Google Chrome のデフォルト値は Lax に設定されています。他のブラウザの特定のバージョンでは、SameSite 属性のデフォルト値が None に設定されている場合があります。

- なし: ブラウザが安全な接続でのみクロスサイトコンテキストで Cookie を使用することを示します。
- **Lax**: ブラウザが同じドメイン上のリクエストやクロスサイトのリクエストに Cookie を使用することを示します。クロスサイトでは、GET リクエストなどの安全な HTTP メソッドのみが Cookie を使用できます。たとえば、あるサブドメイン abc.example.com の GET リクエストは、GET を使用して別のサブドメイン xyz.example.com の Cookie を読み取ることができます。
  クロスサイトの場合、安全な HTTP メソッドはサーバーの状態を変更しないため、安全な HTTP メソッドのみが使用されます。詳しくは、「https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite#lax」を参照してください。
- 厳格:Cookie は同じサイトコンテキストでのみ使用します。

Cookie に SameSite 属性がない場合、Google Chrome は SameSite = Lax の機能を想定しています。
その結果、ブラウザによる Cookie の挿入を必要とするクロスサイトコンテキストを持つ iframe 内のデプロイでは、Google Chrome はクロスサイト Cookie を共有しません。その結果、Web サイト内の iframe が読み込まれないことがあります。

SameSite Cookie 属性を設定する

SameSite という名前の新しい Cookie 属性が VPN および認証、承認、および監査仮想サーバーに追加されます。この属性は、グローバルレベルおよび仮想サーバレベルで設定できます。

SameSite 属性を設定するには、次の操作を行う必要があります。

1. 仮想サーバーの SameSite 属性を設定する
2. クッキーをパッチセットにバインドする (ブラウザがクロスサイトクッキーをドロップした場合)

CLI を使用して **SameSite** 属性を設定する

仮想サーバーレベルで SameSite 属性を設定するには、次のコマンドを使用します。

```
set vpn vserver VP1 -SameSite [STRICT | LAX | None]
set authentication vserver AV1 -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```

グローバルレベルで SameSite 属性を設定するには、次のコマンドを使用します。

```
set aaa parameter -SameSite [STRICT | LAX | None]
set vpn parameter -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```

注：仮想サーバーレベルの設定は、グローバルレベル設定よりも優先されます。Citrix では、仮想サーバーレベルで SameSiteCookie 属性を設定することをお勧めします。

CLI を使用してクッキーをパッチセットにバインドする

ブラウザがクロスサイトクッキーをドロップした場合、その Cookie 文字列を既存の NS_cookies_SameSite パッチセットにバインドして、SameSite 属性が Cookie に追加されるようにすることができます。

例**:**

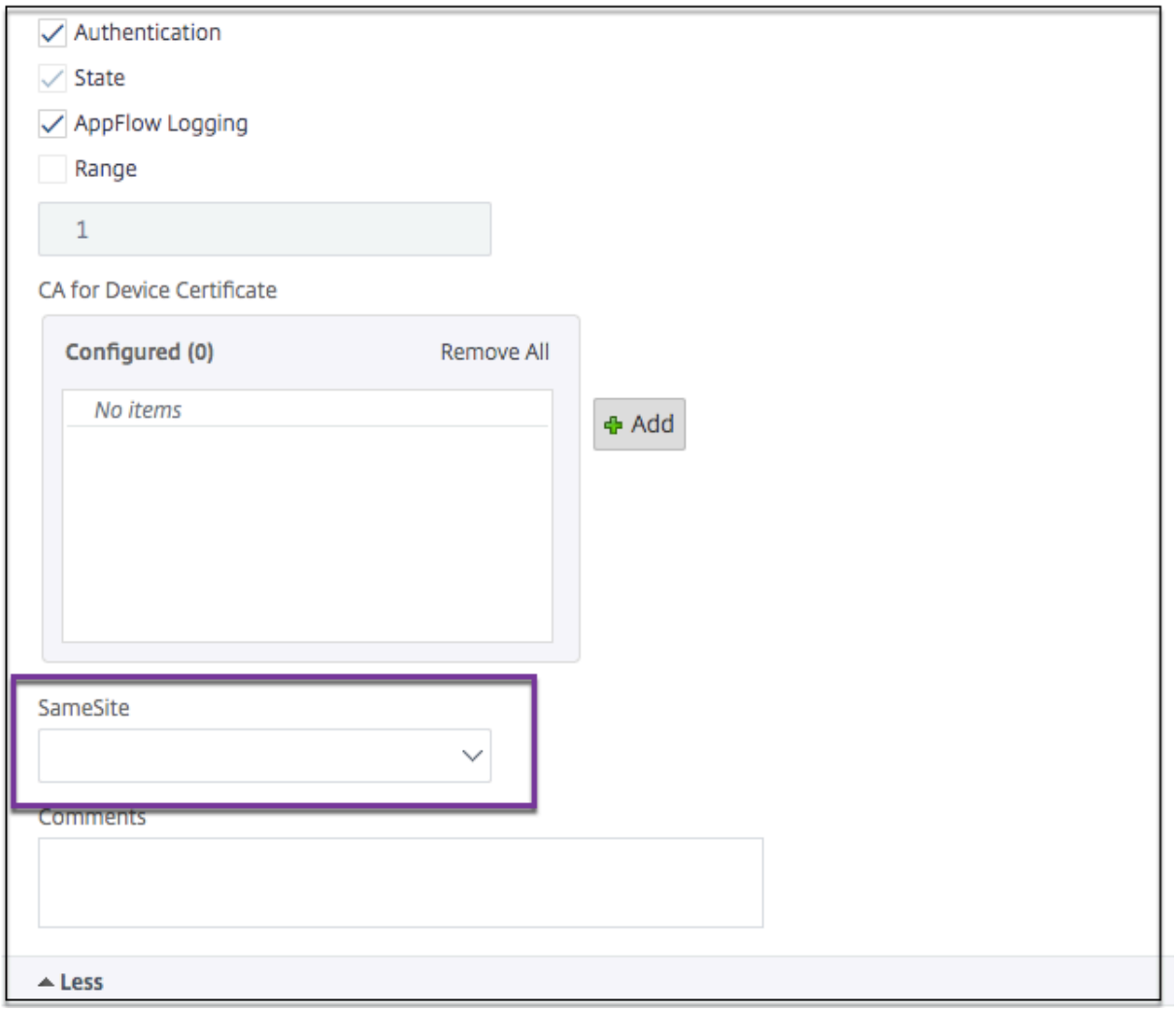
```
bind patset ns_cookies_SameSite ”NSC_TASS”
bind patset ns_cookies_SameSite ”NSC_TMAS”
<!--NeedCopy-->
```

GUI を使用して **sameSite** 属性を設定する

仮想サーバーレベルで **SameSite** 属性を設定するには、次の手順を実行します。

1. セキュリティ **> AAA —** アプリケーショントラフィック **>** 仮想サーバに移動します。
2. 仮想サーバを選択し、[編集（Edit）] をクリックします。

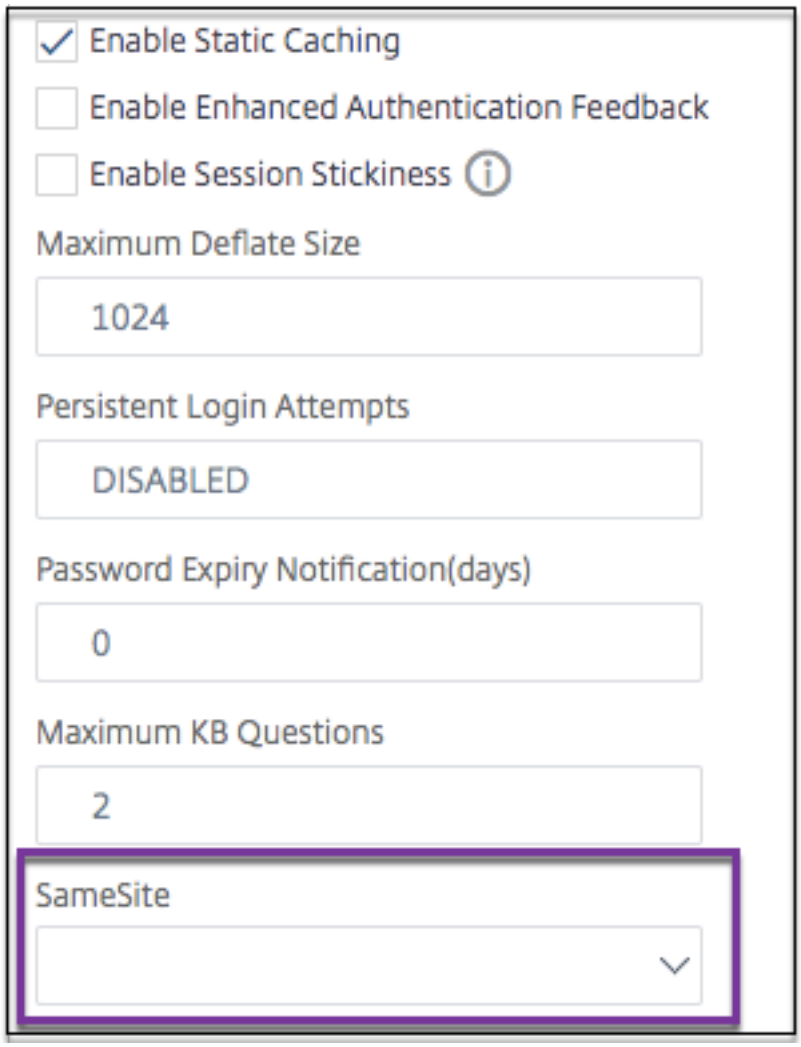
3. [基本設定] セクションの [編集] アイコンをクリックし、[詳細] をクリックします。
4. **SameSite** で、必要に応じてオプションを選択します。



グローバルレベルで **SameSite** 属性を設定するには**:**

1. セキュリティ **> AAA —** アプリケーショントラフィック **>** 認証設定の変更に移動します。

2. [**AAA** パラメータの設定] ページで、[**SameSite**] リストをクリックし、必要に応じてオプションを選択します。



一般的に使用されるプロトコルの認証、承認、および監査の構成

January 31, 2022

Citrix ADC アプライアンスを認証、承認、監査用に構成するには、Citrix ADC アプライアンスおよびクライアントのブラウザで特定の設定が必要です。設定は、認証、認可、および監査に使用されるプロトコルによって異なります。

Kerberos 認証用の Citrix ADC アプライアンスの構成の詳細については、「Kerberos/NTLM による認証、承認、および監査の処理」を参照してください。

Kerberos /NTLM による認証、承認、監査の処理

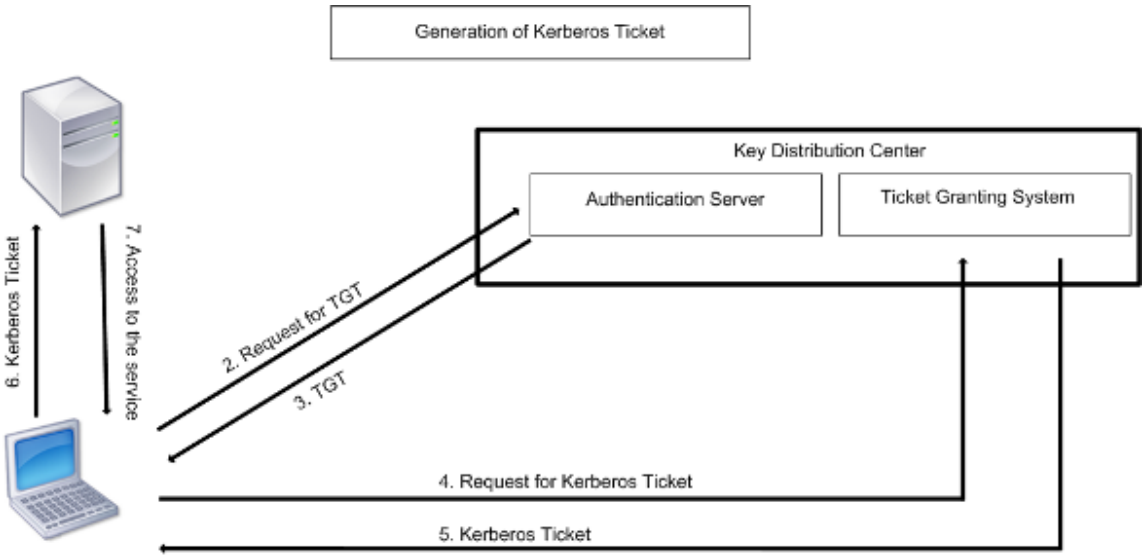
October 7, 2021

コンピューターネットワーク認証プロトコルである Kerberos は、インターネットを介した安全な通信を提供します。主にクライアント/サーバーアプリケーション用に設計されており、クライアントとサーバーがそれぞれ互いの信頼性を保証するための相互認証を提供します。Kerberos は、キー配布センター (KDC) と呼ばれる信頼できるサードパーティを使用します。KDC は、ユーザーを認証する認証サーバー (AS) と、チケット交付サーバー (TGS) で構成されます。

ネットワーク上の各エンティティ (クライアントまたはサーバー) には、自身と KDC のみが認識する秘密鍵があります。このキーの知識は、エンティティの真正性を意味します。ネットワーク上の 2 つのエンティティ間の通信のために、KDC は Kerberos チケットまたはサービスチケットと呼ばれるセッションキーを生成します。クライアントは、特定のサーバのクレデンシャルを AS に要求します。クライアントは、チケット交付チケット（TGT）と呼ばれるチケットを受け取ります。次に、クライアントは、AS から受信した TGT を使用して TGS に接続し、その ID を証明し、サービスを要求します。クライアントがサービスに適格である場合、TGS はクライアントに Kerberos チケットを発行します。次に、クライアントは Kerberos チケットを使用して、サービスをホストしているサーバー (サービスサーバーと呼ばれる) と通信し、サービスの受信が許可されていることを証明します。Kerberos チケットには、設定可能な有効期間があります。クライアントは AS で自身を認証するのは 1 回だけです。物理サーバに複数回接続すると、AS チケットが再利用されます。

次の図は、Kerberos プロトコルの基本的な機能を示しています。

図 1: **Kerberos** の機能



Kerberos 認証には、次の利点があります。

- 認証の高速化。物理サーバーがクライアントから Kerberos チケットを取得すると、サーバーはクライアントを直接認証するのに十分な情報を持ちます。クライアント認証のためにドメイン Controller に接続する必要がないため、認証プロセスが高速になります。
- 相互認証。KDC がクライアントに Kerberos チケットを発行し、クライアントがチケットを使用してサービスにアクセスする場合、認証されたサーバーのみが Kerberos チケットを復号化できます。Citrix ADC アプライアンス上の仮想サーバーが Kerberos チケットを復号化できる場合は、仮想サーバーとクライアントの両

方が認証されていると結論付けることができます。したがって、サーバーの認証は、クライアントの認証とともに行われます。

- Windows と Kerberos をサポートする他のオペレーティング・システムの間のシングル・サインオン。

Kerberos 認証には、次のような欠点があります。

- Kerberos には厳しい時間要件があります。認証が失敗しないように、関連するホストのクロックを Kerberos サーバのクロックと同期させる必要があります。ネットワークタイムプロトコルデーモンを使用してホストクロックの同期を維持することで、この欠点を緩和できます。Kerberos チケットには可用性期間があり、これを設定できます。
- Kerberos では、中央サーバを継続的に使用できるようにする必要があります。Kerberos サーバーがダウンすると、誰もログオンできません。複数の Kerberos サーバーとフォールバック認証メカニズムを使用することで、このリスクを軽減できます。
- すべての認証は一元化された KDC によって制御されるため、ローカルワークステーションのユーザーのパスワードが盗まれるなど、このインフラストラクチャに侵入すると、攻撃者が任意のユーザーを偽装する可能性があります。信頼するデスクトップマシンまたはラップトップのみを使用するか、ハードウェアトークンを使用して事前認証を適用することで、このリスクをある程度軽減できます。

Kerberos 認証を使用するには、Citrix ADC アプライアンスおよび各クライアントで認証を構成する必要があります。

認証、承認、監査での **Kerberos** 認証の最適化

Citrix ADC アプライアンスは、Kerberos 認証中にシステムパフォーマンスを最適化し、改善するようになりました。認証、承認、および監査デーモンは、同じユーザーに対する未処理の Kerberos 要求を記憶し、キー配布センター (KDC) への負荷を回避します。これにより、要求の重複を回避できます。

Citrix ADC がクライアント認証に **Kerberos** を実装する方法

October 7, 2021

> 重要
>
> Kerberos/NTLM 認証は、NetScaler 9.3 nCore リリース以降でのみサポートされており、トラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。

Citrix ADC は、Kerberos 認証に関連するコンポーネントを次のように処理します。

キー配布センター **(KDC)**

Windows 2000 サーバーまたはそれ以降のバージョンでは、ドメインコントローラーと KDC は Windows サーバーの一部です。Windows Server が起動し、実行している場合は、ドメインコントローラと KDC が構成されている

ことを示します。KDC は、Active Directory サーバーでもあります。

> 注
>
> すべての Kerberos の相互作用は、Windows Kerberos ドメインコントローラーで検証されます。

認証サービスとプロトコルのネゴシエーション

Citrix ADC アプライアンスは、認証、承認、監査トラフィック管理認証仮想サーバーでの Kerberos 認証をサポートします。Kerberos 認証が失敗した場合、Citrix ADC は NTLM 認証を使用します。

既定では、Windows 2000 Server 以降のバージョンの Windows Server では、認証、承認、および監査に Kerberos が使用されます。認証タイプとして NEGOTIATE を使用して認証ポリシーを作成すると、Citrix ADC は認証、承認、および監査に Kerberos プロトコルを使用しようとします。クライアントのブラウザが Kerberos チケットを受信できない場合、Citrix ADC は NTLM 認証を使用します。このプロセスは、ネゴシエーションと呼ばれます。

クライアントは、次のいずれかに該当する場合、Kerberos チケットの受信に失敗することがあります。

- Kerberos はクライアントでサポートされていません。
- Kerberos がクライアントで有効になっていません。
- クライアントが KDC 以外のドメインに存在する。
- KDC 上のアクセスディレクトリは、クライアントからアクセスできません。

Kerberos/NTLM 認証の場合、Citrix ADC は、Citrix ADC アプライアンスにローカルに存在するデータを使用しません。

承認

トラフィック管理仮想サーバは、負荷分散仮想サーバまたはコンテンツスイッチング仮想サーバです。

監査

Citrix ADC アプライアンスは、次の監査ログによる Kerberos 認証の監査をサポートします。

- トラフィック管理エンドユーザアクティビティの完全な監査証跡
- SYSLOG と高パフォーマンス TCP ロギング
- システム管理者の完全な監査証跡
- すべてのシステムイベント
- スクリプト可能なログ形式

サポートされる環境

Kerberos 認証では、Citrix ADC 上の特定の環境は必要ありません。クライアント（ブラウザ）が Kerberos 認証をサポートする必要があります。

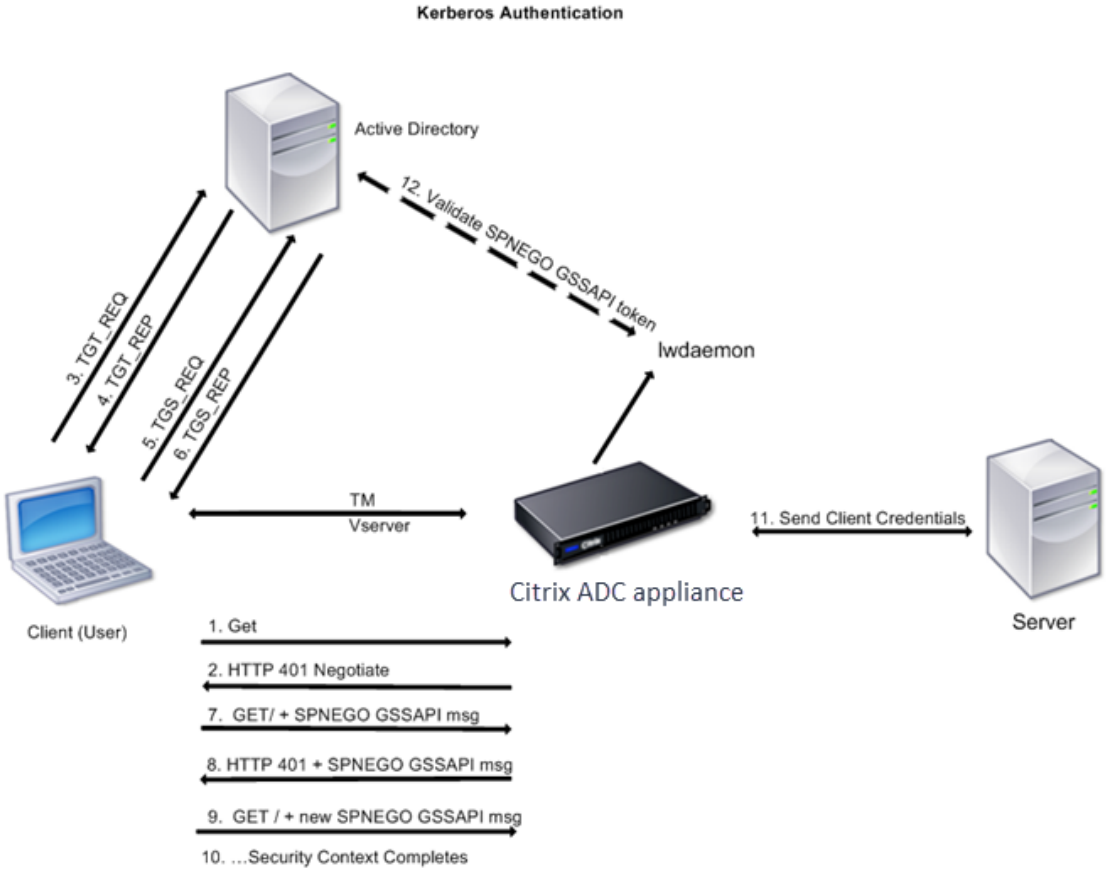
高可用性

高可用性セットアップでは、アクティブな Citrix ADC のみがドメインに参加します。フェイルオーバーの場合、Citrix ADC lwagent デーモンはセカンダリ Citrix ADC アプライアンスをドメインに参加させます。この機能に特定の設定は必要ありません。

Kerberos 認証プロセス

次の図は、Citrix ADC 環境における Kerberos 認証の一般的なプロセスを示しています。

図 1: Citrix ADC での Kerberos 認証プロセス



Kerberos 認証は、次の段階で行われます。

クライアントが **KDC** に対して自身を認証する

1. Citrix ADC アプライアンスは、クライアントから要求を受信します。
2. Citrix ADC アプライアンスのトラフィック管理（負荷分散またはコンテンツスイッチング）仮想サーバーは、クライアントにチャレンジを送信します。
3. このチャレンジに対応するために、クライアントは Kerberos チケットを取得します。

- クライアントは、KDC の認証サーバにチケット認可チケット (TGT) の要求を送信し、TGT を受信します。(図「Kerberos 認証プロセス」の 3、4 を参照してください)。
- クライアントは TGT を KDC のチケット交付サーバーに送信し、Kerberos チケットを受け取ります。(図「Kerberos 認証プロセス」の 5、6 を参照してください)。

> 注
>
> クライアントがライフタイムの期限が切れていない Kerberos チケットをすでに持っている場合、上記の認証プロセスは不要です。さらに、SPNEGO をサポートする Web サービス、.NET、または J2EE などのクライアントは、ターゲットサーバーの Kerberos チケットを取得し、SPNEGO トークンを作成し、HTTP 要求を送信するときに HTTP ヘッダーにトークンを挿入します。クライアント認証プロセスは行われません。

クライアントがサービスを要求します。

1. クライアントは、SPNEGO トークンと HTTP 要求を含む Kerberos チケットを Citrix ADC 上のトラフィック管理仮想サーバーに送信します。SPNEGO トークンに必要な GSSAPI データがあります。
2. Citrix ADC アプライアンスは、クライアントと Citrix ADC の間にセキュリティコンテキストを確立します。Citrix ADC が Kerberos チケットで提供されたデータを受け付けることができない場合、クライアントは別のチケットを取得するように求められます。このサイクルは、GSSAPI データが受け入れられ、セキュリティコンテキストが確立されるまで繰り返されます。Citrix ADC 上のトラフィック管理仮想サーバーは、クライアントと物理サーバー間の HTTP プロキシとして機能します。

Citrix ADC アプライアンスが認証を完了します。

1. セキュリティコンテキストが完了すると、トラフィック管理仮想サーバーは SPNEGO トークンを検証します。
2. 有効な SPNEGO トークンから、仮想サーバーはユーザー ID と GSS 資格情報を抽出し、認証デーモンに渡します。
3. 認証が成功すると、Kerberos 認証が完了します。

Citrix ADC アプライアンスでの **Kerberos** 認証の構成

October 7, 2021

このトピックでは、CLI と GUI を使用して Citrix ADC アプライアンスで Kerberos 認証を構成するための詳細な手順について説明します。

CLI での **Kerberos** 認証の設定

1. 認証、承認、および監査機能を有効にして、アプライアンスでトラフィックの認証を確実にします。

   ns-cli-prompt> **enable ns feature** AAA

2. キータブファイルを Citrix ADC アプライアンスに追加します。Kerberos 認証中にクライアントから受信したシークレットを復号化するには、キータブファイルが必要です。単一のキータブファイルには、Citrix ADC

アプライアンス上のトラフィック管理仮想サーバーにバインドされているすべてのサービスの認証の詳細が含まれています。

まず、Active Directory サーバーでキータブファイルを生成してから、Citrix ADC アプライアンスに転送します。

- Active Directory サーバーにログオンし、Kerberos 認証用のユーザーを追加します。たとえば、「Kerb-SVC-Account」という名前のユーザーを追加するには、次のようにします。

  ネットユーザー **Kerb-SVC** アカウント無料です！ **@ #456 /**追加

  > 注
  >
  > [ユーザーのプロパティ] セクションで、[次のログオン時にパスワードを変更する] オプションが選択されておらず、[パスワードの有効期限] オプションが選択されていることを確認します。

- 上記のユーザーに HTTP サービスをマッピングし、キータブファイルをエクスポートします。たとえば、Active Directory サーバーで次のコマンドを実行します。

  ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM /pass freebsd!@#456 /mapuser newacp\dummy /ptype KRB5_NT_PRINCIPAL

  > 注
  >
  > 複数のサービスに認証が必要な場合は、複数のサービスをマッピングできます。さらに多くのサービスをマッピングする場合は、すべてのサービスに対して上記のコマンドを繰り返します。出力ファイルには同じ名前を指定することも、別の名前を指定することもできます。

- unix **ftp** コマンドまたは任意のファイル転送ユーティリティを使用して、キータブファイルを Citrix ADC アプライアンスに転送します。

3. Citrix ADC アプライアンスは、完全修飾ドメイン名（FQDN）からドメイン Controller IP アドレスを取得する必要があります。したがって、Citrix ADC に DNS サーバーを構成することをお勧めします。

   ns-cli-prompt> **add dns nameserver** <ip-address>

   > 注
   >
   > または、静的なホストエントリを追加するか、その他の方法を使用して、Citrix ADC アプライアンスがドメイン Controller FQDN 名を IP アドレスに解決できるようにします。

4. 認証アクションを設定し、認証ポリシーに関連付けます。

   - ネゴシエートアクションを設定します。

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

注: ドメインユーザおよびドメイン名の設定については、クライアントに移動し、次の例に示すように klist コマンドを使用します。

クライアント: ユーザー名 @ AAA.LOCAL

サーバー:http/onprem_idp.AAA.local @ AAA.LOCAL

add authentication negotiateAction <name> -domain <AAA.LOCAL> -domainUser <HTTP/onprem_idp.aaa.local>

- ネゴシエートポリシーを設定し、ネゴシエートアクションをこのポリシーに関連付けます。

  ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. 認証仮想サーバを作成し、ネゴシエートポリシーをそのサーバに関連付けます。

   - 認証仮想サーバーを作成します。

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - ネゴシエートポリシーを認証仮想サーバにバインドします。

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. 認証仮想サーバをトラフィック管理（ロードバランシングまたはコンテンツスイッチング）仮想サーバに関連付けます。

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の設定を行うことができます。

7. 次の手順を実行して、設定を確認します。

   - FQDN を使用して、トラフィック管理仮想サーバにアクセスします。たとえば、サンプル
   - CLI でセッションの詳細を表示します。

     ns-cli-prompt> **show aaa session**

GUI での Kerberos 認証の設定

1. 認証、認可、および監査機能を有効にします。

   [システム] > [設定] に移動し、[基本機能の構成] をクリックして、認証、承認、監査機能を有効にします。

2. 上記の CLI 手順のステップ 2 の説明に従って、キータブファイルを追加します。

3. DNS サーバーを追加します。

   [トラフィック管理] **> [DNS] > [**ネームサーバー] に移動し、DNS サーバーの IP アドレスを指定します。

4. [ネゴシエート] アクションとポリシーを設定します。

   [セキュリティ] **>** **[AAA-**アプリケーショントラフィック**] >** **[**ポリシー**] >** **[**認証**] >** **[**高度なポリシー**] >** **[**ポリシー] に移動し、アクションタイプとして [ネゴシエート] を使用してポリシーを作成します。[**ADD**] をクリックして新しい認証ネゴシエートサーバを作成するか、[**Edit**] をクリックして既存の詳細を設定します。

5. ネゴシエートポリシーを認証仮想サーバにバインドします。

   [セキュリティ] **>** **[AAA-**アプリケーショントラフィック**] >** **[**仮想サーバ] に移動し、ネゴシエートポリシーを認証仮想サーバに関連付けます。

6. 認証仮想サーバをトラフィック管理（ロードバランシングまたはコンテンツスイッチング）仮想サーバに関連付けます。

   [トラフィック管理**] >** **[**ロードバランシング**] >** **[**仮想サーバー] に移動し、関連する認証設定を指定します。

   > 注
   >
   > コンテンツスイッチング仮想サーバーでも同様の設定を行うことができます。

7. 上記の CLI 手順のステップ 7 で詳述した設定を確認します。

クライアントで **Kerberos** 認証を構成する

October 7, 2021

認証に Kerberos を使用するには、ブラウザで Kerberos サポートを構成する必要があります。Kerberos 準拠の任意のブラウザを使用できます。Internet Explorer および Mozilla Firefox で Kerberos サポートを設定する手順は次のとおりです。その他のブラウザについては、ブラウザのドキュメントを参照してください。

Kerberos 認証用に **Internet Explorer** を構成するには

1. [ツール**]** メニューの **[**インターネットオプション**]** を選択します。
2. [セキュリティ] タブで、[ローカルイントラネット] をクリックし、[サイト] をクリックします。
3. [ローカルイントラネット] ダイアログボックスで、[イントラネットネットワークを自動的に検出する] オプションが選択されていることを確認し、[詳細設定] をクリックします。
4. [ローカルイントラネット] ダイアログボックスで、Citrix ADC アプライアンス上のトラフィック管理仮想サーバーのドメインの Web サイトを追加します。指定したサイトはローカルイントラネットサイトになります。
5. [閉じる**]** または **[OK]** をクリックしてダイアログボックスを閉じます。

Kerberos 認証用に **Mozilla Firefox** を設定するには

1. コンピューターに Kerberos が正しく構成されていることを確認します。
2. URL バーに about: config と入力します。

3. フィルタテキストボックスに「network.negotiate」と入力します。
4. 追加するドメインに、ネットワークを変更します。
5. 追加するドメインに、ネットワークを変更します。

> 注： Windows を実行している場合は、フィルターテキストボックスに sspi と入力し、network.auth.use-sspi オプションを False に変更する必要もあります。

物理サーバーから **Kerberos** 認証をオフロードします

January 31, 2022

Citrix ADC アプライアンスは、認証タスクをサーバーからオフロードできます。Citrix ADC は、クライアントからの要求を認証する物理サーバーではなく、すべてのクライアント要求を認証してから、バインドされた物理サーバーに転送します。ユーザー認証は、Active Directory トークンに基づいています。

Citrix ADC と物理サーバー間の認証は行われず、認証オフロードはエンドユーザーに対して透過的です。Windows コンピューターへの初回ログオン後、エンドユーザーは、ポップアップまたはログオンページに追加の認証情報を入力する必要はありません。

現在の Citrix ADC アプライアンスのリリースでは、Kerberos 認証は、トラフィック管理仮想サーバーの認証、承認、監査にのみ使用できます。Kerberos 認証は、Citrix Gateway アドバンスエディションアプライアンスでの SSL VPN または Citrix ADC アプライアンスの管理ではサポートされません。

Kerberos 認証では、Citrix ADC アプライアンスおよびクライアントブラウザでの構成が必要です。

Citrix ADC アプライアンスで **Kerberos** 認証を構成するには

1. Active Directory にユーザー・アカウントを作成します。ユーザー・アカウントを作成するときは、「ユーザー・プロパティ」セクションで次のオプションを確認します。
   - [次回ログオン時にパスワードを変更する] オプションを選択していないことを確認します。
   - [パスワードの有効期限切れ] オプションを必ず選択してください。
2. AD サーバーで、CLI コマンドプロンプトで次のように入力します。
   - ktpass-princ HTTP/ kerberos.crete.lab.net@crete.lab.net-ptype KRB5_NT_PRINCIPAL-マップユーザー-マップポップセット-<password> アウト C: kerbtabfile.txt kerbuser@crete.lab.net

   > 注
   >
   > 上記のコマンドを 1 行で入力してください。kerbtabfile.txt ファイル：上記のコマンドの出力は、C に書き込まれます。
3. セキュアコピー（SCP）クライアントを使用して、kerbtabfile.txt ファイルを Citrix ADC アプライアンスの /etc ディレクトリにアップロードします。

4. 次のコマンドを実行して、Citrix ADC アプライアンスに DNS サーバーを追加します。

   - DNS ネームサーバーを追加する 1.2.3.4

   Citrix ADC アプライアンスは、DNS サーバーがないと Kerberos 要求を処理できません。Microsoft Windows ドメインで使用されているものと同じ DNS サーバーを使用してください。

5. Citrix ADC コマンドラインインターフェイスに切り替えます。
6. 次のコマンドを実行して、Kerberos 認証サーバーを作成します。

   - 認証の追加ネゴシエーションアクション Kerberos サーバードメイン「crete.lab.net」-ドメインユーザーユーザーユーザーユーザーユーザーパスワード <password>-keytab /var/mykcd.keytab

   > 注
   >
   > キータブが使用できない場合は、ドメイン、ドメインユーザー、および-domainUserPasswd のパラメーターを指定できます。

7. 次のコマンドを実行して、ネゴシエーション・ポリシーを作成します。

   - `add authentication negotiatePolicy Kerberos-Policy ”REQ.IP.DESTIP == 192.168.17.200”KerberosServer<!--NeedCopy-->`

8. 次のコマンドを実行して、認証仮想サーバーを作成します。

   - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`

9. 次のコマンドを実行して、Kerberos ポリシーを認証仮想サーバーにバインドします。

   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`

10. 次のコマンドを実行して、SSL 証明書を認証仮想サーバーにバインドします。GUI の Citrix ADC アプライアンスからインストールできるテスト証明書のいずれかを使用できます。次のコマンドを実行して、ServerTestCert サンプル証明書を使用します。

    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`

11. IP アドレス 192.168.17.200 を使用して、HTTP ロードバランシング仮想サーバーを作成します。

    NetScaler 9.3 リリースが 9.3.47.8 より古い場合は、コマンドラインインターフェイスから仮想サーバーを作成してください。

12. 次のコマンドを実行して、認証仮想サーバーを構成します。

    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`

13. Web ブラウザのアドレスバーにホスト名 Example を入力します。

    Kerberos 認証がブラウザで設定されていないため、Web ブラウザに認証ダイアログボックスが表示されます。

    > 注
    >
    > Kerberos 認証には、クライアント上で特定の設定が必要です。クライアントがホスト名を解決できることを確認します。これにより、Web ブラウザが HTTP 仮想サーバに接続されます。

14. クライアントコンピュータの Web ブラウザで Kerberos を構成します。

    - Internet Explorer での構成については、「Kerberos 認証用の Internet Explorer の構成」を参照してください。
    - Mozilla Firefox での設定については、Internet Explorer の Kerberos 認証の設定を参照してください。

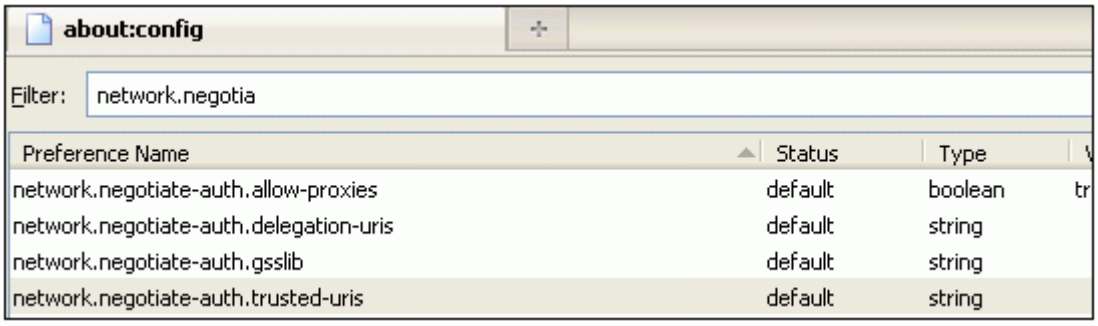
15. 認証なしでバックエンド物理サーバにアクセスできるかどうかを確認します。

Kerberos 認証用に **Internet Explorer** を構成するには

1. [ツール] メニューの [インターネットオプション] を選択します。
2. [セキュリティ] タブをアクティブにします。
3. [セキュリティ設定の変更を表示するゾーンの選択] セクションで、[ローカルイントラネット] を選択します。
4. [サイト] をクリックします。
5. [詳細設定] をクリックします。
6. URL、例] を指定し、[追加] をクリックします。
7. **Internet Explorer** を再起動します。

Kerberos 認証用に **Mozilla Firefox** を設定するには

1. ブラウザのアドレスバーに about: config と入力します。
2. 警告免責事項をクリックします。
3. [フィルタ] ボックスに「ネットワーク」と入力します。
4. [ネットワーク] をダブルクリックします。[ネゴシエート認証. 信頼された **uris]** をクリックします。画面の例を以下に示します。



5. [文字列値の入力] ダイアログボックスで、www.crete.lab.net と指定します。
6. Firefox を再起動します。

認証および認可に関連する問題のトラブルシューティング

September 2, 2022

エラーメッセージをローカライズする

Citrix ADC nFactor システムによって生成されたエラーメッセージをローカライズする

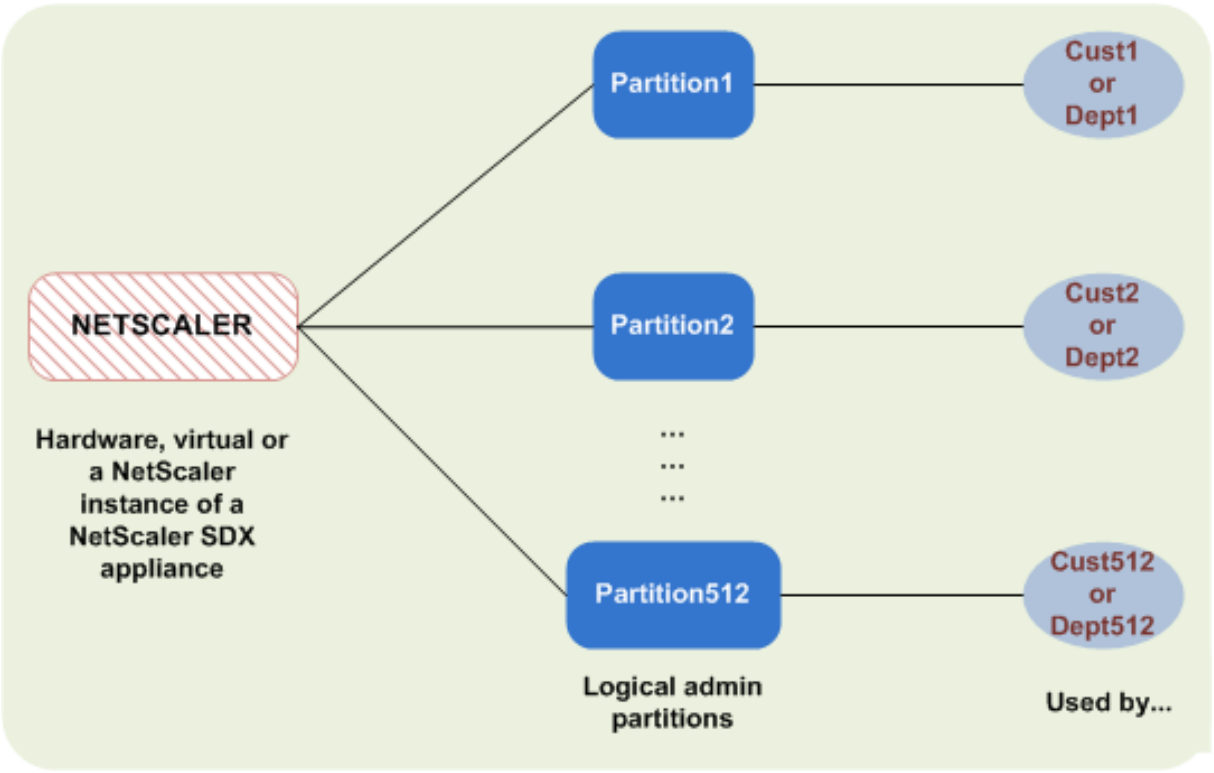
aaad.debug モジュールの認証に関する問題のトラブルシューティング

aaad.debug モジュールを使用した Citrix ADC および Citrix Gateway の認証に関する問題のトラブルシューティング

管理者パーティション

October 7, 2021

Citrix ADC アプライアンスは、管理パーティションと呼ばれる論理エンティティに分割できます。各パーティションは、個別の Citrix ADC アプライアンスとして構成および使用できます。次の図は、さまざまな顧客や部門で使用されている Citrix ADC パーティションを示しています。



パーティション化された Citrix ADC アプライアンスには、単一のデフォルトパーティションと 1 つ以上の管理パーティションがあります。次の表に、2 つのパーティションタイプの詳細を示します。

> 注
>
> パーティション化されたアプライアンスでは、モード BridgeBPDU はデフォルトパーティションでのみ有効になり、管理パーティションでは有効にできません。

可用性:

Citrix ADC アプライアンスには、デフォルトパーティションと呼ばれる単一のパーティションが付属しています。デフォルトのパーティションは、Citrix ADC アプライアンスのパーティション分割後も保持されます。

管理パーティションの構成の説明に従って、明示的に作成する必要があります。

パーティション数:

一つ

Citrix ADC アプライアンスは、1 つ以上（最大 512）の管理パーティションを持つことができます。

ユーザーアクセスとロール:

パーティション固有のコマンドポリシーに関連付けられていないすべての *Citrix* ADC ユーザーは、デフォルトのパーティションにアクセスして構成できます。いつものように、関連するコマンドポリシーは、ユーザーが実行できる操作を制限します。

ユーザーアクセスとロールは、そのパーティションのユーザーも指定する Citrix ADC スーパーユーザーによって作

成されます。管理者パーティションにアクセスして構成できるのは、スーパーユーザーおよびパーティションの関連付けられたユーザーだけです。

> 注
>
> パーティションユーザーにはシェルアクセス権がありません。

ファイル構造:

デフォルトパーティション内のすべてのファイルは、デフォルトの Citrix ADC ファイル構造に格納されます。

たとえば、*/nsconfig* ディレクトリには *Citrix* ADC 構成ファイルが格納され、*/var/log/* ディレクトリには Citrix ADC ログが格納されます。

管理パーティション内のすべてのファイルは、管理パーティションの名前を持つディレクトリパスに格納されます。

たとえば、Citrix ADC 構成ファイル（ns.conf）は/nsconfig/partitions/<partitionName>ディレクトリに保存されます。その他のパーティション固有のファイルは、/var/partitions/<partitionName> ディレクトリに格納されます。

管理パーティション内のその他のパスは次のとおりです。

- ダウンロードされたファイル: /var/partitions/<partitionName>/download/
- ログファイル: /var/partitions/<partitionName>/log/

> 注
>
> 現在、ロギングはパーティションレベルではサポートされていません。したがって、このディレクトリは空で、すべてのログが/var/log/ディレクトリに保存されます。

- SSL CRL 証明書関連ファイル: /var/partitions/<partitionName>/netscaler/ssl

利用可能なリソース:

すべての Citrix ADC リソース。

管理者パーティションに明示的に割り当てられている Citrix ADC リソース。

ユーザーアクセスとロール

パーティション化された Citrix ADC アプライアンスの認証と承認では、ルート管理者はパーティション管理者を 1 つ以上のパーティションに割り当てることができます。パーティション管理者は、他のパーティションに影響を与えずに、そのパーティションに対するユーザーを認証できます。パーティションユーザーは、SNIP アドレスを使用してそのパーティションにのみアクセスする権限があります。ルート管理者とパーティション管理者の両方が、異なるアプリケーションへのアクセスをユーザーに許可することで、ロールベースのアクセス (RBA) を構成できます。

管理者とユーザーロールは、次のように記述できます。

ルート管理者。パーティション化されたアプライアンスに NSIP アドレスを使用してアクセスし、1 つ以上のパーティションへのユーザーアクセスを許可できます。管理者は、パーティション管理者を 1 つ以上のパーティションに割

り当てることもできます。管理者は、NSIP アドレスを使用してデフォルトパーティションからパーティション管理者を作成するか、パーティションに切り替えて、ユーザーを作成し、SNIP アドレスを使用してパーティション管理者アクセスを割り当てることができます。

パーティション管理者。ルート管理者によって割り当てられた NSIP アドレスを使用して、指定されたパーティションにアクセスします。管理者は、パーティションのユーザーアクセスをパーティションにロールベースのアクセスを割り当てて、パーティション固有の設定を使用して外部サーバー認証を構成することもできます。

システムユーザー。NSIP アドレスを通してパーティションにアクセスします。ルート管理者によって指定されたパーティションとリソースにアクセスできます。

パーティションユーザー。SNIP アドレスを使用してパーティションにアクセスします。ユーザーアカウントはパーティション管理者によって作成され、ユーザーはパーティション内でのみリソースにアクセスできます。

確認事項

パーティションでロールベースのアクセスを提供する際に覚えておくべき点をいくつか挙げます。

1. NSIP アドレスを介して GUI にアクセスする Citrix ADC ユーザーは、デフォルトのパーティション認証構成を使用してアプライアンスにログオンします。
2. パーティション SNIP アドレスを使用して GUI にアクセスするパーティションシステムユーザは、パーティション固有の認証設定を使用してアプライアンスにログオンします。
3. パーティション内に作成されたパーティションユーザーは、NSIP アドレスを使用してログインできません。
4. パーティションにバインドされた Citrix ADC ユーザーは、パーティション SNIP アドレスを使用してログインできません。
5. 外部認証サーバ（LDAP、RADIUS、TACACS など）を介して認証するシステムユーザは、SNIP アドレスを介してパーティションにアクセスする必要があります。

パーティション設定でのロールベースのアクセスを管理するためのユースケース

企業組織 www.example.com に複数のビジネスユニットがあり、ネットワーク内のすべてのインスタンスを管理する一元管理者がいるシナリオを考えてみましょう。ただし、各ビジネスユニットに排他的なユーザー権限と環境を提供したいと考えています。

パーティションアプライアンスでデフォルトのパーティション認証設定およびパーティション固有の設定で管理される管理者とユーザーを次に示します。

ジョン: ルート管理者

ジョージ: パーティション管理者

Adam: システムユーザー

Jane: パーティションユーザー

John、パーティション化された Citrix ADC アプライアンスのルート管理者です。John は、アプライアンス内のパーティション（P1、P2、P3、P4、P5 など）にわたってすべてのユーザーアカウントと管理ユーザーアカウントを管

理します。John は、アプライアンスのデフォルトパーティションからエンティティへの詳細なロールベースのアクセスを提供します。John は、ユーザーアカウントを作成し、各アカウントにパーティションアクセスを割り当てます。組織内のネットワークエンジニアである George は、パーティション P2 で実行されているいくつかのアプリケーションに対して、ロールベースのアクセス権を持つことを好みます。ユーザー管理に基づいて、John は George のパーティション管理者ロールを作成し、P2 パーティション内の partition-admin コマンドポリシーに自分のユーザーアカウントを関連付けます。別のネットワークエンジニアであるアダムは、P2 で実行されているアプリケーションにアクセスすることを好みます。John は Adam のシステムユーザアカウントを作成し、ユーザアカウントを P2 パーティションに関連付けます。アカウントが作成されると、Adam はアプライアンスにログインして NSIP アドレスを介して Citrix ADC 管理インターフェイスにアクセスし、ユーザー/グループのバインドに基づいてパーティション P2 に切り替えることができます。

別のネットワークエンジニアである Jane が、パーティション P2 でのみ実行されているアプリケーションに直接アクセスしたいとします。George（パーティション管理者）は、自分のパーティションユーザーアカウントを作成し、自分のアカウントを認可権限のコマンドポリシーに関連付けることができます。パーティション内に作成された Jane のユーザーアカウントが P2 に直接関連付けられるようになりました。これで、Jane は SNIP アドレスを介して Citrix ADC 管理インターフェイスにアクセスでき、他のパーティションに切り替えることはできません。

> 注
>
> Jane のユーザーアカウントがパーティション P2 のパーティション管理者によって作成されている場合、管理者は SNIP アドレス（パーティション内に作成された）を介してのみ Citrix ADC 管理インターフェイスにアクセスできます。管理者は NSIP アドレスを介してインターフェイスにアクセスすることはできません。同様に、Adam のユーザーアカウントがデフォルトパーティションにルート管理者によって作成され、P2 パーティションにバインドされている場合です。管理者は、デフォルトパーティションで作成された NSIP アドレスまたは SNIP アドレス（管理アクセス権が有効）を介してのみ、Citrix ADC 管理インターフェイスにアクセスできます。また、管理パーティションに作成された SNIP アドレスを介してパーティションインターフェイスにアクセスすることは許可されていません。

パーティション管理者の役割と責任を構成する

ルート管理者がデフォルトパーティションで実行する構成を次に示します。

管理パーティションとシステムユーザーの作成 — ルート管理者は、アプライアンスのデフォルトパーティションに管理パーティションとシステムユーザーを作成します。その後、管理者はユーザーを異なるパーティションに関連付けます。1 つ以上のパーティションにバインドされている場合は、ユーザーバインディングに基づいて、あるパーティションから別のパーティションに切り替えることができます。また、1 つ以上のバインドされたパーティションへのアクセスは、ルート管理者によってのみ許可されます。

システムユーザーを特定のパーティションのパーティション管理者として承認する — ユーザーアカウントを作成すると、ルート管理者は特定のパーティションに切り替え、そのユーザーをパーティション管理者として承認します。このためには、partition-admin コマンドポリシーをユーザアカウントに割り当てます。これで、ユーザーはパーティション管理者としてパーティションにアクセスし、パーティション内のエンティティを管理できます。

次に、管理パーティション内のパーティション管理者が実行する構成を示します。

管理パーティションでの SNIP アドレスの設定-パーティション管理者はパーティションにログオンし、SNIP アドレスを作成し、アドレスへの管理アクセスを提供します。

パーティション・コマンド・ポリシーを使用したパーティション・システム・ユーザーの作成とバインド-パーティション管理者はパーティション・ユーザーを作成し、ユーザー・アクセスの範囲を定義します。このためには、ユーザーアカウントをパーティションコマンドポリシーにバインドします。

パーティションコマンドポリシーを使用したパーティションシステムユーザーグループの作成とバインド-パーティション管理者は、パーティションユーザーグループを作成し、ユーザーグループアクセスの範囲を定義します。これは、ユーザグループアカウントをパーティションコマンドポリシーにバインドすることによって行われます。

外部ユーザの外部サーバ認証の設定（オプション）：この設定は、SNIP アドレスを使用してパーティションにアクセスする外部 TACACS ユーザを認証するために行われます。

次に、管理パーティション内のパーティションユーザーに対するロールベースのアクセスの構成で実行されるタスクを示します。

1. 管理パーティションの作成 — 管理パーティションにパーティションユーザーを作成する前に、まずパーティションを作成する必要があります。ルート管理者は、構成ユーティリティまたはコマンドラインインターフェイスを使用して、デフォルトパーティションからパーティションを作成できます。
2. デフォルトパーティションからパーティション P2 へのユーザーアクセスの切り替え：デフォルトパーティションからアプライアンスにアクセスするパーティション管理者は、デフォルトパーティションから特定のパーティションに切り替えることができます。たとえば、ユーザーバインドに基づいて P2 をパーティション分割します。
3. 管理アクセス権が有効なパーティションユーザーアカウントに SNIP アドレスを追加する-管理パーティションへのアクセスを切り替えたら。SNIP アドレスを作成し、そのアドレスへの管理アクセスを提供します。
4. パーティションコマンドポリシーを使用したパーティションシステムユーザーの作成とバインド-パーティション管理者である場合は、パーティションユーザーを作成し、ユーザーアクセスの範囲を定義できます。このためには、ユーザーアカウントをパーティションコマンドポリシーにバインドします。
5. partition コマンドポリシーを使用したパーティション・ユーザー・グループの作成とバインド-パーティション管理者の場合は、パーティション・ユーザー・グループを作成し、ユーザー・アクセス制御の範囲を定義できます。これは、ユーザグループアカウントをパーティションコマンドポリシーにバインドすることによって行われます。

外部ユーザの外部サーバ認証の設定（オプション）：この設定は、SNIP アドレスを使用してパーティションにアクセスする外部 TACACS ユーザを認証するために行われます。

管理パーティションを使用する利点

デプロイメントに管理パーティションを使用すると、次のメリットを利用できます。

- アプリケーションの管理所有権を顧客に委任できます。
- パフォーマンスと使いやすさを損なうことなく、ADC の所有コストを削減します。

- 不当な構成変更から保護します。パーティション化されていない Citrix ADC アプライアンスでは、他のアプリケーションの承認されたユーザーは、アプリケーションに必要な構成を意図的または意図せずに変更できます。望ましくない動作につながる可能性があります。パーティション化された Citrix ADC アプライアンスでは、この可能性は軽減されます。
- パーティションごとに専用 VLAN を使用して、異なるアプリケーション間のトラフィックを分離します。
- アプリケーションのデプロイを高速化し、拡張できるようにします。
- アプリケーションレベルまたはローカライズされた管理とレポート作成を許可します。

いくつかのケースを分析して、管理パーティションを使用できるシナリオを理解しましょう。

ユーザーケース **1:** エンタープライズネットワークでの管理パーティションの使用方法

Foo.com という会社が直面するシナリオを考えてみましょう。

- **Foo.com** には **Citrix** ADC が 1 つあります。
- 5 つの部門があり、各部門には Citrix ADC で展開する必要がある 1 つのアプリケーションがあります。
- 各アプリケーションは、異なるユーザーまたは管理者によって個別に管理する必要があります。
- 他のユーザーは、設定へのアクセスを制限する必要があります。
- アプリケーションまたはバックエンドは、IP アドレスなどのリソースを共有できる必要があります。
- グローバル IT 部門は、すべてのパーティションに共通でなければならない Citrix ADC レベルの設定を制御できる必要があります。
- アプリケーションは互いに独立している必要があります。一方のアプリケーションの構成エラーは、他方のアプリケーションに影響してはいけません。

パーティション化されていない Citrix ADC は、これらの要件を満たすことができません。ただし、Citrix ADC をパーティション化することで、これらの要件をすべて達成できます。

アプリケーションごとにパーティションを作成し、必要なユーザをパーティションに割り当てて、各パーティションの VLAN を指定し、デフォルトパーティションでグローバル設定を定義するだけです。

ユースケース **2:** サービスプロバイダによる管理パーティションの使用方法

BigProvider というサービスプロバイダーが直面するシナリオを考えてみましょう。

- BigProvider には 5 つの顧客があります。3 つの小企業と 2 つの大企業です。
- **SmallBiz**、**SmallerBiz**、**StartupBiz** は、最も基本的な Citrix ADC 機能のみを必要とします。
- ****BigBiz** と **LargeBiz** は大企業であり ** 、大量のトラフィックを引き付けるアプリケーションを持っています。彼らは、より複雑な Citrix ADC 機能のいくつかを使用したいと考えています。

パーティション化されていないアプローチでは、Citrix ADC 管理者は通常、Citrix ADC SDX アプライアンスを使用し、顧客ごとに Citrix ADC インスタンスをプロビジョニングします。

このソリューションは、****BigBiz** および **LargeBiz** に適しています。これは、アプリケーションではパーティション化されていない **Citrix ADC** アプライアンス全体の電力が消費されない必要があるためです。ただし、このソリュ

ーションは、**SmallBiz**、****SmallerBiz**、および ****StartupBiz**** のサービスにはそれほど費用対効果がない可能性があります。

したがって、**BigProvider** は次の解決策を決定します。

- Citrix ADC SDX アプライアンスを使用して、****BigBiz** および **LargeBiz** 専用の **Citrix**ADC インスタンスを起動する。
- 単一の Citrix ADC を使用して、SmallBiz、**SmallerBiz**、****StartupBiz** にそれぞれ **1** つずつ **、3 つのパーティションに分割されます。

Citrix ADC 管理者（スーパーユーザー）は、これらの各顧客の管理パーティションを作成し、パーティションのユーザーを指定します。また、パーティションの Citrix ADC リソースを指定し、各パーティション宛てのトラフィックが使用する VLAN を指定します。

管理者パーティションでの **Citrix ADC** 構成サポート

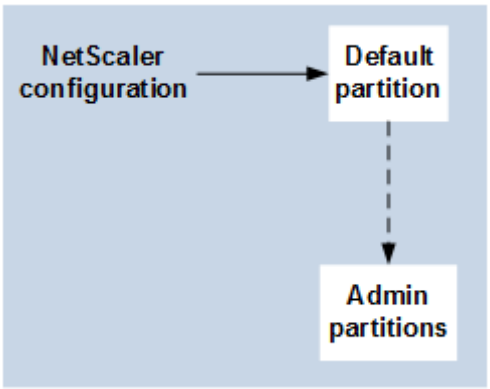
October 7, 2021

Citrix ADC 構成は、次の 3 種類の構成に分類できます。これは、Citrix の構成と構成が実行されるパーティションによって異なります。

> 注
>
> - Citrix ADC クラスターで管理パーティションを設定することはできません。これは、Citrix ADC クラスターをパーティション化できないことを意味します。
> - Citrix ADC 14000 FIPS アプライアンスでは管理パーティションを設定できません。
> - ケース 3 は 、管理パーティションでサポートされていない Citrix ADC 機能を示しています。
> - 負荷分散テンプレートは、管理パーティションではサポートされません。

ケース **1 (**グローバル構成**)**

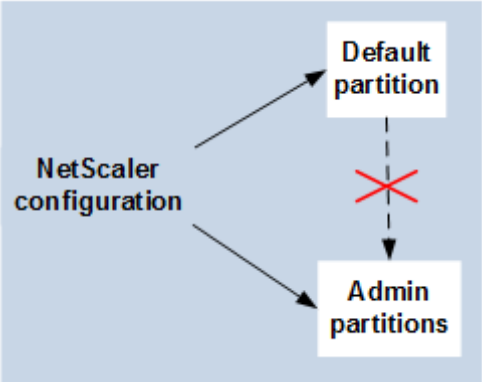
デフォルトパーティションでのみ実行でき、使用可能な設定、またはすべての管理パーティションに影響する設定。



- モニタ、TCP プロファイル、HTTP プロファイルなどの組み込みエンティティの更新。
- syslog、NSLOG、ウェブログ、コンテンツスイッチング、IPSEC、SIP、DHCP、サージ保護、TCP バッファリング、およびシステム収集のグローバルパラメータの更新。
- 高可用性（HA）構成
- インターフェイスと VLAN の変更
- ユーザー構成

ケース **2 (**パーティション固有の構成**)**

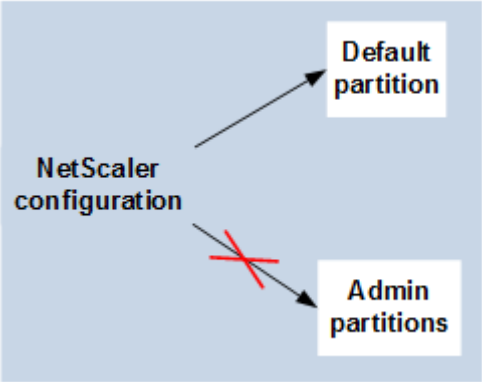
デフォルトパーティションと管理パーティションで独立して実行できる設定。これらの構成は、実行されているパーティションにのみ適用されます。



- パーティションのトラフィックレベルの統計情報の取得。
- パーティション管理者は、そのパーティションにバインドされている VLAN の IP バインディングを更新できます。ただし、インターフェイスバインディングを更新することはできません。
- Citrix ADC 構成をクリアします。
- AppFlow、AppQoE、HTTP 圧縮、DNS、TCP、HTTP、暗号化、レスポンダ、書き換え、SSL の機能固有のパラメーター。
- 仮想サーバ、サービス、モニタなどの機能固有の構成。

ケース **3**

管理パーティションでは実行できない設定。これらの機能はデフォルトパーティションで設定できますが、管理パーティションには影響しません。



> 注:
>
> 特定のリリースの管理パーティションでサポートされている設定は、**Yes** とマークされます。

フィーチャコンポーネント	Citrix ADC 機能	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
ネットワーク	トラフィックドメイン	いいえ (ビルド 60.13 以降ではサポートされていません)	いいえ	いいえ	いいえ
ポリシー	拡張性	はい	はい	はい	はい
負荷分散	DBS Autoscale	はい	はい	はい	はい
負荷分散	DNSSEC	いいえ	いいえ	はい	はい
負荷分散	Diameter	はい	はい	はい	はい
負荷分散	RTSP	いいえ	いいえ	いいえ	いいえ
負荷分散	確かに接続	はい	はい	はい	はい
負荷分散	オートスケールサービスグループ	はい	はい	はい	はい
管理性	RBA 外部認証	はい	はい	はい	はい
管理性	RISE Cisco	いいえ	いいえ	いいえ	いいえ
管理性	ACI-Cisco	はい	はい	はい	はい
管理性	AppExpert	はい	はい	はい	はい
管理性	HDX Insight	いいえ	いいえ	いいえ	いいえ
管理性	インサイト	いいえ	いいえ	いいえ	いいえ

フィーチャコンポーネント	Citrix ADC 機能	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
VPN	Citrix CloudBridge Connector	いいえ	いいえ	いいえ	いいえ
VPN	Citrix Gateway または SSL VPN	いいえ	いいえ	いいえ	いいえ
VPN	SSL VPN ICA プロキシ	いいえ	いいえ	いいえ	いいえ
VPN	Citrix ADC 上の Web インターフェイス	いいえ	いいえ	いいえ	いいえ
SSL	SSL プロファイル	はい	はい	はい	はい
SSL	SSL-FIPS	いいえ	いいえ	いいえ	いいえ
SSL	外部 HSM	いいえ	いいえ	いいえ	いいえ
インフラ	キャッシュリダイレクト	いいえ	いいえ	いいえ	いいえ
インフラ	統合キャッシング	はい	はい	はい	はい
ネットワーク	VXLAN	はい	はい	はい	はい
ネットワーク	グレースフルシャットダウン	はい	はい	はい	はい
ネットワーク	LSN	いいえ	いいえ	いいえ	いいえ
ネットワーク	IPv6 レディロゴ	はい	はい	はい	はい
ネットワーク	vPath	はい	はい	はい	はい
負荷分散	データストリーム	はい	はい	はい	はい
ログ	Web ログ	はい	はい	はい	はい
ネットワーク	L2 Param/L3 Param	はい	はい	はい	はい
ネットワーク	GRE トンネル	はい	はい	はい	はい

フィーチャコンポーネント	Citrix ADC 機能	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
ローディングバランシング	スクリプト可能なモニタリング	はい	はい	はい	はい
負荷分散	GSLB か	はい	はい	はい	はい
インフラ	接続ミラーリング	はい	はい	はい	はい
インフラ	FEO	はい	はい	はい	はい
インフラ	Ns トレース	はい	はい	はい	はい
負荷分散	優先度によるキューイング	はい	はい	はい	はい
ネットワーク	HDOSP	はい	はい	はい	はい
ネットワーク	ネットプロファイル	はい	はい	はい	はい
ネットワーク	ネットワーキング (制限付き機能)	はい	はい	はい	はい
ネットワーク	VRRP (制限付き機能)	はい	はい	はい	はい
ログ	監査ロギング (SYSLOG-TCP、Syslog サーバの LB、SNIP サポート、および Syslog の FQDN サポート)	はい	はい	はい	はい
VPN	Citrix Gateway	いいえ	いいえ	いいえ	いいえ
VPN	AAA-TM	はい	はい	はい	はい
AppFlow	AppFlow	いいえ	はい (IPFIX のみ)	はい (IPFIX のみ)	はい
appFW	アプリケーションファイアウォール	いいえ	いいえ	いいえ	いいえ

フィーチャコンポーネント	Citrix ADC 機能	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
URL 変換	URL 変換	いいえ	いいえ	いいえ	いいえ
負荷分散	TCP バッファリング	いいえ	いいえ	いいえ	いいえ
ポリシー	OCSP レスポンダー	はい	はい	はい	はい
監査ログ	SYSLOG-TCP	いいえ	はい	はい	はい
最適化	フロントエンド最適化	いいえ	はい	はい	はい
AppQoE	AppQoE	はい	はい	はい	はい
ボット	ボット管理	-	-	-	いいえ

前の表では、管理パーティション設定の「制限付き機能」としての機能の一部を示します。次のセクションでは、一部の機能が制限付き機能として言及されている理由について説明します。

- **VRRP**。VRRP は、次の理由により、管理パーティションの [制限付き機能] です。
  - VRID の追加または削除は、デフォルトのパーティションコンテキストからのみ実行できます。ただし、VRID が作成されると、デフォルト以外のパーティション内で使用できます。
  - VRRP 機能は、専用 VLAN でのみサポートされます。
  - VRRP 機能は、管理パーティションで使用される共有 VLAN ではサポートされません。内部でブロックされています。設定中にエラーメッセージは表示されません。プロトコルは、デフォルトパーティションまたは任意の管理パーティションにバインドされた共有 VLAN（タグ付きまたはタグなし）でブロックされます。

> 重要
>
> VRRP を使用したアクティブ/アクティブ展開をサポートするには、メイン VIP とバックアップ VIP で同じ VRID を使用する必要があります。異なる VRD は使用できません。

- ネットワーキング。一部のネットワーク構成（L2 Param および L3 Param）は、パーティションコンテキストでサポートされていないか、有効ではありません。このような構成に遭遇すると、次のエラーメッセージが表示されます。「エラー: この設定オプションは、デフォルト以外のパーティションではサポートされていません。」

管理パーティションを構成する

October 7, 2021

> 重要
>
> - 管理パーティションの作成と構成は、スーパーユーザーのみに許可されます。
> - 特に指定がない限り、管理パーティションをセットアップするための設定は、デフォルトパーティションから行う必要があります。

Citrix ADC アプライアンスをパーティション化すると、実際には単一の Citrix ADC アプライアンスの複数のインスタンスが作成されます。各インスタンスには独自の設定があり、これらの各パーティションのトラフィックは他のパーティションから分離されます。これは、各パーティションに専用 VLAN または共有 VLAN を割り当てることによって行われます。

パーティション化された Citrix ADC には、1 つのデフォルトパーティションと作成される管理パーティションがあります。管理パーティションを設定するには、まず、関連するリソース（メモリ、最大帯域幅、および接続）を持つパーティションを作成する必要があります。次に、パーティションにアクセスできるユーザーと、パーティション上の各ユーザーの承認レベルを指定します。

パーティション化された Citrix ADC へのアクセスは、NSIP アドレスまたはその他の管理 IP アドレスを使用して、パーティション化されていない Citrix ADC にアクセスするのと同じです。ユーザーとして、有効なログオン資格情報を入力すると、バインド先のパーティションに連れて行かれます。作成した構成は、そのパーティションに保存されます。複数のパーティションに関連付けられている場合は、関連付けられた最初のパーティションが表示されます。他のパーティションの 1 つにエンティティを構成する場合は、そのパーティションに明示的に切り替える必要があります。

適切なパーティションにアクセスすると、実行する構成はそのパーティションに保存され、そのパーティションに固有になります。

> 注
>
> - Citrix ADC スーパーユーザーおよびその他の非パーティションユーザーは、デフォルトのパーティションになります。
> - 512 パーティションのユーザーは同時にログインできます。

ヒント

SNIP（管理アクセスが有効）を使用して HTTPS 経由でパーティション化された Citrix ADC アプライアンスにアクセスするには、各パーティションにパーティション管理者の証明書があることを確認してください。パーティション内では、パーティション管理者は次の操作を行う必要があります。

1. 証明書を Citrix ADC に追加します。

```
add ssl certKey ns-server-certificate -cert ns-server.cert-key ns-
server.key
```


2. これはサービスnshttps-<SNIP>-3009にバインドされます。<SNIP>は SNIP アドレス（この場合は 100.10.10.1）に置き換える必要があります。

   ```
   bind ssl service nshttps-100.10.10.1-3009 -certkeyName ns-server-
   certificate
   ```

パーティションリソースの制限

パーティション化された Citrix ADC アプライアンスでは、ネットワーク管理者は、メモリ、帯域幅、接続制限などのパーティションリソースが無制限として構成されたパーティションを作成できます。これは、パーティションリソースの値として Zero を指定することによって行われます。ゼロは、パーティション上でリソースが無制限であり、システム制限まで消費できることを示します。パーティションリソース構成は、トラフィックドメインの展開を管理パーティションに移行する場合や、特定の展開内のパーティションのリソース割り当て制限がわからない場合に便利です。

管理パーティションのリソース制限は次のとおりです。

1. パーティションメモリ。これは、パーティションに割り当てられる最大メモリです。パーティションを作成するときは、必ず値を指定してください。

   > 注
   >
   > NetScaler 12.0 以降では、パーティションを作成するときに、メモリ制限をゼロに設定できます。特定のメモリ制限を使用してパーティションがすでに作成されている場合は、制限を任意の値に減らすか、制限をゼロに設定できます。

   パラメータ:maxMemLimit

   パーティション内の最大メモリは MB 単位で割り当てられます。ゼロの値は、パーティション上のメモリが無制限であり、システム制限まで消費できることを示します。

   デフォルト値:10

2. パーティション帯域幅。パーティションに割り当てられる最大帯域幅。制限を指定する場合は、アプライアンスのライセンススループット内であることを確認してください。それ以外の場合、パーティションで使用される帯域幅は制限されません。指定された制限は、アプリケーションが必要とする帯域幅について責任があります。アプリケーションの帯域幅が指定された制限を超えると、パケットはドロップされます。

   > 注
   >
   > NetScaler 12.0 以降では、パーティションを作成できるときに、パーティションの帯域幅制限をゼロに設定できます。パーティションが特定の帯域幅で既に作成されている場合は、帯域幅を減らすか、制限をゼロに設定できます。

   パラメータ: 最大帯域幅

   パーティション内の最大帯域幅は Kbps 単位で割り当てられます。ゼロの値は、帯域幅が制限されていないことを示します。つまり、パーティションはシステム制限まで消費する可能性があります。

   デフォルト値:10240

最大値:4294967295

3. パーティション接続。パーティションで開くことができる同時接続の最大数。この値は、パーティション内で予想される最大同時フローに対応する必要があります。パーティション接続は、パーティションクォータメモリから計算されます。以前は、接続はデフォルトのパーティションクォータメモリから計算されていました。これは、バックエンドのサーバー側の TCP 接続ではなく、クライアント側でのみ構成されます。この設定値を超えると、新しい接続を確立できません。

   > 注
   >
   > NetScaler 12.0 以降では、開いている接続数がゼロに設定されたパーティションを作成できます。特定の数のオープン接続を持つパーティションをすでに作成している場合は、接続制限を減らすか、制限をゼロに設定できます。

   パラメータ: 最大接続

   パーティションで開くことができる同時接続の最大数。ゼロの値は、開いている接続の数に制限がないことを示します。

   デフォルト値:1024

   最小値:0

   最大値:4294967295

管理パーティションを構成する

管理パーティションを設定するには、次のタスクを実行します。

CLI を使用して管理パーティションでアクセスするには

1. Citrix ADC アプライアンスにログオンします。
2. 正しいパーティションに入っているかどうかを確認します。コマンドプロンプトに、現在選択されているパーティションの名前が表示されます。
3. はいの場合は、次のステップに進みます。
4. いいえ、関連付けられているパーティションのリストを取得し、適切なパーティションに切り替えます。
   - `show system user <username>`
   - **`switch`** `ns partition <partitionName>`
5. これで、パーティション化されていない Citrix ADC として必要な構成を実行できます。

GUI を使用して管理パーティションにアクセスするには

1. Citrix ADC アプライアンスにログオンします。

2. 正しいパーティションに入っているかどうかを確認します。GUI の上部バーには、現在選択されているパーティションの名前が表示されます。
   - はいの場合は、次のステップに進みます。
   - [いいえ] の場合は、[構成] **>** [システム] **>** [パーティション管理] ****[**パーティション] に移動し、切り替えるパーティションを右クリックし、**[**** 切り替え] を選択します。
3. これで、パーティション化されていない Citrix ADC として必要な構成を実行できます。

管理者パーティションを追加する

ルート管理者は、デフォルトパーティションから管理パーティションを追加し、そのパーティションを VLAN 2 にバインドします。

CLI を使用して管理パーティションを作成するには

コマンドプロンプトで入力します。

```
1 add partition <partitionname>
```

ユーザーアクセスをデフォルトパーティションから管理パーティションに切り替える

これで、ユーザーアクセスをデフォルトパーティションからパーティション Par1 に切り替えることができます。

CLI を使用してユーザアカウントをデフォルトパーティションから管理パーティションに切り替えるには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 Switch ns partition <pname>
```

管理アクセスが有効になっているパーティションユーザーアカウントに **SNIP** アドレスを追加する

パーティションで、管理アクセスが有効な SNIP アドレスを作成します。

コマンドラインインターフェイスを使用して管理アクセスが有効になっているパーティションユーザーアカウントに **SNIP** アドレスを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
> add ns ip <ip address> <subnet mask> -mgmtAccess enabled
```


パーティションコマンドポリシーを使用してパーティションユーザーを作成してバインドする

パーティションで、パーティションシステムユーザーを作成し、partition-admin コマンドポリシーでユーザーをバインドします。

CLI を使用して **partition** コマンドポリシーを使用してパーティションシステムユーザーを作成してバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
> add system user <username> <password>

Done
```

partition コマンドポリシーを使用したパーティションユーザーグループの作成とバインド

パーティション Par1 で、パーティションシステムユーザーグループを作成し、パーティション管理、パーティション読み取り専用、パーティションオペレータ、パーティションネットワークなどのパーティションコマンドポリシーを使用してグループをバインドします。

コマンドラインインターフェイスを使用して、パーティションコマンドポリシーを使用してパーティションユーザーグループを作成し、バインドするには:

```
1 > add system group <groupName>
2 > bind system group <groupname> (-userName | -policyName <cmdpolicy> <
    priority> | -partitionName)
```

外部ユーザの外部サーバ認証の設定

パーティション Par1 では、外部サーバ認証を設定して、SNIP アドレスを介してパーティションにアクセスする外部 TACACS ユーザを認証できます。

コマンドラインインターフェイスを使用して外部ユーザーの外部サーバー認証を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 > add authentication tacacsaction <name> -serverip <IP> -tacacsSecret <
    secret key> -authorization ON -accounting ON
2 > add authentication policy <policname> -rule true -action <name>
3 > bind system global <policyname> -priority <value>1
```


GUI を使用して、パーティション内のパーティションシステムユーザーアカウントを構成します

管理パーティションでパーティションユーザーアカウントを構成するには、パーティションユーザーまたはパーティションユーザーグループを作成し、パーティションコマンドポリシーをバインドする必要があります。また、外部ユーザの外部サーバ認証を構成することもできます。

GUI を使用してパーティションにパーティションユーザーアカウントを作成するには

[システム] **>** [ユーザー管理] に移動し、[ユーザー] をクリックしてパーティションシステムユーザーを追加し、コマンドポリシー (partitionadmin/partition/partition-ad-only/パーティション演算子/パーティションネットワーク) にユーザーをバインドします。

GUI を使用してパーティションにパーティションのユーザーグループアカウントを作成するには

[システム] **>** [ユーザー管理] に移動し、[グループ] をクリックしてパーティションシステムのユーザーグループを追加し、ユーザーグループをコマンドポリシー (partitionadmin/partition/partition-read-only/パーティション演算子/パーティションネットワーク) にバインドします。

GUI を使用して外部ユーザの外部サーバ認証を設定するには

[システム] **>** [認証] **>** [基本アクション] に移動し、[**TACACS**] をクリックして、パーティションにアクセスする外部ユーザを認証するための TACACS サーバを設定します。

構成例

次の構成では、パーティションユーザーまたはパーティションユーザーグループを作成し、パーティションコマンドポリシーをバインドする方法を示します。また、外部ユーザを認証するための外部サーバ認証の設定方法についても説明します。

```
1  > add partition Par1
2  > switch ns partition Par1
3  > add ns ip 10.102.29.203 255.255.255.0 -mgmtAccessenabled
4  > add system user John Password
5  > bind system user Jane partition-read-only -priority 1
6  > add system group Retail
7  > bind system group Retail -policyname partition-network 1 (where 1 is
       the priority number)
8  > bind system group Retail  - username Jane
9  > add authentication tacacssaction tacuser  - serverip 10.102.29.200  -
       tacacsSecret Password  - authorization ON  - accounting ON
10 > add authentication policy polname  - rule true  - action tacacsAction
```



```
11  > bind system global polname  -priority 1
```

管理パーティション内のパーティションユーザーおよびパーティションユーザーグループのコマンドポリシー

管理パーティション内のユーザアカウントを認証するコマンド	管理パーティション内で使用可能なコマンドポリシー (組み込みポリシー)	ユーザーアカウントアクセスタイプ
add system user	パーティション管理	SNIP（管理アクセスが有効になっている場合）
add system group	パーティションネットワーク	SNIP（管理アクセスが有効になっている場合）
add authentication <action, policy>, bind system global <policy name>	パーティション-読み取り専用	SNIP（管理アクセスが有効になっている場合）
remove system user	パーティション管理	SNIP（管理アクセスが有効になっている場合）
remove system group	パーティション管理	SNIP（管理アクセスが有効になっている場合）
bind system cmdpolicy to system user; bind system cmdpolicy to system group	パーティション管理	SNIP（管理アクセスが有効になっている場合）

デフォルトの管理パーティションで **LACP** イーサネットチャネルを設定します

リンク集約制御プロトコル（LACP）を使用すると、複数のポートを単一の高速リンク（チャネルとも呼ばれる）に結合できます。LACP 対応アプライアンスは、チャネルを介して LACP データユニット（LACPDU）を交換します。

Citrix ADC アプライアンスのデフォルトパーティションで有効にできる LACP 構成モードは 3 つあります。

1. アクティブ。アクティブモードのポートは LACPDU を送信します。リンクアグリゲーションは、イーサネットリンクのもう一方の端が LACP アクティブモードまたはパッシブモードにある場合に形成されます。
2. パッシブ。パッシブモードのポートは、lacPDU を受信したときにのみ lacPDU を送信します。リンクアグリゲーションは、イーサネットリンクのもう一方の端が LACP アクティブモードにある場合に形成されます。
3. 無効化。リンクアグリゲーションは形成されません。

> 注
>
> デフォルトでは、リンクアグリゲーションはアプライアンスのデフォルトパーティションで無効になっています。

LACP は、イーサネットリンクによって接続されたデバイス間で LACPDU を交換します。これらのデバイスは、通常、アクターまたはパートナーと呼ばれます。

LACPDU データユニットには、次のパラメータが含まれています。

- LACP モード。アクティブ、パッシブ、または無効。
- LACP タイムアウト。パートナーまたは俳優がタイムアウトするまでの待機期間。指定可能な値: ロングとショート。デフォルト: 長い。
- ポートキー。異なるチャネルを区別する。キーが 1 の場合、LA/1 が作成されます。キーが 2 の場合、LA/2 が作成されます。指定可能な値:1 から 8 までの整数。4 ～8 はクラスタ CLAG 用です。
- ポートプライオリティ。最小値:1。最大値:65535 デフォルト:32768。
- システム優先度。このプライオリティをシステム MAC とともに使用して、パートナーとの LACP ネゴシエーション中にシステムを一意に識別するシステム ID を形成します。システムプライオリティを 1 および 65535 から設定します。デフォルト値は 32768 に設定されています。
- インターフェイス。NetScaler 10.1 アプライアンスではチャネルごとに 8 つのインターフェイスをサポートし、NetScaler 10.5 および 11.0 アプライアンスではチャネルごとに 16 インターフェイスをサポートします。

LACPDU を交換した後、アクターとパートナーは設定をネゴシエートし、集約にポートを追加するかどうかを決定します。

LACP の設定と確認

次のセクションでは、管理パーティションで LACP を設定して検証する方法について説明します。

CLI を使用して **Citrix ADC** アプライアンスで **LACP** を構成して検証するには

1. 各インターフェイスで LACP を有効にします。

   ```
   set interface <Interface_ID> -lacpMode PASSIVE -lacpKey 1<!--NeedCopy
   -->
   ```

   インターフェイスで LACP を有効にすると、チャネルが動的に作成されます。また、インターフェイスで LACP を有効にして lacpKey を 1 に設定すると、インターフェイスはチャネル LA/1 に自動的にバインドされます。

   > 注
   >
   > インターフェイスをチャネルにバインドすると、チャネルパラメータがインターフェイスパラメータよりも優先されるため、インターフェイスパラメータは無視されます。チャネルが LACP によって動的に作成される場合、チャネルに対して追加、バインド、アンバインド、または削除の各操作を実行できま

> せん。LACP によって動的に作成されたチャネルは、チャネルのすべてのインターフェイスで LACP を無効にすると、自動的に削除されます。

2. システムプライオリティを設定します。

```
set lacp -sysPriority <Positive_Integer><!--NeedCopy-->
```

3. LACP が期待どおりに動作していることを確認します。

“‘show interface

```
1 ```show channel<!--NeedCopy-->
```

```
show LACP<!--NeedCopy-->
```

> 注
>
> Cisco インターネットワークオペレーティングシステム（iOS）の一部のバージョンでは、switchport trunk native VLAN <VLAN_ID> コマンドを実行すると、Cisco スイッチに LACP PDU のタグが付けられます。これにより、Cisco スイッチと Citrix ADC アプライアンス間の LACP チャネルに障害が発生します。ただし、この問題は、前の手順で設定したスタティックリンク集約チャネルには影響しません。

デフォルトパーティションのすべての管理パーティションの設定を保存する

管理者は、デフォルトパーティションからすべての管理パーティションの設定を一度に保存できます。

CLI を使用して、デフォルトパーティションからすべての管理パーティションを保存する

コマンドプロンプトで入力します。

```
save ns config -all
```

パーティションおよびクラスタベースのカスタムレポートのサポート

Citrix ADC GUI には、現在の表示パーティションまたはクラスタで作成されたカスタムレポートのみが表示されます。

以前は、Citrix ADC GUI は、区別するパーティション名またはクラスタ名を指定せずに、カスタムレポート名をバックエンドファイルに直接保存するために使用されていました。

GUI で現在のパーティションまたはクラスタのカスタムレポートを表示するには

- [レポート] タブに移動します。
- [カスタムレポート] をクリックして、現在のパーティションまたはクラスタで作成されたレポートを表示します。

OAuth IdP のパーティション設定で **VPN** グローバル証明書をバインドするサポート

パーティション設定で、OAuth IdP 展開用の VPN グローバルに証明書をバインドできるようになりました。

CLI を使用してパーティション設定で証明書をバインドするには

コマンドプロンプトで入力します。

```
bind vpn global [-certkeyName <string>] [-userDataEncryptionKey <string
    >]
```

管理パーティションの **VLAN** 設定

April 7, 2022

VLAN は、「専用」VLAN または「共有」VLAN としてパーティションにバインドできます。配置に基づいて、VLAN をパーティションにバインドして、ネットワークトラフィックを他のパーティションから分離できます。

専用 **VLAN–**「共有」オプションが無効になっている 1 つのパーティションにのみバインドされている VLAN であり、タグ付き VLAN である必要があります。たとえば、クライアント/サーバ配置では、セキュリティ上の理由から、システム管理者がサーバ側のパーティションごとに専用 VLAN を作成します。

共有 **VLAN–**「共有」オプションが有効になっている複数のパーティションにバインドされた（共有された）VLAN。たとえば、クライアント/サーバ配置では、システム管理者がクライアント側ネットワークを制御できない場合、VLAN が作成され、複数のパーティション間で共有されます。

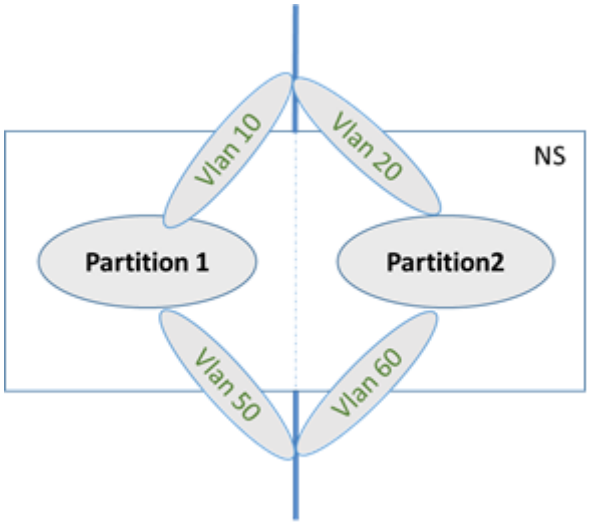
共有 VLAN は、複数のパーティションにわたって使用できます。デフォルトパーティションで作成され、共有 VLAN を複数のパーティションにバインドできます。デフォルトでは、共有 VLAN はデフォルトパーティションに暗黙的にバインドされるため、明示的にバインドすることはできません。

> 注
>
> - 任意のハイパーバイザー（ESX、KVM、Xen、および Hyper-V）プラットフォームに展開された Citrix ADC アプライアンスは、パーティション設定とトラフィックドメインにおける次の条件の両方に準拠している必要があります。
>   - Enable the promiscuous mode, MAC changes, MAC spoofing, or forged transmit for shared VLANs with partition.
>   - Enable the VLAN with port group properties of the virtual switch, if the traffic is through a dedicated VLAN.
> - パーティション化された（マルチテナント）Citrix ADC アプライアンスでは、システム管理者は特定の 1

> つまたは複数のパーティションに流れるトラフィックを分離できます。これは、1 つ以上の VLAN を各パーティションにバインドすることによって行われます。VLAN は、1 つのパーティション専用にすることも、複数のパーティション間で共有することもできます。

専用 **VLAN**

パーティションに流れるトラフィックを分離するには、VLAN を作成し、パーティションに関連付けます。これにより、VLAN は関連付けられたパーティションだけに認識され、VLAN を通過するトラフィックは関連付けられたパーティションでのみ分類され、処理されます。



特定のパーティションに専用 VLAN を実装するには、次の手順を実行します。

1. VLAN（V1）を追加します。
2. ネットワークインターフェイスをタグ付きネットワークインターフェイスとして VLAN にバインドします。
3. パーティション (P1) を作成します。
4. パーティション（P1）を専用 VLAN（V1）にバインドします。

CLI を使用して以下を構成します

- VLAN を作成する

  ```
  add vlan <id>
  ```

例

```
1     add vlan 100
```

- VLAN をバインドする

  ```
  bind vlan <id> -ifnum <interface> -tagged
  ```


例

```
1     bind vlan 100  -ifnum 1/8 -tagged
```

- パーティションを作成する

  ```
  Add ns partition <partition name> [-maxBandwidth <positive_integer>][-maxConn <positive_integer>] [-maxMemLimit <positive_integer>]
  ```

例

```
1     Add ns partition P1  -maxBandwidth 200  -maxconn 50  -maxmemlimit
         90
2
3     Done
```

- パーティションを VLAN にバインドする

  ```
  bind partition <partition-id> -vlan <id>
  ```

例

```
1     bind partition P1  -vlan 100
```

Citrix ADC GUI を使用して専用 VLAN を構成します

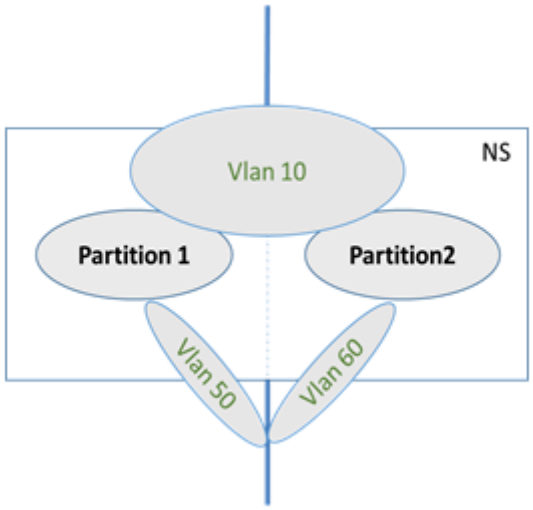
1. [設定] > [システム] > [ネットワーク] > [VLAN*] に移動し、[追加] をクリックして VLAN を作成します。
2. [**VLAN** の作成] ページで、次のパラメータを設定します。
   - VLAN ID
   - エイリアス名
   - 最大伝送ユニット
   - 動的ルーティング
   - IPv6 動的ルーティング
   - パーティションの共有
3. [**Interface Bindings**] セクションで、1 つ以上のインターフェイスを選択し、VLAN にバインドします。
4. [**IP** バインディング] セクションで、1 つ以上の IP アドレスを選択し、VLAN にバインドします。
5. [**OK**] をクリックし、[完了] をクリックします。

共有 **VLAN**

共有 VLAN 設定では、各パーティションに MAC アドレスが割り当てられ、共有 VLAN で受信されるトラフィックは MAC アドレスによって分類されます。サブネットトラフィックを制限できるため、レイヤ 3 VLAN だけをお勧めします。パーティション MAC アドレスは、共有 VLAN 配置にのみ適用され、重要です。

> 注
>
> Citrix ADC バージョン 12.1 ビルド 51.16 以降、パーティション化されたアプライアンス内の共有 VLAN は動的ルーティングプロトコルをサポートします。

次の図は、VLAN（VLAN 10）が 2 つのパーティション間で共有される方法を示しています。



共有 VLAN 構成を展開するには、次の手順を実行します。

1. 共有オプションが「enabled」で VLAN を作成するか、既存の VLAN で共有オプションを有効にします。デフォルトでは、このオプションは「無効」になっています。
2. パーティションインターフェイスを共有 VLAN にバインドします。
3. パーティションを作成します。各パーティションには独自のパーティション MAC アドレスが割り当てられます。
4. パーティションを共有 VLAN にバインドします。

CLI を使用して共有 **VLAN** を構成します

コマンドプロンプトで、次のいずれかのコマンドを入力して、VLAN を追加するか、既存の VLAN の共有パラメータを設定します。

```
add vlan <id> [-sharing (ENABLED | DISABLED)]

set vlan <id> [-sharing (ENABLED | DISABLED)]
```



```
4
5 add vlan 100  - sharing ENABLED
6
7 set vlan 100  - sharing ENABLED
```

CLI を使用してパーティションを共有 **VLAN** にバインドします

コマンドプロンプトで入力します。

```
1 bind partition <partition-id> -vlan <id>
2
3 bind partition P1  - vlan 100
4
5 add ns partition P1  - maxBandwidth 200  - maxconn 50    - maxmemlimit 90
     -partitionMAC<mac_addr
6
7 Done
```

CLI を使用してパーティションの **MAC** アドレスを構成する

```
1 set ns partition <partition name> [-partitionMAC<mac_addr>]
2
3 set ns partition P1  - partitionMAC 22:33:44:55:66:77
```

CLI を使用してパーティションを共有 **VLAN** にバインドします

```
1 bind partition <partition-id> -vlan <id>
2
3 bind partition <partition-id> -vlan <id>
4
5 bind partition P1  - vlan 100
6
7 bind partition P2  - vlan 100
8
9 bind partition P3  - vlan 100
10
11 bind partition P4  - vlan 100
```


Citrix ADC GUI を使用して共有 **VLAN** を構成する

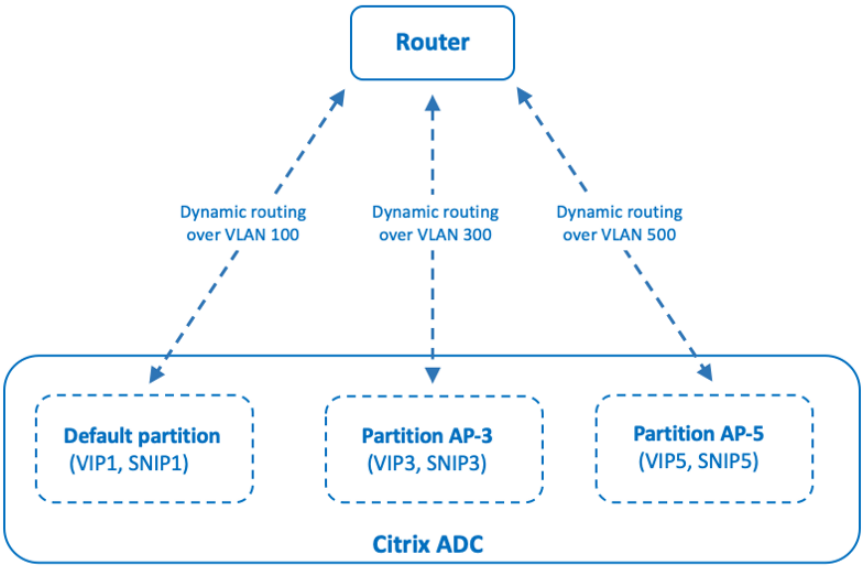
1. **[設定] > [システム] > [ネットワーク] > [VLAN]** に移動し、**VLAN** プロファイルを選択し、[編集] をクリックしてパーティション共有パラメータを設定します。
2. [**Create VLAN**] ページで [**Partitions Sharing**] チェックボックスをオンにします。
3. [**OK**]、[完了] の順にクリックします。

管理パーティション間の共有 **VLAN** を介した動的ルーティング

Citrix ADC アプライアンスの管理パーティションは、複数のテナントをホストする方法を提供します。

Citrix ADC バージョン 12.1 ビルド 51.16 以降、パーティション化されたアプライアンスの共有 VLAN は動的ルーティングプロトコルをサポートします。ルーティングは、管理パーティションに関連付けられた専用 VLAN または共有 VLAN で構成できます。

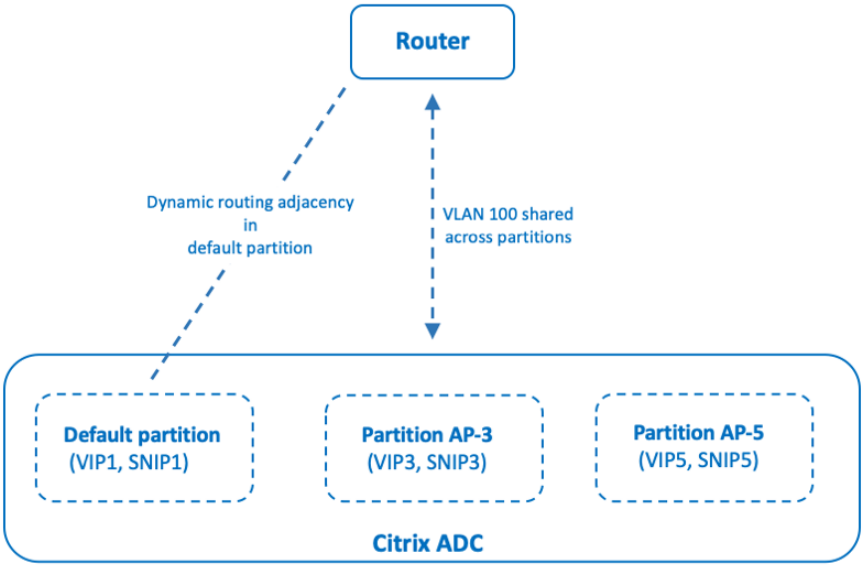
管理パーティションの専用 **VLAN**。専用 VLAN では、テナントのデータパスは 1 つ以上の VLAN を使用して識別されます。その結果、テナントの構成とデータパスの分離が厳密になります。VIP アドレスの状態をアドバタイズするために、動的ルーティングが各パーティションで有効になり、ルーティングの隣接関係がパーティションごとに確立されます。



Dynamic routing over a dedicated VLAN per partition

管理パーティション間で共有 **VLAN**。共有 VLAN では、デフォルト以外のパーティションで設定された VIP アドレスは、デフォルトのパーティションで形成された単一の隣接またはピアリングを介してアドバタイズできます。デフォルト以外のパーティションの SNIP アドレスは、デフォルト以外のパーティションのすべての VIP アドレス（**advertiseOnDefaultPartition** オプションで構成されている）のネクストホップとして使用されます。設定された SNIP アドレスは、ルーティングアドバタイズメントでネクストホップ IP アドレスとしてマークされます。

Citrix ADC アプライアンスの管理パーティションのセットアップ例を考えてみましょう。VLAN100 はデフォルトのパーティション間で共有され、デフォルト以外のパーティションは AP-3 と AP-5 です。SNIP アドレス SNIP1 はデフォルトのパーティションに追加され、SNIP3 は AP-3 に追加され、SNIP5 は AP-5 に追加されます。SNIP1、SNIP3、および SNIP5 は、VLAN-100 を介して到達可能です。VIP アドレス VIP1 はデフォルトのパーティションに追加され、VIP3 は AP-3 に追加され、VIP5 は AP-5 に追加されます。VIP3 と VIP5 は、デフォルトのパーティションで形成された単一の隣接またはピアリングを介してアドバタイズされます。



Dynamic routing over a shared VLAN across partitions

はじめに

デフォルト以外の管理パーティションで共有 VLAN を介して動的ルーティングを構成する前に、次のことを確認してください。

- 動的ルーティングは、デフォルトパーティションの共有 **VLAN** で構成されます。デフォルトパーティションの共有 VLAN での動的ルーティングの構成は、次の手順で構成されます。
  1. 共有 VLAN で動的ルーティングを有効にします。
  2. 動的ルーティングを有効にして SNIP IP アドレスを追加します。この SNIP IP アドレスは、アップストリームとの動的ルーティングに使用されます。
  3. SNIP IP サブネットを共有 VLAN にバインドします。
- **1** つ以上の動的ルーティングプロトコルがデフォルトのパーティションに設定されています。詳細については、ダイナミックルーティングプロトコルの設定を参照してください。

構成の手順

デフォルト以外の管理パーティションの共有 VLAN を介した動的ルーティングの構成は、次の手順で構成されます。

1. デフォルト以外のパーティションに **SNIP IP** アドレスを追加します。この SNIP IP アドレスは、デフォルトパーティションの動的ルーティングに使用されている SNIP IP アドレスと同じサブネット内にある必要があります。
2. 動的ルーティングを使用して、デフォルト以外のパーティションで **VIP** アドレスをアドバタイズするための次のパラメータを設定または有効にします。
   - ホストルートゲートウェイ（hostRtGw）。このパラメーターを、前の手順で追加した SNIP アドレスに設定します。
   - デフォルトのパーティション（advertiseOnDefaultPartition）にアドバタイズします。このパラメーターを有効にします。

構成例

Citrix ADC アプライアンスでの管理パーティション設定の例を考えてみましょう。このアプライアンスには、デフォルト以外の管理パーティション AP-3 が構成されています。共有 VLANVLAN100 は AP-3 にバインドされています。次の設定例は、AP-3 で VLAN100 を介した動的ルーティングを設定します。

手順	構成例
デフォルトの管理パーティション	-
共有 VLAN100 で動的ルーティングを有効にします。	`set vlan 100 -dynamicRouting enabled`
動的ルーティングを有効にして SNIP IP アドレス 192.0.2.10 を追加します。この SNIP IP アドレスは、アップストリームとの動的ルーティングに使用されます。	`add ns ip 192.0.2.10 255.255.255.0 -type SNIP -dynamicRouting enabled`
192.0.2.10 のサブネットを共有 VLAN100 にバインドします。	`bind vlan 100 -IPAddress 192.0.2.10 255.255.255.0`
デフォルト以外の管理パーティション **AP-3**	-
SNIP IP アドレス 192.0.2.30 を追加します。この SNIP IP アドレスは、デフォルトパーティションの SNIP IP アドレス 192.0.2.10 と同じサブネットにあります。	`add ns ip 192.0.2.30 255.255.255.0 -type SNIP`
ダイナミックルーティングを使用して VIP アドレス 203.0.113.300 をアドバタイズする場合は、advertiseOnDefaultPartitionパラメータを有効にし、hostRtGwパラメータを 192.0.2.30 に設定します。	`set ns ip 203.0.113.300 255.255.255.255 -hostRoute enabled -advertiseOnDefaultPartition enabled -hostRtGw 192.0.2.30`

管理パーティション全体の共有 **VLAN** を介した **IPv6** のダイナミックルーティング

IPv6 アドレスが管理パーティション内の共有 VLAN を介して動的にルーティングするには、`enable ns feature IPv6PT`コマンドおよび`set L3Param -ipv6DynamicRouting ENABLED`コマンドを有効にする必要があります。次の設定例は、共有 VLAN 経由の IPv6 のダイナミックルーティングの設定に役立ちます。

構成例

次の設定例では、AP-3 で VLAN 100 を介したダイナミックルーティングを設定します。

手順	構成例
デフォルトの管理パーティション	-
共有 VLAN100 で動的ルーティングを有効にします。	`set vlan 100 -dynamicRouting enabled`
ダイナミックルーティングを有効にした状態で SNIP IP アドレス 2001: b: c: d⊠ 1/64 を追加します。SNIP IP アドレスは、アップストリームとのダイナミックルーティングに使用されます。	`add ns ip6 2001:b:c:d::1/64 -type SNIP -dynamicRouting enabled`
2001: b: c: d⊠ 1/64 のサブネットを共有 VLAN 100 にバインドします。	`bind vlan 100 -IPAddress 2001:b:c:d::1/64`
デフォルト以外の管理パーティション **AP-3**	-
SNIP IP アドレス 2001: b: c: d⊠ 2/64 を追加します。この SNIP アドレスは、デフォルトパーティションの SNIP アドレス 2001: b: c: d⊠ 2/64 と同じサブネット内にあります。	`add ns ip6 2001:b:c:d::2/64 -type SNIP`
ダイナミックルーティングを使用して VIP アドレス 2002⊠ 1/128 をアドバタイズする場合は、`advertiseOnDefaultPartition` パラメータを有効にし、`ip6hostRtGw`パラメータを 2001: b: c: d⊠ 2 に設定します。	`set ns ip6 2002::1/128 -hostRoute enabled -advertiseOnDefaultPartition enabled -ip6hostRtGw 2001:b:c:d::2`

管理パーティションに存在する VIP は、デフォルトパーティションの VTYSH でカーネルルートとして認識される必要があります。

```
> switch partition default
Done

```



```
>vtysh
ns#

ns# sh ipv6 route kernel

IPv6 routing table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
   IA - OSPF inter area, E1 - OSPF external type 1,
   E2 - OSPF external type 2, I - IS-IS, B - BGP
Timers: Uptime

K      2002::1/128 via 2001:b:c:d::2, vlan0, 01:24:15
                                  >> on Default Partition, VIP : 2002::1
   present in AP known via SNIP6 : 2001:b:c:d::2 is present in AP as a
   Kernel Route
```

デフォルトパーティションで OSPFv3/bgp+ の下の「カーネルの再配布」オプションを使用して、アップストリームにアドバタイズできます。

```
ns# sh run router ipv6 ospf
!
router ipv6 ospf 1
redistribute kernel
!
```

Citrix ADC SDX アプライアンスの管理パーティションを持つ共有 VLAN

SDX アプライアンスでは、共有 VLAN で管理パーティションを使用する前に、管理サービスのユーザーインターフェイスを使用して PMAC アドレスを生成および構成する必要があります。管理サービスを使用すると、次の方法でパーティションの MAC アドレスを生成できます。

- ベース MAC アドレスの使用
- カスタム MAC アドレスの指定
- ランダムに MAC アドレスを生成する

> 注
>
> - ランダムに生成される MAC アドレスは、高可用性以外の展開で使用されます。
> - パーティションの MAC アドレスを生成したら、管理パーティションを構成する前に Citrix ADC インスタンスを再起動する必要があります。SDX アプライアンスからのパーティション MAC アドレスの生成の詳細については、SDX アプライアンス内の Citrix ADC インスタンスで管理パーティションを構成するためのパーティション MAC アドレスの生成を参照してください。

管理者パーティションに対する **VXLAN** のサポート

October 7, 2021

パーティション分割された Citrix ADC アプライアンスでは、VLAN の構成と同様に、デフォルトのパーティションで VXLAN を構成できます。VXLAN を構成した後、それを管理パーティションにバインドできます。または、VXLAN がパーティションにバインドされている VLAN を拡張している場合、アプライアンスは VXLAN を同じブロードキャストドメインの下のパーティションにバインドします。これは、VXLAN をパーティションからアンバインドする VLAN のバインド解除に適用できます。

Citrix ADC アプライアンスでの VXLAN の動作の詳細については、「VXLAN」を参照してください。

また、パーティション化された Citrix ADC アプライアンスでの VLAN の動作の詳細については、「管理パーティション化」を参照してください。

VXLAN を設定する前に覚えておくべきポイント

パーティション分割された Citrix ADC アプライアンスで VXLAN を構成する前に、次の点に注意してください。

- VXLAN 経由で VLAN を拡張する場合は、VLAN がパーティションにバインドされていることを確認してください。
- パーティション管理者だけが、管理パーティション内の VXAN の IP および動的ルーティングを構成する必要があります。

共有 VXLAN は、パーティション化されたアプライアンスではサポートされていないため、VXLAN に共有 VLAN にタグを付けることはできません。また、VXLAN にタグを付けると、共有の VLAN にすることはできません。

サポート可能な **VXLAN** 構成

サポート可能な VXLAN 設定を次に示します。

同じブロードキャストドメイン内の **VXLAN** を介した **VLAN** の拡張

次の CLI 手順は、同じブロードキャストドメイン内で VXLAN を介して VLAN を拡張するのに役立ちます。

1. デフォルトパーティションに VLAN を追加する

```
1  add vlan <id>
```

2. 同じブロードキャストドメイン内の VXLAN 上で VLAN を拡張します。


```
add vxlan <vxlan id>  - vlan <id>
```

3. すべての BUM（ブロードキャスト不明マルチキャスト）トラフィックを伝送するようにピア vtep を構成します。

> 注
>
> vtep アドレスはマルチキャストアドレスにすることができます。

```
add bridgetable -mac <mac_addr> -vxlan <positive_integer> -vtep <
    ip_addr> [-vni <positive_integer>][-deviceVlan <
    positive_integer>]
```

4. IP アドレスを VXLAN にバインドします。

```
bind vxlan <id> [-srcIP <ip_addr>][-IPAddress <ip_addr|ipv6_addr
    |*> [<netmask>]]
```

5. VLAN を管理パーティションにバインドします。

```
bind partition <partition-id> -vxlan <id>

add vlan 3000

add vxlan 3000  - vlan 10

add bridgetable  - mac 00:00:00:00:00:00  - vxlan 3000 -vtep
    10.102.58.8  - vni 11

bind vxlan 3000  -  srcIP 10.102.101.15

bind partition p1  - vlan 10
```

管理パーティションに対する **SNMP** のサポート

October 7, 2021

パーティション化された Citrix ADC アプライアンスは、SNMP インフラストラクチャを使用して、パーティションレートの制限とパーティションリソース使用率の詳細の監視を行います。

管理パーティションのレート制限用の **SNMP** トラップ

パーティション分割された Citrix ADC アプライアンスでは、PARTITION-RATE-LIMIT アラームが 9 個の SNMP トラップを生成して、パーティションリソース（帯域幅、接続、メモリなど）が制限に達したか、正常に戻ったことを通知できます。

次の場合、次の 9 つの SNMP トラップが生成されます。

- **partitionCONNThresholdReached**。パーティションのアクティブな接続数が、上限しきい値のパーセンテージを超えています。
- **partitionCONNThresholdNormal**。アクティブな接続の数が、通常のしきい値のパーセンテージ以下です。
- **partitionBWThresholdReached**。パーティションの帯域幅の使用率が、しきい値の高い割合に達した。
- **partitionMEMThresholdReached**。パーティションの現在のメモリ使用量が上限しきい値の割合を超えています。
- **partitionMEMThresholdNormal**。パーティションの現在のメモリ使用量は、通常のしきい値のパーセンテージ以下になります。
- **partitionMEMLimitExceeded**。パーティションの現在のメモリ使用量がメモリ制限のパーセンテージを超えています。
- **partitionCONNLimitExceeded**。パーティションのアクティブな接続数が、設定された制限を超え、新しい接続がドロップされます。
- **partitionCONNLimitNormal**。パーティションのアクティブな接続数が、設定された制限値を下回り、パーティションは新しい接続を受け入れることができます。
- **partitionBWLimitExceeded**。パーティションの現在の帯域幅使用率が、設定されている制限を超えています。

SNMP トラップのしきい値は構成不可能であり、次のとおりです。

- 高しきい値 = 80%（すべてのパーティションレート制限トラップに適用）
- 低しきい値 = 60%（すべてのパーティションレート制限トラップに適用可能）
- メモリ制限 = 95% (パーティションのメモリトラップにのみ適用)

PARTITION-RATE-LIMIT アラームの設定

特定のパーティションで PARTITION-RATE-LIMIT アラームを設定し、SNMP トラップメッセージの生成を有効にします。

1. PARTITION-RATE-LIMIT アラームの有効化
2. PARTITION-RATE-LIMIT アラームの設定
3. SNMP トラップの送信先の設定

CLI を使用して **PARTITION-RATE-LIMIT** アラームを有効にするには

コマンドプロンプトで、次のコマンドを入力します。


```
enable snmp alarm PARTITION-RATE-LIMIT

show snmp alarm PARTITION-RATE-LIMIT
```

CLI を使用して **PARTITION-RATE-LIMIT** アラームを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set snmp alarm PARTITION-RATE-LIMIT [-state ( ENABLED | DISABLED )] [-
    severity <severity>] [-logging ( ENABLED | DISABLED )]
```

CLI を使用して **SNMP** トラップの宛先を構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add snmp trap <trapClass> <trapDestination> [-version <version>] [-td <
    positive_integer>] [-destPort <port>] [-communityName <string>] [-
    srcIP <ip_addr|ipv6_addr>] [-severity <severity>] [-allPartitions (
    ENABLED | DISABLED )]
```

GUI を使用して **PARTITION-Rate-Limit** アラームを設定するには

[システム**] > [SNMP] > [**アラーム] に移動し、[**PARTITION-RATE-LIMIT** アラーム] を選択して、アラームパラメータを設定します。

GUI を使用して **SNMP** トラップの宛先を構成するには

[システム**] > [SNMP] > [**トラップ] に移動し、宛先デバイスの IP アドレスを指定します。

パーティションリソース使用率の **SNMP** モニタリング

SNMP を使用すると、Citrix ADC アプライアンス上でパーティションのリソース（帯域幅、接続、メモリなど）使用率の詳細をリアルタイムで監視できます。これは、SNMP マネージャーから SNMP 要求（SNMP GET、SNMP GET BULK、SNMP GETNEXT、または SNMP WALK など）を送信することによって行われます。

> 注
>
> パーティションリソースを監視するには、デフォルトのパーティションで SNMP コミュニティを設定する必要があります。このスライドでは、*partitionTable* はデフォルトのパーティションに維持され、SNMP 通信はアプライアンスの NSIP アドレスを介して行われます。

Citrix ADC 管理者がアプライアンスのパーティション P1 の帯域幅の使用状況を知りたい場合を考えてみましょう。SNMP マネージャは、対応する OID（パーティション現在の帯域幅）に対する SNMP GET 要求をアプライアンスの NSIP アドレスに送信することによって、この情報を取得します。デフォルトパーティション上の SNMP エージェントは、P1 の現在の帯域幅使用状況を取得し、NSIP アドレスを通じて SNMP Manager に送信します。

次の表に、パーティションテーブルの一部である SNMP カウンタとその説明を示します。

SNMP Parameter	SNMP OID	説明
partitionName	1.3.6.1.4.1.5951.4.1.1.88.1.1	パーティション名
partitionCurrentBandwidth	1.3.6.1.4.1.5951.4.1.1.88.1.2	パーティションの現在の帯域幅使用状況。
partitionCurrentConnections	1.3.6.1.4.1.5951.4.1.1.88.1.3	パーティションのアクティブな接続の現在の数。
partitionMemoryUsagePcnt	1.3.6.1.4.1.5951.4.1.1.88.1.4	パーティションの現在のメモリ使用量 (パーセント)。

管理パーティションの監査ログのサポート

October 7, 2021

パーティション分割された Citrix ADC アプライアンスでは、高度なポリシーを使用して、管理パーティションで監査ログを構成できます。たとえば、特定のパーティションのログ（状態とステータス情報）を表示したい場合があります。複数のユーザーが、パーティション内の承認レベルに基づいてさまざまな機能セットにアクセスします。

確認事項

1. パーティションから生成された監査ログは、単一のログファイルとして保存されます (/var/log/ns.log).
2. 監査ログサーバー（syslog または ns log）のサブネットアドレスを、監査ログメッセージを送信するためのパーティションの送信元 IP アドレスとして構成します。
3. デフォルトパーティションでは、監査ログメッセージの送信元 IP アドレスとして NSIP が使用されます。
4. 監査ログメッセージを表示するには、「show audit messages」コマンドを使用します。

監査ログの構成の詳細については、「監査ログ用の NetScaler アプライアンスの構成」を参照してください。

パーティション分割された **Citrix ADC** アプライアンスでの監査ログの設定

管理パーティションで監査ログを構成するには、次のタスクを実行します。

1. パーティションサブネットの IP アドレスを設定します。管理パーティションの IPv4 SNIP アドレス。
2. 監査ログ（syslog および ns ログ）アクションを構成します。監査アクションは、ログに記録するメッセージと、外部ログサーバにメッセージを記録する方法を指定する情報の集合です。
3. 監査ログ（syslog および ns ログ）ポリシーを構成します。監査ログポリシーは、syslog または ns ログサーバーへのソースパーティションのログメッセージを定義します。
4. 監査ログポリシーを sysGlobal エンティティと nsGlobal エンティティにバインドします。監査ログポリシーをシステムグローバルエンティティにバインドします。
5. 監査ログの統計を確認します。監査ログの統計情報を表示し、設定を評価します。

CLI を使用して以下を構成します

1. パーティションのサブネット IP アドレスを作成します

   ```
   add ns ip <ip address> <subnet mask>
   ```

2. Syslog アクションを作成する

   ```
   add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )] [-transport ( TCP |
   UDP )]
   ```

3. ns ログアクションを作成する

   ```
   add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
   <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )]
   ```

4. Syslog 監査ログポリシーを作成します

   ```
   add audit syslogpolicy syslog-pol1 true audit-action1
   ```

5. ns ログ監査ログポリシーを作成します

   ```
   add audit nslogpolicy nslog-pol1 true audit-action1
   ```

6. 監査ログポリシーを syslogGlobal エンティティにバインドします

   ```
   bind audit sysloggloba -policyName <name> -priority <priority_integer>
    -globalBindType SYSTEM_GLOBAL
   ```

7. 監査ログポリシーを nslogGlobal エンティティにバインドします

   ```
   bind audit nsloggloba -policyName <name> -priority <priority_integer>
   -globalBindType SYSTEM_GLOBAL
   ```

8. 監査ログ統計を表示する

   ```
   stat audit -detail
   ```


例

```
add ns ip 10.102.1.1 255.255.255.0
add audit syslogAction syslog_action1 10.102.1.2 -logLevel
    INFORMATIONAL -dateFormat MMDDYYYY -transport UDP
add audit syslogpolicy syslog-pol1 true syslog_action1
bind audit sysloggloba l -policyName syslog-pol1 -priority 1 -
    globalBindType SYSTEM_GLOBAL
```

ログの保存

SYSLOG または NSLOG サーバーがすべてのパーティションからログ情報を収集すると、ログメッセージとして ns.log ファイルに保存されます。ログメッセージには、次の情報が含まれています。

- パーティション名。
- IP アドレス。
- タイムスタンプ。
- メッセージの種類
- 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)
- メッセージ情報。

共有 **VLAN** 設定の設定済みの **PMAC** アドレスを表示します

April 7, 2022

共有 VLAN 設定でパーティション設定を使用するには、パーティション MAC（PMAC）アドレスと呼ばれる仮想 MAC アドレスが必要です。パーティションは、共有 VLAN での通信に PMAC アドレスを使用します。各パーティションに一意の PMAC アドレスが設定され、そのパーティションにバインドされているすべての共有 VLAN で使用されます。非 SDX プラットフォーム（VPX または MPX）プラットフォームの場合、PMAC アドレスはユーザーが指定するか、Citrix ADC アプライアンスによって内部的に生成されます。PMAC アドレスがパーティションに指定されていない場合、パーティションが最初の共有 VLAN にバインドされたときに内部的に生成されます。SDX プラットフォームの場合、PMAC アドレスは常に SVM ツールで設定してから、パーティションに割り当てる必要があります。

設定された PMAC のリストを表示するには、**Show ns PartitionMAC** コマンドを使用します。このコマンドを使用すると、Citrix ADC CLI または GUI を使用して、設定済みの PMAC を検証できます。このコマンドは、すべての PMAC アドレスと対応するパーティション（割り当てられている場合）を表示します。非 SDX プラットフォームの場合、このコマンドはすべての PMAC アドレスとそれに対応するパーティションを表示します。これは、PMAC アドレスが必要に応じてパーティションに割り当てられるためです（パーティションが共有 VLAN にバインドされている場合）。ただし、SDX プラットフォームの場合、リストに未割り当ての PMAC が含まれている可能性があります。

SDX プラットフォーム用の PMAC を生成する方法については、「パーティション MAC アドレスの生成」トピックを参照してください。

Citrix ADCCLI を使用して PMAC を表示する

コマンドプロンプトで、次のコマンドを入力します。

```
show ns partitionMAC
```

```
Partition MAC Partition Name

1) f2:0c:64:da:f6:d7

2) b4:0c:43:da:f6:d2

3) a6:e7:b2:6c:48:e0

Done
```

Citrix ADC GUI を使用して PMAC アドレスを表示する

1. Citrix ADC アプライアンスにサインインし、「構成」>「システム」>「パーティション **MAC**」の順に選択します。
2. [Partition MAC] ページには、PMAC とそのパーティションのリストが表示されます。

AppExpert

October 7, 2021

以下のトピックでは、AppExpert および Citrix ADC アプライアンスのその他の機能に関する概念的なリファレンスおよび構成手順について説明します。

> 注
>
> ポリシー拡張機能の詳細については、「ポリシー拡張」を参照してください。

- アクション分析: 事前定義された条件に基づいて実行時の統計を収集します。ポリシーとともに使用すると、自動的にリアルタイムのトラフィック最適化のためのインフラストラクチャも提供されます。
- AppExpert アプリケーションとテンプレート: アプリケーション、アプリケーションテンプレート、Citrix Gateway アプリケーション、およびエンティティテンプレートを使用して、Citrix® NetScaler® アプライアンスの構成手順を簡素化します。

- AppQoE：アプリケーションレベルのエクスペリエンスの品質（AppQoE）は、Citrix ADC アプライアンスの既存のポリシーベースのセキュリティ機能を新しいキューイングメカニズムである均等化キューイングを利用する単一の統合機能に統合します。
- エンティティテンプレート：エンティティテンプレートを使用して、ポリシーや仮想サーバーなどの個々の Citrix ADC エンティティを設定および構成する方法について説明します。エンティティテンプレートは、オブジェクトの仕様とデフォルトセットを提供します。
- HTTP コールアウト：ポリシー評価中に特定の基準が満たされたときに、Citrix ADC アプライアンスが生成して外部アプリケーションに送信する HTTP リクエスト。
- パターンセット: デフォルトの構文ポリシーの評価中に文字列の一致を許可します。
- ポリシーと式：Citrix ADC アプライアンスが実行する必要のある操作を決定するルール。
- レート制限：Citrix ADC アプライアンス上の特定のネットワークエンティティまたは仮想エンティティの最大負荷を定義します。
- Responder：リクエストの送信元、リクエストの送信元、およびその他の基準に基づいて、セキュリティとシステム管理に影響します。
- 書き換え：Citrix ADC アプライアンスによって処理される要求または応答の情報を書き換えます。
- 文字列マップ：デフォルトのポリシー構文を使用するすべての Citrix ADC 機能でパターンマッチングを実行します。

アクション分析

October 7, 2021

Web サイトやアプリケーションのパフォーマンスは、最も頻繁に要求されるコンテンツの配信をどのように最適化するかにより決まります。キャッシュや圧縮などの方法は、クライアントへのサービス配信の高速化に役立ちますが、最も頻繁に要求されるリソースを特定し、それらのリソースをキャッシュまたは圧縮できるようにする必要があります。Web サイトやアプリケーショントラフィックに関するリアルタイム統計を集計すれば、最も頻繁に使用されるリソースを特定できます。リソースごとのアクセス頻度や消費帯域幅などの統計によって、サーバーパフォーマンスとネットワーク使用率を改善するために、それらのリソースをキャッシュまたは圧縮する必要があるかどうかを判断できます。応答時間やアプリケーションへの同時接続数などの統計は、サーバー側のリソースを強化する必要があるかどうかを判断するのに役立ちます。

Web サイトやアプリケーションが頻繁に更新されない場合、統計データを収集する製品を使用して、その統計を手動で分析し、コンテンツの配信を最適化できます。ただし、手動による最適化を実行したくない場合、またはウェブサイトやアプリケーションが本質的に動的である場合は、統計データを収集するだけでなく、統計に基づいてリソースの配信を自動的に最適化できるインフラストラクチャが必要です。Citrix ADC アプライアンスでは、この機能はアクション分析機能によって提供されます。この機能は単一の Citrix ADC アプライアンス上で動作し、定義した基準に

基づいてランタイム統計を収集します。Citrix ADC ポリシーと共に使用すると、この機能によって自動的なリアルタイムトラフィックの最適化に必要なインフラストラクチャも提供されます。

アクション分析機能を設定するときは、セレクタと呼ばれるエンティティでデフォルトの構文式を設定することで、統計データ（URL や HTTP メソッドなど）を収集するリクエスト属性を指定します。次に、サンプリング間隔やサンプル数などの設定を構成する ID を構成します。また、セレクタと識別子のペアで指定されたとおりにアプライアンスがトラフィックを評価できるようにするポリシーも設定します。最後に、ポリシーをバインドポイントにバインドして、統計情報の収集を開始します。

アプライアンスには、機能の使用を開始するために使用できる組み込みセレクター、識別子、レスポンダーポリシーのセットも用意されています。

アプライアンスは、次の統計情報を集約します。

- リクエストの数。
- 要求によって消費される帯域幅。
- 応答時間。
- 同時接続の数。

選択した属性に対してレコードの実行時ソートを実行するよう、機能を構成できます。統計データは、コマンドラインインターフェイスまたは構成ユーティリティの Stream Sessions ツールのいずれかを使用して表示できます。

セレクタの設定

October 7, 2021

セレクタは、要求を識別するためのフィルタです。これは、リクエスト内のクライアント IP アドレスや URL などのリクエスト属性を識別する、最大 5 つの個別のデフォルト構文式で構成されます。各式は非複合デフォルト構文式であり、他の式との AND 関係にあると見なされます。次に、セレクタ式のいくつかの例を示します。

- `HTTP.REQ.URL`
- `CLIENT.IP.SRC`
- `HTTP.RES.BODY(1000).AFTER_STR("<string>").BEFORE_STR("<string>")"`
- `CLIENT.IP.SRC.SUBNET(24)`

セレクタは、レート制限およびアクション分析の設定で使用されます。セレクタは、レート制限設定ではオプションですが、アクション分析設定では必須です。

パラメータを指定する順序は重要です。たとえば、あるセレクタで IP アドレスとドメインを（その順序で）構成し、別のセレクタでドメインと IP アドレスを指定すると、Citrix ADC ではこれらの値が一意であると見なされます。これにより、同じトランザクションが 2 回カウントされる可能性があります。また、複数のポリシーで同じセレクタを呼び出すと、Citrix ADC は同じトランザクションを複数回カウントできます。

セレクタ内の式を変更した場合、その式を呼び出すポリシーが新しいポリシーラベルまたはバインドポイントにバインドされると、エラーが発生することがあります。たとえば、myLimitSelector1 という名前のセレクタを作

成し、myLimitID1 から呼び出し、dnsRateLimit1 という名前の DNS ポリシーから識別子を呼び出すとします。myLimitSelector1 の式を変更すると、dnsRateLimit1 を新しいバインドポイントにバインドするときにエラーが発生することがあります。回避策は、これらの式を呼び出すポリシーを作成する前に、これらの式を変更することです。

Citrix ADC アプライアンスは、最も一般的なユースケースの一部に組み込みセレクタの PDF を提供します 。PDF を参照してください。

また、選択したリクエスト属性を識別する式でセレクタを設定することもできます。たとえば、特定のヘッダーを持つリクエストのレコードを作成できます。ヘッダーを評価するには、使用するセレクタにHTTP.REQ.HEADER("<header_name>")を追加します。

コマンドラインインターフェイスを使用してセレクタを設定するには:

コマンドプロンプトで次のコマンドを入力して、セレクタを設定し、設定を確認します。

- `add stream selector <name> <rule> ...`
- `show stream selector`

例

```
> add stream selector myselector HTTP.REQ.URL CLIENT.IP.SRC
 Done
> show stream selector myselector
    Name: myselector
    Expressions:
        1) HTTP.REQ.URL
        2) CLIENT.IP.SRC
 Done
>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセレクタを変更または削除するには:

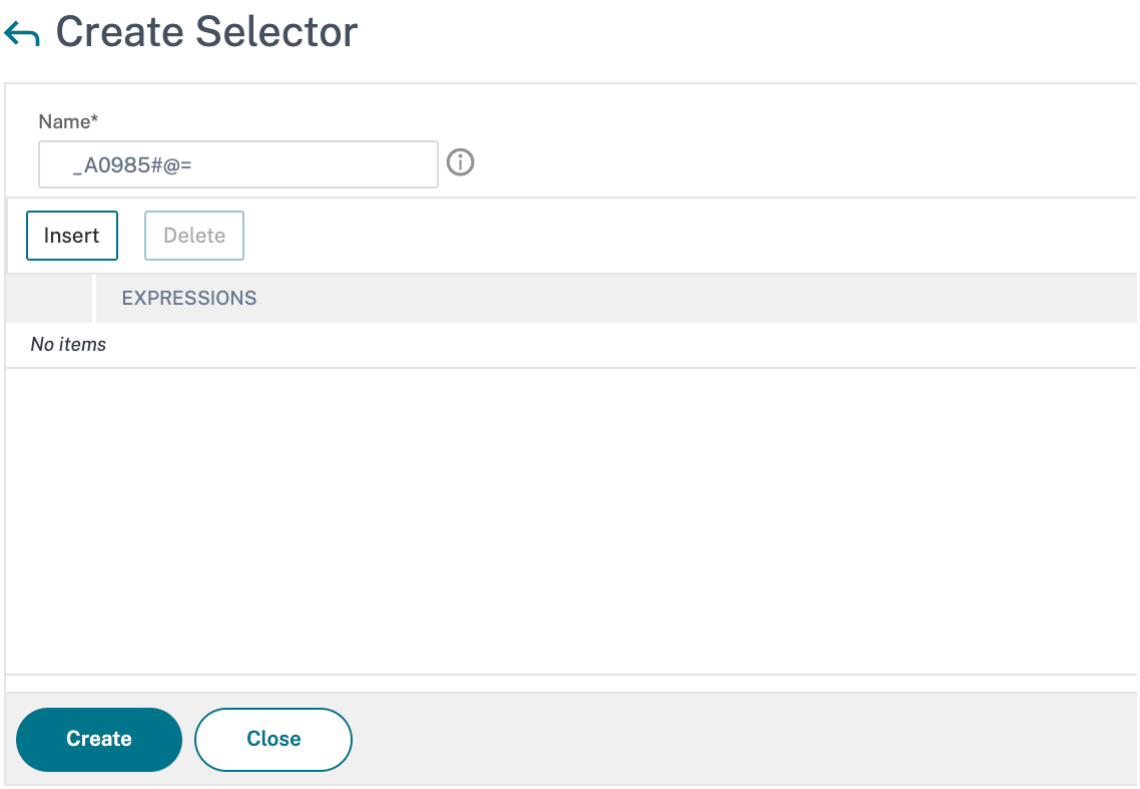
- セレクタを変更するには、set stream selector コマンド、セレクタの名前、および式とともに規則パラメータを入力します。保持する既存の式と、追加する新しい式を入力します。
- セレクタを削除するには、rm stream セレクタコマンドとセレクタの名前を入力します。

GUI を使用してセレクタを設定するには、次の手順を実行します。

1. [**AppExpert**] > [アクション分析] > [セレクタ] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
    - セレクタを作成するには、「追加」をクリックします。
    - セレクタを変更するには、セレクタを選択し、「編集」をクリックします。
3. 「セレクタの作成」または「セレクタの設定」ページで、次のパラメータを設定します。

- Name: セレクタの名前を追加するには、[名前] フィールドに名前を入力します。名前は、ASCII、英数字、またはアンダースコアで始まる必要があります。名前には、ASCII 英数字、アンダースコア、ハッシュ、ピリオド、スペース、コロン、アット文字、等号、ハイフンのみを含める必要があります。
- Expressions. セレクタ設定に式を追加するには、[挿入] をクリックします。セレクター設定から式を削除するには、[式] ボックスで式を選択し、[削除] をクリックします。注記:「式」(Expressions) ボックスに、有効なパラメータを入力します。たとえば、HTTP と入力します。次に、このパラメータの後にピリオドを入力します。ドロップダウンメニューが表示されます。このメニューの内容には、入力した最初のキーワードの後に続くキーワードが表示されます。この式プレフィックスで次のキーワードを選択するには、ドロップダウンメニューで選択項目をダブルクリックします。[式] テキストボックスには、式プレフィックスの最初と 2 番目のキーワード（HTTP.REQ など）が表示されます。完全なエクスプレッションが形成されるまで、エクスプレッションコンポーネントの追加を続けます。

4. [挿入] をクリックします。
5. 非複合式を 5 つまで追加します。
6. [作成]、[閉じる] の順にクリックします。



ストリーム識別子の構成

October 7, 2021

ストリーム識別子を設定して、特定のセレクタによって識別される要求から統計データを収集するためのパラメータを指定します。識別子は、使用するセレクタ、統計収集間隔、サンプル数、およびレコードをソートするフィールドを指定します。

Citrix ADC アプライアンスには、一般的なユースケース用に次の組み込みストリーム識別子が含まれています。すべての組み込み識別子は、サンプルカウントを 1、間隔を 1 分間指定します。さらに、
REQUESTS 属性のデータを並べ替えます。それらは、異なる組み込みセレクタに関連付けられていることだけが異なります。各組み込み識別子は、同じ名前の組み込みセレクタに関連付けられます（たとえば、組み込み識別子 Top_URL は組み込みセレクタの
Top_URL に関連付けられます）。組み込み識別子は次のとおりです。

- Top_URL
- Top_CLIENTS
- Top_URL_CLIENTS_LBVSERVER
- Top_URL_CLIENTS_CSVSERVER
- Top_MSSQL_QUERY_DB_LBVSERVER
- Top_MYSQL_QUERY_DB_LBVSERVER

組み込みセレクタの詳細については、「セレクタの設定」を参照してください。

注：セレクタの文字列結果（HTTP.REQ.URL など）を格納するための最大長は 60 文字です。文字列 (URL など) の長さが 1000 文字で、そのうちの 50 文字で文字列を一意に識別できる場合は、式を使用して必要な 50 文字だけを抽出します。

組み込み識別子の設定は変更できません。ただし、選択した構成で識別子を作成できます。

コマンドラインインターフェイスを使用してストリーム識別子を構成するには

コマンドプロンプトで次のコマンドを入力して、ストリーム ID を構成し、構成を確認します。

- `add stream identifier <name> <selectorName> [-interval <positive_integer>] [-SampleCount <positive_integer>] [-sort <sort>]`
- `show stream identifier <name>`

例

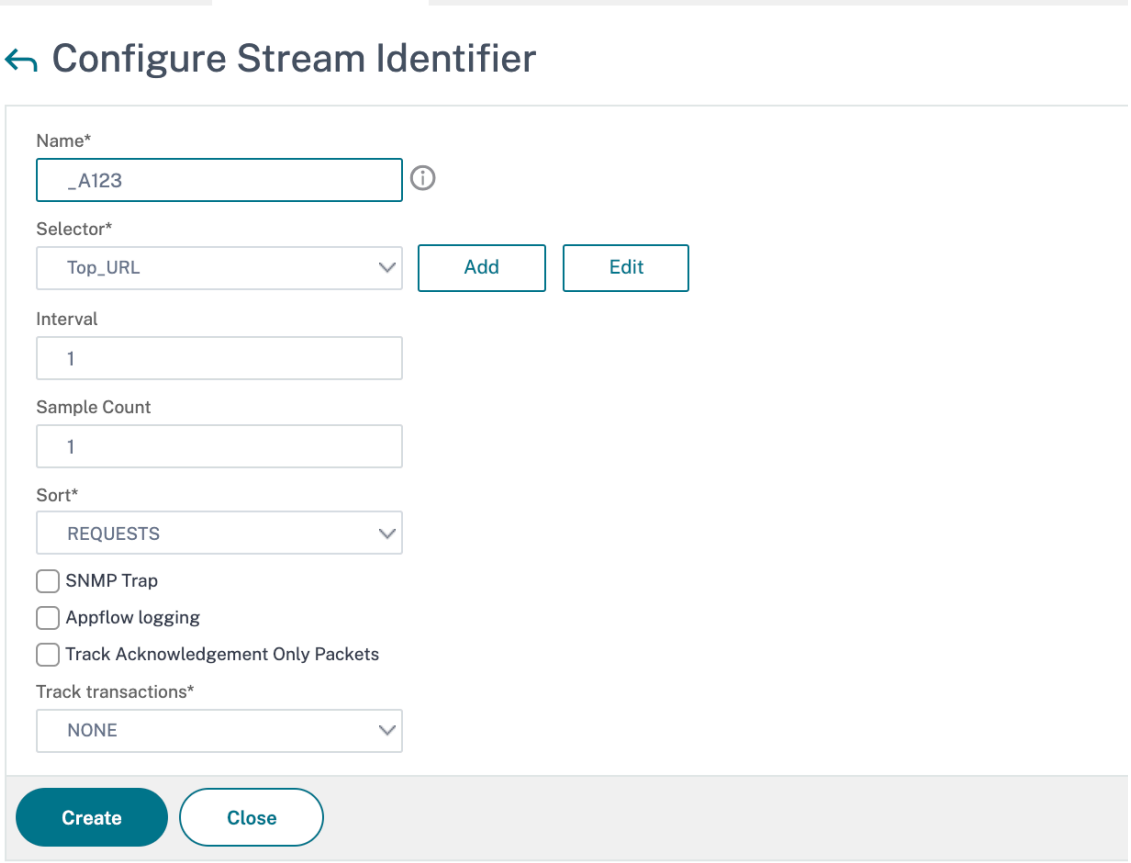
```
1 > add stream identifier myidentifier Top_URL -interval 10 -sampleCount
     100
2  Done
3 <!--NeedCopy-->
```

GUI を使用してストリーム識別子を構成するには

1. [**AppExpert**] > [アクション分析] > [ストリーム識別子] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。

- ストリーム識別子を作成するには、[**Add**] をクリックします。
- ストリーム識別子を変更するには、識別子を選択し、[編集] をクリックします。

3. ストリーム識別子の構成ページで、次のパラメータを設定します。
   - 名前
   - セレクタ
   - Interval
   - サンプル数
   - 並べ替え
4. [**Create**] をクリックしてから、[**Close**] をクリックします。



統計の表示

October 7, 2021

収集された統計情報は、コマンドラインインターフェイスでは表形式で表示でき、設定ユーティリティではグラフィカル形式で表示できます。

次の表に、収集された統計情報を示します。

統計	stat ストリーム識別子<identifier name> コマンドの出力内の列名	説明
要求数	**Req**	直前の<interval>分間にレコードが作成されたリクエストの数。
消費帯域幅	**BandW**	直近の<interval>分数内に受信された要求によって消費された合計帯域幅。要求の合計帯域幅は、要求とその応答によって消費される帯域幅です。値は、次の上位または下位の整数値に四捨五入されます。したがって、期待値と若干異なる可能性があります。たとえば、リクエストの合計帯域幅消費が 2.2 KB の場合。リクエストの 1 つのインスタンスが 2 KB を消費したとして表示されることがあります。2 つのインスタンスは 4 KB を消費したとして表示されますが、3 つのインスタンスは 7 KB を消費したと見なされる場合があります。
Response time	**RspTime**	直近<interval>分に受信したすべてのリクエストの平均応答時間（分）。
同時接続	**Conn**	現在開いている同時接続の合計数。

コマンドラインを使用してストリーム識別子について収集された統計データを表示するには

コマンドプロンプトで入力します。

```
stat stream identifier <name> [<pattern> ...] [-detail] [-fullValues] [-ntimes <positive_integer>] [-logFile <input_filename>] [-sortBy <sortBy> [<sortOrder>]
```


例

例 1 は、BandW 列の出力を降順でソートします。例 2 では、例 1 の [**Req**] 列の出力を昇順でソートします。

例 1

```
> stat stream identifier myidentifier -sortBy BandW Descending -
   fullValues
Stream Session statistics
                    Req                 BandW
User1                508                125924
User2               5020                 12692
User3               2025                  4316

                RspTime                  Conn
User1              5694                     0
User2               109                     0
User3                 3                     0
 Done
<!--NeedCopy-->
```

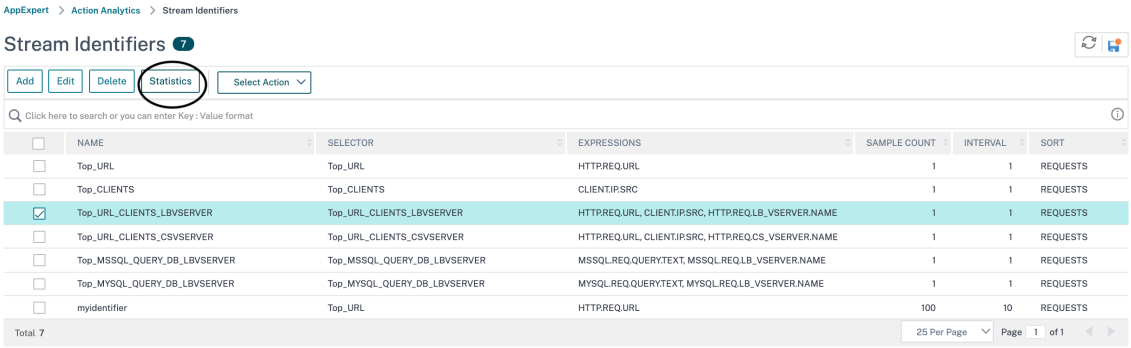
例 2

```
> stat stream identifier myidentifier -sortBy Req Ascending  -
   fullValues
Stream Session statistics
                    Req                 BandW
User1                508                125924
User3               2025                  4316
User2               5020                 12692

                RspTime                  Conn
User1              5694                     0
User3                 3                     0
User2               109                     0
 Done
<!--NeedCopy-->
```

GUI を使用してストリーム識別子について収集された統計データを表示するには

1. [**AppExpert**] > [アクション分析] > [ストリーム識別子] に移動します。
2. セッションを表示するストリーム識別子を選択し、[Statistics] をクリックします。さまざまなセレクタ式について収集された値に基づいて出力をグループ化する方法については、「Statistics」を参照してください。



属性値に対するレコードのグループ化

October 7, 2021

特定の URL が全体およびクライアントごとにアクセスされた回数、クライアントごとの GET リクエストと POST リクエストの合計数などの統計情報は、需要を満たすためにリソースを拡張する必要があるのか、配信用に最適化する必要があるのかについての貴重な洞察を提供します。このような統計情報を取得するには、適切なセレクタ式のセットを使用し、stat ストリーム識別子コマンドで pattern パラメータを使用する必要があります。グループ化は、コマンドで指定されたパターンに基づいています。グループ化は、複数の式の値に対して同時に実行できます。

コマンドラインインターフェイスでは、任意のパターンを使用して出力をグループ化できます。設定ユーティリティのパターンは、さまざまなセレクタ式の値にドリルダウンするときの選択肢によって異なります。たとえば、式HTTP.REQ.URL、CLIENT.IP.SRC、HTTP.REQ.LB_VSERVER.NAMEがその順序にあるセレクタを考えてみましょう。統計のホームページには、これらの各式のアイコンが表示されます。CLIENT.IP.SRCのアイコンをクリックすると、出力はパターンに基づいていますか? で指定します。出力には、各クライアント IP アドレスの統計情報が表示されます。IP アドレスをクリックすると、* <IP address> ? と? <IP address> *のパターンに基づいて出力されます。ここで<IP address> は選択した IP アドレスです。結果の出力では、URL をクリックすると、使用されるパターンは<URL> <IP address> ?になります。

コマンドラインインターフェイスを使用してセレクタ式の値のレコードをグループ化するには

コマンドプロンプトで次のコマンドを入力して、セレクタ式に基づいてレコードをグループ化します。

```
stat stream identifier <name> [<pattern> ...]
```

次の例では、別のパターンを使用して、stat stream identifier コマンドの出力に対するパターンの効果を示します。セレクタ式は、その順序で、HTTP.REQ.URL と HTTP.REQ. ヘッダー（「ユーザーヘッダー」）です。要求には、UserHeader という名前のカスタムヘッダーが含まれています。この例では、特定の統計値はグループ化によって決定されるとおりに変化しますが、特定のフィールドの値の合計は変わりません。

例 1

次のコマンドでは、使用されるパターンは? ?. アプライアンスは、両方のセレクタ式で収集された値に基づいて出力をグループ化します。行ヘッダーは、疑問符 (?) で区切られた式の値で構成されます。ヘッダー/ mysite/mypage1.html を持つ行ですか? Ed は、URL/mysite/mypage1.html のユーザー Ed によって行われたリクエストの統計を表示します。

> 注:
>
> 次のコマンドを「?」ではなく「\?」で入力する必要があります。たとえば、セレクターでは、client.ip.src と client.tcp.srcport という式が使用されています。セレクタのために収集された値の出力をグループ化する STAT コマンドは、'STAT ストリーム識別子 myidentifier ですか??-以下に示すようにフルバリュス。

```
> stat stream identifier myidentifier ? ? -fullValues
Stream Session statistics
                                                Req                  BandW
/mysite/mypage2.html?Grace                        1                   2553
/mysite/mypage1.html?Grace                        2                      4
/mysite/mypage1.html?Ed                           8                     16
/mysite/mypage2.html?Joe                          1                   2554
/mysite/mypage1.html?Joe                          5                     10
/mysite/?Joe                                      1                      4

                                             RspTime                  Conn
/mysite/mypage2.html?Grace                        0                      0
/mysite/mypage1.html?Grace                        0                      0
/mysite/mypage1.html?Ed                           0                      0
/mysite/mypage2.html?Joe                          0                      0
/mysite/mypage1.html?Joe                          0                      0
/mysite/?Joe                                      6                      0
 Done
<!--NeedCopy-->
```

例 2

次のコマンドでは、使用されるパターンは *? です。アプライアンスは、2 番目の式 HTTP.REQ.HEADER（「ユーザーヘッダー」）に対して累積された値に基づいて出力をグループ化します。行には、ユーザー Grace、Ed、Joe によって行われたすべてのリクエストの統計が表示されます。

> 注:
>
> 次のコマンドを「?」の代わりに「?」。

```
> stat stream identifier myidentifier * ?
Stream Session statistics
                  Req     BandW  RspTime      Conn
```



```
Grace                3      2557         0         0
Ed                   8        16         0         0
Joe                  7      2568         6         0
 Done
<!--NeedCopy-->
```

例 3

次のコマンドでは、使用されるパターンは? *、これはデフォルトのパターンです。出力は、最初のセレクタ式で収集された値に基づいてグループ化されます。各行には、1 つの URL の統計情報が表示されます。

> 注:
>
> 次のコマンドを「?」の代わりに「?」。

```
> stat stream identifier myidentifier ? * -fullValues
Stream Session statistics
                                          Req                  BandW
/mysite/mypage2.html                        2                   5107
/mysite/mypage1.html                       15                     30
/mysite/                                    1                      4

                                       RspTime                  Conn
/mysite/mypage2.html                          0                      0
/mysite/mypage1.html                          0                      0
/mysite/                                      6                      0
 Done
<!--NeedCopy-->
```

例 4

次のコマンドでは、使用されるパターンは* *です。アプライアンスは、受信したすべてのリクエストについて、行タイトルなしで 1 セットの集合統計情報を表示します。

```
> stat stream identifier myidentifier * *
Stream Session statistics
                  Req     BandW  RspTime      Conn
                   18      5141         6         0
 Done
<!--NeedCopy-->
```

例 5

次のコマンドでは、パターンは /mysite/mypage1.html * です。アプライアンスは、URL /mysite/mypage1.html に対して受信したすべてのリクエストについて、行タイトルなしの 1 セットの集合統計情報を表示します。

```
> stat stream identifier myidentifier /mysite/mypage1.html *
Stream Session statistics
                  Req    BandW  RspTime     Conn
                   15       30        0        0
 Done
<!--NeedCopy-->
```

ストリームセッションのクリア

October 7, 2021

ストリーム識別子に対して蓄積されたすべてのレコードをフラッシュできます。

コマンドラインインターフェイスを使用してストリームセッションをクリアするには

コマンドプロンプトで次のコマンドを入力して、ストリームセッションをクリアし、結果を確認します。

- ストリームセッションのクリア
- stat stream identifier

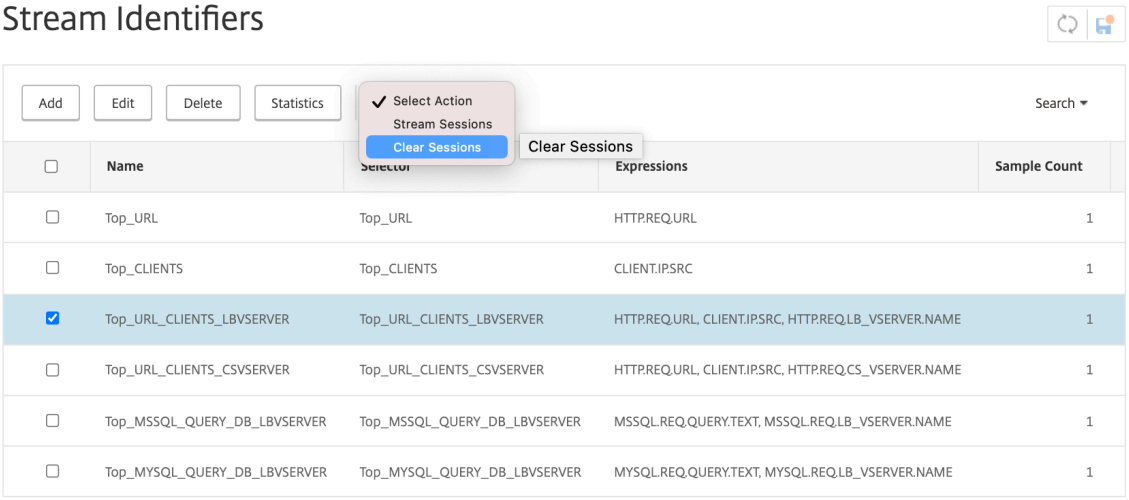
例

この例では、最初に stat stream identifier コマンドを使用し、clear stream session コマンドの結果を確認するために使用される stat stream identifier コマンドと比較できるようにします。

```
>stat stream identifier myidentifier
Stream Session statistics
                  Req    BandW  RspTime     Conn
/aed....html        2        0        0        0
/                 636      303       12        0
 Done
>clear stream session myidentifier
 Done
>stat stream identifier myidentifier
 Done
<!--NeedCopy-->
```


GUI を使用してストリームセッションをクリアするには

1. [**AppExpert**] > [アクション分析] > [ストリーム識別子] に移動します。
2. セッションをクリアするストリーム識別子を選択し、[セッションのクリア] をクリックします。



トラフィックを最適化するためのポリシーの構成

October 7, 2021

アクション分析設定のセレクタと識別子のペアを有効にするには、統計情報を収集するトラフィックフロー内のポイントにペアを関連付ける必要があります。これを行うには、デフォルトの構文ポリシーを設定し、ポリシー規則からストリーム識別子を参照します。圧縮ポリシー、キャッシュポリシー、書き換えポリシー、アプリケーションファイアウォールポリシー、応答側ポリシー、およびブール式に基づくアクションを持つその他のポリシーを使用できます。

アクション分析機能では、データを収集および評価するためのデフォルトの構文式と関数のセットが導入されています。式 `ANALYTICS.STREAM(<identifier_name>)` は、使用する識別子を参照するために使用されます。式 COLLECT_STATS は、統計データを収集するために使用されます。`IS_TOP(<uint>)` や `IS_TOP_FREQUENTS(<uint>)` などの機能は、リアルタイムのトラフィック最適化を自動的に決定するために使用されます。

- **IS_TOP(<number>).** 指定されたオブジェクトが <number> 要素の先頭にあるかどうかを調べます。たとえば、は、上位 10 個の要素の中の要素です。複数の要素が数を持つ場合、それらは本質的に類似しているとみなされます。undef 条件を避けるために、ソート関数をオンにする必要があります。
- **IS_TOP_FREQUENTS(<frequency>).** 指定されたオブジェクトが最上位要素にある <grequency> 要素の先頭にあるかどうかを調べます。たとえば、は、すべてのトップ要素のトップ 50%の間で要素を維持します。同じ値を持つ要素は、本質的に類似しているとみなされます。undef 条件を避けるために、ソート関数をオンにする必要があります。

Citrix ADC アプライアンスがトラフィックからデータを収集するのか、アクションを実行するのかを決定するのは、ポリシー構成です。アプライアンスが統計データのみを収集する必要がある場合は、ルール`ANALYTICS.`

STREAM(<identifier_name>).COLLECT_STATS とアクション NOOP を使用してポリシーを設定できます。NOOP ポリシーは、バインドポイントで最も優先順位の高いポリシーである必要があります。統計情報を収集するだけの場合は、このポリシーで十分です。圧縮やキャッシュの対象など、トラフィックの最適化の決定は、統計データの手動による定期的な評価に基づいている必要があります。

統計情報の収集に加えて、アプライアンスがトラフィックに対するアクションも実行する必要がある場合は、NOOP ポリシーの gotoPriorityExpression パラメータを設定して、目的のルールとアクションを持つ別のポリシーが後で評価されるようにする必要があります。この 2 番目のポリシーには、ANALYTICS.STREAM(<identifier_name>) プレフィックスで始まるルールと、データを評価する関数が必要です。

次に、グローバルに設定およびバインドされた 2 つの応答側ポリシーの例を示します。ポリシー responder_stat_collection を使用すると、アプライアンスは識別子（myidentifier）に基づいて統計情報を収集できます。ポリシー responder_notify は、収集されたデータを評価します。

例

```
> add responder action send_notification respondwith '"You are in the Top 10 list for bandwidth consumption"'
 Done
> add responder policy responder_stat_collection' ANALYTICS.STREAM("myidentifier").COLLECT_STATS' NOOP
Done
> add responder policy responder_notify 'ANALYTICS.STREAM("myidentifier").BANDWIDTH.IS_TOP(10)' send_notification
 Done
> bind responder global responder_stat_collection 10 NEXT
 Done
> bind responder global responder_notify 20 END
 Done
<!--NeedCopy-->
```

ユーザーまたはクライアントデバイスあたりの帯域幅消費を制限する方法

October 7, 2021

Web サイト、アプリケーション、またはファイルホスティングサービスには、すべてのユーザーにサービスを提供するために利用可能な有限なネットワークリソースとサーバーリソースがあります。最も重要なリソースの 1 つは帯域幅です。ユーザベースのサブセットだけが帯域幅を大幅に消費すると、ネットワークの輻輳が発生し、他のユーザに対するリソースのアベイラビリティが低下する可能性があります。ネットワークの輻輳を防ぐために、要求に至るまでの一定の期間にわたって事前設定された帯域幅値を超えた場合、HTML ページを使用してクライアント要求に応答するなど、一時的なサービス拒否技術を使用して、クライアントの帯域幅消費を制限する必要があります。

一般に、帯域幅の消費量は、クライアントデバイスごとまたはユーザーごとに調整できます。このユースケースは、1 時間にわたってクライアントあたりの帯域幅消費を 100 MB に制限する方法を示しています。また、このユースケースは、ユーザー名を提供するカスタムヘッダーを使用して、1 時間にわたって 100 MB にユーザーあたりの帯域幅消費を規制する方法を示しています。どちらの場合も、1 時間の移動期間における帯域幅消費の追跡は、ストリーム識別子の interval パラメータを 60 分に設定することで実現されます。また、このユースケースでは、制限を超えたクライアントに送信する HTML ページをインポートする方法も示します。HTML ページをインポートすると、これらのユースケースでのレスポンダーアクションの設定が簡素化されるだけでなく、同じレスポンダーを必要とするレスポンダーアクションの設定も簡素化されます。

コマンドラインインターフェイスを使用して、ユーザーまたはクライアントデバイスあたりの帯域幅消費を制限するには

コマンドラインインターフェイスで次のタスクを実行して、クライアントまたはユーザーの帯域幅消費を制限するためのアクション分析を構成します。各ステップには、サンプルコマンドとその出力が含まれています。

1. 負荷分散構成を設定します。負荷分散仮想サーバー mysitevip を構成し、必要なすべてのサービスを構成します。サービスを仮想サーバにバインドします。次の例では、10 個のサービスを作成し、そのサービスを mysitevip にバインドします。

```
> add lb vserver mysitevip HTTP 192.0.2.17 80
 Done
> add service service[1-10] 192.0.2.[240-249] HTTP 80
service ”service1” added
service ”service2” added
service ”service3” added
.
.
.
service ”service10” added
 Done
> bind lb vserver vserver1 service[1-10]
service ”service1” bound
service ”service2” bound
service ”service3” bound
.
.
.
service ”service10” bound
 Done
<!--NeedCopy-->
```

2. ストリームセレクタを設定します。次のストリームセレクタのいずれかを設定します。
   - クライアントごとの帯域幅消費を制限するには、クライアント IP アドレスを識別するストリームセレク

タを設定します。

```
> add stream selector myselector CLIENT.IP.SRC
 Done
<!--NeedCopy-->
```

- ユーザー名を提供する要求ヘッダーの値に基づいてユーザーごとの帯域幅消費を制限するには、ヘッダーを識別するストリームセレクタを設定します。次の例では、ヘッダーの名前は UserHeader です。

```
> add stream selector myselector HTTP.REQ.HEADER( “UserHeader ” )
 Done
<!--NeedCopy-->
```

3. ストリーム識別子を設定します。ストリームセレクタを使用するストリーム識別子を設定します。間隔パラメータを 60 分に設定します。

```
> add stream identifier myidentifier myselector -interval 60 -sampleCount 1 -sort BANDWIDTH
 Done
<!--NeedCopy-->
```

4. レスポンダアクションを設定します。帯域幅消費制限を超えたユーザーまたはクライアントに送信する HTML ページをインポートし、レスポンダーのアクションの crossed_limits でページを使用します。

```
> import responder htmlpage http://.1.1.1/stdpages/wait.html crossed-limits.html
This operation may take some time, Please wait...

 Done
> add responder action crossed_limits respondwithhtmlpage crossed-limits.html
 Done
<!--NeedCopy-->
```

5. レスポンダポリシーを設定します。レスポンダーポリシー myrespol1 を ANALYTICS.STREAM (「myidentifier」) .COLLECT_STATS とアクション NOOP を使用して構成します。次に、クライアントまたはユーザーが 100 MB の制限を超えたかどうかを判断するポリシー myrespol2 を構成します。ポリシー myrespol2 は、レスポンダアクション crossed_limits を使用して設定されます。


```
> add responder policy myrespol1 'ANALYTICS.STREAM(”myidentifier”)
   .COLLECT_STATS' NOOP
 Done
> add responder policy myrespol2 'ANALYTICS.STREAM(”myidentifier”)
   .BANDWIDTH.GT(104857600)' crossed_limits
 Done
<!--NeedCopy-->
```

6. レスポンダポリシーをロードバランシング仮想サーバにバインドします。統計データのみを収集するポリシー myrespol1 は、優先順位が高く、GOTO 式を NEXT にする必要があります。

```
> bind lb vserver mysitevip -policyName myrespol1 -priority 1 -
   gotoPriorityExpression NEXT
 Done
> bind lb vserver mysitevip -policyName myrespol2 -priority 2 -
   gotoPriorityExpression END
 Done
<!--NeedCopy-->
```

7. 構成をテストします。複数のクライアントまたはユーザからのテスト HTTP 要求をロードバランシング仮想サーバに送信し、stat stream identifier コマンドを使用して、指定された ID について収集された統計情報を表示することにより、設定をテストします。次の出力は、クライアントの統計を表示します。

```
> stat stream identifier myidentifier -sortBy BandW  - fullValues
Stream Session statistics
                            Req             BandW
192.0.2.30                  5000            3761
192.0.2.31                  29              2602
192.0.2.32                  25              51

                            RspTime         Conn
192.0.2.30                  2               0
192.0.2.31                  0               0
192.0.2.32                  0               0
 Done
>
<!--NeedCopy-->
```


AppExpert アプリケーションおよびテンプレート

October 7, 2021

> 警告
>
> ：アプリケーションテンプレート機能は、Citrix ADC 13.0 ビルド 82.x 以降から廃止され、代替としてスタイルブックを使用することをお勧めします。詳細については、「スタイルブック 」トピックを参照してください。

AppExpert アプリケーションは、Citrix ADC アプライアンスで設定する構成のコレクションです。AppExpert アプリケーションの管理は、GUI（GUI）によって簡素化され、各トラフィックサブセットを処理するためのアプリケーショントラフィックサブセット、および個別のセキュリティおよび最適化ポリシーのセットを指定できます。また、展開手順が 1 つのビューに統合されるため、クライアントのターゲット IP アドレスをすばやく構成し、ホストサーバーを指定できます。

AppExpert アプリケーションを開始するには、まず適切なアプリケーションテンプレートを取得し、そのテンプレートを Citrix ADC アプライアンスにインポートする必要があります。AppExpert アプリケーションがセットアップされたら、アプリケーションが正しく動作していることを確認する必要があります。必要に応じて、要件に合わせて構成をカスタマイズできます。

さまざまなアプリケーションコンポーネント、統計情報、およびアプリケーションビジュアライザーのカウンタを表示して、定期的に構成を検証および監視できます。また、アプリケーションの認証、認可、監査（認証、認可、監査）のポリシーを構成することもできます。

AppExpert アプリケーションの用語

AppExpert アプリケーション機能で使用される用語と、用語が使用されるエンティティの説明を次に示します。

パブリックエンドポイント。Citrix ADC アプライアンスが関連する Web アプリケーションに対するクライアント要求を受信する IP アドレスとポートの組み合わせ。パブリックエンドポイントは、HTTP トラフィックまたはセキュア HTTP (HTTPS) トラフィックのいずれかを受信するように設定できます。Web アプリケーションに対するすべてのクライアント要求は、パブリックエンドポイントに送信する必要があります。AppExpert アプリケーションには、複数のエンドポイントを割り当てることができます。パブリックエンドポイントは、テンプレートをインポートした後に設定します。

アプリケーションユニット。Web アプリケーショントラフィックのサブセットを処理し、関連するコンテンツをホストする一連のサービスを負荷分散する AppExpert アプリケーションエンティティ。アプリケーションユニットが管理する必要のあるトラフィックのサブセットは、ルールによって定義されます。また、各アプリケーションユニットは、管理する要求と応答に対して、独自のトラフィック最適化とセキュリティポリシーのセットを定義します。これらのポリシーに関連付けられている Citrix ADC サービスは、圧縮、キャッシュ、書き換え、レスポンダー、およびアプリケーションファイアウォールです。

デフォルトでは、少なくとも 1 つのアプリケーションユニットを持つすべての AppExpert アプリケーションには、デフォルトのアプリケーションユニットが含まれています。これは削除できません。デフォルトのアプリケーション

ユニットは、要求を識別するための規則に関連付けられず、常にアプリケーションユニットの順序で最後に配置されます。これは、他のアプリケーションユニットに対して設定されたルールと一致しない要求を処理するための一連のポリシーを定義します。これにより、すべてのクライアント要求が確実に処理されます。

アプリケーションユニットとそれに関連付けられた規則、ポリシー、およびアクションは、AppExpert アプリケーションテンプレートに含まれています。

Service. Web アプリケーションインスタンスをホストするサーバーの IP アドレスと、アプリケーションがサーバー上でマップされるポートの組み合わせ（形式\<IP address\>:\<Port\>）。多くのリクエストを処理する Web アプリケーションは、複数のサーバーでホストされます。各サーバーは Web アプリケーションのインスタンスをホストすると言われ、Web アプリケーションの各インスタンスは Citrix ADC アプライアンス上のサービスによって表されます。サービスは展開固有であるため、テンプレートには含まれません。テンプレートをインポートした後、サービスを構成する必要があります。

アプリケーション・ユニット・ルール。アプリケーションユニットのトラフィックサブセットの特性を定義する、従来の式またはデフォルトの構文式。次の規則例は、4 つのイメージタイプで構成されるトラフィックサブセットを識別するデフォルトの構文式です。

```
HTTP.REQ.URL.SUFFIX.EQ("bmp")|| HTTP.REQ.URL.SUFFIX.EQ("gif")|| HTTP.REQ.
URL.SUFFIX.EQ("png")|| HTTP.REQ.URL.SUFFIX.EQ("jpg")
```

デフォルトの構文式とクラシックポリシー式の詳細については、「ポリシーと式」を参照してください。

トラフィックサブセット。トラフィックの最適化とセキュリティポリシーの共通セットを必要とするクライアント要求のセット。トラフィックサブセットはアプリケーションユニットによって管理され、ルールによって定義されます。

appExpert アプリケーションの仕組み

October 7, 2021

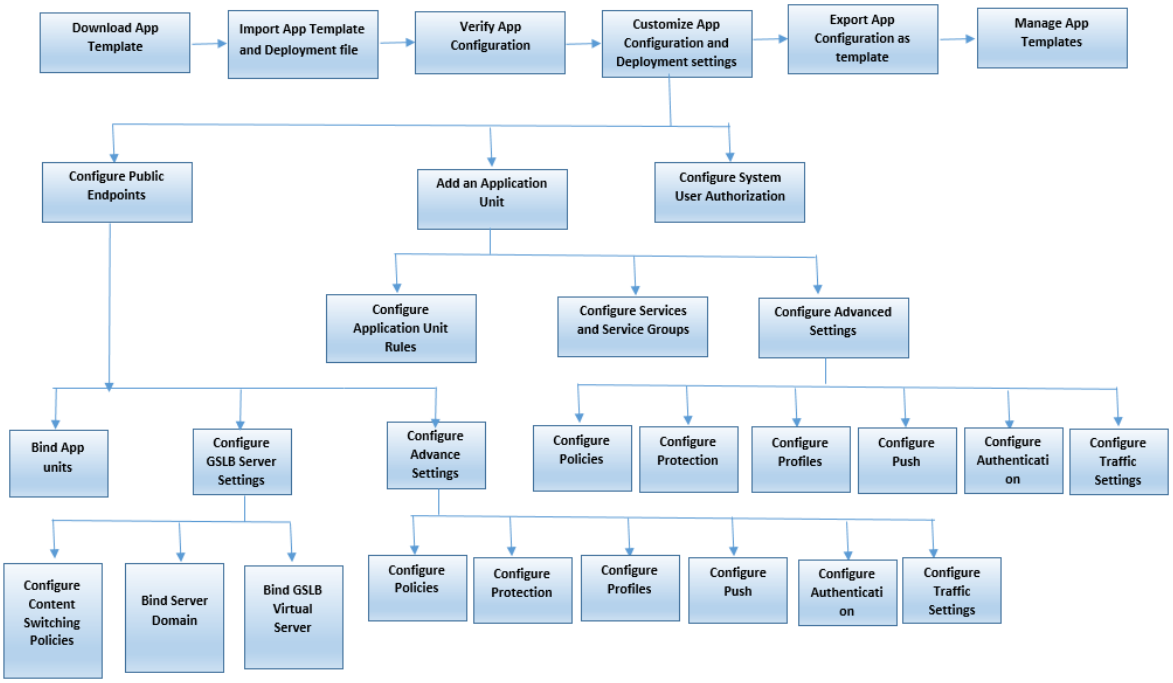
エンドポイントがクライアント要求を受信すると、Citrix ADC アプライアンスは、最上位のアプリケーションユニットに対して構成されたルールに対して要求を評価します。要求がこの規則を満たす場合、要求はアプリケーションユニットに対して設定されているポリシーによって処理され、サービスに転送されます。サービスの選択は、アプリケーションに対してどのサービスが設定されているか、およびアプリケーションユニットに対して設定されたロードバランシングアルゴリズムやパーシステンスメソッドなどの設定によって異なります。

要求がルールを満たさない場合、要求は次の最上位のアプリケーションユニットのルールに対して評価されます。この順序では、要求が規則を満たすまで、各アプリケーションユニット規則に対して要求が評価されます。要求が設定されたルールのいずれも満たさない場合、要求はデフォルトのアプリケーション単位（常に最後のアプリケーション単位）によって処理されます。

AppExpert アプリケーションには、複数のパブリックエンドポイントを設定できます。このような構成では、デフォルトでは、各アプリケーションユニットがすべてのパブリックエンドポイントで受信した要求を処理し、アプリケーション用に構成されているすべてのサービスを負荷分散します。ただし、アプリケーションユニットがパブリックエ

ンドポイントのサブセットからのトラフィックのみを処理し、AppExpert アプリケーション用に構成されたサービスのサブセットのみを負荷分散するように指定できます。

次のフロー図は、組み込みアプリケーションテンプレートを使用するための AppExpert アプリケーションフローシーケンスを示しています。



テンプレートを使用せずにカスタマイズされたアプリケーションを作成する場合は、次の操作を行います。

1. カスタムアプリケーションを作成します。
2. アプリケーションと展開の設定を構成します。
3. 設定を新しいテンプレートファイルにエクスポートします（オプション）。
4. 同様の AppExpert アプリケーション構成を必要とする他の Citrix ADC アプライアンスにテンプレートファイルをインポートする

appExpert の使用を開始する

October 7, 2021

AppExpert アプリケーションを開始するには、まずアプリケーションテンプレートを取得し、そのテンプレートを Citrix ADC アプライアンスにインポートする必要があります。AppExpert アプリケーションがセットアップされたら、アプリケーションが正しく動作していることを確認する必要があります。必要に応じて、要件に合わせて構成をカスタマイズできます。

さまざまなアプリケーションコンポーネントのヒットカウンタを表示することで、定期的に設定を検証および監視で

きます。アプリケーションの認証、認可、監査（AAA）ポリシーを設定することもできます。

アプリケーションを設定するプロセスは、次の 2 つの方法で実行できます。

- 事前構築されたアプリケーションテンプレートの使用
- テンプレートを使用せずにカスタムアプリケーションを作成する。

構築済みのアプリケーションテンプレートを使用してアプリケーションをセットアップする場合は、次の操作を行います。

1. アプリケーションテンプレートをダウンロードします。
2. テンプレートファイルを Citrix ADC アプライアンスにインポートします。
3. アプリケーションのセットアップを確認します。
4. アプリケーションと展開の設定を構成します。
5. 設定を新しいテンプレートファイルにエクスポートします（オプション）。
6. 同様の AppExpert アプリケーション構成を必要とする他の Citrix ADC アプライアンスにテンプレートファイルをインポートします。

Citrix ADC のビデオチュートリアルを使用すると、簡単かつ簡単な方法で Citrix ADC 機能を理解することができます。AppExpert アプリケーションテンプレートを使用してアプリケーションをセットアップする方法については、https://www.youtube.com/watch?v=aqayflvCR_0 ビデオをご覧ください。

アプリケーションテンプレートのダウンロード

October 7, 2021

注： Citrix では、AppExpert アプリケーションテンプレートはサポートされなくなり、コピーをダウンロードすることはできません。Citrix ADC バージョン 13.0 以前を使用している場合は、Citrix サポートに連絡してアプリケーションテンプレートのコピーを入手できます。

AppExpert アプリケーションを設定するには、まず Citrix コミュニティ Web サイト（http://community.citrix.com）からローカルコンピュータまたは Citrix ADC アプライアンスにアプリケーションテンプレートをダウンロードする必要があります。アプリケーションテンプレートはインポートおよびエクスポートされるため、組織内または組織間でアプリケーション固有の設定を簡単に共有できます。アプリケーションテンプレートには、次のエンティティのセットが含まれます。

1. アプリケーションコンポーネント (Web ページ、ファイル、アーカイブ、Web サービスなど)
2. アプリケーションコンポーネントのトラフィック管理エンティティ（仮想サーバの IP アドレスと関連するロードバランシングアルゴリズム、SSL オフロード設定など）。
3. アプリケーショントラフィックの最適化に使用される Citrix ADC ポリシー。

注：アプリケーションテンプレートは、異なる種類の Citrix ADC アプライアンスを構成するために、異なるバージョンで使用できます。

アプリケーション・テンプレートのインポート

October 7, 2021

Citrix ADC ソフトウェアバージョン 9.3 以降の場合、各 AppExpert テンプレートにはテンプレートファイルと展開ファイルの 2 つの XML ファイルがあります。両方のファイルをローカルコンピュータから Citrix ADC アプライアンスにインポートする必要があります。テンプレートファイルは、コンピューターから Citrix ADC アプライアンスの AppExpert アプリケーションテンプレートディレクトリにインポートするか、Citrix ADC アプライアンスにファイルをアップロードしてからアプライアンスからインポートできます。

注：アプライアンスからテンプレートをインポートする場合は、テンプレートで使用可能な変数値を指定する必要があります。

テンプレート・ファイルをインポートすると、アプリケーション構成およびデプロイメント情報がターゲット・アプリケーションに自動的に入力されます。アプライアンスは、NITRO API を介してテンプレートファイルからすべての設定をインポートします。デプロイメント・ファイルをインポートしない場合、コンテンツ・スイッチ仮想サーバー構成が設定されたアプリケーションが生成されます。アプリケーションテンプレートと展開ファイルの形式の詳細については、「Citrix ADC アプリケーションテンプレートと展開ファイルについて」を参照してください。

テンプレートをインポートするときに、デプロイメントファイルを含めない場合は、システムがテンプレートから自動的に生成するアプリケーションでパブリックエンドポイントを設定する必要があります。HTTP 用のエンドポイント、HTTPS 用の別のエンドポイント。HTTPS タイプのパブリックエンドポイントを設定する場合は、必ず SSL 機能を有効にし、サーバ証明書をバインドし、サーバ証明書と証明書キーファイルをインクルードしてください。

テンプレートをインポートした後のエンドポイントの構成の詳細については、「パブリックエンドポイントの設定」を参照してください。

GUI を使用して AppExpert アプリケーションテンプレートファイルを Citrix ADC アプライアンスにインポートするには：

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、[テンプレートのインポート] をクリックします。
3. [インポート] ページで、次のパラメータを設定します。
   a) アプリケーション名（必須）
   b) テンプレートファイル (必須)
   c) 配置ファイルを使用
4. [**Continue**] をクリックして、アプリケーション構成および展開情報をアプリケーションに自動入力します。

Citrix ADC のビデオチュートリアルを使用すると、簡単かつ簡単な方法で Citrix ADC 機能を理解することができます。アプリケーションテンプレートのインポート方法については、https://www.youtube.com/watch?v=AR9TwSD9uJM ビデオをご覧ください。

アプリケーション構成の検証とテスト

October 7, 2021

GUI には、AppExpert アプリケーションのエンティティの状態を示すアイコンが含まれています。これらのアイコンは、アプリケーションおよびアプリケーションユニットに対して表示され、Citrix ADC アプライアンスがサービスやエンティティに対して定期的に実行するヘルスチェックに基づいています。次の表に、アイコンとその意味を示します。

アイコン	エンティティ	意味
	アプリケーション	少なくとも 1 つのパブリックエンドポイントがアップしています。アプリケーションは、稼働中のパブリックエンドポイントからのクライアント要求を受け入れます。
	アプリケーションユニット	アプリケーションユニットがアップしています。少なくとも 1 つのサービスまたはサービスグループがアップしている場合、アプリケーションユニットはアップ状態です。
	アプリケーション	パブリックエンドポイントがアウトアウトアウト（無効）です。このインジケータは、AppExpert アプリケーション用にパブリックエンドポイントが 1 つしか構成されていない場合に表示されます。
	アプリケーション	アプリケーション用に構成されているすべてのエンドポイントがアウトアウトアウトアウトサービスです。このインジケータは、アプリケーションに複数のエンドポイントが設定されている場合にのみ表示されます。
	アプリケーションユニット	アプリケーションユニットに設定されているすべてのサービスがダウンしています。

各アプリケーションとそのアプリケーションユニットのアイコンが常に緑色になっていることを確認する必要があります。アプリケーションに表示されるアイコンが緑色でない場合は、パブリックエンドポイントが正しく設定されていることを確認します。アプリケーションユニットに表示されるアイコンが緑色でない場合は、サービスが正しく設定されていることを確認します。ただし、緑色のインジケータは、関連付けられているすべてのエンティティの状態が UP であることを意味するものではありません。これは、アプリケーションがクライアント要求を処理するのに十分なリソース（エンドポイントとサービス）を持っていることを意味します。関連付けられているすべてのエンティティの状態が UP であることを確認するには、アプリケーションの統計ページですべてのエンティティの健全性をチェックします。

構成のカスタマイズ

October 7, 2021

AppExpert アプリケーションが正しく動作していることを確認したら、要件に合わせて構成をカスタマイズできます。

AppExpert アプリケーション構成が正常に動作していることを確認したら、要件に合わせてアプリケーションと展開設定を構成できます。アプリケーションテンプレートとデプロイメントファイルをインポートすると、使用可能な構成設定（アプリケーションユニット、アプリケーションユニット規則、ポリシー、永続性設定、ロードバランシング方法、プロファイル、トラフィック設定など）がターゲットアプリケーションに自動的に移入されます。このアプリケーションでは、各トラフィックサブセットのパブリックエンドポイント、サービス、サービスグループなどの展開設定を構成できます。AppExpert アプリケーションでテンプレートに含まれていないトラフィックサブセットを管理する場合は、トラフィックサブセットのアプリケーションユニットを追加するか、既存のアプリケーションユニットを変更します。設定をカスタマイズした後、アプリケーションが管理する各トラフィックサブセットの評価順序を指定することもできます。

AppExpert アプリケーションの構成は、次の手順で構成されます。

1. パブリックエンドポイントの設定
2. アプリケーションユニットの設定
3. 評価の順序の指定
4. ビジュアライザを使用したアプリケーション設定の表示

また、テンプレートによって提供されるポリシーを設定することもできます。AppExpert アプリケーションテンプレートに、特定の Citrix ADC 機能（書き換えやアプリケーションファイアウォールなど）のポリシーが含まれていない場合は、独自のポリシーを構成できます。

パブリックエンドポイントの構成

October 7, 2021

AppExpert アプリケーションのインポート時にパブリックエンドポイントを指定しなかった場合は、アプリケーションの作成後にパブリックエンドポイントを指定できます。AppExpert アプリケーションには、HTTP タイプのパブリックエンドポイントと HTTPS タイプのパブリックエンドポイントを 1 つ構成できます。

エンドポイントがすでにアプリケーション用に構成されている場合は、AppExpert アプリケーションからエンドポイントの関連付けを解除し、不要になったエンドポイントを削除できます。パブリックエンドポイントと AppExpert アプリケーションの関連付けを解除すると、そのエンドポイントは関連付けられたアプリケーションユニットから自動的にバインド解除されますが、システムからは削除されません。

AppExpert アプリケーションのパブリックエンドポイントを構成するには:

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、パブリックエンドポイントを構成するアプリケーションを右クリックし、[編集] をクリックします。
3. [アプリケーション] ページで、[パブリックエンドポイント] セクションに移動し、鉛筆アイコンをクリックします。
4. [パブリックエンドポイント] スライダで、次のパラメータを設定します。
   a) パブリックエンドポイントタイプ。ラジオボタンを選択して、端点タイプを定義します。
   b) Name：パブリックエンドポイントの名前。
   c) IP アドレス。パブリックエンドポイントの IP アドレス。
   d) ポート。パブリックエンドポイントのポート番号。
   e) プロトコル。プロトコルの種類を HTTP または HTTPS として選択します。
5. [続行**]** をクリックします。
6. [アプリケーション単位] セクションで、リストからアプリケーション単位を選択します。
7. [**Continue**] をクリックして、ポリシーとサーバの詳細を設定します。
8. 「**OK**」をクリックし、「完了」をクリックします。
9. [閉じる] をクリックします。

[パブリックエンドポイントの構成] ダイアログボックスのパラメータの詳細については、「コンテンツの切り替え」を参照してください。

アプリケーションユニットのサービスおよびサービスグループの構成

October 7, 2021

サービスまたはサービスグループを構成する場合は、既存のサービスまたはサービスグループを変更するか、AppExpert アプリケーションに新しいサービスを追加します。アプリケーションテンプレートのインポート時にサービスまたはサービスグループを指定しなかった場合は、追加します。また、アプリケーションのインスタンスをホストするサーバーの数を増やすときに、サービスとサービスグループを追加します。アプリケーションユニットのサービスおよびサービスグループを構成できるのは、AppExpert アプリケーションのサービスまたはサービスグループを構成した後だけです。

AppExpert アプリケーションのサービスまたはサービスグループを構成するには:

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、アプリケーションを右クリックし、[編集] をクリックします。
3. [アプリケーション] ページで、アプリケーション単位を選択し、[続行] をクリックします。
4. [サービスとサービスグループ] セクションで、次の操作を行います。
   a) [サービスバインディング] スライダで、次のパラメータを設定します。
      i. Service. リストから負荷分散サービスを選択するか、新しいサービスを作成します。
      ii. 重さ。サービスの重み値を指定します。
   b) [バインド]、[完了] の順にクリックします。
   c)「サービスグループバインディング」スライダで、次のパラメータを設定します。
      i. サービスグループ名。負荷分散サービスグループを選択するか、新しいサービスグループを作成します。
      ii. [バインド]、[完了] の順にクリックします。
   d)［完了］をクリックします。
5. [続行] をクリックして、他の構成を設定します。

アプリケーション単位の作成

October 7, 2021

Web アプリケーション実装に固有のトラフィックサブセットまたはテンプレートで定義されていないトラフィックサブセットのアプリケーションユニットを追加する必要がある場合があります。アプリケーション・ユニットを作成するときは、アプリケーション・ユニットのルールを構成する必要があります。

AppExpert アプリケーションのアプリケーション単位を作成するには、次のステップを実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、アプリケーションユニットを追加するアプリケーションを右クリックし、[追加] をクリックします。
3. [アプリケーション] ページで、[アプリケーション単位] セクションに移動し、鉛筆アイコンをクリックします。

アプリケーションユニットのポリシー式を設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、アプリケーションユニットを追加するアプリケーションを右クリックし、[追加] をクリックします。
3. ［アプリケーション］ページで、［アプリケーションユニット］セクションに移動し、**+** アイコンをクリックします。をクリックしてユニットを作成し、ポリシー式を追加します。
4. 新しい式の形式を指定するには、次のいずれかの操作を行います。
   a) [規則] ボックスでクラシック式を構成するように指定するには、[クラシック構文] をクリックします。
   b) [規則] ボックスで詳細な式を構成するように指定するには、[既定の構文] をクリックします。

c) [ルール] ボックスで、式を構成します。

5. [**OK**] をクリックします。

アプリケーション・ユニット・ルールの構成

October 7, 2021

特定のタイプのトラフィックを含めるか除外するように、アプリケーションユニット規則を設定できます。規則を構成するときに、式の構文を定義することもできます。

アプリケーション・ユニット・ルールを構成するには、次のステップを実行します。

1. GUI のナビゲーションウィンドウで、[AppExpert] を展開し、[アプリケーション] をクリックします。
2. 詳細ウィンドウで、ルールを変更するアプリケーション単位を右クリックし、[開く] をクリックします。
3. [アプリケーションユニットの構成] ダイアログボックスで、次の操作を行います。
   a) 新しい式の形式を指定するには、次のいずれかの操作を行います。
      - [規則] ボックスでクラシック式を構成するように指定するには、[クラシック構文] をクリックします。
      - [規則] ボックスで詳細な式を構成するように指定するには、[既定の構文] をクリックします。
   b) [ルール] ボックスで、式を構成します。
4. [**OK**] をクリックします。

アプリケーションユニットのポリシーの設定

October 7, 2021

AppExpert アプリケーションでは、圧縮、キャッシュ、書き換え、レスポンダー、およびアプリケーションファイアウォールのポリシーを構成できます。Citrix Community の Web サイトからダウンロードするテンプレートには、最も一般的なアプリケーション管理要件を満たす一連のポリシーが用意されています。これらのポリシーを微調整またはカスタマイズすることができます。特定のアプリケーションユニットに提供されたポリシーのセットに特定の機能のポリシーが含まれていない場合は、その機能に対して独自のポリシーを作成してバインドできます。

テンプレートを使用せずに AppExpert アプリケーションを作成する場合は、Web アプリケーションに必要なすべてのポリシーを構成する必要があります。

GUI では、さまざまなアイコンを使用して、機能に対してポリシーが設定されているかどうかを示します。アプリケーションユニットでは、特定の機能に対してポリシーが設定されている場合、その機能を表すアイコンが表示されます。たとえば、アプリケーション単位に対して圧縮ポリシーが設定されている場合、アプリケーション単位の [圧縮] 列に圧縮アイコンが表示されます。ポリシーが設定されていない機能については、プラス記号（+）を示すアイコンが表示されます。

注：アプリケーションユニットのポリシーを構成する場合、従来の構文またはデフォルトの構文のいずれかであるポリシーと式を構成する必要がある場合があります。さらに、デフォルトの構文ポリシーを構成するときに、Goto 式などのパラメーターを指定し、ポリシーバンクを呼び出す必要がある場合があります。

両方の形式のポリシーと式を設定する方法については、「ポリシーと式」を参照してください。

圧縮ポリシーの構成

従来のポリシーまたは高度なポリシーのいずれかを使用して圧縮を設定できますが、両方のタイプの圧縮ポリシーを同じアプリケーションユニットにバインドすることはできません。

アプリケーションユニットの圧縮ポリシーを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、構成するアプリケーション単位の行で、[圧縮] 列に表示されるアイコンをクリックします。
3. [圧縮ポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。
   - デフォルトの構文圧縮ポリシーを構成する場合は、[既定の構文に切り替える] をクリックします。クラシック圧縮ポリシーをバインドまたは構成する場合、デフォルトの構文ビューを表示している場合は、[クラシック構文に切り替え] をクリックしてクラシックポリシービューに戻り、バインドされたクラシックポリシーの変更を開始するか、新しいクラシック圧縮ポリシーを作成およびバインドします。
     重要: この設定は、ポリシーを挿入するときに表示されるポリシーも決定します。たとえば、デフォルトの構文ビューを表示している場合、[ポリシーの挿入] をクリックすると、[ポリシー名] 列に表示されるリストには、デフォルトの構文ポリシーのみが含まれます。両方のタイプのポリシーを 1 つのアプリケーションユニットにバインドすることはできません。
   - クラシックポリシーを構成する場合は、ポリシーを要求時と応答時のどちらに評価するかに応じて、[Request] または [Response] をクリックします。
     アプリケーション単位に対して、要求時間と応答時間のクラシック圧縮ポリシーの両方を設定できます。すべての要求時間ポリシーを評価した後、一致するものが見つからない場合、アプライアンスは応答時間ポリシーを評価します。
   - アプリケーションユニットに既にバインドされている圧縮ポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[圧縮ポリシーの構成] ダイアログボックスで、ポリシーを変更し、[OK] をクリックします。
     圧縮ポリシーの変更の詳細については、「圧縮」を参照してください。
   - ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
   - ポリシーに割り当てられた優先度を変更するには、優先度の値をダブルクリックし、新しい値を入力します。
   - 割り当てられた優先順位を再生成するには、再生成の優先順位をクリックします。
   - 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー] をクリックします。次に、[圧縮ポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。

圧縮ポリシーの変更の詳細については、「圧縮」を参照してください。

- デフォルトの構文式を設定する場合は、次の操作を行います。
  - [式に移動] 列で、[式を移動] を選択します。
  - [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。

4. [変更を適用] をクリックし、[閉じる] をクリックします。

キャッシュポリシーの設定

キャッシュポリシーを設定するには、デフォルトの構文ポリシーと式のみを使用できます。

アプリケーションユニットのキャッシュポリシーを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、構成するアプリケーション単位の行で、[キャッシュ] 列に表示されるアイコンをクリックします。
3. [キャッシュポリシーの設定] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。
   - ポリシーをリクエスト時に評価するか、レスポンス時に評価するかに応じて、[Request] または [Response] をクリックします。
     アプリケーション単位のリクエスト時間キャッシュポリシーとレスポンス時間キャッシュポリシーの両方を設定できます。すべての要求時間ポリシーを評価した後、一致するものが見つからない場合、アプライアンスは応答時間ポリシーを評価します。
   - アプリケーションユニットに既にバインドされているキャッシュポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[キャッシュポリシーの構成] ダイアログボックスで、ポリシーを変更し、[**OK**] をクリックします。
     キャッシュポリシーの変更については、統合キャッシュを参照してください。
   - ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
   - ポリシーに割り当てられた優先度を変更するには、優先度の値をダブルクリックし、新しい値を入力します。
   - 割り当てられた優先順位を再生成するには、再生成の優先順位をクリックします。
   - 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー] をクリックします。次に、**[**キャッシュポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。
     キャッシュポリシーの変更については、統合キャッシュを参照してください。
   - [式に移動] 列で、[式を移動] を選択します。
   - [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。
4. [変更を適用] をクリックし、[閉じる] をクリックします。

書き換えポリシーの構成

書き換えポリシーを構成するには、デフォルトの構文ポリシーと式のみを使用できます。

アプリケーションユニットの書き換えポリシーを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、構成するアプリケーションユニットの行で、[Rewrite] 列に表示されているアイコンをクリックします。
3. [書き換えポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。
   - ポリシーをリクエスト時に評価するか、レスポンス時に評価するかに応じて、[Request] または [Response] をクリックします。
     アプリケーションユニットには、要求時間および応答時間書き換えポリシーの両方を設定できます。すべての要求時間ポリシーを評価した後、一致するものが見つからない場合、アプライアンスは応答時間ポリシーを評価します。
   - アプリケーションユニットに既にバインドされている書き換えポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[書き換えポリシーの構成] ダイアログボックスで、ポリシーを変更し、[**OK**] をクリックします。
     書き換えポリシーの変更については、書き換えを参照してください。
   - ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
   - ポリシーに割り当てられた優先度を変更するには、優先度の値をダブルクリックし、新しい値を入力します。
   - 割り当てられた優先順位を再生成するには、再生成の優先順位をクリックします。
   - 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで、[新しいポリシー**]** をクリックします。次に、[書き換えポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。
     書き換えポリシーの変更については、書き換えを参照してください。
   - [式に移動] 列で、[式を移動] を選択します。
   - [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。
4. [変更を適用] をクリックし、[閉じる] をクリックします。

レスポンダポリシーの設定

デフォルトの構文ポリシーと式のみを使用して、レスポンダーポリシーを構成できます。

アプリケーションユニットのレスポンダーポリシーを構成するには、次の操作を行います。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、構成するアプリケーション単位の行で、[レスポンダー] 列のアイコンをクリックします。
3. [レスポンダーポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。

- アプリケーションユニットに既にバインドされているフィルターポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[レスポンダーポリシーの構成] ダイアログボックスでポリシーを変更し、[**OK**] をクリックします。
  レスポンダーポリシーの変更の詳細については、レスポンダーを参照してください。
- ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
- ポリシーに割り当てられた優先度を変更するには、優先度の値をダブルクリックし、新しい値を入力します。
- 割り当てられた優先順位を再生成するには、再生成の優先順位をクリックします。
- 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで [新しいポリシー] をクリックします。次に、[レスポンダーポリシーの作成] ダイアログボックスでポリシーを構成し、[作成] をクリックします。
  レスポンダーポリシーの変更の詳細については、レスポンダーを参照してください。
- [式に移動] 列で、[式を移動] を選択します。
- [Invoke] 列で、現在のポリシーが TRUE と評価された場合に呼び出すポリシーバンクを指定します。

4. [変更を適用] をクリックし、[閉じる] をクリックします。

アプリケーションファイアウォールポリシーの設定

Application Firewall には、従来の構文ポリシーとデフォルトの構文ポリシーと式の両方を設定できます。ただし、あるタイプのポリシーがすでにグローバルにバインドされている場合、またはアプライアンスで構成されている仮想サーバーにバインドされている場合は、別のタイプのポリシーをアプリケーションユニットにバインドできません。たとえば、デフォルトの構文ポリシーがすでにグローバルまたは仮想サーバーにバインドされている場合、クラシックポリシーをアプリケーションユニットにバインドすることはできません。

アプリケーションユニットのアプリケーションファイアウォールポリシーを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、設定するアプリケーションユニットの行で、[**Application Firewall**] 列に表示されているアイコンをクリックします。
3. [アプリケーションファイアウォールポリシーの構成] ダイアログボックスで、実行する構成タスクに応じて、次の 1 つまたは複数の操作を行います。
   - アプリケーションファイアウォールポリシーに設定する式のタイプに応じて、[クラシック式] または [高度な式] をクリックします。
     重要: この設定は、ポリシーを挿入するときに表示されるポリシーも決定します。たとえば、[高度な式] を選択した場合、[ポリシーの挿入] をクリックすると、[ポリシー名] 列に表示されるリストには、デフォルトの構文ポリシーのみが含まれます。両方のタイプのポリシーを 1 つのアプリケーションユニットにバインドすることはできません。このオプションは、いずれかのタイプのポリシーがすでにグローバルまたは仮想サーバにバインドされている場合は使用できません。
   - アプリケーションユニットに既にバインドされているアプリケーションファイアウォールポリシーを変更するには、ポリシーの名前をクリックし、[ポリシーの変更] をクリックします。次に、[アプリケーシ

ョンファイアウォールポリシーの構成] ダイアログボックスで、ポリシーを変更し、[OK] をクリックします。

アプリケーションファイアウォールポリシーの変更の詳細については、「ポリシー」を参照してください。

- ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
- ポリシーに割り当てられた優先度を変更するには、優先度の値をダブルクリックし、新しい値を入力します。
- 割り当てられた優先順位を再生成するには、再生成の優先順位をクリックします。
- 新しいポリシーを挿入するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列に表示されるリストで、[新しいポリシー] をクリックします。次に、**[**アプリケーションファイアウォールポリシーの作成] ダイアログボックスで、ポリシーを構成し、[作成] をクリックします。

  アプリケーションファイアウォールポリシーの変更の詳細については、「ポリシー」を参照してください。

4. [変更を適用] をクリックし、[閉じる] をクリックします。

アプリケーションユニットの設定

October 7, 2021

GUI を使用してアプリケーションユニットを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」**>**「アプリケーションユニット」セクションに移動し、プラスアイコンをクリックして、トラフィックサブセットに新しいアプリケーションユニットを追加します。
2. [アプリケーション単位**]** スライダで、次のパラメータを設定します。

   - 名前
   - 式

   エクスプレッションを挿入するには、エクスプレッションコンポーネントを手動で追加するか、エクスプレッションエディタ (Expression Editor) リンクを使用します。式を手動で追加するには、セレクタコンポーネントを入力し、ピリオド（.）を入力して、次のコンポーネントを選択できるリストを表示します。たとえば、「HTTP」と入力し、ピリオドを入力します。ドロップダウンメニューが表示されます。このメニューの内容には、入力した最初のキーワードの後に続くキーワードが表示されます。ドロップダウンメニューからコンポーネントを選択します。「式 *」テキストボックスに、式に追加したコンポーネント（HTTP.REQ など）が表示されます。完全な式が形成されるまで、コンポーネントを追加し続けます。

   エクスプレッションを作成する場合は、エクスプレッションエディタ (Expression Editor) リンクを使用します。[式エディタ] ページでは、ドロップダウンボックスからコンポーネントを選択して式を作成できます。コンポーネントを選択し、「完了」をクリックして、「アプリケーション単位」ページに式を挿入します。
3. [**Continue**] をクリックして、サービスとサービスグループをバインドします。
4. [**Service**] セクションをクリックして、仮想サービスを選択または追加し、アプリケーションユニットにバインドします。

5. [**Continue**] をクリックし、[**Service Group**] セクションをクリックして、仮想サービスグループを選択または追加し、アプリケーションユニットにバインドします。
6. 「バインドして 続行」をクリックして、アプリケーションユニットの詳細設定（ポリシー、メソッド、永続性、保護、プロファイル、プッシュ、認証、トラフィック設定など）を構成します。
7. 各セクションの プラスアイコンをクリックして、設定パラメータを設定します。
8. [**OK]**、[完了**]** の順にクリックします。

GUI を使用してアプリケーションのアプリケーション単位を編集するには、次のステップを実行します。

「**AppExpert** 」>「アプリケーション」に移動し、アプリケーションを選択して「編集」をクリックします。「アプリケーション単位」セクションで、エンティティを選択し、編集アイコンをクリックして、アプリケーション単位の設定を変更します。

注意: 既存のアプリケーション単位の名前とルール式は変更できません。

Citrix ADC のビデオチュートリアルを使用すると、簡単かつ簡単な方法で Citrix ADC 機能を理解することができます。アプリケーションユニットの構成方法については、https://www.youtube.com/watch?v=bJ5_i8fV2hc ビデオをご覧ください。

アプリケーションのパブリックエンドポイントの構成

October 7, 2021

GUI を使用してアプリケーションのパブリックエンドポイントを構成するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動し、アプリケーション・エンティティを選択して「編集」をクリックします。
2. [**Public Endpoint**] セクションで、[**+**] をクリックして新しいパブリックエンドポイントを設定します。
3. [パブリックエンドポイント] スライダーで、次のいずれかの操作を行います。
   a) [**New**] をクリックして、新しいエンドポイントを作成します。
   b) [既存のパブリックエンドポイント] をクリックして、ドロップダウンリストからエンドポイントを選択します。
4. 次のエンドポイントパラメータを設定します。
   a) 名前
   b) IP アドレス
   c) プロトコル
   d) ポート
5. 「続行」をクリックして、アプリケーションユニット、GSLB サーバーバインディング、ポリシー、プロファイル、プッシュ、トラフィック設定、認証などの追加設定を構成します。
6. [**OK]**、[完了**]** の順にクリックします。
7. [続行] をクリックし、[完了] をクリックします。

GUI を使用してアプリケーションのパブリックエンドポイントを編集するには、次の手順に従います。

「**AppExpert** 」>「アプリケーション」に移動し、アプリケーションを選択して「編集」をクリックします。[パブリックエンドポイント] セクションで、エンドポイントを選択し、ペンアイコンをクリックして、エンドポイントの設定を変更します。

GUI を使用してアプリケーションのパブリックエンドポイントを削除するには、次の手順に従います。

[**AppExpert**] > [アプリケーション] > [パブリックエンドポイント**]** に移動し、ペンアイコンをクリックして、エンティティの横にある削除アイコンを表示します。

Citrix ADC のビデオチュートリアルを使用すると、簡単かつ簡単な方法で Citrix ADC 機能を理解することができます。パブリックエンドポイントの設定方法については、https://www.youtube.com/watch?v=z4v-edQiVpw ビデオをご覧ください。

アプリケーションユニットの評価順序の指定

October 7, 2021

アプリケーション・ユニット・ルールは、GUI に配置された順序で評価されます。最上位のアプリケーションユニットに対して設定されたルールは、常に最初に設定され、その後に第 2 の最上位のアプリケーションユニットに対して設定されたルールが続きます。デフォルトのアプリケーション単位は常に最後に評価されます。

要求がアプリケーション単位に設定された規則と一致すると、要求はアプリケーション単位によって処理され、それ以上の照合は実行されません。したがって、2 つ以上のアプリケーションユニットのトラフィックサブセットが重複する場合は、アプリケーションユニットの評価順序が重要な要素になります。2 つ以上のアプリケーションユニットのトラフィックサブセットが重複する場合は、着信要求をアプリケーションユニットのルールと照合する順序を指定する必要があります。

アプリケーション単位の評価順序を指定する手順は、次のとおりです。

1. 「**AppExpert** 」>「アプリケーション」に移動し、アプリケーションを選択して「編集」をクリックします。[**Application Unit**] セクションで、[**Pencil**] アイコンをクリックし、アプリケーション単位の名前の左側にあるチェックボックスにカーソルを合わせます。チェックボックスの横に表示されるアイコンをクリックし、マウスを押したままにして、アプリケーションを優先リスト内の新しい場所まで上下にドラッグします。

アプリケーションユニットの永続性グループの構成

October 7, 2021

AppExpert アプリケーションのアプリケーションユニットの持続性グループを構成できます。AppExpert アプリケーションのコンテキストでは、持続性グループは、共通の持続性設定を適用するために 1 つのエンティティとして扱

うことができるアプリケーションユニットのグループです。アプリケーションをアプリケーション・テンプレート・ファイルにエクスポートすると、永続性グループ設定が含まれており、AppExpert アプリケーションをインポートすると、アプリケーションユニットに自動的に適用されます。

GUI を使用してアプリケーションの永続性グループを構成するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 「アプリケーションビュー」ダイアログボックスで、持続性グループを構成するアプリケーションユニットのアプリケーション名をクリックし、「持続性グループの構成 」をクリックします。
3. 「持続性グループの構成」ダイアログボックスで、次のいずれかの操作を行います。
   - 持続性グループを追加するには、[**Add**] をクリックします。
   - 永続性グループを変更するには、「開く」をクリックします。
4. 「持続性グループの作成」または 「持続性グループの設定」ダイアログボックスで、次のパラメータを設定します。
   - [Group Name]：永続性グループの名前。Citrix ADC アプライアンスが永続性グループをアプリケーションの構成の一部として認識するには、AppExpert アプリケーションの名前を永続性グループの名前にプレフィックスとして含める必要があります。したがって、デフォルトでは、アプライアンスは [Group Name] ボックスにプレフィクスを表示し、そのプレフィクスを削除することはできません。プレフィックスの後に任意の名前を入力します。
   - パーシステンス：仮想サーバのパーシステンスのタイプ。SOURCEIP を選択した場合は、[IPv4 Netmask] ボックスに、永続性セッションを作成するときにアプライアンスが考慮する必要があるビット数を指定するネットワークマスクを入力します。[クッキードメイン] ボックスと [クッキーの名前] ボックスで、送信するドメイン属性を Set-Cookie ディレクティブで指定し、クッキーの名前をそれぞれ指定します。
   - 「タイムアウト」(Timeout)-永続セッションが有効な期間。
   - [Backup Persistence]：グループのバックアップ永続性のタイプ。
   - [Backup Timeout]：バックアップの永続性が有効になる期間（分単位）。
   - アプリケーションユニット-アプリケーションユニットを永続性グループに追加するには、「使用可能なアプリケーションユニット」ボックスでアプリケーションユニットをクリックし、「追加」をクリックします。持続性グループからアプリケーションユニットを削除するには、「構成されたアプリケーションユニット」ボックスでアプリケーションユニットをクリックし、「削除」をクリックします。
5. [**OK**] をクリックします。

AppExpert アプリケーションの表示とアプリケーションのビジュアライザーを使用したエンティティの構成

October 7, 2021

ビジュアライザー機能は、アプリケーションの構成をグラフィカルに表示します。これには、パブリックエンドポイントの名前、パブリックエンドポイントに割り当てられたアプリケーションユニット、およびアプリケーションにバ

インドされたポリシーとサービスの数が含まれます。ビジュアライザーを使用して、AppExpert アプリケーションの構成の視覚的な概要を取得し、表示されるエンティティの一部を構成できます。デフォルトでは、ビジュアライザーには、選択したアプリケーションのアプリケーションユニット、サービス、およびモニターが表示されます。

アプリケーションビジュアライザーを使用して AppExpert アプリケーションを表示するには:

1. [**AppExpert**] > [アプリケーション**]** に移動し、アプリケーションエンティティを選択して [ビジュアライザー] をクリックします。

ユーザー認証、承認、および監査の構成

October 7, 2021

ユーザーおよびグループの承認を構成して、AppExpert アプリケーションにアクセスできるようにすることができます。権限を設定する AAA ユーザまたはグループがまだ作成されていない場合は、AppExpert から作成してから、アプリケーションアクセスの権限を設定できます。

設定ユーティリティを使用して、アプリケーションの AAA ユーザおよび AAA ユーザグループを設定するには

1.「**AppExpert** 」>「アプリケーション」に移動し、アプリケーション・エンティティを選択して「編集」をクリックします。
2. [詳細設定**]** セクションで、[承認] をクリックし、承認されたユーザーとユーザーグループを構成します。
3. [**AAA** user] セクションをクリックして、許可されたユーザをアプリケーションにバインドします。
4. [**AAA User]** スライダで、パラメータを設定します。
5. [続行] をクリックし、[詳細設定] セクションの [承認ポリシー] をクリックします。
6. [承認ポリシー] スライダーで、承認ポリシーをアプリケーションにバインドします。
7. [続行] をクリックし、[詳細設定**] **** セクションの **[**承認グループ ******] セクションをクリックします。
8. [**AAA Group Binding**] スライダで、認可ユーザグループをアプリケーションにバインドします。
9. [続行] をクリックし、[詳細設定] セクションの [ポリシー] をクリックします。
10. ポリシースライダー、バインドアプリケーションへの 監査の **Syslog** または 監査の **NSLog** 方針。
11. [続行] をクリックし、[完了] をクリックします。

GUI を使用してアプリケーションの AAA ユーザおよび AAA ユーザグループを編集するには、次の手順を実行します。

[**AppExpert**] > [アプリケーション] > [詳細設定**]** に移動し、[認証] をクリックします。次に、編集アイコンをクリックし、ユーザまたはユーザグループの認可設定の値を指定します。

GUI を使用して AAA ユーザおよび AAA ユーザグループを削除するには、次の手順を実行します。

「**AppExpert** 」>「アプリケーション」に移動し、アプリケーションを選択して「編集」をクリックします。[アプリケーション] ページで、[詳細設定] をクリックし、[認証] をクリックします。エンティティの横にある削除アイコンをクリックします。

Citrix ADC アプリケーションの監視

October 7, 2021

AppExpert アプリケーションをカスタマイズした後、アプリケーションとそのすべてのエンティティが正しく動作していることを確認するアプリケーションの統計を表示できます。また、アプリケーションビジュアライザーを使用して、ポリシーや仮想サーバーなどの特定のエンティティに関連付けられた統計を監視することもできます。

また、定期的にさまざまなエンティティのヒットカウンタを表示して、カウンタが更新されていることを確認することもできます。

アプリケーション統計の表示

「アプリケーション」ノードでは、アプリケーションを選択し、アプリケーションの「統計」ページを表示できます。[Statistics] ページでは、パブリックエンドポイントとアプリケーションユニットの状態と状態を監視し、次の統計情報を表示できます。

- 各パブリックエンドポイントとアプリケーションユニットの 1 秒あたりのリクエストとレスポンス。
- 各エンドポイントで、着信トラフィックと発信トラフィックの 1 秒あたりのバイト数。
- アプリケーションユニットのヒットカウンタ、および各アプリケーションユニットのクライアントおよびサーバー接続数。
- アプリケーションユニットにバインドされているサービスの統計情報。

[統計] ページでは、CPU 使用率、メモリ使用量、およびシステムログを表示することもできます。

アプリケーションの統計を表示する手順は、次のとおりです。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、統計を表示するアプリケーションをクリックし、[統計] をクリックします。

アプリケーションビジュアライザーを使用したアプリケーションの監視

アプリケーションビジュアライザを使用して、特定の時点に vservers が受信した 1 秒あたりの要求数と、書き換え、レスポンダー、キャッシュの各ポリシーについて、特定の時点における 1 秒あたりのヒット数を監視できます。

ビジュアライザーで vservers、書き換えポリシー、レスポンダーポリシー、キャッシュポリシーの統計情報を表示するには、次の手順に従います。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、統計情報を表示するアプリケーションを選択し、[ビジュアライザー] をクリックします。
3. 「アプリケーションビジュアライザー」ウィンドウで、次の操作を行います。
    - 統計を表示するには、[統計の表示] をクリックします。
      統計情報は、ビジュアライザーの各ノードに表示されます。この情報はリアルタイムで更新されないため、手動で更新する必要があります。

- 統計情報を更新するには、[統計情報の 更新] をクリックします。

ヒットの表示

さまざまな AppExpert アプリケーションエンティティに提供されるヒットカウンタを使用すると、パブリックエンドポイントとアプリケーションユニットの機能を監視できます。アプリケーションの場合、[Hits] ダイアログボックスには、設定された各パブリックエンドポイントが受信したリクエストの合計数が表示されます。アプリケーションユニットの場合、[ヒット] ダイアログボックスには、アプリケーションユニットが各パブリックエンドポイントから処理したリクエストの数と合計ヒット数が表示されます。ヒットカウンタの表示手順については、設定の確認とテストを参照してください。

アプリケーションの削除

October 7, 2021

アプリケーションとそのアプリケーションユニットが不要になった場合は、削除できます。AppExpert アプリケーションを削除しても、バックエンドサービスは削除されず、アプリケーションが使用したパブリックエンドポイントは他のアプリケーションで使用できるようになります。

アプリケーションを削除するときに、他の場所で使用されていないバインドされたポリシーとアクションを削除するかどうかを指定するように求められます。

GUI を使用してアプリケーションのアプリケーション・ユニットを削除するには、次のステップを実行します。

「**AppExpert** 」>「アプリケーション」に移動し、アプリケーションを選択して「編集」をクリックします。[アプリケーション単位] セクションで、エンティティの横にある削除アイコンをクリックします。

アプリケーションの認証、承認、および監査を構成する

October 7, 2021

アプライアンスに構成するアプリケーションに対して、認証、認可、監査（AAA）を設定できます。アプリケーション用に構成された認証ポリシーは、ユーザーまたはグループがアプリケーションにアクセスしようとしたときに適用する認証の種類を定義します。外部認証を使用する場合、ポリシーによって外部認証サーバも指定されます。アプリケーション用に構成された認可ポリシーは、特定のユーザーまたはグループがアプリケーションにアクセスできるかどうかを指定します。監査ポリシーは、監査ログのタイプ、ロギングが実行されるレベル、およびその他の監査サーバー設定を定義します。認証ポリシーと監査ポリシーでは、従来のポリシー形式が使用されます。

認証ポリシー、認可ポリシー、および監査ポリシーは、任意の順序で設定できます。ただし、アプリケーションに AAA を設定する前に、アプリケーションのパブリックエンドポイントを設定する必要があります。

アプリケーションの認証を構成するには、認証 FQDN、認証仮想サーバー、サーバー証明書、および認証およびセッションポリシーを指定します。認証ポリシーは、アプリケーションに指定された認証仮想サーバーに自動的にバインドされます。

AppExpert アプリケーションの認証を構成するには、次の手順に従います。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   a) [Add] をクリックして、新しいアプリケーションの認証を追加します。
   b) 既存のアプリケーションを変更するには、「編集」をクリックします。
3. 「アプリケーション」ページで、アプリケーションユニットを選択します。
4. アプリケーションユニットのスライダーページで、[詳細設定] セクションの [認証] をクリックします。
5. [認証] セクションで、次のように認証の種類を選択します。
   a) フォームベースの認証
   b) 401 ベースの認証
   c) なし
6. [**OK]** をクリックし、[完了] をクリックします。

アプリケーション認証の構成

ユーザーおよびグループの承認を構成して、AppExpert アプリケーションにアクセスできるようにすることができます。権限を設定する AAA ユーザまたはグループがまだ作成されていない場合は、AppExpert から作成してから、アプリケーションアクセスの権限を設定できます。

AAA ユーザまたはグループが AppExpert アプリケーションにアクセスするための権限を設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、ユーザーまたはグループのアクセスを構成する AppExpert アプリケーションをクリックします。
3. [アプリケーション] ページで、[詳細設定] セクションの [認証] をクリックします。
4. 次のいずれかを行います:
   - 権限を設定する AAA ユーザまたはグループが [Groups/Users] ツリーにすでにある場合は、そのユーザまたはグループを [Groups/Users] ツリーからアプリケーションツリーの [Users] ノードまたは [Groups] ノードにドラッグします。次に、ユーザーまたはグループを右クリックし、[許可] をクリックします。
   - 権限を設定する AAA ユーザまたはグループがアプライアンスで設定されていない場合は、アプリケーションツリーで [Users] または [Groups] を右クリックし、[Add] をクリックします。[AAA グループの作成] または [AAA ユーザの作成] ダイアログボックスで、値を入力し、[作成] をクリックし、[閉じる] をクリックします。
     ユーザーまたはグループは、権限を [許可] に設定して作成されます。アクセス許可の設定を変更するには、グループまたはユーザーを右クリックし、アクセス許可の設定をクリックします。

5. [完了] をクリックし、[閉じる] をクリックします。

アプリケーション監査の構成

アプリケーションの監査ポリシーを構成するときは、ログメッセージの送信先となるサーバー、ログに記録されるメッセージの形式、およびログレベルを指定する必要があります。オプションで、ログ機能や日付形式などの他の設定を構成できます。監査ポリシーは、すべての AppExpert アプリケーションのパブリックエンドポイントに自動的にバインドされます。

アプリケーションの監査ポリシーを構成するには、次の手順に従います。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、監査ポリシーを構成するアプリケーションをクリックします。
3. [Application Unit] スライダーページで、[ポリシー] セクションの [+] アイコンをクリックして、監査ポリシーを構成します。
4. [ポリシー] スライダーページで、ポリシーの種類を [Syslog 監査] または [Nslog 監査] として選択し、[続行] をクリックします。
5. [ポリシーのバインド] セクションで、次のパラメータを設定します。
   a) バインドするポリシーを選択します。バインドのポリシーがない場合は、[+] をクリックして新しいポリシーを作成します。
   b) 新しい監査ポリシーを作成するには、[ポリシー名] の [新しいポリシー**]** をクリックし、[ポリシー] ページで次の操作を行います。
      i. [名前] ボックスに、ポリシーの名前を入力します。
      ii. [名前] ボックスには、サーバー名の先頭に必要な文字列が既に含まれています。文字列は変更できません。
      iii. [監査の種類] リストから、監査の種類 (SYSLOG または NSLOG) を選択します。
      iv. 指定する監査サーバーが既に [サーバー] ボックスの一覧に表示されている場合は、一覧からサーバーを選択し、サーバー設定を変更する場合は [変更] をクリックします。[監査サーバーの構成] ダイアログボックスで、必要に応じて設定を変更し、[OK] をクリックします。[監査サーバーの構成] ダイアログボックスの設定の詳細については、「認証されたセッションの監査」を参照してください。
      v. 新しい監査サーバーを構成する場合は、[新規] をクリックし、[監査サーバーの作成] ダイアログボックスでサーバーの名前を入力し、サーバーの IP アドレス、ポート番号、およびその他の設定を必要に応じて指定します。終了したら、[**OK**] をクリックします。
      vi. [作成] をクリックします。
   c) 作成した新しい監査ポリシーの優先度を変更するには、[優先度] で、優先度を変更するポリシーごとに、優先度の値をダブルクリックし、新しい優先度の値を入力します。
   d) 優先度を再生成するには、[優先度 を再生成] をクリックします。
   e) ポリシーをバインド解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。
   f) ポリシーを変更するには、ポリシーをクリックし、[ポリシーの変更**]** をクリックします。
6. [変更の適用**]** をクリックし、 [閉じる] をクリックします。

アプリケーションの **AAA** の無効化

アプリケーションに AAA を設定したら、そのアプリケーションの AAA 設定を無効にできます。アプリケーションの AAA を無効にしても、設定は失われません。設定を再適用する場合は、アプリケーションの AAA を有効にできます。

アプリケーションの AAA を有効または無効にするには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ペインで、AAA を有効または無効にするアプリケーションをクリックし、次のいずれかの操作を行います。
3. アプリケーションの AAA を無効にするには、[**AAA** をオフにする**]** をクリックします。
4. アプリケーションの AAA を有効にするには、[**Turn On AAA**] をクリックします。

カスタム **Citrix ADC** アプリケーションのセットアップ

October 7, 2021

Citrix ADC アプライアンスを介して管理する Web アプリケーションで AppExpert アプリケーションテンプレートを使用できない場合、または利用可能な AppExpert アプリケーションテンプレートが要件に適合しない場合は、テンプレートなしで AppExpert アプリケーションを作成できます。

テンプレートなしで AppExpert アプリケーションを作成するには、まずアプリケーションユニットとアプリケーションユニットを作成する必要があります。次に、パブリックエンドポイント、サービス、およびサービスグループを構成します。最後に、アプリケーショントラフィックの評価と処理方法を決定するポリシーを設定します。

アプリケーションユニットとアプリケーションユニットを作成し、ポリシーを構成したら、事前に作成された AppExpert アプリケーションテンプレートを使用してアプリケーションを構成する場合と同様に、構成を検証し、正しく動作していることを確認する必要があります。次に、アプリケーションとそのエンティティが正しく動作していることを確認するアプリケーションを監視する必要があります。

アプリケーションの作成

AppExpert アプリケーションを作成すると、アプライアンスによってアプリケーションユニットを追加できるコンテナが作成されます。デフォルトのアプリケーション単位は、最初のアプリケーション単位を作成するまで作成されません。

GUI を使用して AppExpert アプリケーションを作成するには、次のステップを実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、[アプリケーション] を右クリックし、[追加] をクリックします。
3. [アプリケーションの作成] ダイアログボックスの [名前] にアプリケーションの名前を入力し、[**OK**] をクリックします。

アプリケーション単位の作成

Web アプリケーションに関連付けられたトラフィックのサブセットごとに、アプリケーションユニットを作成する必要があります。

GUI を使用して AppExpert アプリケーションのアプリケーション・ユニットを作成するには、次のステップを実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、アプリケーションユニットを追加するアプリケーションを右クリックし、[追加] をクリックします。
3. ［作成］をクリックします。

AppExpert アプリケーションのパブリックエンドポイントの構成

必要なすべてのアプリケーションユニットを作成したら、クライアントが Citrix ADC アプライアンスを介して Web アプリケーションにアクセスできるように、1 つ以上のパブリックエンドポイントを構成する必要があります。

GUI を使用して AppExpert アプリケーションのパブリックエンドポイントを構成するには:

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、パブリックエンドポイントを構成するアプリケーションを右クリックし、[パブリックエンドポイントの構成] をクリックします。
3. アプリケーションの [パブリックエンドポイントの選択] ダイアログボックスで、次のいずれかの操作を行います。
   - 目的のエンドポイントがダイアログボックスに表示されている場合は、対応するチェックボックスをクリックします。
   - すべてのパブリックエンドポイントを指定する場合は、[**Activate All**] をクリックします。
   - AppExpert アプリケーションからエンドポイントの関連付けを解除する場合は、対応するチェックボックスをオフにします。
   - 新しいパブリックエンドポイントを作成する場合は、[**Add**] をクリックします。次に、[パブリックエンドポイントの作成] ダイアログボックスで、エンドポイントの設定を構成し、[**OK**] をクリックします。
     [パブリックエンドポイントの作成] ダイアログボックスでは、エンドポイントの名前、IP アドレス、ポート、およびプロトコルのみを指定できます。パブリックエンドポイントを作成した後で、追加のエンドポイント設定を指定できます。追加のエンドポイント設定を指定するには、エンドポイントを作成した後、[パブリックエンドポイントの選択] ダイアログボックスでエンドポイントをクリックし、[開く] をクリックします。次に、[パブリックエンドポイントの構成] ダイアログボックスで追加の設定を指定し、[**OK**] をクリックします。
     [パブリックエンドポイントの作成**]** および **[**** パブリックエンドポイントの構成 ******] ダイアログボックスのパラメータの詳細については、「コンテンツの切り替え」を参照してください。
   - パブリックエンドポイントを変更する場合は、エンドポイントをクリックし、[開く] をクリックします。次に、［パブリックエンドポイントの構成］ダイアログボックスで、エンドポイントの設定を変更し、［**OK**］をクリックします。

[パブリックエンドポイントの構成] ダイアログボックスのパラメータの詳細については、「コンテンツの切り替え」を参照してください。

4. [閉じる] をクリックします。

アプリケーションユニットのパブリックエンドポイントの構成

アプリケーションユニットの場合は、AppExpert アプリケーションテンプレートから作成されたアプリケーションのパブリックエンドポイントを指定するのと同じ方法でパブリックエンドポイントを指定します。アプリケーションユニットのエンドポイントのサブセットを指定する方法の詳細については、「アプリケーションユニットのエンドポイントの設定」を参照してください。

GUI を使用してアプリケーションユニットのエンドポイントを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、パブリックエンドポイントを指定するアプリケーションユニットを右クリックし、[パブリックエンドポイントの構成] をクリックします。
3. アプリケーションユニットの [パブリックエンドポイントの選択] ダイアログボックスで、次のいずれかの操作を行います。
   - アプリケーションユニットのエンドポイントを初めて指定する場合は、アプリケーションユニットにバインドしないエンドポイントに対応するチェックボックスをオフにします。
   - ダイアログボックスにリストされているが、現在アプリケーションユニットにバインドされていないエンドポイントを指定する場合は、対応するチェックボックスをオンにします。
4. [**OK**] をクリックします。

AppExpert アプリケーションのサービスおよびサービスグループの構成

アプリケーションとサービスグループは、AppExpert アプリケーションのサービスおよびサービスグループを構成した後にのみ、アプリケーションユニットで使用できます。したがって、アプリケーションユニットのサービスを構成する前に、AppExpert アプリケーションのサービスとサービスグループを構成する必要があります。AppExpert アプリケーション用に構成するすべてのサービスとサービスグループは、同じプロトコル (HTTP または HTTPS) を使用する必要があります。テンプレートから作成されていない AppExpert アプリケーションのサービスおよびサービスグループを構成する手順は、テンプレートから作成されたアプリケーションの手順と同じです。

GUI を使用して AppExpert アプリケーションのサービスまたはサービスグループを構成するには:

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、サービスまたはサービスグループを構成するアプリケーションを右クリックし、[バックエンドサービスの構成] をクリックします。
3. [バックエンドサービスの構成] ダイアログボックスで、次のいずれかを実行します。
   - サービスを構成するには、[サービス] タブをクリックします。
   - サービスグループを構成するには、[****** サービスグループ ******] タブをクリックします。
4. [サービスグループ**]** タブまたは **[**サービスグループ] タブで、次のいずれかの操作を行います。

- 目的のサービスまたはサービスグループがタブに表示されている場合は、対応するチェックボックスをオンにします。
- すべてのサービスまたはサービスグループを指定する場合は、[**Activate All**] をクリックします。
- 新しいサービスまたはサービスグループを作成する場合は、[**Add**] をクリックします。次に、[サービスの作成] ダイアログボックスまたは [サービスグループの作成] ダイアログボックスで、サービスまたはサービスグループの設定をそれぞれ構成し、[作成] をクリックします。
- サービスを変更する場合は、そのサービスをクリックし、[開く] をクリックします。次に、[サービスの構成] ダイアログボックスまたは [サービスグループの作成] ダイアログボックスで、サービスまたはサービスグループの設定をそれぞれ構成し、[**OK**] をクリックします。

[サービスの作成]、[サービスの構成]、および [サービスグループの作成] の各ダイアログボックスの設定については、「負荷分散」を参照してください。

アプリケーションユニットのサービスおよびサービスグループの設定

サービスとサービスグループを設定したら、各アプリケーションユニットのサービスとサービスグループを設定する必要があります。ただし、各バックエンドサービスが Web アプリケーションに関連付けられているすべてのコンテンツをホストする場合は、この手順は必要ありません。アプリケーションユニットに関連付けられたコンテンツがバックエンドサーバーのサブセットでのみホストされている場合は、アプリケーションユニットのサービスとサービスグループを設定します。

GUI を使用してアプリケーションユニットのサービスまたはサービスグループを設定するには、次の手順を実行します。

1. 「**AppExpert** 」>「アプリケーション」に移動します。
2. 詳細ウィンドウで、サービスまたはサービスグループを構成するアプリケーションユニットを右クリックし、[バックエンドサービスの構成] をクリックします。
3. [バックエンドサービスの構成] ダイアログボックスで、次のいずれかを実行します。
   - サービスを構成するには、[サービス] タブをクリックします。
   - サービスグループを設定するには、[サービスグループ] タブをクリックします。
4. サービスまたは サービスグループ] タブで、次のいずれかの操作を行います。
   - アプリケーションユニットに設定しないサービスまたはサービスグループに対応するチェックボックスをオフにします。アプリケーションユニットに設定するサービスまたはサービスグループに対応するチェックボックスがオンになっていることを確認します。次に、[Weight] 列で、構成済みの各サービスに割り当てる重みを指定します。
   - すべてのサービスまたはサービスグループを指定するには、[**Activate All**] をクリックします。
5. [メソッド] タブと [持続性] タブと [詳細] タブで、目的のパラメータを指定します。
6. [**OK**] をクリックします。

ポリシーの構成

テンプレートを使用せずに作成された AppExpert アプリケーションのポリシーを構成する手順は、テンプレートから作成された AppExpert アプリケーションの手順と同じです。詳細については、「アプリケーションユニットのポリシーの設定」を参照してください。

テンプレートファイルの作成と管理

October 7, 2021

AppExpert アプリケーションをセットアップし、要件に合わせてカスタマイズしたら、構成からテンプレートを作成し、そのテンプレートを他の管理者と共有できます。または、テンプレートを作成し、同様の AppExpert アプリケーション構成を必要とする他の Citrix ADC アプライアンスにテンプレートをインポートすることもできます。これにより、他のアプライアンスで同様のアプリケーションをセットアップするプロセスが簡素化され、迅速化されます。

AppExpert アプリケーションテンプレートファイルは、Citrix ADC アプライアンスのテンプレートディレクトリまたはローカルコンピューターのフォルダーにエクスポートできます。その後、Citrix ADC アプライアンスとの間でテンプレートをアップロードおよびダウンロードし、アプライアンス上の AppExpert アプリケーションテンプレートディレクトリに保存されているテンプレートの名前を変更できます。

AppExpert アプリケーションテンプレートファイルは、Citrix ADC アプライアンスのテンプレートディレクトリまたはローカルコンピューターのフォルダーにエクスポートできます。その後、Citrix ADC アプライアンスとの間でテンプレートをアップロードおよびダウンロードし、アプライアンス上の AppExpert アプリケーションテンプレートディレクトリに保存されているテンプレートの名前を変更できます。

AppExpert アプリケーションのテンプレートファイルへのエクスポート

October 7, 2021

AppExpert アプリケーションをエクスポートすると、すべてのアプリケーション構成情報がテンプレートファイルにエクスポートされ、すべての展開固有の情報が展開ファイルにエクスポートされます。テンプレートファイルの名前に _deployment という文字列が自動的に付加され、配置ファイルの名前が作成されます。どちらのファイルも XML 形式です。アプリケーションテンプレートファイルを Citrix ADC アプライアンスにエクスポートする場合、テンプレートファイルは/nsconfig/nstemplates/アプリケーションディレクトリに保存され、デプロイメントファイルは/nsconfig/nstemplates/アプリケーション/デプロイメント _files/ディレクトリに保存されます。Citrix Gateway アプリケーションを構成している場合は、テンプレートに Citrix Gateway ポリシーを含めるように選択できます。

GUI を使用して AppExpert アプリケーションをテンプレートファイルにエクスポートするには:

1. 「**AppExpert** 」>「アプリケーション」に移動し、アプリケーション・エンティティを選択して「編集」をクリックします。
2. [アプリケーション] ページで、[テンプレートとして エクスポート] リンクをクリックして、アプリケーションの構成と展開の設定をテンプレートとしてエクスポートします。
3. [アプリケーションのエクスポート] スライダで、次のパラメータを設定します。
   a) テンプレートファイル名
   b) 配置ファイル名
4. [続行] と [完了] をクリックします。
5. 「**AppExpert** 」>「アプリケーション」に移動し、「テンプレートの管理」をクリックして、エクスポートされた構成を「テンプレートファイル」タブおよび「配備ファイル」タブにファイルとして表示します。

コンテンツスイッチ仮想サーバ設定のテンプレートファイルへのエクスポート

October 7, 2021

コンテンツスイッチ構成をアプリケーションテンプレートとしてエクスポートすることもできます。コンテンツスイッチング仮想サーバー構成は、[コンテンツスイッチング仮想サーバー] ペインまたは [コンテンツスイッチングビジュアライザー] から、アプリケーションテンプレートにエクスポートできます。コンテンツスイッチ仮想サーバー、関連するすべての負荷分散仮想サーバー、サービス、サービスグループ、ポリシーなどの構成情報がテンプレートファイルにエクスポートされ、展開固有の情報がすべて展開ファイルにエクスポートされます。テンプレートファイルの名前に「_deployment」という文字列が自動的に付加され、配置ファイルの名前が作成されます。どちらのファイルも XML 形式です。アプリケーションテンプレートファイルを Citrix ADC アプライアンスにエクスポートする場合、テンプレートファイルは Citrix ADC アプライアンス上の/nsconfig/nstemplates/アプリケーションディレクトリに保存され、デプロイメントファイルは/nsconfig/nstemplates/アプリケーション/デプロイメント _files/ディレクトリに保存されます。

アプリケーションテンプレートと展開ファイルの形式の詳細については、「Citrix ADC アプリケーションテンプレートと展開ファイルについて」を参照してください。エクスポートされる構成情報には、コンテンツスイッチング仮想サーバー、関連するすべての負荷分散仮想サーバー、サービス、サービスグループ、およびポリシーが含まれます。

ただし、コンテンツスイッチ仮想サーバーが AppExpert アプリケーションのパブリックエンドポイントとして既に構成されている場合は、構成をテンプレートファイルにエクスポートできません。このシナリオでは、関連付けられた AppExpert アプリケーションをテンプレートにエクスポートする必要があります。

AppExpert アプリケーションをテンプレートファイルにエクスポートする方法の詳細については、「AppExpert アプリケーションのテンプレートファイルへのエクスポート」を参照してください。

GUI を使用して、コンテンツスイッチングビジュアライザからコンテンツスイッチング構成をアプリケーション・テンプレート・ファイルにエクスポートするには、次の手順を実行します。

1. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動します。

2. 詳細ウィンドウで、構成をテンプレートファイルとしてエクスポートするコンテンツスイッチ仮想サーバーの名前をクリックし、[ビジュアライザー] をクリックします。
3. コンテンツスイッチングビジュアライザで、コンテンツスイッチングの vserver のアイコンをクリックし、[関連タスク]、[テンプレートの作成] の順にクリックします。
4. [テンプレートとしてエクスポート...] ダイアログボックスで、テンプレートファイルの名前を入力し、次のいずれかの操作を行います。
   - テンプレートファイルをアプライアンスにエクスポートするには、参照（アプライアンス）が表示されていることを確認します。
   - テンプレート・ファイルをコンピュータにエクスポートするには、「参照」（アプライアンス）ドロップダウン・メニューをクリックし、「ローカル」をクリックして、ファイルを保存する場所を参照し、「保存」をクリックします。
5. 以下の情報を入力します：
   - 概要説明：インポート中に AppExpert アプリケーションテンプレートを紹介するテキスト。このテキストは、テンプレートのインポート時に AppExpert テンプレートウィザードの「アプリケーション名の指定」ページに表示されます。
   - サマリーの説明：テンプレートのインポート時に AppExpert テンプレートウィザードの [サマリ] ページに表示するサマリーです。
   - 作成者: テンプレートの作成者の名前。
   - [**Major**]：テンプレートのメジャーバージョン番号。
   - マイナー：テンプレートのマイナーバージョン番号。この番号はメジャーバージョン番号に追加され、インポート時に、AppExpert テンプレートウィザードの [概要] ページに Major.Minor という形式で表示されます。
6. [OK] をクリックします。

GUI を使用して [Content Switching Virtual Servers] ペインからコンテンツスイッチ設定をアプリケーションテンプレートファイルにエクスポートするには、次の手順を実行します。

1. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、構成をテンプレートファイルとしてエクスポートするコンテンツスイッチ仮想サーバーの名前をクリックし、[AppExpert テンプレートの作成] をクリックします。
3.「コンテンツスイッチング ビジュアライザの手順から、コンテンツスイッチング構成をアプリケーション・テンプレート・ファイルにエクスポートするには」で説明されている手順 4〜6 を実行します。

アプリケーション・テンプレートでの変数の作成

October 7, 2021

アプリケーションテンプレートは、ポリシー式内の変数の宣言と、アプリケーション用に構成されたアクションをサポートします。ポリシー式およびアクションで変数を宣言する機能により、式内の事前設定された値（たとえば、サーバーのホスト名や Rewrite アクションのターゲットなどの設定可能なパラメータ）を、テンプレートのインポート

先の環境に適した値で置き換えることができます。AppExpert アプリケーション・テンプレートに変数が構成されている場合、AppExpert アプリケーション・テンプレートのインポート時に表示される AppExpert テンプレート・ウィザードには、「変数値の指定」ページが表示されます。このページでは、テンプレート用に構成されている変数に適切な値を指定できます。

例として、HTTP 要求の Host ヘッダーの値を評価するように設定されている次のポリシー式を考えます。

```
1 HTTP.REQ.HEADER(”Host”).CONTAINS(”server1”)
2 <!--NeedCopy-->
```

インポート時にサーバー名を設定できるようにするには、文字列「server1」を変数として指定します。テンプレートのインポート時に、「変数」(Variables) タブで変数の新しい値を指定できます。

変数を作成したら、次の操作を実行できます。

- 既存の変数に追加の文字列を割り当てます。文字列の変数を作成した後、同じ式または異なる式の他の部分を選択し、変数に割り当てることができます。変数に割り当てる文字列は同じである必要はありません。インポート時に、変数に割り当てられたすべての文字列は、指定した値に置き換えられます。
- 変数に割り当てられている 1 つまたは複数の文字列を表示します。
- 変数を使用するすべてのエンティティとパラメータのリストを表示します。

アプリケーション・テンプレートのエクスポート・ウィザードでは、次のエンティティの特定のフィールドに変数を定義できます。

- キャッシュポリシー
- 書き換えポリシー
- リライトアクション
- レスポンダーポリシー
- レスポンダーのアクション

GUI を使用してポリシー式またはアクションで変数を設定するには、次の手順を実行します。

1. [**AppExpert] > [**変数**]** に移動します。
2. [変数] ページで、[追加] をクリックします。
3. **[**変数の作成] ページで、次のパラメータを設定します。
   名称。変数の名前。

   スコープ。スコープを「グローバル」または「トランザクション」として選択します。

   タイプ。テキスト、ulong、マップとして変数の種類を選択します。

   失効します。有効期限を入力します。

   満杯の場合 *。マップへの割り当てが設定された max-entries を超える場合に実行するアクション:
   lru-(デフォルト) は、マップ内で最も最近使用されていないエントリを再利用します。

undef-割り当てが、割り当てを実行するポリシーに未定義（Undef）の結果を返すように強制します。
可能な値:undef, lru
デフォルト値:lru.

値がない場合。値の有効期限（秒）。有効期限内に値が参照されない場合、値は削除されます。0（デフォルト）は有効期限がないことを意味します。最小値:0, 最大値:31622400

[初期化値]。この変数の初期化値。代入アクションが値を割り当てる前に参照された場合、シングルトン変数またはマップエントリが設定されます。シングルトン変数またはマップエントリにすでに値が割り当てられている場合、このパラメータを設定してもその変数の値には影響しません。デフォルト:0 (ウルン)、テキストの場合は NULL (最大長): 127

コメント。変数に関する簡単な説明。

4. [閉じる] をクリックします。

テンプレートファイルのアップロードとダウンロード

October 7, 2021

テンプレートファイルは、ローカルコンピュータから Citrix ADC アプライアンスにアップロードすることも、アプライアンスからローカルコンピュータにダウンロードすることもできます。アプライアンスでは、AppExpert アプリケーションテンプレートは常に/nsconfig/nstemplates/applications/である AppExpert アプリケーションテンプレートディレクトリに保存されます。

AppExpert アプリケーションテンプレートをローカルコンピューターから Citrix ADC アプライアンスにアップロードするには:

1. **AppExpert >** アプリケーションに移動します。
2. 詳細ウィンドウで、[テンプレートの管理] をクリックします。
3. [アプリケーションテンプレート] ダイアログボックスで、[アップロード] をクリックします。
4. テンプレートファイルが保存されているディレクトリを参照し、テンプレートファイルをクリックし、[選択] をクリックします。

   テンプレートファイルは、アプライアンスの AppExpert アプリケーションテンプレートディレクトリにアップロードされます。

Citrix ADC アプライアンスからローカルコンピュータに AppExpert アプリケーションテンプレートをダウンロードするには:

1. **AppExpert >** アプリケーションに移動します。
2. 詳細ウィンドウで、[テンプレートの管理] をクリックします。

3. [アプリケーションテンプレート] ダイアログボックスで、ダウンロードする AppExpert アプリケーションテンプレートをクリックし、[ダウンロード] をクリックします。
4. ファイルを保存するディレクトリを参照し、[保存] をクリックします。

Citrix ADC アプリケーションテンプレートと展開ファイルについて

October 7, 2021

Citrix ADC アプリケーションをエクスポートすると、次の 2 つのファイルが自動的に作成されます。

- **Citrix ADC** アプリケーションのテンプレートファイルです。アプリケーション単位、規則、構成済みポリシーなどのアプリケーション構成情報が含まれます。
- 配置ファイル。パブリックエンドポイント、サービス、関連付けられた IP アドレス、設定済みの変数など、デプロイ固有の情報が含まれます。

テンプレートファイルまたは配置ファイルでは、アプリケーション構成情報の各単位は、その単位タイプを表す特定の XML 要素にカプセル化されます。たとえば、各パブリックエンドポイントおよび関連するエンドポイントの詳細は、<appendpoint> および </appendpoint> タグ内にカプセル化され、すべてのエンドポイント要素はタグ <appendpoint_list> と </appendpoint_list> タグ内にカプセル化されます。

注：Citrix ADC アプリケーションをエクスポートした後は、Citrix ADC アプライアンスにアプリケーションをインポートする前に、要素の追加、要素の削除、既存の要素の変更を行うことができます。

Citrix ADC アプリケーションテンプレートの例

以下に、**SharePoint_Team_Site** という Citrix ADC アプリケーションから作成されたテンプレートファイルの例を示します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<template>
<template_info>
   <application_name>SharePoint_Team_Site</application_name>
   <templateversion_major>1</templateversion_major>
   <templateversion_minor>1</templateversion_minor>
   <author>Ed</author>
   <introduction>An application for managing a SharePoint team site
      with images, reports, and, XML content.</introduction>
   <summary>This template includes variables</summary>
   <version_major>9</version_major>
   <version_minor>3</version_minor>
   <build_number>38</build_number>
```



```
</template_info>
<apptemplate>
   <rewrite>
      <rewriteaction_list>
         <rewriteaction>
            <name>Rw_name</name>
            <type>replace</type>
            <target>HTTP.REQ.BODY(10000).AFTER_REGEX(re/number/).
               BEFORE_REGEX(re/address/)</target>
            <stringbuilderexpr>”NA”</stringbuilderexpr>
            <allow_unsafe_pi1>NO</allow_unsafe_pi1>
         </rewriteaction>
         <rewriteaction>
            .
            .
            .
         </rewriteaction>
         .
         .
         .
      </rewriteaction_list>
      <rewritepolicy_list>
         <rewritepolicy>
            <name>Rw_number_NA</name>
            <rule>HTTP.REQ.BODY(100000).CONTAINS(”admin”)</rule>
            <action>Rw_name</action>
         </rewritepolicy>
         <rewritepolicy>
            .
            .
            .
         </rewritepolicy>
         .
         .
         .
      </rewritepolicy_list>
   </rewrite>
   <appunit_list>
      <appunit>
         <name>SharePoint_Team_Sitedefault</name>
         <rule />
         <expressiontype>PE</expressiontype>
         <servicetype>HTTP</servicetype>
         <ipv46>0.0.0.0</ipv46>
         <ipmask>*</ipmask>
```



```
            <port>0</port>
            <range>1</range>
            <persistencetype>NONE</persistencetype>
            <timeout>2</timeout>
            <persistencebackup>NONE</persistencebackup>
            <backuppersistencetimeout>2</backuppersistencetimeout>
            <lbmethod>LEASTCONNECTION</lbmethod>
            <persistmask>255.255.255.255</persistmask>
            <v6persistmasklen>128</v6persistmasklen>
            <pq>OFF</pq>
            <sc>OFF</sc>
            <m>IP</m>
            <datalength>0</datalength>
            <dataoffset>0</dataoffset>
            <sessionless>DISABLED</sessionless>
            <state>ENABLED</state>
            <connfailover>DISABLED</connfailover>
            <clttimeout>180</clttimeout>
            <somethod>NONE</somethod>
            <sopersistence>DISABLED</sopersistence>
            <redirectportrewrite>DISABLED</redirectportrewrite>
            <downstateflush>DISABLED</downstateflush>
            <gt2gb>DISABLED</gt2gb>
            <ipmapping>0.0.0.0</ipmapping>
            <disableprimaryondown>DISABLED</disableprimaryondown>
            <insertvserveripport>OFF</insertvserveripport>
            <authentication>OFF</authentication>
            <authn401>OFF</authn401>
            <push>DISABLED</push>
            <pushlabel>none</pushlabel>
            <l2conn>OFF</l2conn>
         </appunit>
         <appunit>
            .
            .
            .
         </appunit>
         .
         .
         .
      </appunit_list>
      </apptemplate>
      <parameters>
         <property_list>
            <property>
```



```
            <variable_definition_list>
               <variable_definition>
                  <name>body_size</name>
                  <defaultvalue>10000</defaultvalue>
                  <description>Evaluation Scope</description>
                  <startindex>14</startindex>
                  <length>5</length>
               </variable_definition>
               .
               .
               .
            </variable_definition_list>
            <object_type>rewriteaction</object_type>
            <object_name>Rw_name</object_name>
            <name>target</name>
         </property>
         .
         .
         .
      </property_list>
   </parameters>
</template>
<!--NeedCopy-->
```

配置ファイルの例

前の例の **SharePoint_Team_Site** アプリケーションに関連付けられたデプロイメント・ファイルを次に示します。

```
<?xml version="1.0" encoding="UTF8" ?>
<template_deployment>
   <template_info>
      <application_name>SharePoint_Team_Site</application_name>
      <templateversion_major>1</templateversion_major>
      <templateversion_minor>1</templateversion_minor>
      <author>Ed</author>
      <introduction>An application for managing a SharePoint team site
         with images, reports, and, XML content.</introduction>
      <summary>This template includes variables</summary>
      <version_major>9</version_major>
      <version_minor>3</version_minor>
      <build_number>38</build_number>
   </template_info>
```



```
    <appendpoint_list>
       <appendpoint>
          <ipv46>10.111.111.1</ipv46>
          <port>80</port>
          <servicetype>HTTP</servicetype>
       </appendpoint>
    </appendpoint_list>
    <service_list>
       <service>
          <ip>10.102.29.5</ip>
          <port>80</port>
          <servicetype>HTTP</servicetype>
       </service>
       <service>
          .
          .
          .
       </service>
       .
       .
       .
    </service_list>
    <variable_list>
       <variable>
          <name>body_size</name>
          <description>Evaluation Scope</description>
          <value>10000</value>
       </variable>
       <variable>
          .
          .
          .
       </variable>
       .
       .
       .
    </variable_list>
</template_deployment>
<!--NeedCopy-->
```


テンプレートファイルの削除

October 7, 2021

アプリケーションテンプレートとその構成が不要になった場合は、削除できます。テンプレートを削除すると、アプリケーションテンプレートディレクトリに格納されているテンプレート XML ファイルが削除されます。テンプレートファイルを削除すると、削除を確認するプロンプトが表示されます。[はい] をクリックして、選択したファイルをディレクトリから削除します。

GUI を使用してアプリケーション・テンプレート・ディレクトリからテンプレート・ファイルを削除するには、次のステップを実行します。

1. 「**AppExpert**」 **>** 「アプリケーション」に移動し、「テンプレートの管理」をクリックします。[テンプレートファイル**]** タブページまたは **[**配置ファイル ****]** タブページからファイルを選択し、**[**削除 **] をクリックします。

Citrix Gateway アプリケーション

October 7, 2021

Citrix® Citrix ADC® アプライアンスを介して Web アプリケーションを管理するように AppExpert アプリケーションを構成する場合、アプリケーションユニットのセットを作成し、各ユニットのトラフィック最適化とセキュリティポリシーのセットも構成します。各アプリケーションユニットに設定するポリシー（圧縮、キャッシュ、書き換えなどの機能のポリシー）は、そのユニットだけを対象とするトラフィックを評価します。これらのポリシーに加えて、アプリケーション全体の Access Gateway ポリシーを構成して、Access Gateway 経由でアクセスされたときのアプリケーショントラフィックを最適化することもできます。Access Gateway アプリケーション機能を使用すると、AppExpert アプリケーションの Access Gateway ポリシー（認可、トラフィック、クライアントレスアクセス、TCP 圧縮）を設定できます。AppExpert アプリケーションの Citrix Gateway ポリシーを構成した後、作成する AppExpert アプリケーションテンプレートにポリシー構成を含めることができます。

イントラネットサブネット、ファイル共有、およびその他のネットワークリソースに対して Citrix Gateway ポリシーを構成することもできます。最後に、ユーザーが Citrix Gateway のホームページからアクセスできるようにする場合は、AppExpert アプリケーションおよび特定のリソースのブックマークを作成できます。

Citrix Gateway アプリケーション機能でエンティティを構成するには、GUI を使用します。

Citrix Gateway アプリケーションの仕組み

GUI の [アプリケーション] ノードで AppExpert アプリケーションを作成すると、対応する Access Gateway アプリケーションが [Access Gateway アプリケーション] ノードに自動的に作成されます。さらに、AppExpert アプリケーションの構成済みパブリックエンドポイントを使用するルールは、Access Gateway アプリケーションエントリに対して自動的に作成されます。AppExpert アプリケーション用に複数のエンドポイントが構成されている場合、

ルールには設定済みのパブリックエンドポイントがすべて含まれます。Citrix ADC アプライアンスはこのルールを使用して、AppExpert アプリケーションのパブリックエンドポイントで受信したトラフィックに、構成済みの Access Gateway ポリシーを適用します。AppExpert アプリケーションのパブリックエンドポイントで受信したトラフィックは、まず Citrix Gateway ポリシーに対して評価され、AppExpert アプリケーションのアプリケーションユニット用に構成されたポリシーに対して評価されます。

Access Gateway アプリケーションのクライアントレスアクセスポリシー用に作成される規則は、AppExpert アプリケーション用に構成されたパブリックエンドポイントも使用する高度な式です。したがって、AppExpert アプリケーションの Citrix Gateway ポリシーを構成する前に、AppExpert アプリケーションのパブリックエンドポイントを構成する必要があります。

Citrix Gateway 構成をアプリケーションテンプレートに含めると、IP アドレスやポート情報などの展開固有の情報、およびこの情報から作成されたルールはテンプレートに含まれません。

ファイル共有用の **Citrix ADC** 構成の仕組み

Citrix ADC アプライアンスでは、組織のネットワークでホストされているファイル共有の承認ポリシーを構成できます。

ファイル共有を作成するときは、ファイル共有の名前とファイル共有へのネットワークパスを指定します。ネットワークパスでは、サーバーの名前またはサーバーの IP アドレスのいずれかを指定できます。ファイル共有パスのコンポーネントを使用するルールは、ファイル共有に対して自動的に作成されます。このルールにより、アプライアンスはファイル共有サーバーでホストされているファイルに対する要求を識別できます。ファイル共有に対して構成されている承認ポリシーは、受信要求に適用されます。

ファイル共有の Citrix ADC 構成は、AppExpert アプリケーションテンプレートに保存できません。

イントラネットサブネットの **Citrix ADC** 構成の仕組み

ネットワークの一部を形成するイントラネットサブネットについては、Citrix ADC アプライアンスで承認、トラフィック、および TCP 圧縮のポリシーを構成できます。イントラネットサブネットを追加するときは、イントラネットサブネットの IP アドレスとネットマスクを指定します。これら 2 つのパラメーターを使用するルールは、イントラネットサブネットに対して自動的に作成されます。アプライアンスは、宛先 IP アドレスとネットマスクがサブネットの IP アドレスとネットマスクに設定されている要求に対して、設定済みのポリシーを適用します。

イントラネットサブネットの Citrix ADC 構成は、AppExpert アプリケーションテンプレートに保存できません。

その他のリソースカテゴリの仕組み

[その他のリソース] カテゴリでは、選択したルールを使用して、任意のネットワークリソースの Access Gateway ポリシーを構成できます。ネットワークリソースに対する要求を処理するように Citrix ADC アプライアンスを構成する場合、ネットワークリソースに関連付けられた要求を識別する従来の式を構成します。[その他のリソース] で、ネットワークリソースの認可、トラフィック、クライアントレスアクセス、および TCP 圧縮の各ポリシーを設定でき

ます。Citrix ADC アプライアンスは、構成済みの Citrix Gateway ポリシーを、構成済みのルールに一致するすべての要求に適用します。

［その他のリソース］にあるネットワークリソースの Citrix ADC 構成は、AppExpert アプリケーションテンプレートに保存できません。

エンティティの命名規則

Citrix Gateway アプリケーション機能では、この機能で作成するエンティティの一部に命名規則が適用されます。たとえば、イントラネットサブネットのトラフィックポリシー用に作成するプロファイルの名前は、常にイントラネットサブネットの名前の後にアンダースコア (_) が付いた文字列で始まります。エンティティに指定した名前が、この文字列に追加されます。サブネットの名前が「subnet1」の場合、プロファイルの名前は「subnet1_」で始まります。このような命名規則が必要な場合 (エンティティの名前を入力するテキストボックスなど)、ユーザーインタフェースは、エンティティ名の開始に使用する文字列を自動的に挿入し、変更することはできません。

イントラネットサブネットの追加

October 7, 2021

ネットワークで構成されているイントラネットサブネットにバインドされるトラフィックに対して、承認ポリシーとトラフィックポリシーを指定できます。これらのポリシーのルールは、サブネットに指定したパラメータを使用して自動的に作成されます。

GUI を使用してイントラネットサブネットを構成するには、次の手順を実行します。

1. GUI のナビゲーションペインで、[**AppExpert**] を展開し、[アクセスゲートウェイアプリケーション] をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - イントラネットサブネットを追加するには、[イントラネットサブネット] をクリックし、[追加] をクリックします。
   - イントラネットサブネットを変更するには、イントラネットサブネットをクリックし、[開く] をクリックします。
3. [イントラネットサブネットの作成**]** または **[**イントラネットサブネットの構成] ダイアログボックスで、次の操作を行います。
   a) [名前] ボックスに、追加するイントラネットサブネットの名前を入力します。このパラメータは、既存のイントラネットサブネットでは変更できません。
   b) [IP アドレス] ボックスに、イントラネットサブネットの IP アドレスを入力します。
   c) [ネットマスク] ボックスに、イントラネットサブネットに使用するネットマスクを入力します。
   d)「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

他のリソースの追加

October 7, 2021

[その他のリソース] に追加するネットワークリソースについては、リソースに関連付けられたトラフィックのサブセットを識別する従来の式を設定する必要があります。クラシックエクスプレッションの設定の詳細については、を参照してください。

GUI を使用して他のリソース内のリソースを構成するには、次の手順に従います。

1. GUI のナビゲーションペインで、[AppExpert] を展開し、[**Access Gateway Applications]** をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - リソースを追加するには、[その他のリソース] をクリックし、[追加] をクリックします。
   - リソースを変更するには、リソースをクリックし、[開く] をクリックします。
3. [リソースの作成] または [リソースの構成] ダイアログボックスで、次の操作を行います。
   a) [名前] ボックスに、追加するリソースの名前を入力します。このパラメータは、既存のリソースに対して変更できません。
   b) [Rule] ボックスに、追加するリソースに関連付けられているトラフィックのサブセットを識別するルールを入力します。
   または、[構成] をクリックし、[式の作成] ダイアログボックスで規則を作成します。
   c)「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

認可ポリシーの設定

October 7, 2021

AAA ユーザーおよびグループがリソースにアクセスできるように、Citrix Gateway 認証ポリシーを構成できます。

GUI を使用してリソースにアクセスするための AAA ユーザまたはグループの権限を設定するには、次の手順を実行します。

1. GUI のナビゲーションペインで、[AppExpert] を展開し、[**Access Gateway Applications]** をクリックします。
2. 詳細ペインの [Authorization] カラムで、AAA ユーザおよびグループの認可ポリシーを設定するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. 次のいずれかを行います:
   - 権限を設定する AAA ユーザまたはグループが [Groups/Users] ツリーにすでにある場合は、そのユーザまたはグループを [Groups/Users] ツリーから<application name>ツリーの [Users] ノードまたは [Groups] ノードにドラッグします。次に、ユーザーまたはグループを右クリックし、[許可] をクリックします。

- 権限を設定する AAA ユーザまたはグループがアプライアンスで設定されていない場合は、<application name> ツリーで [Users] または [Groups] を右クリックし、[**Add**] をクリックします。[**AAA** グループの作成] または [**AAA** ユーザの作成] ダイアログボックスで、値を入力し、[作成] をクリックし、[閉じる] をクリックします。
  ユーザーまたはグループは、権限を [許可] に設定して作成されます。アクセス許可の設定を変更するには、グループまたはユーザーを右クリックし、アクセス許可の設定をクリックします。

4. ［閉じる］をクリックします。

トラフィックポリシーの設定

October 7, 2021

Citrix Gateway アプリケーションノードでリソースに対して構成するトラフィックポリシーは、アプリケーションへのクライアント接続を制御します。リソースの規則を設定する必要はありません。リソースの作成時に自動的に作成されるルール。要求プロファイルをトラフィックポリシーに関連付けるだけで済みます。トラフィックプロファイルでは、プロトコル、アプリケーションタイムアウト、ファイルタイプの関連付けなどのパラメータを指定します。

リソースのトラフィックポリシーを設定するには

1. GUI のナビゲーションペインで、[AppExpert] を展開し、[Access Gateway Applications] をクリックします。
2. 詳細ペインの [Traffic] 列で、トラフィックポリシーを構成するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. [**Configure Traffic Policy**] ダイアログボックスで、次の操作を行います。
   - 既存のトラフィックポリシーを指定するには、[**Insert Policy**] をクリックし、[Policy Name] 列でポリシーの名前をクリックします。
   - 新しいポリシーを構成するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。[トラフィックポリシーの作成] ダイアログボックスの [名前] ボックスに、アンダースコア (_) の後にポリシーの名前を入力します。次に、[要求プロファイル] で既存の要求プロファイルを選択するか、[新規] をクリックして新しい要求プロファイルを設定します。既存のプロファイルを選択し、[修正] をクリックしてプロファイルを修正することもできます。
     トラフィックポリシーまたはプロファイルの構成の詳細については、「Citrix Gateway」を参照してください。
   - 挿入したポリシーを変更するには、[ポリシー名] 列でポリシー名をクリックし、[ポリシーの変更] をクリックします。関連するプロファイルのみを変更するには、[プロファイル] 列でプロファイルの名前をクリックし、**[**プロファイルの変更] をクリックします。
   - ポリシーに割り当てられた優先度を再生成するには、[優先度 を再生成する] をクリックします。
   - ポリシーの新しいプライオリティ値を指定するには、[Priority] 列で割り当てられているプライオリティをダブルクリックし、必要な値を入力します。

- ポリシーをバインド解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。

4. [変更を適用] をクリックし、[閉じる] をクリックします。

クライアントレスアクセスポリシーの設定

October 7, 2021

クライアントレスアクセスは、Citrix ADC アプライアンス上のリソースに対して構成されている場合、エンドユーザーは Citrix Gateway クライアントソフトウェアを使用せずにリソースにアクセスできます。ユーザーは、Web ブラウザを使用して Outlook Web Access などのリソースにアクセスできます。リソースのクライアントレスアクセスを設定するには、クライアントレスアクセスプロファイルに関連付けられたクライアントレスアクセスポリシーを設定します。

Citrix Gateway アプリケーション］ノードでリソースのクライアントレスアクセスポリシーを構成するには:

1. GUI のナビゲーションペインで、[**AppExpert**] を展開し、[アクセスゲートウェイアプリケーション] をクリックします。
2. 詳細ペインの [クライアントレスアクセス] 列で、クライアントレスアクセスポリシーを構成するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. [クライアントレスアクセスポリシーの設定] ダイアログボックスで、次の操作を行います。
   - 既存のクライアントレスアクセスポリシーを指定するには、[**Insert Policy**] をクリックし、[**Policy Name**] 列でポリシーの名前をクリックします。
   - 新しいクライアントレスアクセスポリシーを設定するには、[**Insert Policy**] をクリックし、[**Policy Name**] カラムで [**New Policy**] をクリックします。[クライアントレスアクセスポリシーの作成] ダイアログボックスの [名前] ボックスに、アンダースコア (_) の後にポリシーの名前を入力します。次に、[プロファイル] で既存のプロファイルを選択するか、[新規] をクリックして新しいプロファイルを設定します。既存のプロファイルを選択し、[修正] をクリックしてプロファイルを修正することもできます。クライアントレスアクセスポリシーまたはプロファイルの構成の詳細については、「Citrix Gateway」を参照してください。
   - 挿入したポリシーを変更するには、[ポリシー名] 列でポリシー名をクリックし、[ポリシーの変更] をクリックします。関連するプロファイルのみを変更するには、[プロファイル] 列でプロファイルの名前をクリックし、[プロファイルの変更] をクリックします。
   - ポリシーの新しいプライオリティ値を指定するには、[Priority] 列で割り当てられているプライオリティをダブルクリックし、必要な値を入力します。
   - ポリシーをバインド解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。
4. [変更を適用] をクリックし、[閉じる] をクリックします。

TCP 圧縮ポリシーの設定

October 7, 2021

アプリケーションの TCP 圧縮ポリシーを設定して、アプリケーションのパフォーマンスを向上させることができます。TCP 圧縮により、ネットワーク遅延が軽減され、帯域幅要件が軽減され、転送速度が向上します。TCP 圧縮ポリシーを設定する場合は、圧縮アクションをポリシーに関連付けます。圧縮アクションでは、圧縮タイプとして [圧縮]、[GZIP]、[減圧]、または [圧縮なし] のいずれかを指定します。圧縮ポリシーと圧縮アクションの詳細については、Citrix Gatewayを参照してください。

Citrix Gateway アプリケーション］ノードでリソースに対して TCP 圧縮ポリシーを構成するには

1. GUI のナビゲーションペインで、[**AppExpert**] を展開し、[アクセスゲートウェイアプリケーション] をクリックします。
2. 詳細ウィンドウの [TCP 圧縮] 列で、TCP 圧縮ポリシーを構成するアプリケーション、ファイル共有、イントラネットサブネット、またはリソースのアイコンをクリックします。
3. [**TCP** 圧縮ポリシーの構成] ダイアログボックスで、次の操作を行います。
   - 既存の TCP 圧縮ポリシーを指定するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列でポリシーの名前をクリックします。
   - 新しい TCP 圧縮ポリシーを作成するには、[ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。[TCP 圧縮ポリシーの作成] ダイアログボックスの [ポリシー名] ボックスで、アンダースコア (「_」) の後にポリシーの名前を入力します。次に、[アクション] で既存のアクションを選択するか、[新規] をクリックして新しいアクションを設定します。[View] をクリックして、構成済みの圧縮タイプを表示することもできます。
     TCP 圧縮ポリシーまたはアクションの構成の詳細については、「Citrix Gateway、Citrix Gateway のアドバンスドエディション」を参照してください。
   - 挿入したポリシーを変更するには、[ポリシー名] 列でポリシー名をクリックし、[ポリシーの変更] をクリックします。
   - ポリシーに割り当てられた優先度を再生成するには、[優先度 を再生成する] をクリックします。
   - ポリシーの新しいプライオリティ値を指定するには、[Priority] 列で割り当てられているプライオリティをダブルクリックし、必要な値を入力します。
   - ポリシーをバインド解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。
4. [変更の適用**]** をクリックし、 [閉じる] をクリックします。

ブックマークを構成する

December 7, 2021

資格のあるユーザーが使用できる内部アプリケーションまたはリソースのブックマークを構成できます。次に、ブッ

クマークをユーザー、ユーザーグループ、または仮想サーバーにグローバルにバインドし、アクセスインターフェイスでユーザーに対して有効にすることができます。作成したブックマークリンクは、エンタープライズ Web サイトの下の Web サイトペインに表示されます。

詳細については、「Web リンクの作成と適用 」トピックを参照してください。

AppQoE

October 7, 2021

アプリケーションレベルのエクスペリエンス品質（AppQoE）は、Citrix ADC アプライアンスの既存のポリシーベースのセキュリティ機能を、新しいキューイングメカニズムである均等化キューイングを活用する単一の統合機能に統合します。均等化キューイングは、負荷分散された Web サーバーおよびアプリケーションへの要求をサービスレベルではなく仮想サーバーレベルで管理します。これにより、ロードバランシング後の個別のストリームとしてではなく、ロードバランシングの前にウェブサイトまたはアプリケーションへのすべての要求のキューイングを 1 つのグループとして処理できます。

AppQoE に統合される機能は、HTTP サービス拒否保護（HDOSP）およびプライオリティキューイング（PQ）です。これらのサービスは総称して、さまざまな問題に対する保護を提供します。

- 単純な過負荷。どのサーバーでも、どのサーバーでも、一度に限られた数の接続しか受け付けられません。保護された Web サイトまたはアプリケーションが一度に受信する要求が多すぎる場合、サージ保護機能はオーバーロードを検出し、サーバーがそれらを受け入れることができるまで余分な接続をキューに入れます。優先度キューイング機能を使用すると、リソースへのアクセスを最も必要とするすべてのユーザーが他の優先度の低い要求を待つことなく、アクセスできるようになります。AppQoE 機能は、要求したリソースが使用できないことをユーザーに通知する代替 Web ページを表示します。
- サービス拒否 **(DOS)** 攻撃。公開されているリソースは、そのサービスを停止させ、正当なユーザーによるアクセスを拒否することを目的とする攻撃に対して脆弱です。サージ保護およびプライオリティキューイング機能は、他のタイプの高負荷に加えて DOS 攻撃の管理に役立ちます。さらに、HTTP サービス拒否保護機能は、Web サイトに対する DOS 攻撃、疑わしい攻撃者にチャレンジを送信し、クライアントが適切な応答を送信しない場合に接続をドロップします。

Citrix ADC オペレーティングシステムの最新バージョンまでは、これらの機能はサービスレベルで実装されていました。つまり、各サービスには独自のキューが割り当てられています。サービスレベルのキューは機能しますが、いくつかの欠点もあります。そのほとんどは、Citrix ADC アプライアンスが、キューイングに依存する保護機能を実装する前に要求の負荷を分散する必要があるためです。キューイングの前に保護機能を実装すると、さまざまな利点があります。そのいくつかを以下に示します。

- プライオリティキューイング機能で設定された接続の絶対プライオリティを維持できます。
- サービスが状態遷移しても、接続はサービスレベルのキューにあるため、フラッシュされません。
- サービス拒否攻撃など、高負荷の期間中、ロードバランシングの前に HTTP DoS が実行されるため、ロードバランサが対処しなければならない前に、ロードバランサからの不要なトラフィックまたは優先度の低いトラ

フィックを検出して迂回できます。

AppQoE は、均等なキューイングの実装に加えて、共通の目標を達成するためにそれぞれ異なるツールセットを提供する一連の機能を統合しています。つまり、ネットワークリソースを過剰または不適切な需要から保護します。これらの機能を共通のフレームワークに配置することで、より簡単に構成および実装できます。

AppQoE の有効化

October 7, 2021

AppQoE を設定するには、まずこの機能を有効にする必要があります。

コマンドラインを使用して AppQoE を有効にするには

コマンドプロンプトで、次のコマンドを入力します。

- enable ns feature appqoe
- show ns feature

例:

```
> enable ns feature appqoe
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 ...
 1)     AppQoE                          AppQoE               ON
 Done
<!--NeedCopy-->
```

GUI を使用して AppQoE を有効にするには

1. [システム] > [設定] に移動します。
2. 詳細ペインで、[高度な機能の構成] をクリックします。
3. [高度な機能の構成] ダイアログボックスで、[**AppQoE**] チェックボックスをオンにします。
4. [**OK**] をクリックします。

appQoE アクション

October 7, 2021

AppQoE 機能を有効にした後、要求を処理するための 1 つ以上のアクションを設定する必要があります。

> 重要:
>
> アクションを作成するために個別のパラメータは必要ありませんが、少なくとも 1 つのパラメータを含める必要があります。そうしないと、アクションを作成できません。

コマンドラインを使用して **AppQoE** アクションを構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appqoe action <name> [-priority <priority>] [-respondWith (ACS|NS)[<customfile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction (**SimpleResponse** | **HICResponse**)]`
- `show appqoe action`

例

プライオリティキューを中および最低のプライオリティキューに対して 10 および 1000 のポリシーキュー深度でプライオリティキューイングを設定するには、次の手順を実行します。

```
> add appqoe action appqoe-act-basic-prhigh -priority HIGH
 Done

> add appqoe action appqoe-act-basic-prmedium -priority MEDIUM -
   polqDepth 10
 Done

> add appqoe action appqoe-act-basic-prlow -priority LOW -polqDepth
   1000
 Done

> show appqoe action

1.       Name: appqoe-act-basic-prhigh
        ActionType: PRIORITY_QUEUING
        Priority: HIGH
        PolicyQdepth: 0
```



```
        Qdepth: 0

1.      Name: appqoe-act-basic-prmedium
        ActionType: PRIORITY_QUEUING
        Priority: MEDIUM
        PolicyQdepth: 10
        Qdepth: 0

1.      Name: appqoe-act-basic-prlow
        ActionType: PRIORITY_QUEUING
        Priority: LOW
        PolicyQdepth: 1000
        Qdepth: 0
Done
<!--NeedCopy-->
```

コマンドラインを使用して既存の **AppQoE** アクションを変更するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appqoe action <name> [-priority <priority>] [-altContentSvcName <string>] [-altContentPath <string>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-maxConn <positive_integer>] [-delay <usecs>] [-dosTrigExpression <expression>] [-dosAction ( SimpleResponse | HICResponse )]`
- `show appqoe action`

コマンドラインを使用して **AppQoE** アクションを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `rm appqoe action <name>`
- `show appqoe action`

AppQoE アクションを構成するためのパラメータ

- 名を入力します。新しいアクションの名前、または変更する既存のアクションの名前。名前は、文字、数字、またはアンダースコア記号で始まり、1 から文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) の記号で構成できます。
- priority. 要求が割り当てられるプライオリティキュー。保護された Web サーバーまたはアプリケーションが負荷が高く、追加の要求を受け付けることができない場合、リソースが使用可能なときに待機中の要求が満たされる順序を指定します。選択肢は次のとおりです。

1. **HIGH.** リソースが使用可能になったらすぐにリクエストを処理します。
2. **MEDIUM.** HIGH 優先度キュー内のすべての要求を処理した後、要求を処理します。
3. **LOW.** HIGH プライオリティキューおよび MEDIUM プライオリティキュー内のすべての要求を処理した後、要求を処理します。
4. **LOWEST.** 優先度の高いキュー内のすべての要求が満たされた後にのみ、要求を処理します。

優先度が構成されていない場合、Citrix ADC アプライアンスはデフォルトで最低優先度のキューに要求を割り当てます。

- respondWith. 指定したしきい値に達したときに、Citrix ADC が指定したレスポンダーのアクションを実行するように構成します。次のいずれかの設定で使用する必要があります。
  - **ACS**：代替コンテンツサービスからコンテンツを配信します。しきい値：maxConn（最大接続数）または遅延。
  - **NS**： Citrix ADC からの組み込み応答を提供します。しきい値：maxConn（最大接続数）または遅延。
  - アクションなし：代替コンテンツを提供しません。maxConn（最大接続数）または遅延しきい値に達した場合に、最低プライオリティキューに接続を割り当てます。
- altContentSvcName. -responseWith ACS が指定されている場合は、代替コンテンツサービスの名前。通常は、代替コンテンツをホストする Web サーバへの絶対 URL です。
- altContentPath. -responseWath (ACS | NS) が指定されている場合、代替コンテンツへのパス。
- olqDepth. このアクションに関連付けられたポリシーキューのポリシーキューの深さのしきい値。このアクションに関連付けられたポリシーキュー内の接続数が指定した数まで増加すると、後続のリクエストは LOWEST ポリシーキューに割り当てられます。最小値:1 最大値:4,294,967,294
- priqDepth. 指定されたプライオリティキューのポリシーキュー深度のしきい値。現在のアクションに関連付けられたポリシーがバインドされている仮想サーバ上の指定されたキュー内の要求の数が、指定された数まで増加すると、後続の要求は最低優先度キューに割り当てられます。最小値:1 最大値:4,294,967,294
- maxConn. ポリシー規則に一致する要求に対してオープンできる接続の最大数。最小値:1 最大値:4,294,967,294
- delay. ポリシー規則に一致する要求の遅延しきい値（マイクロ秒）。一致する要求がしきい値よりも長く遅延した場合、
  Citrix ADC アプライアンスは指定されたアクションを実行します。NO ACTION が指定されている場合、アプライアンスは最も優先度の高いキューに要求を割り当てます。最小値:1 最大値:599999,999
- dosTrigExpression. DoS アクションをトリガーするためのオプションの第 2 レベルのチェックを追加します。
- dosAction. アプライアンスが、または保護されたサーバーが DoS 攻撃を受けていると判断した場合に実行するアクション。可能な値: 単純応答、HIC 応答。

これらの値は、HTTP-DDoS 攻撃を軽減するために、着信要求の信頼性を検証するための HTTP チャレンジ/レスポンス方式を指定します。

HTTP チャレンジレスポンス生成および検証プロセスでは、AppQoE は Cookie を使用してクライアントの応答を検証し、クライアントが本物であると思われることを確認します。チャレンジを送信すると、Citrix ADC アプライアンスは 2 つのクッキーを生成します。

Header cookie (_DOSQ). Citrix ADC アプライアンスが応答を確認できるように、クライアント固有の情報が含まれます。

Body cookie (_DOSH). クライアントマシンの検証に使用される情報。クライアントのブラウザ (HIC の場合はユーザー) がこのクッキーの値を計算します。Citrix ADC アプライアンスは、その値を期待値と比較して、クライアントを確認します。

_DOSH 値を計算するためにアプライアンスがクライアントに送信する情報は、DoS アクション構成に基づいています。

1. 単純応答：この場合、Citrix ADC アプライアンスは値を分割し、最終的な値を結合するための JavaScript コードを生成します。元の値を計算できるクライアントマシンは本物とみなされます。
2. HICResponse：この場合、Citrix ADC アプライアンスは 2 つの 1 桁の数字を生成し、それらの数字のイメージを生成します。次に、バックパッチフレームワークを使用して、アプライアンスはこれらのイメージを base64 文字列として挿入します。

制限事項

1. これは簡単な CAPTCHA の実装ではありません。そのため、その用語は使用されません。
2. 検証番号は、Citrix ADC で生成された数に基づいており、120 では変更されません。この番号は、動的またはクライアント固有である必要があります。

構成ユーティリティを使用して **AppQoE** アクションを構成するには

1. 「アプリ エキスパート」>「**AppQoE**」**>**「アクション」に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいアクションを作成するには、[追加] をクリックします。
   - 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
3. 「**AppQoE** アクションの作成」または「**AppQoE** アクションの設定」画面で、パラメーターの値を入力または選択します。ダイアログ・ボックスの内容は、「AppQoE アクションを構成するためのパラメータ」で説明されているパラメータに対応しています（アスタリスクは必須パラメータを示します）。
   - 名前—name
   - アクション・タイプ-respondWith
   - プライオリティ—priority
   - ポリシーキューの深さ：polqDepth
   - キューの深さ：priqDepth
   - DOS アクション：dosAction
4. [作成**]** または [**OK]** をクリックします。

AppQoE パラメータ

October 7, 2021

AppQoE パラメーターでは、AppQoE セッションのセッション寿命、カスタマイズされた応答を含むファイルのファイル名、およびキューに配置できるクライアント接続の数を構成します。

コマンドラインを使用して **AppQoE** パラメーター設定を構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appqoe parameter [-sessionLife <secs>] [-avgwaitingclient <positive_integer>] [-MaxAltRespBandWidth <positive_integer>] [-dosAttackThresh <positive_integer>]`
- `show appqoe parameter`

AppQoE パラメータを設定するためのパラメータ

- sessionLife

  代替コンテンツを表示してからアプライアンスが同じコンテンツを再表示するまで待機する秒数。デフォルト値:300 最大最小値:1 最大値:4,294,967,294

- avgwaitingclient

  サービス待機キューに入ることができるクライアント要求の平均数。デフォルト値:1000000 最大値:4,294,967,294

- MaxAltRespBandWidth

  代替応答を送信するときに消費する最大帯域幅。最大値に達すると、帯域幅消費が低下するまでアプライアンスは代替コンテンツの送信を終了します。デフォルト値:100 最小値:1 最大値:4,294,967,294

- dosAtckThrsh

  サービス拒否攻撃のしきい値。アプライアンスが DoS 保護対策で応答する前に、キューで待機する必要がある接続の数。デフォルト値:2000 最小値:0 最大値:4,294,967,294

GUI を使用して **AppQoE** パラメータ設定を構成するには

1. 「**AppExpert** 」>「**AppQoE** 」に移動します。
2. 詳細ウィンドウで、[**AppQoE** パラメーターの構成] をクリックします。
3. 「**AppQoE** パラメータの構成」画面で、パラメーターの値を入力または選択します。ダイアログ・ボックスの内容は、「AppQoE パラメータを構成するためのパラメータ」で説明されているパラメータに対応しています(アスタリスクは必須パラメータを示します)。

- セッション寿命（秒）
- sessionLife
- 平均待機クライアント：avgwaitingclient
- 代替応答帯域幅制限（Mbps）：MaxAltRespBandWidth
- DOS 攻撃しきい値：dosAttackThresh

4. [**OK**] をクリックします。

AppQoE ポリシー

October 7, 2021

AppQoE を実装するには、少なくとも 1 つのポリシーを構成して、特定のキューにキューイングされる接続を区別する方法を Citrix ADC に指示する必要があります。

コマンドラインを使用して **AppQoE** ポリシーを構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
```

例:

次の例では、「Android」を含む User-Agent ヘッダーを持つリクエストを選択し、中優先キューに割り当てます。これらのリクエストは、Google Android オペレーティングシステムを実行しているスマートフォンやタブレットから来ています。

```
> add appqoe action appqoe-act-primd -priority MEDIUM
Done
> add appqoe policy appqoe-pol-primd -rule ”HTTP.REQ.HEADER(”User-Agent
    ”).CONTAINS(”Android”)” -action appqoe-act-primd
Done
> sh appqoe policy appqoe-pol-primd
        Name: appqoe-pol-primd
        Rule: HTTP.REQ.HEADER(”User-Agent”).CONTAINS(”Android”)
        Action: appqoe-act-primd
        Hits: 0

Done
<!--NeedCopy-->
```


AppQoE ポリシーを構成するためのパラメータ

- 名を入力します。AppQoE ポリシーの名前。名前は、文字、数字、またはアンダースコア記号で始まり、1 ～ 127 の文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号で構成できます。アクションの種類を識別するのに役立つ名前を選択する必要があります。
- ルール。アプライアンスに処理する必要がある接続を通知する Citrix ADC 式。
- アクション。接続がポリシーに一致したときに実行する AppQoE アクション。

構成ユーティリティを使用して **AppQoE** ポリシーを構成するには

1. [アプリ エキスパート] > [**AppQoE**] > [ポリシー] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[**Edit**] をクリックします。
3. ポリシーを作成する場合は、[**AppQoE** ポリシーの作成] ダイアログの [名前] テキストボックスに、新しいポリシーの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～127 の文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号で構成できます。このポリシーの目的と効果を識別するのに役立つ名前を選択する必要があります。

   既存のポリシーを変更する場合は、この手順をスキップします。既存のポリシーの名前は変更できません。
4. [**Action**] ドロップダウンリストで、ポリシーが接続に一致したときに実行する AppQoE アクションを選択します。プラス (+) をクリックして [**AppQoE** アクションの追加] ダイアログを開き、新しいアクションを追加します。
5. [ルール**]** テキストボックスに、ポリシー式を直接入力するか、[新規] をクリックしてポリシー式を作成します。[新規] をクリックした場合は、次の手順を実行します。

   a) [式の作成] ダイアログボックスで、[追加] をクリックします。
   1 [式の追加] ダイアログボックスで、[頻繁に使用する式**]** ドロップダウンリストから一般的な式を選択するか、[式の構築] ドロップダウンリストを使用して、フィルタリングするトラフィックを定義する式を作成します。
   独自の式を作成する場合は、[式を作成] 領域の左側にある最初のドロップダウンリストから最初の項を選択します。このリストには、次の選択肢があります。

   - HTTP
   - SYS
   - クライアント
   - SERVER
   - ANALYTICS

- TEXT

デフォルトの選択肢は HTTP です。最初のドロップダウンリストで選択した後（またはデフォルトを受け入れる）、式の右側のドロップダウンリストから次の用語を選択できます。このリスト内の用語とそれに続くその他のリストは、以前の選択肢によって変わります。リストには、有効な選択肢である用語のみが表示されます。式が終了するまで、用語の選択を続けます。

a) 必要な式を作成したら、[**OK**] をクリックします。式が [式] テキストボックスに追加されます。

6. ［作成］をクリックします。式が [ルール] テキストボックスに表示されます。

負荷分散仮想サーバーのエンティティテンプレート

October 7, 2021

> 警告
>
> ：エンティティテンプレート機能は、Citrix ADC 13.0 ビルド 82.x 以降から廃止され、代替としてスタイルブックを使用することをお勧めします。詳細については、「スタイルブック」トピックを参照してください。

エンティティテンプレートは、Citrix ADC アプライアンスで負荷分散仮想サーバーテンプレートを作成するための情報の集合です。これは、負荷分散仮想サーバー用に構成される仕様とデフォルトのセットを提供します。デフォルトのセットを定義するテンプレートを使用すると、複数の構成手順を省略しながら、同様の設定が必要な複数の仮想サーバをすばやく構成できます。

負荷分散仮想サーバーの詳細をテンプレートファイルにエクスポートすることで、エンティティテンプレートを作成できます。これは、Citrix ADC GUI を介してのみ実行できます。Citrix ADC GUI を使用して、エンティティテンプレートのエクスポート、インポート、および管理を行います。エンティティテンプレートを他の管理者と共有し、アプライアンスまたはマシンにローカルに保存したテンプレートを管理できます。アプライアンスまたはローカルコンピューターからエンティティテンプレートをインポートすることもできます。

テンプレートを作成する前に、負荷分散仮想サーバーの構成について理解しておく必要があります。

負荷分散仮想サーバテンプレート

負荷分散エンティティテンプレートは、Citrix ADC アプリケーションテンプレートの作成と同じ方法で作成されます。負荷分散仮想サーバーをテンプレートファイルにエクスポートすると、次の 2 つのファイルが自動的に作成されます。

- ロードバランシング仮想サーバテンプレートファイル。負荷分散仮想サーバー用に構成されているパラメーターの値を格納する XML 要素が含まれます。このファイルには、バインドされたポリシーに関する情報を格納するための XML 要素も含まれています。
- 配置ファイル。サービス、サービスグループ、設定済みの変数など、デプロイ固有の情報を格納する XML 要素が含まれます。

テンプレートおよび配置ファイルでは、構成情報の各単位は、その単位タイプを表す特定の XML 要素にカプセル化されます。たとえば、負荷分散方式パラメータ LBMethod は<lbmethod> タグ</lbmethod> とタグ内にカプセル化されます。

> 注:
>
> 負荷分散仮想サーバーをエクスポートした後、構成情報を Citrix ADC アプライアンスにインポートする前に、要素の追加、要素の削除、および既存の要素の変更を行うことができます。

負荷分散仮想サーバーテンプレートの仕組み

負荷分散仮想サーバーのテンプレートを作成するときは、サーバーのデフォルト値を指定します。読み取り専用にする値、表示しない値、およびユーザーが設定できる値を指定します。また、テンプレートのインポートウィザードを構成するページも構成します。指定したすべての情報と設定は、テンプレートファイルに保存されます。
ユーザーがテンプレートを Citrix ADC アプライアンスにインポートすると、GUI によって、テンプレート用に構成したさまざまなページが表示されます。GUI は読み取り専用パラメータ値を表示し、設定可能なパラメータの値を指定するようユーザに要求します。ユーザーが指示に従うと、アプライアンスは設定値を使用してエンティティを作成します。
[トラフィック管理] ノードから、負荷分散仮想サーバーのエンティティテンプレートを作成または変更できます。
仮想サーバーの詳細をテンプレートにエクスポートするには、テンプレートの次のオプションと設定を指定する必要があります。

- パラメータのデフォルト値。
- デフォルト値がユーザーに表示されるかどうか。
- ユーザーがデフォルト値を変更できるかどうか。
- エンティティインポートウィザードのページ数。ページ名、テキスト、使用可能なパラメータを含みます。
- テンプレートを作成するエンティティにバインドする必要があるエンティティ。

たとえば、負荷分散仮想サーバーテンプレートを作成するときに、テンプレートから作成する仮想サーバーにバインドするポリシーを指定できます。ただし、テンプレートにはバインド情報のみが含まれます。バインドされたエンティティは含まれません。エンティティテンプレートが別の Citrix ADC アプライアンスにインポートされる場合、バインドが成功するためには、インポート時にバインドされたエンティティがアプライアンス上に存在する必要があります。ターゲットアプライアンスにバインドされたエンティティが存在しない場合、（テンプレートが構成された）エンティティはバインドなしで作成されます。バインドされたエンティティのサブセットのみがターゲットアプライアンスに存在する場合、それらはテンプレートから作成されたエンティティにバインドされます。
負荷分散仮想サーバーのテンプレートをエクスポートすると、エンティティの構成設定がテンプレートに表示されます。デフォルトでは、バインドされたすべてのエンティティが選択されますが、必要に応じてバインドを変更できます。既存のエンティティに基づいていないテンプレートの場合と同様に、バインディング情報のみが含まれており、エンティティは含まれません。既存の構成設定とともにテンプレートを保存することも、テンプレートの新しい構成を作成するための基礎として使用することもできます。

負荷分散仮想サーバーテンプレートの変数の構成

ロードバランシング仮想サーバテンプレートは、設定されたロードバランシングパラメータ、およびバインドされたポリシーおよびアクションにおける変数の宣言をサポートします。変数を宣言する機能により、事前設定された値をテンプレートのインポート先の環境に適した値で置き換えることができます。

たとえば、テンプレートを作成する負荷分散仮想サーバーにバインドされたポリシーに対して構成された次の式を考えます。式は、HTTP リクエストの受け入れ言語ヘッダーの値を評価します。
`HTTP.REQ.HEADER("Accept-Language").CONTAINS("en-us")`

インポート時にヘッダーの値を設定できるようにするには、文字列 en-us を変数として指定することができます。

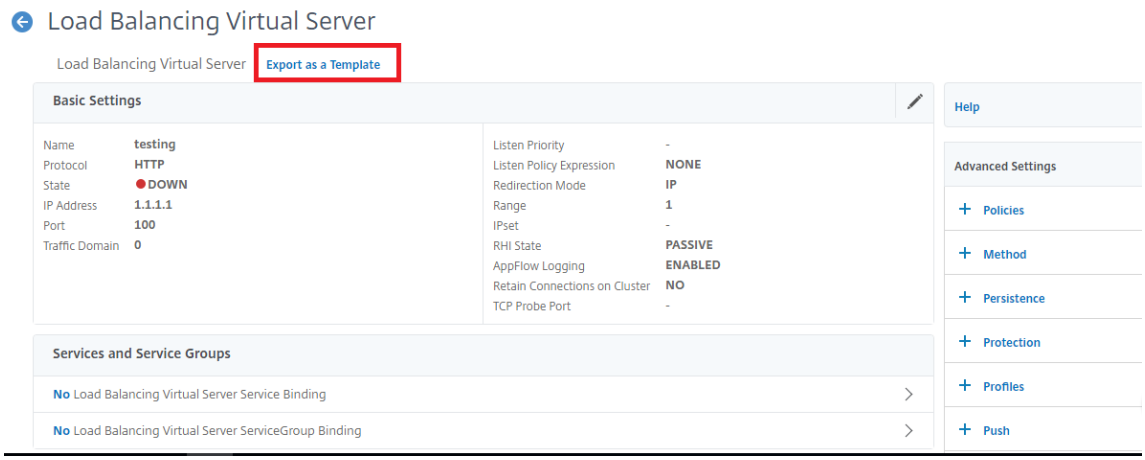
変数を作成したら、次の操作を実行できます。

- 既存の変数にもっと文字列を割り当てます。文字列の変数を作成した後、同じ式または異なる式の他の部分を選択し、変数に割り当てることができます。変数に割り当てる文字列は同じである必要はありません。インポート時に、変数に割り当てられたすべての文字列は、指定した値に置き換えられます。
- 変数に割り当てられている 1 つまたは複数の文字列を表示します。
- 変数を使用するすべてのエンティティとパラメータのリストを表示します

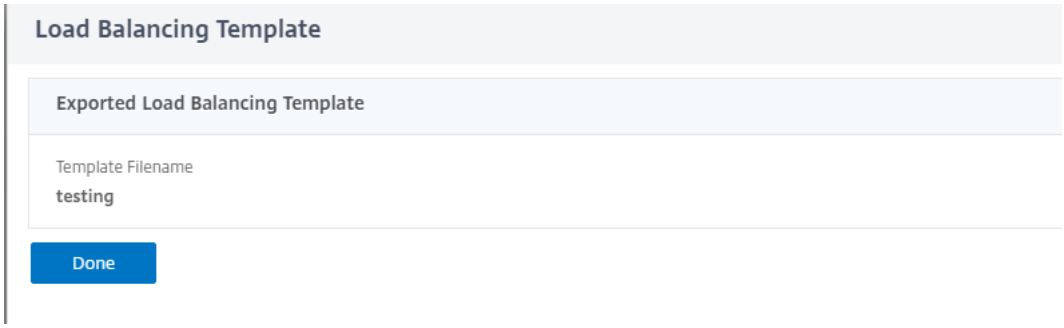
負荷分散仮想サーバーテンプレートで変数を構成するには

Citrix ADC GUI を使用して負荷分散仮想サーバーテンプレートの変数を構成するには、次の手順に従います。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します
2. 詳細ウィンドウで、テンプレートファイルにエクスポートする仮想サーバーを右クリックし、[追加] をクリックします。
3. [負荷分散仮想サーバーの作成] ページで、仮想サーバーのパラメータを設定します。負荷分散仮想サーバーの構成の詳細については、「負荷分散の仕組み」を参照してください。
4. 負荷分散仮想サーバーのパラメータを設定したら、[**Done**] をクリックします。



5. サーバーの詳細を テンプレートファイルとしてエクスポートするには、上部の [テンプレートとしてエクスポート] リンクをクリックします。
6. [負荷分散テンプレートの作成] ページで、テンプレートの設定を入力します。
7. [完了] をクリックします。



負荷分散仮想サーバーテンプレートの変更

テンプレートに設定されたパラメータ、バインディング、ページのみを変更できます。テンプレートの作成時に指定したテンプレートの名前と場所は変更できません。Citrix ADC アプライアンスには、負荷分散仮想サーバーテンプレートを変更するオプションはありません。

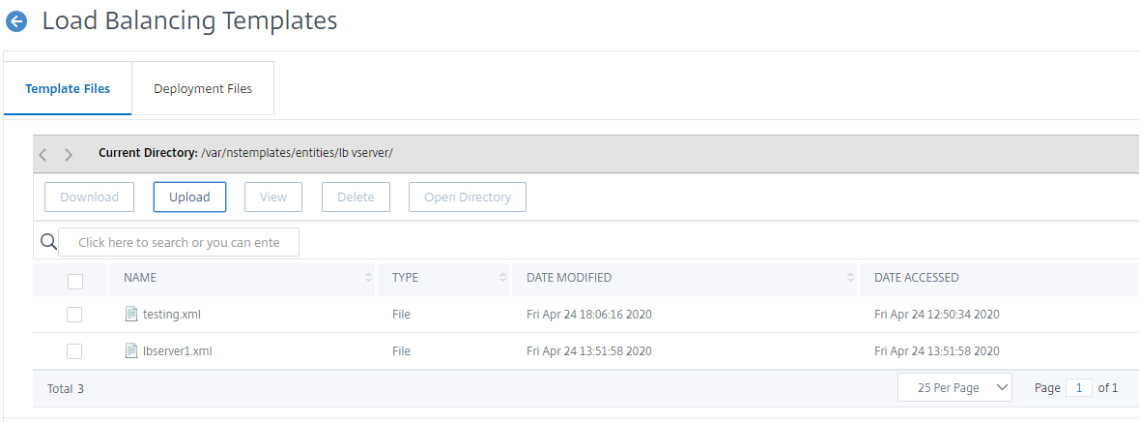
Citrix ADC GUI を使用して負荷分散仮想サーバーを変更するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. [負荷分散仮想サーバー] ページで、エンティティのパラメータを変更します。
3. [完了] をクリックします。
4. [テンプレートとしてエクスポート] リンクをクリックします。
5. 変更された変更は、負荷分散仮想サーバーテンプレートファイルで使用できるようになります。
6. [エクスポートされた負荷分散テンプレート] ページで、[完了] をクリックします。

負荷分散仮想サーバーテンプレートの管理

Citrix ADC GUI を使用して、負荷分散仮想サーバーのテンプレートファイルと展開ファイルを整理できます。

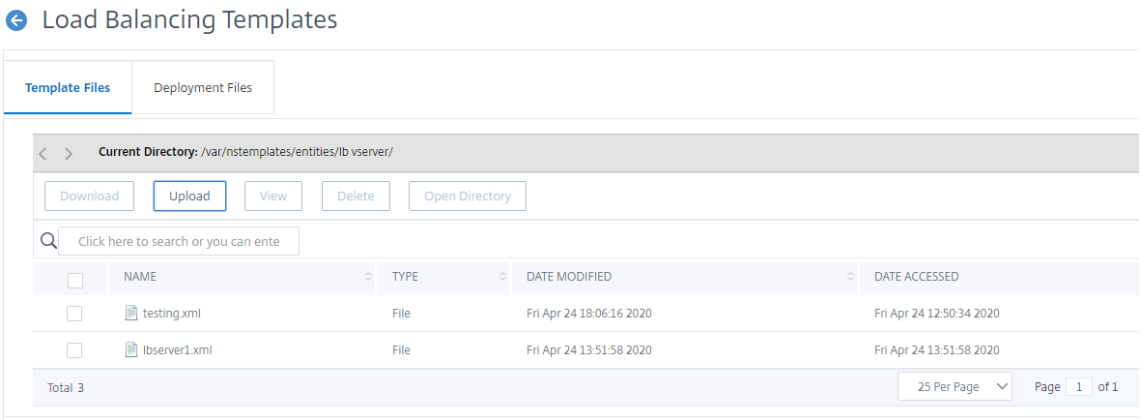
1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. [仮想サーバー] ページで、[テンプレートの管理] アクションを選択します。
3. [負荷分散テンプレート] ページで、[テンプレートファイル] タブをクリックします。
4. 「**Template Files** 」タブ・ページでは、アプライアンス・テンプレート・フォルダに対してテンプレートをアップロードまたはダウンロードできます。



5. [閉じる] をクリックします。

Citrix ADC GUI を使用して負荷分散仮想サーバーエンティティテンプレートをアップロードするには

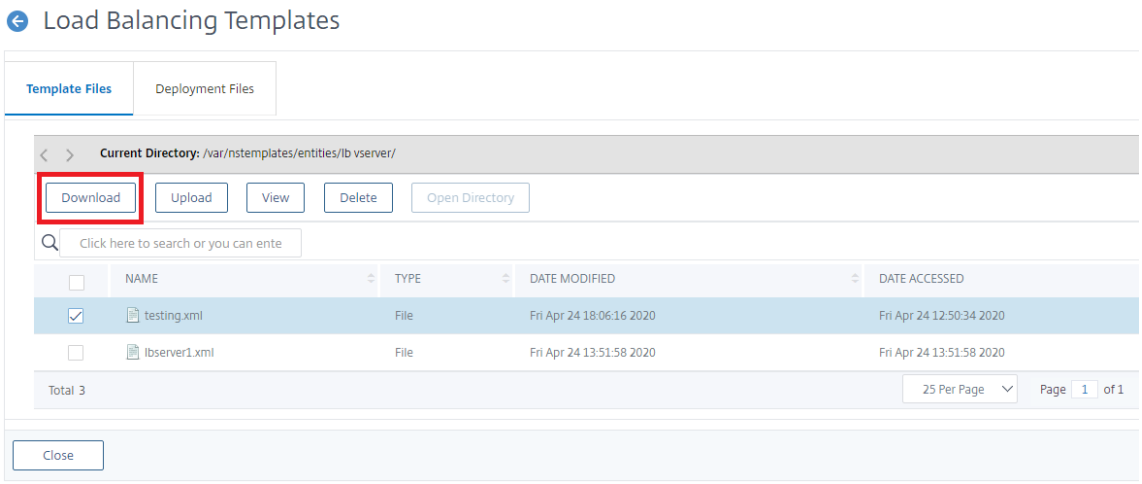
1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. [仮想サーバー] ページで、[アクションの選択] をクリックし、[テンプレートの管理] を選択します。
3. [負荷分散テンプレート] ページで、[テンプレートファイル] タブをクリックします。
4. 「テンプレートファイル」タブページで、「アップロード」をクリックしてテンプレートをアップロードします。
5. [閉じる] をクリックします。



Citrix ADC GUI を使用して負荷分散仮想サーバーエンティティテンプレートをダウンロードするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. [仮想サーバー] ページで、[アクションの選択] をクリックし、[テンプレートの管理] を選択します。
3. [負荷分散テンプレート] ページで、[テンプレートファイル] タブをクリックします。
4. [テンプレートファイル] タブページで、テンプレートファイルを選択し、[ダウンロード] をクリックします。

5. [閉じる] をクリックします。



負荷分散仮想サーバーテンプレートと展開テンプレートの例

次に、「Lbvip」という負荷分散仮想サーバーから作成されたテンプレートファイルの例を示します。

```
COPY

<?xml version="1.0" encoding="UTF-8" ?>
  <template>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <entitytemplate>
      <lbvserver_list>
        <lbvserver>
          <name>Lbvip</name>
          <servicetype>HTTP</servicetype>
          <ipv46>0.0.0.0</ipv46>
          <ipmask>*</ipmask>
          <port>0</port>
          <range>1</range>
          <persistencetype>NONE</persistencetype>
          <timeout>2</timeout>
          <persistencebackup>NONE</persistencebackup>
          <backuppersistencetimeout>2</backuppersistencetimeout>
          <lbmethod>LEASTCONNECTION</lbmethod>
```



```
            <persistmask>255.255.255.255</persistmask>
            <v6persistmasklen>128</v6persistmasklen>
            <pq>OFF</pq>
            <sc>OFF</sc>
            <m>IP</m>
            <datalength>0</datalength>
            <dataoffset>0</dataoffset>
            <sessionless>DISABLED</sessionless>
            <state>ENABLED</state>
            <connfailover>DISABLED</connfailover>
            <clttimeout>180</clttimeout>
            <somethod>NONE</somethod>
            <sopersistence>DISABLED</sopersistence>
            <sopersistencetimeout>2</sopersistencetimeout>
            <redirectportrewrite>DISABLED</redirectportrewrite>
            <downstateflush>DISABLED</downstateflush>
            <gt2gb>DISABLED</gt2gb>
            <ipmapping>0.0.0.0</ipmapping>
            <disableprimaryondown>DISABLED</disableprimaryondown>
            <insertvserveripport>OFF</insertvserveripport>
            <authentication>OFF</authentication>
            <authn401>OFF</authn401>
            <push>DISABLED</push>
            <pushlabel>none</pushlabel>
            <l2conn>OFF</l2conn>
            <appflowlog>DISABLED</appflowlog>
            <icmpvsrresponse>PASSIVE</icmpvsrresponse>
            <lbvserver_cmppolicy_binding_list>
              <lbvserver_cmppolicy_binding>
                <name>Lbvip</name>
                <policyname>NOPOLICY-COMPRESSION</policyname>
                <priority>100</priority>
                <gotopriorityexpression>END</gotopriorityexpression>
                <bindpoint>REQUEST</bindpoint>
              </lbvserver_cmppolicy_binding>
            </lbvserver_cmppolicy_binding_list>
          </lbvserver>
        </lbvserver_list>
      </entitytemplate>
    </template>
<!--NeedCopy-->
```

配置ファイルの例

前の例の仮想サーバーに関連付けられたデプロイメントファイルを次に示します。

COPY

```
<?xml version=”1.0” encoding=”UTF-8” ?>
  <template_deployment>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <service_list>
      <service>
      <ip>1.2.3.4</ip>
      <port>80</port>
      <servicetype>HTTP</servicetype>
      </service>
    </service_list>
    <servicegroup_list>
      <servicegroup>
        <name>svcgrp</name>
        <servicetype>HTTP</servicetype>
        <servicegroup_servicegroupmember_binding_list>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.3.90</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
        <servicegroup_servicegroupmember_binding>
          <ip>1.2.8.0</ip>
          <port>80</port>
        </servicegroup_servicegroupmember_binding>
        <servicegroup_servicegroupmember_binding>
          <ip>1.2.8.1</ip>
          <port>80</port>
        </servicegroup_servicegroupmember_binding>
        <servicegroup_servicegroupmember_binding>
          <ip>1.2.9.0</ip>
          <port>80</port>
        </servicegroup_servicegroupmember_binding>
      </servicegroup_servicegroupmember_binding_list>
    </servicegroup>
  </servicegroup_list>
</template_deployment>
```



```
41
42 <!--NeedCopy-->
```

HTTP callouts

October 7, 2021

特定のタイプの要求の場合、またはポリシー評価中に特定の基準が満たされた場合には、ポリシー評価を短時間停止し、サーバから情報を取得し、取得した情報に依存する特定のアクションを実行することができます。また、特定の種類の要求を受け取ったときに、データベースや Web サーバーでホストされているコンテンツを更新することもできます。HTTP コールアウトを使用すると、これらのすべてのタスクを実行できます。

HTTP コールアウトは、ポリシー評価中に特定の条件が満たされたときに Citrix ADC アプライアンスが生成し、外部アプリケーションに送信する HTTP または HTTPS リクエストです。サーバから取得された情報は、デフォルトの構文ポリシー式で分析でき、適切なアクションを実行できます。HTTP コンテンツスイッチング、TCP コンテンツスイッチング、リライト、レスポンダ、およびトークンベースのロードバランシング方式の HTTP コールアウトを設定できます。

HTTP コールアウトを設定する前に、コールアウトの送信先となるサーバー上にアプリケーションを設定する必要があります。*HTTP* コールアウトエージェントと呼ばれるアプリケーションは、*HTTP* コールアウト要求に必要な情報で応答するように設定する必要があります。HTTP コールアウトエージェントは、Citrix ADC アプライアンスがコールアウトを送信するデータを提供する Web サーバーでもかまいません。HTTP コールアウトへの応答の形式が、ある呼び出しから別の呼び出しに変更されないようにする必要があります。

HTTP コールアウトエージェントを設定したら、Citrix ADC アプライアンスで HTTP コールアウトを構成します。最後に、コールアウトを呼び出すには、適切な Citrix ADC 機能のデフォルトの構文ポリシーにコールアウトを含めて、ポリシーを評価するバインドポイントにポリシーをバインドします。

HTTP コールアウトを設定したら、コールアウトが正しく動作していることを確認する設定を確認する必要があります。

HTTP コールアウトの仕組み

October 7, 2021

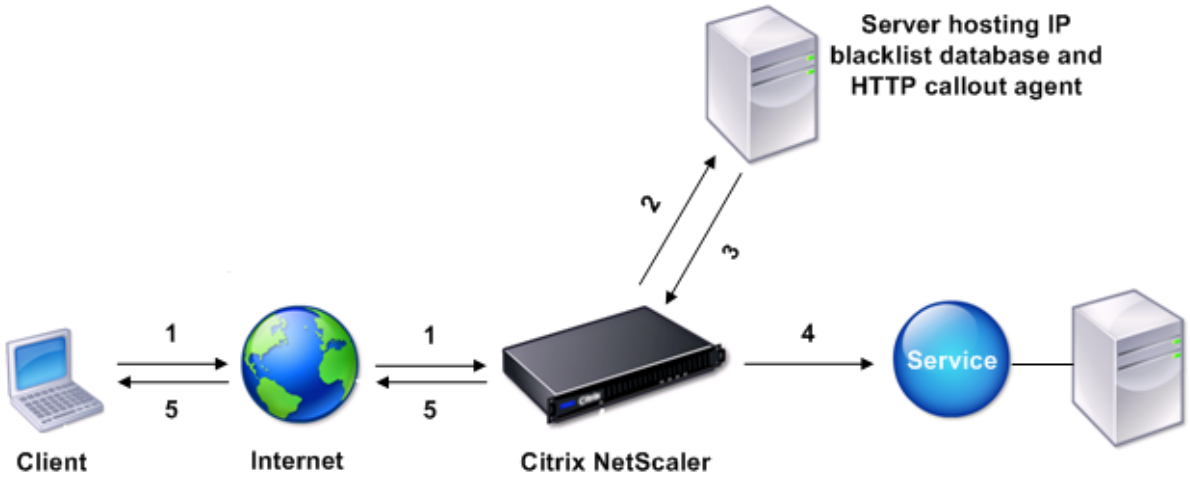
Citrix ADC アプライアンスがクライアント要求を受信すると、アプライアンスはさまざまなバインドポイントにバインドされたポリシーに対して要求を評価します。この評価中に、アプライアンスは HTTP コールアウト式`SYS.HTTP_CALLOUT(<name>)`を検出すると、ポリシー評価を短時間停止し、指定された HTTP コールアウト用に設定されたパラメータを使用して HTTP コールアウトエージェントに要求を送信します。応答を受信すると、アプライアンスは応答の指定された部分を検査し、HTTP コールアウトエージェントからの応答の評価が TRUE また

は FALSE のいずれに評価されたかに応じて、アクションを実行するか、次のポリシーを評価します。たとえば、応答側ポリシーに HTTP コールアウトが含まれている場合、応答の評価が TRUE と評価されると、アプライアンスは応答側ポリシーに関連付けられたアクションを実行します。

HTTP コールアウト構成が正しくないか不完全である場合、またはコールアウトが再帰的に呼び出された場合、アプライアンスは UNDEF 条件を発生させ、未定義のヒットカウンタを更新します。

次の図は、グローバルにバインドされたレスポンダーポリシーから呼び出される HTTP コールアウトの動作を示しています。HTTP コールアウトは、着信要求に関連付けられているクライアントの IP アドレスを含めるように設定されます。Citrix ADC アプライアンスがクライアントからの要求を受信すると、アプライアンスはコールアウト要求を生成し、コールアウトサーバーに送信します。コールアウトサーバーは、ブラックリストに登録された IP アドレスのデータベースと、クライアントの IP アドレスがデータベースにリストされているかどうかをチェックする HTTP コールアウトエージェントをホストします。HTTP コールアウトエージェントは、コールアウト要求を受信し、クライアントの IP アドレスがリストされているかどうかを確認し、Citrix ADC アプライアンスが評価する応答を送信します。応答がクライアントの IP アドレスがブラックリストに登録されていないことを示している場合、アプライアンスは応答を構成済みのサービスに転送します。クライアントの IP アドレスがブラックリストに登録されている場合、アプライアンスはクライアント接続をリセットします

図 1: HTTP コールアウトエンティティモデル



1: Client request
2: HTTP callout request to check whether the client is blacklisted
3: Response from HTTP callout agent
4: Request forwarded to service if 3 indicates a safe IP address
5: Connection RESET if 3 indicates a bad IP address

HTTP リクエストとレスポンスの形式に関する注意事項

October 7, 2021

Citrix ADC アプライアンスは、HTTP コールアウト要求の有効性をチェックしません。したがって、HTTP コールアウトを設定する前に、HTTP 要求の形式を知っておく必要があります。HTTP コールアウトの設定には、HTTP コールアウトエージェントからの応答を評価する式を設定する必要があるため、HTTP レスポンスの形式も知っておく必要があります。

ここでは、次の項について説明します。

- HTTP リクエストのフォーマット
- HTTP レスポンスの形式

HTTP リクエストのフォーマット

HTTP リクエストには、キャリッジリターンと改行で終わる一連の行が含まれます。<CR><LF> or \r\n。

リクエストの最初の行（メッセージ行）には、HTTP メソッドとターゲットが含まれます。たとえば、GET リクエストのメッセージ行には、次の例に示すように、キーワード GET と、フェッチされるオブジェクトを表す文字列が含まれます。

```
GET /mysite/mydirectory/index.html HTTP/1.1\r\n
<!--NeedCopy-->
```

要求の残りの部分には、必須の Host ヘッダー、および該当する場合はメッセージ本文を含む HTTP ヘッダーが含まれます。

リクエストはバンク行（余分な<CR><LF> or \r\n）で終わります。

リクエストの例を次に示します。

```
Get /mysite/index.html HTTP/1.1\r\n
Host: 10.101.101.10\r\n
Accept: \*/\*\r\n
\r\n
<!--NeedCopy-->
```

HTTP レスポンスの形式

HTTP 応答には、ステータスメッセージ、応答 HTTP ヘッダー、および要求されたオブジェクト、または要求されたオブジェクトを提供できない場合は、エラーメッセージが含まれます。

応答の例を次に示します。


```
HTTP/1.1 200 OK\r\n
Content-Length: 55\r\n
Content-Type: text/html\r\n
Last-Modified: Wed, 12 Aug 1998 15:03:50 GMT\r\n
Accept-Ranges: bytes\r\n
ETag:  “04f97692cbd1:377” \r\n
Date: Thu, 19 Jun 2008 19:29:07 GMT\r\n
\r\n
<55-character response>
<!--NeedCopy-->
```

HTTP コールアウトの設定

October 7, 2021

HTTP コールアウトを設定するときは、リクエストのタイプ（HTTP または HTTPS）、送信先、およびリクエストの形式を指定します。応答の期待される形式、および最後に分析する応答の部分。

宛先には、HTTP コールアウトエージェントの IP アドレスとポートを指定します。または、負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーを使用して、HTTP コールアウト要求を管理します。

最初のケースでは、HTTP コールアウト要求は HTTP コールアウトエージェントに直接送信されます。2 番目のケースでは、HTTP コールアウト要求は、指定された仮想サーバーの仮想 IP アドレス (VIP) に送信されます。仮想サーバーは、クライアント要求を処理するのと同じ方法で要求を処理します。たとえば、多数のコールアウトが生成されることが予想される場合、HTTP コールアウトエージェントのインスタンスを複数のサーバーで構成し、これらのインスタンスを (サービスとして) 負荷分散仮想サーバーにバインドし、HTTP コールアウト設定で負荷分散仮想サーバーを指定できます。次に、負荷分散仮想サーバーは、負荷分散アルゴリズムによって決定された設定済みインスタンスの負荷を分散します。

HTTP コールアウト要求の形式については、HTTP コールアウト要求の個々の属性（属性ベースの HTTP コールアウト）を指定するか、HTTP コールアウト要求全体を高度なポリシー式（式ベースの HTTP コールアウト）として指定できます。

次の表では、HTTP コールアウトポリシーの要素について説明します。

詳細については、「ポリシー httpCallout」を参照してください。

パラメーター	説明
名前	コールアウトの名前 (最大 127 文字)

パラメーター	説明
IP アドレスとポート (IP アドレス/ポート) または仮想サーバー名 (vserver)	コールアウトの送信先サーバーの IPv4 または IPv6 アドレス、またはワイルドカード、およびコールアウトの送信先サーバー上のポート、またはワイルドカード。または、サービスタイプが HTTP の負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーの名前。
HTTP メソッド (httpMethod)	HTTP メソッド (httpMethod)。このコールアウトが送信する HTTP リクエストで使用されるメソッド。有効な値:GET または POST。デフォルト:GET。
ホスト式 (HostExpr)	ホスト式 (HostExpr)。Host ヘッダーを構成するための高度なテキスト式。最大長:255。式はリテラル値でも、値を導き出す高度な式でもかまいません。例:「10.101.10.11」,「http.req.header (「ホスト」)」
URL ステム式 (urlStemExpr)	URL ステム式 (urlStemExpr) URL ステムを生成するための高度な文字列式です。最大長:8191. 式には、リテラル文字列または値を導出する式を指定できます。例:「」/mysite/index.html「」「http.req.url」
HTTP ヘッダー (ヘッダー)	HTTP ヘッダー (ヘッダー)。HTTP コールアウトリクエストに HTTP ヘッダーとその値を挿入するための高度なテキスト式。すべてのヘッダーに値を指定します。ヘッダー名を文字列として指定し、ヘッダー値を高度な式として指定します。ヘッダーをスペースで区切って指定します。-headers cip (client.ip.src) hdr (http.req.header (「HDR」)) など。ヘッダーの数は 8 にすることができます。
サーバーに送信する式ベースの要求 (fullReqExpr)	Citrix ADC が 8191 文字の高度な式として送信する正確な HTTP リクエスト。このパラメータを指定する場合は、HttpMethod、HostExpr、urlStemExpr、ヘッダー、およびパラメーターの引数を省略する必要があります。リクエスト式は、コールアウトが使用されるフィーチャによって制約されます。たとえば、HTTP.RES 式は、要求時ポリシーバンクまたは TCP コンテンツスイッチングポリシーバンクでは使用できません。

パラメーター	説明
サーバーに送信する式ベースの要求 (BodyExpr)	リクエストの本文を生成するための高度な文字列式。式には、リテラル文字列または値を導出する式 (client.ip.src など) を含めることができます。-fullReqExpr と相互に排他的です。
パラメーター	コールアウトが送信する HTTP リクエストにクエリパラメータを挿入するための高度な式。設定するすべてのパラメータの値を指定します。コールアウトリクエストが GET メソッドを使用する場合、これらのパラメーターは URL に挿入されます。コールアウトリクエストが POST メソッドを使用する場合、これらのパラメータは POST 本文に挿入されます。クエリパラメータ名を文字列として、値を高度な式として設定します。パラメータ値は URL エンコードされます。パラメータをスペースで区切って指定します 例 パラメータ name1 (「name1」) name2 (http.req.header (「hdr」))。最大 8 つのパラメータを設定できます。
戻り値の型 (ReturnType)	ターゲットアプリケーションがコールアウトへの応答で返すデータのタイプ。有効な値:TEXT: 戻り値をテキスト文字列として扱います。NUM: 戻り値を数値として扱います。BOOL: 戻り値をブール値として扱います。注意: 戻り値の型は、設定後に変更できません。
レスポンスからデータを抽出する式 (ResultExpr)	HTTP コールアウトへの応答から HTTP.RES オブジェクトを抽出する高度な式。最大長は 8191 です。この式の操作は、戻り値の型と一致する必要があります。たとえば、戻り値の種類のテキストを構成する場合、結果式はテキストベースの式である必要があります。戻り値の型が num の場合、結果式 (resultExpr) は次のような数値を返す必要があります。「http.res.body (10000) .length」注: 場合によっては、戻り値の型を TEXT に設定し、サーバーから送信される結果が 16 KB を超えると、結果式が NULL を返すことがあります。たとえば、結果が 16 KB を超える連結文字列になった場合などです。
スキーム	コールアウトサーバーのスキームのタイプ。例:HTTP、https

パラメーター	説明
cacheForsecs	コールアウト応答がキャッシュされる期間（秒）。キャッシュされた応答は、「calloutContentGroup」という名前の統合キャッシュコンテンツグループに保存されます。デュレーションが設定されていない場合、通常のキャッシュ構成を使用してキャッシュしない限り、コールアウト応答はキャッシュされません。このパラメータは、これらの応答に適用される通常のキャッシュ設定よりも優先されます。

注：アプライアンスはリクエストの有効性をチェックしません。リクエストが有効なリクエストであり、機密情報が含まれていないことを確認する必要があります。正しくないまたは不完全な HTTP コールアウト構成は、アクションに関連付けられていない実行時 UNDEF 条件になります。UNDEF 条件は、未定義のヒットカウンタを更新するだけで、誤って設定された HTTP コールアウトをトラブルシューティングできます。ただし、アプライアンスは HTTP コールアウト要求を解析して、コールアウトの特定の Citrix ADC 機能を構成できるようにします。これにより、HTTP コールアウトが自身を呼び出す可能性があります。コールアウトの再帰とその回避方法については、「HTTP コールアウトの再帰を回避する」を参照してください。

最後に、HTTP 要求属性を使用するか、式を使用して HTTP コールアウト要求の形式を定義するかに関係なく、HTTP コールアウトエージェントからの応答の形式と、評価する応答の部分を指定する必要があります。レスポンスタイプには、ブール値、数値、またはテキストを使用できます。この戻り値の型だけに基づいて、コールアウト応答でさらにエクスプレッションメソッドを使用できます。戻り値の型が数値の場合、コールアウト応答で数値ベースの式を使用できます。評価する応答の部分は、式によって指定されます。たとえば、応答にテキストが含まれるように指定した場合、`HTTP.RES.BODY(<unit>)`を使用して、アプライアンスがコールアウトエージェントからの応答の最初の <unit> バイトのみを評価するように指定できます。

コマンドラインでは、最初に
add コマンドを使用して HTTP コールアウトを作成します。コールアウトを追加すると、デフォルト値の GET に設定された HTTP メソッドを除き、すべてのパラメータがデフォルト値の NONE に設定されます。次に、set コマンドを使用してコールアウトのパラメータを設定します。set コマンドは、両方のタイプのコールアウト（属性ベースとエクスプレッションベース）を設定するために使用します。違いは、2 種類のコールアウトを設定するために使用されるパラメータにあります。したがって、次のコマンドライン手順には、属性ベースのコールアウトを設定するための set コマンドと、式ベースのコールアウトを設定するための set コマンドが含まれます。構成ユーティリティでは、これらの構成タスクはすべて 1 つのダイアログボックスで実行されます。

注:HTTP コールアウトをポリシーに入れる前に、戻り値のタイプを除くすべての設定済みパラメータを変更できます。HTTP コールアウトがポリシー内にあると、コールアウトで設定されている式を完全に変更することはできません。たとえば、クライアント.IP.SRC に HTTP.REQ ヘッダー（「マイバル」）を変更することはできません。式に渡される演算子と引数は変更できます。たとえば、`HTTP.REQ.HEADER("myVal1")`を`HTTP.REQ.HEADER("myVal2")`にまたは`HTTP.REQ.HEADER("myVal")`を`HTTP.REQ.HEADER("`

myVal”).AFTER_STR(<string>)に変更できます。set コマンドが失敗した場合は、HTTP コールアウトを作成します。

HTTP コールアウトの設定には、高度なポリシー式を設定する必要があります。高度なポリシー式の構成の詳細については、「高度なポリシー式の構成: はじめに 」トピックを参照してください。

コマンドラインインターフェイスを使用して **HTTP** コールアウトを構成するには

コマンドプロンプトで、次の操作を行います。

HTTP コールアウトを作成します。

```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port<
    port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (
    GET | POST )] [-hostExpr <expression>]  [-urlStemExpr <expression>]
    [-headers <name(value)> ...] [-parameters <name(value)> ...] [-
    bodyExpr <expression>] [-fullReqExpr <expression>] [-scheme ( http |
     https )] [-resultExpr <expression>] [-cacheForSecs <secs>] [-
    comment <string>]

<!--NeedCopy-->
```

例:

```
add policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header(”Host”)'-urlStemExpr ”http
    .req.url” -parameters Name(”My Name”) -headers Name(”MyHeader”)-
    resultExpr ”http.res.body(10000).length”

<!--NeedCopy-->
```

HTTP コールアウトの設定を変更します。

```
set policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr|\*>] [-
    port <port|\*>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod ( GET | POST )] [-hostExpr <string>] [-urlStemExpr <
    string>] [-headers <name(value)> ...] [-parameters <name(value)>
    ...] [-resultExpr <string>]

<!--NeedCopy-->
```


例:

```
> set policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header(”Host”)'-urlStemExpr ”http
    .req.url” -parameters Name(”My Name”) -headers Name(”MyHeader”) -
    resultExpr ”http.res.body(10000).length”
<!--NeedCopy-->
```

FullReqExpr パラメータを使用して HTTP コールアウトを構成します。

```
set policy httpCallout <name> [-vServer <string>] [-returnType <
    returnType>] [-fullReqExpr <string>] [-resultExpr <string>]
<!--NeedCopy-->
```

例:

```
> set policy httpCallout mycallout1 -vserver lbv1 -returnType num
    fullReqExpr q{
 ”GET ” + http.req.url + ”HTTP/” + http.req.version.major + ”.” + http.
    req.version.minor.sub(1) + ”r\nHost:10.101.10.10\r\nAccept: \*/\*\r
    \n\r\n” }


<!--NeedCopy-->
```

HTTP コールアウトの設定を確認します。

```
show policy httpCallout `<name>`

sh policy httpCallout mycallout1
> Name: mycallout1
>Vserver: lbv1 (UP)
Effective Vserver state: UP
Return type: TEXT
Scheme: HTTP
Full REQ expr: ”GET ” + http.req.url + ”HTTP/” + http.req.version.major
    + ”.” + http.req.version.minor.sub(1)+ ”r\nHost:10.101.10.10\r\
    nAccept: \*/\*\r\n\r\n”
Result expr: http.res.body(100)
```

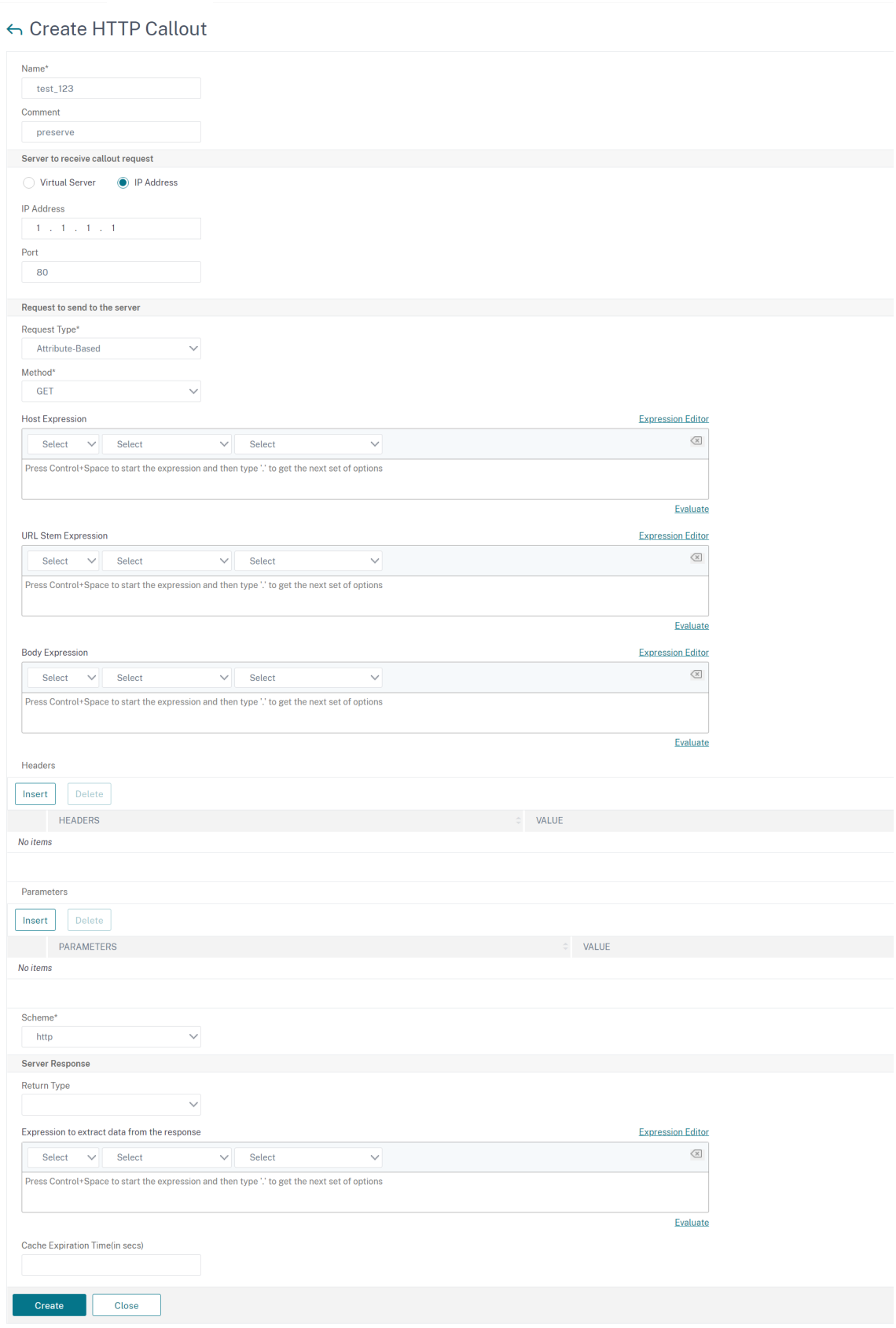


```
Hits: 0
Undef Hits: 0
Done
>

<!--NeedCopy-->
```

構成ユーティリティを使用して **HTTP** コールアウトを構成するには

1. **AppExpert** > **HTTP** コールアウトに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**HTTP** コールアウトの作成] ダイアログボックスで、HTTP コールアウトのパラメータを設定します。パラメータの説明については、チェックボックスの上にマウスカーソルを合わせます。
4. [**Create**] をクリックしてから、[**Close**] をクリックします。



設定の確認

October 7, 2021

HTTP コールアウトが正しく機能するには、すべての HTTP コールアウトパラメータとコールアウトに関連付けられたエンティティが正しく設定されている必要があります。Citrix ADC アプライアンスは HTTP コールアウトパラメータの妥当性をチェックしませんが、バインドされたエンティティ（HTTP コールアウトの送信先となるサーバーまたは仮想サーバー）の状態を示します。次の表に、アイコンとそのアイコンが表示される条件を示します。

アイコン	意味
	HTTP コールアウトエージェントをホストするサーバーの状態、または HTTP コールアウトが送信されるロードバランシング、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーが UP になります。
	HTTP コールアウトエージェントをホストするサーバーの状態、または HTTP コールアウトが送信されるロードバランシング、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーの状態は OUT OF SERVICE です。
	HTTP コールアウトエージェントをホストするサーバーの状態、または HTTP コールアウトが送信されるロードバランシング、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーの状態は DOWN です。

表 1. HTTP コールアウトにバインドされたエンティティの状態を示すアイコン

HTTP コールアウトが正しく機能するには、アイコンが常に緑色になっている必要があります。アイコンが緑色でない場合は、HTTP コールアウトの送信先であるコールアウトサーバーまたは仮想サーバーの状態を確認します。アイコンが緑色であっても、HTTP コールアウトが期待どおりに動作しない場合は、コールアウトに設定されているパラメータを確認します。

また、HTTP コールアウトが呼び出されたポリシーと一致するテスト要求を送信し、ポリシーと HTTP コールアウトのヒットカウンタをチェックし、Citrix ADC アプライアンスがクライアントに送信する応答を確認することで、構成を確認することもできます。

注：HTTP コールアウトは、再帰的に再帰的に自分自身を呼び出すことがあります。この場合、ヒットカウンタは、アプライアンスによって生成されたコールアウトごとに 2 つのカウントずつ増加します。hits カウンタが正しい値を表示するには、HTTP コールアウトを 2 回目に呼び出さないように設定する必要があります。HTTP コールアウトの

再帰を回避する方法の詳細については、「HTTP コールアウトの再帰を回避する」を参照してください。

HTTP コールアウトのヒットカウンタを表示するには

1. **AppExpert** > **HTTP Callouts** に移動します。
2. 詳細ウィンドウで、ヒットカウンタを表示する HTTP コールアウトをクリックし、[詳細] 領域にヒットを表示します。

HTTP コールアウトの呼び出し

October 7, 2021

HTTP コールアウトを設定したら、デフォルトの構文ポリシー規則にSYS.HTTP_CALLOUT(<name>) 式を含めることによって、コールアウトを呼び出します。この式では、<name> は呼び出す HTTP コールアウトの名前です。

コールアウト式とともにデフォルトの構文式演算子を使用して、応答を処理し、適切なアクションを実行できます。HTTP コールアウトエージェントからの応答の戻り値の型によって、応答で使用できる演算子のセットが決まります。分析する応答の一部がテキストの場合は、テキスト演算子を使用して応答を分析できます。たとえば、<string> 次の例のように、CONTAINS () 演算子を使用して、レスポンスの指定した部分に特定の文字列が含まれているかどうかを確認できます。

```
1 SYS.HTTP_CALLOUT(mycallout).contains(”Good IP address”)
2 <!--NeedCopy-->
```

応答側ポリシーで前述の式を使用する場合は、適切な応答側アクションを設定できます。

同様に、評価する応答の一部が数値である場合は、GT（int）などの数値演算子を使用できます。応答にブール値が含まれている場合は、ブール演算子を使用できます。

注：HTTP コールアウトは、再帰的に自分自身を呼び出すことができます。HTTP コールアウトの再帰を回避するには、HTTP コールアウト式と、再帰を防止するデフォルトの構文式を組み合わせます。HTTP コールアウトの再帰を回避する方法については、「HTTP コールアウトの再帰を回避する」を参照してください。

以前に生成されたコールアウトを評価した後にコールアウトを呼び出すポリシーを設定することで、HTTP コールアウトをカスケードすることもできます。このシナリオでは、あるポリシーがコールアウトを呼び出した後、Citrix ADC アプライアンスがコールアウトをコールアウトサーバーに送信する前にコールアウトを解析するときに、2 番目のポリシーセットでコールアウトを評価し、追加のコールアウトを呼び出すことができます。このコールアウトは、3 番目のポリシーセットで評価できます。このような実装については、次の例で説明します。

まず、myCallout1 という HTTP コールアウトを設定してから、応答側ポリシー Pol1 を設定して myCallout1 を呼び出すことができます。次に、2 番目の HTTP コールアウト myCallout2、レスポンダーポリシー Pol2 を構成でき

ます。Pol2 を設定して、myCallout1 を評価し、myCallout2 を呼び出します。両方のレスポンダーポリシーをグローバルにバインドします。

HTTP コールアウトの再帰を避けるために、myCallout1 には「Request1」という一意のカスタム HTTP ヘッダーが設定されています。Pol1 は、HTTP コールアウトの再帰を避けるように設定されており、デフォルトの構文式

```
HTTP.REQ.HEADER(”Request1”).EQ(”Callout Request”).NOT.
<!--NeedCopy-->
```

Pol2 は同じデフォルトの構文式を使用しますが、.NOT 演算子は除外されるため、Citrix ADC アプライアンスが myCallout1 を解析するときにポリシーが評価します。myCallout2 は「Request2」と呼ばれる独自のヘッダーを識別し、Pol2 には myCallout2 が再帰的に呼び出されないようにするためのデフォルトの構文式が含まれています。

例:

```
> add policy httpCallout myCallout1

Done

> set policy httpCallout myCallout1 -IPAddress 10.102.3.95 -port 80 -
    returnType TEXT -hostExpr
 ””10.102.3.95”” -urlStemExpr ””/cgi-bin/check_clnt_from_database.pl””
     -headers Request1
(”Callout Request”) -parameters cip(CLIENT.IP.SRC) -resultExpr ”HTTP.
    RES.BODY(100)”

Done

> add responder policy Pol1 ”HTTP.REQ.HEADER(”Request1”).EQ(”Callout
    Request”).NOT &&
SYS.HTTP_CALLOUT(myCallout1).CONTAINS(”IP Matched”)” RESET

Done

> bind responder global Pol1 100 END -type OVERRIDE

Done

> add policy httpCallout myCallout2

Done

```



```
24 > set policy httpCallout myCallout2 -IPAddress 10.102.3.96 -port 80 -
      returnType TEXT -hostExpr
25 ””10.102.3.96”” -urlStemExpr ””/cgi-bin/
      check_clnt_location_from_database.pl”” -headers Request2
26 (”Callout Request”) -parameters cip(CLIENT.IP.SRC) -resultExpr ”HTTP.
      RES.BODY(200)”
27
28  Done
29
30 > add responder policy Pol2 ”HTTP.REQ.HEADER(”Request2”).EQ(”Callout
      Request”).NOT &&
31  HTTP.REQ.HEADER(”Request1”).EQ(”Callout Request”) && SYS.HTTP_CALLOUT(
       myCallout2).CONTAINS
32 (”APAC”)” RESET
33
34  Done
35
36 > bind responder global Pol2 110 END -type OVERRIDE
37
38 Done
39 <!--NeedCopy-->
```

HTTP コールアウトの再帰の回避

October 7, 2021

Citrix ADC アプライアンスは HTTP コールアウト要求の有効性をチェックしませんが、要求を HTTP コールアウトエージェントに送信する前に一度解析します。この解析により、アプライアンスはコールアウト要求を他の着信要求として処理できます。これにより、コールアウト要求に対して機能するように、いくつかの便利な Citrix ADC 機能（統合キャッシュなど）を設定できます。

ただし、この解析中に HTTP コールアウト要求は同じポリシーを選択できるため、再帰的にそれ自体を呼び出すことができます。アプライアンスは再帰呼び出しを検出し、未定義（UNDEF）状態を生成します。ただし、再帰呼び出しにより、ポリシーと HTTP コールアウト選択カウンタは、それぞれ 1 カウントではなく、それぞれ 2 カウントずつ増加します。

コールアウトがコールアウト自体を呼び出さないようにするには、HTTP コールアウトリクエストの一意な特性を少なくとも 1 つ指定し、コールアウトを呼び出すポリシールールによってこの特性を持つすべてのリクエストが処理されないようにする必要があります。そのためには、ポリシールールに別の既定の構文式を含めます。式は、コールアウト`SYS.HTTP_CALLOUT(<name>)` 式が評価される前に評価されるように、式の前に指定する必要があります。次に例を示します：


```
<Expression that prevents callout recursion> OR SYS.HTTP_CALLOUT(<name
    >)
<!--NeedCopy-->
```

この方法でポリシールールを構成すると、アプライアンスが要求を生成して解析すると、複合ルールが FALSE と評価され、コールアウトが 2 回目に生成されず、選択カウンタが正しくインクリメントされます。

HTTP コールアウト要求に一意の特性を割り当てる方法の 1 つは、コールアウトを設定するときに一意のカスタム HTTP ヘッダーを含めることです。以下は、「myCallout」と呼ばれる HTTP コールアウトの例です。コールアウトは、クライアントの IP アドレスがブラックリストに登録された IP アドレスのデータベースに存在するかどうかをチェックする HTTP 要求を生成します。吹き出しには「Request」というカスタムヘッダーが含まれており、この値は「Callout Request」の値に設定されます。グローバルにバインドされたレスポンダーポリシー「Pol1」は HTTP コールアウトを呼び出しますが、Request ヘッダーがこの値に設定されているすべてのリクエストを除外するため、myCallout の 2 回目の呼び出しが防止されます。2 番目の呼び出しを防ぐ式は HTTP.REQ.HEADER (「リクエスト」) .EQ (「コールアウト要求」) です。

例:

```
> add policy httpCallout myCallout
 Done

> set policy httpCallout myCallout -IPAddress 10.102.3.95 -port 80 -
    returnType TEXT -hostExpr ""10.102.3.95"" -urlStemExpr ""/cgi-bin/
    check_clnt_from_database.pl"" -headers Request("Callout Request") -
    parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"
Done

> add responder policy Pol1 "HTTP.REQ.HEADER("Request").EQ("Callout
    Request").NOT && SYS.HTTP_CALLOUT(myCallout).CONTAINS("IP Matched")"
     RESET
Done

> bind responder global Pol1 100 END -type OVERRIDE
Done
<!--NeedCopy-->
```

> 注:
> リクエスト URL に HTTP コールアウト用に設定された stem 式が含まれているかどうかを確認する式を設定することもできます。ソリューションを実装するには、HTTP コールアウトエージェントが HTTP コールアウトにのみ応答し、アプライアンスを経由する他の要求には応答しないようにします。HTTP コールアウトエージェントが他のクライアント要求を処理するアプリケーションまたは Web サーバーである場合、このような式

> により、アプライアンスはこれらのクライアント要求を処理できなくなります。代わりに、前述のように一意のカスタムヘッダーを使用します。

HTTP コールアウト応答のキャッシュ

October 7, 2021

コールアウト使用時のパフォーマンスを向上させるために、統合キャッシュ機能を使用してコールアウト応答をキャッシュできます。応答は、calloutContentGroup という名前の統合キャッシュコンテンツグループに、指定された期間保存されます。

注記: コールアウト応答をキャッシュするには、統合キャッシュ機能が有効になっていることを確認してください。

コマンドラインインターフェイスを使用してキャッシュ期間を設定するには

コマンドプロンプトで入力します。

```
set policy httpCallout <name> -cacheForSecs <secs>
```

例:

```
1 > set httpcallout httpcallout1 -cacheForSecs 120
2 <!--NeedCopy-->
```

構成ユーティリティを使用してキャッシュ期間を設定するには

1. **AppExpert > HTTP** コールアウトに移動します。
2. 詳細ペインで、キャッシュ期間を設定する HTTP コールアウトを選択し、[開く] をクリックします。
3. [**HTTP** コールアウトの設定] ダイアログボックスで、[キャッシュの有効期限] を指定します。
4. 正しい期間が入力されていることを確認し、[**OK**] をクリックします。

ユースケース**:IP** ブラックリストを使用したクライアントのフィルタリング

October 7, 2021

HTTP コールアウトは、管理者がブラックリストに載っているクライアントからの要求をブロックするために使用できます。クライアントのリストには、一般に知られているブラックリスト、組織で管理するブラックリスト、またはその両方の組み合わせを使用できます。

Citrix ADC アプライアンスは、事前設定されたブラックリストに対してクライアントの IP アドレスをチェックし、IP アドレスがブラックリストに登録されている場合はトランザクションをブロックします。IP アドレスがリストにない場合、アプライアンスはトランザクションを処理します。

この設定を実装するには、次のタスクを実行する必要があります。

1. Citrix ADC アプライアンスでレスポンダーを有効にします。
2. Citrix ADC アプライアンスで HTTP コールアウトを作成し、外部サーバーの詳細およびその他の必須パラメータを使用して設定します。
3. 応答側ポリシーを設定して、HTTP コールアウトへの応答を分析し、ポリシーをグローバルにバインドします。
4. リモートサーバに HTTP コールアウトエージェントを作成します。

レスポンダの有効化

応答側を使用するには、応答側を有効にする必要があります。

GUI を使用してレスポンダを有効にするには

1. レスポンダーライセンスがインストールされていることを確認します。
2. 構成ユーティリティで、[AppExpert] を展開し、[レスポンダー] を右クリックして、[レスポンダー機能を有効にする] をクリックします。

Citrix ADC アプライアンスでの **HTTP** コールアウトの作成

次の表に示すパラメータ設定を使用して、HTTP コールアウト HTTP_Callout を作成します。HTTP コールアウトの作成の詳細については、「HTTP コールアウト PDF の設定 」を参照してください。

レスポンダーポリシーを構成し、グローバルにバインドする

HTTP コールアウトを設定したら、コールアウト設定を確認し、コールアウトを呼び出すようにレスポンダーポリシーを設定します。[
Policies] サブノードでレスポンダーポリシーを作成し、
レスポンダーポリシーマネージャーを使用してグローバルにバインドできますが、このデモでは、
レスポンダーポリシーマネージャーを使用してレスポンダーポリシーを作成し、ポリシーをグローバルにバインドします。

レスポンダーポリシーを作成し、**usin** によってグローバルにバインドするには

1. [**AppExpert**] > [応答者] に移動します。
2. 詳細ペインの [ポリシーマネージャ] で、[ポリシーマネージャ] をクリックします。
3. [レスポンダーポリシーマネージャー] ダイアログボックスで、[グローバルに上書き] をクリックします。

4. [ポリシーの挿入] をクリックし、[ポリシー名] の [新しいポリシー] をクリックします。
5. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] に「ポリシー応答 **1**」と入力します。
   b) [アクション**]** で、[リセット**]** を選択します。
   c) [未定義の結果のアクション**]** で、[グローバルな未定義の結果のアクション] を選択します。
   d) [式**]** に、次の既定の構文式を入力します。

```
1 ”HTTP.REQ.HEADER(”Request”).EQ(”Callout Request”).NOT && SYS.
     HTTP_CALLOUT(HTTP_Callout).CONTAINS(”IP Matched”)”
2 <!--NeedCopy-->
```

   e) [**Create**] をクリックしてから、[**Close**] をクリックします。
6. [変更を適用] をクリックし、[閉じる] をクリックします。

リモートサーバでの **HTTP** コールアウトエージェントの作成

ここで、Citrix ADC アプライアンスからコールアウト要求を受信し、適切に応答する HTTP コールアウトエージェントをリモートコールアウトサーバー上に作成する必要があります。HTTP コールアウトエージェントは、展開ごとに異なるスクリプトです。データベースの種類やサポートされるスクリプト言語など、サーバーの仕様を念頭に置いて記述する必要があります。

次に、指定した IP アドレスが IP ブラックリストの一部であるかどうかを検証するコールアウトエージェントの例を示します。エージェントは Perl スクリプト言語で書かれており、MYSQL データベースを使用しています。

次の CGI スクリプトは、コールアウトサーバー上の指定された IP アドレスをチェックします。

```
1  #!/usr/bin/perl -w
2  print ”Content-type: text/html\n\n”;
3          use DBI();
4          use CGI qw(:standard);
5  #Take the Client IP address from the request query
6          my $ip_to_check = param('cip');
7  # Where a MYSQL database is running
8          my $dsn = 'DBI:mysql:BAD_CLIENT:localhost';
9  # Database username to connect with
10         my $db_user_name =  ‘dbuser’ ;
11 # Database password to connect with
12         my $db_password = 'dbpassword';
13         my ($id, $password);
```



```
# Connecting to the database
        my $dbh = DBI->connect($dsn, $db_user_name, $db_password);
        my $sth = $dbh->prepare(qq{
  select * from bad_clnt  }
 );
        $sth->execute();
        while (my ($ip_in_database) = $sth->fetchrow_array()) {

        chomp($ip_in_database);
# Check for IP match
        if ($ip_in_database eq $ip_to_check) {

              print "\n IP Matched\n";
                                                    $sth->finish();
                                               exit;
                 }

        }

       print "\n IP Failed\n";
       $sth->finish();
       exit;
<!--NeedCopy-->
```

ユースケース：コンテンツを動的に取得および更新するための **ESI** サポート

October 7, 2021

エッジサイドインクルード (ESI) は、エッジレベルの動的 Web コンテンツアセンブリのマークアップ言語です。これは、ネットワークエッジで集約、組み立て、および配信できるキャッシュ可能およびキャッシュ不可能な Web ページコンポーネントを記述するための簡単なマークアップ言語を定義することによって、動的な Web ベースのアプリケーションを高速化するのに役立ちます。Citrix ADC アプライアンスで HTTP コールアウトを使用すると、ESI コンストラクトを読み取り、コンテンツを動的に集約またはアセンブルできます。

この設定を実装するには、次のタスクを実行する必要があります。

1. Citrix ADC アプライアンスで書き換えを有効にします。
2. アプライアンスで HTTP コールアウトを作成し、外部サーバーの詳細およびその他の必須パラメータを使用して設定します。
3. ESI コンテンツをコールアウト応答本文に置き換えるように書き換えアクションを設定します。
4. リライトポリシーを設定してアクションが実行される条件を指定し、リライトポリシーをグローバルにバインドします。

書き換えの有効化

Citrix ADC アプライアンスでリライトを使用する前に、リライトを有効にする必要があります。次の手順では、書き換え機能を有効にする手順について説明します。

GUI を使用して書き換えを有効にするには

1. 書き換えライセンスがインストールされていることを確認します。
2. 構成ユーティリティで、AppExpert を展開し、[書き換え] を右クリックし、[書き換え機能を有効にする] をクリックします。

Citrix ADC アプライアンスでの **HTTP** コールアウトの作成

HTTP コールアウトの作成の詳細については、「HTTP コールアウトの設定」を参照してください。
パラメータ値の詳細については、http-Callout-2 pdf のパラメータと値を参照してください 。

書き換えアクションの設定

書き換えアクション Action-Rewrite-1 を作成して、ESI コンテンツをコールアウト応答本文に置き換えます。次の表に示すパラメータ設定を使用します。

表 2. アクション書き換え 1 のパラメータと値

パラメーター	値
名前	Action-Rewrite-1
種類	置換
ターゲットテキスト参照を選択する式	“HTTP.RES.BODY(500).AFTER_STR (\” <example>\”).BEFORE_STR (\”</example>\”)”
置換テキストの文字列式	“SYS.HTTP_CALLOUT(HTTP-Callout-2)”

構成ユーティリティを使用して書き換えアクションを構成するには

1. **AppExpert** > 書き換え > アクションに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [書き換えアクションの作成] ダイアログボックスの [名前] に「**Action-Rewrite-1** 」と入力します。
4. 「タイプ」で「置き換え」を選択します。
5. 「式」で、ターゲットテキスト参照を選択するには、次のデフォルトの構文式を入力します。


```
1 ”HTTP.RES.BODY(500).AFTER_STR(”<example>”).BEFORE_STR(”<example>”)
    ”
2 <!--NeedCopy-->
```

6. [置換テキストの文字列式] に、次の文字列式を入力します。

```
1 ”SYS.HTTP_CALLOUT(HTTP-Callout-2)”
2 <!--NeedCopy-->
```

7. [**Create**] をクリックしてから、[**Close**] をクリックします。

書き換えポリシーの作成とグローバルなバインド

次の表に示すパラメータ設定を使用して、書き換えポリシー Policy-Rewrite-1 を作成します。
[ポリシー] サブノードで書き換えポリシーを作成し、
書き換えポリシーマネージャを使用してグローバルにバインドできます。または、
書き換えポリシーマネージャを使用して、これらのタスクを同時に実行することもできます。このデモでは、
書き換えポリシーマネージャを使用して両方のタスクを実行します。

表 3. ポリシーリライト 1 のパラメータと値

パラメーター	値
名前	Policy-Rewrite-1
操作（アクション）	Action_Rewrite-1
未定義の結果アクション	-Global undefined-result action-
式	“HTTP.REQ.HEADER(“Name”).CONTAINS (“Callout”).NOT”

構成ユーティリティを使用して書き換えポリシーを構成し、グローバルにバインドするには

1. [**AppExpert**] > [書き換え] に移動します。
2. 詳細ペインの [ポリシーマネージャー] で、[ポリシーマネージャーの書き換え] をクリックします。
3. [ポリシーマネージャを書き換え] ダイアログボックスで、[グローバルに上書き] をクリックします。
4. [ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。
5. [書き換えポリシーの作成] ダイアログボックスで、次の操作を行います。1
   . [名前] に「Policy-Rewrite-1」と入力します。

a)「アクション」で、「アクション-書き換え-1」を選択します。
b) [未定義の結果のアクション] で、[グローバルな未定義の結果のアクション] を選択します。
c) [式] に、次の既定の構文式を入力します。

```
"HTTP.REQ.HEADER("Name").CONTAINS("Callout").NOT"
<!--NeedCopy-->
```

a) [**Create**] をクリックしてから、[**Close**] をクリックします。

6. [変更を適用] をクリックし、[閉じる] をクリックします。

ユースケース**:** アクセス制御と認証

October 7, 2021

セキュリティの高いゾーンでは、クライアントがリソースにアクセスする前に、ユーザーを外部から認証する必要があります。Citrix ADC アプライアンスでは、HTTP コールアウトを使用して、指定された資格情報を評価してユーザーを外部認証できます。この例では、クライアントはリクエストの HTTP ヘッダーを介してユーザー名とパスワードを送信することを前提としています。ただし、URL または HTTP 本文から同じ情報を取得できます。

この設定を実装するには、次のタスクを実行する必要があります。

1. Citrix ADC アプライアンスでレスポンダー機能を有効にします。
2. アプライアンスで HTTP コールアウトを作成し、外部サーバーの詳細およびその他の必須パラメータを使用して設定します。
3. レスポンダーポリシーを設定してレスポンダーを分析し、ポリシーをグローバルにバインドします。
4. リモートサーバ上にコールアウトエージェントを作成します。

レスポンダーの有効化

応答側機能は、Citrix ADC アプライアンスで使用する前に有効にする必要があります。

構成ユーティリティを使用してレスポンダーを有効にするには

1. レスポンダーライセンスがインストールされていることを確認します。
2. 構成ユーティリティで、[AppExpert] を展開し、[レスポンダー] を右クリックし、[レスポンダー機能を有効にする] をクリックします。

Citrix ADC アプライアンスでの **HTTP** コールアウトの作成

次の表に示すパラメータ設定を使用して、HTTP コールアウトである HTTP-Callout-3 を作成します。HTTP コールアウトの作成の詳細については、「HTTP コールアウトの設定」を参照してください。

表 1. HTTP コールアウト 3 のパラメータと値

パラメーター	値	名前
名前	Policy-Responder-3	

パラメーター

値

名前

HTTP-Callout-3

コールアウト要求を受信するサーバー:

IP アドレス

10.103.9.95

ポート

80

サーバーに送信する要求:

方法

GET

ホスト式

10.102.3.95

URL ステム式

“/cgi-bin/authenticate.pl”

ヘッダー:

名前

要求

値式

コールアウト要求

パラメータ:

名前

ユーザー名

値式

HTTP.REQ.HEADER(“Username”).VALUE(0)

名前

パスワード

値式

HTTP.REQ.HEADER(“Password”).VALUE(0)

サーバの応答:

戻り値の型

TEXT

レスポンスからデータを抽出する式

HTTP.RES.BODY (100)

レスポンダーポリシーを作成してレスポンダーを分析する

コールアウトサーバーからの応答をチェックし、送信元 IP アドレスがブラックリストに登録されている場合は接続をリセットするレスポンダーポリシー Policy-Responder-3 を作成します。次の表に示すパラメータ設定を使用してポリシーを作成します。[
Policies] サブノードでレスポンダーポリシーを作成し、
レスポンダーポリシーマネージャーを使用してグローバルにバインドできますが、このデモでは、
レスポンダーポリシーマネージャーを使用してレスポンダーポリシーを作成し、ポリシーをグローバルにバインドします。

表 2. Parameters and Values for Policy-Responder-3

パラメーター	値
名前	Policy-Responder-3
操作（アクション）	RESET
未定義の結果-アクション	-Global undefined-result action-
式	“HTTP.REQ.HEADER(\“Request\”).EQ(\“Callout Request\”).NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\“Authentication Failed\”)”

構成ユーティリティを使用してレスポンダーポリシーを作成し、グローバルにバインドするには

1. [**AppExpert**] > [応答者] に移動します。

2. 詳細ペインの [ポリシーマネージャー] で、[レスポンダーポリシーマネージャー] をクリックします。
3. [レスポンダーポリシーの管理] ダイアログボックスで、[グローバルに上書き] をクリックします。
4. [ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。
5. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] に「ポリシーレスポンダー-3」と入力します。
   b) [アクション] で、[リセット] を選択します。
   c)「未定義結果アクション」で、「グローバル未定義結果アクション」を選択します。
   d) [式] テキストボックスに、次のように入力します。

```
1 ”HTTP.REQ.HEADER(”Request”).EQ(”Callout Request”).NOT && SYS.
     HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(”Authentication Failed”)”
2 <!--NeedCopy-->
```

   a) **Create**、**Close** の順にクリックします。
6. [変更を適用] をクリックし、[閉じる] をクリックします。

リモートサーバでの **HTTP** コールアウトエージェントの作成

ここで、リモートコールアウトサーバーに HTTP コールアウトエージェントを作成する必要があります。HTTP コールアウトエージェントは、Citrix ADC アプライアンスからコールアウト要求を受信し、適切に応答します。コールアウトエージェントは、展開ごとに異なるスクリプトで、データベースの種類やサポートされるスクリプト言語など、サーバーの仕様を念頭に置いて記述する必要があります。

次に、指定されたユーザー名とパスワードが有効かどうかを検証するコールアウトエージェント擬似コードの例を示します。エージェントは、任意のプログラミング言語で実装できます。擬似コードは、コールアウトエージェントを開発するためのガイドラインとしてのみ使用します。プログラムに追加の機能を組み込むことができます。

擬似コードを使用して指定されたユーザー名とパスワードを確認するには

1. 要求で指定されたユーザー名とパスワードを受け入れ、適切にフォーマットします。
2. すべての有効なユーザー名とパスワードを含むデータベースに接続します。
3. 指定された認証情報をデータベースと照合して確認します。
4. HTTP コールアウトで要求される応答をフォーマットします。
5. Citrix ADC アプライアンスにレスポンスを送信します。

ユースケース:OWA ベースのスパムフィルタリング

October 7, 2021

スパムフィルタリングは、既知または信頼できる送信元からのものではない電子メールや、不適切なコンテンツを含む電子メールを動的にブロックする機能です。スパムフィルタリングには、特定の種類のメッセージがスパムであることを示す関連するビジネスロジックが必要です。Citrix ADC アプライアンスが HTTP プロトコルに基づいて Outlook Web Access（OWA）メッセージを処理する場合、HTTP コールアウトを使用してスパムをフィルタリングできます。

HTTP コールアウトを使用して、受信メッセージの任意の部分を抽出し、メッセージが正当なものかスパムかを判断するためのルールで構成された外部コールアウトサーバーをチェックできます。スパムメールの場合、セキュリティ上の理由から、Citrix ADC アプライアンスは送信者にメールがスパムとしてマークされていることを通知しません。

次の例では、電子メールの件名にリストされているさまざまなキーワードについて、非常に基本的なチェックを行います。これらのチェックは、実稼働環境ではより複雑になる可能性があります。

この設定を実装するには、次のタスクを実行する必要があります。

1. Citrix ADC アプライアンスでレスポンダー機能を有効にします。
2. Citrix ADC アプライアンスで HTTP コールアウトを作成し、外部サーバーの詳細およびその他の必須パラメータを使用して設定します。
3. レスポンダーポリシーを作成してレスポンダーを分析し、ポリシーをグローバルにバインドします。
4. リモートサーバ上にコールアウトエージェントを作成します。

レスポンダの有効化

応答側機能は、Citrix ADC アプライアンスで使用する前に有効にする必要があります。

GUI を使用してレスポンダを有効にするには

1. レスポンダーライセンスがインストールされていることを確認します。
2. 構成ユーティリティで、[AppExpert] を展開し、[レスポンダー] を右クリックして、[レスポンダー機能を有効にする] をクリックします。

Citrix ADC アプライアンスでの HTTP コールアウトの作成

次の表に示すパラメータ設定を使用して、HTTP コールアウトである HTTP-Callout-4 を作成します。HTTP コールアウトの作成の詳細については、「HTTP コールアウトの設定」を参照してください。

詳細については、HTTP-Callout-4 pdf のパラメータと値を参照してください 。

レスポンダアクションの作成

レスポンダーアクション「アクション-レスポンダー-4」を作成します。次の表に示すパラメータ設定を使用してアクションを作成します。

パラメーター	値
名前	Action-Responder-4
種類	Respond with
ターゲット	””HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n””

表 2. アクション応答者のパラメータと値-4

構成ユーティリティを使用してレスポンダーアクションを作成するには

1. **AppExpert** > レスポンダー > アクションに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [レスポンダーアクションの作成] ダイアログボックスの **[名前]** に「アクション**-**レスポンダー**-4** 」と入力します。
4. [種類] で、[応答] をクリックします。
5. [ターゲット] に、次のように入力します。

```
1 ””HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By:
     ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\
     nCache-Control: no-cache\r\n\r\n””
2 <!--NeedCopy-->
```

6. [**Create**] をクリックしてから、[**Close**] をクリックします。

HTTP コールアウトを呼び出すレスポンダーポリシーの作成

リクエスト本文をチェックするレスポンダーポリシー Policy-Responder-4 を作成します。本文に「*subject*」という単語が含まれている場合は、HTTP コールアウトを呼び出して電子メールを確認します。次の表に

示すパラメータ設定を使用してポリシーを作成します。[
Policies] サブノードでレスポンダーポリシーを作成し、
レスポンダーポリシーマネージャーを使用してグローバルにバインドできますが、このデモでは、
レスポンダーポリシーマネージャーを使用してレスポンダーポリシーを作成し、グローバルにバインドします。

パラメーター	値
名前	Policy-Responder-4
操作（アクション）	Action-Responder-4
未定義の結果-アクション	-Global undefined-result action-
式	“HTTP.REQ.BODY(1000).CONTAINS(“urn:schemas:httpmail:su && SYS.HTTP_CALLOUT(HTTP-Callout-4)”

構成ユーティリティを使用してレスポンダーポリシーを作成するには

1. [**AppExpert**] > [応答者] に移動します。
2. 詳細ペインの [ポリシーマネージャー] で、[レスポンダーポリシーマネージャー] をクリックします。
3. [レスポンダーポリシーの管理] ダイアログボックスで、[グローバルに上書き] をクリックします。
4. [ポリシーの挿入] をクリックし、[ポリシー名] 列の [新しいポリシー] をクリックします。
5. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] に「ポリシーレスポンダー**-4**」と入力します。
   b) [アクション] で、[アクション**-**応答**-4**] をクリックします。
   c)「未定義の結果アクション」で、「グローバル未定義結果アクション」をクリックします。
   d) [式**]** テキストボックスに、次のように入力します。

```
”HTTP.REQ.BODY(1000).CONTAINS(”urn:schemas:httpmail:subject”)
   && SYS.HTTP_CALLOUT(HTTP-Callout-4)”
<!--NeedCopy-->
```

   e) **Create**、**Close** の順にクリックします。
6. [変更を適用] をクリックし、[閉じる] をクリックします。

リモートサーバでの **HTTP** コールアウトエージェントの作成

これで、リモートコールアウトサーバーに HTTP コールアウトエージェントを作成する必要があります。HTTP コールアウトエージェントは、Citrix ADC アプライアンスからコールアウト要求を受信し、それに応じて応答します。コールアウトエージェントは、展開ごとに異なるスクリプトで、データベースの種類やサポートされるスクリプト言語など、サーバーの仕様を念頭に置いて記述する必要があります。

次の擬似コードは、スパムメールを示すために一般的に理解される単語のリストをチェックするコールアウトエージェントの作成手順を示しています。エージェントは、任意のプログラミング言語で実装できます。擬似コードは、コールアウトエージェントを開発するためのガイドラインとしてのみ使用します。プログラムに追加の機能を組み込むことができます。

疑似コードを使用してスパムメールを特定するには

1. Citrix ADC アプライアンスから提供された電子メールの件名を受け入れます。
2. 電子メールの件名がチェックされるすべての用語を含むデータベースに接続します。
3. メールの件名の単語をスパム単語リストと照合します。
4. HTTP コールアウトで要求される応答をフォーマットします。
5. Citrix ADC アプライアンスにレスポンスを送信します。

ユースケース: 動的コンテンツの切り替え

October 7, 2021

このユースケースは、HTTP コールアウトを使用して、要求の転送先となる負荷分散仮想サーバーの名前を取得することにより、動的なコンテンツスイッチングを提供します。

1. コンテンツスイッチング仮想サーバーを追加します。

```
add cs vserver cs_vserver1 HTTP 10.102.29.196 80
<!--NeedCopy-->
```

2. HTTP コールアウトを作成します。

```
add policy httpCallout http_callout1
<!--NeedCopy-->
```

3. HTTP ヘッダー「X-CLIENT-IP」にクライアント IP アドレスを含む要求からの負荷分散仮想サーバーの名前で応答するように HTTP コールアウトを設定します。

```
> set policy httpCallout http_callout1 -IPAddress 10.217.14.23 -
    port 80 -returnType TEXT -hostExpr ””www.get-lbvip.com”” -
    urlStemExpr ””/index.html”” -headers X-CLIENT-IP(CLIENT.IP.SRC)
     -resultExpr ”HTTP.RES.BODY(1000).AFTER_STR(”<lbvip>”).
    BEFORE_STR(”<lbvip”)”
```



```
<!--NeedCopy-->
```

4. コールアウト応答を取得するようにコンテンツスイッチングアクションを設定します。

```
add cs action cs_action1 -targetVserverExpr 'SYS.HTTP_CALLOUT(
   http_callout1)'
<!--NeedCopy-->
```

> 注:
>
> 負荷分散仮想サーバをコンテンツスイッチング仮想サーバにバインドして、次の項目を考慮する必要があります。
>
> - コールアウトが解決する負荷分散仮想サーバーの非可用性。
> - 吹き出しの実行から生じる UNDEF 条件。

```
> bind cs vserver cs_vserver1 -lbvserver default_lbvip
<!--NeedCopy-->
```

5. コンテンツスイッチングポリシーを設定します。

```
add cs policy cs_policy1 -rule true -action cs_action1
<!--NeedCopy-->
```

6. コンテンツスイッチングポリシーをコンテンツスイッチング仮想サーバーにバインドします。

```
bind cs vserver cs_vserver1 -policyName cs_policy1 -priority 10
<!--NeedCopy-->
```

パターンセットとデータセット

January 25, 2022

多数の文字列パターンに対する文字列照合操作のポリシー式は、長くて複雑になる傾向があります。このような複雑な式の評価によって消費されるリソースは、処理サイクル、メモリ、および構成サイズの点で重要です。パターンマッチングを使用すると、より単純でリソース消費の少ない式を作成できます。

照合するパターンのタイプに応じて、次のいずれかの機能を使用してパターンマッチングを実装できます。

- パターンセットは、デフォルトの構文ポリシー評価時に文字列マッチングに使用されるインデックス付きパターンの配列です。パターンセットの例: イメージタイプ {svg、bmp、PNG、GIF、tiff、jpg}。
- データセットは、パターンセットの特殊な形式です。これは、型数（整数）、IPv4 アドレス、または IPv6 アドレスのパターンの配列です。

patsetとdatasetの違いは、datasetでは境界条件を比較する点です。たとえば、入力文字列が 1.1.1.11 で、1.1.1.1 パターンがpatsetと IPv4 タイプのデータセットにバインドされていると仮定すると、リクエストに IP アドレスが存在するかどうかを確認するようにpatsetとデータセットが設定されます。評価後、patsetは入力に 1.1.1.1 が存在するが、dataset評価が false であることを返します。これは、IP アドレスが他の IP アドレスの一部ではない境界チェックインが原因です。つまり、束縛されたパターンの後には整数があってはならないということです。

多くの場合、パターンセットとデータセットのどちらでも使用できます。ただし、数値データまたは IPv4 アドレスと IPv6 アドレスに特定の一致が必要な場合は、データセットを使用する必要があります。

> 注:
>
> パターンセットとデータセットは、デフォルトの構文ポリシーでのみ使用できます。

パターンセットまたはデータセットを使用するには、まずパターンセットまたはデータセットを作成し、パターンをバインドします。次に、パケット内の文字列を比較するポリシーを設定する場合は、適切な演算子を使用して、パターンセットまたはデータセットの名前を引数として渡します。

パターンセットとデータセットでの文字列マッチングの仕組み

October 7, 2021

パターンセットまたはデータセットにはパターンのセットが含まれており、各パターンには一意のインデックスが割り当てられます。ポリシーがパケットに適用されると、式によって評価される文字列が識別され、演算子は、一致する文字列が見つかるか、すべてのパターンが比較されるまで、その文字列をパターンセットまたはデータセットで定義されたパターンと比較します。次に、その関数に応じて、演算子は、一致するパターンが見つかったかどうかを示すブール値、または文字列に一致するパターンのインデックスを返します。

注記: このトピックでは、パターンセットの作業について説明します。データセットは同じように動作します。パターンセットとデータセットの唯一の違いは、セットで定義されたパターンのタイプです。

パターンが文字列マッチングにどのように使用できるかを理解するために、次のユースケースを考えてみましょう。

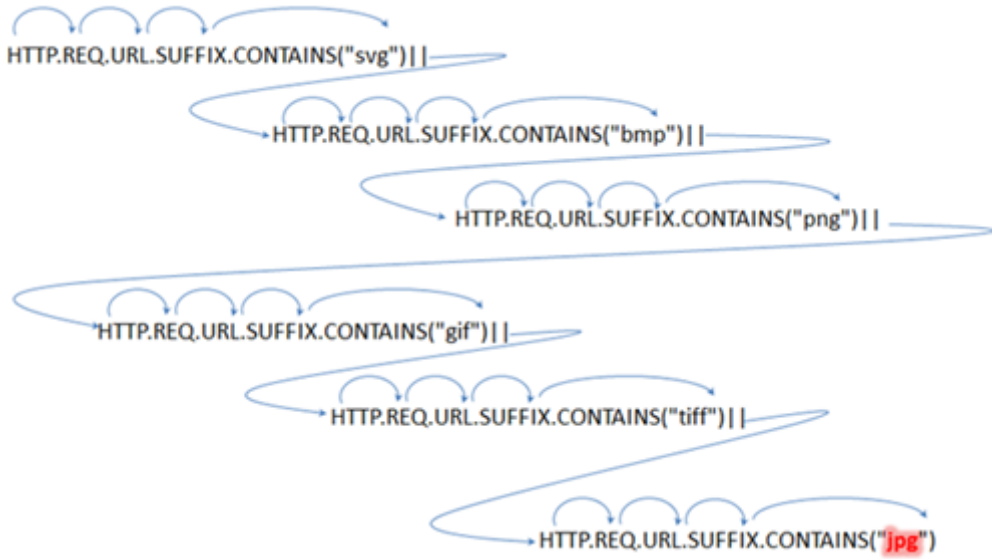
URL サフィックス（ターゲットテキスト）に画像ファイル拡張子が含まれているかどうかを調べます。パターンセットを使用しない場合は、次のように複雑な式を定義する必要があります。

```
1 HTTP.REQ.URL.SUFFIX.CONTAINS("svg") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("
    bmp") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("png") ¦¦
```

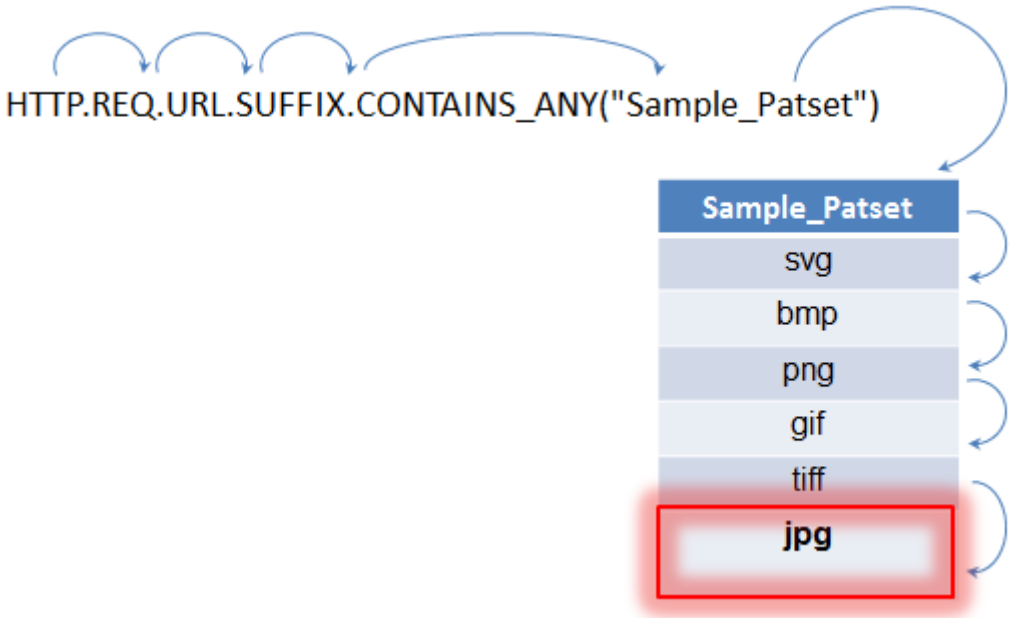


```
2 HTTP.REQ.URL.SUFFIX.CONTAINS(”gif”) ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS(”
    tiff”) ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS(”jpg”)
3 <!--NeedCopy-->
```

URL のサフィックスが「jpg」で、上記の複合式がある場合、Citrix ADC アプライアンスは複合式全体を順番に反復処理して、要求が jpg イメージを参照しているかどうかを判断する必要があります。次の図は、プロセスの手順を示しています。



複合式に数百のサブ式が含まれている場合、上記のプロセスはリソースを大量に消費します。次の図に示すように、パターンセットを呼び出す式が適しています。



上記のようなポリシー評価中、演算子（CONTAINS_ANY）は、一致が見つかるまで、リクエストで識別された文字

列とパターンセットで定義されたパターンを比較します。Sample_Patset 式を使用すると、6 つのサブ式による複数の反復が 1 つに削減されます。

複数の OR 演算で文字列マッチングを実行する複合式を設定する必要がなくなるため、パターンセットまたはデータセットは構成を簡素化し、要求と応答の処理を高速化します。

パターンセットの設定

October 7, 2021

パターンセットを設定するには、パターンとして機能する文字列を指定する必要があります。これらの各パターンに一意のインデックス値を手動で割り当てることも、インデックス値を自動的に割り当てることもできます。

注記: パターンセットでは大文字と小文字が区別されます (大文字と小文字を無視する式を指定しない限り)。したがって、たとえば、文字列パターン「product1」は、文字列パターン「Product1」と同じではありません。

インデックス値について覚えておくべきポイント**:**

- 同じインデックス値を複数のパターンにバインドすることはできません。
- 自動的に割り当てられるインデックス値は、パターンセット内の既存のパターンの最大インデックス値より 1 つ大きい値です。たとえば、パターンセット内の既存のパターンの最大インデックス値が 104 の場合、次に自動的に割り当てられるインデックス値は 105 です。
- 最初のパターンにインデックスを指定しない場合、インデックス値 1 が自動的にそのパターンに割り当てられます。
- 1 つまたは複数のパターンが削除または変更された場合、インデックス値は自動的には再生成されません。たとえば、セットに 1 ～5 のインデックスがある 5 つのパターンが含まれており、インデックスが 3 のパターンが削除された場合、パターンセット内の他のインデックス値は自動的には再生成されず、1 ～4 の値は生成されません。
- パターンに割り当てることができる最大インデックス値は 4294967290 です。その値がセット内のパターンにすでに割り当てられている場合は、新しく追加されたパターンにインデックス値を手動で割り当てる必要があります。現在使用されている値よりも小さい未使用のインデックス値は、自動的に割り当てることはできません。

コマンドラインインターフェイスを使用してパターンセットを構成するには

コマンドプロンプトで、次の操作を行います。

1. パターンセットを作成します。

```
add policy patset <name>
```

例**:**

```
add policy patset samplepatset
```


1. パターンセットにパターンをバインドします。

```
bind policy patset <name> <string> [-index <positive_integer>][-charset
 ( ASCII | UTF_8 )] [-comment <string>]
```

例:

```
bind policy patset samplepatset product1 -index 1 -comment short description
 about the pattern bound to the pattern set
```

注記: パターンセットにバインドするすべてのパターンに対して、この手順を繰り返します。

1. 設定を確認します。

```
show policy patset <name>
```

構成ユーティリティを使用してパターンセットを構成するには

1. 「**AppExpert**」>「パターンセット」に移動します。
2. 詳細ウィンドウで、[追加] をクリックして [パターンセットの作成] ダイアログボックスを開きます。
3. 「名前」(Name) テキストボックスにパターンセットの名前を指定します。
4. 「パターンの指定」(Specify Pattern) で、最初のパターンを入力し、必要に応じて次のパラメータの値を指定します。
   - [バックスラッシュをエスケープ文字として扱う]: パターンに含める可能性のあるバックスラッシュ文字をエスケープ文字として扱うように指定するには、このチェックボックスをオンにします。
   - インデックス: ユーザによって割り当てられたインデックス値（1 ～4294967290）。
5. 正しい文字を入力したことを確認し、[追加] をクリックします。
6. 手順 4 と 5 を繰り返してパターンを追加し、[作成] をクリックします。

ファイルベースのパターンセットを構成する

Citrix ADC アプライアンスは、ファイルベースのパターンセットをサポートしています。

CLI を使用してファイルベースのパターンセットを設定するには

コマンドプロンプトで、次のコマンドを入力します。

- 新しいパターンセットファイルを Citrix ADC アプライアンスにインポートします。

```
1 import policy patsetfile <src> <name> -delimiter <char> -charset
     <ASCII | UTF_8>
2 <!--NeedCopy-->
```


例:

```
import policy patsetfile local:test.csv clientids_list -
   delimiter ,
<!--NeedCopy-->
```

ローカルデバイス、HTTP サーバー、または FTP サーバーからファイルをインポートできます。ローカルデバイスからファイルを追加するには、ファイルが /var/tmp の場所にある必要があります。

- Citrix ADC アプライアンスの既存のパターンセットファイルを更新します。

```
update policy -patsetfile <patset filename>
<!--NeedCopy-->
```

例:

```
update policy -patsetfile clientids_list
<!--NeedCopy-->
```

- パターンセットファイルをパケットエンジンに追加します。

```
add policy -patsetfile <patset filename>
<!--NeedCopy-->
```

例:

```
add policy -patsetfile clientids_list
<!--NeedCopy-->
```

- パターンセットにパターンをバインドします。

```
add policy patset <patset name> -patsetfile <patset filename>
<!--NeedCopy-->
```

例:


```
add policy patset clientid_patset -patsetfile clientids_list
<!--NeedCopy-->
```

- 設定を確認します。

```
show policy patsetfile clientids_list

Name: clientids_list
Patset Name: clientid_patset
Number of Imported Patterns: 8
Number of Bound Patterns: 8
(All the patterns bound successfully)

Done
<!--NeedCopy-->
```

GUI を使用してファイルベースのパターンセットを構成するには

1. **AppExpert->** パターンセットファイルに移動します。
2. インポート**]** ペインで、[インポート] をクリックします。
3. **[**ポリシーパットセットファイルの構成] ページで、インポートするファイルを選択し、[**OK**] をクリックします。
4. インポートしたファイルを選択し、[追加] をクリックします。
5. **[**ポリシーパットセットファイルの作成] ページで詳細を入力し、[作成] をクリックしてポリシーパターンセットを追加します。

データセットの構成

October 7, 2021

データセットを構成するには、サーバーの文字列をパターンとして指定し、タイプ（数値、IPv4 アドレス、または IPv6 アドレス）を割り当て、データセット範囲を構成する必要があります。パターンに一意のインデックス値を手動で割り当てることも、インデックス値を自動的に割り当てることもできます。データセットは、HTTP または任意の 7 層プロトコルに関連していません。これは、テキストまたは文字列に対してのみ機能します。このような NUM、ULong、IPv4、IPv6、MAC、ダブルなどのデータセットの異なる種類があります。タイプを選択し、指定したタイプに基づいてデータセット範囲を定義できます。

> 注:
>
> ポリシーデータセットでは、大文字と小文字が区別されます（大文字と小文字を無視する式を指定しない限り）。したがって、たとえば、MAC アドレス ff: ff: ff: ff は、MAC アドレス FF: FF: FF: FF: FF と同じではありません。

データセットのインデックス値に適用されるルールは、パターンセットに似ています。インデックス値の詳細については、パターンセットの構成を参照してください。

データセットを構成するには

データセットを構成するには、次の手順を完了する必要があります。

1. ポリシーデータセットの追加
2. ポリシーデータセットへのバインドパターン
3. ポリシー式の追加
4. ポリシー設定の確認

ポリシーデータセットの追加

コマンドプロンプトで、次の操作を行います。

```
add policy dataset <name> <type>
```

例:

```
add policy dataset ds1 ipv4 -comment numbers
```

データセットへのパターンのバインド

コマンドプロンプトで入力します。

```
bind policy dataset <name> <value> [-index <positive_integer>] [-endRange <string>] [-comment <string>]
```

例:

```
bind policy dataset ds1 1.1.1.1 -endRange 1.1.1.10 -comment short description about the pattern bound to the data set
```

> 注:
>
> データセットにバインドするすべてのパターンについて、この手順を繰り返す必要があります。1 つのデータセットにバインドできるパターンの数は最大 5000 個です。
>
> また、データセット範囲は、データセットにバインドされた他の範囲と重複してはいけません。また、データセットにバインドされた単一の値を含めることはできません。重複する範囲を持つデータセットをバインドする

> と、エラーが発生します。

例:

```
add policy dataset ip_set ipv4
 Done
bind policy dataset ip_set 2.2.2.25
 Done
bind policy dataset ip_set 2.2.2.20 -endRange 2.2.2.30
ERROR: The range overlaps an existing range or includes a value bound
   to the dataset.
<!--NeedCopy-->
```

データセットにバインドされた単一の値と等しい場合、またはデータセットにバインドされた範囲の下限値と上限 (下限 <= 値 && 値 <-上限) の間にある値は、データセット内にあると見なされます。

ポリシーデータセットでポリシー式を使用する

コマンドプロンプトで入力します。

```
add policy expression exp1 http.req.body(100).contains_any(”ds1”)
```

ここで、
式は、データセット ds1 にバインドされたパターン（または範囲内のパターン）が、HTTP リクエスト本体の最初の 100 バイトに存在するかどうかをチェックします。

データセット構成の確認

コマンドプロンプトで入力します。

```
show policy dataset ds1
> show policy dataset ds1
```

例:

```
        Dataset:        ds1
         Type:   IPV4
1)       Bound Dataset Range from: 1.1.1.1       through: 1.1.1.10
          Index:  1
<!--NeedCopy-->
```


構成ユーティリティを使用してデータセットを構成するには

次の手順に従って、ポリシーデータセットを設定します。

1. [**AppExpert] > [**データセット] に移動します。
2. 詳細ペインの [データセット] で、[追加] をクリックします。
3. **[**データセットの構成] ページで、次のパラメータを設定します。

   a) Name：ポリシーデータセットの名前。
   b) タイプ。データセットにバインドする値のタイプ。

   データセットの構成

4. [挿入] をクリックして、特定のタイプのデータセット値をバインドします。

   a) 値。データセットに関連付けられた指定されたタイプの値。
   b) インデックス。データセットのインデックス値。
   c) 終了範囲。データセットエントリ。これは<value>から<end_range>までの範囲です。
   d) コメント。データセットに関する簡単な説明。

   データセットバインディング

5. [挿入して 閉じる] をクリックします。
6. コメントを入力します。
7. **[**作成**]** して **[**閉じる**]** をクリックします。

パターンセットとデータセットの使用

October 7, 2021

パターンセットまたはデータセットを引数として受け取るデフォルトの構文ポリシー式を使用して、文字列の一致操作を実行できます。

使用方法は次のとおりです。

```
<text>.<operator>(”<name>”)
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

- <text> は、パケット内の文字列を識別する式です。例: HTTP.REQ.HEADER(“Host”).
- <operator>は、パターンセットタイプ表 pdf で説明されている演算子の 1 つです。

使用例については、サンプル使用法を参照してください。

使用サンプル

October 7, 2021

式でのパターンセットの使用方法を理解するために、「imagetypes」という名前のパターンセットの例を考えてみましょう。

パターン	インデックス値
svg か	1
bmp	2
PNG	3
gif	4
TIFF	5
jpg	6

表 1. Pattern set “imagetypes”

例 **1:** HTTP リクエストのサフィックスが「imagetypes」パターンセットで定義されているファイル拡張子の 1 つであるかどうかを判断します。

- 式。HTTP.REQ.URL.SUFFIX.EQUALS_ANY(“imagetypes”)
- サンプル **URL**。 `http://www.example.com/homepageicon.jpg`
- 結果。TRUE

例 **2:** HTTP リクエストのサフィックスが「imagetypes」パターンセットで定義されているファイル拡張子の 1 つであるかどうかを判断し、そのパターンのインデックスを返します。

- 式。HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”)
- サンプル **URL**。 `http://www.example.com/mylogo.png`
- 結果。 4 (パターン「gif」のインデックス値)

例 **3:** パターンのインデックス値を使用して、URL サフィックスが指定されたインデックス値の範囲内にあるかどうかを判断します。

- 式。HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(3) && HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“i
- サンプル **URL**。 `http://www.example.com/mylogo.png`
- 結果。TRUE (gif ファイルタイプのインデックス値は 4 です。)

例 **4:** ファイル拡張子の bmp、jpg、および png に対して 1 つのポリシーセットを実装し、gif、tiff、および svg ファイルに対して異なるポリシーセットを実装します。

一致するパターンのインデックスを返す式を使用して、Web アプリケーションのトラフィックサブセットを定義できます。コンテンツスイッチ仮想サーバーのコンテンツスイッチングポリシーでは、次の 2 つの式を使用できます。

- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).LE(3)
- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(4)

変数

October 7, 2021

変数は、トークンの形式で情報を格納する名前付きオブジェクトです。これらのトークンは、内部計算とポリシー処理のために、Citrix ADC アプライアンス上の異なるトランザクション内およびトランザクション間で使用されます。

Citrix ADC アプライアンスは、次のタイプの変数の作成をサポートしています。

- シングルトン変数。ulong および text (最大サイズ) のいずれかの型の値を指定できます。ulong 型は符号なし 64 ビット整数で、テキスト型はバイトのシーケンス、max-size はシーケンスの最大バイト数です。
- 変数をマップします。マップはキーに関連付けられた値を保持します。各キーと値のペアはマップエントリと呼ばれます。各エントリのキーは、マップ内で一意です。マップは次のように指定されます。

  マップ (キーの種類、値の種類、最大値)。

  各項目の意味は次のとおりです。

  - *key_type* は、キーのデータ型です。これは、タイプテキスト（最大サイズ）です。
  - *value_type* は、マップの値のデータ型です。これは、タイプが ulong またはテキスト（最大サイズ）であってもよい。
  - *max-values* は、マップに含めることができるエントリの最大数です。それはタイプ ulong です。

これらの変数の値は、ポリシーアクションで呼び出される必要がある割り当てを使用して設定されます。

> 注: 変数は、高可用性セットアップまたはクラスターではまだサポートされていません。

変数のスコープ

マップ変数またはシングルトン変数は、グローバルスコープを持つことができます。また、シングルトン変数のスコープを 1 つのトランザクションに制限することもできます。

- グローバルスコープ変数-グローバルスコープの変数（デフォルト）は 1 つのインスタンスしか持たず、そのインスタンスは Citrix ADC アプライアンスのすべてのコアおよびクラスタまたは HA 構成のすべてのノードで同じ値を持ちます。グローバル変数の値は、明示的に削除されるまで、有効期限が切れるまで、またはスタンドアロンアプライアンスが再起動されるか、クラスタまたは HA 構成のすべてのノードが再起動されるまで存在します。

- トランザクションスコープ変数 -トランザクションスコープを持つ変数は、Citrix ADC アプライアンスによって処理されるトランザクションごとに、独自の値を持つ個別のインスタンスを持ちます。トランザクション処理が完了すると、トランザクション変数の値が削除されます。

> 注：トランザクションスコープ変数は、Citrix ADC リリース 10.5.e 以降で使用できます。

変数の構成と使用

October 7, 2021

最初に変数を作成し、次に値を代入するか、変数に対して実行する必要のある操作を指定する必要があります。これらの操作を実行した後で、割り当てをポリシーアクションとして使用できます。

注: いったん構成すると、変数の設定を変更したりリセットしたりすることはできません。変数を変更する必要がある場合は、変数と変数へのすべての参照（式と代入）を削除する必要があります。その後、変数を新しい設定で再追加し、参照（式と代入）を再追加することができます。

コマンドラインインターフェイスを使用して変数を構成するには

1. 変数を作成します。

```
add ns variable <name> -type <string> [-scope global] [-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )] [-ifNoValue ( undef | init )] [-init <string>] [-expires <positive_integer>] [-comment <string>]
<!--NeedCopy-->
```

注：コマンド・パラメータの説明については、マニュアル・ページ「man add ns 変数」を参照してください。

例 1:「my_counter」という名前の ulong 変数を作成し、1 に初期化します。

```
add ns variable my_counter  - type ulong -init 1
<!--NeedCopy-->
```

例 2:「ユーザー _ 特権 _ マップ」という名前のマップを作成します。マップには、最大長 15 文字のキーと、最大長 10 文字のテキスト値、最大 10000 エントリが含まれます。

```
add ns variable user_privilege_map -type map(text(15),text(10),10000)
<!--NeedCopy-->
```


注: マップに 10000 個の期限切れのエントリが含まれている場合、新しいキーの割り当ては、最も最近使用されていないエントリの 1 つを再利用します。デフォルトでは、存在しないキーの値を取得しようとする式は、空のテキスト値を初期化します。

値を代入するか、変数に対して実行する操作を指定します。これは、割り当てを作成することによって行われます。

```
1 add ns assignment <name> -variable <expression> [-set <expression> | -
     add <expression> | -sub <expression> | -append <expression> | -clear
     ] [-comment <string>]
2 <!--NeedCopy-->
```

注：変数は、変数セレクタ（$）を使用して参照されます。したがって、
$variable1 は、テキストまたは ulong 変数を参照するために使用されます。同様に、
$variable2[キー式**]** は、マップ変数を参照するために使用されます。

例 1:「inc_my_counter」という名前の割り当てを定義し、自動的に「my_counter」変数に 1 を追加します。

```
1 add ns assignment inc_my_counter -variable $my_counter -add 1
2 <!--NeedCopy-->
```

例 2:「set_user_privilege」という名前の割り当てを定義し、”get_user_privilege」HTTP コールアウトによって返された値を持つクライアントの IP アドレスのエントリを「user_map」変数に追加します。

```
1 add ns assignment set_user_privilege -variable $user_privilege_map[
     client.ip.src.typecast_text_t] -set sys.http.callout(
     get_user_privilege)
2 <!--NeedCopy-->
```

注：そのキーのエントリがすでに存在する場合、値は置き換えられます。それ以外の場合は、キーと値の新しいエントリが追加されます。user_privilege_map の前の宣言に基づいて、マップにすでに 10000 エントリがある場合、最も最近使用されていないエントリの 1 つが新しいキーと値に再利用されます。

1. ポリシーで変数割り当てを呼び出します。

   マップ変数を操作できる 2 つの関数があります。

   - **$name.valueExists(key-expression).** キー式によって選択されたマップ内の値がある場合は true を返します。それ以外の場合は false を返します。この関数は、マップ・エントリが存在する場合は有効期限と LRU 情報を更新しますが、値が存在しない場合は新しいマップ・エントリを作成しません。

- **$name.valueCount.** 現在変数に保持されている値の数を返します。これは、マップ内のエントリの数です。シングルトン変数の場合、変数が初期化されていない場合は 0、そうでない場合は 1 です。

例**:** 圧縮ポリシーを使用して「set_user_privilege」という名前の割り当てを呼び出します。

```
add cmp policy set_user_privilege_pol -rule $user_privilege_map.
    valueExists(client.ip.src.typecast_text_t).not -resAction
    set_user_privilege
<!--NeedCopy-->
```

レスポンス側に **HTTP** ヘッダーを挿入するユースケース

次の例は、シングルトン変数の例を示しています。

テキスト型のシングルトン変数を追加します。この変数は、最大 100 バイトのデータを保持できます。

```
add ns variable http_req_data -type text(100) -scope transaction
<!--NeedCopy-->
```

変数に HTTP リクエストデータを格納するために使用される代入アクションを追加します。

```
add ns assignment set_http_req_data -variable $http_req_data -set http.
    req.body(100)
<!--NeedCopy-->
```

変数から値をフェッチする HTTP ヘッダーを挿入する書き換えアクションを追加します。

```
add rewrite action act_ins_header insert_http_header user_name
    $http_req_data.after_str(”user_name”).before_str(”password”)
<!--NeedCopy-->
```

要求時間に評価する書き換えポリシーを追加し、データを格納するための割り当てアクションを実行します。このポリシーにヒットすると、代入アクションを実行し、データを ns 変数（http_req_data）に格納します。

```
add rewrite policy pol_set_variable true set_http_req_data

bind rewrite global pol_set_variable 10 -type req_dEFAULT
<!--NeedCopy-->
```


応答時間に評価する書き換えポリシーを追加し、応答に HTTP ヘッダーを追加します。

```
add rewrite policy pol_ins_header true act_ins_header

bind rewrite global pol_ins_header 10 -type res_dEFAULT
<!--NeedCopy-->
```

割り当てアクション

Citrix ADC アプライアンスでは、ポリシールールが true と評価されると、ポリシーにバインドされた割り当てアクションがトリガーされます。アクションによって、変数の値が更新されます。この値は後続のポリシールール評価で使用できます。このようにして、同じ変数を更新して、同じ機能内の後続のポリシー評価に使用できます。以前は、関連付けられた割り当てアクションのポリシーが true と評価されたときに、機能内のすべてのポリシーを評価した後にのみ割り当てアクションを実行していました。したがって、割り当てアクションによって設定された変数値は、機能内の後続のポリシールール評価で使用できません。

この機能は、Citrix ADC アプライアンス上のクライアントのアクセスリストを制御するユースケースでよりよく理解できます。アクセス決定は、本体に「BLOCK」または「ALLOW」を含む応答GET /client-access?<client-IP-address> を返すリクエストで、別の Web サービスによって提供されます。HTTP コールアウトは、着信要求に関連付けられているクライアントの IP アドレスを含めるように設定されます。Citrix ADC アプライアンスがクライアントからの要求を受信すると、アプライアンスはコールアウト要求を生成し、コールアウトサーバーに送信します。コールアウトサーバーは、ブラックリストに登録された IP アドレスのデータベースと、クライアントの IP アドレスがデータベースにリストされているかどうかをチェックする HTTP コールアウトエージェントをホストします。HTTP コールアウトエージェントは、コールアウト要求を受信し、クライアントの IP アドレスがリストされているかどうかを確認し、応答を送信します。応答は、本体の「BLOCK」または「ALLOW」とともに、ステータスコード、200、302 です。ステータスコードに基づいて、アプライアンスはポリシー評価を実行します。ポリシー評価が true の場合、割り当てアクションはただちにトリガーされ、action は値を変数に設定します。アプライアンスは、同じモジュール内の後続のポリシー評価にこの変数値を使用し、設定します。

割り当てアクションを構成するためのユースケース

次の手順に従って、割り当てアクションを設定し、後続のポリシーに変数を使用します。

1. アクセス決定は、本体に BLOCK または ALLOW を含む応答を返すリクエストで、別の Web サービスによって提供されます。

   ```
   GET /url-service>/url-allowed?<URL path>
   ```

2. URL のアクセス決定を保持するマップ変数を設定します。

   ```
   add ns variable url_list_map -type 'map(text(1000),text(10),10000)'
   ```


3. Web サービスにアクセス要求を送信する HTTP コールアウトを設定します。

```
add policy httpCallout url_list_callout -vserver url_vs -returnType
TEXT -urlStemExpr '"/url-allowed?" + HTTP.REQ.URL.PATH'-resultExpr '
HTTP.RES.BODY(10)'
```

4. 割り当てアクションを設定して、コールアウトを呼び出してアクセス決定を取得し、URL のマップエントリに割り当てます。

```
add ns assignment client_access_assn -variable '$client_access_map[
CLIENT.IP.SRC.TYPECAST_TEXT_T]'-set SYS.HTTP_CALLOUT(client_access_callout
)
```

5. URL 要求がブロックされた場合に 403 応答を送信するように応答側アクションを設定します。

```
add responder action url_list_block_act respondwith '"HTTP/1.1 403
Forbidden\\r\\n\\r\\n"'
```

6. URL のマップエントリが設定されていない場合は、レスポンダーポリシーを設定します。即時アクション拡張により、このポリシーの評価時にマップエントリの値が設定されます。拡張以前は、すべてのレスポンダーポリシーが評価され、別の Web サービスによって決定が行われるまで、割り当ては行われませんでした。

```
add responder policy url_list_assn_pol '!$url_list_map.VALUEEXISTS(HTTP
.REQ.URL.PATH)'url_list_assn
```

7. マップエントリの値が BLOCK の場合、URL へのアクセスをブロックするレスポンダーポリシーを設定します。即時アクション拡張により、前述のポリシーによって設定されたマップエントリをこのポリシーで使用できます。機能強化以前は、この時点ではマップエントリの設定が解除されます。

```
add responder policy client_access_block_pol '$client_access_map[CLIENT
.IP.SRC.TYPECAST_TEXT_T] == "BLOCK"'client_access_block_act
```

8. 応答側ポリシーを仮想サーバにバインドします。注: 別の仮想サーバー上の HTTP コールアウトに対してポリシーを実行したくないため、ポリシーをグローバルにバインドすることはできません。

```
bind lb vserver vs -policyName client_access_assn_pol -priority 10 -
gotoPriorityExpression NEXT -type REQUEST
bind lb vserver vs -policyName client_access_block_pol -priority 20 -
gotoPriorityExpression END -type REQUEST
```

構成ユーティリティを使用して変数を構成するには

1. **AppExpert > NS** 変数に移動し、変数を作成します。
2. **AppExpert > NS** 割り当てに移動し、変数に値を割り当てます。
3. 割り当てをアクションとして構成する適切な機能領域に移動します。

ユースケース: ユーザー権限のキャッシュ

October 7, 2021

このユースケースでは、外部 Web サービスからユーザー権限（「GOLD」、「SILVER」など）を取得する必要があります。

このユースケースを実現するには、次の操作を実行します

HTTP コールアウトを作成して、外部 Web サービスからユーザ権限を取得します。

```
add policy httpcallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port <port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod ( GET | POST )] [-hostExpr <string>] [-urlStemExpr <string>] [-headers <name(value)> ...] [-parameters <name(value)> ...] [-bodyExpr <string>][-fullReqExpr <string>] [-scheme ( http | https )] [-resultExpr <string>] [-cacheForSecs <secs>] [-comment <string>]

add policy httpcallout get_user_privilege -ipaddress 10.217.193.84 -port 80 -returnType text -httpMethod GET -hostExpr '"/get_user_privilege"' -resultExpr 'http.res.body(5)'
<!--NeedCopy-->
```

権限を変数に格納します。

```
add ns variable <name> -type <string> [-scope ( global | transaction )][-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )][-ifNoValue ( undef | init )] [-init <string>] [-expires <positive_integer>] [-comment <string>]

add ns variable user_privilege_map -type map(text(15),text(10),10000) -expires 1200

add ns assignment set_user_privilege -variable $user_privilege_map[client.ip.src] -set sys.http_callout(get_user_privilege)
<!--NeedCopy-->
```

クライアントの IP アドレスのキャッシュされたエントリがすでに存在するかどうかをチェックするポリシーを作成します。存在しない場合は、HTTP コールアウトを呼び出して、クライアントのマップエントリを設定します。


```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy set_user_privilege_pol -rule $user_privilege_map.
   valueExists(client.ip.src).not -resAction set_user_privilege>
<!--NeedCopy-->
```

クライアントのキャッシュされた特権エントリが「GOLD」である場合に圧縮するポリシーを作成します。

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy compress_if_gold_privilege_pol -rule '
   $user_privilege_map[client.ip.src].eq(”GOLD”)' -resAction compress
<!--NeedCopy-->
```

圧縮ポリシーをグローバルにバインドします。

```
bind cmp global <policyName> [-priority <positive_integer>] [-state (
   ENABLED | DISABLED )] [-gotoPriorityExpression <expression>] [-type
   <type>] [-invoke (<labelType> <labelName>) ]

bind cmp global set_user_privilege_pol -priority 10 NEXT

bind cmp global compress_if_gold_privilege_pol -priority 20 END
<!--NeedCopy-->
```

ユースケース: セッション数の制限

June 1, 2022

このユースケースでは、要件はアクティブなバックエンドセッションの数を制限することです。展開では、各セッションログインの URL にはログインがあり、各セッションログアウトには URL にログアウトがあります。ログインが成功すると、バックエンドは一意の 10 文字の値を持つ sessionid Cookie を設定します。

このユースケースを実現するには、次の操作を実行します。

1. アクティブな各セッションを格納できるマップ変数を作成します。マップのキーはセッション ID です。変数の有効期限は 600 秒 (10 分) に設定されています。


```
> add ns variable session_map -type map(text(10),ulong,100) -
    expires 600
<!--NeedCopy-->
```

2. map 変数に次の代入を作成します。

   - sessionid のエントリを作成し、その値を 1 に設定します (この値は実際には使用されません)。

```
> add ns assignment add_session -variable '$session_map[http.
    req.cookie.value(”sessionid”)]' -set 1
<!--NeedCopy-->
```

   - セッション ID のエントリを解放します。これにより、session_map の値カウントが暗黙的に減少します。

```
> add ns assignment delete_session -variable '$session_map[
    http.req.cookie.value(”sessionid”)]' -clear
<!--NeedCopy-->
```

3. 次のレスポンダーポリシーを作成します。

   - HTTP リクエストにそのセッション ID のマップエントリが存在するかどうかを確認する。add_session 割り当ては、マップエントリが存在しない場合に実行されます。

```
> add responder policy add_session_pol 'http.req.url.contains
    (”example”) || $session_map.valueExists(http.req.cookie.
    value(”abc”))' add_session
<!--NeedCopy-->
```

     注:
     add_session_pol ポリシーの
     valueExists () 関数は、セッションのマップエントリへの参照としてカウントされるため、各リクエストはそのセッションの有効期限タイムアウトをリセットします。10 分経過してもセッションの要求が受信されない場合、セッションのエントリの割り当ては解除されます。

   - セッションがいつログアウトしたかを確認する。delete_session 割り当てが実行されます。

```
add responder policy delete_session_pol ”http.req.url.
    contains(”Logout”)” delete_session
```



```
2   <!--NeedCopy-->
```

- ログイン要求があるかどうか、およびアクティブなセッションの数が 100 を超えているかどうかを確認する。これらの条件が満たされると、セッション数を制限するために、ユーザーはサーバーがビジーであることを示すページにリダイレクトされます。

```
1   add responder action redirect_too_busy redirect "/too_busy.
       html"
2   add responder policy check_login_pol "http.req.url.contains("
       example") && $session_map.valueCount > 100"
       redirect_too_busy
3   <!--NeedCopy-->
```

4. レスポンダーポリシーをグローバルにバインドします。

```
1 bind responder global add_session_pol 30 next
2 bind responder global delete_session_pol 10
3 bind responder global check_login_pol 20
4 <!--NeedCopy-->
```

ポリシーと式

October 7, 2021

以下のトピックでは、Citrix® Citrix ADC® アプライアンスで高度なポリシーを構成するために必要となる概念情報と参照情報を提供します。

Citrix ADC アプライアンスでサポートされているすべての高度なポリシー式については、「ポリシー式」を参照してください。

ポリシーと式の概要	式、ポリシー、アクションの目的、およびさまざまな Citrix ADC アプリケーションがこれらをどのように使用するかについて説明します。
詳細ポリシーの設定	高度なポリシーの構造と、それらを個別に、またはポリシーバンクとして構成する方法について説明します。
高度な式の設定: はじめに	式の構文とセマンティクスについて説明し、式とポリシーの設定方法について簡単に説明します。
高度な式: テキストの評価	テキスト (HTTP POST 要求の本文やユーザー証明書のコンテンツなど) を操作するときに設定する式を記述します。

高度な式: 日付、時刻、および数値の操作	任意のタイプの数値データ（たとえば、URL の長さ、クライアントの IP アドレス、HTTP 要求が送信された日付と時刻）を操作するときに設定する式について説明します。
高度な式:HTTP、TCP、および UDP データの解析	IP アドレスと IPv6 アドレス、MAC アドレス、および HTTP および TCP トラフィックに固有のデータを解析するための式について説明します。
高度な式:SSL 証明書の解析	SSL トラフィックおよびクライアント証明書の式を設定する方法について説明します。たとえば、証明書または証明書発行者の有効期限を取得する方法などです。

高度な式:IP アドレスと MAC アドレス、スループット、VLAN ID	他の章で説明されていないクライアントまたはサーバ関連のデータを操作するために使用できる式について説明します。	型キャストデータ	ある型のデータを別の型に変換するための式を記述する。	Regular Expressions	高度な式の演算子への引数として正規表現を渡す方法について説明します。
クラシックポリシーとエクスプレッションの設定	クラシックポリシーとクラシック式と呼ばれるより単純なポリシーとエクスプレッションの設定方法の詳細を提供する	エクスプレッションリファレンス	クラシックと高度なエクスプレッションの引数のリファレンス	高度なエクスプレッションの概要 Expression and Policies	クイックリファレンス形式とチュートリアルの形式で、独自に使用できるようにカスタマイズできる、クラシックおよび高度な式とポリシーの例
書き換えの高度なポリシーのチュートリアル例	書き換え機能で使用する高度なポリシーの例				
クラシックポリシーのチュートリアル例	アプリケーションファイアウォールや SSL などの Citrix ADC 機能のクラシックポリシーの例				

Apache mod_rewrite ルールの高度なポリシーへの移行	Apache HTTP Server の mod_rewrite エンジンを使用して記述した関数の例。Citrix ADC で書き換えポリシーとレスポンダーポリシーに変換した後の関数の例。

ポリシーと式の概要

October 7, 2021

多くの Citrix ADC 機能では、機能がデータを評価する方法がポリシーによって制御されます。ポリシーは、rule と呼ばれる論理式を使用してデータを評価し、評価に基づいて 1 つ以上のアクションを適用します。または、ポリシーでプロファイルを適用して、複雑なアクションを定義することもできます。

Citrix ADC の一部の機能では、従来のポリシーよりも優れた機能を提供するデフォルトの構文ポリシーが使用されます。新しいリリースの Citrix ADC ソフトウェアに移行し、デフォルトの構文ポリシーを使用する機能に対して従来のポリシーを構成している場合は、ポリシーを高度なポリシーインフラストラクチャに手動で移行する必要があります。

クラシックおよび高度なポリシー

October 7, 2021

> 警告
>
> 従来のポリシー式は、Citrix ADC 12.0 ビルド 56.20 以降では廃止され、代わりに高度なポリシーを使用することをお勧めします。詳細については、「高度なポリシー」を参照してください。

従来のポリシーでは、トラフィックおよびその他のデータの基本特性を評価します。たとえば、従来のポリシーでは、HTTP 要求または応答に特定のタイプのヘッダーまたは URL が含まれているかどうかを識別できます。

高度なポリシーでは、従来のポリシーと同じタイプの評価を実行できます。さらに、高度なポリシーインフラストラクチャ（PI）を使用すると、より多くのデータ（HTTP リクエストの本文など）を分析し、ポリシールールでより多

くの操作（リクエストの本文のデータを HTTP ヘッダーに変換するなど）を構成できます。

ポリシーにアクションまたはプロファイルを割り当てるだけでなく、Citrix ADC 機能に関連する処理の特定のポイントにポリシーをバインドします。バインドポイントは、ポリシーをいつ評価するかを決定する要素の 1 つです。

高度なポリシーを使用する利点

デフォルトの構文ポリシーでは、クラスオブジェクトモデルに基づいて構築された強力な式言語が使用され、さまざまな Citrix ADC 機能の動作を構成する機能を強化するオプションがいくつか用意されています。高度なポリシーインフラストラクチャ (PI) を使用すると、次の操作を実行できます。

- レイヤ 2 ～7 のネットワークトラフィックのきめ細かな分析を実行します。
- HTTP または HTTPS リクエストまたはレスポンスのヘッダーまたは本体の任意の部分を評価します。
- デフォルトレベル、オーバーライド、および仮想サーバーレベルで高度なポリシーインフラストラクチャ (PI) がサポートする複数のバインドポイントにポリシーをバインドします。
- goto 式を使用して、式の評価の結果によって決定される他のポリシーおよびバインドポイントに制御を転送します。
- パターンセット、ポリシーラベル、レート制限識別子、HTTP コールアウトなどの特殊なツールを使用すると、複雑なユースケースに対して効果的にポリシーを設定できます。

また、構成ユーティリティは、高度なポリシーインフラストラクチャ（PI）および式に対する堅牢なグラフィカルユーザーインターフェイスのサポートを拡張し、ネットワークプロトコルに関する知識が限られているユーザーは、ポリシーをすばやく簡単に構成できます。構成ユーティリティには、高度なポリシーのポリシー評価機能も含まれています。この機能を使用すると、高度なポリシーを評価し、コミットする前にその動作をテストできるため、構成エラーのリスクが軽減されます。

詳細ポリシーの基本コンポーネント

詳細ポリシーの特徴を次に示します。

- Name：各ポリシーには一意の名前があります。
- Rule. このルールは、Citrix ADC 機能でトラフィックまたは他のオブジェクトを評価できるようにする論理式です。たとえば、Citrix ADC は、特定の IP アドレスから送信された HTTP リクエストかどうか、または HTTP リクエストのキャッシュ制御ヘッダーに「No-Cache」値があるかどうかを決定するルールを有効にできます。

高度なポリシーでは、SSL VPN クライアントの従来の式を除き、クラシックポリシーで使用できるすべての式を使用できます。さらに、高度なポリシーを使用すると、より複雑な式を構成できます。

- バインディング。Citrix ADC が必要なときにポリシーを呼び出せるようにするには、ポリシーを 1 つまたは複数のバインドポイントに関連付けます（バインドします）。

ポリシーは、グローバルにバインドすることも、仮想サーバにバインドすることもできます。詳細については、「ポリシーバインディングについて」を参照してください。

- 関連付けられたアクション。アクションは、ポリシーとは別のエンティティです。ポリシー評価では、最終的に Citrix ADC がアクションを実行します。

たとえば、統合キャッシュのポリシーは、.png ファイルまたは.jpeg ファイルに対する HTTP 要求を識別できます。このポリシーに関連付けるアクションによって、これらのタイプのリクエストに対する応答がキャッシュから提供されることが決まります。

一部の機能では、プロファイルと呼ばれるより複雑な命令セットの一部としてアクションを構成します。

Citrix ADC の機能によってポリシーが使用される方法

Citrix ADC は、動作のポリシーに依存するさまざまな機能をサポートしています。次の表は、Citrix ADC の機能でポリシーを使用する方法をまとめたものです。

[フィーチャ名]	ポリシーの種類	機能でのポリシーの使用方法
システム	クラシック	認証機能の場合、ポリシーにはさまざまな認証方法の認証スキームが含まれます。たとえば、LDAP および証明書ベースの認証スキームを設定できます。監査機能でもポリシーを設定します。
DNS	詳細設定	要求に対して DNS 解決を実行する方法を決定する。
SSL	クラシックとアドバンス	暗号化機能を適用するタイミングを決定し、証明書情報をクリアテキストに追加する。エンドツーエンドのセキュリティを提供するために、メッセージが復号化された後、SSL 機能はクリアテキストを再暗号化し、SSL を使用して Web サーバと通信します。
圧縮	クラシックとアドバンス	圧縮されるトラフィックのタイプを決定する。
統合キャッシング	詳細設定	HTTP 応答がキャッシュ可能かどうかを判断します。
レスポンダー	詳細設定	レスポンダー関数の動作を構成する。

[フィーチャ名]	ポリシーの種類	機能でのポリシーの使用方法
保護機能	クラシック	フィルタ、SureConnect、およびプライオリティキューイング機能の動作を構成する。
コンテンツスイッチ	クラシックとアドバンス	着信要求の特性に基づいて、応答を処理するサーバまたはサーバグループを判断する。要求特性には、デバイスタイプ、言語、Cookie、HTTP メソッド、コンテンツタイプ、および関連するキャッシュサーバが含まれます。
AAA-トラフィック管理	クラシック。例外: トラフィックポリシーは Advanced Policy Infrastructure（PI）のみをサポートし、認可ポリシーは、クラシックおよびアドバンスポリシーインフラストラクチャ（PI）	ユーザーがログインしてセッションを確立する前に、クライアント側のセキュリティをチェックする。シングルサインオン（SSO）が必要かどうかを決定するトラフィックポリシーでは、デフォルトの構文のみを使用します。承認ポリシーは、アプライアンスを介してイントラネットリソースにアクセスするユーザーとグループを承認します。
キャッシュリダイレクト	クラシック	応答がキャッシュから提供されるか、オリジンサーバーから提供されるかを判断する。

[フィーチャ名]	ポリシーの種類	機能でのポリシーの使用方法
リライト	詳細設定	提供する前に変更する HTTP データを識別します。ポリシーは、データを変更するためのルールを提供します。たとえば、HTTP データを変更して、新しいホームページ、新しいサーバー、または受信要求のアドレスに基づいて選択したサーバーに要求をリダイレクトできます。また、セキュリティ上の理由から、応答のサーバー情報をマスクするようにデータを変更できます。URL トランスフォーマー関数は、URL を変換する必要があるかどうかを評価するために、HTTP トランザクションおよびテキストファイル内の URL を識別します。
アプリケーションファイアウォール	クラシックとアドバンス	ファイアウォールを通過するトラフィックおよび許可するべきでないデータの特性を識別する。
Citrix Gateway、クライアントレスアクセス機能	詳細設定	Citrix Gateway を使用して一般的な Web アクセスの書き換えルールを定義する。
Citrix Gateway	クラシック	Citrix Gateway が認証、承認、監査、およびその他の機能をどのように実行するかを決定します。

アクションとプロファイルについて

ポリシー自体はデータに対してアクションを実行しません。ポリシーは、トラフィックを評価するための読み取り専用ロジックを提供します。ポリシー評価に基づいて操作を実行する機能を有効にするには、アクションまたはプロファイルを設定し、それらをポリシーに関連付けます。

注記**:** アクションとプロファイルは、特定の機能に固有です。フィーチャへのアクションとプロファイルの割り当てについては、個々の機能のマニュアルを参照してください。

アクションについて

アクションは、ポリシー内の式の評価に応じて、Citrix ADC が実行する手順です。たとえば、ポリシーの式がリクエスト内の特定の送信元 IP アドレスと一致する場合、このポリシーに関連付けられているアクションによって、接続が許可されるかどうかが決まります。

Citrix ADC が実行できる操作の種類は、機能によって異なります。たとえば、Rewrite では、アクションはリクエスト内のテキストを置き換えたり、リクエストの宛先 URL を変更したりできます。統合キャッシュでは、アクションによって HTTP 応答がキャッシュから提供されるか、オリジンサーバーから提供されるかが決まります。

Citrix ADC の一部の機能では、アクションは事前に定義されており、その他の機能では設定可能です。場合によっては（Rewrite など）、関連付けられたポリシールールの設定に使用するのと同じタイプの式を使用してアクションを設定します。

プロファイルについて

Citrix ADC の一部の機能では、プロファイル、またはアクションとプロファイルの両方をポリシーに関連付けることができます。プロファイルは、機能が複雑な機能を実行できるようにする設定の集まりです。たとえば、アプリケーションファイアウォールでは、XML データのプロファイルは、データの不正な XML 構文や SQL インジェクションの証拠など、複数のスクリーニング操作を実行できます。

特定の機能でのアクションとプロファイルの使用

次の表は、Citrix ADC の各機能におけるアクションとプロファイルの使用方法についてまとめたものです。表は網羅的ではありません。機能に対するアクションとプロファイルの具体的な使用方法の詳細については、機能のマニュアルを参照してください。

機能	アクションの使用	プロファイルの使用
Application firewall	プロファイルと同義語	すべてのアプリケーションファイアウォール機能は、プロファイルを使用して、パターンベースの学習を含む複雑な動作を定義します。これらのプロファイルをポリシーに追加します。

機能	アクションの使用	プロファイルの使用
Citrix Gateway	Citrix Gateway の機能では、「事前認証」というアクションが使用されます。許可アクションと拒否アクションを使用します。これらのアクションをプロファイルに追加します。「承認」。許可アクションと拒否アクションを使用します。これらのアクションをポリシーに追加します。TCP 圧縮。さまざまなアクションを使用します。これらのアクションをポリシーに追加します。	プロファイルを使用する機能は、事前認証、セッション、トラフィック、およびクライアントレスアクセスです。プロファイルを設定したら、ポリシーに追加します。
リライト	URL 書き換えアクションを設定し、ポリシーに追加します。	使用しません。
統合キャッシング	ポリシー内でキャッシュおよび無効化アクションを構成する	使用しません。
AAA-トラフィック管理	認証タイプを選択するか、許可アクションを [許可] または [拒否] に設定するか、監査を SYSLOG または NSLOG に設定します。	デフォルトのタイムアウトおよび認可アクションを使用してセッションプロファイルを設定できます。
保護機能	フィルター、圧縮、レスポンダー、および SureConnect の機能について、ポリシー内でアクションを構成します。	使用しません。
SSL	SSL ポリシー内でアクションを設定します。	使用しません。
システム	アクションは暗黙的に行われます。認証機能では、[許可] または [拒否] のいずれかになります。[監査] の場合、[監査オン] または [監査オフ] になります。	使用しません。
DNS	アクションは暗黙的に行われます。これは、ドロップパケットまたは DNS サーバーの場所のいずれかです。	使用しません。

機能	アクションの使用	プロファイルの使用
SSL オフロード	アクションは暗黙的に行われます。これは、SSL 仮想サーバーまたはサービスに関連付けるポリシーに基づいています。	使用しません。
圧縮	データに適用する圧縮のタイプを決定する	使用しません。
コンテンツスイッチ	アクションは暗黙的に行われます。要求がポリシーと一致する場合、要求はポリシーに関連付けられた仮想サーバに送信されます。	使用しません。
キャッシュリダイレクト	アクションは暗黙的に行われます。リクエストがポリシーと一致する場合、リクエストはオリジンサーバーに送信されます。	使用しません。

ポリシーバインディングについて

ポリシーは、ポリシーを呼び出せるエンティティに関連付けられるか、またはエンティティにバインドされます。たとえば、すべての仮想サーバーに適用される要求時間の評価にポリシーをバインドできます。特定のバインドポイントにバインドされたポリシーのコレクションは、ポリシーバンクを構成します。

次に、ポリシーのさまざまなタイプのバインドポイントの概要を示します。

- グローバル要求時間。ポリシーは、要求時に機能のすべてのコンポーネントで使用できます。
- グローバル応答時間。ポリシーは、応答時に機能のすべてのコンポーネントで使用できます。
- 要求時間、仮想サーバ固有。

ポリシーは、特定の仮想サーバーの要求時間処理にバインドできます。たとえば、要求時間ポリシーをキャッシュリダイレクト仮想サーバーにバインドして、特定の要求がキャッシュの負荷分散仮想サーバーに転送され、他の要求がオリジンの負荷分散仮想サーバーに送信されるようにできます。

- 応答時間、仮想サーバ固有。ポリシーは、特定の仮想サーバの応答時間処理にバインドすることもできます。
- ユーザー定義のポリシーラベル。Advanced Policy Infrastructure（PI）の場合、ポリシーラベルを定義し、ポリシーラベルの下に一連の関連ポリシーを収集することにより、ポリシー（ポリシーバンク）のカスタムグループを設定できます。
- その他のバインドポイント。追加のバインドポイントを使用できるかどうかは、ポリシーのタイプ（クラシックポリシーまたは高度なポリシー）と、関連する Citrix ADC 機能の詳細によって異なります。たとえば、Citrix Gateway 用に構成する従来のポリシーには、ユーザーとグループのバインドポイントがあります。

高度なポリシーバインドの詳細については、「高度なポリシーを使用するポリシーをバインドする 」および「仮想サーバーのポリシーバンクを構成する」を参照してください。クラシックポリシーバインディングの詳細については、「クラシックポリシーを構成する」を参照してください。

ポリシーの評価順序について

クラシックポリシーの場合、ポリシーグループおよびグループ内のポリシーは、次の項目に応じて特定の順序で評価されます。

- ポリシーのバインドポイント。たとえば、ポリシーが仮想サーバーの要求時間処理にバインドされているか、グローバル応答時間処理にバインドされているかなどです。たとえば、Citrix ADC は、要求時にすべての要求時クラシックポリシーを評価してから、仮想サーバー固有のポリシーを評価します。
- ポリシーのプライオリティレベル。評価プロセスの各ポイントについて、ポリシーに割り当てられた優先度レベルによって、同じバインドポイントを共有する他のポリシーと比較して評価の順序が決まります。たとえば、Citrix ADC が要求時間、仮想サーバー固有のポリシーのバンクを評価する場合、優先度の最も低い値に割り当てられたポリシーから開始します。従来のポリシーでは、すべてのバインドポイントでプライオリティレベルが一意である必要があります。

高度なポリシーでは、従来のポリシーと同様に、Citrix ADC は、処理全体の特定のポイントでポリシーのグループ化（バンク）を選択します。次に、高度なポリシーの基本的なグループ化（バンク）の評価の順序を示します。

1. 要求時間のグローバル上書き
2. 要求時間、仮想サーバ固有（仮想サーバごとに 1 つのバインドポイント）
3. 要求時間のグローバルデフォルト
4. 応答時間のグローバル上書き
5. 応答時間の仮想サーバ固有
6. レスポンスタイムグローバルデフォルト

ただし、前述のどのポリシーのバンクでも、評価の順序は従来の政策よりも柔軟です。ポリシーバンク内では、優先度レベルに関係なく評価される次のポリシーをポイントできます。また、他のバインドポイントおよびユーザー定義のポリシーバンクに属するポリシーバンクを呼び出すことができます。

トラフィックフローに基づく評価順序

トラフィックが Citrix ADC を通過し、さまざまな機能によって処理されると、各機能によってポリシー評価が実行されます。ポリシーがトラフィックと一致すると、Citrix ADC はアクションを保存し、データが Citrix ADC から出るまで処理を続けます。その時点で、Citrix ADC は通常、一致するすべてのアクションを適用します。統合キャッシュは、最終的な Cache または NoCache アクションのみを適用しますが、例外です。

一部のポリシーは、他のポリシーの結果に影響を与えます。次に例を示します。

- 応答が統合キャッシュから提供される場合、Citrix ADC の他の機能によっては、応答または応答を開始した要求が処理されません。
- コンテンツフィルタ機能によって応答の提供が妨げられる場合、後続の機能は応答を評価しません。

アプリケーションファイアウォールが着信要求を拒否した場合、他の機能では要求を処理できません。

クラシックおよび高度なポリシー式

October 7, 2021

ポリシーの最も基本的なコンポーネントの 1 つは、そのルールです。ポリシールールは、ポリシーがトラフィックを分析できるようにする論理式です。ポリシーの機能のほとんどは、その式から派生しています。

式は、トラフィックまたはその他のデータの特性を 1 つ以上のパラメータおよび値と一致させます。たとえば、式を使用すると、Citrix ADC で次の処理を実行できます。

- 要求に証明書が含まれているかどうかを判断します。
- TCP 要求を送信したクライアントの IP アドレスを確認します。
- HTTP リクエストに含まれるデータ（一般的なスプレッドシートやワープロアプリケーションなど）を特定します。
- HTTP リクエストの長さを計算します。

クラシックエクスプレッションについて

クラシック式を使用すると、データの基本的な特性を評価できます。これらは、文字列の一致やその他の操作を実行する構造化された構文を持っています。

古典的なエクスプレッションの簡単な例を次に示します。

- HTTP レスポンスには、特定のタイプのキャッシュコントロールヘッダーが含まれています。

res.http.header キャッシュコントロールにパブリックが含まれています

- HTTP 応答には、画像データが含まれます。

res.http.header コンテンツタイプに画像が含まれています

- SSL 要求には証明書が含まれています。

クライアント証明書が存在する

高度なポリシー式について

デフォルトの構文ポリシーを使用する機能では、高度な式も使用されます。詳細ポリシーを使用する機能の詳細については、表「Citrix ADC 機能、ポリシータイプ、ポリシーの使用状況」を参照してください。

高度なポリシー式には、他にもいくつかの用途があります。ポリシールールで詳細式を設定する以外に、次の状況で詳細
式を構成します。

- 統合キャッシュ:

  高度なポリシー式を使用して、統合キャッシュ内のコンテンツグループのセレクタを構成します。

- 負荷分散:

  高度なポリシー式を使用して、負荷分散に TOKEN メソッドを使用する負荷分散仮想サーバーのトークン抽出を構成します。

- 書き換え:

  書き換えアクションを構成するには、高度なポリシー式を使用します。

- レートベースのポリシー:

  詳細ポリシー式を使用して、さまざまなサーバへのトラフィックレートを制御するポリシーを設定するときに、制限セレクタを設定します。

次に、高度なポリシー式の簡単な例を示します。

- HTTP リクエスト URL に含まれる文字数は 500 文字以下です。

```
http.req.url.length \<= 500
```

- HTTP 要求には、500 文字未満の cookie が含まれています。

```
http.req.cookie.length \< 500
```

- HTTP リクエスト URL には、特定のテキスト文字列が含まれています。

```
http.req.url.contains(".html")
```

NSPEPI ツールを使用したポリシー式の変換

October 16, 2022

> 注:
>
> NSPEPI と事前設定チェックツールは、公開 GitHub からダウンロードできます。詳細については、Github NEPEPI ページと GitHub 事前設定ページを参照して、ツールをダウンロードする詳細な手順を確認してください。お客様には、GitHub で利用可能なツールを使用して、最も完全で最新のバージョンを使用することをお勧めします。

従来のポリシーベースの機能は、NetScaler 12.0 ビルド 56.20 以降で廃止されました。別の方法として、高度なポリシーインフラストラクチャを使用することをお勧めします。この取り組みの一環として、Citrix ADC 12.1 ビルド 56.20 以降にアップグレードする場合、クラシックポリシーベースの機能を、対応する非推奨の機能に置き換える必要があります。また、クラシックポリシーと式を高度なポリシーと式に変換する必要があります。また、すべての新しい Citrix ADC 機能は、高度なポリシーインフラストラクチャのみをサポートします。

この`nspepi`ツールは、次のことを実行できます。

1. クラシックポリシー式を高度なポリシー式に変換します。
2. 特定のクラシックポリシーとそのエンティティバインディングを高度なポリシーおよびバインディングに変換します。
3. いくつかの非推奨の機能を、対応する非推奨の機能に変換します。
4. 従来のフィルタコマンドを高度なフィルタコマンドに変換します。

> 注:
>
> nspepiツールが ns.conf 構成ファイルを正常に変換すると、変換されたファイルは、接頭辞「new_」が付いた新しいファイルとして表示されます。変換された設定ファイルにエラーまたは警告がある場合は、変換プロセスの一環として手動で修正する必要があります。変換したら、テスト環境でファイルをテストし、それを使用して実際の ns.conf 構成ファイルを置き換える必要があります。テスト後、新しく変換または修正された ns.conf 構成ファイル用にアプライアンスを再起動する必要があります。

クラシックポリシーまたは式のみをサポートする機能は廃止され、対応する非推奨の機能に置き換えることができます。

> 注:
>
> 旧バージョンのnspepiツールに関する情報は PDF 形式で入手できます。詳細については、「PDF 12.1-51.16 より前の nspepi ツールを使用した従来のポリシー変換 」を参照してください。

変換の警告とエラーファイル

変換にツールを使用する前に、注意すべき警告はほとんどありません。

1. すべての警告とエラーがコンソールに出力されます。設定ファイルが保存されている場所に警告ファイルが作成されます。
2. 警告およびエラーファイルの名前は、入力ファイルと同じですが、ファイル名に接頭辞「warn_」が追加されています。式の変換中 (-e を使用する場合)、警告はカレントディレクトリに「warn_expr」という名前で表示されます。

> 注:
>
> このファイルは、日付/時刻スタンプとログレベルを含む標準のログファイル形式です。ツールが複数回実行されるので、ファイルの以前のインスタンスは「.1」、「.2」などの接尾辞で保持されます。最大で 10 個のインスタンスが保持されます。

変換されたファイル形式

設定ファイル (「-f」を使用) を変換する場合、変換されたファイルは、同じ名前でプレフィックス「new_」の入力設定ファイルが存在するディレクトリと同じディレクトリに置かれます。

nspepi 変換ツールによって処理されるコマンドまたは機能

自動変換処理中に処理されるコマンドは次のとおりです。

- 次のクラシックポリシーとその式は、高度なポリシーと式に変換されます。変換には、エンティティバインディングとグローバルバインディングが含まれます。

1. add appfw policy
2. add cmp policy
3. add cr policy
4. add cs policy
5. add tm sessionPolicy
6. add filter action
7. add filter policy
8. 負荷分散、コンテンツスイッチング、キャッシュリダイレクト、およびグローバルへのポリシーバインディングをフィルタします。

> 注:
>
> ただし、「add tm SessionPolicy」の場合、詳細ポリシーでグローバルオーバーライドにバインドすることはできません。

- 「Add lb 仮想サーバー」で設定されたルールパラメーターは、クラシック式から高度な式に変換されます。
- 「追加 ns HttpProfile」または「ns httpProfile を設定」コマンドで設定された SPDY パラメータが「-http2 ENABLED」に変更されます。
- 名前付き式(「ポリシー式の追加」コマンド)。各クラシック名前付きポリシー式は、接頭辞として「nspepi_adv_」が設定された、対応する Advanced 名前付き式に変換されます。さらに、変換されたクラシック式での名前付き式の使用は、対応する Advanced 名前付き式に変更されます。さらに、すべての名前付き式には 2 つの名前付き式があります。1 つは Classic で、もう 1 つは Advanced です (以下を参照)。
- トンネルトラフィックポリシー変換がサポートされています。
- CMP、CR、およびトンネルでの組み込みのクラシックポリシーバインディングの処理。
- Patclass フィーチャは Pat セットフィーチャに変換されます。
- 「書き換えアクションを追加」コマンドの「-pattern」パラメータは、「-search」パラメータを使用するように変換されます。
- SYS.EVAL_CLASSIC_EXPR は、同等の非推奨の高度な式に変換されます。これらの式は、高度な定義式を使用できるすべてのコマンドで見ることができます。
- 高度な定義式の Q および S プレフィックスは、非推奨の同等の高度な式に変換されます。これらの式は、高度な定義式を使用できるすべてのコマンドで見ることができます。

例:

```
add policy expression classic_expr ns_true
Converts to:
```



```
add policy expression classic_expr ns_true
add policy expression nspepi_adv_classic_expr TRUE
<!--NeedCopy-->
```

- 「set cmp パラメータ」コマンドで設定された policyType パラメータが削除されます。デフォルトでは、ポリシータイプは「Advanced」です。

従来のフィルタコマンドを高度なフィルタコマンドに変換する

このnspepiツールは、追加、バインドなどの従来のフィルタ操作に基づくコマンドを高度なフィルタコマンドに変換できます。

ただし、nepepi ツールは次のフィルタコマンドをサポートしていません。

1. add filter action <action Name> FORWARD <service name>
2. add filter action <action name> ADD prebody
3. add filter action <action name> ADD postbody

> 注:
>
> 1. ns.conf に既存の書き換え機能またはレスポンダ機能があり、それらのポリシーがGOTO式ENDまたはUSER_INVOCATION_RESULTとしてグローバルにバインドされ、バインドタイプがREQ_XまたはRES_Xである場合、バインドフィルタコマンドは部分的に変換され、コメントアウトされます。手動作業を行うために警告が表示されます。
> 2. 既存の書き換え機能またはレスポンダー機能があり、そのポリシーがGOTO - ENDまたはUSER_INVOCATION_RESULTを使用して HTTPS タイプの仮想サーバー (負荷分散、コンテンツスイッチング、キャッシュリダイレクトなど) にバインドされている場合、ツールはバインドフィルターコマンドを部分的に変換してからコメントアウトします。手動作業を行うために警告が表示されます。

例

以下は入力例です。

```
add lb vserver v1 http 1.1.1.1 80 -persistenceType NONE -cltTimeout
    9000
add cs vserver csv1 HTTP 1.1.1.2 80 -cltTimeout 180 -persistenceType
    NONE
add cr vserver crv1 HTTP 1.1.1.3 80 -cacheType FORWARD
add service svc1 1.1.1.4 http 80
add filter action fact_add add 'header:value'
add filter action fact_variable add 'H1:%%HTTP.TRANSID%%'
add filter action fact_prebody add prebody
add filter action fact_error_act1 ERRORCODE 200 ”<HTML>Good URL</HTML>”
```



```
add filter action fact_forward_act1 FORWARD svc1
add filter policy fpol_add_res -rule ns_true -resAction fact_add
add filter policy fpol_error_res -rule ns_true -resAction
    fact_error_act1
add filter policy fpol_error_req -rule ns_true -reqAction
    fact_error_act1
add filter policy fpol_add_req -rule ns_true -reqAction fact_add
add filter policy fpol_variable_req -rule ns_true -reqAction
    fact_variable
add filter policy fpol_variable_res -rule ns_true -resAction
    fact_variable
add filter policy fpol_prebody_req -rule ns_true -reqAction
    fact_prebody
add filter policy fpol_prebody_res -rule ns_true -resAction
    fact_prebody
add filter policy fpol_forward_req -rule ns_true -reqAction
    fact_forward_act1
bind lb vserver v1 -policyName fpol_add_res
bind lb vserver v1 -policyName fpol_add_req
bind lb vserver v1 -policyName fpol_error_res
bind lb vserver v1 -policyName fpol_error_req
bind lb vserver v1 -policyName fpol_variable_res
bind lb vserver v1 -policyName fpol_variable_req
bind lb vserver v1 -policyName fpol_forward_req
bind cs vserver csv1 -policyName fpol_add_req
bind cs vserver csv1 -policyName fpol_add_res
bind cs vserver csv1 -policyName fpol_error_res
bind cs vserver csv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_add_req
bind cr vserver crv1 -policyName fpol_add_res
bind cr vserver crv1 -policyName fpol_error_res
bind cr vserver crv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_forward_req
bind filter global fpol_add_req
bind filter global fpol_add_res
bind filter global fpol_error_req
bind filter global fpol_error_res
bind filter global fpol_variable_req
bind filter global fpol_variable_res
bind filter global fpol_variable_res -state DISABLED
bind filter global fpol_prebody_req
bind filter global fpol_forward_req
After conversion, warning/error messages will be displayed for manual
    effort.
Warning files:
```



```
cat warn_<input file name>:
2019-11-07 17:13:34,724: ERROR - Conversion of [add filter action
    fact_prebody add prebody] not supported in this tool.
2019-11-07 17:13:34,739: ERROR - Conversion of [add filter action
    fact_forward_act1 FORWARD svc1] not supported in this tool.
2019-11-07 17:13:38,042: ERROR - Conversion of [add filter policy
    fpol_prebody_req -rule ns_true -reqAction fact_prebody] not
    supported in this tool.
2019-11-07 17:13:38,497: ERROR - Conversion of [add filter policy
    fpol_prebody_res -rule ns_true -resAction fact_prebody] not
    supported in this tool.
2019-11-07 17:13:39,035: ERROR - Conversion of [add filter policy
    fpol_forward_req -rule ns_true -reqAction fact_forward_act1] not
    supported in this tool.
2019-11-07 17:13:39,060: WARNING - Following bind command is commented
    out because state is disabled. Advanced expressions only have a
    fixed ordering of the types of bindings without interleaving, except
     that global bindings are allowed before all other bindings and
    after all bindings. If you have global bindings in the middle of non
    -global bindings or any other interleaving then you will need to
    reorder all your bindings for that feature and direction. Refer to
    nspepi documentation. If command is required please take a backup
    because comments will not be saved in ns.conf after triggering 'save
     ns config': bind filter global fpol_variable_res -state DISABLED


<!--NeedCopy-->
```

次に、出力例を示します。変換されたすべてのコマンドにはコメントが付きます。

```
cat new_<input file name>
add rewrite action fact_add insert_http_header header ””value””
add filter action fact_prebody add prebody
add filter action fact_forward_act1 FORWARD svc1
add filter policy fpol_prebody_req -rule ns_true -reqAction
    fact_prebody
add filter policy fpol_prebody_res -rule ns_true -resAction
    fact_prebody
add filter policy fpol_forward_req -rule ns_true -reqAction
    fact_forward_act1
bind lb vserver v1 -policyName fpol_forward_req
bind cr vserver crv1 -policyName fpol_forward_req
#bind filter global fpol_variable_res -state DISABLED
```



```
bind filter global fpol_prebody_req
bind filter global fpol_forward_req
add rewrite action nspepi_adv_fact_variable insert_http_header H1 HTTP.RES.TXID
add rewrite action fact_variable insert_http_header H1 HTTP.REQ.TXID
add responder action fact_error_act1 respondwith "HTTP.REQ.VERSION.APPEND(" 200 OK\r
nConnection: close\r
nContent-Length: 21\r\n\r
n<HTML>Good URL</HTML>")"
add rewrite action nspepi_adv_fact_error_act1 replace_http_res "HTTP.REQ.VERSION.APPEND(" 200 OK\r
nConnection: close\r
nContent-Length: 21\r\n\r
n<HTML>Good URL</HTML>")"
add rewrite policy fpol_add_res TRUE fact_add
add rewrite policy fpol_error_res TRUE nspepi_adv_fact_error_act1
add responder policy fpol_error_req TRUE fact_error_act1
add rewrite policy fpol_add_req TRUE fact_add
add rewrite policy fpol_variable_req TRUE fact_variable
add rewrite policy fpol_variable_res TRUE nspepi_adv_fact_variable
set cmp parameter -policyType ADVANCED
bind rewrite global fpol_add_req 100 NEXT -type REQ_DEFAULT
bind rewrite global fpol_variable_req 200 NEXT -type REQ_DEFAULT
bind rewrite global fpol_add_res 100 NEXT -type RES_DEFAULT
bind rewrite global fpol_error_res 200 NEXT -type RES_DEFAULT
bind rewrite global fpol_variable_res 300 NEXT -type RES_DEFAULT
bind responder global fpol_error_req 100 END -type REQ_DEFAULT
bind lb vserver v1 -policyName fpol_add_res -type RESPONSE -priority 100 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_res -type RESPONSE -priority 200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_res -type RESPONSE -priority 300 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_add_req -type REQUEST -priority 100 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_req -type REQUEST -priority 200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_req -type REQUEST -priority 100 -gotoPriorityExpression END
bind cs vserver csv1 -policyName fpol_add_req -type REQUEST -priority 100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_add_res -type RESPONSE -priority 100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_res -type RESPONSE -
```



```
    priority 200 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END
bind cr vserver crv1 -policyName fpol_add_req -type REQUEST -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_res -type RESPONSE -
    priority 200 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END

<!--NeedCopy-->
```

既存の書き換えまたはレスポンダーポリシーバインディングに **goto** 式 **END** または **USE_INNVOCATION** がある場合、従来のフィルタコマンドを高度な機能コマンドに変換します

この変換では、書き換えポリシーが 1 つ以上の仮想サーバーにバインドされ、サーバーに END または USE_INVOCATION_RESULT がある場合、ツールはコマンドをコメントアウトします。

例

次に、入力コマンドの例を示します。

```
COPY
add filter policy fpol1 -rule ns_true -resAction reset
add filter policy fpol2 -rule ns_true -reqAction reset
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
bind rewrite policylabel pl pol1 1
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
    -invoke policylabel pl
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
    REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
```



```
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind lb vserver v1_http -policyName fpol1
bind cs vserver csv1_http -policyName fpol1
bind lb vserver v2_http -policyName fpol2
bind cs vserver csv2_http -policyName fpol2
bind cr vserver crv2_http -policyName fpol2
bind filter global fpol1 -priority 100
bind filter global fpol2 -priority 100
<!--NeedCopy-->
```

次に、出力コマンドの例を示します。

```
COPY
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
bind rewrite policylabel pl pol1 1
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
add rewrite policy fpol1 TRUE RESET
add responder policy fpol2 TRUE RESET
#bind lb vserver v1_http -policyName fpol1 -type RESPONSE
#bind cs vserver csv1_http -policyName fpol1 -type RESPONSE
#bind rewrite global fpol1 100 -type RES_DEFAULT
#bind lb vserver v2_http -policyName fpol2 -type REQUEST
#bind cs vserver csv2_http -policyName fpol2 -type REQUEST
#bind cr vserver crv2_http -policyName fpol2 -type REQUEST
#bind responder global fpol2 100 -type REQ_DEFAULT
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
    -invoke policylabel pl
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
    REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
```

```
22 bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
      gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
23 bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
      gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST-
24
25 <!--NeedCopy-->
```

nspepi 変換ツールで処理されないコマンドまたは機能

次に、自動変換処理の一部として処理されないコマンドをいくつか示します。

- 一部のバインディングは、グローバルバインドポイントと非グローバルバインドポイント間、ユーザーとグループ間、および異なるエンティティへのバインディング間で優先度のインターリーブがあると、変換できません。これらは、影響を受ける設定がコメントアウトされ、エラーが生成されます。このような構成は、手動で変換する必要があります。
- クラシックポリシーとアドバンスドポリシーの両方を cmp global にバインドできます。クラシックポリシーを高度なポリシーに変換すると、機能が変更されるケースは多くあります。いくつかのポリシーをコメントアウトすることで解決できるコマンドを変換しました。それでも、変換できないコマンドがいくつかあります。このような場合、エラーが発生し、変換は手動で行う必要があります。
- Classic 組み込みの名前付き式のすべての用途が、同等の高度な名前付き式に変換されるわけではありません。
- クライアントセキュリティ式は処理されません。
- コンテンツスイッチおよびキャッシュリダイレクト仮想サーバーの「-precedence」オプションは処理されません。
- SureConnect (SC)
- プライオリティキューイング（PQ）
- HTTP サービス拒否 (HDOS)
- HTML インジェクション
- 認証
- 承認
- VPN
- Syslog
- Nslog
- ファイルベースの Classic 式は処理されません。

> 注:
>
> patClass/Filter のようないくつかの機能では、コマンドの構文が変更されます。cmd ポリシーがある場合は、お客様の要件に応じて cmd ポリシーを変更する必要があります。

既知の問題

この`nspepi`ツールでは、次のエラーが生成される可能性があります。

- 式を変換するときに問題がある場合。
- 名前付きポリシー式で-clientSecurityMessage パラメータが使用されている場合。これは、このパラメータが詳細ポリシー式でサポートされていないためです。

> 注:
>
> -state オプションを無効にした従来のポリシーバインディングはすべてコメントアウトされます。-state オプションは、高度なポリシーバインディングでは使用できません。

nspepi ツールを実行する

以下は、nspepiツールを実行するためのコマンドライン例です。このツールは、シェルのコマンドラインから実行されます（そのためには、NetScaler「CLI」に「シェル」コマンドを入力する必要があります）。変換を実行するには、「-f」または」-e」のいずれかを指定する必要があります。「-d」の使用は、Citrix 担当者がサポート目的で分析することを意図しています。

```
usage: nspepi [-h] (-e <classic policy expression> | -f <path to ns
    config file>)[-d] [-v] [-V

Convert classic policy expressions to advanced policy expressions and
    deprecated commands to non-deprecated
commands.

optional arguments:
-h, --help show this help message and exit
-e <classic policy expression>, --expression <classic policy expression
    >
convert classic policy expression to advanced policy
expression (maximum length of 8191 allowed)
-f <path to ns config file>, --infile <path to ns config file>
convert netscaler config file
-d, --debug log debug output
-v, --verbose show verbose output
-V, --version show program's version number and exit
<!--NeedCopy-->
```

使用例**:**

1. nspepi -e ”req.tcp.destport == 80”
2. nspepi -f ns.conf

次に、CLI を使用したnspepiツールの実行例をいくつか示します。

—e パラメータの出力例**:**


```
root@ns# nspepi -e ”req.http.header foo == ”bar””
”HTTP.REQ.HEADER(”foo”).EQ(”bar”)”
<!--NeedCopy-->
```

-f パラメータの出力例**:**

```
root@ns# cat sample.conf
add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180 -
    originUSIP OFF
add cr policy cr_pol1 -rule ns_true
bind cr vserver cr_vs -policyName cr_pol1
<!--NeedCopy-->
```

-f パラメータを指定して **nspepi** を実行中:

```
nspepi -f sample.conf
<!--NeedCopy-->
```

変換されたコンフィグは新しいファイルnew_sample.confで使用できます。
warn_sample.confファイルをチェックして、生成された可能性のある警告またはエラーがないか確認します。

-f パラメータと-v パラメータの出力例

```
nspepi -f sample.conf -v
INFO - add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180
     -originUSIP OFF
INFO - add cr policy cr_pol1 -rule TRUE -action ORIGIN
INFO - bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

変換されたコンフィグは新しいファイルnew_sample.confで使用できます。
warn_sample.confファイルをチェックして、生成された可能性のある警告またはエラーがないか確認します。

変換された設定ファイル**:**

```
root@ns# cat new_sample.conf
```



```
add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180 -
    originUSIP OFF
add cr policy cr_pol1 -rule TRUE -action ORIGIN
set cmp parameter -policyType ADVANCED
bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
    gotoPriorityExpression END -type REQUEST

<!--NeedCopy-->
```

エラーや警告のない設定例の出力例**:**

```
nspepi -f sample_2.conf
<!--NeedCopy-->
```

変換されたコンフィグは新しいファイル`new_sample_2.conf`で使用できます。
`warn_sample_2.conf`ファイルをチェックして、生成された可能性のある警告またはエラーがないか確認します。

警告付きの設定例の出力例**:**

```
root@ns# cat sample_2.conf
add policy expression security_expr ”req.tcp.destport == 80” -
    clientSecurityMessage ”Not allowed”
set cmp parameter -policyType CLASSIC
add cmp policy cmp_pol1 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol2 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
bind cmp global cmp_pol1
bind cmp global cmp_pol2 -state DISABLED
bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
    RES_DEFAULT
bind lb vserver lb_vs -policyName cmp_pol2
root@ns#
<!--NeedCopy-->
```

-f パラメータを指定した **nspepi** の実行例**:**

```
root@ns# nspepi -f sample_2.conf
ERROR - Error in converting expression security_expr : conversion of
    clientSecurityMessage based expression is not supported.
```



```
3 WARNING - Following bind command is commented out because state is
    disabled. Advanced expressions only have a fixed ordering of the
    types of bindings without interleaving, except that global bindings
    are allowed before all other bindings and after all bindings. If you
     have global bindings in the middle of non-global bindings or any
    other interleaving then you will need to reorder all your bindings
    for that feature and direction. Refer to nspepi documentation. If
    command is required please take a backup because comments will not
    be saved in ns.conf after triggering 'save ns config': bind cmp
    global cmp_pol2 -state DISABLED
4 Warning - Bindings of advanced CMP policies to cmp global are commented
     out, because initial global cmp parameter is classic but advanced
    policies are bound. Now global cmp parameter policy type is set to
    advanced. If commands are required please take a backup because
    comments will not be saved in ns.conf after triggering 'save ns
    config'. Advanced expressions only have a fixed ordering of the
    types of bindings without interleaving, except that global bindings
    are allowed before all other bindings and after all bindings. If you
     have global bindings in the middle of non-global bindings or any
    other interleaving then you will need to reorder all your bindings
    for that feature and direction. Refer to nspepi documentation.
5 root@ns#
6 <!--NeedCopy-->
```

変換されたファイル:

```
1  root@ns# cat new_sample_2.conf
2  add policy expression security_expr ”req.tcp.destport == 80” -
       clientSecurityMessage ”Not allowed”
3  set cmp parameter -policyType ADVANCED
4  add cmp policy cmp_pol1 -rule TRUE -resAction COMPRESS
5  add cmp policy cmp_pol2 -rule TRUE -resAction COMPRESS
6  add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
7  #bind cmp global cmp_pol2 -state DISABLED
8  #bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
       RES_DEFAULT
9  bind cmp global cmp_pol1 -priority 100 -gotoPriorityExpression END -
       type RES_DEFAULT
10 bind lb vserver lb_vs -policyName cmp_pol2 -priority 100 -
       gotoPriorityExpression END -type RESPONSE
11 root@ns#
12 <!--NeedCopy-->
```


警告ファイル:

```
root@ns# cat warn_sample_2.conf
2019-02-28 06:20:10,590: ERROR - Error in converting expression
    security_expr : conversion of clientSecurityMessage based expression
     is not supported.
2019-02-28 06:20:12,187: WARNING - Following bind command is commented
    out because state is disabled. Advanced expressions only have a
    fixed ordering of the types of bindings without interleaving, except
     that global bindings are allowed before all other bindings and
    after all bindings. If you have global bindings in the middle of non
    -global bindings or any other interleaving then you will need to
    reorder all your bindings for that feature and direction. Refer to
    nspepi documentation. If command is required please take a backup
    because comments will not be saved in ns.conf after triggering 'save
     ns config': bind cmp global cmp_pol2 -state DISABLED
2019-02-28 06:20:12,191: WARNING - Bindings of advanced CMP policies to
     cmp global are commented out, because initial global cmp parameter
    is classic but advanced policies are bound. Now global cmp parameter
     policy type is set to advanced. If commands are required please
    take a backup because comments will not be saved in ns.conf after
    triggering 'save ns config'. Advanced expressions only have a fixed
    ordering of the types of bindings without interleaving, except that
    global bindings are allowed before all other bindings and after all
    bindings. If you have global bindings in the middle of non-global
    bindings or any other interleaving then you will need to reorder all
     your bindings for that feature and direction. Refer to nspepi
    documentation.
root@ns#
<!--NeedCopy-->
```

バインディングの優先順位

高度なポリシーでは、グローバルと非グローバルの間、および異なるバインディングタイプ間の優先度による任意のインターリーブは許可されません。このようなクラシックポリシーの優先順位のインターリーブに依存する場合は、高度なポリシールールに準拠し、希望する動作になるように優先順位を調整する必要があります。
詳細ポリシーの優先順位は、バインドポイントに対してローカルです。バインドポイントは、プロトコル、機能、方向、およびエンティティの一意の組み合わせです（エンティティは、特定の仮想サーバ、ユーザ、グループ、サービス、およびグローバルオーバーライドまたはグローバルデフォルトのいずれかです）。ポリシーの優先順位は、バインドポイント全体では従いません。

特定のプロトコル、機能、および方向について、高度なポリシーの評価順序を以下に示します。

- グローバルオーバーライド。
- (現在の) 認証、承認、および監査ユーザー。
- 認証、認可、および監査グループ（ユーザーがメンバーである）の重み順です。複数のグループの重みが同じ場合、順序は定義されません。
- 要求を受信したか、コンテンツスイッチングが選択された LB 仮想サーバー。
- コンテンツスイッチング仮想サーバー、要求を受信したキャッシュリダイレクト仮想サーバー。
- 負荷分散によって選択されたサービス。
- グローバルデフォルト。

認可ポリシーの評価では、次の順序になります。

- システムのオーバーライド。
- 要求を受信した仮想サーバー、または CS が選択された負荷分散仮想サーバー。
- 要求を受信したコンテンツスイッチング仮想サーバー。
- システムデフォルト。

各バインドポイント内では、ポリシーは、番号の小さいものから高い番号の優先順位で評価されます。ポリシーは、使用されているプロトコルとメッセージの受信元の方向についてのみ評価されます。

手動での優先順位の再設定が必要な従来のポリシーバインディング

ここでは、ニーズを満たすために手動で優先順位を再設定する必要があるクラシックポリシーバインディングのタイプをいくつか示します。これらはすべて、特定のフィーチャと方向に対するものです。

- 上記のエンティティタイプリストの方向とは逆のプライオリティ番号が増加するクラシックプライオリティ。たとえば、負荷分散仮想サーバーバインディングよりも低いコンテンツスイッチング仮想サーバーバインディングです。
- 認証、承認、および監査グループをインターリーブする従来の優先順位。あるグループの一部が他のグループの前にあり、さらに別のパートはその他のグループの一部の後にあります。
- 認証、認可、および監査グループの重みの順序以外の数が増加する従来の優先順位。
- 一部の非グローバルプライオリティよりも小さく、同じグローバルプライオリティのクラシックグローバルプライオリティは、他の非グローバルプライオリティ（つまり、非グローバルプライオリティのセグメント、続いて 1 つ以上のグローバル、その後に非グローバルプライオリティ）よりも大きくなります。

従来のポリシー廃止に関する **FAQ**

October 7, 2021

- **Citrix ADC 12.0** 以降のリリースで廃止された従来のポリシーは何ですか**?**

  「非推奨ポリシー 」表に記載されているすべての機能と機能は、Citrix ADC リリース 12.0 ビルド 56.20 から廃止されます。非推奨の機能およびポリシーの詳細については、次の表（PDF 形式）を参照することをお勧め

します。

- 非推奨のポリシーとその代替ポリシーについては、表 1 。
- 表 2 非推奨の Citrix ADC 機能とその代替機能と構成の詳細を示します。

- 従来のポリシーベースの機能と機能を高度なポリシーに変換するにはどうすればよいですか。

  Citrix ADC 独自のnspepiツールを使用して、コマンド、式、構成を変換できます。nspepi ツールは、Citrix ADC 構成のすべてのクラシック式を高度なポリシー式に変換するのに役立ちます。このnspepiツールの詳細については、「NSPEPI ツールを使用したポリシー式の変換」を参照してください。

- クラシックポリシーベースの機能および機能が廃止されるのはどのリリースからですか。

  Citrix ADC 12.0 は 56.20 以降をビルドします。

- クラシックポリシーベースの機能をサポートしないビルドにアプライアンスをアップグレードするときに従うべき手順は何ですか。

  Citrix ADC リリース 13.0 より後のリリースにアプライアンスをアップグレードする前に、高度なポリシーを使用することをお勧めします。詳細については、「高度なポリシー」を参照してください。

- **Citrix ADC** アプライアンスで非推奨の機能がサポートされる期間はどれくらいですか?

  Citrix は、Citrix ADC リリース 13.0 より後のリリースでのクラシックポリシーとその使用をサポートしません。

- 構成ファイルを変換した後、アプライアンスを再起動する必要がありますか。

  はい、ns.configファイルの変換に成功したら、Citrix ADC インスタンスを再起動する必要があります。

先に進む前に

October 7, 2021

式とポリシーを設定する前に、関連する Citrix ADC の機能とデータの構造を次のように理解しておく必要があります。

- 関連する機能に関するドキュメントをお読みください。
- 構成するデータの種類のデータストリームを確認します。

設定するトラフィックまたはコンテンツのタイプに対してトレースを実行することもできます。これにより、式で指定する必要があるパラメータと値、およびこれらのパラメータと値に対する操作が分かります。

注: Citrix ADC は、機能内でクラシックポリシーまたは高度なポリシーのいずれかをサポートします。同じフィーチャに両方のタイプを含めることはできません。過去数リリースでは、Citrix ADC の一部の機能が、従来のポリシーと式の使用から、高度なポリシーと式に移行されました。関心のある機能が [Advanced] ポリシー形式に変更されている場合は、古い情報を手動で移行する必要があります。次に、ポリシーを移行する必要があるかどうかを判断するためのガイドラインを示します。

- リリース 9.0 より前のバージョンの統合キャッシュ機能でクラシックポリシーを設定し、バージョン 9.0 以降にアップグレードしても、影響はありません。レガシーポリシーはすべて、アドバンスポリシー形式に移行されます。
- その他の機能については、機能が Advanced ポリシーに移行されている場合は、従来のポリシーと式を Advanced 構文に手動で移行する必要があります。

高度なポリシーインフラストラクチャの構成

October 7, 2021

DNS、書き換え、レスポンダー、統合キャッシュなど、さまざまな Citrix ADC 機能、および Citrix Gateway のクライアントレスアクセス機能に関する高度なポリシーを作成できます。ポリシーは、これらの機能の動作を制御します。

ポリシーを作成するときは、名前、規則（式）、機能固有の属性、およびデータがポリシーと一致したときに実行されるアクションを割り当てます。ポリシーを作成したら、ポリシーをグローバルにバインドするか、仮想サーバーの要求時間処理または応答時間処理のいずれかによって呼び出されるタイミングを決定します。

同じバインドポイントを共有する ポリシーは、ポリシーバンクと呼ばれます。たとえば、仮想サーバーにバインドされているすべてのポリシーが、仮想サーバーのポリシーバンクを構成します。ポリシーをバインドするときは、優先度レベルを割り当てて、バンク内の他のポリシーと相対的に呼び出されるタイミングを指定します。優先度レベルを割り当てるだけでなく、Goto 式を指定することで、バンク内のポリシーの任意の評価順序を設定できます。

組み込みのバインドポイントまたは仮想サーバーに関連付けられたポリシーバンクに加えて、ポリシーラベルを構成できます。ポリシーラベルは、任意の名前で識別されるポリシーバンクです。ポリシーラベルとその中のポリシーは、グローバルまたは仮想サーバ固有のポリシーバンクから呼び出します。ポリシーラベルまたは仮想サーバポリシーバンクは、複数のポリシーバンクから呼び出すことができます。

一部の機能では、ポリシーマネージャを使用してポリシーを設定およびバインドできます。

ポリシーで使用される識別子の名前の規則

October 7, 2021

名前付き式、HTTP コールアウト、パターンセット、およびレート制限機能の識別子の名前は、ASCII アルファベットまたはアンダースコア（_）で始める必要があります。残りの文字は、ASCII 英数字またはアンダースコア (_) です。

これらの識別子の名前は、次の予約語で始めることはできません。

- 単語 ALT、TRUE、FALSE、または Q または S の 1 文字の識別子。
- 特殊構文インジケータ RE (正規表現の場合) または XP (XPath 式の場合)。
- 式プレフィックス。現在は、次のとおりです。

- クライアント
- EXTEND
- HTTP
- SERVER
- SYS
- TARGET
- TEXT
- URL
- MYSQL
- MSSQL

さらに、これらの識別子の名前は、ポリシーインフラストラクチャで使用される列挙定数の名前と同じにすることはできません。たとえば、識別子の名前は、IGNORECASE、YEAR、または LATIN2_CZECH_CS（MySQL 文字セット）にすることはできません。

注： Citrix ADC アプライアンスは、識別子とこれらの単語および列挙定数との大文字と小文字を区別しない比較を実行します。たとえば、識別子の名前を TRUE、True、または true で始めることはできません。

ポリシーの作成または変更

October 13, 2021

すべてのポリシーには、いくつかの共通要素があります。ポリシーの作成は、少なくとも、ポリシーの名前付けと規則の設定で構成されます。さまざまな機能のポリシー設定ツールには、重複する領域がありますが、相違点もあります。アクションをポリシーに関連付けるなど、特定の機能に対するポリシーの設定の詳細については、機能のマニュアルを参照してください。

ポリシーを作成するには、まずポリシーの目的を決定します。たとえば、イメージファイルの HTTP 要求、または SSL 証明書を含むクライアント要求を識別するポリシーを定義できます。ポリシーで使用する情報の種類を知るだけでなく、ポリシーが分析するデータの形式も知る必要があります。

次に、ポリシーがグローバルに適用可能かどうか、または特定の仮想サーバーに関連しているかどうかを判断します。また、ポリシーが評価される順序（ポリシーのバインド方法によって決定されます）が、設定しようとしているポリシーに与える影響についても考慮してください。

CLI を使用したポリシーの作成

コマンドプロンプトで次のコマンドを入力して、ポリシーを作成し、構成を確認します。

```
1 -  add responder|dns|cs|rewrite|cache policy <policyName> -rule <
     expression> [<feature-specific information>]
```



```
2
3 -  show rewrite policy <name>
4 <!--NeedCopy-->
```

例 **1**:

```
1  add rewrite policy ”pol_remove-ae” true ”act_remove-ae”
2  Done
3  > show rewrite policy pol_remove-ae
4              Name: pol_remove-ae
5              Rule: true
6              RewriteAction: act_remove-ae
7              UndefAction: Use Global
8              Hits: 0
9              Undef Hits: 0
10             Bound to: GLOBAL RES_OVERRIDE
11             Priority: 90
12             GotoPriorityExpression: END
13 Done
14 <!--NeedCopy-->
```

例 **2**:

```
1  add cache policy BranchReportsCachePolicy -rule q{
2   http.req.url.query.value(”actionoverride”).contains(”branchReport s”)
       }
3    -action cache
4  Done
5  show cache policy BranchReportsCachePolicy
6              Name: BranchReportsCachePolicy
7              Rule: http.req.url.query.value(”actionoverride”).contains(”
                  branchReports”)
8              CacheAction: CACHE
9              Stored in group: DEFAULT
10             UndefAction: Use Global
11             Hits: 0
12             Undef Hits: 0
13 Done
14 <!--NeedCopy-->
```

注**:** コマンド・ラインでは、ポリシー・ルール（式）内の引用符（q）をエスケープするか、q デリミタで区切る必要があります。詳細については、「高度なポリシー式の構成: はじめに」を参照してください。

GUI を使用したポリシーの作成または変更

1. ナビゲーションウィンドウで、ポリシーを構成する機能の名前を展開し、[**Policies**] をクリックします。たとえば、[コンテンツの切り替え]、[統合キャッシュ]、[**DNS**]、[書き換え]、または [レスポンダー] を選択できます。
2. 詳細ウィンドウで [追加] をクリックするか、既存のポリシーを選択して [開く] をクリックします。ポリシー設定ダイアログボックスが表示されます。
3. 次のパラメータの値を指定します。(アスタリスクは必須パラメータを示します。括弧内の用語については、「ポリシーを作成または変更するためのパラメータ」の対応するパラメータを参照してください。)
4. [**Create**] をクリックしてから、[**Close**] をクリックします。
5. [保存] をクリックします。ポリシーが追加されます。
   注：ポリシーを作成したら、設定ペインでポリシーエントリをクリックして、ポリシーの詳細を表示できます。強調表示および下線付きの詳細は、対応するエンティティ (名前付き式など) へのリンクです。

ポリシー設定例

October 7, 2021

次の例は、コマンドラインインターフェイスでポリシーと関連するアクションを入力する方法を示しています。構成ユーティリティでは、式は、統合キャッシュまたは書き換え機能の機能構成ダイアログボックスの [式] ウィンドウに表示されます。

次に、キャッシュポリシーの作成例を示します。キャッシュポリシーのアクションは組み込まれているため、ポリシーとは別に設定する必要はありません。

```
1   add cache policy BranchReportsCachePolicy -rule q{
2  http.req.url.query.value(”actionoverride”).contains(”branchReports”) }
3   -action cache
4 <!--NeedCopy-->
```

書き換えポリシーとアクションの例を次に示します。

```
1  add rewrite action myAction1 INSERT_HTTP_HEADER ”myHeader” ”
      valueForMyHeader”
2  add rewrite policy myPolicy1 ”http.req.url.contains(”myURLstring”)”
      myAction1
3 <!--NeedCopy-->
```


注**:** コマンド・ラインでは、ポリシー・ルール（式）内の引用符（q）をエスケープするか、q デリミタで区切る必要があります。詳細については、「高度なポリシー式の構成: はじめに」を参照してください。

ポリシーマネージャを使用したポリシーの設定とバインド

October 13, 2021

> 警告:
>
> Citrix ADC 12.0 ビルド 56.20 以降では、従来のポリシー式はサポートされなくなりました。また、高度なポリシーを使用することをお勧めします。詳細については、「高度なポリシー」を参照してください。

一部のアプリケーションでは、Citrix ADC 構成ユーティリティに専用の Policy Manager が用意されており、ポリシーバンクの構成を簡素化できます。また、使用されていないポリシーやアクションを検索して削除することもできます。

Policy Manager は、現在、書き換え、統合キャッシュ、レスポンダー、圧縮機能で使用できます。

ここでは、この項の手順に相当するキーボード操作を示します。

- Policy Manager でセルを編集するには、Tab キーでセルに移動し、[F2] をクリックするか、キーボードの [Space] バーを押します。
- ドロップダウンメニューでエントリを選択するには、Tab キーを使用してエントリに移動し、スペースバーを押してドロップダウンメニューを表示し、上方向キーおよび下方向キーを使用して目的のエントリに移動し、もう一度スペースバーを押してエントリを選択します。
- ドロップダウンメニューの選択をキャンセルするには、Esc キーを押します。
- ポリシーを挿入するには、Tab キーを使用して挿入ポイントの上にある行に移動し、Control キーを押しながら Insert キーを押すか、[ポリシーの挿入] をクリックします。
- ポリシーを削除するには、Tab キーを使用してポリシーを含む行に移動し、Delete キーを押します。

注: ポリシーを削除すると、Citrix ADC は銀行内の他のポリシーの Goto Expression 値を検索します。これらの [Goto Expression] の値のいずれかが、削除されたポリシーのプライオリティレベルと一致する場合、それらは削除されます。

ポリシーマネージャーを使用してポリシーバインディングを構成する

1. ナビゲーションペインで、ポリシーを設定する機能をクリックします。選択肢は、[レスポンダー]、[統合キャッシュ]、[書き換え]、[圧縮] です。
2. 詳細ペインで、[ポリシーマネージャ] をクリックします。
3. 圧縮用のクラシックポリシーバインディングを構成する場合は、[圧縮ポリシーマネージャ] ダイアログボックスで、[クラシック構文に切り替える] をクリックします。ダイアログボックスが従来の構文ビューに切り替わり、[高度なポリシーに切り替える] ボタンが表示されます。ポリシーバインディングの設定を完了する前に、

[詳細ポリシー] を使用するポリシーのバインディングを設定する場合は、[詳細ポリシーに切り替える] ボタンをクリックします。

4. Responder 以外の機能では、バインドポイントを指定するには、[要求] または [応答] をクリックし、要求時間または応答時間のバインドポイントのいずれかをクリックします。オプションは、[グローバル上書き]、[LB 仮想サーバー]、[CS 仮想サーバー]、[デフォルトグローバル]、または [ポリシーラベル] です。レスポンダーを構成している場合は、[要求] および [応答] フロータイプは使用できません。

5. ポリシーをこのバインドポイントにバインドするには、[Insert Policy] をクリックし、以前に構成したポリシー、NOPOLICY ラベル、または [New policy] オプションを選択します。選択するオプションに応じて、次の選択肢があります。

   - 新しいポリシー**:** 「ポリシーの作成または変更」の説明に従ってポリシーを作成し、ポリシーバンク内の各エントリの形式に記載されているように、プライオリティレベル、GoTo 式、およびポリシー呼び出しを設定します。
   - 既存のポリシー、**NOPOLICY**、または NOPOLICY\<feature name\>:ポリシーバンクの各エントリの表で説明されているように、プライオリティレベル、GoTo 式、およびポリシー呼び出しを設定します。**NOPOLICY** またはNOPOLICY\<feature name\> オプションは、詳細ポリシーを使用するポリシーに対してのみ使用できます。

6. 上記の手順を繰り返して、このポリシーバンクにエントリを追加します。

7. エントリの優先度レベルを変更するには、次のいずれかを実行します。

   - エントリの [優先順位] フィールドをダブルクリックし、値を編集します。
   - ポリシーをクリックし、表内の別の行にドラッグします。
   - [優先度を再生成] をクリックします。

   3 つのいずれの場合も、他のすべてのポリシーの優先度レベルは、必要に応じて新しい値に対応するように変更されます。整数値のある Goto 式も自動的に更新されます。たとえば、優先順位の値を 10 から 100 に変更すると、[Goto Expression] の値が 10 のすべてのポリシーが値 100 に更新されます。

8. テーブルの行のポリシー、アクション、またはポリシーバンクの呼び出しを変更するには、エントリの右側にある下矢印をクリックし、次のいずれかの操作を行います。

   - ポリシーを変更するには、別のポリシー名を選択するか、[新しいポリシー] を選択して、ポリシーの作成または変更の手順に従います。
   - 「次へ移動」式を変更するには、「次」、「終了」、「USE_INVOCATION_RESULT」を選択するか、または複数を選択して、このポリシーバンク内の別のエントリの優先度レベルを返す式を入力します。
   - 呼び出しを変更するには、既存のポリシーバンクを選択するか、[新しいポリシーラベル] をクリックして、ポリシーをポリシーラベルにバインドするの手順に従います。

9. このバンクからポリシーまたはポリシーラベルの呼び出しをバインド解除するには、ポリシーまたはポリシーラベルを含む行の任意のフィールドをクリックし、[ポリシーのバインド解除] をクリックします。

10. 完了したら、[変更を適用] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインドされたことが示されます。

ポリシーマネージャーを使用して未使用のポリシーを削除する

1. ナビゲーションペインで、ポリシーバンクを設定する機能をクリックします。選択肢は、[レスポンダー]、[統合キャッシュ]、または [書き換え] です。
2. 詳細ペインで、[<Feature Name>ポリシーマネージャ] をクリックします。
3. [機能名] > [ポリシー管理] ダイアログボックスで、[クリーンアップ構成] をクリックします。
4. [クリーンアップ構成] ダイアログボックスで、削除する項目を選択し、[削除] をクリックします。
5. [削除] ダイアログボックスで、[はい] をクリックします。
6. [閉じる] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常に削除されたことが示されます。

ポリシーのバインド解除

October 7, 2021

ポリシーを再割り当てまたは削除する場合は、まずバインドを削除する必要があります。

CLI を使用して、統合キャッシュ、書き換え、または圧縮の詳細ポリシーをグローバルにバインド解除します

コマンドプロンプトで次のコマンドを入力して、統合キャッシュ、書き換え、または圧縮の詳細ポリシーをグローバルにバインド解除し、構成を確認します。

```
1 -  unbind cache|rewrite|cmp global <policyName> [-type req_override|
     req_default|res_override|res_default] [-priority <positiveInteger>]
2
3 -  show cache|rewrite|cmp global
4 <!--NeedCopy-->
```

例:

```
1 > unbind cache global_nonPostReq
2 Done
3 > show cache global
4     1)      Global bindpoint: REQ_DEFAULT
5             Number of bound policies: 1
6
7     2)      Global bindpoint: RES_DEFAULT
8             Number of bound policies: 1
```



```
9
10 Done
11 <!--NeedCopy-->
```

優先度は、NOPOLICY という名前の「ダミー」ポリシーの場合にのみ必要です。

CLI を使用してレスポンダーポリシーをグローバルにバインド解除する

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーのバインドをグローバルに解除し、設定を確認します。

```
1 - unbind responder global <policyName> [-type override|default] [-
    priority <positiveInteger>]
2
3 - show responder global
4 <!--NeedCopy-->
```

例:

```
1 > unbind responder global pol404Error
2 Done
3 > show responder global
4     1)      Global bindpoint: REQ_DEFAULT
5             Number of bound policies: 1
6 Done
7 <!--NeedCopy-->
```

優先度は、NOPOLICY という名前の「ダミー」ポリシーの場合にのみ必要です。

CLI を使用して DNS ポリシーをグローバルにバインド解除する

コマンドプロンプトで次のコマンドを入力して、DNS ポリシーをグローバルにバインド解除し、構成を確認します。

```
1 - unbind responder global <policyName>
2
3 - unbind responder global
4 <!--NeedCopy-->
```

例:


```
unbind dns global dfgdfg
Done
show dns global
    Policy name : dfgdfggfhg
            Priority : 100
            Goto expression : END
Done
<!--NeedCopy-->
```

CLI を使用して仮想サーバから高度なポリシーをバインド解除する

コマンドプロンプトで次のコマンドを入力して、仮想サーバーから高度なポリシーをバインド解除し、構成を確認します。

```
-  unbind cs vserver <name> -policyName <policyName> [-priority <
   positiveInteger>] [-type REQUEST|RESPONSE]

-  show lb vserver <name>
<!--NeedCopy-->
```

例:

```
unbind cs vserver vs-cont-switch  -policyName pol1
Done
> show cs vserver vs-cont-switch
            vs-cont-switch (10.102.29.10:80) - HTTP Type: CONTENT
            State: UP
            Last state change was at Wed Aug 19 08:56:55 2009 (+18 ms)
            Time since last state change: 0 days, 02:47:55.750
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            State Update: DISABLED
            Default:        Content Precedence: RULE
            Vserver IP and Port insertion: OFF
            Case Sensitivity: ON
            Push: DISABLED  Push VServer:
            Push Label Rule: none
```



```
Done
<!--NeedCopy-->
```

優先度は、NOPOLICY という名前の「ダミー」ポリシーの場合にのみ必要です。

統合キャッシュ、レスポンダー、書き換え、または圧縮のバインドを解除する **GUI** を使用した高度なポリシーのグローバル化

1. ナビゲーションペインで、バインドを解除するポリシーの機能をクリックします（たとえば、統合キャッシュ）。
2. 詳細ペインで、[<Feature Name>ポリシーマネージャ] をクリックします。
3. [**Policy Manager**] ダイアログ・ボックスで、バインドを解除するポリシーのバインド・ポイント（[Advanced Global] など）を選択します。
4. バインドを解除するポリシー名をクリックし、[ポリシーのバインド解除] をクリックします。
5. [変更を適用] をクリックします。
6. [閉じる] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインド解除されたことが示されます。

GUI を使用して **DNS** ポリシーをグローバルにバインド解除する

1. **Traffic Management** > **DNS** > **Policies** に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックします。
3. [グローバルバインド] ダイアログボックスで、[ポリシー] を選択し、[ポリシーのバインド解除] をクリックします。
4. [**OK**] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインド解除されたことが示されます。

GUI を使用して、ロードバランシングまたはコンテンツスイッチング仮想サーバーから高度なポリシーをバインド解除する

1. [トラフィック管理] に移動し、[負荷分散] または [コンテンツスイッチング] を展開し、[仮想サーバー] をクリックします。
2. 詳細ウィンドウで、ポリシーのバインドを解除する仮想サーバーをダブルクリックします。
3. [ポリシー] タブの [アクティブ] 列で、バインドを解除するポリシーの横にあるチェックボックスをオフにします。
4. [**OK**] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常にバインド解除されたことが示されます。

ポリシーラベルの作成

October 7, 2021

ポリシーバンクを設定する組み込みのバインドポイントに加えて、ユーザー定義のポリシーラベルを構成し、ポリシーを関連付けることもできます。

ポリシー・ラベル内では、ポリシーをバインドし、ポリシー・ラベルのポリシー・バンク内の他のポリシーと比較して各ポリシーの評価順序を指定します。Citrix ADC では、次のような任意の評価順序を定義することもできます。

- 「goto」式を使用して、現在のエントリの後に評価されるバンク内の次のエントリを指すことができます。
- ポリシーバンクのエントリを使用して、別のバンクを呼び出すことができます。

各機能によって、ポリシーラベルにバインドできるポリシーのタイプ、ラベルをバインドできる負荷分散仮想サーバーのタイプ、およびラベルを起動できるコンテンツスイッチ仮想サーバーのタイプが決まります。たとえば、TCP ポリシーラベルは、TCP ロードバランシング仮想サーバにのみバインドできます。HTTP ポリシーをこのタイプのポリシーラベルにバインドすることはできません。また、TCP ポリシーラベルは、TCP コンテンツスイッチング仮想サーバからのみ呼び出すことができます。

新しいポリシーラベルを設定したら、組み込みバインドポイントの 1 つ以上のバンクから呼び出すことができます。

CLI を使用してキャッシュポリシーラベルを作成する

コマンドプロンプトで次のコマンドを入力して、キャッシュポリシーラベルを作成し、構成を確認します。

```
-  add cache policylabel <labelName> -evaluates req|res

-  show cache policylabel<labelName>
<!--NeedCopy-->
```

例:

```
> add cache policylabel lbl-cache-pol -evaluates req
Done

> show cache policylabel lbl-cache-pol
            Label Name: lbl-cache-pol
            Evaluates: REQ
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```


CLI を使用したコンテンツスイッチングポリシーラベルの作成

コマンドプロンプトで次のコマンドを入力して、Content Switching ポリシーラベルを作成し、構成を確認します。

```
-  add cs policylabel <labelName> http|tcp|rtsp|ssl

-  show cs policylabel <labelName>
<!--NeedCopy-->
```

例:

```
> add cs policylabel lbl-cs-pol http
Done
> show cs policylabel lbl-cs-pol
            Label Name: lbl-cs-pol
            Label Type: HTTP
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

CLI を使用して書き換えポリシーラベルを作成する

コマンドプロンプトで次のコマンドを入力して、書き換えポリシーラベルを作成し、構成を確認します。

```
-  add rewrite policylabel <labelName> http_req|http_res|url|text|
   clientless_vpn_req|clientless_vpn_res

-  show rewrite policylabel <labelName>
<!--NeedCopy-->
```

例:

```
> add rewrite policylabel lbl-rewrt-pol http_req
Done

> show rewrite policylabel lbl-rewrt-pol
            Label Name: lbl-rewrt-pol
            Transform Name: http_req
```



```
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーラベルを作成する

コマンドプロンプトで次のコマンドを入力して、Responder ポリシーラベルを作成し、構成を確認します。

```
-  add responder policylabel <labelName>

-  show responder policylabel <labelName>
<!--NeedCopy-->
```

例:

```
> add responder policylabel lbl-respndr-pol
Done

> show responder policylabel lbl-respndr-pol
            Label Name: lbl-respndr-pol
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

注: このポリシーラベルは、ポリシーバンクから呼び出します。詳細については、「ポリシーラベルへのポリシーのバインド」を参照してください。

GUI を使用したポリシー・ラベルの作成

1. ナビゲーションペインで、ポリシーラベルを作成する機能を展開し、[**Policy Labels**] をクリックします。[統合キャッシュ]、[書き換え]、[コンテンツの切り替え]、または [レスポンダー] から選択できます。
2. 詳細ペインで、[**Add**] をクリックします。
3. [Name] ボックスに、このポリシーラベルの一意の名前を入力します。
4. ポリシーラベルの機能固有の情報を入力します。たとえば、統合キャッシュの場合、[評価] ドロップダウンメニューで、このポリシーラベルに要求時間ポリシーを含める場合は [REQ] を選択し、このポリシーラベルに応答時間ポリシーを含める場合は [RES] を選択します。[書き換え] では、変換名を選択します。
5. [作成] をクリックします。

6. 組み込みのポリシーバンクの 1 つを構成して、このポリシーラベルを呼び出します。詳細については、「ポリシーラベルへのポリシーのバインド」を参照してください。ステータスバーにポリシーラベルが正常に作成されたことを示すメッセージが表示されます。

ポリシーラベルにポリシーをバインドする

組み込みバインドポイントにバインドされるポリシーバンクと同様に、ポリシーラベル内の各エントリは、ポリシーラベルにバインドされるポリシーです。グローバルにバインドされるポリシーまたは vserver にバインドされるポリシーと同様に、ポリシーラベルにバインドされる各ポリシーは、現在のエントリが処理された後に評価されるポリシーバンクまたはポリシーラベルを呼び出すことができます。次の表は、ポリシーラベルのエントリをまとめたものです。

- **Name**：ポリシーの名前。ポリシーを評価せずに別のポリシーバンクを呼び出す場合は、「ダミー」ポリシー名 NOPOLICY。

  ポリシーバンクでは NOPOLICY を複数回指定できますが、名前付きポリシーは一度しか指定できません。

- **Priority**：整数。この設定は、Goto 式で使用できます。
- **Goto** 式。このバンクで評価する次のポリシーを決定します。次のいずれかの値を指定できます。
  - 次へ。次に高い優先度を持つポリシーに移動します。
  - 終了。評価を停止します。
  - **INVOCATION_RESULT**。このエントリが別のポリシーバンクを呼び出す場合に適用されます。呼び出されたバンクの最後の Goto の値が END の場合、評価は停止します。最後の Goto が END 以外の場合、現在のポリシーバンクが NEXT を実行します。
  - 正の数：評価される次のポリシーの優先度番号。
  - 数式を指定します。評価される次のポリシーのプライオリティ番号を生成する式。

  Goto は、ポリシーバンクでのみ進めることができます。

  Goto 式を省略すると、END を指定した場合と同じです。

- 呼び出しタイプ。ポリシーバンクタイプを指定します。値には、次のいずれかを指定できます。
  - 仮想サーバを要求します。仮想サーバーに関連付けられた要求時間ポリシーを呼び出します。
  - 応答サーバ。仮想サーバに関連付けられた応答時間ポリシーを呼び出します。
  - ポリシーラベル。バンクのポリシーラベルで識別される、別のポリシーバンクを呼び出します。
- 呼び出し名。［呼び出しタイプ］に指定した値に応じて、仮想サーバーまたはポリシー・ラベルの名前。

ポリシーラベルまたは仮想サーバポリシーバンクの構成

October 7, 2021

ポリシーを作成し、ポリシーをバインドしてポリシーバンクを作成したら、ラベルまたはポリシーバンク内でポリシーを追加構成できます。たとえば、外部ポリシーバンクの呼び出しを設定する前に、ポリシーバンクを設定するまで待つことができます。

このセクションでは、以下のトピックについて説明します:

- ポリシー・ラベルの構成
- 仮想サーバーのポリシーバンクを構成する

ポリシー・ラベルの構成

ポリシーラベルは、一連のポリシーと、他のポリシーラベルおよび仮想サーバ固有のポリシーバンクの呼び出しで構成されます。Invoke パラメータを使用すると、他のポリシーバンクからポリシーラベルまたは仮想サーバ固有のポリシーバンクを呼び出すことができます。特殊な目的の NoPolicy エントリを使用すると、式（ルール）を処理せずに外部バンクを呼び出すことができます。NoPolicy エントリは、ルールが含まれていない「ダミー」ポリシーです。

Citrix ADC コマンドラインからポリシーラベルを構成する場合は、以下のコマンド構文に注意してください。

- gotoPriorityExpression は、表 2 の説明に従って構成されます。詳細ポリシーを使用したバインドポリシーの「ポリシーバンク内のエントリ」セクションの「ポリシーバンク内のエントリ」のポリシーバンク内の各エントリの形式。
- type 引数は必須です。これは、この引数がオプションである従来のポリシーのバインドとは異なります。
- 仮想サーバにバインドされたポリシーのバンクは、ポリシーラベルの呼び出しに使用するのと同じ方法を使用して呼び出すことができます。

CLI を使用したポリシーラベルの設定

コマンドプロンプトで次のコマンドを入力して、ポリシーラベルを構成し、構成を確認します。

```
1 -  bind cache|rewrite|responder policylabel <policylabelName> -
     policyName <policyName> -priority <priority> [-
     gotoPriorityExpression <gotoprioritыExpression>] [-invoke reqvserver
     |resvserver|policylabel <policyLabelName>|<vserverName>]
2
3 -  show cache|rewrite|responder policylabel <policylabelName>
4 <!--NeedCopy-->
```

例**:**

```
1 bind cache policylabel _reqBuiltinDefaults -policyName  _nonGetReq -
     priority 100
2 Done
```



```
show cache policylabel _reqBuiltinDefaults
            Label Name: _reqBuiltinDefaults
            Evaluates: REQ
            Number of bound policies: 3
            Number of times invoked: 0
    1)      Policy Name: _nonGetReq
            Priority: 100
            GotoPriorityExpression: END
    2)      Policy Name: _advancedConditionalReq
            Priority: 200
            GotoPriorityExpression: END

    3)      Policy Name: _personalizedReq
            Priority: 300
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

CLI を使用して **NOPOLICY** エントリを使用して書き換えポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、NOPOLICY エントリを含む書き換えポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
-  bind rewrite global <policyName> <priority> <gotoPriorityExpression>
    -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke
   reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>

-  show rewrite global
<!--NeedCopy-->
```

例:

```
> bind rewrite global NOPOLICY 100 -type REQ_DEFAULT -invoke
   policylabel lbl-rewrt-pol
Done
> show rewrite global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

    2)      Global bindpoint: REQ_OVERRIDE
            Number of bound policies: 1
```



```
9  Done
10 <!--NeedCopy-->
```

CLI を使用して統合キャッシュポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、統合キャッシュポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind cache global NOPOLICY -priority <priority> -
     gotoPriorityExpression <gotoprioritыExpression> -type REQ_OVERRIDE|
     REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|
     policylabel <policyLabelName>|<vserverName>
2
3 -  show cache global
4 <!--NeedCopy-->
```

例**:**

```
1 bind cache global NOPOLICY -priority 100 -gotoPriorityExpression END -
     type REQ_DEFAULT -invoke policylabel lbl-cache-pol
2 Done
3 > show cache global
4     1)      Global bindpoint: REQ_DEFAULT
5             Number of bound policies: 2
6
7     2)      Global bindpoint: RES_DEFAULT
8             Number of bound policies: 1
9
10 Done
11 <!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、Responder ポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind responder global NOPOLICY <priority> <gotoprioritыExpression> -
     type OVERRIDE|DEFAULT -invoke vserver|policylabel <policyLabelName
     >|<vserverName>
```



```
2
3 -  show responder global
4 <!--NeedCopy-->
```

例:

```
1 > bind responder global NOPOLICY 100 NEXT -type DEFAULT -invoke
     policylabel lbl-respndr-pol
2 Done
3 > show responder global
4      1)      Global bindpoint: REQ_DEFAULT
5              Number of bound policies: 2
6
7 Done
8 <!--NeedCopy-->
```

GUI を使用したポリシー・ラベルの設定

1. ナビゲーションペインで、ポリシーラベルを構成する機能を展開し、[Policy Labels] をクリックします。選択肢は、統合キャッシュ、書き換え、またはレスポンダーです。
2. 詳細ウィンドウで、構成するラベルをダブルクリックします。
3. このポリシーラベルに新しいポリシーを追加する場合は、[ポリシーの挿入] をクリックし、[ポリシー名] フィールドで [新しいポリシー] を選択します。ポリシーの追加の詳細については、「ポリシーを作成または変更する」を参照してください。ポリシーバンクを起動し、呼び出しの前に規則を評価しない場合は、[Insert Policy] をクリックし、[Policy Name] フィールドで [NOPOLICY] を選択します。
4. このポリシーラベルのエントリごとに、次の項目を設定します。
   - ポリシー名:

     これは、このバンクに挿入したポリシー名、新しいポリシー、または NOPOLICY エントリによって既に決定されます。
   - 優先度:

     バンク内の評価の絶対順序を決定する数値、または Goto 式と組み合わせて使用される数値。
   - 式:

     ポリシールール。ポリシー式については、次の章で詳しく説明します。概要については、「高度なポリシー式の設定: はじめに」を参照してください。
   - アクション:

     このポリシーが TRUE と評価された場合に実行されるアクション。

- **Goto Expression:**

  オプションです。優先度レベルを強化し、評価する次のポリシーまたはポリシーバンクを決定します。Goto 式で使用可能な値の詳細については、表 2 を参照してください。詳細ポリシーを使用したバインドポリシーの「ポリシーバンク内のエントリ」セクションの「ポリシーバンク内のエントリ」のポリシーバンク内の各エントリの形式。

- 呼び出し**:**

  オプションです。別のポリシーバンクを呼び出します。

5. [**OK**] をクリックします。ステータスバーにポリシーラベルが正常に構成されたことを示すメッセージが表示されます。

仮想サーバーのポリシーバンクを構成する

仮想サーバーのポリシーのバンクを構成できます。ポリシーバンクには個別のポリシーを含めることができます。ポリシーバンク内の各エントリは、必要に応じて、別の仮想サーバ用に設定したポリシーラベルまたはポリシーバンクを呼び出すことができます。ポリシーラベルまたはポリシーバンクを呼び出す場合は、ポリシー名の代わりに NOPOLICY の「ダミー」エントリを選択することで、式（規則）をトリガーすることなく実行できます。

CLI を使用して仮想サーバポリシーバンクにポリシーを追加する

コマンドプロンプトで次のコマンドを入力して、仮想サーバーポリシーバンクにポリシーを追加し、構成を確認します。

```
1 -  bind lb|cs vserver <virtualServerName> <serviceType> [-policyName <
     policyName>] [-priority <positiveInteger>] [-gotoPriorityExpression
     <expression>] [-type REQUEST|RESPONSE]
2
3 -  show lb|cs vserver <virtualServerName>
4 <!--NeedCopy-->
```

例**:**

```
1 add lb vserver vs-cont-sw TCP
2 Done
3 show lb vserver vs-cont-sw
4             vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
5             State: DOWN
6             Last state change was at Wed Aug 19 10:04:02 2009 (+279 ms)
7             Time since last state change: 0 days, 00:02:14.420
```



```
8               Effective State: DOWN
9               Client Idle Timeout: 9000 sec
10              Down state flush: ENABLED
11              Disable Primary Vserver On Down : DISABLED
12              No. of Bound Services :  0 (Total)       0 (Active)
13              Configured Method: LEASTCONNECTION
14              Mode: IP
15              Persistence: NONE
16              Connection Failover: DISABLED
17 Done
18 <!--NeedCopy-->
```

CLI を使用して **NOPOLICY** エントリを使用して仮想サーバポリシーバンクからポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、NOPOLICY エントリを含む仮想サーバポリシーバンクからポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind lb|cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
     NOPOLICY-CACHE|NOPOLICY-RESPONDER -priority <integer> -type REQUEST|
     RESPONSE -gotoPriorityExpression <gotoprioritыExpression> -invoke
     reqVserver|resVserver|policyLabel <vserverName>|<labelName>
2
3 -  show lb vserver
4 <!--NeedCopy-->
```

例**:**

```
1 > bind lb vserver vs-cont-sw -policyname NOPOLICY-REWRITE -priority 200
      -type REQUEST -gotoPriorityExpression NEXT -invoke policyLabel lbl-
     rewrt-pol
2 Done
3 <!--NeedCopy-->
```

GUI を使用して仮想サーバポリシーバンクを構成する

1. 左側のナビゲーションペインで、[**** トラフィック管理] **>** [負荷分散]、[トラフィック管理] **>** [コンテンツスイッチング]、[トラフィック管理] **>** [SSL オフロード]、[セキュリティ] **>** [AAA-アプリケーショントラフィック] または ****Citrix Gateway** で必要に応じて選択し、［仮想サーバー］をクリックします。
2. 詳細ウィンドウで、構成する仮想サーバーを選択し、[開く] をクリックします。

3. [仮想サーバーの構成] ダイアログボックスで、[ポリシー] タブをクリックします。
4. このバンクに新しいポリシーを作成するには、仮想サーバのポリシーバンクに追加するポリシーまたはポリシーラベルのタイプのアイコンをクリックし、[ポリシーの挿入] をクリックします。ポリシー・ルールを評価せずにポリシー・ラベルを呼び出す場合は、NOPOLICY「ダミー」ポリシーを選択します。
5. このポリシーバンクに既存のエントリを設定するには、次のように入力します。
   - 優先度**:**

     バンク内の評価の絶対順序を決定する数値、または Goto 式と組み合わせて使用される数値。
   - 式**:**

     ポリシールール。ポリシー式については、次の章で詳しく説明します。概要については、「高度なポリシー式の設定: はじめに」を参照してください。
   - アクション**:**

     このポリシーが TRUE と評価された場合に実行されるアクション。
   - **Goto Expression:**

     オプションです。次のポリシーまたはポリシーバンクの評価を決定します。Goto 式の指定可能な値の詳細については、「詳細ポリシーを使用してポリシーをバインドする」の「ポリシーバンク内のエントリ」を参照してください。
   - 呼び出し**:**

     オプションです。別のポリシーバンクを呼び出すには、呼び出すポリシーラベルまたは仮想サーバポリシーバンクの名前を選択します。
6. [**OK**] をクリックします。ステータスバーにメッセージが表示され、ポリシーが正常に構成されたことが示されます。

ポリシーラベルまたは仮想サーバポリシーバンクの起動または削除

October 13, 2021

一度だけバインドできるポリシーとは異なり、ポリシーラベルまたは仮想サーバーのポリシーバンクを呼び出すことによって、何度でも使用できます。呼び出しは、次の 2 つの場所から実行できます。

- ポリシーバンク内の名前付きポリシーのバインディングから。
- ポリシーバンクの NOPOLICY「ダミー」エントリのバインディングから。

通常、ポリシーラベルは、呼び出されるポリシーと同じタイプである必要があります。たとえば、レスポンダーポリシーからレスポンダーポリシーラベルを呼び出します。

注: コマンドラインでポリシーバンク内のグローバル NOPOLICY エントリをバインドまたはバインド解除する場合は、優先度を指定して、1 つの NOPOLICY エントリを別のエントリと区別します。

CLI を使用して書き換えまたは統合キャッシュ・ポリシー・ラベルを呼び出す

コマンドプロンプトで、次のいずれかのコマンドを入力して、書き換えまたは統合キャッシュポリシーラベルを呼び出し、構成を確認します。

```
1 -  bind cache global <policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
2
3 -  bind rewrite global<policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
4
5 -  show cache global|show rewrite global
6 <!--NeedCopy-->
```

例:

```
1 > bind cache global _nonPostReq2 -priority 100 -type req_override -
     invoke
2     policylabel lbl-cache-pol
3 Done
4 > show cache global
5     1)      Global bindpoint: REQ_DEFAULT
6             Number of bound policies: 2
7
8     2)      Global bindpoint: RES_DEFAULT
9             Number of bound policies: 1
10
11    3)      Global bindpoint: REQ_OVERRIDE
12            Number of bound policies: 1
13
14 Done
15 <!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーラベルを呼び出す

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーラベルを呼び出し、構成を確認します。


```
-  bind responder global <policy_Name> <priority_as_positive_integer>
   [<gotoPriorityExpression>] -type REQ_OVERRIDE|REQ_DEFAULT|OVERRIDE|
   DEFAULT -invoke vserver|policylabel <label_name>

-  show responder global
<!--NeedCopy-->
```

例:

```
> bind responder global pol404Error1 300 -invoke policylabel lbl-
   respndr-pol
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

Done
<!--NeedCopy-->
```

CLI を使用した仮想サーバポリシーバンクの呼び出し

コマンドプロンプトで次のコマンドを入力して、仮想サーバーポリシーバンクを呼び出し、構成を確認します。

```
-  bind lb vserver <vserver_name> -policyName <policy_Name> -priority <
   positive_integer> [-gotoPriorityExpression <expression>] -type
   REQUEST|RESPONSE -invoke reqvserver|resvserver|policylabel <
   policy_Label_Name>

-  bind lb vserver <vserver_name>
<!--NeedCopy-->
```

例:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 100
Done

> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
```



```
            Last state change was at Wed Jul 15 05:54:24 2009 (+166 ms)
            Time since last state change: 28 days, 06:37:49.250
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
       100   Hits: 0

    2)      Policy : pol-ssl Priority:0
    3)      Policy : ns_cmp_msapp Priority:100
    4)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

CLI を使用して書き換えまたは統合キャッシュポリシーラベルを削除する

コマンドプロンプトで、次のコマンドのいずれかを入力して、書き換えまたは統合キャッシュポリシーラベルを削除し、構成を確認します。

```
-  unbind rewrite global <policyName> -priority <positiveInteger> -type
    REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  unbind cache global <policyName> -priority <positiveInteger> -type
   REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  show rewrite global|show cache global
<!--NeedCopy-->
```

例:


```
> unbind rewrite global NOPOLICY -priority 100 -type REQ_OVERRIDE
> show rewrite global
Done
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

CLI を使用してレスポンダーポリシーラベルを削除する

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーラベルを削除し、構成を確認します。

```
-  unbind responder global <policyName> -priority <positiveInteger> -
   type OVERRIDE|DEFAULT

-  show responder global
<!--NeedCopy-->
```

例:

```
> unbind responder global NOPOLICY -priority 100 -type REQ_DEFAULT
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

CLI を使用した仮想サーバポリシーラベルの削除

コマンドプロンプトで、次のコマンドのいずれかを入力して、仮想サーバーポリシーラベルを削除し、構成を確認します。

```
-  unbind lb vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
   positiveInteger>
```



```
2
3 -  unbind cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
     NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
     positiveInteger>
4
5 -  show lb vserver|show cs vserver
6 <!--NeedCopy-->
```

例:

```
1  > unbind lb vserver lbvip -policyName ns_cmp_msapp -priority 200
2  Done
3  > show lb vserver lbvip
4              lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
5              State: DOWN
6              Last state change was at Wed Jul 15 05:54:24 2009 (+161 ms)
7              Time since last state change: 28 days, 06:47:54.600
8              Effective State: DOWN
9              Client Idle Timeout: 180 sec
10             Down state flush: ENABLED
11             Disable Primary Vserver On Down : DISABLED
12             Port Rewrite : DISABLED
13             No. of Bound Services :  0 (Total)       0 (Active)
14             Configured Method: LEASTCONNECTION
15             Mode: IP
16             Persistence: NONE
17             Vserver IP and Port insertion: OFF
18             Push: DISABLED  Push VServer:
19             Push Multi Clients: NO
20             Push Label Rule: none
21
22     1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
           100   Hits: 0
23
24     1)      Policy : pol-ssl Priority:0
25     2)      Policy : cf-pol Priority:1      Inherited
26 Done
27 <!--NeedCopy-->
```

GUI を使用してポリシーラベルまたは仮想サーバポリシーバンクを呼び出す

1. ポリシーをグローバルにバインドする、ポリシーを仮想サーバーにバインドする、またはポリシーをポリシー

ラベルにバインドするで説明されているように、ポリシーをバインドします。または、ポリシー名の代わりにNOPOLICY の「ダミー」エントリを入力することもできます。これは、ポリシーバンクを評価する前にポリシーを評価したくない場合に行います。

2. [Invoke] フィールドで、トラフィックがバインドされたポリシーと一致するかどうかを評価するポリシーラベルまたは仮想サーバポリシーバンクの名前を選択します。ステータスバーに表示されるメッセージは、ポリシーラベルまたは仮想サーバポリシーバンクが正常に呼び出されたことを示します。

GUI を使用したポリシーラベル呼び出しの削除

1. ポリシーを開き、[Invoke] フィールドをクリアします。ポリシーのバインドを解除すると、ラベルの呼び出しも削除されます。ステータスバーにポリシーラベルが正常に削除されたことを示すメッセージが表示されます。

高度なポリシー式の設定**:** はじめに

October 7, 2021

高度なポリシーでは、高度なポリシー式で指定した情報に基づいてデータが評価されます。高度なポリシー式は、データ要素（HTTP ヘッダー、送信元 IP アドレス、Citrix ADC システム時刻、POST 本体データなど）を分析します。Citrix ADC の一部の機能では、ポリシーで高度なポリシー式を構成するだけでなく、ポリシーのコンテキスト外で高度なポリシー式を構成することもできます。

高度なポリシー式を作成するには、分析するデータを識別するプレフィックスを選択し、データに対して実行する操作を指定します。たとえば、オペレーションでは、指定したテキスト文字列とデータを照合したり、テキスト文字列を HTTP ヘッダーに変換したりできます。その他の操作では、返される文字列と一連の文字列または文字列パターンが一致します。複合式を設定するには、ブール演算子と算術演算子を指定し、かっこを使用して評価の順序を制御します。

高度なポリシー式には、従来の式を含めることもできます。頻繁に使用する式に名前を割り当てると、式を繰り返し作成する必要がなくなります。

ポリシーおよびその他のいくつかのエンティティには、Citrix ADC が通過するトラフィックのパケットを評価したり、Citrix ADC システム自体からデータを抽出したり、外部アプリケーションに要求（「コールアウト」）を送信したり、別のデータを分析したりするために使用するルールが含まれます。ルールは、トラフィックと比較される論理式の形式をとり、最終的に TRUE または FALSE の値を返します。

ルールの要素自体は、TRUE または FALSE、文字列、または数値を返すことができます。

Advanced ポリシー式を設定する前に、ポリシーまたは他のエンティティが評価するデータの特性を理解する必要があります。たとえば、統合キャッシュ機能を使用する場合、ポリシーによって、キャッシュに格納できるデータが決定されます。統合キャッシュでは、Citrix ADC が受信する HTTP リクエストとレスポンスの URL、ヘッダー、およびその他のデータを把握する必要があります。この知識により、実際のデータと一致するポリシーを構成し、Citrix

ADC で HTTP トラフィックのキャッシュを管理できるようになります。この情報は、ポリシーで設定する必要のある式のタイプを決定するのに役立ちます。

高度なポリシー式の基本要素

October 7, 2021

Advanced ポリシー式は、少なくとも 1 つのプレフィクス（またはプレフィクスの代わりに使用される 1 つの要素）で構成されます。ほとんどの式では、プレフィックスが識別するデータに対して実行される操作も指定します。最大 1,499 文字の式を次のようにフォーマットします。

```
<prefix>.<operation> [<compound-operator> <prefix>.<operation>. . .]
```

各項目の意味は次のとおりです。

- <prefix>

  は、式を開始するためのアンカーポイントです。

  プレフィックスは、データの単位を識別するピリオドで区切られたキーです。たとえば、次のプレフィックスは、HTTP リクエストで Content-Type という名前のヘッダーが存在するかどうかを調べます。

  http.req.header(“Content-Type”)

  プレフィックスを単独で使用して、プレフィックスが識別するオブジェクトの値を返すこともできます。

- <operation>

  は、プレフィクスで識別されるデータに対して実行される評価を示します。

  たとえば、次の式を考えます。

  http.req.header(“Content-Type”).eq(“text/html”)

  この式では、演算子コンポーネントは次のとおりです。

  eq(“text/html”)

  この演算子により、Citrix ADC は Content-Type ヘッダーを含む HTTP リクエストを評価し、特にこのヘッダーの値が文字列「text/html」と等しいかどうかを判断します。詳細については、「操作」を参照してください。

- <compound-operator>

  は、複数のプレフィックスまたは prefix.operation 要素から複合式を形成するブール演算子または算術演算子です。

  たとえば、次の式を考えます。

  http.req.header(“Content-Type”).eq(“text/html”) && http.req.url.contains(“.html”)

プレフィックス

式のプレフィックスは、データの離散部分を表します。たとえば、式プレフィックスは、HTTP URL、HTTP Cookie ヘッダー、または HTTP POST リクエストの本文内の文字列を表すことができます。式のプレフィックスは、次のようなさまざまなデータ型を識別して返すことができます。

- TCP/IP パケット内のクライアント IP アドレス
- Citrix ADC システム時刻
- HTTP 経由の外部コールアウト
- TCP または UDP レコードタイプ

ほとんどの場合、式プレフィックスは次のキーワードのいずれかで始まります。

- CLIENT:
  - 次の例のように、要求を送信しているクライアントまたは応答を受信しているクライアントの特性を識別します。
  - プレフィクス client.ip.dst は、要求または応答の宛先 IP アドレスを指定します。
  - プレフィクス client.ip.src は、送信元 IP アドレスを指定します。
- HTTP:
  - 次の例のように、HTTP リクエストまたはレスポンス内の要素を識別します。
  - プレフィックス http.req.body（整数）は、HTTP リクエストの本文を複数行テキストオブジェクトとして指定します。整数で指定された文字位置までです。
  - プレフィックス http.req.header（「ヘッダー名」）は、ヘッダー名で指定された HTTP ヘッダーを指定します。
  - プレフィックス http.req.url は、URL エンコード形式の HTTP URL を指定します。
- SERVER:

  要求を処理中または応答を送信しているサーバー内の要素を識別します。
- SYS:

  トラフィックを処理している Citrix ADC の特性を識別します。

  注： DNS ポリシーは、SYS、CLIENT、および SERVER オブジェクトのみをサポートします。

  さらに、Citrix Gateway では、クライアントレス VPN 機能は次の種類のプレフィックスを使用できます。
- TEXT:

  リクエストまたはレスポンス内のテキストエレメントを識別します。
- TARGET:

  接続のターゲットを識別します。
- URL:

  HTTP リクエストまたはレスポンスの URL 部分の要素を識別します。

一般的な経験則として、任意の式プレフィックスを自己完結型の式にすることができます。たとえば、次のプレフィックスは、文字列引数で指定された HTTP ヘッダーの内容を返す完全な式です (引用符で囲みます)。

```
http.res.header.("myheader")
```

または、プレフィックスと単純な演算を組み合わせて、TRUE と FALSE の値を決定することもできます。たとえば、次の値は TRUE または FALSE を返します。

```
http.res.header.("myheader").exists
```

また、次の例のように、式内の個々のプレフィックスおよび複数のプレフィックスに対して複雑な操作を使用することもできます。

```
http.req.url.length + http.req.cookie.length <= 500
```

指定できる式プレフィックスは、Citrix ADC の機能によって異なります。次の表は、フィーチャごとに対象となる式のプレフィックスについて説明しています。

機能	機能で使用される式プレフィックスのタイプ
DNS	SYS, CLIENT, SERVER
保護機能のレスポンダー	HTTP, SYS, CLIENT
コンテンツスイッチ	HTTP, SYS, CLIENT
リライト	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN
統合キャッシング	HTTP, SYS, CLIENT, SERVER
Citrix Gateway、クライアントレスアクセス	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN

表 1. さまざまな Citrix ADC 機能で許可される式プレフィックスのタイプ

注：機能で許可される式プレフィクスの詳細については、その機能のドキュメントを参照してください。

単一要素式

最も単純なタイプの Advanced ポリシー式には、要素が 1 つ含まれています。この要素は、次のいずれかになります。

- 本当です。詳細ポリシー式は、単に true という値で構成できます。このタイプの式は、常に TRUE の値を返します。これは、ポリシーアクションを連鎖させ、Goto 式をトリガーする場合に便利です。
- 偽です。詳細ポリシー式は、単に false という値で構成できます。このタイプの式は、常に FALSE の値を返します。
- 複合式のプレフィックス。たとえば、プレフィックス HTTP.REQ.HOSTNAME はホスト名を返す完全な式で、HTTP.REQ.URL は URL を返す完全な式です。プレフィックスは、演算および追加のプレフィックスと組み

合わせて使用して、複合式を形成することもできます。

操作

ほとんどの式では、プレフィックスが識別するデータに対する操作も指定します。たとえば、次のプレフィックスを指定するとします。

```
http.req.url
```

このプレフィックスは、HTTP リクエスト内の URL を抽出します。この式プレフィックスでは、式内で演算子を使用する必要はありません。ただし、HTTP 要求 URL を処理する式を設定する場合は、URL の特定の特性を分析する操作を指定できます。次に、いくつかの可能性を示します。

- URL 内の特定のホスト名を検索します。
- URL 内の特定のパスを検索します。
- URL の長さを評価します。
- タイムスタンプを示す文字列を URL で検索し、GMT に変換します。

次に、Server という名前の HTTP ヘッダーを識別するプレフィックスと、ヘッダー値で IIS という文字列を検索する操作の例を示します。

```
http.res.header("Server").contains("IIS")
```

ホスト名を識別するプレフィックスと、名前の値として文字列「www.mycompany.com」を検索する操作の例を次に示します。

```
http.req.hostname.eq("www.mycompany.com")
```

式プレフィックスの基本的な操作

次の表では、式プレフィックスに対して実行できる基本的な操作について説明します。

操作	判断内容
CONTAINS(<string>)	オブジェクトが一致します <string>。以下は例です: http.req.header（「キャッシュ制御」）.contains（「キャッシュなし」）
EXISTS	特定の項目がオブジェクト内に存在します。以下は例です: http.res. ヘッダー（「MyHdr」）.exists
EQ(<text>)	特定の非数値値がオブジェクトに存在します。以下に例を示します。
EQ(<integer>)	特定の数値はオブジェクトに存在します。次に、クライアント.ip.dst.eq (10.100.10.100) の例を示します。

操作	判断内容
LT(<integer>)	オブジェクトの値が特定の値より小さい場合。以下は例である。コンテンツの長さの例 (5000)
GT(<integer>)	オブジェクトの値が特定の値よりも大きい。以下に例を示します。内容の長さ.gt (5)

次の表は、使用可能な操作の種類をいくつかまとめたものです。

操作タイプ	説明
テキスト操作	個々の文字列と文字列のセットを、ターゲットの任意の部分と一致させます。ターゲットには、文字列全体、文字列の先頭、または文字列の先頭と末尾の間のテキストの任意の部分を指定できます。たとえば、「XYZSomeText」から文字列「XYZ」を抽出できます。または、HTTP ヘッダー値を異なる文字列の配列と比較することもできます。テキストを別の種類のデータに変換することもできます。例を次に示します。文字列を整数値に変換し、URL 内のクエリ文字列からリストを作成し、文字列を時間値に変換します。
数値演算	数値演算には、算術演算子の適用、コンテンツの長さの評価、リスト内の項目数、日付、時刻、および IP アドレスが含まれます。

複合高度なポリシー式

October 7, 2021

高度なポリシー式は、ブール演算子または算術演算子とアトミック演算で構成できます。次の複合式には、ブール値 AND があります。

```
http.req.hostname.eq(”mycompany.com”)&& http.req.method.eq(post)
```

次の式は、2 つのターゲットの値を加算し、その結果を 3 番目の値と比較します。

```
http.req.url.length + http.req.cookie.length \<= 500
```

複合式には、任意の数の論理演算子と算術演算子を使用できます。

次の式は、HTTP リクエストの長さを評価します。この式は URL と Cookie に基づいています。
この式は、ヘッダー内のテキストを評価します。また、この 2 つの結果に対してブール値 AND も表示されます。

```
http.req.url.length + http.req.cookie.length \<= 500 && http.req.header.
contains(”some text”)
```

括弧を使用して、複合式での評価の順序を制御できます。

複合式のブール演算子

複合式は、次の演算子を使用して構成します。

- &&.

  この演算子は論理 AND です。式を TRUE と評価するには、すべてのコンポーネントが TRUE と評価される必要があります。

  例:

  http.req.url.hostname.eq(“myHost”) && http.req.header(“myHeader”).exists

- ||.

  この演算子は論理 OR です。式のいずれかのコンポーネントが TRUE と評価された場合、式全体が TRUE になります。

- !.

  P 式に論理 NOT を指定します。

Citrix ADC 構成ユーティリティでは、[式の追加] ダイアログボックスに AND、NOT、OR 演算子が表示されることがあります。ただし、これらの複合式は限定的に使用できます。演算子 &&、||、! を使用することをお勧めしますブール論理を使用する複合式を構成するには。

複合式の括弧

括弧を使用して、式の評価順序を制御できます。以下はその例です:

```
http.req.url.contains(”myCompany.com”)|| (http.req.url.hostname.eq(”myHost”
)&& http.req.header(”myHeader”).exists)
```

次に、別の例を示します。

```
(http.req.header(”Content-Type”).exists && http.req.header(”Content-Type”).
eq(”text/html”))|| (http.req.header(”Transfer-Encoding”).exists || http.req
.header(”Content-Length”).exists)
```


文字列の複合演算

次の表に、文字列データに対する複合演算を設定するために使用できる演算子を示します。

文字列値を生成する操作	説明
str + str	演算子の左側の式の値と右側の値を連結します。例:http.req.hostname + http.req.url.protocol
str + num	演算子の左側にある式の値と、右側の数値を連結します。例:http.req.hostname + http.req.url.content_length
num + str	演算子の左側にある式の数値を、右側の文字列値と連結します。例:http.req.url.content_length + http.req.url.hostname
str + ip	演算子の左側にある式の文字列値を、右側の IP アドレス値と連結します。例:http.req.hostname + 10.00.000.00
IP + str	演算子の左側にある式の IP アドレス値と右側の文字列値を連結します。例:client.ip.dst + http.req.url.hostname
str1 ALT str2	string1 の評価によって undef 例外が発生するか、結果がヌル文字列である場合は string2 を使用します。それ以外の場合は string1 を使用し、string2 は決して評価しません。例:http.req.hostname alt client.ip.src

TRUE または FALSE の結果を生成する文字列に対する演算	説明
str == str	演算子の両側の文字列が同じかどうかを評価します。以下は例です： http.req. ヘッダー（「マイヘッダー」） == http.res. ヘッダー（「マイヘッダー」）
str <= str	演算子の左側の文字列が、右側の文字列と同じか、アルファベット順に先行するかを評価します。
str >= str	演算子の左側の文字列が右側の文字列と同じか、アルファベット順に続くかを評価します。
str < str	演算子の左側の文字列が、右側の文字列の前にアルファベット順にあるかどうかを評価します。
str > str	演算子の左側の文字列が、右側の文字列の後にあるかどうかをアルファベット順に評価します。

TRUE または FALSE の結果を生成する文字列に対する演算	説明
str !!= str	演算子の両側の文字列が異なるかどうかを評価します。

文字列に対する論理演算	説明
bool && bool	この演算子は論理 AND です。複合式のコンポーネントを評価する場合、AND によって結合されるすべてのコンポーネントが TRUE と評価される必要があります。以下は例を示しています。(「ビューレポート & & 私のページラベルを表示」)
bool || bool	この演算子は論理 OR です。複合式のコンポーネントを評価するときに、OR に属する式のいずれかのコンポーネントが TRUE と評価されると、式全体が TRUE になります。以下は、例です。http.req.url.contains (「.js」) || http.res.header。(「コンテンツタイプ」)。(「javascript」) を含む
bool	式に対して論理 NOT を実行します。

数値の複合演算

複合数値式を設定できます。たとえば、次の式は、HTTP ヘッダーの長さと URL の長さの合計を表す数値を返します。

```
http.req.header.length + http.req.url.length
```

次の表では、数値データの複合式を構成するために使用できる演算子について説明します。

数値の算術演算	説明
num + num	演算子の左式の値を右側の式値に追加します。例:http.req.content_length + http.req.url.length
num – num	左の数式値から演算子の右側の式値を減算します。
数 × 数値	演算子の左の式の値に右の式の値を乗算します。例: クライアント.interface.rx スループット * 9
num / num	演算子の左式の値を右の式の値で割ります。

数値の算術演算	説明
num % num	モジュロ、または演算子の左側の式の値を、右側の式の値で除算した剰余を計算します。たとえば、「15 mod 4」の値は 3、「12 mod 4」の値は 0 になります。
~number	数値のビット単位の論理否定を適用した後の数値を返します。次の例では、数値式が 12 (バイナリ 1100) を返すと仮定しています:~ 数値式です。〜演算子を適用した結果は、-11（バイナリ 1110011、合計 32 ビット、すべてが左側）です。32 ビット未満のすべての戻り値は、演算子を適用する前に暗黙的に左側にゼロを持ち、32 ビット幅にします。
number ^ number	等しい長さの 2 つのビットパターンを比較し、各数値引数の対応するビットの各ペアに対して XOR 演算を実行し、ビットが異なる場合は 1 を返し、同じ場合は 0 を返します。整数引数と現在の数値にビット単位の XOR を適用した後の数値を返します。ビット単位の比較の値が同じ場合、戻り値は 0 になります。次の例では、数値. 式 1 は 12 (バイナリ 1100) を返し、数値. 式 2 は 10 (バイナリ 1010) を返します。数値. 式 1 ^ 数値. 式 2 ^ 演算子を式全体に適用した結果は 6 (バイナリ 0110) です。32 ビット未満のすべての戻り値は、演算子を適用する前に暗黙的に左側にゼロを持ち、32 ビット幅にします。
数値 | 数値	数値値にビット単位の OR を適用した後の数値を返します。ビット単位の比較のいずれかの値が 1 の場合、戻り値は 1 になります。次の例では、数値. 式 1 が 12 (バイナリ 1100) を返し、数値. 式 2 が 10 (バイナリ 1010) を返すと仮定しています。数値. 式 1 | 数値. 式 2 | 演算子を式全体に適用した結果は 14 (バイナリ 1110) です。32 ビット未満のすべての戻り値は、演算子を適用する前に暗黙的に左側にゼロを持ち、32 ビット幅にします。

数値の算術演算	説明
number & number	同じ長さの 2 つのビットパターンを比較し、対応するビットの各ペアに対してビット単位の AND 演算を実行します。両方のビットに値 1 が含まれている場合は 1、いずれかのビットが 0 の場合は 0 を返します。次の例では、数値. 式 1 が 12 (バイナリ 1100) を返し、数値. 式 2 は 10 (バイナリ 1010) を返すと仮定しています。数値. 式 1 と数値. 式 2 式全体が 8 (バイナリ 1000) と評価されます。32 ビット未満のすべての戻り値は、演算子を適用する前に暗黙的に左側にゼロを持ち、32 ビット幅にします。
num « num	右側の number 引数のビット数によって、数値値のビット単位の左シフトの後に数値を返します。シフトされるビット数は 32 の整数です。次の例では、numeric.expression1 は 12 (バイナリ 1100) を返し、numeric.expression2 が 3 を返すと仮定しています。数式 1 « 数値. 式 2 LSHIFT 演算子を適用した結果は 96 (バイナリ 110000) です。32 ビット未満のすべての戻り値は、演算子を適用する前に暗黙的に左側にゼロを持ち、32 ビット幅にします。
num » num	整数引数のビット数による数値値のビット単位の右シフト後の数値を返します。シフトされるビット数は 32 の整数です。次の例では、数値. 式 1 が 12 (バイナリ 1100) を返し、数値. 式 2 が 3 を返すと仮定しています。数値. 式 1 » 数値. 式 2 RSHIFT 演算子を適用した結果は 1 (バイナリ 0001) です。32 ビット未満のすべての戻り値は、演算子を適用する前に暗黙的に左側にゼロを持ち、32 ビット幅にします。

TRUE または FALSE の結果を生成する数値演算子	説明
num == num	演算子の左側の式の値が、右側の式の値と等しいかどうかを判断します。
num! = num	演算子の左側の式の値が右側の式の値と等しくないかどうかを判定します。

TRUE または FALSE の結果を生成する数値演算子	説明
num > num	演算子の左側の式の値が右側の式の値より大きいかどうかを判定する。
num < num	演算子の左側の式の値が右側の式の値より小さいかどうかを判定する。
num >= num	演算子の左側の式の値が右側の式の値以上であるかどうかを判定する。
num <= num	演算子の左側の式の値が右側の式の値以下かどうかを判定する

ポリシーインフラストラクチャのデータ型の関数

Citrix ADC ポリシーインフラストラクチャでは、次の数値データ型がサポートされています。

- 整数 (32 ビット)
- 符号なしロング (64 ビット)
- ダブル (64 ビット)

単純な式は、これらのすべてのデータ型を返すことができます。また、算術演算子と論理演算子を使用してこれらのデータ型の値を評価または返す複合式を作成することもできます。また、ポリシー式では、これらの値をすべて使用できます。符号なし long 型のリテラル定数は、文字列 ul を数値に追加することで指定できます。double 型のリテラル定数には、ピリオド (.)、指数、またはその両方が含まれます。

算術演算子、論理演算子、および型昇格

複合式では、次の標準的な算術演算子と論理演算子を double データ型と符号なしロングデータ型に使用できます。

- +、-、*、/
- %, ~, ^, &, |, «, and » (do not apply to double)
- ==、!=、>、<、>=、<=

これらの演算子はすべて、C プログラミング言語と同じ意味を持ちます。

整数型、符号なし long、および double 型のオペランド間の混合演算のすべての場合において。型昇格は、同じ型のオペランドに対して操作を行うために行われます。この演算では、優先順位の高いオペランドに下位優先タイプが昇格されます。優先順位の順序（高い順）は次のとおりです。

- Double

- 符号なしロング
- 整数

したがって、数値の結果を返す操作は、操作に含まれる最も高いタイプの結果を返します。

たとえば、オペランドが整数型で符号なし長型の場合、整数オペランドは自動的に符号なし長型に変換されます。この型変換は、単純な式で行われます。式の接頭辞によって識別されるデータの型が、関数に引数として渡されるデータの型と一致しません。HTTP.REQ.CONTENT_LENGTH.DIV (3ul) のオペレーションでは、接頭辞 HTTP.REQ.CONTENT_LENGTH は、符号なしロングになる整数を返します。符号なし長整数型:DIV () 関数の引数として渡されるデータ型で、符号なし長除算が実行されます。同様に、引数は式で昇格することができます。たとえば、HTTP.REQ.HEADER (「myHeader」) .TYPECAST_DOUBLE_AT.DIV (5) は整数 5 を倍精度型に昇格させ、倍精度除算を行います。

ある型のデータを別の型のデータにキャストする式については、データの型キャストを参照してください。

式での文字セットの指定

October 7, 2021

Citrix ADC アプライアンスのポリシーインフラストラクチャでは、ASCII および UTF-8 文字セットがサポートされています。デフォルトの文字セットは ASCII です。式を設定するトラフィックが ASCII 文字だけで構成されている場合は、式に文字セットを指定する必要はありません。アプライアンスは、バイナリ文字を含むすべての文字列リテラルと文字リテラルを許可します。ただし、UTF-8 文字セットでは、文字列リテラルと文字リテラルが有効な UTF-8 である必要があります。

```
CLIENT.TCP.PAYLOAD(100).CONTAINS("\xff\x02")
```

式では、SET_CHAR_SET () 関数を式のポイントで導入する必要があります。このポイント以降は、指定した文字セットでデータ処理を実行する必要があります。たとえば、式 HTTP.REQ.BODY (1000) .AFTER_REGEX (再/次の例/) .BEFORE_REGEX (前の例/) .CONTAINS_ANY (「Greek_ alphabet」) で、パターンセット「Greek_alphabet」に格納されている文字列が UTF-8 の場合、SET_CHAR_SET を含める必要があります。(UTF_8) 関数 <string> 次のように、CONTAINS_ANY (「<string>」) 関数の直前に記述します。

```
HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In
the preceding example/).SET_CHAR_SET(UTF_8).CONTAINS_ANY("Greek_ alphabet")
```

SET_CHAR_SET () 関数は、後でその文字セットを変更する別の SET_CHAR_SET () 関数によって上書きされない限り、式内の以降のすべての処理 (つまり、後続のすべての関数) の文字セットを設定します。したがって、特定の単純な式のすべての関数が UTF-8 用である場合は、テキストを識別する関数 (HEADER (「<name>」) または BODY () 関数など) の直後に SET_CHAR_SET (UTF_8 <int>) 関数を含めることができます。上記の最初の段落に続く 2 番目の例では、AFTER_REGEX () 関数および BEFORE_REGEX () 関数に渡された ASCII 引数が UTF-8 文字列に変更された場合、次のように、BODY (1000) 関数の直後に SET_CHAR_SET (UTF_8) 関数を含めることができます。


```
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).AFTER_REGEX(re/Bücher/).BEFORE_REGEX
(re/Wörterbuch/).CONTAINS_ANY("Greek_alphabet")
```

UTF-8 文字セットは ASCII 文字セットのスーパーセットなので、文字セットを UTF-8 に変更しても、ASCII 文字セットに設定された式は期待どおりに動作し続けます。

異なる文字セットを持つ複合式

複合式では、式の 1 つのサブセットが ASCII 文字セットのデータを操作するように設定され、残りの式が UTF-8 文字セットのデータを操作するように設定されている場合、式が評価されるときに、各式に指定された文字セットが考慮されます。を個別に設定できます。ただし、複合式を処理する場合、演算子を処理する直前に、アプライアンスは返される ASCII 値の文字セットを UTF-8 に昇格させます。たとえば、次の複合式では、最初の単純な式は ASCII 文字セットのデータを評価し、2 番目の単純な式は UTF-8 文字セットのデータを評価します。

```
HTTP.REQ.HEADER("MyHeader")== HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8)
```

ただし、複合式を処理する場合、「等しい」ブール演算子を評価する直前に、Citrix ADC アプライアンスは、HTTP.REQ.HEADER（「MyHeader」）によって返される値の文字セットを UTF-8 に昇格させます。

次の例の最初の単純な式は、ASCII 文字セットのデータを評価します。ただし、Citrix ADC アプライアンスが複合式を処理する場合、2 つの単純な式の結果を連結する直前に、アプライアンスは HTTP.REQ.BODY (10) から返される値の文字セットを UTF-8 に昇格させます。

```
HTTP.REQ.BODY(10)+ HTTP.REQ.HEADER("MyHeader").SET_CHAR_SET(UTF_8)
```

したがって、複合式は UTF-8 文字セットのデータを返します。

トラフィックの文字セットに基づいて文字セットを指定する

トラフィックの特性に基づいて、文字セットを UTF-8 に設定できます。評価されるトラフィックの文字セットが UTF-8 であるかどうかがわからない場合は、最初の式が UTF-8 トラフィックをチェックし、後続の式が文字セットを UTF-8 に設定する複合式を設定できます。次に、リクエストの最初の 1000 バイトに UTF-8 文字列 Bücher が含まれているかどうかをチェックする前に、リクエストの Content-Type ヘッダーの「charset」の値を最初にチェックする複合式の例を示します。

```
HTTP.REQ.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).TYPECAST_NVLIST_T
('=', '; ', '"').VALUE("charset").EQ("UTF-8")&& HTTP.REQ.BODY(1000).SET_CHAR_SET
(UTF_8).CONTAINS("Bücher")
```

評価されるトラフィックの文字セットが UTF-8 であると確信できる場合は、例の 2 番目の式で十分です。

式内の文字および文字列リテラル

式の評価中に、現在の文字セットが ASCII であっても、文字リテラルと文字列リテラルは、それぞれ一重引用符 (‘’) と引用符 (「」) で囲まれており、UTF-8 文字セットではリテラルと見なされます。特定の式で、関数が ASCII 文字セ

ットの文字リテラルまたは文字列リテラルで動作していて、リテラルに ASCII 以外の文字が含まれている場合、エラーが返されます。

> 注:
>
> 高度なポリシー式の文字列リテラルは、ポリシー式と同じ長さになりました。式は、1499 バイトまたは 8191 バイトの長さにすることができます。

16 進数および 8 進数形式の値

式を設定する場合、8 進数および 16 進数の形式で値を入力できます。ただし、各 16 進数または 8 進数バイトは UTF-8 バイトと見なされます。無効な UTF-8 バイトは、値を手動で入力するか、クリップボードから貼り付けるかにかかわらず、エラーになります。たとえば、「xcex20」は、「c8」の後に「20」を続けることができないため、無効な UTF-8 文字です (マルチバイトの UTF-8 文字列の各バイトには上位ビットが設定されている必要があります)。無効な UTF-8 文字のもう 1 つの例は「xce xa9」です。これは、16 進文字は空白文字で区切られているためです。

UTF-8 文字列を返す関数

text>.XPATH と<text>.XPATH_JSON 関数のみが常に UTF-8 文字列を返します。次の MYSQL ルーチンは、プロトコル内のデータに応じて、実行時にどの文字セットを返すかを決定します。

- MYSQL_CLIENT_T.USER
- MYSQL_CLIENT_T.DATABASE
- MYSQL_REQ_QUERY_T.COMMAND
- MYSQL_REQ_QUERY_T.TEXT
- MYSQL_REQ_QUERY_T.TEXT(<unsigned **int**>)
- MYSQL_RES_ERROR_T.SQLSTATE
- MYSQL_RES_ERROR_T.MESSAGE
- MYSQL_RES_FIELD_T.CATALOG
- MYSQL_RES_FIELD_T.DB
- MYSQL_RES_FIELD_T.TABLE
- MSQL_RES_FIELD_T.ORIGINAL_TABLE
- MYSQL_RES_FIELD_T.NAME
- MYSQL_RES_FIELD_T.ORIGINAL_NAME
- MYSQL_RES_OK_T.MESSAGE
- MYSQL_RES_ROW_T.TEXT_ELEM(<unsigned **int**>)

UTF-8 の端末接続設定

ターミナル接続（PuTTY などを使用して）を使用して Citrix ADC アプライアンスへの接続を設定する場合は、データの転送に使用する文字セットを UTF-8 に設定する必要があります。

高度なポリシー式の最小関数と最大関数

高度なポリシー式では、以下の最小関数と最大関数がサポートされています。

1. (<expression1>.max(<expression2>) -2 つの値の最大値を返します。
2. (<expression1>.min(<expression2>) -2 つの値の最小値を返します。

高度なポリシー式のクラシック式

September 2, 2022

> 警告:
>
> Citrix ADC 12.0 ビルド 56.20 以降では、従来のポリシー式はサポートされなくなりました。また、高度なポリシーを使用することをお勧めします。詳細については、「高度なポリシー式の構成: はじめに」を参照してください。

古典的な表現は、トラフィックの基本的な特性を表します。場合によっては、高度なポリシー式でクラシック式を使用したい場合があります。

以下は、クラシック式を使用するすべての高度なポリシー式の構文です。

SYS.EVAL_CLASSIC_EXPR (「エクスプレッション」)

> 注:
>
> SYS.EVAL_CLASSIC_EXPR 式の構文とメタデータは廃止される予定です。手動で変換するか、nspepi ツールを使用して、クラシック式を高度な式に変換できます。

SYS.EVAL_CLASSIC_EXPR (「式」) 式の例を以下に示します。

```
1 sys.eval_classic_expr(”req.ssl.client.cipher.bits > 1000”)
2 sys.eval_classic_expr(”url contains abc”)
3 sys.eval_classic_expr(”req.ip.sourceip == 10.102.1.61 -netmask
     255.255.255.255”)
4 sys.eval_classic_expr(”time >= *:30:00GMT”)
5 sys.eval_classic_expr(”e1 || e2”)
6 sys.eval_classic_expr(”req.http.urllen > 50”)
7 sys.eval_classic_expr(”dayofweek == wedGMT”)
8 <!--NeedCopy-->
```


ポリシーで高度なポリシー式を構成する

October 7, 2021

1 つのポリシーで最大 1,499 文字の Advanced ポリシー式を設定できます。詳細ポリシー式のユーザインターフェイスは、式を設定する機能、および式をポリシー用に設定するのか、別の用途用に設定するのかによって異なります。

コマンドラインで式を設定する場合は、二重引用符 (「.」または「..」) を使用して式を区切ります。式内では、バックスラッシュ () を使用して追加の引用符をエスケープします。たとえば、式で引用符をエスケープする標準的なメソッドを次に示します。

”\”abc\””

‘\”abc\”’

コマンドラインで疑問符やその他のバックスラッシュをエスケープするには、バックスラッシュを使用する必要があります。たとえば、式の http.req.url が含まれています (「?」) は、疑問符が解析されるようにバックスラッシュを必要とします。疑問符を入力すると、コマンドラインにバックスラッシュ文字が表示されないことに注意してください。一方、バックスラッシュをエスケープした場合 (例:’http.req.url.contains(“\\http”)’)、エスケープ文字はコマンドラインでエコーされます。

エントリを読みやすくするには、式全体の引用符をエスケープします。式の先頭に、エスケープシーケンス「q」と次の特殊文字の 1 つを入力します。/{< ~$^+=&%@‘?.

次のように、式の最後に特殊文字のみを入力します。

```
q@http.req.url.contains(”sometext”) && http.req.cookie.exists@

q~http.req.url.contains(”sometext”) && http.req.cookie.exists~
<!--NeedCopy-->
```

{区切り文字を使用する式は} で閉じられることに注意してください。

一部の機能 (統合キャッシュやレスポンダーなど) では、ポリシー構成ダイアログボックスに式を構成するためのセカンダリダイアログボックスが表示されます。このダイアログでは、式設定中の各ポイントで使用可能な選択肢を表示するドロップダウンリストから選択できます。これらの設定ダイアログを使用するときは算術演算子を使用できませんが、その他の高度なポリシー式機能を使用できます。算術演算子を使用するには、式を自由形式で作成します。

CLI を使用して高度なポリシー構文ルールを構成する

> 注:
>
> 既定の構文ポリシーの名前が [詳細ポリシー] に変更されました。

コマンドプロンプトで次のコマンドを入力して、既定の構文規則を構成し、構成を確認します。

1. `add cache|dns|rewrite|cs policy policyName **-rule** expression featureSpecificPa **-action**`
2. `show cache|dns|rewrite|cs policy policyName`
   次に、キャッシュポリシーの設定例を示します。

   例:

```
> add cache policy pol-cache -rule http.req.content_length.le(5) -
   action INVAL
 Done

> show cache policy pol-cache
        Name: pol-cache
        Rule: http.req.content_length.le(5)
        CacheAction: INVAL
        Invalidate groups: DEFAULT
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

GUI を使用して既定の構文ポリシー式を構成する

1. ナビゲーションウィンドウで、ポリシーを構成する機能の名前をクリックします。たとえば、[統合キャッシュ]、[レスポンダー]、[DNS]、[書き換え]、または [コンテンツの切り替え] を選択し、[ポリシー] をクリックします。
2. [追加] をクリックします。
3. ほとんどのフィーチャでは、[式] フィールドをクリックします。コンテンツを切り替えるには、[設定] をクリックします。
4. [プレフィックス] アイコン (家) をクリックし、ドロップダウンリストから最初の式のプレフィックスを選択します。たとえば、レスポンダーでは、オプションは HTTP、SYS、および CLIENT です。次の適用可能なオプションのセットがドロップダウンリストに表示されます。

5. 次のオプションをダブルクリックして選択し、ピリオド (.) を入力します。ここでも、適用可能なオプションのセットが別のドロップダウンリストに表示されます。
6. 入力フィールド (かっこで囲まれた) が表示されるまで、オプションの選択を続けます。入力フィールドが表示されたら、括弧内に適切な値を入力します。たとえば、GT (int) (より大きい、整数の形式) を選択した場合は、括弧内に整数を指定します。テキスト文字列は引用符で区切られます。次に例を示します。

   ```
   HTTP.REQ.BODY(1000).BETWEEN("this","that")
   ```

7. 複合式の 2 つの部分間に演算子を挿入するには、[演算子] アイコン (シグマ) をクリックし、演算子のタイプを選択します。次に、ブール値 OR（二重縦棒 || によって示される）で構成された式の例を示します。

   ```
   HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this
   ","that")
   ```

8. 名前の付いた式を挿入するには、[追加] アイコン (プラス記号) の横にある下向き矢印をクリックし、名前の付いた式を選択します。
9. ドロップダウンメニューを使用して式を設定し、組み込み式を挿入するには、[追加] アイコン (プラス記号) をクリックします。**[**式の追加] ダイアログボックスは、メインダイアログボックスと同じように機能しますが、オプションを選択するためのドロップダウンリストや、かっこではなくデータ入力用のテキストフィールドを提供します。このダイアログボックスには、よく使用する式を挿入する [頻繁に使用する式] ドロップダウンリストもあります。式の追加が完了したら、[**OK**] をクリックします。
10. 入力が終わったら、[作成] をクリックします。ステータスバーにメッセージが表示され、ポリシー式が正常に設定されたことが示されます。

GUI を使用して既定の構文式をテストする

1. ナビゲーションウィンドウで、ポリシーを構成する機能の名前をクリックし (たとえば、[統合キャッシュ]、[レスポンダー]、[DNS]、[書き換え]、または [コンテンツの切り替え] を選択)、[ポリシー] をクリックします。
2. ポリシーを選択し、[**Open**] をクリックします。
3. 式をテストするには、[評価] アイコン (チェックマーク) をクリックします。
4. [式エバリュエーター] ダイアログボックスで、式に一致するフロータイプを選択します。
5. [**HTTP** 要求データ] フィールドまたは [**HTTP** 応答データ] フィールドで、式で解析する HTTP 要求または応答を貼り付け、[評価] をクリックします。完全な HTTP リクエストまたはレスポンスを指定する必要があり、ヘッダーと本文は空白行で区切る必要があります。HTTP ヘッダーをトラップするプログラムによっては、応答もトラップされません。ヘッダーのみをコピーして貼り付ける場合は、ヘッダーの最後に空白行を挿入して、完全な HTTP 要求または応答を形成します。
6. [閉じる] をクリックして、このダイアログボックスを閉じます。

名前付き詳細ポリシー式を構成する

October 7, 2021

複数のポリシーで同じ式を複数回再入力する代わりに、名前付き式を設定し、その式をポリシーで使用したいときはいつでもその名前を参照できます。たとえば、次の名前付き式を作成できます。

- ThisExpression:

  ```
  http.req.body(100).contains("this")
  ```

- ThatExpression:

  ```
  http.req.body(100).contains("that")
  ```

その後、これらの名前付き式をポリシー式で使用できます。たとえば、次に示すのは、前述の例に基づく正当な表現です。

この式	その式です

高度なポリシー式の名前を関数のプレフィックスとして使用できます。名前付きエクスプレッションは、単純なエクスプレッションまたは複合エクスプレッションのいずれかになります。関数は、名前付き式によって返されるデータの型を操作できる関数でなければなりません。

例 **1:** プレフィックスとしての単純な名前付き式

テキスト文字列を識別する次の単純な名前付き式は、<string> テキストデータを操作する AFTER_STR (「<string>」) 関数のプレフィックスとして使用できます。

```
HTTP.REQ.BODY(1000)
```

式の名前が top1KB の場合は、HTTP.REQ.BODY (1000) .AFTER_STR (「ユーザー名」) の代わりにトップ 1KB.AFTER_STR (「ユーザー名」) を使用できます。

例 **2:** プリフィックスとしての複合名前付き式

basic_header_value という名前の複合式を作成して、次のように要求内のユーザー名、コロン (:)、およびユーザーのパスワードを連結できます。

```
add policy expression basic_header_value "HTTP.REQ.USER.NAME + \":\" + HTTP
.REQ.USER.PASSWD"
```

次に、次の例に示すように、リライトアクションで式の名前を使用できます。


```
add rewrite action insert_b64encoded_authorization insert_http_header
authorization '"Basic " + basic_header_value.b64encode'-bypassSafetyCheck
 YES
```

この例では、カスタムヘッダーの値を構築するために使用される式では、B64 エンコードアルゴリズムが、複合名前付き式によって返される文字列に適用されます。

また、名前付き式（単独で、または関数のプレフィックスとして）を使用して、書き換えで置換ターゲットのテキスト式を作成することもできます。

CLI を使用して名前付きデフォルト構文式を設定する

コマンドプロンプトで次のコマンドを入力して、名前付き式を構成し、構成を確認します。

```
1 -  add policy expression <name><value>
2
3 -  show policy expression <name>
4 <!--NeedCopy-->
```

例:

```
1 > add policy expression myExp "http.req.body(100).contains("the other")
     "
2 Done
3
4 > show policy expression myExp
5     1)      Name: myExp  Expr: "http.req.body(100).contains("the other"
         )"  Hits: 0 Type : ADVANCED
6 Done
7 <!--NeedCopy-->
```

式は最大 1,499 文字です。

GUI を使用して名前付き式を構成する

1. ナビゲーションウィンドウで、[**AppExpert**] を展開し、[式] をクリックします。
2. [高度な式] をクリックします。
3. [追加] をクリックします。
4. 式の名前と説明を入力します。
5. 高度なポリシー式の構成で説明されているプロセスを使用して、式を設定します。ステータスバーにメッセージが表示され、ポリシー式が正常に設定されたことが示されます。

ポリシーのコンテキスト外で高度なポリシー式を構成する

October 7, 2021

以下を含む多くの関数では、ポリシーの一部ではない高度なポリシー式が必要になることがあります。

- 統合キャッシュ・セレクタ:

  セレクタの定義では、複数の非複合式（選択レット）を定義します。各セレクトレットは、他のセレクトレットと暗黙の論理 AND 関係にあります。

- 負荷分散:

  負荷分散仮想サーバーの負荷分散の TOKEN 方法の式を構成します。

- 書き換えアクション:

  式は、構成するリライトアクションのタイプに応じて、リライトアクションの場所と実行するリライトのタイプを定義します。たとえば、DELETE アクションはターゲット式のみを使用します。REPLACE アクションでは、ターゲット式と式を使用して、置換テキストを設定します。

- レートベースのポリシー:

  制限セレクタを設定するには、高度なポリシー式を使用します。これらのセレクタは、さまざまなサーバへのトラフィックレートを抑制するポリシーを設定するときに使用できます。セレクターの定義では、最大 5 つの非複合式 (セレクトレット) を定義します。各セレクトレットは、暗黙の論理 AND と他のセレクトレットです。

CLI を使用して、ポリシー外で高度なポリシー式を設定します（キャッシュセレクタの例）

コマンドプロンプトで次のコマンドを入力して、ポリシーの外部で高度なポリシー式を設定し、設定を確認します。

```
1 -  add cache selector <selectorName> <rule>
2 -  show cache selector <selectorName>
3 <!--NeedCopy-->
```

例:

```
1 > add cache selector mainpageSelector ”http.req.cookie.value(”ABC_def”)
     ”
2     ”http.req.url.query.value(”_ghi”)”selector ”mainpageSelector” added
3 Done
4 > show cache selector mainpageSelector
5             Name: mainpageSelector
6             Expressions:
```



```
7                       1) http.req.cookie.value(”ABC_def”)
8                       2) http.req.url.query.value(”_ghi”)
9  Done
10 <!--NeedCopy-->
```

次に、ポリシーでの高度なポリシー式の設定で説明されているように、より読みやすい q 区切り文字を使用する同等のコマンドを示します。

```
1  > add cache selector mainpageSelector2 q~http.req.cookie.value(”ABC_def
       ”)~
2      q~http.req.url.query.value(”_ghi”)~selector ”mainpageSelector2”
           added
3  Done
4  > show cache selector mainpageSelector2
5              Name: mainpageSelector2
6              Expressions:
7                       1) http.req.cookie.value(”ABC_def”)
8                       2) http.req.url.query.value(”_ghi”)
9  Done
10 <!--NeedCopy-->
```

高度なポリシー式: テキストの評価

October 7, 2021

要求または応答のテキストを評価する高度なポリシー式を使用してポリシーを設定できます。高度なポリシーテキスト式は、HTTP ヘッダーで文字列マッチングを実行する単純な式から、テキストをエンコードおよびデコードする複雑な式までさまざまです。テキスト式は、大文字と小文字を区別したり、大文字と小文字を区別したり、スペースを使用したり無視したりするように設定できます。テキスト式とブール演算子を組み合わせることによって、複雑なテキスト式を構成することもできます。

式プレフィクスと演算子を使用して、HTTP 要求、HTTP 応答、VPN およびクライアントレス VPN データを評価できます。ただし、テキスト式プレフィクスは、トラフィックのこれらの要素の評価に限定されません。

テキスト式について

October 7, 2021

Citrix ADC アプライアンスを通過するテキストを操作するために、さまざまな式を設定できます。次に、デフォルトの構文式を使用してテキストを解析する方法の例を示します。

- 特定の HTTP ヘッダーが存在することを確認します。

  たとえば、特定のサーバーに要求を転送するために、特定の Accept-Language ヘッダーを含む HTTP 要求を識別できます。

- 特定の HTTP URL に特定の文字列が含まれているかどうかを判断します。

  たとえば、特定の URL に対するリクエストをブロックできます。文字列は、別の文字列の先頭、中間、または末尾に発生する可能性があります。

- 特定のアプリケーションに送信される POST 要求を識別します。

  たとえば、キャッシュされたアプリケーションデータを更新するために、データベースアプリケーションに送信されるすべての POST 要求を識別できます。

HTTP リクエストとレスポンスのデータストリームを表示するための特別なツールがあることに注意してください。ツールを使用して、データストリームを表示できます。

テキストの操作について

テキストベースの式は、データの要素を識別するための少なくとも 1 つのプレフィックスと、通常は (必ずしもそうではありませんが) そのプレフィックスに対する操作で構成されます。テキストベースの操作は、リクエストまたはレスポンスの任意の部分に適用できます。テキストの基本的な操作には、さまざまなタイプの文字列の一致が含まれます。

たとえば、次の式は、ヘッダー値を文字列と比較します。

```
http.req.header("myHeader").contains("some-text")
```

次の式は、リクエスト内のファイルタイプの一致の例です。

```
http.req.url.suffix.contains("jpeg")
```

```
http.req.url.suffix.eq("jpeg")
```

前述の例では、contains 演算子は部分一致を許可し、eq 演算子は完全一致を検索します。

評価する前に文字列をフォーマットするための他の操作も使用できます。たとえば、テキスト操作を使用して、引用符や空白を取り除いたり、文字列をすべての小文字に変換したり、文字列を連結したりできます。

> 注記:
>
> 複雑な操作を使用して、パターンに基づいてマッチングを実行したり、あるタイプのテキストフォーマットを別のタイプに変換したりできます。

詳しくは、次のトピックを参照してください:

- パターンセットとデータセット。
- 正規表現。
- データを型キャストします。

テキスト式の複合と優先順位

さまざまな演算子を適用して、テキストのプレフィックスや式を組み合わせることができます。たとえば、次の式は、各プレフィックスの戻り値を連結します。

```
http.req.hostname + http.req.url
```

次に、論理 AND を使用する複合テキスト式の例を示します。この式の両方のコンポーネントは、式と一致する要求に対して TRUE である必要があります。

```
http.req.method.eq(post)&& http.req.body(1024).startswith(”destination=”)
```

> メモ:
>
> コンパウンドの演算子の詳細については、「複合高度なエクスプレッション」を参照してください。

テキスト式のカテゴリ

設定できるテキスト式の主なカテゴリは次のとおりです。

- HTTP ヘッダー、HTTP URL、および HTTP リクエストの POST ボディ内の情報。

  詳細については、「HTTP リクエストとレスポンスのテキストの式プレフィックス」を参照してください。

- VPN またはクライアントレス VPN に関する情報。

  詳細については、「VPN およびクライアントレス VPN の式プレフィックス」を参照してください。

- TCP ペイロード情報。

  TCP ペイロード式の詳細については、「高度なポリシー式:HTTP、TCP、および UDP データの解析」を参照してください。

- SSL (セキュアソケットレイヤー) 証明書のテキスト。

  SSL および SSL 証明書データのテキスト式については、「高度なポリシー式:SSL 証明書の解析」および「[SSL 証明書の日付の式](/ja-jp/citrix-adc/13/appexpert/policies-and-expressions/adv-policy-exp-working-with-dates-times-and-numbers/exp-for-ssl-certificate-date.html)」を参照してください。

> 注:
>
> POST リクエストの本文などのドキュメント本体を解析すると、パフォーマンスに影響を与える可能性があります。ドキュメント本体を評価するポリシーのパフォーマンスへの影響をテストできます。

テキスト式のガイドライン

パフォーマンスの観点からは、通常、式内でプロトコル対応関数を使用するのが最適です。たとえば、次の式は、プロトコル対応関数を使用します。

```
HTTP.REQ.URL.QUERY
```


前の式は、文字列の解析に基づく次の同等の式よりも優れています。

```
HTTP.REQ.URL.AFTER_STR("?")
```

最初のケースでは、式は URL クエリを具体的に調べます。2 番目のケースでは、式は疑問符が最初に出現するデータをスキャンします。

また、次の式のように、テキストの構造化解析によるパフォーマンス上の利点もあります。

```
HTTP.REQ.HEADER("Example").TYPECAST_LIST_T(',').GET(1)
```

(型キャストの詳細については、データの型キャストを参照してください。カンマ区切りのデータを収集してリストに構造化する型キャスト式は、通常、次の非構造化同等のものよりも優れています。

```
HTTP.REQ.HEADER("Example").AFTER_STR(",").BEFORE_STR(",")
```

最後に、非構造化テキスト式は、通常、正規表現よりもパフォーマンスが優れています。たとえば、構造化されていないテキスト式を次に示します。

```
HTTP.REQ.HEADER("Example").AFTER_STR("more")
```

前の式は、通常、正規表現を使用する次の同等のものよりも優れたパフォーマンスを提供します。

```
HTTP.REQ.HEADER("Example").AFTER_REGEX(re/more/)
```

正規表現の詳細については、「正規表現」を参照してください。

HTTP リクエストとレスポンスのテキストの式プレフィックス

June 24, 2022

HTTP リクエストまたはレスポンスには、通常、ヘッダー、ヘッダー値、URL、POST 本文などのテキストが含まれます。HTTP リクエストまたはレスポンスで、これらのテキストベースの項目の 1 つ以上を操作するように式を設定できます。

パラメーターについて詳しくは、「Citrix ADC 高度なポリシー式リファレンス」を参照してください。

高度な式を使用して設定する方法の詳細については、次のトピックを参照してください。

- 複合高度なポリシー式
- 高度なポリシー式:IP アドレスと MAC アドレス、スループット、VLAN ID
- 高度なポリシー式:SSL の解析
- 高度なポリシー式: 日付、時刻、数字の操作
- 高度なポリシー表現の基本要素
- 高度なポリシー式: テキストの評価
- 高度なポリシー式:HTTP、TCP、および UDP データの解析
- デフォルトの構文式とポリシーの概要例

VPN およびクライアントレス VPN の式プレフィクス

October 7, 2021

高度なポリシーエンジンは、VPN またはクライアントレス VPN データの解析に固有のプレフィクスを提供します。このデータには、次のものが含まれます。

- VPN トラフィックのホスト名、ドメイン、および URL。
- VPN トラフィックのプロトコル。
- VPN トラフィック内のクエリ。

これらのテキスト要素は、多くの場合、URL および URL の構成要素です。これらの要素にテキストベースの操作を適用するだけでなく、URL の解析に固有の操作を使用してこれらの要素を解析できます。詳細については、「URL のセグメントを抽出するための式」を参照してください。

VPN 式プレフィクスの詳細については、VPN 式表を参照してください。

テキストに対する基本的な操作

October 7, 2021

テキストの基本的な操作には、文字列の一致、文字列の長さの計算、大文字と小文字の区別を制御する操作が含まれます。式に引数として渡される文字列には、空白を含めることができますが、文字列は 255 文字を超えることはできません。

文字列比較関数

次の表に、関数がブール値 TRUE または FALSE を返す基本的な文字列照合操作を示します。

機能	説明
<text>.CONTAINS(<string>)	ターゲットに<string>が含まれている場合、ブール値 TRUE を返します。例: http.req.url.contains(".jpeg")
<text>.EQ(<string>)	ターゲットがと完全に一致する場合に、ブール型 TRUE 値を返します<string>。たとえば、次の式は、ホスト名が「myhostabc」の URL に対してブール値 TRUE を返します。http.req.url.hostname.eq("myhostabc")

機能	説明
<text>.STARTSWITH(<string>)	ターゲットが<string>で始まる場合は、ブール型 TRUE 値を返します。たとえば、次の式は、ホスト名が「myhostabc」の URL に対してブール値 TRUE を返します。http.req.url.hostname.startswith(”myhost”)
<text>.ENDSWITH(<string>)	ターゲットがで終わる場合に、ブール型 TRUE 値を返します <string>。たとえば、次の式は、ホスト名が「myhostabc」の URL に対してブール値 TRUE を返します。http.req.url.hostname.endswith(”abc”)
<text>.NE(<string>)	プレフィックスが文字列引数と等しくない場合は、ブール型 TRUE 値を返します。プレフィックスが文字列以外の値を返す場合、関数の引数は、プレフィックスによって返される値の文字列表現と比較されます。関数はSET_TEXT_MODE(IGNORECASE)or SET_TEXT_MODE(NOIGNORECASE)、ASCII 文字セットと UTF-8 文字セットの両方で使用できます。
<text>.GT(<string>)	プレフィックスが文字列引数よりもアルファベット順に大きい場合は、ブール型 TRUE 値を返します。プレフィックスが文字列以外の値を返す場合、関数の引数は、プレフィックスによって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE) またはSET_TEXT_MODE(NOIGNORECASE) 、ASCII 文字セットと UTF-8 文字セットの両方で使用できます。
<text>.GE(<string>)	プレフィックスが文字列引数にアルファベット順より大きいか等しい場合、ブール型 TRUE 値を返します。プレフィックスが文字列以外の値を返す場合、関数の引数は、プレフィックスによって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE) またはSET_TEXT_MODE(NOIGNORECASE) 、ASCII 文字セットと UTF-8 文字セットの両方で使用できます。

機能	説明
<text>.LT(<string>)	プレフィックスが文字列引数よりもアルファベット順に小さい場合は、ブール型 TRUE 値を返します。プレフィックスが文字列以外の値を返す場合、関数の引数は、プレフィックスによって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE) またはSET_TEXT_MODE(NOIGNORECASE) 、ASCII 文字セットと UTF-8 文字セットの両方で使用できます。
<text>.LE(<string>)	プレフィックスがアルファベット順に文字列引数以下の場合、ブール型 TRUE 値を返します。プレフィックスが文字列以外の値を返す場合、関数の引数は、プレフィックスによって返される値の文字列表現と比較されます。関数は、SET_TEXT_MODE(IGNORECASE) またはSET_TEXT_MODE(NOIGNORECASE) 、ASCII 文字セットと UTF-8 文字セットの両方で使用できます。

文字列の長さを計算する

<text>.LENGTH演算は、文字列内の文字数（バイト数ではない）に等しい数値を返します。

<text>.LENGTH

たとえば、特定の長さを超えるリクエスト URL を特定できます。次に、この例を実装する式を示します。

```
HTTP.REQ.URL.LENGTH < 500
```

文字列内の文字または要素の数を取った後、それらに数値演算を適用することができます。詳細については、「デフォルトの構文式: 日付、時刻、および数字の操作」を参照してください。

テキストの大文字小文字の考慮、無視、変更

次の関数は、文字列内の文字の大文字 (大文字または小文字) を操作します。

機能	説明
<text>.SET_TEXT_MODE(IGNORECASE|NOIGNORECASE)	この関数は、すべてのテキスト操作で大文字と小文字の区別をオンまたはオフにします。

機能	説明
<text>.TO_LOWER	ターゲットを小文字に変換し、最大 2 キロバイト (KB) のテキストブロックにします。ターゲットが 2 KB を超える場合、UNDEF を返します。たとえば、文字列「ABCd:」は「abcd:」に変換されます。
<text>.TO_UPPER	ターゲットを大文字に変換します。ターゲットが 2 KB を超える場合、UNDEF を返します。たとえば、文字列「abcD:」は「ABCD:」に変換されます。.

文字列から特定の文字を取り除く

STRIP_CHARS (<string>) 関数を使用すると、デフォルトの構文式プレフィックス (入力文字列) によって返されるテキストから特定の文字を削除できます。引数で指定した文字のすべてのインスタンスは、入力文字列から削除されます。結果の文字列には、文字列とパターンセットの一致に使用するメソッドを含め、任意のテキストメソッドを使用できます。

たとえば、CLIENT.UDP.DNS. ドメイン.STRIP_CHARS (「.-_」) という式では、STRIP_CHARS (<string>) 関数によって、プレフィクス CLIENT.UDP.DNS.DOMAIN によって返されたドメイン名からすべてのピリオド (.)、ハイフン (-)、およびアンダースコア (_) が取り除かれます。返されるドメイン名が「.dom_ai_n-name」の場合、この関数は文字列「adomainname」を返します。

次の例では、結果の文字列を「listofdomains」というパターンセットと比較します。

```
CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_").CONTAINS_ANY("listofdomains")
```

注: STRIP_CHARS(<string>)関数によって返される文字列に対して書き換えを実行することはできません。

次の関数は、指定された文字列入力の先頭と末尾から一致する文字を取り除きます。

機能	説明
<text>.STRIP_START_CHARS(s)	一致しない最初の文字が見つかるまで、入力文字列の先頭から一致する文字を取り出し、残りの文字列を返します。除去する文字は、引用符で囲んで単一の文字列として指定する必要があります。たとえば、ヘッダーの名前が TestLang で、:*/en_us:* がその値である場合、*HTTP.RES.HEADER (「TestLang」) .STRIP_START_CHARS (「:」)* は、最初に一致しない文字 e が見つかるまで、ヘッダーの値の先頭から指定された文字を取り除き、returnsen_us: を文字列を返します。

機能	説明
<text>.STRIP_END_CHARS(s)	最初に一致しない文字が見つかったまでの入力文字列の末尾から一致する文字を取り出し、文字列の残りの部分を返します。除去する文字は、引用符で囲んで単一の文字列として指定する必要があります。たとえば、ヘッダーの名前が TestLang で、*:/en_us:* がその値である場合、*HTTP.RES.HEADER (「TestLang」) .STRIP_START_CHARS (「:」)* は、ヘッダーの値の末尾から指定された文字を取り除き、一致しない最初の文字 s が見つかるまで:/_en_us を文字列を返します。

別の文字列に文字列を追加する

APPEND () 関数を使用すると、前述の関数によって返される値の文字列表現に、引数の文字列表現を追加できます。上記の関数は、数値、符号なし long、double、時刻値、IPv4 アドレス、または IPv6 アドレスを返す関数です。引数には、テキスト文字列、数値、符号なし long、double、時刻値、IPv4 アドレス、または IPv6 アドレスを使用できます。結果の文字列値は、+ 演算子を使用して取得した文字列値と同じです。

テキストに対する複雑な操作

July 19, 2022

単純な文字列照合に加えて、特定の文字列ではなく、文字列の長さとテキストブロックのパターンを調べる式を設定できます。

テキストベースの操作では、次の点に注意してください。

- 文字列引数をとる操作では、文字列は 255 文字を超えることはできません。
- 式に文字列を指定するときは、空白を含めることができます。

文字列の長さに対する操作

次の操作では、文字列を文字カウントで抽出します。

文字カウント操作	説明
<text>.TRUNCATE(<count>)	ターゲットの末尾を<count>の文字数で切り捨てた後の文字列を返します。文字列全体が<count>より短い場合、何も返されません。

文字カウント操作	説明
<text>.TRUNCATE(<character>,<count>)	<count>で指定された文字数だけ<character>の後のテキストを切り捨てた後の文字列を返します。
<text>.PREFIX(<character>,<count>)	<character>の出現回数が最も多い<count>ターゲット内で最も長いプレフィクスを選択します。
<text>.SUFFIX(<character>,<count>)	<character>の出現回数が最も多い<count>ターゲット内で最も長いサフィックスを選択します。たとえば、次のレスポンス本文を考えてみましょう peninsula。次の式は、sula: http.res.body(100).suffix('n',0)の値を返します。次の式が戻ります insula: http.res.body(100).suffix('n',1)。次の式は、peninsula: http.res.body(100).suffix('n',2)の値を返します。次の式は、peninsula: http.res.body(100).suffix('n',3)の値を返します。
<text>.SUBSTR(<starting_offset>, <length>)	ターゲットオブジェクトから<length>文字数を含む文字列を選択します。<starting_offset>の後の文字列の抽出を開始します。オフセットの後の文字数が<length>引数の値より少ない場合は、残りの文字をすべて選択します。
<text>.SKIP(<character>, <count>)	<character>の出現回数が最も多い<count>最長のプレフィックスをスキップした後に、ターゲットから文字列を選択します。

文字列の一部に対する操作

いずれかの操作を使用して大きな文字列のサブセットを抽出する方法については、文字列演算表を参照してください。

2 つの文字列の英数字の順序を比較する操作

COMPARE 操作は、2 つの異なる文字列の最初の不一致文字を検査します。この操作は、辞書の用語を並べ替えるときに使用される方法である辞書順に基づいています。

この演算は、比較された文字列の最初の一致しない文字の ASCII 値間の算術差を返します。次の相違点は例です。

- 「abc」と「and」の差は-1 です（3 番目のペアワイズ文字比較に基づく）。
- 「@」と「abc」の差は-33 です。

- 「1」と「abc」の差は-47 です。

COMPARE 操作の構文を以下に示します。

```
<text>.COMPARE(<string>)
```

テキストを表すバイト文字列から整数を抽出する

テキストを表すバイト文字列をバイト列として扱い、シーケンスから 8 ビット、16 ビット、または 32 ビットを抽出し、抽出したビットを整数に変換する方法については、整数抽出表を参照してください 。

テキストをハッシュ値に変換する

HASH 関数を使用して、テキスト文字列をハッシュ値に変換できます。この関数は、演算の結果として 31 ビットの正の整数を返します。式の形式は次のとおりです。

```
<text>.HASH
```

この関数は、大文字と小文字と空白を無視します。たとえば、演算後、2 つの文字列 Ab c と a bc は同じハッシュ値を生成します。

Base64 エンコーディングアルゴリズムを適用してテキストをエンコードおよびデコードする

次の 2 つの関数は、Base64 エンコーディングアルゴリズムを適用してテキスト文字列をエンコードおよびデコードします。

機能	説明
text.B64ENCODE	Base64 エンコーディングアルゴリズムを適用して、テキスト文字列 (text で指定) をエンコードします。
text.B64DECODE	Base64 デコードアルゴリズムを適用して、Base64 でエンコードされた文字列 (テキストで指定) をデコードします。テキストが B64 エンコード形式でない場合、この操作は UNDEF を生成します。

EXTEND 関数を使用して、書き換えアクションの検索を絞り込みます

EXTEND 関数は、パターンまたはパターンセットを指定し、HTTP パケットの本体をターゲットとする書き換えアクションで使用されます。パターン一致が見つかると、EXTEND 関数は、一致する文字列の両側の定義済みのバイト数だけ検索範囲を拡張します。その後、正規表現を使用して、この拡張領域の一致に対して書き換えを実行できます。EXTEND 関数で設定された書き換えアクションは、正規表現のみを使用して HTTP 本文全体を評価する書き換えアクションよりも速く書き換えを実行します。

EXTEND 関数の形式は EXTEND (m, n) です。ここで、m と n は、検索の範囲が一致パターンの前後に拡張されるバイト数です。一致が見つかると、新しい検索範囲は、一致する文字列の直前の m バイト、文字列自体、および文字列の後の n バイトで構成されます。正規表現を使用して、この新しい文字列の一部を書き換えることができます。

EXTEND 関数は、関数を使用する書き換えアクションが次の要件を満たしている場合にのみ使用できます。

- 検索は、パターンまたはパターンセット (正規表現ではない) を使用して実行されます。
- 書き換えアクションは、HTTP パケットの本体のみを評価します。

また、EXTEND 関数は、次のタイプの書き換えアクションでのみ使用できます。

- replace_all
- insert_after_all
- delete_all
- insert_before_all

たとえば、本文の最初の 1000 バイトの”http://exampleurl.com/” と”http://exampleurl.au/” のすべてのインスタンスを削除できます。これを行うには、文字列 exampleurl のすべてのインスタンスを検索し、一致が見つかったときに文字列の両側で検索範囲を拡張し、正規表現を使用して拡張領域で書き換えを実行する書き換えアクションを設定できます。次の例では、検索の範囲を、一致する文字列の左に 20 バイト、右に 50 バイト拡張します。

```
add rewrite action delurl_example delete_all 'HTTP.REQ.BODY(1000)'-pattern
exampleurl -refineSearch 'extend(20,50).regex_select(re##http://exampleurl
.(com|au)##)'
```

テキストを **16** 進数形式に変換する

次の関数は、テキストを 16 進数形式に変換し、結果の文字列を抽出します。

```
<text>.BLOB_TO_HEX(<string>)
```

たとえば、この関数はバイト文字列「abc」を「61:62:63」に変換します。

テキストの暗号化と復号化

既定の構文式では、ENCRYPT および DECRYPT 関数を使用してテキストを暗号化および復号化できます。特定の Citrix ADC アプライアンスまたは高可用性（HA）ペアの ENCRYPT 機能によって暗号化されたデータは、同じ Citrix ADC アプライアンスまたは HA ペアの DECRYPT 機能による復号化を目的としています。アプライアンスは、RC4、DES3、AES128、AES192、および AES256 暗号化方式をサポートしています。暗号化に必要なキー値は、ユーザーが指定できません。暗号化方式が設定されると、アプライアンスは指定された方式に適したランダムなキー値を自動的に生成します。デフォルトの方法は AES256 暗号化です。これは最も安全な暗号化方法であり、Citrix が推奨する暗号化方法です。

暗号化方式を変更する場合、またはアプライアンスが現在の暗号化方式に対して新しいキー値を生成する場合を除き、暗号化を設定する必要はありません。

注**:** XML ペイロードを暗号化および復号化することもできます。XML ペイロードを暗号化および復号化する関数については、XML ペイロードの暗号化と復号化を参照してください。

暗号化の構成

起動時に、アプライアンスは set ns encryptionParams コマンドをデフォルトで AES256 暗号化方式で実行し、AES256 暗号化に適したランダムに生成されたキー値を使用します。アプライアンスはキー値も暗号化し、暗号化されたキー値とともにコマンドを Citrix ADC 構成ファイルに保存します。したがって、AES256 暗号化方式は、ENCRYPT 関数と DECRYPT 関数に対してデフォルトで有効になっています。設定ファイルに保存されているキー値は、アプライアンスが再起動するたびにコマンドを実行しても、再起動後も保持されます。

暗号化方式を変更する場合、またはアプライアンスで現在の暗号化方式に対して新しいキー値を生成する場合は、set ns encryptionParams コマンドを手動で実行するか、構成ユーティリティを使用できます。CLI を使用して暗号化方式を変更するには、「例 **1:** 暗号化方式の変更」に示すように、**method** パラメータのみを設定します。アプライアンスで現在の暗号化方式に対して新しいキー値を生成する場合は、「例 **2**: 現在の暗号化方式に対する新しいキー値の生成」に示すように、**method** パラメータを現在の暗号化方式に設定し、**KeyValue** パラメータを空の文字列（”“）に設定します。新しいキー値を生成したら、設定を保存する必要があります。設定を保存しない場合、アプライアンスは次に再起動するまで新しく生成されたキー値のみを使用し、その後、保存された設定のキー値に戻ります。

GUI を使用した暗号化の設定

1. [システム] > **[**設定**]** に移動します。
2. [設定] 領域で、[暗号化パラメータの変更] をクリックします。
3. [暗号化パラメータの変更] ダイアログボックスで、次のいずれかの操作を行います。
   - 暗号化方式を変更するには、[方式] ボックスの一覧で、使用する暗号化方式を選択します。
   - 現在の暗号化方式の新しいキー値を生成するには、[選択した方法の新しいキーを生成] をクリックします。
4. 「**OK**」をクリックします。

暗号化関数と復号化関数を使用する

ENCRYPT 関数と DECRYPT 関数は、テキストを返す任意の式プレフィックスとともに使用できます。たとえば、クッキー暗号化のリライトポリシーで ENCRYPT 関数と DECRYPT 関数を使用できます。次の例では、書き換えアクションは、バックエンドサービスによって設定される myCookie という名前の Cookie を暗号化し、クライアントから返されたときに同じ Cookie を復号化します。

```
1 add rewrite action my-cookie-encrypt-action replace ”HTTP.RES.
    SET_COOKIE.COOKIE(”MyCookie”).VALUE(0)” ”HTTP.RES.SET_COOKIE.COOKIE(
    ”MyCookie”).VALUE(0).ENCRYPT” -bypassSafetyCheck YES
2
```



```
3 add rewrite action my-cookie-decrypt-action replace "HTTP.REQ.COOKIE.
    VALUE("MyCookie")" "HTTP.REQ.COOKIE.VALUE("MyCookie").DECRYPT" -
    bypassSafetyCheck YES
4 <!--NeedCopy-->
```

暗号化と復号化のポリシーを設定したら、設定を保存してポリシーを有効にします。

サードパーティ暗号化用の暗号化キーを構成する

デフォルトの構文式では、ENCRYPT および DECRYPT 関数を使用して、リクエストまたはレスポンスのテキストを暗号化および復号化できます。アプライアンス（スタンドアロン、高可用性、またはクラスタ）の ENCRYPT 機能によって暗号化されたデータは、同じアプライアンスによる DECRYPT 機能によって復号化されることを意図しています。アプライアンスは、RC4、DES、Triple-DES、AES92、AES256 暗号化方式をサポートしており、これらの各方式はデータの暗号化と復号化の両方に秘密鍵を使用します。これらの方法のいずれかを使用して、自己暗号化とサードパーティ暗号化の 2 つの方法でデータを暗号化および復号化できます。

アプライアンス（スタンドアロン、高可用性、またはクラスタ）の自己暗号化機能は、ヘッダー値を評価してデータを暗号化してから復号化します。これを理解する一例として、HTTP クッキーの暗号化があります。この式は、ヘッダーを評価し、送信応答の Set-Cookie ヘッダー内の HTTP Cookie 値を暗号化し、クライアントからの後続の着信要求の Cookie ヘッダーで返されたときに Cookie 値を復号化します。キー値はユーザー設定できません。代わりに、set ns encryptionParams コマンドで暗号化方式が設定されている場合、アプライアンスは設定された方式のランダムなキー値を自動的に生成します。デフォルトでは、このコマンドは AES256 暗号化方式を使用します。これは高度にセキュリティ保護された方法であり、この方法をお勧めします。

サードパーティの暗号化機能は、サードパーティアプリケーションを使用してデータを暗号化または復号化します。たとえば、クライアントはリクエスト内のデータを暗号化し、アプライアンスはバックエンドサーバーに送信する前にデータを復号化したり、その逆を行ったりします。これを実行するには、アプライアンスとサードパーティアプリケーションが秘密キーを共有する必要があります。アプライアンスでは、暗号化キーオブジェクトを使用して秘密キーを直接設定できます。キー値は、より強力な暗号化のためにアプライアンスによって自動的に生成されます。アプライアンスとサードパーティアプリケーションの両方がデータの暗号化と復号化に同じキーを使用できるように、サードパーティ製アプライアンスで同じキーを手動で設定します。

注: サードパーティの暗号化を使用して、XML ペイロードを暗号化および復号化することもできます。XML ペイロードを暗号化および復号化する関数については、「XML ペイロードの暗号化と復号化」を参照してください。

暗号メソッド

暗号方式は、平文のバイト列を暗号文バイト列に変換する暗号化機能と、暗号文を平文に戻す復号機能の 2 つの機能を提供する。暗号方式は、鍵と呼ばれるバイトシーケンスを使用して、暗号化と復号化を実行します。暗号化と復号に同じ鍵を使用する暗号方式は、対称と呼ばれます。暗号化と復号化に異なるキーを使用する暗号方式は非対称です。非対称暗号の最も注目すべき例は公開鍵暗号法である。公開鍵暗号では、暗号化のために誰でも利用できる公開鍵と、復号機のみが知っている秘密鍵を使用する。

優れた暗号方式は、鍵を持っていない場合、暗号文を解読（「クラック」）することが不可能になります。「実行不可能」とは、サイファーテキストをクラックすると、価値よりも多くの時間とコンピューティングリソースがかかることを意味します。コンピュータがより強力で安価になるにつれて、以前はクラックが不可能であった暗号がより実現可能になります。また、時間が経つにつれて、暗号メソッド（またはその実装）に欠陥が見つかり、クラッキングが容易になります。したがって、古い暗号方式よりも新しい暗号方式が優先されます。一般に、長い長さのキーは、短いキーよりもセキュリティが優れていますが、暗号化と復号化の時間が長くなります。

暗号方式は、ストリーム暗号またはブロック暗号を使用できます。RC4 は、ほとんどセキュリティで保護されたストリーム暗号であり、レガシーアプリケーションにのみ使用されます。ブロック暗号にはパディングを含めることができます。

ストリーム暗号

ストリーム暗号方式は、個々のバイトに対して動作します。Citrix ADC アプライアンスで使用できるストリーム暗号は 1 つだけです。RC4 は、128 ビット（16 バイト）のキー長を使用します。与えられたキーに対して、RC4 はバイトの擬似乱数シーケンスを生成し、キーストリームを呼び出す。キーストリームは、暗号文を生成するために平文と X 論理変換される。RC4 はもはや安全とは見なされず、レガシーアプリケーションで必要な場合にのみ使用すべきです。

ブロック暗号

ブロック暗号方式は、固定されたバイトブロックで動作します。Citrix ADC アプライアンスは、データ暗号化標準（DES）と高度な暗号化標準（AES）の 2 つのブロック暗号を提供します。DES は 8 バイトのブロックサイズと(Citrix ADC アプライアンスでは）キーの長さの 2 つの選択肢を使用します。64 ビット（8 バイト）（そのうち 56 ビットはデータ、8 ビット）はパリティであり、トリプル DES は 192 ビット（24 バイト）のキー長です。AES のブロックサイズは 16 バイトで、（Citrix ADC では）キーの長さには 128 ビット（16 バイト)、192 ビット（24 バイト)、および 256 ビット（32 バイト）の 3 つの選択肢があります。

パディング

ブロック暗号のプレーンテキストがブロックの整数でない場合は、より多くのバイトのパディングが必要になることがあります。例えば、平文が「xyzzy」（16 進数 78797a7a79）であるとする。8 バイトの Triple-DES ブロックの場合、8 バイトを作成するには、この値をパディングする必要があります。パディングスキームは、復号化関数が復号後の元の平文の長さを決定できるようにする必要があります。現在使用されているパディングスキームをいくつか次に示します (n は追加されたバイト数です)。

- PKCS7: n バイトの値を n 個ずつ加算します。たとえば、78797a7a79030303。これは、OpenSSL と ENCRYPT () ポリシー関数で使用されるパディングスキームです。PKCS5 のパディング方式は PKCS7 と同じです。
- ANSI X.923: n-1 個のゼロバイトと値 n の最後のバイトを加算します。たとえば、78797a7a79000003。
- ISO 10126: n-1 個のランダムなバイトと値 n の最後のバイトを加算する。たとえば、78797a7a79xxxx03 と指定します。xx には任意のバイト値を指定できます。DECRYPT () ポリシー関数はこのパディングスキームを

受け入れ、PKCS7 および ANSI X.923 スキームを受け入れることもできます。

- ISO/IEC 7816-4:0x80 バイトと n-1 ゼロバイトを追加します。たとえば、78797a7a79800000。これは OneAndZeroS パディングとも呼ばれます。
- ゼロ:n 個のゼロバイトを追加します。例：78797a7a79000000。これは NUL バイトを含まないプレーンテキストでのみ使用できます。

パディングが使用され、平文がブロックの整数である場合、復号関数が元の平文の長さを明確に決定できるように、通常、余分なブロックが追加されます。PCKS7 と 8 バイトブロックの場合、これは 0808080808080808 になります。

動作モード

ブロック暗号にはさまざまな動作モードがあり、平文の複数のブロックをどのように暗号化するかを指定します。一部のモードは、暗号化プロセスを開始するために使用される平文とは別のデータのブロックである初期化ベクトル (IV) を使用します。暗号化ごとに異なる IV を使用することをお勧めします。そうすれば、同じ平文でも異なる暗号文が生成されます。IV は秘密である必要はないので、暗号文の前に付加されます。モードには次のものが含まれます。

- 電子コードブック (ECB): 平文の各ブロックは個別に暗号化されます。IV は使用されません。プレーンテキストが暗号ブロックサイズの倍数でない場合は、パディングが必要です。同じ平文と鍵は、常に同じ暗号文を生成します。このため、ECB は他のモードよりも安全性が低いと考えられ、レガシーアプリケーションにのみ使用する必要があります。
- 暗号ブロック連鎖（CBC）：平文の各ブロックは、暗号化される前に、前の暗号文ブロック、または最初のブロックの IV と XOR 化されます。プレーンテキストが暗号ブロックサイズの倍数でない場合は、パディングが必要です。これは、Citrix ADC EncryptionParams メソッドで使用されるモードです。
- Cipher Feedback (CFB): 前の暗号文ブロック、または最初のブロックの IV は暗号化され、出力は現在の平文ブロックと XOR され、現在の暗号文ブロックが作成されます。フィードバックは、1 ビット、8 ビット、または 128 ビットです。平文は暗号文と XOR されるので、パディングは不要である。
- 出力フィードバック (OFB): キーストリームは、暗号を連続的に IV に適用し、キーストリームブロックを平文で XOR することによって生成されます。パディングは必要ありません。

サードパーティ暗号化用の暗号化キーを構成する

次に、暗号化キーの設定で実行される設定タスクを示します。

1. 暗号化キーを追加する。指定された暗号方式の暗号キーを、指定されたキー値で設定します。
2. 暗号化キーの変更。設定済みの暗号化キーのパラメータを編集できます。
3. 暗号化キーの設定解除。設定済みの暗号化キーのパラメータをデフォルト値に設定します。名前を持つ EncryptionKey 値が存在する必要があります。パディングを DEFAULT (メソッドで決定) に設定し、既存の IV を削除します。これにより、ENCRYPT () はランダムな IV を生成します。既存のコメントを削除します。メソッドとキー値はリセットできません。
4. 暗号化キーを削除する。設定されている暗号化キーを削除します。キーには参照を設定できません。

5. 暗号化キーを表示します。設定された暗号化キーまたはすべての設定済みキーのパラメータを表示します。名前を省略すると、キー値は表示されません。

CLI を使用して暗号化キーを追加する

コマンドプロンプトで入力します。

```
add ns encryptionKey <name> -method <method> [-keyValue <keyvalue>][-
padding (OFF | ON)] [-iv <hexstring>] -keyValue <keyvalue> [-comment <
string>]
```

各項目の意味は次のとおりです。

```
1 <method> = ( NONE | RC4 | DES3 | AES128 | AES192 | AES256 | DES | DES-
     CBC | DES-CFB | DES-OFB | DES-ECB | DES3-CBC | DES3-CFB | DES3-OFB |
      DES3-ECB | AES128-CBC | AES128-CFB | AES128-OFB | AES128-ECB |
     AES192-CBC | AES192-CFB | AES192-OFB | AES192-ECB | AES256-CBC |
     AES256-CFB | AES256-OFB | AES256-ECB ) <hexstring> = hex-encoded
     byte sequence
2 <!--NeedCopy-->
```

上記の暗号化方式は、CBC をデフォルトの動作モードとする動作モードを指定します。したがって、DES、DES2、AES128、AES192、AES256 方式は、DES-CBC、DES3-CBC、AES128-CBC、AES192-CBC、AES256-CBC 方式と同等である。

CLI を使用して暗号化キーを変更する

コマンドプロンプトで入力します。

```
set ns encryptionKey <name> [-method <method>] [-keyValue <keyvalue>] [-
padding ( OFF | ON )] [-iv <string>] [-comment <string>]
```

CLI を使用して暗号化キーを設定解除する

コマンドプロンプトで入力します。

```
unset ns encryptionKey <name> [-padding] [-iv] [-comment]
```

CLI を使用して暗号化キーを削除する

コマンドプロンプトで入力します。

```
rm ns encryptionKey <name>
```


CLI を使用して暗号化キーを表示する

コマンドプロンプトで入力します。

例**:**

```
show ns encryptionKey [<name>]

add ns encryptionKey my_key -method aes256 -keyValue 26
    ea5537b7e0746089476e5658f9327c0b10c3b4778c673a5b38cee182874711  - iv
    c2bf0b2e15c15004d6b14bcdc7e5e365
set ns encryptionKey my_key -keyValue
    b8742b163abcf62d639837bbee3cef9fb5842d82d00dfe6548831d2bd1d93476
unset ns encryptionKey my_key -iv
rm ns encryptionKey my_key
show ns encryptionKey my_key
Name: my_key
Method: AES256
Padding: DEFAULT
Key Value: (not disclosed)
<!--NeedCopy-->
```

GUI を使用して暗号化キーを追加する

[システム] > [暗号化キー] に移動し、[追加] をクリックして暗号化キーを作成します。

GUI を使用して暗号化キーを変更する

[システム] > [暗号化キー] に移動し、[編集] をクリックして、設定された暗号化キーのパラメータを変更します。

GUI を使用して暗号化キーを削除する

[システム] > [暗号化キー] に移動し、[削除] をクリックします。

サードパーティ暗号化用の暗号化および復号化機能

以下は、サードパート暗号化に使用される ENCRYPT 関数です。

```
ENCRYPT (encryptionKey, out_encoding)
```

各項目の意味は次のとおりです。

アプライアンスの入力データは、暗号化されるテキストです。

encryptionKey: 暗号化方法、秘密キーの値、およびその他の暗号化パラメータを提供するために、設定された暗号化キーオブジェクトを指定するオプションの文字列パラメータ。省略した場合、このメソッドは set ns encryptionParams コマンドに関連付けられた自動生成されたキー値を使用します。

out_encoding: この値は、出力のエンコード方法を指定します。省略すると、BASE64 エンコーディングが使用されます。

入力:

```
BASE64: original PEM base64-encoding: 6 bits (0..63) encoded as one
    ASCII character:
                  0..23 = 'A'..'Z', 24..51 = 'a'..'z', 52..61 = '0'..'9
                      ', 62 = '+', 63 = '/', '=' = pad byte.
       BASE64URL:  URL and Filename safe base64-encoding: same as BASE64
            except 62 = '-', 63 = '_'
       HEX_UPPER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '.
       HEX_LOWER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'a'..'f
           '.
       HEX_COLONS: Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '; ':' between each hex byte. Matches BLOB_TO_HEX() output
           format
       HEX:  For input, accepts HEX_UPPER, HEX_LOWER, and HEX_COLONS
           format. For output, produces HEX_LOWER format
<!--NeedCopy-->
```

出力: 出力は、指定されたメソッドとキーを使用して暗号化され、指定された出力エンコーディングを使用してエンコードされたテキストです。IV を必要とするブロックメソッドおよびモードの暗号化されたテキストの前に生成された IV を挿入し、encryptionKey に IV が指定されていないか、または encryptionKey が省略されます。

以下は、サードパーティの復号化に使用される DECRYPT 関数です。

```
DECRYPT(encryptionKey, in_encoding)
```

各項目の意味は次のとおりです。

入力データは、指定されたメソッドを使用した暗号化されたテキストであり、指定された入力エンコーディングを使用してエンコードされたキーです。このテキストは、IV を必要とするブロックメソッドおよびモードの暗号化されたテキストの前に生成された IV を含むことが想定され、encryptionKey に IV が指定されていないか、または encryptionKey が省略されます。

encryptionKey: 暗号化方式、秘密鍵、およびその他の暗号化パラメータを提供するために、設定された encryptionKey オブジェクトを指定するオプションの文字列パラメータ。省略すると、encryptionParams 設定に関連付けられたメソッドと自動生成されたキーが使用されます。

in_encoding-入力のエンコード方法を指定するオプションの列挙パラメータ。これらの値は、ENCRYPT の out_encoding と同じです。省略すると、BASE64 エンコーディングが想定されます。

出力データは、エンコードされていない復号化されたテキストです。

バリアントとオプションのパラメータ

以下に、これらの関数のバリアントとオプションのパラメータを示します。

バリアント	説明
ENCRYPT	encryptionParams コマンドと BASE64 出力エンコーディングパラメータを使用します。
ENCRYPT (out_encoding)	EncryptionParams を使用し、出力エンコーディングパラメータを指定します。
ENCRYPT(encryptionKey)	指定した encryptionKey および BASE64 出力エンコーディングパラメータを使用します。
ENCRYPT(encryptionKey, out_encoding)	指定した encryptionKey と出力エンコーディングパラメーターを使用します。
DECRYPT	encryptionParams コマンドと BASE64 入力エンコーディングパラメータを使用します。
DECRYPT (out_encoding)	encryptionParams コマンドと指定された入力エンコーディングパラメータを使用します。
DECRYPT(encryptionKey)	指定した encryptionKey および BASE64 入力エンコーディングパラメータを使用します。
DECRYPT(encryptionKey, out_encoding)	指定した encryptionKey と入力エンコーディングパラメータを使用します。

HMAC キーの設定

Citrix ADC アプライアンスは、メッセージ送信者とメッセージ受信者の間で共有される秘密鍵を使用して、入力テキストのダイジェスト方法またはハッシュを計算するハッシュメッセージ認証コード（HMAC）関数をサポートします。ダイジェスト方式 (RFC 2104 手法から派生したもの) は、送信者を認証し、メッセージの内容が変更されていないことを確認します。たとえば、クライアントが共有 HMAC キーを含むメッセージを Citrix ADC アプライアンスに送信すると、詳細（PI）ポリシー式は HMAC 関数を使用して、選択したテキストのハッシュベースのコードを計算します。次に、受信者が秘密鍵を含むメッセージを受信すると、HMAC を元の HMAC と比較して再計算し、メッセージが改ざんされているかどうかを判断します。HMAC 機能は、スタンドアロンアプライアンス、および高可用性構成またはクラスタ内のアプライアンスによってサポートされます。これを使用することは、暗号化キーの設定に似ています。

add ns hmackey コマンドと set ns hmackey コマンドには、HMAC 計算に使用するダイジェスト方式と共有秘

密鍵を指定するパラメータが含まれています。

HMAC キーを設定するには、次の手順を実行する必要があります。

1. HMAC キーの追加。指定されたキー値で HMAC キーを設定します。
2. HMAC キーの変更。設定された HMAC キーのパラメータを変更します。ダイジェストメソッドは、キー値の長さがダイジェストによって決定されないため、キー値を変更することなく変更できます。ただし、ダイジェストを変更するときは、新しいキーを指定することをお勧めします。
3. HMAC キーの設定を解除します。設定された HMAC キーのパラメータをデフォルト値に設定します。という名前の HMAckEy オブジェクトが存在する必要があります。設定解除できるパラメータはコメントのみです。コメントは削除されます。
4. HMAC キーを削除する。設定済みのキーを削除します。キーには参照を設定できません。
5. HMAC キーを表示します。設定された HMAC AC キーまたはすべての設定済みキーのパラメータを表示します。名前を省略すると、キー値は表示されません。

一意でランダムな **HMAC** キーを構成する

一意の HMAC キーを自動的に生成できます。アプライアンスがクラスタ構成の場合、HMAC キーはプロセスの開始時に生成され、すべてのノードとパケットエンジンに配布されます。これにより、HMAC キーは、クラスタ内のすべてのパケットエンジンとすべてのノードで同じになります。

コマンドプロンプトで入力します。

```
add ns hmacKey <your_key> -digest <digest> -keyValue <keyvalue>
```

例**:**

```
add ns hmacKey <name> -digest sha1 -keyValue AUTO
```

各項目の意味は次のとおりです。

- 名前の構文は正しく、既存のキーの名前と重複しません。
- 「AUTO」KeyValue を設定コマンドで使用すると、既存の encrytionKey オブジェクトと HMAckEy オブジェクトの新しいキーを生成できます。

> 注意**:**
>
> 自動キー生成は、Citrix ADC アプライアンスがキーを使用してデータを暗号化および復号化する場合、または HMAC キーの生成と検証を行う場合に便利です。キー値自体は表示時にすでに暗号化されているため、生成されたキー値を取得して他の当事者が使用することはできません。

例**:**

```
add ns hmacKey my_hmac_key -digest sha1 -keyValue 0c753c6c5ef859189cacdf95b506d02c179
```

上記の暗号化方式は、CBC をデフォルトの動作モードとする動作モードを指定します。したがって、DES、DES2、AES128、AES192、AES256 方式は、DES-CBC、DES3-CBC、AES128-CBC、AES192-CBC、AES256-CBC 方式と同等である。

CLI を使用した HMAC キーの変更

このコマンドは、HMAC キーに設定されたパラメータを変更します。キー値の長さはダイジェストによって決定されないため、キー値を変更せずにダイジェストを変更できます。ただし、ダイジェストを変更するときは、新しいキーを指定することをお勧めします。コマンドプロンプトで入力します。

```
set ns hmacKey <name> [-digst <digest>] [-keyValue <keyvalue>]
[-comment <string>]

<!--NeedCopy-->
```

CLI を使用した HMAC キーの設定解除

このコマンドは、HMAC キー用に設定されたパラメータをデフォルト値に設定します。という名前の HMAckEy オブジェクトが存在する必要があります。設定解除できる唯一のパラメータは comment オプションで、これは削除されます。コマンドプロンプトで入力します。

```
unset ns hmacKey <name> -comment
```

CLI を使用して HMAC キーを削除する

このコマンドは、設定された hmac キーを削除します。キーは参照を持つことができません。コマンドプロンプトで入力します。

```
rm ns hmacKey <name>
```

CLI を使用して HMAC キーを表示する

コマンドプロンプトで入力します。

```
show ns encryptionKey [<name>]

add ns hmacKey my_hmac_key -digest sha1 -keyValue 0
    c753c6c5ef859189cacdf95b506d02c1797407d
set ns hmacKey my_hmac_key -keyValue
    f348c594341a840a1f641a1cf24aa24c15eb1317
rm ns hmacKey my_hmac_key
show ns hmacKey my_hmac_key
           Name: my_hmac_key
     Digest: SHA1
     Key Value: (not disclosed)
```



```
<!--NeedCopy-->
```

高度なポリシー式: 日付、時刻、および数値の操作

October 7, 2021

Citrix ADC アプライアンスが処理するほとんどの数値データは、日付と時刻で構成されます。アプライアンスは、日付と時刻の操作に加えて、HTTP 要求や応答の長さなど、その他の数値データを処理します。このデータを処理するには、番号を処理する高度なポリシー式を設定できます。

数値式は、数値を返す式のプレフィックスで構成されます。場合によっては、数値に対して操作を実行できる演算子です。数値を返す式のプレフィックスの例は`SYS.TIME.DAY`、`HTTP.REQ.CONTENT_LENGTH` `HTTP.RES.BODY.LENGTH`。 `Numeric`演算子はデータを数値形式で返すプレフィックス式で機能します。たとえば、`GT(<int>)` 演算子は、整数を返す任意のプレフィックス式（HTTP.REQ.CONTENT_LENGTH など）で使用できます。

式内の日付と時刻の形式

October 7, 2021

日付と時刻（Citrix ADC システム時刻や SSL 証明書の日付など）に対応するポリシーで高度なポリシー式を構成する場合は、次のように時間形式を指定します。

`GMT|LOCAL [<yyyy>] [<month>] [<d>] [<h>] [<m>] [<s>]`

各項目の意味は次のとおりです:

- <yyyy> は、GMT または LOCAL の後の 4 桁の年です。
- <month> は、月を表す 3 文字の省略形です (たとえば、Jan、Dec など)。
- <d> は曜日または日付の整数です。

  月曜日、火曜日などに曜日を指定することはできません。月の特定の日に整数を指定するか、月の第 1、第 2、第 3 の曜日として日付を指定します。曜日を指定する例を次に示します。

  - Sun_1 は月の第 1 日曜日を指定します。
  - Sun_3 は月の第 3 日曜日を指定します。
  - Wed_3 は月の第 3 水曜日を指定します。
  - 30 は、月の正確な日付の例です。
- <h> は時間（たとえば、10h）です。
- <s> は秒数（30 秒など）です。

次の式は、日付が 2008 年 1 月から 2009 年 1 月の間 (GMT に基づく) である場合に true になります。

```
http.req.date.between(GMT 2008 Jan, GMT 2009 Jan)
```

次の式は、GMT に基づいて、3 月および暦年の 3 月に続くすべての月に対して真です。

```
sys.time.ge(GMT 2008 Mar)
```

日付と時刻を指定する場合は、大文字と小文字が区別され、エントリ間の空白の正確な数を保持する必要があることに注意してください。

```
**Note:**

In an expression that requires two time values, both must use GMT or
    both must use LOCAL. You cannot mix the two in an expression.

Unlike when you use the SYS.TIME prefix in an advanced policy
    expression, if you specify SYS.TIME in a rewrite action, the Citrix
    ADC returns a string in conventional date format (for example, Sun,
    06 Nov 1994 08:49:37 GMT). For example, the following rewrite action
     replaces the http.res.date header with the Citrix ADC system time
    in a conventional date format:

add rewrite action sync_date replace http.res.date sys.time
```

Citrix ADC システム時刻の式

October 7, 2021

SYS.TIME 式プレフィックスは、Citrix ADC のシステム時刻を抽出します。Citrix ADC システム時間に応じて、特定のイベントが特定の時間に発生したか、特定の時間範囲内に発生したかを示す式を設定できます。

次の表に、SYS.TIME プレフィックスを使用して作成できる式を示します。

- **SYS.TIME.BETWEEN(<time1>, <time2>):**

  戻り値がより遅い場合、ブール型 TRUE を返します <time1> <time2>。

  <time1>、<time2> 引数を次のようにフォーマットします。

  - 両方とも GMT または両方の LOCAL である必要があります。
  - <time2> は <time1> より後の値である必要があります。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、月の第 1 日曜日である場合は、次のように指定できます。

- sys.time.between(GMT 2004, GMT 2006)
- sys.time.between(GMT 2004 Jan, GMT 2006 Nov)
- sys.time.between(GMT 2004 Jan, GMT 2006)
- sys.time.between(GMT 2005 May Sun_1, GMT 2005 May Sun_3)
- sys.time.between(GMT 2005 May 1, GMT May 2005 1)
- sys.time.between(LOCAL 2005 May 1, LOCAL May 2005 1)

- **SYS.TIME.DAY:**

  月の現在の日を 1 ～31 の数値として返します。

- **SYS.TIME.EQ(<time>):**

  現在の時刻が <time> 引数と等しい場合は、ブール型 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、その月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.eq(GMT 2005) (この例では TRUE。)
  - sys.time.eq(GMT 2005 Dec) (この例では FALSE。)
  - sys.time.eq (LOCAL 2005 May) (この例では、現在のタイムゾーンに応じて TRUE または FALSE に評価されます。)。
  - sys.time.eq(GMT 10h) (この例では TRUE。)
  - sys.time.eq(GMT 10h 30s) (この例では TRUE。)
  - sys.time.eq(GMT May 10h) (この例では TRUE。)
  - sys.time.eq(GMT Sun) (この例では TRUE。)
  - sys.time.eq(GMT May Sun_1) (この例では TRUE。)

- **SYS.TIME.NE(<time>):**

  現在の時刻が <time> 引数と等しくない場合に、ブール型 TRUE を返します。

- **SYS.TIME.GE(<time>):**

  現在の時刻が <time> 以降または等しい場合に、ブール型 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、その月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.ge(GMT 2004) (この例では TRUE。)
  - sys.time.ge(GMT 2005 Jan) (この例では TRUE。)
  - sys.time.ge(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - sys.time.ge (GMT 8h) (この例では TRUE)
  - sys.time.ge (GMT 30m) (この例では FALSE)
  - sys.time.ge(GMT May 10h)(この例では TRUE)
  - sys.time.ge(GMT May 10h 0m)(この例では TRUE)
  - sys.time.ge(GMT Sun)(この例では TRUE)

- sys.time.ge(GMT May Sun_1) (この例では TRUE)

- **SYS.TIME.GT(<time>):**

  時間値が <time> 引数より後の場合は、ブール型 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、その月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.gt(GMT 2004) (この例では TRUE)
  - sys.time.gt(GMT 2005 Jan) (この例では TRUE)
  - sys.time.gt(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - sys.time.gt(GMT 8h) (この例では TRUE)
  - sys.time.gt(GMT 30m) (この例では FALSE)
  - sys.time.gt(GMT May 10h) (この例では FALSE)
  - sys.time.gt(GMT May 10h 0m)(この例では TRUE)
  - sys.time.gt(GMT Sun) (この例では FALSE)
  - sys.time.gt(GMT May Sun_1) (この例では FALSE)

- **SYS.TIME.HOURS:**

  現在の時間を 0 ～23 の整数で返します。

- **SYS.TIME.LE(<time>):**

  現在の時間値が <time> 引数に先行するまたは等しい場合は、ブール型 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、その月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - システムタイム.le (GMT 2006) (この例では TRUE)
  - sys.time.le(GMT 2005 Dec) (この例では TRUE)
  - sys.time.le(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - sys.time.le (GMT 8 h) (この例では FALSE)
  - sys.time.le (GMT 30m) (この例では TRUE)
  - sys.time.le(GMT May 10h) (この例では TRUE。)
  - sys.time.le(GMT Jun 11h) (この例では TRUE。)
  - sys.time.le(GMT Wed)(この例では TRUE)
  - sys.time.le(GMT May Sun_1) (この例では TRUE)

- **SYS.TIME.LT(<time>):**

  現在の時間値が <time> 引数に先行する場合は、ブール型 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、その月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧内に表示されます)。

  - sys.time.lt(GMT 2006) (この例では TRUE。)
  - sys.time.lt.time.lt(GMT 2005 Dec) (この例では TRUE。)

- sys.time.lt(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
- sys.time.lt (GMT 8 h) (この例では FALSE を返します)。
- sys.time.lt(GMT 30m) (この例では TRUE。)
- sys.time.lt(GMT May 10h)(この例では FALSE)
- sys.time.lt(GMT Jun 11h) (この例では TRUE。)
- sys.time.lt(GMT Wed) (この例では TRUE。)
- sys.time.lt(GMT May Sun_1) (この例では FALSE。)

- **SYS.TIME.MINUTES:**

  現在の分を 0 ～59 の整数で返します。

- **SYS.TIME.MONTH:**

  現在の月を抽出し、1 (1 月) ～12 (12 月) の整数を返します。

- **SYS.TIME.RELATIVE_BOOT:**

  直近またはスケジュールされた再起動までの秒数を計算し、整数を返します。

  最も近いブート時間が過去のものである場合、整数は負の値になります。将来の場合、整数は正です。

- **SYS.TIME.RELATIVE_NOW:**

  Citrix ADC 現在のシステム時刻から指定した時刻までの秒数を計算し、差を示す整数を返します。

  指定した時刻が過去の場合、整数は負の値になります。将来の場合、整数は正の値になります。

- **SYS.TIME.SECONDS:**

  現在の Citrix ADC システム時刻から秒を抽出し、その値を 0～59 の整数として返します。

- **SYS.TIME.WEEKDAY:**

  現在の平日を 0 (日曜日) ～6 (土曜日) の値で返します。

- **SYS.TIME.WITHIN (<time1>, <time2>):**

  <time1> たとえば、日や時間などの時間要素を省略すると、その範囲内で最も低い値を持つと見なされます。でエレメントを省略すると <time2>、そのエレメントは範囲内で最も高い値を持つと見なされます。

  時間の要素の範囲は、月 1-12、日 1-31、平日 0-6、時間 0-23、分 0-59、秒 0-59 です。年を指定する場合は、<time1> との両方で年を指定する必要があります <time2>。

  たとえば、時刻が GMT 2005 5 月 10 日 10h 15m 30 秒で、月の第 2 火曜日である場合、次の値を指定できます (評価結果は括弧内に表示されます)。

  - sys.time.within(GMT 2004, GMT 2006)(この例では TRUE)
  - sys.time.within(GMT 2004 Jan, GMT 2006 Mar)(FALSE、5 月は 1 月～3 月の範囲外です)。
  - sys.time.within(GMT Feb, GMT)(TRUE, 5 月は 2 月から 12 月の範囲です。)
  - sys.time.within(GMT Sun_1, GMT Sun_3)(TRUE、第 2 火曜日は第 1 日曜日と第 3 日曜日の間です。)

  - sys.time.within(GMT 2005 May 1 10h, GMT May 2005 1 17h)（この例では、TRUE です)。
  - sys.time.within(LOCAL 2005 May 1, LOCAL May 2005 1) (Citrix ADC システムのタイムゾーンに応じて、TRUE または FALSE。)

- **SYS.TIME.YEAR:**

  現在のシステム時刻から年を抽出し、その値を 4 桁の整数として返します。

SSL 証明書の日付の式

October 7, 2021

次のプレフィックスを含む式を設定することで、SSL 証明書の有効期間を判断できます。

```
CLIENT.SSL.CLIENT_CERT
```

次の例では、有効期限の特定の時刻を証明書内の情報と一致させます。

```
client.ssl.client_cert.valid_not_after.eq(GMT 2009)
```

次の表に、SSL 証明書に対する時間ベースの操作を示します。必要な式を取得するには、最初の列の式の 証明書を、プレフィックス式「CLIENT.SSL.CLIENT_CERT」に置き換えます。

- <certificate>.VALID_NOT_AFTER:

  証明書の有効期限の最後の日を返します。戻り値の形式は、1970 年 1 月 1 日からの秒数です（0 時間、0 分、0 秒)。

- **<certificate>.VALID_NOT_AFTER.BETWEEN(<time1>, <time2>):**

  証明書の有効性が <time1><time2> 引数と引数の間にある場合は、ブール型 TRUE 値を返します。<time1> との両方 <time2> を完全に指定する必要があります。次に例を示します。

  GMT 1995 Jan は完全に指定されています.

  GMT Jan が完全には指定されていません

  GMT 1995 20 は完全に指定されていません。

  GMT Jan Mon_2 が完全に指定されていません。

  <time1> および <time2> 引数は、GMT または LOCAL の両方 <time2> で指定し、より大きい値を指定する必要があります <time1>。

  たとえば、GMT 2005 5 月 1 日 10h 15m 30 秒で、月の第 1 日曜日を指定する場合は、次のように指定できます (評価結果は括弧で囲まれています)。

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)

- . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
- . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
- . . .between(LOCAL 2005 May 1, LOCAL May 2005 1) (Citrix ADC システムタイムゾーンに応じて TRUE または FALSE。)

- **<certificate>.VALID_NOT_AFTER.DAY:**

  証明書が有効である月の最終日を抽出し、日付に応じて 1 ～31 の数値を返します。

- **<certificate>.VALID_NOT_AFTER.EQ (<time>):**

  時間が <time> 引数と等しい場合は、ブール型 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、月の第 1 日曜日である場合は、次のように指定できます (この例の評価結果は括弧で囲まれています)。

  - . . .eq(GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . . .eq(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - . . .eq(GMT 10h) (TRUE)
  - . . .eq(GMT 10h 30s) (TRUE)
  - . . .eq(GMT May 10h) (TRUE)
  - . . .eq(GMT Sun) (TRUE)
  - . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GE(<time>):**

  時間値が <time> 引数以上の場合、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧で囲まれています)。

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GT (<time>):**

  時間値が <time> 引数より大きい場合は、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧で囲まれています)。

- . . .gt(GMT 2004) (TRUE)
- . . .gt(GMT 2005 Jan) (TRUE)
- . . .gt(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
- . . .gt(GMT 8h) (TRUE)
- . . .gt(GMT 30m) (FALSE)
- . . .gt(GMT May 10h) (FALSE)
- . . .gt(GMT Sun) (FALSE)
- . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.HOURS:**

  証明書が有効である直近の 1 時間を抽出し、その値を 0 ～23 の整数として返します。

- **<certificate>.VALID_NOT_AFTER.LE(<time>):**

  時間が <time> 引数に先行するまたは等しい場合は、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧で囲まれています)。

  - . . .le(GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - . . .le(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .le(GMT 8h) (FALSE)
  - . . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.LT(<time>):**

  時間が <time> 引数に先行する場合は、ブール型 TRUE を返します。

  たとえば、現在の時刻が GMT 2005 5 月 1 日 10h 15m 30 秒で、月の第 1 日曜日である場合は、次のように指定できます。

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.MINUTES:**

  証明書が有効である最後の分を抽出し、その値を 0 ～59 の整数として返します。

- **<certificate>.VALID_NOT_AFTER.MONTH:**

  証明書が有効である最後の月を抽出し、その値を 1 (1 月) ～12 (12 月) の整数として返します。

- **<certificate>.VALID_NOT_AFTER.Relative_BOOT:**

  直近またはスケジュールされた再起動までの秒数を計算し、整数を返します。最も近いブート時間が過去のものである場合、整数は負の値になります。将来の場合、整数は正です。

- **<certificate>.VALID_NOT_AFTER.RELATIVE_NOW;**

  現在のシステム時刻から指定された時刻までの秒数を計算し、整数を返します。時刻が過去にある場合、整数は負の値になります。将来の場合、整数は正の値になります。

- **<certificate>.VALID_NOT_AFTER.SECONDS:**

  証明書が有効である最後の秒を抽出し、その値を 0 ～59 の整数として返します。

- **<certificate>.VALID_NOT_AFTER.WEEKDAY:**

  証明書が有効である最後の平日を抽出します。0 (日曜日) から 6 (土曜日) までの数値を返します。

- **<certificate>.VALID_NOT_AFTER.WITHIN(<time1>, <time2>):**

  時刻が <time1> およびの要素によって定義されたすべての範囲内にある場合、ブール型 (Boolean) の値を返します <time2>。

  <time1> から時間の要素を省略すると、その範囲内で最小値を持つと見なされます。から要素を省略すると <time2>、その範囲の最大値を持つと見なされます。で年を指定 <time1> する場合は、で年を指定する必要があります <time2>。

  時間の要素の範囲は、月 1-12、日 1-31、平日 0-6、時間 0-23、分 0-59、秒 0-59 です。結果が TRUE になるには、時刻の各要素が、で指定した対応する範囲に存在している必要があります <time1> <time2>。

  たとえば、時刻が GMT 2005 5 月 10 日 10h 15m 30 秒で、月の第 2 火曜日である場合、次のように指定できます (評価結果は括弧で囲まれています)。

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, 5 月は 1 月から 3 月の範囲ではありません。)
  - . . .within(GMT Feb, GMT) (TRUE, 5 月は 2 月～12 月の範囲内)
  - . . .within(GMT Sun_1, GMT Sun_3) (TRUE, 第 2 火曜日は第 1 日曜日から第 3 日曜日の範囲内にある)
  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (Citrix ADC システムのタイムゾーンに応じて、TRUE または FALSE)

- **<certificate>.** 年後無効**:**

  証明書が有効である最後の年を抽出し、4 桁の整数を返します。

- **<certificate>.VALID_NOT_BEFORE:**

  クライアント証明書が有効になる日付を返します。

  戻り値の形式は、1970 年 1 月 1 日からの秒数です（0 時間、0 分、0 秒）。

- **<certificate>.VALID_NOT_BEFORE.BETWEEN(<time1>, <time2>):**

  時間値が 2 つの時間引数の間にある場合、ブール型 TRUE を返します。<time1> <time2> 引数と引数の両方を完全に指定する必要があります。

  次に例を示します。

  GMT 1995 Jan は完全に指定されています.
  GMT Jan が完全には指定されていません。
  GMT 1995 20 は完全に指定されていません。
  GMT Jan Mon_2 が完全に指定されていません。
  time 引数は GMT または LOCAL の両方で、<time2> は <time1> より大きい値を指定する必要があります。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧で囲まれています)。

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - . . .between(LOCAL 2005 May 1, LOCAL May 2005 1) (Citrix ADC システムタイムゾーンに応じて TRUE または FALSE。)

- **<certificate>.VALID_NOT_BEFORE.DAY:**

  証明書が有効である月の最終日を抽出し、その日を表す 1 ～31 の数値としてその値を返します。

- **<certificate>.VALID_NOT_BEFORE.EQ(<time>):**

  時間が <time> 引数と等しい場合は、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧で囲まれています)。

  - . . .eq(GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . . .eq(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - . . .eq(GMT 10h) (TRUE)
  - . . .eq(GMT 10h 30s) (TRUE)

  - . . .eq(GMT May 10h) (TRUE)
  - . . .eq(GMT Sun) (TRUE)
  - . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GE(<time>):**

  時間が <time> 引数より大きい (後) または等しい場合は、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧で囲まれています)。

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GT(<time>):**

  時間が <time> 引数の後に発生する場合は、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (評価結果は括弧で囲まれています)。

  - . . .gt(GMT 2004) (TRUE)
  - . . .gt(GMT 2005 Jan) (TRUE)
  - . . .gt(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - . . .gt(GMT 8h) (TRUE)
  - . . .gt(GMT 30m) (FALSE)
  - . . .gt(GMT May 10h) (FALSE)
  - . . .gt(GMT May 10h 0m) (TRUE)
  - . . .gt(GMT Sun) (FALSE)
  - . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.HOURS:**

  証明書が有効である直近の 1 時間を抽出し、その値を 0 ～23 の整数として返します。

- **<certificate>.VALID_NOT_BEFORE.LE (<time>)

  時間が <time> 引数に先行するまたは等しい場合は、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧で囲まれています)。

- . . .le(GMT 2006) (TRUE)
- . . .le(GMT 2005 Dec) (TRUE)
- . . .le(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
- . . .le(GMT 8h) (FALSE)
  . - . .le(GMT 30m) (TRUE)
- . . .le(GMT May 10h) (TRUE)
- . . .le(GMT Jun 11h) (TRUE)
- . . .le(GMT Wed) (TRUE)
- . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.LT(<time>):**

  時間が <time> 引数に先行する場合は、ブール型 TRUE を返します。

  たとえば、時刻の値が GMT 2005 5 月 1 日 10h 15m 30 秒で、2005 年 5 月の第 1 日曜日である場合、次のように指定できます (この例の評価結果は括弧で囲まれています)。

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May) (現在のタイムゾーンに応じて TRUE または FALSE。)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_MINUTES:**

  証明書が有効である最後の分を抽出します。現在の分を 0 ～59 の整数で返します。

- **<certificate>.VALID_NOT_BEFORE.MONTH:**

  証明書が有効である先月を抽出します。現在の月を 1 (1 月) ～12 (12 月) の整数で返します。

- **<certificate>.VALID_BEFORE.RELATIVE_BOOT:**

  直近またはスケジュールされた Citrix ADC 再起動までの秒数を計算し、整数を返します。最も近いブート時刻が過去にある場合、整数は負の値になります。将来の場合、整数は正の値になります。

- **<certificate>.VALID_NOT_NOT_NOW:**

  Citrix ADC 現在のシステム時刻から指定された時刻までの秒数を整数で返します。指定した時刻が過去の場合、整数は負の値になります。将来の場合、整数は正です。

- **<certificate>.VALID_BEFORE.SECONDS:**

  証明書が有効である最後の秒を抽出します。現在の秒を 0 ～59 の整数で返します。

- **<certificate>.VALID_NOT_BEFORE.WEEKDAY:**

  証明書が有効である最後の平日を抽出します。平日を 0 (日曜日) から 6 (土曜日) までの数値で返します。

- **<certificate>.VALID_NOT_BEFORE.WITHIN(<time1>, <time2>):**

  時間の各要素が、<time1> <time2> 引数で定義された範囲内に存在する場合、ブール型 TRUE を返します。

  <time1> から時間の要素を省略すると、その範囲内で最小値を持つと見なされます。から時間の要素を省略すると <time2>、その範囲内で最大値を持つと見なされます。で年を指定する場合は <time1>、で指定する必要があります <time2>。時間の要素の範囲は、月 1-12、日 1-31、平日 0-6、時間 0-23、分 0-59、秒 0-59 です。

  たとえば、時刻が GMT 2005 5 月 10 日 10h 15m 30 秒で、月の第 2 火曜日である場合、次のように指定できます (評価結果は括弧で囲まれています)。

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, 5 月は 1 月から 3 月の範囲ではありません。)
  - . . .within(GMT Feb, GMT) (TRUE、5 月は 2 月～12 月の範囲です。)
  - . . .within(GMT Sun_1, GMT Sun_3) (TRUE、第 2 火曜日は第 1 日曜日と第 3 日曜日の間です。)
  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (Citrix ADC システムのタイムゾーンに応じて、TRUE または FALSE)

- **<certificate>.VALID_NOT_BEFORE.YEAR:**

  証明書が有効である最後の年を抽出します。現在の年を 4 桁の整数で返します。

HTTP 要求日と応答日の式

October 7, 2021

次の式プレフィックスは、HTTP Date ヘッダーの内容をテキストまたは日付オブジェクトとして返します。これらの値は、次のように評価できます。

- 数字で指定します。HTTP Date ヘッダーの数値は、1970 年 1 月 1 日以降の秒数の形式で返されます。

  たとえば、式 http.req.date.mod (86400) は、その日の開始からの秒数を返します。これらの値は、他の非日付の数値データと同じ演算を使用して評価できます。詳細については、日付と時刻以外の数値データの式接頭辞を参照してください。

- HTTP ヘッダーとして。日付ヘッダーは、他の HTTP ヘッダーと同じ操作を使用して評価できます。

  詳細については、「デフォルトの構文式:HTTP、TCP、および UDP データの解析」を参照してください。

- テキストとして。日付ヘッダーは、他の文字列と同じ操作を使用して評価できます。

詳細については、「高度なポリシー式: テキストの評価」を参照してください。

前	説明
HTTP.REQ.DATE	HTTP Date ヘッダーの内容をテキストまたは日付オブジェクトとして返します。認識される日付形式は次のとおりです。RFC822。Sun, 06 Jan 1980 08:49:37 GMT, RFC850. Sunday, 06-Jan-80 09:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.
HTTP.RES.DATE か	HTTP Date ヘッダーの内容をテキストまたは日付オブジェクトとして返します。認識される日付形式は次のとおりです。RFC822。Sun, 06 Jan 1980 8:49:37 GMT, RFC850. Sunday, 06-Jan-80 9:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.

曜日を文字列として短い形式と長い形式で生成する

October 7, 2021

関数WEEKDAY_STRING_SHORT とWEEKDAY_STRINGは、それぞれ短い形式と長い形式で、曜日を文字列として生成します。返される文字列は常に英語です。これらの関数で使用されるプレフィックスは、曜日を整数形式で返す必要があります。プレフィックスによって返される値の許容範囲は 0 ～6 です。したがって、許容範囲内の整数を返す任意のプレフィックスを使用できます。UNDEF 条件は、戻り値がこの範囲にない場合、またはメモリ割り当てが失敗した場合に発生します。

次に、関数の説明を示します。

機能	説明
<prefix>.WEEKDAY_STRING_SHORT	曜日を短い形式で返します。短い形式は常に 3 文字の長さで、最初の大文字と残りの文字は小文字です。たとえば、SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT は、WEEKDAY 関数によって返される値が 0 の場合は Sun を返し、プレフィックスによって返される値が 6 の場合は Sat を返します。

機能	説明
<prefix>.WEEKDAY_STRING	曜日を長い形式で返します。長い形式は常に初期大文字で、残りの文字は小文字になります。たとえば、SYS.TIME.WEEKDAY.WEEKDAY_STRINGは、WEEKDAY 関数によって返される値が 0 の場合に Sunday を返し、プレフィックスによって返される値が 6 の場合に Saturday を返します。

日付と時刻以外の数値データの式プレフィックス

October 7, 2021

時間に基づいて動作する式を設定する以外に、次のタイプの数値データに対して式を構成できます。

- HTTP リクエストの長さ、リクエスト内の HTTP ヘッダーの数など。

  詳細については、「日付以外の数値 HTTP ペイロードデータの式」を参照してください。

- IP アドレスと MAC アドレス。

  詳細については、「IP アドレスと IP サブネットの式」を参照してください。

- インターフェイス ID およびトランザクションスループットレートに関するクライアントおよびサーバデータ。

  詳細については、「数値クライアントおよびサーバーデータの式」を参照してください。

- 日付以外のクライアント証明書の数値データ。

  証明書の有効期限までの日数や暗号化キーのサイズなど、これらのプレフィクスの詳細については、「SSL 証明書の数値データのプレフィックス」を参照してください。

数値をテキストに変換する

October 7, 2021

次の関数は、式のプレフィックスによって返された数値からバイナリ文字列を生成します。これらの関数は、バイナリデータの置換文字列として TCP 書き換え機能で特に有用です。TCP 書き換え機能の詳細については、書き換えを参照してください。

すべての関数は、テキスト型の値を返します。一部の関数がパラメータとして受け入れるエンディアンは、LITTLE_ENDIAN または BIG_ENDIAN のいずれかです。

機能	説明
<number>.SIGNED8_STRING	数を表す 8 ビットの符号付きバイナリ文字列を生成します。値が範囲外である場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STR
<number>.UNSIGNED8_STRING	数を表す 8 ビットの符号なしバイナリ文字列を生成します。値が範囲外である場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED
<number>.SIGNED16_STRING(<endianness>)	数を表す 16 ビットの符号付きバイナリ文字列を生成します。値が範囲外である場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)
<number>.UNSIGNED16_STRING(<endianness>)	数を表す 16 ビットの符号なしバイナリ文字列を生成します。値が範囲外である場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LIT-TLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN)
<number>.SIGNED32_STRING(<endianness>)	数値を表す 32 ビットの符号付きバイナリ文字列を生成します。例: HTTP.REQ.BODY(100).AFTER_STR(“delim”).GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_ENDIAN)
<unsigned_long_number>.UNSIGNED8_STRING	数を表す 8 ビットの符号なしバイナリ文字列を生成します。値が範囲外である場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIG
<unsigned_long_number>.UNSIGNED16_STRING	数を表す 16 ビットの符号なしバイナリ文字列を生成します。値が範囲外である場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LIT-TLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UNSIG

機能	説明
<unsigned_long_number>.UNSIGNED32_STRING(数値data type32ビット)	32ビットの符号なしバイナリ文字列を生成します。値が範囲外である場合、undef 条件が発生します。例: HTTP.REQ.BODY(100).AFTER_STR(“delim2”).GET_UNSIGNED3 BIG_ENDIAN).ADD(2).UNSIGNED32_STRING(BIG_ENDIAN)

仮想サーバーベースの式

October 7, 2021

SYS.VSERVER(”<vserver-name>”)式プレフィクスを使用すると、仮想サーバを識別できます。このプレフィックスで次の関数を使用して、指定した仮想サーバーに関する情報を取得できます。

- **THROUGHPUT.** 仮想サーバーのスループットを Mbps (メガビット/秒) で返します。返される値は、符号なしの long 数値です。

  使用法**:** SYS.VSERVER(“vserver”).THROUGHPUT

- **CONNECTIONS.** 仮想サーバーによって管理されている接続の数を返します。返される値は、符号なしの long 数値です。

  使用法**:** SYS.VSERVER(“vserver”).CONNECTIONS

- **STATE.** 仮想サーバーの状態を返します。返される値は、UP、DOWN、または OUT_OF_SERVICE です。したがって、これらの値の 1 つを EQ () 演算子に引数として渡して、TRUE または FALSE のブール値を比較できます。

  使用法**:**SYS.VSERVER(“vserver”).STATE

- **HEALTH.** 指定された仮想サーバの UP 状態にあるサービスの割合を返します。返される値は整数です。

  使用法**:** SYS.VSERVER(“vserver”).HEALTH

- **RESPTIME.** 応答時間をマイクロ秒数を表す整数で返します。応答時間は、仮想サーバーにバインドされたすべてのサービスからの平均 TTFB（最初のバイトまでの時間）です。

  使用法**:** SYS.VSERVER(“vserver”).RESPTIME

- **SURGECOUNT.** 仮想サーバーのサージキュー内の要求の数を返します。返される値は整数です。

  使用法**:**SYS.VSERVER(“vserver”).SURGECOUNT

例 **1**:

次の書き換えポリシーは、負荷分散仮想サーバー LBvserver での接続数が 10000 を超えると、書き換え処理を中止します。

```
add rewrite policy norewrite_pol sys.vserver("LBvserver").connections.gt
(10000)norewrite
```

例 **2**:

次の書き換えアクションでは、カスタムヘッダー TP が挿入されます。このヘッダーの値は、仮想サーバー LBvserver の全体です。

```
add rewrite action tp_header insert_http_header TP SYS.VSERVER("LBvserver")
.THROUGHPUT
```

例 **3:**

次の監査ログメッセージアクションは、仮想サーバにバインドされたサービスの平均 TTFB を newnslog ログファイルに書き込みます。

```
add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS
Response Time to Servers:\" + sys.vserver(\"ssllb\").resptime + \" millisec
\""-logtoNewnslog YES -bypassSafetyCheck YES
```

高度なポリシー式：**HTTP**、**TCP**、および **UDP** データの解析

January 31, 2022

高度なポリシー式を設定して、HTTP 要求または応答のペイロードを評価できます。HTTP 接続に関連付けられたペイロードには、HTTP ヘッダー（標準ヘッダーまたはカスタムヘッダー）、本文、接続 URL が含まれます。また、TCP または UDP パケットでペイロードを評価して処理することもできます。たとえば、HTTP 接続の場合、特定の HTTP ヘッダーが存在するかどうか、または URL に特定のクエリパラメーターが含まれているかどうかを確認できます。

URL エンコーディングを変換する式を構成し、その後の評価のために HTML または XML の「安全」コーディングを適用できます。XPATH および JSON プレフィックスを使用して、XML ファイルと JSON ファイルの日付を評価することもできます。

AAA.USER、AAA.LOGIN などの認証式の詳細については、「認証、承認、および監査ログイン」および「AAA.AUTHERTICATION 式」を参照してください。Citrix ADC AAA ユーザー認証のトピック 。

また、テキストベースおよび数値の Advanced ポリシー式を使用して、HTTP 要求および応答データを評価することもできます。詳細については、「高度なポリシー式: テキストの評価 」および「デフォルトの構文式: 日付、時刻、および数値の操作」を参照してください。

着信 **IP** パケット内のプロトコルを識別するための式

October 7, 2021

次の表に、着信パケットのプロトコルを識別するために使用できる式を示します。

式	説明
CLIENT.IP.PROTOCOL	クライアントから送信される IPv4 パケット内のプロトコルを識別します。
CLIENT.IPV6.PROTOCOL	クライアントから送信される IPv6 パケットのプロトコルを識別します。
SERVER.IP.PROTOCOL	サーバから送信される IPv4 パケットのプロトコルを識別します。
SERVER.IPV6.PROTOCOL	サーバから送信される IPv6 パケットのプロトコルを識別します。

PROTOCOL 関数の引数

インターネット割り当て番号機関 (IANA) プロトコル番号を PROTOCOL 関数に渡すことができます。たとえば、着信パケットのプロトコルが TCP であるかどうかを判断する場合は、CLIENT.IP.PROTOCOL.EQ (6) を使用できます。ここで、6 は、TCP の IANA によって割り当てられたプロトコル番号です。プロトコルによっては、プロトコル番号の代わりに列挙値を渡すことができます。たとえば、CLIENT.IP.PROTOCOL.EQ(6) の代わりに CLIENT.IP.PROTOCOL.EQ(TCP) を使用できます。次の表に、列挙値を使用できるプロトコルと、PROTOCOL 関数で使用する対応する列挙値を示します。

プロトコル	列挙値
Transmission Control Protocol (TCP)	TCP
ユーザーデータグラムプロトコル (UDP)	UDP
インターネット制御メッセージプロトコル (ICMP)	ICMP
IPv4 および IPv6 で認証サービスを提供するための IP 認証ヘッダー (AH)	AH
カプセル化セキュリティペイロード (ESP) プロトコル	ESP
一般ルーティングカプセル化（GRE）	GRE
IP-within-IP Encapsulation Protocol	IPIP

プロトコル	列挙値
IPv6 用インターネット制御メッセージプロトコル (ICMPv6)	ICMPv6
IPv6 のフラグメントヘッダー	FRAGMENT

ユースケースのシナリオ

プロトコル式は、要求ベースのポリシーと応答ベースのポリシーの両方で使用できます。この式は、負荷分散、WAN 最適化、コンテンツスイッチング、書き換え、リッスンポリシーなど、さまざまな Citrix ADC 機能で使用できます。EQ () や NE () などの関数とともに式を使用して、ポリシー内のプロトコルを識別し、アクションを実行できます。

式の使用例を次に示します。

- Branch Repeater の負荷分散構成では、ワイルドカード仮想サーバーのリスンポリシーで式を使用できます。たとえば、リッスンポリシー CLIENT.IP.PROTOCOL.EQ (TCP) を使用してワイルドカード仮想サーバーを構成して、仮想サーバーが TCP トラフィックのみを処理し、TCP 以外のトラフィックをすべてブリッジするようにできます。リッスンポリシーの代わりにアクセスコントロールリストを使用できますが、リッスンポリシーを使用すると、処理されるトラフィックをより適切に制御できます。
- 種類 ANY のコンテンツスイッチング仮想サーバーの場合、着信パケットのプロトコルに基づいて要求を切り替えるコンテンツスイッチングポリシーを構成できます。たとえば、すべての TCP トラフィックを 1 つの負荷分散仮想サーバーに送信し、すべての TCP 以外のトラフィックを別の負荷分散仮想サーバーに送信するようにコンテンツスイッチングポリシーを構成できます。
- クライアントベースの式を使用して、プロトコルに基づいて永続性を設定できます。たとえば、CLIENT.IP.PROTOCOL を使用して、着信 IPv4 パケットのプロトコルに基づいて永続性を構成できます。

HTTP およびキャッシュ制御ヘッダーの式

October 7, 2021

HTTP トラフィックを評価する一般的な方法の 1 つは、要求または応答のヘッダーを調べることです。ヘッダーは、次のような多くの機能を実行できます。

- 送信者に関するデータを含むクッキーを提供する。
- 送信されるデータのタイプを特定します。
- データが移動したルート（Via ヘッダー）を特定します。

> 注
>
> ヘッダーデータとテキストデータの両方を評価するために操作を使用する場合、ヘッダーベースの操作は常にテキストベースの操作よりも優先されます。たとえば、AFTER_STR 操作をヘッダーに適用すると、現在のヘッダータイプのすべてのインスタンスのテキストベースの AFTER_STR 操作が上書きされます。

HTTP ヘッダーのプレフィックス

HTTP ヘッダーを抽出する式プレフィクスの 「HTTP ヘッダーのプレフィックス」テーブル。

HTTP ヘッダーの操作

HTTP ヘッダーのプレフィックスで指定できる操作については 、「HTTP ヘッダーの操作」表を参照してください。

キャッシュ制御ヘッダーのプレフィックス

次のプレフィックスは、特にキャッシュ制御ヘッダーに適用されます。

HTTP ヘッダープレフィックス	説明
HTTP.REQ.CACHE_CONTROL	HTTP リクエストのキャッシュ制御ヘッダーを返します。
HTTP.RES.CACHE_CONTROL	HTTP レスポンスでキャッシュ制御ヘッダーを返します。

キャッシュ制御ヘッダーの操作

HTTP ヘッダーの任意の操作を Cache-Control ヘッダーに適用できます。

さらに、次の操作では、特定の種類のキャッシュ制御ヘッダーを識別します。これらのヘッダータイプについては、RFC 2616 を参照してください。

HTTP ヘッダー操作	説明
`Cache-Control header.NAME(<integer>)`	<integer>で指定された名前/値リスト内の n 番目のコンポーネントに対応する Cache-Control ヘッダーの名前をテキスト値として返します。名前/値コンポーネントのインデックスは 0 ベースです。integer 引数で指定された<integer> がリスト内のコンポーネント数より大きい場合、長さ 0 のテキストオブジェクトが返されます。以下はその例です。 `http.req.cache_control.name(3).contains("some_text")`
Cache-Control header.IS_INVALID	要求または応答に Cache-Control ヘッダーが存在しない場合は、ブール型 (Boolean) の値を返します。以下はその例です。 `http.req.cache_control.is_invalid`
`Cache-Control header.IS_PRIVATE`	キャッシュ制御ヘッダーがプライベート値を持っている場合、ブール型 TRUE を返します。以下はその例です。`http.req.cache_control.is_private`
Cache-Control header.IS_PUBLIC	キャッシュ制御ヘッダーがプライベート値を持っている場合、ブール型 TRUE を返します。以下に例を示します。
Cache-Control header.IS_NO_STORE	キャッシュ制御ヘッダーが値 No-Store を持っている場合、ブール値を返します。以下は例を示しています。
Cache-Control header.IS_NO_CACHE	キャッシュ制御ヘッダーが値 No-Cache を持っている場合、ブール値を返します。以下に例を示します。キャッシュキャッシュの制御はキャッシュされません。
Cache-Control header.IS_MAX_AGE	キャッシュコントロールヘッダーが Max-Age の値を持っている場合、ブール型 (Boolean) の値を返します。以下に、http.req.cache_ コントロール _ 最大年齢の例を示します。
Cache-Control header.IS_MIN_FRESH	キャッシュ制御ヘッダーが Min-Fresh の値を持っている場合、ブール値を返します。以下に例を示します。
Cache-Control header.IS_MAX_STALE	キャッシュコントロールヘッダーが Max-Stale の値を持っている場合、ブール型 (Boolean) の値を返します。以下に例を示します。

HTTP ヘッダー操作	説明
Cache-Control header.IS_MUST_REVALIDATE	キャッシュ制御ヘッダーに値を持っている場合、ブール値 TRUE を返します。以下に例を示します。再検証する必要があります。
Cache-Control header.IS_NO_TRANSFORM	キャッシュ制御ヘッダーの値が No-Transform の場合、ブール型 (Boolean) の値を返します。以下は例を示しています。
Cache-Control header.IS_ONLY_IF_CACHED	キャッシュ制御ヘッダーが Only-If-Cached の値を持っている場合、ブール型 (Boolean) の値を返します。キャッシュされた場合のみ: http.req.cache_control.is_only_ キャッシュされた場合のみ
Cache-Control header.IS_PROXY_REVALIDATE	キャッシュ制御ヘッダーがプロキシ再確認の値を持っている場合、ブール型 TRUE を返します。次に、http.req.cache_control.is_ プロキシ再検証の例を示します。
Cache-Control header.IS_S_MAXAGE	キャッシュコントロールヘッダーが S-Maxage の値を持つ場合に、ブール型 (Boolean) の値を返します。以下に例を示します。
Cache-Control header.IS_UNKNOWN	Cache-Control ヘッダーの型が不明な場合は、ブール型 TRUE を返します。以下に、http.req.cache_ コントロール _ 未知の例を示します。
Cache-Control header.MAX_AGE	キャッシュ制御ヘッダー Max-Age の値を返します。このヘッダーが存在しないか無効な場合は、0 が返されます。以下に例を示します。
Cache-Control header.MAX_STALE	キャッシュ制御ヘッダーの最大古い値を返します。このヘッダーが存在しないか無効な場合は、0 が返されます。以下に例を示します。
Cache-Control header.MIN_FRESH	キャッシュ制御ヘッダーの Min-Fresh の値を返します。このヘッダーが存在しないか無効な場合は、0 が返されます。以下に例を示します。
Cache-Control header.S_MAXAGE	キャッシュ制御ヘッダー S-Maxage の値を返します。このヘッダーが存在しないか無効な場合は、0 が返されます。以下に例を示します。

URL のセグメントを抽出するための式

October 7, 2021

URL と URL の一部（ホスト名など）または URL パスのセグメントを抽出できます。たとえば、次の式は、URL からイメージファイルのサフィックスを抽出することによって、イメージファイルの HTTP 要求を識別します。

```
http.req.url.suffix.eq("jpeg")|| http.req.url.suffix.eq("gif")
```

URL のほとんどの式はテキストで動作し、HTTP リクエストとレスポンスのテキストの式プレフィックスで説明されています。このセクションでは、GET オペレーションについて説明します。GET 操作は、次のプレフィックスを付けて使用すると、テキストを抽出します。

- HTTP.REQ.URL.PATH
- VPN.BASEURL.PATH
- VPN.CLIENTLESS_BASEURL.PATH

次の表に、HTTP URL のプレフィクスを示します。

URL プレフィックス	説明
HTTP.REQ.URL.PATH.GET(<n>)	URL パスからスラッシュ（「/」）で区切られたリストを返します。たとえば、次の URL を考えてみます。<http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>。次の式は、この URL から dir1 を返します。<http.req.url.path.get(1)>。次の式は、ディレクトリ 2 を返します。
HTTP.REQ.URL.PATH.GET_REVERSE(<n>)	パスの末尾から始まる、URL パスからスラッシュ（「/」）で区切られたリストを返します。たとえば、次の URL を考えてみます。<http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>。次の式は、この URL から index.html を返します。<http.req.url.path.get_reverse(0)>。次の式は、ディレクトリ 3 を返します。

日付以外の HTTP ステータスコードおよび数値の HTTP ペイロードデータの式

October 7, 2021

次の表に、日付以外の HTTP データ内の数値のプレフィックスを示します。

前	説明
HTTP.REQ.CONTENT_LENGTH	HTTP リクエストの長さを数値で返します。以下に例を示します。内容の長さ < 500
HTTP.RES.CONTENT_LENGTH	HTTP レスポンスの長さを数値で返します。以下に例を示します。内容の長さは 1000 以下です。
HTTP.RES.STATUS	応答ステータスコードを返します。
HTTP.RES.IS_REDIRECT	レスポンスコードがリダイレクトに関連付けられている場合は、ブール型 TRUE を返します。リダイレクト応答コードは次のとおりです。300（複数選択）、301（永続的に移動）、302（検出）、303（その他参照）、305（プロキシを使用）、307（一時リダイレクト）。注：ステータスコード 304 は、リダイレクト HTTP レスポンスステータスコードとはみなされません。ステータスコード 306 は使用されていません。

SIP の式

October 7, 2021

Citrix ADC の詳細ポリシー式言語には、セッション開始プロトコル（SIP）接続で動作するいくつかの式が含まれています。これらの式は、要求/応答ベースで動作する、サポートされているプロトコルのポリシーで使用することを意図しています。これらの式は、コンテンツスイッチング、レート制限、レスポンダ、および書き換えポリシーで使用できます。

レスポンダポリシーで使用される SIP 式には、特定の制限が適用されます。SIP ロードバランシング仮想サーバーでは、DROP、NOOP、または RESPONDWITH アクションだけが許可されます。レスポンダーポリシーは、負荷分散仮想サーバー、上書きグローバルバインドポイント、既定のグローバルバインドポイント、または sip_udp ポリシーラベルにバインドできます。

SIP プロトコルで使用されるヘッダー形式は、HTTP プロトコルで使用されるヘッダー形式と似ているため、新しい式の多くは HTTP アナログとよく似ています。各 SIP ヘッダーは、SIP メソッド、URL、およびバージョンを含む行で構成され、続いて HTTP ヘッダーのような一連の名前と値のペアが続きます。

次に、その下の式テーブルで参照される SIP ヘッダーの例を示します。

```
1 INVITE sip:16@www.sip.com:5060;transport=udp SIP/2.0
2 Record-Route: <sip:200.200.100.22;lr=on>
```



```
3  Via: SIP/2.0/UDP 200.200.100.22;branch=z9hG4bK444b.c8e103d1.0;rport
       =5060;
4      received=10.102.84.18
5  Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;
6      received=10.102.84.160
7  From: ”12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53
       cc0185
8  To: ”16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
9  Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180
10 Max-Forwards: 69CSeq: 101 INVITE
11 User-Agent: Cisco-CP7940G/8.0
12 Contact: <sip:12@10.102.84.180:5060;transport=udp>
13 Expires: 180
14 Accept: application/sdp
15 Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE
16 Supported: replaces,join,norefersub
17 Content-Length: 277
18 Content-Type: application/sdp
19 Content-Disposition: session;handling=optiona
20 <!--NeedCopy-->
```

SIP リファレンステーブル

次の表に、SIP ヘッダーで動作する式の一覧を示します。最初のテーブルには、リクエストヘッダーに適用される式が含まれています。ほとんどの応答ベースの式は、対応する要求ベースの式とほぼ同じです。対応する要求式から応答式を作成するには、式の最初の 2 つのセクションを SIP.REQ から SIP.RES に変更し、その他の明白な調整を行います。2 番目のテーブルには、応答に固有の応答式が含まれており、対応する要求はありません。次の表の任意の要素を完全な式として使用することも、さまざまな演算子を使用してこれらの式要素を他の要素と組み合わせて、より複雑な式を形成することもできます。

SIP 要求式

式	説明
SIP.REQ.METHOD	SIP 要求のメソッドで操作します。サポートされている SIP 要求メソッドは、ACK、BYE、CANCEL、INFO、INVITE、MESSAGE、NOTIFY、OPTIONS、PRACK、PUBLISH、REFER、REGISTER、SUBSCRIBE、UPDATE です。この式はテキストクラスの派生であるため、テキストに適用可能なすべての操作がこのメソッドに適用されます。たとえば、SIP 要求が INVITE SIP: 16 @10 .102.84. 181:5060; トランスポート =udp SIP/2.0 の場合、この式は INVITE を返します。
SIP.REQ.URL	SIP 要求 URL で操作します。この式はテキストクラスの派生であるため、テキストに適用可能なすべての操作がこのメソッドに適用されます。たとえば、SIP リクエストが「INVITE SIP: 16 @10 .102.84. 181:5060; トランスポート =udp SIP/2.0」の場合、次の式は 16 @10 .102.84. 181:5060; トランスポート =udp を返します。
SIP.REQ.URL.PROTOCOL	URL プロトコルを返します。たとえば、SIP URL の「16@www.sip.com: 5060; トランスポート =udp」の場合、この式は sip を返します。
SIP.REQ.URL.HOSTNAME	SIP URL のホスト名部分を返します。たとえば、「16@www.sip.com: 5060; トランスポート =udp」の場合、この式は「www.sip.com: 5060」を返します。
SIP.REQ.URL.HOSTNAME.PORT	SIP URL ホスト名のポート部分を返します。ポートが指定されていない場合、この式はデフォルトの SIP ポート 5060 を返します。たとえば、SIP ホスト名が www.sip.com: 5060 の場合、この式は 5060 を返します。
SIP.REQ.URL.HOSTNAME.DOMAIN	SIP URL ホスト名のドメイン名部分を返します。ホストが IP アドレスの場合、この式は誤った結果を返します。たとえば、SIP ホスト名が www.sip.com: 5060 の場合、この式は sip.com を返します。SIP ホスト名が 192.168.43. 15:5060 の場合、この式はエラーを返します。

式	説明
SIP.REQ.URL.HOSTNAME.SERVER	ホストのサーバー部分を返します。たとえば、SIP ホスト名が www.sip.com: 5060 の場合、この式は www を返します。
SIP.REQ.URL.USERNAME	@ 文字の前のユーザー名を返します。たとえば、SIP URL が「16@www.sip.com: 5060; トランスポート =udp」の場合、この式は 16 を返します。
SIP.REQ.VERSION	要求内の SIP バージョン番号を返します。たとえば、SIP 要求が INVITE SIP: 16 @10 .102.84. 181:5060; トランスポート =udp SIP/2.0 の場合、この式は SIP/2.0 を返します。
SIP.REQ.VERSION.MAJOR	メジャーバージョン番号（ピリオドの左側の番号）を返します。たとえば、SIP バージョン番号が SIP/2.0 の場合、この式は 2 を返します。
SIP.REQ.VERSION.MINOR	マイナーバージョン番号 (ピリオドの右側にある数値) を返します。たとえば、SIP バージョン番号が SIP/2.0 の場合、この式は 0 を返します。
SIP.REQ.CONTENT_LENGTH	コンテンツ-Length ヘッダーの内容を返します。この式は thesip_header_t クラスの派生型であるため、SIP ヘッダーで使用できるすべての操作を使用できます。たとえば、コンテンツ長:277 の SIP コンテンツ長ヘッダーの場合、この式は 277 を返します。
SIP.REQ.TO	To ヘッダーの内容を返します。たとえば、[宛先] のヘッダーの場合、<sip: 16@sip_example.com > タグ =00127f54ec85a6d90c14f45-53cc0185 は、この式は「16」を返します <sip: 16@sip_example.com >。タグ =00127f54ec85a6d90c14f45-53ccc0185 を返します。
SIP.REQ.TO.ADDRESS	sip_url オブジェクトにある SIP URI を返します。SIP URI で使用できるすべての操作を使用できます。たとえば、[宛先] ヘッダーの [宛先] の場合、<sip: 16@sip_example.com > タグ =00127f54ec85a6d90c14f45-53cc0185 の場合、この式は sip: 16@sip_example.com を返します。

式	説明
SIP.REQ.TO.DISPLAY_NAME	To ヘッダーの表示名部分を返します。たとえば、[宛先] の SIP ヘッダーの場合、<sip: 16@sip_example.com > タグ =00127f54ec85a6d90c14f45-53cc0185、この式は 16 を返します。
SIP.REQ.TO.TAG	TO ヘッダーの「タグ」名値のペアから「タグ」値を返します。たとえば、[宛先] ヘッダーの場合、タグ <sip: 16@sip_example.com > =「16」、タグ = 00127f54cc14f45-53cc0185 は、この式は、00127f54ec85a6d90c14f45-53cc0185 を返します。
SIP.REQ.FROM	From ヘッダーの内容を返します。たとえば、「12」<sip: 12@sip_example.com >; タグ =00127f54ec85a6d90c14f45-53cc0185 の差出人ヘッダーの場合、この式は sip: 12@sip_example.com を返します。
SIP.REQ.FROM.ADDRESS	sip_url オブジェクトにある SIP URI を返します。SIP URI で使用できるすべての操作を使用できます。たとえば、「12」<sip: 12@sip_example.com >; タグ =00127f54ec85a6d90c14f45-53cc0185 の差出人ヘッダーの場合、この式は sip: 12@sip_example.com を返します。
SIP.REQ.FROM.DISPLAY_NAME	To ヘッダーの表示名部分を返します。たとえば、「12」<sip: 12@sip_example.com >; タグ =00127f54ec85a6d90c14f45-53cc0185 の差出人ヘッダーの場合、この式は 12 を返します。
SIP.REQ.FROM.TAG	TO ヘッダーの「タグ」名と値のペアから「タグ」値を返します。たとえば、”12”<sip:12@sip_example.com>; タグ =00127f54cc14f45-53cc0185 の差出人ヘッダーの場合、この式は 00127f54ec85a6d90c14f45-53cc0185 を返します。

式	説明
SIP.REQ.VIA	完全な Via ヘッダーを返します。リクエスト内に複数の Via ヘッダーがある場合は、最後の Via ヘッダーを返します。たとえば、サンプル SIP ヘッダー内の 2 つの Via ヘッダーの場合、次の式は Via を返します。SIP/2.0/UDP 10.102.84。180:5060; ブランチ =z9hG4bK03e76d0b; ポート =5060; 受信 =10.102.84.160。
SIP.REQ.VIA.SENTBY_ADDRESS	リクエストを送信したアドレスを返します。たとえば、ビアヘッダーの場合、SIP/2.0/UDP 10.102.84. 180:5060; ブランチ =z9hG4bK03e76d0b; ポート =5060; 受信 =10.102.84.160、この式は 10.102.84.180 を返します。
SIP.REQ.VIA.SENTBY_PORT	要求を送信したポートを返します。たとえば、ビアヘッダーの場合、SIP/2.0/UDP 10.102.84. 180:5060; ブランチ =z9hG4bK03e76d0b; ポート =5060; 受信 =10.102.84.160、この式は 5060 を返します。
SIP.REQ.VIA.RPORT	rport の名前と値のペアから値を返します。たとえば、ビアヘッダーの場合、SIP/2.0/UDP 10.102.84. 180:5060; ブランチ =z9hG4bK03e76d0b; ポート =5060; 受信 =10.102.84.160、この式は 5060 を返します。
SIP.REQ.VIA.BRANCH	ブランチ名と値のペアから値を返します。たとえば、ビアヘッダーの場合、SIP/2.0/UDP 10.102.84. 180:5060; ブランチ =9hG4bK03e76d0b; ポート =5060; 受信 =10.102.84.160、この式は z9hG4bK03e76d0b を返します。
SIP.REQ.VIA.RECEIVED	受信した名前と値のペアから値を返します。たとえば、ビアヘッダーの場合、SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re この式は 10.102.84.160 を返します。

式	説明
SIP.REQ.CALLID	Callid ヘッダーの内容を返します。この式は thesip_header_t クラスの派生型であるため、SIP ヘッダーで使用できるすべての操作を使用できます。For example, for a SIP Callid header ofCall-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180, this expression returns00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180.
SIP.REQ.CSEQ	CSEQ から CSEQ 番号を整数で返します。たとえば、CSeq: 101 INVITE の SIP CSEQ ヘッダーの場合、この式は 101 を返します。
SIP.REQ.HEADER(<header_name>)	指定された SIP ヘッダーを返します。<header_name> は、目的のヘッダーの名前を置き換えます。たとえば、SIP からヘッダーを返すには、SIP.REQ.HEADER (「差出人」) と入力します。
SIP.REQ.HEADER(<header_name>).INSTANCE(<line_number>)	指定された SIP ヘッダーの指定されたインスタンスを返します。同じ SIP ヘッダーの複数のインスタンスが発生する可能性があります。このような SIP ヘッダーの特定のインスタンス（たとえば、特定の Via ヘッダー）が必要な場合は、に番号を入力してそのヘッダーを指定できます <line_number>。ヘッダーインスタンスは最後（0）から最初に一致します。言い換えれば、SIP.REQ.HEADER（「ビア」）.INSTANCE（0）は、VIA ヘッダーの最後のインスタンスを返し、SIP.REQ.HEADER（「ビア」）.INSTANCE（1）は最後のインスタンスを返しますが、VIA ヘッダーのいずれかを返します。たとえば、SIP ヘッダーの例で使用した場合、SIP. 要求ヘッダー (「経由」) . インスタンス (1) は、次のとおりを返します:SIP/2.0/UDP 10.102.84. 180:5060; ブランチ =z9hG4bK03e76d0b; rport=5060.

式	説明
SIP.REQ.HEADER(<header_name>).VALUE(<line_	指定された SIP ヘッダーの指定されたインスタンスの内容を返します。使用法は前の式とほぼ同じです。たとえば、上記のテーブルエントリの SIP ヘッダーの例で使用した場合、SIP.REQ.HEADER (「経由」) .VALUE (1) は SIP/2.0/UDP 10.102.84 を返します。180:5060; ブランチ = z9hG4bK03e76d0b; rport = 5060 を返します。
SIP.REQ.HEADER(<header_name>).COUNT	特定のヘッダーのインスタンスの数を整数で返します。たとえば、上記の SIP ヘッダーの例で使用すると、SIP.REQ.HEADER (「Via」) .COUNT は 2 を返します。
SIP.REQ.HEADER(<header_name>).EXISTS	指定されたヘッダーが存在するかどうかに応じて、true または false のブール値を返します。たとえば、上記の SIP ヘッダーの例で使用した場合、SIP.REQ.HEADER (「期限切れ」) .EXISTS は true を返し、一方、SIP.REQ.HEADER (「発信者 ID」) .EXISTS は false を返します。
SIP.REQ.HEADER(<header_name>).LIST	指定されたヘッダーのコンマ区切りのパラメータリストを返します。たとえば、上記の SIP ヘッダーの例で使用した場合、SIP.REQ.HEADER (「許可」) .LIST は ACK、BYE、キャンセル、受信、通知、オプション、参照、登録、更新を返します。文字列.GET (<list_item_number>) を追加して、特定のリスト項目を選択できます。たとえば、上記のリストから最初の項目 (ACK) を取得するには、SIP.REQ.HEADER (「許可」) .LIST.GET (0) と入力します。2 番目の項目 (BYE) を抽出するには、SIP.REQ.HEADER (「許可」) .LIST.GET (1) と入力します。注：指定されたヘッダーに名前/値のペアのリストが含まれている場合は、名前/値のペア全体が返されます。

式	説明
SIP.REQ.HEADER(<header_name>).TYPECAST_SI	型キャスト <header_name> をに <in_header_name> 指定します。任意のテキストを thesip_header_t クラスに型キャストすることができます。その後、すべてのヘッダーベースの操作を使用できます。この操作を実行すると、で使用できるすべての操作を適用できます <in_header_name>。たとえば、式 SIP.REQ. コンテンツの長さ. タイプタイプキャストは、コンテンツ長ヘッダーのすべてのインスタンスを型キャストします。この操作を実行した後、指定したヘッダーのすべてのインスタンスにすべてのヘッダー操作を適用できます。
SIP.REQ.HEADER(<header_name>).CONTAINS(<string>)	指定されたテキスト文字列が指定されたヘッダーのいずれかのインスタンスに存在する場合は、ブール値 true を返します。指定されたヘッダーのすべてのインスタンスに対して操作します。ヘッダーインスタンスは最後（0）から最初に一致します。
SIP.REQ.HEADER(<header_name>).EQUALS_ANY	に関連付けられたパターンが、<patset> 指定されたヘッダーのいずれかのインスタンス内のコンテンツと一致する場合は、ブール値 true を返します。指定されたヘッダーのすべてのインスタンスに対して操作します。ヘッダーインスタンスは最後（0）から最初に一致します。
SIP.REQ.HEADER(<header_name>).CONTAINS_ANY(<patset>)	に関連付けられたパターンが、<patset> 指定されたヘッダーのいずれかのインスタンス内のコンテンツと一致する場合は、ブール型 true を返します。指定されたヘッダーのすべてのインスタンスに対して操作します。ヘッダーインスタンスは最後（0）から最初に一致します。
SIP.REQ.HEADER(<header_name>).CONTAINS_IN	そのパターンが <patset> 指定されたヘッダーのいずれかのインスタンス内のコンテンツと一致する場合に、関連付けられた一致パターンのインデックスを返します。指定されたヘッダーのすべてのインスタンスに対して操作します。ヘッダーインスタンスは最後（0）から最初に一致します。

式	説明
SIP.REQ.HEADER(<header_name>).EQUALS_INDEX(<patset>)	<header_name> が <patset> 指定されたヘッダーのいずれかのインスタンスと一致する場合に、関連付けられた一致パターンのインデックスを返します。指定されたヘッダーのすべてのインスタンスに対して操作します。ヘッダーインスタンスは最後（0）から最初に一致します。
SIP.REQ.HEADER(<header_name>).SUBSTR(<stri	指定された文字列が指定されたヘッダーのインスタンスに存在する場合、この式はその文字列を返します。たとえば、SIP ヘッダー経由の場合：SIP/2.0/UDP 10.102.84. 180:5060; ブランチ =z9hG4bK03e76d0b; ポート =5060; 受信 =10.102.84.160”、SIP. 要求. ヘッダー（「経由」）.SUBSTR（「rport=5060」）は「rport=5060」を返します。.REQ.HEADER (「経由」) .SUBSTR (「rport = 5061」) は空の文字列を返します。
SIP.REQ.HEADER(<header_name>).AFTER_STR(<string>)	指定された文字列が指定されたヘッダーのいずれかのインスタンスに存在する場合、この式はその文字列の直後の文字列を返します。たとえば、SIP ヘッダービア:SIP/2.0/UDP 10.102.84. 180:5060; ブランチ =z9hG4bK03e76d0b; ポート =5060; 受信 =10.102.84.160、式 SIP. 要求. ヘッダー (「経由」) .AFTER_STR (「rport=」) は 5060 を返します。

式	説明
SIP.REQ.HEADER(<header_name>).REGEX_MATC	指定された正規表現（regex）が指定されたヘッダーのいずれかのインスタンスと一致する場合、ブール値 true を返します。正規表現は、次の形式で指定する必要があります。re <delimiter> 正規表現 <same delimiter>。正規表現の長さは 1499 文字を超えることはできません。PCRE 正規表現ライブラリに準拠している必要があります。PCRE 正規表現構文については、http://www.pcre.org/pcre.txtを参照してください。pcrepattern のマニュアルページには、PCRE 正規表現を使用したパターンの指定に関する有用な情報も記載されています。この式でサポートされている正規表現の構文は、PCRE と若干の違いがあります。後方参照は許可されていません。再帰的な正規表現は避けるべきです。いくつかの仕事がありますが、多くはそうではありません。ドット (.) メタ文字は改行と一致します。Unicode はサポートされません。SET_TEXT_MODE（IGNORECASE）は、(? i) 正規表現で指定された内部オプション。
SIP.REQ.HEADER(<header_name>).REGEX_SELECT(<regex>)	指定された正規表現が指定されたヘッダーのいずれかのインスタンス内の任意のテキストと一致する場合、この式はテキスト を返します。たとえば、SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re は式 SIP.REQ.HEADER(“Via”).REGEX_SELECT(“received=[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}”) received=10.102.84.160 を返します。
SIP.REQ.HEADER(<header_name>).AFTER_REGE	指定された正規表現が、指定されたヘッダーの任意のインスタンス内の任意のテキストと一致する場合、この式は、そのテキストの直後の文字列を 返します。たとえば、SIP ヘッダーの経由:SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER(“Via”).AFTER_REGEX(“received=”) returns10.102.84.160.

式	説明
SIP.REQ.HEADER(<header_name>).BEFORE_REGEX(<regex>)	指定された正規表現が指定されたヘッダーのいずれかのインスタンス内の任意のテキストと一致する場合、この式は、そのテキストの直前の文字列を返します。たとえば、SIP ヘッダーの経由：SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER(“Via”).BEFORE_REGEX(“[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}”) returns received=.
SIP.REQ.FULL_HEADER	終端の CR/LF を含む SIP ヘッダー全体を返します。
SIP.REQ.IS_VALID	リクエスト形式が有効な場合、ブール値 true を返します。
SIP.REQ.BODY(<length>)	指定された長さまで、リクエスト本文を返します。指定された長さがリクエスト本文の長さより大きい場合、この式はリクエスト本文全体を返します。
SIP.REQ.LB_VSERVER	現在の要求を処理している負荷分散仮想サーバー (LB vserver) の名前を返します。
SIP.REQ.CS_VSERVER	現在の要求を処理しているコンテンツスイッチ仮想サーバー (CS vserver) の名前を返します。

SIP 応答式

式	説明
SIP.RES.STATUS	SIP 応答ステータスコードを返します。たとえば、応答の最初の行が「SIP/2.0 100」の場合、この式は「100」を返します。
SIP.RES.STATUS.MSG	SIP 応答ステータスメッセージを返します。たとえば、応答の最初の行が SIP/2.0 100 Trying である場合、この式は Trying を返します。
SIP.RES.IS_REDIRECT	レスポンスコードがリダイレクトの場合、ブール値 true を返します。
SIP.RES.METHOD	CSeq ヘッダ内の要求メソッド文字列から抽出された応答メソッドを返します。

HTTP、HTML、XML エンコーディングおよび「安全」文字の操作

October 7, 2021

以下の操作は、リクエストまたはレスポンス内の HTML データのエンコードと、POST 本文の XML データを使用します。

- **<text>.HTML_XML_SAFE:**
  次の例のように、特殊文字を XML セーフ形式に変換します。

  左向きの山括弧 (<) は < に変換されます
  < 右向きの山括弧 () は >
  アンパサンド (&) に変換されます &
  この操作は、クロスサイトスクリプティング攻撃から保護します。変換されたテキストの最大長は 2048 バイトです。これは読み取り専用の操作です。

  変換を適用すると、式で指定した追加の演算子が、選択したテキストに適用されます。次に例を示します。

  html_xml_safe. には（「myQueryString」）が含まれています。

- **<text>.HTTP_HEADER_SAFE:**
  入力テキストのすべての改行 (‘n’) 文字を ‘%0A’ に変換し、入力を HTTP ヘッダーで安全に使用できるようにします。

  この操作は、応答分割攻撃から保護します。

  変換されたテキストの最大長は 2048 バイトです。これは読み取り専用の操作です。

- **<text>.HTTP_URL_SAFE:**
  安全でない URL 文字を ‘%xx’ 値に変換します。ここで、「xx」は入力文字の 16 進数ベースの表現です。たとえば、アンパサンド (&) は、URL セーフエンコードでは%26 と表されます。変換されたテキストの最大長は 2048 バイトです。これは読み取り専用の操作です。

  以下は、URL セーフ文字です。その他はすべて安全ではありません:

  - 英数字:a ～z、A ～Z、0 ～9
  - アスタリスク:「*」
  - アンパサンド:「&」
  - アットマーク:「@」
  - コロン:「:」
  - カンマ:「,」
  - ドル:「$」
  - ドット:「. 」
  - 等しい:「=」
  - 感嘆符:「! 」
  - ハイフン:「-」

- 左かっこ:「(“, “)」
- パーセント:「%」
- プラス:「+」
- セミコロン:「;」
- 一重引用符:「’」
- スラッシュ:「/」
- 疑問符:「? 」
- チルダ:「~」
- アンダースコア:「_」

- **<text>.MARK_SAFE:**

  任意のタイプのデータ変換を適用せずに、テキストを安全としてマークします。

- **<text>.SET_TEXT_MODE (URLENCODED|NOURLENCODED)**

  バイトストリーム内のすべての%HH エンコーディングを変換します。この操作は、（バイトではなく）文字で動作します。デフォルトでは、1 バイトは ASCII エンコーディングの文字を表します。ただし、URLENCODED モードを指定すると、3 バイトで文字を表すことができます。

  次の例では、PREFIX (3) 操作でターゲットの最初の 3 文字を選択します。

  ```
  http.req.url.hostname.prefix(3)
  ```

  次の例では、Citrix ADC はターゲットから最大 9 バイトを選択できます。

  ```
  http.req.url.hostname.set_text_mode(urlencoded).prefix(3)
  ```

- **<text>.SET_TEXT_MODE (**プラススペース | プラススペースなし**):**

  プラス文字 (+) の処理方法を指定します。PLUS_AS_SPACE オプションは、プラス文字を空白に置き換えます。たとえば、テキスト「hello+world」は「hello world」になります。NO_PLUS_AS_SPACE オプションでは、プラス文字はそのまま残されます。

- **<text>.SET_TEXT_MODE(BACKSLASH_ENCODED|NO_BACKSLASH_ENCODED):**

  で表されるテキストオブジェクトに対してバックスラッシュのデコードを実行するかどうかを指定します <text>。

  BACKSLASH_ENCODED が指定されている場合、SET_TEXT_MODE 演算子は、テキスト・オブジェクトに対して次の操作を実行します。

  - 「XXX」のすべての出現は、文字「Y」に置き換えられます（ここで、XXX は 8 進数の数字を表し、Y は XXX に相当する ASCII を表します）。このタイプのエンコーディングの 8 進値の有効範囲は 0 ～377 です。たとえば、エンコードされたテキスト「http72//」と「http072//」の両方がにデコードされます。コロン (:) は<http://>、8 進値「72」に相当する ASCII 値です。
  - 「xHH」のすべての出現は、文字「Y」に置き換えられます（HH は 16 進数の数値を表し、Y は HH に相当する ASCII を表します。たとえば、エンコードされたテキスト「httpx3a//」はにデコードされます。コロン（:）は<http://>、16 進値「3a」に相当する ASCII です。

- 「uWWXX」のすべての出現は、文字シーケンス「YZ」に置き換えられます（WW と XX は 2 つの異なる 16 進値を表し、Y と Z は WW と XX にそれぞれ相当する ASCII 値を表します。たとえば、エンコードされたテキスト「http%u3a2f/」と「http%u003a//」は両方とも<http://>にデコードされます。ここで、「3a」と「2f」は 2 つの 16 進値で、コロン（:）とスラッシュ（「/」）はそれぞれの ASCII 値を表します。をそれぞれ指定します。
- 「b」、「n」、「t」、「f」、「r」のすべての出現は、対応する ASCII 文字に置き換えられます。

NO_BACKSLASH_ENCODED が指定されている場合、テキストオブジェクトに対してバックスラッシュのデコードは実行されません。

- **<text>.SET_TEXT_MODE (BAD_ENCODE_RAISE_UNDEF|BAD_ENCODE_RAISE_UNDEF):**

  URLENCODED モードまたは BACKSLASH_ENCODED モードが設定されていて、<text> で指定されたエンコーディングモードに対応する不正なエンコーディングがで表されるテキストオブジェクトで検出された場合、関連付けられた未定義のアクションを実行します。

  NO_BAD_ENCODE_RAISE_UNDEF が指定されている場合、<text> によって表されるテキストオブジェクトで不正なエンコーディングが検出された場合、関連付けられた未定義のアクションは実行されません。

TCP、UDP、および VLAN データの式

October 7, 2021

TCP および UDP データは、文字列または数値の形式をとります。TCP および UDP データの文字列値を返す式プレフィクスについては、任意のテキストベースの操作を適用できます。詳細については、「高度なポリシー式: テキストの評価」を参照してください。

送信元ポートなどの数値を返す式プレフィクスについては、算術演算を適用できます。詳細については、「式の接頭辞の基本操作 」および「数値の複合演算」を参照してください。

次の表に、TCP および UDP データを抽出するプレフィクスを示します。

GET オペレーション	説明
CLIENT.TCP.PAYLOAD(<integer>)	TCP ペイロードデータを文字列として返します。ペイロードの最初の文字から始まり、<integer> 引数の文字数だけ継続します。このプレフィックスには、任意のテキストベースの操作を適用できます。
CLIENT.TCP.SRCPORT	現在のパケットの送信元ポートの ID を数値で返します。
CLIENT.TCP.DSTPORT	現在のパケットの宛先ポートの ID を数値で返します。

GET オペレーション	説明
CLIENT.TCP.OPTIONS	クライアントによって設定された TCP オプションを返します。TCP オプションの例としては、最大セグメントサイズ (MSS)、ウィンドウスケール、選択的確認応答 (SACK)、タイムスタンプオプションがあります。<type> <m> このプレフィックスには、COUNT、TYPE ()、および TYPE_NAME () 演算子を使用できます。サーバによって設定される TCP オプションについては、SERVER.TCP.OPTIONS プレフィクスを参照してください。
CLIENT.TCP.OPTIONS.COUNT	クライアントが設定した TCP オプションの数を返します。
CLIENT.TCP.OPTIONS.TYPE(<type>)	型 (またはオプションの種類) が引数として指定されている TCP オプションの値を返します。この値は、ビッグエンディアン形式 (またはネットワークバイト順序) のバイト文字列として返されます。パラメータ: タイプ-タイプ値
CLIENT.TCP.OPTIONS.TYPE_NAME(<m>)	列挙定数が引数として指定されている TCP オプションの値を返します。引数として渡すことができる列挙定数は、REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, MAXSEG です。これらの列挙定数の代わりに TCP オプションの種類を指定するには、CLIENT.TCP.OPTIONS.TYPE (<type>) を使用します。その他の TCP オプションについては、クライアント.TCP.OPTIONS.TYPE (<type>) を使用する必要があります。パラメータ:m-TCP オプション列挙定数
CLIENT.TCP.REPEATER_OPTION.EXISTS	リピータ TCP オプションが存在する場合、ブール型 TRUE を返します。
CLIENT.TCP.REPEATER_OPTION.IP	リピータ TCP オプションからブランチリピータの IPv4 アドレスを返します。
CLIENT.TCP.REPEATER_OPTION.MAC	リピータ TCP オプションからブランチリピータの MAC アドレスを返します。
CLIENT.UDP.DNS.DOMAIN	DNS ドメイン名を返します。

GET オペレーション	説明
CLIENT.UDP.DNS.DOMAIN.EQ(“<hostname>”)	ドメイン名が <hostname> 引数に一致する場合に、ブール型 TRUE を返します。比較では、大文字と小文字は区別されません。次に、クライアント.udp.dns.ドメイン.eq（「www.mycompany.com」）の例を示します。
CLIENT.UDP.DNS.IS_AAAAREC	レコードタイプが AAAA の場合、ブール型 TRUE を返します。これらのタイプのレコードは、前方参照の IPv6 アドレスを示します。
CLIENT.UDP.DNS.IS_ANYREC	任意のレコードタイプの場合に、ブール型 TRUE を返します。
CLIENT.UDP.DNS.IS_AREC	レコードがタイプ A の場合、ブール型 TRUE を返します。タイプ A レコードがホストアドレスを提供します。
CLIENT.UDP.DNS.IS_CNAMEREC	レコードのタイプが CNAME の場合は、ブール型 TRUE を返します。リソースを識別するために複数の名前を使用するシステムでは、正規名と 1 つのエイリアスがあります。CNAME は正規名を提供します。
CLIENT.UDP.DNS.IS_MXREC	レコードのタイプが MX (メールエクスチェンジャ) の場合は、ブール型 TRUE を返します。この DNS レコードは、優先順位とホスト名を表します。同じドメイン名の MX レコードは、ドメイン内の電子メールサーバーと各サーバーの優先順位を指定します。
CLIENT.UDP.DNS.IS_NSREC	レコードのタイプが NS の場合は、ブール型 TRUE を返します。これは、関連付けられた A レコードを持つホスト名を含むネームサーバレコードです。これにより、NS レコードに関連付けられているドメイン名を検索できるようになります。
CLIENT.UDP.DNS.IS_PTRREC	レコードのタイプが PTR の場合は、ブール型 TRUE を返します。これはドメイン名ポインタであり、ドメイン名を IPv4 アドレスに関連付けるためによく使用されます。
CLIENT.UDP.DNS.IS_SOAREC	レコードのタイプが SOA の場合は、ブール型 TRUE を返します。これは権威記録の始まりです。

GET オペレーション	説明
CLIENT.UDP.DNS.IS_SRVREC	レコードのタイプが SRV の場合は、ブール型 TRUE を返します。これは MX レコードのより一般的なバージョンです。
CLIENT.UDP.DSTPORT	現在のパケットの UDP 宛先ポートの数値 ID を返します。
CLIENT.UDP.SRCPORT	現在のパケットの UDP 送信元ポートの数値 ID を返します。
CLIENT.UDP.RADIUS	現在のパケットの RADIUS データを返します。
CLIENT.UDP.RADIUS.ATTR_TYPE(<type>)	引数として指定された属性タイプの値を返します。
CLIENT.UDP.RADIUS.USERNAME	RADIUS ユーザー名を返します。
CLIENT.TCP.MSS	現在の接続の最大セグメントサイズ (MSS) を数値で返します。
CLIENT.VLAN.ID	現在のパケットが Citrix ADC に入った VLAN の数値 ID を返します。
SERVER.TCP.DSTPORT	現在のパケットの宛先ポートの数値の ID を返します。
SERVER.TCP.SRCPORT	現在のパケットの送信元ポートの数値の ID を返します。
SERVER.TCP.OPTIONS	サーバーによって設定された TCP オプションを返します。TCP オプションの例としては、最大セグメントサイズ (MSS)、ウィンドウスケール、選択的確認応答 (SACK)、タイムスタンプオプションがあります。 <type> <m> このプレフィックスには、COUNT、TYPE ()、および TYPE_NAME () 演算子を使用できます。クライアントによって設定される TCP オプションについては、CLIENT.TCP.OPTIONS プレフィクスを参照してください。
SERVER.TCP.OPTIONS.COUNT	サーバーが設定した TCP オプションの数を返します。
SERVER.TCP.OPTIONS.TYPE(<type>)	型 (またはオプションの種類) が引数として指定されている TCP オプションの値を返します。この値は、ビッグエンディアン形式 (またはネットワークバイト順序) のバイト文字列として返されます。パラメータ: タイプ-タイプ値

GET オペレーション	説明
SERVER.TCP.OPTIONS.TYPE_NAME(<m>)	列挙定数が引数として指定されている TCP オプションの値を返します。引数として渡すことができる列挙定数は、REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, MAXSEG です。これらの列挙定数の代わりに TCP オプションの種類を指定するには、CLIENT.TCP.OPTIONS.TYPE (<type>) を使用します。その他の TCP オプションについては、クライアント.TCP.OPTIONS.TYPE (<type>) を使用する必要があります。パラメータ:m-TCP オプション列挙定数
SERVER.VLAN	現在のパケットが Citrix ADC に入った VLAN 上で動作する。
SERVER.VLAN.ID	現在のパケットが Citrix ADC に入った VLAN の数値 ID を返します。

DNS メッセージを評価し、そのキャリアプロトコルを識別するための式

October 7, 2021

DNS 要求と応答を評価するには、それぞれ DNS.REQ と DNS.RES で始まる式を使用します。DNS メッセージの送信に使用されているトランスポート層プロトコルを特定することもできます。

次の関数は、DNS クエリの内容を返します。

機能	説明
DNS.REQ.QUESTION.DOMAIN	DNS クエリの質問セクションでドメイン名 (QNAME フィールドの値) を返します。ドメイン名はテキスト文字列として返され、EQ ()、NE ()、およびテキストを扱うその他の関数に渡すことができます。

機能	説明
DNS.REQ.QUESTION.TYPE	DNS クエリのクエリの種類 (QTYPE フィールドの値) を返します。このフィールドは、ネームサーバが照会されるリソースレコードの種類 (A、NS、CNAME など) を示します。戻り値は、EQ () 関数および NE () 関数を使用して、次のいずれかの値と比較できます。A、AAAA、NS、SRV、PTR、CNAME、SOA、MX、および ANY。注:TYPE 関数では、EQ () 関数と NE () 関数のみを使用できます。Example: DNS.REQ.QUESTION.TYPE.EQ(MX)

次の関数は、DNS 応答の内容を返します。

機能	説明
DNS.RES.HEADER.RCODE	DNS 応答のヘッダーセクションで応答コード (RCODE フィールドの値) を返します。RCODE 関数では、EQ () 関数と NE () 関数のみを使用できます。次の可能な値は、エラー、フォーマット、SERVFAIL、NXDOMAIN、NOTIMP、および拒みました。
DNS.RES.QUESTION.DOMAIN	DNS レスポンスの質問セクションでドメイン名 (QNAME フィールドの値) を返します。ドメイン名はテキスト文字列として返され、EQ ()、NE ()、およびテキストを扱うその他の関数に渡すことができます。
DNS.RES.QUESTION.TYPE	DNS レスポンスの質問セクションでクエリタイプ（QTYPE フィールドの値）を返します。このフィールドは、応答に含まれるリソースレコードの種類 (A、NS、CNAME など) を示します。戻り値は、EQ () 関数および NE () 関数を使用して、次のいずれかの値と比較できます。A、AAAA、NS、SRV、PTR、CNAME、SOA、MX、および ANY。TYPE 関数では、EQ () 関数と NE () 関数のみを使用できます。Example: DNS.RES.QUESTION.TYPE.EQ(SOA)

次の関数は、トランスポート層のプロトコル名を返します。

機能	説明
DNS.REQ.TRANSPORT	DNS クエリの送信に使用されたトランスポート層プロトコルの名前を返します。返される値は TCP と UDP です。TRANSPORT 関数では、EQ () 関数と NE () 関数のみを使用できます。Example: DNS.REQ.TRANSPORT.EQ(TCP)
DNS.RES.TRANSPORT	DNS 応答に使用されたトランスポート層プロトコルの名前を返します。返される値は TCP と UDP です。TRANSPORT 関数では、EQ () 関数と NE () 関数のみを使用できます。Example: DNS.RES.TRANSPORT.EQ(TCP)

XPath および HTML、XML、または JSON の式

October 7, 2021

高度なポリシーインフラストラクチャでは、HTML、XML、および JavaScript オブジェクト記法 (JSON) ファイルからデータを評価および取得するための式がサポートされています。これにより、HTML、XML、または JSON ドキュメント内の特定のノードを検索し、ファイルにノードが存在するかどうかを判断し、XML コンテキスト内のノード (たとえば、特定の親を持つノード、または特定の値を持つ特定の属性を持つノード) を特定し、そのようなノードの内容を返すことができます。さらに、書き換え式で XPath 式を使用できます。

XPath の高度なポリシー式の実装は、HTML または XML テキストを指定する高度なポリシー式のプレフィックス (「HTTP.REQ.BODY」など) と、XPath 式を引数として受け取る XPATH 演算子で構成されます。

HTML ファイルは、タグとテキスト要素のほとんどが自由形式のコレクションです。XPATH_HTML 演算子を使用すると、引数として XPath 式を受け取り、HTML ファイルを処理できます。JSON ファイルは、名前と値のペアのコレクションまたは値の順序付きリストのいずれかです。XPATH_JSON 演算子を使用すると、引数として XPath 式を受け取り、JSON ファイルを処理できます。

- **<text>.XPATH (xpathex):**

  XML ファイルを操作し、ブール値を返します。

  たとえば、次の式は、XML ファイルの最初の 1000 バイト内に「creator」というノードが「Book」ノードの下に存在する場合、ブール値 TRUE を返します。

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%boolean(//Book/creator)%)
  ```

  パラメーター:

  xpathex-XPath ブール式

- **<text>.XPATH (xpathex):**

  XML ファイルを操作し、データ型「double」の値を返します。

  たとえば、次の式は、文字列が XML ファイルの最初の 1000 バイトにある場合、文字列「36」（価格値）をデータ型「double」の値に変換します。

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%number(/Book/price)%)
  ```

  パラメーター:

  xpathex-XPath 数値式

例:

```
    <Book>
    <creator>
        <Person>
            <name>Milton</name>
        </Person>
    </creator>
    <title>Paradise Lost</title>
    </Book>
<!--NeedCopy-->
```

- **<text>.XPATH (xpathex):**

  XML ファイルを操作し、ノードセットまたは文字列を返します。ノードセットは、標準の XPath 文字列変換ルーチンを使用して、対応する文字列に変換されます。

  たとえば、次のエクスプレッションは、本体の最初の 1000 バイトで「/Book/creator」(ノードセット) で囲まれているすべてのノードを選択します。

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%/Book/creator%)
  ```

  パラメーター:

  xpathex-XPath 式

- **<text>.XPATH_HTML(xpathex)**

  HTML ファイルを操作し、テキスト値を返します。

  たとえば、次の式は HTML ファイルに対して動作し、title HTML 要素が最初の 1000 バイトに見つかった場合、<title\></title\> タグで囲まれたテキストを返します。

  ```
  HTTP.REQ.BODY(1000).XPATH_HTML(xp%/html/head/title%)
  ```

  パラメーター:

  xpathex-XPath テキスト式

- **<text>.XPATH_HTML_WITH_MARKUP(xpathex)**

  HTML ファイルを操作し、文書の選択された部分全体を含む文字列を返します。これには、囲む要素タグなどのマークアップも含まれます。

  次の式は HTML ファイルに対して動作し <\ title>、マークアップを含むタグ内のすべてのコンテンツを選択します。

  ```
  HTTP.REQ.BODY(1000).XPATH_HTML_WITH_MARKUP( xp%/html/head/title%)
  ```

  式によって選択される HTML 本文の部分は、さらなる処理のためにマークされます。

  パラメーター:

  xpathex-XPath 式

- **<text>.XPATH_JSON(xpathex)**

  JSON ファイルを操作し、ブール値を返します。

  たとえば、次の JSON ファイルを考えてみます。

  { ”Book”:{ ”creator”:{ ”person”:{ ”name”:’<name>’ } }, ”title”:’<title>’ } }

  次の式は JSON ファイルを操作し、JSON ファイルの最初の 1000 バイトに「creator」という名前の親ノードが「Book」である場合に、ブール値 TRUE を返します。

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%boolean(/Book/creator)%)
  ```

  パラメーター:

  xpathex-XPath ブール式

- **<text>.XPATH_JSON(xpathex)**

  JSON ファイルを操作し、データ型「double」の値を返します。

  たとえば、次の JSON ファイルを考えてみます。

  {「本」: {「クリエイター」: {「人」: {「名前」:<name>”},「タイトル」: ‘<title>’,「価格」: ”36”}}

  次の式は JSON ファイルを操作し、JSON ファイルの最初の 1000 バイトに文字列が存在する場合、文字列「36」をデータ型「double」の値に変換します。

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%number(/Book/price)%)
  ```

  パラメーター:

  xpathex-XPath 数値式

- **<text>.XPATH_JSON(xpathex)**

  JSON ファイルを操作し、ノードセットまたは文字列を返します。ノードセットは、標準の XPath 文字列変換ルーチンを使用して、対応する文字列に変換されます。

  たとえば、次の JSON ファイルを考えてみます。

{「本」: {「クリエイター」: {「人」: {「名前」:<name>"},「タイトル」: '<title>' } }

次の式は、JSON ファイルの本体の最初の 1000 バイトの「/Book」(ノードセット) で囲まれたすべてのノードを選択し、対応する文字列値を返します。<name><title>":

```
HTTP.REQ.BODY(1000).XPATH_JSON(xp%/Book%)
```

パラメーター:

xpathex-XPath 式

- **<text>.XPATH_JSON_WITH_MARKUP(xpathex)**

  XML ファイルを操作し、結果ノードのドキュメント全体を含む文字列を返します。これには、囲む要素タグを含むマークアップも含まれます。

  たとえば、次の JSON ファイルを考えてみます。

  {「本」: {「クリエイター」: {「人」: {「名前」:<name>"},「タイトル」: '<title>' } }

  次の式は JSON ファイルを操作し、本体の最初の 1000 バイトで「/Book/creator」で囲まれているすべてのノードを選択します。このノードは"creator:{ person:{ name:'<name>' } }" です。

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON_WITH_MARKUP(xp%/Book/creator%)
  ```

  式によって選択される JSON 本体の部分は、さらなる処理のためにマークされます。

  パラメーター:

  xpathex-XPath 式

- **<text>.XPATH_WITH_MARKUP(xpathex):**

  XML ファイルを操作し、結果ノードのドキュメント全体を含む文字列を返します。これには、囲む要素タグを含むマークアップも含まれます。

  たとえば、次の式は XML ファイルに対して動作し、本文の最初の 1000 バイトの「/Book/creator」で囲まれたノードをすべて選択します。

  ```
  HTTP.REQ.BODY(1000).XPATH_WITH_MARKUP(xp%/Book/creator%)
  ```

  式によって選択される JSON 本体の部分は、さらなる処理のためにマークされます。

  パラメーター:

  xpathex-XPath 式

XML ペイロードの暗号化と復号化

October 7, 2021

高度なポリシー式で XML_ENCRYPT () 関数と XML_DECRYPT () 関数を使用して、XML データを暗号化および復号化できます。これらの関数は、http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/で定義されている W3C XML 暗号化標準に準拠しています。XML_ENCRYPT () および XML_DECRYPT () は、XML 暗号化仕様のサブセットをサポートします。サブセットでは、データ暗号化はバルク暗号方式 (RC4、DES3、AES128、AES192、または AES256) を使用し、RSA 公開キーを使用してバルク暗号キーを暗号化します。

注**:** ペイロード内のテキストを暗号化および復号化する場合は、ENCRYPT 関数と DECRYPT 関数を使用する必要があります。これらの関数の詳細については、テキストの暗号化と復号化を参照してください。

XML_ENCRYPT () 関数と XML_DECRYPT () 関数は、テキストの ENCRYPT コマンドおよび DECRYPT コマンドで使用される暗号化/復号化サービスに依存しません。暗号方式は、XML_ENCRYPT () 関数の引数として明示的に指定されています。このXML_DECRYPT( ) 関数は、<xenc:EncryptedData> 要素から指定された暗号メソッドに関する情報を取得します。次に、XML 暗号化および復号化関数の概要を示します。

- XML_ENCRYPT(<certKeyName>, <method> [, <flags>])**. Returns an <xenc:EncryptedData>要素には、暗号化された入力テキストと暗号化キーが含まれています。暗号化キー自体は RSA を使用して暗号化されます。
- XML_DECRYPT(<certKeyName>)。暗号方式と RSA 暗号化キーを含む入力<xenc:EncryptedData> 要素から復号化されたテキストを返します。

注**:** <xenc:EncryptedData>要素は、W3C XML 暗号化仕様で定義されています。

次に、引数の説明を示します。

- **certKeyName**: XML_ENCRYPT () の場合は RSA パブリックキー、または XML_DECRYPT () の場合は RSA プライベートキーを含む X.509 証明書を選択します。証明書キーは、add ssl certKey コマンドで事前に作成されている必要があります。
- **method:** XML データの暗号化に使用する暗号化方法を指定します。可能な値: RC4, DES3, AES128, AES192, AES256.
- **flags :**XML_ENCRYPT( ) によって生成される<xenc:EncryptedData> 要素に含まれる次のオプションのキー情報 (<ds:KeyInfo>) を指定するビットマスク。
  - **1** -証明書キー名を持つキー名要素を含めます。エレメントは<ds:KeyName>です。
  - **2** -証明書からの RSA 公開キーとともに KeyValue 要素を含めます。エレメントは<ds:KeyValue>です。
  - **4** -証明書のシリアル番号と発行者 DN とともに X509IssuerSerial 要素を含めます。エレメントは<ds:X509IssuserSerial>です。
  - **8** -証明書のサブジェクト名 DN とともに X509 サブジェクト名要素を含めます。エレメントは<ds:X509SubjectName>です。
  - **16** -証明書全体に X509Certificate 要素を含めます。エレメントは<ds:X509Certificate>です。

式で **XML_ENCRYPT ()** 関数と **XML_DECRYPT ()** 関数を使用します

XML 暗号化機能では、SSL 証明書とキーのペアを使用して、キーの暗号化に X.509 証明書（RSA 公開キー付き）、キーの復号化に RSA 秘密キーを提供します。したがって、式で XML_ENCRYPT () 関数を使用する前に、SSL 証明書とキーのペアを作成する必要があります。次のコマンドは、X.509 証明書 my-cert.pem、プライベートキーファイル my-key.pem を使用して、SSL 証明書とキーのペアである my-certkey を作成します。

```
add ssl certKey my-certkey -cert my-cert.pem -key my-key.pem -passcrypt
kxPeMRYnitY=
```

次の CLI コマンドは、XML コンテンツの暗号化と復号化のための書き換えアクションとポリシーを作成します。

```
1  add rewrite action my-xml-encrypt-action replace "HTTP.RES.BODY(10000).
       XPATH_WITH_MARKUP(xp%/%)" "HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp
       %/%).XML_ENCRYPT("my-certkey", AES256, 31)" -bypassSafetyCheck YES
2
3  add rewrite action my-xml-decrypt-action replace "HTTP.REQ.BODY(10000).
       XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)" "HTTP.REQ.BODY(10000).
       XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%).XML_DECRYPT("my-certkey"
       )" -bypassSafetyCheck YES
4
5  add rewrite policy my-xml-encrypt-policy "HTTP.REQ.URL.CONTAINS("xml-
       encrypt")" my-xml-encrypt-action
6
7  add rewrite policy my-xml-decrypt-policy "HTTP.REQ.BODY(10000).XPATH(xp
       %boolean(//xenc:EncryptedData)%)" my-xml-decrypt-action
8
9  bind rewrite global my-xml-encrypt-policy 30
10
11 bind rewrite global my-xml-decrypt-policy 30
12 <!--NeedCopy-->
```

上記の例では、書き換えアクション my-xml-encrypt-action は、AES-256 バルク暗号化方式と my-certkey からの RSA 公開キーを使用して、リクエストの XML ドキュメント全体 (XPATH_WITH_MARKUP (xp%/%)) を暗号化します。アクションは、暗号化されたデータと暗号化されたキーを含む<xenc:EncryptedData> エレメントで文書を置き換えます。31 で表されるフラグには、すべてのオプション<ds:KeyInfo> 要素が含まれます。

アクション my-xml-decrypt-action は、応答の最初の<xenc:EncryptedData> 要素を復号化します (XPATH_WITH_MARKUP (xp%//xenc: 暗号化データ%))。これには、次の CLI コマンドを使用して、xenc XML 名前空間を事前に追加する必要があります。

```
add ns xmlnamespace xenc http://www.w3.org/2001/04/xmlenc##
```

my-xml-decrypt-action アクションは、my-certkey の RSA 秘密キーを使用して暗号化されたキーを復号化し、要素で指定された一括暗号化方式を使用して暗号化された内容を復号化します。最後に、アクションによって、暗号

化されたデータ要素が復号化されたコンテンツに置き換えられます。

書き換えポリシー my-xml-encrypt-policy は、xml 暗号化を含む URL のリクエストに my-xml-encrypt-action を適用します。アクションは、Citrix ADC アプライアンスで構成されたサービスからの応答全体を暗号化します。

書き換えポリシー my-xml-decrypt-policy は、<xenc:EncryptedData> 要素を含むリクエストに my-xml-decrypt-アクションを適用します ((XPATH (xp%//xenc:EncryptedData%) は空でない文字列を返します)。アクションは、Citrix ADC アプライアンスで構成されたサービスにバインドされた要求内の暗号化されたデータを復号化します。

高度なポリシー式:SSL の解析

October 7, 2021

SSL 証明書と SSL クライアントの hello メッセージを解析するための高度なポリシー式があります。

SSL 証明書の解析

高度なポリシー式を使用して、X.509 SSL (セキュアソケットレイヤー) クライアント証明書を評価できます。クライアント証明書は、ユーザーの ID を認証するために使用できる電子ドキュメントです。クライアント証明書には、少なくとも、バージョン情報、シリアル番号、署名アルゴリズム ID、発行者名、有効期間、サブジェクト（ユーザー）名、公開キー、および署名が含まれています。

SSL 接続とクライアント証明書内のデータの両方を調べることができます。たとえば、低強度の暗号を使用する SSL 要求を特定の負荷分散仮想サーバーファームに送信できます。次のコマンドは、要求内の暗号強度を解析し、40 以下の暗号強度を照合するコンテンツスイッチングポリシーの例です。

```
1 add cs policy p1 -rule ”client.ssl.cipher_bits.le(40)”
2 <!--NeedCopy-->
```

別の例として、要求にクライアント証明書が含まれているかどうかを決定するポリシーを設定できます。

```
1 add cs policy p2 -rule ”client.ssl.client_cert exists”
2 <!--NeedCopy-->
```

または、クライアント証明書の特定の情報を検査するポリシーを構成できます。たとえば、次のポリシーでは、証明書の有効期限が 1 日以上前にあることを確認します。


```
add cs policy p2 -rule ”client.ssl.client_cert exists && client.ssl.
    client_cert.days_to_expire.ge(1)”
<!--NeedCopy-->
```

> 注
>
> 証明書の日付と時刻の解析については、「式内の日付と時刻のフォーマット」および 「SSL 証明書の日付の式」を参照してください。

テキストベースの **SSL** および証明書データのプレフィックス

次の表では、SSL トランザクションとクライアント証明書でテキストベースの項目を識別する式プレフィックスについて説明します。

表 1. SSL およびクライアント証明書データのテキストまたはブール値を返すプレフィックス

前	説明
CLIENT.SSL.CLIENT_CERT	現在の SSL トランザクションの SSL クライアント証明書を返します。
CLIENT.SSL.CLIENT_CERT.TO_PEM	SSL クライアント証明書をバイナリ形式で返します。
CLIENT.SSL.CIPHER_EXPORTABLE	SSL 暗号暗号がエクスポート可能な場合は、ブール型 TRUE を返します。
CLIENT.SSL.CIPHER_NAME	SSL 接続から呼び出された場合は SSL 暗号の名前を返し、非 SSL 接続から呼び出された場合は NULL 文字列を返します。
CLIENT.SSL.IS_SSL	現在の接続が SSL ベースの場合に TRUE を返します。

SSL 証明書の数値データのプレフィックス

次の表では、SSL 証明書の日付以外の数値データを評価するプレフィックスについて説明します。これらのプレフィックスは、[式接頭辞の基本操作と数値の複合演算で説明されている操作で使用できます](/ja-jp/citrix-adc/13/appexpert/policies-and-expressions/adv-policy-expressions-getting-started/compound-advanced-policy-expressions.html)。

表 2. SSL 証明書の日付以外の数値データを評価するプレフィックス

前	説明
CLIENT.SSL.CLIENT_CERT.DAYS_TO_EXPIRE	証明書が有効である日数を返します。期限切れの証明書の場合は-1 を返します。

前	説明
CLIENT.SSL.CLIENT_CERT.PK_SIZE	証明書で使用されている公開キーのサイズを返します。
CLIENT.SSL.CLIENT_CERT.VERSION	証明書のバージョン番号を返します。接続が SSL ベースでない場合は、ゼロ (0) を返します。
CLIENT.SSL.CIPHER_BITS	暗号化キーのビット数を返します。接続が SSL ベースでない場合は 0 を返します。
CLIENT.SSL.VERSION	次のように、SSL プロトコルのバージョンを表す数値を返します。0。トランザクションは SSL ベースではありません。; 0x002。トランザクションは SSLv2 です。トランザクションは SSLv3 です。0x301。トランザクションは TLSv1 です。トランザクションは TLS 1.1 です。トランザクションは TLS 1.2 です。0x304 です。トランザクションは TLS 1.3 です。

> 注
>
> 証明書の有効期限に関連する式については、「SSL 証明書日付の式」を参照してください。

SSL 証明書の式

SSL 証明書を解析するには、次のプレフィックスを使用する式を設定します。

CLIENT.SSL.CLIENT_CERT

ここでは、証明書に対して構成できる式について説明します。ただし、証明書の有効期限を調べる式は除きます。時間ベースの操作については、「高度なポリシー式: 日付、時刻、および数字の操作」を参照してください。

次の表に、CLIENT.SSL.CLIENT_CERT プレフィックスに指定できるオペレーションを示します。

表 3. クライアント.SSL.CLIENT_CERT プレフィックスで指定できるオペレーション

SSL 証明書の操作	説明
`<certificate>.EXISTS`	クライアントに SSL 証明書がある場合は、ブール型 TRUE を返します。

SSL 証明書の操作	説明
<certificate>.ISSUER	証明書内の発行者の識別名 (DN) を名前/値リストとして返します。等号 (「=」) は名前と値の区切り文字で、スラッシュ (「/」) は名前と値のペアを区切る区切り文字です。返される DN の例を次に示します。 /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.com
<certificate>.ISSUER.IGNORE_EMPTY_ELEMENTS	Issuer を返し、名前/値リストの空の要素を無視します。たとえば、Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com を考えてみましょう。次の Rewrite アクションは、前の発行者の定義に基づいて 6 のカウントを返します。 sh rewrite action insert_ssl_header Name: insert_ssl Operation: insert_http_header Target:Cert-Issuer Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT。ただし、値を次のように変更すると、返されるカウントは 9 です。 CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT

SSL クライアントを解析する

SSL クライアントの hello メッセージを解析するには、次のプレフィックスを使用する式を設定します。

前	説明
CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXC(	式で指定された 16 進コードを、クライアントの hello メッセージで受信した暗号スイートの 16 進コードと一致させます。
CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION	クライアントの hello メッセージヘッダーで受信したバージョン。
CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE	クライアントまたはサーバーがセッションの再ネゴシエーションを開始する場合は true を返します。

前	説明
CLIENT.SSL.CLIENT_HELLO.IS_REUSE	アプライアンスが、クライアント hello メッセージで受信した 0 以外のセッション ID に基づいて SSL セッションを再利用する場合、true を返します。
CLIENT.SSL.CLIENT_HELLO.IS_SCSV	シグナリング暗号スイート値（SCSV）機能がクライアントの hello メッセージでアドバタイズされている場合は true を返します。フォールバック SCSV の 16 進コードは 0x5600 です。
CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET	0 以外の長さのセッションチケット拡張が、クライアント hello メッセージでアドバタイズされている場合は true を返します。
CLIENT.SSL.CLIENT_HELLO.LENGTH	クライアントの hello メッセージヘッダーで受信された長さ。
CLIENT.SSL.CLIENT_HELLO.SNI	クライアント hello メッセージのサーバ名拡張子で受信したサーバ名を返します。
CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPRC	クライアント hello メッセージで受信された ALPN 拡張内のアプリケーションプロトコルが、式で指定されたプロトコルと一致する場合は true を返します。

これらの式は、CLIENTHELLO_REQ バインドポイントで使用できます。詳細については、SSL ポリシーバインディングを参照してください。

高度なポリシー式：**IP** アドレスと **MAC** アドレス、スループット、**VLAN ID**

October 7, 2021

IPv4 および IPv6 アドレス、MAC アドレス、IP サブネット、インターフェイスポートでのスループットレート（Rx、Tx、RxTx）、およびパケットを受信する VLAN の ID などの有用なクライアントおよびサーバデータを返す高度なポリシー式プレフィクスを使用できます。その後、さまざまな演算子を使用して、これらの式のプレフィックスによって返されるデータを評価できます。

IP アドレスおよび **IP** サブネットの式

高度なポリシー式を使用して、インターネットプロトコルバージョン 4 (IPv4) またはインターネットプロトコルバージョン 6 (IPv6) 形式のアドレスとサブネットを評価できます。IPv6 アドレスおよびサブネットの式プレフィクスに

は、プレフィクスに IPv6 が含まれます。IPv4 アドレスおよびサブネットの式プレフィクスには、プレフィクスに IP が含まれます。次に、要求が特定の IPv4 サブネットから発信されたかどうかを識別する式の例を示します。

```
client.ip.src.in_subnet(147.1.0.0/16)
<!--NeedCopy-->
```

次に、パケットの受信元のサブネットを調べて、Host ヘッダーで書き換えアクションを実行する、書き換えポリシーの例を示します。これら 2 つのポリシーが設定されている場合、実行される書き換えアクションはリクエストのサブネットによって異なります。これら 2 つのポリシーは、IPv4 アドレス形式の IP アドレスを評価します。

```
- add rewrite action URL1-rewrite-action replace "http.req.header("Host")" ""www.mycompany1.com""
- add rewrite policy URL1-rewrite-policy "http.req.header("Host").contains("www.test1.com") && client.ip.src.in_subnet(147.1.0.0/16)" URL1-rewrite-action
- add rewrite action URL2-rewrite-action replace "http.req.header("Host")" ""www.mycompany2.com""
- add rewrite policy URL2-rewrite-policy "http.req.header("Host").contains("www.test2.com") && client.ip.src.in_subnet(10.202.0.0/16)" URL2-rewrite-action
<!--NeedCopy-->
```

> 注
>
> 上記の例は、Citrix ADC コマンドラインインターフェイス（CLI）で入力するコマンドです。したがって、各引用符の前にバックスラッシュ（\）を付ける必要があります。詳細については、「ポリシーでの高度なポリシー式の設定」を参照してください。

IPV4 アドレスおよび IP サブネットのプレフィクス

次の表に、IPv4 アドレスとサブネットを返すプレフィックス、および IPv4 アドレスのセグメントを示します。IPv4 アドレスに固有の数値演算子と演算子を、これらのプレフィックスとともに使用できます。数値演算の詳細については、「式接頭辞の基本操作」および「数値の複合演算」を参照してください。

表 1. IP アドレスと MAC アドレスを評価するプレフィクス

前	説明
CLIENT.IP.SRC	現在のパケットの送信元 IP を IP アドレスまたは数値で返します。

前	説明
CLIENT.IP.DST	現在のパケットの宛先 IP を IP アドレスまたは数値で返します。
SERVER.IP.SRC	現在のパケットの送信元 IP を IP アドレスまたは数値で返します。
SERVER.IP.DST	現在のパケットの宛先 IP を IP アドレスまたは数値で返します。

IPV4 アドレスの操作

IPV4 操作のプレフィクス表では、IPv4 アドレスを返すプレフィクスで使用できる演算子について説明します。

IPv6 式について

IPv6 アドレス形式では、以前の IPv4 形式よりも柔軟性が高くなります。IPv6 アドレスは 16 進数形式（RFC 2373）です。次の例では、例 1 は IPv6 アドレス、例 2 は IPv6 アドレスを含む URL、例 3 は IPv6 アドレスとポート番号を含んでいます。

例 **1**:

```
9901:0ab1:22a2:88a3:3333:4a4b:5555:6666
<!--NeedCopy-->
```

例 **2**:

```
http://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]/
<!--NeedCopy-->
```

例 **3:**

```
https://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]:8080/
<!--NeedCopy-->
```

例 3 では、IP アドレスとポート番号（8080）を括弧で区切ります。

IPv6 式を他の式と組み合わせるには、'+' 演算子のみを使用できます。出力は、個々の式から返される文字列値の連結です。IPv6 式では、他の算術演算子は使用できません。次の構文は例です。


```
client.ipv6.src + server.ip.dst
<!--NeedCopy-->
```

たとえば、クライアントの送信元 IPv6 アドレスがABCD:1234::ABCDで、サーバの宛先 IPv4 アドレスが10.100.10.100の場合、前の式は”ABCD:1234::ABCD10.100.10.100”を返します。

Citrix ADC アプライアンスが IPv6 パケットを受信すると、未使用の IPv4 アドレス範囲から一時的な IPv4 アドレスを割り当て、パケットの送信元アドレスをこの一時アドレスに変更します。応答時に、発信パケットの送信元アドレスは元の IPv6 アドレスに置き換えられます。

> 注
>
> IPv6 式は、ブール結果を生成する式を除く他の式と組み合わせることができます。

IPv6 アドレスの式プレフィックス

次の表の式プレフィックスによって返される IPv6 アドレスは、テキストデータとして扱うことができます。たとえば、プレフィクス client.ipv6.dst は、宛先 IPv6 アドレスをテキストとして評価できる文字列として返します。

次の表に、IPv6 アドレスを返す式プレフィックスについて説明します。

表 3. テキストを返す IPv6 式プレフィックス

前	説明
CLIENT.IPV6	現在のパケットでの IPv6 アドレス上で動作する。
CLIENT.IPV6.DST	IP ヘッダーの宛先フィールドの IPv6 アドレスを返します。
CLIENT.IPV6.SRC	IP ヘッダーの送信元フィールドの IPv6 アドレスを返します。以下はその例です。`client.ipv6.src.in_subnet(2007::2008/64)` `client.ipv6.src.get1.le(2008)`
SERVER.IPV6	現在のパケットでの IPv6 アドレス上で動作する。
SERVER.IPV6.DST	IP ヘッダーの宛先フィールドの IPv6 アドレスを返します。
SERVER.IPV6.SRC	IP ヘッダーの送信元フィールドの IPv6 アドレスを返します。以下はその例です。`server.ipv6.src.in_subnet(2007::2008/64)` `server.ipv6.src.get1.le(2008)`

IPv6 プレフィクスの操作

次の表に、IPv6 アドレスを返すプレフィックスで使用できる演算子を示します。

表 4. IPv6 アドレスを評価する操作

IPv6 オペレーション	説明
<ipv6>.EQ(<IPv6_address>	IP アドレス値が<IPv6_address> 引数と同じ場合は、ブール型 TRUE を返します。以下はその例です。client.ipv6.dst.eq(ABCD:1234::ABCD)
<ipv6>.GET1. . .GET8	IPv6 アドレスのセグメントを数値で返します。次の例では、アドレスの 5 番目のビットセットである ipv6 アドレス 1000:1001:CD 10:0000:89 AB:4567:CDEF:client.ipv6.dst.get5 extracts 0000からセグメントを取得します。client.ipv6.dst.get6 extracts 89AB. client.ipv6.dst.get7 extracts 4567. これらのセグメントに対して数値演算を実行できます。IPv6 アドレス全体を取得する場合は、数値演算を実行できないことに注意してください。これは、CLIENT.IPV6.SRC などの IPv6 アドレス全体を返す式が、アドレスをテキスト形式で返すためです。
<ipv6>.IN_SUBNET(<subnet>)	<subnet>引数で指定されたサブネットに IPv6 アドレス値がある場合、ブール型 TRUE を返します。以下はその例です。client.ipv6.dst.eq(1000:1001:CD10:0000:0000:89AB:4567:CDEF/60)
<ipv6>.IS_IPV4	IPv4 クライアントの場合はブール値 TRUE を返し、そうでない場合はブール値 FALSE を返します。
<ipv6>.SUBNET(<n>)	引数として指定されたサブネットマスクを適用した後、IPv6 アドレスを返します。サブネットマスクには、0 ～128 の値を指定できます。たとえば、次のようになります: CLIENT.IPV6.SRC.SUBNET(24)

MAC アドレスの式

MAC アドレスは ##: ##: ##: ##: ##: ##: ##: ## ## の形式でコロンで区切られた 16 進値で構成されます。各「#」は、0 ～9 の数値または A ～F の文字を表します。送信元および宛先 MAC アドレスの評価には、デフォルトの構文式のプレフィクスと演算子を使用できます。

MAC アドレスのプレフィクス

次の表に、MAC アドレスを返すプレフィクスを示します。

表 5. MAC アドレスを評価するプレフィクス

前	説明
client.ether.dstmac	イーサネットヘッダーの宛先フィールドの MAC アドレスを返します。
client.ether.srcmac	イーサネットヘッダーの送信元フィールドの MAC アドレスを返します。

MAC アドレスの操作

次の表に、MAC アドレスを返すプレフィックスで使用できる演算子を示します。

表 6. MAC アドレスでの操作

前	説明
<mac address>.EQ(<address>)	MAC アドレス値が<address> 引数と同じ場合は、ブール型 TRUE を返します。
<mac address>.GET1. . .GET4	GET 操作で指定された MAC アドレスのセグメントから抽出された数値を返します。たとえば、MAC アドレスが 12:34:56:78:9 a: bc の場合、次は 34 を返します。client.ether.dstmac.get2

数値クライアントおよびサーバデータの式

次の表では、スループット、ポート番号、VLAN ID など、数値クライアントおよびサーバデータを操作するためのプレフィクスについて説明します。

表 7. 数値クライアントおよびサーバデータを評価するプレフィックス

前	説明
client.interface.rxthroughput	過去 7 秒間の raw 受信トラフィックのスループット（KBps）を表す整数を返します。
client.interface.txthroughput	過去 7 秒間の raw 送信トラフィックのスループット（KBps）を表す整数を返します。

前	説明
client.interface.rxtxthroughput	過去 7 秒間の未加工の送受信トラフィックのスループット（KBps）を表す整数を返します。
server.interface.rxthroughput	過去 7 秒間の raw 受信トラフィックのスループット（KBps）を表す整数を返します。
server.interface.txthroughput	過去 7 秒間の raw 送信トラフィックのスループット（KBps）を表す整数を返します。
server.interface.rxtxthroughput	過去 7 秒間の未加工の送受信トラフィックのスループット（KBps）を表す整数を返します。
server.vlan.id	現在のパケットが Citrix ADC に入った VLAN の数値 ID を返します。
クライアント.vlan.id	現在のパケットが Citrix ADC に入った VLAN の数値 ID を返します。

高度なポリシー式**:** ストリーム分析関数

October 7, 2021

ストリーム分析式は、ANALYTICS.STREAM (<identifier_name>) プレフィックスで始まります。次のリストでは、このプレフィックスで使用できる関数について説明します。

- **COLLECT_STATS**

  ポリシーに対して評価されるリクエストから統計データを収集し、リクエストごとにレコードを作成します。

- **REQUESTS**

  指定されたレコードのグループ化に対して存在する要求の数を返します。返される値は、符号なし long 型です。

- 帯域幅

  指定されたレコードグループの帯域幅統計情報を返します。返される値は、符号なし long 型です。

- **RESPTIME**

  指定されたレコードのグループ化の応答時間の統計を返します。返される値は、符号なし long 型です。

- **CONNECTIONS**

  指定されたレコードのグループ化に対して存在する同時接続の数を返します。返される値は、符号なし long 型です。

- **IS_TOP(n)**

  指定したレコードグループの統計値が上位 n 個のグループのうちの 1 つである場合、ブール型 (Boolean) の値を返します。それ以外の場合は、ブール値 FALSE を返します。

- **CHECK_LIMIT**

  指定したレコードグループの統計が、事前設定された制限に達した場合、ブール型 (Boolean) の値を返します。それ以外の場合は、ブール値 FALSE を返します。

高度なポリシー式**:DataStream**

October 7, 2021

Citrix ADC アプライアンスのポリシーインフラストラクチャには、アプリケーションサーバーのファームと関連するデータベースサーバーの間でアプライアンスを展開するときに、データベースサーバーのトラフィックを評価および処理するために使用できる式が含まれています。

このセクションでは、以下のトピックについて説明します:

- MySQL プロトコルの式
- SQL サーバーの接続を評価するための式

MySQL プロトコルの式

次の式は、MySQL データベースサーバーに関連付けられたトラフィックを評価します。ポリシーで要求ベースの式 (MYSQL.CLIENT および MYSQL.REQ で始まる式) を使用して、コンテンツスイッチング仮想サーバーのバインドポイントで要求切り替えの決定を行い、応答ベースの式 (MYSQL.RES で始まる式) を使用して、ユーザーに対するサーバー応答を評価できます。構成されたヘルスモニタ。

- **MYSQL.CLIENT.** MySQL 接続のクライアントプロパティで操作します。
- **MYSQL.CLIENT.CAPABILITIES.** 認証中にクライアントがハンドシェイク初期化パケットの capabilities フィールドに設定したフラグのセットを返します。設定されるフラグの例としては、「クライアント検索」、「クライアント圧縮」、および「クライアント SSL」があります。
- **MYSQL.CLIENT.CHAR_SET.** クライアントが使用する文字セットに割り当てられた列挙定数を返します。<m> <m> 比較の結果を示すブール値を返す EQ () および NE () 演算子は、このプレフィックスとともに使用されます。文字セット列挙定数は次のとおりです。
  - LATIN2_CZECH_CS
  - DEC8_SWEDISH_CI
  - CP850_GENERAL_CI
  - GREEK_GENERAL_CI
  - LATIN1_GERMAN1_CI

- HP8_ENGLISH_CI
- KOI8R_GENERAL_CI
- LATIN1_SWEDISH_CI
- LATIN2_GENERAL_CI
- SWE7_SWEDISH_CI
- ASCII_GENERAL_CI
- CP1251_BULGARIAN_CI
- LATIN1_DANISH_CI
- HEBREW_GENERAL_CI
- LATIN7_ESTONIAN_CS
- LATIN2_HUNGARIAN_CI
- KOI8U_GENERAL_CI
- CP1251_UKRAINIAN_CI
- CP1250_GENERAL_CI
- LATIN2_CROATIAN_CI
- CP1257_LITHUANIAN_CI
- LATIN5_TURKISH_CI
- LATIN1_GERMAN2_CI
- ARMSCII8_GENERAL_CI
- UTF8_GENERAL_CI
- CP1250_CZECH_CS
- CP866_GENERAL_CI
- KEYBCS2_GENERAL_CI
- MACCE_GENERAL_CI
- MACROMAN_GENERAL_CI
- CP852_GENERAL_CI
- LATIN7_GENERAL_CI
- LATIN7_GENERAL_CS
- MACCE_BIN
- CP1250_CROATIAN_CI
- LATIN1_BIN
- LATIN1_GENERAL_CI
- LATIN1_GENERAL_CS
- CP1251_BIN
- CP1251_GENERAL_CI
- CP1251_GENERAL_CS
- MACROMAN_BIN
- CP1256_GENERAL_CI
- CP1257_BIN

- CP1257_GENERAL_CI
- ARMSCII8_BIN
- ASCII_BIN
- CP1250_BIN
- CP1256_BIN
- CP866_BIN
- DEC8_BIN
- GREEK_BIN
- HEBREW_BIN
- HP8_BIN
- KEYBCS2_BIN
- KOI8R_BIN
- KOI8U_BIN
- LATIN2_BIN
- LATIN5_BIN
- LATIN7_BIN
- CP850_BIN
- CP852_BIN
- SWE7_BIN
- UTF8_BIN
- GEOSTD8_GENERAL_CI
- GEOSTD8_BIN
- LATIN1_SPANISH_CI
- UTF8_UNICODE_CI
- UTF8_ICELANDIC_CI
- UTF8_LATVIAN_CI
- UTF8_ROMANIAN_CI
- UTF8_SLOVENIAN_CI
- UTF8_POLISH_CI
- UTF8_ESTONIAN_CI
- UTF8_SPANISH_CI
- UTF8_SWEDISH_CI
- UTF8_TURKISH_CI
- UTF8_CZECH_CI
- UTF8_DANISH_CI
- UTF8_LITHUANIAN_CI
- UTF8_SLOVAK_CI
- UTF8_SPANISH2_CI
- UTF8_ROMAN_CI

  - UTF8_PERSIAN_CI
  - UTF8_ESPERANTO_CI
  - UTF8_HUNGARIAN_CI
  - INVAL_CHARSET

- **MYSQL.CLIENT.DATABASE.** クライアントがデータベースサーバーに送信する認証パケットで指定されたデータベースの名前を返します。これは、データベース名属性です。
- MYSQL.CLIENT.USER。クライアントがデータベースに接続しようとしている (認証パケット内の) ユーザー名を返します。これはユーザー属性です。
- **MYSQL.REQ.** MySQL リクエストで操作します。
- **MYSQL.REQ.COMMAND.** 要求内のコマンドの種類に割り当てられた列挙定数を指定します。<m> <m> 比較の結果を示すブール値を返す EQ () および NE () 演算子は、このプレフィックスとともに使用されます。列挙定数値を次に示します。

  - SLEEP
  - QUIT
  - INIT_DB
  - QUERY
  - FIELD_LIST
  - CREATE_DB
  - DROP_DB
  - REFRESH
  - SHUTDOWN
  - STATISTICS
  - PROCESS_INFO
  - CONNECT
  - PROCESS_KILL
  - デバッグ
  - PING
  - TIME
  - DELAYED_INSERT
  - CHANGE_USER
  - BINLOG_DUMP
  - TABLE_DUMP
  - CONNECT_OUT
  - REGISTER_SLAVE
  - STMT_PREPARE
  - STMT_EXECUTE
  - STMT_SEND_LONG_DATA
  - STMT_CLOSE

- STMT_RESET
- SET_OPTION
- STMT_FETCH

- **MYSQL.REQ.QUERY.** MySQL リクエスト内のクエリを識別します。
- **MYSQL.REQ.QUERY.COMMAND.** MySQL クエリの最初のキーワードを返します。
- **MYSQL.REQ.QUERY.SIZE.** リクエストクエリのサイズを整数形式で返します。SIZE メソッドは、HTTP リクエストまたはレスポンスの長さを返す CONTENT_LENGTH メソッドに似ています。
- **MYSQL.REQ.QUERY.TEXT.** クエリ全体をカバーする文字列を返します。
- **MYSQL.REQ.QUERY.TEXT(<n>).** MySQL クエリの最初の n バイトを文字列として返します。これは HTTP.BODY（<n>）に似ています。

  パラメーター:

  n-返されるバイト数

- **MYSQL.RES** MySQL の応答で操作します。
- **MYSQL.RES.ATLEAST_ROWS_COUNT(<i>).** レスポンスに i 以上の行数があるかどうかを調べ、結果を示すブール値 TRUE または FALSE を返します。

  パラメーター:

  i-行数

- **MYSQL.RES.ERROR.** MySQL エラーオブジェクトを識別します。エラーオブジェクトには、エラー番号とエラーメッセージが含まれます。
- **MYSQL.RES.ERROR.MESSAGE.** サーバーのエラー応答から取得したエラーメッセージを返します。
- **MYSQL.RES.ERROR.NUM.** サーバーのエラー応答から取得したエラー番号を返します。
- **MYSQL.RES.ERROR.SQLSTATE.** サーバーのエラー応答の SQLSTATE フィールドの値を返します。MySQL サーバーは、エラー番号の値を SQLSTATE 値に変換します。
- **MYSQL.RES.FIELD(<i>).** i^{番目に対応するパケットを識別します。} サーバーの応答内の個々のフィールド。各フィールドパケットは、関連付けられたカラムのプロパティを記述します。パケット数（i）は 0 から始まります。

  パラメーター:

  i-パケット番号

- **MYSQL.RES.FIELD(<i>).CATALOG.** フィールドパケットのカタログプロパティを返します。
- **MYSQL.RES.FIELD(<i>).CHAR_SET.** カラムの文字セットを返します。<m> <m> 比較の結果を示すブール値を返す EQ () および NE () 演算子は、このプレフィックスとともに使用されます。

- **MYSQL.RES.FIELD(<i>).DATATYPE.** カラムのデータ型を表す列挙定数を返します。これは、カラムのタイプ（enum_field_type とも呼ばれる）属性です。<m> <m> 比較の結果を示すブール値を返す EQ () および NE () 演算子は、このプレフィックスとともに使用されます。さまざまなデータタイプに使用できる値は次のとおりです。
  - DECIMAL
  - TINY
  - SHORT
  - LONG
  - FLOAT
  - DOUBLE
  - NULL
  - TIMESTAMP
  - LONGLONG
  - INT24
  - DATE
  - TIME
  - DATETIME
  - YEAR
  - NEWDATE
  - VARCHAR（MySQL 5.0 の新機能）
  - BIT（MySQL 5.0 の新機能）
  - NEWDECIMAL（MySQL 5.0 の新機能）
  - ENUM
  - SET
  - TINY_BLOB
  - MEDIUM_BLOB
  - LONG_BLOB
  - BLOB
  - VAR_STRING
  - STRING
  - GEOMETRY
- **MYSQL.RES.FIELD(<i>).DB.** フィールドパケットのデータベース識別子（db）属性を返します。
- **MYSQL.RES.FIELD(<i>).DECIMALS.** 型が DECIMAL または NUMERIC の場合、小数点以下の桁数を返します。これは、フィールドパケットの小数点以下の属性です。
- **MYSQL.RES.FIELD(<i>).FLAGS.** フィールドパケットの flags プロパティを返します。次に、16 進数のフラグ値を示します。
  - 0001: NOT_NULL_FLAG
  - 0002: PRI_KEY_FLAG

  - 0004: UNIQUE_KEY_FLAG
  - 0008: MULTIPLE_KEY_FLAG
  - 0010: BLOB_FLAG
  - 0020: UNSIGNED_FLAG
  - 0040: ZEROFILL_FLAG
  - 0080: BINARY_FLAG
  - 0100: ENUM_FLAG
  - 0200: AUTO_INCREMENT_FLAG
  - 0400: TIMESTAMP_FLAG
  - 0800: SET_FLAG

- **MYSQL.RES.FIELD(<i>).LENGTH.** カラムの長さを返します。これは、フィールドパケットの length 属性の値です。返される値は、実際の値よりも大きい場合があります。たとえば、VARCHAR (2) 列のインスタンスは、文字が 1 つしかない場合でも、値 2 を返すことがあります。
- **MYSQL.RES.FIELD(<i>).NAME.** カラム識別子（AS 句の後の名前）を返します。これは、フィールドパケットの name 属性です。
- **MYSQL.RES.FIELD(<i>).ORIGINAL_NAME.** 元のカラム ID (AS 句がある場合) を返します。これは、フィールドパケットの org_name 属性です。
- **MYSQL.RES.FIELD(<i>).ORIGINAL_TABLE.** カラムの元のテーブル ID (AS 句がある場合) を返します。これは、フィールドパケットの org_table 属性です。
- **MYSQL.RES.FIELD(<i>).TABLE.** カラムのテーブル ID を返します（AS 句がある場合）。フィールドパケットのテーブル属性です。
- **MYSQL.RES.FIELDS_COUNT.** 応答内のフィールドパケット数（OK パケットの field_count 属性）を返します。
- **MYSQL.RES.OK** データベースサーバから送信された OK パケットを識別します。
- **MYSQL.RES.OK.AFFECTED_ROWS.** INSERT、UPDATE、または DELETE クエリによって影響を受けるローの数を返します。これは、OK パケットの affected_rows 属性の値です。
- **MYSQL.RES.OK.INSERT_ID.** OK パケットの unique_id 属性を識別します。自動インクリメント ID が現在の MySQL ステートメントまたはクエリによって生成されない場合、unique_id の値と、式によって返される値は 0 になります。
- **MYSQL.RES.OK.MESSAGE.** OK パケットのメッセージプロパティを返します。
- **MYSQL.RES.OK.STATUS.** OK パケットの server_status 属性内のビット文字列を識別します。クライアントは、サーバーの状態を使用して、現在のコマンドが実行中のトランザクションの一部であるかどうかを確認できます。server_status ビット文字列のビットは、次のフィールドに対応します (指定された順序で)。
  - IN TRANSACTION
  - AUTO_COMMIT

  - MORE RESULTS
  - MULTI QUERY
  - BAD INDEX USED
  - NO INDEX USED
  - CURSOR EXISTS
  - LAST ROW SEEN
  - DATABASE DROPPED
  - NO BACKSLASH ESCAPES

- **MYSQL.RES.OK.WARNING_COUNT.** OK パケットの warning_count 属性を返します。
- **MYSQL.RES.ROW(<i>).** i^{番目に対応するパケットを識別します。} データベースサーバーの応答内の個々の行。

  パラメーター:

  i-行番号

- **MYSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>).** j^{番目のかどうかをチェックします。} I^{番目の列} テーブルの行が NULL です。C の規則に従って、インデックス i と j の両方が 0 から始まります。したがって、行 i と列 j は実際には (i+1)^{番目のものです} 行と (j+1)^{番目の行} 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.ROW(<i>).IS_NULL_ELEM(j).** j^{番目のかどうかをチェックします。} i^{番目の列} テーブルの行が NULL です。C の規則に従って、インデックス i と j の両方が 0 から始まります。したがって、行 i と列 j は実際には (i+1)^{番目のものです} 行と (j+1)^{番目の行} 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.ROW(<i>).NUM_ELEM(<j>).** j^{番目の整数値を返します。} i^{番目の列} テーブルの行。C の規則に従って、インデックス i と j の両方が 0 から始まります。したがって、行 i と列 j は実際には (i+1)^{番目のものです} 行と (j+1)^{番目の行} 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.ROW(<i>).TEXT_ELEM(j).** j^{番目の文字列を返します。} i^{番目の列} テーブルの行。C の規則に従って、インデックス i と j の両方が 0 から始まります。したがって、行 i と列 j は実際には (i+1)^{番目のものです} 行と (j+1)^{番目の行} 列、それぞれ。

  パラメーター:

  i-行番号

  j-列番号

- **MYSQL.RES.TYPE.** 応答型の列挙定数を返します。この値には、エラー、OK、および結果セットを指定できます。<m> <m> 比較の結果を示すブール値を返す EQ () および NE () 演算子は、このプレフィックスとともに使用されます。

Microsoft SQL サーバー接続を評価するための式

次の式は、Microsoft SQL Server データベースサーバーに関連付けられたトラフィックを評価します。ポリシーでは、要求ベースの式 (MSSQL.CLIENT および MSSQL.REQ で始まる式) を使用して、コンテンツスイッチング仮想サーバーのバインドポイントで要求切り替えの決定を行い、応答ベースの式 (MSSQL.RES で始まる式) を使用して、ユーザーに対するサーバーの応答を評価できます。構成されたヘルスモニタ。

式	説明
MSSQL.CLIENT.CAPABILITIES	LOGIN7 認証パケットのオプションフラグ 1、オプションフラグ 2、オプションフラグ 3、およびタイプフラグフィールドを 4 バイトの整数としてその順序で返します。各フィールドの長さは 1 バイトで、一連のクライアント機能を指定します。
MSSQL.CLIENT.DATABASE	クライアントデータベースの名前を返します。返される値は text 型です。
MSSQL.CLIENT.USER	クライアントが認証したユーザー名を返します。返される値は text 型です。
MSSQL.REQ.COMMAND	Microsoft SQL Server データベースサーバーに送信される要求内のコマンドの種類を識別する列挙定数を返します。返される値は text 型です。列挙定数の値の例としては、クエリ、応答、RPC、および注意があります。<m> <m> 比較の結果を示すブール値を返す EQ () および NE () 演算子は、この式で使用されます。
MSSQL.REQ.QUERY.COMMAND	SQL クエリの最初のキーワードを返します。返される値は text 型です。
MSSQL.REQ.QUERY.SIZE	リクエスト内の SQL クエリのサイズを返します。返される値は数値です。

式	説明
MSSQL.REQ.QUERY.TEXT	SQL クエリ全体を文字列として返します。返される値は text 型です。
MSSQL.REQ.QUERY.TEXT(<n>)	SQL クエリの最初の n バイトを返します。返される値は text 型です。パラメータ:n-バイト数
MSSQL.REQ.RPC.NAME	リモートプロシージャコール (RPC) 要求で呼び出されているプロシージャの名前を返します。名前は文字列として返されます。
MSSQL.REQ.RPC.IS_PROCID	リモートプロシージャコール (RPC) 要求にプロシージャ ID または RPC 名が含まれているかどうかを示すブール型 (Boolean) の値を返します。TRUE の戻り値は要求にプロシージャ ID が含まれていることを示し、FALSE の戻り値は要求に RPC 名が含まれていることを示します。
MSSQL.REQ.RPC.PROCID	リモートプロシージャコール (RPC) 要求のプロシージャ ID を整数で返します。
MSSQL.REQ.RPC.BODY 注**:** 10.1 より前のリリースでは使用できません。	SQL リクエストの本文を、カンマで区切られた「a=b」句で表されるパラメータ形式の文字列として返します。ここで、「a」は RPC パラメータ名、「b」はその値です。
MSSQL.REQ.RPC.BODY 注**:** 10.1 より前のリリースでは使用できません。	SQL リクエストの本文の一部を、カンマで区切られた「a=b」句で表されるパラメータ形式の文字列として返します。ここで、「a」は RPC パラメータ名、「b」はその値です。パラメータは、リクエストの最初の「n」バイトからのみ返され、SQL ヘッダーをスキップします。完全な名前と値のペアだけが返されます。
MSSQL.RES.ATLEAST_ROWS_COUNT(i)	応答に少なくとも i 行数があるかどうかをチェックします。返される値は、ブール値 TRUE または FALSE です。パラメータ:i-行数
MSSQL.RES.DONE.ROWCOUNT	INSERT、UPDATE、または DELETE クエリによって影響を受けるローの数を返します。返される値は、符号なし long 型です。
MSSQL.RES.DONE.STATUS	Microsoft SQL Server データベースサーバーによって送信された DONE トークンからステータスフィールドを返します。返される値は数値です。

式	説明
MSSQL.RES.ERROR.MESSAGE	Microsoft SQL Server データベースサーバーによって送信された ERROR トークンからエラーメッセージを返します。これは、ERROR トークンの MsgText フィールドの値です。返される値は text 型です。
MSSQL.RES.ERROR.NUM	Microsoft SQL Server データベースサーバーによって送信された ERROR トークンからエラー番号を返します。これは、ERROR トークンの Number フィールドの値です。返される値は数値です。
MSSQL.RES.ERROR.STATE	Microsoft SQL Server データベースサーバーによって送信された ERROR トークンからエラー状態を返します。これは、ERROR トークンの State フィールドの値です。返される値は数値です。
MSSQL.RES.FIELD(<i>).DATATYPE	サーバーレスポンスの i 番目のフィールドのデータ型を返します。<m> <m> 比較の結果を示すブール値を返す EQ () 関数と NE () 関数は、このプレフィックスとともに使用されます。たとえば、次の式は、DATATYPE 関数が応答の 3 番目のフィールドの datetime の値を返す場合に、ブール値 TRUE を返します。MSSQL.RES.FIELD (<2>) .DATATYPE.EQ (日時) パラメーター:i-行番号
MSSQL.RES.FIELD(<i>).LENGTH	サーバ応答の i 番目のフィールドの可能な最大長を返します。返される値は数値です。パラメータ:i-行番号
MSSQL.RES.FIELD(<i>).NAME	サーバーレスポンスの i 番目のフィールドの名前を返します。返される値は text 型です。パラメータ:i-行番号
MSSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>)	テーブルの i 番目の行の j 番目の列から、double 型の値を返します。値が倍精度値でない場合は、UNDEF 条件が発生します。C の規則に従って、インデックス i と j の両方が 0（ゼロ）から始まります。したがって、行 i と列 j は、実際には (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号

式	説明
MSSQL.RES.ROW(<i>).NUM_ELEM(j)	テーブルの i 番目の行の j 番目の列から整数値を返します。値が整数値でない場合は、UNDEF 条件が発生します。C の規則に従って、インデックス i と j の両方が 0（ゼロ）から始まります。したがって、行 i と列 j は、実際には (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号
MSSQL.RES.ROW(<i>).IS_NULL_ELEM(j)	テーブルの i 番目の行の j 番目の列が NULL かどうかをチェックし、結果を示すブール値 TRUE または FALSE を返します。C の規則に従って、インデックス i と j の両方が 0（ゼロ）から始まります。したがって、行 i と列 j は、実際には (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号
MSSQL.RES.ROW(<i>).TEXT_ELEM(j)	テーブルの i 番目の行の j 番目の列からテキスト文字列を返します。C の規則に従って、インデックス i と j の両方が 0（ゼロ）から始まります。したがって、行 i と列 j は、実際には (i + 1) 番目の行と (j + 1) 番目の列です。パラメータ:i-行番号 j-列番号
MSSQL.RES.TYPE	応答の種類を識別する列挙定数を返します。考えられる戻り値は、エラー、OK、および結果セットです。<m> <m> 比較の結果を示すブール値を返す EQ () および NE () 演算子は、この式で使用されます。

データの型キャスト

October 7, 2021

リクエストとレスポンスから、あるタイプ（テキストや整数など）のデータを抽出し、別のタイプのデータに変換できます。たとえば、文字列を抽出し、文字列を時刻形式に変換できます。また、HTTP リクエスト本体から文字列を抽出して HTTP ヘッダーのように扱うことも、あるタイプのリクエストヘッダーから値を抽出して別のタイプのレスポンスヘッダーに挿入することもできます。

データの型キャスト後、新しいデータ型に適した任意の操作を適用できます。たとえば、HTTP ヘッダーにテキストをタイプキャストする場合、HTTP ヘッダーに適用可能な任意の操作を戻り値に適用できます。

型キャストデータの詳細については、「型キャスト操作」の PDF を参照してください。

正規表現

October 7, 2021

CONTAINS (”<string>”) または EQ (”<string>”) 演算子で実行する操作よりも複雑な文字列マッチング操作を実行する場合は、正規表現を使用します。Citrix® Citrix ADC® アプライアンスのポリシーインフラストラクチャには、テキストマッチングの引数として正規表現を渡すことができる演算子が含まれています。正規表現で動作する演算子の名前には、文字列 REGEX が含まれます。引数として渡す正規表現は、「”<http://www.pcre.org/pcre.txt>.”」で説明されている正規表現の構文に準拠している必要があります。正規表現の詳細については、「”<http://www.regular-expressions.info/quickstart.html>”」および「”<http://www.silverstones.com/thebat/Regex.html>.”」を参照してください。

正規表現で動作する演算子のターゲットテキストは、テキストまたは HTTP ヘッダーの値のいずれかです。次に、正規表現演算子を使用してテキストを操作するデフォルトの構文式の形式を示します。

```
<text>.<regex_operator>(re<delimiter><regex_pattern><delimiter>)
```

この<text>文字列は、パケット内のテキスト文字列を識別するデフォルトの構文式プレフィックス (HTTP.REQ.URL など) を表します。文字列は、<regex_operator>正規表現演算子を表します。正規表現は常に文字列 re で始まります。正規表現を表す文字列<regex_pattern>は、<delimiter>で表される一致区切り文字のペアで囲みます。

次の式例は、HTTP パケット内の URL に文字列*.jpeg （* はワイルドカード）が含まれているかどうかを調べ、結果を示すブール値 TRUE または FALSE を返します。正規表現は、区切り文字として機能するスラッシュマーク (/) のペアで囲まれています。

```
http.req.url.regex_match(re/.<asterisk>\.jpeg/)
```

正規表現演算子を組み合わせて、検索の範囲を定義または絞り込むことができます。たとえば、<text>.AFTER_REGEX(reregex_pattern1).BEFORE_REGEX(reregex_pattern2)は、文字列マッチングのターゲットが、パターン regex_pattern1 と regex_pattern2 の間のテキストであることを指定します。正規表現演算子によって定義されたスコープでは、テキスト演算子を使用できます。たとえば、CONTAINS (”<string>”) 演算子を使用して、定義されたスコープに文字列 abc が含まれているかどうかを確認できます。

```
<text>.AFTER_REGEX(re/regex_pattern1).BEFORE_REGEX(re/regex_pattern2/).
CONTAINS(”abc”)
```

> 注
>
> 正規表現を評価するプロセスは、単純な文字列引数で動作する CONTAINS (”<string>”) や EQ (”<string>”) などの演算子よりも本質的に時間がかかります。正規表現は、要件が他の演算子の範囲を超えている場合にのみ使用してください。

正規表現の基本特性

October 7, 2021

Citrix ADC アプライアンスで定義されている正規表現の特性は次のとおりです。

- 正規表現は常に文字列「re」で始まり、その後に、使用する正規表現を囲む区切り文字（区切り文字と呼ばれます）が続きます。

たとえば、re# # <regex_pattern> # は区切り文字として番号記号 (#) を使用します。

- 正規表現は 1499 文字を超えることはできません。
- 数字のマッチングは、文字列 d (バックスラッシュの後に d) を使用して行うことができます。
- 空白は、s (バックスラッシュの後に s) を使用して表すことができます。
- 正規表現には空白を含めることができます。

Citrix ADC の構文と PCRE の構文の違いを次に示します。

- Citrix ADC では、正規表現での後方参照は許可されません。
- 再帰的な正規表現を使用しないでください。
- ドットメタ文字は、改行文字にも一致します。
- Unicode はサポートされていません。
- 操作 SET_TEXT_MODE (IGNORECASE) は、(? i）正規表現の内部オプション。

正規表現の操作

October 7, 2021

次の表に、正規表現で動作する演算子を示します。特定のデフォルトの構文式で正規表現演算子によって実行される操作は、式プレフィックスがテキストまたは HTTP ヘッダーを識別するかどうかによって異なります。ヘッダーを評価する操作は、指定されたヘッダータイプのすべてのインスタンスに対するテキストベースの操作よりも優先されます。演算子を使用する場合は、<text> テキストを識別するために構成するデフォルトの構文式プレフィクスに置き換えます。

正規表現演算	説明
<text>.BEFORE_REGEX(<regular expression>)	<regular expression> 引数に一致する文字列の前のテキストを選択します。正規表現がターゲットのどのデータとも一致しない場合、式は長さ 0 のテキストオブジェクトを返します。次の式は、「テキスト/プレーン」から文字列「テキスト」を選択します。 http.res.header（「コンテンツタイプ」）.before_regex（RE #/#）
<text>.AFTER_REGEX(<regular expression>)	<regular expression> 引数に一致する文字列の後のテキストを選択します。正規表現がターゲットのどのテキストとも一致しない場合、式は長さ 0 のテキストオブジェクトを返します。次の式は、「myExample」から「例」を抽出します。http.req. ヘッダー（「etag」）.after_regex（再/私の/）
<text>.REGEX_SELECT(<regular expression>)	<regular expression> 引数に一致する文字列を選択します。正規表現がターゲットと一致しない場合、長さ 0 のテキストオブジェクトが返されます。次の例では、Via ヘッダーから文字列「NS-CACHE-9.0:90」を抽出します。 http.req.header(“via”).regex_select(re!NS-CACHE-\d.\d:\s*\d{1,3}!)

正規表現演算	説明
<text>.REGEX_MATCH(<regular expression>)	ターゲットが <regular expression> 1499 文字までの引数に一致する場合に TRUE を返します。正規表現は、次の形式である必要があります。re <delimiter> 正規表現 < delimiter> 両方の区切り文字は同じである必要があります。さらに、正規表現は Perl 互換 (PCRE) の正規表現ライブラリ構文に準拠している必要があります。詳しくは、http://www.pcre.org/pcre.txtを参照してください。特に、pcrepattern のマニュアルページを参照してください。ただし、次の点に注意してください。後方参照は許可されていません。再帰的な正規表現は推奨されません。ドットメタ文字は、改行文字にも一致します。Unicode 文字セットはサポートされていません。SET_TEXT_MODE(IGNORECASE) が正規表現で指定された (?i) 内部オプションを上書きします。以下はその例である。http.req.hostname.regex_match (re/[[: alpha:]] + (abc) {2,3}/) と http.req.url.set_text_mode (urlencoded) .regex_match (re# (a*b*+c) #) 次の例は ab と a に一致する:http.req.url.regex_match (re/a (?i) b/) 次の例では、ab、aB、Ab、Ab、AB とマッチします。http.req.url.set_text_mode (ignoreCase) .regex_match (re/ab/) 次の例では、大文字と小文字を区別しない複数行の一致を実行します。http.req.body.regex_match (re/ (?ixm) (^ab (.*) cd$)/)

従来のポリシーと式の構成

October 7, 2021

Citrix ADC の一部の機能では、従来のポリシーと従来の表現が使用されます。デフォルトの構文ポリシーと同様に、従来のポリシーは、グローバルまたは仮想サーバに固有のものになります。ただし、クラシックポリシーの設定方法とバインドポイントは、ある程度はデフォルトの構文ポリシーの設定方法と異なります。デフォルトの構文式と同様に、名前付き式を設定し、複数のクラシックポリシーで名前付き式を使用できます。

次の表は、従来のポリシーを使用して構成できる Citrix ADC 機能をまとめたものです。

ここをクリックしてテーブルを表示します 。

クラシックポリシーを構成する

October 7, 2021

構成ユーティリティまたはコマンドラインインターフェイスを使用して、従来のポリシーと従来の式を構成できます。ポリシールールは 1,499 文字を超えることはできません。ポリシールールを構成するときに、名前付きクラシック式を使用できます。名前付きエクスプレッションの詳細については、名前付きクラシックエクスプレッションの作成を参照してください。ポリシーを構成したら、グローバルまたは仮想サーバにバインドします。

Citrix ADC の各機能のポリシー構成方法には若干の違いがあることに注意してください。

注：構文 SYS.EVAL_CLASSIC_EXPR（クラシック式）を使用し、引数としてクラシック式を指定すると、クラシック式をデフォルトの構文式に埋め込むことができます。

CLI を使用したクラシックポリシーの作成

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
-  add cmp policy <name> -rule <expression> -action <action>

-  show cmp policy [<policyName>]
<!--NeedCopy-->
```

例

次のコマンドは、まず圧縮アクションを作成してから、そのアクションを適用する圧縮ポリシーを作成します。

```
> add cmp action cmp-act-compress compress
 Done
> show cmp action cmp-act-compress
1)      Name: cmp-act-compress  Compression Type: compress
 Done
> add cmp pol cmp-pol-compress -rule ExpCheckIp -resAction cmp-act-
   compress
 Done
> show cmp pol cmp-pol-compress
1)      Name: cmp-pol-compress  Rule: ExpCheckIp
```



```
10          Response action: cmp-act-compress       Hits: 0
11   Done
12  >
13  <!--NeedCopy-->
```

GUI を使用したクラシック式を使用したポリシーの作成

1. ナビゲーションペインで、ポリシーを設定する機能を展開し、機能に応じて次の操作を行います。
   - [コンテンツの切り替え]、[キャッシュリダイレクト]、および [アプリケーションファイアウォール] で、[ポリシー] をクリックします。
   - [SSL] の場合は、[ポリシー] をクリックし、詳細ウィンドウで [ポリシー] タブをクリックします。
   - [システム認証] で [認証] をクリックし、詳細ウィンドウで [ポリシー] タブをクリックします。
   - [フィルタ]、[SureConnect]、および [優先キューイング] で、[保護機能] を展開し、目的の機能を選択し、詳細ウィンドウで [ポリシー] タブをクリックします。
   - Citrix Gateway の場合は、[Citrix Gateway]、[ポリシー] の順に展開し、目的の機能を選択し、詳細ペインで [ポリシー] タブをクリックします。
2. ほとんどの機能では、[追加] ボタンをクリックします。
3. [<feature name> ポリシーの 作成] ダイアログボックスの [名前 *] テキストボックスに、ポリシーの名前を入力します。

   注**:** ポリシー名の先頭には、文字またはアンダースコアを使用する必要があります。ポリシー名は、文字、数字、ハイフン（-）、ピリオド（.）、シャープ記号（#）、スペース（）、アンダースコア（_）など、1～31 文字で構成できます。
4. ほとんどの機能では、アクションまたはプロファイルを関連付けます。たとえば、アクションを選択する必要がある場合や、Citrix Gateway またはアプリケーションファイアウォールポリシーの場合は、ポリシーに関連付けるプロファイルを選択します。プロファイルは、分析対象のデータがポリシー・ルールに一致したときに適用される一連のアクションとして機能する構成オプションのセットです。
5. このポリシーで照合するデータの種類を説明する式を作成します。

   作成するポリシーのタイプに応じて、定義済みの式を選択することも、新しい式を作成することもできます。

   名前付き式は、ポリシールールで名前で参照できる定義済みの式です。
6. [作成**]** をクリックして、新しいポリシーを作成します。
7. [**Close**] をクリックして、作成中のポリシーのタイプの [Policies] 画面に戻ります。

クラシックエクスプレッションを構成する

October 7, 2021

クラシックエクスプレッションは、次のエクスプレッション要素で構成され、階層順にリストされます。

- **[フロータイプ]:** 接続が着信か発信かを指定します。フロータイプは、着信接続の場合は REQ、発信接続の場合は RES です。
- プロトコル。プロトコルを指定します。選択できるプロトコルは、HTTP、SSL、TCP、および IP です。
- 修飾子。選択したプロトコルに依存するプロトコル属性。
- 演算子。接続データに対して実行するテストのタイプ。オペレータの選択は、テストする接続情報によって異なります。テストする接続情報がテキストの場合は、テキスト演算子を使用します。数値の場合は、標準の数値演算子を使用します。
- **Value.** 接続データ要素（フロータイプ、プロトコル、修飾子によって定義される）がテストされる文字列または数値。値は、リテラルまたは式のいずれかです。リテラルまたは式は、接続データ要素のデータ型と一致する必要があります。

ポリシーでは、ブール演算子と比較演算子を使用して、従来の式を組み合わせて、より複雑な式を作成できます。

式要素は、左から右に解析されます。左端の要素は REQ または RES のいずれかであり、それぞれ要求または応答を指定します。連続する用語は、特定の接続タイプとその接続タイプの特定の属性を定義します。各用語は、ピリオドによって先行または後続の用語から分離されます。引数は括弧内に表示され、渡された式要素の後に続きます。

次の古典的な式フラグメントは、着信接続のクライアント送信元 IP を返します。

```
REQ.IP.SOURCEIP
```

この例では、リクエスト内の IP アドレスを識別します。式要素 SOURCEIP は、送信元 IP アドレスを指定します。この式フラグメントは、それ自体では役に立たないかもしれません。追加の式要素である演算子を使用して、戻り値が特定の条件を満たしているかどうかを判断できます。次の式は、クライアント IP がサブネット 200.0.0.0/8 にあるかどうかをテストし、ブール値 TRUE または FALSE を返します。

```
REQ.IP.SOURCEIP == 200.0.0.0 -netmask 255.0.0.0
```

CLI を使用したクラシックポリシー式の作成

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
1 -  set appfw policy <name> -rule <expression> -action <action>
2
3 -  show appfw policy <name>
4 <!--NeedCopy-->
```

例

```
1 > set appfw policy GenericApplicationSSL_ 'HTTP.REQ.METHOD.EQ(”get”)'
     APPFW_DROP
```



```
 Done
> show appfw policy GenericApplicationSSL_
        Name: GenericApplicationSSL_    Rule: HTTP.REQ.METHOD.EQ("get")
        Profile: APPFW_DROP     Hits: 0
        Undef Hits: 0
        Policy is bound to following entities
        1) REQ VSERVER app_u_GenericApplicationSSLPortalPages
           PRIORITY : 100
 Done
<!--NeedCopy-->
```

GUI を使用してクラシックポリシーの式を追加する

この手順では、[式の追加] ダイアログボックスについて説明します。ポリシーを設定する機能によっては、このダイアログボックスに到達するルートが異なる場合があります。

1. 「GUI を使用して従来の式を使用してポリシーを作成するには」の手順 1～4 を実行します。
2. [式の追加] ダイアログボックスの [式の種類] で、作成する式の種類をクリックします。
3. [フロータイプ] の下向き矢印をクリックし、フロータイプを選択します。

フロータイプは、通常 REQ または RES です。REQ オプションは、ポリシーをすべての着信接続または要求に適用するように指定します。RES オプションは、すべての発信接続または応答にポリシーを適用します。

アプリケーションファイアウォールポリシーの場合は、式タイプを [一般式] に設定し、フロータイプを [REQ] に設定しておきます。アプリケーションファイアウォールは、各要求と応答を 1 つのペアになったエンティティとして扱います。したがって、すべてのアプリケーションファイアウォールポリシーは REQ で始まります。

1. [Protocol] の下向き矢印をクリックし、ポリシー式に使用するプロトコルを選択します。選択肢は次のとおりです:
   - HTTP Web サーバーに送信される HTTP 要求を評価します。従来の式の場合、HTTP には HTTPS リクエストが含まれます。
   - SSL。現在の接続に関連付けられている SSL データを評価します。
   - TCP。現在の接続に関連付けられている TCP データを評価します。
   - 知的財産権です。現在の接続に関連付けられている IP アドレスを評価します。
2. [Qualifier] で下向き矢印をクリックし、ポリシーの修飾子を選択します。
   修飾子は、評価されるデータのタイプを定義します。表示される修飾子の一覧は、手順 4 で選択したプロトコルによって異なります。
   HTTP プロトコルでは、次の選択肢が表示されます。
   - METHOD. 特定の HTTP メソッドを使用する HTTP リクエストをフィルタリングします。
   - URL。特定の Web ページの HTTP 要求をフィルタリングします。
   - URLQUERY. 特定のクエリ文字列を含む HTTP リクエストをフィルタリングします。
   - VERSION. 指定された HTTP プロトコルのバージョンに基づいて HTTP 要求をフィルタリングします。

- HEADER. 特定の HTTP ヘッダーに基づいてフィルタリングします。
- URLLEN. URL の長さに基づいてフィルタリングします。
- URLQUERY. URL のクエリ部分に基づいてフィルタリングします。
- URLQUERYLEN. URL のクエリ部分の長さのみに基づいてフィルタリングします。

3. [Operator] の下向き矢印をクリックし、ポリシー式の演算子を選択します。一般的な演算子は次のとおりです。

演算子	説明
==	指定された値に正確に一致するか、指定された値と完全に等しいです。
!=	指定された値と一致しません。
>	指定された値よりも大きい。
<	指定された値より小さい。
>=	指定した値以上である。
<=	指定された値以下である。
CONTAINS	指定された値を格納します。
CONTENTS	指定されたヘッダー、URL、または URL クエリの内容を返します。
EXISTS	指定されたヘッダーまたはクエリが存在します。
NOTCONTAINS	指定された値を含まない。
NOTEXISTS	指定されたヘッダーまたはクエリは存在しません。

1. 「値」(Value) テキストボックスが表示されたら、必要に応じて文字列または数値を入力します。たとえば、フロータイプとして「REQ」、プロトコルとして「HTTP」、修飾子として「HEADER」を選択し、「値」フィールドにヘッダー文字列の値を入力し、「ヘッダー名」テキストボックスに文字列と一致するヘッダータイプを入力します。
2. [**OK**] をクリックします。
3. 複合エクスプレッションを作成するには、[追加] をクリックします。実行する複合のタイプは、[ポリシーの作成] ダイアログボックスで次の選択内容によって異なります。
   - 任意の式に一致します。式は論理 OR 関係にあります。
   - **[**すべての式に一致する**]**。式は論理 AND 関係にあります。
   - 表形式の式。[AND]、[OR]、および [かっこ] ボタンをクリックして、評価を制御します。
   - 高度なフリーフォーム。式コンポーネントを [式] フィールドに直接入力し、[AND]、[OR]、[かっこ] ボタンをクリックして評価を制御します。

クラシックポリシーをバインドする

October 7, 2021

ポリシータイプに応じて、クラシックポリシーをグローバルにバインドすることも、仮想サーバにバインドすることもできます。ポリシーバインドポイントについては、「クラシックポリシーを使用する機能におけるポリシーのポリシータイプとバインドポイント」の表を参照してください。

注**:** クラシックポリシーは、複数のバインドポイントにバインドできます。

CLI を使用してクラシックポリシーをグローバルにバインドする

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
-  bind cmp global <policyName> [-priority <positive_integer>]

-  show cmp global
<!--NeedCopy-->
```

例

```
> bind cmp global cmp-pol-compress -priority 2
 Done
> show cmp global
1)      Policy Name: cmp-pol-compress   Priority: 2
2)      Policy Name: ns_nocmp_xml_ie    Priority: 8700
3)      Policy Name: ns_nocmp_mozilla_47        Priority: 8800
4)      Policy Name: ns_cmp_mscss       Priority: 8900
5)      Policy Name: ns_cmp_msapp       Priority: 9000
6)      Policy Name: ns_cmp_content_type        Priority: 10000
 Done
>
<!--NeedCopy-->
```

CLI を使用してクラシックポリシーを仮想サーバーにバインドする

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。


```
-  bind lb vserver <name> [<targetVserver>] [-policyName <string> [-
   priority <positive_integer>]

-  show lb vserver<name>
<!--NeedCopy-->
```

例

```
> bind lb vserver lbtemp -policyName cmp-pol-compress -priority 1
 Done
> show lb vserver lbtemp
        lbtemp (10.102.29.101:80) - HTTP        Type: ADDRESS
        State: UP
        Last state change was at Tue Oct 27 06:40:38 2009 (+557 ms)
        Time since last state change: 0 days, 02:00:40.330
        Effective State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: Bound service's state
           changed to UP
        Group: vserver-grp
        Mode: IP
        Persistence: COOKIEINSERT (version 0)   Persistence Backup:
           SOURCEIP   Persistence Mask: 255.255.255.255
        Persistence Timeout: 2 min      Backup Persistence Timeout: 2
           min
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
1) http-one (10.102.29.252: 80) - HTTP State: UP        Weight: 1
        Persistence Cookie Value : NSC_wtfswfs-hsq=
           ffffffff096e03ed45525d5f4f58455e445a4a423660
1)      Policy : cmp-pol-compress Priority:1
 Done
>
<!--NeedCopy-->
```


GUI を使用してクラシックポリシーをグローバルにバインドする

注記: この手順では、「グローバルバインディング」(Global Bindings) ダイアログボックスについて説明します。ポリシーをグローバルにバインドする機能によっては、このダイアログボックスに到達するルートが異なる場合があります。

1. ナビゲーションウィンドウで、クラシックポリシーをグローバルにバインドする機能を展開し、グローバルにバインドするポリシーを見つけます。

   注：コンテンツスイッチング、キャッシュリダイレクト、SureConnect、優先キューイング、または Citrix Gateway 認証のポリシーをグローバルにバインドすることはできません。

2. 詳細ペインで、[グローバルバインディング] をクリックします。
3. [グローバルに<feature name> ポリシーをバインド/バインド解除] ダイアログボックスで、[ポリシーの挿入] をクリックします。
4. [ポリシー名] 列で、グローバルにバインドする既存のポリシーの名前をクリックするか、[新しいポリシー] をクリックして [ポリシーの作成] <Feature name> ダイアログボックスを開きます。
5. ポリシーを選択するか、新しいポリシーを作成したら、[Priority] 列に優先度の値を入力します。

   数値が小さいほど、他のポリシーに対してこのポリシーが適用される時間が短くなります。たとえば、プライオリティ 10 が割り当てられたポリシーは、プライオリティ 100 のポリシーの前に適用されます。異なるポリシーに同じプライオリティを使用できます。クラシックポリシーを使用するすべての機能では、接続が一致した最初のポリシーだけが実装されるため、ポリシーの優先順位は意図した結果を得るために重要です。

   ベストプラクティスとして、各ポリシー間に 50（または 100）の間隔で優先度を設定することで、ポリシーを追加する余地を残します。

6. [**OK**] をクリックします。

GUI を使用して従来のポリシーを仮想サーバーにバインドする

1. ナビゲーションウィンドウで、クラシックポリシーをバインドする仮想サーバーを含む機能を展開します (たとえば、クラシックポリシーをコンテンツスイッチ仮想サーバーにバインドする場合は、[トラフィック管理] > [コンテンツの切り替え] の順に展開します)。次に、[仮想サーバー] をクリックします。
2. 詳細ウィンドウで、仮想サーバーを選択し、[開く] をクリックします。
3. [<Feature> 仮想サーバーの構成] ダイアログボックスの [ポリシー] タブで、目的のタイプポリシーの機能アイコンをクリックし、[ポリシーの挿入] をクリックします。
4. [Policy Name] 列で、仮想サーバにバインドする既存のポリシーの名前をクリックするか、[A] をクリックして [Create <feature name> Policy] ダイアログ・ボックスを開きます。
5. ポリシーを選択するか、新しいポリシーを作成したら、[Priority] 列で優先度を設定します。

   ポリシーをコンテンツスイッチ仮想サーバーにバインドする場合は、[ターゲット] 列で、ポリシーと一致するトラフィックの送信先となる負荷分散仮想サーバーを選択します。

6. [**OK**] をクリックします。

クラシックポリシーを表示する

October 7, 2021

構成ユーティリティまたはコマンドラインを使用して、クラシックポリシーを表示できます。ポリシーの名前、式、バインディングなどの詳細を表示できます。

CLI を使用したクラシックポリシーとそのバインド情報の表示

コマンドプロンプトで次のコマンドを入力して、クラシックポリシーとそのバインド情報を表示します。

```
show <featureName> policy [policyName]
```

例

```
1 > show appfw policy GenericApplicationSSL_
2         Name: GenericApplicationSSL_    Rule: ns_only_get_adv
3         Profile: GenericApplicationSSL_Prof1    Hits: 0
4         Undef Hits: 0
5         Policy is bound to following entities
6         1) REQ VSERVER app_u_GenericApplicationSSLPortalPages
             PRIORITY : 100
7  Done
8 <!--NeedCopy-->
```

注意**:** ポリシー名を省略すると、すべてのポリシーがバインドの詳細なしでリストされます。

GUI を使用したクラシックポリシーおよびポリシーバインディングの表示

1. ナビゲーションウィンドウで、ポリシーを表示する機能を展開します (たとえば、アプリケーションファイアウォールポリシーを表示する場合は、[アプリケーションファイアウォール] を展開します)。次に、[ポリシー] をクリックします。
2. 詳細ウィンドウで、次の 1 つまたは複数の操作を行います。
   - 特定のポリシーの詳細を表示するには、ポリシーをクリックします。詳細が、設定ペインの [Details] 領域に表示されます。
   - 特定のポリシーのバインドを表示するには、ポリシーをクリックし、[バインドの表示] をクリックします。

- グローバルバインドを表示するには、ポリシーをクリックし、[グローバルバインディング] をクリックします。コンテンツスイッチング、キャッシュリダイレクト、SureConnect、優先キューイング、または Citrix Gateway 承認ポリシーをグローバルにバインドすることはできません。

名前の付いたクラシックエクスプレッションを作成する

October 7, 2021

名前付きクラシックエクスプレッションは、割り当てられた名前を使用して参照できるクラシックエクスプレッションです。多くの場合、大規模または複雑で、より大きな複合式の一部を形成する古典的な式を構成する必要があります。また、頻繁に使用する必要があるクラシック式を、複数の複合式またはクラシックポリシーで構成することもできます。これらのシナリオでは、必要なクラシック式を作成し、任意の名前で保存し、複合式またはポリシーからその式を名前で参照できます。これにより、構成時間が節約され、複雑な複合式の可読性が向上します。さらに、名前付きクラシックエクスプレッションに対する変更は 1 回だけ行う必要があります。

一部の名前付き式は組み込みで、これらのサブセットは読み取り専用です。組み込みの名前付き式は、[全般]、[アンチウイルス]、[パーソナルファイアウォール]、および [インターネットセキュリティ] の 4 つのカテゴリに分類されます。一般的な名前付き式には、さまざまな用途があります。たとえば、[一般] カテゴリから、ns_true および ns_false という式を使用して、すべてのトラフィックに対して返される値を TRUE または FALSE に指定できます。また、特定の種類のデータ（HTM、DOC、GIF ファイルなど）を識別したり、キャッシュヘッダーが存在するかどうかを確認したり、クライアントと Citrix ADC 間のパケットのラウンドトリップ時間が長い（80 ミリ秒以上）かどうかを判断することもできます。

アンチウイルス、パーソナルファイアウォール、およびインターネットセキュリティの名前付き式は、クライアントで特定のプログラムとバージョンの存在をテストし、主に Citrix Gateway ポリシーで使用されます。

メモ**:** 組み込みの名前付き式を変更または削除することはできません。

CLI を使用して名前付きクラシック式を作成する

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
1 -  add expression <name> <value> [-comment <string>] [-
     clientSecurityMessage <string>]
2 -  show expression [<name> | -type CLASSIC
3 <!--NeedCopy-->
```

例


```
> add expression classic_ne ”REQ.HTTP.URL CONTAINS www.example1.com” -
   comment ”Checking the URL for www.example1.com”
 Done
> show expression classic_ne
1)      Name: classic_ne  Expr: REQ.HTTP.URL CONTAINS www.example1.com
    Hits: 0 Type : CLASSIC
        Comment: ”Checking the URL for www.example1.com”
 Done
>
<!--NeedCopy-->
```

GUI を使用して名前付きクラシック式を作成する

1. ナビゲーションウィンドウで、[AppExpert] を展開し、[式] を展開し、[クラシック式] をクリックします。
2. 詳細ペインで、[Add] をクリックします。

   メモ: 式リストの組み込み式の一部は読み取り専用です。
3. [ポリシー式の作成] ダイアログボックスで、次のパラメータの値を指定します。

   - 式名 *—name
   - クライアントセキュリティメッセージ: clientSecurityMessage
   - コメント: comment

   * 必須パラメータ
4. 式を作成するには、次のいずれかの操作を行います。

   - このエクスプレッションへの入力は、[名前付きエクスプレッション] ドロップダウンリストから選択できます。
   - GUI を使用したクラシックポリシーの式の追加で説明されているように、新しい式を作成できます。
5. 完了したら、[閉じる] をクリックします。新しいエクスプレッションが作成されたことを確認するには、[クラシックエクスプレッション] リストの一番下までスクロールして表示します。

式参照-高度なポリシー式

October 7, 2021

> 警告
>
> Q および S プレフィックスは、Citrix ADC 12.0 ビルド 56.20 以降では廃止され、高度なポリシー式ではサポートされなくなりました。

次の表は、デフォルトの構文式プレフィックスの一覧です。これらのプレフィックスの説明とそれらに対して指定できる演算子への相互参照があります。プレフィックスによっては、複数のタイプの演算子を使用できることに注意してください。たとえば、テキスト用の演算子または HTTP ヘッダーの演算子を使用して Cookie を解析できます。

次の表の任意の要素を完全な式として使用することも、さまざまな演算子を使用してこれらの式要素を他の要素と組み合わせて、より複雑な式を形成することもできます。

注意**:** 次の表の「説明」列には、プレフィックスの使用方法とプレフィックスの適用可能な演算子に関する追加情報への相互参照が含まれています。

詳細については、表を表示するには、式 PDF を参照してください。

エクスプレッションリファレンス-クラシックエクスプレッション

October 7, 2021

> 警告
>
> Citrix ADC 12.0 ビルド 56.20 以降では、従来のポリシー式はサポートされなくなりました。また、高度なポリシーを使用することをお勧めします。詳細については、「高度なポリシー」を参照してください。

画面の左側の目次に表示されるサブトピックには、Citrix ADC クラシック式を示す表が含まれています。

演算子の表では、各演算子の結果タイプが説明の先頭に表示されます。他の表では、各式のレベルが説明の先頭に表示されます。名前付き式の場合、各式が全体として表示されます。

オペレーター

式要素	定義
==	ブール値。現在の式が引数と等しい場合に TRUE を返します。テキスト操作の場合、比較する項目は互いに正確に一致する必要があります。数値演算の場合、項目は同じ数値に評価される必要があります。
!=	ブール値。現在の式が引数と等しくない場合に TRUE を返します。テキスト操作では、比較する項目が互いに正確に一致してはなりません。数値演算の場合、項目は同じ数値に評価できません。
CONTAINS	ブール値。現在の式に、引数で指定された文字列が含まれている場合に TRUE を返します。
NOTCONTAINS	ブール値。現在の式に、引数で指定された文字列が含まれていない場合に TRUE を返します。

式要素	定義
CONTENTS	Text。現在の式の内容を返します。
EXISTS	ブール値。現在の式で指定された項目が存在する場合は TRUE を返します。
NOTEXISTS	ブール値。現在の式で指定された項目が存在しない場合に TRUE を返します。
>	ブール値。現在の式が引数より大きい数に評価される場合に TRUE を返します。
<	ブール値。現在の式が引数より小さい数に評価される場合に TRUE を返します。
>=	ブール値。現在の式が引数以上の数に評価される場合に TRUE を返します。
<=	ブール値。現在の式が引数以下の数に評価される場合に TRUE を返します。

一般的な表現

式要素	定義
REQ	[フロータイプ]: 着信（または要求）パケット上で動作する。
REQ.HTTP	プロトコル。HTTP 要求に対して操作します。
REQ.HTTP.METHOD	修飾子。HTTP メソッドを指定します。
REQ.HTTP.URL	修飾子。URL を指定します。
REQ.HTTP.URLTOKENS	修飾子。URL トークンを指定します。
REQ.HTTP.VERSION	修飾子。HTTP バージョンを指定します。
REQ.HTTP.HEADER	修飾子。HTTP ヘッダーを指定します。
REQ.HTTP.URLLEN	修飾子。URL の文字数を指定します。
REQ.HTTP.URLQUERY	修飾子。URL のクエリ部分を指定します。
REQ.HTTP.URLQUERYLEN	修飾子。URL のクエリ部分の長さを指定します。
REQ.SSL	プロトコル。SSL 要求に対して操作します。
REQ.SSL.CLIENT.CERT	修飾子。クライアント証明書全体を指定します。
REQ.SSL.CLIENT.CERT.SUBJECT	修飾子。クライアント証明書のサブジェクトを指定します。

式要素	定義
REQ.SSL.CLIENT.CERT.ISSUER	修飾子。クライアント証明書の発行者を指定します。
REQ.SSL.CLIENT.CERT.SIGALGO	修飾子。クライアント証明書で使用される検証アルゴリズムを指定します。
REQ.SSL.CLIENT.CERT.VERSION	修飾子。クライアント証明書のバージョンを指定します。
REQ.SSL.CLIENT.CERT.VALIDFROM	修飾子。クライアント証明書が無効になる日付を指定します。
REQ.SSL.CLIENT.CERT.VALIDTO	修飾子。クライアント証明書が無効になる日付を指定します。
REQ.SSL.CLIENT.CERT.SERIALNUMBER	修飾子。クライアント証明書のシリアル番号を指定します。
REQ.SSL.CLIENT.CIPHER.TYPE	修飾子。クライアントが使用する暗号化プロトコルを指定します。
REQ.SSL.CLIENT.CIPHER.BITS	修飾子。クライアントの SSL キーで使用されるビット数を指定します。
REQ.SSL.CLIENT.SSL.VERSION	修飾子。クライアントが使用している SSL バージョンを指定します。
REQ.TCP	プロトコル。着信 TCP パケット上で動作する。
REQ.TCP.SOURCEPORT	修飾子。着信パケットの送信元ポートを指定します。
REQ.TCP.DESTPORT	修飾子。着信パケットの宛先ポートを指定します。
REQ.IP	プロトコル。着信 IP パケット上で動作する。
REQ.IP.SOURCEIP	修飾子。着信パケットの送信元 IP を指定します。
REQ.IP.DESTIP	修飾子。着信パケットの宛先 IP を指定します。
RES	[フロータイプ]: 発信（または応答）パケット上で動作する。
RES.HTTP	プロトコル。HTTP 応答で操作します。
RES.HTTP.VERSION	修飾子。HTTP バージョンを指定します。
RES.HTTP.HEADER	修飾子。HTTP ヘッダーを指定します。
RES.HTTP.STATUSCODE	修飾子。HTTP レスポンスのステータスコードを指定します。
RES.TCP	プロトコル。着信 TCP パケット上で動作する。
RES.TCP.SOURCEPORT	修飾子。発信パケットの送信元ポートを指定します。

式要素	定義
RES.TCP.DESTPORT	修飾子。発信パケットの宛先ポートを指定します。
RES.IP	プロトコル。発信 IP パケット上で動作する。
RES.IP.SOURCEIP	修飾子。発信パケットの送信元 IP を指定します。これは、IPv4 形式または IPv6 形式にすることができます。例：add expr exp3 “sourceip == 10.102.32.123 –netmask 255.255.255.0 && destip == 2001::23/120”.
RES.IP.DESTIP か	修飾子。発信パケットの宛先 IP を指定します。

クライアントセキュリティ式

次のソフトウェアを使用して Access Gateway でクライアント設定を構成するための式。

- アンチウイルス
- パーソナルファイアウォール
- アンチスパム
- インターネットセキュリティ

使用例については、http://support.citrix.com/article/CTX112599を参照してください。

実際の式	定義
CLIENT.APPLICATION.AV(<NAME>.VERSION == <VERSION>)	クライアントが、指定されたアンチウイルスプログラムおよびバージョンを実行しているかどうかを確認します。
CLIENT.APPLICATION.AV(<NAME>.VERSION != <VERSION>)	クライアントが、指定されたアンチウイルスプログラムおよびバージョンを実行していないかどうかをチェックします。
CLIENT.APPLICATION.PF(<NAME>.VERSION == <VERSION>)	クライアントが、指定されたパーソナルファイアウォールプログラムおよびバージョンを実行しているかどうかを確認します。
CLIENT.APPLICATION.PF(<NAME>.VERSION != <VERSION>)	クライアントが、指定されたパーソナルファイアウォールプログラムおよびバージョンを実行していないかどうかを確認します。
CLIENT.APPLICATION.IS(<NAME>.VERSION == <VERSION>)	クライアントが指定のインターネットセキュリティプログラムおよびバージョンを実行しているかどうかを確認します。

実際の式	定義
CLIENT.APPLICATION.IS(<NAME>.VERSION != <VERSION>)	クライアントが、指定されたインターネットセキュリティプログラムおよびバージョンを実行していないかどうかをチェックします。
CLIENT.APPLICATION.AS(<NAME>.VERSION == <VERSION>)	クライアントが、指定されたスパム対策プログラムおよびバージョンを実行しているかどうかを確認します。
CLIENT.APPLICATION.AS(<NAME>.VERSION != <VERSION>)	クライアントが、指定されたスパム対策プログラムおよびバージョンを実行していないかどうかを確認します。

ネットワークベースの式

式	定義
REQ	[フロータイプ]: 着信パケットまたは要求パケット上で動作する。
REQ.VLANID	修飾子。仮想 LAN（VLAN）ID 上で動作する。
REQ.INTERFACE.ID	修飾子。指定された Citrix ADC インターフェイスの ID 上で動作する。
REQ.INTERFACE.RXTHROUGHPUT	修飾子。指定された Citrix ADC インターフェイスの未加工受信パケットスループット上で動作する。
REQ.INTERFACE.TXTHROUGHPUT	修飾子。指定された Citrix ADC インターフェイスの生の送信パケットスループット上で動作する。
REQ.INTERFACE.RXTXTHROUGHPUT	修飾子。指定された Citrix ADC インターフェイスの raw 受信パケットおよび送信パケットスループット上で動作する。
REQ.ETHER.SOURCEMAC	修飾子。送信元 MAC アドレス上で動作する。
REQ.ETHER.DESTMAC	修飾子。宛先 MAC アドレス上で動作する。
RES	[フロータイプ]: 発信（または応答）パケット上で動作する。
RES.VLANID	修飾子。仮想 LAN（VLAN）ID 上で動作する。
RES.INTERFACE.ID	修飾子。指定された Citrix ADC インターフェイスの ID 上で動作する。

式	定義
RES.INTERFACE.RXTHROUGHPUT	修飾子。指定された Citrix ADC インターフェイスの未加工受信パケットスループット上で動作する。
RES.INTERFACE.TXTHROUGHPUT	修飾子。指定された Citrix ADC インターフェイスの生の送信パケットスループット上で動作する。
RES.INTERFACE.RXTXTHROUGHPUT	修飾子。指定された Citrix ADC インターフェイスの raw 受信パケットおよび送信パケットスループット上で動作する。
RES.ETHER.SOURCEMAC	修飾子。送信元 MAC アドレス上で動作する。
RES.ETHER.DESTMAC	修飾子。宛先 MAC アドレス上で動作する。

日付/時刻式

式	定義
TIME	修飾子。日付と時刻（GMT）で動作。
DATE	修飾子。日付、GMT に作動します。
DAYOFWEEK	指定された曜日 GMT（グリニッジ標準時）に動作する。

ファイル・システムの表現

Citrix Gateway のファイル転送ユーティリティ（VPN ポータル）を使用してファイル共有にアクセスするユーザーおよびグループの承認ポリシーでファイルシステム式を指定できます。これらの式は、Citrix Gateway のファイル転送認証機能と連動し、ファイルサーバー、フォルダー、ファイルへのユーザーアクセスを制御します。たとえば、認可ポリシーでこれらの式を使用して、ファイルの種類とサイズに基づいてアクセスを制御できます。

詳細については、ファイル名式 pdf を参照してください。

注：ファイルシステム式は正規表現をサポートしていません。

組み込みの名前付き式 (一般)

式	定義
ns_all_apps_ncomp	宛先ポートが 0 ～65535 の間の接続をテストします。つまり、すべてのアプリケーションをテストします。

式	定義
ns_cachecontrol_nocache	値「no-cache」を含む HTTP キャッシュ制御ヘッダーとの接続をテストします。
ns_cachecontrol_nostore	値「no-store」を含む HTTP キャッシュ制御ヘッダーとの接続をテストします。
ns_cmpclient	クライアントをテストして、圧縮されたコンテンツを受け入れるかどうかを判断します。
ns_content_type	「text」を含む HTTP コンテンツタイプヘッダーとの接続をテストします。
ns_css	「text/css」を含む HTTP コンテンツタイプヘッダーとの接続をテストします。
ns_ext_asp	.asp という文字列を含む任意の URL への HTTP 接続をテストします。つまり、アクティブなサーバーページ (ASP) への接続です。
ns_ext_cfm	文字列.cfm を含む任意の URL への HTTP 接続をテストします。
ns_ext_cgi	文字列.cgi を含む任意の URL への HTTP 接続をテストします。つまり、共通 Gateway インターフェイス（CGI）スクリプトへの接続です。
ns_ext_ex	.ex という文字列を含む任意の URL への HTTP 接続をテストします。
ns_ext_exe	文字列「.ex」を含む任意の URL への HTTP 接続、つまり実行可能ファイルへの接続をテストします。
ns_ext_htx	.htx という文字列を含む任意の URL への HTTP 接続をテストします。
ns_ext_not_gif	.png という文字列を含まない URL への HTTP 接続をテストします。つまり、GIF イメージではない URL への接続をすべてテストします。
ns_ext_not_jpeg	.jpg という文字列を含まない URL への HTTP 接続をテストします。つまり、JPEG イメージではない URL への接続をテストします。
ns_ext_shtml	文字列.shtml を含む任意の URL への HTTP 接続をテストします。つまり、サーバー解析された HTML ページへの接続です。
ns_false	常に FALSE の値を返します。

式	定義
ns_farclient	クライアントは Citrix ADC とは異なる地域にあり、クライアントの IP アドレス内の地域によって決定されます。次のリージョンが事前定義されています。192.0.0.0 – 193.255.255.255: Multi-regional, 194.0.0.0 – 195.255.255.255: European Union, 196.0.0.0 – 197.255.255.255: Other1, 198.0.0.0 – , 199.255.255.255: North America, 200.0.0.0 – 201.255.255.255: Central and South America, 202.0.0.0 – 203.255.255.255: Pacific Rim, 204.0.0.0 – 205.255.255.255: Other2, and 206.0.0.0 – 207.255.255.255: Other3
ns_header_cookie	Cookie ヘッダーを含む HTTP 接続をテストします。
ns_header_pragma	Pragma:no-cache ヘッダーを含む HTTP 接続をテストします。
ns_mozilla_47	ユーザエージェントヘッダーに Mozilla/4.7 という文字列が含まれている HTTP 接続をテストします。つまり、Mozilla 4.7 ウェブブラウザを使用しているクライアントからの接続です。
ns_msexcel	コンテンツタイプヘッダーに application/vnd.msexcel という文字列が含まれている HTTP 接続をテストします。つまり、Microsoft Excel スプレッドシートを送信する接続です。
ns_msie	ユーザーエージェントヘッダーに MSIE という文字列が含まれている HTTP 接続をテストします。つまり、Internet Explorer Web ブラウザの任意のバージョンを使用しているクライアントからの接続です。
ns_msppt	コンテンツタイプヘッダーに application/vnd.ms-powerpoint という文字列が含まれている HTTP 接続をテストします。つまり、Microsoft PowerPoint ファイルを送信する接続です。
ns_msword	コンテンツタイプヘッダーに application/vnd.msword という文字列が含まれている HTTP 接続をテストします。つまり、Microsoft Word ファイルを送信する接続です。

式	定義
ns_non_get	GET 以外の HTTP メソッドを使用する HTTP 接続をテストします。
ns_slowclient	クライアントと Citrix ADC 間の平均ラウンドトリップ時間が 80 ミリ秒を超える場合に TRUE を返します。
ns_true	すべてのトラフィックに対して TRUE を返します。
ns_url_path_bin	URL パスをテストして、/bin/ ディレクトリを指しているかどうかを確認します。
ns_url_path_cgibin	URL パスをテストして、CGI-BIN ディレクトリを指しているかどうかを確認します。
ns_url_path_exec	URL パスをテストして、/exec/ディレクトリを指しているかどうかを確認します。
ns_url_tokens	URL トークンの存在をテストします。
ns_xmldata	XML データの存在をテストします。

組み込みの名前付き式（アンチウイルス）

式	定義
McAfee Virus Scan 11	クライアントが McAfee VirusScan の最新バージョンを実行しているかどうかをテストします。
McAfee Antivirus	クライアントが McAfee Antivirus のいずれかのバージョンを実行しているかどうかをテストします。
Symantec AntiVirus 10 (with Updated Definition File)	クライアントが Symantec AntiVirus の最新バージョンを実行しているかどうかをテストします。
Symantec AntiVirus 6.0	クライアントが Symantec AntiVirus 6.0 を実行しているかどうかをテストします。
Symantec AntiVirus 7.5	クライアントが Symantec AntiVirus 7.5 を実行しているかどうかをテストします。
TrendMicro OfficeScan 7.3	クライアントがトレンドマイクロシステムズの OfficeScan バージョン 7.3 を実行しているかどうかをテストします。
TrendMicro AntiVirus 11.25	クライアントがトレンドマイクロシステムズのアンチウイルス（バージョン 11.25）を実行しているかどうかをテストします。

式	定義
Sophos Antivirus 4	クライアントが Sophos Antivirus バージョン 4 を実行しているかどうかをテストします。
Sophos Antivirus 5	クライアントが Sophos Antivirus バージョン 5 を実行しているかどうかをテストします。
Sophos Antivirus 6	クライアントが Sophos Antivirus バージョン 6 を実行しているかどうかをテストします。

組み込みの名前付き式（パーソナルファイアウォール）

式	定義
TrendMicro OfficeScan 7.3	クライアントがトレンドマイクロシステムズの OfficeScan バージョン 7.3 を実行しているかどうかをテストします。
Sygate Personal Firewall 5.6	クライアントが Sygate パーソナルファイアウォールバージョン 5.6 を実行しているかどうかをテストします。
ZoneAlarm Personal Firewall 6.5	クライアントが ZoneAlarm パーソナルファイアウォールバージョン 6.5 を実行しているかどうかをテストします。

組み込みの名前付き式 **(**クライアントセキュリティ**)**

式	定義
Norton Internet Security	クライアントがノートンインターネットセキュリティのいずれかのバージョンを実行しているかどうかをテストします。

デフォルトの構文式とポリシーの概要例

October 7, 2021

次の表に、独自の既定の構文式の基礎として使用できる既定の構文式の例を示します。

表 1. デフォルトの構文式の例

式タイプ	サンプル式
HTTP リクエストで使用されるメソッドを見てください。	`http.req.method.eq(post)http.req.method.eq(get)`

HTTP 要求（req）または応答（res）のキャッシュコントロールまたはプラグマヘッダ値を確認します。	`http.req.header(”Cache-Control”).contains(”no-store”)`
	`http.req.header(”Cache-Control”).contains(”no-cache”)`
	`http.req.header(”Pragma”).contains(”no-cache”)`
	`http.res.header(”Cache-Control”).contains(”private”)`
	`http.res.header(”Cache-Control”).contains(”public”)`
	`http.res.header(”Cache-Control”).contains(”must-revalidate”)http.res.header(”Cache-Control”).contains (”proxy-revalidate”)`
	`http.res.header(”Cache-Control”).contains(”max-age”)`

リクエスト（req）またはレスポンス（res）にヘッダーの存在を確認します。	`http.req.header(”myHeader”).exists` `http.res.header(”myHeader”).exists`
ファイル拡張子に基づいて、HTTP リクエスト内の特定のファイルの種類を探します。	`http.req.url.contains(”.html”)http.req.url.contains(”.cgi”)http.req.url.contains(”.asp”)http.req.url.contains(”.exe”)http.req.url.contains(”.cfm”)http.req.url.contains(”.ex”)http.req.url.contains(”.shtml”)http.req.url.contains(”.htx”)http.req.url.contains(”/cgi-bin/”)http.req.url.contains(”/exec/”)http.req.url.contains(”/bin/”)`
HTTP リクエストで特定のファイルの種類以外のものを探します。	`http.req.url.contains(”.png”).not;` `http.req.url.contains(”.jpeg”).not`

式タイプ	サンプル式
Content-Type ヘッダーに基づいて、HTTP 応答で送信されるファイルの種類を確認します。	`http.res.header("Content-Type").contains("text")http.res.header("Content-Type").contains "application/msword")http.res.header("Content-Type").contains("vnd.ms-excel")http.res.header("Content-Type").contains("application/vnd.ms-powerpoint"); http.res.header("Content-Type").contains("text/css"); http.res.header("Content-Type").contains("text/xml"); http.res.header("Content-Type").contains("image/")`
この応答に有効期限ヘッダーが含まれているかどうかを確認します。	`http.res.header("Expires").exists`
レスポンスで Set-Cookie ヘッダーを確認します。	`http.res.header("Set-Cookie").exists`
応答を送信したエージェントを確認します。	`http.res.header("User-Agent").contains("Mozilla/4.7")http.res.header("User-Agent").contains("MSIE")`
リクエストの本文の最初の 1024 バイトが文字列「some text」で始まるかどうかを確認します。	`http.req.body(1024).contains("some text")`

次の表に、一般的に使用される関数のポリシー設定とバインディングの例を示します。

表 2. デフォルトの構文式とポリシーの例

目的	例
書き換え機能を使用して、HTTP 応答の本文内http://with https://の出現を置き換えます。	`add rewrite action httpRewriteAction replace_all http.res.body(50000)"\"https://\""-pattern http://add rewrite policy demo_rep34312 "http.res.body(50000).contains(\"http://\")" httpRewriteAction`

目的	例
HTTP 本体の最初の 1000 バイトで「abcd」のすべての出現を「1234」に置き換えます。	add rewrite action abcdTo1234Action replace_all ”http.req.body(1000)”” \”1234\””-pattern abcd add rewrite policy abcdTo1234Policy ”http.req.body(1000).contains(\”abcd\”)” abcdTo1234Action bind rewrite global abcdTo1234Policy 100 END -type REQ_OVERRIDE
HTTP バージョンを 1.0 にダウングレードして、サーバが HTTP 応答をチャンクしないようにします。	add rewrite action downgradeTo1.0 Action replace http.req.version.minor ”\”0\””add rewrite policy downgradeTo1.0Policy ”http.req.version.minor.eq(1)”downgradeTo1.0 Action bind lb vserver myLBVserver -policyName downgradeTo1.0Policy -priority 100 -gotoPriorityExpression NEXT -type REQUEST
すべての応答で HTTP または HTTPS プロトコルへの参照を削除します。これにより、ユーザーの接続が HTTP の場合は HTTP を使用してリンクが開かれ、ユーザーの接続が HTTPS の場合は HTTPS を使用してリンクが開かれます。	add rewrite action remove_http_https replace_all ”http.res.body(1000000).set_text_mode(ignorecase)””\”//\””-pattern ”re~https?://|HTTPS?://~”add rewrite policy remove_http_https **true** remove_http_https bind lb vserver test_vsvr -policyName remove_http_https -priority 20 -gotoPriorityExpression NEXT -type RESPONSE

目的	例
すべての URL で http: のインスタンスを https: に書き換えます。	`add responder action httpToHttpsAction redirect "\"https://\" + http.req.hostname + http.req.url"-bypassSafetyCheck YES add responder policy httpToHttpsPolicy "!CLIENT.SSL.IS_SSL" httpToHttpsAction bind responder global httpToHttpsPolicy 1 END -type OVERRIDE`
URL A から URL B にリダイレクトするように URL を変更します。この例では、パスに「file5.html」が追加されます。	`add responder action appendFile5Action redirect \"http://\" + http.req.hostname + http.req.url + \"/file5.html\""-bypassSafetyCheck YES add responder policy appendFile5Policy "http.req.url.eq(\"/testsite\")" appendFile5Action bind responder global appendFile5Policy 1 END -type OVERRIDE`
外部 URL を内部 URL にリダイレクトします。	`add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server(''"www.my.host.com"'add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("www.external.host.com")'act_external_to_internal bind rewrite global pol_external_to_internal 100 END -type REQ_OVERRIDE`

目的	例
クエリ文字列を持つリクエストを www.Webn.example.com にリダイレクトします。値 n は、クエリ文字列のサーバーパラメータ (server=5 など) から派生します。	`add rewrite action act_redirect_query REPLACE q##http.req.header(”Host”).before_str(”.example.com”)''”Web”+ http.req.url.query.value(”server”)## add rewrite policy pol_redirect_query q##http.req.header(”Host”).eq(”www.example.com”)&& http.req.url.contains(”?”)'act_redirect_query##`
URL からの 1 秒あたりのリクエスト数を制限します。	`add ns limitSelector ip_limit_selector http.req.url ”client.ip.src”add ns limitIdentifier ip_limit_identifier -threshold 4 -timeSlice 3600 -mode request_rate -limitType smooth -selectorName ip_limit_selector add responder action my_Web_site_redirect_action redirect ”\”http://www.mycompany.com/\””add responder policy ip_limit_responder_policy ”http.req.url.contains(\”myasp.asp\”)&& sys.check_limit (\”ip_limit_identifier\”)”my_Web_site_redirect_action bind responder global ip_limit_responder_policy 100 END -type default`
クライアントの IP アドレスを確認しますが、要求を変更せずに要求を渡します。	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER (”x-forwarded-for”).EXISTS HTTP.REQ.HEADER (”client-ip”).EXISTS'NOREWRITE bind rewrite global check_client_ip_policy 100 END`

目的	例
リクエストから古いヘッダーを削除し、NS クライアントヘッダーを挿入します。	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER(”x-forwarded-for”).EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER(”client-ip”).EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER(”x-forwarded-for”).EXISTS HTTP.REQ.HEADER(”client-ip”).EXISTS 'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END`

目的	例
リクエストから古いヘッダーを削除し、NS クライアントヘッダーを挿入してから、挿入されたヘッダーの値に古いヘッダーと Citrix ADC アプライアンスの接続 IP アドレスが含まれるように、「ヘッダーの挿入」アクションを変更します。この例では、最後のセット書き換えアクションを除いて、前の例を繰り返していることに注意してください。	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER(”x-forwarded-for”).EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER(”client-ip”).EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER(”x-forwarded-for”).EXISTS HTTP.REQ.HEADER(”client-ip”).EXISTS 'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END set rewrite action insert_ns_client_header -stringBuilderExpr 'HTTP.REQ.HEADER(”x-forwarded-for”).VALUE(0)+ ” ” + HTTP.REQ.HEADER(”client-ip”).VALUE(0)+ ” ” + CLIENT.IP.SRC'-bypassSafetyCheck YES`

書き換えのデフォルト構文ポリシーのチュートリアルの例

October 7, 2021

書き換え機能を使用すると、HTTP ヘッダーの任意の部分を変更できます。また、応答の場合は HTTP 本文を変更できます。この機能を使用すると、不要な HTTP ヘッダーの削除、内部 URL のマスキング、Web ページのリダイレクト、クエリやキーワードのリダイレクトなど、いくつかの便利なタスクを実行できます。

次の例では、まず書き換えアクションと書き換えポリシーを作成します。次に、ポリシーをグローバルにバインドします。

このドキュメントでは、次の詳細について説明します。

- 外部 URL の内部 URL へのリダイレクト
- クエリのリダイレクト
- HTTP から HTTPS への書き換え
- 不要なヘッダーの削除
- Web サーバーのリダイレクトの削減
- サーバヘッダーのマスキング
- プレーンテキストを URL エンコードされた文字列に変換し、反対の方法

コマンドおよび構文の説明の詳細については、Rewrite コマンドリファレンスページを参照してください 。

外部 **URL** を内部 **URL** にリダイレクトする

この例では、外部 URL を内部 URL にリダイレクトする書き換えアクションおよび書き換えポリシーを作成する方法について説明します。書き換えを実行する act_external_to_internal というアクションを作成します。次に、pol_external_to_internal というポリシーを作成します。

CLI を使用して外部 **URL** を内部 **URL** にリダイレクトするには

- 書き換えアクションを作成するには、コマンドプロンプトで次のように入力します。

```
add rewrite action act_external_to_internal REPLACE “http.req.hostname.
server” “\” host_name_of_internal_Web_server\” ”
```

- 書き換えポリシーを作成するには、Citrix ADC コマンドプロンプトで次のように入力します。

```
add rewrite policy pol_external_to_internal ”http.req.hostname.server.eq(\”
host_name_of_external_Web_server\”)”act_external_to_internal
```

- ポリシーをグローバルにバインドします。

構成ユーティリティを使用して外部 **URL** を内部 **URL** にリダイレクトするには

1. **AppExpert** > 書き換え > アクションに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [書き換えアクションを作成] ダイアログボックスで、act_external_to_internal という名前を入力します。
4. HTTP サーバのホスト名を内部サーバ名に置き換えるには、「タイプ」リストボックスから「置換」を選択します。
5. ヘッダー名フィールドに「**Host**」と入力します。
6. 置換テキストフィールドの文字列式に、Web サーバーの内部ホスト名を入力します。
7. [**Create**] をクリックしてから、[**Close**] をクリックします。
8. ナビゲーションペインで、[**Policies**] をクリックします。
9. 詳細ペインで、[**Add**] をクリックします。
10. 「名前」フィールドに「pol_external_to_internal」と入力します。このポリシーは、Web サーバーへの接続を検出します。
11. [アクション] ドロップダウンメニューで、アクション act_external_to_internal を選択します。
12. 式エディタで、次の式を作成します。

```
HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")
<!--NeedCopy-->
```

1. 新しいポリシーをグローバルにバインドします。

クエリのリダイレクト

この例では、クエリを適切な URL にリダイレクトする書き換えアクションおよび書き換えポリシーを作成する方法について説明します。この例では、リクエストに **www.example.com** に設定されたホストヘッダーと、文字列 **/query.cgi?** の **GET** メソッドが含まれていることを前提としています。サーバ **= 5** です。リダイレクトは、ホストヘッダーからドメイン名を抽出し、クエリ文字列から番号を抽出し、ユーザーのクエリをサーバー **Web5.example.com** にリダイレクトします。ここで、ユーザーのクエリの残りの部分が処理されます。

> 注:
>
> 次のコマンドは複数の行に表示されますが、改行せずに 1 行で入力する必要があります。

CLI を使用してクエリを適切な **URL** にリダイレクトするには

- HTTP サーバのホスト名を内部サーバ名に置き換える act_redirect_query という名前の書き換えアクションを作成するには、次のように入力します。

```
add rewrite action act_redirect_query REPLACE http.req.header("Host").
before_str(".example.com")'"Web" + http.req.url.query.value("server")'
```


- pol_redirect_query という名前の書き換えポリシーを作成するには、Citrix ADC コマンドプロンプトで次のコマンドを入力します。このポリシーは、クエリ文字列を含む Web サーバーへの接続を検出します。クエリ文字列を含まない接続には、このポリシーを適用しないでください。

```
add rewrite policy pol_redirect_query 'http.req.header(”Host”).eq(www.
example.com)&& http.req.url.contains(”?”)'act_redirect_query
```

- 新しいポリシーをグローバルにバインドします。

この書き換えポリシーは、非常に特殊なものであり、他の書き換えポリシーの前に実行する必要があるため、高い優先度を割り当てることをお勧めします。優先度 1 を割り当てると、最初に評価されます。

HTTP から **HTTPS** への書き換え

この例では、文字列「HTTP」で始まるすべての URL を検索し、その文字列を「https」に置き換えるように Web サーバーレスポンスを書き換える方法について説明します。サーバーを HTTP から HTTPS に移動した後に Web ページを更新する必要がないようにするために使用できます。

CLI を使用して **HTTP URL** を **HTTPS** にリダイレクトするには

- 文字列「HTTP」のすべてのインスタンスを文字列「https」に置き換える act_replace_http_with_https という名前の書き換えアクションを作成するには、次のコマンドを入力します。

```
add rewrite action act_replace_http_with_https replace_all 'http.res.body
(100)''”https”'-pattern http
```

- Web サーバへの接続を検出する pol_replace_http_with_https という名前の書き換えポリシーを作成するには、次のコマンドを入力します。

```
add rewrite policy pol_replace_http_with_https TRUE act_replace_http_with_https
 NOREWRITE
```

- 新しいポリシーをグローバルにバインドします。

この書き換え操作のトラブルシューティングについては、「ケーススタディ:HTTP リンクを HTTPS に変換するための書き換えポリシーが機能しない」を参照してください。

不要なヘッダーの削除

この例では、Rewrite ポリシーを使用して不要なヘッダーを削除する方法について説明します。具体的には、次のヘッダーを削除する方法の例を示します。

- エンコーディングヘッダーを受け入れます。HTTP 応答から Accept Encoding ヘッダーを削除すると、応答の圧縮が防止されます。
- コンテンツの場所ヘッダー。HTTP 応答から Content Location ヘッダーを削除すると、サーバーがセキュリティ侵害を許す可能性のある情報をハッカーに提供できなくなります。

HTTP 応答からヘッダーを削除するには、書き換えアクションと書き換えポリシーを作成し、ポリシーをグローバルにバインドします。

CLI を使用して適切な書き換えアクションを作成するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、Accept Encoding ヘッダーを削除して応答の圧縮を防止するか、Content Location ヘッダーを削除します。

- `add rewrite action "act_remove-ae"delete_http_header "Accept-Encoding"`
- `add rewrite action "act_remove-cl"delete_http_header "Content-Location"`

CLI を使用して適切な書き換えポリシーを作成するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、Accept Encoding ヘッダーまたは Content Location ヘッダーを削除します。

- `add rewrite policy "pol_remove-ae"true "act_remove-ae"`
- `add rewrite policy "pol_remove-cl"true "act_remove-cl"`

CLI を使用してポリシーをグローバルにバインドするには

コマンドプロンプトで、必要に応じて次のコマンドのいずれかを入力し、作成したポリシーをグローバルにバインドします。

- `bind rewrite global pol_remove_ae 100`
- `bind rewrite global pol_remove_cl 200`

Web サーバーのリダイレクトの削減

この例では、書き換えポリシーを使用して、ホームページへの接続と、末尾がスラッシュ (/) で終わる他の URL をサーバーの既定のインデックスページに変更し、リダイレクトを防止し、サーバーの負荷を軽減する方法について説明します。

CLI を使用してデフォルトのホームページを含めるようにディレクトリレベルの **HTTP** 要求を変更するには

- スラッシュで終わる URL を変更して、デフォルトのホームページ index.html を含むように変更する action-default-homepage という名前の書き換えアクションを作成するには、次のように入力します。

```
add rewrite action "action-default-homepage"replace http.req.url.path "\"/index.html\""
```

- ホームページへの接続を検出して新しいアクションを適用する policy-default-homepage という名前の書き換えポリシーを作成するには、次のように入力します。


```
add rewrite policy ”policy-default-homepage”q\##http.req.url.path.EQ(”/”)”
action-default-homepage”\##
```

- 新しいポリシーをグローバルにバインドして有効にします。

サーバヘッダーのマスキング

この例では、書き換えポリシーを使用して、Web サーバーからの HTTP 応答の Server ヘッダー内の情報をマスクする方法について説明します。このヘッダーには、ハッカーがウェブサイトを侵害するために使用できる情報が含まれています。ヘッダーをマスクしても、熟練したハッカーがサーバーに関する情報を見つけることを妨げることはありませんが、Web サーバーのハッキングが難しくなり、ハッカーが保護されていないターゲットを選択するように促します。

CLI からの応答で **Server** ヘッダーをマスクするには

1. Server ヘッダーの内容を無意味な文字列に置き換える act_mask-server という名前の書き換えアクションを作成するには、次のように入力します。

```
add rewrite action ”act_mask-server”replace ”http.RES.HEADER(\”Server\”)””
\”Web Server 1.0\””
```

1. すべての接続を検出する pol_mask-server という名前の書き換えポリシーを作成するには、次のように入力します。

```
add rewrite policy ”pol_mask-server”true ”act_mask-server”
```

1. 新しいポリシーをグローバルにバインドして有効にします。

プレーンテキストを **URL** エンコードされた文字列に変換する方法と反対の方法

次の式は、プレーンテキストを URL エンコードされた文字列に変換し、反対の方法で変換します。

1. URL_RESERVED_CHARS_SAFE（URL エンコードする文字列）。

例:

```
1 (”abc def&123”).URL_RESERVED_CHARS_SAFE
2 Output will be
3  “abc%20def%26123”  which is url encoded.
4 <!--NeedCopy-->
```

1. SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE. (URL ENCODED to string)

例:


```
("abc%20def%26123").SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE
Output will be
 "abc def&123"
<!--NeedCopy-->
```

クラシックポリシーのチュートリアルの例

October 7, 2021

以下の例では、Citrix Gateway、アプリケーションファイアウォール、SSL など、特定の Citrix ADC 機能に対する従来のポリシー設定の便利な例について説明します。

このドキュメントでは、次の詳細について説明します。

- 有効なクライアント証明書を確認するための Citrix Gateway ポリシー
- ショッピングカートアプリケーションを保護するためのアプリケーションファイアウォールポリシー
- スクリプト Web ページを保護するためのアプリケーションファイアウォールポリシー
- 特定の IP からパケットをドロップする DNS ポリシー
- 有効なクライアント証明書を要求する SSL ポリシー

有効なクライアント証明書をチェックするための **Citrix Gateway** ポリシー

以下のポリシーにより、Citrix ADC は、企業の SSL VPN への接続を確立する前に、クライアントが有効な証明書を提示するようにします。

コマンドラインインターフェイスを使用して有効なクライアント証明書を確認するには

- クライアント証明書の認証を実行するアクションを追加します。

  ```
  add ssl action act1 -clientAuth DOCLIENTAUTH
  ```

- クライアント要求を評価する SSL ポリシーを作成します。

  ```
  add ssl policy pol1 -rule "REQ.HTTP.METHOD == GET"-action act1
  ```

- Web サーバーに送信される要求の HTTP ヘッダーに証明書発行者の詳細を挿入する書き換えアクションを追加します。

  ```
  add rewrite action act2 insert_http_header "CertDN"CLIENT.SSL.CLIENT_CERT
  .SUBJECT
  ```

- クライアント証明書が存在する場合は、証明書発行者の詳細を挿入する書き換えポリシーを作成します。

  ```
  add rewrite policy pol2 "CLIENT.SSL.CLIENT_CERT.EXISTS"act2
  ```


これらの新しいポリシーを Citrix ADC VIP にバインドして有効にします。

ショッピングカートのアプリケーションを保護するためのアプリケーションファイアウォールポリシー

ショッピングカートのアプリケーションは、クレジットカード番号や有効期限などの重要な顧客情報を処理し、バックエンドデータベースサーバーにアクセスします。多くのショッピングカートのアプリケーションでは、レガシー CGI スクリプトも使用されています。このスクリプトには、書いた時点で不明であったセキュリティ上の欠陥が含まれていても、現在はハッカーや ID 泥棒に知られています。

ショッピングカートのアプリケーションは、次の攻撃に対して特に脆弱です。

- クッキーの改ざん。ショッピングカートアプリケーションが Cookie を使用し、ユーザーがアプリケーションに戻った Cookie に対して適切なチェックを行わない場合、攻撃者は Cookie を変更し、別のユーザーの資格情報でショッピングカートアプリケーションにアクセスする可能性があります。攻撃者は、そのユーザーとしてログオンすると、正当なユーザーに関する機密性の高い個人情報を取得したり、正当なユーザーのアカウントを使用して注文を行ったりする可能性があります。
- **SQL** インジェクション。ショッピングカートアプリケーションは、通常、バックエンドデータベースサーバーにアクセスします。アプリケーションが Web フォームのフォームフィールドでユーザーが返すデータに対して適切な安全性チェックを実行しない限り、攻撃者は Web フォームを使用して、不正な SQL コマンドをデータベースサーバーに挿入する可能性があります。攻撃者は通常、この種の攻撃を使用して、データベースから機密性の高い個人情報を取得したり、データベース内の情報を修正したりします。

次の構成は、これらの攻撃やその他の攻撃からショッピングカートのアプリケーションを保護します。

構成ユーティリティを使用してショッピングカートアプリケーションを保護するには

1. [セキュリティ**] > [**アプリケーションファイアウォール**] > [**プロファイル] に移動し、[追加**]** をクリックします。
2. [アプリケーションファイアウォールプロファイルの作成] ダイアログボックスの [プロファイル名] フィールドに shopping_cart と入力します。
3. [プロファイルの種類] ドロップダウンリストで、[Web アプリケーション] を選択します。
4.「設定」で、「詳細設定」のデフォルト設定を選択します。
5. [作成] をクリックしてから、[閉じる] をクリックします。
6. 詳細ビューで、新しいプロファイルをダブルクリックします。
7. [Web アプリケーションプロファイルの構成] ダイアログボックスで、次の説明に従って新しいプロファイルを構成します。
    - [チェック] タブをクリックし、[開始 URL チェック] をダブルクリックし、[開始 URL チェックの変更] ダイアログボックスで [全般] タブをクリックしてブロックを無効にし、ラーニング、ロギング、統計、および URL 閉鎖を有効にします。[OK] をクリックし、[閉じる] をクリックします。

コマンドラインを使用している場合は、プロンプトで次のように入力し、Enter キーを押して、これらの設定を構成します。

```
set appfw profile shopping_cart -startURLAction LEARN LOG STATS -
startURLClosure ON
```

- Cookie の一貫性チェックとフォームフィールドの一貫性チェックでは、ブロックを無効にし、ラーニング、ロギング、統計を有効にします。開始の URL チェックの設定と同様の方法を使用します。

  コマンドラインを使用している場合は、次のコマンドを入力してこれらの設定を構成します。

  ```
  set appfw profile shopping_cart -cookieConsistencyAction LEARN LOG
  STATS

  set appfw profile shopping_cart -fieldConsistencyAction LEARN LOG
  STATS
  ```

- SQL インジェクション・チェックでは、「SQL インジェクション・チェックの変更」ダイアログ・ボックスの「全般」タブの「チェック・アクション」セクションで、ブロックを無効にし、特殊文字のラーニング、ロギング、統計および変換を有効にします。

  コマンドラインを使用している場合は、プロンプトで次のように入力し、Enter キーを押して、これらの設定を構成します。

  ```
  set appfw profile shopping_cart -SQLInjectionAction LEARN LOG STATS
   -SQLInjectionTransformSpecialChars ON
  ```

- クレジット・カード・チェックでは、ブロックを無効にし、クレジット・カード番号のロギング、統計、マスキングを有効にし、支払方法として受け入れるクレジット・カードの保護を有効にします。

  - 構成ユーティリティを使用している場合は、[クレジットカードチェックの変更] ダイアログボックスの [全般] タブの [チェックアクション] セクションで、ブロック、ログ、統計、およびマスキング (または x-out) を構成します。特定のクレジットカードの保護は、同じダイアログボックスの [設定] タブで構成します。
  - コマンドラインを使用している場合は、プロンプトで次のように入力し、Enter キーを押して、これらの設定を構成します。

  ```
  set appfw profile shopping_cart -creditCardAction LOG STATS -
  creditCardXOut ON -creditCard <name> [<name>...]
  ```

  <name> あなたが保護したいクレジットカードの名前を置き換えるために. ビザの場合は、VISA を代用します。マスターカードの場合は、マスターカードの代わりになります。アメリカン・エキスプレスの場合は、アメックスに置き換えます。「検出」の場合は、「検出」に置き換えます。ダイナースクラブは、ダイナースクラブに代わるものです。JCB の場合は、JCB に置き換えます。

8. ショッピングカートアプリケーションへの接続を検出し、shopping_cart プロファイルをそれらの接続に適用する shopping_cart という名前のポリシーを作成します。

ショッピングカートへの接続を検出するには、着信接続の URL を調べます。ショッピングカートのアプリケーションを別のホストでホストする場合（セキュリティやその他の理由による賢明な措置）、URL でそのホストが存在することを確認することができます。他のトラフィックを処理するホスト上のディレクトリでショッピングカートをホストする場合は、接続が適切なディレクトリまたは HTML ページに移動していることを確認する必要があります。

これらのいずれかを検出するプロセスは同じです。次の式に基づいてポリシーを作成し、<string>の適切なホストまたは URL を置き換えます。

```
1 REQ.HTTP.HEADER URL CONTAINS <string>
2 <!--NeedCopy-->
```

- 構成ユーティリティを使用している場合は、アプリケーション・ファイアウォールの「ポリシー」ページに移動し、「追加...」ボタンをクリックして新しいポリシーを追加します。次に、201 ページ以降の「構成ユーティリティを使用してクラシック式を使用してポリシーを作成するには」で説明するポリシー作成プロセスに従います。
- コマンドラインを使用している場合は、プロンプトで次のコマンドを入力し、Enter キーを押します。

  ```
  add appfw policy shopping_cart ”REQ.HTTP.HEADER URL CONTAINS <
  string>”shopping_cart
  ```

9. 新しいポリシーをグローバルにバインドして有効にします。

このポリシーがショッピングカートへのすべての接続と一致し、別のより一般的なポリシーによってプリエンプトされないようにする必要があるため、高い優先順位を割り当てる必要があります。1 をプライオリティとして割り当てると、他のポリシーはこのポリシーを優先させることはできません。

スクリプト化された **Web** ページを保護するためのアプリケーションファイアウォールポリシー

埋め込まれたスクリプト、特にレガシー JavaScript を含む Web ページは、「同じ生成元ルール」に違反することがよくあります。このルールでは、スクリプトが配置されているサーバー以外のサーバー上のコンテンツにアクセスしたり変更したりすることはできません。このセキュリティ上の脆弱性をクロスサイトスクリプティングと呼びます。アプリケーションファイアウォールのクロスサイトスクリプティング規則は、通常、クロスサイトスクリプティングを含むリクエストを除外します。

残念ながら、システム管理者がこれらのスクリプトをチェックし、安全であることがわかっていても、古い JavaScript を持つ Web ページが機能しなくなる可能性があります。次の例では、他の Web サイトに対してこの重要なフィルタを無効にせずに、信頼できるソースからの Web ページのクロスサイトスクリプティングを許可するようにアプリケーションファイアウォールを構成する方法について説明します。

コマンドラインインターフェイスを使用して **Web** ページをクロスサイトスクリプティングで保護するには

- コマンドラインで詳細プロファイルを作成するには、次のように入力します。

```
add appfw profile pr_xssokay -defaults advanced
```

- プロファイルを構成するには、次のように入力します。

```
set appfw profile pr_xssokay -startURLAction NONE -startURLClosure OFF
-cookieConsistencyAction LEARN LOG STATS -fieldConsistencyAction LEARN
LOG STATS -crossSiteScriptingAction LEARN LOG STATS$”
```

- スクリプト化された Web ページへの接続を検出し、pr_xssOK プロファイルを適用するポリシーを作成します。次のように入力します。

```
add appfw policy pol_xssokay ”REQ.HTTP.HEADER URL CONTAINS ^\\.pl\\?$
|| REQ.HTTP.HEADER URL CONTAINS ^\\.js$”pr_xssokay
```

- ポリシーをグローバルにバインドします。

構成ユーティリティを使用して **Web** ページをクロスサイトスクリプトで保護するには

1. [セキュリティ] **>** [アプリケーションファイアウォール] **>** [プロファイル] に移動します。
2. 詳細ビューで、[追加] をクリックします。
3. [アプリケーションファイアウォールプロファイルの作成] ダイアログボックスで、詳細な既定で Web アプリケーションプロファイルを作成し、pr_xssOK という名前を付けます。[作成] をクリックしてから、[閉じる] をクリックします。
4. 詳細ビューで、プロファイルをクリックし、[開く] をクリックし、[Web アプリケーションプロファイルの構成] ダイアログボックスで、次に示すように pr_xssOK プロファイルを構成します。

   URL チェックを開始: すべてのアクションをクリアします。

   - Cookie の一貫性チェック: ブロックを無効にします。
   - フォームフィールドの一貫性チェック: ブロックを無効にします。
   - クロスサイトスクリプティングチェック: ブロックを無効にします。

   これにより、安全であることがわかっているクロスサイトスクリプティングを含む Web ページを含む正当な要求がブロックされるのを防ぐことができます。
5. [ポリシー] をクリックし、[追加] をクリックします。
6. [アプリケーションファイアウォールポリシーの作成] ダイアログボックスで、スクリプト化された Web ページへの接続を検出し、pr_xssOK プロファイルを適用するポリシーを作成します。

   - ポリシー名: pol_xssOK
   - 関連付けられたプロファイル:pr_xssOK

Policy expression: “REQ.HTTP.HEADER URL CONTAINS ^\.pl\?$	REQ.HTTP.HEADER URL CONTAINS ^\.js$”

7. 新しいポリシーをグローバルにバインドして有効にします。

特定の **IP** からのパケットをドロップする **DNS** ポリシー

次の例では、DDOS 攻撃で使用されるような不要な IP またはネットワークからの接続を検出し、それらの場所からのすべてのパケットをドロップする DNS アクションおよび DNS ポリシーを作成する方法について説明します。この例では、IANA 予約済み IP ブロック 192.168.0.0/16 内のネットワークを示しています。敵対的なネットワークは、通常、パブリックにルーティング可能な IP 上にあります。

コマンドラインインターフェイスを使用して特定の **IP** からのパケットをドロップするには

- 敵対的なネットワークからの接続を検出し、それらのパケットをドロップする pol_dos_drop という名前の DNS ポリシーを作成するには、次のように入力します。

  ```
  add dns policy pol_ddos_drop 'client.ip.src.in_subnet(192.168.253.128/25) || client.ip.src.in_subnet(192.168.254.32/27)'-drop YES'
  ```

  192.168.0.0/16 の範囲のネットワーク例では、ブロックする各ネットワークの ##.##.##.##.##.###/## 形式の IP とネットマスクを置き換えます。各 CLIENT.IP.SRC.IN_SUBNET（##.##.##.##.##./##）コマンドを OR 演算子で区切って、必要な数のネットワークを含めることができます。

- 新しいポリシーをグローバルにバインドして有効にします。

有効なクライアント証明書を要求する **SSL** ポリシー

次の例は、クライアントとの SSL 接続を開始する前に、ユーザーのクライアント証明書の有効性をチェックする SSL ポリシーを示しています。

期限切れのクライアント証明書を持つユーザーからの接続をブロックするには

- コマンドラインインターフェイスにログオンします。

  GUI を使用している場合は、[SSL ポリシー] ページに移動し、[データ] 領域で [アクション] タブをクリックします。

- act_current_client_cert という名前の SSL アクションを作成します。このアクションでは、Citrix ADC との SSL 接続を確立するためにユーザーが現在のクライアント証明書を持っている必要があります。


```
add ssl action act_current_client_cert-clientAuth DOCLIENTAUTH -clientCert
 ENABLED -certHeader ”clientCertificateHeader”-clientCertNotBefore
ENABLED -certNotBeforeHeader ”Mon, 01 Jan 2007 00:00:00 GMT”
```

- クエリ文字列を含む Web サーバーへの接続を検出する pol_current_client_cert という名前の SSL ポリシーを作成します。

  ```
  add ssl policy pol_current_ client_cert 'REQ.SSL.CLIENT.CERT.VALIDFROM
  \>= ”Mon, 01 Jan 2007 00:00:00 GMT”'act_block_ssl
  ```

- 新しいポリシーをグローバルにバインドします。

  この SSL ポリシーは、より具体的な SSL ポリシーが適用されない限り、すべてのユーザーの SSL 接続に適用する必要があるため、低優先度を割り当てることができます。1,000（1000）のプライオリティを割り当てると、他の SSL ポリシーが最初に評価されます。つまり、このポリシーは、より具体的なポリシー基準に一致しない接続にのみ適用されます。

Apache mod_rewrite ルールのデフォルト構文への移行

October 7, 2021

Apache HTTP サーバは、HTTP リクエスト URL を書き換えるための mod_rewrite と呼ばれるエンジンを提供します。mod_rewrite ルールを Apache から Citrix ADC に移行すると、バックエンドサーバーのパフォーマンスが向上します。さらに、Citrix ADC は通常、複数の Web サーバー（時には数千もの）の負荷分散を行うため、ルールを Citrix ADC に移行した後、これらのルールを一元的に制御できます。

次に、mod_rewrite 関数の例と、Citrix ADC 上のリライトポリシーとレスポンダーポリシーへのこれらの関数の変換を示します。

URL のバリエーションを正規の URL に変換する

一部の Web サーバーでは、1 つのリソースに対して複数の URL を持つことができます。正規の URL を使用および配布する必要がありますが、他の URL はショートカットまたは内部 URL として存在する可能性があります。最初のリクエストに使用された URL に関係なく、正規の URL がユーザーに表示されるようにすることができます。

次の例では、URL /~user を /u/user に変換しています。

URL を変換するための Apache の mod_rewrite ソリューション

```
1 RewriteRule ^/~([^/]+)/?(.*) /u/$1/$2[R]
2 <!--NeedCopy-->
```


URL を変換するための **Citrix ADC** ソリューション

```
add responder action act1 redirect '"/u/"+HTTP.REQ.URL.AFTER_STR("/~")'
    -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.STARTSWITH("/~") && HTTP.REQ.
   URL.LENGTH.GT(2)' act1
bind responder global pol1 100
<!--NeedCopy-->
```

ホスト名のバリエーションを正規ホスト名に変換する

サイトにアクセスするために特定のホスト名の使用を強制することができます。たとえば、example.com の代わりに www.example.com の使用を強制することができます。

80 以外のポートで実行されているサイトに特定のホスト名を適用するための **Apache mod_rewrite** ソリューション

```
RewriteCond %{
 HTTP_HOST }
    !^www.example.com
RewriteCond %{
 HTTP_HOST }
    !^$
RewriteCond %{
 SERVER_PORT }
  !^80$
RewriteRule ^/(.*)         http://www.example.com:%{
 SERVER_PORT }
 /$1 [L,R]
<!--NeedCopy-->
```

ポート **80** で実行されているサイトに特定のホスト名を適用するための **Apache mod_rewrite** ソリューション

```
RewriteCond %{
 HTTP_HOST }
    !^www.example.com
RewriteCond %{
 HTTP_HOST }
    !^$
```



```
RewriteRule ^/(.*)          http://www.example.com/$1 [L,R]
<!--NeedCopy-->
```

80 以外のポートで実行されているサイトに特定のホスト名を適用する **Citrix ADC** ソリューション

```
add responder action act1 redirect '"http://www.example.com:"+CLIENT.
    TCP.DSTPORT+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com
    ")&&!HTTP.REQ.HOSTNAME.EQ("")&&!HTTP.REQ.HOSTNAME.PORT.EQ(80)&&HTTP.
    REQ.HOSTNAME.CONTAINS("example.com")' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

ポート **80** で実行されているサイトに特定のホスト名を適用するための **Citrix ADC** ソリューション

```
add responder action act1 redirect '"http://www.example.com"+HTTP.REQ.
    URL' -bypassSafetyCheck yes
add responder policy pol1  '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.
    com")&&!HTTP.REQ.HOSTNAME.EQ("")&&HTTP.REQ.HOSTNAME.PORT.EQ(80)&&
    HTTP.REQ.HOSTNAME.CONTAINS("example.com")' act1
bind responder global  pol1 100 END
<!--NeedCopy-->
```

ドキュメントルートの移動

通常、Web サーバーのドキュメントルートは、URL「/」に基づいています。ただし、ドキュメントルートには任意のディレクトリを指定できます。トラフィックが最上位の「/」ディレクトリから別のディレクトリに変更された場合は、ドキュメントルートにリダイレクトできます。

次の例では、ドキュメントのルートを/から /e/www に変更します。最初の 2 つの例は、ある文字列を別の文字列に置き換えるだけです。3 番目の例は、ルートディレクトリを置き換えるとともに、URL の残りの部分（パスとクエリ文字列）を保持するため、たとえば /example/file.html を /e/www/example/file.html にリダイレクトします。

ドキュメントルートを移動するための **Apache** の **mod_rewrite** ソリューション

```
RewriteEngine on
RewriteRule   ^/$  /e/www/  [R]
<!--NeedCopy-->
```


ドキュメントルートを移動するための **Citrix ADC** ソリューション

```
add responder action act1 redirect '"/e/www/"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.EQ("/")' act1
bind responder global pol1 100
<!--NeedCopy-->
```

ドキュメントルートを移動し、リクエストにパス情報を追加するための **Citrix ADC** ソリューション

```
add responder action act1 redirect '"/e/www"+HTTP.REQ.URL' -
    bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.URL.STARTSWITH("/e/www/")' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

ホームディレクトリを新しい **Web** サーバーに移動する

Web サーバー上のホームディレクトリに送信された要求を別の Web サーバーにリダイレクトできます。たとえば、新しい Web サーバーが古いサーバーを置き換える場合、ホームディレクトリを新しい場所に移行するときに、移行したホームディレクトリに対する要求を新しい Web サーバーにリダイレクトする必要があります。

次の例では、新しい Web サーバーのホスト名は newserver です。

別の **Web** サーバーにリダイレクトするための **Apache** の **mod_rewrite** ソリューション

```
RewriteRule    ^/(.+)  http://newserver/$1     [R,L]
<!--NeedCopy-->
```

別の **Web** サーバーにリダイレクトするための **Citrix ADC** ソリューション（方法 **1**）

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -
    bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.REGEX_MATCH(re#^/(.+)#)'   act1
bind responder global pol1 100 END
<!--NeedCopy-->
```


別の **Web** サーバーにリダイレクトするための **Citrix ADC** ソリューション（方法 **2**）

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -
    bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.LENGTH.GT(1)' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

構造化されたホームディレクトリの操作

通常、数千人のユーザーがいるサイトには、構造化されたホームディレクトリレイアウトがあります。たとえば、各ホームディレクトリは、ユーザー名の最初の文字を使用して名前が付けられたサブディレクトリの下に配置できます。たとえば、jsmith (/~jsmith/anypath) のホームディレクトリは /home/j/smith/.www/anypath であり、rvalveti (/~rvalveti/anypath) のホームディレクトリは /home/r/r/rvalveti/.www/anypath です。

次の例では、要求をホームディレクトリにリダイレクトします。

構造化されたホームディレクトリ用の **Apache** の **mod_rewrite** ソリューション

```
RewriteRule    ^/~(([a-z])[a-z0-9]+)(.*)   /home/$2/$1/.www$3
<!--NeedCopy-->
```

構造化されたホームディレクトリ用の **Citrix ADC** ソリューション

構造化されたホームディレクトリ用の Citrix ADC ソリューション

```
add rewrite action act1 replace 'HTTP.REQ.URL'  '"/home/"+ HTTP.REQ.URL
    .AFTER_STR("~").PREFIX(1)+"/"+ HTTP.REQ.URL.AFTER_STR("~").
    BEFORE_STR("/")+"/.www"+HTTP.REQ.URL.SKIP('/',1)'  -
    bypassSafetyCheck yes
add rewrite policy pol1  'HTTP.REQ.URL.PATH.STARTSWITH("/~")' act1
bind rewrite global pol1 100

<!--NeedCopy-->
```

無効な **URL** を他の **Web** サーバーにリダイレクトする

URL が有効でない場合は、別の Web サーバーにリダイレクトする必要があります。たとえば、URL で指定されたファイルが URL で指定されたサーバー上に存在しない場合は、別の Web サーバーにリダイレクトする必要がありま

す。

Apache では、mod_rewrite を使用してこのチェックを実行できます。Citrix ADC では、HTTP コールアウトは、サーバー上のスクリプトを実行して、サーバー上のファイルをチェックできます。次の Citrix ADC 例では、file_check.cgi という名前のスクリプトが URL を処理し、この情報を使用してサーバー上にターゲットファイルが存在するかどうかをチェックします。スクリプトは TRUE または FALSE を返し、Citrix ADC はスクリプトが返す値を使用してポリシーを検証します。

リダイレクトの実行に加えて、Citrix ADC はカスタムヘッダーを追加することも、2 番目の Citrix ADC の例のように、レスポンス本文にテキストを追加することもできます。

URL が間違っている場合のリダイレクトのための **Apache** の **mod_rewrite** ソリューション

```
RewriteCond   /your/docroot/%{
 REQUEST_FILENAME }
  !-f
RewriteRule   ^(.+)      http://webserverB.com/$1 [R]

<!--NeedCopy-->
```

URL が間違っている場合のリダイレクトのための **Citrix ADC** ソリューション（方法 **1**）

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr
     '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).
    CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"'   -
    parameters query=http.req.url.path -headers Name("ddd")
add responder action act1 redirect '"http://webserverB.com"+HTTP.REQ.
    URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.
    HTTP_CALLOUT(call)' act1
bind responder global pol1 100

<!--NeedCopy-->
```

URL が間違っている場合のリダイレクトのための **Citrix ADC** ソリューション（方法 **2**）

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr
     '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).
```



```
    CONTAINS(”True”)'  -urlStemExpr '”/cgi-bin/file_check.cgi”'   -
    parameters query=http.req.url.path -headers Name(”ddd”)
3 add responder  action act1 respondwith  '”HTTP/1.1 302 Moved
    Temporarily\r\nLocation: http://webserverB.com”+HTTP.REQ.URL+”\r\n\r
    \nHTTPCallout Used”' -bypassSafetyCheck yes
4 add responder policy pol1 '!HTTP.REQ.HEADER(”Name”).EXISTS  &&  !SYS.
    HTTP_CALLOUT(call)' act1
5 bind responder global pol1 100
6
7 <!--NeedCopy-->
```

時間に基づいて **URL** を書き換え

時間に基づいて URL を書き換えることができます。次の例では、時間帯に応じて、example.html のリクエストを example.day.html または example.night.html に変更します。

時間に基づいて **URL** を書き換えるための **Apache** の **mod_rewrite** ソリューション

```
1  RewriteCond   %{
2   TIME_HOUR }
3   %{
4   TIME_MIN }
5    >0700
6  RewriteCond   %{
7   TIME_HOUR }
8   %{
9   TIME_MIN }
10   <1900
11 RewriteRule   ^example.html$ example.day.html [L]
12 RewriteRule   ^example.html$ example.night.html
13
14 <!--NeedCopy-->
```

時間に基づいて **URL** を書き換えるための **Citrix ADC** ソリューション

```
1 add rewrite action act1 insert_before 'HTTP.REQ.URL.PATH.SUFFIX('.',0)'
     '”day.”'
2 add rewrite action act2  insert_before 'HTTP.REQ.URL.PATH.SUFFIX('.',0)
    ' '”night.”'
3 add rewrite  policy pol1 'SYS.TIME.WITHIN(LOCAL 07h 00m,LOCAL 18h 59m)'
     act1
```



```
add rewrite policy pol2 'true'  act2
bind rewrite global pol1 101
bind rewrite global pol2 102

<!--NeedCopy-->
```

新しいファイル名へのリダイレクト（ユーザーには見えない）

Web ページの名前を変更した場合、下位互換性のために古い URL を引き続きサポートし、ページの名前が変更されたことをユーザーが認識できないようにすることができます。

次の例の最初の 2 つのベースディレクトリは /~quux/ です。3 番目の例では、任意のベースディレクトリと、URL 内のクエリ文字列が存在しています。

固定された場所でファイル名の変更を管理するための **Apache mod_rewrite** ソリューション

```
RewriteEngine  on
RewriteBase    /~quux/
RewriteRule    ^foo.html$  bar.html

<!--NeedCopy-->
```

特定の場所でファイル名の変更を管理するための **Citrix ADC** ソリューション

```
add rewrite action act1 replace 'HTTP.REQ.URL.AFTER_STR("/~quux").
   SUBSTR("foo.html")' '"bar.html"'
add rewrite policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/foo.html")' act1
bind rewrite global pol1 100

<!--NeedCopy-->
```

URL のベースディレクトリまたはクエリ文字列に関係なく、ファイル名の変更を管理するための **Citrix ADC** ソリューション

```
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX('/',0)' '"bar
   .html"'
Add rewrite policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("foo.html")' act1
Bind rewrite global pol1 100

<!--NeedCopy-->
```


新しいファイル名へのリダイレクト **(**ユーザー表示の **URL)**

Web ページの名前を変更する場合、下位互換性を保つために古い URL を引き続きサポートし、ブラウザに表示される URL を変更することでページの名前が変更されたことをユーザーに確認できるようにすることができます。

次の例の最初の 2 つでは、ベースディレクトリが /~quux/ のときにリダイレクトが行われます。3 番目の例では、任意のベースディレクトリと、URL 内のクエリ文字列が存在しています。

ブラウザに表示されるファイル名と **URL** を変更するための **Apache** の **mod_rewrite** ソリューション

```
RewriteEngine on
RewriteBase     /~quux/
RewriteRule     ^old.html$ new.html   [R]

<!--NeedCopy-->
```

ブラウザに表示されるファイル名と **URL** を変更するための **Citrix ADC** ソリューション

```
add responder action act1 redirect 'HTTP.REQ.URL.BEFORE_STR("foo.html")
   +"new.html"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/old.html")'
   act1
bind responder global pol1 100

<!--NeedCopy-->
```

URL のベースディレクトリやクエリ文字列に関係なく、ブラウザに表示されるファイル名と **URL** を変更するための **Citrix ADC** ソリューション

```
add responder action act1 redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("old.
   html")+"new.html"+HTTP.REQ.URL.AFTER_STR("old.html")' -
   bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("old.html")' act1
bind responder global pol1 100

<!--NeedCopy-->
```


ブラウザに依存するコンテンツの収容

ブラウザ固有の制限（少なくとも重要なトップレベルページの場合）に対応するために、ブラウザの種類とバージョンに制限を設定する必要がある場合があります。たとえば、最新の Netscape バリアントには最大バージョン、Lynx ブラウザには最小バージョン、その他すべてのバージョンには平均機能バージョンを設定できます。

次の例では、HTTP ヘッダー「ユーザーエージェント」に対して動作し、このヘッダーが「Mozilla/3」で始まる場合、ページ MyPage.html が MyPage.NS.html に書き換えられます。ブラウザが「リンクス」または「Mozilla」バージョン 1 または 2 の場合、URL は MyPage.20.html になります。他のすべてのブラウザは、ページ MyPage.32.html を受け取ります。

ブラウザ固有の設定のための **Apache** の **mod_rewrite** ソリューション

```
RewriteCond %{
 HTTP_USER_AGENT }
   ^Mozilla/3.*
RewriteRule ^MyPage.html$ MyPage.NS.html [L]
RewriteCond %{
 HTTP_USER_AGENT }
   ^Lynx/.* [OR]
RewriteCond %{
 HTTP_USER_AGENT }
   ^Mozilla/[12].*
RewriteRule ^MyPage.html$ MyPage.20.html [L]
RewriteRule ^fMyPage.html$ MyPage.32.html [L]
Citrix ADC solution for browser-specific settings
add patset pat1
bind patset pat1 Mozilla/1
bind Patset pat1 Mozilla/2
bind patset pat1 Lynx
bind Patset pat1 Mozilla/3
add rewrite action act1 insert_before 'HTTP.REQ.URL.SUFFIX' '"NS."'
add rewrite action act2 insert_before 'HTTP.REQ.URL.SUFFIX' '"20."'
add rewrite action act3 insert_before 'HTTP.REQ.URL.SUFFIX' '"32."'
add rewrite policy pol1 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX
   ("pat1").EQ(4)' act1
add rewrite policy pol2 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX
   ("pat1").BETWEEN(1,3)' act2
add rewrite policy pol3 '!HTTP.REQ.HEADER("User-Agent").STARTSWITH_ANY
   ("pat1")' act3
bind rewrite global pol1 101 END
bind rewrite global pol2 102 END
bind rewrite global pol3 103 END

```



```
<!--NeedCopy-->
```

ロボットによるアクセスの遮断

ロボットが特定のディレクトリまたは一連のディレクトリからページを取得することをブロックして、これらのディレクトリとの間のトラフィックを容易にすることができます。特定の場所に基づいてアクセスを制限することも、User-Agent HTTP ヘッダーの情報に基づいて要求をブロックすることもできます。

次の例では、ブロックされる Web の場所は /~quux/foo/arc/ で、ブロックされる IP アドレスは 123.45.67.8 と 123.45.67.9 で、ロボットの名前は NameOfBadRobot です。

パスとユーザーエージェントヘッダーをブロックするための **Apache mod_rewrite** ソリューション

```
RewriteCond %{
 HTTP_USER_AGENT }
    ^NameOfBadRobot.*
RewriteCond %{
 REMOTE_ADDR }
        ^123.45.67.[8-9]$
RewriteRule ^/~quux/foo/arc/.+   -   [F]

<!--NeedCopy-->
```

パスとユーザーエージェントヘッダーをブロックするための **Citrix ADC** ソリューション

```
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"
    '
add responder policy pol1 'HTTP.REQ.HEADER("User_Agent").STARTSWITH("
    NameOfBadRobot")&&CLIENT.IP.SRC.EQ(123.45.67.8)&&CLIENT.IP.SRC.EQ
    (123.45.67.9) && HTTP.REQ.URL.STARTSWITH("/~quux/foo/arc")' act1
bind responder global pol1 100

<!--NeedCopy-->
```

インライン画像へのアクセスをブロックする

ユーザーが頻繁に自分の使用のためにインライングラフィックをコピーするためにサーバーに行くと（そして不要なトラフィックを生成する）場合は、HTTP Referer ヘッダーを送信するブラウザーの機能を制限することができます。

次の例では、グラフィックスは Exampleにあります。

インラインイメージへのアクセスをブロックするための **Apache mod_rewrite** ソリューション

```
RewriteCond %{
 HTTP_REFERER }
  !^$
RewriteCond %{
 HTTP_REFERER }
  !^http://www.quux-corp.de/~quux/.\*$
RewriteRule .\*.png$ - [F]

<!--NeedCopy-->
```

インラインイメージへのアクセスをブロックするための **Citrix ADC** ソリューション

```
add patset pat1
bind patset pat1 .png
bind patset pat1 .jpeg
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"
    '
add responder policy pol1 '!HTTP.REQ.HEADER("Referer").EQ("") && !HTTP.
    REQ.HEADER("Referer").STARTSWITH("http://www.quux-corp.de/~quux/")&&
    HTTP.REQ.URL.ENDSWITH_ANY("pat1")' act1
bind responder global pol1 100

<!--NeedCopy-->
```

延長なしリンクの作成

ユーザーがサーバー側でアプリケーションまたはスクリプトの詳細を知らないようにするには、ファイル拡張子をユーザーに表示しないようにします。これを行うには、拡張なしのリンクをサポートしたい場合があります。この動作は、書き換えルールを使用してすべての要求に拡張機能を追加したり、要求に拡張機能を選択的に追加したりすることで実現できます。

次の例の最初の 2 つは、すべてのリクエスト URL に拡張機能を追加することを示しています。最後の例では、2 つのファイル拡張子のうちの 1 つが追加されます。最後の例では、mod_rewrite モジュールはファイル拡張子を簡単に見つけることができます。これは、このモジュールが Web サーバー上に存在するためです。これとは対照的に、Citrix ADC は HTTP コールアウトを呼び出して、Web サーバー上の要求されたファイルの拡張子をチェックする必要があります。このコールアウト応答に基づいて、Citrix ADC は要求 URL に.html または.php 拡張子を追加します。

> 注
>
> 2 番目の Citrix ADC 例では、HTTP コールアウトを使用して、サーバーでホストされている file_check.cgi という名前のスクリプトを照会します。このスクリプトは、コールアウトで指定された引数が有効なファイル名であるかどうかをチェックします。

すべてのリクエストに**.php** 拡張子を追加するための **Apache** の **mod_rewrite** ソリューション

```
RewriteRule ^/?([a-z]+)$ $1.php [L]

<!--NeedCopy-->
```

すべてのリクエストに**.php** 拡張子を追加するための **Citrix ADC** ポリシー

```
add rewrite action act1 insert_after 'HTTP.REQ.URL' '".php"'
add rewrite policy pol1 'HTTP.REQ.URL.PATH.REGEX_MATCH(re#^/([a-z]+)$#)
    ' act1
bind rewrite global pol1 100
<!--NeedCopy-->
```

.html または**.php** の拡張子をリクエストに追加するための **Apache** の **mod_rewrite** ソリューション

```
RewriteCond %{
 REQUEST_FILENAME }
 .php -f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.php [L]
RewriteCond %{
 REQUEST_FILENAME }
 .html - f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.html [L]
<!--NeedCopy-->
```

.html または**.php** の拡張子をリクエストに追加するための **Citrix ADC** ポリシー

```
add HTTPCallout Call_html
add HTTPCallout Call_php
set policy httpCallout Call_html -IPAddress 10.102.59.101 -port 80 -
    hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.
```



```
    BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"
    '   -parameters query=http.req.url+".html"
4 set policy httpCallout Call_php -IPAddress 10.102.59.101 -port 80 -
    hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.
    BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"
    ' -parameters query=http.req.url+".php"
5 add patset pat1
6 bind patset pat1 .html
7 bind patset pat1 .php
8 bind patset pat1 .asp
9 bind patset pat1 .cgi
10 add rewrite  action act1 insert_after 'HTTP.REQ.URL.PATH'  '".html"'
11 add rewrite  action act2 insert_after "HTTP.REQ.URL.PATH"  '".php"'
12 add rewrite policy pol1 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.
    HTTP_CALLOUT(Call_html)' act1
13 add rewrite policy pol2 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.
    HTTP_CALLOUT(Call_php)' act2
14 bind rewrite global pol1 100 END
15 bind rewrite global pol2 101 END
16
17 <!--NeedCopy-->
```

作業 **URI** を新しい形式にリダイレクトする

次のような作業中の URL のセットがあるとします。

```
1 /index.php?id=nnnn
2
3 <!--NeedCopy-->
```

これらの URL を /nnnn に変更し、検索エンジンがインデックスを新しい URI 形式に更新するようにするには、次の操作を行う必要があります。

- 検索エンジンがインデックスを更新するように、古い URI を新しい URI にリダイレクトします。
- 新しい URI を古い URI に書き直して、index.php スクリプトが正しく実行されるようにします。

これを行うには、クエリ文字列にマーカーコードを挿入し（マーカーコードが訪問者に表示されないようにする）、index.php スクリプトのマーカーコードを削除します。

次の例では、クエリ文字列にマーカーが存在しない場合にのみ、古いリンクから新しい形式にリダイレクトします。新しい形式を使用するリンクが古い形式に再書き込まれ、クエリ文字列にマーカーが追加されます。

Apache mod_rewrite solution

```
RewriteCond %{
 QUERY_STRING }
  !marker
RewriteCond %{
 QUERY_STRING }
  id=([-a-zA-Z0-9_+]+)
RewriteRule ^/?index.php$ %1? [R,L]
RewriteRule ^/?([-a-zA-Z0-9_+]+)$  index.php?marker&id=$1 [L]
Citrix ADC solution
add responder action act_redirect redirect 'HTTP.REQ.URL.PATH.
    BEFORE_STR(”index.php”)+HTTP.REQ.URL.QUERY.VALUE(”id”)' -
    bypassSafetyCheck yes
add responder policy pol_redirect '!HTTP.REQ.URL.QUERY.CONTAINS(”marker
    ”)&& HTTP.REQ.URL.QUERY.VALUE(”id”).REGEX_MATCH(re/[-a-zA-Z0-9_+]+/)
     && HTTP.REQ.URL.PATH.CONTAINS(”index.php”)' act_redirect
bind responder global pol_redirect 100 END
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX('/',0)' '”
    index.phpmarker&id=”+HTTP.REQ.URL.PATH.SUFFIX('/',0)' -
    bypassSafetyCheck yes
add rewrite policy pol1 '!HTTP.REQ.URL.QUERY.CONTAINS(”marker”)'  act1
bind rewrite global pol1 100 END

<!--NeedCopy-->
```

選択したページにセキュアサーバが使用されていることの確認

選択した Web ページにセキュリティで保護されたサーバーのみが使用されるようにするには、次の Apache mod_rewrite コードまたは Citrix ADC レスポンダーポリシーを使用できます。

Apache mod_rewrite solution

```
RewriteCond %{
 SERVER_PORT }
  !^443$
RewriteRule ^/?(page1|page2|page3|page4|page5)$  https://www.example.
    com/%1 [R,L]

<!--NeedCopy-->
```


正規表現を使用した **Citrix ADC** ソリューション

```
add responder action res_redirect redirect  '"https://www.example.com"+
    HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.
    REQ.URL.REGEX_MATCH(re/page[1-5]/)'  res_redirect
bind responder global pol_redirect 100 END

<!--NeedCopy-->
```

パターンセットを使用した **Citrix ADC** ソリューション

```
add patset pat1
bind patset pat1 page1
bind patset pat1 page2
bind patset pat1 page3
bind patset pat1 page4
bind patset pat1 page5
add responder action res_redirect redirect  '"https://www.example.com"+
    HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.
    REQ.URL.CONTAINS_ANY("pat1")'  res_redirect
bind responder global pol_redirect 100 END

<!--NeedCopy-->
```

リライトとレスポンダーポリシーの例

October 7, 2021

書き換えポリシーとレスポンダーポリシーの例をいくつか示します。

例 **1:** コマンドラインインターフェイスを使用してローカル **Client-IP** ヘッダーを追加するには

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.
    IP.SRC'
add rewrite policy pol_ins_client http.req.is_valid act_ins_client
bind rewrite global pol_ins_client 300 END

```



```
namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:06:48 GMT
* Server Apache/2.2.15 (CentOS) is not blacklisted
< Server: Apache/2.2.15 (CentOS)
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: "816c5-5-58bbc1e73cdd3"
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
< NS-Client: 10.102.1.98
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$

<!--NeedCopy-->
```

例 **2: HTTP** サーバータイプをマスクする

```
add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server") ""Web Server 1.0""
add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:15:42 GMT
* Server Web Server 1.0 is not blacklisted
```



```
< Server: Web Server 1.0
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: ”816c5-5-58bbc1e73cdd3”
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$
<!--NeedCopy-->
```

例 **3: URL** が受信されたときに別の **URL** にリダイレクトして応答する

```
> add responder action act1 redirect ””www.google.com””
 Done
> add responder policy pol1 'HTTP.REQ.URL.CONTAINS(”file”)' act1
 Done
> bind responder global pol1 1
 Done
>

name:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 302 Found : Moved Temporarily
< Location: www.google.com
< Connection: close
< Cache-Control: no-cache
< Pragma: no-cache
<
* Closing connection 0
name@obelix:~$
<!--NeedCopy-->
```

例 **4:** 任意の式またはテキストにできるメッセージで応答する


```
add responder action act123 respondwith ””Please reach out to
    administrator””
add responder policy pol1 ”HTTP.REQ.URL.CONTAINS(”file”)” act123
bind responder global pol1 100 END

name@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10..Responder Action and Policy:

>add responder action Redirect-Action redirect ””https://xyz.abc.com/
    dispatcher/SAML2AuthService?siteurl=wmav”” -responseStatusCode 302

>add responder policy Redirect-Policy ”HTTP.REQ.HOSTNAME.CONTAINS(”abc”
    )” Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -
    gotoPriorityExpression END -type REQUEST.
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
* Connection #0 to host 10.10.10.10 left intact
Please reach out to administratort_name@obelix:~$
<!--NeedCopy-->
```

例 **5: HTML** インポートされたページで応答する

```
import responder htmlpage http://10.10.10.10)/testsite/file5.html
    page112
add responder action act1 respondwithHtmlpage page1
add responder policy pol1 true act1
bind responder global pol1 100

name@obelix:~$ curl -v http://10.10.10.10)/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10)) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
```



```
> User-Agent: curl/7.35.0
> Host: 10.102.58.140
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Content-Length: 5
< Content-Type: text/html
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_name@obelix:~$
<!--NeedCopy-->
```

例 **6:** レスポンダーポリシーを使用したホスト名に基づく **URL** のリダイレクト

```
Responder Action and Policy:

>add responder action Redirect-Action redirect ””https://xyz.abc.com/
    dispatcher/SAML2AuthService?siteurl=wmav”” -responseStatusCode 302

>add responder policy Redirect-Policy ”HTTP.REQ.HOSTNAME.CONTAINS(”abc”
    )” Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

レート制限

October 7, 2021

レート制限機能を使用すると、Citrix ADC アプライアンス上の特定のネットワークエンティティまたは仮想エンティティの最大負荷を定義できます。この機能を使用すると、エンティティに関連付けられたトラフィックのレートを監視し、トラフィックレートに基づいてリアルタイムで予防措置を実行するようにアプライアンスを設定できます。この機能は、アプライアンスに大量の要求を送信している敵対的なクライアントからネットワークが攻撃を受けている場合に特に便利です。クライアントに対するリソースの可用性に影響するリスクを軽減し、ネットワークとアプライアンスが管理するリソースの信頼性を向上させることができます。

仮想サーバー、URL、ドメイン、URL とドメインの組み合わせなど、仮想エンティティとユーザー定義エンティティに関連付けられたトラフィックのレートを監視および制御できます。トラフィックが高すぎる場合はトラフィックレ

ートを調整し、トラフィックレートに基づいて情報キャッシュを設定し、トラフィックレートが事前定義された制限を超えた場合は、特定のロードバランシング仮想サーバにトラフィックをリダイレクトできます。HTTP、TCP、および DNS 要求にレートベースのモニタリングを適用できます。

特定のシナリオのトラフィックレートを監視するには、レート制限 *ID* を設定します。レート制限識別子は、タイムスライスと呼ばれる指定された期間に許可される（特定のタイプの）要求または接続の最大数などの数値のしきい値を指定します。

オプションで、ストリームセレクタと呼ばれるフィルタを設定し、識別子を設定するときにレート制限識別子に関連付けることができます。オプションのストリームセレクタと制限識別子を設定したら、デフォルトの構文ポリシーから制限識別子を呼び出す必要があります。識別子は、書き換え、応答側、DNS、統合キャッシュなど、識別子が有用である任意の機能から呼び出すことができます。

レート制限 ID の SNMP トラップをグローバルに有効または無効にできます。タイムスライスごとに複数のトラップを生成するように指定していない限り、各トラップには、レート制限 ID の設定済みデータ収集間隔（タイムスライス）の累積データが含まれます。SNMP トラップおよびマネージャの設定の詳細については、「SNMP」を参照してください。

ストリームセレクタの設定

October 7, 2021

トラフィックストリームセレクタは、アクセスを抑制するエンティティを識別するためのオプションのフィルタです。セレクタは要求または応答に適用され、レートストリーム識別子によって分析できるデータポイント（キー）を選択します。これらのデータポイントは、IP アドレス、サブネット、ドメイン名、TCP または UDP 識別子、URL 内の特定の文字列や拡張機能など、トラフィックのほぼすべての特性に基づくことができます。

ストリームセレクタは、selectlets と呼ばれる個々のデフォルトの構文式で構成されています。各選択レットは、非複合デフォルト構文式です。トラフィックストリームセレクタには、selectlets と呼ばれる非複合式を 5 つまで含めることができます。各選択レットは、他の式と AND 関係にあるとみなされます。次に、選択レットのいくつかの例を示します。

```
1 http.req.url
2 http.res.body(1000>after_str(”car_model”).before_str(”made_in”)”
3 ”client.ip.src.subnet(24)”
4 <!--NeedCopy-->
```

パラメータを指定する順序は重要です。たとえば、あるセレクタで IP アドレスとドメインを（その順序で）構成し、別のセレクタでドメインと IP アドレスを指定すると、Citrix ADC ではこれらの値が一意であると見なされます。これにより、同じトランザクションが 2 回カウントされる可能性があります。また、複数のポリシーで同じセレクタを呼び出すと、Citrix ADC は同じトランザクションを複数回カウントできます。

注意: ストリームセレクタで式を変更した場合、それを呼び出すポリシーが新しいポリシーラベルまたはバインドポイントにバインドされると、エラーが発生することがあります。たとえば、myStreamSelector1 という名前のストリームセレクタを作成し、myLimitID1 から呼び出し、dnsRateLimit1 という名前の DNS ポリシーから識別子を呼び出すとします。myStreamSelector1 の式を変更すると、dnsRateLimit1 を新しいバインドポイントにバインドするときにエラーが発生することがあります。回避策は、これらの式を呼び出すポリシーを作成する前に、これらの式を変更することです。

コマンドラインインターフェイスを使用してトラフィックストリームセレクタを設定するには

コマンドプロンプトで入力します。

```
add stream selector <name> <rule> ...
<!--NeedCopy-->
```

例:

```
add stream selector myStreamSel HTTP.REQ.URL CLIENT.IP.SRC
<!--NeedCopy-->
```

構成ユーティリティを使用してストリームセレクタを構成するには

[AppExpert] > [レート制限] > [セレクタ] に移動し、[追加] をクリックして、関連する詳細を指定します。

トラフィックレート制限識別子の設定

October 7, 2021

レート制限識別子は、特定の時間間隔内にトラフィック量が指定された値を超えたかどうかをチェックします。トラフィック量が特定の時間間隔内に制限を超えた場合、識別子は「ブール値 TRUE」を返します。ポリシー規則の複合デフォルト構文式に制約条件識別子を含める場合は、ストリームセレクタを含める必要があります。を指定しない場合、制限識別子は、複合式によって識別されるすべての要求または応答に適用されます。

> 注:
>
> 文字列の結果を格納するための最大長 (HTTP.REQ.URL など) は 60 文字です。文字列（URL など）の長さが 1000 文字で、そのうちの 50 文字で文字列を一意に識別できる場合は、式を使用して必要な 50 文字を抽出できます。

コマンドラインインターフェイスからトラフィック制限識別子を設定するには

コマンドプロンプトで入力します。

```
add ns limitIdentifier <limitIdentifier> -threshold <positive_integer>
    -timeSlice <positive_integer> -mode <mode> -limitType ( BURSTY |
    SMOOTH ) -selectorName <string> -maxBandwidth <positive_integer> -
    trapsInTimeSlice <positive_integer>
<!--NeedCopy-->
```

引数の説明

limitIdentifier。レート制限識別子の名前。ASCII 文字またはアンダースコア (_) 文字で始まり、ASCII 英数字またはアンダースコア文字のみで構成する必要があります。予約語は使用しないでください。これは必須の引数です。最大長：31

threshold。要求 (モードが REQUEST_RATE として設定) がタイムスライスごとに追跡されるときに、指定されたタイムスライスで許可される要求の最大数。接続（モードが CONNECTION に設定されている）が追跡されるとき、それは通過する接続の総数です。デフォルト値：1 最小値：1 最大値：4294967295

timeSlice。10 の倍数で指定される時間間隔（ミリ秒単位）。要求がしきい値を超えるかどうかをチェックするために追跡されます。この引数は、モードが REQUEST_RATE に設定されている場合にのみ必要です。デフォルト値:1000 最小値:10 最大値:4294967295

mode。追跡するトラフィックのタイプを定義します。

1. REQUEST_RATE。リクエスト/タイムスライスを追跡します。
2. 接続。アクティブなトランザクションを追跡します。

limitType。スムーズまたはバースト要求タイプ。

selectorName。レート制限セレクタの名前。この引数が NULL の場合、レート制限は、仮想サーバーまたは Citrix ADC が受信するすべてのトラフィックに適用されます（制限識別子が仮想サーバーにバインドされているかグローバルにバインドされているかによって異なります）。フィルタはありません。最大長：**31**

最大帯域幅。許可される最大帯域幅（kbps 単位）。最小値:0 最大値:4294967287

例**:**

BURSTY モードでのトラフィックレート制限識別子の設定：

```
add ns limitIdentifier 100_request_limit -threshold 100 -timeSlice 1000
     -mode REQUEST_RATE -limitType BURSTY -selectorName
    limit_100_requests_selector -trapsInTimeSlice 30
<!--NeedCopy-->
```


SMOOTH モードでのトラフィックレート制限識別子の設定:

```
add ns limitidentifier limit_req -mode request_rate -limitType smooth -
    timeslice 1000 -Threshold 2000 -trapsInTimeSlice 200
<!--NeedCopy-->
```

構成ユーティリティを使用してトラフィック制限識別子を構成するには

AppExpert > レート制限 > 制限識別子の順に移動し、[追加] をクリックして、関連する詳細を指定します。

トラフィックレートポリシーの設定とバインド

October 7, 2021

レートベースのアプリケーションの動作を実装するには、適切な Citrix ADC 機能でポリシーを構成します。機能は、デフォルトの構文ポリシーをサポートする必要があります。この機能がトラフィックレートを分析できるようにするには、ポリシー式に次の式プレフィクスが含まれている必要があります。

```
sys.check_limit(<limit_identifier>)
<!--NeedCopy-->
```

ここで、limit_identifier は制限識別子の名前です。

ポリシー式は、少なくとも 2 つのコンポーネントを含む複合式である必要があります。

- レート制限 ID が適用されるトラフィックを識別する式。次に例を示します:

```
    http.req.url.contains(”my_aspx.aspx”).
<!--NeedCopy-->
```

- レート制限識別子を識別する式。たとえば、sys.check_limit (「my_limit_identifier」)。これは、ポリシー式の最後の式である必要があります。

コマンドラインインターフェイスを使用してレートに基づくポリシーを構成するには

コマンドプロンプトで次のコマンドを入力して、レートベースのポリシーを構成し、構成を確認します。


```
add cache|dns|rewrite|responder policy <policy_name> -rule expression
    && sys.check_limit(”<LimitIdentifierName>”) [<feature-specific
    information>]
<!--NeedCopy-->
```

次に、レートベースのポリシー規則の完全な例を示します。この例では、ポリシーに関連付けられているレスポンダーアクション send_direct_url が設定されていることを前提としています。sys.check_limit パラメータは、ポリシー式の最後の要素である必要があります。

```
add responder policy responder_threshold_policy ”http.req.url.contains(
    ”myindex.html”) && sys.check_limit(”my_limit_identifier”)”
    send_direct_url
<!--NeedCopy-->
```

ポリシーをグローバルまたは仮想サーバーにバインドする方法については、「デフォルトの構文ポリシーのバインド」を参照してください。

構成ユーティリティを使用してレートベースのポリシーを構成するには

1. ナビゲーションウィンドウで、ポリシーを構成する機能 (統合キャッシュ、書き換え、レスポンダーなど) を展開し、[ポリシー] をクリックします。
2. 詳細ペインで、[Add] をクリックします。[名前] に、ポリシーの一意の名前を入力します。
3. [式] で、ポリシールールを入力し、式の最終コンポーネントとして sys.check_limit パラメータが含まれていることを確認します。次に例を示します:

   ```
   http.req.url.contains(”my_aspx.aspx”) && sys.check_limit(”
       my_limit_identifier”)
   <!--NeedCopy-->
   ```

4. ポリシーに関する機能固有の情報を入力します。

   たとえば、ポリシーをアクションまたはプロファイルに関連付ける必要がある場合があります。詳細については、機能固有のマニュアルを参照してください。

5. [Create] をクリックしてから、[Close] をクリックします。
6. [保存] をクリックします。

トラフィックレートの表示

October 7, 2021

1 つ以上の仮想サーバを通過するトラフィックがレートベースのポリシーと一致する場合、このトラフィックのレートを表示できます。レート統計情報は、レートベースのポリシーのルールで指定した制限識別子に保持されます。複数のポリシーが同じ制限識別子を使用している場合、特定の制限識別子を使用するすべてのポリシーに対するヒットによって定義されたトラフィックレートを表示できます。

コマンドラインインターフェイスを使用してトラフィックレートを表示するには

コマンドプロンプトで次のコマンドを入力して、トラフィックレートを表示します。

```
show ns limitSessions <limitIdentifier>
<!--NeedCopy-->
```

例:

```
sh limitsession myLimitSession
<!--NeedCopy-->
```

構成ユーティリティを使用してトラフィックレートを表示するには

1. [AppExpert] > [レート制限] > [リミット識別子] に移動します。
2. トラフィックレートを表示する制限識別子を選択します。
3. 「セッションの表示」ボタンをクリックします。1 つ以上の仮想サーバを通過するトラフィックが、この制限識別子を使用するレート制限ポリシーと一致している場合（ヒットがこの識別子に設定されたタイムスライス内にある場合）、[Session Details] ダイアログボックスが表示されます。それ以外の場合は、「セッションが存在しません」というメッセージが表示されます。

レートベースポリシーのテスト

October 7, 2021

レートベースのポリシーをテストするには、レートベースのポリシーがバインドされている任意の仮想サーバにトラフィックを送信できます。

タスクの概要**:** レートベースのポリシーのテスト

1. ストリームセレクタ（オプション）およびレート制限識別子（必須）を設定します。次に例を示します:

```
add stream selector sel_subnet Q.URL ”CLIENT.IP.SRC.SUBNET(24)”
add ns limitIdentifier k_subnet -Threshold 4 -timeSlice 3600 -mode
    REQUEST_RATE  -limittype smooth  -selectorName sel_subnet  -
   trapsInTimeSlice 8
<!--NeedCopy-->
```

2. レート制限 ID を使用するポリシーに関連付けるアクションを設定します。次に例を示します:

```
add responder action resp_redirect redirect ””http://response_site
    .com/””
<!--NeedCopy-->
```

3. sys.check_limit 式プレフィクスを使用してレート制限識別子を呼び出すポリシーを設定します。たとえば、ポリシーでは、次のように、特定のサブネットから到着するすべての要求にレート制限識別子を適用できます。

```
add responder policy resp_subnet ”SYS.CHECK_LIMIT(”k_subnet”)”
   resp_redirect
<!--NeedCopy-->
```

4. ポリシーをグローバルにバインドするか、仮想サーバにバインドします。次に例を示します:

```
bind responder global resp_subnet 6 END -type DEFAULT
<!--NeedCopy-->
```

5. ブラウザのアドレスバーで、テスト HTTP クエリを仮想サーバーに送信します。次に例を示します:

```
http://<IP of a vserver>/testsite/test.txt
<!--NeedCopy-->
```

6. Citrix ADC コマンドプロンプトで、次のように入力します。

```
show ns limitSessions <limitIdentifier>
<!--NeedCopy-->
```


例

```
> sh limitsession k_subnet
1)      Time Remaining:       98 secs  Hits: 2
                             Action Taken: 0
        Total Hash:    1718618  Hash String: /test.txt
        IPs gathered:
                1) 10.217.253.0
        Active Transactions: 0
 Done
>
<!--NeedCopy-->
```

7. クエリを繰り返し、制限識別子の統計を再度チェックして、統計が正しく更新されていることを確認します。

レートベースポリシーの例

October 7, 2021

次の表に、レートベースのポリシーの例を示します。

Table 1. Examples of Rate-Based Policies

Purpose	Example

Limit the number of requests per second from a URL	`add stream selector ipStreamSelector http.req.url "client.ip.src"`
	`add ns limitIdentifier ipLimitIdentifier -threshold 4 -timeSlice 1000`
	`-mode request_rate -limitType smooth -selectorName ipStreamSelector`
	
	`add responder action myWebSiteRedirectAction redirect`
	`"\"http://www.mycompany.com/\""`
	
	`add responder policy ipLimitResponderPolicy "http.req.url.contains(\"myasp.asp\")`
	`&& sys.check_limit(\"ipLimitIdentifier\")" myWebSiteRedirectAction`
	
	`bind responder global ipLimitResponderPolicy 100 END -type default`

Cache a response if the request URL rate exceeds 5 per 20000 milliseconds	`add stream selector cacheStreamSelector http.req.url`
	`add ns limitidentifier cacheRateLimitIdentifier -threshold 5 -timeSlice 2000`
	`-selectorName cacheStreamSelector`
	
	`add cache policy cacheRateLimitPolicy -rule "http.req.method.eq(get)`
	`&& sys.check_limit(\"cacheRateLimitIdentifier\")" -action cache`
	
	`bind cache global cacheRateLimitPolicy -priority 10`

Drop a connection on the basis of cookies received in requests from www.yourcompany.com if the requests exceed the rate limit	`add stream selector reqCookieStreamSelector "http.req.cookie`
	`.value(\"mycookie\")" "client.ip.src.subnet(24)"`
	
	`add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000`
	`-selectorName reqCookieStreamSelector`
	
	`add responder action sendRedirectUrl redirect '\"http://www.mycompany.com\"`
	`+ http.req.url' -bypassSafetyCheck YES`
	
	`add responder policy rateLimitCookiePolicy`
	`"http.req.url.contains(\"www.yourcompany.com\")`
	`&& sys.check_limit(\"myLimitIdentifier\")" sendRedirectUrl`

Drop a DNS packet if the requests from a particular client IP address and DNS domain exceed the rate limit	`add stream selector dropDNSStreamSelector client.udp.dns.domain client.ip.src`
	`add ns limitIdentifier dropDNSRateIdentifier -timeslice 20000 -mode request_rate`
	`-selectorName dropDNSStreamSelector -maxBandwidth 1 -trapsintimeslice 20`
	
	`add dns policy dnsDropOnClientRatePolicy "sys.check_limit`
	`(\"dropDNSRateIdentifier\")" -drop yes`

Limit the number of HTTP requests that arrive from the same host (with a subnet mask of 32) and that have the same destination IP address.	`add stream selector ipv6_sel "CLIENT.IPv6.src.subnet(32)" CLIENT.IPv6.dst Q.URL`
	`add ns limitIdentifier ipv6_id -imeSlice 20000 -selectorName ipv6_sel`
	`add lb vserver ipv6_vip HTTP 3ffe::209 80 -persistenceType NONE -cltTimeout 180`
	`add responder action redirect_page redirect "\"http://redirectpage.com/\""`
	`add responder policy ipv6_resp_pol "SYS.CHECK_LIMIT(\"ipv6_id\")" redirect_page`
	`bind responder global ipv6_resp_pol 5 END -type DEFAULT`

レートベースポリシーのサンプルユースケース

October 7, 2021

次のシナリオでは、グローバルサーバー負荷分散 (GSLB) におけるレートベースのポリシーの 2 つの使用方法について説明します。

- 最初のシナリオでは、DNS リクエストのレートが 1 秒あたり 1000 を超えた場合に、新しいデータセンターにトラフィックを送信するレートベースのポリシーの使用について説明します。
- 2 番目のシナリオでは、特定の期間内にローカル DNS (LDNS) クライアントに対して 5 つ以上の DNS 要求が到着すると、追加の要求はドロップされます。

トラフィックレートに基づくトラフィックのリダイレクト

このシナリオでは、近接ベースの負荷分散方法と、特定のリージョンの DNS 要求を識別するレート制限ポリシーを構成します。レート制限ポリシーでは、1 秒あたり 1000 個の DNS 要求のしきい値を指定します。DNS ポリシーは、「Europe.GB.17.London.UK-East.ISP-UK」地域の DNS リクエストにレート制限ポリシーを適用します。DNS ポリシーでは、要求 1001 から開始して 1 秒間隔の終了までレート制限しきい値を超える DNS 要求は、「North America.US.TX.Dallas.US-East.ISP-US」リージョンに関連付けられている IP アドレスに転送されます。

次の構成は、このシナリオを示しています。

```
add stream selector DNSSelector1 client.udp.dns.domain

add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000
     -selectorName DNSSelector1

add dns policy DNSLimitPolicy1 "client.ip.src.matches_location("Europe.
    GB.17.London.\*.\*") &&
sys.check_limit("DNSLimitIdentifier1")" -preferredLocation "North
    America.US.TX.Dallas.\*.\*"

bind dns global DNSLimitPolicy1 5
<!--NeedCopy-->
```

トラフィックレート基準での **DNS** 要求の削除

次のグローバルサーバロードバランシングの例では、特定の間隔で最大 5 つの DNS 要求を、解決のために LDNS クライアントに送信することを許可するレート制限ポリシーを設定します。このレートを超える要求はすべて廃棄されます。この種類のポリシーは、Citrix ADC をリソースの悪用から保護するのに役立ちます。たとえば、このシナリオ

では、接続の存続可能時間 (TTL) が 5 秒である場合、このポリシーは、LDNS がドメインを再クエリするのを防ぎます。代わりに、Citrix ADC にキャッシュされたデータを使用します。

```
add stream selector LDNSSelector1 client.udp.dns.domain client.ip.src

add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice
    1000 -selectorName LDNSSelector1

add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(".") && sys.
    check_limit("LDNSLimitIdentifier1")" -drop YES

bind dns global LDNSPolicy1 6

show gslb vserver gvip

gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)       3 (Active)
Persistence: NONE       Persistence ID: 100
Disable Primary Vserver on Down: DISABLED       Site Persistence: NONE
Backup Session Timeout: 0
Empty Down Response: DISABLED
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1.     site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2.      site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3.      site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
```



```
4.      www.gslbindia.com        TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
<!--NeedCopy-->
```

トラフィックドメインのレート制限

October 7, 2021

トラフィックドメインのレート制限を設定できます。Citrix ADC 式言語の次の式は、トラフィックドメインに関連付けられたトラフィックを識別します。

- client.traffic_domain.id

特定のトラフィックドメイン、一連のトラフィックドメイン、またはすべてのトラフィックドメインに関連付けられたトラフィックに対して、レート制限を設定できます。

トラフィックドメインのレート制限を構成するには、構成ユーティリティまたは Citrix ADC コマンドラインを使用して、Citrix ADC アプライアンスで次の手順を実行します。

1. client.traffic_domain.id 式を使用して、トラフィックドメインに関連付けられたトラフィックを識別し、レート制限するストリームセレクタを設定します。
2. レート制限の対象となるトラフィックの最大しきい値などのパラメータを指定するレート制限 ID を設定します。また、このステップでは、ストリームセレクタをレートリミッタに関連付けます。
3. レート制限 ID を使用するポリシーに関連付けるアクションを設定します。
4. sys.check_limit 式プレフィクスを使用してレート制限識別子を呼び出すポリシーを設定し、アクションをこのポリシーに関連付けます。
5. ポリシーをグローバルにバインドします。

たとえば、ID が 10 と 20 の 2 つのトラフィックドメインが Citrix ADC NS1 で設定されているとします。トラフィックドメイン 10 では、LB1-TD-1 はサーバ S1 および S2 を負荷分散するように設定され、LB2-TD1 はサーバ S3 および S4 を負荷分散するように設定されています。

トラフィックドメイン 20 では、LB1-TD-2 はサーバ S5 および S6 をロードバランシングするように設定され、LB2-TD2 はサーバ S7 および S8 をロードバランシングするように設定されます。

次の表に、設定例でのトラフィックドメインのレート制限ポリシーの例を示します。

目的	CLI コマンド
各トラフィックドメインのリクエスト数を 1 秒あたり 10 に制限します。	add stream selector tdratelimit-1 CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier limitidf-1 -threshold 10 -selectorName tdratelimit-1 -trapsInTimeSlice 0 add responder policy ratelimit-pol “sys.check_limit(\“limitidf-1\”)” DROP bind responder global ratelimit-pol 1
各トラフィックドメインについて、クライアントあたり 1 秒あたり 5 件のリクエスト数を制限します。	add stream selector tdandclientip CLIENT.IP.SRC,CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td_limitidf -threshold 5 -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy tdratelimit-pol “sys.check_limit(\“td_limitidf\”)” DROP bind responder global tdratelimit-pol 2
特定のトラフィックドメイン（トラフィックドメイン 10 など）に送信されるリクエストの数を 3 秒ごとに 30 件に制限します。	add stream selector tdratelimit CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td10_limitidf -threshold 30 -timeSlice 3000 -selectorName tdratelimit -trapsInTimeSlice 5 add responder policy td10ratelimit “client.traffic_domain.id==10 && sys.check_limit(\“td10_limitidf\”)” DROP bind responder global td10ratelimit 3
特定のトラフィックドメイン（トラフィックドメイン 20 など）の接続数を、クライアントあたり 1 秒あたり 5 に制限します。	add stream selector tdandclientip CLIENT.IP.SRC CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td20_limitidf -threshold 5 -mode CONNECTION -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy td20_ratelimit “client.traffic_domain.id==20 && sys.check_limit(\“td20_limitidf\”)” DROP bind responder global td20_ratelimit 4

パケットレベルでレート制限を設定する

October 7, 2021

ストリームセレクタとレスポンダポリシーを設定して、セレクタによって識別されるすべての接続を通過するパケットレベルで統計情報を収集できます。1 秒あたりのパケット数が設定されたしきい値を超えると、ポリシーによって設定されたアクション（RESET または DROP）が適用されます。これらのポリシーは、すべてのタイプの仮想サーバに対して構成できます。すべてのサイズのパケットが考慮されます。

パケットレベルでレート制限を設定するには、次の作業を実行します

1. 負荷分散を有効にする
2. ストリームセレクタの追加
3. ストリーム識別子の追加
4. レスポンダーポリシーの追加
5. 負荷分散仮想サーバの追加
6. レスポンダーポリシーのバインド

負荷分散機能を有効にするには

コマンドプロンプトで入力します。

```
enable ns feature lb
<!--NeedCopy-->
```

ストリームセレクタを追加するには

コマンドプロンプトで入力します。

```
add stream selector packetlimitselector client.ip.src client.tcp.
   srcport client.ip.dst client.tcp.dstport
<!--NeedCopy-->
```

ストリーム識別子を追加するには

コマンドプロンプトで入力します。

```
add stream identifier packetlimitidentifier packetlimitselector -
   interval 1
<!--NeedCopy-->
```


ACK のみのパケットのトラッキングを有効にするには

コマンドプロンプトで入力します。

```
set stream identifier packetlimitidentifier  - trackAckOnlyPackets
    ENABLED
<!--NeedCopy-->
```

レスポンダーポリシーを追加するには

コマンドプロンプトで入力します。

```
add responder policy packet_rate_sessionpolicy ”ANALYTICS.STREAM(”
    packetlimitidentifier”).COLLECT_STATS(”PACKET_LIMIT”, <
    max_threshold_PPS>, ACTION, 0/1)” NOOP
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

- <max_threshold_PPS> は、1 秒あたりの接続で許可される最大パケット数です。
- アクションは、ドロップまたはリセットすることができます。
- 0 または 1 は制限タイプを表します。0 は BURSTY 制限タイプを表し、1 は SMOOTH 制限タイプを表します。

例**:**

```
add responder policy packet_rate_sessionpolicy ”ANALYTICS.STREAM(”
    packetlimitidentifier”).COLLECT_STATS(”PACKET_LIMIT”, 40, RESET, 0)”
     NOOP
<!--NeedCopy-->
```

負荷分散仮想サーバーを追加するには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-lb-1 HTTP 10.102.20.200 80
<!--NeedCopy-->
```


レスポンダーポリシーをバインドするには

セレクタとレスポンダポリシーの設定後、ポリシーをグローバルにバインドすることも、特定の仮想サーバにバインドすることもできます。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
bind responder global <policyName> <priority> [<gotoPriorityExpression
    >] [-type <type>] [-invoke  (<labelType>  <labelName>) ]
<!--NeedCopy-->
```

または

```
bind lb vserver <name>@  (-policyName <string>@ [-priority <
    positive_integer>]
<!--NeedCopy-->
```

例:

```
bind responder global packet_rate_sessionpolicy 101 END -type
    REQ_DEFAULT

bind responder global packet_rate_sessionpolicy 102 END -type

bind lb vserver v1 -policyname packet_rate_sessionpolicy -priority 10
<!--NeedCopy-->
```

レスポンダー

October 7, 2021

> 警告
>
> 従来のポリシーを使用したフィルター機能は廃止されました。代替手段として、高度なポリシーインフラストラクチャで書き換え機能とレスポンダー機能の使用をお勧めします。

今日の複雑な Web 構成では、表面上に表示される HTTP 要求に対する応答が異なることがよくあります。ユーザーが Web ページを要求するとき、ユーザーの地理的な場所、ブラウザーの仕様、またはブラウザーが受け入れる言語と優先順位に応じて、異なるページを提供することができます。DDoS 攻撃が発生している、またはハッキングの試みを開始している IP 範囲からリクエストが送信されている場合は、接続をドロップできます。

レスポンダーは、TCP、DNS (UDP)、および HTTP などのプロトコルをサポートしています。アプライアンスでレスポンダーを有効にすると、サーバーの応答は、要求の送信者、送信元、およびセキュリティおよびシステム管理に影響するその他の基準に基づいて行うことができます。この機能はシンプルで迅速に使用できます。より複雑な機能の呼び出しを避けることで、複雑な処理を必要としない要求の処理に費やされる CPU サイクルと時間を削減できます。

財務情報などの機密データを処理する場合、クライアントが安全な接続を使用してサイトを参照するようにする場合は、`http://`の代わりに`https://`を使用して、要求を安全な接続にリダイレクトできます。

レスポンダーを使用するには、次の操作を行います。

- アプライアンスでレスポンダー機能を有効にします。
- レスポンダーのアクションを構成します。アクションは、カスタムレスポンスの生成、リクエストの別の Web ページへのリダイレクト、または接続のリセットです。
- レスポンダーポリシーを構成します。ポリシーは、アクションを実行する必要があるリクエスト（トラフィック）を決定します。
- 各ポリシーをバインドポイントにバインドして有効にします。バインドポイントとは、Citrix ADC アプライアンスがトラフィックを調べて、ポリシーと一致するかどうかを確認するエンティティを指します。たとえば、バインドポイントを負荷分散仮想サーバーにすることができます。

どのポリシーにも一致しない要求に対しても既定のアクションを指定できます。また、エラーメッセージを生成するアクションの安全性チェックをバイパスすることもできます。

Citrix ADC の書き換え機能は、Citrix ADC が処理する要求または応答の一部の情報を書き換えるのに役立ちます。次のセクションでは、2 つの機能の違いをいくつか示します。

[書き換え] オプションと [レスポンダー] オプションの比較

書き換え機能と応答側機能の主な違いは次のとおりです。

レスポンダーは、レスポンスまたはサーバーベースの式には使用できません。レスポンダーは、クライアントパラメーターに応じて、次のシナリオでのみ使用できます。

- 新しい Web サイトまたは Web ページへの HTTP リクエストのリダイレクト
- カスタム応答で応答する
- 要求レベルでの接続の削除またはリセット

レスポンダーポリシーがある場合、Citrix ADC はクライアントからの要求を検証し、該当するポリシーに従ってアクションを実行し、応答をクライアントに送信し、クライアントとの接続を閉じます。

書き換えポリシーがある場合、Citrix ADC はクライアントからの要求またはサーバーからの応答を調べて、該当するポリシーに従ってアクションを実行し、トラフィックをクライアントまたはサーバーに転送します。

一般に、要求ベースのパラメータに基づいてアプライアンスが接続をリセットまたはドロップする場合は、レスポンダーを使用することをお勧めします。レスポンダーを使用して、トラフィックをリダイレクトしたり、カスタムメッセージで応答したりできます。HTTP リクエストとレスポンスのデータを操作するには、書き換えを使用します。

レスポンダー機能の有効化

October 7, 2021

レスポンダー機能を使用するには、まずレスポンダー機能を有効にする必要があります。

Citrix ADC コマンドラインを使用してレスポンダー機能を有効にするには:

コマンドプロンプトで次のコマンドを入力して、レスポンダ機能を有効にし、構成を確認します。

- `enable ns feature <feature>`
- `show ns feature`

例:

```
enable ns feature Responder
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 .
 .
 .
 1)     Responder                       RESPONDER            ON
 2)     HTML Injection                  HTMLInjection        ON
 3)     Citrix ADC Push                  push                 OFF
 Done
>
<!--NeedCopy-->
```

GUI を使用してレスポンダ機能を有効にするには、次の手順を実行します。

1. ナビゲーションペインで、[システム] を展開し、[設定] をクリックします。
2. 詳細ウィンドウの [モードと 機能] で、[高度な機能の変更] をクリックします。
3. [高度な機能の構成] ダイアログボックスで、[レスポンダー] チェックボックスをオンにし、[**OK**] をクリックします。
4. 機能を有効化/無効化しますか? ダイアログボックスで、[はい] をクリックします。ステータスバーに、機能が有効になっていることを示すメッセージが表示されます。

レスポンダーのアクションを構成する

October 7, 2021

応答側機能を有効にした後、要求を処理するための 1 つ以上のアクションを設定する必要があります。レスポンダーは、次のタイプのアクションをサポートします。

- **Respond with**。要求を Web サーバーに転送せずに、Target 式で定義された応答を送信します。（Citrix ADC アプライアンスは、Web サーバーの代わりに使用され、Web サーバーとして機能します）。このタイプのアクションを使用して、単純な HTML ベースのレスポンスを手動で定義します。通常、Respond with アクションのテキストは、Web サーバーのエラーコードと簡潔な HTML ページで構成されます。
- **Respond with SQL OK**。ターゲット式で定義された指定された SQL OK レスポンスを送信します。このタイプのアクションを使用して、SQL OK 応答を SQL クエリに送信します。
- **Respond with SQL Error**。ターゲット式で定義された指定された SQL エラー応答を送信します。この種類のアクションを使用して、SQL エラー応答を SQL クエリに送信します。
- **Respond with HTML page**。指定された HTML ページを応答として送信します。以前にアップロードされた HTML ページのドロップダウンリストから選択するか、新しい HTML ページをアップロードできます。このタイプのアクションを使用して、インポートされた HTML ページをレスポンスとして送信します。アプライアンスは、responsewithhtmlpage レスポンダーアクションでカスタムヘッダーを使用して応答します。最大 8 つのカスタムヘッダーを構成できます。
- **Redirect**。要求を別の Web ページまたは Web サーバーにリダイレクトします。Redirect アクションは、DNS に存在するが、実際の Web サーバーが存在しない「ダミー」の Web サイトに元々送信された要求を、実際の Web サイトにリダイレクトできます。また、検索リクエストを適切な URL にリダイレクトすることもできます。通常、リダイレクトアクションのリダイレクトターゲットは完全な URL で構成されます。

Citrix ADC コマンドラインを使用してレスポンダーのアクションを構成するには：

指定したレスポンダアクションの現在の設定を表示します。アクション名を指定しない場合は、Citrix ADC アプライアンスで現在構成されているすべてのレスポンダーアクションのリストを省略して表示します。

コマンドプロンプトで次のコマンドを入力して、レスポンダーアクションを構成し、構成を確認します。

- `add responder action <name> <type> <target> [-bypassSafetyCheck (YES | NO)]`
- `show responder action`

パラメーター：

- **Name**：レスポンダアクションの名前。最大長：127
- **type**。レスポンダアクションのタイプ。次のようになります。（応答します）。
- ターゲット。応答するものを指定する式
- **htmlpage**. htmlpage で応答するように指定するオプション

- **bypassSafetyCheck**。安全でない式を許可する安全性チェック。注：この属性は非推奨です。
- **hits**。アクションが実行された回数。
- **referenceCount**。アクションへの参照の数。
- **undefHits**。アクションの結果が UNDEF になった回数。
- **comment**。このレスポンダーのアクションに関するあらゆる種類の情報。
- **builtin**。レスポンダアクションが組み込まれているかどうかを判断するフラグ

例:

```
To create a responder action that displays a “Not Found” error page for URLs that do not exist:

> add responder action act404Error respondWith '”HTTP/1.1 404 Not Found\r\n\r\n”+ HTTP.REQ.URL.HTTP_URL_SAFE + ” does not exist on the web server.”'
Done

> show responder action

1) Name: act404Error
Operation: respondwith
Target: ”HTTP/1.1 404 Not Found” + HTTP.REQ.URL.HTTP_URL_SAFE + ” does not exist on the web server.”
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

To create a responder action that displays a “Not Found” error page for URLs that do not exist:

add responder action act404Error respondWith '”HTTP/1.1 404 Not Found\r\n\r\n”+ HTTP.REQ.URL.HTTP_URL_SAFE + ” does not exist on the web server.”'
Done
> show responder action

1) Name: act404Error
Operation: respondwith
Target: ”HTTP/1.1 404 Not Found” + HTTP.REQ.URL.HTTP_URL_SAFE + ” does not exist on the web server.”
```



```
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
<!--NeedCopy-->
```

Citrix ADC コマンドラインを使用して既存のレスポンダーアクションを変更するには:

コマンドプロンプトで次のコマンドを入力して、既存のレスポンダーアクションを変更し、構成を確認します。

- `set responder action <name> -target <string> [-bypassSafetyCheck ( YES | NO )]`
- `show responder action`

例:

```
set responder action act404Error  -target '"HTTP/1.1 404 Not Found\r\n\
    r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web
    server."'
 Done
> show responder action

1)      Name: act404Error
        Operation: respondwith
        Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE +
            " does not exist on the web server."
        BypassSafetyCheck : NO
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done
<!--NeedCopy-->
```

Citrix ADC コマンドラインを使用してレスポンダーのアクションを削除するには:

コマンドプロンプトで次のコマンドを入力して、レスポンダーアクションを削除し、構成を確認します。

- `rm responder action <name>`
- `show responder action`

例:

```
rm responder action act404Error
```



```
 Done

> show responder action
 Done
<!--NeedCopy-->
```

Citrix ADC コマンドラインを使用して、レスポンダーアクションにカスタムヘッダーを追加するには、次の手順に従います。

Citrix ADC アプライアンスは、レスポンダーアクションでカスタムヘッダーを使用して応答できるようになりました。最大 8 つのカスタムヘッダーを構成できます。以前は、アプライアンスはContent-type:text/htmlとContent-Length:<value> 静的ヘッダーでのみ応答していました。

> 注:
>
> カスタムヘッダー構成では、「Content-Type」ヘッダー値を上書きすることもできます。

コマンドプロンプトで、次のコマンドを入力します。

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment <string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <expression>] [-headers <name(value)> ...]
```

各項目の意味は次のとおりです。

名を入力します。レスポンダアクションの名前。文字、数字、またはアンダースコア文字 (_) で始まり、文字、数字、ハイフン (-)、ピリオド (.) ハッシュ (#)、スペース ()、アットマーク (@)、等しい (=)、コロン (:)、およびアンダースコア文字のみを含める必要があります。レスポンダーポリシーの追加後に変更できます。

タイプ。レスポンダアクションのタイプ。使用可能な設定は次のように機能します。

1. respondwith<target> -ターゲットとして指定された式でリクエストに応答します。
2. respondwithhtmlpage-アップロードされた HTML ページオブジェクトをターゲットとして指定してリクエストに応答します。
3. redirect-ターゲットとして指定された URL にリクエストをリダイレクトします。
4. sqlresponse_ok-SQL OK レスポンスを送信します。
5. sqlresponse_error-SQL エラーレスポンスを送信します。これは必須の引数です。可能な値：noop、応答、リダイレクト、応答ページへの返信、sqlresponse_ok、sqlresponse_error

ターゲット。応答する内容を指定する式。通常、リダイレクトポリシーの URL またはデフォルトの構文式です。リクエスト内の情報を参照する Citrix ADC のデフォルトの構文式に加えて、stringbuilder 式には、テキストと HTML、および新しい行と段落を定義する単純なエスケープコードを含めることができます。各 stringbuilder 式要素（Citrix ADC のデフォルトの構文式または文字列）を二重引用符で囲みます。要素を結合するには、プラス (+) 文字を使用します。

htmlpage. 応答の htmlpage ポリシーの場合、応答として使用する HTML ページオブジェクトの名前。最初にページオブジェクトをインポートする必要があります。最大長：31

コメント。このレスポンダーのアクションに関するあらゆる種類の情報。最大長：255

responseStatusCode。HTTP 応答ステータスコード (例:200、302、404 など) リダイレクトアクションタイプのデフォルト値は 302 で、応答 withhtmlpage の場合は 200 です。最小値:100 最大値:599

reasonPhrase。HTTP レスポンスの理由フレーズを指定する式。reason フレーズは、引用符で囲まれた文字列リテラルまたは PI 式です。たとえば、「無効な URL:」+ HTTP.REQ.URL の最大長は 8191 です。

[ヘッダ] HTTP レスポンスに挿入する 1 つ以上のヘッダー。各ヘッダーは「name (expr)」として指定されます。expr は、実行時に評価され、名前付きヘッダーの値を提供する式です。レスポンダーアクションには、最大 8 つのヘッダーを構成できます。

GUI を使用してレスポンダアクションを構成するには、次の手順を実行します。

1. **AppExpert > Responder > Actions** に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - アクションを作成するには、[追加] をクリックします。
   - 既存のアクションを変更するには、アクションを選択して、[開く] をクリックします。
3. アクション を作成するか、既存のアクションを修正するかによって、「作成」(Create) または「**OK**」(OK) をクリックします。
4. [閉じる] をクリックします。ステータスバーに、機能が有効になっていることを示すメッセージが表示されます。
5. 応答側のアクションを削除するには、アクションを選択し、[削除] をクリックします。ステータスバーに、機能が無効になったことを示すメッセージが表示されます。

[式の追加] ダイアログボックスを使用して式を追加するには

1. [レスポンダーアクションの作成] または [レスポンダーアクションの構成] ダイアログボックスで、[追加] をクリックします。
2. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。
   - HTTP HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
   - SYS。1 つ以上の保護された Web サイトリクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - CLIENT。要求を送信したコンピュータ。リクエストの送信者の一部を調べる場合は、これを選択します。
   - ANALYTICS. リクエストに関連付けられた分析データ。リクエストメタデータを調べる場合は、これを選択します。
   - SIP. SIP 要求。SIP 要求の一部の側面を調べる場合は、これを選択します。選択すると、右端のリストボックスには、式の次の部分に適した用語が一覧表示されます。
3. 2 番目のリストボックスで、式の 2 番目の項を選択します。選択内容は、前のステップで行った選択内容によって異なり、コンテキストに適切です。2 番目の選択を行うと、[式を構築] ウィンドウの下にある [ヘルプ] ウィンドウ (空白) に、選択した用語の目的と使用方法を示すヘルプが表示されます。

4. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。

グローバル **HTTP** アクションの設定

HTTP 要求がタイムアウトしたときにレスポンダアクションを呼び出すように、グローバル HTTP アクションを設定できます。この機能を設定するには、最初に呼び出すレスポンダアクションを作成する必要があります。次に、そのレスポンダーアクションでタイムアウトに応答するように、グローバル HTTP タイムアウトアクションを設定します。

Citrix ADC コマンドラインを使用してグローバル HTTP アクションを構成するには:

コマンドプロンプトで、次のコマンドを入力します。

- `set ns httpProfile -reqTimeoutAction <responder action name>`
- `save ns config`

`<responder action name>`で、レスポンダアクションの名前を置き換えます。

HTML ページのインポートの設定

Citrix ADC アプライアンスは、カスタムメッセージで応答すると, 我々は、HTML ファイルで応答することができます.`import responder htmlpage`コマンドを使用してファイルをインポートし、このファイルを`add responder action <act name> respondwithhtmlpage <file name>`コマンドで使用することができます。Citrix ADC GUI を使用してファイルをインポートすることもできます。必要な HTML ページをアプライアンスフォルダにインポートし、レスポンダの実行時にページをアップロードできます。

CLI を使用した **HTML** ページのインポート

コマンドプロンプトで入力します。

```
import responder htmlpage [<src>] <name> [-comment <string>] [-overwrite][-CAcertFile <string>]
```

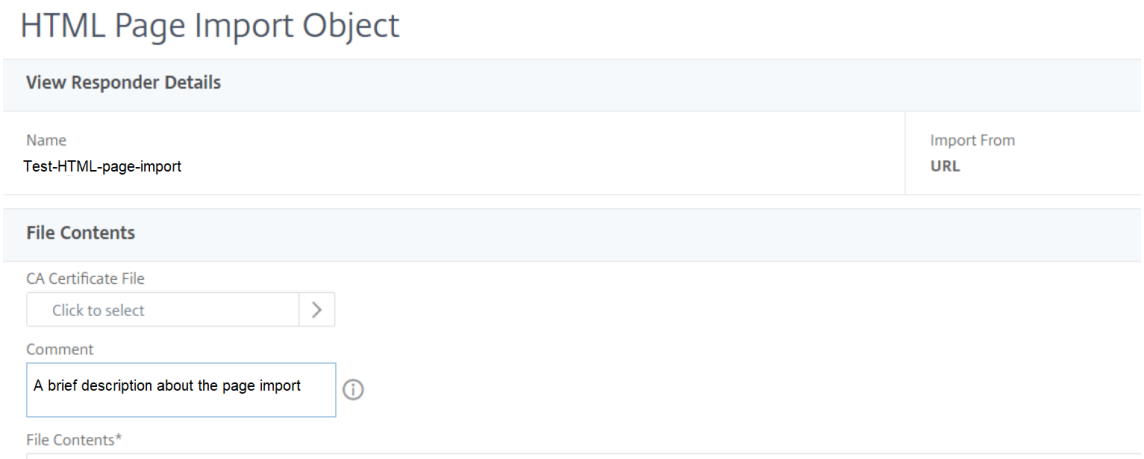
例**:**

```
import responder htmlpage http://www.example.com/page.html my-responder-page -CAcertFile my_root_ca_cert
```

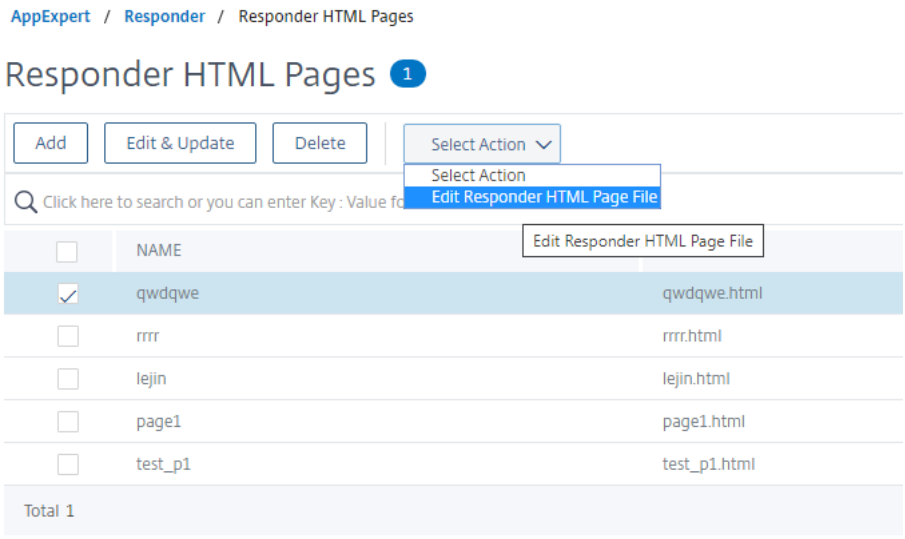
ここで、
CA 証明書は、クライアント証明書の検証に使用されます。証明書は、「import ssl certfile」CLI コマンドを使用してインポートするか、API または GUI を介して同等の方法でインポートする必要があります。証明書名が構成されていない場合は、デフォルトのルート CA 証明書が証明書の検証に使用されます。

Citrix ADC GUI を使用して **HTML** ページをインポートする

1. **AppExpert** > レスポンダー >**HTML** ページのインポートに移動します。
2. レスポンダー **HTML** インポートの詳細ウィンドウで、[追加] をクリックします。
3. **HTML** ページのインポート・オブジェクト・ページで、次のパラメータを設定します。
   a) Name：HTML ページの名前。
   b) [読み込み元]: ファイル、テキスト、またはテキストから読み込まれます。
   c) URL。HTML ファイルの URL の場所を入力するために選択します。
   d) File。アプライアンスのディレクトリから HTML ファイルを選択します。
   e) Text。HTML ファイルをテキストとして選択します。
4. ［続行］をクリックします。
5. レスポンダの HTML ページの詳細を確認します。
6. ［完了］をクリックします。



HTML ページを編集するには、ファイルを選択し、[アクションの選択] ドロップダウンリストから [レスポンダー **HTML** ページファイルの編集] をクリックします。



レスポンダーポリシーの設定

October 7, 2021

レスポンダーのアクションを構成したら、次にレスポンダーポリシーを構成して、Citrix ADC アプライアンスが応答するリクエストを選択する必要があります。レスポンダポリシーは、1 つ以上の式で構成される規則に基づいています。ルールはアクションに関連付けられています。アクションは、リクエストがルールに一致した場合に実行されます。

注：レスポンダーポリシーの作成と管理については、Citrix ADC コマンドプロンプトでは使用できないサポートが GUI に表示されます。

Citrix ADC コマンドラインを使用してレスポンダーポリシーを構成するには：

コマンドプロンプトで入力します。

- `add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction <actionName>`
- `show responder policy <name>`

例：

```
1 > add responder policy policyThree ”CLIENT.IP.SRC.IN_SUBNET
     (222.222.0.0/16)” RESET
2  Done
3 > show responder policy policyThree
4
```



```
        Name: policyThree
        Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
        Responder Action: RESET
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0
 Done
<!--NeedCopy-->
```

Citrix ADC コマンドラインを使用して既存のレスポンダーポリシーを変更するには:

コマンドプロンプトで入力します。

- `set responder policy <name> [-rule <expression>] [-action <string>] [-undefAction <string>]`
- `show responder policy <name>`

Citrix ADC コマンドラインを使用してレスポンダーポリシーを削除するには:

コマンドプロンプトで入力します。

- `rm responder policy <name>`
- `show responder policy`

例:

```
>rm responder policy pol404Error
 Done

> show responder policy
 Done
<!--NeedCopy-->
```

GUI を使用してレスポンダポリシーを設定するには、次の手順を実行します。

1. **AppExpert >** レスポンダー **>** ポリシーに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
3. 新しいポリシーを作成するか、既存のポリシーを変更するかに応じて、[**Create**] または [**OK**] をクリックします。
4. [閉じる] をクリックします。ステータスバーに、機能が設定されたことを示すメッセージが表示されます。

レスポンダーポリシーのバインド

February 21, 2022

ポリシーを有効にするには、Citrix ADC を通過するすべてのトラフィックに適用されるようにグローバルにバインドするか、特定の仮想サーバーにバインドして、宛先 IP アドレスがその仮想サーバーの VIP である要求にのみポリシーを適用するようにする必要があります。

ポリシーをバインドするときは、そのポリシーにプライオリティを割り当てます。プライオリティによって、定義したポリシーが評価される順序が決まります。優先度は、任意の正の整数に設定できます。

Citrix ADC オペレーティングシステムでは、ポリシーの優先順位は逆の順序で機能します。数値が大きいほど優先度は低くなります。たとえば、プライオリティが 10、100、1000 の 3 つのポリシーがある場合、プライオリティ 10 が割り当てられたポリシーが最初に実行され、次にポリシーにプライオリティ 100 が割り当てられ、最後にポリシーにオーダー 1000 が割り当てられます。レスポンダー機能では、リクエストが最初に一致するポリシーのみが実装され、一致する可能性のある追加のポリシーは実装されないため、意図した結果を得るにはポリシーの優先度が重要です。

ポリシーをグローバルにバインドするときに、各ポリシーの間隔を 50 または 100 に設定して優先度を設定することで、他のポリシーを任意の順序で追加する余地を十分に確保し、必要な順序で評価するように設定できます。その後、既存のポリシーの優先順位を再割り当てすることなく、いつでもポリシーを追加できます。

Citrix ADC でのポリシーのバインドの詳細については、「ポリシーと式」を参照してください。

> 注:
>
> レスポンダーポリシーは TCP ベースの仮想サーバーにバインドされます。

Citrix ADC コマンドラインを使用してレスポンダーポリシーをグローバルにバインドするには:

コマンドプロンプトで次のコマンドを入力して、レスポンダーポリシーをグローバルにバインドし、構成を確認します。

- `bind responder global <policyName> <priority> [<gotoPriorityExpression> [-type <type>] [-invoke (<labelType> <labelName>)]`
- `show responder global`

例:

```
> bind responder global poliError 100
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

 Done
<!--NeedCopy-->
```


Citrix ADC コマンドラインを使用してレスポンダーポリシーを特定の仮想サーバーにバインドするには:

コマンドプロンプトで入力します。

- `bind lb vserver <name> -policyname <policy_name> -priority <priority>`
- `show lb vserver vs-loadbal <name>`

例:

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
2)      vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:46 2009 (+213 ms)
        Time since last state change: 2 days, 00:00:04.260
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```


GUI を使用してレスポンダーポリシーをグローバルにバインドするには、次の手順を実行します。

1. **AppExpert** > レスポンダー > ポリシーに移動します。
2. [レスポンダーポリシー] ページで、レスポンダーポリシーを選択し、[ポリシーマネージャー] をクリックします。
3. [レスポンダーポリシーマネージャー] ダイアログボックスの [バインドポイント] メニューで、[既定のグローバル] を選択します。
4. [**Insert Policy**] をクリックして新しい行を挿入し、すべての非バインドレスポンダーポリシーのドロップダウンリストを表示します。
5. リストにあるポリシーの 1 つをクリックします。このポリシーは、グローバルにバインドされたレスポンダーポリシーの一覧に挿入されます。
6. [変更を適用] をクリックします。
7. [閉じる] をクリックします。ステータスバーに、構成が正常に完了したことを示すメッセージが表示されます。

GUI を使用してレスポンダーポリシーを特定の仮想サーバーにバインドするには、次の手順を実行します。

1. トラフィック管理 > 負荷分散 > 仮想サーバーに移動します。
2. [負荷分散仮想サーバー] ページで、レスポンダーポリシーをバインドする仮想サーバーを選択し、[開く] をクリックします。
3. [仮想サーバーの構成（負荷分散）] ダイアログボックスで、［ポリシー］ タブを選択します。このタブには、Citrix ADC アプライアンスに構成されているすべてのポリシーの一覧が表示されます。
4. この仮想サーバにバインドするポリシーの名前の横にあるチェックボックスをオンにします。
5. [**OK**] をクリックします。ステータスバーに、構成が正常に完了したことを示すメッセージが表示されます。

レスポンダーポリシーの既定のアクションの設定

October 7, 2021

Citrix ADC アプライアンスは、要求がレスポンダーポリシーと一致しない場合、未定義のイベント（UNDEF イベント）を生成します。その後、アプライアンスは、未定義のイベントに割り当てられたデフォルトのアクションを実行します。デフォルトでは、アクションは要求を次の機能（負荷分散、コンテンツフィルタリングなど）に転送します。この既定の動作により、要求が特定のレスポンダーアクションを Web サーバーに送信する必要がないことが保証されます。また、クライアントは、要求したコンテンツへのアクセスを受け取ります。

ただし、Citrix ADC アプライアンスが保護する 1 つ以上の Web サイトが、無効な要求や悪意のある要求を多数受け取る場合は、クライアント接続をリセットするか、要求を破棄するようにデフォルトのアクションを変更できます。このタイプの構成では、正当な要求と一致する 1 つ以上のレスポンダーポリシーを作成し、それらの要求を元の宛先にリダイレクトするだけです。その後、Citrix ADC アプライアンスは、設定したデフォルトのアクションで指定された他の要求をすべてブロックします。

未定義のイベントには、次のいずれかのアクションを割り当てることができます。

- **NOOP**。NOOP アクションは、レスポンダ処理を中止しますが、パケットフローは変更しません。アプライアンスは、応答側のポリシーに一致しない要求を処理し続け、別の機能が介入して要求をブロックまたはリダイレクトしない限り、要求された URL に転送します。このアクションは、Web サーバーへの通常の要求に適しており、既定の設定です。
- **RESET**。未定義のアクションが RESET に設定されている場合、アプライアンスはクライアント接続をリセットし、Web サーバーとのセッションを再確立する必要があることをクライアントに通知します。このアクションは、存在しない Web ページに対する繰り返し要求や、保護された Web サイトをハッキングまたはプローブしようとする接続に適しています。
- **DROP**。未定義のアクションが DROP に設定されている場合、アプライアンスはクライアントに何らかの形で応答せずに要求をサイレントにドロップします。このアクションは、DDoS 攻撃またはサーバーに対するその他の持続的な攻撃の一部であると思われる要求に適しています。

注: UNDEF イベントは、クライアント要求に対してのみトリガされます。応答に対して UNDEF イベントはトリガーされません。

Citrix ADC コマンドラインを使用して未定義のアクションを設定するには:

コマンドプロンプトで次のコマンドを入力して、未定義のアクションを設定し、構成を確認します。

- `set responder param -undefAction (RESET|DROP|NOOP)[-timeout <msecs>]`
- `show responder param`

各項目の意味は次のとおりです。

timeout: すべてのポリシーと選択したアクションを中断することなく処理できる最大時間（ミリ秒）。タイムアウトに達すると、評価によって UNDEF が発生し、それ以上の処理は実行されません。

最小値:1

最大値:5000

例**:**

```
>set responder param -undefAction RESET -timeout 3900
 Done
> show responder param
Action Name: RESET
Timeout: 3900
 Done
>
<!--NeedCopy-->
```

GUI を使用して未定義のアクションを設定する

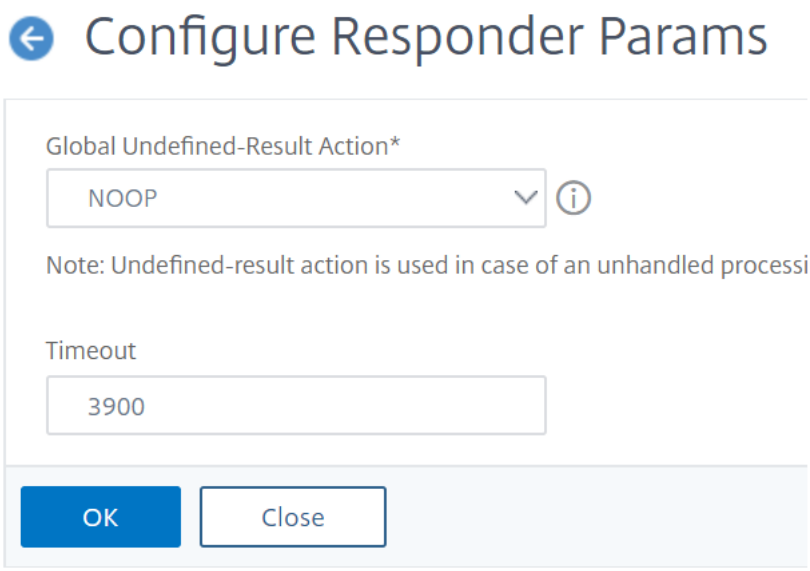
1. [**AppExpert**] > [レスポンダー] に移動し、[設定] の [レスポンダーの設定の変更**]** リンクをクリックします。

2. [レスポンダーパラメータの設定] ページで、次のパラメータを設定します。

   a) グローバル未定義の結果アクション。応答側のポリシーとアクションで未処理の処理処理例外では、未定義の結果アクションが優先されます。**NOOP**、**RESET** または **DROP** を選択します。

   b) タイムアウト。すべてのポリシーと選択したアクションを中断することなく処理できる最大時間（ミリ秒）。タイムアウトに達すると、評価によって UNDEF が発生し、それ以上の処理は実行されません。

3. [**OK**] をクリックします。



。

レスポンダーのアクションとポリシーの例

January 25, 2022

レスポンダーのアクションとポリシーは強力で複雑ですが、比較的単純なアプリケーションから始めることができます。

例**:** 指定された **IP** からのアクセスのブロック

次の手順では、CIDR 222.222.0.0/16 から送信されたクライアントによる保護された Web サイトへのアクセスをブロックします。レスポンダは、要求された URL へのアクセスがクライアントに許可されていないことを示すエラーメッセージを送信します。

Citrix ADC コマンドラインを使用してアクセスをブロックするには:

コマンドプロンプトで、次のコマンドを入力してアクセスをブロックします。

- add responder action act_unauthorized respond with “HTTP/1.1 403 Forbidden\r\n\r\n” + “Client: “ + CLIENT.IP.SRC + “ is not authorized to access URL:” + “HTTP.REQ.URL.HTTP_URL_SAFE”’

- add responder policy pol_un “CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)” act_unauthorized
- bind responder global pol_un 10

GUI を使用してアクセスをブロックするには、次の手順を実行します。

1. ナビゲーションウィンドウで [レスポンダー] を展開し、[アクション] をクリックします。
2. 詳細ペインで、[**Add**] をクリックします。
3. [レスポンダーアクションの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] テキストボックスに act_unauthorized と入力します。
   b) [種類] で、[返信方法] を選択します。
   c) [ターゲット] テキスト領域に、次の文字列を入力します。「HTTP/1.1 403 禁止された\ r\ n\ r\ n」+「クライアント:” + CLIENT.IP.SRC + “は URL にアクセスする権限がありません:」+ HTTP.REQ.URL.HTTP_URL_SAFE
   d) [**Create**] をクリックしてから、[**Close**] をクリックします。
   構成した act_unauthorized という名前のレスポンダーアクションが [レスポンダーアクション] ページに表示されます。
4. ナビゲーションペインで、[**Policies**] をクリックします。
5. 詳細ペインで、[**Add**] をクリックします。
6. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] テキストボックスに pol_unauthorized と入力します。
   b) [アクション] で [act_unauthorized] を選択します。
   c) [式] ウィンドウで、次のルールを入力します。CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)
   d) [作成] をクリックし、[閉じる] をクリックします。
   構成した pol_unauthorized という名前のレスポンダーポリシーが [レスポンダーポリシー] ページに表示されます。
7. レスポンダーポリシーのバインドで説明されているように、新しいポリシー pol_unauthorized をグローバルにバインドします。

例**:** クライアントを新しい **URL** にリダイレクトする

次の手順では、CIDR 222.222.0.0/16 内から保護された Web サイトにアクセスするクライアントを指定された URL にリダイレクトします。

Citrix ADC コマンドラインを使用してクライアントをリダイレクトするには:

コマンドプロンプトで次のコマンドを入力してクライアントをリダイレクトし、構成を確認します。

- add responder action act_redirect redirect ”<http://www.example.com/404.html>”
- show responder action act_redirect
- add responder policy pol_redirect “CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)” act_redirect
- show responder policy pol_redirect
- bind responder global pol_redirect 10

例**:**


```
> add responder action act_redirect redirect `" http ://www.example.com
    /404.html "`
Done

> add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET
    (222.222.0.0/16)" act_redirect
Done
<!--NeedCopy-->
```

GUI を使用してクライアントをリダイレクトするには、次の手順を実行します。

1. **AppExpert** > レスポンダー > アクションに移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [レスポンダーアクションの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] テキストボックスに act_redirect と入力します。
   b) [タイプ] で [リダイレクト] を選択します
   c) [ターゲット] テキスト領域に、次の文字列を入力します。"<http://www.example.com/404.html>"
   d) [作成] をクリックし、[閉じる] をクリックします。
      act_redirect という名前のレスポンダーアクションが [レスポンダーアクション] ページに表示されます。
4. ナビゲーションペインで、[**Policies**] をクリックします。
5. 詳細ペインで、[**Add**] をクリックします。
6. [レスポンダーポリシーの作成] ダイアログボックスで、次の操作を行います。
   a) [名前] テキストボックスに pol_redirect と入力します。
   b) [アクション] で [act_redirect] を選択します。
   c) [式] ウィンドウで、次のルールを入力します。CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)
   d) [作成] をクリックし、[閉じる] をクリックします。
      構成した pol_redirect という名前のレスポンダーポリシーが [レスポンダーポリシー] ページに表示されます。
7. レスポンダーポリシーのバインドの説明に従って、新しいポリシー pol_redirect をグローバルにバインドします。

レスポンダの直径サポート

October 7, 2021

レスポンダー機能で Diameter プロトコルがサポートされるようになりました。応答側は、HTTP および TCP 要求と同様に Diameter 要求に応答するように設定できます。たとえば、特定の Diameter オリジンからの要求に応答

するように Responder を構成し、モバイルデバイス用に拡張された Web ページへのリダイレクトを設定できます。Diameter ヘッダーと属性値ペア（AVP）の検査をサポートする多数の Citrix ADC 式が追加されました。これらの式は、インデックス、ID、または名前による特定の AVP のルックアップをサポートし、各 AVP の情報を調べ、適切な応答を送信します。

Diameter 要求に応答するようにレスポンダーを構成するには、次の手順に従います。

コマンドプロンプトで、次のコマンドを入力します。

- `add responder action <actname> RESPONDWITH "DIAMETER.NEW_REDIRECT(\"aaa://host.example.com\")"`

  <actname>には、新しいアクションの名前を置き換えます。名前は、1 ～127 文字の長さで構成され、文字、数字、およびハイフン (-) およびアンダースコア (_) 記号を使用できます。aaa://host.example.comには、接続をリダイレクトする Diameter ホストの URL を代入します。

- add responder policy <polname> “diameter.req.avp(264).value.eq(“host1.example.net”)” <actname>

  <polname>では、新しいポリシーの名前に置き換えます。<actname>と同様に、名前は 1 ～127 文字の長さで構成され、文字、数字、ハイフン (-) およびアンダースコア (_) 記号を含めることができます。host1.example.net の場合は、リダイレクトする要求の発信元ホストの名前を置き換えます。<actname>には、作成したアクションの名前を置き換えます。

- `bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST`

  <vservername>には、ポリシーをバインドするロードバランシング仮想サーバの名前を置き換えます。<polname>には、作成したポリシーの名前を置き換えます。<priority>では、ポリシーの優先度を置き換えます。

例:

「host1.example.net」から発信された Diameter 要求に応答するレスポンダーアクションとポリシーを作成して、「host.example.com」にリダイレクトするには、次のアクションとポリシーを追加し、図のようにポリシーをバインドします。

```
1 > add responder action act_resp-dm-redirect RESPONDWITH "DIAMETER.
     NEW_REDIRECT("aaa://host.example.com")"
2 Done
3
4 > add responder pol_resp-dm-redirect "diameter.req.avp(264).value.eq("
     host1.example.net")" act_resp-dm-redirect
5 Done
6
7 > bind lb vserver vs1 -policyName pol_resp-dm-redirect -priority 10 -
     type REQUEST
```



```
Done
<!--NeedCopy-->
```

レスポンダーに対する **RADIUS** のサポート

October 7, 2021

Citrix ADC の式言語には、RADIUS 要求から情報を抽出して操作できる式が含まれています。これらの式を使用すると、レスポンダー機能を使用して RADIUS 要求に応答できます。レスポンダのポリシーとアクションは、RADIUS 要求に適切または関連する任意の式を使用できます。使用可能な式を使用すると、RADIUS メッセージタイプを識別し、接続から任意の属性値ペア（AVP）を抽出し、その情報に基づいて異なる応答を送信できます。RADIUS 接続のすべての応答側ポリシーを呼び出すポリシーラベルを作成することもできます。

RADIUS 式を使用すると、要求の送信先の RADIUS サーバとの通信を必要としない単純な応答を作成できます。レスポンダーポリシーが接続と一致すると、Citrix ADC は RADIUS 認証サーバーに接続せずに適切な RADIUS レスポンスを構築して送信します。たとえば、RADIUS 要求の送信元 IP アドレスがレスポンダーポリシーで指定されたサブネットからのものである場合、Citrix ADC はその要求に対してアクセス拒否メッセージで応答するか、単に要求をドロップできます。

ポリシーラベルを作成して、特定のタイプの RADIUS 要求を、それらの要求に適した一連のポリシーを通してルーティングすることもできます。

注：現在の RADIUS 式は、RADIUS IPv6 属性では機能しません。

RADIUS をサポートする式に関する Citrix ADC マニュアルでは、RADIUS 通信の基本構造と目的を理解していることを前提としています。RADIUS の詳細については、RADIUS サーバのマニュアルを参照するか、RADIUS プロトコルの概要をオンラインで検索してください。

RADIUS のレスポンダーポリシーの構成

以下の手順では、Citrix ADC コマンドラインを使用してレスポンダーのアクションとポリシーを構成し、RADIUS 固有のグローバルバインドポイントにポリシーをバインドします。

レスポンダーのアクションとポリシーを構成し、ポリシーをバインドするには、次の手順に従います。

コマンドプロンプトで、次のコマンドを入力します。

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>`
  ここで、`<bindPoint>`は RADIUS 固有のグローバルバインドポイントの 1 つを表します。

レスポンダの **RADIUS** 式

レスポンダーの構成では、次の Citrix ADC 式を使用して、RADIUS 要求のさまざまな部分を参照できます。

接続タイプの識別**:**

- RADIUS.IS_CLIENT。接続が RADIUS クライアント (要求) メッセージである場合は TRUE を返します。
- RADIUS.IS_SERVER 接続が RADIUS サーバー (応答) メッセージである場合は TRUE を返します。

リクエスト式**:**

- `RADIUS.REQ.CODE`。RADIUS 要求タイプに対応する番号を返します。num_at クラスの導関数です。たとえば、RADIUS アクセス要求は 1 を返します。RADIUS アカウンティング要求は 4 を返します。
- `RADIUS.REQ.LENGTH`。ヘッダーを含む RADIUS 要求の長さを返します。num_at クラスの導関数です。
- `RADIUS.REQ.IDENTIFIER`。RADIUS 要求 ID を返します。この番号は、各要求に割り当てられた番号で、要求を対応する応答に一致させることができます。num_at クラスの導関数です。
- `RADIUS.REQ.AVP(<AVP Code No>).VALUE`。型 text_t の文字列として、この AVP の最初の出現の値を返します。
- `RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)`。AVP の指定されたインスタンスを RAVP_t 型の文字列として返します。特定の RADIUS AVP は、RADIUS メッセージ内で複数回発生する可能性があります。INSTANCE (0) は最初のインスタンスを返し、INSTANCE (1) は 2 番目のインスタンスを返します。以下同様に最大 16 個のインスタンスを返します。
- `RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)`。AVP の指定されたインスタンスの値を text_t 型の文字列として返します。
- `RADIUS.REQ.AVP(<AVP code no>).COUNT`。RADIUS 接続内の特定の AVP のインスタンス数を整数で返します。
- `RADIUS.REQ.AVP(<AVP code no>).EXISTS`。指定されたタイプの AVP がメッセージ内に存在する場合は TRUE、存在しない場合は FALSE を返します。

レスポンス式**:**

RADIUS 応答式は、REQ を置き換える点を除いて、RADIUS 要求式と同じです。

AVP 値の型キャスト**:**

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィクス、時刻の各データ型にタイプキャストする式をサポートしています。構文は、他の Citrix ADC 型キャスト式と同じです。

例**:**

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィクス、時刻の各データ型にタイプキャストする式をサポートしています。構文は、他の Citrix ADC 型キャスト式と同じです。


```
RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
<!--NeedCopy-->
```

AVP タイプ式**:**

Citrix ADC は、RFC2865 および RFC2866 に記載されている割り当てられた整数コードを使用して、RADIUS AVP 値を抽出する式をサポートしています。テキストエイリアスを使用して、同じタスクを実行することもできます。次に、いくつかの例を示します。

- RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value. RADIUS ユーザ名の値を抽出します。
- RADIUS.REQ.AVP (4)。VALUE or RADIUS.REQ. ACCT_SESSION_ID.value. メッセージからアカウントセッション ID AVP（コード 44）を抽出します。
- RADIUS.REQ.AVP (26)。VALUE or RADIUS.REQ.VENDOR_SPECIFIC.VALUE. ベンダー固有の値を抽出します。

最も一般的に使用される RADIUS AVP の値は、同じ方法で抽出できます。

RADIUS バインドポイント**:**

RADIUS 式を含むポリシーには、4 つのグローバルバインドポイントを使用できます。

- RADIUS_REQ_OVERRIDE. 優先度/優先要求ポリシーキュー。
- RADIUS_REQ_DEFAULT. 標準要求ポリシーキュー。
- RADIUS_RES_OVERRIDE. 優先度/優先応答ポリシーキュー。
- RADIUS_RES_DEFAULT. 標準応答ポリシーキュー。

RADIUS 応答固有の式:

- RADIUS_RESPONDWITH. 指定された RADIUS 応答で応答します。応答は、RADIUS 式と適用可能なその他の式の両方で、Citrix ADC 式を使用して作成されます。
- RADIUS.NEW_ANSWER. 新しい RADIUS 応答をユーザに送信します。
- RADIUS.NEW_ACCESSREJECT. RADIUS 要求を拒否します。
- RADIUS.NEW_AVP. 指定された新しい AVP を応答に追加します。

使用例

次に、レスポンダを使用した RADIUS の使用例を示します。

特定のネットワークからの **RADIUS** 要求のブロック

特定のネットワークからの認証要求をブロックするように応答側機能を設定するには、まず要求を拒否する応答側アクションを作成します。ブロックするネットワークからの要求を選択するポリシーでアクションを使用します。次のように指定して、応答側ポリシーを RADIUS 固有のグローバルバインドポイントにバインドします。

- 優先度

- nextExpr 値として END を指定すると、このポリシーが一致したときにポリシー評価が停止します。
- RADIUS_REQ_OVERRIDE は、ポリシーを割り当てるキューです。これにより、デフォルト・キューに割り当てられたポリシーの前に評価されます。

特定のネットワーク ** からのログオンをブロックするように Responder を構成するには

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder global <polName> <priority> <nextExpr> -type <bindPoint>`

例:

```
> add responder action rspActRadiusReject respondwith radius.
   new_accessreject
Done

> add responder policy rspPolRadiusReject client.ip.src.in_subnet
   (10.224.85.0/24) rspActRadiusReject
Done

> bind responder global rspPolRadiusReject 1 END -type
   RADIUS_REQ_OVERRIDE
<!--NeedCopy-->
```

レスポンダー機能に対する **DNS** サポート

October 7, 2021

応答側機能は、HTTP および TCP 要求と同様に DNS 要求に応答するように設定できます。たとえば、UDP 経由で DNS 応答を送信するように設定し、クライアントからの DNS 要求が TCP 経由で送信されるように設定できます。多くの Citrix ADC 式は、リクエスト内の DNS ヘッダーの検査をサポートしています。これらの式は、特定のヘッダーフィールドを調べ、適切な応答を送信します。

- **DNS** 式。レスポンダー構成では、次の Citrix ADC 式を使用して、DNS 要求のさまざまな部分を参照できます。

式	説明
DNS.NEW_RESPONSE	要求に基づいて、新しい空の DNS 応答を作成します。
DNS.NEW_RESPONSE <AA, TC, rcode>	指定されたパラメータに基づいて新しい DNS 応答を作成します。

- **DNS** バインドポイント。DNS 式を含むポリシーでは、次のグローバルバインドポイントを使用できます。

バインドポイント	説明
DNS_REQ_OVERRIDE	優先度/優先要求ポリシーキュー。
DNS_REQ_DEFAULT	標準要求ポリシーキュー。

デフォルトのバインドポイントに加えて、タイプ DNS のポリシーラベルを作成し、それらに DNS ポリシーをバインドできます。

DNS のレスポンダーポリシーの構成

以下の手順では、Citrix ADC コマンドラインを使用してレスポンダーのアクションとポリシーを構成し、レスポンダー固有のグローバルバインドポイントにポリシーをバインドします。

DNS 要求に応答するようにレスポンダーを構成するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

1. `add responder action <actName> <actType>`

   <actname>には、新しいアクションの名前を置き換えます。名前の長さは 1 ～127 文字で、文字、数字、ハイフン (-)、アンダースコア (_) を使用できます。<actType>には、レスポンダアクションタイプ *respondWith* を置き換えます。

2. `add responder policy <polName> <rule> <actName>`

   <polname>では、新しいポリシーの名前に置き換えます。<actname>では、名前は 1 ～127 文字で、文字、数字、ハイフン（-）、およびアンダースコア（_）を使用できます。<actname>には、作成したアクションの名前を置き換えます。

3. `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>`

   <bindPoint>には、応答側固有のグローバルバインドポイントの 1 つを指定します。<polName>には、作成したポリシーの名前を置き換えます。<priority>に、ポリシーのプライオリティを指定します。

設定例**-TCP** 経由ですべての **DNS** 要求を強制します。

TCP 経由ですべての DNS 要求を適用するには、TC ビットと rcode を NOERROR として設定するレスポンダーアクションを作成します。

```
> add responder action resp_act_set_tc_bit respondwith DNS.NEW_RESPONSE
    (true, true, NOERROR)
Done

```



```
> add responder policy enforce_tcp  dns.REQ.TRANSPORT.EQ(udp)
   resp_act_set_tc_bit
Done

>bind lb vserver dns_udp  - policyName  enforce_tcp   -type request  -
   priority 100
Done
<!--NeedCopy-->
```

レスポンダーの **MQTT** サポート

October 7, 2021

レスポンダー機能は MQTT プロトコルをサポートします。着信 MQTT メッセージのパラメーターに基づいてアクションを実行するようにレスポンダーポリシーを構成できます。

アクションは、新しい接続に対して次のいずれかで応答します。

- DROP
- RESET
- NOOP
- 新しい MQTT CONNACK 応答を開始するためのレスポンダーアクション。

MQTT のレスポンダーポリシーの構成

レスポンダー機能を有効にした後、MQTT リクエストを処理するための 1 つ以上のアクションを構成する必要があります。次に、レスポンダーポリシーを構成します。レスポンダーポリシーをグローバルにバインドすることも、特定の負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドすることもできます。

レスポンダーポリシーをグローバルにバインドするには、次のバインドポイントを使用できます。

- MQTT_REQ_DEFAULT
- MQTT_REQ_OVERRIDE
- MQTT_JUMBO_REQ_DEFAULT
- MQTT_JUMBO_REQ_OVERRIDE

次のバインドポイントを使用して、レスポンダーポリシーをコンテンツスイッチングまたは負荷分散仮想サーバーにバインドできます。

- REQUEST
- MQTT_JUMBO_REQ （このバインドポイントはジャンボパケットにのみ使用されます）

CLI を使用して MQTT リクエストに応答するようにレスポンダーを構成するには

コマンドプロンプトで、次のコマンドを入力します。

レスポンダーのアクションを構成します。

```
add responder action <actName> <actType>
<!--NeedCopy-->
```

- actnameには、新しいアクションの名前を置き換えます。名前の長さは 1～127 文字で、文字、数字、ハイフン（-）、およびアンダースコアを含めることができます。(_) シンボル。
- actTypeの代わりに、レスポンダーアクションタイプ、respondwith を使用します。

例:

```
add responder action mqtt_connack_unsup_ver respondwith MQTT.
   NEW_CONNACK(132)
<!--NeedCopy-->
```

レスポンダーポリシーを構成します。Citrix ADC アプライアンスは、このレスポンダーポリシーによって選択された MQTT リクエストに応答します。

```
add responder policy <polName> <rule> <actname>
<!--NeedCopy-->
```

- polnameでは、新しいポリシーの名前に置き換えます。
- actnameは、作成したアクションの名前に置き換えてください。

例:

```
add responder policy reject_lower_version "MQTT.HEADER.COMMAND.EQ(
   CONNECT) && MQTT.VERSION.LT(3)" mqtt_connack_unsup_ver
<!--NeedCopy-->
```

レスポンダーポリシーを特定の負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドします。ポリシーは、宛先 IP アドレスがその仮想サーバーの VIP である MQTT リクエストにのみ適用されます。

```
bind lb vserver <name> -policyName <policy_name> -priority <priority>

```



```
3 bind cs vserver <name> -policyName <policy_name> -priority <priority>
4 <!--NeedCopy-->
```

- policy_nameは、作成したポリシーの名前に置き換えてください。
- priorityに、ポリシーのプライオリティを指定します。

例:

```
1 bind lb vserver lb1 -policyName reject_lower_version -priority 50
2
3 bind cs vserver mqtt_frontend_cs -policyName reject_lower_version -
     priority 5
4 <!--NeedCopy-->
```

使用例 1: ユーザー名またはクライアント ID に基づいてクライアントをフィルタリングする

管理者は、MQTT CONNECT メッセージのユーザー名またはクライアント ID に基づいて接続を拒否するように MQTT レスポンダーポリシーを構成できます。

クライアント ID に基づいてクライアントをフィルタリングするためのサンプル構成

```
1 add policy patset filter_clients
2 bind policy patset filter_clients client1
3
4 add responder action mqtt_connack_invalid_client respondwith MQTT.
     NEW_CONNACK(2)
5
6 add responder policy reject_clients "MQTT.HEADER.COMMAND.EQ(CONNECT) &&
      mqtt.connect.clientid.equals_any("filter_clients")"
     mqtt_connack_invalid_client
7
8 bind cs vserver mqtt_frontend_cs -policyName reject_clients -priority 5
9 <!--NeedCopy-->
```

使用例 2: ジャンボパケットを処理するために MQTT メッセージの最大メッセージ長を制限する

管理者は、メッセージの長さが特定のしきい値を超えた場合にクライアント接続をドロップするように MQTT レスポンダーポリシーを構成するか、要件に基づいて必要なアクションを実行できます。

ジャンボパケットを処理するために、次のルールパターンのいずれかを持つレスポンダーポリシーがジャンボバインドポイントにバインドされます。

- MQTT.MESSAGE_LENGTH
- MQTT.COMMAND
- MQTT.FROM_CLIENT
- MQTT.FROM_SERVER

ジャンボバインドポイントにバインドされたポリシーは、ジャンボパケットに対してのみ評価されます。

MQTT メッセージの最大メッセージ長を制限するためのサンプル構成

```
set lb parameter -dropmqttjumbomessage no

add responder policy drop_large_message MQTT.MESSAGE_LENGTH.GT(100000)
    reset

bind cs vserver mqtt_frontend_cs -policyName drop_large_message -
    priority 10
<!--NeedCopy-->
```

この例では、dropmqttjumbomessage パラメーターは NO に設定されています。したがって、ADC アプライアンスは、64,000 バイトより長く 1,00,000 バイト未満の長さのメッセージを処理します。長さが 1,00,000 バイトを超えるメッセージはリセットされます。

レスポンダを使用して **HTTP** リクエストを **HTTPS** にリダイレクトする方法

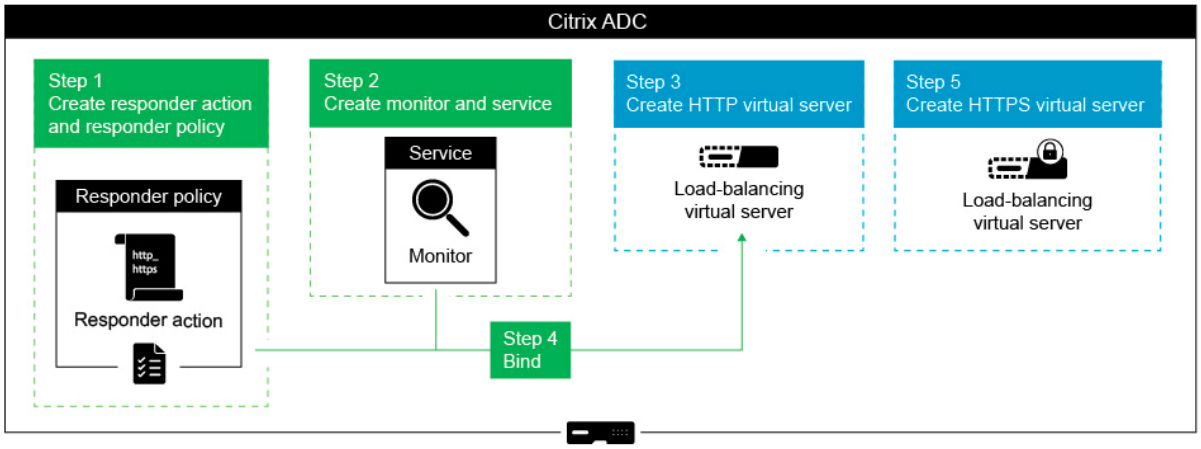
October 7, 2021

この記事では、ロードバランシング仮想サーバーの IP アドレスを使用してレスポンダー機能を設定し、クライアント要求を HTTP から HTTPS にリダイレクトする方法について説明します。

ユーザーが HTTP リクエストを送信してセキュリティで保護された Web サイトにアクセスしようとするシナリオを考えてみましょう。リクエストを削除する代わりに、リクエストをセキュアな Web サイトにリダイレクトすることもできます。レスポンダ機能を使用すると、ユーザーがアクセスしようとするパスと URL クエリを変更することなく、セキュリティで保護された Web サイトに要求をリダイレクトできます。

Citrix ADC レスポンダーが要求を **HTTP** から **HTTPS** にリダイレクトする方法

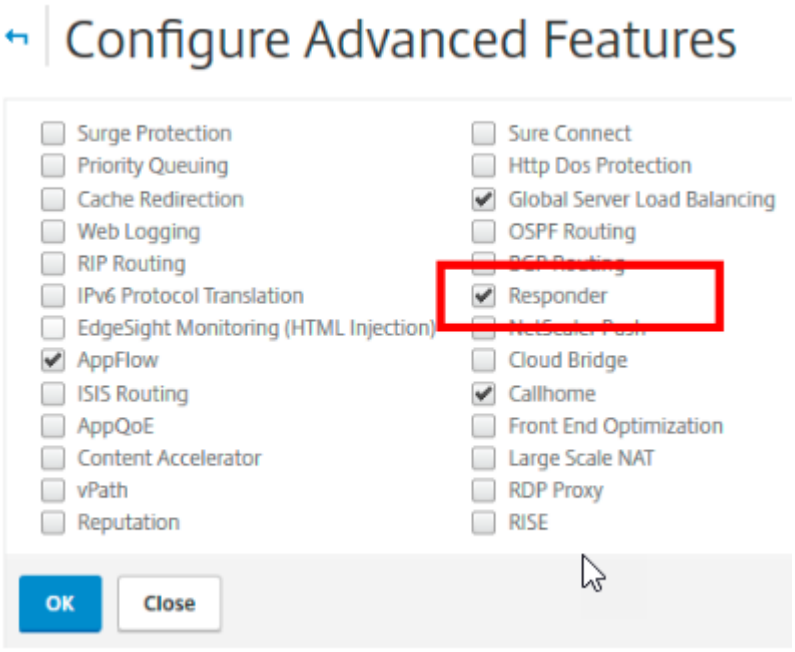
次の図は、アプライアンスが要求をリダイレクトする方法のステップバイステップフローを示しています。



> 注：ナビゲーションパスとスクリーンショットは NetScaler 11.0 から取得されます。

レスポンダー機能を NetScaler アプライアンスの負荷分散 VIP アドレスとともに構成し、クライアント要求を HTTP から HTTPS にリダイレクトするには、次の手順を実行します。

1. アプライアンスでレスポンダ機能を有効にします。[システム] > [設定] > [高度な機能の構成] > [レスポンダー] に移動します。



2. レスポンダアクションを作成し、「名前」フィールドに適切な名前（http_to_https_actn など）を指定します。
3. 応答側アクションを作成するには、ナビゲーションペインで [**AppExpert**] > [応答側] を展開し、[アクション] をクリックして [追加] をクリックします。
4. [タイプとしてリダイレクト] を選択します。

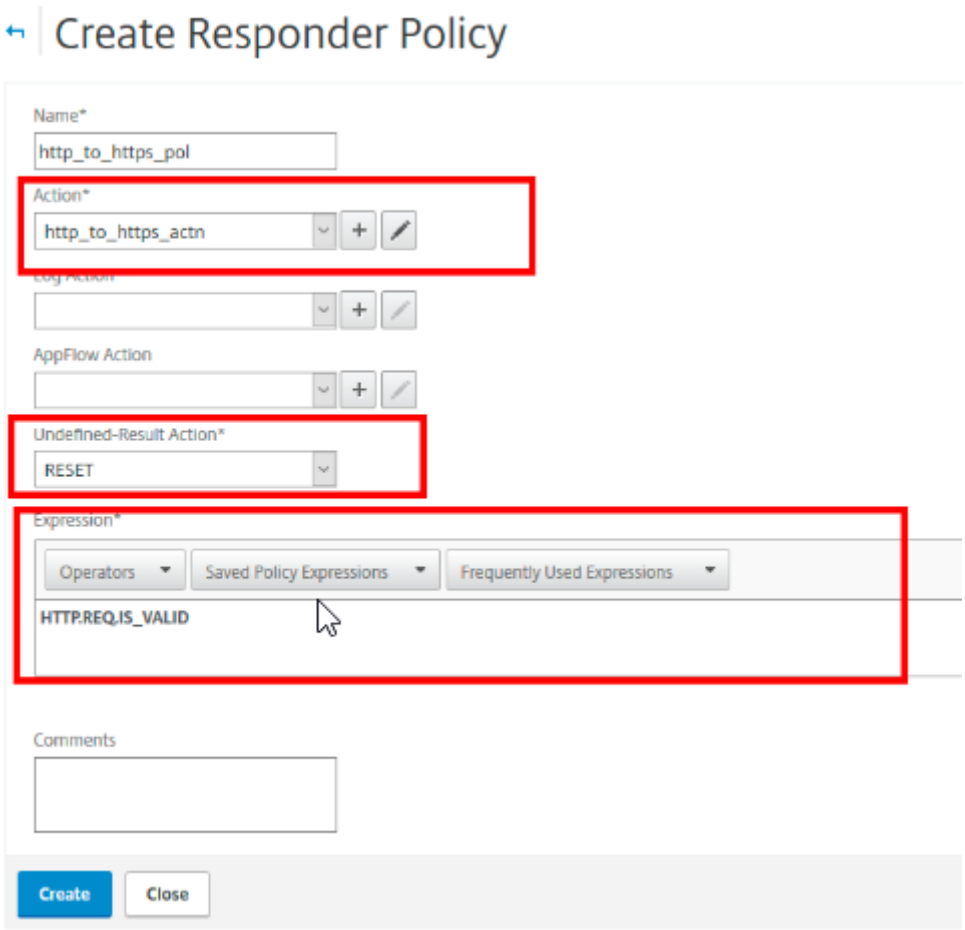
5. [式] フィールドに、次の式を入力します。

   ```
   "https://"+ HTTP.REQ.HOSTNAME.HTTP_URL_SAFE + HTTP.REQ.URL.PATH_AND_QUERY
   .HTTP_URL_SAFE。
   ```

6. NetScaler バージョン 9.0 および 10.0 では、[安全性チェックのバイパス] オプションがクリアされていることを確認します。

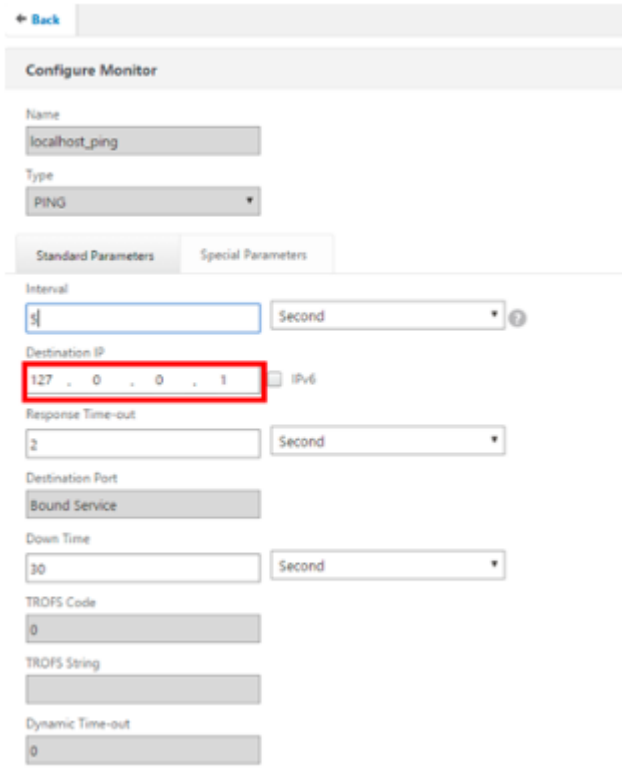
   注：このオプションは、NetScaler 11.0 以降では存在しません。

7. レスポンダーポリシーを作成し、[名前] フィールドに http_to_https_pol などの適切な名前を指定します。
8. レスポンダーポリシーを作成するには、ナビゲーションウィンドウで [**AppExpert**] > [レスポンダー] を展開し、[ポリシー] をクリックして、[追加] をクリックします。
9. 「アクション」(Action) リストから、作成したアクション名を選択します。
10. [未定義のアクション] リストから、[リセット] を選択します。
11. 次のスクリーンショットに示すように、[式] フィールドに **HTTP.REQ.IS_VALID** 式を入力します。

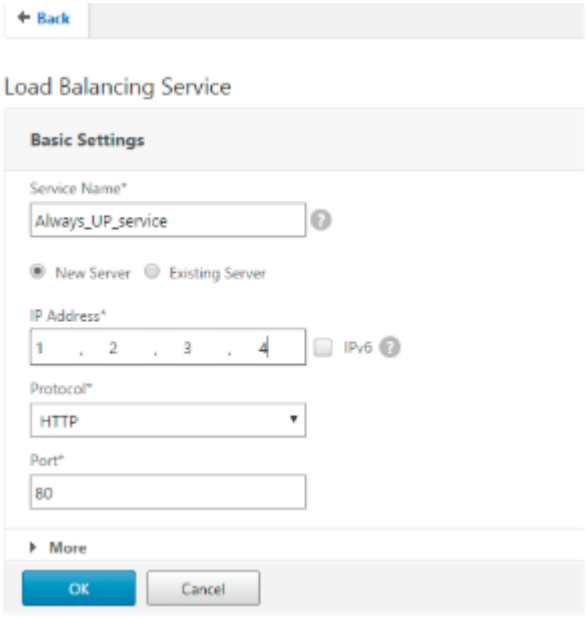


1. ステータスが常に UP としてマークされるモニタを作成し、「名前」フィールドに適切な名前 (localhost_ping など) を指定します。

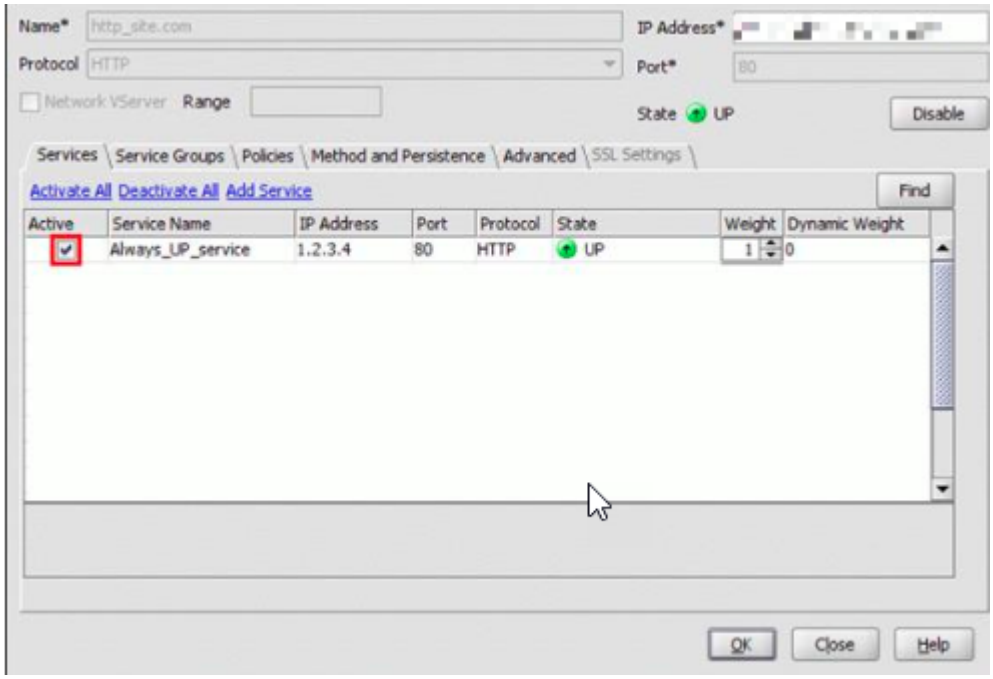
2. モニターを作成するには、ナビゲーションウィンドウで [負荷分散] を展開し、[モニター] をクリックし、[追加] をクリックします。
3. [宛先 **IP**] フィールドで、次のスクリーンショットに示すように 127.0.0.1 の IP アドレスを指定します。



4. サービスを作成し、「名前」フィールドに適切な名前（**Always_UP_service** など）を指定します。
5. サービスを作成するには、ナビゲーションウィンドウで [負荷分散] を展開し、[サービス] をクリックし、[追加] をクリックします。
6. [**Server**] フィールドに存在しない IP アドレスを指定します。



7. [ポート] フィールドに 80 を指定します。
8. 作成したモニターを [使用可能なモニター] リストから追加します。
9. ロード・バランシング仮想サーバーを作成し、「名前」フィールドに適切な名前を指定します。
10. 負荷分散仮想サーバーを作成するには、ナビゲーションウィンドウで、[負荷分散] を展開し、[サービス] をクリックし、[追加] をクリックします。
11. [IP アドレス] フィールドに Web サイトの IP アドレスを指定します。
12. [プロトコル] リストから [HTTP] を選択します。
13. [ポート] フィールドに 80 と入力します。
14. NetScaler バージョン 9.0 および 10.0 では、次のスクリーンショットに示すように、「サービス」タブで作成したサービスの「アクティブ」オプションを選択します。このオプションは、NetScaler バージョン 11.0 では推奨されません。



15. ［ポリシー］タブをクリックします。
16. 作成したレスポンダーポリシーを、Web サイトの HTTP ロードバランシング VIP アドレスにバインドします。
17. Web サイトの IP アドレスと 443 ポートを持つ、セキュアな負荷分散仮想サーバーを作成します。

アプライアンスのコマンドラインインターフェイスから前述の手順と同様の構成を作成するには、次のコマンドを実行します。

```
enable ns feature responder
add responder action http_to_https_actn redirect ””https://” + http.req
    .hostname.HTTP_URL_SAFE + http.REQ.URL.PATH_AND_QUERY.HTTP_URL_SAFE”
add responder policy http_to_https_pol HTTP.REQ.IS_VALID
    http_to_https_actn RESET
add lb monitor localhost_ping PING -LRTM ENABLED -destIP 127.0.0.1
add service Always_UP_service 1.2.3.4 HTTP 80 -gslb NONE -maxClient 0 -
    maxReq 0 -cip ENABLED dummy -usip NO -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP YES
bind lb monitor localhost_ping Always_UP_service
add lb vserver http_site.com HTTP 10.217.96.238 80 -persistenceType
    COOKIEINSERT -timeout 0 -cltTimeout 180
bind lb vserver http_site.com Always_UP_service
bind lb vserver http_site.com -policyName http_to_https_pol -priority 1
     -gotoPriorityExpression END
<!--NeedCopy-->
```


> 注意:
>
> - リダイレクトが機能するには、ポート 80 負荷分散リダイレクト仮想サーバーのステータスが UP である必要があります。
> - HTTPS 仮想サーバーがアクティブでない場合、Web ブラウザーが正しくリダイレクトされないことがあります。
> - このリダイレクト設定により、複数のドメインが同じ IP アドレスにバインドされている状況に対応できます。
> - クライアントが無効な HTTP 要求をリダイレクト仮想サーバに送信すると、アプライアンスは RESET メッセージ・コードを送信します。

トラブルシューティング

October 7, 2021

レスポンダー機能を構成した後も期待どおりに動作しない場合は、一般的なツールを使用して Citrix ADC リソースにアクセスし、問題を診断できます。

トラブルシューティングのリソース

最適な結果を得るには、次のリソースを使用して、Citrix ADC アプライアンスの統合キャッシュに関する問題のトラブルシューティングを行います。

- ns.conf ファイル
- クライアントと Citrix ADC アプライアンスからの関連するトレースファイル

上記のリソースに加えて、次のツールもトラブルシューティングを迅速に行えます。

- IEhttpheaders または同様のユーティリティ
- Citrix ADC トレースファイル用にカスタマイズされた Wireshark アプリケーション

レスポンダーの問題のトラブルシューティング

- 問題

  レスポンダー機能は構成されていますが、レスポンダーのアクションが機能していません。

  解像度

  - 機能が有効になっていることを確認します。
  - いずれかのポリシーのヒットカウンタをチェックして、カウンタが増加しているかどうかを確認します。
  - ポリシーとアクションが正しく構成されていることを確認します。
  - アクションとポリシーが適切にバインドされていることを確認します。

  - クライアントと Citrix ADC アプライアンスのパケットトレースを記録し、それらを分析して問題へのポインタを取得します。
  - iehttpHeaters パケットトレースをクライアントに記録し、HTTP 要求と応答を確認して、問題へのポインタを取得します。

- 問題

  メンテナンスページを作成する必要があります。

  解像度

1. サービスと仮想サーバーを構成します。
2. バックアップ仮想サーバに、バインドされたサービスを構成します。これにより、Web サイトのステータスが常に UP として表示されるようになります。
3. バックアップ仮想サーバをバックアップとして使用するように、プライマリ仮想サーバを構成します。
4. 適切なターゲットを持つレスポンダーアクションを作成します。以下は、参考のための例です。

   ```
   add responder action sorry_page respondwith q{ ”HTTP/1.0 200 OK”+”\r\n\
   r\n”+ ”<html><body>Sorry, this page is not available\</body>\</html>”+
   ”\r\n”}
   ```

5. レスポンダーポリシーを作成し、アクションをそれにバインドします。
6. レスポンダポリシーをバックアップ仮想サーバにバインドします。

リライト

October 7, 2021

> 警告
>
> 従来のポリシーを使用したフィルター機能は廃止されました。代替手段として、高度なポリシーインフラストラクチャで書き換え機能とレスポンダー機能の使用をお勧めします。

書き換えとは、Citrix ADC アプライアンスが処理する要求または応答の情報を書き換えることです。書き換えは、Web サイトの実際の構成に関する不必要な詳細を公開することなく、要求されたコンテンツへのアクセスを提供するのに役立ちます。書き換え機能が役に立つ状況については、以下で説明します。

- セキュリティを向上させるため、Citrix ADC はレスポンス本文内のすべての`http://links`を`https://`に書き換えることができます。
- SSL オフロード展開では、応答内の安全でないリンクをセキュアなリンクに変換する必要があります。rewrite オプションを使用すると、Citrix ADC からクライアントへの送信応答にセキュリティで保護されたリンクがあることを確認するために、すべての `http://links`を`https://`に書き換えることができます。

- Web サイトでエラーページを表示する必要がある場合は、既定の 404 エラーページの代わりにカスタムエラーページを表示できます。たとえば、エラーページの代わりに Web サイトのホームページまたはサイトマップを表示すると、訪問者は Web サイトから移動するのではなく、サイトに残ります。
- 新しい Web サイトを起動するが、古い URL を使用する場合は、[書き換え] オプションを使用できます。
- サイト内のトピックに複雑な URL がある場合は、単純で覚えやすい URL（「クール URL」とも呼ばれます）で書き直すことができます。
- Web サイトの URL に既定のページ名を追加できます。たとえば、会社の Web サイトのデフォルトページが`http://www.abc.com/index.php`の場合、ブラウザのアドレスバーに「abc.com」と入力すると、URL を「abc.com/index.php」に書き換えることができます。

書き換え機能を有効にすると、Citrix ADC は HTTP 要求と応答のヘッダーと本文を変更できます。

HTTP 要求と応答を書き換えるには、構成する書き換えポリシーで、プロトコル対応の Citrix ADC ポリシー式を使用できます。HTTP 要求と応答を管理する仮想サーバは、
HTTP または
SSL のタイプである必要があります。HTTP トラフィックでは、次のアクションを実行できます。

- リクエストの URL を変更する
- ヘッダーの追加、変更、削除
- 本文またはヘッダー内の特定の文字列を追加、置換、または削除します。

TCP ペイロードを書き換えるには、ペイロードをバイトの生のストリームと見なします。TCP 接続を管理する各仮想サーバーは、TCP または SSL_TCP タイプである必要があります。TCP 書き換えという用語は、HTTP データではない TCP ペイロードの書き換えを指すために使用されます。TCP トラフィックでは、TCP ペイロードの任意の部分を追加、変更、または削除できます。

書き換え機能の使用例については、書き換えアクションとポリシーの例を参照してください。

[書き換え] オプションと [レスポンダー] オプションの比較

書き換え機能と応答側機能の主な違いは次のとおりです。

レスポンダーは、レスポンスまたはサーバーベースの式には使用できません。レスポンダーは、クライアントパラメーターに応じて、次のシナリオでのみ使用できます。

- 新しい Web サイトまたは Web ページへの http リクエストのリダイレクト
- カスタム応答で応答する
- 要求レベルでの接続の削除またはリセット

レスポンダーポリシーの場合、Citrix ADC はクライアントからの要求を調べ、適用可能なポリシーに従ってアクションを実行し、クライアントに応答を送信し、クライアントとの接続を閉じます。

書き換えポリシーの場合、Citrix ADC はクライアントからの要求またはサーバーからの応答を調べ、適用可能なポリシーに従ってアクションを実行し、トラフィックをクライアントまたはサーバーに転送します。

一般に、Citrix ADC でクライアントまたは要求ベースのパラメーターに基づいて接続をリセットまたは切断する場合は、レスポンダーを使用することをお勧めします。レスポンダーを使用してトラフィックをリダイレクトするか、カスタムメッセージで応答します。HTTP リクエストとレスポンスのデータを操作するには、書き換えを使用します。

書き換えの仕組み

October 7, 2021

書き換えポリシーは、規則とアクションで構成されます。このルールは、書き換えが適用されるトラフィックを決定し、アクションによって Citrix ADC が実行するアクションを決定します。複数の書き換えポリシーを定義できます。ポリシーごとに、バインドポイントと優先順位を指定します。

バインドポイントとは、Citrix ADC がトラフィックを調べて、書き換えポリシーを適用できるかどうかを検証するトラフィックフロー内のポイントを指します。ポリシーを特定の負荷分散またはコンテンツスイッチング仮想サーバーにバインドするか、Citrix ADC で処理されるトラフィック全体にポリシーを適用する場合は、ポリシーをグローバルにできます。これらのポリシーは、グローバルポリシーと呼ばれます。

ユーザー定義のポリシーに加えて、Citrix ADC にはいくつかのデフォルトポリシーがあります。デフォルトポリシーを変更または削除することはできません。

ポリシーを評価するために、Citrix ADC は下記の順序に従います。

- グローバルポリシー
- 特定の仮想サーバにバインドされたポリシー
- デフォルトポリシー

注：Citrix ADC は、ポイントにバインドされている場合にのみ書き換えポリシーを適用できます。

Citrix ADC は、以下の手順で書き換え機能を実装します。

- Citrix ADC アプライアンスは、グローバルポリシーをチェックし、個々のバインドポイントでポリシーをチェックします。
- 複数のポリシーがバインドポイントにバインドされている場合、Citrix ADC は優先度の順にポリシーを評価します。プライオリティが最も高いポリシーが最初に評価されます。各ポリシーを評価した後、ポリシーが TRUE と評価された場合（トラフィックがルールに一致した場合）、ポリシーに関連付けられたアクションが、実行するアクションのリストに追加されます。一致は、ポリシー規則で指定された特性が、評価される要求または応答の特性と一致する場合に発生します。
- どのポリシーでも、アクションに加えて、現在のポリシーが評価された後に評価されるポリシーを指定できます。このポリシーは「式に移動」と呼ばれます。どのポリシーでも、式に移動（gotoPriorityExpr）が指定されている場合、Citrix ADC は式に移動（Go to Expression）ポリシーを評価し、次に優先順位の高いポリシーを無視します。

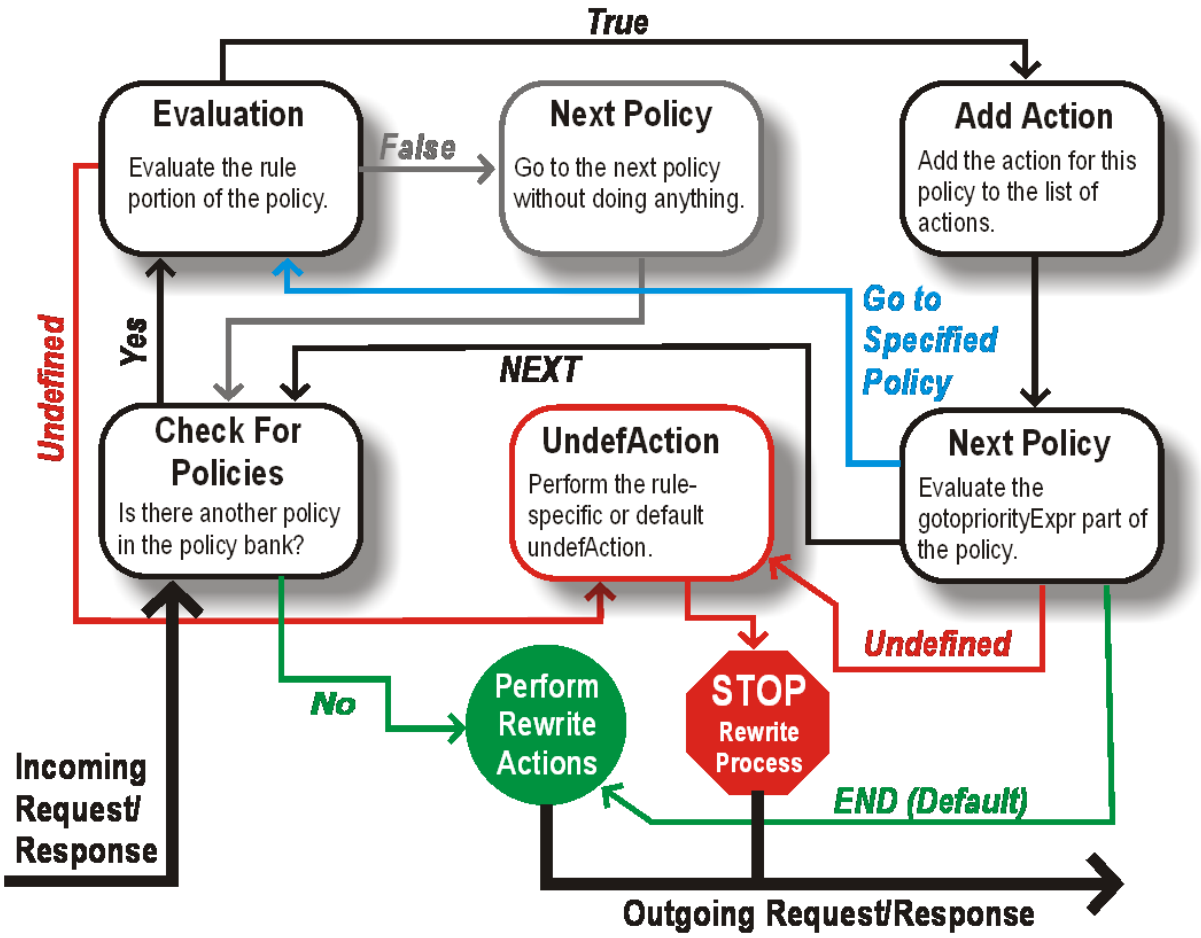
ポリシーのプライオリティを指定して、Go to Expression ポリシーを示すことができます。ポリシーの名前は使用できません。特定のポリシーを評価した後に Citrix ADC で他のポリシーの評価を停止する場合は、「式に移動」を「終了」に設定します。

- すべてのポリシーが評価された後、またはポリシーの「式に移動」が「終了」に設定されている場合、Citrix ADC はアクションのリストに従ってアクションの実行を開始します。

書き換えポリシーの構成の詳細については、書き換えポリシーの構成および書き換えポリシーのバインドについては、「書き換えポリシーのバインド」を参照してください。

次の図は、書き換え機能を使用した場合に、Citrix ADC が要求または応答をどのように処理するかを示しています。

図 1: 書き換えプロセス



ポリシー評価

プライオリティが最も高いポリシーが最初に評価されます。Citrix ADC は、一致するものが見つかっても、リライトポリシーの評価を停止しません。Citrix ADC で設定されているすべてのリライトポリシーを評価します。

- ポリシーが TRUE と評価された場合、Citrix ADC は以下の手順に従います。

- ポリシーで「式に移動」が「終了」に設定されている場合、Citrix ADC は他のすべてのポリシーの評価を停止し、書き換えを開始します。
- gotoPriority 式は、「次へ」、「終了」、いくつかの整数、または「INVOCATION_LIST」に設定できます。値は、次のプライオリティを持つポリシーを決定します。次の表は、式の値ごとに Citrix ADC が実行するアクションを示しています。

式の値	操作（アクション）
NEXT	次のプライオリティを持つポリシーが評価されます。
END	ポリシーの評価が停止します。
<an integer>	指定されたプライオリティのポリシーが評価されます。
INVOCATION_LIST	[次へ移動] または [終了] は、呼び出しリストの結果に基づいて適用されます。

- ポリシーが FALSE と評価された場合、Citrix ADC は優先順位の順で評価を継続します。
- ポリシーが UNDEFINED（エラーのために受信したトラフィックで評価できない）と評価されると、Citrix ADC は UNDEFINED 条件（undefAction と呼ばれる）に割り当てられたアクションを実行し、ポリシーのさらなる評価を停止します。

Citrix ADC は、評価が完了した後にのみ、実際の書き換えを開始します。これは、ポリシーによって識別されるアクションのリストを参照し、TRUE と評価され、書き換えを開始します。リスト内のすべてのアクションを実装した後、Citrix ADC は必要に応じてトラフィックを転送します。

> 注:
>
> ポリシーで、HTTP ヘッダーまたは本文、または TCP ペイロードの同じ部分で競合または重複するアクションが指定されていないことを確認します。このような競合が発生すると、Citrix ADC は未定義の状況を検出し、書き換えを中止します。

リライトアクション

Citrix ADC アプライアンスで、本文内のテキストの追加、置換、削除、ヘッダーの追加、変更、削除など、TCP ペイロードの変更をリライトアクションとして実行するアクションを指定します。書き換えアクションの詳細については、「書き換えアクションの設定」を参照してください。

次の表では、ポリシーが TRUE と評価されたときに Citrix ADC が実行できる手順について説明します。

操作（アクション）	結果
Ins	ポリシーに指定された書き換えアクションが実行されます。

操作（アクション）	結果
NOREWRITE	要求または応答は書き換えられません。Citrix ADC は、メッセージの一部を書き換えることなくトラフィックを転送します。
RESET	接続は TCP レベルで中断されます。
DROP	メッセージはドロップされます。

> 注:
>
> どのポリシーでも、undefaction（ポリシーが UNDEFINED に評価されたときに実行されるアクション）を NOREWRITE、RESET、または DROP として設定できます。

書き換え機能を使用するには、
次の手順を実行します。

- Citrix ADC でこの機能を有効にします。
- 書き換えアクションを定義します。
- 書き換えポリシーを定義します。
- ポリシーをバインドポイントにバインドして、ポリシーを有効にします。

書き換え機能の有効化

October 7, 2021

HTTP または TCP 要求または応答を書き換える場合は、Citrix ADC アプライアンスで書き換え機能を有効にします。この機能が有効になっている場合、Citrix ADC は指定されたポリシーに従って書き換えアクションを実行します。詳細については、「書き換えの仕組み」を参照してください。

コマンドラインインターフェイスを使用して書き換え機能を有効にするには

コマンドプロンプトで次のコマンドを入力して、書き換え機能を有効にし、構成を確認します。

- enable ns feature REWRITE
- show ns feature

例:

```
1 > enable ns feature REWRITE
2  Done
```



```
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 .
 .
 .
 1)     Rewrite                         REWRITE             ON
 .
 .
 1)     Citrix ADC Push                  push                OFF
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して書き換え機能を有効にするには

1. ナビゲーションウィンドウで、[システム] をクリックし、[設定] をクリックします。
2. 詳細ウィンドウの [モードと機能] で、[基本機能の構成] をクリックします。
3. [基本機能の構成] ダイアログボックスで、[書き換え] チェックボックスをオンにし、[**OK**] をクリックします。
4. [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。ステータスバーに、選択した機能が有効になっていることを示すメッセージが表示されます。

書き換えアクションの設定

October 13, 2021

> 警告
>
> 書き換えアクションのパターン機能は、Citrix ADC 12.0 ビルド 56.20 以降では廃止され、代替として、［検索書き換えアクション］パラメータを使用することをお勧めします。

書き換えアクションは、サーバーまたはクライアントに送信する前に要求または応答に加えられた変更を示します。

式は、次のことを定義しています。

- アクションタイプを書き換えます。
- 書き換えアクションの場所です。
- アクションの設定タイプを書き換えます。

たとえば、DELETE アクションはターゲット式のみを使用します。REPLACE アクションは、ターゲット式と式を使用して置換テキストを設定します。

書き換え機能を有効にした後は、組み込みの書き換えアクションで十分でない限り、1 つ以上のアクションを設定する必要があります。すべての組み込みアクションの名前は、文字列 ns_cvpn で始まり、その後に文字列とアンダースコア文字が続きます。組み込みアクションは、クライアントレス VPN 要求または応答の一部のデコード、JavaScript または XML データの変更など、便利で複雑なタスクを実行します。組み込みアクションは、表示、有効化、および無効化できますが、変更または削除することはできません。

> 注:
>
> HTTP 書き換えにのみ使用できるアクションタイプは、[書き換え アクションタイプ] 列で識別されます。

詳細については、「**Type** パラメータ」を参照してください。

コマンドラインインターフェイスを使用して書き換えアクションを作成する

コマンドプロンプトで次のコマンドを入力して、書き換えアクションを作成し、構成を確認します。

- `add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-search <expression>] [refineSearch <expression>] [-comment<string>]`
- `show rewrite action <name>`

詳細については、書き換えアクションタイプとその引数テーブルを参照してください 。

書き換え機能には、次の組み込みアクションがあります。

- noReWrite-リクエストまたはレスポンスを書き換えずにユーザーに送信します。
- RESET-ユーザーがリクエストを再送信できるように、接続をリセットしてユーザーのブラウザに通知します。
- DROP-ユーザーに応答を送信せずに接続をドロップします。

次のフロータイプの 1 つがすべてのアクションに暗黙的に関連付けられます。

- Request-アクションはリクエストに適用されます。
- Response-アクションは応答に適用されます。
- ニュートラル-アクションはリクエストとレスポンスの両方に適用されます。

名前

ユーザー定義の書き換えアクションの名前。文字、数字、またはアンダースコア文字 (_) で始まり、ハイフン (-)、ピリオド (.) ハッシュ (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア文字のみを含める必要があります。書き換えポリシーの追加後に変更できます。

型パラメータ

Type パラメータには、ユーザー定義の書き換えアクションのタイプが表示されます。

Type パラメータの値は次のとおりです。

- REPLACE <target> <string_builder_expr>。文字列を文字列ビルダー式に置き換えます。

例:

```
> add rewrite action replace_http_act replace http.res.body(100) '"
    new_replaced_data"'
Done
> sh rewrite action replace_http_act
Name: replace_http_act
Operation: replace
Target:http.res.body(100)
Value:"new_replaced_data"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_ALL <target> <string_builder_expr1> -(pattern|search)<string_builder_expr2>。<target>で指定されたリクエストまたはレスポンスで、<string_builder_expr1>で定義されているすべての文字列を、<string_builder_expr2>で定義された文字列に置き換えます。PCRE 形式のパターンまたは検索機能を使用して、置換する文字列を検索できます。

例:

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 "www.abc.com"
Done
> bind policy patset pat_list_2 "www.def.com"
Done
> add rewrite action refineSearch_act_31 replace_all "HTTP.RES.BODY
    (100000)" ""https://""-search "patset("pat_list_2")" -refineSearch "
    EXTEND(7,0).REGEX_SELECT(re#http://#)"
Done

> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:"https://"
```



```
Search: patset(”pat_list_2”)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_HTTP_RES <string_builder_expr>。完全な HTTP レスポンスを文字列ビルダー式で定義された文字列に置き換えます。

例:

```
> add rewrite action replace_http_res_act replace_http_res '”HTTP/1.1 200 OK\r\n\r\nSending from ADC”'
 Done
> sh rewrite action replace_http_res_act
Name: replace_http_res_act
Operation: replace_http_res
Target:”HTTP/1.1 200 OK
Sending from ADC”
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_SIP_RES <target>。SIP 応答全体を、<target>で指定された文字列に置き換えます。

例:

```
> add rewrite action replace_sip_res_act replace_sip_res '”HTTP/1.1 200 OK\r\n\r\nSending from ADC”'
Done
> sh rewrite action replace_sip_res_act
Name: replace_sip_res_act
Operation: replace_sip_res
Target:”HTTP/1.1 200 OK
Sending from ADC”
Hits: 0
Undef Hits: 0
Action Reference Count: 0
```



```
Done

<!--NeedCopy-->
```

- INSERT_HTTP_HEADER <header_string_builder_expr> <contents_string_builder_expr>。で指定された HTTP ヘッダーを挿入します。<header_string_builder_expr> およびヘッダーコンテンツで指定された内容 <contents_string_builder_expr>。

例:

```
> add rewrite action ins_cip_header insert_http_header "CIP" "CLIENT.IP.SRC"
Done
> sh rewrite action ins_cip_header
Name: ins_cip_header
Operation: insert_http_header
Target:CIP
Value:CLIENT.IP.SRC
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_HTTP_HEADER <target>。<target>で指定された HTTP ヘッダーを削除します。

例:

```
> add rewrite action del_true_client_ip_header delete_http_header "True-Client-IP"
Done
> sh rewrite action del_true_client_ip_header
Name: del_true_client_ip_header
Operation: delete_http_header
Target:True-Client-IP
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```


- CORRUPT_HTTP_HEADER <target>。<target>で指定したすべての出現する HTTP ヘッダーのヘッダー名を破損した名前に置き換え、レシーバによって認識されないようにします。例:MY_HEADER は MHY_ADER に変更されます。

例:

```
> add rewrite action corrupt_content_length_hdr corrupt_http_header ”
   Content-Length”
Done
> sh rewrite action corrupt_content_length_hdr
Name: corrupt_content_length_hdr
Operation: corrupt_http_header
Target:Content-Length
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE <string_builder_expr1> <string_builder_expr1>。<string_builder_expr1>で指定された文字列を検索し、その前の<string_builder_expr2>に文字列を挿入します。

```
> add rewrite action insert_before_ex_act insert_before http.res.body
   (100) '”Add this string in the starting”'
Done
> sh rewrite action insert_before_ex_act
Name: insert_before_ex_act
Operation: insert_before
Target:http.res.body(100)
Value:”Add this string in the starting”
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE_ALL <target> <string_builder_expr1> -(pattern|search)<string_builder_expr2>。<target>で指定されたリクエストまたはレスポンスで、<string_builder_expr1>で指定された文字列のすべてのオカレンスを検索し、<string_builder_expr2>で指定された文字列を前に挿入します。PCRE 形式のパターンまた

は検索機能を使用して、文字列を検索できます。

例:

```
> add policy patset pat
 Done
> bind policy patset pat abcd
 Done
> add rewrite action refineSearch_act_1 insert_before_all http.res.body
   (10) 'target.prefix(10) + ”refineSearch_testing”' -search patset(”
   pat”) -refineSearch extend(10,10)
 Done
> sh rewrite action refineSearch_act_1
Name: refineSearch_act_1
Operation: insert_before_all
Target:http.res.body(10)
Refine Search:extend(10,10)
Value:target.prefix(10) + ”refineSearch_testing”
Search: patset(”pat”)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_AFTER <string_builder_expr1> <string_builder_expr2>。で指定された文字列を検索します。<string_builder_expr1> で指定した文字列を挿入します。<string_builder_expr2> それの後に。

例:

```
> add rewrite action insert_after_act insert_after http.req.body(100) '
   ”add this string after 100 bytes”'
Done
> sh rewrite action insert_after_act
Name: insert_after_act
Operation: insert_after
Target:http.req.body(100)
Value:”add this string after 100 bytes”
Hits: 0
Undef Hits: 0
Action Reference Count: 0
```



```
Done

<!--NeedCopy-->
```

- INSERT_AFTER_ALL <target> <string_builder_expr1> -(pattern|search) <string_builder_expr>。<target>で指定されたリクエストまたはレスポンスで、<string_builder_expr1>で指定された文字列のすべての出現箇所を検索し、<string_builder_expr2>で指定した文字列をそれぞれの後に挿入します。PCRE 形式のパターンまたは検索機能を使用して、文字列を検索できます。

例:

```
> add rewrite action refineSearch_act_2 insert_after_all http.res.body
   (100) '"refineSearch_testing"' -search text("abc") -refineSearch
   extend(0, 10)
Done
> sh rewrite action refineSearch_act_2
Name: refineSearch_act_2
Operation: insert_after_all
Target:http.res.body(100)
Refine Search:extend(0, 10)
Value:"refineSearch_testing"
Search: text("abc")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE <target>。指定したターゲットを検索して削除します。

例:

```
> add rewrite action delete_ex_act delete http.req.header("HDR")
Done
> sh rewrite action delete_ex_act
Name: delete_ex_act
Operation: delete
Target:http.req.header("HDR")
Hits: 0
Undef Hits: 0
```



```
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_ALL <target> -(pattern|search)<string_builder_expr>。<target>で指定されたリクエストまたはレスポンスで、<string_builder_expr>で指定された文字列のすべてのオカレンスを検索して削除します。PCRE 形式のパターンまたは検索機能を使用して、文字列を検索できます。

例:

```
>add rewrite action refineSearch_act_4 delete_all "HTTP.RES.BODY(50000)
    " -search text("Windows Desktops") -refineSearch "EXTEND(40,40).
    REGEX_SELECT(re#\s`\*`<AppData>.`\*`\s`\*`<\/AppData>#)"
Done
> show REWRITE action refineSearch_act_4
Name: refineSearch_act_4
Operation: delete_all
Target:HTTP.RES.BODY(50000)
Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s`\*`<AppData>.`\*`\s
    `\*`</AppData>#)
Search: text("Windows Desktops")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DIAMETER_HEADER_FIELD <target> <field value>。リクエストまたはレスポンスで、<target>で指定されたヘッダーフィールドを変更します。Diameter.req.flags.SET(<flag>)またはstringbuilderexpressionとしてのDiameter.req.flags.UNSET<flag>を使用して、フラグを設定または解除します。

例:

```
> add rewrite action replace_diameter_field_ex_act
    replace_diameter_header_field diameter.req.flags diameter.req.flags.
    set(PROXIABLE)
Done
> sh rewrite action replace_diameter_field_ex_act
Name: replace_diameter_field_ex_act
```



```
Operation: replace_diameter_header_field
Target:diameter.req.flags
Value:diameter.req.flags.set(PROXIABLE)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DNS_HEADER_FIELD <target>。リクエストまたはレスポンスで、<target>で指定されたヘッダーフィールドを変更します。

例:

```
> add rewrite action replace_dns_hdr_act replace_dns_header_field dns.
   req.header.flags.set(AA)
Done
> sh rewrite action replace_dns_hdr_act
Name: replace_dns_hdr_act
Operation: replace_dns_header_field
Target:dns.req.header.flags.set(AA)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DNS_ANSWER_SECTION <target>。応答の DNS 応答セクションを置き換えます。これは現在、A および AAAA レコードにのみ適用されます。DNS.NEW_RRSET_AおよびNS.NEW_RRSET_AAAA式を使用して、新しい回答セクションを構成します。

例:

```
> add rewrite action replace_dns_ans_act replace_dns_answer_section
   DNS.NEW_RRSET_A("1.1.1.1", 10)
Done
> sh rewrite action replace_dns_ans_act
Name: replace_dns_ans_act
Operation: replace_dns_answer_section
Target:DNS.NEW_RRSET_A("1.1.1.1", 10)
Hits: 0
```



```
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE<target>。ターゲットの [クライアントレス VPN 形式] で指定されたパターンをデコードします。

例:

```
> add rewrite action cvpn_decode_act_1 clientless_vpn_decode http.req.
   body(100)
Done
> sh rewrite action cvpn_decode_act_1
Name: cvpn_decode_act_1
Operation: clientless_vpn_decode
Target:http.req.body(100)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE_ALL<target>-search<expression>。検索パラメータで指定されたすべてのパターンをクライアントレス VPN 形式でデコードします。

例:

```
> add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
   -search text(”abcd”)
Done
> sh rewrite action act1
Name: act1
Operation: clientless_vpn_decode_all
Target:http.req.body(100)
Search: text(”abcd”)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

```



```
13 <!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE<target>。ターゲットで指定されたパターンをクライアントレス VPN 形式でエンコードします。

例:

```
1 > add rewrite action cvpn_encode_act_1 clientless_vpn_encode http.req.
     body(100)
2 Done
3 > sh rewrite action cvpn_encode_act_1
4 Name: cvpn_encode_act_1
5 Operation: clientless_vpn_encode
6 Target:http.req.body(100)
7 Hits: 0
8 Undef Hits: 0
9 Action Reference Count: 0
10 Done
11
12 <!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE_ALL<target>-search<expression>。指定されたすべてのパターンをクライアントレス VPN 形式でエンコードします。

例:

```
1 > add rewrite action act2 clientless_vpn_encode_all http.req.body(100)
     -search text(”abcd”)
2 Done
3 > sh rewrite action act2
4 Name: act1
5 Operation: clientless_vpn_encode_all
6 Target:http.req.body(100)
7 Search: text(”abcd”)
8 Hits: 0
9 Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

- CORRUPT_SIP_HEADER<target>。<target>で指定したすべての SIP ヘッダーのヘッダー名を破損した名前に置き換え、受信者がそれを認識しないようにします。

例:

```
> add rewrite action corrupt_sip_hdr_act corrupt_sip_header SIP_HDR
Done
> sh rewrite action corrupt_sip_hdr_act
Name: corrupt_sip_hdr_act
Operation: corrupt_sip_header
Target:SIP_HDR
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_SIP_HEADER <header_string_builder_expr> <contents_string_builder_expr >。<header_string_builder_expr>で指定された SIP ヘッダーと、<contents_string_builder_expr >で指定されたヘッダーコンテンツを挿入します。

例:

```
> add rewrite action insert_sip_hdr_act insert_sip_header SIP_HDR '"
   inserting_sip_header"'
 Done
>sh rewrite action insert_sip_hdr_act
Name: insert_sip_hdr_act
Operation: insert_sip_header
Target:SIP_HDR
Value:"inserting_sip_header"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_SIP_HEADER<target>。<target>で指定された SIP ヘッダーを削除します。

例:

```
> add rewrite action delete_sip_hdr delete_sip_header  SIP_HDR
```



```
Done
> sh rewrite action delete_sip_hdr
Name: delete_sip_hdr
Operation: delete_sip_header
Target:SIP_HDR
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

ターゲットパラメータ

Target パラメータリクエストまたはレスポンスのどの部分を書き換えるかを指定する式を指定します。

StringBuilderExpr

StringBuilderExpr 指定された場所のリクエストまたは応答に挿入されるコンテンツを指定する式を指定します。この式は、指定された文字列を置き換えます。

例 **1.** クライアント **IP** を使用した **HTTP** ヘッダーの挿入:

```
> add rewrite action insertact INSERT_HTTP_HEADER ”client-IP” CLIENT.IP
    .SRC
Done
> show rewrite action insertact
Name: insertact
Operation: insert_http_header
Target:Client-IP
Value:CLIENT.IP.SRC
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

例 **2. TCP** ペイロード内の文字列の置換（**TCP** 書き換え）:


```
> add rewrite action client_tcp_payload_replace_all REPLACE_ALL
  'client.tcp.payload(1000)' '”new-string”' -search text(”old-string”)
Done
> show rewrite action client_tcp_payload_replace_all
Name: client_tcp_payload_replace_all
Operation: replace_all
Target:client.tcp.payload(1000)
Value:”new-string”
Search: text(”old-string”)
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

リクエストまたはレスポンスの一部を検索して書き換え

検索機能は、リクエストまたはレスポンスで必要なパターンのすべてのインスタンスを検索するのに役立ちます。

検索機能は、次のアクションタイプで使用する必要があります。

- INSERT_BEFORE_ALL
- INSERT_AFTER_ALL
- REPLACE_ALL
- DELETE_ALL
- CLIENTLESS_VPN_ENCODE_ALL
- CLIENTLESS_VPN_DECODE_ALL

検索機能は、次のアクションタイプでは使用できません。

- INSERT_HTTP_HEADER
- INSERT_BEFORE
- INSERT_AFTER
- REPLACE
- DELETE
- DELETE_HTTP_HEADER
- CORRUPT_HTTP_HEADER
- REPLACE_HTTP_RES
- CLIENTLESS_VPN_ENCODE
- CLIENTLESS_VPN_DECODE
- INSERT_SIP_HEADER

- DELETE_SIP_HEADER
- CORRUPT_SIP_HEADER
- REPLACE_DIAMETER_HEADER_FIELD
- REPLACE_DNS_ANSWER_SECTION
- REPLACE_DNS_HEADER_FIELD
- REPLACE_SIP_RES

次の検索タイプがサポートされています。

- Text-
  リテラル文字列例 ⊠ 検索テキスト (「hello」)
- 正規表現-リクエストまたはレスポンスの複数の文字列を照合するために使用されるパターン例 ⊠ 検索正規表現 (re~^hello*~)
- XPATH-XML を検索する XPATH 式。
  例 ⊠-search xpath(xp%/a/b%)
- JSON-JSON を検索するための XPATH 式。
  例 ⊠ 検索 xpath_json (xp%/a/b%)
  HTML-HTML を検索する XPATH
  式例 ⊠ 検索 xpath_html (xp%/html/body%)
  パッチセット-これはパッチセットエンティティにバインドされたすべてのパターンを検索します。
  例:-search patset(“patset1”)
- Datset-データセットエンティティにバインドされたすべてのパターンを検索します。
  例 ⊠-search dataset(“dataset1”)
- AVP-
  直径/Radius メッセージの例 ⊠ 検索 avp (999) で複数の AVP を照合するために使用される AVP 番号

検索結果を絞り込む

検索の絞り込み機能を使用して、検索結果を絞り込むための追加条件を指定できます。検索の絞り込み機能は、検索機能が使用されている場合にのみ使用できます。
検索の絞り込みパラメータは常に「拡張 (m, n)」操作で始まります。ここで、’m’ は検索結果の左側のバイト数を指定し、’n’ は検索結果の右側のバイト数を指定して、選択した領域を拡張します。

設定されている書き換えアクションが次の場合:

```
1 > add rewrite action test_refine_search replace_all http.res.body(10) '
    ” testing_refine_search” ' -search text(”abc”) -refineSearch extend
    (1,1)
2 And the HTTP response body is abcxxxx456.
```



```
3
4 <!--NeedCopy-->
```

次に、検索パラメータはパターン「abc」を検出し、refineSearch パラメータは一致したパターンの左側に余分な 1 バイト、右側に余分な 1 バイトをチェックするように設定されているためです。結果として置換されるテキストは abcx です。したがって、このアクションの出力は`testing_refine_searchxxx456`です。

例 **1: INSERT_BEFORE_ALL** アクションタイプで **[絞り込み]** 検索機能を使用する。

```
1 > add policy patset pat
2 Done
3 > bind policy patset pat abcd
4 Done
5 > add rewrite action refineSearch_act_1 insert_before_all http.res.body
     (10) 'target.prefix(10) + ”refineSearch_testing”' -search patset(”
     pat”) -refineSearch extend(10,10)
6 Done
7 > sh rewrite action refineSearch_act_1
8 Name: refineSearch_act_1
9 Operation: insert_before_all
10 Target:http.res.body(10)
11 Refine Search:extend(10,10)
12 Value:target.prefix(10) + ”refineSearch_testing”
13 Search: patset(”pat”)
14 Hits: 0
15 Undef Hits: 0
16 Action Reference Count: 0
17 Done
18
19 <!--NeedCopy-->
```

例 **2: INSERT_AFTER_ALL** アクションタイプで「絞り込み」サーチ機能を使用する。

```
1 > add rewrite action refineSearch_act_2 insert_after_all http.res.body
     (100) '”refineSearch_testing”' -search text(”abc”) -refineSearch
     extend(0, 10)
2 Done
3 > sh rewrite action refineSearch_act_2
4 Name: refineSearch_act_2
5 Operation: insert_after_all
6 Target:http.res.body(100)
7 Refine Search:extend(0, 10)
```



```
Value:”refineSearch_testing”
Search: text(”abc”)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

例 **3: REPLACE_ALL** アクションタイプで **[**絞り込み**]** 検索機能を使用する。

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 ”www.abc.com”
Done
> bind policy patset pat_list_2 ”www.def.com”
Done
> add rewrite action refineSearch_act_31 replace_all ”HTTP.RES.BODY
    (100000)” ””https://”” -search ”patset(”pat_list_2”)” -refineSearch
    ”EXTEND(7,0).REGEX_SELECT(re#http://#)”
Done
> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:”https://”
Search: patset(”pat_list_2”)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

例 **4: DELETE_ALL** アクションタイプで **[**検索の絞り込み**]** 機能を使用する。

```
>add rewrite action refineSearch_act_4 delete_all ”HTTP.RES.BODY(50000)
    ” -search text(”Windows Desktops”) -refineSearch ”EXTEND(40,40).
    REGEX_SELECT(re#\s*<AppData>.\*\s\*<\/AppData>#)”
> show REWRITE action refineSearch_act_4
Name: refineSearch_act_4
```



```
4  Operation: delete_all
5  Target:HTTP.RES.BODY(50000)
6  Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s*<AppData>.\*\s\*</
       AppData>#)
7  Search: text(”Windows Desktops”)
8  Hits: 0
9  Undef Hits: 0
10 Action Reference Count: 0
11 Done
12 >
13 <!--NeedCopy-->
```

例 **5: CLIENTLESS_VPN_ENCODE_ALL** アクションタイプで「検索の絞り込み」機能を使用する。

”’

```
add rewrite action act2 clientless_vpn_encode_all http.req.body(100) -search text(“abcd”)
Done
sh rewrite action act2
Name: act1
Operation: clientless_vpn_encode_all
Target:http.req.body(100)
Search: text(“abcd”)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
```

”’

例 **6: CLIENTLESS_VPN_DECODE_ALL** アクションタイプで「検索の絞り込み」機能を使用する。

```
1  > add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
       -search text(”abcd”)
2  Done
3  > sh rewrite action act1
4  Name: act1
5  Operation: clientless_vpn_decode_all
6  Target:http.req.body(100)
7  Search: text(”abcd”)
8  Hits: 0
9  Undef Hits: 0
10 Action Reference Count: 0
11 Done
```



```
12 >
13 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して既存の書き換えアクションを変更する

コマンドプロンプトで次のコマンドを入力して、既存の書き換えアクションを変更し、構成を確認します。

- set rewrite action <name> [-target<expression>] [-stringBuilderExpr<expression>] [-pattern<expression> | -search <expression>] [-refineSearch <expression>] [-comment<string>]
- show rewrite action <name>

例:

```
1 > set rewrite action insertact -target ”Client-IP”
2  Done
3 > show rewrite action insertact
4
5 Name: insertact
6 Operation: insert_http_header   Target:Client-IP
7 Value:CLIENT.IP.SRC
8 Hits: 0
9 Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して書き換えアクションを削除する

コマンドプロンプトで次のコマンドを入力して、書き換えアクションを削除します。

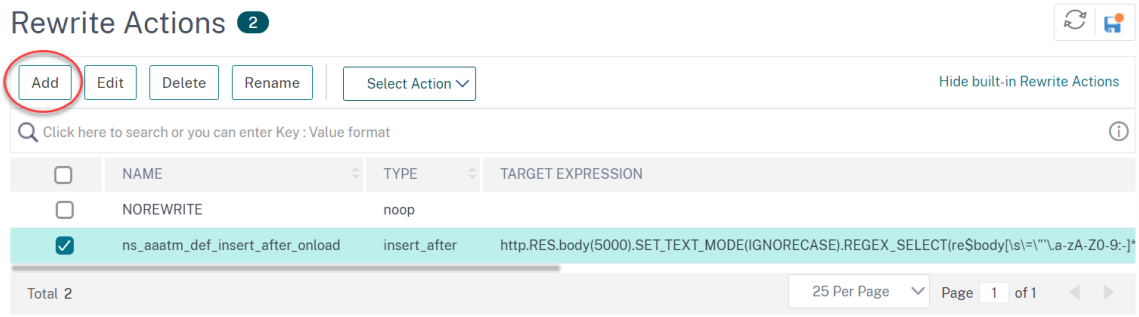
```
rm rewrite action <name>
```

例:

```
1 > rm rewrite action insertact
2 Done
3
4 <!--NeedCopy-->
```


構成ユーティリティを使用して書き換えアクションを構成する

1. [**AppExpert**] > [**Rewrite**] > [**Actions**] の順に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - アクションを作成するには、[追加] をクリックします。
   - 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
3. [作成] または [**OK**] をクリックします。アクションが正常に構成されたことを示すメッセージがステータスバーに表示されます。
4. 手順 2 ～4 を繰り返して、必要な数の書き換えアクションを作成または変更します。
5. [閉じる] をクリックします。



[式の追加] ダイアログボックスを使用して式を追加する

1. [書き換えアクションの作成] または [書き換えアクションの設定] ダイアログボックスで、入力する引数 type のテキスト領域で、[追加] をクリックします。
2. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。
   - HTTP HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
   - SYS。保護された Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - クライアント。要求を送信したコンピュータ。リクエストの送信者の側面を調べる場合は、これを選択します。

選択すると、右端のリストボックスに、式の次の部分に適した用語がリストされます。

1. 2 番目のリストボックスで、式の 2 番目の用語を選択します。選択肢は、前のステップで行った選択によって異なり、コンテキストに適切です。2 番目の選択を行った後、[式の構築] ウィンドウの下のヘルプウィンドウ (空白) に、選択した用語の目的と使用法を説明するヘルプが表示されます。

2. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。
   PI 式の言語およびレスポンダーポリシーの式の作成の詳細については、「ポリシーと式」を参照してください。

サンプル HTTP データに対してリライトアクションを使用した場合のエフェクトをテストする場合は、リライト式エバリュエータを使用できます。

TCP ペイロードを書き換え

TCP 書き換えのアクション内のターゲット式は、次のいずれかの式プレフィックスで開始する必要があります。

- **CLIENT.TCP.PAYLOAD**。クライアント要求の TCP ペイロードを書き換えるため。たとえば、CLIENT.TCP.PAYLOAD (10000) .AFTER_STR (“string1”) などです。
- **SERVER.TCP.PAYLOAD**。サーバー応答の TCP ペイロードを書き換えるため。たとえば、SERVER.TCP.PAYLOAD (1000) .B64DECODE.BETUMEN (“string1”,” string2”) など。

[書き換えアクションエバリュエータ] ダイアログボックスを使用して書き換えアクションを評価する

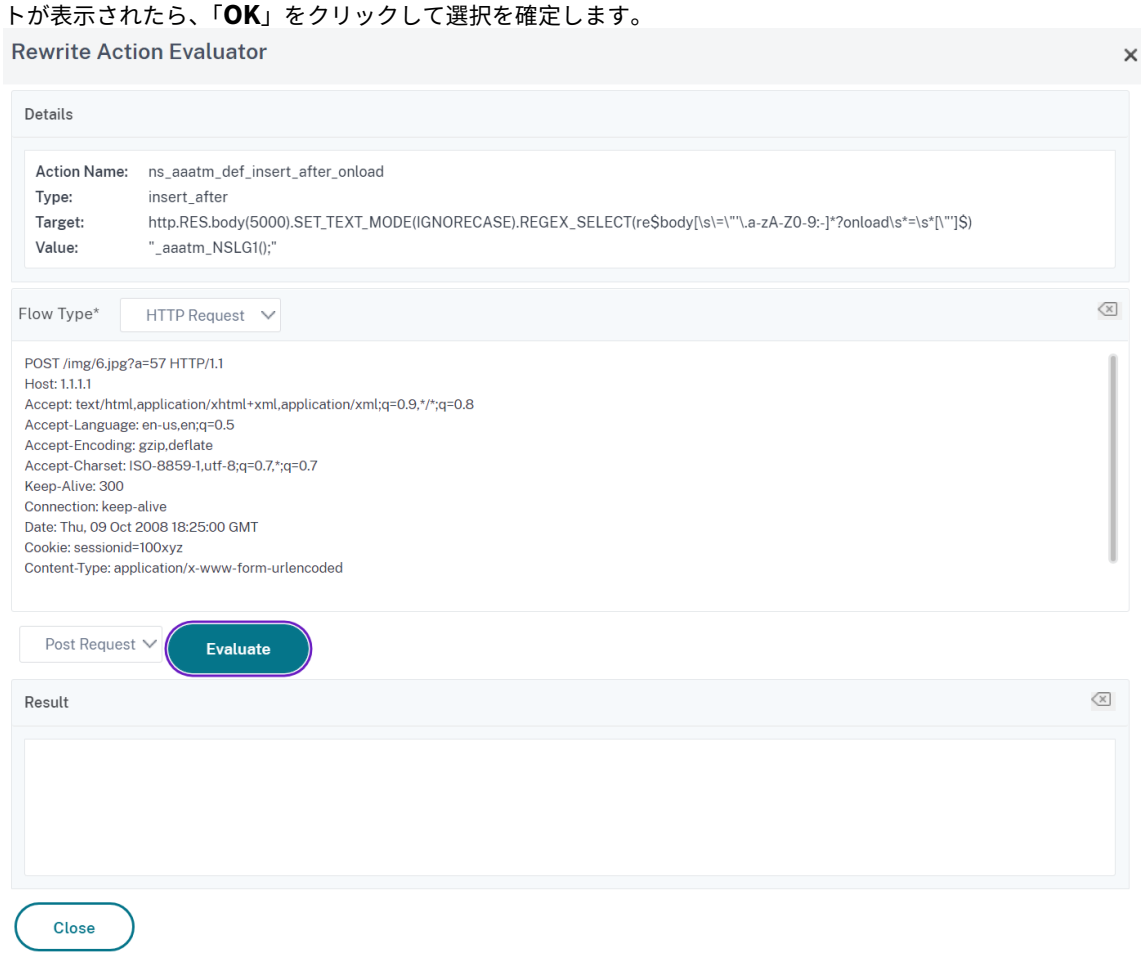
1. [書き換えアクション] の詳細ウィンドウで、評価する書き換えアクションを選択し、[評価] をクリックします。

2. [式エバリュエーターを書き換え] ダイアログボックスで、次のパラメータの値を指定します。(アスタリスクは必須パラメータを示します)。

   「書き換えアクション」(Rewrite Action)-評価する書き換えアクションがまだ選択されていない場合は、ドロップダウンリストから選択します。書き換えアクションを選択すると、「詳細」セクションに、選択した書き換えアクションの詳細が表示されます。
   「新規」(New)-「新規作成」(New) を選択して「書き換えアクションを作成」(Create Rewrite Action) ダイアログボックスを開
   「変更」(Modify)-「修正」(Modify) を選択して「書き換えアクションを設定」(Configure) ダイアログボックスを開き、選択した書き換え
   フロータイプ：選択した書き換えアクションを HTTP リクエストデータまたは HTTP 応答データのどちらでテストするかを指定します。デフォルトは [リクエスト] です。応答データでテストする場合は、[応答] を選択します。
   HTTP 要求/応答データ *：書き換えアクションエバリュエーターがテストに使用する HTTP データを提供するスペースを提供します。データをウィンドウに直接貼り付けるか、[Sample] をクリックしてサンプル HTTP ヘッダーを挿入できます。
   行末を表示-サンプル HTTP データの各行の末尾に UNIX スタイルの行末文字（\ n）を表示するかどうかを指定します。
   [サンプル]：HTTP リクエスト/レスポンスデータウィンドウにサンプル HTTP データを挿入します。GET または POST データを選択できます。
   [参照（Browse）]：ローカルブラウズウィンドウが開き、ローカルまたはネットワークロケーションのサン

プル HTTP データを含むファイルを選択できます。
[クリア（Clear）]：[HTTP 要求/応答データ] ウィンドウから現在のサンプル HTTP データを消去します。

3. [評価] をクリックします。書き換えアクションエバリュエーターは、選択したサンプルデータに対する書き換えアクションの効果を評価し、[結果] ウィンドウで選択した [書き換え] アクションによって変更された結果を表示します。追加および削除は、ダイアログボックスの左下隅の凡例に示されているように強調表示されます。
4. すべてのアクションが希望する効果があると判断するまで、書き換えアクションを評価し続けます。
   - 選択した書き換えアクションを修正し、変更したバージョンをテストするには、[修正] をクリックして [書き換えアクションの構成] ダイアログボックスを開き、変更内容を保存し、もう一度 [評価] をクリックします。
   - 同じリクエストまたはレスポンスデータを使用して、別の書き換えアクションを評価するには、[書き換えアクション] ドロップダウンリストからそれを選択し、もう一度 [評価] をクリックします。
5. [閉じる] をクリックして、[式の書き換えエバリュエーター] を閉じ、[書き換え操作] ウィンドウに戻ります。
6. 書き換えアクションを削除するには、削除する書き換えアクションを選択し、「削除」をクリックし、プロンプトが表示されたら、「**OK**」をクリックして選択を確定します。



書き換えポリシーの構成

October 7, 2021

必要な書き換えアクションを作成したら、少なくとも 1 つの書き換えポリシーを作成して、Citrix ADC アプライアンスで書き換える要求を選択する必要があります。

書き換えポリシーは、ルール自体が 1 つ以上の式で構成されます。また、リクエストまたはレスポンスがルールに一致した場合に実行される関連アクション。HTTP リクエストとレスポンスを評価するためのポリシールールは、リクエストまたはレスポンスのほぼすべての部分に基づくことができます。

TCP リライトアクションを使用して、TCP ペイロード以外のデータを書き換えることはできません。TCP 書き換えポリシーのポリシールールは、トランスポート層内の情報に基づいて行うことができます。そしてトランスポート層の下の層。

設定されたルールが要求または応答と一致すると、対応するポリシーがトリガーされ、関連付けられたアクションが実行されます。

> 注:
> コマンドラインインターフェイスまたは構成ユーティリティを使用して、書き換えポリシーを作成および構成できます。コマンドラインインターフェイスと Citrix ADC ポリシー式言語に精通していないユーザーは、通常、構成ユーティリティの使用がはるかに簡単になります。

コマンドラインインターフェイスを使用して新しい書き換えポリシーを追加するには

コマンドプロンプトで次のコマンドを入力して、新しい書き換えポリシーを追加し、構成を確認します。

- <add rewrite policy <name> <expression> <action> [<undefaction>]
- <show rewrite policy <name>

例 **1. HTTP** コンテンツの書き換え:

```
> add rewrite policy policyNew ”HTTP.RES.IS_VALID” insertact NOREWRITE
 Done
> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertact
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```


例 **2. TCP** ペイロードの書き換え **(TCP** 書き換え**)**:

```
> add rewrite policy client_tcp_payload_policy CLIENT.IP.SRC.EQ
   (172.168.12.232) client_tcp_payload_replace_all
 Done
> show rewrite policy client_tcp_payload_policy
        Name: client_tcp_payload_policy
        Rule: CLIENT.IP.SRC.EQ(172.168.12.232)
        RewriteAction: client_tcp_payload_replace_all
        UndefAction: Use Global
        LogAction: Use Global
        Hits: 0
        Undef Hits: 0

 Done
>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して既存の書き換えポリシーを変更するには

コマンドプロンプトで次のコマンドを入力して、既存の書き換えポリシーを変更し、構成を確認します。

- <set rewrite policy <name>-rule <expression-action <action> [<undefaction>]
- <show rewrite policy <name>

例:

```
> set rewrite policy policyNew -rule ”HTTP.RES.IS_VALID” -action
   insertaction
 Done

> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertaction
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して書き換えポリシーを削除するには

コマンドプロンプトで次のコマンドを入力して、書き換えポリシーを削除します。

rm rewrite policy <name>

例:

```
> rm rewrite policy policyNew
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して書き換えポリシーを構成するには

1. [**AppExpert**] > [**Rewrite**] > [**Policies**] の順に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ポリシーを作成するには、[追加] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
3. [作成] または [**OK**] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。
4. 手順 2 ～4 を繰り返して、必要な数の書き換えアクションを作成または変更します。
5. [閉じる] をクリックします。書き換えポリシーを削除するには、削除する書き換えポリシーを選択し、「削除」をクリックし、プロンプトが表示されたら、「**OK**」をクリックして選択を確定します。

コンテンツセキュリティヘッダー、**XSS** 保護、**HSTS**、**X-Content-Type-Options**、およびコンテンツセキュリティポリシーの書き換えポリシーを作成する

コマンドプロンプトで次の書き換えアクションコマンドを入力して、書き換えを使用して NetScaler を介して提供される Web ページにセキュリティヘッダーを追加します。

```
add rewrite action insert_STS_header insert_http_header Strict-
    Transport-Security ""max-age=157680000""
add rewrite action rw_act_insert_XSS_header insert_http_header X-Xss-
    Protection ""1; mode=block""
add rewrite action rw_act_insert_Xcontent_header insert_http_header X-
    Content-Type-Options ""nosniff""
add rewrite action rw_act_insert_Content_security_policy
    insert_http_header Content-Security-Policy ""default-src 'self' ;
    script-src 'self' 'unsafe-inline' 'unsafe-eval' ; style-src 'self' '
    unsafe-inline' 'unsafe-eval'; img-src 'self' data:""
<!--NeedCopy-->
```


コマンドプロンプトで次の書き換えポリシーコマンドを入力して、書き換えを使用して NetScaler を介して提供される Web ページにセキュリティヘッダーを追加します。

```
add rewrite policy enforce_STS true insert_STS_header
add rewrite policy rw_pol_insert_XSS_header ”HTTP.RES.HEADER(”X-Xss-Protection”).EXISTS.NOT” rw_act_insert_XSS_header
add rewrite policy rw_pol_insert_XContent TRUE rw_act_insert_Xcontent_header
add rewrite policy rw_pol_insert_Content_security_policy TRUE rw_act_insert_Content_security_policy
<!--NeedCopy-->
```

コマンドプロンプトで次のコマンドを入力して、Goto Expression NEXT を使用してレスポンスの仮想サーバーにポリシーをバインドします。

```
bind vpn vserver access -policy enforce_STS -priority 100 -gotoPriorityExpression NEXT -type RESPONSE
bind vpn vserver ”VSERVERNAME” -policy rw_pol_insert_XSS_header -priority 110 -gotoPriorityExpression NEXT -type RESPONSE
bind vpn vserver access -policy rw_pol_insert_XContent -priority 120 -gotoPriorityExpression NEXT -type RESPONSE
bind vpn vserver access -policy rw_pol_insert_Content_security_policy -priority 130 -gotoPriorityExpression NEXT -type RESPONSE
<!--NeedCopy-->
```

構成ユーティリティを使用して、コンテンツセキュリティヘッダー、**XSS** 保護、**HSTS**、**X-Content-Type-Options**、およびコンテンツセキュリティポリシーの書き換えポリシーを構成する

1. **AppExpert** > 書き換え > アクションに移動します。
2. [追加] をクリックして、ヘッダーごとに書き換えアクションを作成します。
3. **AppExpert** > 書き換え > ポリシーに移動します。
4. [追加] をクリックして、書き換えポリシーを作成し、アクションにリンクします。
5. Goto 式 **NEXT** を使用して、レスポンスでポリシーを仮想サーバーにバインドします。

> 注:
> SSLVPN では、以下のコンテンツセキュリティアクションを使用する必要があります。

```
add rewrite action Rewrite_Insert_Content-Security-Policy insert_http_header Content-Security-Policy ””default-src 'self' ;
```



```
        script-src 'self' 'unsafe-inline' 'unsafe-eval' ; style-src 'self' '
        unsafe-inline' 'unsafe-eval'; img-src 'self' http://localhost:* data
        :;"
2   <!--NeedCopy-->
```

ブラウザが localhost HTTP 呼び出しを使用して Cookie/GW 情報をプラグインに渡すため、localhost 例外が必要です。CSP には「self」しか持たないため、仮想サーバーへの呼び出しだけが許可されます。

書き換えポリシーのバインド

October 7, 2021

書き換えポリシーを作成したら、それをバインドして有効にする必要があります。Citrix ADC を通過するすべてのトラフィックにポリシーを適用する場合は、ポリシーをグローバルにバインドするか、特定の仮想サーバーまたはバインドポイントにバインドして、その仮想サーバーまたはバインドポイントの着信トラフィックのみをそのポリシーに誘導することができます。受信要求が書き換えポリシーと一致する場合、そのポリシーに関連付けられたアクションが実行されます。

HTTP 要求と応答を評価するための書き換えポリシーは、HTTP または SSL タイプの仮想サーバーにバインドすることも、REQ_OVERRIDE、REQ_DE
FAULT、RES_OVERRIDE、および RES_DEFAULT バインドポイントにバインドすることもできます。TCP 書き換えの書き換えポリシーは、TCP または SSL_TCP タイプの仮想サーバー、または OTHERTCP_REQ_OVERRIDE、OTHERTCP_REQ_DEFAULT、OTHERTCP_RES_OVERRIDE、および O
THERTCP_RES_DEFAULT バインドポイントにのみバインドできます。

注：OTHERTCP という用語は、Citrix ADC アプライアンスのコンテキストで使用され、TCP パケットがカプセル化するプロトコルに関係なく、生のバイトストリームとして扱うすべての TCP または SSL_TCP 要求および応答を指します。

ポリシーをバインドするときは、ポリシーに優先度を割り当てます。優先順位によって、定義したポリシーが評価される順序が決まります。プライオリティは任意の正の整数に設定できます。

Citrix ADC オペレーティングシステムでは、ポリシーの優先順位は逆の順序で機能します。値が大きいほど、優先順位は低くなります。たとえば、プライオリティ 10、100、1000 の 3 つのポリシーがある場合、プライオリティ 10 が割り当てられたポリシーが最初に適用され、次にポリシーにプライオリティ 100 が割り当てられ、最後に 1000 というプライオリティが割り当てられます。

Citrix ADC オペレーティングシステムの他のほとんどの機能とは異なり、書き換え機能は、要求がポリシーに一致した後も引き続きポリシーを評価して実装します。ただし、要求または応答に対する特定のアクションポリシーの影響は、別のアクションの前後に実行されるかどうかによって異なることがよくあります。優先度は、意図した結果を得るために重要です。

他のポリシーを任意の順序で追加できる十分な余地を残すことができます。また、ポリシーをバインドするときに、各ポリシー間に 50 または 100 の間隔で優先度を設定することで、希望する順序で評価されるように設定することもできます。これを行うと、既存のポリシーの優先順位を再割り当てすることなく、いつでもポリシーを追加できます。

書き換えポリシーをバインドする場合、goto 式 (gotoPriorityExpression) をポリシーに割り当てるオプションもあります。goto 式には、goto 式を含むポリシーよりも高い優先順位を持つ別のポリシーに割り当てられた優先順位と一致する任意の正の整数を指定できます。goto 式をポリシーに割り当てた場合、要求または応答がポリシーと一致すると、Citrix ADC は、優先度が goto 式と一致するポリシーに即座に移動します。現在のポリシーよりも小さいが、goto 式のプライオリティ番号よりも大きいプライオリティ番号を持つポリシーはスキップされ、これらのポリシーは評価されません。

コマンドラインインターフェイスを使用して書き換えポリシーをグローバルにバインドするには

コマンドプロンプトで次のコマンドを入力して、書き換えポリシーをグローバルにバインドし、設定を確認します。

- `bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [-type <type>] [-invoke (<labelType> <labelName>)]`
- `show rewrite global`

例:

```
>bind rewrite global policyNew 10
 Done

> show rewrite global
1)      Global bindpoint: RES_DEFAULT
        Number of bound policies: 1

2)      Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1

 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して書き換えポリシーを特定の仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、書き換えポリシーを特定の仮想サーバーにバインドし、構成を確認します。

- `bind lb vserver <name>@ (<serviceName>@ [-weight <positive_integer>])| <serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )] [-invoke (<labelType> <labelName>)] )`

- show lb vserver <name>

例:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
>
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
        Time since last state change: 28 days, 01:57:26.350
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:50
2)      Policy : cf-pol Priority:1      Inherited
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して書き換えポリシーをバインドポイントにバインドするには

1. [**AppExpert**] > [書き換え] > [ポリシー] に移動します。
2. 詳細ペインで、グローバルにバインドする書き換えポリシーを選択し、[**Policy Manager**] をクリックします。
3. [ポリシーマネージャを書き換え] ダイアログボックスの [バインドポイント] メニューで、次のいずれかの操作を行います。
   a) HTTP 書き換えポリシーのバインディングを構成する場合は、[**HTTP**] をクリックし、要求ベースの書き換えポリシーと応答ベースの書き換えポリシーのどちらを構成するかに応じて、[**Request**] または [**Response**] をクリックします。

b) TCP 書き換えポリシーのバインディングを構成する場合は、[**TCP**] をクリックし、クライアント側の TCP 書き換えポリシーとサーバー側の TCP 書き換えポリシーのどちらを構成するかに応じて、[クライアント] または [サーバー] をクリックします。

4. リライト・ポリシーをバインドするバインド・ポイントをクリックします。[**Rewrite Policy Manager**] ダイアログ・ボックスに、選択したバインド・ポイントにバインドされているすべてのリライト・ポリシーが表示されます。
5. [**Insert Policy**] をクリックして新しい行を挿入し、使用可能なバインドされていないすべての書き換えポリシーを含むドロップダウンリストを表示します。
6. バインドポイントにバインドするポリシーをクリックします。ポリシーは、バインドポイントにバインドされた書き換えポリシーのリストに挿入されます。
7. [優先順位] 列では、優先度を任意の正の整数に変更できます。このパラメーターの詳細については、「書き換えポリシーをバインドするためのパラメーター」の「優先度」を参照してください。
8. 現在のポリシーが一致した場合に、ポリシーをスキップして特定のポリシーに直接移動する場合は、[Goto Expression] カラムの値を、次に適用するポリシーのプライオリティと同じ値に変更します。このパラメーターの詳細については、「書き換えポリシーをバインドするためのパラメーター」の「gotoPriorityExpression」を参照してください。
9. ポリシーを変更するには、ポリシーをクリックしてから、[ポリシーの変更] をクリックします。
10. ポリシーをバインド解除するには、ポリシーをクリックし、[ポリシーのバインド解除] をクリックします。
11. アクションを変更するには、[アクション] 列で変更するアクションをクリックし、[アクションの変更] をクリックします。
12. 呼び出しラベルを変更するには、**[**呼び出し**]** 列で変更する呼び出しラベルをクリックし、**[**呼び出し ****** ラベルの変更] をクリックします。**
13. 現在構成しているバインドポイントにバインドされているすべてのポリシーのプライオリティを再生成するには、[**Regenerate Priorities**] をクリックします。ポリシーでは、他のポリシーと相対的に既存の優先順位が保持されますが、優先順位は 10 の倍数で再番号が付けられます。
14. [変更を適用**]** をクリックします。
15. [閉じる] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

構成ユーティリティを使用して書き換えポリシーを特定の仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーの詳細ウィンドウ領域の一覧で、書き換えポリシーをバインドする仮想サーバーを選択し、[開く] をクリックします。
3. [仮想サーバーの構成（負荷分散）] ダイアログボックスで、[ポリシー] タブを選択します。Citrix ADC で構成されたすべてのポリシーがリストに表示されます。
4. この仮想サーバにバインドするポリシーの名前の横にあるチェックボックスをオンにします。
5. [**OK**] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

書き換えポリシーラベルの設定

October 7, 2021

単一のポリシーでサポートされるよりも複雑なポリシー構造を構築する場合は、ポリシーラベルを作成し、ポリシーと同じようにバインドできます。ポリシーラベルは、ポリシーがバインドされるユーザ定義のポイントです。ポリシーラベルが呼び出されると、そのラベルにバインドされているすべてのポリシーが、設定した優先順位に従って評価されます。ポリシー・ラベルには、1 つまたは複数のポリシーを含めることができ、それぞれに独自の結果を割り当てることができます。ポリシー・ラベル内の 1 つのポリシーが一致すると、次のポリシーに進み、別のポリシー・ラベルまたは適切なリソースを呼び出したり、ポリシー評価の即時終了を行ったり、ポリシー・ラベルを呼び出したポリシーに制御が戻ったりすることがあります。

書き換えポリシーラベルは、名前、ポリシーラベルに含まれるポリシーのタイプを示すトランスフォーム名、およびポリシーラベルにバインドされたポリシーのリストで構成されます。ポリシーラベルにバインドされている各ポリシーには、書き換えポリシーの設定で説明されているすべての要素が含まれます。

注：コマンドラインインターフェイスまたは設定ユーティリティを使用して、書き換えポリシーラベルを作成および構成できます。コマンドラインインターフェイスと Citrix ADC ポリシーインフラストラクチャ（PI）言語に精通していないユーザーは、通常、構成ユーティリティの使用がはるかに容易になります。

コマンドラインインターフェイスを使用して書き換えポリシーラベルを構成するには

新しい書き換えポリシーラベルを追加するには、コマンドプロンプトで次のコマンドを入力します。

add rewrite policylabel <labelName> <transform>

たとえば、polLabelHTTPResponses という名前の書き換えポリシーラベルを追加して、HTTP 応答で動作するすべてのポリシーをグループ化するには、次のように入力します。

add rewrite policylabel polLabelHTTPResponses http_res

既存の書き換えポリシーラベルを変更するには、Citrix ADC コマンドプロンプトで次のコマンドを入力します。

set rewrite policy <name> <transform>

注：set rewrite policy コマンドは、add rewrite policy コマンドと同じオプションを取ります。

書き換えポリシーラベルを削除するには、Citrix ADC コマンドプロンプトで次のコマンドを入力します。

rm rewrite policy<name>

たとえば、polLabelHTTPResponses という名前の書き換えポリシーラベルを削除するには、次のように入力します。

rm rewrite policy polLabelHTTPResponses

構成ユーティリティを使用して書き換えポリシーラベルを構成するには

1. [**AppExpert**] > [書き換え] > [ポリシーラベル] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいポリシーラベルを作成するには、[**Add**] をクリックします。
   - 既存のポリシーラベルを変更するには、ポリシーを選択し、[開く] をクリックします。
3. ポリシーラベルにバインドされているリストからポリシーを追加または削除します。
   - リストにポリシーを追加するには、[**Insert Policy**] をクリックし、ドロップダウンリストからポリシーを選択します。リストで [New Policy] を選択し、書き換えポリシーの設定の手順に従って、新しいポリシーを作成してリストに追加できます。
   - リストからポリシーを削除するには、そのポリシーを選択し、[ポリシーのバインド解除] をクリックします。
4. [Priority] 列の番号を編集して、各ポリシーのプライオリティを変更します。

   [優先度を再生] をクリックして、ポリシーの番号を自動的に再設定することもできます。
5. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

   ポリシー・ラベルを削除するには、ポリシー・ラベルを選択し、「削除」をクリックします。ポリシー・ラベルの名前を変更するには、ポリシー・ラベルを選択し、「名前の変更」をクリックします。ポリシーの名前を編集し、[**OK**] をクリックして変更を保存します。

デフォルトの書き換えアクションの設定

October 7, 2021

未定義のイベントは、Citrix ADC がポリシーを評価できない場合にトリガーされます。通常、ポリシー内の論理エラーまたはその他のエラーまたは Citrix ADC のエラー状態が検出されたためです。書き換えポリシーの評価でエラーが発生した場合は、指定された未定義のアクションが実行されます。書き換えポリシーレベルで設定された未定義のアクションは、グローバルに設定された未定義のアクションの前に実行されます。

Citrix ADC では、次の 3 種類の未定義アクションがサポートされています。

- undefAction NOREWRITE

  書き換え処理を中止しますが、パケットフローは変更しません。つまり、Citrix ADC は、どの書き換えポリシーにも一致しない要求と応答を処理し続け、別の機能が要求をブロックまたはリダイレクトしない限り、最終的に要求した URL に転送します。このアクションは、Web サーバーへの通常の要求に適しており、デフォルトの設定です。

- undefAction RESET

クライアント接続をリセットします。つまり、Citrix ADC は、Web サーバーとのセッションを再確立する必要があることをクライアントに通知します。このアクションは、存在しない Web ページの繰り返し要求、または保護された Web サイトをハッキングまたはプローブしようとする可能性のある接続に適しています。

- undefAction DROP

  クライアントに何らかの形で応答せずに要求をサイレントにドロップします。つまり、Citrix ADC はクライアントに応答せずに接続を破棄するだけです。このアクションは、DDoS 攻撃またはサーバーに対する別の持続的な攻撃の一部であると思われる要求に適しています。

注: 未定義のイベントは、要求と応答のフロー固有のポリシーの両方に対してトリガーできます。

コマンドラインインターフェイスを使用して既定のアクションを構成するには

コマンドプロンプトで次のコマンドを入力して、既定の動作を構成し、構成を確認します。

- <set rewrite param -undefAction (NOREWRITE | RESET | DROP)
- <show rewrite param

例:

```
1 > set rewrite param -undefAction NOREWRITE
2  Done
3 > show rewrite param
4         Action Name: NOREWRITE
5  Done
6 <!--NeedCopy-->
```

構成ユーティリティを使用して既定のアクションを構成するには

1. [AppExpert] > [書き換え] に移動します。
2. 詳細ペインの [書き換えの概要] で、[書き換え設定の変更] リンクをクリックします。[書き換えパラメータの設定] ダイアログボックスが表示されます。
3. 「グローバル未定義結果アクション」で、次のいずれかのオプションを選択します。
   - NoRewrite—NOREWRITE
   - Reset—RESET
   - Drop—DROP
4. [OK] をクリックします。グローバル未定義のアクションは、選択した値に設定されます。

安全性チェックのバイパス

October 7, 2021

リライトアクションを作成すると、アクションの作成に使用した式が安全であるかどうかが Citrix ADC によって検証されます。HTTP リクエストに含まれる URL などのランタイムデータから Citrix ADC によって作成された式は、予期しないエラーを引き起こす可能性があります。Citrix ADC は、安全でない式などのエラーを引き起こす式を報告します。

場合によっては、式が安全である可能性があります。たとえば、Citrix ADC は、Web サーバーが一時的に利用できないために URL が解決されない場合でも、解決されない URL を含む式を検証できません。安全性チェックを手動でバイパスして、これらの式を許可することができます。

コマンドラインインターフェイスを使用して安全性チェックをバイパスするには

コマンドプロンプトで次のコマンドを入力して、安全性チェックをバイパスし、構成を確認します。

- <set rewrite action <name> -bypassSafetyCheck YES
- <show rewrite action <name>

例:

```
1  > set rewrite action insertact -bypassSafetyCheck YES
2   Done
3  > show rewrite action insertact
4
5          Name: insertact
6          Operation: insert_http_header   Target:Client-IP
7          Value:CLIENT.IP.SRC
8          BypassSafetyCheck : YES
9          Hits: 0
10         Undef Hits: 0
11         Action Reference Count: 2
12  Done
13 <!--NeedCopy-->
```

構成ユーティリティを使用して安全性チェックをバイパスするには

1. **AppExpert** > 書き換え > アクションに移動します。
2. 詳細ウィンドウで、安全チェックから除外する書き換え操作を選択し、[開く] をクリックします。
3. [書き換えアクションを設定] ダイアログボックスで、[安全チェックをバイパスする] チェックボックスをオンにします。

4. [**OK**] をクリックします。

書き換えアクションとポリシーの例

October 7, 2021

このセクションの例では、さまざまな便利なタスクを実行するために書き換えを設定する方法を示します。この例は、中規模の製造会社である Example Manufacturing Inc. のサーバールームで発生しています。この会社は、Web サイトを使用して販売、配送、およびカスタマーサポートのかなりの部分を管理しています。

サンプル製造には 2 つのドメインがあります。example.com は Web サイトと顧客への電子メールで、イントラネットは example.net です。お客様は、サンプル Web サイトを使用して、注文、見積依頼、製品の調査、カスタマーサービスおよびテクニカルサポートに連絡します。

Example の収益ストリームの重要な部分として、Web サイトは迅速に対応し、顧客データの機密性を維持する必要があります。したがって、例には複数の Web サーバーがあり、Citrix ADC アプライアンスを使用して、Web サイトの負荷を分散し、Web サーバーとの間のトラフィックを管理します。

サンプルシステム管理者は、書き換え機能を使用して次のタスクを実行します。

例 **1:** 古い **X-Forwarded-For** ヘッダーとクライアント **IP** ヘッダーの削除

Example Inc. は、受信要求から古い X-Forwarded-For およびクライアント IP HTTP ヘッダーを削除します。

例 **2**: ローカルクライアント **IP** ヘッダーの追加

Example Inc. は、受信要求に新しいローカル Client-IP ヘッダーを追加します。

例 **3:** セキュア接続とセキュアでない接続のタグ付け

Example Inc. は、受信要求に、接続が安全な接続であるかどうかを示すヘッダーをタグ付けします。

例 **4: HTTP** サーバタイプのマスク

Example Inc. は HTTP Server: ヘッダーを変更して、不正なユーザや悪意のあるコードがそのヘッダーを使用して使用する HTTP サーバソフトウェアを判別できないようにします。

例 **5:** 外部 **URL** を内部 **URL** にリダイレクトする

Example Inc. は、Web サーバーの実際の名前とサーバールームの構成に関する情報をユーザーから隠し、Web サイトの URL を短く覚えやすくし、サイトのセキュリティを強化しています。

例 **6: Apache** 書き換えモジュールルールの移行

Example Inc. は、Apache リライトルールを Citrix ADC アプライアンスに移動し、Apache PERL ベースのスクリプト構文を Citrix ADC リライトルールの構文に変換します。

例 **7:** マーケティングキーワードのリダイレクト

Example Inc. のマーケティング部門は、会社の Web サイトで事前定義された特定のキーワード検索の簡略化された URL を設定します。

例 **8:** クエリをクエリされたサーバーにリダイレクトします。

Example Inc. は、特定のクエリ要求を適切なサーバーにリダイレクトします。

例 **9:** ホームページのリダイレクト

Example Inc. は最近、小規模な競合他社を買収し、取得した会社のホームページへの要求を独自の Web サイトのページにリダイレクトするようになりました。

例 **10**: ポリシーベースの **RSA** 暗号化

Example Inc. は、PEM RSA 公開キーを使用して、事前定義およびユーザー定義のヘッダーまたは本文のコンテンツを暗号化します。

これらの各タスクでは、システム管理者が書き換えアクションとポリシーを作成し、Citrix ADC 上の有効なバインドポイントにバインドする必要があります。

例 **1:** 古い **X-Forwarded-For** ヘッダーとクライアント **IP** ヘッダーの削除

October 7, 2021

Example Inc. は、受信要求から古い X-Forwarded-For ヘッダーとクライアント IP HTTP ヘッダーを削除して、表示される唯一の X-Forwarded-For ヘッダーがローカルサーバーによって追加されるようにします。この構成は、Citrix ADC コマンドラインまたは構成ユーティリティを使用して実行できます。Example Inc. のシステム管理者は、古い学校のネットワークエンジニアであり、可能な限り CLI を使用することを好むが、設定ユーティリティのインターフェイスを理解して、新しいシステム管理者にその使用方法をチームに表示できるようにしたいと考えています。

次の例では、CLI と構成ユーティリティの両方を使用して各設定を実行する方法を示します。この手順は、ユーザーが書き換えアクションの作成、書き換えポリシーの作成、およびバインドポリシーの基本をすでに知っていると仮定して省略されています。

- 書き換えアクションの作成の詳細については、「書き換えアクションの設定」を参照してください。
- 書き換えポリシーの作成の詳細については、「書き換えポリシーの設定」を参照してください。
- 書き換えポリシーのバインドの詳細については、「書き換えポリシーのバインド」を参照してください。

コマンドラインインターフェイスを使用して要求から古い **X-Forwarded** ヘッダーとクライアント **IP** ヘッダーを削除するには

コマンドプロンプトで、次のコマンドを次の順序で入力します。


```
add rewrite action act_del_xfor delete_http_header x-forwarded-for
add rewrite action act_del_cip delete_http_header client-ip
add rewrite policy pol_check_xfor 'HTTP.REQ.HEADER(”x-forwarded-for”).EXISTS' act_del_xfor
add rewrite policy pol_check_cip 'HTTP.REQ.HEADER(”client-ip”).EXISTS' act_del_cip
bind rewrite global pol_check_xfor 100 200
bind rewrite global pol_check_cip 200 300
<!--NeedCopy-->
```

構成ユーティリティを使用して要求から古い **X-Forwarded** ヘッダーとクライアント **IP** ヘッダーを削除するには

[書き換えアクションの作成] ダイアログボックスで、次の説明とともに 2 つの書き換えアクションを作成します。

名前	種類	引数（複数可）
act_del_xfor	delete_http_header	x-forwarded-for
act_del_cip	delete_http_header	client-ip

[書き換えポリシーの作成] ダイアログボックスで、次の説明を使用して 2 つの書き換えポリシーを作成します。

名前	式	操作（アクション）
pol_check_xfor	‘HTTP.REQ.HEADER(“x-forwarded-for”).EXISTS’	act_del_xfor
pol_check_cip	‘HTTP.REQ.HEADER(“client-ip”).EXISTS’	act_del_cip

両方のポリシーをグローバルにバインドし、以下に示す優先順位と goto 式の値を割り当てます。

名前	優先度	**Goto** 式
pol_check_xfor	100	200
pol_check_cip	200	300

すべての X-Forwarded-For およびクライアント IP HTTP ヘッダーが、着信要求から削除されるようになりました。

例 2：ローカルクライアント IP ヘッダーの追加

October 7, 2021

Example Inc. は、受信要求にローカルのクライアント IP HTTP ヘッダーを追加したいと考えています。この例には、同じ基本タスクの 2 つのわずかに異なるバージョンが含まれています。

コマンドラインインターフェイスを使用してローカル Client-IP ヘッダーを追加するには

コマンドプロンプトで、次のコマンドを次の順序で入力します。

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.IP.SRC'
add rewrite policy pol_ins_client 'HTTP.REQ.HEADER(”x-forwarded-for”).EXISTS || HTTP.REQ.HEADER(”client-ip”).EXISTS' act_ins_client
bind rewrite global pol_ins_client 300 END
<!--NeedCopy-->
```

構成ユーティリティを使用してローカル Client-IP ヘッダーを追加するには

[書き換えアクションの作成] ダイアログボックスで、次の説明を含む書き換えアクションを作成します。

名前	種類	引数（複数可）
act_ins_client	insert_http_header	NS-Client ‘CLIENT.IP.SRC’

[書き換えポリシーの作成] ダイアログボックスで、次の説明を含む書き換えポリシーを作成します。

名前	式	操作（アクション）
pol_ins_client	‘HTTP.REQ.HEADER(“x-forwarded-for”).EXISTS || HTTP.REQ.HEADER(“client-ip”).EXISTS’	act_ins_client

ポリシーをグローバルにバインドし、次に示す優先順位と goto 式値を割り当てます。

名前	優先度	Goto 式
pol_ins_client	100	Next

例 **3:** セキュア接続とセキュアでない接続のタグ付け

October 7, 2021

Example Inc. は、受信要求に、接続が安全な接続であるかどうかを示すヘッダーをタグ付けします。これにより、Citrix ADC が接続を復号化した後、サーバーは安全な接続を追跡できます。

この構成を実装するには、まず次の表に示す値を使用して書き換えアクションを作成します。これらのアクションは、ポート 80 への接続を非セキュア接続としてラベル付け、ポート 443 への接続をセキュア接続としてラベル付けします。

アクション名	書き換えアクションのタイプ	ヘッダー名	値
Action-Rewrite-SSL_YES	INSERT_HTTP_HEADER	SSL	はい

アクション名	書き換えアクションのタイプ	ヘッダー名	値
Action-Rewrite-SSL_NO	INSERT_HTTP_HEADER	SSL	いいえ

次に、次の表に示す値を使用して書き換えポリシーを作成します。これらのポリシーは、着信要求をチェックして、ポート 80 に送信される要求とポート 443 に送信される要求を決定します。次に、ポリシーによって正しい SSL ヘッダーが追加されます。

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-SSL_YES	Action-Rewrite-SSL_YES	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(443)
Policy-Rewrite-SSL_NO	Action-Rewrite-SSL_NO	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(80)

最後に、書き換えポリシーを Citrix ADC にバインドし、最初のポリシーの優先順位を 200、2 番目のポリシーの優先順位を 300 に割り当て、両方のポリシーの goto 式を END に設定します。

ポート 80 への各着信接続には、SSL：NO HTTP ヘッダーが追加され、ポート 443 への各着信接続には SSL：YES HTTP ヘッダーが追加されます。

例 **4: HTTP** サーバタイプのマスク

October 7, 2021

Example Inc. は、HTTP Server: ヘッダーを変更して、権限のないユーザーや悪意のあるコードが HTTP サーバーが使用するソフトウェアを識別するためにヘッダーを使用できないようにします。

HTTP Server: ヘッダーを変更するには、次の表の値を使用して、書き換えアクションとリライトポリシーを作成します。

アクション名	書き換えアクションのタイプ	ターゲット参照を選択する式	置換テキストの文字列式
Action-Rewrite-Server_Mask	REPLACE	HTTP.RES.HEADER(“Se	“Web Server 1.0”

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-Server_Mask	Action-Rewrite-Server_Mask	NOREWRITE	HTTP.RES.IS_VALID

コマンドの例**:**

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER(”Server”)”\”Web Server 1.0\””
```

```
> add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
```

次に、書き換えポリシーをグローバルにバインドし、優先順位 100 を割り当てて、ポリシーの Goto Priority Expression を END に設定します。

HTTP Server: ヘッダーは、「Web Server 1.0」を読み取るように変更され、Example Inc. Web サイトで使用されている実際の HTTP サーバーソフトウェアをマスキングします。

例 **5:** 外部 **URL** を内部 **URL** にリダイレクトする

January 31, 2022

Example Inc. は、Web サーバーのセキュリティを向上させるために、実際のサーブルームの設定をユーザーから隠したいと考えています。

これを行うには、次の表に示す値を使用して書き換えアクションを作成します。リクエストヘッダーの場合、テーブル内のアクションはwww.example.comを web.hq.example.net に変更します。レスポンスヘッダーの場合、アクションは逆の動作をして、web.hq.example.netをwww.example.comに変換します。

[アクション名]	書き換えアクションのタイプ	ターゲットリファレンスを選択する式	置換テキストの文字列式
Action-Rewrite-Request_Server_Repla	REPLACE	HTTP.REQ.HOSTNAME.S	“Web.hq.example.net”
Action-Rewrite-Response_Server_Replace	REPLACE	HTTP.RES.HEADER(“Server”)	“www.example.com”

最初のポリシーは、受信リクエストが有効かどうかをチェックし、有効である場合は、Action-Rewrite-Request_Server_Replace アクションを実行します。2 番目のポリシーは、応答がサーバーweb.hq.example.netから発信されているかどうかを確認します。その場合、Action-Rewrite-Response_Server_Replace_Server_Replace アクションが実行されます。

外部 URL をリダイレクトするための書き換えアクションとポリシーの例。

```
add rewrite action Action-Rewrite-Request_Server_Replace REPLACE HTTP.REQ.HOSTNAME.SERVER '"Web.hq.example.net"'

add rewrite action Action-Rewrite-Response_Server_Replace REPLACE HTTP.RES.HEADER("Server")'"www.example.com"'

add rewrite policy Policy-Rewrite-Request_Server_Replace HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")Action-Rewrite-Request_Server_Replace NOREWRITE

add rewrite policy Policy-Rewrite-Response_Server_Replace HTTP.REQ.HEADER("Server").EQ("Web.hq.example.net")Action-Rewrite-Response_Server_Replace
```

最後に、書き換えポリシーをバインドし、それぞれに優先度 500 を割り当てます。これは、異なるポリシーバンクにあり、競合しないためです。両方のバインディングで goto 式を NEXT に設定する必要があります。

```
bind rewrite global Policy-Rewrite-Request_Server_Replace 500 END -type REQ_DEFAULT

bind rewrite global Policy-Rewrite-Response_Server_Replace 500 END -type RES_DEFAULT
```


リクエストヘッダーのすべてのwww.example.comのインスタンスがweb.hq.example.netに変更され、レスポンスヘッダーのすべてのweb.hq.example.netのインスタンスがwww.example.comに変更されるようになりました。

例 **6: Apache** 書き換えモジュールルールの移行

October 7, 2021

Example Inc. は現在、Apache 書き換えモジュールを使用して、Web サーバーに送信された検索要求を処理し、要求 URL の情報に基づいてそれらの要求を適切なサーバーにリダイレクトしています。Example Inc. は、これらのルールを Citrix ADC プラットフォームに移行することで、設定を簡素化したいと考えています。

Example が現在使用しているいくつかの Apache 書き換えルールを以下に示します。これらのルールは、SiteID 文字列がない場合、または SiteID 文字列がゼロ (0) に等しい場合は、特別な結果ページに検索リクエストをリダイレクトします。これらの条件が適用されない場合は、標準結果ページにリダイレクトします。

以下は、現在の Apache 書き換えルールです。

- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} !SiteId= [OR]
- RewriteCond %{QUERY_STRING} SiteId=0
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ results2.html [P,L]
- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results.html [P,L]

Citrix ADC でこれらの Apache 書き換えルールを実装するには、次の表の値を使用して書き換えアクションを作成します。

アクション名	書き換えアクションのタイプ	ターゲット参照を選択する式	置換テキストの文字列式
Action-Rewrite-Display_Results_NulSit	REPLACE	HTTP.REQ.URL	“/results2.html”
Action-Rewrite-Display_Results	REPLACE	HTTP.REQ.URL	“/results2.html”

次に、次の表に示すように、値を使用して書き換えポリシーを作成します。

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-Display_Results_NulSit	Action-Rewrite-Display_Results_NulSit	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MOD && (!HTTP.REQ.URL.QUERY.CONTAINS(“S || HTTP.REQ.URL.QUERY.CONTAINS(“Si || HTTP.REQ.URL.QUERY.SET_TEXT_MO
Policy-Rewrite-Display_Results	Action-Rewrite-Display_Results	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MOD || HTTP.REQ.URL.QUERY.SET_TEXT_MO

最後に、書き換えポリシーをバインドし、最初のポリシーに優先順位 600、2 番目のポリシーに優先順位 700 を割り当てて、両方のバインドに対して goto 式を NEXT に設定します。

Citrix ADC は、Apache 書き換えモジュールのルールが移行される前の Web サーバーと同じようにこれらの検索要求を処理するようになりました。

例 **7:** マーケティングキーワードのリダイレクト

October 7, 2021

Example Inc. のマーケティング部門は、会社の Web サイトで特定の定義済みキーワード検索の簡易 URL を設定したいと考えています。これらのキーワードについては、以下に示すように URL を再定義したいと考えています。

- 外部 URL:

```
http://www.example.com/\<marketingkeyword\>
```

- 内部 URL:

```
http://www.example.com/go/kwsearch.asp?keyword=\<marketingkeyword\>
```

マーケティングキーワードのリダイレクトを設定するには、次の表の値を使用して書き換えアクションを作成します。

アクション名	書き換えアクションのタイプ	ターゲット位置を選択する式	置換テキストの文字列式
Action-Rewrite-Modify_URL	INSERT_BEFORE	HTTP.REQ.URL.PATH.GE (1)	””go/kwsearch.aspkeyword=”l”

次に、次の表の値を使用して書き換えポリシーを作成します。

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-Modify_URL	Action-Rewrite-Modify_URL	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“v

最後に、書き換えポリシーをバインドし、プライオリティを 800 に割り当てます。以前の書き換えポリシーとは異なり、このポリシーは、条件に一致するリクエストに最後に適用する必要があります。このため、Citrix ADC 管理者は Goto 優先式を END に設定します。

マーケティングキーワードを使用したリクエストは、キーワード検索 CGI ページにリダイレクトされ、検索が実行され、残りのポリシーはすべてスキップされます。

例 **8:** クエリーをクエリーされたサーバーにリダイレクトする

October 7, 2021

Example Inc. は、次に示すように、クエリ要求を適切なサーバーにリダイレクトしたいと考えています。

- <Request: GET /query.cgi?server=5HOST: www.example.com
- <Redirect URL: <http://web-5.example.com/>

このリダイレクトを実装するには、まず次の表の値を使用して書き換えアクションを作成します。

アクション名	書き換えアクションのタイプ	ターゲット参照を選択する式	置換テキストの文字列式
Action-Rewrite-Replace_Hostheader	REPLACE	HTTP.REQ.HEADER(“Ho	“server-“ + HTTP.REQ.URL.QUERY.VALUE(“web”) ıple.com”)

次に、次の表の値を使用して書き換えポリシーを作成します。

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-Replace_Hostheader	Action-Rewrite-Replace_Hostheader	NOREWRITE	HTTP.REQ.HEADER(”Host”).EQ(”www.example.com”)

コマンドの例**:**


```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("
Server")"\"Web Server 1.0\""
Done

> add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-
Rewrite-Server_Mask NOREWRITE
Done
```

最後に、書き換えポリシーをバインドし、優先度 900 を割り当てます。このポリシーは、条件に一致するリクエストに最後に適用されたポリシーである必要があるため、goto 式を END に設定します。

<http://www.example.com/query.cgi?server>=で始まる任意の URL への着信要求は、クエリのサーバー番号にリダイレクトされます。

例 9: ホームページのリダイレクト

October 7, 2021

New Company, Inc. は最近、小規模な競合企業である購入企業を買収し、購入企業のホームページを、次に示すように自社の Web サイトの新しいページにリダイレクトしたいと考えています。

- 古い URL:http://www.purchasedcompany.com/*
- 新しい URL:http://www.newcompany.com/products/page.htm

リクエストを「購買済会社」ホームページへリダイレクトするには、次の表の値を使用して書き換えアクションを作成します。

アクション名	書き換えアクションのタイプ	ターゲット参照を選択する式	置換テキストの文字列式
Action-Rewrite-Replace_URLr	REPLACE	HTTP.REQ.URL.PATH_A	“/products/page.htm”
Action-Rewrite-Replace_Host	REPLACE	HTTP.REQ.HOSTNAME	「www.newcompany.com」

次に、次の表の値を使用して書き換えポリシーを作成します。

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-Replace-None	Action-Rewrite-Replace-None	NOREWRITE	!HTTP.REQ.HOSTNAME.SERVER.EQ(“

ポリシー名	アクション名	未定義のアクション	式
Policy-Rewrite-Replace-Host	Action-Rewrite-Replace_Host	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“v
Policy-Rewrite-Replace-URL	Action-Rewrite-Replace_URL	NOREWRITE	HTTP.REQ.IS_VALID

最後に、書き換えポリシーをグローバルにバインドし、最初のポリシーに優先度 100、2 番目のポリシーに優先度 200、3 番目のポリシーに優先度 300 を割り当てます。これらのポリシーは、条件に一致するリクエストに最後に適用されるポリシーである必要があります。このため、goto 式を 1 番目と 3 番目のポリシーでは END に、2 番目のポリシーでは 300 に設定します。これにより、残りのすべての要求が正しく処理されます。

買収した会社の古い Web サイトへの要求は、新しい会社のホームページの正しいページにリダイレクトされます。

例 **10**：ポリシーベースの **RSA** 暗号化

October 7, 2021

RSA アルゴリズムは、PKEY_ENCRYPT_PEM () 関数を使用して、事前定義された HTTP およびユーザー定義のヘッダーまたは本文の内容を暗号化します。この関数は、RSA 公開キー（秘密キーではない）のみを受け入れ、暗号化されたデータは公開キーの長さより長くすることはできません。暗号化されるデータがキーの長さよりも短い場合、アルゴリズムは RSA_PKCS1 パディング方法を使用します。

サンプルシナリオでは、この関数を B64ENCODE () 関数と共に使用して、HTTP ヘッダー値を RSA 公開鍵で暗号化された値に置き換えることができます。暗号化されるデータは、RSA 秘密キーを使用して受信者によって復号化されます。

この機能は、書き換えポリシーを使用して実装できます。これを行うには、次のタスクを完了する必要があります。

1. RSA 公開キーをポリシー式として追加します。
2. 書き換えアクションを作成します。
3. 書き換えポリシーを作成します。
4. 書き換えポリシーをグローバルとしてバインドします。
5. RSA 暗号化の確認

Citrix ADC コマンドインターフェイスを使用したポリシーベースの **RSA** 暗号化

Citrix ADC コマンドインターフェイスを使用してポリシーベースの RSA 暗号化を構成するには、次のタスクを実行します。

Citrix ADC コマンドインターフェイスを使用して、**RSA** 公開キーをポリシー式として追加するには：


```
add policy expression pubkey '"-----BEGIN RSA PUBLIC KEY-----
    MIGJAoGBAKl5vgQEj73Kxp+9
    yn1v5gPR1pnc4oLM2a0kaWwBOsB6rzCIy6znwnvwCY1xRvQhRlJSAyJbloL7wZFIJ2FOR8Cz
    +8ZQWXU2syG+udi4EnWqLgFYowF9zK+o79az597eNPAjsHZ/C2oL/+6qY5a/
    f1z8bQPrHC4GpFfAEJhh/+NnAgMBAAE=-----END RSA PUBLIC KEY-----"'
<!--NeedCopy-->
```

Citrix ADC コマンドインターフェイスを使用して、**HTTP** ヘッダー要求を暗号化するアクションを書き換えるには:

```
add rewrite action encrypt_act insert_http_header encrypted_data
HTTP.REQ.HEADER("data_to_encrypt").PKEY_ENCRYPT_PEM(pubkey).B64ENCODE
```

Citrix ADC コマンドインターフェイスを使用して書き換えポリシーを追加するには:

```
add rewrite policy encrypt_pol 'HTTP.REQ.HEADER("data_to_encrypt").
    EXISTS' encrypt_act
<!--NeedCopy-->
```

Citrix ADC コマンドインターフェイスを使用してグローバルに書き換えポリシーをバインドするには:

```
bind rewrite global encrypt_pol 10 -type RES_DEFAULT
```

Citrix ADC コマンドインターフェイスを使用して **RSA** 暗号化を確認するには:

```
>curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
    http://10.217.24.7/`

* About to connect() to 10.217.24.7 port 80 (#0)

*  Trying 10.217.24.7...

*  connected

*  Connected to 10.217.24.7 (10.217.24.7) port 80 (#0)

> GET / HTTP/1.1
> User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
    OpenSSL/0.9.8y zlib/1.2.3
> Host: 10.217.24.7
> Accept: \*/\*
> data_to_encrypt: Now is the time that tries men's souls
>
```



```
< HTTP/1.1 200 OK
< Date: Mon, 09 Oct 2017 05:22:37 GMT
< Server: Apache/2.2.24 (FreeBSD) mod_ssl/2.2.24 OpenSSL/0.9.8y DAV/2
< Last-Modified: Thu, 20 Feb 2014 20:29:06 GMT
< ETag: ”6bd9f2-2c-4f2dc5b570880”
< Accept-Ranges: bytes
< Content-Length: 44
< Content-Type: text/html
< encrypted_data: UliegKBJqZd7JdaC49XMLEK1+eQN2rEfevypW91gKvBVlaKM9N9/
   C2BKuztS99SE0xQaisidzN5IgeIcpQMn+
   CiKYVlLzPG1RuhGaqHYzIt6C8A842da7xE4OlV5SHwScqkqZ5aVrXc3EwtUksna7jOLr40aLeXnnB
   /DB11pUAE=
<
* Connection #0 to host 10.217.24.7 left intact
<html><body><h1>It works!</h1></body></html>* Closing connection #0

<!--NeedCopy-->
```

暗号化する同じデータでこの curl コマンドを実行したあと、暗号化されたデータは実行ごとに異なることを示しています。これは、パディングによって暗号化するデータの先頭にランダムなバイトが挿入され、暗号化されたデータが毎回異なるためです。

```
>curl -v -H ”data_to_encrypt: Now is the time that tries men's souls”
   http://10.217.24.7/`

< encrypted_data:
   DaOjtl1Pl4DlQKf58MMeL4cFwFvZwhjMqv5aUYM5Iyzk4UpwIYhpRvgTNu2lXEVc1H0tcR1EGC
   /ViQncLc4EbTurCWLbzjce3+fknnMmzF0lRT6ZZXWbMvsNFOxDA1SnuAgwxWXy/
   ooe9Wy6SYsL2oi1sr5wTG+RihDd9zP+P14=

>curl -v -H ”data_to_encrypt: Now is the time that tries men's souls”
   http://10.217.24.7/

. . .

< encrypted_data: eej6YbGP68yHn48qFUvi+fkG+OiO8j3yYLScrRBU+
   TPQ8WeDVaWnDNAVLvL0ZYHHAU1W2YDRYb+8
   cdKHLpW36QbI6Q5FfBuWKZSI2hSyUvypTpCoAYcHXFv0ns+tRtg0EPNNj+
   lyGjKQWtFi6K8IXXISoDy42FblKIlaA7gEriY=
<!--NeedCopy-->
```


GUI を使用したポリシー・ベースの RSA 暗号化

GUI では、次のタスクを実行できます。

GUI を使用して **RSA** 公開キーをポリシー式として追加するには、次の手順を実行します。

1. Citrix ADC アプライアンスにサインインし、「構成」>「**AppExpert** 」>「高度な式」の順に選択します。
2. 詳細ペインで、[**Add**] をクリックして、RSA 公開キーを高度なポリシー式として定義します。
3.「式の作成」ページで、次のパラメータを設定します。
   a) 式名。高度な式の名前。
   b) 式。式エディタを使用して、RSA 公開キーを高度な式として定義します。
   c) コメント。式の簡単な説明。
4. ［作成］をクリックします。

GUI を使用して **HTTP** ヘッダー要求を暗号化するアクションを書き換える手順は、次のとおりです。

1. Citrix ADC アプライアンスにサインインし、「構成」>「**AppExpert** 」>「書き換え」**>**「アクション」の順に選択します。
2. 詳細ウィンドウで、[追加] をクリックして書き換えアクションを追加します。
3.「書き換えアクションの作成」画面で、次のパラメータを設定します。
   a) Name：書き換えアクションの名前。
   b) タイプ。「INSERT_HTTP_HEADER」としてアクションタイプを選択します。
   c) アクションタイプを使用して、ヘッダーを挿入します。書き換えが必要な HTTP ヘッダーの名前を入力します。
   d) 式。アクションに関連付けられている詳細ポリシー式の名前。
   e) コメント。書き換えアクションの簡単な説明。
4. ［作成］をクリックします。

GUI を使用して書き換えの詳細ポリシーを追加するには、次の手順を実行します。

1. Citrix ADC アプライアンスにサインインし、「構成」>「**AppExpert** 」>「書き換え」>「ポリシー」の順に選択します。
2. [書き換えポリシー] ページで、[追加] をクリックして書き換えポリシーを追加します。
3. [書き換えポリシーの作成**]** ページで、次のパラメータを設定します。
   a) Name：書き換えポリシーの名前。
   b) 操作。要求または応答がこの書き換えポリシーと一致する場合に実行する書き換えアクションの名前。
   c) ログアクション。要求がこのポリシーに一致する場合に使用するメッセージログアクションの名前。
   d) 未定義の結果アクション。ポリシー評価の結果が未定義の場合に実行するアクション。
   e) 式。アクションをトリガーする高度なポリシー式の名前。
   f) コメント。書き換えアクションの簡単な説明。
4. ［作成］をクリックします。

GUI を使用して書き換えポリシーをグローバルにバインドするには、次の手順を実行します。

1. Citrix ADC アプライアンスにサインインし、「構成」>「**AppExpert** 」>「書き換え」>「ポリシー」の順に選

択します。

2. [ポリシーの書き換え] 画面で、バインドする書き換えポリシーを選択し、[**Policy Manager**] をクリックします。
3. [ポリシーマネージャの書き換え] ページの [バインドポイント] セクションで、次のパラメータを設定します。
   a) バインドポイント。バインドポイントを [デフォルトグローバル] として選択します。
   b) プロトコル。プロトコルの種類を HTTP として選択します。
   c) [接続タイプ]。接続タイプを「要求」として選択します。
   d) [続行] をクリックして、[ポリシーのバインド] セクションを表示します。
   e) [ポリシーのバインド] セクションで、書き換えポリシーを選択し、バインドパラメータを設定します。
4. ［バインド］ をクリックします。

例 **11:** パディング操作なしのポリシーベースの **RSA** 暗号化

October 7, 2021

PKEY_ENCRYPT_PEM_NO_PADDING () ポリシー関数は、RSA 暗号化を実行する前に、パディング操作なしで RSA アルゴリズムを使用します。ポリシー関数は、PKEY_ENCRYPT_PEM () 関数と同じように動作しますが、RSA_PKCS1_PADDING ではなく RSA_NO_PADDING メソッドを使用する点が異なります。pkey パラメータは、PEM エンコードされた RSA 公開キーを持つテキスト文字列です。PKEY_ENCRYPT_PEM () と同様に、キーに対してポリシー式を使用できます。

この機能は、書き換えポリシーを使用して実装できます。これを行うには、次のタスクを完了する必要があります。

1. RSA 公開キーをポリシー式として追加します。
2. 書き換えアクションを作成します。

Citrix ADC コマンドインターフェイスを使用したポリシーベースの **RSA** 暗号化

Citrix ADC コマンドインターフェイスを使用してポリシーベースの RSA 暗号化を構成するには、次のタスクを実行します。

Citrix ADC コマンドインターフェイスを使用して、パディングポリシー式なしで **RSA** 公開キーを追加するには:

```
add expression rsa_pub_key_4096 '"-----BEGIN RSA PUBLIC KEY-----" + "
    MIICCgKCAgEArrwBldKd48xrpOSRPMrg+eNAO0ODU6t5b/WYQLdElqNv7WpefBrA" +
    "nwI2s619gEU1r4zoLqL7l5ALtt5Z+F0JBYfOzBzOky0GtEJ5iX5GP4QxT65J3nHH" +
     "4MTF3acmjvXxclmaKXEFlaVIzW7FTr3Luw/CnOjflAB403Q6F9VBVvQmOVYWnqoI"
    + "+0q1VIg6Q1pAcvdKBiOf85BBofE5EIbZ/1Jt0CdbsV568l+8ve7BnSUncFHoRR3O"
     + "/VfSsDuNWZf7n3RNMzxEuIA72UGPzNYFQzvcPOdzd0aN7jAXw0mgC/NSvKzGKHlo
    " + "mUYYBzlVQdDMZWnd6jSzsBRXSXxsNEy/
    RuXwplrA5epo7JdCoMkfeI4vUXm6MNr8" + "
```



```
      TQdFqIc1pdnOsbRf9ec62XbcfR7P8CDTsmLSaagx3rjenPdB+LTWKw2VUF+YONIg” +
      ”jM3fyFef9ovVhLhS5HvMqFGs8P75W+d7BOIbIu3EngACiEJOpYSsETD4WgPK6Iyv” +
       ”j6cxsLeYMtElTb0fBIIqysCHdmjF3M1lqdqp4dKs3+W798GJZYM5MxZKUzrBi0Xu”
      + ”e7GtSh2aimsfQureUD+0z0RN2umeDsYcA1ghXMclDP+jLS1lnrv0Yvo+TKcm9b8G”
       + ”uR/drbCrCsGyWFW+bsAu3AWz9S6TePurP5unRmNNvXpH5DRgsYl3d50CAwEAAQ
      ==” + ”-----END RSA PUBLIC KEY-----”
2 <!--NeedCopy-->
```

Citrix ADC コマンドインターフェイスを使用して、パディングなしポリシー式の書き換えアクションを追加するには:

```
add rewrite action rsa_encrypt_act insertHttpHeader encrypted 'HTTP.REQ.
HEADER(”plaintext”).PKEY_ENCRYPT_PEM_NO_PADDING(rsa_pub_key_4096)
```

GUI を使用した、パディングなしのポリシー・ベースの RSA 暗号化

GUI では、次のタスクを実行できます。

GUI を使用して、パディングなし操作用の **RSA** 公開キーをポリシー式として追加するには、次の手順を実行します。

1. Citrix ADC アプライアンスにサインインし、「構成」>「**AppExpert** 」>「高度な式」の順に選択します。
2. 詳細ペインで、[**Add**] をクリックして、RSA 公開キーを高度なポリシー式として定義します。
3. 「式の作成」ページで、次のパラメータを設定します。
   a) 式名。高度な式の名前。
   b) 式。式エディタを使用して、RSA 公開キーを高度な式として定義します。
      注: ポリシー式の最大文字列長は 255 文字です。1024 ビットよりも長いキーについては、キーをより小さなチャンクに分割し、チャンクを「chunk1」+「chunk2」+ として連結する必要があります。
   c) コメント。式の簡単な説明。
4. [作成] をクリックします。

GUI を使用してアクションの書き換えを追加するには、次のステップを実行します。

1. Citrix ADC アプライアンスにサインインし、「構成」>「**AppExpert** 」>「書き換え」**>**「アクション」の順に選択します。
2. 詳細ウィンドウで、[追加] をクリックして書き換えアクションを追加します。
3. 「書き換えアクションの作成」画面で、次のパラメータを設定します。
   a) Name: 書き換えアクションの名前。
   b) タイプ。「INSERT_HTTP_HEADER」としてアクションタイプを選択します。
   c) アクションタイプを使用して、ヘッダーを挿入します。書き換えが必要な HTTP ヘッダーの名前を入力します。
   d) 式。アクションに関連付けられている詳細ポリシー式の名前。
   e) コメント。書き換えアクションの簡単な説明。
4. [作成] をクリックします。

例 12：Citrix ADC アプライアンスでクライアント要求内のホスト名および URL を変更するように書き換えを構成する

October 7, 2021

Citrix ADC アプライアンスの書き換え機能は、クライアント要求で使用可能な URL を、バックエンドサーバーが認識できる別の URL に変換するために使用されます。書き換え機能を使用すると、次の利点を得ることができます。

- クライアントによって要求されるリソースへの実際の URL を非表示にすることで、セキュリティを強化します。
- 不正なユーザアクセスがネットワークリソースにアクセスするのを防ぎます。

現在の組織が別の組織によって取得されている例を考えてみましょう。管理者は、買収された組織のすべてのユーザーに新しい Web アドレスについて通知することは難しい仕事になります。このシナリオでは、書き換え機能を使用すると、取得した組織のウェブサイトのためのクライアント要求でホスト名や URL を変更することが便利になります。rewrite を使用すると、Web サイトがメンテナンス中のときに、クライアント要求の URL を一時的に変更できます。

次のセクションでは、書き換え機能を使用してクライアント要求のホスト名と URL を変更する手順について説明します。

ユーザーが Web ブラウザに`http://www.example.com` URL を入力する例を考えてみましょう。Web サイト管理者は、Citrix ADC アプライアンスにクライアント要求内の前の URL をとして変換することを望んでいます`http://myexample.example.net.in/resource/inventory/s?t=112`。

上記の例では、ウェブサイト管理者は、Citrix ADC アプライアンスで「example.com」のドメイン名を「myexample.example.net.in」に置き換え、URL を「resource/inventory/s?t=112」に置き換えることを望んでいます。

CLI を使用して、次の操作を実行します

1. SSH を使用して Citrix ADC アプライアンスにログオンします。
2. 書き換えアクションを追加します。
   - `add rewrite action rewrite_host_hdr_act replace "HTTP.REQ.HEADER(\"Host\")""\"myexample.example.net.in\""`
   - `add rewrite action rewrite_url_act replace HTTP.REQ.URL.PATH_AND_QUERY "\"/resource/inventory/s?t=112\""`
3. 書き換えアクションに書き換えポリシーを追加します。
   - `add rewrite policy rewrite_host_hdr_pol "HTTP.REQ.HEADER(\"Host\").CONTAINS(\"www.example.com\")"rewrite_host_hdr_act`

- `add rewrite policy rewrite_url_pol "HTTP.REQ.HEADER(\"Host\").CONTAINS(\"www.example.com\")"rewrite_url_act`

4. リライトポリシーを仮想サーバーにバインドします。

   - `bind lb vserver rewrite_LB -policyName rewrite_host_hdr_pol -priority 10 -gotoPriorityExpression 20 -type REQUEST`
   - `bind lb vserver rewrite_LB -policyName rewrite_url_pol -priority 20 -gotoPriorityExpression END -type REQUEST`

URL 変換

October 7, 2021

URL 変換機能では、外部ユーザーが見る外部バージョンから、Web サーバーおよび IT スタッフのみが見る内部 URL に、指定された要求内のすべての URL を変更する方法を提供します。ネットワーク構造をユーザーに公開することなく、ユーザー要求をシームレスにリダイレクトできます。また、ユーザーが覚えにくい複雑な内部 URL を変更して、よりシンプルで覚えやすい外部 URL に変更することもできます。

> 注
>
> URL 変換機能を使用する前に、書き換え機能を有効にする必要があります。書き換え機能を有効にするには、書き換え機能の有効化を参照してください。
>
> URL 変換機能は、HTML レスポンス本文の URL を書き換え、JavaScript やその他の変数には適用されません。

URL 変換の設定を開始するには、それぞれが特定の変換を記述するプロファイルを作成します。各プロファイル内に、変換を詳細に記述する 1 つ以上のアクションを作成します。次に、変換する HTTP 要求のタイプを識別するポリシーを作成し、各ポリシーを適切なプロファイルに関連付けます。最後に、各ポリシーをグローバルにバインドして有効にします。

URL 変換プロファイルの設定

October 7, 2021

プロファイルは、特定の URL 変換を一連のアクションとして記述します。プロファイルは、主にアクションのコンテナとして機能し、アクションが実行される順序を決定します。ほとんどの変換では、外部ホスト名とオプションのパスが、異なる内部ホスト名とパスに変換されます。最も有用な変換は単純で、必要な操作は 1 つだけですが、複数のアクションを使用して複雑な変換を実行できます。

アクションを作成してプロファイルに追加することはできません。最初にプロファイルを作成してから、それにアクションを追加する必要があります。CLI では、アクションの作成とアクションの設定は別個の手順です。プロファイルの作成とプロファイルの設定は、CLI と設定ユーティリティーの両方で別々の手順です。

Citrix ADC コマンドラインを使用して URL 変換プロファイルを作成するには

Citrix ADC コマンドプロンプトで、以下のコマンドを以下の順に入力して、URL 変換プロファイルを作成し、構成を確認します。次に、2 番目と 3 番目のコマンドを繰り返して、追加のアクションを設定できます。

- add transform profile <profileName> -type URL [-onlyTransformAbsURLinBody (ON|OFF)] \[-comment <comment>]
- add transform action <name> <profileName> <priority>
- set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainFrom <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment ”<string>”]
- show transform profile <name>

例:

```
> add transform profile shoppingcart -type URL
 Done
> add transform action actshopping shoppingcart 1000
 Done
> set transform action actshopping -priority 1000 -reqUrlFrom 'shopping
   .example.com' -reqUrlInto 'www.example.net/shopping' -resUrlFrom '
   www.example.net/shopping' -resUrlInto 'shopping.example.com' -
   cookieDomainFrom 'example.com' -cookieDomainInto 'example.net' -
   state ENABLED -comment 'URL transformation for shopping cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```


Citrix ADC コマンドラインを使用して既存の URL 変換プロファイルまたはアクションを変更するには

Citrix ADC コマンドプロンプトで次のコマンドを入力して、既存の URL 変換プロファイルまたはアクションを変更し、構成を確認します。

注：それぞれトランスフォームプロファイルまたは set トランスフォームアクションコマンドを使用します。set transform profile コマンドは、add transform profile コマンドと同じ引数を取ります。set transform アクションは、初期設定に使用されたコマンドと同じです。

- set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment ”<string>”]
- show transform profile <name>

例:

```
> set transform action actshopping -priority 1000 -reqUrlFrom '
    searching.example.net' -reqUrlInto 'www.example.net/searching' -
    resUrlFrom 'www.example.net/searching' -resUrlInto 'searching.
    example.com' -cookieDomainInto 'example.net' -state ENABLED -comment
     'URL transformation for searching cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```

Citrix ADC コマンドラインを使用して URL 変換プロファイルとアクションを削除するには

まず、各アクションに対して次のコマンドを 1 回入力して、そのプロファイルに関連付けられているすべてのアクションを削除します。

- rm transform action <name>プロファイルに関連付けられたすべてのアクションを削除したら、以下に示すようにプロファイルを削除します。
- rm transform profile <name>

構成ユーティリティを使用して **URL** 変換プロファイルを作成するには

1. ナビゲーションウィンドウで、[書き換え] を展開し、[URL 変換] を展開し、[プロファイル] をクリックします。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**URL** 変換プロファイルの作成] ダイアログボックスで、パラメータの値を入力または選択します。ダイアログ・ボックスの内容は、「URL 変換プロファイルを構成するためのパラメータ」で説明されているパラメータに対応しています（アスタリスクは必須パラメータを示します）。
   - 名前 *—name
   - Comment-コメント
   - レスポンス本文内の絶対 URL のみを変換します。
4. [**Create**] をクリックしてから、[**Close**] をクリックします。プロファイルが正常に構成されたことを示すメッセージがステータスバーに表示されます。

構成ユーティリティを使用して **URL** 変換プロファイルとアクションを構成するには

1. ナビゲーションウィンドウで、[書き換え] を展開し、[URL 変換] を展開し、[プロファイル] をクリックします。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[開く] をクリックします。
3. [**URL** 変換プロファイルの設定] ダイアログボックスで、次のいずれかの操作を行います。
   - 新しいアクションを作成するには、[追加] をクリックします。
   - 既存のアクションを変更するには、アクションを選択して、[開く] をクリックします。
4. パラメータの値を入力または選択して、「**URL** 変換アクションの作成」または「**URL** 変換アクションの修正」ダイアログボックスに入力します。ダイアログ・ボックスの内容は、「URL 変換プロファイルを構成するためのパラメータ」で説明されているパラメータに対応しています（アスタリスクは必須パラメータを示します）。
   - アクション名 *—name
   - コメント: comment
   - 優先度 *: priority
   - リクエスト URL の送信元-reqUrlFrom
   - リクエスト URL の先-reqUrlInto
   - からのレスポンス URL-resUrlFrom
   - レスポンス URL の入力先-resUrlInto
   - クッキードメインから—cookieDomainFrom
   - クッキードメインへ—cookieDomainInto
   - 有効-state
5. 変更を保存します。
   - 新しいアクションを作成する場合は、[作成]、[閉じる] の順にクリックします。
   - 既存のアクションを変更する場合は、「**OK**」をクリックします。
     プロファイルが正常に構成されたことを示すメッセージがステータスバーに表示されます。
6. 手順 3 ～5 を繰り返して、追加のアクションを作成または変更します。
7. アクションを削除するには、アクションを選択し、[削除] をクリックします。プロンプトが表示されたら、[OK] をクリックして削除を確認します。

8. 「**OK**」をクリックして変更を保存し、「URL 変換プロファイルの修正」ダイアログボックスを閉じます。
9. プロファイルを削除するには、詳細ペインでプロファイルを選択し、[削除] をクリックします。プロンプトが表示されたら、[**OK**] をクリックして削除を確認します。

URL 変換ポリシーの設定

October 7, 2021

URL 変換プロファイルを作成したら、次に URL 変換ポリシーを作成し、Citrix ADC がそのプロファイルを使用して変換する要求と応答を選択します。URL 変換では、各要求とその応答が 1 つの単位と見なされるため、URL 変換ポリシーは要求を受信したときにのみ評価されます。ポリシーが一致すると、Citrix ADC は要求と応答の両方を変換します。

注:URL 変換および書き換え機能は、リクエスト処理中に同じ HTTP ヘッダー上で動作することはできません。このため、リクエストに URL 変換を適用する場合は、変更する HTTP ヘッダーがどれも書き換えアクションによって操作されていないことを確認する必要があります。

Citrix ADC コマンドラインを使用して URL 変換ポリシーを構成するには

新しいポリシーを作成する必要があります。コマンドラインでは、既存のポリシーだけを削除できます。Citrix ADC コマンドプロンプトで次のコマンドを入力して、URL 変換ポリシーを構成し、構成を確認します。

- `<add transform policy <name> <rule> <profileName>`
- `<show transform policy <name>`

例:

```
1  > add transform policy polsearch  HTTP.REQ.URL.SUFFIX.EQ(”Searching”)
       prosearching
2   Done
3  > show transform policy polsearch
4  1)      Name: polsearch
5          Rule: HTTP.REQ.URL.SUFFIX.EQ(”Searching”)
6          Profile: prosearching
7          Priority: 0
8          Hits: 0
9   Done
10 <!--NeedCopy-->
```


Citrix ADC コマンドラインを使用して **URL** 変換ポリシーを削除するには

Citrix ADC コマンドプロンプトで次のコマンドを入力して、URL 変換ポリシーを削除します。

```
rm transform policy <name>
```

例:

```
1 > rm transform policy polsearch
2 Done
3 <!--NeedCopy-->
```

構成ユーティリティを使用して **URL** 変換ポリシーを構成するには

1. ナビゲーションウィンドウで、[書き換え]、[URL 変換] の順に展開し、[ポリシー] をクリックします。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいポリシーを作成するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
3. [**URL** 変換ポリシーの作成] または [**URL** 変換ポリシーの構成] ダイアログボックスで、パラメータの値を入力または選択します。ダイアログ・ボックスの内容は、「URL 変換ポリシーを構成するためのパラメータ」で説明されているパラメータに対応しています（アスタリスクは必須パラメータを示します）。
   - Name*：name（以前に設定したポリシーでは変更できません）
   - プロファイル *—profileName
   - 式—rule

   新しいポリシーの式の作成についてヘルプが必要な場合は、[Expression] テキストボックスにカーソルがある状態で、Ctrl キーを押したままスペースバーを押します。式を作成するには、下記の説明に従って直接入力するか、[式の追加] ダイアログボックスを使用します。
4. [プレフィックス] をクリックし、式のプレフィックスを選択します。

   選択肢は次のとおりです：
   - HTTP：HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
   - SYS：保護されている Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - CLIENT：要求を送信したコンピュータ。リクエストの送信者の一部を調べる場合は、これを選択します。
   - SERVER—要求が送信されたコンピュータ。要求の受信者のいくつかの側面を調べる場合は、これを選択します。

- URL：リクエストの URL。リクエストの送信先の URL の一部を調べる場合は、これを選択します。
- TEXT：要求内の任意のテキスト文字列。リクエスト内のテキスト文字列を調べる場合に選択します。
- TARGET — リクエストのターゲット。リクエストターゲットのいくつかの側面を調べる場合は、これを選択します。

プリフィックスを選択すると、Citrix ADC では 2 つの部分からなるプロンプトウィンドウが表示され、上部に次の選択肢が表示され、下部に選択した選択肢が意味する簡単な説明が表示されます。選択肢は、選択したプレフィクスによって異なります。

5. 次の用語を選択します。

プレフィックスとして HTTP を選択した場合、REQ は HTTP 要求を指定し、RES は HTTP 応答を指定します。RES は HTTP 応答を指定します。別のプレフィックスを選択した場合、選択肢はより多様になります。特定の選択肢に関するヘルプを表示するには、その選択肢を 1 回クリックすると、下部のプロンプトウィンドウにその選択肢に関する情報が表示されます。

目的の選択肢が確定したら、ダブルクリックして [式] ウィンドウに挿入します。

1. ピリオドを入力し、前のリストボックスの右側に表示されるリストボックスから用語の選択を続行します。式が終了するまで、値の入力を求めるプロンプトが表示されるテキストボックスに、適切なテキスト文字列または数値を入力します。
2. 新しいポリシーを作成するか、既存のポリシーを変更するかに応じて、[**Create**] または [**OK**] をクリックします。
3. [閉じる] をクリックします。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

[式の追加] ダイアログボックスを使用して式を追加するには

1. [レスポンダーアクションの作成] または [レスポンダーアクションの構成] ダイアログボックスで、[追加] をクリックします。
2. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。
   - HTTP HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。
   - SYS。保護された Web サイト。リクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
   - CLIENT。要求を送信したコンピュータ。リクエストの送信者の一部を調べる場合は、これを選択します。
   - SERVER。要求が送信されたコンピュータ。要求の受信者のいくつかの側面を調べる場合は、これを選択します。
   - URL。リクエストの URL。リクエストの送信先の URL の一部を調べる場合は、これを選択します。

- TEXT. リクエスト内の任意のテキスト文字列。リクエスト内のテキスト文字列を調べる場合に選択します。
- TARGET. リクエストのターゲット。リクエストターゲットのいくつかの側面を調べる場合は、これを選択します。
  選択すると、右端のリストボックスには、式の次の部分に適した用語が一覧表示されます。

3. 2 番目のリストボックスで、式の 2 番目の項を選択します。選択内容は、前のステップで行った選択内容によって異なり、コンテキストに適切です。2 番目の選択を行うと、[式を構築] ウィンドウの下にある [ヘルプ] ウィンドウ (空白) に、選択した用語の目的と使用方法を示すヘルプが表示されます。
4. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。

URL 変換ポリシーのグローバルバインディング

October 7, 2021

URL 変換ポリシーを設定したら、それらをグローバルまたはバインドポイントにバインドして有効にします。バインド後、URL 変換ポリシーに一致する要求または応答は、そのポリシーに関連付けられたプロファイルによって変換されます。

ポリシーをバインドするときは、そのポリシーに優先度を割り当てます。優先順位によって、定義したポリシーが評価される順序が決まります。プライオリティは任意の正の整数に設定できます。Citrix ADC OS では、ポリシーの優先順位は逆の順序で機能します。値が大きいほど、優先順位は低くなります。

URL 変換機能では、要求が一致する最初のポリシーだけが実装され、一致する可能性のある追加のポリシーは実装されないため、意図した結果を得るにはポリシーの優先順位が重要です。最初のポリシーに低い優先度（1000 など）を設定した場合、より高い優先度を持つ他のポリシーが要求と一致しない場合にのみ、Citrix ADC にポリシーを実行するように指示します。最初のポリシーに高い優先度（1 など）を指定した場合は、Citrix ADC に最初にそのポリシーを実行するように指示し、一致するその他のポリシーもスキップします。ポリシーをグローバルにバインドするときに、各ポリシー間に 50 または 100 の間隔で優先順位を設定することで、優先順位を再割り当てすることなく、他のポリシーを任意の順序で追加できる十分な余裕を残すことができます。

注:URL 変換ポリシーは、TCP ベースの仮想サーバーにバインドできません。

Citrix ADC コマンドラインを使用して URL 変換ポリシーをバインドするには

Citrix ADC コマンドプロンプトで次のコマンドを入力して、URL 変換ポリシーをグローバルにバインドし、構成を確認します。

- `bind transform global <policyName> <priority>`
- `show transform global`

例:

```
> bind transform global polisearching 100
 Done
> show transform global
1)      Policy Name: polisearching
        Priority: 100

 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **URL** 変換ポリシーをバインドするには

1. ナビゲーションウィンドウで、[書き換え] を展開し、[URL 変換] を展開し、[** ポリシー] をクリックします。
2. 詳細ペインで、[ポリシーマネージャ] をクリックします。
3. [**Transform Policy Manager**] ダイアログボックスで、ポリシー ** をバインドするバインドポイントを選択します。選択肢は次のとおりです。
   - [グローバルをオーバーライド]。このバインドポイントにバインドされたポリシーは、Citrix ADC アプライアンスのすべてのインターフェイスからのすべてのトラフィックを処理し、他のポリシーの前に適用されます。
   - **LB** 仮想サーバー。負荷分散仮想サーバーにバインドされたポリシーは、その負荷分散仮想サーバーによって処理されるトラフィックにのみ適用され、Default Global ポリシーの前に適用されます。[LB Virtual Server] を選択した後、このポリシーをバインドする特定の負荷分散仮想サーバーも選択する必要があります。
   - **CS** 仮想サーバ。コンテンツスイッチ仮想サーバーにバインドされたポリシーは、そのコンテンツスイッチ仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトグローバルポリシーの前に適用されます。[CS Virtual Server] を選択した後、このポリシーをバインドする特定のコンテンツスイッチ仮想サーバーも選択する必要があります。
   - デフォルトグローバル。このバインドポイントにバインドされたポリシーは、Citrix ADC アプライアンスのすべてのインターフェイスからのすべてのトラフィックを処理します。
   - ポリシーラベル。ポリシーラベルにバインドされたポリシーは、ポリシーラベルがルーティングするトラフィックを処理します。ポリシーラベルは、このトラフィックにポリシーを適用する順序を制御します。
4. [Insert Policy] を選択して新しい行を挿入し、使用可能なバインドされていない URL 変換ポリシーをすべて含むドロップダウンリストを表示します。
5. バインドするポリシーを選択するか、[New Policy] を選択して新しいポリシーを作成します。選択または作成したポリシーが、グローバルにバインドされた URL 変換ポリシーのリストに挿入されます。
6. バインドに追加の調整を行います。
   - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。

[優先度を再生成] を選択して、優先度を均等に再番号付けすることもできます。

- ポリシー式を変更するには、そのフィールドをダブルクリックして [Configure Transform Policy] ダイアログボックスを開き、ポリシー式を編集できます。
- [Goto Expression] を設定するには、[Goto Expression] 列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。
- 「呼び出し」オプションを設定するには、「呼び出し」列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。

7. ステップ 3 ～6 を繰り返して、グローバルにバインドする URL 変換ポリシーを追加します。
8. [**OK**] をクリックして変更を保存します。ポリシーが正常に構成されたことを示すメッセージがステータスバーに表示されます。

リライト機能に対する **RADIUS** のサポート

October 7, 2021

Citrix ADC の式言語には、要求と応答の RADIUS メッセージから情報を抽出したり、操作したりできる式が含まれています。これらの式を使用すると、書き換え機能を使用して、RADIUS メッセージを宛先に送信する前に RADIUS メッセージの一部を変更できます。書き換えポリシーおよびアクションでは、RADIUS メッセージに適切な、または関連する任意の式を使用できます。使用可能な式を使用すると、RADIUS メッセージタイプを識別し、接続から任意の属性値ペア（AVP）を抽出し、RADIUS AVP を変更できます。RADIUS 接続用のポリシーラベルを作成することもできます。

リライトルールでは、新しい RADIUS 式をさまざまな目的で使用できます。たとえば、次のようなことができます。

- RADIUS ユーザ名 AVP のドメイン部分を削除して、シングルサインオン（SSO）を簡素化します。
- 電話会社の操作で使用される MSISDN フィールドなど、ベンダー固有の AVP を挿入し、加入者情報を格納します。

ポリシーラベルを作成して、特定のタイプの RADIUS 要求を、それらの要求に適した一連のポリシーを通してルーティングすることもできます。

> 注:
>
> リライト用の RADIUS には、次の制限事項があります。

- Citrix ADC は、書き換えられた RADIUS 要求または応答を再署名しません。RADIUS 認証サーバが署名付き RADIUS メッセージを必要とする場合、認証は失敗します。
- 現在使用可能な RADIUS 式は、RADIUS IPv6 属性では機能しません。

RADIUS をサポートする式に関する Citrix ADC マニュアルでは、RADIUS 通信の基本構造と目的を理解していることを前提としています。RADIUS の詳細については、RADIUS サーバのマニュアルを参照するか、RADIUS プロトコルの概要をオンラインで検索してください。

RADIUS の書き換えポリシーの設定

以下の手順では、Citrix ADC コマンドラインを使用して書き換えアクションとポリシーを構成し、書き換え固有のグローバルバインドポイントにポリシーをバインドします。

Rewrite アクションとポリシーを設定し、ポリシーをバインドするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

- `add rewrite action <actName> <actType>`
- `add rewrite policy <polName> <rule> <actName>`
- `bind rewrite policy <polName> <priority> <nextExpr> -type <bindPoint>` ここで、<bindPoint>は、書き換え固有のグローバルバインドポイントの 1 つを表します。

リライト用の **RADIUS** 式

書き換え構成では、次の Citrix ADC 式を使用して、RADIUS 要求または応答のさまざまな部分を参照できます。

接続のタイプの識別:

- `RADIUS.IS_CLIENT`

  接続が RADIUS クライアント (要求) メッセージである場合は TRUE を返します。

- `RADIUS.IS_SERVER`

  接続が RADIUS サーバー (応答) メッセージである場合は TRUE を返します。

リクエスト式:

- `RADIUS.REQ.CODE`

  RADIUS 要求タイプに対応する番号を返します。num_at クラスの導関数です。たとえば、RADIUS アクセス要求は 1 を返します。RADIUS アカウンティング要求は 4 を返します。

- `RADIUS.REQ.LENGTH`

  ヘッダーを含む RADIUS 要求の長さを返します。
  num_at クラスの導関数です。

- `RADIUS.REQ.IDENTIFIER`

  RADIUS 要求 ID を返します。この番号は、各要求に割り当てられた番号で、要求を対応する応答に一致させることができます。
  num_at クラスの導関数です。

- `RADIUS.REQ.AVP(<AVP Code No>).VALUE`

  型
  text_t の文字列として、この AVP の最初の出現の値を返します。

- `RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)`

  AVP の指定されたインスタンスを RAVP_t 型の文字列として返します。特定の RADIUS AVP は、RADIUS メッセージ内で複数回発生する可能性があります。INSTANCE (0) は最初のインスタンスを返し、INSTANCE (1) は 2 番目のインスタンスを返します。以下同様に最大 16 個のインスタンスを返します。

- `RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)`

  AVP の指定されたインスタンスの値を
  text_t 型の文字列として返します。

- `RADIUS.REQ.AVP(<AVP code no>).COUNT`

  RADIUS 接続内の特定の AVP のインスタンス数を整数で返します。

- `RADIUS.REQ.AVP(<AVP code no>).EXISTS`

  指定されたタイプの AVP がメッセージ内に存在する場合は TRUE、存在しない場合は FALSE を返します。

応答式:

RADIUS 応答式は、REQ を置き換える点を除いて、RADIUS 要求式と同じです。

AVP 値のタイプキャスト:

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィクス、時刻の各データ型にタイプキャストする式をサポートしています。構文は、他の Citrix ADC 型キャスト式と同じです。

例:

ADC は、RADIUS AVP 値をテキスト、整数、符号なし整数、ロング、符号なしロング、ipv4 アドレス、ipv6 アドレス、ipv6 プレフィクス、時刻の各データ型にタイプキャストする式をサポートしています。構文は、他の Citrix ADC 型キャスト式と同じです。

```
1 RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
2 <!--NeedCopy-->
```

AVP 型式:

Citrix ADC は、RFC2865 および RFC2866 に記載されている割り当てられた整数コードを使用して、RADIUS AVP 値を抽出する式をサポートしています。テキストエイリアスを使用して、同じタスクを実行することもできます。次に、いくつかの例を示します。

- `RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value`

  RADIUS ユーザ名の値を抽出します。

- `RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT\\_SESSION\\_ID.value`

  メッセージからアカウントセッション ID AVP（コード 44）を抽出します。

- RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR_SPECIFIC.VALUE

  ベンダー固有の値を抽出します。

最も一般的に使用される RADIUS AVP の値は、同じ方法で抽出できます。

RADIUS バインドポイント:

RADIUS 式を含むポリシーには、4 つのグローバルバインドポイントを使用できます。

- RADIUS_REQ_OVERRIDE

  優先度/優先要求ポリシーキュー。

- RADIUS_REQ_DEFAULT

  標準要求ポリシーキュー。

- RADIUS_RES_OVERRIDE

  優先度/優先応答ポリシーキュー。

- RADIUS_RES_DEFAULT

  標準応答ポリシーキュー。

RADIUS 書き換え固有の式:

- RADIUS.NEW_AVP

  指定された RADIUS AVP を文字列として返します。

- RADIUS.NEW_AVP_INTEGER32

  指定された RADIUS AVP を整数として返します。

- RADIUS.NEW_AVP_UNSIGNED32

  指定された RADIUS AVP を符号なし整数として返します。

- RADIUS.NEW_VENDOR_SPEC_AVP(<ID>, <definition>)

  指定された拡張ベンダー固有の AVP を接続に追加します。<ID>には、長い数値を代入します。<definition>には、AVP のデータを含む文字列を置き換えます。

- RADIUS.REQ.AVP_START

  RADIUS ヘッダーの終わりと AVP の開始との間の位置を返します。書き換えアクションで使用されます。

  例:

```
1     add rewrite action insert1 insert_after radius.req.avp_start radius
          .new_avp(33, "NEW AVP")
2  <!--NeedCopy-->
```


- RADIUS.REQ.AVP_END

  RADIUS メッセージ内の RADIUS メッセージの最後（またはすべての AVP の最後）の位置を返します。書き換えアクションを実行するときに使用します。

  例:

```
1     add rewrite action insert2 insert_before radius.req.avp_end "radius
          .new_avp(33, "NEW AVP")"
2 <!--NeedCopy-->
```

- RADIUS.REQ.AVP_LIST

  RADIUS メッセージ内の AVP の開始位置と、ヘッダーを除いた RADIUS メッセージの長さを返します。つまり、RADIUS メッセージ内のすべての AVP を返します。書き換えアクションを実行するために使用されます。

  例:

```
1     add rewrite action insert3 insert_before_all radius.req.avp_list "
          radius.new_avp(33, "NEW AVP")" -search "avp(33)"
2 <!--NeedCopy-->
```

RADIUS の有効な書き換えアクションタイプ:

RADIUS 式で使用できるリライトアクションタイプは次のとおりです。

- INSERT_AFTER
- INSERT_BEFORE
- INSERT_AFTER_ALL
- INSERT_BEFORE_ALL
- 削除
- DELETE_ALL
- REPLACE
- REPLACE_ALL

すべてINSERT_ actionsを使用して、RADIUS AVP を RADIUS 接続に挿入できます。

使用例

次に、書き換えを伴う RADIUS の使用例を示します。

ユーザ名 **AVP** の書き換え

RADIUS ユーザ名 AVP からドメインストリングを削除するように書き換え機能を設定するには、まず次の例に示すように、rewrite REPLACE アクションを作成します。すべての RADIUS 要求を選択する書き換えポリシーでアクションを使用します。ポリシーをグローバルバインドポイントにバインドします。その場合は、優先度を適切なレベルに設定して、ブロックまたは拒否ポリシーを最初に有効にします。ただし、ブロックまたは拒否されていないすべての要求が書き換えられます。ポリシー評価を続行するには、「次へ進む」式（gotoPriorityExpr）を「次へ」に設定し、ポリシーを RADIUS_REQ_DEFAULT キューにアタッチします。

例:

```
add rewrite action rwActRadiusDomainDel replace radius.req.user_name q/
    RADIUS.NEW_AVP(1,RADIUS.REQ.USER_NAME.VALUE.AFTER_STR(" "))/
add rewrite policy RadiusRemoveDomainPol true rwActRadiusDomainDel
<!--NeedCopy-->
```

> 注:
>
> RADIUS の書き換えポリシーは、Gateway 仮想サーバには適用されません。Gateway 仮想サーバがロードバランシングに使用される場合は、RADIUS を構成し、書き換えポリシーを RADIUS ロードバランシング仮想サーバにバインドする必要があります。

ベンダー固有の **AVP** の挿入

MSISDN フィールドの内容を含むベンダー固有の AVP を挿入するように Rewrite アクションを設定するには、まず MSISDN フィールドを要求に挿入する書き換えの INSERT アクションを作成します。すべての RADIUS 要求を選択する Rewrite ポリシーのアクションを使用します。ポリシーをグローバルにバインドし、次の例に示すように、プライオリティを適切なレベルに設定し、その他のパラメータを設定します。

例:

```
add rewrite action rwActRadiusInsMSISDN insert_after radius.req.
    avp_start RADIUS.NEW_VENDOR_SPEC_AVP(<VENDOR ID>, "RADIUS.NEW_AVP(<
    Attribute Code>, <MSISDN>)")
add rewrite policy rwPolRadiusInsMSISDN true rwActRadiusInsMSISDN
bind rewrite global rwPolRadiusInsMSISDN 100 NEXT -type
    RADIUS_REQ_DEFAULT
<!--NeedCopy-->
```


書き換えの直径サポート

October 7, 2021

書き換え機能で Diameter プロトコルがサポートされるようになりました。Rewrite を設定して、HTTP または TCP 要求と応答と同様に Diameter 要求と応答を変更できます。これにより、Rewrite を使用して Diameter 要求のフローを管理し、必要な変更を加えることができます。たとえば、Diameter リクエストの「Origin-Host」値が適切でない場合は、Rewrite を使用して Diameter サーバーが許容できる値に置き換えることができます。

Diameter 要求を修正するように書き換えを設定するには

Diameter 要求の Origin-Host を別の値に置き換えるように書き換え機能を構成するには、コマンドプロンプトで次のコマンドを入力します。

- <add rewrite action <actname> replace “DIAMETER.REQ.AVP(264,\“Citrix ADC.example.net\”)”
  <actname> には新しいアクションの名前を置き換えます。名前は、1 ～127 文字の長さで構成され、文字、数字、およびハイフン (-) およびアンダースコア (_) 記号を使用できます。Citrix ADC.example.net の場合は、元のホスト名の代わりに使用するホストオリジンに置き換えます。
- add rewrite policy <polname> “diameter.req.avp(264).value.eq(\“host.example.com\”)” <actname>
  <polname> は新しいポリシーの名前を置き換えます。と同様に <actname>、名前は 1 ～127 文字の長さで構成され、文字、数字、ハイフン (-) およびアンダースコア (_) 記号を含めることができます。host.example.com の場合は、変更するホストオリジンの名前に置き換えます。<actname> には、作成したアクションの名前を置き換えます。
- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST
  <vservername> には、ポリシーをバインドするロードバランシング仮想サーバの名前を置き換えます。には <polname>、作成したポリシーの名前を置き換えます。では <priority>、ポリシーの優先度を置き換えます。

例:

「host.example.com」のすべての Diameter Host-Origins を「Citrix ADC.example.net」に変更するリライトアクションとポリシーを作成するには、次のアクションとポリシーを追加し、図のようにポリシーをバインドします。

```
1 > add rewrite action rw_act_replace_avp replace ”diameter.req.avp(264)”
      ”diameter.new.avp(264,”Citrix ADC.example.net”)”
2 > add rewrite policy rw_diam_pol ”diameter.req.avp(264).value.eq(”
     client.realm2.net”)” rw_act_replace_avp
3 > bind lb vserver vs1 -policyName rw_diam_pol -priority 10 -type
     REQUEST
4
5 Done
6 <!--NeedCopy-->
```


書き換え機能に対する **DNS** サポート

October 7, 2021

HTTP または TCP 要求と応答の場合と同様に、DNS 要求と応答を変更するように書き換え機能を設定できます。書き換えを使用して DNS 要求のフローを管理し、ヘッダーまたは応答セクションで必要な変更を加えることができます。たとえば、DNS 応答のヘッダーフラグに AA ビットが設定されていない場合、rewrite を使用して DNS 応答に AA ビットを設定し、クライアントに送信できます。

DNS の式

書き換え構成では、次の Citrix ADC 式を使用して、DNS 要求または応答のさまざまな部分を参照できます。

エクスプレッションと説明を参照してください。

DNS バインドポイント

DNS 式を含むポリシーでは、次のグローバルバインドポイントを使用できます。

バインドポイント	説明
DNS_REQ_OVERRIDE	要求ポリシーキューを上書きします。
DNS_REQ_DEFAULT	標準要求ポリシーキュー。
DNS_RES_OVERRIDE	応答ポリシーキューを上書きします。
DNS_RES_DEFAULT	標準応答ポリシーキュー。

デフォルトのバインドポイントに加えて、タイプ DNS_REQ または DNS_RES のポリシーラベルを作成し、それらに DNS ポリシーをバインドできます。

DNS の書き換えアクションタイプ

- **replace_dns_answer_section**: このアクションは、DNS 応答セクションを DNS ポリシーで定義された式に置き換えます。
- **replace_dns_header_field**: DNS 要求のオペコードタイプをチェックします。DNS 要求のオペコードの種類が、指定されたオペコードの種類と一致するかどうかを示す True または False を返します。この操作により、DNS ヘッダーセクションが DNS ポリシーで定義された式に置き換えられます。

DNS の書き換えポリシーの構成

以下の手順では、Citrix ADC コマンドラインを使用して書き換えアクションとポリシーを構成し、書き換え固有のグローバルバインドポイントにポリシーをバインドします。

書き換えアクションとポリシーを構成し、**DNS** のポリシーをバインドします

コマンドプロンプトで、次のコマンドを入力します。

1. `add rewrite action <actName> <actType>`

   <actname>には、新しいアクションの名前を置き換えます。名前の長さは 1 ～127 文字で、文字、数字、ハイフン (-)、アンダースコア (_) を使用できます。<actType>で、DNS 式に用意されている書き換えアクションの種類を指定します。

2. `add rewrite policy <polName> <rule> <actName>`

   <polname>では、新しいポリシーの名前に置き換えます。<actname>では、名前は 1 ～127 文字で、文字、数字、ハイフン（-）、およびアンダースコア（_）を使用できます。<actname>には、作成したアクションの名前を置き換えます。

3. `bind rewrite global <polName> <priority> <gotoPriorityExpression> -type <bindPoint>`

   <polName>には、作成したポリシーの名前を置き換えます。<priority>に、ポリシーのプライオリティを指定します。<bindPoint>では、書き換え固有のグローバルバインドポイントの 1 つを代入します。

例:

DNS 要求の **AA** ビットを設定して、仮想サーバーの負荷を分散します。

次のコマンドは、Citrix ADC アプライアンスが処理するすべてのクエリに対して権限のある DNS サーバーとして機能するように構成します。

```
1 add rewrite action set_aa replace_dns_header_field dns.req.header.flags
     .set(aa)
2 add rewrite policy pol !dns.req.header.flags.is_set(aa)  set_aa
3 bind rewrite global  pol  100  -type dns_res_override
4 <!--NeedCopy-->
```

回答とヘッダーのセクションを変更します。

サーバが NX ドメインで応答する場合、応答を指定した IP アドレスに置き換えるように書き換えアクションを設定できます。NOPOLICY-REWRITE を使用すると、式（ルール）を処理せずに外部バンクを呼び出すことができます。このエントリは、ルールを含まないが、ポリシーラベルまたは仮想サーバ固有のポリシーバンクにエントリを転送するダミーポリシーです。


```
add rewrite action set_aa_res replace_dns_header_field "dns.res.header.flags.set(aa)"
add rewrite action modify_nxdomain_res replace_dns_answer_section "dns.new_rrset_a("10.102.218.160",300)"
add rewrite policy set_res_aa true set_aa_res
add add rewrite policy modify_answer "dns.RES.HEADER.RCODE.EQ(nxdomain) && dns.RES.QUESTION.TYPE.EQ(A)"
modify_nxdomain_res
add rewrite policylabel MODIFY_NODATA dns_res
bind rewrite policylabel MODIFY_NODATA modify_answer 10 END
bind rewrite policylabel MODIFY_NODATA set_res_aa 11 END
bind lb vserver v1 -policyName NOPOLICY-REWRITE -priority 11 -gotoPriorityExpression END -type
RESPONSE -invoke policylabel MODIFY_NODATA
<!--NeedCopy-->
```

制限事項:

- リライトポリシーは、Citrix ADC アプライアンスが DNS プロキシサーバーとして構成されていて、キャッシュミスがある場合にのみ評価されます。
- ヘッダー内の再帰可能 (RA) フラグが YES に設定されている場合、RA フラグは書き換えで変更されません。
- ヘッダーの RA フラグが YES に設定されている場合、書き換え操作に関係なくヘッダーの CD フラグが変更されます。

文字列マップ

October 7, 2021

文字列マップを使用すると、デフォルトのポリシー構文を使用するすべての Citrix ADC 機能でパターンマッチングを実行できます。文字列マップは、キーと値のペアで構成される Citrix ADC エンティティです。キーと値は、ASCII 形式または UTF-8 形式の文字列です。文字列比較では、2 つの新しい関数、`MAP_STRING(<string_map_name>)` および`IS_STRINGMAP_KEY(<string_map_name>)` を使用します。

文字列マップを使用するポリシー設定は、ポリシー式を使用して文字列マッチングを実行するポリシーよりもパフォーマンスが向上し、多数のキーと値のペアで文字列マッチングを実行するために必要なポリシーが少なくなります。文字列マップも直感的で、設定が簡単で、構成が小さくなります。

文字列マップの仕組み

文字列マップは、パターンセットと構造が似ています（パターンセットはインデックス値の文字列へのマッピングを定義し、文字列マップは文字列へのマッピングを定義します）。文字列マップの設定コマンド（add、bind、unbind、

remove、show などのコマンド）は、構文的に構成と似ています。コマンドを実行します。また、パターンセット内のインデックス値と同様に、文字列マップ内の各キーはマップ全体で一意である必要があります。次の表は、url_string_map という文字列マップを示しています。このマップには、キーと値として URL が含まれています。

キー	値
/url_1.html	http://www.redirect_url_1.com/url_1.html
/url_2.html	http://www.redirect_url_2.com/url_2.html
/url_3.html	http://www.redirect_url_1.com/url_1.html

表 1. 文字列マップ「url_ 文字列マップ」

次の表では、文字列マップ内のキーとの文字列マッチングを有効にするために導入された 2 つの関数について説明します。文字列のマッチングは、常にキーで実行されます。さらに、次の関数は、文字列マップ内のキーと、式プレフィックスによって返される完全な文字列を比較します。説明の例は、前の例を示しています。

文字列マップ内のキーとの文字列マッチングを有効にするために導入された 2 つの関数の詳細については、文字列マップ関数表 pdf を参照してください。

文字列マップの設定

最初に文字列マップを作成し、次にキーと値のペアをバインドします。文字列マップは、コマンドラインインターフェイス（CLI）または設定ユーティリティから作成できます。

コマンドラインインターフェイスを使用して文字列マップを設定するには

コマンドプロンプトで、次の操作を行います。

1. 文字列マップを作成します。

```
add policy stringmap <name> -comment <string>
```

1. キーと値のペアを文字列マップにバインドします。

```
bind policy stringmap <name> <key> <value> [-comment <string>]
```

例:

```
1 bind policy stringmap url_string_map1 ”/url_1.html” ”http://www.
     redirect_url_1.com/url_1.html”
2 <!--NeedCopy-->
```


Citrix ADC GUI を使用して文字列マップを構成するには

[**AppExpert**] > [文字列マップ] に移動し、[追加] をクリックして、関連する詳細を指定します。

例**:** リダイレクトアクションを含むレスポンダーポリシー

次のユースケースは、リダイレクトアクションを含むレスポンダーポリシーです。以下の例では、最初の 4 つのコマンドで文字列マップ url_string_map を作成し、前の例で使用した 3 つのキーと値のペアをバインドします。マップを作成し、キーと値のペアをバインドしたら、応答側アクション (act_url_redirects) を作成します。応答側アクション (act_url_redirects) は、文字列マップの対応する URL または www.default.com にクライアントをリダイレクトします。また、リクエストされた URL が url_string_map のいずれかのキーと一致しているかどうかをチェックし、設定されたアクションを実行するレスポンダポリシー（pol_url_redirects）も設定します。最後に、レスポンダーポリシーを、評価するクライアント要求を受信するコンテンツスイッチ仮想サーバーにバインドします。

```
add stringmap url_string_map

bind stringmap url_string_map /url_1.html http://www.redirect_url_1.com/
url_1.html

bind stringmap url_string_map /url_2.html http://www.redirect_url_2.com/
url_2.html

bind stringmap url_string_map /url_3.html http://www.redirect_url_1.com/
url_1.html

add responder action act_url_redirects redirect 'HTTP.REQ.URL.MAP_STRING(”
url_string_map”)ALT ”www.default.com”'-bypassSafetyCheck yes

add responder policy pol_url_redirects TRUE act_url_redirects

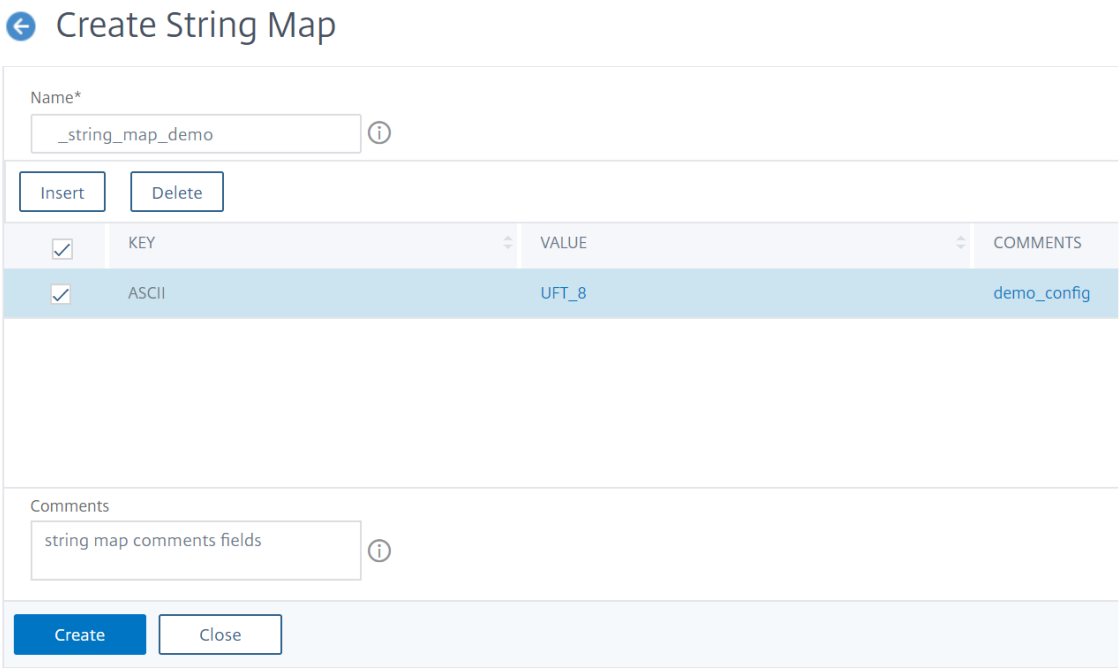
bind cs vserver csw_redirect -policyname pol_url_redirects -priority 1 -
type request
```

Citrix ADC GUI を使用して文字列マップを構成するには

文字列マップを設定するには、以下の手順に従ってください。

1. ナビゲーションウィンドウで、[**AppExpert**] を展開し、[文字列マップ] をクリックします。
2. 詳細ウィンドウで、[追加] をクリックします。
3. **[**文字列マップの作成] ページで、次のパラメータを設定します。
    - Name：文字列マップの名前。
    - キー値を設定します。文字列マップにバインドされた ASCII ベースのキー値エントリ
    - コメント。文字列マップにバインドされたキー値に関する簡単な説明。

4. [作成] して [閉じる] をクリックします。



URL セット

October 7, 2021

この機能を使用すると、100 万個の URL をブラックリストに登録できます。ここでは、次のトピックについて説明します。

- はじめに
- URL 評価のための高度なポリシー式の使用
- URL セットの構成
- URL パターンのセマンティクス
- ブラックリストに登録された URL カテゴリ

はじめに

October 7, 2021

制限された Web サイトへのアクセスを防止するために、Citrix ADC アプライアンスは特殊な URL マッチングアルゴリズムを使用します。このアルゴリズムは、最大 100 万 (1,000,000) のブラックリストに登録された URL のリストを含むことができる URL セットを使用します。各エントリには、URL カテゴリとカテゴリグループをインデック

ス付きパターンとして定義するメタデータを含めることができます。また、このアプライアンスは、インターネット執行機関（政府ウェブサイトを含む）やインターネット組織が管理する機密性の高い URL セットの URL を定期的にダウンロードすることもできます。URL セットが Web サイトからダウンロードされ、アプライアンスにインポートされると、アプライアンスは URL セットを（これらの機関の要求により）暗号化し、機密情報が保持され、エントリが改ざんされることはありません。

Citrix ADC アプライアンスは、高度なポリシーを使用して、着信 URL をブロック、許可、またはリダイレクトする必要があるかどうかを判断します。これらのポリシーは、高度な式を使用して、ブラックリストに登録されたエントリに対して着信 URL を評価します。エントリにはメタデータを含めることができます。メタデータを持たないエントリには、完全一致文字列に基づいて URL を評価する式を使用できます。他の URL では、完全に一致する文字列をチェックする式に加えて、URL のメタデータを評価する式を使用することもできます。

ISP/通信事業者向けの安全なインターネットアクセスポリシーの使用例

URL セットを使用すると、ISP（ISP）または電話会社のお客様は、次のような政府による安全なインターネットアクセスポリシーを適用できます。

1. 違法なインターネットサイト（児童虐待、薬物など）へのアクセスをブロックする
2. お子様の安全なブラウジング

Citrix ADC アプライアンスを使用すると、インターネット執行機関または独立したインターネット組織が管理する URL セットを定期的にダウンロードできます。アプライアンスは定期的にリストをダウンロードし、安全に更新します。リストは機密の URL セットとして保存されるため、改ざんや人間が読めることはありません。定期的にダウンロードされる URL セットは、URL 評価の目的でブラックリストに登録されたセットとして機能します。

プライベート URL が設定されていて、リストの内容が機密に保たれ、ネットワーク管理者がリスト内のブラックリスト URL を知らない場合。ポリシーが正しく構成され、正しいリストが参照されていることを確認するには、Canary URL を構成し、URL セットに追加する必要があります。管理者は、Canary URL を使用して、アプライアンスを通じて要求できます。プライベート URL セットを使用して、すべての URL 要求に対して検索されます。

URL 評価用の高度なポリシー式

October 7, 2021

次の表に、URL セット内のエントリを含む着信 URL を評価するために使用できる式を示します。

注 <URL expression>: HTTP.REQ.URL は、次のように使用されるように一般化されています

式	操作
<URL expression>.URLSET_MATCHES_ANY	URL が URL セット内のエントリと完全に一致する場合に TRUE と評価されます。

式	操作
<URL expression>GET_URLSET_METADATA (<URLSET>)	GET_URLSET_METADATA () 式は、URL が URL セット内の任意のパターンに完全に一致する場合に、関連するメタデータを返します。一致しなかった場合は、空の文字列が返されます。
<URL expression> .GET_URLSET_METADATA(<URLSET>).EQ(<METADATA>	一致するメタデータが<METADATA>と等しい場合は TRUE と評価されます。
<URL expression> .GET_URLSET_METADATA(<URLSET>).TYPECAST_LIST_T(‘,’).GET(0).EQ(<CATEGORY>)	一致するメタデータがカテゴリの先頭にある場合は TRUE と評価されます。このパターンは、メタデータ内の別々のフィールドをエンコードするために使用できますが、最初のフィールドのみと一致します。
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	ホストパラメータと URL パラメータを結合します。このパラメータは、照合のため に<URL expression>として使用できます。

URL セットの設定

October 7, 2021

Citrix ADC プラットフォームで URL セットを構成し、URL を制限するには、次のタスクを実行します。

1. URL セットをインポートします (ダウンロードして暗号化)。Citrix ADC アプライアンスで URL セットをインポートすると、次のことが可能になることができます。

- URL ファイルをダウンロードします。
- アプライアンスにファイルを追加するには。
- ファイルを暗号化する。
  URL セットをシステムに追加するまで、ユーザーには表示されません。

セットは次の方法でダウンロードできます。

- リモートサーバーから URL セットを 1 回ダウンロードして、次のように指定します。http://myserver.com/file_with_urlset.csv
- ADC 内の/var/tmp/パスの下にファイルを追加し、以下の例のようにコマンドを使用します。

```
1 > shell cat /var/tmp/test_urlset.csv
2 example.com
```



```
3 google.com
4 > import policy urlset top10
5 k -url local:test_urlset.csv -delimiter ”,” -rowSeparator ”n” -interval
      10 -privateSet -canaryUrl http://www.in.gr
6 Done
7
8 <!--NeedCopy-->
```

インポートされた URL セットは、データベース内の異なるカテゴリおよびカテゴリグループに分類されます。これは、URL セットファイルのメタデータにカテゴリが存在する場合にのみ有効です。

注: メタデータなしで URL パターンがある可能性があります。

ファイルをインポートしたら、ファイルのプロパティを更新、削除、または表示することができます。ファイルがアプライアンスにプッシュされた後、行を追加してエントリを変更できます。

インポートされたセットは、暗号化されたファイル形式で Citrix ADC ディレクトリに保存されます。インポートされたリストには、数百万の URL エントリが含まれています。次の’インポートされたリストには、最大 100 万の URL エントリを含めることができます。それ以外の場合、アプライアンスは値が制限を超えていることを示すエラーメッセージを返します。インポートされた URL セットにメタデータを含むブラックリストに登録されているエントリがある場合、インポート時にアプライアンスによってメタデータが検出されます。

URL セットをインポートしてアプライアンスに追加すると、着信 URL 評価時に正しい URL セットを識別するために、高度なポリシーで URL セットを使用できます。`HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URLSET_MATCHES_ANY(<URL set name>)`

1. Citrix ADC アプライアンスの URL セットの更新。ファイルをアプライアンスにプッシュしたら、この間隔でコマンドラインインターフェイスを使用して URL ファイルを手動で更新できます。
2. URL セットのエクスポート。URL セットのバックアップが必要な場合は、URL パターンのリストをエクスポートし、そのコピーを宛先 URL に保存できます。エクスポートする前に、URL セットがプライベートとしてマークされているかどうかを確認します。がプライベートとマークされている場合、URL セットはエクスポートできません。エクスポート機能は、プライベートセットでは機能しません。したがって、新しい URL セット`myurl`は、プライベートセットが定義されずにインポートされ、以下のようにローカルパス内の別のファイルにエクスポートされます。

```
1 > shell touch /var/tmp/test_urlset_export.csv
2 Done
3 > shell cat /var/tmp/test_urlset_export.csv
4 Done
5 > shell cat /var/tmp/test_urlset.csv
6 example.com
7 google.com
8 Done
```



```
> export urlset myurl -url local:test_urlset_export.csv

> import urlset myurl -url local:test_urlset.csv
Done
(a non-private urlset is imported)

<!--NeedCopy-->
```

1. URL セットの削除。ブラックリストに登録されたエントリの URL セットを削除する場合は、remove コマンドを使用して、Citrix ADC アプライアンスから URL セットを削除できます。
2. URL セットの表示。show コマンドを使用して、URL セットのプロパティを表示できます。

> 注**:** クエリ部分を含む URL は、インポート中に削除されます。

例**:**

```
show urlset
Name: top100 PatternCount: 100 Delimiter:  RowSeparator: Interval: 0
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、メタを含む **URL** セットをインポートする

コマンドプロンプトで、次のように入力します。

```
import urlset <name> [-overwrite] [ - delimiter <character>] [-
    rowSeparator <character>] [-url] <url> [-interval <seconds>] [-
    privateSet] [-canaryUrl <URL>]
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

区切り文字は、デフォルト値が 44 に設定された CSV ファイルレコードです。

RowSeparator は、デフォルト値は 10 に設定された CSV ファイルの行区切り文字です。

間隔は、URL セットの更新が行われる最も近い 15 分に四捨五入された時間間隔を秒単位で指定します。

CanaryUrl は、URL セットの内容が機密保持されるときのテストに使用される URL です。

例

```
import policy urlset -url local:test_urlset.csv -delimiter ”,”-rowSeparator
 ”n”-interval 10 -privateSet -canaryUrl http://www.in.gr
```


インポートされた **URL** セットに対して明示的なサブドメイン一致を実行する

インポートされた URL セットに対して明示的なサブドメイン一致を実行できるようになりました。新しいパラメーター「SubdomainExactMatch」が「インポートポリシー URLSet」コマンドに追加されます。パラメーターを有効にすると、URL フィルタリングアルゴリズムは明示的なサブドメイン一致を実行します。たとえば、着信 URL が「news.example.com」で、URL セット内のエントリが「example.com」の場合、アルゴリズムは URL と一致しません。

コマンドプロンプトで入力します。

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-rowSeparator
 <character>] -url [-interval <secs>] [-privateSet][-subdomainExactMatch]
[-canaryUrl <URL>]
```

例:

```
import policy urlset forth_urlset -url local:test_urlset.csv -interval 3600
 -subdomainExactMatch
```

コマンドラインインターフェイスを使用して **URL** セットを表示するには

コマンドプロンプトで入力します。

```
show urlset <name>
```

例:

コマンドプロンプトで入力します。

```
1          URLset      Count
2          ------      -----
3  1)      top1k       100
4   Done
5
6  > show urlset top1k
7          Count       Delimiter   Interval    RowSeparator
8          -----       ---------   --------    ------------
9          100               ,         0          0x0a
10  Done
11 >
12
13 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用してインポートされた **URL** セットを表示するには

コマンドプロンプトで入力します。


```
show urlset -imported
```

例:

コマンドプロンプトで入力します。

```
1          URLset
2          ------
3  1)      top1k
4   Done
5  <!--NeedCopy-->
```

URL セットを表示するには **<urlset_name>** コマンドラインインターフェイスを使用する

コマンドプロンプトで入力します。

```
show urlset <name>
```

コマンドラインインターフェイスを使用して **URL** セットをエクスポートするには

コマンドプロンプトで入力します。

```
export urlset <name> <url>
```

コマンドラインインターフェイスを使用して **URL** セットを追加するには

コマンドプロンプトで入力します。

```
add urlset <urlset_name>
```

コマンドラインインターフェイスを使用して **URL** セットを更新するには

コマンドプロンプトで入力します。

```
update urlset <name>
```

コマンドラインインターフェイスを使用して **URL set** コマンドを削除するには

コマンドプロンプトで入力します。

```
remove urlset <name>
```

例:

> 注:
>
> URLSet をインポートまたはエクスポートする前に、test_urlset_export.csvおよびtest_urlset.csvファイルが作成され、/var/tmpディレクトリの下で使用可能であることを確認する必要があります。

```
import policy urlset -url local:test_urlset.csv -delimiter ”,” -rowSeparator ”n” -interval 10 -privateSet -overwrite  - canaryUrl http://www.in.gr

add policy urlset top10k

update policy urlset top10k

sh policy urlset

sh policy urlset top10k

export policy urlset urlset1 -url local:test_urlset_export.csv

import policy urlset top10k -url local:test_urlset.csv  - privateSet

add policy urlset top10k

update policy urlset top10k

show policy urlset top10k
<!--NeedCopy-->
```

インポートされた **URL** セットを表示する

追加された URL セットに加えて、読み込んだ URL セットも表示できるようになりました。これを行うには、「show url set」コマンドに新しいパラメータ「imported」が追加されます。このオプションを有効にすると、アプライアンスはインポートされたすべての URL セットを表示し、インポートされた URL セットが追加された URL セットと区別されます。

コマンドプロンプトで入力します。

```
show policy urlset [<name>] [-imported]
```

例:

```
show policy urlset -imported
```


GUI を使用して URL セットをインポートするには

AppExpert > **URL** セットに移動し、インポートをクリックして URL セットをダウンロードします。

GUI を使用して URL セットを追加するには

AppExpert > **URL** セットに移動し、[追加] をクリックして、ダウンロードした URL セットの URL セットファイルを作成します。

GUI を使用して URL セットを編集するには

AppExpert > **URL** セットに移動し、URL セットを選択し、編集をクリックして変更します。

GUI を使用して URL セットを更新するには

AppExpert > **URL** セットに移動し、 URL セットを選択し、**URL** セットの更新をクリックして、ファイルに加えられた最新の変更で URL セットを更新します。

GUI を使用して URL セットをエクスポートするには

AppExpert > URL セットに移動し、URL セットを選択し、[**URL** セットのエクスポート] をクリックして、セット内の URL パターンを宛先 URL にエクスポートし、その場所に保存します。

URL パターンのセマンティクス

October 7, 2021

次の表に、フィルタリングするページのリストを指定するために使用する URL パターンを示します。たとえば、URL パターン`http://www.example.com/bar`は 1 つのページ`http://www.example.com/bar`に一致します。URL が www.example.com/bar で始まるすべてのページをカバーするには、末尾に「*」を明示的に追加する必要があります。

詳細については、「URL パターンメタデータマッピングテーブル 」を参照してください。

URL のカテゴリ

October 7, 2021

以下は、ブラックリストに登録されたカテゴリのリストです。

S.NO	ブラックリストに載るカテゴリー
1	違法行為
2	違法薬物
3	薬物
4	マリファナ
5	テロリズム/過激派
6	武器
7	誹謗/中傷
8	暴力/自殺
9	主義主張全般
10	アダルト/ポルノ
11	ヌード
12	性的サービス
13	アダルト検索/リンク
14	ハッキング/クラッキング
15	マルウェア
16	リモートプロキシ
17	検索エンジンのキャッシュ
18	翻訳
19	出会い系
20	結婚式/結婚生活
21	相場
22	オンライン取引
23	保険
24	金融商品
25	ギャンブル全般
26	宝くじ
27	オンラインゲーム
28	ゲーム
29	オークション

S.NO	ブラックリストに載るカテゴリー
30	ショッピング/小売
31	不動産
32	IT オンラインショッピング
33	Web 上でのチャット
34	インスタントメッセージ
35	Web ベースのメール
36	メールサービス加入
37	掲示板
38	IT 掲示板
39	個人の Web ページ/ブログ
40	ダウンロード
41	プログラムのダウンロード
42	ストレージサービス
43	ストリーミングメディア
44	雇用
45	キャリアアップ
46	副業
47	猟奇描写
48	スペシャルイベント
49	人気トピック
50	アダルト雑誌/ニュース
51	喫煙
52	飲酒
53	アルコール製品
54	フェチ
55	性的表現（テキスト）
56	コスプレ/仮装
57	オカルト
58	家庭および家族

S.NO	ブラックリストに載るカテゴリー
59	プロスポーツ
60	スポーツ全般
61	ライフイベント
62	旅行/観光
63	公的機関（観光）
64	公共交通
65	宿泊施設
66	ミュージック
67	ホロスコープ/占星術/運勢判断
68	芸能人/著名人
69	飲食/グルメ
70	娯楽/施設/アクティビティ
71	既成宗教
72	宗教
73	政治
74	広告/バナー
75	懸賞/プレゼント
76	スパム
77	ニュース
78	自動車
79	ビジネスおよび商業
80	コンピューターおよびインターネット
81	教育
82	自治体
83	状況
84	インターネット電話
85	軍事
86	ピアツーピア/トレント
87	レジャーおよび趣味

S.NO	ブラックリストに載るカテゴリー
88	リファレンス
89	検索エンジンおよびポータル
90	性教育
91	SMS および携帯電話サービス
92	モバイルアプリおよびパブリッシャー
93	スパイウェア
94	コンテンツ配信ネットワークおよびインフラストラクチャ
95	子供向けサイト
96	水着および下着
97	芸術および文化イベント
98	ホスティングサイト
99	慈善事業および非営利団体
100	写真検索および写真共有サイト
101	着信音
102	ファッションおよび美容
103	モバイルアプリストア
104	パークドメイン
105	絵文字
106	移動体通信事業者
107	ボットネット
108	感染サイト
109	フィッシングサイト
110	キーロガー
111	モバイルマルウェア
112	コンテンツなし
113	農業
114	アーキテクチャ
115	各種協会/業界団体/労働組合

S.NO	ブラックリストに載るカテゴリー
116	書籍・電子書籍
117	ボットネットから命令元への通信
118	DDNS
119	サポートされていない URL
120	法律
121	地域コミュニティ
122	その他
123	オンラインマガジン
124	ペット/獣医
125	著作権や商標権の侵害
126	プライベート IP アドレス
127	リサイクル/環境
128	科学
129	社会および文化
130	輸送サービスと貨物
131	写真および映画
132	博物館および歴史
133	E ラーニング
134	ソーシャルネットワーク全般
135	Facebook
136	Facebook：投稿
137	Facebook：コメント
138	Facebook：友達
139	Facebook：写真のアップロード
135	Facebook：イベント
141	Facebook：アプリ
142	Facebook：チャット
143	Facebook：質問
144	Facebook：動画のアップロード

S.NO	ブラックリストに載るカテゴリー
145	Facebook：グループ
146	Facebook：ゲーム
147	LinkedIn
148	LinkedIn：アップデート
110	LinkedIn：メール
150	LinkedIn：つながり
151	LinkedIn：求人情報
152	Twitter
153	Twitter：投稿
154	Twitter：メール
155	Twitter：フォロー
156	YouTube
157	YouTube：コメント
158	YouTube：動画のアップロード
159	YouTube：共有
160	Instagram
161	Instagram：アップロード
162	Instagram：コメント
163	Instagram：ダイレクトメッセージ
164	Tumblr
165	Tumblr：掲示板
166	Tumblr：コメント
167	Tumblr：写真または動画のアップロード
168	Google+
169	Google+：投稿
170	Google+：コメント
171	Google+：写真のアップロード
172	Google+：動画のアップロード
173	Google+：動画チャット

S.NO	ブラックリストに載るカテゴリー
174	Pinterest
175	Pinterest：ピン
176	Vine：アップロード
252	Vine：コメント
178	Vine：メッセージ
179	Ask.fm
180	Ask.fm：質問
181	Ask.fm：回答
182	Yik Yak
183	Yik Yak：投稿
184	Yik Yak：コメント
185	WordPress
186	WordPress：投稿
187	WordPress：アップロード

AppFlow

October 7, 2021

Citrix ADC アプライアンスは、データセンター内のすべてのアプリケーショントラフィックを一元的に制御します。これは、アプリケーションパフォーマンスの監視、分析、およびビジネスインテリジェンスアプリケーションにとって有効なフローとユーザーセッションレベルの情報を収集します。また、Web ページのパフォーマンスデータとデータベース情報も収集します。AppFlow は、インターネットプロトコルフロー情報 eXport (IPFIX) 形式を使用して情報を送信します。これは、RFC 5101 で定義されているオープンなインターネット技術標準化委員会 (IETF) 標準です。IPFIX（Cisco 社製 NetFlow の標準化バージョン）は、ネットワークフロー情報を監視するために幅広く使用されています。AppFlow は、アプリケーションレベルの情報、Web ページのパフォーマンスデータ、およびデータベース情報を表す新しい情報要素を定義します。

トランスポートプロトコルとして UDP を使用して、AppFlow は フローレコードと呼ばれる収集されたデータを 1 つまたは複数の IPv4 コレクターに送信します。コレクターはフローレコードを集約し、リアルタイムレポートまたは履歴レポートを生成します。

AppFlow は、HTTP、SSL、TCP、SSL_TCP フロー、HDX Insight フローのトランザクションレベルでの可視性

を提供します。監視対象のフロータイプのサンプリングとフィルタリングを行うことが可能です。

> 注:
>
> HDX Insight の詳細については、HDX Insightを参照してください。

AppFlow は、アクションとポリシーを使用して、選択したフローのレコードを特定のコレクターセットに送信します。AppFlow アクションは、どのコレクターのセットが AppFlow レコードを受信するかを指定します。高度な式に基づくポリシーは、関連する AppFlow アクションで指定されたコレクターにフローレコードが送信されるフローを選択するように構成できます。

フローの種類を制限するには、仮想サーバーで AppFlow を有効にします。AppFlow では、仮想サーバーの統計情報も提供しています。

また、AppFlow を特定のサービス向けに有効化してアプリケーションサーバーを表現し、そのアプリケーションサーバーへのトラフィックを監視することもできます。

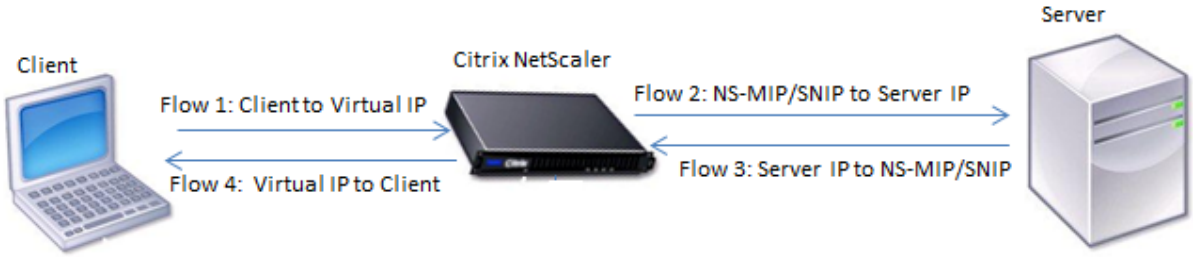
注: この機能は、Citrix ADC nCore ビルドでのみサポートされています。

AppFlow のしくみ

最も一般的な展開シナリオでは、受信トラフィックは Citrix ADC アプライアンスの仮想 IP アドレス（VIP）に流れ、サーバーに負荷分散されます。送信トラフィックは、サーバーから Citrix ADC 上のマッピングされた IP アドレスまたはサブネット IP アドレスに送信され、VIP からクライアントに送信されます。フローは、送信元 IP、送信元ポート、destIP、destPort、およびプロトコルの 5 つのタプルによって識別される IP パケットの単方向集合です。

次の図は、AppFlow 機能の動作を示しています。

図 1: Citrix ADC のフロー・シーケンス



図に示すように、トランザクションの各レッグのネットワークフロー ID は、トラフィックの方向によって異なります。

フローレコードを形成するフローは、次のとおりです。

フロー 1:<Client-IP, Client-Port, VIP-IP, VIP-port, Protocol>

フロー 2:<NS-MIP/SNIP, NS-port, Server-IP, Server-Port, Protocol>

フロー 3:<Server-IP, Server-Port, NS-MIP/SNIP, NS-Port, Protocol>

フロー 4:<VIP-IP, VIP-port, Client-IP, Client-Port, Protocol>

コレクタがトランザクション内の 4 つのフローすべてをリンクできるようにするため、AppFlow は各フローにカスタム transactionID 要素を追加します。HTTP などのアプリケーションレベルのコンテンツスイッチングの場合、単一のクライアント TCP 接続を、要求ごとに異なるバックエンド TCP 接続に負荷分散することができます。AppFlow は、各トランザクションのレコードのセットを提供します。

フローレコード

AppFlow レコードには、フローの開始と終了のタイムスタンプ、パケットカウント、バイトカウントなど、標準の NetFlow または IPFIX 情報が含まれます。AppFlow レコードには、アプリケーションレベルの情報（HTTP URL、HTTP 要求メソッドと応答ステータスコード、サーバーの応答時間、待機時間など）も含まれます。Web ページのパフォーマンスデータ（ページの読み込み時間、ページのレンダリング時間、ページで費やされた時間など）。およびデータベース情報（データベースプロトコル、データベース応答ステータス、データベース応答サイズなど）。IPFIX フローレコードは、フローレコードを送信する前に送信する必要があるテンプレートに基づいています。

テンプレート

AppFlow では、フローのタイプごとにテンプレートのセットを定義します。各テンプレートには、標準の情報要素（IE）とエンタープライズ固有の情報要素（EIE）のセットが含まれています。IPFIX テンプレートは、フローレコード内の情報要素（Internet Explorer）の順序とサイズを定義します。テンプレートは、RFC 5101 で説明されているように、定期的にコレクタに送信されます。

テンプレートには、次の EIE を含めることができます。

- transactionID

  アプリケーションレベルのトランザクションを識別する符号なし 32 ビットの数値。HTTP の場合、これは要求と応答のペアに対応します。この要求と応答のペアに対応するすべてのフローレコードは、同じトランザクション ID を持ちます。最も一般的なケースでは、このトランザクションに対応するuniflowレコードが 4 つあります。Citrix ADC がそれ自体で（統合キャッシュまたはセキュリティポリシーによって提供される）応答を生成する場合、このトランザクションのフローレコードは 2 つしかない可能性があります。

- connectionID

  レイヤ 4 接続（TCP または UDP）を識別する符号なし 32 ビットの数値。Citrix ADC フローは双方向であり、フローの方向ごとに 2 つの個別のフローレコードがあります。この情報要素は、2 つのフローをリンクするために使用できます。

  Citrix ADC の場合、connectionID は、接続の進行状況を追跡するための接続データ構造の識別子です。たとえば、HTTP トランザクションでは、特定の connectionID に、その接続で行われた複数の要求に対応する複数の transactionID 要素が含まれる場合があります。

- tcpRTT

TCP 接続で測定されたラウンドトリップ時間（ミリ秒単位）。これは、ネットワーク上のクライアントまたはサーバーの遅延を判断するためのメトリックとして使用できます。

- httpRequestMethod

  トランザクションで使用される HTTP メソッドを示す 8 ビットの数値。番号から方式へのマッピングを含むオプションテンプレートが、テンプレートとともに送信されます。

- httpRequestSize

  要求のペイロードサイズを示す符号なし 32 ビットの数値。

- httpRequestURL

  クライアントによって要求された HTTP URL。

- httpUserAgent

  Web サーバーへの受信要求のソース。

- httpResponseStatus

  応答ステータスコードを示す符号なし 32 ビットの数値。

- httpResponseSize

  応答サイズを示す符号なし 32 ビットの数値。

- httpResponseTimeToFirstByte

  応答の最初のバイトを受信するのにかかった時間を示す符号なし 32 ビットの数値。

- httpResponseTimeToLastByte

  応答の最後のバイトを受信するのにかかった時間を示す符号なし 32 ビットの数値。

- flowFlags

  異なるフロー条件を示すために使用される符号なし 64 ビットフラグ。

ウェブページのパフォーマンスデータの **EIE**

- clientInteractionStartTime

  ブラウザが応答の最初のバイトを受信して、画像、スクリプト、スタイルシートなどのページのオブジェクトをロードする時刻。

- clientInteractionEndTime

  画像、スクリプト、スタイルシートなど、ページのすべてのオブジェクトをロードするためにブラウザが応答の最後のバイトを受信した時刻。

- clientRenderStartTime

  ブラウザがページのレンダリングを開始する時刻。

- clientRenderEndTime

  ブラウザが埋め込みオブジェクトを含むページ全体のレンダリングを終了した時刻。

データベース情報の **EIE**

- dbProtocolName

  データベースプロトコルを示す符号なしの 8 ビットの数値。有効な値は、MS SQL の場合は 1、MySQL の場合は 2 です。

- dbReqType

  トランザクションで使用されるデータベース要求メソッドを示す符号なしの 8 ビットの数値。MS SQL の場合、有効な値は 1 がクエリの場合、2 がトランザクションの場合、3 が RPC の場合、です。MySQL の有効な値については、MySQL のドキュメントを参照してください。

- dbReqString

  ヘッダーのないデータベース要求文字列を示します。

- dbRespStatus

  Web サーバから受信したデータベース応答のステータスを示す符号なし 64 ビットの数値。

- dbRespLength

  応答サイズを示す符号なし 64 ビットの数値。

- dbRespStatString

  Web サーバから受信した応答ステータス文字列。

AppFlow 機能の構成

October 7, 2021

AppFlow は、他のほとんどのポリシーベースの機能と同じ方法で構成します。まず、AppFlow 機能を有効にします。次に、フローレコードの送信先となるコレクタを指定します。その後、アクションを定義します。アクションは、設定されたコレクタのセットです。次に、1 つ以上のポリシーを設定し、各ポリシーにアクションを関連付けます。このポリシーは、関連するアクションに送信されるフローレコードの選択要求を Citrix ADC アプライアンスに指示します。最後に、各ポリシーをグローバルに、または特定の仮想サーバーにバインドして、ポリシーを有効にします。

さらに AppFlow パラメータを設定して、テンプレートの更新間隔を指定し、httpURL、httpCookie、および httpReferer 情報のエクスポートを有効にすることもできます。各コレクターで、エクスポータのアドレスとして Citrix ADC IP アドレスを指定する必要があります。

> 注
>
> Citrix ADC をコレクターのエクスポーターとして構成する方法については、特定のコレクターのドキュメントを参照してください。

構成ユーティリティは、ユーザーがポリシーとアクションを定義するのに役立つツールを提供します。これは、Citrix ADC アプライアンスが特定のフローのレコードをコレクターのセットにエクスポートする方法を正確に決定します（アクション）。コマンドラインインターフェイスは、コマンドラインを好む経験豊富なユーザーに対応する CLI ベースのコマンドのセットを提供します。

AppFlow の有効化

AppFlow 機能を使用するには、まず AppFlow 機能を有効にする必要があります。

> 注
>
> AppFlow は、nCore Citrix ADC アプライアンスでのみ有効にできます。

コマンドラインインターフェイスを使用して **AppFlow** 機能を有効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
enable ns feature AppFlow
<!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** 機能を有効にするには

[システム**] > [**設定**]** に移動し、**[**詳細機能の設定**]** をクリックして、**[AppFlow]** オプションを選択します。

コレクタの指定

コレクタは、Citrix ADC アプライアンスによって生成された AppFlow レコードを受信します。AppFlow レコードを送信するには、少なくとも 1 つのコレクタを指定する必要があります。デフォルトでは、コレクタは UDP ポート 4739 で IPFIX メッセージをリッスンします。コレクタを設定するときに、デフォルトポートを変更できます。同様に、デフォルトでは、NSIP が AppFlow トラフィックの送信元 IP として使用されます。コレクターを構成するときに、このデフォルトの送信元 IP を SNIP アドレスに変更できます。未使用のコレクターを削除することもできます。

コマンドラインインターフェイスを使用してコレクタを指定するには

> 重要
>
> Citrix ADC リリース 12.1 ビルド 55.13 以降、使用するコレクターのタイプを指定できます。`add appflow collector`コマンドに新しいパラメータ「Transport」が導入されました。デフォルトでは、コレクターはIPFIX メッセージをリッスンします。「Transport」パラメータを使用して、コレクタのタイプを`logstream`または`ipfix`またはリセットのいずれかに変更できます。構成の詳細については、例を参照してください。

コマンドプロンプトで次のコマンドを入力して、コレクタを追加し、構成を確認します。

```
1 -  add appflow collector <name> -IPAddress <ipaddress> -port <
     port_number> -netprofile <netprofile_name> -Transport <Transport>
2
3 -  show appflow collector <name>
4 <!--NeedCopy-->
```

例

```
1 add appflow collector col1 -IPaddress 10.102.29.251 -port 8000 -
     netprofile n2 -Transport ipfix
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して複数のコレクタを指定するには

コマンドプロンプトで次のコマンドを入力して、同じデータを追加して複数のコレクターに送信します。

```
1  add appflow collector <collector1> -IPAddress <IP>
2
3  add appflow collector <collector2> -IPAddress <IP>
4
5  add appflow action <action> -collectors <collector1> <collector2>
6
7  add appflow policy <policy> true <action>
8
9  bind lbvserver <lbvserver> -policy <policy> -priority <priority>
10 <!--NeedCopy-->
```


構成ユーティリティを使用して **1** つ以上のコレクタを指定するには

「システム」**>**「**AppFlow**」**>**「コレクター」に移動し、AppFlow コレクターを作成します。

AppFlow アクションの構成

AppFlow アクションはセットコレクターであり、関連付けられた AppFlow ポリシーが一致した場合にフローレコードが送信されます。

コマンドラインインターフェイスを使用して **AppFlow** アクションを構成するには

コマンドプロンプトで、次のコマンドを入力して AppFlow アクションを構成し、構成を確認します。

```
add appflow action <name> --collectors <string> ... [-
    clientSideMeasurements (Enabled|Disabled) ] [-comment <string>]

show appflow action
<!--NeedCopy-->
```

例

```
add appflow action apfl-act-collector-1-and-3 -collectors collector-1
    collecter-3
<!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** アクションを構成するには

「システム」**>**「**AppFlow**」**>**「アクション」に移動し、AppFlow アクションを作成します。

AppFlow ポリシーの構成

AppFlow アクションを構成したら、次に AppFlow ポリシーを構成する必要があります。AppFlow ポリシーは、1 つ以上の式で構成されるルールに基づいています。

> 注
>
> AppFlow ポリシーを作成および管理するために、構成ユーティリティは、コマンドラインインターフェイスでは利用できない支援を提供します。

コマンドラインインターフェイスを使用して **AppFlow** ポリシーを構成するには

コマンドプロンプトで次のコマンドを入力して、AppFlow ポリシーを追加し、構成を確認します。

```
add appflow policy <name> <rule> <action>

show appflow policy <name>
<!--NeedCopy-->
```

例

```
add appflow policy apfl-pol-tcp-dsprt client.TCP.DSTPORT.EQ(22) apfl-
    act-collector-1-and-3
<!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** ポリシーを構成するには

「システム」**>**「**AppFlow**」**>**「ポリシー」に移動し、AppFlow ポリシーを作成します。

[式の追加**]** ダイアログボックスを使用して式を追加するには

1. [式の追加] ダイアログボックスの最初のリストボックスで、式の最初の用語を選択します。

   - HTTP
     HTTP プロトコル。HTTP プロトコルに関連するリクエストのいくつかの側面を調べる場合は、オプションを選択します。
   - SSL

     保護されたウェブサイト。リクエストの受信者に関連するリクエストのいくつかの側面を調べる場合は、オプションを選択します。
   - クライアント

     要求を送信したコンピュータ。リクエストの送信者のある側面を調べたい場合は、オプションを選択してください。
     選択すると、右端のリストボックスには、式の次の部分に適した用語が一覧表示されます。

2. 2 番目のリストボックスで、式の 2 番目の項を選択します。選択内容は、前のステップで行った選択内容によって異なり、コンテキストに適切です。2 番目の選択を行うと、[式を構築] ウィンドウの下にある [ヘルプ] ウィンドウ (空白) に、選択した用語の目的と使用方法を示すヘルプが表示されます。

3. 式が終了するまで、前のリストボックスの右側に表示されるリストボックスから用語を選択するか、値の入力を求めるテキストボックスに文字列または数値を入力します。

AppFlow ポリシーのバインド

ポリシーを有効にするには、ポリシーをグローバルにバインドして、Citrix ADC を通過するすべてのトラフィックに適用するか、特定の仮想サーバーに適用して、その仮想サーバーに関連するトラフィックにのみポリシーが適用されるようにする必要があります。

ポリシーをバインドするときは、ポリシーに優先度を割り当てます。優先順位によって、定義したポリシーが評価される順序が決まります。プライオリティは任意の正の整数に設定できます。

Citrix ADC オペレーティングシステムでは、ポリシーの優先順位は逆の順序で機能します。値が大きいほど、優先順位は低くなります。たとえば、優先度が 10、100、および 1000 の 3 つのポリシーがある場合、優先度 10 が割り当てられたポリシーが最初に実行されます。その後、ポリシーは 100 の優先度で割り当てられ、最後にポリシーは 1000 の順序を割り当てました。

他のポリシーを任意の順序で追加する余地を十分に残して、必要な順序で評価するように設定することができます。グローバルにバインドするときに、各ポリシー間に 50 または 100 の間隔で優先順位を設定することで実現できます。その後、既存のポリシーの優先度を変更せずに、いつでもポリシーを追加できます。

コマンドラインインターフェイスを使用して **AppFlow** ポリシーをグローバルにバインドするには

コマンドプロンプトで次のコマンドを入力して、AppFlow ポリシーをグローバルにバインドし、構成を確認します。

```
bind appflow global <policyName> <priority> [<gotoPriorityExpression [-
    type <type>] [-invoke (<labelType> <labelName>)]

show appflow global
<!--NeedCopy-->
```

例

```
bind appflow global af_policy_lb1_10.102.71.190 1 NEXT -type
    REQ_OVERRIDE -invoke vserver google
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **AppFlow** ポリシーを特定の仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、AppFlow ポリシーを特定の仮想サーバーにバインドし、構成を確認します。

```
bind lb vserver <name> -policyname <policy_name> -priority <priority>
<!--NeedCopy-->
```

例

```
bind lb vserver google -policyname af_policy_google_10.102.19.179 -
    priority 251
<!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** ポリシーをグローバルにバインドするには

システムに移動 > **AppFlow** で、[**AppFlow** ポリシーマネージャー] をクリックし、関連するバインドポイント（デフォルトのグローバル）と接続タイプを選択して、AppFlow ポリシーをバインドします。

構成ユーティリティを使用して **AppFlow** ポリシーを特定の仮想サーバーにバインドするには

「トラフィック管理」**>**「負荷分散」**>**「仮想サーバー」に移動し、仮想サーバーを選択して「ポリシー」をクリックし、AppFlow ポリシーをバインドします。

仮想サーバーの **AppFlow** の有効化

特定の仮想サーバーを通過するトラフィックのみを監視する場合は、それらの仮想サーバー専用の AppFlow を有効にします。AppFlow を有効にして、負荷分散、コンテンツの切り替え、キャッシュリダイレクト、SSL VPN、GSLB、認証仮想サーバーを有効にすることができます。

コマンドラインインターフェイスを使用して仮想サーバーに対して **AppFlow** を有効にするには

コマンドプロンプトで入力します。

```
set cs vserver <name> <protocol> <IPAddress> <port> -appflowLog ENABLED
<!--NeedCopy-->
```


例

```
set cs vserver Vserver-CS-1 HTTP 10.102.29.161 80 -appflowLog ENABLED
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーで **AppFlow** を有効にするには

「トラフィック管理」**>**「コンテンツスイッチング」**>**「仮想サーバー」に移動し、仮想サーバーを選択し、AppFlow ロギングオプションを有効にします。

サービスの **AppFlow** の有効化

負荷分散仮想サーバーにバインドするサービスに対して AppFlow を有効にできます。

コマンドラインインターフェイスを使用してサービスに対して **AppFlow** を有効にするには

コマンドプロンプトで入力します。

```
set service <name> -appflowLog ENABLED
<!--NeedCopy-->
```

例

```
set service ser -appflowLog ENABLED
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスの **AppFlow** を有効にするには

「トラフィック管理」**>**「負荷分散」**>**「サービス」に移動し、サービスを選択し、AppFlow ロギングオプションを有効にします。

AppFlow パラメータの設定

AppFlow パラメータを設定して、コレクタへのデータのエクスポートをカスタマイズできます。

コマンドラインインターフェイスを使用して **AppFlow** パラメータを設定するには

> 重要
>
> - Citrix ADC リリース 12.1 ビルド 55.13 から、SNIP の代わりに NSIP を使用して`Logstream`レコードを送信できます。`set appflow param`コマンドに新しいパラメータ「LogStreamOvernSip」が導入されました。デフォルトでは、「logstreamOverNSIP」パラメーターは「無効」になっているため、「有効」にする必要があります。構成の詳細については、例を参照してください。
> - Citrix ADC リリース 13.0 ビルド 58.x リリース以降、AppFlow 機能で WebSaaS アプリケーションオプションを有効にできます。Citrix Gateway サービスから Web または SaaS アプリケーションのデータ使用量を受信できるようにすることができます。構成の詳細については、例を参照してください。

コマンドプロンプトで次のコマンドを入力して、AppFlow パラメータを設定し、設定を確認します。

```
1 -  set appflow param [-templateRefresh <secs>] [-appnameRefresh <secs>]
      [-flowRecordInterval <secs>] [-udpPmtu <positive_integer>] [-
     httpUrl ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-httpCookie ( \*\*
     ENABLED\*\* | \*\*DISABLED\*\* )] [-httpReferer ( \*\*ENABLED\*\* |
     \*\*DISABLED\*\* )] [-httpMethod ( \*\*ENABLED\*\* | \*\*DISABLED
     \*\* )] [-httpHost ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
     httpUserAgent ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
     httpXForwardedFor ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )][-
     clientTrafficOnly ( \*\*YES\*\* | \*\*NO\*\*)] [-
     webSaaSAppUsageReporting ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
     logstreamOverNSIP ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )]
2
3 -  show appflow Param
4 <!--NeedCopy-->
```

例

```
1 set appflow Param -templateRefresh 240 -udpPmtu 128 -httpUrl enabled -
     webSaaSAppUsageReporting ENABLED -logstreamOverNSIP ENABLED
2 <!--NeedCopy-->
```

構成ユーティリティを使用して **AppFlow** パラメーターを設定するには

「システム」**>**「**AppFlow**」に移動し、「**AppFlow** 設定の変更」をクリックして、関連する AppFlow パラメータを指定します。

サブスクライバー **ID** の難読化のサポート

Citrix ADC リリース 13.0 ビルド 35.xx 以降、AppFlow 構成が拡張され、レイヤー 4 またはレイヤー 7 の AppFlow レコードで MSISDN を難読化するための「subscriberIdObfuscation」アルゴリズムがサポートされるようになりました。ただし、アルゴリズムを MD5 または SHA256 として構成する前に、まず AppFlow パラメーターとして有効にする必要があります。このパラメータは、デフォルトでは無効になっています。

CLI を使用してサブスクライバー **ID** 難読化アルゴリズムを構成するには

コマンドプロンプトで入力します。

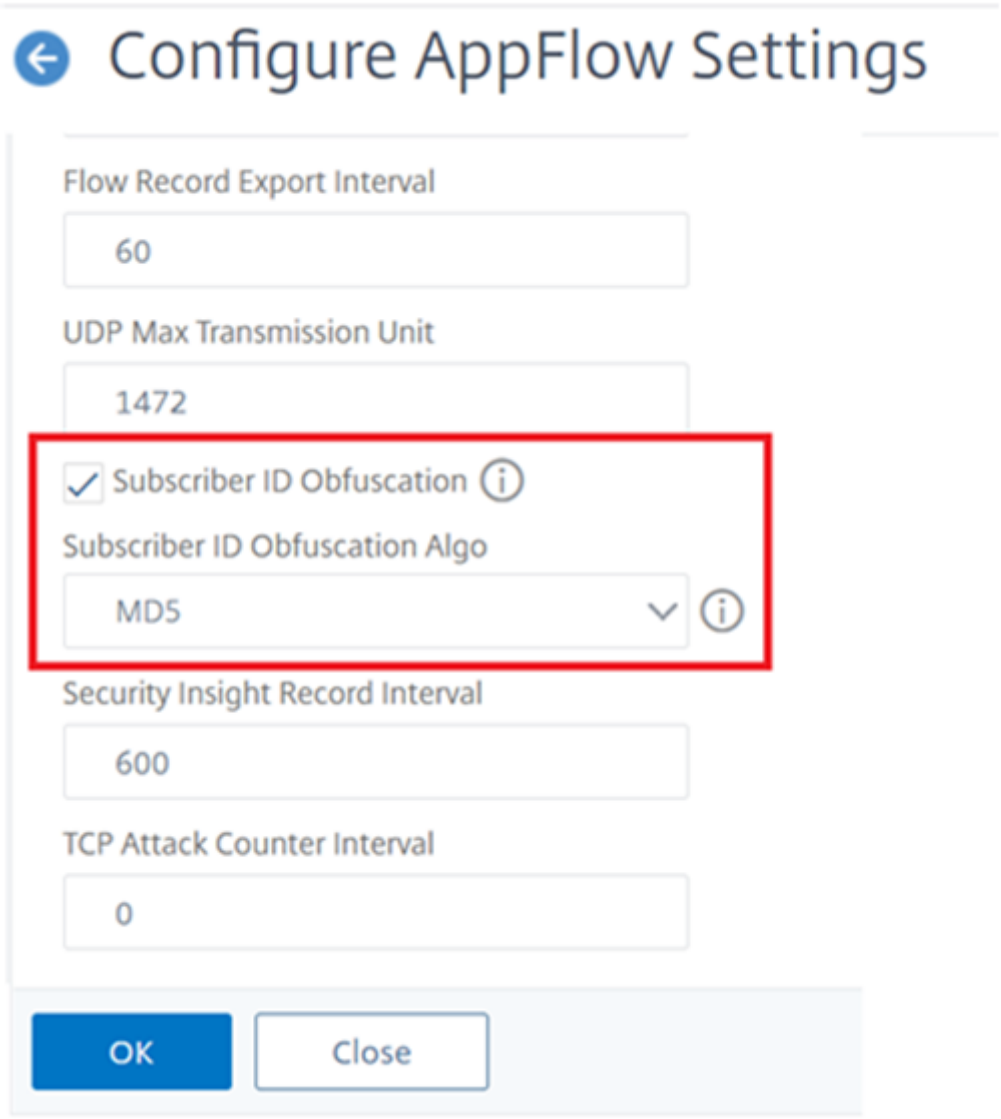
```
set appflow param [-subscriberIdObfuscation ( ENABLED | DISABLED )  [-
    subscriberIdObfuscationAlgo ( MD5 | SHA256 )]]
<!--NeedCopy-->
```

例

```
set appflow param  -subscriberIdObfuscation ENABLED  -
    subscriberIdObfuscationAlgo SHA256
<!--NeedCopy-->
```

GUI を使用してサブスクライバー **ID** 難読化アルゴリズムを構成するには

1. [**System**] **>** [**AppFlow**] の順に選択します。
2. AppFlow の詳細ペインで、[設定**]** の下の [AppFlow 設定の変更] をクリックします。
3. [AppFlow 設定の構成] ページで、次のパラメーターを設定します。
   - サブスクライバー **ID** の難読化。難読化 MSISDN のオプションを有効にします L4/L7 AppFlow レコード。
   - サブスクライバー **ID** 難読化アルゴリズム。アルゴリズムタイプを MD5 または SHA256 として選択します。
4. [**OK]** をクリックして [閉じる] をクリックします。



例:**DataStream** 用の **AppFlow** の設定

次の例は、コマンドラインインターフェイスを使用して AppFlow を DataStream 用に構成する手順を示しています。

```
enable feature appflow

add db user sa password freebsd

add lbvserver lb0 MSSQL 10.102.147.97 1433 -appflowLog ENABLED

add service sv0 10.103.24.132 MSSQL 1433 -appflowLog ENABLED
```



```
8
9  bind lbvserver lb0 sv0
10
11 add appflow collector col0 -IPAddress 10.102.147.90
12
13 add appflow action act0 -collectors col0
14
15 add appflow policy pol0 ”mssql.req.query.text.contains(”select”)” act0
16
17 bind lbvserver lb0 -policyName pol0 -priority 10
18 <!--NeedCopy-->
```

Citrix ADC アプライアンスがデータベース要求を受信すると、アプライアンスは構成済みのポリシーに対して要求を評価します。一致が見つかると、ポリシーで構成された AppFlow コレクターに詳細が送信されます。

Web ページのパフォーマンスデータを AppFlow コレクタにエクスポートする

October 7, 2021

EdgeSight Monitoring アプリケーションは、Citrix ADC 環境で提供されるさまざまな Web アプリケーションのパフォーマンスを監視できる Web ページ監視データを提供します。これで、このデータを AppFlow コレクタにエクスポートして、ウェブページアプリケーションの詳細な分析を取得できます。IPFIX 標準に基づく AppFlow は、EdgeSight の監視だけよりもウェブアプリケーションのパフォーマンスに関するより具体的な情報を提供します。

EdgeSight Monitoring データを AppFlow コレクターにエクスポートするように、負荷分散とコンテンツスイッチング仮想サーバーの両方を構成できます。AppFlow エクスポート用に仮想サーバーを構成する前に、AppFlow アクションを EdgeSight モニタリングレスポンダーポリシーに関連付けます。

次のウェブページのパフォーマンスデータは AppFlow にエクスポートされます。

- ページの読み込み時間。ブラウザがレスポンスの最初のバイトの受信を開始してからユーザーがページとのやり取りを開始するまでの経過時間（ミリ秒単位）。この段階では、すべてのページコンテンツが読み込まれない可能性があります。
- ページレンダリング時間。ブラウザがレスポンスの最初のバイトを受け取ってから、すべてのページコンテンツがレンダリングされるか、ページ読み込みアクションがタイムアウトするまでの経過時間（ミリ秒）。
- ページでの滞在時間。ユーザーがページ上で費やした時間。あるページリクエストから次のページリクエストまでの時間を表します。

AppFlow は、インターネットプロトコルフロー情報 eXport (IPFIX) 形式を使用してパフォーマンスデータを送信します。これは、RFC 5101 で定義されているオープンなインターネット技術標準化委員会 (IETF) 標準です。AppFlow テンプレートでは、次のエンタープライズ固有の情報要素（EIE）を使用して情報をエクスポートします。

- クライアントのロード終了時刻。ブラウザがレスポンスの最後のバイトを受信し、画像、スクリプト、スタイルシートなどのページのすべてのオブジェクトをロードする時間。
- クライアントのロード開始時刻。ブラウザーがレスポンスの最初のバイトを受信して、画像、スクリプト、スタイルシートなどのページのオブジェクトをロードする時間。
- クライアントレンダリング終了時間。ブラウザが埋め込みオブジェクトを含むページ全体のレンダリングを終了した時刻。
- クライアントレンダリング開始時刻。ブラウザがページのレンダリングを開始した時刻。

Web ページのパフォーマンスデータを **AppFlow** コレクタにエクスポートするための前提条件

AppFlow アクションを AppFlow ポリシーに関連付ける前に、次の前提条件が満たされていることを確認します。

- AppFlow 機能が有効化され、構成されています。
- レスポンダー機能が有効になりました。
- EdgeSight Monitoring 機能が有効になりました。
- EdgeSight Monitoring は、パフォーマンスデータを収集するアプリケーションのサービスにバインドされた負荷分散またはコンテンツスイッチング仮想サーバー上で有効になっています。

AppFlow アクションを **EdgeSight** モニタリングレスポンダーポリシーに関連付ける

Web ページのパフォーマンスデータを AppFlow コレクタにエクスポートするには、AppFlow アクションを EdgeSight モニタリングレスポンダーポリシーに関連付ける必要があります。AppFlow アクションは、トラフィックを受信するコレクタのセットを指定します。

CLI を使用して **AppFlow** アクションを **EdgeSight** モニタリングレスポンダーポリシーに関連付けるには

コマンドプロンプトで入力します。

```
set responder policy <name> -appflowAction <action_Name>
<!--NeedCopy-->
```

例

```
set responder policy pol -appflowAction actn
<!--NeedCopy-->
```


GUI を使用して **AppFlow** アクションを **EdgeSight** モニタリングレスポンダーポリシーに関連付けるには

1. **AppExpert** > **Responder** > **Policies** に移動します。
2. 詳細ウィンドウで、EdgeSight Monitoring レスポンダーポリシーを選択し、[開く] をクリックします。
3. [レスポンダーポリシーの構成] ダイアログボックスの [**AppFlow** アクション] ドロップダウンリストで、Web ページのパフォーマンスデータを送信するコレクタに関連付けられた AppFlow アクションを選択します。
4. [**OK**] をクリックします。

EdgeSight 統計を **AppFlow** コレクタにエクスポートするように仮想サーバーを構成する

EdgeSight 統計情報を仮想サーバーから AppFlow コレクターにエクスポートするには、AppFlow アクションを仮想サーバーに関連付ける必要があります。

GUI を使用して負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーに **AppFlow** アクションを関連付けるには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。[トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動することもできます。
2. 詳細ウィンドウで、仮想サーバーまたは複数の仮想サーバーを選択し、[**EdgeSight Monitoring** を有効にする] をクリックします。
3. [EdgeSight Monitoring を有効にする] ダイアログボックスで、[**EdgeSight** 統計情報を **Appflow** にエクスポート] チェックボックスをオンにします。
4. [AppFlow アクション] ドロップダウンリストから、**AppFlow** アクションを選択します。AppFlow アクションは、EdgeSight Monitoring 統計をエクスポートする AppFlow コレクタのリストを定義します。複数の負荷分散仮想サーバーを選択した場合、同じ AppFlow アクションがそれらにバインドされたレスポンダーポリシーに関連付けられます。必要に応じて、選択した負荷分散仮想サーバーごとに構成された AppFlow アクションを個別に変更できます。
5. [**OK**] をクリックします。

Citrix ADC 高可用性ペアでのセッションの信頼性

July 28, 2022

ICA セッション中にネットワークの中断またはデバイスのフェイルオーバーが発生した場合、セッションの再接続では、セッション画面の保持またはクライアントの自動再接続という 2 つのメカニズムのいずれかを使用できます。

セッションの信頼性。推奨モードは、ユーザーにとってスムーズなエクスペリエンスです。この中断は、短時間のネットワークの中断ではほとんど目立ちません。

クライアントの自動再接続。フォールバックオプションには、クライアントの再起動が含まれます。このメカニズムはユーザーにとって混乱を招き、常にサポートされているわけではありません。

HDX Insight が有効になっている場合、Receiver は ICA セッション画面の保持機能を使用して ICA セッションをシームレスに再接続できます。
この機能は、スタンドアロン構成と Citrix ADC HA ペア構成の両方で、また Citrix ADC フェイルオーバーが発生した場合でも機能します。

> 注:
>
> - Citrix ADC アプライアンスは、ソフトウェアバージョン 11.1 ビルド 49.16 以降で実行されている必要があります。
> - Citrix ADC アプライアンスにアクティブな接続がある場合は、セッション画面の保持モードを有効または無効にしないでください。
> - 接続がまだアクティブなときにこの機能を有効または無効にすると、HDX Insight はフェイルオーバーの発生後にこれらのセッションの解析を停止します。その結果、セッションに関する情報が失われます。
> - 高可用性セットアップでのセッション画面の保持は、Citrix ADC ソフトウェアバージョン 11.1 49.16 以降ではデフォルトで無効になっています。セッション画面の保持は、セットアップの両方のノードが同じビルド（たとえば、リリース 11.1 ビルド 53）を実行している場合にのみ、高可用性セットアップでサポートされます。つまり、両方のノードが異なるビルドを実行している場合（たとえば、一方のノードにリリース 11.1 のビルド 53 があり、もう一方のノードにリリース 11.1 のビルド 56 がある場合）、セッション画面の保持は高可用性セットアップではサポートされません。SSL VDA のセッションの信頼性は、次の条件が満たされた場合にサポートされます。
>   - The “EnableSRonHAFailover” parameter in the `set ica parameter` command must be YES.
>   - The HTTPS must be used instead of HTTP while configuring the virtual server.
> - HDX Insight を有効にすると、EnableSronhaFailover パラメーターが無効になっていても、高可用性フェイルオーバー後に基本暗号化アプリケーションとデスクトップが再接続されます。

CLI を使用してセッション画面の保持を設定するには、次の手順を実行します。

1. コマンドラインで、既定のシステム管理者の資格情報を使用してシステムにログオンします。
2. HA フェールオーバーでセッション画面の保持を有効にするには、プロンプトで次のように入力します。
   `set ica parameter EnableSRonHAFailover YES`
3. HA フェールオーバーでセッション画面の保持を無効にするには、プロンプトで次のように入力します。
   `set ica parameter EnableSRonHAFailover NO`

GUI を使用して **HA** フェールオーバーでセッション画面の保持を有効にするには、次の手順を実行します。

1. Web ブラウザで、HA ペアのプライマリ Citrix ADC インスタンスの IP アドレスを入力します（例: http://192.168.100.1）。
2. [**User Name**] と [**Password**] に管理者の資格情報を入力します。
3. [構成] タブで、[システム] > [設定] に移動し、[**ICA** パラメータの変更] をクリックします。
4. [**ICA** パラメータの変更] セクションで、[高可用性フェイルオーバー時のセッション画面の保持] を選択しま

す。

5.「**OK**」をクリックします。

制限事項

- この機能を有効にすると、この機能によって ICA 圧縮が無効になるため、帯域幅の消費が増加します。また、プライマリノードとセカンダリノード間の追加のトラフィックにより、それらの同期が維持されます。
- この機能はアクティブ-パッシブモードでのみサポートされています。アクティブ-アクティブモードは現在サポートされていません。
- HDX Insight が有効で、HA ノブのセッション画面の保持が［いいえ］に設定されている場合、Citrix ADC 高可用性フェイルオーバーシナリオでは ACR 再接続モードのみがサポートされます。HDX Insight が無効になっている場合、HA ノブはセッション画面の保持を無効にしません。

セッション再接続セマンティクスの表は次のとおりです。

セッションはセマンティクスを再接続

ステータス	RONHA フェイルオーバーを有効にするはい	EnableSRonHAFailover No（デフォルト）
HDX Insight 有効	ICA セッションのセッション再接続は機能する	ICA セッションのセッション再接続が機能しない
HDX Insight 無効	ICA セッションのセッション再接続は機能する	ICA セッションのセッション再接続は機能する

> 注意事項

- ICA セッションのセッション画面の保持は、Citrix Gateway でそのまま使用できます。
- 次の両方の条件が満たされている場合、ICA セッションのセッション画面の保持は機能しません。
  - HDX Insight が有効になっている
  - EnableSRonHAFailover が NO に設定されている
- HDX Insight が無効になっている場合、EnableSronhaFailover ノブを「はい」または「いいえ」に設定しても違いはありません。

Citrix Web App Firewall

October 7, 2021

次のトピックでは、Citrix Web AppFirewall 機能のインストールと構成の詳細について説明します。

はじめに	Web セキュリティの概要と Web App Firewall しくみ。
構成	Web サイト、Web サービス、または Web2.0 サイトを保護するように WebAppFirewall を構成する方法。
署名	シグニチャに関する詳細な説明と、サポートされている脆弱性スキャンツールからシグニチャを構成する方法、および例を使用して独自のシグニチャを定義する方法。
セキュリティー検査の概要	構成情報と例を含む、Web AppFirewall のセキュリティチェックの詳細な説明。
プロファイル	Web App Firewall でのプロファイルの構成方法および使用方法の説明。
ポリシー	Web App Firewall の設定時にポリシーがどのように使用されるかを説明し、便利なポリシーの例も示します。
インポート	Web App Firewall でさまざまな種類のインポートされたファイルを使用する方法、およびファイルのインポートとエクスポートの方法の説明。
グローバル設定	すべてのプロファイルに適用される Web App Firewall 機能の説明、およびそれらの設定方法。
使用例	特定の種類のより複雑な Web サイトおよび Web サービスを最適に保護するために WebAppFirewall を設定する方法を示す拡張例。
ログ、統計、およびレポート	Web App Firewall のログ、統計、およびレポートにアクセスして使用する方法。WebAppFirewall の構成に役立ちます。

Citrix Web App Firewall では、さまざまなアプリケーションセキュリティ要件を満たすように簡単に構成できます。一連のセキュリティチェックで構成される Web App Firewall プロファイルを使用して、詳細なパケットレベルの検査を行うことで、要求と応答の両方を保護できます。各プロファイルには、基本保護または高度な保護を選択するオプションがあります。保護によっては、他のファイルの使用が必要になる場合があります。たとえば、XML 検証チェックでは、WSDL ファイルまたはスキーマファイルが必要な場合があります。プロファイルでは、署名やエラーオブジェクトなどの他のファイルも使用できます。これらのファイルはローカルで追加することも、事前にインポートして将来使用するためにアプライアンスに保存することもできます。

各ポリシーはトラフィックのタイプを識別し、そのトラフィックは、ポリシーに関連付けられたプロファイルに指定されたセキュリティチェック違反がないか検査されます。ポリシーは、ポリシーのスコープを決定する異なるバインドポイントを持つことができます。たとえば、特定の仮想サーバにバインドされているポリシーが呼び出され、その仮想サーバを通過するトラフィックだけが評価されます。ポリシーは、指定された優先順位に従って評価され、要求または応答に一致する最初のポリシーが適用されます。

- Web App Firewall 保護の迅速な展開

  Web App Firewall セキュリティをすばやく展開するには、次の手順を使用します。

  1. Web App Firewall プロファイルを追加し、アプリケーションのセキュリティ要件に適したタイプ（html、xml、JSON）を選択します。
  2. 必要なセキュリティレベル (基本または詳細) を選択します。
  3. 署名や WSDL などの必要なファイルを追加またはインポートします。
  4. ファイルを使用するようにプロファイルを設定し、その他の必要な変更をデフォルト設定に加えます。
  5. このプロファイルに WebAppFirewall ポリシーを追加します。
  6. ポリシーをターゲットのバインドポイントにバインドし、優先順位を指定します。

- Web App Firewall エンティティ

  Profile: Web App Firewall プロファイルは、検索対象と何をすべきかを指定します。要求と応答の両方を検査して、チェックする必要のある潜在的なセキュリティ違反と、トランザクションの処理時に実行する必要のあるアクションを決定します。プロファイルは、HTML、XML、または HTML と XML のペイロードを保護できます。アプリケーションのセキュリティ要件に応じて、基本プロファイルまたは詳細プロファイルのいずれかを作成できます。基本プロファイルは、既知の攻撃から保護できます。より高いセキュリティが必要な場合は、高度なプロファイルをデプロイして、アプリケーションリソースへの制御されたアクセスを許可し、ゼロデイ攻撃をブロックできます。ただし、基本プロファイルを変更して高度な保護を提供することはできますが、その逆も可能です。複数のアクション (ブロック、ログ、学習、変換など) を選択できます。高度なセキュリティチェックでは、セッション Cookie と非表示のフォームタグを使用して、クライアント接続の制御と監視を行うことができます。Web App Firewall プロファイルは、トリガーされた違反を学習し、緩和ルールを提案できます。

  基本保護: 基本プロファイルには、開始 URL および拒否 URL 緩和ルールの事前構成済みセットが含まれます。これらの緩和ルールは、どの要求を許可し、どの要求を拒否する必要があるかを決定します。着信要求はこれらのリストと照合され、設定されたアクションが適用されます。これにより、ユーザーはリラクゼーションルールの最小構成でアプリケーションを保護することができます。開始 URL ルールは、強制的な閲覧から保護します。ハッカーによって悪用される既知の Web サーバーの脆弱性は、デフォルトの [拒否 URL] ルールのセットを有効にすることで検出およびブロックできます。バッファオーバーフロー、SQL、クロスサイトスクリプティングなど、一般的に起動される攻撃も簡単に検出できます。

  Advanced Protections: 名前が示すように、高度な保護は、より高いセキュリティ要件を持つアプリケーションに使用されます。リラクゼーションルールは、特定のデータのみへのアクセスを許可し、残りのデータをブロックするように構成されています。この肯定的なセキュリティモデルは、基本的なセキュリティチェックでは検出されない可能性のある未知の攻撃を軽減します。すべての基本的な保護に加えて、高度なプロファ

イルは、ブラウジングの制御、クッキーのチェック、さまざまなフォームフィールドの入力要件の指定、フォームの改ざんやクロスサイトリクエスト偽造攻撃からの保護によって、ユーザーセッションを追跡します。トラフィックを監視し、適切な緩和を展開するラーニングは、多くのセキュリティチェックでデフォルトで有効になっています。使いやすいが、高度な保護は、セキュリティを強化するだけでなく、より多くの処理を必要とし、パフォーマンスに影響を与えることができるキャッシュの使用を許可しないため、十分に考慮する必要があります。

インポート：インポート機能は、Web App Firewall プロファイルで外部ファイル、つまり外部または内部 Web サーバーでホストされているファイル、またはローカルマシンからコピーする必要がある場合に役立ちます。ファイルをインポートしてアプライアンスに保存すると、特に外部 Web サイトへのアクセスを制御する必要がある場合、コンパイルに時間がかかる場合、大きなファイルを HA デプロイメント間で同期する必要がある場合、または次の方法でファイルを再利用できる場合に便利です。複数のデバイス間でコピーします。次に例を示します：

- 外部 Web サーバーでホストされている WSDL は、外部 Web サイトへのアクセスをブロックする前にローカルにインポートできます。
- Cenzic などの外部スキャンツールで生成された大きな署名ファイルは、Citrix アプライアンスのスキーマを使用してインポートおよびプリコンパイルできます。
- カスタマイズされた HTML または XML エラーページは、外部 Web サーバーからインポートすることも、ローカルファイルからコピーすることもできます。

署名：シグニチャは、パターンマッチングを使用して悪意のある攻撃を検出し、トランザクションのリクエストとレスポンスの両方をチェックするように構成できるため、強力です。これらは、カスタマイズ可能なセキュリティソリューションが必要な場合に推奨されるオプションです。シグニチャの一致が検出されたときに実行するアクションには、複数の選択肢（ブロック、ログ、学習、変換など）を使用できます。Web App Firewall には、1,300 を超えるシグニチャルールで構成される既定のシグニチャオブジェクトが組み込まれており、自動更新機能を使用して最新のルールを取得することもできます。他のスキャンツールで作成されたルールもインポートできます。シグニチャオブジェクトは、Web AppFirewall プロファイルで指定された他のセキュリティチェックと連携できる新しいルールを追加することでカスタマイズできます。シグニチャルールは複数のパターンを持つことができ、すべてのパターンが一致する場合にのみ違反にフラグを立てることができるため、誤検出を回避できます。ルールのリテラル `fastmatch` パターンを注意深く選択すると、処理時間を大幅に最適化できます。

ポリシー：Web App ファイアウォールポリシーは、トラフィックをフィルタリングし、異なるタイプに分割するために使用されます。これにより、アプリケーションデータに対してさまざまなレベルのセキュリティ保護を実装する柔軟性が得られます。機密性の高いデータへのアクセスは高度なセキュリティチェック検査に向けることができ、機密性の低いデータは基本レベルのセキュリティ検査によって保護されます。ポリシーは、無害なトラフィックのセキュリティチェック検査をバイパスするように設定することもできます。セキュリティを高めるにはより多くの処理が必要となるため、ポリシーを慎重に設計するとともに、最適なパフォーマンスを実現できます。ポリシーの優先度によって評価される順序が決まり、バインドポイントによってアプリケーションのスコープが決まります。

ハイライト

1. さまざまなタイプのデータを保護し、さまざまなリソースに対して適切なレベルのセキュリティを実装し、パフォーマンスを最大限に引き出すことで、幅広いアプリケーションを保護できます。
2. セキュリティ構成を追加または変更する柔軟性。基本保護と高度な保護を有効または無効にすることで、セキュリティチェックを強化または緩和できます。
3. HTML プロファイルを XML または Web2.0 に変換するオプション (HTML+XML) プロファイルと逆に、さまざまなタイプのペイロードにセキュリティを追加する柔軟性を提供します。
4. 簡単にデプロイされたアクションにより、攻撃をブロックしたり、ログで監視したり、統計情報を収集したり、攻撃文字列を変換して無害にします。
5. 着信要求を検査して攻撃を検出し、サーバーから送信された応答を検査することによって機密データの漏えいを防止する機能。
6. トラフィックパターンから学習して、簡単に編集可能な緩和ルールに関する推奨事項を取得でき、例外を許可するように展開できます。
7. ハイブリッドセキュリティモデル。カスタマイズ可能なシグニチャのパワーを適用して、特定のパターンに一致する攻撃をブロックし、基本または高度なセキュリティ保護に対してポジティブセキュリティモデルチェックを使用する柔軟性を提供します。
8. PCI-DSS 準拠に関する情報を含む、包括的な設定レポートの可用性。

よくある質問と導入ガイド

October 7, 2021

Q: アプリケーションのセキュリティ保護に **Citrix Web App Firewall** が推奨されるのはなぜですか**?**

次の機能を備えた CitrixWeb App Firewall は、包括的なセキュリティソリューションを提供します。

- ハイブリッドセキュリティモデル： Citrix ADC ハイブリッドセキュリティモデルを使用すると、ポジティブなセキュリティモデルとネガティブなセキュリティモデルの両方を活用して、アプリケーションに最適な構成を考え出すことができます。
  - ポジティブセキュリティモデルは、バッファオーバーフロー、CGI-BIN パラメータ操作、Form/Hidden フィールド操作、強制ブラウジング、Cookie またはセッションポイズニング、ACL の破損、クロスサイトスクリプティング（クロスサイトスクリプティング）、コマンドインジェクション、SQL インジェクション、機密情報の漏洩のトリガーエラー、暗号化の安全でない使用、サーバーの設定ミス、バックドアおよびデバッグオプション、レートベースのポリシー施行、よく知られたプラットフォームの脆弱性、ゼロデイエクスプロイト、クロスサイトリクエスト偽造（CSRF）、およびクレジットカードやその他の機密データの漏洩。
  - ネガティブセキュリティモデルでは、L7 および HTTP アプリケーションの脆弱性から保護するために、リッチセットシグニチャを使用します。Web App Firewall は、Cenzic、Qualys、ホワイトハット、

IBM が提供するサードパーティ製のスキャンツールと統合されています。組み込みの XSLT ファイルを使用すると、ネイティブフォーマットの Snort ベースのルールと組み合わせて使用できるルールを簡単にインポートできます。自動更新機能は、新しい脆弱性の最新の更新プログラムを取得します。

ポジティブなセキュリティモデルは、どのデータにアクセスできるかを完全に制御できるオプションを提供するため、セキュリティに対する高いニーズを持つアプリケーションを保護するうえで望ましい選択です。あなたが望むものだけを許可し、残りの部分をブロックします。このモデルには、数回クリックするだけで展開できるセキュリティチェック設定が組み込まれています。ただし、セキュリティが厳しくなるほど、処理オーバーヘッドは大きくなることに注意してください。

ネガティブなセキュリティモデルは、カスタマイズされたアプリケーションに適しています。シグニチャを使用すると、複数の条件を組み合わせることができます。一致と指定されたアクションは、すべての条件が満たされた場合にのみトリガーされます。あなたは望ましくないものだけをブロックし、残りを許可します。特定の場所で特定の高速一致パターンを使用すると、処理オーバーヘッドを大幅に削減してパフォーマンスを最適化できます。アプリケーションの特定のセキュリティニーズに基づいて独自のシグネチャルールを追加するオプションを使用すると、独自のカスタムセキュリティソリューションを柔軟に設計できます。

- 要求と応答側の検出と保護**:** 受信要求を検査して不審な動作を検出し、適切なアクションを実行できます。また、応答を確認して、機密データの漏洩を検出して保護することができます。
- **HTML**、**XML**、**JSON** ペイロード用の豊富な組み込み保護セット**:** Web App Firewall には 19 種類のセキュリティチェックが用意されています。そのうちの 6 つ（開始 URL や拒否 URL など）は、HTML データと XML データの両方に適用されます。5 つのチェック（フィールドの一貫性、フィールド形式など）は HTML に固有で、8 つのチェック（XML 形式や Web サービスの相互運用性など）は XML ペイロードに固有です。この機能には、豊富なアクションとオプションが含まれています。たとえば、URL クロージャを使用すると、Web サイト内のナビゲーションを制御および最適化し、正当なすべての URL を許可するように緩和ルールを設定しなくても、強制的なブラウジングから保護できます。応答に含まれるクレジットカード番号などの機密データを削除するか、除外するかを選択できます。SOAP Array 攻撃保護、XML サービス拒否（XDoS）、WSDL スキャン防止、添付ファイルのチェック、または任意の数の XML 攻撃であっても、アプリケーションが Web App Firewall によって保護されているときにデータを保護する防御シールドがあることを知っているという安心感があります。シグニチャを使用すると、XPATH-Expressions を使用して本文内の違反や JSON ペイロードのヘッダーを検出するルールを設定できます。
- **GWT**： SQL、クロスサイトスクリプティング、およびフォームフィールドの整合性チェック違反から保護するための Google WebToolkit アプリケーションの保護のサポート。
- **Java** フリーのユーザーフレンドリーなグラフィカル・ユーザー・インターフェイス（**GUI**）**:** 直感的な GUI と事前構成されたセキュリティ・チェックにより、いくつかのボタンをクリックするだけで簡単にセキュリティを展開できます。ウィザードの指示に従って、プロファイル、ポリシー、シグニチャ、バインディングなどの必要な要素を作成します。HTML5 ベースの GUI は、任意の Java の依存関係の自由です。パフォーマンスは、古い Java ベースのバージョンのパフォーマンスよりも大幅に優れています。
- 使いやすく、自動化可能な **CLI**： GUI で利用可能な設定オプションのほとんどは、コマンドラインインターフェイス（CLI）でも利用可能です。CLI コマンドはバッチファイルで実行でき、自動化が容易です。

- **REST API** のサポート：Citrix ADC NITRO プロトコルは、Web App Firewall 構成を自動化し、セキュリティ違反を継続的に監視するための関連する統計を収集する REST API の豊富なセットをサポートしています。
- 学習**:** Web App Firewall は、トラフィックを監視してセキュリティを微調整することで学習できる機能は非常にユーザーフレンドリーです。学習エンジンはルールを推奨しています。これにより、正規表現に習熟しなくても緩和を簡単に展開できます。
- **RegEx** エディタのサポート：正規表現は、ルールを統合して検索を最適化したいというジレンマに対するエレガントなソリューションを提供します。正規表現のパワーを活用して、URL、フィールド名、シグニチャパターンなどを設定できます。豊富な組み込みの GUI RegEx エディタを使用すると、式のクイックリファレンスが提供され、RegEx の精度を検証してテストする便利な方法が提供されます。
- カスタマイズされたエラーページ：ブロックされたリクエストは、エラー URL にリダイレクトすることができます。また、サポートされている変数と Citrix のデフォルトの構文（高度な PI 式）を使用して、クライアントのトラブルシューティング情報を埋め込むカスタマイズされたエラーオブジェクトを表示するオプションもあります。
- **PCI-DSS**、統計情報、およびその他の違反レポート**:** 豊富なレポートにより、PCI-DSS コンプライアンス要件への対応、トラフィックカウンタに関する統計情報の収集、およびすべてのプロファイルまたは 1 つのプロファイルの違反レポートの表示が容易になります。
- ログからのログとクリック・トゥ・ルール：詳細なログは、ネイティブおよび CEF 形式でもサポートされています。Web App Firewall では、syslog ビューアで対象のログメッセージをフィルタリングできます。ボタンをクリックするだけで、ログメッセージを選択し、対応する緩和ルールを展開できます。ログメッセージを柔軟にカスタマイズできるほか、Web ログの生成もサポートできます。詳細については、「Web App Firewall ログ」トピックを参照してください。
- 違反ログをトレースレコードに含める**:** トレースレコードにログメッセージを含める機能により、リセットやブロックなどの予期しない動作を簡単にデバッグできます。
- クローン作成：便利なインポート/エクスポートプロファイルオプションを使用すると、Citrix ADC アプライアンス間でセキュリティ設定をクローンできます。[学習したデータのエクスポート] オプションを使用すると、学習したルールを Excel ファイルに簡単にエクスポートできます。その後、適用する前に、アプリケーション所有者によってレビューおよび承認を受けることができます。
- **AppExpert** テンプレート (構成設定のセット) は、Web サイトに対して適切な保護を提供するように設計できます。これらのクッキーカッターテンプレートをテンプレートにエクスポートすることで、同様の保護を他のアプライアンスに展開するプロセスを簡素化し、迅速化できます。

詳細については、AppExpert テンプレートのトピックを参照してください。

- セッションレスセキュリティチェック**:** セッションレスセキュリティチェックを展開すると、メモリフットプリントを削減し、処理を迅速化するのに役立ちます。
- 他の **Citrix ADC** 機能との相互運用性： Web App Firewall は、書き換え、URL 変換、統合キャッシュ、CVPN、レート制限などの Citrix ADC の他の機能とシームレスに動作します。

- ポリシーでの **PI** 式のサポート**:** 高度な PI 式のパワーを活用して、アプリケーションのさまざまな部分に異なるレベルのセキュリティを実装するポリシーを設計できます。
- **IPv6** のサポート**:** Web App Firewall は IPv4 プロトコルと IPv6 プロトコルの両方をサポートします。
- 地理位置情報ベースのセキュリティ保護： Citrix のデフォルトの構文（PI 式）を使用して、ロケーションベースのポリシーを柔軟に設定できます。このポリシーは、組み込みのロケーションデータベースと組み合わせて使用して、ファイアウォールの保護をカスタマイズできます。悪意のある要求が発信された場所を特定し、特定の地理的な場所から発信された要求に対して、必要なレベルのセキュリティチェック検査を実施できます。
- パフォーマンス**:** リクエスト側の ストリーミングにより、パフォーマンスが大幅に向上します。フィールドが処理されるとすぐに、結果のデータがバックエンドに転送され、残りのフィールドの評価が続行されます。処理時間の改善は、大きなポストを処理する場合に特に重要です。
- その他のセキュリティ機能： Web App Firewall には、データのセキュリティを確保するのに役立つその他のセキュリティ設定がいくつかあります。たとえば、**[**機密フィールド**]** を使用すると、ログメッセージ内の機密情報の漏洩をブロックできます。[**HTML** コメントの削除**]** を使用すると、応答から HTML コメントを削除してからクライアントに転送できます。フィールドタイプは、アプリケーションに提出されたフォームで許可される入力を指定するために使用することができます。

Q: Web App Firewall を設定するには何が必要ですか**?**

以下を実行します:

- Web App Firewall プロファイルを追加し、アプリケーションのセキュリティ要件に適したタイプ（html、xml、web2.0）を選択します。
- 必要なセキュリティレベル (基本または詳細) を選択します。
- 署名や WSDL などの必要なファイルを追加またはインポートします。
- ファイルを使用するようにプロファイルを設定し、その他の必要な変更をデフォルト設定に加えます。
- このプロファイルの Web App Firewall ポリシーを追加します。
- ポリシーをターゲットのバインドポイントにバインドし、優先順位を指定します。

Q: どのプロファイルタイプを選択すべきかを知るにはどうすればよいですか**?**

Web App Firewall プロファイルは、HTML ペイロードと XML ペイロードの両方を保護します。アプリケーションの必要性に応じて、HTML プロファイルまたは XML プロファイルを選択できます。アプリケーションが HTML データと XML データの両方をサポートしている場合は、Web2.0 プロファイルを選択できます。

Q: 基本プロファイルと詳細プロファイルの違いは何ですか**?** 必要なものをどのように決めるのですか**?**

基本プロファイルまたは事前プロファイルを使用するかどうかは、アプリケーションのセキュリティ上の必要性によって異なります。基本プロファイルには、開始 URL 緩和ルールと拒否 URL 緩和ルールの事前構成セットが含まれています。これらの緩和規則によって、許可される要求と拒否される要求が決まります。着信要求は事前設定されたルールと一致し、設定されたアクションが適用されます。ユーザーは、緩和規則の最小構成でアプリケーションを保護

できます。開始 URL ルールは、強制的な閲覧から保護します。ハッカーによって悪用される既知の Web サーバーの脆弱性は、デフォルトの [拒否 URL] ルールのセットを有効にすることで検出およびブロックできます。バッファオーバーフロー、SQL、クロスサイトスクリプティングなどの一般的に起動された攻撃も簡単に検出できます。

名前が示すように、高度な保護は、より高いセキュリティ要件を持つアプリケーション用です。リラクゼーションルールは、特定のデータのみへのアクセスを許可し、残りのデータをブロックするように構成されています。この肯定的なセキュリティモデルは、基本的なセキュリティチェックでは検出されない可能性のある未知の攻撃を軽減します。すべての基本的な保護に加えて、高度なプロファイルは、ブラウジングの制御、クッキーのチェック、さまざまなフォームフィールドの入力要件の指定、フォームの改ざんや Cross-Site Request Forgery 攻撃からの保護によってユーザーセッションを追跡します。トラフィックを監視し、適切な緩和を推奨するラーニングは、多くのセキュリティチェックでデフォルトで有効になっています。使いやすくなりますが、高度な保護では、セキュリティが厳しくなるだけでなく、処理能力も増えるため、十分に考慮する必要があります。一部の事前セキュリティチェックでは、キャッシュの使用が許可されていないため、パフォーマンスに影響する可能性があります。

基本プロファイルまたは詳細プロファイルのどちらを使用するかを決定するときは、次の点に注意してください。

- 基本プロファイルおよび高度なプロファイルは、テンプレートを開始するだけです。基本プロファイルを変更して高度なセキュリティ機能を展開できます。その逆も同様です。
- 高度なセキュリティー検査では、より多くの処理が必要となり、パフォーマンスに影響を与える可能性があります。アプリケーションに高度なセキュリティが必要でない限り、基本プロファイルから始めて、アプリケーションに必要なセキュリティを強化することができます。
- アプリケーションで必要な場合を除き、すべてのセキュリティーチェックを有効にする必要はありません。

Q: ポリシーとは何ですか**?** バインドポイントを選択して優先度を設定するにはどうすればよいですか**?**

Web App Firewall ポリシーは、さまざまなレベルのセキュリティ実装を設定するために、トラフィックを論理グループに分類するのに役立ちます。ポリシーのバインドポイントを慎重に選択して、どのトラフィックがどのポリシーと一致するかを決定します。たとえば、すべての着信リクエストをチェックする場合 SQL/cross-site スクリプト攻撃では、一般的なポリシーを作成してグローバルにバインドできます。または、機密データを含むアプリケーションをホストする仮想サーバーのトラフィックに、より厳格なセキュリティチェックを適用する場合は、ポリシーをその仮想サーバーにバインドできます。

優先順位を慎重に割り当てると、トラフィック処理が強化されます。より具体的なポリシーには高いプライオリティを、一般的なポリシーには低いプライオリティを割り当てたい場合。数値が大きいほど、プライオリティは低くなります。プライオリティが 10 のポリシーは、プライオリティが 15 のポリシーよりも先に評価されます。

異なる種類のコンテンツに対して異なるレベルのセキュリティを適用できます。たとえば、画像やテキストなどの静的オブジェクトの要求は、1 つのポリシーを使用してバイパスでき、他の機密コンテンツに対する要求は、2 番目のポリシーを使用して厳密なチェックを受けることができます。

Q: アプリケーションを保護するためのルールの設定はどのようにすればよいですか**?**

Web App Firewall は、あなたのウェブサイトのためのセキュリティの適切なレベルを設計することが非常に容易になります。複数の Web App Firewall ポリシーをさまざまな Web App Firewall プロファイルにバインドして、アプリケーションにさまざまなレベルのセキュリティチェック検査を実装できます。最初にログを監視して、どのセキュリティ脅威が検出され、どの違反がトリガーされているかを確認できます。緩和ルールを手動で追加するか、Web App Firewall で推奨される学習済みルールを利用して、必要な緩和をデプロイして、誤検出を回避できます。

Citrix Web App Firewall は、GUI で ビジュアライザをサポートしているため、ルール管理が非常に簡単です。1 つの画面ですべてのデータを簡単に表示し、1 回のクリックで複数のルールに対してアクションを実行できます。ビジュアライザーの最大の利点は、複数のルールを統合するために正規表現を推奨することです。デリミタとアクション URL に基づいて、ルールのサブセットを選択できます。ビジュアライザのサポートは、1) 学習したルールと 2) 緩和ルールの表示に使用できます。

1. 学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。ルールをスキップ (無視) することもできます。
2. デプロイされた緩和のビジュアライザーには、新しいルールを追加するか、既存のルールを編集するオプションがあります。ノードを選択し、緩和ビジュアライザの [有効] または [無効] ボタンをクリックして、ルールのグループを有効または無効にすることもできます。

Q: 署名とは何ですか**?** どのシグニチャを使用するかを知るにはどうすればよいですか**?**

シグニチャは、複数のルールを持つことができるオブジェクトです。各ルールは、指定した一連のアクションに関連付けることができる 1 つ以上のパターンで構成されます。Web App Firewall には、1,300 を超えるシグニチャルールで構成される既定のシグニチャオブジェクトが組み込まれています。また、自動更新機能を使用して最新のルールを取得して新しい脆弱性から保護することもできます。他のスキャンツールで作成されたルールもインポートできます。

シグニチャは、悪意のある攻撃を検出するためにパターンマッチングを使用し、トランザクションの要求と応答の両方をチェックするように構成できるため、非常に強力です。これらは、カスタマイズ可能なセキュリティソリューションが必要な場合に推奨されるオプションです。シグニチャの一致が検出されると、複数のアクションの選択（ブロック、ログ、学習、変換など）を使用できます。デフォルトのシグニチャは、web-cgi、web-coldfusion、web-frontpage、web-iis、web-php、web-client、web-activex、web-shell-shock、web-struts などのさまざまなタイプのアプリケーションを保護するためのルールをカバーしています。アプリケーションのニーズに合わせて、特定のカテゴリに属するルールを選択してデプロイできます。

署名の使用に関するヒント:

- デフォルトのシグニチャオブジェクトのコピーを作成し、それを変更して、必要なルールを有効にし、必要なアクションを設定できます。
- シグニチャオブジェクトは、他のシグニチャルールと連携して機能する新しいルールを追加することでカスタマイズできます。
- 署名ルールは、Web App Firewall プロファイルに指定されたセキュリティチェックと連動するように設定することもできます。違反を示す一致がシグニチャとセキュリティチェックによって検出された場合、より制限

の厳しいアクションが強制されます。

- シグニチャルールは複数のパターンを持つことができ、すべてのパターンが一致したときにだけ違反にフラグを付けるように設定できるため、誤検出を回避できます。
- ルールに対してリテラル高速一致パターンを慎重に選択すると、処理時間を大幅に最適化できます。

Q: Web App Firewall は、Citrix ADC の他の機能と連携しますか?

Web App Firewall は、Citrix ADC アプライアンスに完全に統合され、他の機能とシームレスに動作します。他の Citrix ADC セキュリティ機能を Web App Firewall と組み合わせて使用することで、アプリケーションのセキュリティを最大限に高めることができます。たとえば、**AAA-TM** を使用して、ユーザーの認証、コンテンツへのアクセス許可の確認、無効なログイン試行を含むアクセスの記録を行うことができます。書き換えを使用すると、URL を変更したり、ヘッダーを追加、変更、削除したりできます。また、**Responder** を使用すると、カスタマイズされたコンテンツをさまざまなユーザーに配信できます。あなたは、トラフィックを監視し、それが高すぎる場合はレートを調整するためにレート 制限を使用して、あなたのウェブサイトの最大負荷を定義することができます。**HTTP** サービス拒否 **(DoS)** 保護は、実際の HTTP クライアントと悪意のある DoS クライアントを区別するのに役立ちます。Web App Firewall ポリシーを仮想サーバーにバインドすることで、セキュリティチェック検査の範囲を絞り込むことができます。負荷分散機能を使用して頻繁に使用するアプリケーションを管理することで、ユーザーエクスペリエンスを最適化できます。画像やテキストなどの静的オブジェクトの要求は、セキュリティチェック検査をバイパスし、統合されたキャッシュまたは 圧縮を利用して、そのようなコンテンツの帯域幅の使用を最適化できます。

Q: ペイロードは、Web App Firewall およびその他の Citrix ADC 機能によってどのように処理されますか?

Citrix ADC アプライアンスの L7 パケットフローの詳細を示す図は、［機能の処理順序 ］セクションにあります。

Q: Web App Firewall デプロイに推奨されるワークフローは何ですか?

Citrix Web App Firewall の最新のセキュリティ保護を使用する利点がわかったので、セキュリティニーズに最適なソリューションの設計に役立つ追加情報を収集できます。以下をお勧めします:

- 環境を知る**:** 環境を知ることで、ニーズに最適なセキュリティ保護ソリューション (署名、セキュリティチェック、またはその両方) を特定できます。構成を開始する前に、次の情報を収集する必要があります。
  - **OS**: どのような OS がありますか（MS Windows、Linux、BSD、Unix など）?
  - **Web** サーバー: どの Web サーバー（IIS、Apache または Citrix ADC エンタープライズサーバー）を実行していますか?
  - アプリケーション**:** アプリケーションサーバーで実行されているアプリケーションの種類 (たとえば、ASP.NET、PHP、コールドフュージョン、ActiveX、フロントページ、ストラット、CGI、アパッチ Tomcat、ドミノ、WebLogic)?
  - カスタマイズされたアプリケーションや既製のアプリケーション（Oracle、SAP など）がありますか。どのバージョンを使用していますか?

  - **SSL**：SSL が必要ですか? その場合、証明書の署名にはどのキーサイズ (512、1024、2048、4096) が使用されますか。
  - トラフィック量：アプリケーションの平均トラフィックレートはいくらですか? トラフィックに季節的または時間特有の急増がありますか?
  - サーバーファーム**:** サーバーはいくつありますか? 負荷分散を使用する必要がありますか。
  - データベース：どのタイプのデータベース（MS-SQL、MySQL、Oracle、Postgres、SQLite、nosql、Sybase、Informix など）を使用していますか?
  - **DB** 接続：どのような種類のデータベース接続（DSN、ファイルごとの接続文字列、単一のファイル接続文字列）があり、どのドライバが使用されていますか?
- セキュリティニーズの特定**:** 最大限のセキュリティ保護を必要とするアプリケーションまたは特定のデータ、脆弱性の低いアプリケーション、およびセキュリティ検査を安全にバイパスできるアプリケーションを評価できます。これにより、最適な構成を考え出すことや、トラフィックを分離するための適切なポリシーとバインドポイントの設計に役立ちます。たとえば、画像、MP3 ファイル、ムービーなどの静的 Web コンテンツに対する要求のセキュリティ検査をバイパスするポリシーを設定し、動的コンテンツに対する要求に高度なセキュリティチェックを適用するように別のポリシーを構成できます。複数のポリシーとプロファイルを使用して、同じアプリケーションの異なるコンテンツを保護できます。
- ライセンス要件： Citrix は、負荷分散、コンテンツスイッチング、キャッシュ、圧縮、レスポンダー、書き換え、コンテンツフィルタリングなどの豊富な機能を利用して、アプリケーションのパフォーマンスを最適化する統合ソリューションを提供します。必要な機能を特定することで、必要なライセンスを決定できます。
- **Citrix ADC** アプライアンスのインストールとベースライン：仮想サーバーを作成し、それを介してテストトラフィックを実行して、システムを通過するトラフィックの速度と量を把握します。この情報は、容量要件を特定し、適切なアプライアンス（VPX、MPX、SDX）を選択するのに役立ちます。
- **Web App** ファイアウォールの展開**:** Web App Firewall ウィザードを使用して、簡単なセキュリティ構成を続行します。ウィザードでは、いくつかの画面が表示され、プロファイル、ポリシー、署名、およびセキュリティチェックを追加するように求められます。
  - プロファイル**:** わかりやすい名前と、プロファイルに適したタイプ (HTML、XML、または WEB 2.0) を選択します。ポリシーとシグニチャは、同じ名前を使用して自動生成されます。
  - [**Policy**]：自動生成されたポリシーには、デフォルトの [Expression]（true）があります。これにより、すべてのトラフィックが選択され、グローバルにバインドされます。使用する特定のポリシーを念頭に置いておかない限り、これは良い出発点です。
  - 保護**:** このウィザードでは、ハイブリッドセキュリティモデルを利用できます。ハイブリッドセキュリティモデルでは、さまざまな種類のアプリケーションを保護するための豊富な規則を提供する既定の署名を使用できます。シンプルな編集モードでは、さまざまなカテゴリ（CGI、Cold Fusion、PHP など）を表示できます。1 つ以上のカテゴリを選択して、アプリケーションに適用できる特定の規則セットを識別できます。選択したカテゴリのすべてのシグニチャルールを有効にするには、[**Action**] オプションを使用します。セキュリティを強化する前にトラフィックを監視できるように、ブロッキングが無効になっていることを確認します。[続行] をクリックします。[詳細保護の指定] ウィンドウで、必要に応じて変更を加えてセキュリティチェック保護を展開できます。ほとんどの場合、初期セキュリティ構成では基本的な保護で十分です。セキュリティ検査データの代表的なサンプルを収集するために、しばらく

の間トラフィックを実行します。

- セキュリティの強化**:** Web App Firewall をデプロイし、しばらくの間トラフィックを観察した後、緩和をデプロイしてからブロックを有効にすることで、アプリケーションのセキュリティを強化することができます。ラーニング、ビジュアライザー、および クリックしてデプロイルールは、設定を微調整して適切なレベルのリラクゼーションを考え出すことを非常に簡単にできる便利な機能です。この時点で、ポリシー表現を変更したり、追加のポリシーとプロファイルを設定して、さまざまな種類のコンテンツに対して必要なレベルのセキュリティを実装することもできます。
- デバッグ**:** アプリケーションの予期しない動作が見られる場合、Web App Firewall には、簡単にデバッグできるさまざまなオプションが用意されています。
  * ログ。正当な要求がブロックされた場合、まず、ns.log ファイルをチェックして、予期しないセキュリティー検査違反がトリガーされているかどうかを確認します。
  * 機能を無効にします。違反は発生しませんが、アプリケーションがリセットまたは部分的な応答を送信するなど、予期しない動作がまだ発生する場合は、Web App Firewall 機能をデバッグ用に無効にできます。問題が解決しない場合は、Web App Firewall が疑わしいものとして除外されます。
  * ログメッセージでレコードをトレースします。問題が Web App Firewall に関連しているように見え、詳細な検査が必要な場合は、セキュリティ違反メッセージを nstrace に含めることができます。トレースで「Follow TCP stream」を使用すると、ヘッダー、ペイロード、対応するログメッセージなど、個々のトランザクションの詳細を同じ画面に表示できます。この機能の使用方法の詳細については、付録を参照してください。

Citrix Web アプリケーションファイアウォールの概要

October 13, 2021

Citrix Web App Firewall は、セキュリティ違反、データ損失、および機密のビジネス情報や顧客情報にアクセスする Web サイトへの不正な変更の可能性を防ぎます。これは、リクエストとレスポンスの両方をフィルタリングし、悪意のあるアクティビティの証拠がないか調べ、そのようなアクティビティを示すリクエストをブロックすることで行われます。あなたのサイトは、一般的なタイプの攻撃からだけでなく、新しい、まだ未知の攻撃からも保護されています。Web サーバーと Web サイトを不正アクセスから保護することに加えて、Web App Firewall は、レガシー CGI コードまたはスクリプト、Web フレームワーク、Web サーバーソフトウェア、およびその他の基盤となるオペレーティングシステムの脆弱性から保護します。

Citrix Web App Firewall は、スタンドアロンアプライアンスとして、または Citrix ADC 仮想アプライアンス (VPX) の機能として利用できます。Web App Firewall のマニュアルでは、Citrix ADC という用語は、Web App Firewall が実行されているプラットフォームを指します。プラットフォームが専用のファイアウォールアプライアンス、他の機能も構成されている Citrix ADC、Citrix ADC VPX のいずれであっても、Web App Firewall が実行されているプラットフォームを指します。

Web App Firewall を使用するには、保護された Web サイトに設定したルールに違反する接続をブロックするセキ

ュリティ構成を少なくとも 1 つ作成する必要があります。作成する可能性のあるセキュリティ構成の数は、Web サイトの複雑さによって異なります。場合によっては、単一の構成で十分です。その他の場合、特にインタラクティブな Web サイト、データベースサーバーにアクセスする Web サイト、ショッピングカートを備えたオンラインストアなどの場合、特定の種類の攻撃に対して脆弱ではないコンテンツに多大な労力を費やすことなく機密データを最適に保護するために、いくつかの異なる構成が必要になる場合があります。グローバル設定のデフォルトは、すべてのセキュリティ構成に影響するままにしておくことがよくあります。ただし、構成の他の部分と競合する場合や、カスタマイズする場合は、グローバル設定を変更できます。

Web アプリケーションのセキュリティ

Web アプリケーションセキュリティは、HTTP プロトコルと HTTPS プロトコルを使用して通信するコンピュータやプログラムのネットワークセキュリティです。これは、セキュリティの欠陥や弱点がたくさんある広い領域です。サーバーとクライアントの両方のオペレーティングシステムにはセキュリティ上の問題があり、攻撃に対して脆弱です。CGI、Java、JavaScript、PERL、PHP などの Web サーバーソフトウェアと Web サイト対応テクノロジには、根本的な脆弱性があります。Web 対応アプリケーションと通信するブラウザやその他のクライアントアプリケーションにも脆弱性があります。訪問者との対話を可能にするサイトを含む、最も単純な HTML 以外のテクノロジーを使用する Web サイトには、多くの場合、独自の脆弱性があります。

以前は、セキュリティ侵害はしばしば面倒でしたが、今日ではほとんどそうではありません。たとえば、ハッカーが Web サーバーにアクセスし、Web サイトに不正な変更を加えた（改ざんした）攻撃は、以前は一般的でした。彼らは通常、仲間のハッカーに自分のスキルを実証したり、標的を絞った人や会社を恥ずかしい超えて何のモチベーションを持っていたハッカーによって起動されました。しかし、現在のセキュリティ侵害のほとんどは、お金への欲求によって動機づけられています。大多数は、次の目標の 1 つまたは両方を達成しようとします。機密で潜在的に価値のある個人情報を取得すること、または Web サイトまたは Web サーバーへの不正アクセスと制御を取得すること。

特定の形態の Web 攻撃は、個人情報の取得に重点を置いています。これらの攻撃は、攻撃者が完全に制御するのを防ぐのに十分な安全性を備えた Web サイトに対しても可能であることがよくあります。攻撃者が Web サイトから取得できる情報には、顧客の名前、住所、電話番号、社会保障番号、クレジットカード番号、医療記録、およびその他の個人情報が含まれます。攻撃者は、この情報を使用したり、他の人に販売することができます。このような攻撃によって得られる情報の多くは、法律によって保護され、そのすべてが習慣と期待によって保護されています。このタイプの違反は、個人情報が危険にさらされている顧客に深刻な結果をもたらす可能性があります。せいぜい、これらの顧客は、他人が自分のクレジットカードを悪用したり、自分の名前で不正なクレジットアカウントを開設したり、自分の ID を完全に流用したり（個人情報の盗難）しないように警戒する必要があります。最悪の場合、顧客は台無しに信用格付けに直面する可能性があり、あるいは彼らは何の部分もなかった犯罪行為のために非難される。

他の Web 攻撃は、Web サイトまたはそれが動作するサーバー、あるいはその両方の制御を取得する（または 侵害する）ことを目的としています。Web サイトまたはサーバーの制御を取得したハッカーは、それを使用して、不正なコンテンツをホストしたり、別の Web サーバーでホストされたコンテンツのプロキシとして機能したり、SMTP サービスを提供して一方的なバルク電子メールを送信したり、DNS サービスを提供して他の侵害されたコンテンツでそのようなアクティビティをサポートしたりできます Web サーバー。侵害された Web サーバーでホストされているほとんどの Web サイトは、疑わしい、または完全に不正なビジネスを促進しています。たとえば、ほとんどのフィッシング Web サイトと子エクスプロイテーション Web サイトは、侵害された Web サーバーでホストされています。

これらの攻撃から Web サイトと Web サービスを保護するには、識別可能な特性を持つ既知の攻撃をブロックし、未知の攻撃から保護することができる多層防御が必要です。未知の攻撃は、Web サイトと Web サービスへの通常のトラフィックとは異なって見えるために検出されることがよくあります。

既知のウェブ攻撃

Web サイトの最初の防衛線は、存在することがわかっており、Web セキュリティの専門家によって監視および分析されている多数の攻撃に対する保護です。HTML ベースの Web サイトに対する一般的な攻撃の種類は次のとおりです。

- バッファオーバーフロー攻撃。長い URL、長い Cookie、または長い情報を Web サーバーに送信すると、システムがハングしたり、クラッシュしたり、基盤となるオペレーティングシステムへの不正アクセスが提供されたりします。バッファオーバーフロー攻撃は、不正な情報にアクセスしたり、Web サーバにアクセスしたり、またはその両方を侵害したりするために使用できます。
- クッキーのセキュリティ攻撃。通常、偽造された資格情報を使用して不正なコンテンツへのアクセスを取得することを期待して、変更された Cookie を Web サーバーに送信します。
- 強制的なブラウジング。ホームページ上のハイパーリンクのある URL や Web サイト上の他の一般的な開始 URL に移動せずに、Web サイト上の URL に直接アクセスします。強制ブラウジングの個々のインスタンスは、ユーザーが Web サイトのページをブックマークしたことを示している可能性がありますが、存在しないコンテンツ、またはユーザーが直接アクセスしてはならないコンテンツに繰り返しアクセスしようとすると、Web サイトのセキュリティに対する攻撃を表すことがよくあります。強制ブラウジングは通常、不正な情報へのアクセスを得るために使用されますが、サーバーを侵害する試みでバッファオーバーフロー攻撃と組み合わせることもできます。
- **Web** フォームのセキュリティ攻撃。不適切なコンテンツを Web フォームで Web サイトに送信する。不適切なコンテンツには、変更された非表示フィールド、HTML、または英数字データのみを対象とするフィールドのコード、短い文字列のみを受け入れるフィールドの長すぎる文字列、整数のみを受け入れるフィールドの英数字文字列、およびワイドが含まれる可能性がありますあなたのウェブサイトがそのウェブフォームで受け取ることを期待していない他のさまざまなデータ。Web フォームのセキュリティ攻撃は、通常、バッファオーバーフロー攻撃と組み合わせて、Web サイトから不正な情報を取得するか、Web サイトを完全に侵害するために使用できます。

Web フォームセキュリティに対する 2 つの特殊なタイプの攻撃には、特に言及する必要があります。

- **SQL** インジェクション攻撃。SQL データベースに 1 つまたは複数のコマンドを実行させることを目的として、アクティブな SQL コマンドを Web フォームまたは URL の一部として送信します。SQL インジェクション攻撃は、通常、不正な情報を取得するために使用されます。
- クロスサイトスクリプティング攻撃。ウェブページ上の URL またはスクリプトを使用して、同一生成元ポリシーに違反します。これにより、スクリプトが別のウェブサイトのコンテンツからプロパティを取得したり、コンテンツを変更したりすることが禁止されます。スクリプトは Web サイト上の情報を取得してファイルを変更できるため、別の Web サイトのコンテンツへのスクリプトアクセスを許可すると、攻撃者は不正な情報を取得したり、Web サーバーを侵害したりする手段を提供できます。

通常、XML ベースの Web サービスに対する攻撃は、Web サービスへの不適切なコンテンツ送信の試行、または Web サービスに対するセキュリティ侵害の試みの 2 つのカテゴリのうち少なくとも 1 つに分類されます。XML ベースの Web サービスに対する一般的な攻撃には、次のようなものがあります。

- 悪意のあるコードまたはオブジェクト。機密情報を直接取得したり、攻撃者に Web サービスまたは基になるサーバーの制御を与える可能性のあるコードまたはオブジェクトを含む XML リクエスト。
- 不正な形式の **XML** 要求。W3C XML 仕様に準拠していないため、セキュリティで保護されていない Web サービスのセキュリティに違反する XML 要求
- サービス拒否 **(DoS)** 攻撃。対象の Web サービスを圧倒し、正当なユーザが Web サービスへのアクセスを拒否するという目的で、大量に繰り返し送信される XML 要求。

XML ベースの標準的な攻撃に加えて、XML Web サービスおよび Web 2.0 サイトも SQL インジェクション攻撃やクロスサイトスクリプティング攻撃に対して脆弱です。

- **SQL** インジェクション攻撃。SQL データベースにそのコマンドを実行させることを目的として、アクティブな SQL コマンドを XML ベースの要求で送信します。HTML SQL インジェクション攻撃と同様に、XML SQL インジェクション攻撃は通常、不正な情報を取得するために使用されます。
- クロスサイトスクリプティング攻撃。XML ベースのアプリケーションに含まれるスクリプトを使用して、同一オリジンポリシーに違反します。このポリシーでは、スクリプトが別のアプリケーションのコンテンツからプロパティを取得したり、コンテンツを変更したりすることはできません。スクリプトは XML アプリケーションを使用して情報を取得してファイルを変更できるため、別のアプリケーションに属するコンテンツへのスクリプトアクセスを許可すると、不正な情報を取得したり、アプリケーションを侵害したり、またはその両方を行う手段が攻撃者に与えられる可能性があります。

既知の Web 攻撃は通常、特定の攻撃に対して常に表示され、正当なトラフィックに表示されてはならない特定の特性（シグネチャ）に対して Web サイトトラフィックをフィルタリングすることで阻止できます。このアプローチには、比較的少ないリソースを必要とし、偽陽性のリスクが比較的少ないという利点があります。したがって、Web サイトや Web サービスへの攻撃に対抗し、基本的な署名保護を構成するための貴重なツールです。

不明なウェブ攻撃

Web サイトやアプリケーションに対する最大の脅威は、既知の攻撃ではなく、未知の攻撃によるものです。ほとんどの未知の攻撃は、セキュリティ会社がまだ効果的な防御を開発していない新たに開始された攻撃（ゼロデイ攻撃）と、多くの Web サイトまたは Web サービスではなく特定の Web サイトまたは Web サービスに対する慎重に標的化された攻撃（槍攻撃）。これらの攻撃は、既知の攻撃と同様に、機密性の高い個人情報を取得し、Web サイトまたは Web サービスを侵害し、それをさらなる攻撃に使用できるようにすること、またはこれらの両方の目的を目的としています。

ゼロデイ攻撃は、すべてのユーザーにとって大きな脅威です。通常、これらの攻撃は既知の攻撃と同じタイプです。ゼロデイ攻撃には、注入された SQL、クロスサイトスクリプト、クロスサイトリクエストフォージェリ、または既知の攻撃に似た別のタイプの攻撃が含まれます。通常、対象となるソフトウェア、Web サイト、または Web サービスの開発者が気付いていないか、知っている脆弱性を対象としています。したがって、セキュリティ会社はこれらの攻撃に対する防御策を開発しておらず、たとえそうであっても、ユーザーはパッチを入手してインストールしたり、こ

れらの攻撃から保護するために必要な回避策を実行したりしていません。ゼロデイ攻撃の発見から防御の可用性（脆弱性ウィンドウ）までの時間は短縮されていますが、加害者は、多くの Web サイトや Web サービスが攻撃に対する特定の保護を欠いている数時間または数日を数えることができます。

スピア攻撃は大きな脅威ですが、より選択されたユーザーグループにとっては大きな脅威です。一般的なタイプの槍攻撃である槍フィッシュは、特定の銀行や金融機関の顧客、または（あまり一般的ではありませんが）特定の会社や組織の従業員を対象としています。その銀行や金融機関の実際の通信に精通しているユーザーが認識できる、大雑把に書かれた偽造であることが多い他のフィッシュとは異なり、槍フィッシュは文字通り完璧で説得力があります。それらには、一見、見知らぬ人が知っていたり、入手できたりしてはならない、個人に固有の情報を含めることができます。したがって、スピアフィッシング詐欺師は、要求された情報を提供するようにターゲットを説得することができます。この情報を使用して、フィッシング詐欺師はアカウントを略奪したり、他のソースから不正に取得したお金を処理したり、他のさらに機密性の高い情報にアクセスしたりできます。

これらのタイプの攻撃には、標準シグニチャと同様に、特定の特性を探すスタティックパターンを使用することはできませんが、通常は検出できる特定の特性があります。この種の攻撃を検出するには、ヒューリスティックフィルタリングやポジティブなセキュリティモデルシステムなど、より高度でリソース集約的なアプローチが必要です。ヒューリスティックフィルタリングは、特定のパターンではなく、動作のパターンのために見えます。ポジティブセキュリティモデルシステムは、保護している Web サイトまたは Web サービスの通常の動作をモデル化し、通常の使用モデルに適合しない接続をブロックします。URL ベースおよび Web フォームベースのセキュリティチェックは、Web サイトの通常の使用をプロファイルし、ヒューリスティックと確実なセキュリティの両方を使用して異常または予期しないトラフィックをブロックし、ユーザーが Web サイトを操作する方法を制御します。ヒューリスティックとポジティブなセキュリティの両方が、適切に設計および導入され、シグニチャが見逃すほとんどの攻撃を捕捉できます。ただし、シグニチャよりもかなり多くのリソースを必要とするため、誤検知を避けるために適切に設定する時間を費やす必要があります。したがって、これらは主要な防御線としてではなく、署名またはその他のリソースをあまり消費しないアプローチのバックアップとして使用されます。

シグニチャに加えてこれらの高度な保護を構成することで、ハイブリッドセキュリティモデルを作成します。これにより、Web App Firewall は、既知の攻撃と未知の攻撃の両方に対して包括的な保護を提供できます。

Citrix Web アプリケーションファイアウォールの仕組み

Web App Firewall をインストールするときに、ポリシー、プロファイル、およびシグニチャオブジェクトで構成される初期セキュリティ設定を作成します。ポリシーは、フィルタリングするトラフィックを識別する規則であり、トラフィックがフィルタリングされたときに許可またはブロックする動作のパターンとタイプがプロファイルによって識別されます。シグニチャと呼ばれる最も単純なパターンは、プロファイル内ではなく、プロファイルに関連付けられたシグニチャオブジェクト内で指定されます。

シグニチャは、既知の攻撃タイプに一致する文字列またはパターンです。Web App Firewall には、7 つのカテゴリに数千を超えるシグニチャが含まれており、それぞれが特定のタイプの Web サーバーおよび Web コンテンツに対する攻撃を狙っています。新しい脅威が特定されると、Citrix は新しい署名でリストを更新します。設定時に、保護する必要がある Web サーバとコンテンツに適したシグニチャカテゴリを指定します。シグニチャは、処理オーバーヘッドが低く、優れた基本的な保護を提供します。アプリケーションに特殊な脆弱性がある場合、またはシグニチャ

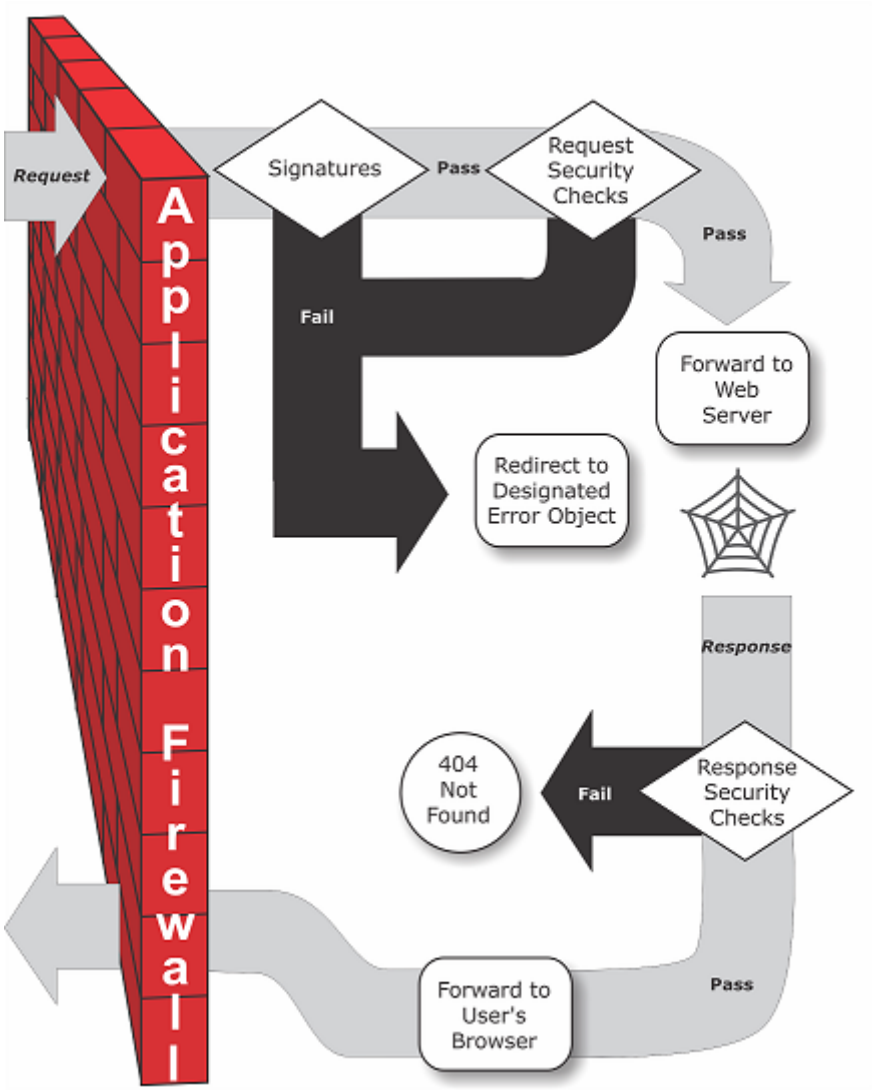
が存在しない攻撃を検出した場合は、独自のシグニチャを追加できます。

より高度な保護をセキュリティチェックと呼びます。セキュリティチェックは、攻撃を示したり、保護された Web サイトや Web サービスに対する脅威を構成したりする可能性のある特定のパターンや動作の種類に対する要求を、より厳密にアルゴリズムで検査することです。たとえば、セキュリティを侵害する可能性のある特定の種類の操作を実行しようとする要求や、社会保障番号やクレジットカード番号などの機密性の高い個人情報を含む応答を識別できます。構成時に、保護する必要がある Web サーバーとコンテンツに適したセキュリティー検査を指定します。セキュリティー検査には制限があります。それらの多くは、適切な例外（緩和）を設定するときに追加しないと、正当な要求と応答をブロックできます。Web サイトの通常の使用を監視し、推奨される例外を作成するアダプティブラーニング機能を使用する場合、必要な例外を特定することは難しくありません。

Web App Firewall は、サーバーとユーザー間のレイヤー 3 ネットワークデバイスまたはレイヤー 2 ネットワークブリッジとしてインストールできます。通常は、会社のルーターまたはファイアウォールの背後にあります。保護する Web サーバと、ユーザがそれらの Web サーバにアクセスするためのハブまたはスイッチ間のトラフィックを傍受できる場所にインストールする必要があります。次に、Web サーバーに直接ではなく Web App Firewall にリクエストを送信し、ユーザーに直接ではなく Web App ファイアウォールに応答するようにネットワークを設定します。Web App Firewall は、内部ルールセットとユーザーの追加と変更の両方を使用して、最終的な宛先に転送する前にトラフィックをフィルタリングします。有害であると検出したアクティビティをブロックまたはレンダリングし、残りのトラフィックを Web サーバに転送します。次の図は、フィルタリングプロセスの概要を示しています。

> 注:
>
> この図は、着信トラフィックへのポリシーの適用を省略しています。このスライドは、ポリシーがすべての要求を処理するセキュリティ設定を示しています。また、この設定では、シグニチャオブジェクトが設定され、プロファイルに関連付けられ、セキュリティチェックが設定されています。

図 1: Web App Firewall フィルタリングフローチャート



図に示すように、ユーザーが保護された Web サイトで URL を要求すると、Web App Firewall は最初にその要求を調べて、署名と一致しないことを確認します。リクエストが署名と一致する場合、Citrix Web Application Firewall はエラーオブジェクト（Web App Firewall アプライアンスにあり、インポート機能を使用して構成できる Web ページ）を表示するか、指定されたエラー URL にリクエストを転送します（エラーページ）。シグニチャにはセキュリティチェックほど多くのリソースが必要ないため、セキュリティチェックを実行する前にシグニチャによって検出された攻撃を検出して停止すると、サーバの負荷が軽減されます。

要求が署名検査に合格すると、Web App Firewall は有効になっている要求セキュリティー検査を適用します。リクエストのセキュリティチェックは、リクエストが Web サイトまたは Web サービスに適切であり、脅威をもたらす可能性のある資料が含まれていないことを確認します。たとえば、セキュリティー検査では、リクエストが予期せぬタイプであるか、予期しないコンテンツをリクエストするか、予期せぬ悪意のある Web フォームデータ、SQL コマンド、またはスクリプトが含まれている可能性があることを示す標識がないか調べます。要求がセキュリティチェックに失敗した場合、Web App Firewall は要求をサニタイズしてから Citrix ADC アプライアンス（または Citrix ADC 仮想アプライアンス）に送り返すか、エラーオブジェクトを表示します。要求がセキュリティチェックに合格すると、

Citrix ADC アプライアンスに返送され、他の処理が完了し、保護された Web サーバーに要求が転送されます。

Web サイトまたは Web サービスがユーザーに応答を送信すると、Web AppFirewall は有効になっている応答セキュリティチェックを適用します。応答セキュリティチェックでは、機密性の高い個人情報の漏洩、Web サイトの改ざんの兆候、または存在してはならないその他のコンテンツがないか応答を調べます。応答がセキュリティチェックに失敗した場合、Web App Firewall は、存在してはならないコンテンツを削除するか、応答をブロックします。応答がセキュリティチェックに合格すると、Citrix ADC アプライアンスに返信され、ユーザーに転送されます。

Citrix Web アプリケーションファイアウォールの機能

Web App Firewall 基本的な機能は、ポリシー、プロファイル、シグニチャです。これらは、既知の Web 攻撃、不明な Web 攻撃、および WebApp Firewall しくみで説明されているハイブリッドセキュリティモデルを提供します。特に注意すべき点は、学習機能です。学習機能は、保護されたアプリケーションへのトラフィックを観察し、特定のセキュリティチェックに対して適切な構成設定を推奨しています。

インポート機能は、Web App Firewall にアップロードするファイルを管理します。これらのファイルは、Web App Firewall によってさまざまなセキュリティチェックで使用され、またはセキュリティチェックに一致する接続に応答するときに使用されます。

ログ、統計、およびレポート機能を使用して、Web App Firewall のパフォーマンスを評価し、より多くの保護が必要になる可能性を特定できます。

Citrix Web アプリケーションファイアウォールによるアプリケーショントラフィックの変更方法

Citrix Web アプリケーションファイアウォールは、保護する Web アプリケーションの動作に影響を与えます。

- Cookies
- HTTP ヘッダー
- フォーム/データ

Citrix Web アプリケーションファイアウォールのセッションクッキー

セッションの状態を維持するために、Citrix ADC Web App Firewall は独自のセッション Cookie を生成します。この Cookie は、Web ブラウザと Citrix ADC Web アプリケーションファイアウォールの間でのみ渡され、Web サーバーには渡されません。いずれかのハッカーがセッション cookie を変更しようとすると、アプリケーションファイアウォールは、要求をサーバーに転送する前に cookie を削除し、要求を新しいユーザーセッションとして扱います。セッションクッキーは、Web ブラウザが開いている限り存在します。Web ブラウザーを閉じると、アプリケーションファイアウォールのセッション cookie が有効になります。セッションの状態は、クライアントがアクセスした URL とフォームの情報を維持します。

構成可能な Web App Firewall セッション Cookie は`citrix_ns_id`です。

Citrix ADC ビルド 12.154 および 13.0 以降では、Cookie の整合性はセッションレスであり、アプライアンスによって生成されたセッション Cookie `citrix_ns_id` の追加を強制しません。

Citrix Web App Firewall クッキー

多くの Web アプリケーションは、ユーザーまたはセッション固有の情報を追跡するために Cookie を生成します。この情報は、ユーザー設定またはショッピングカートアイテムです。Web アプリケーション Cookie は、次の 2 つのタイプのいずれかになります。

- パーシステントクッキー -これらのクッキーはコンピュータにローカルに保存され、次回サイトにアクセスしたときに再び使用されます。通常、このタイプの Cookie には、ログオン、パスワード、設定などのユーザーに関する情報が含まれます。
- セッション **Cookie** または一時 **Cookie-** これらの Cookie はセッション中にのみ使用され、セッションの終了後に破棄されます。このタイプのクッキーには、ショッピングカートアイテムやセッション認証情報などのアプリケーション状態情報が含まれます。

ハッカーは、ユーザーセッションをハイジャックしたり、ユーザーとしてマスカレードするために、アプリケーションクッキーを変更または盗むことができます。アプリケーションファイアウォールは、アプリケーション Cookie をハッシュしてから、デジタル署名を使用して Cookie を追加することにより、このような試みを防ぎます。アプリケーションファイアウォールは、クッキーを追跡することにより、クライアントブラウザとアプリケーションファイアウォールの間でクッキーが変更されたり、侵害されたりしないことを保証します。アプリケーションファイアウォールは、アプリケーション Cookie を変更しません。

Citrix Web アプリケーションファイアウォールは、アプリケーション Cookie を追跡するために、次のデフォルトの Cookie を生成します。

- パーシステント **Cookie**:`citrix_ns_id_wlf`。注：wlf の略は永遠に生きる。
- セッション **Cookie** または一時 **Cookie**:`citrix_ns_id_wat`。注：wat の略は過渡的に動作します。アプリケーション Cookie を追跡するために、アプリケーションファイアウォールは、永続アプリケーション Cookie またはセッションアプリケーション Cookie をまとめてグループ化し、すべての Cookie をまとめてハッシュおよび署名します。したがって、アプリケーションファイアウォールは、すべての永続的なアプリケーション`wlf` Cookie を追跡する 1 つの Cookie を生成し、すべてのアプリケーションセッション`wat` Cookie を追跡する 1 つの Cookie を生成します。

次の表は、Web アプリケーションによって生成された Cookie に基づいて、アプリケーションファイアウォールによって生成された Cookie の数と種類を示しています。

Citrix ADC Web App Firewall 前	変更後は以下の通り
1 つの永続クッキー	永続的な Cookie：`citix_ns_id_wlf`
1 つの一時的な Cookie	一時的な Cookie：`citix_ns_id_wat`
複数の永続的な Cookie、複数の一時的な Cookie	1 つの永続的な Cookie：`citrix_ns_id_wlf`、1 つの一時的な Cookie：`citix_ns_id_wat`

Citrix Web App Firewall では、アプリケーション Cookie を暗号化できます。また、アプリケーションファイアウォールは、アプリケーションによって送信されたセッションクッキーをプロキシするオプションを提供します。これ

は、アプリケーションファイアウォールのセッションデータの残りの部分と一緒に保存し、クライアントに送信しません。クライアントが、アプリケーションファイアウォールのセッション Cookie を含む要求をアプリケーションに送信すると、アプリケーションファイアウォールは、要求を元のアプリケーションに送信する前に、送信された Cookie を要求に挿入し直します。アプリケーションファイアウォールは、HttpOnly および/またはセキュアフラグをクッキーに追加することもできます。

アプリケーションファイアウォールが **HTTP** ヘッダーに与える影響

HTTP 要求と HTTP 応答はどちらも、ヘッダーを使用して 1 つ以上の HTTP のメッセージに関する情報を送信します。ヘッダーは、名前とコロン、スペース、および値を含む各行の一連の行です。たとえば、Host ヘッダーの形式は次のとおりです。

```
Host: www.citrix.com
```

ヘッダーフィールドの中には、リクエストヘッダーとレスポンスヘッダーの両方で使用されるものもあれば、リクエストまたはレスポンスのいずれかにのみ適したものもあります。アプリケーションファイアウォールは、アプリケーションのセキュリティを維持するために、1 つ以上の HTTP 要求または応答の一部のヘッダーを追加、変更、または削除する場合があります。

Citrix Web アプリケーションファイアウォールによって削除された要求ヘッダー

キャッシュに関連するリクエストヘッダーの多くは、セッションのコンテキスト内のすべてのリクエストを表示するために削除されます。同様に、リクエストにエンコードヘッダーが含まれていて、Web サーバーが圧縮応答を送信できるようにする場合、アプリケーションファイアウォールはこのヘッダーを削除するため、圧縮されていないサーバー応答の内容が Web App Firewall によって検査され、クライアントへの機密データの漏洩が防止されます。

アプリケーションファイアウォールは、次の要求ヘッダーを削除します。

- Range：失敗したファイル転送または部分的なファイル転送からのリカバリに使用されます。
- If-Range — クライアントがキャッシュ内にそのオブジェクトの一部がすでに含まれている場合に、部分的なオブジェクトを取得できるようにします (条件付き GET)。
- If-Modified-Sins — 要求されたオブジェクトがこのフィールドに指定された時間以降に変更されていない場合、エンティティはサーバーから返されません。HTTP304 未変更エラーが発生します。
- If-None-Match：最小限のオーバーヘッドで、キャッシュされた情報を効率的に更新できます。
- Accept-Encoding — gzip などの特定のオブジェクトに対して許可されるエンコード方法。

Citrix Web アプリケーションファイアウォールによって変更された要求ヘッダー

Web ブラウザーが HTTP/1.0 以前のプロトコルを使用している場合、ブラウザーは、各応答を受信した後で TCP ソケット接続を継続的に開き、閉じます。これにより、Web サーバーにオーバーヘッドが追加され、セッション状態の維持が妨げられます。ザ・HTTP/1.1 プロトコルにより、セッション中も接続を開いたままにすることができます。アプリケーションファイアウォールは、Web ブラウザで使用されるプロトコルに関係なく、アプリケーションファイア

ウォールと Web サーバー間で HTTP/1.1 を使用するように、次の要求ヘッダーを変更します。
接続: キープアライブ

Citrix Web アプリケーションファイアウォールによって追加された要求ヘッダー

アプリケーションファイアウォールは、リバースプロキシとして機能し、セッションの元の送信元 IP アドレスをアプリケーションファイアウォールの IP アドレスに置き換えます。したがって、Web サーバーログに記録されたすべての要求は、要求がアプリケーションファイアウォールから送信されたことを示します。

Citrix Web アプリケーションファイアウォールによって削除されたレスポンスヘッダー

アプリケーションファイアウォールは、クレジットカード番号の削除やコメントの削除などのコンテンツをブロックまたは変更する場合があり、サイズが一致しない可能性があります。このようなシナリオを防ぐために、アプリケーションファイアウォールは次のヘッダーを削除します。

[Content-Length] — 受信者に送信されるメッセージのサイズを示します。
アプリケーションファイアウォールによって変更された応答ヘッダー

アプリケーションファイアウォールによって変更された応答ヘッダーの多くは、キャッシュに関連しています。HTTP (S) 応答のキャッシュヘッダーを変更して、Web ブラウザーが常に Web サーバーに最新のデータを求める要求を送信し、ローカルキャッシュを使用しないようにする必要があります。ただし、ASP アプリケーションによっては、動的なコンテンツを表示するために個別のプラグインを使用する場合があり、データをブラウザに一時的にキャッシュする機能が必要な場合があります。FFC、URL 閉鎖、CSRF チェックなどの高度なセキュリティ保護が有効になっている場合に、データの一時的なキャッシュを許可するために、アプリケーションファイアウォールは、次のロジックを使用して、サーバー応答のキャッシュ制御ヘッダーを追加または変更します。

- サーバーが Pragma：no-cache を送信した場合、アプリケーションファイアウォールは変更を行いません。
- クライアントリクエストが HTTP1.0 の場合、アプリケーションファイアウォールはプラグマ：no-cache を挿入します。
- クライアント要求が HTTP 1.1 で、キャッシュ制御：非ストアがある場合、アプリケーションファイアウォールは変更を行いません。
- クライアント要求が HTTP 1.1 であり、サーバー応答にストアまたはキャッシュディレクティブがないキャッシュ制御ヘッダーがある場合、アプリケーションファイアウォールは、任意の変更を行いません。
- クライアント要求が HTTP 1.1 で、サーバー応答にキャッシュ制御ヘッダーがないか、キャッシュ制御ヘッダーにストアまたはキャッシュなしディレクティブがない場合、アプリケーションファイアウォールは次のタスクを完了します。

1. キャッシュコントロールを挿入します：最大年齢 = 3、再検証する必要があります、プライベート。
2. Inserts X-Cache-Control-orig = Original value of Cache-Control Header.
3. 最終更新ヘッダーを削除します。
4. ETAG を置き換えます。

5. サーバーによって送信された期限切れヘッダーの X-Expire-Orig= 元の値を挿入します。
6. Expires ヘッダーを変更し、Web ページの有効期限を過去に設定して、常に再度取得されるようにします。
7. 受け入れ範囲を変更し、none に設定します。

アプリケーションファイアウォールが StripComments のように応答を変更したときに、クライアントブラウザに一時的にキャッシュされたデータを置き換えるには、X-out/Remove SafeObject、xout、またはクレジットカードまたは URL 変換の削除、アプリケーションファイアウォールは次のアクションを実行します。

1. クライアントに転送する前に、サーバーから Last-Modified を削除します。
2. ETag は、アプリケーションファイアウォールによって決定された値に置き換えます。

Citrix Web App Firewall によって追加されたレスポンスヘッダー

- `Transfer-Encoding`: チャンクです。このヘッダーは、応答を送信する前に応答の合計長を知ることなく、クライアントに情報をストリーミングします。このヘッダーは、content-length ヘッダーが削除されるために必要です。
- `Set-Cookie`: アプリケーションファイアウォールによって追加されるクッキー。
- `Xet-Cookie`: セッションが有効で、応答がキャッシュで期限切れになっていない場合は、キャッシュからサービスを提供でき、セッションはまだ有効であるため、新しい Cookie を送信する必要はありません。このようなシナリオでは、セット Cookie は Xet-Cookie に変更されます。Web ブラウザ用。

フォームデータの影響

アプリケーションファイアウォールは、サーバーから送信された元のフォームの内容を変更しようとする攻撃から保護します。また、クロスサイトリクエストフォージェリ攻撃から保護することもできます。アプリケーションファイアウォールは、非表示のフォームタグを挿入することで実現します as_fid ページ内。

例: `<input type=”hidden”name=”as_fid”value=”VRgWq0I196Jmg/+LOY7C”/>`

隠しフィールド as_fid は、フィールドの一貫性のために使用されます。このフィールドは、非表示フィールド名と値のペアを含むフォームのすべてのフィールドを追跡し、サーバーから送信されたフォームのフィールドがクライアント側で変更されないようにするために、Application Firewall によって使用されます。CSRF チェックでは、この一意のフォームタグ as_fid を使用して、ユーザーが送信したフォームがこのセッションでユーザーに提供され、ハッカーがユーザーセッションをハイジャックしようとしていないことを確認します。

セッションレスフォームチェック

アプリケーションファイアウォールは、セッションレスフィールドの一貫性を使用してフォームデータを保護するオプションも提供します。これは、フォームに大量の動的非表示フィールドがあり、アプリケーションファイアウォールによるセッションごとのメモリ割り当てが高くなる可能性があるアプリケーションに便利です。セッションレスフィールドの整合性チェックは、構成した設定に基づいて、POST 要求のみ、または GET 要求と POST 要求の両方に対して、別の非表示フィールド as_ffc_field を挿入することによって実行されます。アプリケーションファイアウォールは、フォームをクライアントに転送するときに、メソッド GET を POST に変更します。その後、アプライアン

スはサーバーに送信するときに、メソッドを GET に戻します。as_ffc_field の値には、提供されるフォームの暗号化されたダイジェストが含まれているため、大きくなる可能性があります。セッションレスフォームチェックの例を次に示します。

```
<input type="hidden" name="as_ffc_field" value="CwAAAAVIGLD/
    luRRi1Wu1rbYrFYargEDcO5xVAxsEnMP1megXuQfiDTGbwk0fpgndMHqfMbzfAFdjwR+
    TOm1oT
+u+Svo9+NuloPhtnbkxGtNe7gB/o8GlxEcK9ZkIIVv3oIL/
    nIPSRWJljgpWgafzVx7wtugNwnn8/
    GdnhneLCJTaYU7ScnC6LexJDLisI1xsEeONWt8Zm
+vJTa3mTebDY6LVyhDpDQfBgI1XLgfLTexAUzSNWHYyloqPruGYfnRPw+
    DIGf6gGwn1BYLEsRHKNbjJBrKpOJo9JzhEqdtZ1g3bMzEF9PocPvM1Hpvi5T6VB
/YFunUFM4f+bD7EAVcugdhovzb71CsSQX5+qcC1B8WjQ==" />
<!--NeedCopy-->
```

HTML コメントのストリッピング

アプリケーションファイアウォールは、クライアントに送信する前に、応答内のすべての HTML コメントを削除するためのオプションも提供します。これは、フォームだけでなく、すべての応答ページにも影響します。アプリケーションファイアウォールは、「<!-」と「->」のコメントタグ。タグは、HTML ソースコードのその場所にコメントが存在したことを示すために残ります。他の HTML または JavaScript タグに埋め込まれたテキストは無視されます。一部のアプリケーションは、コメントタグ内に JavaScript が誤って埋め込まれていると、正しく動作しないことがあります。Application Firewall によってコメントが削除される前と後のページのソースコードの比較は、削除されたコメントに必要な JavaScript が埋め込まれているかどうかを識別するのに役立ちます。

クレジットカード保護

アプリケーションファイアウォールは、応答のヘッダーと本文を検査し、応答をクライアントに転送する前にクレジットカード番号を削除または x-out するオプションを提供します。現在、アプリケーションファイアウォールは、次の主要なクレジットカードの保護を提供しています：アメリカン・エキスプレス、ダイナースクラブ、ディスカバー、JCB、マスターカード、およびビザ。x-out アクションは、[ブロック] アクションとは無関係に機能します。

安全な物体保護

クレジットカード番号と同様に、Application Firewall Safe Object セキュリティチェックを使用して、応答内の機密コンテンツを削除または除外することで、他の機密データの漏洩を防ぐこともできます。

クロスサイトスクリプティングはアクションを変換します

トランスフォームでクロスサイトスクリプティングが有効になっている場合、Web AppFirewall はリクエストの ”<” into ”%26lt;” and ”>” into ”%26gt;” を変更します。Web App Firewall の checkRequestHeaders 設定が有効になっている場合、Web App Firewall はリクエストヘッダーを検査し、ヘッダーと Cookie のこれらの文字も変換します。transform アクションは、サーバーから最初に送信された値をブロックまたは変換しません。Web App Firewall で許可されている、クロスサイトスクリプティング用のデフォルトの属性とタグのセットがあります。拒否されたクロスサイトスクリプティングパターンのデフォルトリストも提供されています。これらは、署名オブジェクトを選択して [管理] をクリックすることでカスタマイズできます。SQL/cross-site GUI でのスクリプトパターンダイアログ。

SQL 特殊文字の変換

アプリケーションファイアウォールには、SQL 特殊文字に対する次のデフォルトの変換ルールがあります。

ライセンス	変更後は以下の通り	**Transformation**
’ (single quote that is, %27)	”	Another single quote
\ (backslash that is %5C)	|Another backslash added	
; (semicolon that is %3B)		Dropped

特殊文字の変換が有効で、checkRequestHeaders が ON に設定されている場合、特殊文字の変換はヘッダーと Cookie でも行われます。
注：User-Agent、Accept-Encoding などの一部のリクエストヘッダーにはセミコロンが含まれているため、SQL 変換の影響を受ける可能性があります。

EXPECT ヘッダーが破損する **CitrixWeb** アプリケーションファイアウォールの動作

1. NetScaler が EXPECT ヘッダーを含む HTTP 要求を受信するたびに、NetScaler はバックエンドサーバーに代わってクライアントに EXPECT：100-continue 応答を送信します。
2. この動作は、リクエストをサーバーに転送する前に、アプリケーションファイアウォール保護をリクエスト全体に対して実行する必要があるためです。NetScaler はクライアントからリクエスト全体を取得する必要があります。
3. `100 continue` 応答を受信すると、クライアントは要求を完了する要求の残りの部分を送信します。
4. その後、NetScaler はすべての保護を実行し、要求をサーバーに転送します。
5. NetScaler が完全なリクエストを転送しているため、最初のリクエストに含まれていた EXPECT ヘッダーは廃止され、その結果、NetScaler はこのヘッダーを破損して、サーバーに送信します。
6. 要求を受信したサーバーは、破損しているヘッダーを無視します。

Web App Firewall の設定

October 13, 2021

Citrix Web App Firewall（Web アプリケーションファイアウォール）は、次のいずれかの方法で構成できます。

- **Web App Firewall** ウィザード。設定プロセスを順を追って実行できる一連の画面で構成されるダイアログボックス。
- **Citrix Web** インターフェイス **AppExpert** テンプレート。Web サイトに適切な保護を提供するように設計された AppExpert テンプレート（構成設定のセット）。この AppExpert テンプレートには、多くの Web サイトを保護するための適切な Web AppFirewall 構成設定が含まれています。
- **Citrix ADC** の **GUI** です。Web ベースの設定インターフェイス。
- **Citrix ADC** コマンドラインインターフェイス。コマンドライン設定インターフェイス。

Web App Firewall ウィザードを使用することをお勧めします。ほとんどのユーザーは、Web App Firewall を設定する最も簡単な方法であると感じられ、間違いを防ぐように設計されています。主に Web サイトを保護するために使用する新しい Citrix ADC または VPX がある場合は、Web アプリファイアウォールだけでなくアプライアンス全体に適切なデフォルト構成を提供するため、Web インターフェイス AppExpert テンプレートの方が適している場合があります。。GUI とコマンド・ライン・インタフェースはいずれも、経験豊かなユーザーを対象としています。主に、既存の構成を変更したり、詳細オプションを使用したりします。

Web App Firewall ウィザード

Web App Firewall ウィザードは、簡単な構成の各部分を構成するように求める複数の画面で構成されるダイアログボックスです。Web App Firewall は、指定した情報から適切な構成要素を作成します。これは、最も簡単で、ほとんどの目的で、Web App Firewall を構成するための最良の方法です。

ウィザードを使用するには、任意のブラウザで GUI に接続します。接続が確立したら、Web App Firewall が有効になっていることを確認してから、Web App Firewall ウィザードを実行して、構成情報の入力を求められます。ウィザードを初めて使用するときに、要求された情報をすべて入力する必要はありません。代わりに、デフォルト設定を受け入れ、比較的簡単な構成タスクを実行して重要な機能を有効にし、Web App Firewall が重要な情報を収集して構成を完了できるようにすることができます。

たとえば、処理するトラフィックを選択するための規則を指定するように求められた場合は、デフォルトを受け入れ、すべてのトラフィックが選択されます。シグニチャのリストが表示されたら、適切なカテゴリのシグニチャを有効にして、それらのシグニチャの統計情報の収集を有効にできます。この初期設定では、高度な保護（セキュリティーチェック）をスキップできます。ウィザードは、適切なポリシー、シグニチャオブジェクト、およびプロファイル（まとめてセキュリティ設定）を自動的に作成し、ポリシーをグローバルにバインドします。Web App Firewall は、保護された Web サイトへの接続のフィルタリングを開始し、有効化した 1 つ以上のシグニチャに一致するすべての接続を記録し、各シグニチャが一致する接続に関する統計を収集します。Web App Firewall が一部のトラフィックを処理した後、ウィザードを再度実行し、ログと統計情報を調べて、有効にしたシグニチャのいずれかが正当なトラフィックと一致するかどうかを確認できます。どのシグニチャがブロックするトラフィックを識別しているかを判断した

ら、それらのシグニチャのブロッキングを有効にできます。Web サイトまたは Web サービスが複雑でなく、SQL を使用せず、機密性の高い個人情報にアクセスできない場合、この基本的なセキュリティ構成はおそらく十分な保護を提供します。

たとえば、ウェブサイトが動的な場合は、追加の保護が必要になる場合があります。スクリプトを使用するコンテンツでは、クロスサイトスクリプティング攻撃に対する保護が必要な場合があります。ショッピングカート、多くのブログ、ほとんどのコンテンツ管理システムなどの SQL を使用する Web コンテンツでは、SQL インジェクション攻撃に対する保護が必要になる場合があります。社会保障番号やクレジットカード番号などの機密性の高い個人情報を収集するウェブサイトやウェブサービスでは、意図しない情報漏えいに対する保護が必要になる場合があります。特定の種類の Web サーバーまたは XML サーバーソフトウェアでは、そのソフトウェアに合わせた攻撃から保護する必要があります。もう 1 つの考慮事項は、Web サイトや Web サービスの特定の要素が、他の要素とは異なる保護を必要とする可能性があるということです。Web App Firewall のログと統計情報を調べると、必要な追加の保護を特定するのに役立ちます。

Web サイトおよび Web サービスに必要な高度な保護を決定したら、ウィザードを再度実行してこれらの保護を構成できます。特定のセキュリティチェックでは、チェックで正当なトラフィックがブロックされないように、例外（緩和）を入力する必要があります。手動で行うこともできますが、通常、アダプティブ学習機能を有効にして、必要なリラクゼーションを推奨するほうが簡単です。ウィザードを必要な回数だけ使用して、基本的なセキュリティ構成を強化したり、追加のセキュリティ構成を作成したりできます。

ウィザードを使用すると、ウィザードを使用しなかった場合に手動で実行する必要のあるタスクが自動化されます。ポリシー、シグニチャオブジェクト、およびプロファイルが自動的に作成され、設定名の入力を求められたときに指定した名前が割り当てられます。このウィザードでは、プロファイルに高度な保護設定を追加し、シグニチャオブジェクトをプロファイルにバインドし、プロファイルとポリシーが関連付けられ、ポリシーを Global にバインドしてポリシーを有効にします。

ウィザードでは実行できないタスクがいくつかあります。ウィザードを使用して、グローバル以外のバインドポイントにポリシーをバインドすることはできません。プロファイルを設定の特定の部分にのみ適用する場合は、バインドを手動で設定する必要があります。ウィザードでは、エンジン設定やその他の特定のグローバル構成オプションを構成することはできません。ウィザードでは保護の詳細設定を構成できますが、1 回のセキュリティチェックで特定の設定を変更する場合は、GUI の手動構成画面で変更する方が簡単です。

Web App Firewall ウィザードの使用方法の詳細については、「Web App Firewall ウィザード」を参照してください。

Citrix Web インターフェイス AppExpert テンプレート

AppExpert テンプレートは、複雑なエンタープライズアプリケーションを構成および管理するための、よりシンプルで異なるアプローチです。GUI の AppExpert 表示は、テーブルで構成されています。アプリケーションは左端の列に表示され、そのアプリケーションに適用可能な Citrix ADC 機能が右側の列に表示されます。（AppExpert インターフェイスでは、アプリケーションに関連付けられている機能を「アプリケーション単位」と呼びます）。AppExpert インターフェイスでは、各アプリケーションの対象トラフィックを設定し、各機能を個別に設定する代わりに、圧縮、キャッシュ、書き換え、フィルタリング、レスポンダー、Web App Firewall のルールを有効にします。

Web インターフェイス AppExpert テンプレートには、次の Web App Firewall シグネチャとセキュリティチェックのルールが含まれています。

- **URL** チェックを拒否します。セキュリティ上のリスクがあることがわかっているコンテンツ、または指定した他の URL への接続を検出します。
- バッファオーバーフローチェック。保護された Web サーバでバッファオーバーフローを引き起こす試みを検出します。
- クッキーの一貫性チェック。保護された Web サイトによって設定された Cookie への悪意のある変更を検出します。
- フォームフィールドの一貫性チェック。保護された Web サイト上の Web フォームの構造への変更を検出します。
- **CSRF** フォームタグチェックです。クロスサイトリクエストフォージェリ攻撃を検出します。
- [フィールド形式] チェック。保護された Web サイトの Web フォームにアップロードされた不適切な情報を検出します。
- **HTML SQL** インジェクションチェック。不正な SQL コードを挿入する試みを検出します。
- **HTML** クロスサイトスクリプティングチェック。クロスサイトスクリプティング攻撃を検出します。

AppExpert テンプレートのインストールと使用の詳細については、AppExpert アプリケーションとテンプレートを参照してください。

Citrix GUI

GUI は、Web アプリケーションファイアウォール機能のすべての設定オプションへのアクセスを提供する Web ベースのインターフェイスです。これには、他の設定ツールやインターフェイスからは利用できない高度な設定や管理オプションが含まれます。特に、多くの高度なシグニチャオプションは GUI でのみ設定できます。学習機能によって生成された推奨事項は、GUI でのみ確認できます。GUI でのみグローバル以外のバインドポイントにポリシーをバインドできます。

GUI の説明については、「Web App Firewall 構成インターフェイス」を参照してください。GUI を使用して Web App Firewall を構成する方法の詳細については、GUI を使用した手動設定を参照してください。

GUI を使用して Web App Firewall を設定する手順については、GUI を使用した手動設定を参照してください。citrix-adc GUI の詳細については、「Web App Firewall 構成インターフェイス」を参照してください。

Citrix ADC コマンドラインインターフェイス

Citrix ADC のコマンドラインインターフェイスは、FreeBSD の bash シェルに基づく修正された UNIX シェルです。コマンドラインインターフェイスから Web App Firewall を設定するには、他の UNIX シェルと同様に、プロンプトでコマンドを入力し、Enter キーを押します。NetScaler コマンドラインを使用して、Web App Firewall ほとんどのパラメータとオプションを構成できます。例外はシグニチャ機能であり、その多くのオプションは GUI または Web App Firewall ウィザードを使用してのみ設定でき、その推奨事項は GUI でのみ確認できる学習機能です。

Citrix ADC コマンドラインを使用して Web App Firewall を構成する方法については、「コマンドラインインターフェイスを使用した手動構成」を参照してください。

Citrix Web App Firewall 有効化

October 7, 2021

セキュリティ構成を作成する前に、アプライアンスで Citrix Web App Firewall 機能を有効にする必要があります。

確認事項

- 専用の Citrix Web App Firewall アプライアンスを構成する場合、または既存のアプライアンスをアップグレードする場合、この機能はすでに有効になっています。ここで説明する手順のいずれかを実行する必要はありません。
- 新しい Citrix ADC または VPX を使用している場合は、構成する前に Citrix Web App Firewall 機能を有効にする必要があります。
- 以前のバージョンから Citrix ADC または VPX をアップグレードする場合は、事前に Citrix Web App Firewall 機能を有効にする必要があります。

> 注:
>
> 以前のバージョンから Citrix ADC または VPX をアップグレードする場合は、Citrix Web App Firewall を有効にする前に、アプライアンスのライセンスを更新する必要があります。適切なライセンスを取得するには、Citrix の担当者またはリセラーにお問い合わせください。

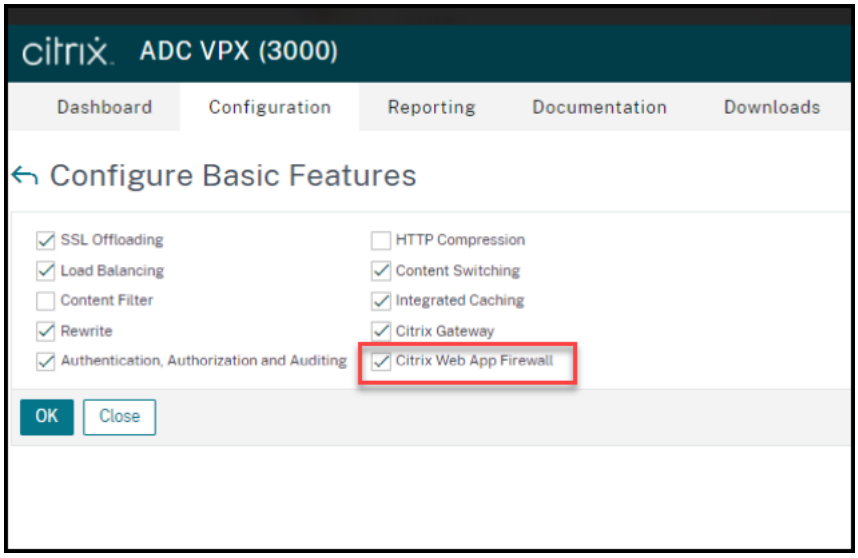
コマンドインターフェイスを使用して **Citrix Web App Firewall** を有効にする

コマンドプロンプトで、次のコマンドを入力します。

```
enable ns feature AppFW
```

GUI を使用して **Web App Firewall** を有効にする

1. [システム] > [設定] に移動します。
2. 詳細ペインで、[高度な機能の構成] をクリックします。
3. ［高度な機能の構成］ページで、**Citrix Web App Firewall** を選択します。
4. ［**OK**］をクリックします。



Web App Firewall ウィザード

October 13, 2021

ほとんどのウィザードとは異なり、Citrix Web App Firewall Wizard は、初期構成プロセスを簡素化するだけでなく、以前に作成した構成を変更し、Web AppFirewall のセットアップを維持するように設計されています。一般的なユーザーは、ウィザードを複数回実行し、毎回一部の画面をスキップします。

Web App Firewall ウィザードは、プロファイル、ポリシー、および署名を自動的に作成します。

ウィザードを開く

Web App Firewall ウィザードを実行するには、GUI を開き、次の手順を実行します。

1. **Security** > **Application Firewall** に移動します。
2. 詳細ウィンドウの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。ウィザードが開きます。

GUI の詳細については、「Web App Firewall 構成インターフェイス」を参照してください。

ウィザードの画面

Web App Firewall ウィザードでは、表形式のページに次の画面が表示されます。

1. [名前の指定]: この画面で、新しいセキュリティ構成を作成するときに、プロファイルの意味のある名前と適切なタイプ (HTML、XML、または WEB 2.0) を指定します。デフォルトのポリシーとシグニチャは、同じ名前を使用して自動生成されます。

プロファイル名

名前は、文字、数字、またはアンダースコア記号で始まり、1 ～31 の文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) の記号で構成されます。新しいセキュリティ構成で保護されているコンテンツを他のユーザーが簡単に識別できるようにする名前を選択します。

> 注:
>
> ウィザードでは、ポリシーとプロファイルの両方にこの名前が使用されるため、最大文字数は 31 文字です。手動で作成されたポリシーの名前は、最大 127 文字です。

既存の構成を変更する場合は、[既存の構成の変更] を選択し、[名前] ドロップダウンリストで、変更する既存の構成の名前を選択します。

> 注:
>
> この一覧には、グローバルまたはバインドポイントにバインドされているポリシーのみが表示されます。アプリケーションファイアウォールウィザードを使用してバインドされていないポリシーを変更することはできません。手動でグローバルまたはバインドポイントにバインドするか、手動で変更する必要があります。(手動で変更する場合は、GUI で) アプリケーションファイアウォール > ポリシー > [ファイアウォール] ペインで、ポリシーを選択し、[開く] をクリックします。

プロファイルの種類

また、この画面でプロファイルの種類を選択します。プロファイルタイプによって、設定可能な高度な保護 (セキュリティチェック) のタイプが決まります。特定の種類のコンテンツは、特定の種類のセキュリティ脅威に対して脆弱ではないため、使用可能なチェックのリストを制限することで、構成中の時間を節約できます。Web App Firewall プロファイルの種類は次のとおりです。

- Web アプリケーション (HTML)。XML または Web2.0 テクノロジーを使用しない HTML ベースの Web サイト。
- XML アプリケーション (XML、SOAP)。任意の XML ベースの Web サービス。
- ウェブ 2.0 アプリケーション (HTML、XML、REST)。ATOM ベースのサイト、ブログ、RSS フィード、Wiki など、HTML および XML ベースのコンテンツを組み合わせた任意の Web 2.0 サイト。

注**:** Web サイトで使用されているコンテンツの種類が不明な場合は、[Web 2.0 アプリケーション] を選択して、すべての種類の Web アプリケーションコンテンツを保護できます。

2. [Specify Rule]: この画面で、現在の設定が検査するトラフィックを定義するポリシールール（式）を指定します。Web サイトおよび Web サービスを保護するための初期設定を作成する場合は、デフォルト値の **true** を受け入れることができます。これにより、すべての Web トラフィックが選択されます。

このセキュリティ設定で、アプライアンスを介してルーティングされるすべての HTTP トラフィックではなく、特定のトラフィックを調べる場合は、調べるトラフィックを指定するポリシールールを記述できます。ルールは、完全に機能するオブジェクト指向プログラミング言語である Citrix ADC 式言語で記述されています。

注：デフォルトの式の構文に加えて、下位互換性のために、Citrix ADC オペレーティングシステムは、Citrix ADC Classic、nCore アプライアンスと仮想アプライアンスでの Citrix ADC クラシック式構文をサポートしています。従

来の式は、Citrix ADC クラスタアプライアンスおよび仮想アプライアンスではサポートされません。既存の構成を Citrix ADC クラスターに移行する現在のユーザーは、従来の式を含むポリシーをデフォルトの式の構文に移行する必要があります。

- Citrix ADC 式の構文を使用して Web App Firewall ルールを作成する簡単な説明と便利なルールのリストについては、「ファイアウォールポリシー」を参照してください。
- Citrix ADC 式の構文でポリシールールを作成する方法の詳細については、「ポリシーと式」を参照してください。

4. 署名を選択: この画面で、Web サイトおよび Web サービスを保護するために使用する署名のカテゴリーを選択します。

これは必須の手順ではありません。必要に応じて、[ディープ保護の指定] 画面に移動できます。[Select Signatures] 画面をスキップすると、プロファイルと関連付けられたポリシーのみが作成され、署名は作成されません。

[新しい署名の作成] または [既存の署名の選択] を選択できます。

新しいセキュリティ設定を作成する場合、選択したシグニチャカテゴリは有効になり、デフォルトでは新しいシグニチャオブジェクトに記録されます。新しいシグニチャオブジェクトには、[名前の指定] 画面で入力した名前と同じ名前が割り当てられます。

シグニチャオブジェクトを設定済みで、作成中のセキュリティ設定に関連付けられたシグニチャオブジェクトとして使用する場合は、[**Select Existing Signature**] をクリックし、[Signatures] リストからシグニチャオブジェクトを選択します。

既存のセキュリティ設定を変更する場合は、[Select Existing Signature] をクリックして、別のシグニチャオブジェクトをセキュリティ設定に割り当てることができます。

[新しい署名の作成] をクリックすると、編集モードとして [簡易] または [詳細] を選択できます。

1. 署名保護の指定（簡易モード）

シンプルモードでは、IIS（インターネットインフォメーションサーバー）、PHP、ActiveX などの一般的なアプリケーションの保護定義のプリセットリストを使用して、署名を簡単に構成できます。シンプルモードのデフォルトのカテゴリは次のとおりです。

- CGI。PERL スクリプト、Unix シェルスクリプト、Python スクリプトなど、任意の言語の CGI スクリプトを使用する Web サイトへの攻撃に対する保護。
- Cold Fusion. AdobeSystems®ColdFusion®Web 開発プラットフォームを使用する Web サイトへの攻撃に対する保護。
- FrontPage. Microsoft®FrontPage®Web 開発プラットフォームを使用する Web サイトへの攻撃に対する保護。
- PHP. PHP オープンソースの Web 開発スクリプト言語を使用する Web サイトへの攻撃に対する保護。
- クライアント側。Microsoft Internet Explorer、Mozilla Firefox、Opera ブラウザ、Adobe AcrobatReader などの保護された Web サイトへのアクセスに使用されるクライアント側ツールへの攻撃に対する保護。

- Microsoft の IIS です。Microsoft Internet Information Server（IIS）を実行している Web サイトへの攻撃に対する保護
- [その他]。Web サーバーやデータベースサーバーなど、他のサーバー側ツールに対する攻撃に対する保護。

この画面では、[署名の選択] 画面で選択した署名カテゴリに関連付けられたアクションを選択します。設定できるアクションは次のとおりです。

- ブロック
- ログ
- 統計情報

デフォルトでは、[Log] アクションと [Stats] アクションは有効ですが、[Block] アクションは無効です。アクションを構成するには、[設定] をクリックします。[アクション] ドロップダウンリストを使用して、選択したすべてのカテゴリのアクション設定を変更できます。
6. 署名保護の指定（詳細モード）

詳細モードでは、シグニチャ定義をより細かく制御でき、さらに多くの情報を提供します。シグニチャ定義を完全に制御する場合は、詳細モードを使用します。

この画面の内容は、「シグニチャオブジェクトの構成または変更」で説明されている「シグニチャオブジェクトの変更」(Modify Signatures Object)ダイアログボックスの内容と同じです。この画面では、[アクション] ドロップダウンリストまたは 3 つのドットが付いた円として表示されるアクションメニューをクリックして、アクションを構成できます。

7. 深い保護の指定：この画面で、Web サイトおよび Web サービスを保護するために使用する高度な保護（セキュリティチェックまたは単にチェックとも呼ばれます）を選択します。使用できるチェックは、[名前の指定] 画面で選択したプロファイルタイプによって異なります。Web 2.0 アプリケーションプロファイルでは、すべてのチェックを使用できます。

詳細については、「セキュリティチェックの概要」および「高度なフォーム保護チェック」を参照してください。

有効化した高度な保護のアクションを設定します。設定できるアクションは次のとおりです。

- Block: 署名に一致する接続をブロックします。デフォルトでは、無効になっています。
- Log: 後で分析できるように、シグニチャと一致する接続をログに記録します。デフォルトで有効。
- Stats: シグニチャごとに、一致した接続数を示す統計情報を保持し、ブロックされた接続のタイプに関するその他の情報を提供します。デフォルトでは、無効になっています。
- 学ぶ。この Web サイトまたは Web サービスへのトラフィックを観察し、このチェックに繰り返し違反する接続を使用して、チェックに推奨される例外またはチェックの新しいルールを生成します。一部のチェックでのみ使用できます。学習機能の詳細については、学習機能の構成と使用、および学習のしくみ、例外（緩和）を設定する方法、またはチェックの学習ルールを展開する方法については、GUI を使用した手動設定を参照してください。

アクションを構成するには、チェックボックスをクリックして保護を選択し、[アクションの設定] をクリックして必要なアクションを選択します。必要に応じて他のパラメータを選択し、[**OK**] をクリックして [アクションの設定] ウィンドウを閉じます。

特定のチェックのすべてのログを表示するには、そのチェックを選択し、[ログ] をクリックして、[Web App Firewall ログ] の説明に従って Syslog Viewer を表示します。セキュリティチェックが保護された Web サイトまたは Web サービスへの正当なアクセスをブロックしている場合は、不要なブロックを示すログを選択し、[展開] をクリックして、そのセキュリティチェックの緩和を作成および実装できます。

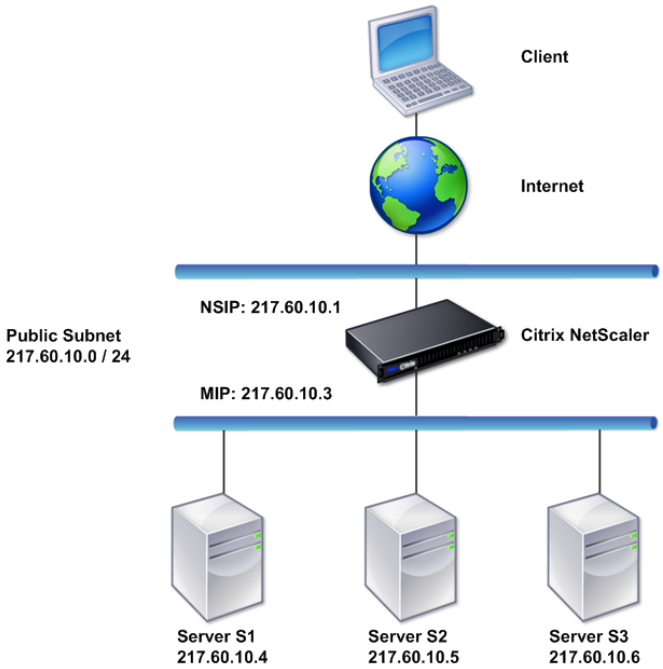
アクション設定の指定が完了したら、[完了] をクリックしてウィザードを完了します。

次に、Web App Firewall ウィザードを使用して特定の種類の構成を実行する方法を示す 4 つの手順を示します。

新しい構成を作成する

Applicaiton Firewall ウィザードを使用して、新しいファイアウォール設定とシグニチャオブジェクトを作成する手順は、次のとおりです。

1. **Security** > **Application Firewall** に移動します。
2. 詳細ペインの [はじめに] で、[** アプリケーションファイアウォール] をクリックします。ウィザードが開きます。



3. [名前の指定] 画面で、[** 新しい構成の作成] を選択します。
4. [名前] フィールドに名前を入力し、[次へ] をクリックします。
5. [ルールの指定] 画面で、もう一度 [次へ] をクリックします。

6. [署名の選択] 画面で、編集モードとして [新しい署名と 簡易] を選択し、[次へ] をクリックします。
7. [署名保護の指定] 画面で、必要な設定を構成します。ブロッキングについて考慮すべきシグニチャと、シグニチャのブロッキングを安全に有効にできるタイミングを決定する方法の詳細については、「シグニチャ」を参照してください。
8. [ディープ保護の指定] 画面で、[アクション 設定] で必要なアクションとパラメータを設定します。
9. 完了したら、[完了] をクリックしてアプリケーションファイアウォールウィザードを閉じます。

既存の構成を変更する

既存の設定および既存のシグニチャカテゴリを変更する手順は、次のとおりです。

1. **Security** > **Application Firewall** に移動します。
2. 詳細ウィンドウの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。ウィザードが開きます。
3. [名前の指定] 画面で [既存の構成の変更] を選択し、[名前] ボックスの一覧で新しい構成中に作成したセキュリティ構成を選択し、[次へ] をクリックします。
4. [ルールの指定] 画面で、[次へ] をクリックしてデフォルト値「true」のままにします。ルールを変更する場合は、「カスタムポリシー式を設定する」で説明されている手順に従います。
5. [署名の選択] 画面で、[既存の署名の選択] をクリックします。[既存の署名] ドロップダウンリストから適切なオプションを選択し、[次へ] をクリックします。署名保護の詳細画面が表示されます。
   注：既存の署名を選択した場合、署名保護のデフォルトの編集モードは拡張されます。
6. [署名保護の指定] 画面で必要な設定を行い、[次へ] をクリックします。ブロッキングについて考慮すべきシグニチャと、シグニチャのブロッキングを安全に有効にできるタイミングを決定する方法の詳細については、「シグニチャ」を参照してください。
7. [ディープ保護の指定] 画面で設定を構成し、[次へ] をクリックします。
8. 完了したら、[完了] をクリックして **Web App Firewall** ウィザードを閉じます。

署名なしで新しい設定を作成する

[Application Firewall Wizard] を使用して [Select Signatures] 画面をスキップし、プロファイルおよび関連付けられたポリシーだけを含み、シグニチャは含まない新しい設定を作成する手順は、次のとおりです。

1. **Security** > **Application Firewall** に移動します。
2. 詳細ウィンドウの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。ウィザードが開きます。
3. [名前の指定] 画面で、[新しい構成の作成] を選択します。
4. [名前] フィールドに名前を入力し、[次へ] をクリックします。
5. 指定規則画面で、再び ネックス t をクリックしてください。
6. [署名の選択] 画面で、[スキップ] をクリックします。
7. [ディープ保護の指定] 画面で、[アクション 設定] で必要なアクションとパラメータを設定します。

8. 完了したら、[完了] をクリックして、アプリケーションファイアウォールウィザードを閉じます。

カスタムポリシー式を設定する

アプリケーションファイアウォールウィザードを使用して、特定のコンテンツのみを保護する特殊なセキュリティ構成を作成するには、次の手順に従います。この場合、初期構成を変更するのではなく、新しいセキュリティ構成を作成します。この種類のセキュリティ構成にはカスタム規則が必要です。そのため、ポリシーでは、選択した Web トラフィックにのみ構成が適用されます。

1. **Security** > **Application Firewall** に移動します。
2. 詳細ウィンドウの [はじめに] で、[アプリケーションファイアウォールウィザード] をクリックします。
3. [名前の指定] 画面の [名前] ボックスに新しいセキュリティ構成の名前を入力し、[種類] ボックスの一覧からセキュリティ構成の種類を選択して、[次へ] をクリックします。
4. [規則の指定] 画面で、この Web アプリケーションで保護するコンテンツのみに一致する規則を入力します。[頻繁に使用するエクスプレッション] ドロップダウンリストと [エクスプレッションエディタ] を使用して、カスタムエクスプレッションを作成します。完了したら、[次へ] をクリックします。
5. [署名の選択] 画面で編集モードを選択し、[次へ] をクリックします。
6. [署名保護の指定] 画面で、必要な設定を構成します。
7. [ディープ保護の指定] 画面で、[アクション 設定] で必要なアクションとパラメータを設定します。
8. 完了したら、[完了] をクリックして アプリケーションファイアウォールウィザードを閉じます。

手動構成

October 7, 2021

Global 以外のバインドポイントにプロファイルをバインドする場合は、バインドを手動で構成する必要があります。また、特定のセキュリティチェックでは、必要な例外を手動で入力するか、学習機能を有効にして Web サイトおよび Web サービスに必要な例外を生成する必要があります。これらのタスクの一部は、Web App Firewall ウィザードを使用して実行できません。

Web App Firewall の動作に慣れており、手動設定を好む場合は、シグニチャオブジェクトとプロファイルを手動で設定し、シグニチャオブジェクトをプロファイルに関連付けて、設定する Web トラフィックに一致するルールでポリシーを作成し、ポリシーを関連付けることができます。をプロファイルとともに使用します。次に、ポリシーをグローバルまたはバインドポイントにバインドして有効にし、完全なセキュリティ構成を作成しました。

手動設定では、GUI（グラフィカルインターフェイス）またはコマンドラインを使用できます。GUI を使用することをお勧めします。すべての構成タスクをコマンドラインで実行できるわけではありません。シグニチャの有効化、学習したデータの確認など、特定のタスクは GUI で実行する必要があります。他のほとんどのタスクは、GUI で実行する方が簡単です。

構成の複製

GUI（GUI）またはコマンドラインインターフェイス（CLI）を使用して Web App Firewall を手動で構成すると、構成は /nsconfig/ns.conf ファイルに保存されます。このファイルのコマンドを使用して、別のアプライアンスに設定を複製できます。コマンドを CLI に 1 つずつ切り取って貼り付けるか、複数のコマンドを /var/tmp フォルダ内のテキストファイルに保存してバッチファイルとして実行することができます。次に、別のアプライアンスの /nsconfig/ns.conf ファイルからコピーされたコマンドを含むバッチファイルの実行例を示します。

```
> batch -f /var/tmp/appfw_add.txt
```

> 警告:
>
> インポートコマンドは、ns.conf ファイルには保存されません。ns.conf ファイルからコマンドを実行して別のアプライアンスに構成をレプリケートする前に、構成で使用されるすべてのオブジェクト（シグニチャ、エラーページ、WSDL、スキーマなど）を、構成をレプリケートするアプライアンスにインポートする必要があります。ns.conf ファイルに保存された Web App Firewall プロファイルを追加するための add コマンドには、インポートされたオブジェクトの名前が含まれる場合がありますが、参照オブジェクトがそのアプライアンス上に存在しない場合、別のアプライアンスで実行するとそのようなコマンドが失敗することがあります。

構成のレプリケーションのインポートまたはエクスポートの詳細については、「シグネチャエクスポート」および「一般的なインポートエクスポートのトピック」を参照してください。

Citrix ADC GUI を使用した手動構成

October 7, 2021

Web App Firewall 機能を手動で構成する必要がある場合は、Citrix ADC GUI 手順を使用することをお勧めします。

シグニチャオブジェクトを作成して設定するには

シグニチャを設定する前に、適切なデフォルトシグニチャオブジェクトテンプレートからシグニチャオブジェクトを作成する必要があります。コピーに新しい名前を割り当て、コピーを構成します。デフォルトシグニチャオブジェクトを直接設定または変更することはできません。次の手順では、シグニチャオブジェクトを設定する基本的な手順を示します。詳細な手順については、シグニチャ機能の手動設定を参照してください。

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ウィンドウで、テンプレートとして使用する署名オブジェクトを選択し、[追加] をクリックします。

   選択肢は次のとおりです:

   - デフォルトシグニチャ。シグニチャルール、SQL インジェクションルール、およびクロスサイトスクリプティングルールが含まれます。
   - **XPath** インジェクション。デフォルトシグニチャのすべての項目が含まれ、さらに XPath インジェクションルールも含まれます。

3. [署名オブジェクトの追加] ダイアログボックスで、新しい署名オブジェクトの名前を入力し、[OK] をクリックし、[閉じる] をクリックします。名前は、文字、数字、またはアンダースコア記号で始まり、1 ～31 文字の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、アンダースコア (_) の各記号で構成されます。
4. 作成したシグニチャオブジェクトを選択し、[開く] をクリックします。
5. **[**署名オブジェクトの修正] ダイアログボックスで、左側の **[**フィルタ条件の表示] オプションを設定して、設定するフィルタ項目を表示します。

   これらのオプションを変更すると、指定した結果が右側の「フィルタ結果」(Filtered Results) ウィンドウに表示されます。シグニチャのカテゴリの詳細については、「署名」を参照してください。
6. [**Filtered Results**] 領域で、該当するチェックボックスをオンまたはオフにして、シグニチャの設定を構成します。
7. 終了したら、[閉じる] をクリックします。

GUI を使用して **Web App Firewall** プロファイルを作成するには

Web App Firewall プロファイルを作成するには、構成の詳細をいくつか指定する必要があります。

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**Web** アプリファイアウォールプロファイルの作成] ダイアログボックスで、プロファイルの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～31 の文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号で構成されます。
4. ドロップダウンリストからプロファイルタイプを選択します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して **Web App Firewall** プロファイルを構成するには

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。
3. [**Web App Firewall** プロファイルの構成] ダイアログボックスの [セキュリティチェック] タブで、セキュリティチェックを構成します。

   - チェックのアクションを有効または無効にするには、リストで、そのアクションのチェックボックスをオンまたはオフにします。

- これらのチェックに他のパラメータを設定するには、一覧でそのチェックの右端にある青い山形をクリックします。表示されるダイアログボックスで、パラメータを設定します。これらは小切手によって異なります。

  チェックボックスをオンにして、ダイアログボックスの下部にある [開く] をクリックして、[リラクゼーションを設定] ダイアログボックスまたは [ルールを設定] ダイアログボックスを表示することもできます。これらのダイアログボックスは、チェックによっても異なります。そのほとんどには、[チェック] タブと [一般] タブがあります。チェックが緩和またはユーザー定義規則をサポートしている場合、「チェック」(Checks) タブには「追加」(Add) ボタンがあります。このボタンでは、チェックのリラクゼーションまたはルールを指定できます。(緩和とは、特定のトラフィックをチェックから除外するための規則です)。緩和がすでに設定されている場合は、緩和を選択し、[開く] をクリックして修正できます。

- 学習した例外またはチェックのルールを確認するには、チェックを選択し、[学習した違反] をクリックします。[学習済みルールの管理] ダイアログボックスで、学習済みの例外またはルールを順に選択します。

  - 例外または規則を編集して一覧に追加するには、[編集と展開] をクリックします。
  - 変更せずに例外またはルールを受け入れるには、「展開」をクリックします。
  - リストから例外または規則を削除するには、[スキップ] をクリックします。

- レビューする例外またはルールのリストをリフレッシュするには、「リ フレッシュ」をクリックします。

- ラーニングビジュアライザーを開き、それを使用して学習したルールを確認するには、[ビジュアライザ] をクリックします。

- チェックに一致する接続のログエントリを確認するには、チェックを選択し、[ログ] をクリックします。この情報を使用して、どのチェックが一致する攻撃であるかを判断し、それらのチェックのブロックを有効にすることができます。この情報を使用して、正当なトラフィックと一致するチェックを特定することもできます。これにより、正当な接続を許可するように適切な免除を構成できます。ログの詳細については、ログ、統計、およびレポートを参照してください。

- チェックを完全に無効にするには、リストで、そのチェックの右側にあるすべてのチェックボックスをオフにします。

4. [設定] タブで、プロファイル設定を構成します。

   - 以前に作成および設定したシグニチャのセットにプロファイルを関連付けるには、**[**共通設定（Common Settings）] で、[Signatures] ドロップダウンリストでそのシグニチャセットを選択します。

     > 注:
     >
     > [共通設定] セクションを表示するには、ダイアログボックスの右側にあるスクロールバーを使用する必要があります。

   - HTML または XML エラーオブジェクトを構成するには、該当するドロップダウンリストからオブジェクトを選択します。

> 注:
>
> 最初に、[Import] ペインで使用するエラーオブジェクトをアップロードする必要があります。

- 既定の XML コンテンツタイプを構成するには、[既定の要求] および [既定の応答] テキストボックスにコンテンツタイプの文字列を直接入力するか、[許可されたコンテンツタイプの管理] をクリックして、許可されたコンテンツタイプのリストを管理します。

5. 学習機能を使用する場合は、[学習] をクリックし、プロファイルの学習設定を構成します。詳細については、「機能の構成と学習機能」を参照してください。
6. 「**OK**」をクリックして変更を保存し、「プロファイル」ペインに戻ります。

Web App Firewall ルールまたは緩和の構成

このダイアログボックスでは、設定するセキュリティチェックに応じて、2 種類の情報を構成します。ほとんどの場合、セキュリティチェックに例外 (またはリラクゼーション) を構成します。[URL の拒否] チェックまたは [フィールド形式] チェックを設定している場合は、追加 (または規則) を設定します。どちらのプロセスも同じです。

Citrix ADC GUI を使用して緩和ルールを構成するには

1. ［セキュリティ］ > **Citrix Web App Firewall** ］ >［プロファイル］ に移動します。
2. [プロファイル] ペインで、構成するプロファイルを選択し、[編集] をクリックします。
3. [**Web App Firewall** プロファイルの構成] ページで、[詳細設定] セクションの [緩和規則] をクリックします。[緩和ルール] セクションには、Web App Firewall 緩和ルールの完全なリストが含まれています。
4. 構成するセキュリティルールをクリックし、[編集] をクリックします。
5. [URL 緩和ルール] ページには、アクションのリストと、このルールに対して構成できる既存の緩和またはルールのリストが含まれています。手動で緩和を追加していない場合、または学習エンジンが推奨する緩和を承認していない場合は、リストが空になる可能性があります。リストの下には、リストの緩和を追加、変更、削除、有効化、無効化できるボタンの行があります。
6. 緩和または規則を追加または変更するには、次のいずれかの操作を行います。
   - 新しい緩和を追加するには、[追加] をクリックします。
   - 既存の緩和を変更するには、変更する緩和を選択し、[開く] をクリックします。

   「**URL** 緩和ルールの開始」ページが表示されます。タイトルを除き、これらのダイアログボックスは同じです。
7. 次の説明に従って、ダイアログボックスに入力します。各チェックのダイアログボックスは異なります。以下のリストは、ダイアログボックスに表示される可能性のあるすべての要素を示しています。
   - [**Enabled**] チェックボックス: この緩和または規則をアクティブに使用する場合に選択します。オフにすると、緩和または規則が非アクティブになります。

- 添付ファイルコンテンツタイプ：XML 添付ファイルのコンテンツタイプ属性。テキスト領域に、許可する XML 添付ファイルの Content-Type 属性と一致する正規表現を入力します。
- [**Action URL**]：テキスト領域に、Web フォームに入力されたデータの配信先となる URL を定義する PCRE 形式の正規表現を入力します。
- [**Cookie**]：テキスト領域に、Cookie を定義する PCRE-形式の正規表現を入力します。
- フィールド名— Web フォームのフィールド名要素には、[フィールド名]、[フォームフィールド]、またはその他の類似の名前のラベルが付けられます。テキスト領域に、フォームフィールドの名前を定義する PCRE-形式の正規表現を入力します。
- **From Origin URL**：テキスト領域で、Web フォームをホストする URL を定義する PCRE 形式の正規表現を入力します。
- **From Action URL**：テキスト領域で、Web フォームに入力されたデータの配信先となる URL を定義する PCRE 形式の正規表現を入力します。
- **[Name]**：XML 要素または属性名。テキスト領域に、要素または属性の名前を定義する PCRE-形式の正規表現を入力します。
- **URL**— URL エレメントには、アクション URL、拒否 URL、フォームアクション URL、フォームのオリジン URL、開始 URL、または単に URL というラベルが付けられます。テキスト領域に、URL を定義する PCRE-形式の正規表現を入力します。
- **[Format**]：[format] セクションには、リストボックスとテキストボックスを含む複数の設定が含まれます。次のいずれかが表示されます。
  - [**Type**]：[Type] ドロップダウンリストでフィールドタイプを選択します。新しいフィールドタイプの定義を追加するには、[管理] をクリックします。
  - **[Minimum Length**]：ユーザに強制的にこのフィールドに入力する場合は、最小長を表す正の整数を入力します。デフォルト：0（フィールドを空白のままにできます）
  - **[Maximum length**]：このフィールドのデータの長さを制限するには、最大長を表す正の整数を文字数で入力します。デフォルト：65535
- [**Location**]：緩和が適用されるリクエストの要素をドロップダウンリストから選択します。HTML セキュリティー検査では、次の選択肢があります。
  - FORMFIELD-Web フォームのフォームフィールド。
  - HEADER—Request headers.
  - クッキー-クッキーヘッダーを設定します。

  XML セキュリティー検査では、次の選択肢があります。
  - エレメント-XML エレメント。
  - 属性：XML 属性。
- 最大添付ファイルサイズ：XML 添付ファイルに許可される最大サイズ（バイト単位）。

- コメント：テキスト領域にコメントを入力します。オプションです。

注：正規表現を必要とするエレメントの場合は、正規表現を入力するか、「正規表現トークン」メニューを使用して正規表現エレメントとシンボルを直接テキストボックスに挿入するか、「正規表現エディタ」をクリックして「正規表現の追加」を開きます。ダイアログボックスを開き、これを使用して式を作成します。

8. 緩和または規則を削除するには、削除する規則を選択し、[削除] をクリックします。
9. 緩和または規則を有効にするには、緩和または規則を選択し、[有効にする] をクリックします。
10. 緩和または規則を無効にするには、緩和または規則を選択し、[無効] をクリックします。
11. 統合された対話型グラフィック表示で、既存のすべての緩和の設定と関係を構成するには、[ビジュアライザー] をクリックし、表示ツールを使用します。

> 注記**:**
>
> 「ビジュアライザー」ボタンは、すべてのチェック緩和ダイアログボックスに表示されるわけではありません。

12. このチェックの学習ルールを確認するには、[学習] をクリックし、学習機能を設定して使用するにはの手順を実行します。
13. [**OK**] をクリックします。

Citrix ADC GUI を使用して学習ルールを構成するには

1. [セキュリティ] > **Citrix Web App Firewall**] > [プロファイル] に移動します。
2. [プロファイル] ペインでプロファイルを選択し、[編集] をクリックします。
3. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定から 学習したルール] をクリックします。[Learning **Rules**] セクションには、現在のプロファイルで使用可能で、学習機能をサポートするセキュリティチェックのリストが表示されます。
4. 学習しきい値を設定するには、セキュリティチェックを選択し、[**Settings**] をクリックします。
5. 「動的プロファイリングおよび学習ルールの設定」ページでは、設定を設定できます。詳細については、「動的プロファイル設定」を参照してください。

   - 最小数のしきい値。構成するセキュリティチェックの学習設定によっては、最小数のしきい値は、監視する必要があるユーザーセッションの合計の最小数、監視する必要がある要求の最小数、または特定のフォームフィールドの最小回数を基準にすることがあります。学習された緩和が生成される前に。デフォルト:1
   - 回数のしきい値の割合。構成するセキュリティチェックの学習設定によっては、しきい値の割合は、セキュリティチェックに違反した監視されたユーザーセッションの合計の割合、要求の割合、またはフォームフィールドが特定のフィールドタイプに一致した回数の割合を、学習された緩和が生成されます。デフォルト:0

6. 学習済みデータをすべて削除し、学習機能をリセットして、最初から再び観測を開始するには、[すべての学習済みデータを削除] アクションを選択します。

   > 注:
   >
   > このボタンは、レビューされていないか、承認されていないか、スキップされた学習済み推奨事項のみを削除します。受け入れられ、展開された学習済み緩和は削除されません。

7. 学習エンジンを特定の IP セットからのトラフィックに制限するには、[**Trusted Learning Client**] をクリックし、使用する IP アドレスをリストに追加します。

   a) 信頼できるラーニングクライアントの一覧に IP アドレスまたは IP アドレスの範囲を追加するには、[追加] をクリックします。
   b) [**AppFirewall** プロファイルから信頼された **Clint** バインディングへ] ページで、[追加] をクリックします。
   c) [有効] チェックボックスをオンにして、機能を有効にします。
   d) [信頼された学習クライアント **] ボックスに、IP アドレスまたは IP アドレスの範囲を CIDR 形式で入力します。
   e) [コメント] テキスト領域に、この IP アドレスまたは範囲を説明するコメントを入力します。
   f) **[**作成**]** して **[**閉じる**]** をクリックします。

8. 既存の IP アドレスまたは範囲を変更するには、IP アドレスまたは範囲をクリックし、[編集] をクリックします。名前を除いて、表示されるダイアログボックスは [信頼できる学習クライアントの追加] ダイアログボックスと同じです。

9. IP アドレスまたは範囲を無効または有効にし、リストに表示したままにするには、IP アドレスまたは範囲をクリックし、必要に応じて [無効] または [有効化] をクリックします。

10. IP アドレスまたは範囲を完全に削除するには、IP アドレスまたは範囲をクリックし、[削除] をクリックします。

11. ［閉じる］をクリックして［**Citrix Web App Firewall** プロファイル］ページに戻ります。

Citrix ADC GUI を使用して **Citrix Web App Firewall** ポリシーを作成するには

1. ［セキュリティ］> ［**Citrix Web App** ファイアウォール］> ［ポリシー］に移動します。
2. ［ポリシー］ページで、**Citrix Web App Firewall** ポリシー］リンクをクリックします。
3. ［Citrix Web App Firewall ポリシー］ページで［追加］をクリックします。
4. ［Citrix Web App Firewall ポリシーの作成］ページで、次のパラメーターを設定します。

   a) Name：名前は、文字、数字、またはアンダースコア記号で始まり、1 ～128 文字の英数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号で構成できます。

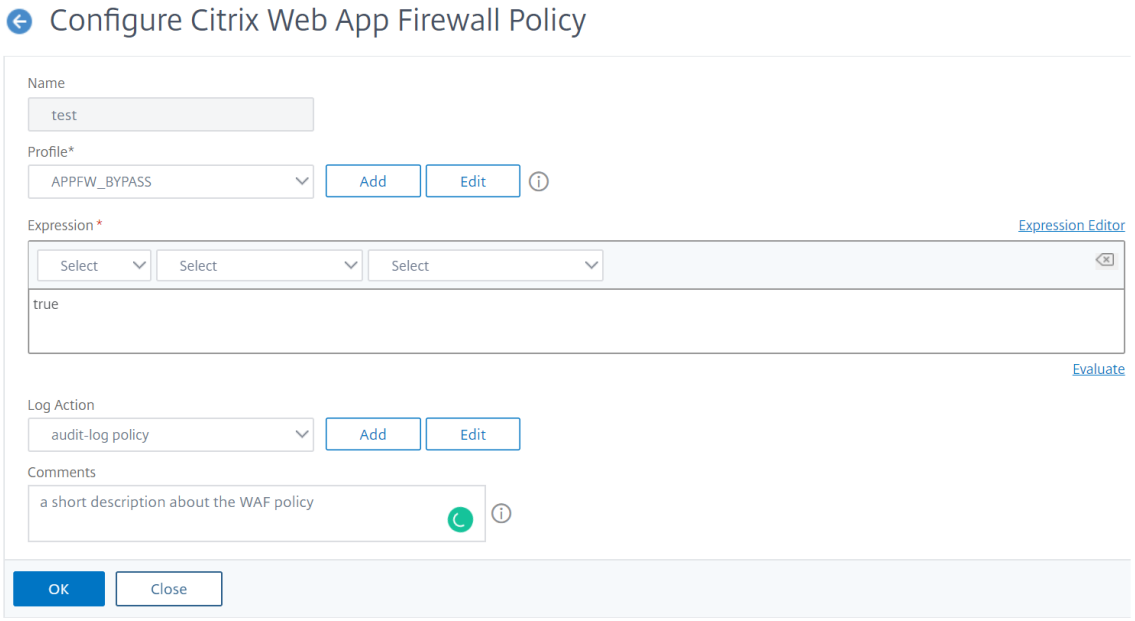
b) [プロファイル]: [Profile] ドロップダウンリストから、このポリシーに関連付けるプロファイルを選択します。[New] をクリックしてポリシーに関連付けるプロファイルを作成し、[Modify] をクリックして既存のプロファイルを 変更できます。

c) 式。[式] テキスト領域で、ポリシーの規則を作成します。

d) ログアクション。ログアクションを追加するか、既存のログアクションを変更できます。

e) コメント。ポリシーに関する簡単な説明。

5. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。



Web App Firewall ルール (式) を作成または構成するには

ポリシールール（式とも呼ばれます）は、ポリシーに関連付けられたプロファイルを使用して Web App Firewall がフィルタリングする Web トラフィックを定義します。他の Citrix ADC ポリシールール（または 式）と同様に、Web App Firewall ルールでは Citrix ADC 式の構文が使用されます。この構文は、強力で、柔軟で、拡張可能です。この一連の命令で完全に記述するには複雑すぎます。次の手順を使用して、単純なファイアウォールポリシールールを作成することも、ポリシー作成プロセスの概要として読み取ることもできます。

1. まだ行っていない場合は、Web AppFirewall ウィザードまたは Citrix ADC GUI の適切な場所に移動して、ポリシールールを作成します。

   - Web App Firewall ウィザードでポリシーを構成する場合は、ナビゲーションペインで ［**Citrix Web App Firewall** ウィザード］ をクリックし、詳細ペインで ［**Citrix Web App Firewall** ウィザード］ をクリックし、［ルールの指定］ タブページに移動します。
   - [規則の指定] ページで、ドロップダウンリストから式の接頭辞を選択します。選択肢は次のとおりです:
   - **HTTP** HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。

- **SYS**。1 つ以上の保護された Web サイトリクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
- **CLIENT**。要求を送信したコンピュータ。リクエストの送信者の一部を調べる場合は、これを選択します。
- **SERVER**。要求が送信されたコンピュータ。要求の受信者のいくつかの側面を調べる場合は、これを選択します。

プレフィックスを選択すると、Web App Firewall に 2 つの部分からなるプロンプトウィンドウが表示されます。このウィンドウには、次の選択肢が上部に表示され、選択した選択肢が意味する内容が下部に表示されます。

2. 次の用語を選択してください。

   プレフィックスとして HTTP を選択した場合、唯一の選択肢は REQ で、要求と応答のペアを指定します。(Web App Firewall は、個別にではなく、1 つのユニットとして要求と応答に対して動作します)。別のプレフィックスを選択した場合、選択肢はより多様になります。特定の選択肢に関するヘルプを表示するには、その選択肢を 1 回クリックすると、下部のプロンプトウィンドウにその選択肢に関する情報が表示されます。

   目的の用語を決定したら、ダブルクリックして [式] ウィンドウに挿入します。

3. 選択した期間の後の期間を入力します。次に、前の手順で説明したように、次の用語を選択するように求められます。用語で値を入力する必要がある場合は、適切な値を入力します。たとえば、HTTP.REQ.HEADER (「」) を選択した場合は、ヘッダー名を引用符で囲んで入力します。

4. 式が終了するまで、プロンプトから用語を選択し、必要な値を入力します。

   次に、特定の目的のための式の例を示します。

   - 特定の **Web** ホスト。特定の Web ホストからのトラフィックを照合するには、次の手順を実行します。

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

shopping.example.com の場合は、照合するウェブホストの名前を置き換えます。

- 特定の **Web** フォルダまたはディレクトリ。Web ホスト上の特定のフォルダまたはディレクトリからのトラフィックを照合するには:

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
```

www.example.com のように、Web ホストの名前を置き換えます。フォルダには、一致させるコンテンツへのフォルダまたはパスを置き換えます。たとえば、ショッピングカートが /solutions/orders というフォルダにある場合は、その文字列をフォルダに置き換えます。

- コンテンツの特定の種類：**GIF** 画像。GIF 形式の画像を一致させるには:

```
HTTP.REQ.URL.ENDSWITH(".png")
```

他の形式の画像と一致させるには、.png の代わりに別の文字列を使用します。

- コンテンツの特定のタイプ：スクリプト。CGI-BIN ディレクトリにあるすべての CGI スクリプトを照合するには:

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
```

.js 拡張子を持つすべての JavaScript を照合するには、次の手順に従います。

```
HTTP.REQ.URL.ENDSWITH(".js")
```

ポリシー式の作成の詳細については、「ポリシーと式」を参照してください。

> 注:
>
> コマンドラインを使用してポリシーを構成する場合は、Citrix ADC 式内の二重引用符をエスケープしてください。たとえば、GUI に入力すると、次の式は正しいです。

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

ただし、コマンドラインで入力する場合は、代わりに次のように入力する必要があります。

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

```
1 ![Policy expression configuration](/en-us/citrix-adc/media/waf-rule.png
    )
```

[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには

[式の追加] ダイアログボックス（式エディターとも呼ばれます）は、Citrix ADC 式言語に慣れていないユーザーに、フィルタリングするトラフィックに一致するポリシーを作成できます。

1. Web App Firewall ウィザードまたは Citrix ADC GUI で適切な場所に移動します（まだ実行していない場合）。
   - **Web App Firewall** ウィザードでポリシーを構成する場合は、ナビゲーションペインで [**Web App Firewall**] をクリックし、詳細ウィンドウで [**Web App Firewall** ウィザード] をクリックし、[ルールの指定] 画面に移動します。
   - ポリシーを手動で構成する場合は、ナビゲーションペインで **[**Web App Firewall**]**、**[**ポリシー**]**、**[**ファイアウォール**]** の順に展開します。詳細ウィンドウで、ポリシーを作成するには、[追加] をクリックします。既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
2. [ルールの指定] 画面の [**Web App Firewall** プロファイルの作成] ダイアログボックスまたは [**Web App** ファイアウォールプロファイルの構成] ダイアログボックスで、[追加] をクリックします。
3. [式の追加] ダイアログボックスの [式の構成] 領域の最初のリストボックスで、次のいずれかの接頭辞を選択します。
   - **HTTP** HTTP プロトコル。HTTP プロトコルに関連するリクエストの側面を調べる場合は、これを選択します。デフォルトの選択肢。

- **SYS**。1 つ以上の保護された Web サイトリクエストの受信者に関連するリクエストの側面を調べる場合は、これを選択します。
- **CLIENT**。要求を送信したコンピュータ。リクエストの送信者の一部を調べる場合は、これを選択します。
- **SERVER**。要求が送信されたコンピュータ。要求の受信者のいくつかの側面を調べる場合は、これを選択します。

4. 2 番目のリストボックスで、次の語句を選択します。使用可能な用語は、前のステップで選択した内容によって異なります。これは、ダイアログボックスでコンテキストに有効な用語のみが含まれるようにリストが自動的に調整されるためです。たとえば、前のリストボックスで [HTTP] を選択した場合、唯一の選択肢は [REQ] です。Web App Firewall はリクエストと関連する応答を単一のユニットとして処理し、両方をフィルタリングするため、個別に応答を指定する必要はありません。2 番目の用語を選択すると、2 番目の用語の右側に 3 番目のリストボックスが表示されます。ヘルプウィンドウには 2 番目の用語の説明が表示され、エクスプレッションのプレビューウィンドウにはエクスプレッションが表示されます。
5. 3 番目のリストボックスで、次の用語を選択します。右側に新しいリストボックスが表示され、ヘルプウィンドウが変わり、新しい用語の説明が表示されます。エクスプレッションのプレビュー (Preview Expression) ウィンドウが更新され、そのポイントまで指定したエクスプレッションが表示されます。
6. 式が完成するまで、用語の選択を続行し、引数の入力を求められたら、引数を入力します。間違えたり、すでに用語を選択した後に式を変更したい場合は、別の用語を選択するだけです。式が変更され、変更した項の後に追加した引数またはそれ以上の項がクリアされます。
7. 式の作成が完了したら、[OK] をクリックして [式の追加] ダイアログボックスを閉じます。式が [式] テキスト領域に挿入されます。

Citrix ADC GUI を使用して **Web App Firewall** ポリシーをバインドするには

1. 次のいずれかを行います:
   - [セキュリティ] > [**Web App Firewall**] に移動し、詳細ペインで [アプリケーションファイアウォールポリシーマネージャ] をクリックします。
   - [セキュリティ] > [**Citrix Web App Firewall**] > [ポリシー] > [ファイアウォール] に移動し、[Citrix Web App Firewall ポリシー] ペインで [ポリシーマネージャ] をクリックします。
2. [アプリケーションファイアウォールポリシーマネージャ] ダイアログボックスで、ポリシーをバインドするバインドポイントをドロップダウンリストから選択します。選択肢は次のとおりです。
   - [グローバルをオーバーライド**]**。このバインドポイントにバインドされたポリシーは、Citrix ADC アプライアンスのすべてのインターフェイスからのすべてのトラフィックを処理し、他のポリシーの前に適用されます。
   - **LB** 仮想サーバー。負荷分散仮想サーバーにバインドされたポリシーは、その負荷分散仮想サーバーによって処理されるトラフィックにのみ適用され、Default Global ポリシーの前に適用されます。[LB Virtual Server] を選択した後、このポリシーをバインドする特定の負荷分散仮想サーバーも選択する必要があります。
   - **CS** 仮想サーバ。コンテンツスイッチ仮想サーバーにバインドされたポリシーは、そのコンテンツスイッチ仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトグローバルポリシーの前

に適用されます。[CS Virtual Server] を選択した後、このポリシーをバインドする特定のコンテンツスイッチ仮想サーバーも選択する必要があります。

- デフォルトグローバル。このバインドポイントにバインドされたポリシーは、Citrix ADC アプライアンスのすべてのインターフェイスからのすべてのトラフィックを処理します。
- ポリシーラベル。ポリシーラベルにバインドされたポリシーは、ポリシーラベルがルーティングするトラフィックを処理します。ポリシーラベルは、このトラフィックにポリシーを適用する順序を制御します。
- なし。ポリシーをバインドポイントにバインドしないでください。

3. ［続行］をクリックします。既存の Web App Firewall ポリシーのリストが表示されます。
4. バインドするポリシーをクリックして選択します。
5. バインドに追加の調整を行います。
   - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。[優先度 を再生成] を選択して、優先度を均等に再番号付けすることもできます。
   - ポリシー式を変更するには、そのフィールドをダブルクリックして [**Web App Firewall Policy** の設定] ダイアログボックスを開き、ポリシー式を編集できます。
   - [Goto Expression] を設定するには、[**Goto Expression**] 列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。
   - [Invoke] オプションを設定するには、[Invoke] 列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。
6. 手順 3 ～6 を繰り返して、グローバルにバインドする Web App Firewall ポリシーを追加します。
7. ［**OK**］をクリックします。ポリシーが正常にバインドされたことを示すメッセージがステータスバーに表示されます。

手動設定コマンドラインインターフェイスを使用する

October 7, 2021

> 注:
>
> Web App Firewall 機能を手動で構成する必要がある場合は、Citrix ADC GUI 手順を使用することをお勧めします。

Web App Firewall 機能は、**Citrix ADC** コマンドインターフェイスから構成できます。ただし、重要な例外があります。コマンドインターフェイスからシグニチャを有効にすることはできません。7 つのカテゴリに約 1,000 個のデフォルトシグニチャがあり、このタスクはコマンドインターフェイスには複雑すぎます。コマンドラインから機能を有効または無効にしたり、パラメータを設定することはできますが、手動リラックスは設定できません。適応型学習機能を設定し、コマンドラインから学習を有効にすることはできますが、学習済み緩和または学習済みルールを確認して承認またはスキップすることはできません。コマンドラインインターフェイスは、Citrix ADC アプライアンスおよび Web App Firewall 使い慣れた上級ユーザーを対象としています。

Citrix ADC コマンドラインを使用して Web App Firewall を手動で構成するには、任意の telnet またはセキュア・

シェル・クライアントを使用して Citrix ADC コマンドラインにログオンします。

コマンドラインインターフェイスを使用してプロファイルを作成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw profile <name> [-defaults ( basic | advanced )]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `save ns config`

例

次の例では、基本デフォルトを使用して pr-basic という名前のプロファイルを追加し、HTML のプロファイルタイプを割り当てます。これは、HTML Web サイトを保護するためのプロファイルの適切な初期設定です。

```
1 add appfw profile pr-basic -defaults basic
2 set appfw profile pr-basic -type HTML
3 save ns config
4 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用してプロファイルを構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> <arg1> [<arg2> ...]`。<arg1>はパラメーターを表し、<arg2>は別のパラメーターか<arg1>のパラメーターに割り当てられる値を表します。特定のセキュリティチェックを構成するときに使用するパラメータの詳細については、高度な保護とそのサブトピックを参照してください。その他のパラメータについては、「縦断を作成するためのパラメータ」を参照してください。
- `save ns config`

例

次の例は、基本デフォルトで作成された HTML プロファイルを設定して、単純な HTML ベースの Web サイトの保護を開始する方法を示しています。この例では、ほとんどのセキュリティチェックで統計情報のロギングとメンテナンスを有効にします。ただし、フォールスポジティブ率が低く、特別な設定を必要としないチェックに対してのみ、ブロッキングを有効にします。また、安全でない HTML や安全でない SQL の変換もオンになります。これにより、攻撃は防止されますが、Web サイトへのリクエストはブロックされません。ロギングと統計を有効にすると、後でログを確認して、特定のセキュリティチェックでブロックを有効にするかどうかを判断できます。


```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
<!--NeedCopy-->
```

ポリシーを作成および構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw policy <name> <rule> <profile>`
- `save ns config`

例

次の例では、pl-blog という名前のポリシーを追加し、ホスト blog.example.com との間で送受信されるすべてのトラフィックを代行受信し、そのポリシーをプロファイル pr-blog に関連付けます。

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
<!--NeedCopy-->
```

Web App Firewall ポリシーをバインドするには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw global <policyName> <priority>`
- `save ns config`

例

次の例では、pl-blog という名前のポリシーをバインドし、プライオリティ 10 を割り当てます。

```
bind appfw global pl-blog 10
save ns config
<!--NeedCopy-->
```

PE ごとのセッション制限を設定するには

コマンドプロンプトで、次のコマンドを入力します。

- set appfw settings <session limit>

例

次に、PE ごとのセッション制限を設定する例を示します。

```
> set appfw settings -sessionLimit 500000`

Done

Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```

署名

October 7, 2021

Web App Firewall シグネチャは、特定の設定可能なルールを提供し、既知の攻撃からウェブサイトを保護するタスクを簡素化します。シグニチャは、オペレーティングシステム、Web サーバー、Web サイト、XML ベースの Web サービス、またはその他のリソースに対する既知の攻撃のコンポーネントであるパターンを表します。事前設定された Web App Firewall の組み込みルールまたはネイティブルールの豊富なセットは、使いやすいセキュリティソリューションを提供し、パターンマッチングのパワーを適用して攻撃を検出し、アプリケーションの脆弱性から保護します。

独自の署名を作成することも、組み込みテンプレートで署名を使用することもできます。Web App Firewall には、次の 2 つの組み込みテンプレートがあります。

- デフォルトシグニチャ：このテンプレートには、SQL インジェクションキーワード、SQL 特殊文字列、SQL 変換ルール、および SQL ワイルドカード文字の完全なリストに加えて、1,300 を超えるシグニチャの事前設定済みリストが含まれています。また、クロスサイトスクリプティングの拒否パターン、クロスサイトスクリプティングの許可された属性とタグも含まれています。これは読み取り専用テンプレートです。コンテンツは表示できますが、このテンプレート内のものは追加、編集、削除できません。それを使用するには、コピーを作成する必要があります。独自のコピーでは、トラフィックに適用するシグニチャルールを有効にし、シグニチャルールがトラフィックと一致したときに実行するアクションを指定できます。

Web App Firewall シグニチャは、Snortによって公開されているルールから得られます。Snort は、さまざまな攻撃やプローブを検出するためのリアルタイムトラフィック分析を実行できるオープンソースの侵入防御システムです。

- ***Xpath** インジェクションパターン: このテンプレートには、事前設定されたリテラルキーワードと PCRE キーワードと、XPath (XML パス言語) インジェクション攻撃を検出するために使用される特別な文字列のセットが含まれています。

空白の署名：組み込みの *Default Signatures テンプレートのコピーを作成する以外に、空白の署名テンプレートを使用して署名オブジェクトを作成できます。空白の署名オプションを使用して作成する署名オブジェクトには、ネイティブの署名ルールはありませんが、*Default テンプレート、それはすべてを持っています SQL/cross-site 組み込みエンティティのスクリプト。

外部フォーマット署名: Web App Firewall は、外部フォーマット署名もサポートしています。Citrix Web App Firewall でサポートされている XSLT ファイルを使用して、サードパーティのスキャンレポートをインポートできます。外部形式ファイルをネイティブ形式に変換する次のスキャンツールには、組み込みの XSLT ファイルのセットが用意されています。

- Cenzic
- Web アプリケーション向けの高度なセキュリティ
- IBM アプリスキャン・エンタープライズ
- IBM アプリスキャンの標準。
- Qualys
- Qualys Cloud
- Whitehat
- Hewlett Packard Enterprise WebInspect
- ラピッド 7 アプリスパイダー
- アクニティックス

アプリケーションのセキュリティ保護

セキュリティが厳しくなると、処理オーバーヘッドが増加します。署名には、アプリケーションの保護を最適化するのに役立つ次の展開オプションがあります。

- ネガティブセキュリティモデル: ネガティブセキュリティモデルでは、事前設定された豊富なシグニチャルールを使用して、パターンマッチングのパワーを適用し、攻撃を検出し、アプリケーションの脆弱性から保護します。あなたは望ましくないものだけをブロックし、残りを許可します。アプリケーションの特定のセキュリ

ティニーズに基づいて、独自のシグネチャルールを追加して、独自のカスタムセキュリティソリューションを設計できます。

- ハイブリッドセキュリティモデル: 署名の使用に加えて、ポジティブセキュリティチェックを使用して、アプリケーションに最適な構成を作成できます。署名を使用して不要なものをブロックし、肯定的なセキュリティチェックを使用して許可されているものを強制します。

署名を使用してアプリケーションを保護するには、署名オブジェクトを使用するように 1 つ以上のプロファイルを設定する必要があります。ハイブリッドセキュリティ構成では、シグネチャオブジェクト内の SQL インジェクションとクロスサイトスクリプティングパターン、および SQL 変換ルールは、シグネチャルールだけでなく、シグネチャオブジェクトを使用している Web App Firewall プロファイルに構成されたポジティブセキュリティチェックによっても使用されます。

Web App Firewall は、保護された Web サイトおよび Web サービスへのトラフィックを調べ、署名と一致するトラフィックを検出します。一致は、ルール内のすべてのパターンがトラフィックに一致する場合にのみトリガーされます。一致が発生すると、ルールに対して指定されたアクションが呼び出されます。リクエストがブロックされると、エラーページまたはエラーオブジェクトを表示できます。ログメッセージは、アプリケーションに対して起動された攻撃を識別するのに役立ちます。統計を有効にすると、Web App ファイアウォールは、Web App Firewall 署名またはセキュリティチェックに一致するリクエストに関するデータを保持します。

トラフィックがシグニチャとポジティブセキュリティチェックの両方に一致する場合、2 つのアクションのうち、より厳しい制限が適用されます。たとえば、ブロックアクションが無効になっているシグネチャルールにリクエストが一致し、アクションがブロックされている SQL Injection ポジティブセキュリティチェックにも一致する場合、リクエストはブロックされます。この場合、署名違反は<not blocked>としてログに記録されますが、要求は SQL インジェクションチェックによってブロックされます。

カスタマイズ: 必要に応じて、独自のルールをシグネチャオブジェクトに追加できます。カスタマイズすることもできます SQL/cross-site スクリプトパターン。アプリケーションの特定のセキュリティニーズに基づいて独自のシグネチャルールを追加するオプションを使用すると、独自のカスタムセキュリティソリューションを柔軟に設計できます。あなたは望ましくないものだけをブロックし、残りを許可します。特定の場所で特定の高速一致パターンを使用すると、処理オーバーヘッドを大幅に削減してパフォーマンスを最適化できます。SQL インジェクションおよびクロスサイトスクリプティングパターンは、追加、変更、または削除できます。組み込みの RegEx エディタと式エディタは、パターンを設定し、その精度を確認するのに役立ちます。

自動更新: 署名オブジェクトを手動で更新して最新の署名ルールを取得するか、Web App Firewall がクラウドベースの Web App Firewall 更新サービスから署名を自動的に更新できるように自動更新機能を適用できます。

> 注:
>
> 自動更新中に新しいシグニチャルールが追加されると、デフォルトで無効になります。更新された署名を定期的に確認し、アプリケーションの保護に関連する新しく追加されたルールを有効にする必要があります。
>
> IIS サーバーで署名をホストするように CORS を構成する必要があります。
>
> Citrix ADC GUI から URL にアクセスすると、ローカルの Web サーバーでは署名の自動更新機能が機能しま

> せん。

はじめに

Citrix 署名を使用してアプリケーションを保護するのは簡単で、いくつかの簡単な手順で実行できます。

1. シグネチャオブジェクトを追加します。

- ウィザードを使用すると、プロファイルおよびポリシーの追加、署名の選択と有効化、署名と肯定的セキュリティチェックに対するアクションの指定など、Web App Firewall 設定全体を作成するように求められます。署名オブジェクトが自動的に作成されます。
- *Default Signatures テンプレートから署名オブジェクトのコピーを作成するか、空のテンプレートを使用して独自のカスタマイズされたルールで署名を作成するか、外部形式署名を追加できます。ルールを有効にし、適用するアクションを設定します。

1. このシグニチャオブジェクトを使用するように、ターゲット Web App Firewall プロファイルを設定します。
2. トラフィックを送信して機能を検証する

ハイライト

- デフォルトシグニチャオブジェクトはテンプレートです。編集や削除はできません。これを使用するには、コピーを作成する必要があります。独自のコピーでは、アプリケーションに必要なルールと各ルールに対して必要なアクションを有効にできます。アプリケーションを保護するには、このシグニチャを使用するようにターゲットプロファイルを設定する必要があります。
- シグニチャパターンの処理にはオーバーヘッドがあります。すべてのシグニチャルールを有効にするのではなく、アプリケーションの保護に適用可能なシグニチャのみを有効にしてください。
- シグニチャの一致をトリガーするには、ルール内のすべてのパターンが一致する必要があります。
- 独自のカスタマイズされたルールを追加して、着信要求を検査し、SQL インジェクション攻撃やクロスサイトスクリプティング攻撃など、さまざまな種類の攻撃を検出できます。また、応答を検査するルールを追加して、クレジットカード番号などの機密情報の漏洩を検出してブロックすることもできます。
- ルールを追加または編集して、既存の署名オブジェクトのコピーを作成し、微調整することができます。SQL/cross-site 別のアプリケーションを保護するためのスクリプトパターン。
- 自動更新を使用すると、Web App Firewall の既定の規則の最新バージョンをダウンロードできます。新しい更新プログラムの可用性を確認するための継続的な監視は必要ありません。
- シグニチャオブジェクトは、複数のプロファイルで使用できます。シグニチャオブジェクトを使用するように 1 つ以上のプロファイルを設定した後でも、シグニチャを有効または無効にしたり、アクション設定を変更したりできます。独自のカスタムシグニチャルールを手動で作成および変更できます。変更は、このシグニチャオブジェクトを使用するように現在設定されているすべてのプロファイルに適用されます。
- HTML、XML、JSON、GWT など、さまざまなタイプのペイロードで違反を検出するようにシグニチャを設定できます。
- 構成済みの署名オブジェクトをエクスポートし、別の Citrix ADC アプライアンスにインポートして、カスタマイズした署名ルールを簡単に複製できます。

シグニチャは、既知の脆弱性に関連付けられているパターンです。シグニチャ保護を使用して、これらの脆弱性を悪用しようとするトラフィックを特定し、特定のアクションを実行できます。

シグニチャはカテゴリに分類されます。アプリケーションの保護に適したカテゴリ内のルールのみを有効にすることで、パフォーマンスを最適化し、処理のオーバーヘッドを減らすことができます。

シグニチャ機能の手動設定

October 7, 2021

署名を使用して Web サイトを保護するには、ルールを確認し、適用するルールを有効にして構成する必要があります。ルールはデフォルトで無効になっています。Web サイトで使用するコンテンツの種類に適用できるすべてのルールを有効にすることをお勧めします。

シグニチャ機能を手動で設定するには、ブラウザを使用して GUI に接続します。次に、組み込みテンプレート、既存の署名オブジェクト、またはファイルをインポートして、署名オブジェクトを作成します。次に、シグニチャオブジェクトの設定または変更の説明に従って、新しいシグニチャオブジェクトを設定します。

署名オブジェクトの追加と削除

April 25, 2022

Web App Firewall に新しい署名オブジェクトを追加するには、次の操作を行います。

- 組み込みテンプレートをコピーする。
- 既存のシグニチャオブジェクトをコピーする。
- 外部ファイルからシグニチャオブジェクトをインポートする。

シグニチャファイルには、CPU 使用率、最新の適用年、重大度の詳細が含まれます。シグニチャファイルが定期的に変更およびアップロードされるたびに、CPU 使用率、最新の年、および CVE 重大度を確認できます。これらの値を確認したら、アプライアンスのシグニチャを有効にするか無効にするかを決定できます。

テンプレートまたは既存のシグニチャオブジェクトをコピーするには、GUI を使用する必要があります。GUI またはコマンドラインを使用して、シグニチャオブジェクトをインポートできます。GUI またはコマンドラインを使用して、シグニチャオブジェクトを削除することもできます。

テンプレートからシグニチャオブジェクトを作成するには

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ウィンドウで、テンプレートとして使用するシグニチャオブジェクトを選択します。

   選択肢は次のとおりです：

- デフォルトのシグニチャ。シグニチャルール、SQL インジェクションルール、クロスサイトスクリプティングルールが含まれます。
- **XPath** インジェクション。XPath インジェクションパターンが含まれます。
- 既存のシグニチャオブジェクト。

> 注意**:**
>
> テンプレートとして使用するシグニチャタイプを選択しない場合、Web App Firewall はシグニチャを最初から作成するように求めるプロンプトを表示します。

3. ［追加］をクリックします。
4. ［シグニチャオブジェクトの追加］ダイアログボックスで、新しいシグニチャオブジェクトの名前を入力し、［OK］をクリックします。名前は、英字、数字、またはアンダースコア記号で始まり、1 ～31 個の英数字、およびハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、およびアンダースコア (_) 記号で構成できます。
5. ［閉じる］をクリックします。

ファイルをインポートしてシグニチャオブジェクトを作成するには

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. シグニチャオブジェクトの追加（**Add Signatures Object** ）ダイアログボックスで、インポートするシグニチャの形式を選択します。
   - Citrix ADC 形式の署名ファイルをインポートするには、［ネイティブ形式］タブを選択します。
   - 外部シグニチャ形式ファイルを読み込むには、[外部形式] タブを選択します。
4. シグニチャオブジェクトの作成に使用するファイルを選択します。
   - ネイティブの Citrix ADC 形式の署名ファイルをインポートするには、［インポート］セクションで［ローカルファイルからインポート］または［URL からインポート］を選択して、ファイルへのパスまたは URL を入力または参照します。
   - Cenzic、IBM AppScan、Qualys、または Whitehat 形式のファイルをインポートするには、「XSLT」セクションで「組み込み XSLT ファイルを使用」、「ローカルファイルを使用」、または「URL から参照」を選択します。次に、[組み込み XSLT ファイルを使用] を選択した場合は、リストから適切なファイル形式を選択します。[ローカルファイルを使用] または [URL から参照] を選択した場合は、ファイルへのパスまたは URL を入力するか、参照して選択します。
5. [追加] をクリックし、[閉じる] をクリックします。

コマンドラインを使用してファイルをインポートしてシグニチャオブジェクトを作成するには

コマンドプロンプトで、次のコマンドを入力します。

- `import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge] [-sha1 <string>]`

- save ns config

例 **#1**

次の例では、signatures.xml という名前のファイルから署名オブジェクトを作成し、MySignatures という名前を割り当てます。

```
import appfw signatures local:signatures.xml MySignatures
save ns config
<!--NeedCopy-->
```

GUI を使用してシグニチャオブジェクトを削除するには

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、削除するシグニチャオブジェクトを選択します。
3. [削除] をクリックします。

コマンドラインを使用してシグニチャオブジェクトを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- rm appfw signatures <name>
- save ns config

シグニチャオブジェクトの設定または変更

October 7, 2021

シグニチャオブジェクトは、作成後に設定するか、既存のシグニチャオブジェクトを変更して、シグニチャカテゴリまたは特定のシグニチャを有効または無効にし、シグニチャが接続に一致したときの Web App Firewall の応答方法を設定します。

シグニチャオブジェクトを設定または変更するには

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ウィンドウで、構成するシグニチャオブジェクトを選択し、[開く] をクリックします。

3. **[**署名オブジェクトの修正] ダイアログボックスで、左側の **[**フィルタ条件の表示] オプションを設定して、設定するフィルタ項目を表示します。

   これらのオプションを変更すると、要求した結果が右側の「フィルタ結果」(Filtered Results) ウィンドウに表示されます。

   - 選択したカテゴリのシグニチャだけを表示するには、該当するシグニチャカテゴリのチェックボックスをオンまたはオフにします。シグニチャカテゴリは次のとおりです。

名前	この署名が保護する攻撃の種類
cgi	CGI スクリプト。Perl および UNIX シェルスクリプトを含む。
クライアント	ブラウザーと他のクライアント。
coldfusion	Adobe SystemsColdFusion アプリケーションサーバーを使用する Web サイト。
frontpage	Microsoft の FrontPage サーバーを使用する Web サイト。
iis か	Microsoft Internet Information Server（IIS）を使用する Web サイト。
misc	さまざまな攻撃。
php	PHP を使用する Web サイト
web-activex	ActiveX コントロールを含む Web サイト。
web-struts	java-ee ベースのアプレットである ApacheStruts を含む Web サイト。

   - 特定のチェックアクションが有効になっているシグニチャのみを表示するには、各アクションの [ON] チェックボックスをオンにし、他のアクションの [ON] チェックボックスをオフにして、すべての [OFF] チェックボックスをオフにします。特定のチェックアクションが無効になったシグネチャのみを表示するには、該当する OFF チェックボックスを選択してすべての ON チェックボックスを選択解除します。チェックアクションが有効か無効かにかかわらずシグネチャを表示するには、該当するアクションの ON と OFF の両方のチェックボックスを選択するか選択解除します。チェックアクションは以下のとおりです:

条件	説明
有効	シグネチャが有効。Web App Firewall はトラフィックを処理するときに有効になっているシグネチャのみをチェックします。

条件	説明
ブロック	この署名に一致する接続はブロックされています。
ログ	この署名に一致するすべての接続で提供されたログエントリ。
統計情報	Web App Firewall は、チェックで生成された統計のこの署名に一致する接続を含みます。

- 特定のストリングを含むシグネチャのみを表示するには、フィルター条件の下のテキストボックスに文字列を入植し、検索をクリックします。
- すべての表示フィルター条件をデフォルト設定にリセットしすべてのシグネチャを表示するには、Show All をクリックします。

4. 特定の署名の詳細については、署名を選択し、[詳細] フィールドの青い二重矢印をクリックします。[署名規則の脆弱性の詳細] メッセージボックスが表示されます。これには、シグニチャの目的に関する情報と、このシグニチャが対処する脆弱性または脆弱性に関する外部の Web ベースの情報へのリンクが含まれています。外部リンクにアクセスするには、そのリンクの説明の左側にある青い二重矢印をクリックします。
5. 適切なチェックボックスをオンにして、署名の設定を構成します。
6. シグニチャオブジェクトにローカルシグニチャルールを追加する場合、または既存のローカルシグニチャルールを変更する場合は、シグネチャエディタ（SignaturesEditor）を参照してください。
7. SQL インジェクション、クロスサイトスクリプト、または Xpath インジェクションパターンが必要ない場合は、[OK] をクリックし、[閉じる] をクリックします。それ以外の場合は、詳細ペインの左下隅にある [管理] をクリックします SQL/cross-site スクリプトパターン。
8. 管理で SQL/cross-site [パターンのスクリプト] ダイアログボックスの [フィルターされた結果] ウィンドウで、構成するパターンのカテゴリとパターンに移動します。SQL インジェクションパターンの詳細については、HTML SQL インジェクションチェックを参照してください。クロスサイトスクリプティングパターンの詳細については、HTML クロスサイトスクリプティングチェックを参照してください。
9. 新しいパターンを追加するには:
   a) 新しいパターンを追加する分岐を選択します。
   b)「フィルタ結果」ウィンドウの下部セクションのすぐ下にある「追加」ボタンをクリックします。
   c) 署名アイテムの作成ダイアログボックスで、追加するパターンを「要素」テキストボックスに入力します。変換規則ブランチに変換パターンを追加する場合は、[要素] の [開始点] テキストボックスに、変更するパターンを入力し、[終了点] テキストボックスに、前のパターンを変更するパターンを入力します。
   d) [**OK**] をクリックします。
10. 既存のパターンを修正するには:
    a)「フィルタ結果」(**Filtered Results**) ウィンドウで、変更するパターンを含むブランチを選択します。

b)「フィルタ結果」(**Filtered Results**) ウィンドウの下にある詳細ウィンドウで、変更するパターンを選択します。

c) [修正**]** をクリックします。

d) **[**署名項目の修正] ダイアログボックスの [要素] テキストボックスで、パターンを変更します。変形パターンを修正する場合は、「エレメント」(Elements) の「始点」(From) および「終点」(To) テキストボックスのいずれかまたは両方のパターンを修正できます。

e) [**OK**] をクリックします。

11. パターンを削除するには、削除するパターンを選択し、[**Filtered Results**] ウィンドウの下の詳細ペインの下にある [**Remove**] ボタンをクリックします。プロンプトが表示されたら、[閉じる] をクリックして選択を確定します。

12. パターンカテゴリをクロスサイトスクリプティングブランチに追加するには:

a) パターンカテゴリを追加するブランチを選択します。

b)「フィルタ結果」ウィンドウのすぐ下にある「追加」ボタンをクリックします。

> 注: 現在、クロスサイトスクリプティングブランチに追加できるのはパターンという名前のカテゴリを 1 つだけなので、[追加] をクリックした後、デフォルトの選択であるパターンを受け入れる必要があります。

c) [**OK**] をクリックします。

13. ブランチを削除するには、そのブランチを選択し、「フィルタ結果」(**Filtered Results**) ウィンドウのすぐ下にある「削除」(Remove) ボタンをクリックします。プロンプトが表示されたら、[**OK]** をクリックして選択を確定します。

> 注記**:** デフォルトブランチを削除すると、そのブランチ内のすべてのパターンが削除されます。これにより、その情報を使用するセキュリティチェックが無効になります。

14. SQL インジェクション、クロスサイトスクリプト、および XPath インジェクションパターンの変更が完了したら、[**OK**] をクリックし、[閉じる] をクリックして [署名オブジェクトの変更] ダイアログボックスに戻ります。

15. [**OK]** をクリックして変更を保存し、シグニチャオブジェクトの設定が完了したら、[**Close**] をクリックします。

署名を使用した **JSON** アプリケーションの保護

October 7, 2021

JavaScript オブジェクト記法 (JSON) は、JavaScript スクリプト言語から派生したテキストベースのオープンスタンダードです。JSON は、単純なデータ構造とオブジェクトと呼ばれる連想配列の人間が読める表現に適していま

す。これは、XML の代替として機能し、主に、Web アプリケーションとの通信のためにシリアライズされたデータ構造を送信するために使用されます。JSON ファイルは通常、.json 拡張子を付けて保存されます。

JSON ペイロードは、通常、**application/json** として指定された MIME タイプで送信されます。JSON の他の「標準」コンテンツタイプは次のとおりです。

- **application/x-javascript**
- **text/javascript**
- **text/x-javascript**
- **text/x-json**

Citrix Web App Firewall 署名を使用した JSON アプリケーションの保護

JSON リクエストを許可するために、アプライアンスは JSON コンテンツタイプで事前設定されています。以下の show-command 出力を参照してください。

```
> sh appfw jsonContentType
1)      JSONContenttypevalue:  ”^application/json$” IsRegex:  REGEX
Done
<!--NeedCopy-->
```

Citrix Web App Firewall は、次のコンテンツタイプについてのみポスト本文を処理します。

- **application/x-www-form-urlencoded**
- **multipart/form-data**
- **text/x-gwt-rpc**

application/json（または他の許可されたコンテンツタイプ）を含む他のコンテンツタイプヘッダーで受信された要求は、ヘッダー検査後にバックエンドに転送されます。このようなリクエストの投稿本文は、SQL やクロスサイトスクリプティングなどのプロファイルのセキュリティチェックが有効になっている場合でも、セキュリティチェック違反がないか検査されません。

JSON アプリケーションを保護し、違反を検出するために、Web App Firewall シグネチャを使用できます。許可されたコンテンツタイプヘッダーを含むすべての要求は、署名の照合のために Web App Firewall によって処理されます。独自にカスタマイズした署名ルールを追加して、JSON ペイロードを処理し、さまざまなセキュリティチェック検査（クロスサイトスクリプティング、SQL、フィールドの整合性など）を実行し、ヘッダーと投稿の本文の違反を検出して、指定されたものを取得できます。行動。

> ヒント
>
> 他の組み込みのデフォルトとは異なり、事前設定された JSON コンテンツタイプは、CLI または GUI（GUI）を使用して編集または削除できます。JSON アプリケーションに対する正当なリクエストがブロックされ、コンテンツタイプ違反が発生する場合は、コンテンツタイプの値が正確に設定されていることを確認します。Web

> App Firewall がコンテンツタイプヘッダーを処理する方法の詳細については、「コンテンツタイプの保護」を参照してください。

コマンドラインインターフェイスを使用して **JSON** コンテンツタイプを追加または削除するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add appfw jsonContentType ^application/json$ IsRegEx REGEX
```

```
rm appfw JSONContentType "^application/json$"
```

GUI を使用して **JSON** コンテンツタイプを管理するには

[セキュリティ] > [**Web App Firewall**] に移動し、[設定] セクションで [**JSON** コンテンツタイプの管理] を選択します。

[**Web App Firewall JSON** コンテンツタイプの設定] パネルで、アプリケーションのニーズに合わせて JSON コンテンツタイプを追加、編集、または削除します。

JSON ペイロードでの攻撃を検出するためのシグニチャ保護の設定

有効な JSON コンテンツタイプに加えて、JSON リクエストで検出された場合にセキュリティ違反を示すパターンを指定するように署名を設定する必要があります。指定されたアクション（block や log など）は、着信要求がシグニチャルールのすべてのターゲットパターンに対して一致をトリガーしたときに実行されます。

カスタマイズしたシグネチャルールを追加するには、GUI を使用することをお勧めします。[システム] > [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。ターゲットシグニチャオブジェクトをダブルクリックして、[**Web App Firewall Signatures**] パネルにアクセスします。[追加] ボタンをクリックして、アクション、カテゴリ、ログ文字列、規則パターンなどを設定します。Web App Firewall は、許可されたすべてのコンテンツタイプペイロードで署名の一致を検査しますが、ルールで JSON 式を指定することで処理を最適化できます。新しいルールパターン を追加する場合は、[**Match**] のドロップダウンオプションで [**Expression**] を選択し、JSON ペイロードからターゲットの一致式を指定して、検査が必要な特定のリクエストを特定します。式は **TEXT** で始まる必要があります。プレフィックス。他のルールパターンを追加して、攻撃を識別するための追加のマッチパターンを指定できます。

次の例は、シグニチャルールを示しています。指定した XPATH_JSON 式に一致するクロスサイトスクリプトタグが JSON ペイロードの POST 本文で検出されると、シグニチャの一致がトリガーされます。

JSON ペイロードでクロスサイトスクリプティングを検出するための署名の例

```
1 <SignatureRule actions="log,stats" category="JSON" enabled="ON" id="
     1000001" severity="" source="" type="" version="1">
2
3   <PatternList>
```



```
4
5      <RequestPatterns>
6
7        <Pattern>
8
9          <Location area="HTTP_POST_BODY"/>
10
11         <Match type="Expression">TEXT.XPATH_JSON(xp%/glossary/title%).
               CONTAINS("example glossary")</Match>
12
13       </Pattern>
14
15       <Pattern>
16
17         <Location area="HTTP_METHOD"/>
18
19         <Match type="LITERAL">POST</Match>
20
21       </Pattern>
22
23       <Pattern>
24
25         <Location area="HTTP_POST_BODY"/>
26
27         <Match type="CrossSiteScripting"/>
28
29        </Pattern>
30
31     </RequestPatterns>
32
33   </PatternList>
34
35   <LogString>Cross-site scripting violation detected in json payload</
        LogString>
36
37   <Comment/>
38
39 </SignatureRule>
40 <!--NeedCopy-->
```

ペイロードの例

次のペイロードは、クロスサイトスクリプティングタグ **<Gotcha!!>** が含まれているため、シグニチャの一致をトリガーします。を選択します。


```
{
 ”glossary”: {
 ”title”: ”example glossary”,”GlossDiv”: {
 ”title”: ”S”,”GlossList”: {
 ”GlossEntry”: {
 ”ID”: ”SGML”,”SortAs”: ”SGML”,”GlossTerm”: ”Standard Generalized
    Markup Language”,”Acronym”: ”SGML”,”Abbrev”: ”ISO 8879:1986”,”
    GlossDef”: {
 ”para”: ”A meta-markup language, used to create markup languages \*\*<
    Gotcha!!>\*\* such as DocBook.”,”GlossSeeAlso”: [”GML”, ”XML”] }
 ,”GlossSee”: ”markup” }
  }
  }
  }
  }

<!--NeedCopy-->
```

ログメッセージの例

```
Aug 21 12:21:42 <local0.info> 10.217.31.239 08/21/2015:23:21:42 GMT ns
    0-PPE-1 : APPFW APPFW_SIGNATURE_MATCH 1471 0 :  10.217.253.62 990-
    PPE0 NtJnVMNnvPeQJnaUzXYW/GTvAQsA010 prof1 http://10.217.31.212/FFC/
    login_post.php Signature violation rule ID 1000001: cross-site
    scripting violation detected in json payload  <not blocked>
<!--NeedCopy-->
```

> 注
>
> クロスサイトスクリプトタグ（**<Gotcha!! >**）、シグネチャルールの一致はトリガーされません。

ハイライト

- JSON ペイロードを保護するには、Web App Firewall シグネチャを使用して、クロスサイトスクリプティング、SQL、およびその他の違反を検出します。
- JSON コンテンツタイプが、許可されたコンテンツタイプとしてアプライアンスで設定されていることを確認します。
- ペイロードのコンテンツタイプが、設定された JSON コンテンツタイプと一致していることを確認します。
- シグニチャルールで設定されたすべてのパターンが、トリガーされるシグニチャ違反と一致していることを確認します。

- 署名ルールを追加するときは、JSON ペイロードの式と一致するルールパターンが少なくとも 1 つ必要です。シグニチャルール内のすべての PI 式は、プレフィクス TEXT. で始まり、ブールである必要があります。

ポリシーと署名を使用して、**SQL** およびクロスサイトスクリプティングでエンコードされたペイロードでアプリケーションまたは **JSON** コンテンツタイプを保護します

Citrix Web App Firewall は、ポリシーと署名を使用して、アプリケーションまたは JSON コンテンツタイプを保護できます。

ポリシーを使用して、アプリケーションまたは **JSON** コンテンツタイプの **SQL** インジェクションを検査する

SQL インジェクションをサポートするために、次のポリシーを追加し、仮想サーバーにグローバルにバインドする必要があります。

```
add appfw policy sqli_1 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A)
|(?<=[^a-zA-Z0-9_])))(select|insert|delete|update|drop|create|alter|grant
|revoke|commit|rollback|shutdown|union|intersect|minus|case|decode|where
|group|begin|join|exists|distinct|add|modify|constraint|null|like|exec|
execute|char|or|and|sp_sdidebug)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK

add appfw policy sqli_2 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A)
|(?<=[^a-zA-Z0-9_]))(xp_availablemedia|xp_cmdshell|xp_deletemail|xp_dirtree
|xp_dropwebtask|xp_dsninfo|xp_enumdsn|xp_enumerrorlogs|xp_enumgroups|
xp_enumqueuedtasks|xp_eventlog|xp_findnextmsg|xp_fixeddrives|xp_getfiledetails
|xp_getnetname|xp_grantlogin|xp_logevent|xp_loginconfig|xp_logininfo|
xp_makewebtask|xp_msver|xp_regread|xp_perfend|xp_perfmonitor|xp_perfsample
|xp_perfstart|xp_readerrorlog|xp_readmail|xp_revokelogin|xp_runwebtask|
xp_schedulersignal|xp_sendmail|xp_servicecontrol|xp_snmp_getstate|xp_snmp_raisetrap
|xp_sprintf|xp_sqlinventory|xp_sqlregister|xp_sqltrace|xp_sscanf|xp_startmail
|xp_stopmail|xp_subdirs|xp_unc_to_drive)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK


add appfw policy sqli_3 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A)
|(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|MSysACEs|MSysObjects|MSysQueries
|MSysRelationships)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK

add appfw policy sqli_4 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A
)|(?<=[^a-zA-Z0-9_]))(SYS\.USER_OBJECTS|SYS\.TAB|SYS\.USER_TABLES|SYS\.
```



```
USER_VIEWS|SYS\.ALL_TABLES|SYS\.USER_TAB_COLUMNS|SYS\.USER_CONSTRAINTS|SYS
\.USER_TRIGGERS|SYS\.USER_CATALOG|SYS\.ALL_CATALOG|SYS\.ALL_CONSTRAINTS|SYS
\.ALL_OBJECTS|SYS\.ALL_TAB_COLUMNS|SYS\.ALL_TAB_PRIVS|SYS\.ALL_TRIGGERS|SYS
\.ALL_USERS|SYS\.ALL_VIEWS|SYS\.USER_ROLE_PRIVS|SYS\.USER_SYS_PRIVS|SYS\.
USER_TAB_PRIVS)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK
```

署名を使用してアプリケーションまたは **JSON** コンテンツタイプを検査する

次のシグニチャルールをアプリケーションファイアウォールプロファイルのシグニチャオブジェクトに追加して、JSON content-type の SQL インジェクションをサポートできます。

> 注:
>
> ポスト本文の署名は CPU を大量に消費します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2013-2018 Citrix Systems, Inc. All rights reserved. -->
<SignaturesFile schema_version="6" version="0" minor_schema_version="0"
    >
     <Signatures>
         <SignatureRule id="4000000" enabled="ON" actions="log,block"
            category="sql" source="" severity="" type="" version="1"
            sourceid="" harmscore="">
             <PatternList>
                 <RequestPatterns>
                     <Pattern>
                         <Location area="HTTP_POST_BODY"/>
                         <Match type="Expression">TEXT.SET_TEXT_MODE(
                            IGNORECASE).SET_TEXT_MODE(URLENCODED).
                            DECODE_USING_TEXT_MODE.REGEX_MATCH(re#(((\A)
                            |(?<=[^a-zA-Z0-9_])))(select|insert|delete|
                            update|drop|create|alter|grant|revoke|commit
                            |rollback|shutdown|union|intersect|minus|
                            case|decode|where|group|begin|join|exists|
                            distinct|add|modify|constraint|null|like|
                            exec|execute|char|or|and|sp_sdidebug)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                     </Pattern>
                     <Pattern type="fastmatch">
                         <Location area="HTTP_METHOD"/>
                         <Match type="LITERAL">T</Match>
                     </Pattern>
                 </RequestPatterns>
```



```
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
        <SignatureRule id="4000001" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
           sourceid="" harmscore="">
            <PatternList>
                <RequestPatterns>
                    <Pattern>
                        <Location area="HTTP_POST_BODY"/>
                        <Match type="Expression">TEXT.SET_TEXT_MODE(
                           IGNORECASE).SET_TEXT_MODE(URLENCODED).
                           DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                           |(?<=[^a-zA-Z0-9_]))(xp_availablemedia|
                           xp_cmdshell|xp_deletemail|xp_dirtree|
                           xp_dropwebtask|xp_dsninfo|xp_enumdsn|
                           xp_enumerrorlogs|xp_enumgroups|
                           xp_enumqueuedtasks|xp_eventlog|
                           xp_findnextmsg|xp_fixeddrives|
                           xp_getfiledetails|xp_getnetname|
                           xp_grantlogin|xp_logevent|xp_loginconfig|
                           xp_logininfo|xp_makewebtask|xp_msver|
                           xp_regread|xp_perfend|xp_perfmonitor|
                           xp_perfsample|xp_perfstart|xp_readerrorlog|
                           xp_readmail|xp_revokelogin|xp_runwebtask|
                           xp_schedulersignal|xp_sendmail|
                           xp_servicecontrol|xp_snmp_getstate|
                           xp_snmp_raisetrap|xp_sprintf|xp_sqlinventory
                           |xp_sqlregister|xp_sqltrace|xp_sscanf|
                           xp_startmail|xp_stopmail|xp_subdirs|
                           xp_unc_to_drive)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                    </Pattern>
                    <Pattern type="fastmatch">
                        <Location area="HTTP_METHOD"/>
                        <Match type="LITERAL">T</Match>
                    </Pattern>
                </RequestPatterns>
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
        <SignatureRule id="4000002" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
```



```
            sourceid="" harmscore="">
40           <PatternList>
41               <RequestPatterns>
42                   <Pattern>
43                       <Location area="HTTP_POST_BODY"/>
44                       <Match type="Expression">TEXT.SET_TEXT_MODE(
                             IGNORECASE).SET_TEXT_MODE(URLENCODED).
                             DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                             |(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|
                             MSysACEs|MSysObjects|MSysQueries|
                             MSysRelationships)((
45 Z)|(?=[^a-zA-Z0-9_]))#)</Match>
46                   </Pattern>
47                   <Pattern type="fastmatch">
48                       <Location area="HTTP_METHOD"/>
49                       <Match type="LITERAL">T</Match>
50                   </Pattern>
51               </RequestPatterns>
52           </PatternList>
53           <LogString>sql Injection</LogString>
54           <Comment/>
55       </SignatureRule>
56       <SignatureRule id="4000003" enabled="ON" actions="log,block"
             category="sql" source="" severity="" type="" version="1"
             sourceid="" harmscore="">
57           <PatternList>
58               <RequestPatterns>
59                   <Pattern>
60                       <Location area="HTTP_POST_BODY"/>
61                       <Match type="Expression">TEXT.SET_TEXT_MODE(
                             IGNORECASE).SET_TEXT_MODE(URLENCODED).
                             DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                             |(?<=[^a-zA-Z0-9_]))(SYS.USER_OBJECTS|SYS.
                             TAB|SYS.USER_TABLES|SYS.USER_VIEWS|SYS.
                             ALL_TABLES|SYS.USER_TAB_COLUMNS|SYS.
                             USER_CONSTRAINTS|SYS.USER_TRIGGERS|SYS.
                             USER_CATALOG|SYS.ALL_CATALOG|SYS.
                             ALL_CONSTRAINTS|SYS.ALL_OBJECTS|SYS.
                             ALL_TAB_COLUMNS|SYS.ALL_TAB_PRIVS|SYS.
                             ALL_TRIGGERS|SYS.ALL_USERS|SYS.ALL_VIEWS|SYS
                             .USER_ROLE_PRIVS|SYS.USER_SYS_PRIVS|SYS.
                             USER_TAB_PRIVS)((
62 Z)|(?=[^a-zA-Z0-9_]))#)</Match>
63                   </Pattern>
64                   <Pattern type="fastmatch">
```



```
65                          <Location area="HTTP_METHOD"/>
66                          <Match type="LITERAL">T</Match>
67                      </Pattern>
68                  </RequestPatterns>
69              </PatternList>
70              <LogString>sql Injection</LogString>
71              <Comment/>
72          </SignatureRule>
73      </Signatures>
74  </SignaturesFile>
75
76  <!--NeedCopy-->
```

シグネチャオブジェクトの更新

October 7, 2021

Web App Firewall が現在の脅威に対する保護を提供していることを確認するには、署名オブジェクトを頻繁に更新する必要があります。デフォルトの WebApp Firewall シグニチャと、サポートされている脆弱性スキャンツールからインポートしたシグニチャの両方を定期的に更新する必要があります。

Citrix は、Web App Firewall デフォルトの署名を定期的に更新します。デフォルトのシグニチャは、手動または自動で更新できます。いずれの場合も、Citrix 担当者または Citrix リセラーに、アップデートにアクセスするための URL を問い合わせてください。[エンジンの設定] および [署名の自動更新設定] ダイアログボックスで、Citrix ネイティブ形式の署名の自動更新を有効にできます。

脆弱性スキャンツールのほとんどのメーカーは、定期的にツールを更新します。ほとんどの Web サイトも頻繁に変更されます。ツールを更新し、Web サイトを定期的に再スキャンして、結果の署名をファイルにエクスポートし、Web AppFirewall 構成にインポートする必要があります。

> ヒント
>
> Citrix ADC コマンドラインから Web App Firewall 署名を更新する場合は、最初にデフォルトの署名を更新してから、さらに更新コマンドを発行して、デフォルトの署名に基づく各カスタム署名ファイルを更新する必要があります。最初にデフォルトシグニチャを更新しないと、バージョンの不一致エラーにより、カスタムシグニチャファイルの更新が妨げられます。
>
> 注
>
> 次は、サードパーティ製のシグニチャオブジェクトとユーザ定義のシグニチャオブジェクトとネイティブルールおよびユーザ定義のシグニチャオブジェクトをマージする場合に適用されます。
>
> バージョン 0 のシグニチャが新しいインポートされたファイルにマージされると、結果のシグニチャはバージ

> ョン 0 のままになります。
>
> つまり、インポートされたファイル内のすべてのネイティブ（または組み込み）ルールは、マージ後に無視されます。これは、バージョン 0 のシグネチャがマージ後もそのまま維持されるようにするためです。
>
> インポートされたファイルにマージするネイティブルールを含めるには、マージの前にバージョン 0 から既存の署名を更新する必要があります。つまり、既存のシグニチャのバージョン 0 の性質を放棄する必要があります。
>
> Citrix ADC リリースアップグレードがある場合、ファイル「default_signatures.xml」が新しいビルドに追加され、ファイル「updated_signature.xml」が古いビルドから削除されます。アップグレード後、署名の自動更新機能が有効になっている場合、アプライアンスは既存の署名をビルドの最新バージョンに更新し、「updated_signature.xml」ファイルを生成します。

コマンドラインを使用して、ソースから **Web App Firewall** 署名を更新するには

コマンドプロンプトで、次のコマンドを入力します。

- `update appfw signatures <name> [-mergedefault]`
- `save ns config`

例

次の例では、デフォルトの署名オブジェクトから MySignatures という名前の署名オブジェクトを更新し、デフォルトの署名オブジェクト内の新しい署名と既存の署名をマージします。このコマンドは、承認された脆弱性スキャンツールなど、他のソースからインポートされたユーザが作成したシグニチャまたはシグニチャを上書きしません。

```
update appfw signatures MySignatures -mergedefault
save ns config
<!--NeedCopy-->
```

Citrix フォーマットファイルからの署名オブジェクトの更新

Citrix は、Web App Firewall 署名を定期的に更新します。Web App Firewall が最新のリストを使用していることを確認するには、Web AppFirewall の署名を定期的に更新する必要があります。アップデートにアクセスするための URL については、Citrix 担当者または Citrix リセラーにお問い合わせください。

コマンドラインを使用して **Citrix** 形式のファイルから署名オブジェクトを更新するには

コマンドプロンプトで、次のコマンドを入力します。

- `update appfw signatures <name> [-mergeDefault]`

- `save ns config`

GUI を使用して **Citrix** 形式のファイルからシグニチャオブジェクトを更新するには

1. [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。
2. 詳細ウィンドウで、更新するシグニチャオブジェクトを選択します。
3.「アクション」ドロップダウンリストで、「マージ」を選択します。
4. 署名オブジェクトの更新ダイアログボックスで、次のいずれかのオプションを選択します。
   - **[Import from URL]**: Web URL からシグニチャアップデートをダウンロードする場合は、このオプションを選択します。
   - **Import from Local File**: ローカルハードドライブ、ネットワークハードドライブ、またはその他のストレージデバイス上のファイルからシグニチャアップデートをインポートする場合は、このオプションを選択します。
5. テキスト領域で URL を入力するか、ローカルファイルを参照します。
6. [更新] をクリックします。更新ファイルがインポートされ、[署名の更新] ダイアログボックスが、[署名オブジェクトの修正**]** ダイアログボックスの形式とほぼ同じ形式に変わります。[**Update Signatures Object**] ダイアログボックスには、新規または変更されたシグニチャ規則、SQL インジェクションまたはクロスサイトスクリプティングパターン、および XPath インジェクションパターン（存在する場合）を含むすべてのブランチが表示されます。
7. 新しいシグニチャと変更されたシグニチャを確認して設定します。
8. 完了したら、[**OK**] をクリックし、[閉じる] をクリックします。

サポートされている脆弱性スキャンツールからのシグニチャオブジェクトの更新

> 注:
>
> ファイルからシグニチャオブジェクトを更新する前に、脆弱性スキャンツールからシグニチャをエクスポートしてファイルを作成する必要があります。

脆弱性スキャンツールからシグニチャをインポートおよび更新するには

1. [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。
2. 詳細ウィンドウで、更新する署名オブジェクトを選択し、[マージ] をクリックします。
3. [署名オブジェクトの更新] ダイアログボックスの [外部形式] タブの [インポート] セクションで、次のいずれかのオプションを選択します。
   - **[Import from URL]**: Web URL からシグニチャアップデートをダウンロードする場合は、このオプションを選択します。
   - **Import from Local File**: ローカル、ネットワークハードドライブ、またはその他のストレージデバイス上のファイルからシグニチャアップデートをインポートする場合は、このオプションを選択します。
4. テキスト領域で、URL を入力するか、ローカルファイルへのパスを参照または入力します。
5. [XSLT] セクションで、次のいずれかのオプションを選択します。

- [組み込み **XSLT** ファイルを使用]: 組み込み XSLT ファイルを使用する場合は、このオプションを選択します。
- **[**ローカル **XSLT** ファイルを使用]: ローカルコンピュータ上の XSLT ファイルを使用する場合は、このオプションを選択します。
- [**URL** から **XSLT** を参照]: Web URL から XSLT ファイルをインポートするには、このオプションを選択します。

6. 「組み込み XSLT ファイルを使用」を選択した場合は、「組み込み XSLT」ドロップダウンリストで、使用するファイルを次のオプションから選択します。
   - **Cenzic.**
   - **Deep_Security_for_Web_Apps.**
   - **Hewlett_Packard_Enterprise_WebInspect.**
   - **IBM-AppScan-Enterprise.**
   - **IBM-AppScan-Standard.**
   - **Qualys.**
   - **Whitehat.**
7. ［更新］をクリックします。更新ファイルがインポートされ、[シグニチャの更新] ダイアログボックスは、「シグニチャオブジェクトの構成と変更」で説明されている「シグニチャオブジェクトの変更」ダイアログボックスとほぼ同じ形式に変更されます。[**Update Signatures Object**] ダイアログボックスには、新規または変更されたシグニチャ規則、SQL インジェクションまたはクロスサイトスクリプティングパターン、および XPath インジェクションパターン（存在する場合）を含むすべてのブランチが表示されます。
8. 新しいシグニチャと変更されたシグニチャを確認して設定します。
9. 完了したら、[**OK**] をクリックし、[閉じる] をクリックします。

署名の自動更新

October 7, 2021

Web アプリケーションファイアウォールの Signature Auto Update 機能を使用すると、ユーザーは最新のシグニチャを取得して、新しい脆弱性から Web アプリケーションを保護できます。自動更新機能は、最新の更新プログラムを取得するための継続的な手動介入を必要とせずに、より優れた保護を提供します。

シグニチャは 1 時間ごとに自動更新され、最新の更新の可用性を定期的にチェックする必要はありません。署名自動更新を有効にすると、Citrix ADC アプライアンスは署名をホストしているサーバーに接続して、新しいバージョンが利用可能かどうかを確認します。

カスタマイズ可能な場所

最新のアプリケーションファイアウォールのシグニチャは Amazon でホストされ、最新のアップデートをチェックするためのデフォルトの署名 URL として設定されています。

ただし、ユーザは、これらのシグニチャマッピングファイルを内部サーバにダウンロードするオプションがあります。その後、ユーザーは別のシグニチャ URL パスを設定して、ローカルサーバからシグニチャマッピングファイルをダウンロードできます。自動更新機能を機能させるには、外部サイトにアクセスするように DNS サーバーを構成する必要があります。

署名の更新

appfw デフォルトシグニチャオブジェクトを使用して作成されたすべてのユーザー定義シグニチャオブジェクトのバージョンが 0 より大きい。シグニチャ自動更新を有効にすると、すべてのシグニチャが自動的に更新されます。

ユーザが Cenzic や Qualys などの外部フォーマットでシグニチャをインポートした場合、シグニチャはバージョンをゼロとしてインポートします。同様に、ユーザーが空のテンプレートを使用して署名オブジェクトを作成した場合は、ゼロバージョンの署名として作成されます。これらのシグニチャは自動的には更新されません。これは、使用されていないデフォルトシグニチャを管理するオーバーヘッドにユーザーが関心がない場合があるためです。

ただし、Web アプリケーションファイアウォールでは、これらのシグニチャを手動で選択して更新して、既存のルールにデフォルトのシグニチャルールを追加することもできます。シグニチャを手動で更新すると、バージョンが変更され、シグニチャは他のシグニチャとともに自動的に更新されます。

シグニチャ自動更新を構成する

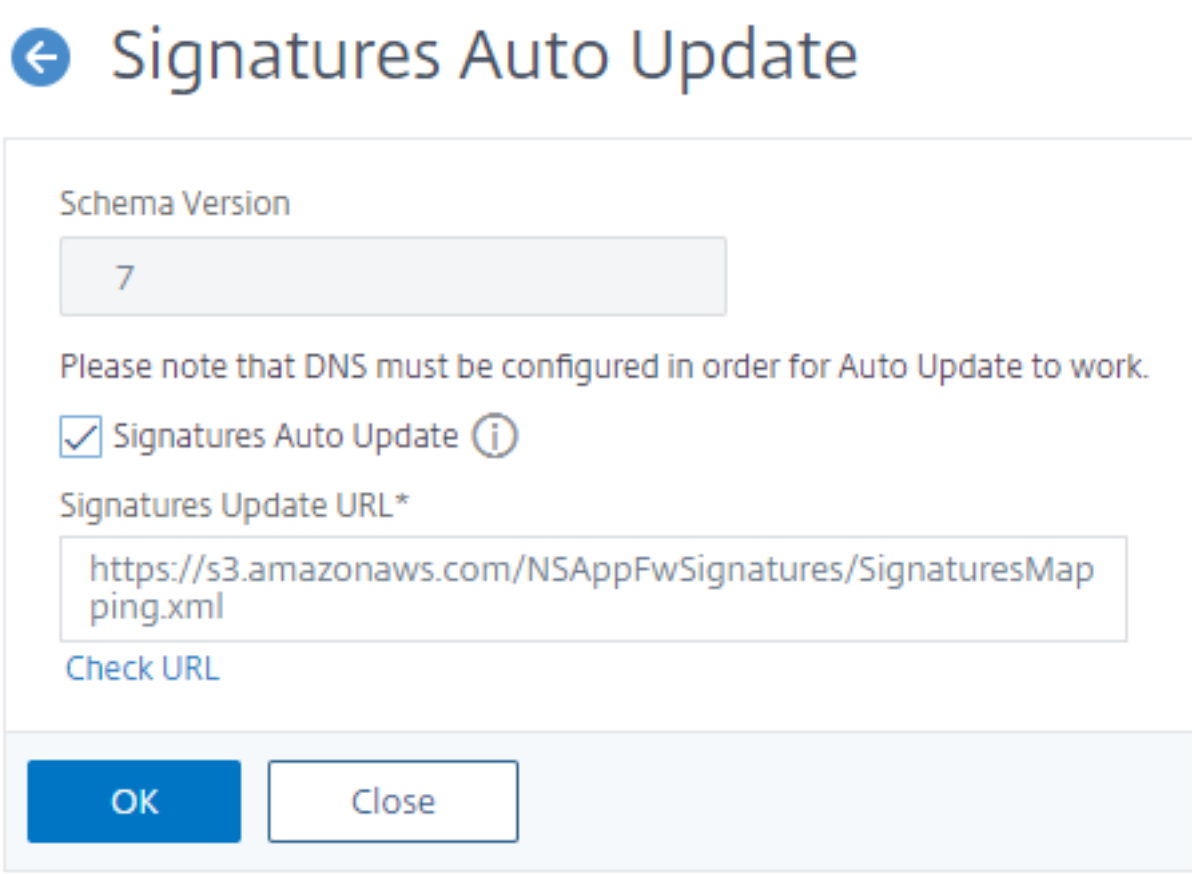
CLI を使用してシグニチャ自動更新機能を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 set appfw settings SignatureAutoUpdate on
2 set appfw settings SignatureUrl https://s3.amazonaws.com/
     NSAppFwSignatures/SignaturesMapping.xml
3 <!--NeedCopy-->
```

GUI を使用してシグニチャ自動更新を設定するには、次の手順を実行します。

1. [セキュリティ] ノードを展開します。
2. [アプリケーションファイアウォール] ノードを展開します。
3. [署名] ノードを選択します。
4.「アクション」から「設定を自動更新」を選択します。
5. [シグニチャの自動更新] オプションを有効にします。
6. 必要に応じて、シグニチャ更新 URL のカスタマイズされたパスを指定できます。[リセット] をクリックして、デフォルトの`s3.amazonaws.com server`にリセットします。
7.「**OK**」をクリックします。



シグニチャを手動で更新する

ゼロバージョンシグニチャまたはその他のユーザ定義シグニチャを手動で更新するには、まずデフォルトシグニチャの最新のアップデートを取得し、これを使用してターゲットユーザ定義シグニチャを更新する必要があります。

CLI から次のコマンドを実行して、シグニチャファイルを更新します。

```
update appfw signatures ”*Default Signatures”
update appfw signatures cenzic  - mergedefault
<!--NeedCopy-->
```

> 注:
>
> `Default Signatures` 大文字と小文字は区別されます。前のコマンドの Cenzic は、更新されるシグニチャファイルの名前です。

インターネットにアクセスせずにデフォルトの署名をインポートする

最新のアップデートを入手するには、Amazon (AWS) サーバーを指すようにプロキシサーバーを設定することをお勧めします。ただし、NetScaler アプライアンスに外部サイトへのインターネット接続がない場合、ユーザーは更新された署名ファイルをローカルサーバーに保存できます。その後、アプライアンスはローカルサーバからシグニチャをダウンロードできます。このシナリオでは、ユーザーは常に **Amazon** サイトをチェックして、最新のアップデートを入手する必要があります。**Citrix** 公開キーを使用して改ざんから保護して作成された対応する **sha1** ファイルに対して、署名ファイルをダウンロードして検証できます。

Signatures ファイルをローカルサーバにコピーするには、次の手順を実行します。

1. <MySignatures>などのローカルディレクトリを、ローカルサーバー上に作成します。
2. AWS サイトを開きます。
3. SignaturesMapping.xmlファイルを<MySignatures>フォルダにコピーします。

SignaturesMapping.xmlファイルを開くと、シグネチャ用のすべての xml ファイルと、サポートされているさまざまなバージョンの対応する sha1 ファイルが表示されます。次のスクリーンショットでは、そのようなペアの 1 つが強調表示されています。

1. <MySignatures>フォルダにサブディレクトリ<sigs>を作成します。
2. *.xml.sha1ファイルの対応する*.xml files listed in the <file>タグにリストされている<sha1> タグとSignaturesMapping.xmlファイルのすべてのペアを<sigs>フォルダにコピーします。次に、<sigs> フォルダにコピーされるサンプルファイルをいくつか示します。

```
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml.sha1
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml.sha1
```

> 注:
>
> <MySignatures> フォルダには任意の名前を付けて、任意の場所に置くことができます。ただし、サブディレクトリ<sigs>は、マッピングファイルがコピーされる<MySignatures>フォルダ内のサブディレクトリである必要があります。また、SignaturesMapping.xml に示すように、サブディレクトリ名<sigs>は正確な名前で、大文字と小文字が区別されていることを確認してください。すべてのシグネチャファイルとそれに対応する sha1 ファイルは、この<sigs>ディレクトリの下にコピーする必要があります。

ホストされている Amazon web サーバーからローカルサーバーにコンテンツをミラーリングした後、新しいローカルウェブサーバーへのパスを変更して、自動更新用の SignatureUrl に設定します。たとえば、アプライアンスのコマンドラインインターフェイスから次のコマンドを実行します。

```
1 set appfw settings SignatureUrl https://myserver.example.net/
     MySignatures/SignaturesMapping.xml
2 <!--NeedCopy-->
```


更新するシグニチャの数によっては、更新処理に数分かかる場合があります。更新操作が完了するまでに十分な時間を確保してください。

エラーに直面した場合「URL にアクセス中にエラーが発生しました!」設定中に、手順に従って解決してください。

1. コンテンツセキュリティポリシー (CSP) セキュリティが URL アクセスをブロックしないように、URLhttps://myserver.example.netを /netscaler/ns_gui/admin_ui/php/application/controllers/common/utils.phpに追加します。これらの設定はアップグレードでは保持されないことに注意してください。ユーザーはアップグレード後に再度追加する必要があります。

```
$configuration_view_connect_src = "connect-src 'self' https://app.pendo
    .io https://s3.amazonaws.comhttps://myserver.example.net;";
<!--NeedCopy-->
```

1. ユーザは、https://myserver.example.net/MySignatures/SignaturesMapping.xmlの次の CORS ヘッダーに応答するように Web サーバhttps://myserver.example.netを設定する必要があります。

```
Access-Control-Allow-Methods: GET
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 3000
<!--NeedCopy-->
```

シグニチャを更新するためのガイドライン

シグニチャの更新時には、次のガイドラインが使用されます。

- シグニチャ更新の URL に同じバージョンまたは新しいバージョンの署名オブジェクトが含まれていると、シグニチャが更新されます。
- 各シグニチャルールは、ルール ID とバージョン番号に関連付けられます。たとえば、次のようになります: <SignatureRule id="803"version="16"…>
- 既存のものと同じ ID およびバージョン番号を持つ着信シグニチャファイルのシグニチャルールは、パターンやログ文字列が異なる場合でも無視されます。
- 新しい ID を持つシグニチャルールが追加されます。すべてのアクションと enabled フラグが新しいファイルから使用されます。

  > 注:

> アプリケーションの要件に従って、これらの新しく追加されたルールを有効にし、その他のアクション設定を変更するために、更新されたシグニチャを定期的に確認する必要があります。

- 同じ ID を持つが、新しいバージョン番号を持つルールは、既存のルールを置き換えます。既存のルールからのすべてのアクションと有効フラグが保持されます。

> ヒント：
>
> CLI からシグニチャを更新する場合は、まずデフォルトシグニチャを更新する必要があります。次に、デフォルトシグニチャに基づく各カスタムシグニチャファイルを更新するには、update コマンドを追加する必要があります。最初にデフォルトシグニチャを更新しないと、バージョンの不一致エラーによってカスタムシグニチャファイルの更新が防止されます。

Snort ルールの統合

October 7, 2021

Web アプリケーションに対する悪意のある攻撃では、内部ネットワークを保護することが重要です。悪意のあるデータは、インターフェイスレベルで Web アプリケーションに影響を与えるだけでなく、悪意のあるパケットもアプリケーション層に到達します。このような攻撃に対処するには、内部ネットワークを調べる侵入検知および防止システムを構成することが重要です。

Snort ルールはアプライアンスに統合され、アプリケーション層でのデータパケットにおける悪意のある攻撃を検査します。snort ルールをダウンロードし、WAF シグニチャルールに変換できます。シグネチャには、DOS 攻撃、バッファオーバーフロー、ステルスポートスキャン、CGI 攻撃、SMB プローブ、OS フィンガープリントの試行などの悪意のあるアクティビティを検出できるルールベースの構成があります。Snort ルールを統合することで、インターフェイスとアプリケーションレベルでセキュリティソリューションを強化できます。

Snort ルールの設定

設定では、まず Snort ルールをダウンロードし、次に WAF シグニチャルールにインポートします。ルールを WAF シグニチャに変換すると、そのルールを WAF セキュリティチェックとして使用できます。snort ベースのシグニチャルールは、着信データパケットを調べて、ネットワークに悪意のある攻撃があるかどうかを検出します。

新しいパラメータ「vendorType」が import コマンドに追加され、Snort ルールを WAF シグニチャに変換します。

パラメータ「vendorType」は、Snort ルールの場合のみ、SNORT に設定されます。

コマンドインターフェイスを使用して **snort** ルールをダウンロードする

以下の URL から Snort ルールをテキストファイルとしてダウンロードできます。

```
https://www.snort.org/downloads/community/snort3-community-rules.tar.gz
```


コマンドインターフェイスを使用して **snort** ルールをインポートする

ダウンロード後、Snort ルールをアプライアンスにインポートできます。

コマンドプロンプトで入力します。

```
import appfw signatures <src> <name> [-xslt <string>] [-comment <string>]
[-overwrite] [-merge [-preservedefactions]] [-sha1 <string>] [-VendorType
Snort]
```

例:

```
import appfw signatures http://www.example.com/ns/signatures.xml sig-snort -
comment “signatures from snort rules” -VendorType snort
```

引数:

Src インポートされたシグニチャオブジェクトを格納する場所の URL（プロトコル、ホスト、パス、ファイル名）。

> 注:
>
> インポートするオブジェクトが、アクセスのためにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。最大長の必須引数：2047

Name：Citrix ADC 上の署名オブジェクトに割り当てる名前。最大長の必須引数:31

コメント。signature オブジェクトに関する情報を保持する方法の説明。最大長:255
上書き同じ名前の既存のシグニチャオブジェクトを上書きします。

マージ。既存の署名を新しい署名ルールにマージします。

防腐剤。シグニチャルールの def アクションを保持します。

ベンダータイプ。WAF シグニチャを生成するサードパーティベンダー。可能な値: スノト。

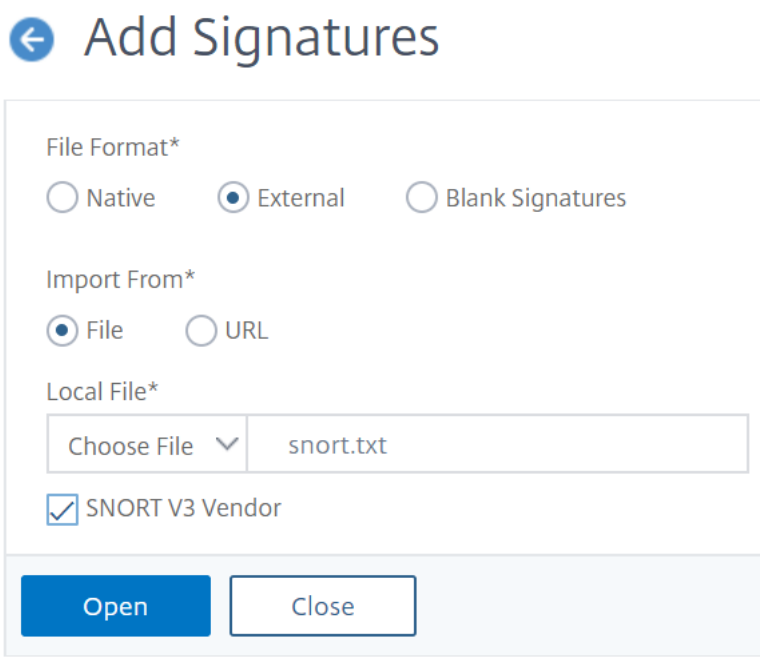
Citrix ADC GUI を使用して **snort** ルールを構成する

Snort ルールの GUI 設定は、Cenzic、Qualys、Whitehat のような他の外部 Web アプリケーションスキャナの設定に似ています。

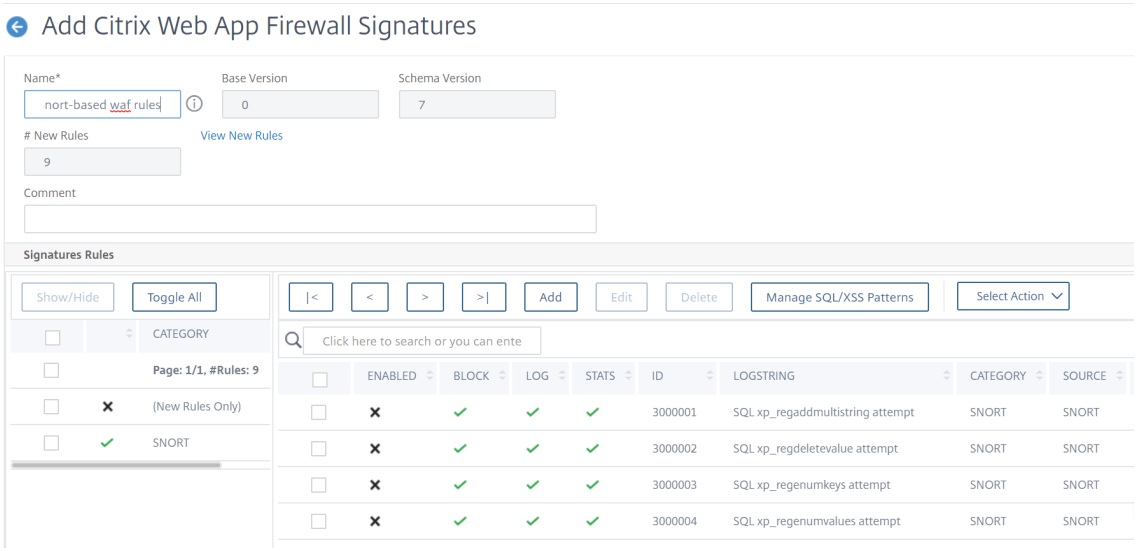
Snort を設定するには、以下の手順に従います。

1. ［構成］ **>** ［セキュリティ］ **>** ［**Citrix Web App Firewall**］ **>** ［署名］
2. [署名] ページで、[追加] をクリックします。
3. **[**署名の追加] ページで、次のパラメータを設定して Snort ルールを設定します。
   a) ファイル形式。ファイルフォーマットを外部として選択します。
   b) インポート元。インポートオプションを snort ファイルまたは URL として選択し、URL を入力します。
   c) スノート V3 ベンダー. ファイルまたは URL から Snort ルールをインポートするには、このチェックボックスをオンにします。

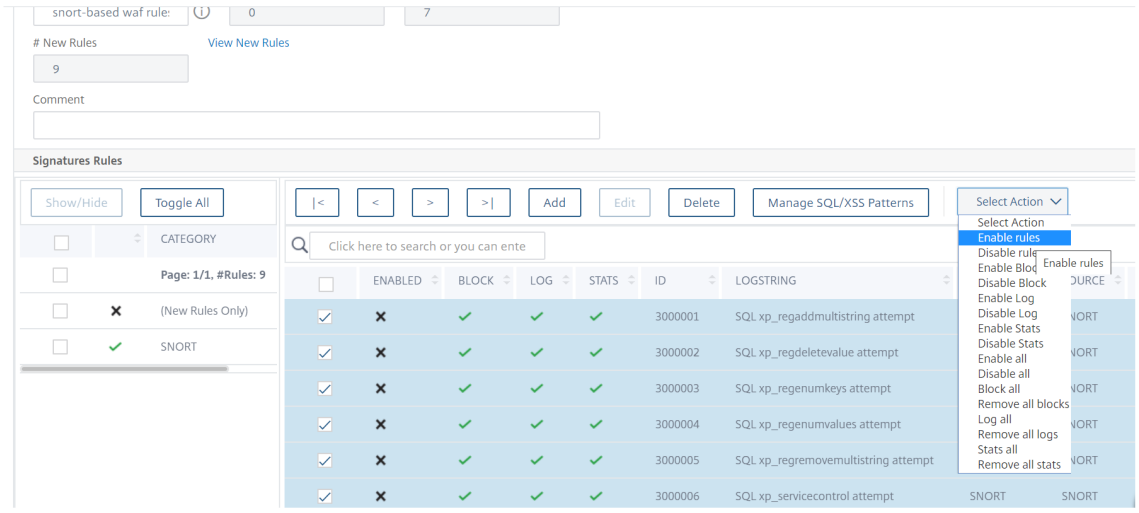
4. ［開く］をクリックします。



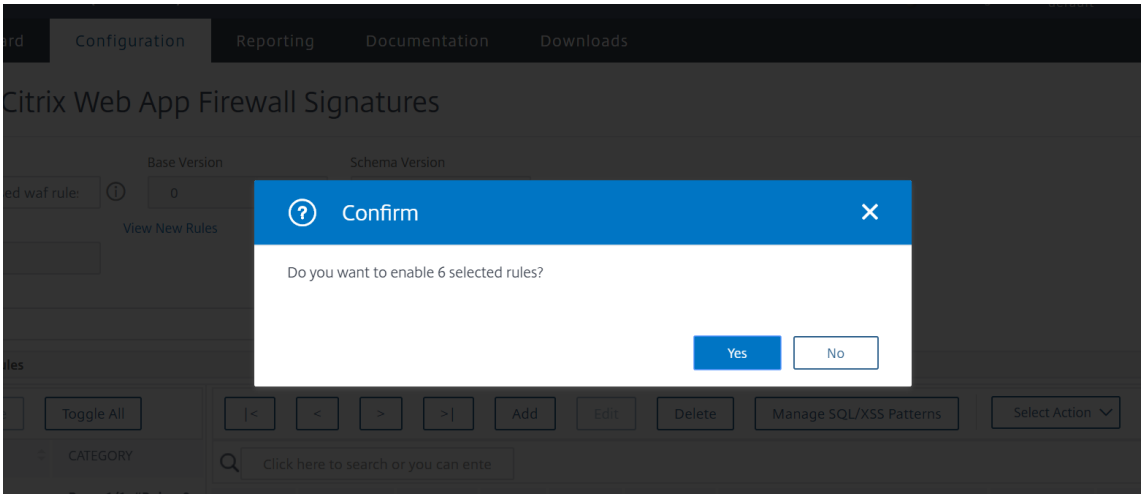
アプライアンスは、Snort ルールを snort ベースの WAF シグニチャルールとしてインポートします。



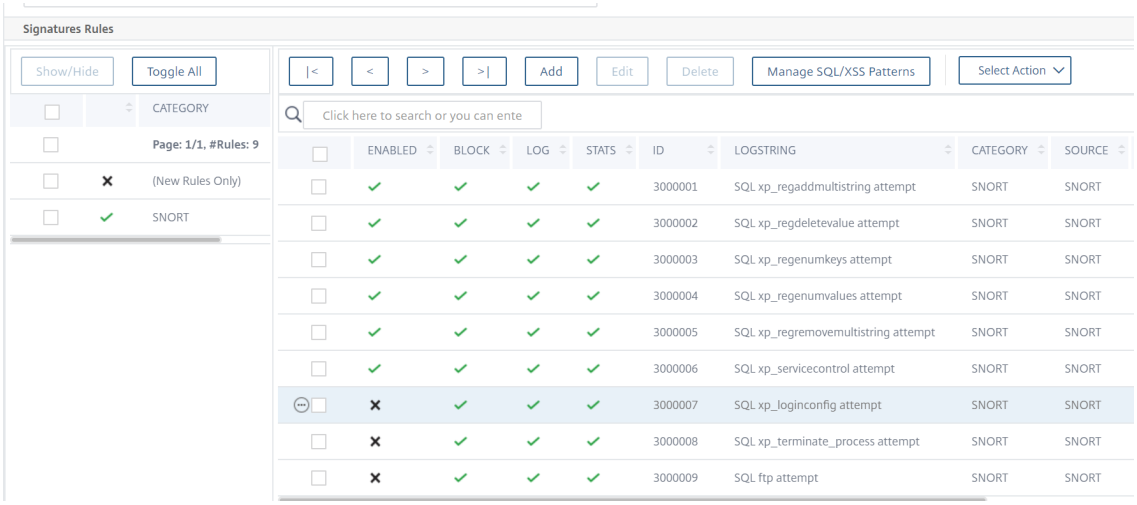
ベストプラクティスとして、フィルタアクションを使用して、アプライアンスで WAF 署名ルールとしてインポートする snort ルールを有効にする必要があります。



5. 確認するには、[はい] をクリックします。



6. 選択したルールがアプライアンス上で有効になります。



7. [**OK**] をクリックします。

シグネチャオブジェクトのファイルへのエクスポート

October 7, 2021

署名オブジェクトをファイルにエクスポートして、別の Citrix ADC にインポートできるようにします。

シグネチャオブジェクトをファイルにエクスポートするには

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ペインで、構成するシグニチャオブジェクトを選択します。
3.「アクション」ドロップダウンリストで、「エクスポート」を選択します。
4.「署名オブジェクトのエクスポート」ダイアログボックスの「ローカルファイル」テキストボックスに、署名オブジェクトをエクスポートするファイルのパスと名前を入力するか、「参照」ダイアログを使用してパスと名前を指定します。
5.［**OK**］をクリックします。

署名エディター

October 7, 2021

シグニチャエディタを使用して、ユーザ定義（ローカル）シグニチャルールを既存のシグニチャオブジェクトに追加または変更できます。ローカル署名ルールは、Citrix のデフォルトシグニチャルールと同じ属性を持ち、同じように機能します。デフォルトシグニチャの場合と同様に、有効または無効にし、シグニチャアクションを設定します。

既存の署名が一致しない既知の攻撃から Web サイトとサービスを保護する必要がある場合は、ローカルルールを追加します。たとえば、新しいタイプの攻撃を発見し、Web サーバー上のログを調べてその特性を判断したり、新しいタイプの攻撃に関するサードパーティ情報を取得したりできます。

シグニチャルールの中心には、ルール パターンがあります。ルールパターンは、ルールが一致するように設計されている攻撃の特性をまとめて表します。各パターンは、単純な文字列、PCRE 形式の正規表現、または組み込みの SQL インジェクションまたはクロスサイトスクリプティングパターンで構成できます。

新しいパターンを追加するか、攻撃に一致するように既存のパターンを変更することで、シグニチャルールを変更できます。たとえば、攻撃に対する変更を確認したり、Web サーバー上のログを調べたり、サードパーティの情報から適切なパターンを判断したりできます。

署名エディタを使用してローカルシグニチャルールを追加または変更するには

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ウィンドウで、編集する署名オブジェクトを選択し、[開く] をクリックします。

3. **[**署名オブジェクトの修正] ダイアログボックスの [フィルタされた結果] ウィンドウの下の画面中央で、次のいずれかの操作を行います。

   - 新しいローカルシグニチャルールを追加するには、[Add] をクリックします。
   - 既存のローカルシグニチャルールを変更するには、そのルールを選択し、[開く] をクリックします。

4. [ローカル署名規則の追加] ダイアログボックスまたは **[**ローカル署名規則の変更] ダイアログボックスで、適切なチェックボックスをオンにして、署名の動作を構成します。

   - 有効。新しいシグニチャルールを有効にします。これを選択しない場合、この新しいシグニチャルールは設定に追加されますが、非アクティブになります。
   - ブロック。このシグニチャルールに違反する接続をブロックします。
   - ログ。このシグネチャ規則の違反を Citrix ADC ログに記録します。
   - **Stat**。このシグニチャルールの違反を統計情報に含めます。
   - **Remove-** 応答からシグニチャルールに一致する情報を除外します。(応答ルールにのみ適用されます)。
   - **X-Out**。シグニチャルールと一致する情報を文字 X でマスクします (応答ルールにのみ適用)。
   - **Allow Duplicates**。このシグニチャオブジェクトでこのシグニチャルールの重複を許可します。

5. [**Category**] ドロップダウンリストから、新しいシグニチャルールのカテゴリを選択します。

   また、リストの右側にあるアイコンをクリックし、[署名ルールカテゴリの追加] ダイアログボックスを使用して新しいカテゴリをリストに追加して、カテゴリを作成することもできます。変更するルールは、新しいカテゴリに自動的に追加されます。手順については、「シグニチャルールカテゴリを追加するには」を参照してください。

6. [**LogString**] テキストボックスに、ログで使用するシグニチャルールの簡単な説明を入力します。

7. 「コメント」テキスト・ボックスにコメントを入力します。(オプション)

8. [詳細...] をクリックし、詳細オプションを変更します。

   a) このシグニチャルールを適用する前に HTML コメントを削除するには、[Strip Comments] ドロップダウンリストで [All] または [Exclude Script Tag] を選択します。
   b) CSRF リファラーヘッダーのチェックを設定するには、[CSRF リファラーヘッダーチェック] ラジオボタン配列で、[存在する場合] または [常に表示] ラジオボタンのいずれかを選択します。
   c) このローカルシグニチャルールに割り当てられたルール ID を手動で変更するには、[Rule ID] テキストボックスで番号を変更します。ID は、ローカルシグニチャルールにまだ割り当てられていない 1000000 ～1999999 の正の整数である必要があります。
   d) 新しいシグニチャルールにバージョン番号を割り当てるには、[Version Number] テキストボックスで番号を変更します。
   e) ソース ID を割り当てるには、[ソース ID] テキストボックスで文字列を変更します。
   f) ソースを指定するには、「ソース」(Source) ドロップダウンリストから「ローカル」(Local) または「スナート」(Snort) を選択するか、リストの右側にある「追加」(Add) アイコンをクリックして新しいソースを追加します。
   g) このローカルシグニチャルールの違反に危害スコアを割り当てるには、[Harm Score] テキストボックスに 1 ～10 の数値を入力します。

h) このローカルシグニチャルールに重大度を割り当てるには、[Severity] ドロップダウンリストで [High]、[Medium]、または [Low] を選択するか、リストの右側にある [Add] アイコンをクリックして新しい重大度を追加します。

i) このローカルシグニチャルールに違反タイプを割り当てるには、[Type] ドロップダウンリストで [Vulnerable] または [Warning] を選択するか、リストの右側にある [Add] アイコンをクリックして新しい違反タイプを追加します。

9. [パターン] リストで、パターンを追加または編集します。

   - パターンを追加するには、[追加] をクリックします。[新しい署名規則パターンの作成] ダイアログボックスで、署名規則に 1 つまたは複数のパターンを追加し、[**OK**] をクリックします。
   - パターンを編集するには、パターンを選択して [開く] をクリックします。[署名規則のパターンの編集] ダイアログボックスで、パターンを変更し、[**OK**] をクリックします。

   パターンの追加または編集の詳細については、「シグニチャルールパターン」を参照してください。

10. [**OK**] をクリックします。

署名ルールカテゴリを追加するには

October 7, 2021

シグニチャルールをカテゴリに配置すると、個々のシグニチャではなく、シグニチャグループのアクションを設定できます。次の理由により、これを行うことができます。

- 選択の容易さ。たとえば、特定のグループ内のすべてのシグニチャルールが、特定のタイプの Web サーバソフトウェアまたはテクノロジーに対する攻撃から保護されているとします。保護された Web サイトがそのソフトウェアまたはテクノロジーを使用している場合は、それらすべてを有効にする必要があります。有効になっていない場合は、いずれも有効にしたくありません。
- 初期設定の容易さ。シグニチャのグループのデフォルトを 1 つずつではなく、カテゴリとして設定するのが最も簡単です。その後、必要に応じて個々のシグニチャに変更を加えることができます。
- 継続的な構成の容易さ。特定のカテゴリに属するなど、特定の基準を満たすシグニチャだけを表示できる場合は、シグニチャの設定が簡単です。

1. [セキュリティ] > [**Web App Firewall**] > [署名] に移動します。
2. 詳細ウィンドウで、構成するシグニチャオブジェクトを選択し、[開く] をクリックします。
3. **[**署名オブジェクトの修正] ダイアログボックスで、画面中央の [フィルタされた結果] ウィンドウの下にある [追加] をクリックします。
4. [ローカル署名規則の追加] ダイアログボックスで、[カテゴリ] ドロップダウンリストの右側にあるアイコンをクリックします。
5. [署名規則カテゴリの追加] ダイアログボックスの [新しいカテゴリ] テキストボックスに、新しい署名カテゴリの名前を入力します。名前は 1 ～64 文字で構成できます。
6. [**OK**] をクリックします。

シグニチャルールパターン

October 7, 2021

パターンを追加したり、既存のパターンを変更して、シグニチャが一致した場合に攻撃を特徴付ける文字列または式を指定できます。攻撃によって発生するパターンを検出するには、Web サーバー上のログを調べます。ツールを使用してリアルタイムで接続データを観察したり、攻撃に関するサードパーティのレポートから文字列や式を取得したりできます。

> 注意:
>
> シグネチャルールに追加する新しいパターンは、既存のパターンと AND 関係になります。潜在的な攻撃がシグニチャと一致するためにすべてのパターンと一致する必要がないようにするには、既存のシグニチャルールにパターンを追加しないでください。

各パターンは、単純な文字列、PCRE 形式の正規表現、または組み込みの SQL インジェクションまたはクロスサイトスクリプティングパターンで構成できます。正規表現に基づくパターンを追加する前に、PCRE 形式の正規表現を理解していることを確認する必要があります。PCRE 式は複雑で強力です。それらの仕組みを理解できない場合は、意図せず自分が望まなかったもの（偽陽性）に一致するパターン、または自分が望むもの（偽 ** 陰性）に一致しないパターンを作成できます。

デフォルト以外のコンテンツタイプのカスタム署名パターン

Citrix ADC Web App Firewall（WAF）で、正規化されたコンテンツを検査するための新しい場所がサポートされるようになりました。デフォルトでは、WAF はデフォルト以外のコンテンツタイプのエンコードされたペイロードをブロックしません。これらのコンテンツタイプがホワイトリストに登録され、設定済みのアクションが適用されない場合、SQL およびクロスサイトスクリプティング保護チェックでは、エンコードされたペイロードでの SQL 攻撃またはクロスサイトスクリプティング攻撃はフィルタリングされません。この問題を解決するために、ユーザはこの新しい場所（HTTP_CANON_POST_BODY）を使用してカスタムシグニチャルールを作成できます。このルールでは、エンコードされたペイロードでデフォルト以外のコンテンツタイプを調べます。SQL またはクロスサイトスクリプティング攻撃がある場合、ポストボディの正規化後にトラフィックをブロックします。

> 注:
>
> このサポートは、HTTP 要求にのみ適用されます。

PCRE-形式の正規表現に慣れていない場合は、次のリソースを使用して基本を学習したり、特定の問題に関するヘルプを参照したりできます。

- 「正規表現のマスタリング」第 3 版。ジェフリー・フリードルによる著作権 (c) 2006. オライリー・メディア, ISBN: 9780596528126.
- 「正規表現クックブック」. 著作権 (c) 2009 ヤン・ゴイバーツとスティーブン・レビサンによる. オライリーメディア, ISBN:
- **PCRE** マニュアルページ/仕様 (テキスト/公式):http://www.pcre.org/pcre.txt

- **PCRE** マンページ/仕様

http://www.gammon.com.au/pcre/index.html

- **Wikipedia PCRE** エントリー: http://en.wikipedia.org/wiki/PCRE
- **PCRE** メーリングリスト
  PCRE-dev-PCRE 開発

非 ASCII 文字を PCRE 形式の正規表現でエンコードする必要がある場合、Citrix ADC プラットフォームでは 16 進数の UTF-8 コードのエンコードがサポートされます。詳細については、PCRE 文字エンコーディング形式を参照してください。

シグニチャルールパターンを設定するには

1. **Security > Citrix Web App Firewall > Signatures** に移動します。
2. 詳細ウィンドウで、構成するシグニチャオブジェクトを選択し、[開く] をクリックします。
3. **[**シグニチャオブジェクトの変更] ダイアログボックスで、画面の中央の [フィルタされた結果] ウィンドウの下にある [追加] をクリックしてシグニチャルールを作成するか、既存のシグニチャルールを選択して [開く] をクリックします。

   > 注:
   >
   > 変更できるのは、追加したシグニチャルールだけです。デフォルトのシグニチャルールは変更できません。

   操作に応じて、[ローカル署名規則の追加] または [ローカル署名規則の変更] ダイアログボックスが表示されます。両方のダイアログボックスの内容は同じです。
4. ダイアログボックスの「パターン」ウィンドウで、「追加」(**Add**) をクリックして新しいパターンを追加するか、「追加」(**Add**) ボタンの下のリストから既存のパターンを選択して「開く」(**Open**) をクリックします。アクションに応じて、**[**新しい署名規則パターンの作成] ダイアログボックスまたは [署名規則パターンの編集] ダイアログボックスが表示されます。両方のダイアログボックスの内容は同じです。
5. [パターンタイプ] ドロップダウンリストで、パターンが一致させる接続のタイプを選択します。
   - 挿入された SQL コード、Web フォームへの攻撃、クロスサイトスクリプト、不適切な URL など、リクエストの要素または機能に一致するパターンがある場合は、[**Request**] を選択します。
   - パターンが応答要素または機能 (クレジットカード番号や安全なオブジェクトなど) を一致させる場合は、[**Response**] を選択します。
6. [位置] 領域で、このパターンで調べる要素を定義します。
   [Location] 領域には、このパターンを調べるための HTTP 要求または応答の要素が表示されます。「位置」(Location) エリアに表示される選択肢は、選択したパターンタイプによって異なります。パターンタイプとして [Request] を選択した場合、HTTP リクエストに関連する項目が表示されます。[応答] を選択すると、HTTP 応答に関連する項目が表示されます。

さらに、「エリア」(Area) ドロップダウンリストから値を選択すると、「位置」(Location) 領域の残りの部分がインタラクティブに変更されます。このセクションに表示される可能性のあるすべての構成項目を次に示します。

- エリア。HTTP 接続の特定の部分を記述する要素のドロップダウンリスト。選択肢は次のとおりです。
  - **HTTP_ANY**. HTTP 接続のすべての部分。
  - **HTTP_COOKIE**. 任意のクッキー変換が実行された後、HTTP リクエストヘッダー内のすべてのクッキー。
    注：HTTP 応答 “Set-Cookie：” ヘッダーを検索しません。
  - **HTTP_FORM_FIELD**. URL デコード、パーセントデコード、および余分な空白の削除後のフォームフィールドとその内容。<Location>タグを使用すると、検索するフォームフィールド名のリストをさらに制限できます。
  - **HTTP_HEADER**. クロスサイトスクリプティングまたは URL デコード変換後の HTTP ヘッダーの値部分。
  - **HTTP_METHOD**. HTTP リクエストメソッド。
  - **HTTP_ORIGIN_URL**. Web フォームのオリジン URL。
  - **HTTP_POST_BODY**. HTTP ポスト本文とそれに含まれる Web フォームデータ。
  - **HTTP_RAW_COOKIE**.「クッキー:」名部分を含むすべての HTTP リクエストクッキー。
    注：HTTP 応答 “Set-Cookie：” ヘッダーを検索しません。
  - **HTTP_RAW_HEADER**. HTTP ヘッダー全体。個々のヘッダーは改行文字（n）または改行/改行文字列（rn）で区切られます。
  - **HTTP_RAW_RESP_HEADER**. URL 変換が完了した後の応答ヘッダーの名前と値の部分、および完全な応答状態を含む、応答ヘッダー全体。HTTP_RAW_HEADER と同様に、個々のヘッダーは改行文字 (n) または改行/改行文字列 (rn) で区切られます。
  - **HTTP_RAW_SET_COOKIE**. URL 変換が実行された後の Set-Cookie ヘッダー全体
    注:URL 変換では、Set-Cookie ヘッダーのドメイン部分とパス部分の両方を変更できます。
  - **HTTP_RAW_URL**. クエリまたはフラグメント部分を含む、URL 変換が実行される前のリクエスト URL 全体。
  - **HTTP_RESP_HEADER**. URL 変換が実行された後の完全な応答ヘッダーの値部分。
  - **HTTP_RESP_BODY**. HTTP 応答本文
  - **HTTP_SET_COOKIE**. HTTP 応答ヘッダーのすべての「Set-Cookie」ヘッダー。
  - **HTTP_STATUS_CODE**. HTTP ステータスコード。
  - **HTTP_STATUS_MESSAGE**. HTTP ステータスメッセージ。
  - **HTTP_URL**. UTF-* 文字セットへの変換、URL デコード、空白の除外、および相対 URL の絶対値への変換後の HTTP ヘッダー内の URL の値部分。クエリポートまたはフラグメントポートを除く。HTML エンティティのデコードは含まれません。
  - URL。[
    Area] 設定で指定された要素内で見つかった URL を検査します。次のいずれかの設定を選択します。
  - **Any.** すべての URL をチェックします。

- リテラル。リテラル文字列を含む URL をチェックします。[リテラル] を選択すると、テキストボックスが表示されます。テキストボックスにリテラル文字列を入力します。
- PCRE PCRE 形式の正規表現に一致する URL をチェックします。この選択を選択すると、正規表現ウィンドウが表示されます。ウィンドウに正規表現を入力します。正規表現トークンを使用すると、カーソル位置に共通の正規表現要素を挿入できます。または、[正規表現エディタ] をクリックして [正規表現エディタ] ダイアログボックスを表示して、必要な正規表現の作成を支援できます。
- 式。Citrix ADC デフォルト式と一致する URL をチェックします。
- フィールド名。[
  Area] 選択で指定された要素で見つかったフォームフィールド名を調べます。**Any.** すべての URL をチェックします。
- リテラル。リテラル文字列を含む URL をチェックします。[リテラル] を選択すると、テキストボックスが表示されます。テキストボックスにリテラル文字列を入力します。
- PCRE PCRE 形式の正規表現に一致する URL をチェックします。この選択を選択すると、正規表現ウィンドウが表示されます。ウィンドウに正規表現を入力します。正規表現トークンを使用して一般的な正規表現要素を挿入するか、正規表現エディタを使用して必要な正規表現を構築することができます。
- 式。Citrix ADC デフォルト式と一致する URL をチェックします。

7. [パターン] 領域で、パターンを定義します。パターンは、一致させるパターンを定義するリテラル文字列または PCRE 形式の正規表現です。[パターン] 領域には、次の要素が含まれます。
- 一致。署名に使用できる検索方法のドロップダウンリスト。このリストは、パターンタイプが [要求] か [応答] かによって異なります。

   リクエストマッチタイプ
   PCRE。PCRE 形式の正規表現。

   > 注:
   >
   > PCRE を選択すると、パターンウィンドウの下にある正規表現ツールが有効になります。これらのツールは、他のほとんどのタイプのパターンには役に立ちません。

   - **Injection**. 指定された場所に挿入された SQL を探すように Web App Firewall に指示します。Web App Firewall に SQL インジェクションのパターンが既にあるため、[パターン] ウィンドウが消えます。
   - **CrossSiteScripting**. Web App Firewall に、指定した場所でクロスサイトスクリプトを検索するように指示します。Web App Firewall にクロスサイトスクリプトのパターンが既に存在するため、[パターン] ウィンドウが消えます。
   - 式。Citrix ADC のデフォルト表現言語の表現は、Citrix ADC アプライアンスで Web App Firewall ポリシーを作成する場合と同じ表現言語です。Citrix ADC 式言語は、もともとポリシールール用に開発されたものですが、柔軟性の高い汎用言語で、署名パターンの定義にも使用できます。

   [式] を選択すると、[パターン] ウィンドウの下に Citrix ADC 式エディターが表示されます。式エディタとその使用方法の詳細については、「[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには」を参照してください。

回答の一致タイプ:

```
- Literal. A literal string
- PCRE. A PCRE-format regular expression.
```

> 注
>
> PCRE を選択すると、パターンウィンドウの下にある正規表現ツールが有効になります。これらのツールは、他のほとんどのタイプのパターンには役に立ちません。

- **Credit Card**. サポートされている 6 種類のクレジットカード番号のいずれかに一致する組み込みパターン。

> 注:
>
> [式] 一致タイプは、応答側の署名では使用できません。

- パターンウィンドウ (ラベルなし)

このウィンドウで、照合するパターンを入力し、追加データを入力します。

- リテラル。検索する文字列をテキスト領域で入力します。
- CRE. テキスト領域に正規表現を入力します。正規表現エディタを使用して必要な正規表現を作成するか、正規表現トークンを使用して一般的な正規表現要素をカーソル位置に挿入します。UTF-8 文字を有効にするには、[UTF-8] をクリックします。
- 式。テキスト領域に Citrix ADC の詳細式を入力します。[Prefix] を使用して式の最初の用語を選択するか、[Operator] を使用して一般的な演算子をカーソル位置に挿入します。[追加] をクリックして [式の追加] ダイアログボックスを開き、必要な正規表現の作成についてさらに支援します。「評価」(Evaluate) をクリックして「高度な式評価子」(Advanced Expression Evaluator) を開き、式の効果を判断します。
- [オフセット]: このパターンで一致を開始する前にスキップする文字数。このフィールドを使用して、最初の文字以外のポイントで文字列の検査を開始します。
- 奥行き。一致するかどうかを調べる開始点からの文字数。このフィールドを使用して、大きな文字列の検索を特定の文字数に制限します。
- 最小長さ。検索する文字列は、指定したバイト数以上でなければなりません。短い文字列は一致しません。
- 最大長さ。検索する文字列は、指定したバイト数を超えないようにしてください。長い文字列は一致しません。
- 検索方法。ファストマッチというラベルの付いたチェックボックス。パフォーマンスを向上させるために、リテラルパターンに対してのみファストマッチを有効にすることができます。

8. [**OK**] をクリックします。
9. 上記の 4 つの手順を繰り返して、さらにパターンを追加または変更します。
10. パターンの追加または変更が完了したら、[**OK**] をクリックして変更を保存し、[署名] ウィンドウに戻ります。

> 注意:
>
> [ローカル署名規則 の追加] または [ローカル署名規則の変更] ダイアログボックスで [**OK**] をクリックするまで、変更は保存されません。変更を破棄しない限り、[**OK**] をクリックせずにこれらのダイアログボックスを閉じないでください。

ルールをインポートおよびマージするには

October 7, 2021

シグニチャエディタを使用して GUI からインポートおよびマージ操作を実行するときに、新しい、更新、複製、および無効なルールを確認できるようになりました。

署名エディタには、次の 4 つの新しい行が表示されます。

1. 新しいルール
2. 更新されたルール
3. 重複ルール
4. 無効なルール

[新しい規則のみ] フィルタと [更新された規則のみ] フィルタの出力は、署名エディタの [編集] ウィンドウの [カテゴリフィルタ] ペインにも表示されます。

新しい、複製、無効および更新されたルールの対応するリンクを表示するには、GUI からファイルをインポートする必要があります。

シグニチャルールをインポートする手順:

1. Citrix ADC web GUI で **Configuration > Security > Citrix Web App Firewall Signatures** に移動します。[署名] ウィンドウで、[追加] をクリックします。[ファイル形式] > [ネイティブ]、**[**インポート元] > [**URL**] を選択し、[URL] フィールドに上記のリンクを追加します。URL にアクセスできない場合は、XML データをテキストファイル形式でダウンロードできます 。
2. [開く] をクリックすると、シグネチャファイルが開き、新しいルールと無効なルールのリンクが表示されます。
3. 3rdパーティシグニチャルールをインポートすると、インポートされた.xml ファイルに 90 個の新しいルールと 9 個の重複ルールが表示されます。URL にアクセスできない場合は、XML データをテキストファイル形式でダウンロードできます 。

高可用性の展開とビルドのアップグレードにおける署名の更新

October 7, 2021

シグニチャの更新は、プライマリノードで行われます。1 次ノードでシグニチャが更新される間、更新されたファイルは並行して 2 次ノードと同時に同期されます。

デフォルトシグニチャは常に最初に更新され、次に残りのユーザ定義シグニチャが更新されます。

Amazon AWS への接続

デフォルトのルート NSIP は、Amazon AWS への接続に使用されます。SNIP が使用される特定のユースケースのシナリオがあり、複数の SNIP が存在する場合、ホスティングサイトから ARP 応答を受信した最初のシナリオがルートを保持します。

バージョンアップグレード中の署名の更新

アップグレードの場合、NS にシグニチャの古いベースバージョンがある場合、* 新しいシグニチャバージョンが利用可能になると、デフォルトシグニチャが自動的に更新されます。

スキーマが変更された場合、バージョンのアップグレード時に、すべてのシグニチャオブジェクトのスキーマバージョンが更新されます。

ただし、ユーザ定義シグニチャのベースバージョンでは、リリース 10.5 とリリース 11.0 の動作が異なります。

リリース 10.5 では、デフォルトのシグニチャのみが更新され、残りのシグニチャのベースバージョンはビルドのアップグレード後も変更されません。

リリース 11.0 では、この動作が変更されました。アプライアンスをアップグレードして新しいビルドをインストールすると、*Default 署名オブジェクトだけでなく、アプライアンス内に現在存在するその他のユーザー定義署名もすべて更新され、ビルドのアップグレード後も同じバージョンになります。

10.5 および 11.0 リリースビルドの両方で、自動更新が設定されている場合、*Default Signatures とゼロ以外のバージョンのシグニチャはすべて最新リリースされたシグニチャバージョンに自動的に更新され、同じベースバージョンになります。

セキュリティー検査の概要

October 7, 2021

Web App Firewall の高度な保護（セキュリティチェック）は、保護された Web サイトおよび Web サービスに対する複雑または未知の攻撃をキャッチするように設計された一連のフィルターです。セキュリティチェックでは、ヒューリスティック、ポジティブセキュリティ、およびその他の手法を使用して、シグネチャだけでは検出できない攻撃を検出します。セキュリティチェックを構成するには、Web App Firewall プロファイルを作成して構成します。このプロファイルとは、Web App Firewall で使用するセキュリティチェックと、セキュリティチェックに失敗した要求または応答の処理方法を指示します。プロファイルは、シグニチャオブジェクトおよびセキュリティ設定を作成するためのポリシーに関連付けられます。

Web App Firewall には 20 のセキュリティチェックが用意されています。このチェックは、攻撃の対象となるタイプと、設定する複雑さが大きく異なります。セキュリティー検査は、次のカテゴリに分類されます。

- 一般的なセキュリティチェック。コンテンツに関係しない、またはすべてのタイプのコンテンツに同等に適用できる Web セキュリティのあらゆる側面に適用されるチェック。
- **HTML** セキュリティチェック。HTML リクエストと応答を調べるチェック。これらのチェックは、HTML ベースの Web サイト、および HTML と XML の混合コンテンツを含む Web2.0 サイトの HTML 部分に適用されます。
- **XML** セキュリティチェック。XML 要求と応答を調べるチェック。これらのチェックは、XML ベースの Web サービスと Web 2.0 サイトの XML 部分に適用されます。

セキュリティチェックは、オペレーションシステムと Web サーバーソフトウェアの脆弱性、SQL データベースの脆弱性、Web サイトと Web サービスの設計とコーディングのエラー、ホストまたはアクセスできるサイトのセキュリティ保護の失敗など、さまざまな種類の攻撃から保護します。機密情報。

すべてのセキュリティー検査には、チェックアクションという一連の構成オプションがあります。チェックアクションは、Web App Firewall がチェックに一致する接続を処理する方法を制御します。3 つのチェックアクションは、すべてのセキュリティチェックで使用できます。これには、次の種類のアカウントがあります。

- ブロック。シグニチャに一致する接続をブロックします。デフォルトでは、無効になっています。
- ログ。後で分析できるように、シグニチャに一致するログ接続。デフォルトで有効。
- 統計。シグニチャごとに、一致した接続数を示す統計情報を保持し、ブロックされた接続のタイプに関するその他の情報を提供します。デフォルトでは、無効になっています。

4 番目のチェックアクション「**Learn**」は、チェックアクションの半分以上に対して使用できます。保護された Web サイトまたは Web サービスへのトラフィックを監視し、セキュリティチェックに繰り返し違反する接続を使用して、チェックに対する推奨される例外（緩和）またはチェックの新しいルールを生成します。チェックアクションに加えて、特定のセキュリティチェックには、チェックでどの接続がそのチェックに違反しているかを判断したり、チェックに違反する接続に対する Web App Firewall の応答を構成したりするためのルールを制御するパラメータがあります。これらのパラメータはチェックごとに異なり、各チェックのドキュメントで説明されています。

セキュリティチェックを構成するには、「Web App Firewall ウィザード」で説明されているように Web App Firewall ウィザードを使用するか、GUI を使用した手動構成の説明に従って、セキュリティチェックを手動で構成できます。緩和や規則を手動で入力したり、学習したデータを確認するなど、一部のタスクは、コマンドラインではなく GUI を使用してのみ実行できます。通常、ウィザードの使用は最適な構成方法ですが、ウィザードに精通していて、1 回のセキュリティチェックで構成を調整するだけであれば、手動構成が簡単になる場合もあります。

セキュリティチェックの構成方法にかかわらず、セキュリティチェックごとに特定のタスクを実行する必要があります。多くのチェックでは、そのセキュリティチェックのブロックを有効にする前に、正当なトラフィックのブロックを防ぐために例外（緩和）を指定する必要があります。これは、一定量のトラフィックがフィルタリングされた後にログエントリを観察し、必要な例外を作成することによって、手動で行うことができます。ただし、通常、学習機能を有効にしてトラフィックを観察し、必要な例外を推奨するほうがはるかに簡単です。

Web App Firewall は、トランザクションの処理中にパケットエンジン (PE) を使用します。各パケットエンジンには、ほとんどの展開シナリオで十分な 100K セッションに制限があります。ただし、Web App Firewall が大量のト

ラフィックを処理していて、セッションタイムアウトが大きい値に設定されている場合、セッションが蓄積されることがあります。有効な Web App Firewall セッション数が PE あたり 100 K を超えると、Web アプリケーションファイアウォールのセキュリティチェック違反が Security Insight アプライアンスに送信されないことがあります。セッションタイムアウトの値を小さくするか、セッションレス URL 閉鎖またはセッションレスフィールドの一貫性を持つセキュリティチェックにセッションレスモードを使用すると、セッションが蓄積されるのを防ぐのに役立ちます。トランザクションが長いセッションを必要とするシナリオで、これが実行可能なオプションでない場合は、より多くのパケットエンジンを備えたハイエンドプラットフォームにアップグレードすることをお勧めします。

キャッシュされた AppFirewall のサポートが追加され、コアごとの CLI による最大セッション設定は 50K セッションに設定されます。

トップレベルの保護

October 7, 2021

Web App Firewall 保護のうち 4 つは、一般的な種類の Web 攻撃に対して特に効果的であるため、他のものよりも一般的に使用されます。これには、次の種類のアカウントがあります。

- **HTML** クロスサイトスクリプティング。スクリプトが配置されている Web サイトとは異なる Web サイトのコンテンツにアクセスまたは変更しようとするスクリプトの要求と応答を調べます。このチェックは、このようなスクリプトを検出すると、要求または応答を宛先に転送する前にスクリプトを無害にするか、接続をブロックします。
- **HTML SQL** インジェクション。SQL データベースに SQL コマンドを挿入しようとする試みについて、フォームフィールドデータを含む要求を検査します。このチェックは、挿入された SQL コードを検出すると、要求をブロックするか、または Web サーバーに要求を転送する前に、挿入された SQL コードを無害にします。

  注意: 次の両方の条件が構成に当てはまる場合は、Web AppFirewall が正しく構成されていることを確認する必要があります。

  - [HTML クロスサイトスクリプティング] チェックまたは [HTML SQL インジェクション] チェック (またはその両方) を有効にした場合、
  - 保護された Web サイトは、ファイルのアップロードを受け入れるか、大きな POST 本文データを含むことができる Web フォームを含みます。

  このケースを処理するための Web App Firewall の構成の詳細については、「アプリケーションファイアウォールの構成」を参照してください。
- バッファオーバーフロー。Web サーバーでバッファオーバーフローが発生する試みを検出する要求を調べます。
- クッキーの整合性。ユーザーリクエストとともに返された Cookie を調べ、Web サーバーがそのユーザーに設定した Cookie と一致することを確認します。変更された Cookie が見つかった場合、要求が Web サーバーに転送される前に、要求から取り除かれます。

バッファオーバーフローのチェックは簡単です。通常は、すぐにブロッキングを有効にできます。他の 3 つのトップレベルチェックはかなり複雑で、トラフィックをブロックするために安全に使用するには、設定が必要です。これらのチェックを手動で構成するのではなく、学習機能を有効にして、必要な例外を生成できるようにすることを強くお勧めします。

HTML クロスサイトスクリプティングチェック

October 7, 2021

HTML クロスサイトスクリプティング（クロスサイトスクリプティング）チェックでは、クロスサイトスクリプティング攻撃の可能性がないか、ユーザー要求のヘッダーと POST ボディの両方を調べます。クロスサイトスクリプトが見つかった場合、攻撃を無害にするためにリクエストを変更（変換）するか、リクエストをブロックします。

> 注:
>
> HTML クロスサイトスクリプティング（クロスサイトスクリプティング）チェックは、コンテンツタイプ、コンテンツの長さなどに対してのみ機能します。クッキーでは機能しません。また、Web アプリケーションファイアウォールプロファイルで「CheckRequestHeaders」オプションを有効にしてください。

同一生成元ルールに違反する HTML クロスサイトスクリプティングスクリプトを使用することで、保護された Web サイトでのスクリプトの誤用を防ぐことができます。このルールでは、スクリプトはサーバー以外のサーバーのコンテンツにアクセスしたり変更したりしてはなりません。同じオリジン規則に違反するスクリプトはクロスサイトスクリプトと呼ばれ、スクリプトを使用して別のサーバー上のコンテンツにアクセスまたは変更する習慣をクロスサイトスクリプティングと呼びます。クロスサイトスクリプティングがセキュリティ上の問題である理由は、クロスサイトスクリプティングを許可する Web サーバーは、その Web サーバー上ではなく、攻撃者が所有し制御しているサーバーなど、異なる Web サーバー上で攻撃される可能性があるためです。

残念ながら、多くの企業では、同じオリジンルールに違反する JavaScript が強化された Web コンテンツの大規模なインストールベースを持っています。このようなサイトで HTML クロスサイトスクリプティングチェックを有効にすると、適切な例外を生成して、チェックが正当な活動をブロックしないようにする必要があります。

Web App Firewall は、HTML クロスサイトスクリプティング保護を実装するためのさまざまなアクションオプションを提供します。ブロック、ログ、統計、学習アクションに加えて、送信されたリクエストの スクリプトタグをエンコードするエンティティによって、クロスサイトスクリプトを変換するオプションもあります。クロスサイトスクリプティングの完全な URL を確認するを構成して、クエリパラメータだけでなく URL 全体を検査してクロスサイトスクリプティング攻撃を検出するかどうかを指定できます。**InspectQueryContentTypes** パラメーターを構成して、特定のコンテンツタイプのクロスサイトスクリプティング攻撃の要求クエリ部分を検査できます。

リラクゼーションを展開して、誤検出を回避できます。Web App Firewall 学習エンジンは、緩和ルールの設定に関する推奨事項を提供できます。

アプリケーション用に最適化された HTML クロスサイトスクリプティング保護を構成するには、次のいずれかのアクションを構成します。

- [**Block**]：ブロックを有効にすると、要求でクロスサイトスクリプティングタグが検出されると、ブロックアクションがトリガーされます。
- **[Log**]：ログ機能を有効にすると、HTML クロスサイトスクリプティングチェックによって実行されるアクションを示すログメッセージが生成されます。ブロックが無効になっている場合、クロスサイトスクリプティング違反が検出されたヘッダーまたはフォームフィールドごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。同様に、クロスサイトスクリプティングタグが複数のフィールドで変換される場合でも、変換操作に対して要求ごとに 1 つのログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとした可能性があります。
- **Stats**：有効の場合、stats 機能は違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされている場合は、構成を再確認して、新しい緩和ルールを構成するか、既存の緩和ルールを変更する必要があるかどうかを確認する必要があります。
- 学習：どの緩和ルールがアプリケーションに最適かが不明な場合は、学習機能を使用して、学習したデータに基づいて HTML クロスサイトスクリプティングルールの推奨事項を生成できます。Web App Firewall 学習エンジンは、トラフィックを監視し、観測された値に基づいて学習の推奨事項を提供します。パフォーマンスを損なうことなく最適なメリットを得るには、ラーニングオプションを短時間有効にして、ルールの代表的なサンプルを取得してから、ルールを展開してラーニングを無効にします。
- クロスサイトスクリプトの変換：有効にすると、Web App Firewall は HTML クロスサイトスクリプティングチェックに一致するリクエストに対して次の変更を行います。
  - 左山かっこ (<) から HTML 文字エンティティに相当する (<)
  - 右山かっこ（>）から HTML 文字エンティティに相当する（>）

これにより、ブラウザが <script>などの安全でない html タグを解釈して、悪意のあるコードを実行することがなくなります。リクエストヘッダーのチェックと変換の両方を有効にすると、リクエストヘッダーで見つかった特殊文字も変更されます。保護された Web サイトのスクリプトにクロスサイトスクリプティング機能が含まれているが、Web サイトがそれらのスクリプトに依存して正しく動作しない場合は、ブロックを安全に無効にし、変換を有効にすることができます。この設定により、正当な Web トラフィックがブロックされなくなり、クロスサイトスクリプティング攻撃が阻止されます。

- クロスサイトスクリプティングの完全な **URL** を確認してください。完全な URL のチェックが有効になっている場合、Web App Firewall は、URL のクエリ部分だけをチェックするのではなく、URL 全体を調べて HTML クロスサイトスクリプティング攻撃を検出します。
- リクエストヘッダーを確認してください。リクエストヘッダーチェックが有効になっている場合、Web App Firewall は、URL だけでなく、HTML クロスサイトスクリプティング攻撃のリクエストのヘッダーを調べます。GUI を使用する場合は、Web App Firewall プロファイルの [設定] タブでこのパラメータを有効にできます。
- **InspectQueryContentTypes**。リクエストクエリ検査が構成されている場合、App Firewall は、特定のコンテンツタイプに対するクロスサイトスクリプティング攻撃のリクエストのクエリを検査します。GUI を使用する場合は、App Firewall プロファイルの [設定] タブでこのパラメータを設定できます。

> 重要:
>
> ストリーミングの変更の一環として、クロスサイトスクリプティングタグの Web App Firewall 処理が変更されました。この変更は、11.0 以降のビルドに適用されます。この変更は、要求側のストリーミングをサポートする 10.5.e の拡張ビルドにも関係します。以前のリリースでは、開かっこ (<), or close bracket (>) または開かっこ (<>) の両方が存在する場合は、クロスサイトスクリプティング違反のフラグが付けられました。この動作は、要求側ストリーミングのサポートを含むビルドで変更されました。閉じ括弧文字（>）だけが攻撃と見なされなくなりました。開き角かっこ (<) が存在しても要求はブロックされ、攻撃と見なされます。クロスサイトスクリプティング攻撃にフラグが付けられます。

クロスサイトスクリプティングきめ細かなリラックス

Web App Firewall には、特定のフォームフィールド、ヘッダー、または Cookie をクロスサイトスクリプティングインスペクションのチェックから除外するオプションがあります。緩和規則を設定することで、これらのフィールドの 1 つ以上の検査を完全にバイパスできます。

Web App Firewall では、緩和ルールを微調整することで、より厳格なセキュリティを実装できます。アプリケーションでは、特定のパターンを許可する柔軟性が要求される場合がありますが、セキュリティインスペクションをバイパスするように緩和規則を設定すると、ターゲットフィールドがクロスサイトスクリプティング攻撃パターンの検査から除外されるため、アプリケーションが攻撃に対して脆弱になる可能性があります。クロスサイトスクリプティングのきめ細かなリラクゼーションでは、特定の属性、タグ、およびパターンを許可するオプションが提供されます。残りの属性、タグ、パターンはブロックされます。たとえば、Web App Firewall には、現在 125 を超える拒否パターンの既定のセットがあります。ハッカーはクロスサイトスクリプト攻撃でこれらのパターンを使用できるため、Web App Firewall は潜在的な脅威としてフラグを立てます。特定の場所に対して安全と見なされる 1 つまたは複数のパターンをリラックスできます。潜在的に危険なクロスサイトスクリプティングの残りのパターンは、ターゲットの場所に対して引き続きチェックされ、セキュリティチェック違反がトリガーされ続けます。これで、より厳密な制御が可能になりました。

緩和で使用されるコマンドには、**[**値のタイプ**]** と **[**値の式] のオプションのパラメータがあります。値のタイプは空白のままにすることも、[タグ]、[属性]、または [パターン] を選択することもできます。値タイプを空白のままにすると、指定した URL の設定済みフィールドはクロスサイトスクリプティングチェック検査から除外されます。値タイプを選択する場合は、値式を指定する必要があります。値式が正規表現かリテラル文字列かを指定できます。入力が許可リストと拒否リストと照合されると、緩和ルールで設定された指定された式のみが除外されます。

Web App Firewall には、次のクロスサイトスクリプティングが組み込まれています。

1. クロスサイトスクリプティング許可される属性: **abbr**、**accesskey**、**align**、**alt**、**axis**、**bgcolor**、**border**、セルのパディング、セルの間隔、**char**、**charoff**、**charset** など、52 のデフォルトで許可される属性があります。
2. クロスサイトスクリプティング許可されたタグ: **address**、**basefont**、**bgsound**、**big**、**blockquote**、**bg**、**br**、**caption**、**center**、**cite**、**dd**、**del** など 47 のデフォルト許可のタグがありますが、
3. クロスサイトスクリプティング拒否されたパターン: **FSCommand**、**javascript**:、**onAbort**、**onActivate** などの 129 のデフォルトの拒否パターンがあります

> 警告
>
> Web App Firewall アクション URL は正規表現です。HTML クロスサイトスクリプティング緩和ルールを設定する場合、**Name** および **Value Expression** をリテラルまたは RegEx に指定できます。正規表現は強力です。特に、PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。例外として追加するルールを正確に定義し、それ以外は何も定義していないことを確認します。ワイルドカード、特にドットとアスタリスク (.*) メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、望ましくない結果が生じることがあります。たとえば、ブロックする意図がない Web コンテンツへのアクセスをブロックしたり、HTML クロスサイトスクリプティングチェックでブロックされた攻撃を許可したりするなどです。

考慮すべきポイント**:**

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- フィールド名は、複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。クロスサイトスクリプティングの値の種類は、1) タグ、2) 属性、3) パターンです。
- フィールド名/URL の組み合わせごとに複数の緩和ルールを持つことができます
- フォームフィールド名およびアクション URL では、大文字と小文字は区別されません。

コマンドラインを使用した **HTML** クロスサイトスクリプティングチェックの設定

コマンドラインを使用して HTML クロスサイトスクリプティングチェックアクションおよびその他のパラメーターを構成するには

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して HTML クロスサイトスクリプティングチェックを設定できます。

- ページの下部にあるappfw プロファイル「パラメータの説明」を設定します 。)
- `<name> -crossSiteScriptingAction (([block] [learn] [log] [stats])| [** none**])`
- ページの下部にあるappfw プロファイル「パラメータの説明」を設定します 。)
- `<name> **-crossSiteScriptingTransformUnsafeHTML** (ON | OFF)`
- ページの下部に表示されるappfw プロファイルパラメーターの説明を設定します 。
- `<name> -crossSiteScriptingCheckCompleteURLs (ON | OFF)`
- ページの下部に表示されるappfw プロファイルパラメーターの説明を設定します 。
- `<name> - checkRequestHeaders (ON | OFF)` パラメーターの説明は、ページの下部に記載されています。」
- `<name> - CheckRequestQueryNonHtml (ON | OFF)` パラメーターの説明は、ページの下部に記載されています。」

HTML クロスサイトスクリプティングを構成するには、コマンドラインを使用して緩和ルールを確認します

バインドを追加または削除するには、次のように bind または unbind コマンドを使用します。

- `bind appfw profile <name> -crossSiteScripting <String> [isRegex (REGEX | NOTREGEX)] <formActionURL> [-location <location>] [-valueType (Tag|Attribute|Pattern)[<valueExpression>] [-isValueRegex (REGEX | NOTREGEX)]]`
- `unbind appfw profile <name> -crossSiteScripting <String> <formActionURL> [-location <location>] [-valueType (Tag |Attribute|Pattern)[<valueExpression>]]`

GUI を使用して HTML クロスサイトスクリプティングチェックを構成する

GUI では、アプリケーションに関連付けられたプロファイルのペインで HTML クロスサイトスクリプティングチェックを設定できます。

GUI を使用して HTML クロスサイトスクリプティングチェックを構成または変更するには

1. アプリケーションファイアウォールに移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ウィンドウで、[セキュリティチェック] をクリックします。

セキュリティチェックテーブルには、すべてのセキュリティチェックに対して現在構成されているアクション設定が表示されます。設定には 2 つのオプションがあります。

a. HTML クロスサイトスクリプティングの「ブロック」、「ログ」、「統計」、および「学習」アクションを有効または無効にするには、表のチェックボックスをオンまたはオフにして「**OK**」をクリックし、「保存して閉じる」をクリックします。[セキュリティチェック] ウィンドウを閉じます。

b。このセキュリティチェックのオプションをさらに構成する場合は、[**HTML** クロスサイトスクリプティング] をダブルクリックするか、行を選択して [アクション設定] をクリックして、次のオプションを表示します。

クロスサイトスクリプトの変換：安全でないスクリプトタグを変換します。

クロスサイトスクリプティングの完全な **URL** を確認する：URL のクエリ部分だけをチェックするのではなく、完全な URL でクロスサイトスクリプト違反がないか確認します。

上記の設定のいずれかを変更したら、[**OK**] をクリックして変更を保存し、[セキュリティチェック] テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。[**OK**] をクリックして［セキュリティチェック］セクションで行ったすべての変更を保存し、［保存して閉じる］をクリックして［セキュリティチェック］ウィンドウを閉じます。

チェック要求ヘッダーの設定を有効または無効にするには、**[**詳細設定**]** ウィンドウの [プロファイル設定] をクリックします。[共通設定**]** で、[要求ヘッダーをチェック] チェックボックスをオンまたはオフにします。[**OK**] をクリックします。[プロファイル設定] ペインの右上にある X] アイコンを使用してこのセクションを閉じるか、このプロファイルの構成が完了したら、[完了] をクリックしてアプリケーションファイアウォール > プロファイルに戻ることができます。

チェック要求クエリ非 **HTML** 設定を有効または無効にするには、[詳細設定**]** ウィンドウの [プロファイル設定] をクリックします。[共通設定**]** で、[**HTML** 以外の要求クエリを確認する] チェックボックスをオンまたはオフにしま

す。[**OK**] をクリックします。**[Profile Settings]** ペインの右上にある [X] アイコンを使用してこのセクションを閉じるか、このプロファイルの設定が完了したら、[完了] をクリックして [**App Firewall] > [Profile**] に戻ります。

GUI を使用して HTML クロスサイトスクリプティングの緩和ルールを構成するには

1. アプリケーションファイアウォールに移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定**]** ペインで、[緩和規則] をクリックします。
3. 「リラクゼーション規則」テーブルで、「**HTML** クロスサイトスクリプティング」エントリをダブルクリックするか、エントリを選択して「編集」をクリックします。
4. [**HTML** クロスサイトスクリプティング緩和規則] ダイアログで、緩和規則の [追加]、[編集**]**、**[**削除**]**、**[**有効化]、または [無効化] の操作を実行します。

> 注
>
> 新しいルールを追加する場合、[値の ** タイプフィールド**]** で **[**タグ**]** または **[**属性**]** または **[**パターン**]** オプションを選択しない限り、**[**値の式 **] フィールドは表示されません。

ビジュアライザーを使用して HTML クロスサイトスクリプティング緩和ルールを管理するには

すべての緩和規則をまとめて表示するには、[リラクゼーション規則] テーブルの [**HTML** クロスサイトスクリプティング] 行をハイライト表示し、[ビジュアライザー] をクリックします。展開されたリラクゼーションのビジュアライザーには、新しいルールを 追加するか、既存のルールを 編集するオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効または 無効にすることもできます。

GUI を使用してクロスサイトスクリプティングパターンを表示またはカスタマイズするには

GUI を使用して、クロスサイトスクリプティングが許可される属性または許可されるタグのデフォルトのリストを表示またはカスタマイズできます。クロスサイトスクリプティング拒否パターンのデフォルトリストを表示またはカスタマイズすることもできます。

既定のリストは、[アプリケーションファイアウォール] > [署名] > [既定の署名] で指定されます。プロファイルに署名オブジェクトをバインドしない場合、Default Signatures オブジェクトで指定されたデフォルトのクロスサイトスクリプティング許可リストと拒否リストが、クロスサイトスクリプティングのセキュリティチェック処理のためにプロファイルで使用されます。デフォルトのシグネチャオブジェクトで指定されているタグ、属性、およびパターンは読み取り専用です。編集や修正はできません。これらを変更または変更する場合は、Default Signatures オブジェクトのコピーを作成して、ユーザー定義シグニチャオブジェクトを作成します。新しい User-defined シグニチャオブジェクトの許可リストまたは拒否リストを変更し、カスタマイズされた許可リストおよび拒否リストを使用するトラフィックを処理しているプロファイルでこのシグニチャオブジェクトを使用します。

1. デフォルトのクロスサイトスクリプティングパターンを表示するには:

a. 「アプリケーションファイアウォール」>「署名」に移動し、「デフォルト署名」を選択して「編集」をクリックします。次に、[管理] をクリックします SQL/cross-site スクリプトパターン。

管理 SQL/cross-site スクリプトパステーブルは、クロスサイトスクリプティングに関連する次の 3 行を示しています。

`xss/allowed/attribute`

`xss/allowed/tag`

`xss/denied/pattern`

b. 行を選択して「要素を管理」をクリックすると、Web App Firewall のクロスサイトスクリプティングチェックで使用される対応する クロスサイトスクリプティング要素（タグ、属性、パターン）が表示されます。

1. クロスサイトスクリプティング要素をカスタマイズするには: ユーザー定義の署名オブジェクトを編集して、許可されたタグ、許可された属性、および拒否されたパターンをカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除したりできます。

a. [アプリケーションファイアウォール] > [署名] に移動し、ターゲットの [ユーザー定義署名] を選択し、[編集] をクリックします。[管理] をクリックします SQL/cross-site 管理を表示するためのスクリプトパターン SQL/cross-site スクリプトパステーブル。

b. ターゲットのクロスサイトスクリプティング行を選択します。

i. ［要素の管理］をクリックして、対応するクロスサイトスクリプティング要素を［追加］、［編集］、または［削除］します。

ii. [削除] をクリックして、選択した行を削除します。

> 警告:
>
> デフォルトのクロスサイトスクリプティング要素を削除または変更する前、またはクロスサイトスクリプティングパスを削除して行全体を削除する前に、注意する必要があります。シグニチャルールとクロスサイトスクリプティングセキュリティチェックは、アプリケーションを保護するための攻撃を検出するためにこれらの要素に依存しています。クロスサイトスクリプティング要素をカスタマイズすると、編集中に必要なパターンが削除された場合、アプリケーションがクロスサイトスクリプティング攻撃に対して脆弱になる可能性があります。

HTML クロスサイトスクリプティングチェックでの **Learn** 機能の使用

「学習」アクションが有効になっている場合、Citrix Web App Firewall 学習エンジンはトラフィックを監視し、クロスサイトスクリプティング URL 違反を学習します。クロスサイトスクリプティング URL ルールを定期的に検査し、誤検知のシナリオがないかどうかを展開できます。

> 注:
>
> クラスター構成では、クロスサイトスクリプティング URL ルールを展開するには、すべてのノードが同じバージョンである必要があります。

HTML クロスサイトスクリプティング学習の機能強化。Web App Firewall の学習機能拡張は、Citrix ADC ソフトウェアのリリース 11.0 で導入されました。きめ細かな HTML クロスサイトスクリプティングリラクゼーションをデプロイするために、Web App Firewall はきめ細かな HTML クロスサイトスクリプティング学習を提供します。学習エンジンは、観測された値のタイプ（タグ、属性、パターン）および入力フィールドで観測された対応する値式に関する推奨事項を作成します。ブロックされたリクエストをチェックして、現在のルールが制限が厳しく、緩和する

必要があるかどうかを判断するだけでなく、学習エンジンによって生成されたルールを確認して、違反を引き起こしている値タイプと値式を判断し、で対処する必要があります。リラクゼーションルール。

> 注:
>
> Web App Firewall の学習エンジンは、名前の最初の 128 バイトだけを区別できます。フォームに、最初の 128 バイトに一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別できないことがあります。同様に、展開された緩和ルールは、そのようなフィールドを HTML クロスサイトスクリプティングインスペクションから誤って緩和することがあります。
> ヒント
>
> 12 文字を超えるクロスサイトスクリプティングタグは、正しく学習またはログに記録されません。
>
> 学習のためにより長いタグ長が必要な場合は、表示されない大きなタグをに追加できます。AS_cross-site scripting_ALLOWED_TAGS_LIST 長さ 'x' の場合。**

コマンドラインインターフェイスを使用して学習データを表示または使用するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `show appfw learningdata <profilename> crossSiteScripting`
- `rm appfw learningdata <profilename> -crossSiteScripting <string> <formActionURL> [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> **crossSiteScripting*`

GUI を使用して学習したデータを表示または使用するには

1. アプリケーションファイアウォールに移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ウィンドウで、[学習ルール] をクリックします。[学習済みルール] テーブルで [**HTML** クロスサイトスクリプティング] エントリを選択し、ダブルクリックすると、学習済みルールにアクセスできます。このテーブルには、[フィールド名]、[アクション **URL**]、[値のタイプ]、[値]、および [ヒット数] 列が表示されます。学習したルールを展開するか、緩和ルールとして展開する前にルールを編集できます。ルールを破棄するには、ルールを選択して [スキップ] ボタンをクリックします。一度に編集できるルールは 1 つだけですが、複数のルールを選択して展開またはスキップできます。

また、「学習ルール」(Learned Rules) テーブルで「HTML クロスサイトスクリプティング」(**HTML Cross-Site Scripting**) エントリを選択し、「ビジュアライザ」(**Visualizer**) をクリックして、学習したすべての違反の統合ビューを表示することもできます。ビジュアライザを使用すると、学習したルールを簡単に管理できます。データの包括的なビューを 1 つの画面に表示し、1 回のクリックでルールのグループに対するアクションを簡単に実行できます。ビジュアライザーの最大の利点は、複数のルールを統合するために正規表現を推奨することです。区切り文字とアクション URL に基づいて、これらのルールのサブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 のルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。

HTML クロスサイトスクリプティングチェックでログ機能を使用する

ログアクションを有効にすると、HTML クロスサイトスクリプティングのセキュリティチェック違反が監査ログに次のように記録されます。APPFW_cross-site スクリプト違反。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、/var/log/ フォルダー内の ns.logs を末尾に移動し、HTML クロスサイトスクリプティング違反に関連するログメッセージにアクセスします。

```
Shell
tail -f /var/log/ns.log | grep APPFW_cross-site scripting
```

CEF ログ形式のクロスサイトスクリプティングセキュリティチェック違反ログメッセージの例:

```
1 Jul 11 00:45:51 <local0.info> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
    .0|APPFW|\*\*APPFW_cross-site scripting\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=4840 method=GET request=http://aaron.
    stratum8.net/FFC/CreditCardMind.html?abc=%3Cdef%3E msg=\*\*Cross-
    site script check failed for field abc=”Bad tag: def”\*\* cn1=133
    cn2=294 cs1=pr_ffc cs2=PPE1 cs3=eUljypvLa0BbabwfGVE52Sewg9U0001 cs4=
    ALERT cs5=2015 act=\*\*not blocked\*\*
2 <!--NeedCopy-->
```

トランスフォームアクションを示すネイティブログ形式のクロスサイトスクリプティングセキュリティチェック違反ログメッセージの例

```
1 Jul 11 01:00:28 <local0.info> 10.217.31.98 07/11/2015:01:00:28 GMT ns
    0-PPE-0 : default APPFW \*\*APPFW_cross-site scripting\*\* 132 0 :
    10.217.253.62 392-PPE0 eUljypvLa0BbabwfGVE52Sewg9U0001 pr_ffc http:
    //aaron.stratum8.net/FFC/login.php?login_name=%3CBOB%3E&passwd=&
    drinking_pref=on &text_area=&loginButton=ClickToLogin&as_sfid=
    AAAAAAVFqmYL68IGvkrcn2pzehjfIkm5E6EZ9FL8YLvIW_41AvAATuKYe9N7uGThSpEAxbb0iBx55j
    -FC4llF \*\*Cross-site script special characters seen in fields <
    transformed>\*\*
2 <!--NeedCopy-->
```

GUI を使用してログメッセージにアクセスする

Citrix GUI には、ログメッセージを分析するための便利なツール（Syslog Viewer）が含まれています。Syslog ビューアには、次の複数のオプションがあります。

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。[**HTML** クロスサイトスクリプティング] 行をハイライト表示し、[ログ] をクリックします。プロファイルの HTML クロスサイトスクリプティングチェックから直接ログにアクセスすると、GUI によってログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
- 「**Citrix ADC** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティー検査違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- [アプリケーションファイアウォール] > [ポリシー] > [監査] に移動します。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。**HTML** クロスサイトスクリプティングチェックのログメッセージを選択するには、モジュールのドロップダウンリストオプションで **APPFW** を選択してフィルタリングします。**[Event Type]** リストには、選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、APPFW_cross-site [スクリプト] チェックボックスをオンにして [適用] ボタンをクリックすると、HTML クロスサイトスクリプティングのセキュリティチェック違反に関連するログメッセージのみが Syslog ビューアに表示されます。

特定のログメッセージの行にカーソルを置くと、ログメッセージの下に、モジュール、イベントタイプ、イベント **ID**、クライアント **IP** などの複数のオプションが表示されます。これらのオプションのいずれかを選択して、ログメッセージ内の対応する情報を強調表示することができます。

[クリックして展開] 機能は、GUI でのみ使用できます。Syslog Viewer を使用すると、ログを表示するだけでなく、Web App Firewall セキュリティチェック違反のログメッセージに基づいて HTML クロスサイトスクリプティング緩和ルールを展開することもできます。この操作では、ログメッセージは CEF ログ形式である必要があります。クリックして展開機能は、ブロック (またはブロックしない) アクションによって生成されたログメッセージに対してのみ使用できます。変換操作に関するログメッセージのリラクゼーションルールは展開できません。

Syslog Viewer から緩和ルールを展開するには、ログメッセージを選択します。選択した行の [Syslog Viewer] ボックスの右上隅にチェックボックスが表示されます。チェックボックスをオンにし、[アクション] リストからオプションを選択して、緩和ルールを展開します。**[**編集と配備]、[配備]、[すべて配備] は、**[**アクション**]** オプションとして利用できます。

[クリックして展開する] オプションを使用して展開される HTML クロスサイトスクリプティングルールには、細粒緩和の推奨事項は含まれません。

GUI を使用してクリックしてデプロイする機能を構成する

1. Syslog ビューアで、モジュールオプションで **APPFW** を選択します。
2. を選択 APP_cross-site 対応するログメッセージをフィルタリングするための イベントタイプとしての スクリプト。

3. 展開するルールを識別するには、チェックボックスをオンにします。
4. 緩和ルールを展開するには、オプションの [**Action**] ドロップダウンリストを使用します。
5. 対応する緩和規則セクションに規則が表示されていることを確認します。

HTML クロスサイトスクリプティング違反の統計

stats アクションが有効な場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、HTML クロスサイトスクリプティングチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロックアクションが有効になっている場合、3 つの HTML クロスサイトスクリプティング違反を含むページのリクエストでは、最初の違反が検出されるとすぐにページがブロックされるため、stats カウンタが 1 つ増えます。ただし、ブロックが無効になっている場合、同じ要求を処理すると、違反とログの統計カウンタが 3 ずつ増加します。これは、違反ごとに個別のログメッセージが生成されるためです。

コマンドラインを使用して HTML クロスサイトスクリプティングチェック統計を表示するには

コマンドプロンプトで入力します。

```
> sh appfw stats
```

特定のプロファイルの統計情報を表示するには、次のコマンドを使用します。

```
> **stat appfw profile** <profile name>
```

GUI を使用して **HTML** クロスサイトスクリプティング統計を表示する

1. [セキュリティ] > [アプリケーションファイアウォール] **>** [プロファイル] **>** [統計] に移動します。
2. 右側のペインで、[統計リンク] にアクセスします。
3. スクロールバーを使用して、HTML クロスサイトスクリプティング違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

- **HTML** クロスサイトスクリプティング攻撃保護の組み込みサポート—Citrix Web App Firewall は、許可された属性とタグの組み合わせ、および受信したペイロードの拒否されたパターンを監視することにより、クロスサイトスクリプティング攻撃から保護します。クロスサイトスクリプティングチェックで使用される組み込みのデフォルト許可タグ、許可された属性、拒否されたパターンはすべて、/netscaler/default_custom_settings.xml ファイルで指定されています。
- **Customizatio**n：タグ、属性、およびパターンのデフォルトリストを変更して、アプリケーションの特定のニーズに合わせてクロスサイトスクリプティングセキュリティチェック検査をカスタマイズできます。デフォルトのシグニチャオブジェクトのコピーを作成するか、既存のエントリを変更するか、新しいシグニチャオブジェクトを追加します。カスタマイズした設定を使用するには、このシグニチャオブジェクトをプロファイルにバインドします。

- ハイブリッドセキュリティモデル-シグニチャと詳細なセキュリティ保護の両方で、SQL/cross-site プロファイルにバインドされている署名オブジェクトで指定されたスクリプトパターン。署名オブジェクトがプロファイルにバインドされていない場合、SQL/cross-site デフォルトの署名オブジェクトに存在するスクリプトパターンが使用されます。
- **Transform**：変換操作について、次の点に注意してください。

変換操作は、他のクロスサイトスクリプティングアクション設定とは独立して機能します。変換が有効で、ブロック、ログ、統計、学習がすべて無効になっている場合、クロスサイトスクリプティングタグが変換されます。

ブロックアクションが有効になっている場合は、変換アクションよりも優先されます。

- きめの細かいリラクゼーションと学習。緩和ルールを微調整して、セキュリティチェック検査からクロスサイトスクリプティング要素のサブセットを緩和し、残りを検出します。学習エンジンは、観測されたデータに基づいて、特定の値のタイプと値の式を推奨しています。
- **[**クリックして展開]：syslog ビューアで 1 つまたは複数のクロスサイトスクリプティング違反ログメッセージを選択し、緩和ルールとして展開します。
- 文字セット：プロファイルのデフォルトの文字セットは、アプリケーションのニーズに基づいて設定する必要があります。デフォルトでは、プロファイル文字セットは英語 (米国) (ISO-8859-1) に設定されています。指定された文字セットなしで要求を受信した場合、Web App Firewall は ISO-8859-1 であるかのように要求を処理します。開き角かっこ文字（< ）または閉じ角かっこ文字 (>) これらの文字が他の文字セットでエンコードされている場合、クロスサイトスクリプティングタグとして解釈されません。たとえば、リクエストに UTF-8 文字列 “%uff1cscript%uff1e**“**が含まれているが、リクエストページで文字セットが指定されていない場合、プロファイルのデフォルトの文字セットが Unicode として指定されていない限り、クロスサイトスクリプティング違反はトリガーされない可能性があります。

HTML SQL インジェクションチェック

April 25, 2022

多くの Web アプリケーションには、SQL を使用してリレーショナルデータベースサーバーと通信する Web フォームがあります。悪意のあるコードやハッカーが、安全ではない Web フォームを使用して SQL コマンドを Web サーバーに送信する可能性があります。Web App Firewall HTML SQL インジェクションチェックは、セキュリティを侵害する可能性のある不正な SQL コードのインジェクションに対する特別な防御を提供します。Web App Firewall は、ユーザーリクエストで不正な SQL コードを検出すると、リクエストを変換して SQL コードを非アクティブにするか、リクエストをブロックします。Web App Firewall は、1) POST 本文、2) ヘッダー、および 3) Cookie の 3 つの場所に挿入された SQL コードの要求ペイロードを調べます。注入された SQL コードのリクエストのクエリ部分を調べるには、特定のコンテンツタイプに対してアプリケーションファイアウォールプロファイル設定 ‘inspectQueryContentTypes’ を構成してください。

キーワードと特殊文字のデフォルトのセットは、SQL 攻撃の起動に一般的に使用される既知のキーワードと特殊文字を提供します。新しいパターンを追加したり、デフォルトセットを編集して SQL チェックインスペクションをカス

タマイズしたりできます。Web App Firewall には、SQL インジェクション保護を実装するためのさまざまなアクションオプションが用意されています。Web App Firewall プロファイルには、ブロック、ログ、統計、学習の各アクションに加えて、**SQL** 特殊文字を変換して攻撃を無害にするオプションも用意されています。

アクションに加えて、SQL インジェクション処理用に構成できるパラメーターがいくつかあります。**SQL** ワイルドカード文字をチェックできます。SQL インジェクションタイプを変更し、4 つのオプション (**sqlKeyword**、sqlSPLChar**、****sqlSplCharandKeyword**、**sqlSPLCharorKeyword**) のいずれかを選択して、ペイロードの処理時に SQL キーワードと SQL 特殊文字を評価する方法を指定できます。[**SQL Comments Handling]** パラメーターには、SQL インジェクション検出中に検査または除外する必要のあるコメントのタイプを指定するオプションがあります。

リラクゼーションを展開すると、誤検出を回避できます。Web App Firewall 学習エンジンは、緩和ルールの設定に関する推奨事項を提供できます。

アプリケーションに最適化された SQL インジェクション保護を構成するには、次のオプションを使用できます。

Block—入力が SQL インジェクションタイプの仕様と一致する場合にのみブロックアクションがトリガーされます。たとえば、**SQLSplCharANDKeyword** が SQL インジェクションタイプとして設定されている場合、入力で SQL 特殊文字が検出された場合でも、リクエストにキーワードが含まれていなくてもリクエストはブロックされません。SQL インジェクションタイプが **sqlSPLChar** または sqlSPLCharorKeyword** のいずれかに設定されている場合、このようなリクエストはブロックされます。

Log: ログ機能を有効にすると、SQL インジェクションチェックによって実行されるアクションを示すログメッセージが生成されます。ブロックアクションが無効になっている場合、SQL 違反が検出された入力フィールドごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。同様に、SQL 特殊文字が複数のフィールドで変換された場合でも、変換操作に対してリクエストごとに 1 つのログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。

[**Stats**]: 有効にすると、統計機能は違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当なリクエストがブロックされる場合、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを確認するために、構成を再確認しなければならない場合があります。

学習— どの SQL 緩和ルールがアプリケーションに適しているかわからない場合は、学習機能を使用して、学習したデータに基づいて推奨事項を生成できます。Web App Firewall 学習エンジンはトラフィックを監視し、観測された値に基づいて SQL 学習の推奨事項を提供します。パフォーマンスを損なうことなく最適な効果を得るには、学習オプションを短時間有効にしてルールの代表的なサンプルを取得し、ルールを展開して学習を無効にすることをお勧めします。

SQL 特殊文字の変換— Web App Firewall では、一重引用符 (‘)、バックスラッシュ (\ )、セミコロン (;) の 3 つの文字を SQL セキュリティチェック処理の特殊文字と見なします。SQL 変換機能は、HTML リクエストの SQL インジェクションコードを変更して、リクエストが無害になるようにします。変更された HTML リクエストはサーバーに送信されます。デフォルトの変換ルールはすべて /netscaler/default_custom_settings.xml ファイルに指定されています。

変換操作では、リクエストに以下の変更を加えることで、SQL コードが非アクティブになります。

- 一重引用符 (‘) から二重引用符 (「)。
- バックスラッシュ (\ ) をダブルバックスラッシュ (\ ) にします。
- セミコロン (;) は完全に削除されます。

この 3 つの文字 (特殊文字列) は、SQL Server にコマンドを発行するために必要です。SQL コマンドの前に特別な文字列を付けない限り、ほとんどの SQL サーバーではそのコマンドが無視されます。したがって、変換が有効な場合に Web App Firewall が実行する変更により、攻撃者はアクティブな SQL を挿入できなくなります。これらの変更が加えられた後、リクエストは保護された Web サイトに安全に転送されます。保護された Web サイト上の Web フォームが SQL 特殊文字列を正当に含むことができるが、Web フォームが正しく動作するために特殊文字列に依存しない場合、Web App Firewall が保護された Web サイトに提供する保護を低下させずに、ブロックを無効にして変換を有効にして、正当な Web フォームデータのブロックを防止できます。

変換操作は、**SQL** インジェクションタイプ設定とは独立して機能します。変換が有効で、SQL インジェクションタイプが SQL キーワードとして指定されている場合、リクエストにキーワードが含まれていなくても SQL 特殊文字が変換されます。

> ヒント
>
> 通常、変換とブロッキングのどちらかを有効にしますが、両方を有効にすることはできません。ブロックアクションが有効になっている場合は、変換アクションよりも優先されます。ブロッキングを有効にしている場合、変換の有効化は冗長です。

SQL ワイルドカード文字の確認-ワイルドカード文字を使用して、SQL (SQL-SELECT) ステートメントの選択範囲を広げることができます。これらのワイルドカード演算子は、[いいね！] および [**NOT** いいね] 演算子と共に使用して、値を類似の値と比較できます。パーセント (%) およびアンダースコア (_) 文字は、ワイルドカードとしてよく使用されます。パーセント記号は、MS-DOS で使用されるアスタリスク (*) ワイルドカード文字に似ており、フィールド内の 0 文字、1 文字、または複数の文字に一致します。アンダースコアは MS-DOS の疑問符 (?) と似ています。ワイルドカード文字。これは、式の 1 つの数字または文字に一致します。

たとえば、次のクエリを使用して文字列検索を実行し、名前に D 文字が含まれるすべての顧客を検索できます。

「**%D%**」のような名前のカスタマーから ***** を選択してください。:

次の例では、演算子を組み合わせて、2 番目と 3 番目に 0 がある給与値をすべて検索します。

顧客から ***** を選択 **WHERE** 給与 **‘_ 00%’**:

DBMS ベンダーによっては、演算子を追加してワイルドカード文字を拡張しています。Citrix Web App Firewall は、これらのワイルドカード文字を挿入することによって開始される攻撃から保護できます。デフォルトの 5 つのワイルドカード文字は、パーセント (%)、アンダースコア (_)、キャレット (^)、開き角かっこ ([)、閉じ角かっこ (]) です。この保護は、HTML プロファイルと XML プロファイルの両方に適用されます。

デフォルトのワイルドカード文字は、***Default Signatures** で指定されたリテラルのリストです。

- `<wildchar type=” LITERAL” >%</wildchar>`
- `<wildchar type=” LITERAL” >_</wildchar>`

- <wildchar type=" LITERAL" >^</wildchar>
- <wildchar type=" LITERAL" >[</wildchar>
- <wildchar type=" LITERAL" >]</wildchar>

攻撃のワイルドカード文字は [^A-F] のように PCRE になります。Web App Firewall は PCRE ワイルドカードもサポートしていますが、ほとんどの攻撃をブロックするには、上記のリテラルワイルドカード文字で十分です。

> 注:
>
> SQL ワイルドカード文字のチェックは、SQL の特殊文字チェックとは異なります。誤検出を避けるため、このオプションは注意して使用する必要があります。

SQL インジェクションタイプを含むリクエストの確認— Web App Firewall には、アプリケーションの個々のニーズに基づいて、SQL インジェクションインスペクションに必要なレベルの厳格さを実装するための 4 つのオプションが用意されています。SQL 違反を検出するために、リクエストはインジェクションタイプの指定と照合されます。SQL インジェクションタイプには、次の 4 つのオプションがあります。

- **SQL** 特殊文字とキーワード：SQL 違反をトリガーするには、SQL キーワードと SQL 特殊文字の両方を入力に含める必要があります。この最も制限の少ない設定もデフォルト設定です。
- **SQL** 特殊文字-SQL 違反をトリガーするには、入力に少なくとも 1 つの特殊文字が含まれている必要があります。
- **SQL** キーワード：SQL 違反をトリガーするには、指定した SQL キーワードのうち少なくとも 1 つが入力に存在している必要があります。このオプションは十分に考慮せずに選択しないでください。誤検出を避けるため、入力にキーワードが含まれていないことを確認します。
- **SQL** 特殊文字またはキーワード：セキュリティチェック違反をトリガーするには、入力にキーワードまたは特殊文字列が含まれている必要があります。

> ヒント:
>
> SQL 特殊文字を含む入力をチェックするように Web App Firewall を構成すると、Web アプリケーションファイアウォールは特殊文字を含まない Web フォームフィールドをスキップします。ほとんどの SQL サーバーは、前に特殊文字が付いていない SQL コマンドを処理しないため、このオプションを有効にすると、Web App Firewall の負荷を大幅に軽減し、保護された Web サイトを危険にさらすことなく処理を高速化できます。

SQL コメントの処理-デフォルトでは、Web App Firewall はすべての SQL コメントに注入された SQL コマンドをチェックします。ただし、多くの SQL サーバーでは、SQL 特殊文字が前に付いていても、コメント内の内容は無視されます。処理を高速化するために、SQL サーバーがコメントを無視する場合、挿入された SQL のリクエストを調べるときにコメントをスキップするように Web App Firewall を構成できます。SQL コメント処理オプションは次のとおりです。

- **ANSI**—UNIX ベースの SQL データベースで通常使用される ANSI 形式の SQL コメントをスキップします。例:
  - — (2 つのハイフン)-これは、2 つのハイフンで始まり、行末で終わるコメントです。
  - {}-中カッコ（中カッコはコメントを囲みます。{はコメントの前にあり、} はその後に続きます。中括弧は 1 行または複数行のコメントを区切ることができますが、コメントはネストできません)

  - /**/ : C style comments (Does not allow nested comments). Please note /*! <comment that begin with slash followed by asterisk and exclamation mark is not a comment > */
  - MySQL Server は C スタイルのコメントのいくつかのバリアントをサポートしています。これらのコードを使用すると、MySQL 拡張を含むが、移植可能なコードを、次の形式のコメントを使用して記述できます。/*! MySQL-specific code */
  - . #: Mysql コメント: これは # 文字で始まるコメントです。

- ネスト: ネストされた SQL コメントをスキップします。このコメントは、Microsoft SQL Server で通常使用されます。たとえば、; — (2 つのハイフン)、**/* */** (ネストされたコメントを許可します)
-
- **[**すべてのコメントをチェック]: 何もスキップせずに、注入された SQL のリクエスト全体をチェックします。これがデフォルトの設定です。

> ヒント
>
> 通常、バックエンドデータベースが Microsoft SQL Server 上で実行されていない限り、[ネスト] または [ANSI/ネスト] オプションを選択しないでください。他のほとんどの種類の SQL Server ソフトウェアは、ネストされたコメントを認識しません。ネストされたコメントが、別の種類の SQL Server 宛ての要求に表示される場合は、そのサーバーのセキュリティ侵害の試みを示している可能性があります。

[**Check Request headers**]: フォームフィールドの入力の検査に加えて、HTML SQL インジェクション攻撃のリクエストヘッダーを調べる場合は、このオプションを有効にします。GUI を使用する場合は、Web App Firewall プロファイルの **[詳細設定]-> [プロファイル設定]** ペインでこのパラメーターを有効にすることができます。

> 注:
>
> Check Request ヘッダーフラグを有効にすると、**User-Agent** ヘッダーの緩和ルールを設定する必要がある場合があります。のような **SQL** キーワードや、SQL 特殊文字のセミコロン（**;**）が存在すると、偽陽性が発生し、このヘッダーを含む要求がブロックされる可能性があります。
> 警告
>
> リクエストヘッダーのチェックと変換の両方を有効にすると、ヘッダーで見つかった SQL 特殊文字も変換されます。受け入れ、受け入れ文字セット、受け入れエンコーディング、受け入れ言語、期待、およびユーザーエージェントヘッダーは、通常、セミコロン（;）が含まれています。Request ヘッダーのチェックと変換を同時に有効にすると、エラーが発生する場合があります。

inspectQueryContentTypes — 特定のコンテンツタイプに対する SQL インジェクション攻撃のリクエストクエリ部分を調べる場合は、このオプションを設定します。GUI を使用する場合は、App Firewall プロファイルの [詳細設定]-> [プロファイル設定] ペインでこのパラメーターを構成できます。

SQL ファイングレインリラクゼーション

Web App Firewall では、SQL インジェクションインスペクションチェックから特定のフォームフィールド、ヘッダー、または Cookie を除外するオプションがあります。SQL インジェクションチェックのリラクゼーションルールを設定することで、これらのフィールドの 1 つ以上のインスペクションを完全にバイパスできます。

Web App Firewall では、緩和ルールを微調整することで、より厳格なセキュリティを実装できます。アプリケーションでは、特定のパターンを許可する柔軟性が要求される場合がありますが、セキュリティインスペクションをバイパスするように緩和規則を設定すると、ターゲットフィールドが SQL 攻撃パターンの検査から免除されるため、アプリケーションが攻撃に対して脆弱になる可能性があります。SQL のきめ細かい緩和は、特定のパターンを許可し、残りをブロックするオプションを提供します。たとえば、Web App Firewall には現在、100 を超える SQL キーワードのデフォルトセットがあります。ハッカーは SQL Injection 攻撃でこれらのキーワードを使用できるため、Web App Firewall は潜在的な脅威としてフラグを立てます。特定の場所で安全と見なされる 1 つ以上のキーワードをリラックスできます。潜在的に危険な SQL キーワードの残りの部分は、ターゲットの場所がチェックされ、セキュリティチェック違反が引き続きトリガーされます。これで、より厳密な制御が可能になりました。

緩和で使用されるコマンドには、[値タイプ] と **[**** 値式] のオプションパラメータがあります。値式が正規表現かリテラル文字列かを指定できます。値の型は空白のままにすることも、[キーワード]、[SpecialString]、または [WildChar]** を選択することもできます。

> 警告:
>
> 正規表現は強力です。特に PCRE 形式の正規表現に慣れていない場合は、作成した正規表現をすべて再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットアスタリスク（.*）メタ文字またはワイルドカードの組み合わせを不注意に使用すると、ブロックする意図がない Web コンテンツへのアクセスをブロックしたり、HTML SQL インジェクションチェックでブロックされた攻撃を許可するなど、望ましくない結果が生じる可能性があります。

考慮すべきポイント**:**

- 値式はオプションの引数です。フィールド名には値式がない場合があります。
- フィールド名は複数の値式にバインドできます。
- 値式には値型を割り当てる必要があります。SQL 値の型は、1) キーワード、2) SpecialString、または 3) WildChar のいずれかになります。
- フィールド名と URL の組み合わせごとに複数の緩和ルールを設定できます。

コマンドラインを使用した **SQL** インジェクションチェックの設定

コマンドラインを使用して SQL インジェクションアクションとその他のパラメーターを構成するには、次の手順を実行します。

コマンドラインインターフェイスでは、**set appfw profile** コマンドまたは **add appfw profile** コマンドのいずれかを使用して、SQL インジェクション保護を設定できます。ブロック、学習、ログ、統計アクションを有効にし、SQL インジェクション攻撃文字列で使用される特殊文字を変換して攻撃を無効にするかどうかを指定できます。ペイロードで検出する SQL 攻撃パターンの種類 (キーワード、ワイルドカード文字、特殊文字列) を選択し、Web App

Firewall で SQL インジェクション違反のリクエストヘッダーも検査するかどうかを指定します。**unset appfw profile** コマンドを使用して、構成した設定をデフォルトに戻します。次のコマンドはそれぞれ 1 つのパラメータのみを設定しますが、1 つのコマンドに複数のパラメータを含めることができます。

- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -SQLInjectionAction (([block] [learn] [log] [stats])| [none])
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -SQLInjectionTransformSpecialChars (**ON** | OFF)
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -**SQLInjectionCheckSQLWildChars** (**ON** |**OFF**)
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- **<name> -**SQLInjectionType** ([**SQLKeyword**] | [**SQLSplChar**] | [**SQLSplCharANDKeyword**] | [**SQLSplCharORKeyword**])
- ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -**SQLInjectionParseComments** ([**checkall**] | [**ansi|nested**] | [**ansinested**])
- **ページの下部に表示されるアプリケーションファイアウォールプロファイル「パラメータの説明」を設定します。
- <name> -CheckRequestHeaders (ON | OFF) ページの下部に表示されるパラメータの説明。
- <name> - CheckRequestQueryNonHtml (ON | OFF) ページの下部に表示されるパラメータの説明。

コマンドインターフェイスを使用して SQL インジェクション緩和ルールを設定するには

バインドを追加または削除するには、次のように bind または unbind コマンドを使用します。

- bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX|NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor|SpecialString|Wildchar)[<valueExpression>][-isValueRegex (REGEX | NOTREGEX)]]
- unbind appfw profile <name> -SQLInjection <String> <formActionURL> [-location <location>] [-valueTyp (Keyword|SpecialString|Wildchar)[<valueExpression>]]

> 注:
>
> SQL キーワードと SQL 特殊文字のリストを含むビューシグニチャオブジェクトを表示することで、デフォル

> トのシグニチャファイルのコンテンツから SQL キーワードのリストを検索できます。

GUI を使用した SQL インジェクションのセキュリティー検査の設定

GUI では、アプリケーションに関連付けられたプロファイルのペインで SQL Injection セキュリティー検査を構成できます。

GUI を使用して SQL インジェクションチェックを構成または変更するには

1. [アプリケーションファイアウォール**] > [**プロファイル] に移動し、ターゲットプロファイルを強調表示して [編集] をクリックします。
2. [詳細設定] ウィンドウで、[セキュリティチェック] をクリックします。

セキュリティー検査テーブルには、すべてのセキュリティー検査に対して現在構成されているアクション設定が表示されます。設定には次の 2 つのオプションがあります。

a. HTML SQL インジェクションの [ブロック]、[ログ]、[統計]、および [学習] の各アクションを有効または無効にするには、テーブルのチェックボックスをオンまたはオフにして、[**OK**] をクリックし、[保存して閉じる] をクリックして [セキュリティチェック] ウィンドウを閉じます。

b. このセキュリティ・チェックのその他のオプションを構成する場合は、「HTML SQL インジェクション」をダブルクリックするか、行を選択して「アクションの設定」をクリックして、次のオプションを表示します。

SQL 特殊文字の変換：要求内の任意の SQL 特殊文字を変換します。

SQL ワイルドカード文字の確認：ペイロード内の SQL ワイルドカード文字を攻撃パターンと見なします。

「次を含むリクエストのチェック」— チェックする SQL インジェクションのタイプ (sqlKeyword、sqlSPLChar、sqlSPLCharandKeyword、または sqlSPLCharorKeyword)。

SQL コメントの処理-チェックするコメントのタイプ ([すべてのコメントをチェック]、[ANSI]、[ネスト]、または [ANSI/ネスト])。

上記の設定のいずれかを変更したら、「**OK**」をクリックして変更内容を保存し、「セキュリティチェック」(Security Checks) テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。[**OK**］をクリックして［セキュリティチェック］セクションで行った変更をすべて保存し、［保存して閉じる］をクリックして［セキュリティチェック］ウィンドウを閉じます。

GUI を使用して SQL インジェクション緩和ルールを構成するには

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを強調表示して [編集] をクリックします。
- [詳細設定] ウィンドウで、[緩和規則] をクリックします。
- 「緩和規則」テーブルで、「**HTML SQL Injection** 」エントリをダブルクリックするか、エントリを選択して「編集」をクリックします。
- 「**HTML SQL** インジェクション緩和規則」ダイアログで、緩和規則の「追加」、「編集」、「削除」、「有効化」、または「無効化」

> 注
>
> 新しいルールを追加すると、[値の型フィールド] で [キーワード] または [SpecialString] または [WildChar] オプションを選択しない限り、[値の ** 式 **] フィールドは表示されません。

ビジュアライザーを使用して SQL インジェクション緩和ルールを管理するには

すべての緩和ルールをまとめて表示するには、[**HTML SQL Injection**] 行をハイライト表示して [ビジュアライザー] をクリックします。デプロイされたリラクゼーションのビジュアライザーには、[新しいルールを追加] または **[**既存のルールを編集**]** のオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効または 無効にすることもできます。

GUI を使用して射出パターンを表示またはカスタマイズする

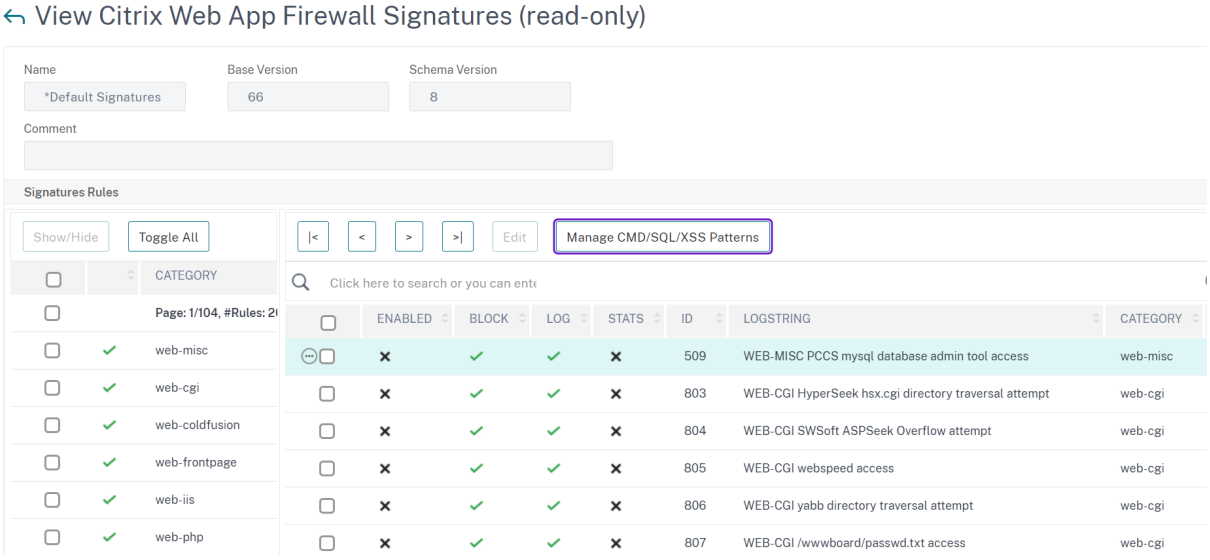
GUI を使用して、射出パターンを表示またはカスタマイズできます。

デフォルトの SQL パターンは、デフォルトのシグニチャファイルで指定されます。シグニチャオブジェクトをプロファイルにバインドしない場合、デフォルトシグニチャオブジェクトで指定されたデフォルトのインジェクションパターンが、コマンドインジェクションセキュリティチェック処理のためにプロファイルによって使用されます。デフォルトのシグニチャオブジェクトで指定されている規則とパターンは読み取り専用です。編集や修正はできません。これらのパターンを変更または変更する場合は、デフォルトの SSignatures オブジェクトのコピーを作成して、ユーザー定義署名オブジェクトを作成します。新しいユーザ定義シグニチャオブジェクトのコマンドインジェクションパターンを変更し、これらのカスタマイズされたパターンを使用するトラフィックを処理しているプロファイルでこのシグニチャオブジェクトを使用します。

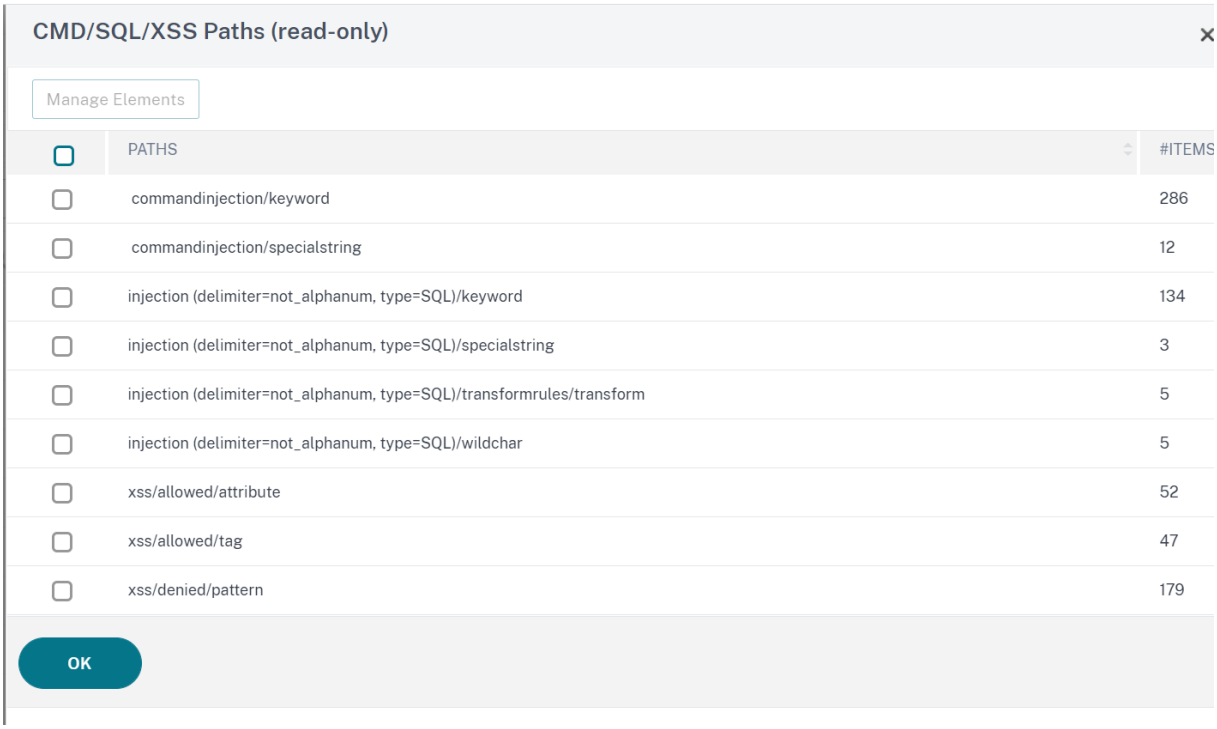
詳細については、「署名」を参照してください。

GUI を使用してデフォルトの射出パターンを表示するには、次の手順を実行します。

1. [アプリケーションファイアウォール**] > [**署名] に移動し、[***** デフォルトシグニチャ**]** を選択し、[編集] をクリックします。



1. [**CMD/SQL/XSS** パターンの管理] をクリックします。「**SQL/**クロスサイトスクリプティングパスの管理」テーブルには、CMD/SQL/XS インジェクションに関連するパターンが表示されます。



1. 行を選択し、[要素の管理] をクリックして、Web App Firewall コマンドインジェクションチェックで使用される対応する注入パターン (キーワード、特殊文字列、変換ルール、またはワイルドカード文字) を表示します。

SQL インジェクションチェックでの学習機能の使用

学習アクションが有効になると、Web App Firewall 学習エンジンはトラフィックを監視し、トリガーされた違反を学習します。学習したルールは定期的に検査できます。十分に検討した後、学習したルールを SQL インジェクション緩和ルールとして展開できます。

SQL インジェクション学習の拡張：Citrix ADC ソフトウェアのリリース 11.0 で、Web App Firewall の学習拡張機能が導入されました。細かい SQL インジェクション緩和をデプロイするために、Web App Firewall はきめ細かい SQL インジェクション学習を提供します。学習エンジンは、観測された値のタイプ（keyword、SpecialString、Wildchar）および入力フィールドで観測された対応する値式に関する推奨事項を作成します。ブロックされたリクエストをチェックして、現在のルールが制限が厳しく、緩和する必要があるかどうかを判断するだけでなく、学習エンジンによって生成されたルールを確認して、違反を引き起こしている値タイプと値式を判断し、で対処する必要があります。リラクゼーションルール。

> 重要
>
> Web App Firewall の学習エンジンでは、名前の最初の 128 バイトしか区別できません。フォームに、最初の 128 バイトに一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別できないことがあります。同様に、デプロイされた緩和ルールは、そのようなフィールドを SQL インジェクション検査から誤って緩

> 和する可能性があります。
>
> （注）User-Agent ヘッダーの SQL チェックインをバイパスするには、次の緩和ルールを使用します。
>
> ```
> bind appfw profile your_profile_name -SQLInjection User-Agent ".*"-
> location HEADER
> ```

コマンドラインインターフェイスを使用して学習データを表示または使用するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `show appfw learningdata <profilename> SQLInjection`
- `rm appfw learningdata <profilename> -SQLInjection <string> <formActionURL > [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> SQLInjection`

GUI を使用して学習済みデータを表示または使用するには

1. [アプリケーションファイアウォール**] > [**プロファイル] に移動し、ターゲットプロファイルを強調表示して [編集] をクリックします。
2. [詳細設定] ペインで、[学習済みルール] をクリックします。「学習済みルール」テーブルで **HTML SQL Injection** エントリを選択してダブルクリックすると、学習したルールにアクセスできます。学習したルールを展開したり、緩和ルールとして展開する前にルールを編集したりできます。ルールを破棄するには、ルールを選択して「スキップ」(**Skip**) ボタンをクリックします。一度に編集できるルールは 1 つだけですが、展開またはスキップするルールは複数選択できます。

また、「学習済みルール」(Learned Rules) テーブルで「**HTML SQL Injection** 」エントリを選択し、「ビジュアライザー」(**Visualizer**) をクリックして、学習済みのすべての違反の統合ビューを表示することもできます。ビジュアライザを使用すると、学習したルールを簡単に管理できます。1 つの画面でデータの包括的なビューを表示し、1 回のクリックでルールのグループに対するアクションの実行を容易にします。ビジュアライザーの最大の利点は、正規表現を推奨して複数のルールを統合できることです。デリミタとアクション URL に基づいて、これらのルールのサブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 個のルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。

SQL インジェクションチェックでログ機能を使用する

ログアクションを有効にすると、HTML SQL インジェクションのセキュリティチェック違反が **APPFW_SQL** 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて **/var/log/** フォルダ内の **ns.logs** を末尾にして、SQL インジェクション違反に関連するログメッセージにアクセスします。


```
> Shell

## tail -f /var/log/ns.log | grep APPFW_SQL
```

リクエストが変換されたときの HTML SQL インジェクションログメッセージの例

```
Jun 26 21:08:41 <local0.info> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
    .0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=54001
     method=GET request=http://aaron.stratum8.net/FFC/login.php?
    login_name=%27+or&passwd=and+%3B&drinking_pref=on&text_area=select
    +*+from+%5C+%3B&loginButton=ClickToLogin&as_sfid=AAAAAAXjnGN5gLH-
    hvhTOpIySEIqES7BjFRs5Mq0fwPp-3ZHDi5yWlRWByj0cVbMyy-
    Ens2vaaiULKOcUri4OD4kbXWwSY5s7I3QkDsrvIgCYMC9BMvBwY2wbNcSqCwk52lfE0k
    %3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c msg= Special
    characters seen in fields cn1=74 cn2=762 cs1=pr_ffc cs2=PPE1 cs3=9
    ztIlf9p1H7p6Xtzn6NMygTv/QM0002 cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

ポストリクエストがブロックされた場合の HTML SQL インジェクションログメッセージの例

```
Jun 26 21:30:34 <local0.info> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
    .0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=9459
    method=POST request=http://aaron.stratum8.net/FFC/login_post.php msg
    =SQL Keyword check failed for field text_area=”(')” cn1=78 cn2=834
    cs1=pr_ffc cs2=PPE1 cs3=eVJMMPtZ2XgylGrHjkx3rZLfBCI0002 cs4=ALERT
    cs5=2015 act=blocked
<!--NeedCopy-->
```

> 注
>
> 10.5.e ビルド (エンハンスメントビルド) と 11.0 以降のビルドでのストリーミング変更の一環として、入力データをブロック単位で処理するようになりました。RegEx パターンマッチングは、連続した文字列マッチングで 4K に制限されるようになりました。この変更により、SQL 違反ログメッセージには、以前のビルドとは異なる情報が含まれる場合があります。入力のキーワードと特殊文字は、多くのバイトで区切ることができます。データを処理するときに、入力値全体をバッファリングするのではなく、SQL キーワードと特殊文字列を追跡します。ログメッセージには、フィールド名に加えて、SQL キーワード、SQL 特殊文字、または SQL キーワードと SQL 特殊文字の両方が含まれるようになりました。これらの文字は、構成された設定によって決定されます。次の例に示すように、残りの入力はログメッセージに含まれなくなります。
>
> 例:
>
> 10.5 では、Web App Firewall が SQL 違反を検出すると、次に示すように、入力文字列全体がログメッセージに含まれることがあります。

> SQL Keyword check failed for field text=\"select a name from testbed1 ;(;)\".*<blocked>
>
> リクエストサイドストリーミングおよび 11.0 以降のビルドをサポートする 10.5.e の拡張ビルドでは、次に示すように、フィールド名、キーワード、および特殊文字（該当する場合）のみをログメッセージに記録します。
>
> SQL Keyword check failed for field **text="select(;)"<blocked>
>
> この変更は、application/x-www-form-urlencoded、マルチパート/フォームデータ、または text/x-gwt-rpc コンテンツタイプを含むリクエストに適用されます。**JSON** または **XML** ペイロードの処理中に生成されるログメッセージは、この変更の影響を受けません。

GUI を使用してログメッセージにアクセスするには

Citrix GUI には、ログメッセージを分析するための便利なツール（**Syslog Viewer**）が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります。

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。**HTML SQL** インジェクション行を強調表示して、[ログ] をクリックします。プロファイルの HTML SQL インジェクションチェックから直接ログにアクセスすると、GUI によってログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
- 「**Citrix ADC** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。[Audit Messages] セクションで、[**Syslog messages**] リンクをクリックして [Syslog Viewer] を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティチェック違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- アプリケーションファイアウォール > ポリシー > 監査に移動します。[Audit Messages] セクションで、[**Syslog messages**] リンクをクリックして [Syslog Viewer] を表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。**HTML SQL** インジェクションチェックのログメッセージを選択するには、「モジュール」のドロップダウン・リスト・オプションで「**APPFW**」を選択してフィルタリングします。**[Event Type]** リストには、選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、「**APPFW_SQL** 」チェック・ボックスを選択して「適用」ボタンをクリックすると、**SQL Injection** セキュリティ・チェック違反に関するログ・メッセージのみが Syslog Viewer に表示されます。

特定のログメッセージの行にカーソルを置くと、[モジュール]、[イベントタイプ]、[イベント **ID**]、[クライアント **IP**] などの複数のオプションがログメッセージの下に表示されます。これらのオプションを選択すると、ログメッセージ内の対応する情報を強調表示できます。

[クリックして展開] 機能は GUI でのみ使用できます。Syslog Viewer を使用すると、ログを表示するだけでなく、Web App Firewall セキュリティチェック違反のログメッセージに基づいて HTML SQL インジェクション緩和ルールを展開することもできます。この操作では、ログメッセージは CEF ログ形式である必要があります。クリックして展開機能は、ブロック (またはブロックしない) アクションによって生成されたログメッセージに対してのみ使用できます。変換操作に関するログメッセージの緩和ルールは展開できません。

Syslog Viewer から緩和ルールを展開するには、ログメッセージを選択します。選択した行の [**Syslog Viewer**] ボックスの右上隅にチェックボックスが表示されます。このチェックボックスをオンにし、[アクション] リストから緩和ルールを展開するオプションを選択します。**[**** 編集とデプロイ ****]**、**[**** デプロイ**]**、**[**すべてデプロイ] は、******

[Click to Deploy] オプションを使用してデプロイされる SQL インジェクションルールには、細粒度緩和の推奨事項は含まれません。

GUI の **[**クリックしてデプロイ**]** 機能を使用するには、次の手順を実行します。

1. Syslog Viewer で、[モジュール] オプションの [アプリケーションファイアウォール] を選択します。
2. 対応するログメッセージをフィルタするには、イベントタイプとして **APP_SQL** を選択します。
3. 展開するルールを識別するには、このチェックボックスをオンにします。
4. オプションの [アクション（Action）] ドロップダウンリストを使用して、緩和ルールを展開します。
5. ルールが対応する [緩和ルール] セクションに表示されていることを確認します。

SQL インジェクション違反の統計

統計アクションが有効になっている場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、SQL インジェクションチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロック・アクションが有効になっている場合、3 つの SQL インジェクション違反を含むページのリクエストでは、最初の違反が検出されるとすぐにページがブロックされるため、stats カウンタが 1 つずつ増加します。ただし、ブロックが無効になっている場合、同じ要求を処理すると、違反ごとに個別のログメッセージが生成されるため、違反とログの統計カウンタが 3 ずつ増加します。

コマンドラインを使用して **SQL** インジェクション・チェック統計を表示するには、次のステップを実行します。

コマンドプロンプトで入力します。

> **sh appfw** 統計

特定のプロファイルの統計情報を表示するには、以下のコマンドを使用します。

```
> stat appfw profile <profile name>
```

GUI を使用して HTML SQL インジェクションの統計情報を表示するには

1. システム > セキュリティ > アプリケーションファイアウォールに移動します。
2. 右側のペインで、[統計リンク] にアクセスします。
3. スクロールバーを使用して、HTML SQL インジェクション違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

SQL インジェクションチェックについては、次の点に注意してください。

- **SQL** インジェクション保護の組み込みサポート-Citrix Web App Firewall は、フォームパラメータ内の SQL キーワードと特殊文字の組み合わせを監視することで、SQL インジェクションから保護します。すべての SQL キーワード、特殊文字、ワイルドカード文字、およびデフォルトの変換ルールは、/netscaler/default_custom_settings.xml ファイルに指定されています。
- カスタマイズ：デフォルトのキーワード、特殊文字、ワイルドカード文字、および変換ルールを変更して、アプリケーションの特定のニーズに合わせて SQL セキュリティチェックインスペクションをカスタマイズできます。デフォルトのシグニチャオブジェクトのコピーを作成するか、既存のエントリを変更するか、新しいシグニチャオブジェクトを追加します。このシグニチャオブジェクトをプロファイルにバインドして、カスタマイズした設定を利用します。
- ハイブリッドセキュリティモデル-シグニチャとディープセキュリティ保護の両方で、プロファイルにバインドされたシグニチャオブジェクトで指定された SQL/Cross-Site スクリプティングパターンが使用されます。シグニチャオブジェクトがプロファイルにバインドされていない場合は、デフォルトのシグニチャオブジェクトに存在する SQL/Cross-Site スクリプティングパターンが使用されます。
- [**Transform**]：変換操作について、次の点に注意してください。
  - 変換操作は、他の SQL インジェクションアクション設定とは独立して動作します。変換が有効で、ブロック、ログ、統計、学習がすべて無効になっている場合、SQL の特殊文字は変換されます。
  - SQL 変換が有効な場合、SQL 特殊文字が非ブロックモードで変換された後、ユーザー要求がバックエンドサーバーに送信されます。ブロックアクションが有効になっている場合は、変換アクションよりも優先されます。インジェクションタイプが SQL 特殊文字として指定され、ブロックが有効な場合、変換アクションにもかかわらずリクエストはブロックされます。
- [きめ細かな緩和と学習]：緩和ルールを微調整して、SQL 要素のサブセットをセキュリティチェック検査から緩和し、残りは検出します。学習エンジンは、観測されたデータに基づいて、特定の値のタイプと値の式を推奨しています。
- [**Click to Deploy**]：syslog ビューアで 1 つまたは複数の SQL 違反ログメッセージを選択し、緩和ルールとして展開します。

HTML および **JSON** ペイロードに対する **SQL** 文法ベースの保護

October 7, 2021

Citrix Web App Firewall は、HTTPおよびJSONペイロードでの SQL インジェクション攻撃を検出するためにパターン一致アプローチを使用します。このアプローチでは、事前定義された一連のキーワードと（または）特殊文字のセットを使用して攻撃を検出し、違反としてフラグを立てます。このアプローチは効果的ですが、多くの誤検出が発生し、1 つ以上の緩和ルールが追加される可能性があります。特に、HTTP または JSON リクエストで「Select」や「From」などの一般的に使用される単語が使用されている場合。HTMLおよびJSONペイロードの SQL 文法保護チェックを実装することで、誤検知を減らすことができます。

既存のパターンマッチアプローチでは、HTTP リクエストに事前定義されたキーワードまたは特殊文字が存在する場合、SQL インジェクション攻撃が識別されます。この場合、ステートメントは有効な SQL 文である必要はありませ

ん。しかし、文法ベースのアプローチでは、SQL インジェクション攻撃は、キーワードまたは特殊文字が SQL 文に存在するか、SQL 文の一部である場合にのみ検出され、誤検出シナリオが減少します。

SQL 文法ベースの保護使用シナリオ

HTTP リクエストに「チケットを選択してユニオンステーションで会おう」というステートメントを考えてみましょう。ステートメントは有効な SQL ステートメントではありませんが、既存のパターンマッチアプローチは、SQL インジェクション攻撃としてリクエストを検出します。これは、ステートメントが「Select」、「and」、「Union」などのキーワードを使用するためです。ただし、SQL 文法アプローチの場合、キーワードが有効な SQL 文に存在しないか、有効な SQL 文の一部ではないため、ステートメントは違反攻撃として検出されません。

文法ベースのアプローチは、JSONペイロードの SQL インジェクション攻撃を検出するように設定することもできます。緩和ルールを追加するには、既存の緩和ルールを再利用できます。「ValueType」「キーワード」のルールでは、細粒緩和ルールは、SQL 文法にも適用されます。JSON SQL 文法では、既存の URL ベースのメソッドを再利用できます。

CLI を使用して SQL 文法ベースの保護を構成する

SQL 文法ベースの検出を実装するには、Web App Firewall プロファイルで「sqLinjectionGrammer」パラメーターを構成する必要があります。デフォルトでは、パラメータは無効になっています。学習以外の既存の SQL インジェクションアクションはすべてサポートされています。アップグレード後に作成された新しいプロファイルは、SQL インジェクション文法をサポートしており、引き続きデフォルトのタイプを「特殊文字またはキーワード」とし、明示的に有効にする必要があります。

コマンドプロンプトで入力します。

```
1 add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
     SQLInjectionGrammar ON/OFF
2 <!--NeedCopy-->
```

例:

```
add appfw profile profile1 -SQLInjectionAction Block -SQLInjectionGrammar ON
```

CLI を使用して SQL パターンマッチ保護と文法ベースの保護を構成します

文法ベースとパターンマッチの両方のアプローチを有効にした場合、アプライアンスは最初に文法ベースの検出を実行し、アクションタイプをブロックに設定した SQL インジェクション検出がある場合、リクエストはブロックされます（パターンマッチを使用して検出を確認せずに）。

コマンドプロンプトで入力します。


```
add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
    SQLInjectionGrammar ON  - SQLInjectionType <Any action other than  '
    None' : SQLSplCharANDKeyword/ SQLSplCharORKeyword/ SQLSplChar/
    SQLKeyword>
<!--NeedCopy-->
```

例:

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar ON -
SQLInjectionType SQLSplChar
```

CLI を使用して SQL インジェクションチェックを文法ベースの保護のみで構成する

コマンドプロンプトで入力します。

```
add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
    SQLInjectionGrammar ON  - SQLInjectionType None
<!--NeedCopy-->
```

例:

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar ON -
SQLInjectionType None
```

CLI を使用して SQL 文法ベースの保護のための緩和ルールをバインドする

アプリケーションでペイロード内の特定の「ELEMENT」または「ATTRIBUTE」のSQLインジェクションチェックをバイパスする必要がある場合は、緩和ルールを設定する必要があります。

> 注:
>
> ValueType「キーワード」を含む緩和ルールは、アプライアンスがSQL文法を使用して検出を実行する場合にのみ評価されます。

射出検査緩和SQL ルールコマンドには、次の構文があります。コマンドプロンプトで入力します。

```
bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX|
    NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor
    |SpecialString|Wildchar) [<valueExpression>][-isValueRegex (REGEX |
    NOTREGEX) ]]
<!--NeedCopy-->
```


例:

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -isregex
regEX
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
Keyword 'selec[a-z]+' -isvalueRegex regEX
```

CLI を使用して JSON ペイロードの SQL 文法ベースの保護を構成する

JSON ペイロードの SQL 文法ベースの検出を実装するには、Web App Firewall プロファイルで「jsonSQLinjectionGrammer」パラメータを設定する必要があります。デフォルトでは、パラメータは無効になっています。学習以外の既存の SQL インジェクションアクションはすべてサポートされています。アップグレード後に作成された新しいプロファイルは、SQL インジェクション文法をサポートしており、引き続きデフォルトのタイプを「特殊文字またはキーワード」として使用するため、明示的に有効にする必要があります。

コマンドプロンプトで入力します。

```
1 add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
     action-name> -JSONSQLInjectionGrammar ON/OFF
2 <!--NeedCopy-->
```

例:

```
add appfw profile profile1 -type JSON -JSONSQLInjectionAction Block -JSONSQLInjectionG
 ON
```

CLI を使用して SQL パターン一致保護と文法ベースの保護を構成します

文法ベースチェックとパターンマッチチェックの両方を有効にした場合、アプライアンスは最初に文法ベースの検出を実行し、アクションタイプをブロックに設定した SQL インジェクション検出がある場合、リクエストはブロックされます（パターンマッチを使用して検出を確認せずに）。

> 注:
>
> ValueType「キーワード」を持つ緩和ルールは、アプライアンスが SQL 文法を使用して検出を実行する場合にのみ評価されます。

コマンドプロンプトで入力します。


```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType <Any
     action other than  'None' : SQLSplCharANDKeyword/
    SQLSplCharORKeyword/ SQLSplChar/ SQLKeyword>
<!--NeedCopy-->
```

例:

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block - JSONSQLInjectionGramma
 ON -JSONSQLInjectionType SQLSplChar
```

CLI を使用して JSON ペイロードの SQL 文法ベースの保護を構成する

コマンドプロンプトで入力します。

```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType None
    `\
<!--NeedCopy-->
```

例:

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block - JSONSQLInjectionGramma
 ON -JSONSQLInjectionType None
```

CLI を使用して JSON SQL 文法ベースの保護のための URL ベースの緩和ルールをバインドします

アプリケーションでペイロード内の特定の「ELEMENT」または「ATTRIBUTE」のJSONコマンドインジェクションインスペクションをバイパスする必要がある場合は、緩和ルールを設定できます。
射出検査緩和JSONルールコマンドには、次の構文があります。コマンドプロンプトで入力します。

```
bind appfw profile <profile name>  - JSONCMDURL <expression> -comment <
    string> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED ) -state (
    ENABLED | DISABLED )
<!--NeedCopy-->
```

例:

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -isregex
regEX
```

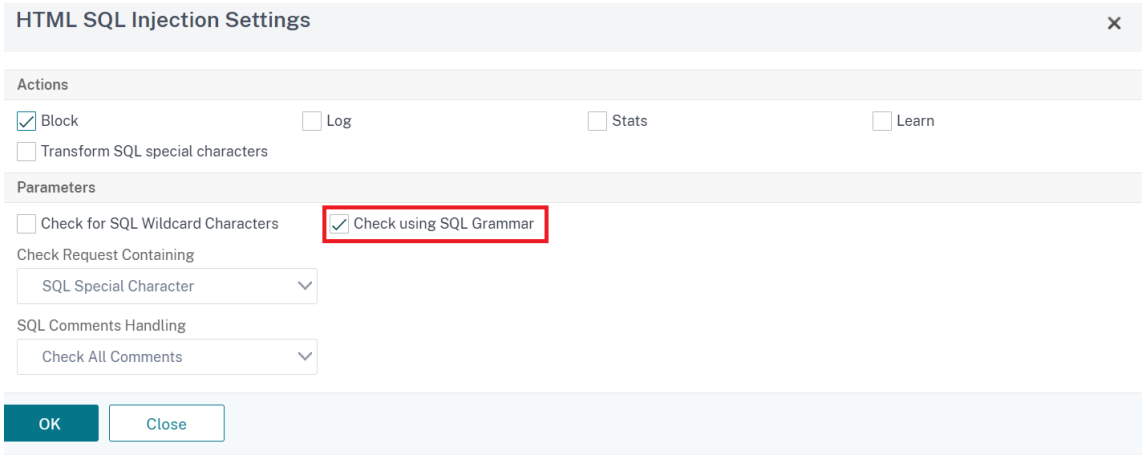


```
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
Keyword 'selec[a-z]+' -isvalueRegex regEX
```

GUI を使用して SQL 文法ベースの保護を構成する

GUI 手順を実行して、文法ベースの HTML SQL インジェクション検出を設定します。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. **Citrix Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで、[**HTML SQL** インジェクションの設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。
6. [アクションの設定] をクリックして、[**HMTL SQL** インジェクション設定] ページにアクセスします。



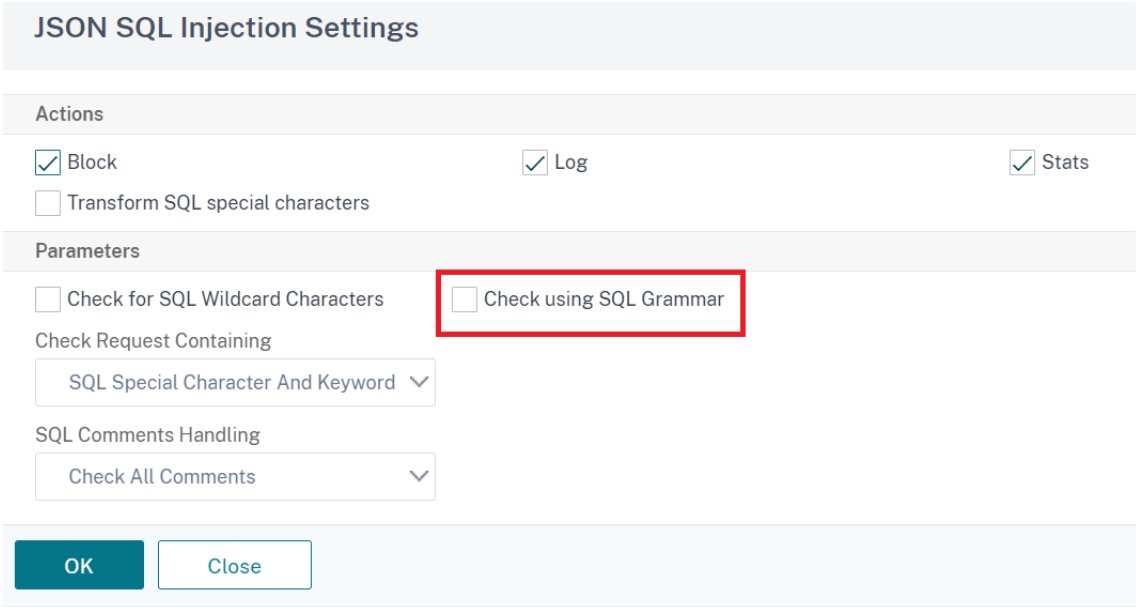
7. [**SQL** 文法を使用してチェックする] チェックボックスをオンにします。
8. [**OK**] をクリックします。

GUI を使用して JSON ペイロードの SQL 文法ベースの保護を構成する

GUI 手順を実行して、文法ベースの JSON SQL インジェクション検出を設定します。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. **Citrix Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティーチェック] セクションで、[**JSON SQL** インジェクション設定] に移動します。

5. チェックボックスの近くにある実行可能アイコンをクリックします。
6. [アクション設定] をクリックして、[**JSON SQL** インジェクション設定] ページにアクセスします。
7. [**SQL** 文法を使用してチェックする] チェックボックスをオンにします。
8. [**OK**] をクリックします。



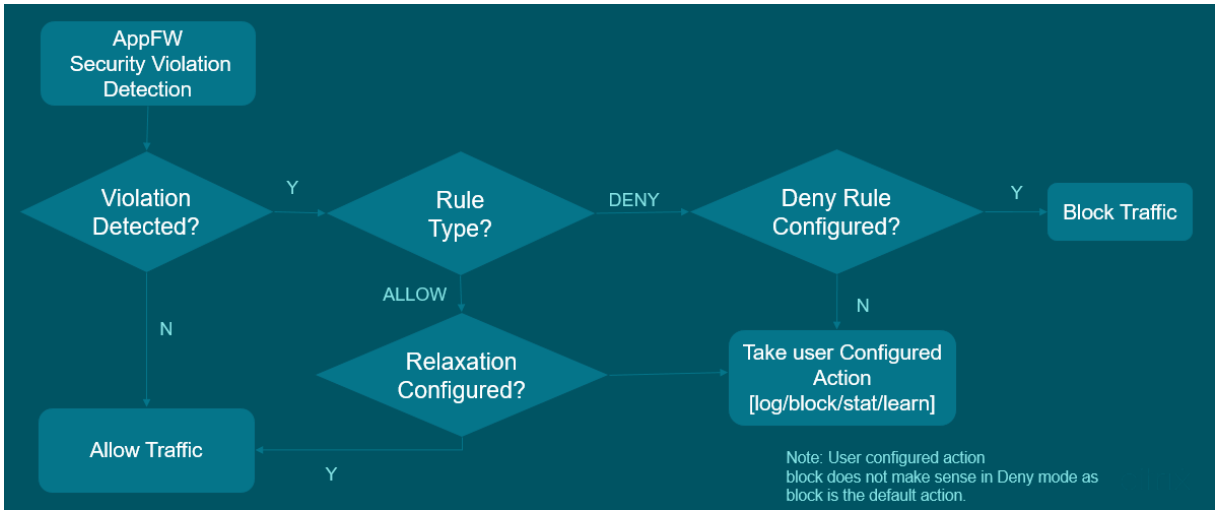
HTML SQL インジェクション攻撃を処理するための緩和ルールと拒否ルール

October 7, 2021

着信トラフィックがある場合、違反検出ロジックはトラフィック違反をチェックします。HTML SQL インジェクション攻撃が検出されない場合、トラフィックは通過できます。ただし、違反が検出された場合、緩和ルール（許可）と拒否ルールによって、違反の処理方法が定義されます。セキュリティチェックが許可モード（デフォルトモード）で設定されている場合、ユーザが緩和ルールまたは許可ルールを明示的に設定していない限り、検出された違反はブロックされます。

許可モードに加えて、セキュリティチェックを拒否モードで構成し、違反の処理に拒否ルールを使用することもできます。このモードでセキュリティチェックが設定されている場合、ユーザーが明示的に拒否ルールを設定している場合、検出された違反はブロックされます。拒否ルールが構成されていない場合は、ユーザー設定アクションが適用されます。

次の図は、動作モードの動作を許可および拒否する方法を説明しています。



1. 違反が検出されると、緩和ルール（許可）および拒否ルールによって、違反の処理方法が定義されます。
2. セキュリティチェックが拒否モードに設定されている場合（許可モードに設定されている場合は、ステップ 5 に進みます）、拒否ルールを明示的に設定していない限り、違反はブロックされます。
3. 違反が拒否ルールと一致する場合、アプライアンスはトラフィックをブロックします。
4. トラフィック違反がルールと一致しない場合、アプライアンスはユーザー定義アクション（ブロック、リセット、またはドロップ）を適用します。
5. セキュリティチェックが許可モードで構成されている場合、Web App Firewall モジュールは、許可ルールが構成されているかどうかを確認します。
6. 違反が許可ルールと一致する場合、アプライアンスはトラフィックをバイパスすることを許可し、それ以外の場合はブロックされます。

セキュリティチェックイン緩和および強制モードを構成する

コマンドプロンプトで入力します。

```
1 set appfw profile <name>   - SQLInjectionAction [block stats learn]  -
     SQLInjectionRuleType [ALLOW DENY]
2 <!--NeedCopy-->
```

例**:**

```
set appfw profile prof1 sqlInjectionAction block -sqlInjectionRuleType
ALLOW DENY
```

緩和ルールと強制ルールを **Web** アプリケーションファイアウォールプロファイルにバインドする

コマンドプロンプトで入力します。


```
bind appfw profile <name> -SQLInjection <string> <formActionURL>
<!--NeedCopy-->
```

例:

```
bind appfw profile p1 -SQLInjection field_f1 ”/login.php”-RuleType ALLOW
bind appfw profile p2 -SQLInjection field_f1 ”/login.php”-RuleType ALLOW
```

HTML コマンドインジェクション保護チェック

October 7, 2021

HTML コマンドインジェクションチェックは、着信トラフィックにシステムセキュリティを壊したり、システムを変更したりする不正なコマンドがあるかどうかを調べます。検出時にトラフィックに悪意のあるコマンドがある場合、アプライアンスは要求をブロックするか、設定済みのアクションを実行します。

Citrix Web App Firewall プロファイルが、コマンドインジェクション攻撃の新しいセキュリティチェックで強化されました。コマンドインジェクションセキュリティチェックによってトラフィックが検査され、悪意のあるコマンドが検出されると、アプライアンスは要求をブロックするか、設定済みのアクションを実行します。

コマンドインジェクション攻撃では、攻撃者は Citrix ADC オペレーティングシステム上で不正なコマンドを実行することを目指しています。これを達成するために、攻撃者は脆弱なアプリケーションを使用してオペレーティングシステムのコマンドを注入します。アプリケーションが安全でないデータ（フォーム、Cookie、またはヘッダー）をシステムシェルに渡した場合、Citrix ADC アプライアンスはインジェクション攻撃に対して脆弱です。

コマンドインジェクション保護のしくみ

1. 着信要求の場合、WAF はトラフィックにキーワードまたは特殊文字があるかどうか調べます。着信要求に、拒否されたキーワードまたは特殊文字のいずれにも一致するパターンがない場合、要求は許可されます。それ以外の場合、要求は設定されたアクションに基づいてブロック、ドロップ、またはリダイレクトされます。
2. リストからキーワードまたは特殊文字を除外する場合は、緩和ルールを適用して、特定の条件下でセキュリティチェックをバイパスできます。
3. ログ記録を有効にして、ログメッセージを生成できます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとした可能性があります。
4. また、統計機能を有効にして、違反およびログに関する統計データを収集することもできます。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要

求がブロックされている場合は、構成を再確認して、新しい緩和ルールを構成する必要があるか、既存のルールを変更する必要があるかを確認する必要があります。

コマンドインジェクションチェックで拒否されたキーワードと特殊文字

コマンドインジェクション攻撃を検出してブロックするために、アプライアンスはデフォルトのシグネチャファイルに定義された一連のパターン（キーワードと特殊文字）を持っています。次に、コマンドインジェクション検出中にブロックされたキーワードのリストを示します。

```
    <commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
…
</commandinjection>
<!--NeedCopy-->
```

シグニチャファイルに定義される特殊文字は次のとおりです。
| ; & $ > < '\ ! >> ##

CLI を使用したコマンドインジェクションチェックの設定

コマンドラインインターフェイスでは、set ザ profile コマンドまたは add ザ profile コマンドを使用して、コマンドインジェクション設定を構成できます。ブロック、ログ、および統計アクションを有効にすることができます。また、ペイロードで検出するキーワードと文字列文字も設定する必要があります。

コマンドプロンプトで入力します。

```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -CMDInjectionType
 <CMDInjectionType>]
```

> 注:
>
> デフォルトでは、コマンドインジェクションアクションは「なし」に設定されています。また、デフォルトのコマンドインジェクションタイプはCmdSplCharANDKeyWordとして設定されています。

例:

```
set appfw profile profile1 -cmdInjectionAction block -CMDInjectionType
CmdSplChar
```

ここで、使用可能なコマンドインジェクションアクションは次のとおりです。

- なし-コマンドインジェクション保護を無効にします。
- Log：セキュリティ・チェックのコマンド・インジェクション違反をログに記録します。

- Block-コマンドインジェクションのセキュリティチェックに違反するトラフィックをブロックします。
- 統計-コマンドインジェクションのセキュリティ違反の統計を生成します。

ここで、使用可能なコマンド射出タイプは次のとおりです。

- Cmd SplChar. 特殊文字をチェックする
- cmdKeyword コマンドインジェクションキーワードをチェックします
- cmdsplcharand キーワードです。特殊文字とコマンドインジェクションをチェックします。キーワードとブロックは、両方が存在する場合のみ。
- cmdsplCharorKeyword。特殊文字とコマンドインジェクションをチェックし、それらのいずれかが見つかった場合はブロックします。

コマンドインジェクション保護チェックのリラクゼーションルールの設定

アプリケーションで、ペイロード内の特定の ELEMment または attribute に対するコマンドインジェクション検査をバイパスする必要がある場合は、緩和規則を設定できます。

射出検査緩和規則コマンドの構文は次のとおりです。

```
bind appfw profile <profile name> -cmdInjection <string> <URL> -isregex <
REGEX/NOTREGEX>
```

ヘッダーの正規表現の緩和ルールの例

```
bind appfw profile sample -CMDInjection hdr ”http://10.10.10.10/”-location
heaDER -valueType Keyword '[a-z]+grep'-isvalueRegex REGEX
```

その結果、インジェクションはコマンドインジェクションチェックを免除し、「grep」のバリアントを含むヘッダーhdrを許可します。

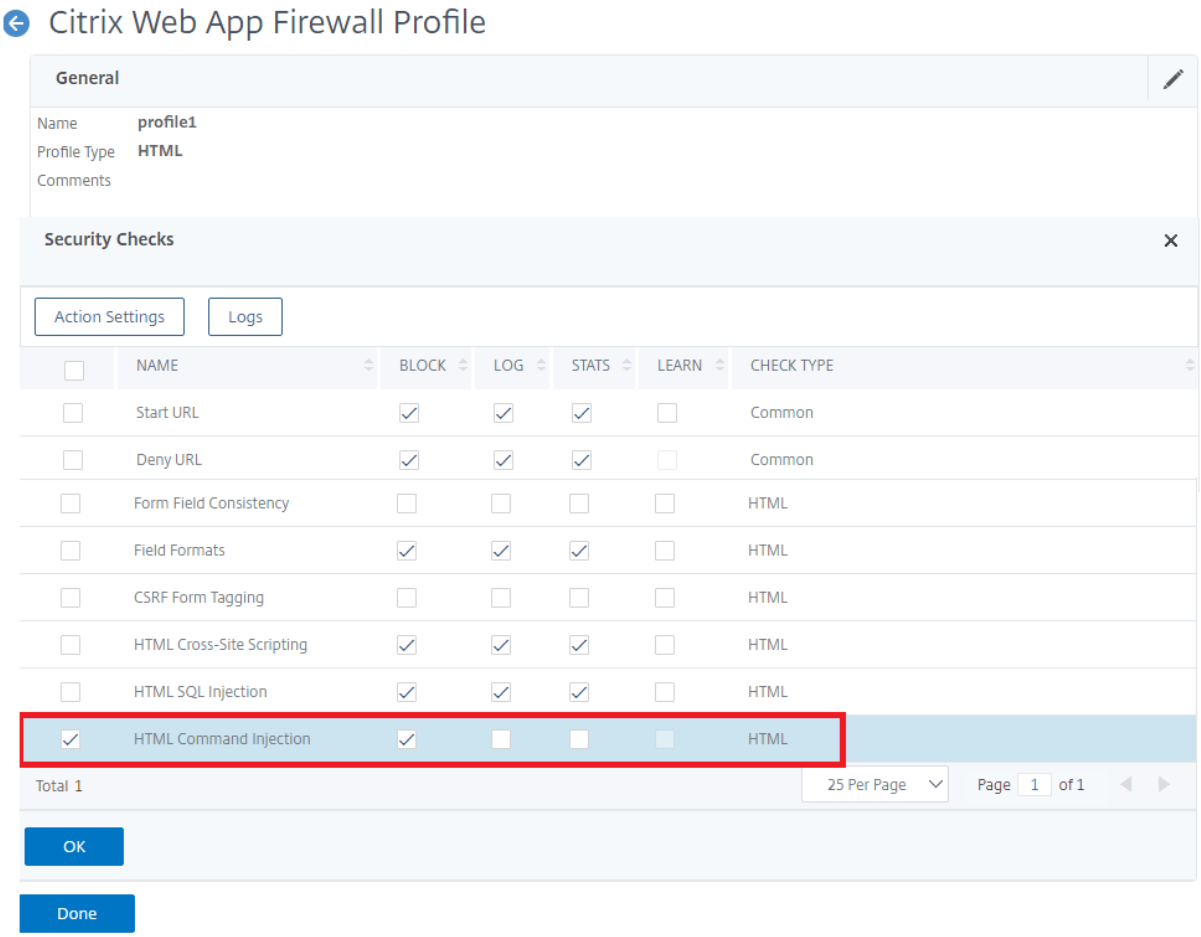
クッキー内の正規表現として **valueType** を持つリラクゼーションルールの例

```
bind appfw profile sample -CMDInjection ck_login ”http://10.10.10.10/”-
location cookie -valueType Keyword 'pkg[a-z]+'-isvalueRegex REGEX
```

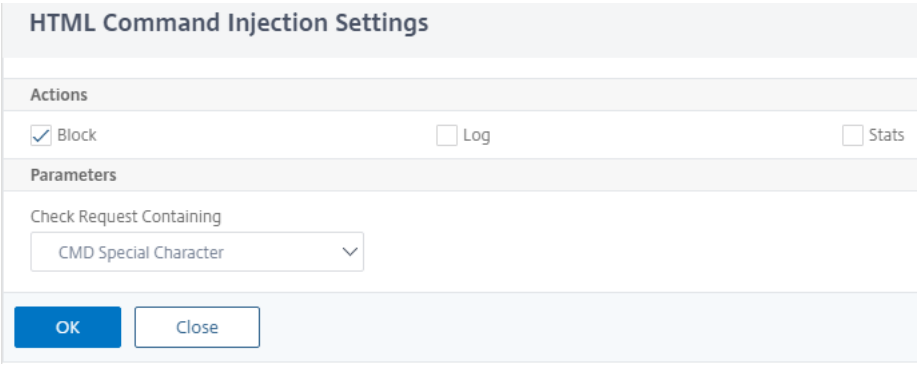
Citrix ADC GUI を使用したコマンドインジェクションチェックの構成

コマンドインジェクションチェックを設定するには、次の手順を実行します。

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. [プロファイル] ページで、プロファイルを選択し、[編集] をクリックします。
3. **Citrix Web App Firewall** プロファイルページで、［詳細設定］セクションに移動し、［セキュリティチェック］をクリックします。



1. 「セキュリティチェック」セクションで、「**HTML** コマンドインジェクション」を選択し、「アクション設定」をクリックします。
2. [**HTML** コマンドインジェクション設定] ページで、次のパラメータを設定します。
   a) アクション。コマンドインジェクションセキュリティチェックに対して実行するアクションを 1 つ以上選択します。
   b) 次の内容を含むリクエストをチェックします。コマンドインジェクションパターンを選択して、着信要求にパターンがあるかどうかを確認します。
3. [**OK**] をクリックします。



GUI を使用してコマンドインジェクションパターンを表示またはカスタマイズする

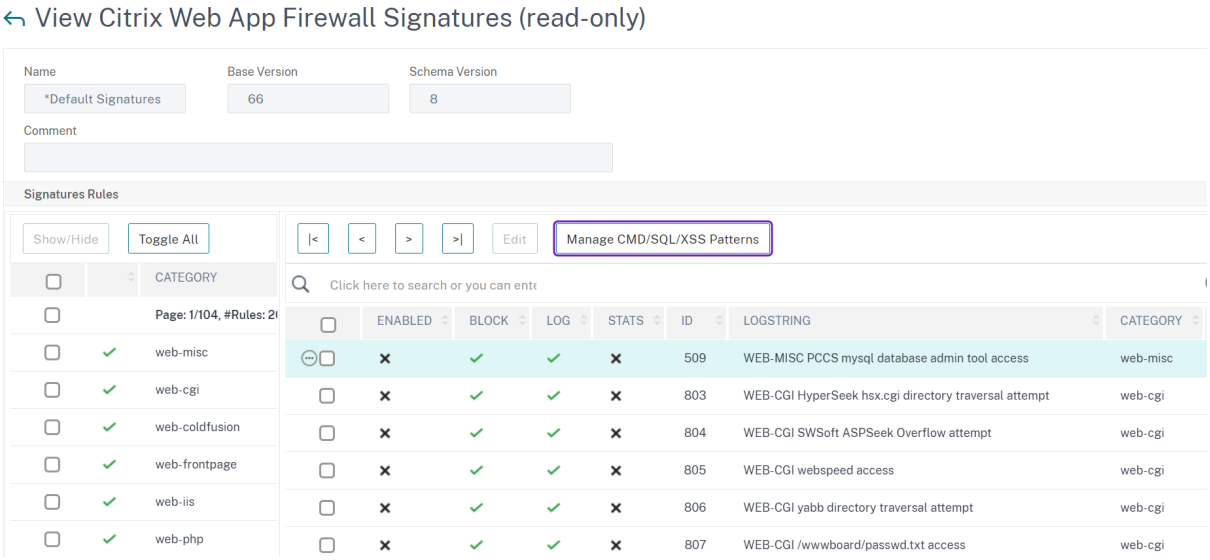
GUI を使用して、**HTML** コマンドインジェクションパターンを表示またはカスタマイズできます。

デフォルトのコマンド注入パターンは、デフォルトシグニチャファイルで指定されます。シグニチャオブジェクトをプロファイルにバインドしない場合、デフォルトシグニチャオブジェクトで指定されたデフォルトの HTML コマンド注入パターンが、コマンドインジェクションセキュリティチェック処理のためにプロファイルによって使用されます。デフォルトのシグニチャオブジェクトで指定されている規則とパターンは読み取り専用です。編集や修正はできません。これらのパターンを変更または変更する場合は、デフォルトの SSignatures オブジェクトのコピーを作成して、ユーザー定義署名オブジェクトを作成します。新しいユーザ定義シグニチャオブジェクトのコマンドインジェクションパターンを変更し、これらのカスタマイズされたパターンを使用するトラフィックを処理しているプロファイルでこのシグニチャオブジェクトを使用します。

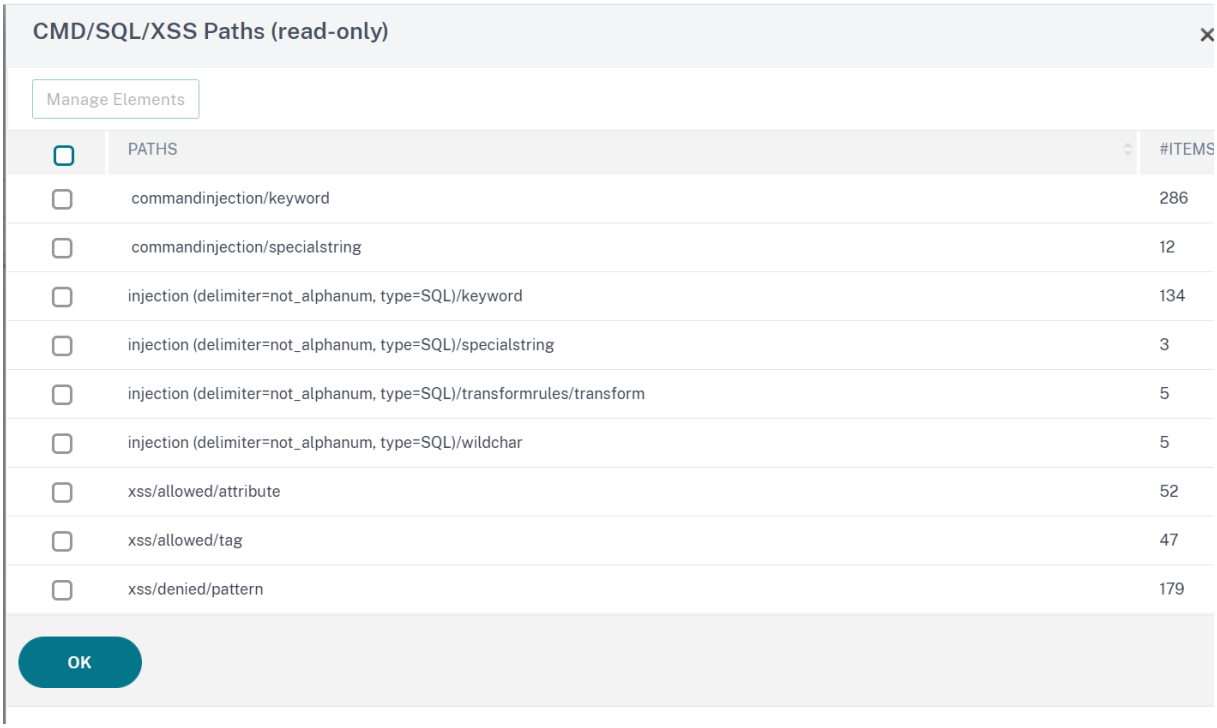
詳細については、「署名」を参照してください。

GUI を使用してデフォルトのコマンドインジェクションパターンを表示するには、次の手順を実行します。

1. [アプリケーションファイアウォール] **>** [署名] に移動し、[***** デフォルトシグニチャ] を選択し、[編集] をクリックします。



1. [**CMD/SQL/XSS** パターンの管理] をクリックします。**CMD/SQL/XSS** パス **(**読み取り専用**)** テーブルには、CMD/SQL/XSSインジェクションに関連するパターンが表示されます。



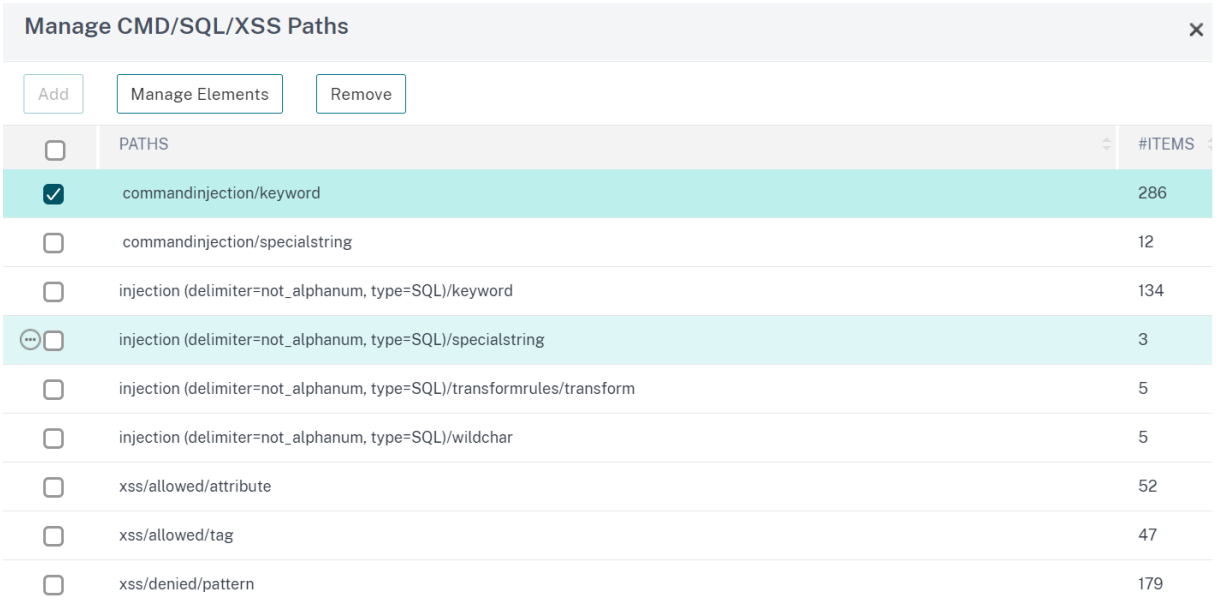
1. 行を選択し、[要素の管理] をクリックして、Web App Firewall コマンドインジェクションチェックで使用される対応するコマンド注入パターン (キーワード、特殊文字列、変換ルール、またはワイルドカード文字) を表示します。

GUI を使用してコマンドインジェクションパターンをカスタマイズするには

ユーザー定義の署名オブジェクトを編集して、**CMD** キーワード、特殊文字列、およびワイルドカード文字をカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除したりできます。コマンドインジェクション特殊文字列の変換ルールを変更できます。

1. [アプリケーションファイアウォール**] > [**署名**]** に移動し、ターゲットの [ユーザ定義署名] を選択し、[追加] をクリックします。[**CMD/SQL/XSS** パターンの管理] をクリックします。
2. [**CMD/SQL/XSS** パスの管理] ページで、ターゲットの CMD インジェクション行を選択します。
3. [要素を管理]、[コマンドインジェクション要素を追加]、または [削除] をクリックします。

> 警告:
>
> デフォルトのコマンドインジェクション要素を削除または変更する前に、注意が必要です。または、CMD パスを削除して行全体を削除する必要があります。シグニチャルールとコマンドインジェクションセキュリティチェックは、これらの要素に基づいてコマンドインジェクション攻撃を検出し、アプリケーションを保護します。SQL パターンをカスタマイズすると、編集中に必要なパターンが削除されると、アプリケーションがコマンドインジェクション攻撃に対して脆弱になる可能性があります。



コマンドインジェクショントラフィックおよび違反統計情報の表示

Citrix Web App Firewall 統計ページには、セキュリティトラフィックとセキュリティ違反の詳細が表形式またはグラフ形式で表示されます。

コマンドインターフェイスを使用してセキュリティの統計情報を表示するには。

コマンドプロンプトで入力します。

```
stat appfw profile profile1
```

Appfw プロファイルトラフィック

統計	レート (/s)	合計
要求	0	0
要求バイト数	0	0
応答	0	0
送信バイト数	0	0
中止します	0	0
リダイレクトします	0	0
長期平均応答時間（ミリ秒）	–	0
最近の平均応答時間（ミリ秒）	–	0

HTML/XML/JSON 違反の統計	レート (/s)	合計
開始 URL	0	0
URL を拒否する	0	0
リファラヘッダー	0	0
バッファオーバーフロー	0	0
クッキーの整合性	0	0
クッキーのハイジャック	0	0
CSRF フォームタグ	0	0
HTML クロスサイトスクリプティング	0	0
HTML SQL インジェクション	0	0
フィールドの書式	0	0
フィールドの一貫性	0	0
クレジットカード	0	0
セーフオブジェクト	0	0
署名の違反	0	0
コンテンツの種類	0	0
JSON サービス拒否攻撃	0	0
JSON SQL injection	0	0
JSON Cross-Site Scripting	0	0
ファイルのアップロードの種類	0	0
コンテンツタイプ XML ペイロードを推定	0	0
HTML CMD インジェクション	0	0
XML 形式	0	0
XML サービス拒否 (XDoS)	0	0
XML メッセージの検証	0	0
Web サービスの相互運用性	0	0
XML SQL Injection	0	0
XML Cross-Site Scripting	0	0

HTML/XML/JSON 違反の統計	レート (/s)	合計
XML 添付ファイル	0	0
SOAP 障害違反	0	0
XML 汎用違反	0	0
違反合計	0	0

HTML/XML/JSON ログ統計情報	レート (/s)	合計
開始 URL ログ	0	0
URL ログを拒否する	0	0
リファラーヘッダーログ	0	0
バッファオーバーフローログ	0	0
クッキーの整合性ログ	0	0
クッキーのハイジャックログ	0	0
タグログからの CSRF	0	0
HTML クロスサイトスクリプティングログ	0	0
HTML クロスサイトスクリプティング変換ログ	0	0
HTML SQL インジェクションのログ	0	0
HTML SQL 変換ログ	0	0
フィールド形式のログ	0	0
フィールド整合性ログ	0	0
クレジットカード	0	0
クレジットカード変換ログ	0	0
セーフオブジェクトログ	0	0
シグニチャログ	0	0
コンテンツタイプログ	0	0
JSON サービス拒否ログ	0	0
JSON SQL インジェクションログ	0	0

HTML/XML/JSON ログ統計情報	レート (/s)	合計
JSON クロスサイトスクリプティングログ	0	0
ファイルアップロードタイプのログ	0	0
コンテンツタイプ XML ペイロード L を推測する	0	0
HTML コマンドインジェクションログ	0	0
XML 形式のログ	0	0
XML サービス拒否 (XDoS) ログ	0	0
XML メッセージ検証ログ	0	0
WSI ログ	0	0
XML SQL インジェクションログ	0	0
XML クロスサイトスクリプティングログ	0	0
XML 添付ファイルログ	0	0
SOAP 障害ログ	0	0
XML 汎用ログ	0	0
ログメッセージの合計	0	0

サーバエラー応答統計レート **(/s) >** 合計 |
|—|-|-|
HTTP クライアントエラー (4xx Resp) | 0 | 0|
HTTP サーバエラー (5xx Resp) | 0 |

Citrix ADC GUI を使用した **HTML** コマンドインジェクション統計の表示

コマンドインジェクションの統計情報を表示するには、次の手順を実行します。

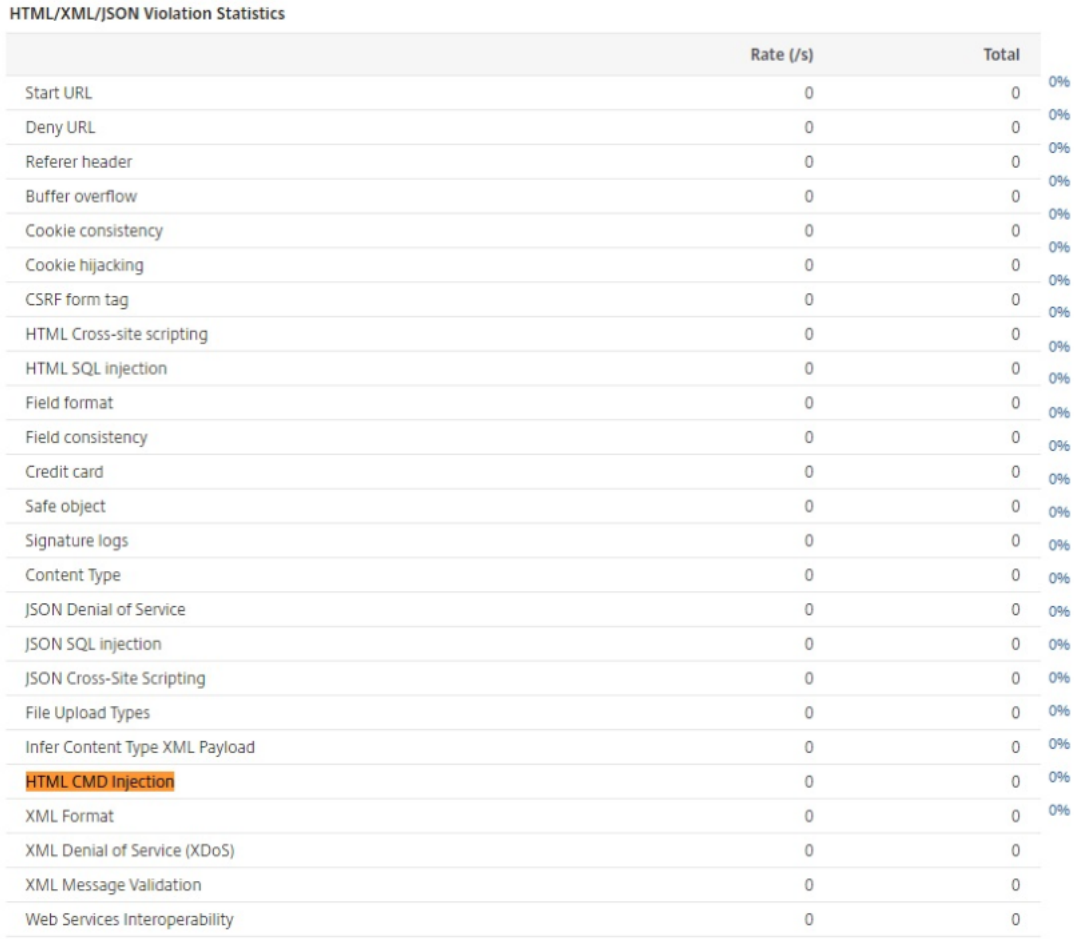
1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ウィンドウで、Web App Firewall プロファイルを選択し、[統計] をクリックします。
3. **Citrix Web App Firewall** 統計ページには、HTML コマンドインジェクショントラフィックと違反の詳細が表示されます。

4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィック形式で表示できます。

HTML コマンドインジェクショントラフィックの統計情報

HTML SQL Injection logs	0	0
HTML SQL transform logs	0	0
Field format logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0
JSON Cross-Site Scripting logs	0	0
File upload types logs	0	0
Infer Content Type XML Payload Logs	0	0
HTML Command Injection logs	0	0
XML Format logs	0	0
XML Denial of Service(XDoS) logs	0	0
XML Message Validation logs	0	0
WSI logs	0	0
XML SQL Injection logs	0	0
XML XSS logs	0	0
XML Attachment logs	0	0

HTML コマンドインジェクション違反の統計



HTML/XML/JSON Violation Statistics

	Rate (/s)	Total	
Start URL	0	0	0%
Deny URL	0	0	0%
Referer header	0	0	0%
Buffer overflow	0	0	0%
Cookie consistency	0	0	0%
Cookie hijacking	0	0	0%
CSRF form tag	0	0	0%
HTML Cross-site scripting	0	0	0%
HTML SQL injection	0	0	0%
Field format	0	0	0%
Field consistency	0	0	0%
Credit card	0	0	0%
Safe object	0	0	0%
Signature logs	0	0	0%
Content Type	0	0	0%
JSON Denial of Service	0	0	0%
JSON SQL injection	0	0	0%
JSON Cross-Site Scripting	0	0	0%
File Upload Types	0	0	0%
Infer Content Type XML Payload	0	0	0%
HTML CMD Injection	0	0	0%
XML Format	0	0	0%
XML Denial of Service (XDoS)	0	0	
XML Message Validation	0	0	
Web Services Interoperability	0	0	

XML 外部エンティティ (XXE) 攻撃防止

October 7, 2021

XML 外部エンティティ (XXE) 攻撃保護は、Web アプリケーションが存在する信頼されたドメイン外のエンティティに関して、受信ペイロードに許可されていない XML 入力があるかどうかを検査します。XXE 攻撃は、外部エンティティへの参照を含む入力を含む XML ペイロードを解析する弱い XML パーサーがある場合に発生します。

Citrix ADC アプライアンスでは、XML パーサーが正しく構成されていない場合、この脆弱性を悪用した影響は危険です。これにより、攻撃者は Web サーバー上の機密データを読み取ることができます。サービス拒否攻撃などを実行します。したがって、XXE 攻撃からアプライアンスを保護することが重要です。Web アプリケーションファイアウォールは、コンテンツタイプが XML として識別される限り、XXE 攻撃からアプライアンスを保護できます。悪意のあるユーザーがこの保護メカニズムをバイパスするのを防ぐため、HTTP ヘッダーの「推論」コンテンツタイプが本文のコンテンツタイプと一致しない場合、WAF は受信要求をブロックします。このメカニズムにより、ホワイトリストに登録されたデフォルトまたはデフォルト以外のコンテンツタイプが使用されるときに XXE 攻撃保護バイパスが回避されます。

Citrix ADC アプライアンスに影響を与える可能性のある XXE 脅威には、次のようなものがあります。

- 機密データ漏洩
- サービス拒否（DOS）攻撃
- サーバー側の偽造要求
- ポートスキャン

XML 外部エンティティ (XXE) インジェクション保護を構成する

コマンドインターフェイスを使用して XML 外部エンティティ (XXE) チェックを構成するには:
コマンドラインインターフェイスで、Application fireall profile コマンドを追加または変更して **XXE** 設定を構成できます。ブロック、ログ、および統計アクションを有効にすることができます。

コマンドプロンプトで入力します。

```
set appfw profile <name> [-inferContentTypeXmlPayloadAction <inferContentTypeXmlPayloa
 <block | log | stats | none>]
```

> 注:
>
> デフォルトでは、XXE アクションは「none」に設定されています。

例**:**

```
set appfw profile profile1 -inferContentTypeXmlPayloadAction Block
```

ここで、アクションタイプは次のとおりです。

ブロック: リクエストは、リクエスト内の URL の例外なくブロックされます。

ログ:HTTP 要求ヘッダーの content-type とペイロードの間に不一致が発生した場合は、違反する要求に関する情報をログメッセージに含める必要があります。

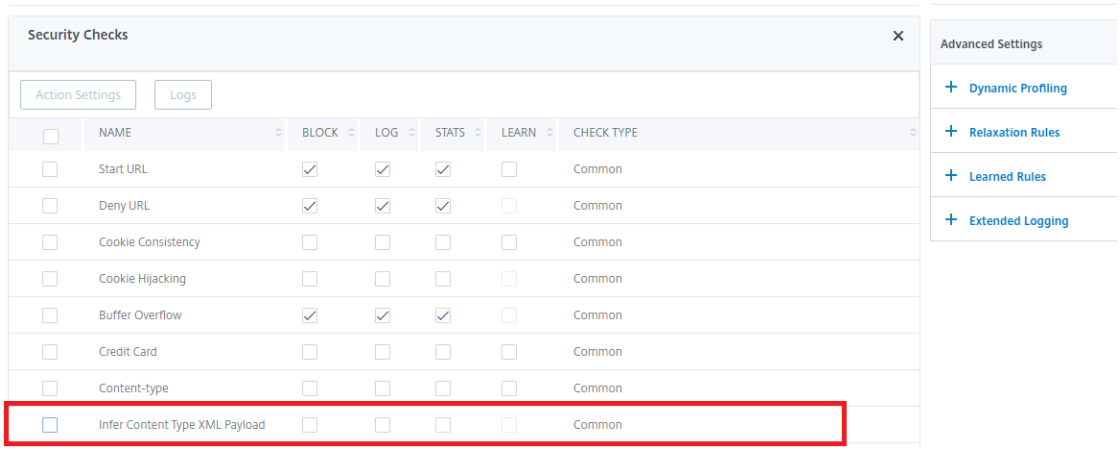
統計: コンテンツタイプの不一致が検出されると、この違反タイプの対応する統計値が増加します。

なし: コンテンツタイプの不一致が検出された場合、アクションは実行されません。[なし] は、他のアクションタイプと組み合わせることはできません。デフォルトのアクションは [なし] に設定されています。

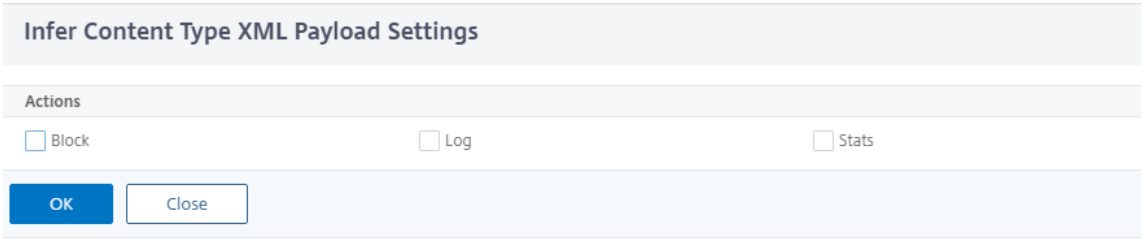
Citrix ADC GUI を使用して XXE インジェクションチェックを構成する

XXE インジェクションチェックを設定するには、次の手順を実行します。

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. [プロファイル] ページで、プロファイルを選択し、[編集] をクリックします。
3. **Citrix Web App Firewall** プロファイルページで、［詳細設定］セクションに移動し、［セキュリティチェック］をクリックします。



4. 「セキュリティチェック」セクションで、「コンテンツタイプの **XML** ペイロードを推定」を選択し、「アクション設定」をクリックします。
5. [コンテンツタイプ XML ペイロードの設定] ページで、次のパラメータを設定します。
   a) アクション。XXE インジェクションセキュリティチェックに対して実行するアクションを 1 つ以上選択します。
6. [**OK**] をクリックします。



XXE インジェクショントラフィックおよび違反統計情報の表示

Citrix Web App Firewall 統計ページには、セキュリティトラフィックとセキュリティ違反の詳細が表形式またはグラフ形式で表示されます。

コマンドインターフェイスを使用してセキュリティの統計情報を表示するには。

コマンドプロンプトで入力します。

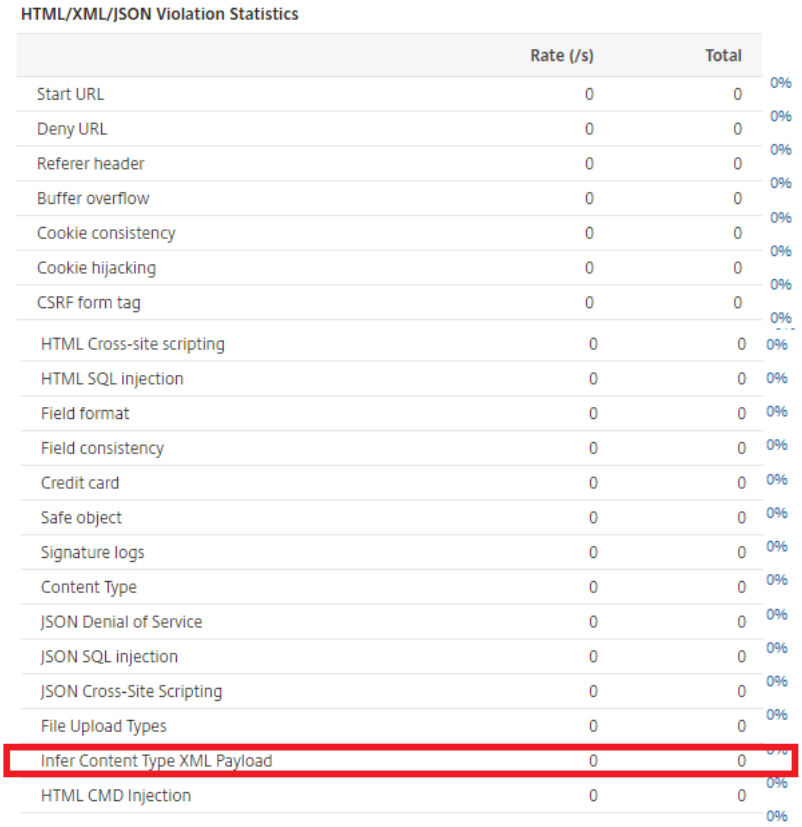
```
stat appfw profile profile1
```

Citrix ADC GUI を使用した **XXE** インジェクションの統計情報の表示

XXE 射出統計を表示するには、次の手順を実行します。

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ウィンドウで、Web App Firewall プロファイルを選択し、[統計] をクリックします。

3. **Citrix Web App Firewall** 統計ページに、XXE コマンドインジェクショントラフィックと違反の詳細が表示されます。
4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィック形式で表示できます。



HTML/XML/JSON Violation Statistics

	Rate (/s)	Total	
Start URL	0	0	0%
Deny URL	0	0	0%
Referer header	0	0	0%
Buffer overflow	0	0	0%
Cookie consistency	0	0	0%
Cookie hijacking	0	0	0%
CSRF form tag	0	0	0%
HTML Cross-site scripting	0	0	0%
HTML SQL injection	0	0	0%
Field format	0	0	0%
Field consistency	0	0	0%
Credit card	0	0	0%
Safe object	0	0	0%
Signature logs	0	0	0%
Content Type	0	0	0%
JSON Denial of Service	0	0	0%
JSON SQL injection	0	0	0%
JSON Cross-Site Scripting	0	0	0%
File Upload Types	0	0	0%
Infer Content Type XML Payload	0	0	0%
HTML CMD Injection	0	0	0%

バッファオーバーフローチェック

October 7, 2021

バッファオーバーフローチェックは、Web サーバ上でバッファオーバーフローを引き起こす試みを検出します。Web App Firewall は、URL、Cookie、またはヘッダーが設定された長さよりも長いことを検出すると、バッファオーバーフローを引き起こす可能性があるため、要求をブロックします。

バッファオーバーフローチェックは、処理できるよりも大きいデータ文字列を受信すると、クラッシュまたは予期せぬ動作をする、安全でないオペレーティングシステムまたは Web サーバソフトウェアに対する攻撃を防止します。適切なプログラミング技術は、入ってくるデータをチェックし、長すぎる文字列を拒否または切り捨てることによって、バッファオーバーフローを防止します。しかし、多くのプログラムは、すべての受信データをチェックしないため、バッファオーバーフローに対して脆弱です。この問題は特に、古いバージョンの Web サーバーソフトウェアとオペレーティングシステムに影響しますが、その多くは引き続き使用されています。

バッファオーバーフローのセキュリティチェックでは、ブロック、ログ、および 統計の各アクションを設定できます。さらに、次のパラメータを設定することもできます。

- **URL** の最大長。要求された URL で Web App Firewall が許可する最大長。より長い URL を持つリクエストはブロックされます。可能な値: 0～65535。デフォルト: 1024
- クッキーの最大長。Web App Firewall がリクエスト内のすべての Cookie を許可する最大長です。より長いクッキーを使用したリクエストは、違反をトリガーします。可能な値: 0～65535。デフォルト： 4096
- ヘッダーの最大長。Web App Firewall で HTTP ヘッダーを許可する最大長。長いヘッダーを持つリクエストはブロックされます。可能な値: 0～65535。デフォルト： 4096
- クエリ文字列の長さ。受信リクエストのクエリ文字列に許可される最大長。クエリが長くなるリクエストはブロックされます。可能な値: 0～65535。デフォルト: 1024
- 要求の長さの合計です。着信要求に許可される最大要求長。長い長さの要求はブロックされます。可能な値: 0～65535。デフォルト:24820

コマンドラインを使用したバッファオーバーフローのセキュリティチェックの設定

コマンドラインを使用してバッファオーバーフローのセキュリティチェック操作およびその他のパラメーターを構成するには

コマンドプロンプトで入力します。

```
add appfw profile <name> -bufferOverflowMaxURLLength <positive_integer> -
bufferOverflowMaxHeaderLength <positive_integer> - bufferOverflowMaxCookieLength
 <positive_integer> -bufferOverflowMaxQueryLength <positive_integer> -
bufferOverflowMaxTotalHeaderLength <positive_integer>
```

例**:**

```
add appfw profile profile1 -bufferOverflowMaxURLLength 7000 -bufferOverflowMaxHeaderLe
 7250 - bufferOverflowMaxCookieLength 7100 -bufferOverflowMaxQueryLength
7300 -bufferOverflowMaxTotalHeaderLength 7300
```

Citrix ADC GUI を使用してバッファオーバーフローのセキュリティチェックを構成する

1. **Security** > **Web App Firewall** および **Profiles** に移動します。
2. [プロファイル] ページで、プロファイルを選択し、[編集] をクリックします。
3. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] セクションに移動し、[セキュリティチェック] をクリックします。
4. 「セキュリティチェック」セクションで、「バッファオーバーフロー」を選択し、「アクション設定」をクリックします。
5. バッファオーバーフロー設定ページで、次のパラメータを設定します。
   a. アクション。コマンドインジェクションセキュリティチェックに対して実行するアクションを 1 つ以上選択

します。

b. URL の最大長。保護された Web サイトの URL の最大長（文字単位）。より長い URL を持つリクエストはブロックされます。

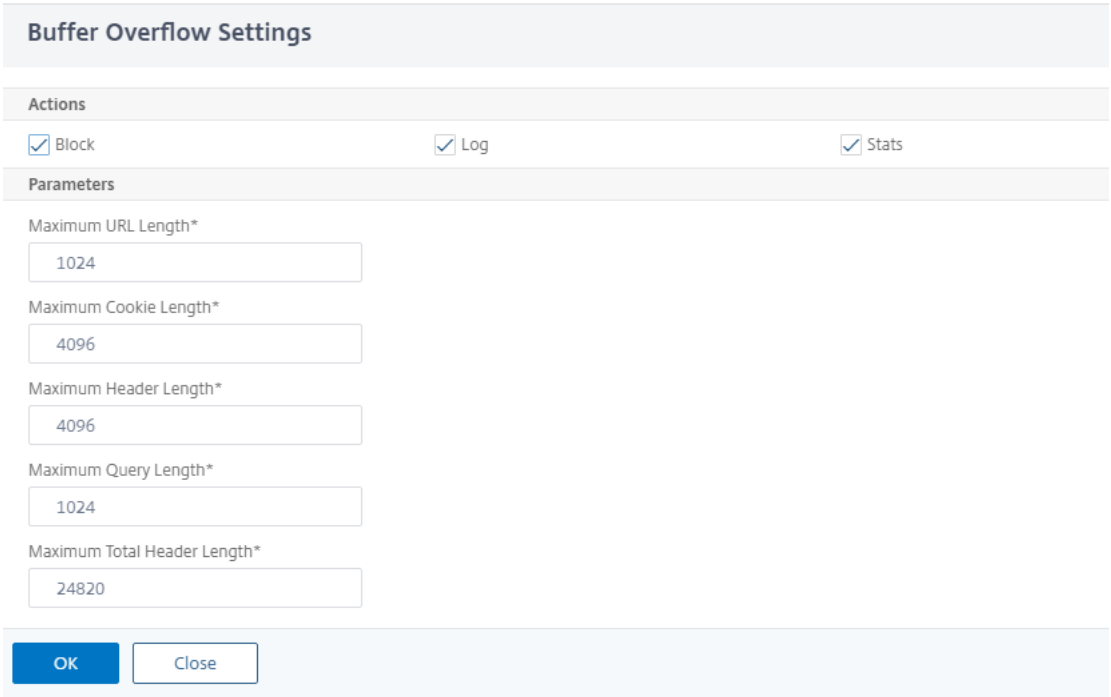
c. Cookie の最大長。保護された Web サイトに送信される Cookie の最大長（文字数）。クッキーの長いリクエストはブロックされます。

d. ヘッダーの最大長保護された Web サイトに送信されるリクエストの HTTP ヘッダーの最大長（文字単位）。長いヘッダーを持つリクエストはブロックされます。

e. クエリーの最大長。保護された Web サイトに送信されるクエリ文字列の最大長（バイト単位）。長いクエリ文字列を持つリクエストはブロックされます。

f. ヘッダーの最大合計長。保護された Web サイトに送信されるリクエストの HTTP ヘッダーの合計長に対する最大長（バイト単位）。HttpProfile におけるこの値と MaxHeaderLen の最小値が使用されます。長い長さの要求はブロックされます。

6. [**OK]** をクリックして [閉じる] をクリックします。



バッファオーバーフローのセキュリティチェックでのログ機能の使用

ログアクションを有効にすると、バッファオーバーフローのセキュリティチェック違反が APPFW_BUFFEROVERFLOW_URL、**APPFW_BUFFEROVERFLOW_COOKIE**、および **APPFW_BUFFEROVERFLOW_HDR** 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

GUI を使用してログを確認する場合は、クリックツーデプロイ機能を使用して、ログで示される緩和を適用できます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、**/var/log/** フォルダの **ns.log** s を末尾にして、バッファオーバーフロー違反に関連するログメッセージにアクセスします。

```
> \*\*Shell\*\*
> \*\*tail -f /var/log/ns.log | grep APPFW_BUFFEROVERFLOW\*\*
<!--NeedCopy-->
```

非ブロックモードでのバッファオーバーフロー MaxCookieLength 違反を示す CEF ログメッセージの例

```
Oct 22 17:35:20 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|\*\*APPFW_BUFFEROVERFLOW_COOKIE\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=41198 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html \*\*msg=Cookie header length(43) is
    greater than maximum allowed(16).\*\* cn1=119 cn2=465 cs1=
    owa_profile cs2=PPE1 cs3=wvOOOb+cJ2ZRbstZpyeNXIqLj7Y0001 cs4=ALERT
    cs5=2015 \*\*act=not blocked\*\*
<!--NeedCopy-->
```

非ブロックモードでのバッファオーバーフロー最大 URL 長違反を示す CEF ログメッセージの例

```
Oct 22 18:39:56 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|\*\*APPFW_BUFFEROVERFLOW_URL\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=19171 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html \*\*msg=URL length(39) is greater than
    maximum allowed(20).\*\* cn1=707 cn2=402 cs1=owa_profile cs2=PPE0
    cs3=kW49GcKbnwKByByi3+jeNzfgWa80000 cs4=ALERT cs5=2015 \*\*act=not
    blocked\*\*
<!--NeedCopy-->
```

ブロックモードでのバッファオーバーフロー MaxHeaderLength 違反を示すネイティブフォーマットログメッセージの例

```
Oct 22 18:44:00 <local0.info> 10.217.31.98 10/22/2015:18:44:00 GMT ns
    0-PPE-2 : default APPFW \*\*APPFW_BUFFEROVERFLOW_HDR\*\* 155 0 :
    10.217.253.62 374-PPE2 khhBEeY4DB8V2D3H2sMLkXmfWnA0002 owa_profile
    \*\*Header(User-Agent) length(82) is greater than maximum allowed
    (10)\*\* : http://aaron.stratum8.net/ \*\*<blocked>\*\*
<!--NeedCopy-->
```


GUI を使用してログメッセージにアクセスするには

Citrix GUI には、ログメッセージを分析するための便利なツール（**Syslog Viewer**）が含まれています。Syslog ビューアには、次の複数のオプションがあります。

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。[バッファオーバーフロー] 行をハイライト表示し、[ログ] をクリックします。プロファイルの Buffer Overflow Security Check から直接ログにアクセスすると、GUI によってログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
- また、「**NetScaler** 」>「システム」>「監査」に移動して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティー検査違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- [アプリケーションファイアウォール] > [ポリシー] > [監査] に移動します。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

XML ベースの Syslog ビューアには、関心のあるログメッセージだけを選択するためのさまざまなフィルタオプションがあります。バッファオーバーフローチェックのログメッセージを選択するには、[モジュール] のドロップダウンリストオプションで [**APPFW**] を選択してフィルタします。イベントタイプリストには、バッファオーバーフローのセキュリティチェックに関連するすべてのログメッセージを表示 するための、**APPFW_BUFFEROVERFLOW_COOKIE**、**PPFW_BUFFEROVERFLOW_HDR** の 3 つのオプションがあります。1 つまたは複数のオプションを選択して、さらに選択を絞り込むことができます。たとえば、[**APPFW_BUFFEROVERFLOW_COOKIE**] チェックボックスをオンにし、[適用] ボタンをクリックすると、Syslog ビューアには、Cookie ヘッダーの バッファオーバーフローのセキュリティチェック違反に関するログメッセージだけが表示されます。特定のログメッセージの行にカーソルを置くと、モジュール、イベントタイプ、イベント **ID**、クライアント **IP** などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログメッセージ内の対応する情報を強調表示することができます。

Click-to-Deploy: GUI には、クリックツーデプロイ機能があります。これは、現在、**URL** の長さ違反に関連するバッファオーバーフローのログメッセージに対してのみサポートされています。Syslog ビューアーを使用して、トリガーされた違反を表示するだけでなく、ブロックされたメッセージの観察された長さに基づいて情報に基づいた決定を実行することもできます。現在の値が制限的すぎて誤検出をトリガーしている場合は、メッセージを選択して展開し、現在の値をメッセージに表示される URL の長さの値に置き換えることができます。この操作では、ログメッセージは CEF ログ形式である必要があります。ログメッセージに対して緩和を展開できる場合は、行の [**Syslog Viewer**] ボックスの右端にチェックボックスが表示されます。チェックボックスをオンにし、[アクション] リストからオプションを選択して、緩和を展開します。**[**編集と配備]、[配備]、[すべて配備] は、**[**アクション**]** オプションとして利用できます。**APPFW_BUFFEROVERFLOW_URL** フィルタを使用すると、設定された URL の長さ違反に関連するすべてのログメッセージを分離できます。

個々のログメッセージを選択した場合、**3** つのアクションオプションの [編集と配備]、[配備**]**、および **[**すべて配備**]**

を使用できます。[編集と配備] を選択すると、[バッファオーバーフローの設定] ダイアログが表示されます。リクエストで観察された新しい URL の長さが、[最大 **URL** 長] 入力フィールドに挿入されます。編集を行わずに [**Close**] をクリックすると、現在設定されている値は変更されません。[**OK**] ボタンをクリックすると、[URL の最大長] の新しい値が以前の値に置き換えられます。

> 注
>
> 表示された [バッファオーバーフロー設定] ダイアログ ** の [ブロック]、[ログ]、および [統計アクション] チェックボックスはオフになっています。[編集と配備] オプションを選択する場合は、再設定する必要があります。[OK] をクリックする前に、必ずこれらのチェックボックスを有効にします。そうしないと、新しい **URL** の長さが構成されますが、アクションは none** に設定されます。

複数のログメッセージのチェックボックスをオンにした場合は、[Deploy All] オプションまたは [**DeployAll**] オプションを使用できます。展開されたログメッセージの URL 長が異なっている場合、設定された値は、選択したメッセージで観察された最大の URL 長の値に置き換えられます。ルールを展開すると、バッファーのオーバーフロー最大 **URLLength** の値が変更されるだけです。設定されたアクションは保持され、変更されません。

GUI でクイック展開機能を使用するには

1. Syslog ビューアで、モジュールオプションで **APPFW** を選択します。
2. 対応するログ・メッセージをフィルタリングするには､イベント・タイプとして｢**APPFW_BUFFEROVERFLOW_URL** ｣チェック・ボックスを有効にします。
3. ルールを選択するには、チェックボックスをオンにします。
4. 緩和をデプロイするには、オプションの [**Action**] ドロップダウンリストを使用します。
5. [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択し、[セキュリティチェック] をクリックして [バッファオーバーフローの設定] ペインにアクセスし、[**URL** の最大長] の値が更新されたことを確認します。

バッファオーバーフロー違反の統計情報

stats アクションが有効な場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行したときに、バッファオーバーフローのセキュリティチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロックアクションが有効になっている場合、最初の違反が検出されたときにページがブロックされるため、3 つのバッファオーバーフロー違反を含むページに対するリクエストは stats カウンタを 1 つずつ増やします。ただし、block が無効になっている場合、同じ要求を処理すると、違反ごとに stat カウンタがインクリメントされます。これは、違反ごとに個別のログメッセージが生成されるためです。

コマンドラインを使用してバッファオーバーフローセキュリティチェックの統計情報を表示するには

コマンドプロンプトで入力します。

```
> sh appfw stats
```

特定のプロファイルの統計情報を表示するには、次のコマンドを使用します。


```
> stat appfw profile <profile name>
```

GUI を使用してバッファオーバーフロー統計を表示するには

1. [システム] > [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 右側のペインで、[統計リンク] にアクセスします。
3. スクロールバーを使用して、バッファオーバーフローの違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

- バッファオーバーフローセキュリティチェックでは、許可される URL、Cookie、およびヘッダーの最大長を強制する制限を設定できます。
- **Block**、**Log**、**Stats** アクションを使用すると、トラフィックを監視し、アプリケーションに最適な保護を設定できます。
- Syslog ビューアを使用すると、バッファオーバーフロー違反に関連するすべてのログメッセージをフィルタリングおよび表示できます。
- クイック展開機能は、バッファーオーバーフロー最大 **URLLength** 違反に対してサポートされています。個々のルールを選択して展開することも、複数のログメッセージを選択して、URL の最大許容長である現在設定されている値を調整して緩和することもできます。選択したグループの URL の最大値が新しい値として設定され、現在違反としてフラグが付けられているすべての要求が許可されます。
- Web App Firewall は、着信リクエストを検査する際に、個々の Cookie を評価するようになりました。Cookie ヘッダーで受信した 1 つの Cookie の長さが、設定されている **BufferOverflowMaxCookieLength** を超えると、バッファオーバーフロー違反がトリガーされます。

> 重要
>
> リリース 10.5.e（59.13xx.e ビルドより前のいくつかの暫定的な拡張ビルド）と 11.0 リリース（65.x より前のビルド）では、Cookie ヘッダーの Web App Firewall 処理が変更されました。これらのリリースでは、すべての Cookie が個別に評価され、Cookie ヘッダーで受信された 1 つの Cookie の長さが設定された BufferOverflowMaxCookieLength を超えると、バッファオーバーフロー違反がトリガーされます。この変更の結果、10.5 以前のリリースビルドでブロックされた要求が許可される可能性があります。これは、cookie の長さを決定するために cookie ヘッダー全体の長さが計算されないためです。** 状況によっては、サーバーに転送される Cookie の合計サイズが許容値よりも大きく、サーバーが「400 Bad Request」で応答することがあります。
>
> この変更は元に戻されました。11.0 リリース 65.x 以降のビルドに加えて、10.5.e->59.13xx.e およびそれ以降の 10.5.e 拡張ビルドの動作は、リリース 10.5 の非拡張ビルドの動作に似ています。クッキーの長さを計算する際に、生の Cookie ヘッダー全体が考慮されるようになりました。クッキーの長さの決定には、スペースと、名前と値のペアを区切るセミコロン (;) 文字も含まれます。

Google ウェブツールキットのウェブアプリファイアウォールのサポート

October 7, 2021

注：この機能は、Citrix ADC リリース 10.5.e で使用できます。

Google Web Toolkit（GWT）リモートプロシージャコール（RPC）メカニズムに従った Web サーバーは、GWT サポートを有効にするための特定の構成を必要とせずに Citrix Web AppFirewall によって保護できます。

GWT とは何ですか?

GWT は、XMLHttpRequest、および JavaScript の専門知識を持っていない人々によって複雑な高性能 Web アプリケーションを構築し、最適化するために使用されます。このオープンソースの無料開発ツールキットは、小規模および大規模アプリケーションの開発に広く使用されており、フライトやホテルなどの検索結果などのブラウザベースのデータを表示するためによく使用されます。GWT は、ほとんどのブラウザやモバイルデバイス上で実行することができ、最適化された JavaScript スクリプトを書くための Java API とウィジェットのコアセットを提供します。GWT RPC フレームワークは、Web アプリケーションのクライアントとサーバーコンポーネントが HTTP 経由で Java オブジェクトを交換することが容易になります。GWT RPC サービスは、SOAP または REST に基づく Web サービスと同じではありません。これらは、単にクライアント上のサーバーと GWT アプリケーション間でデータを転送するための軽量な方法です。GWT は、メソッド呼び出しと戻り値の引数を交換する Java オブジェクトのシリアル化を処理します。

GWT を使用する一般的なウェブサイトについては、

https://www.quora.com/What-web-applications-use-Google-Web-Toolkit-%28GWT%29

GWT リクエストの仕組み

GWT RPC 要求は、パイプで区切られ、引数の数が可変です。これは HTTP POST のペイロードとして搬送され、次の値を持ちます。

1. Content-type = text/x-gwt-rpc. 文字セットには任意の値を指定できます。
2. Method = POST.

コンテンツタイプが「テキスト/ x-gwt-rpc」である場合、GET と POST HTTP 要求の両方が有効な GWT 要求とみなされます。クエリ文字列は、GWT 要求の一部としてサポートされるようになりました。アプリケーションファイアウォールプロファイルの「InspectQueryContentTypes」パラメータを「OTHER」に設定して、コンテンツタイプの「text/x-gwt-rpc」のリクエストクエリ部分を調べます。

次の例は、GWT 要求の有効なペイロードを示しています。

```
5|0|8|http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com
    .test.client.TestService|testMethod|java.lang.String|java.lang.
    Integer| myInput1|java.lang.Integer/3438268394|1|2|3|4|2|5|6|7|8|1|
<!--NeedCopy-->
```

リクエストは、次の 3 つの部分に分けることができます。

a)Header: 5|0|8|

上記のリクエストの最初の 3 桁5|0|8| は、それぞれ「バージョン、Subversion、およびテーブルのサイズ」を表します。これらは正の整数でなければなりません。

b）文字列テーブル:

```
http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.
client.TestService|testMethod|java.lang.String|java.lang.Integer|myInput1|
java.lang.Integer/3438268394|
```

上記のパイプ区切り文字列テーブルのメンバーには、ユーザーが指定した入力が含まれています。これらの入力は、Web App Firewall チェックのために解析され、次のように識別されます。

- 第 1:http://localhost:8080/test/

  これはリクエスト URL です。

- 第 2 回:16878339F02B83818D264AE430C20468

  一意の 16 進識別子。この文字列に 16 進以外の文字が含まれている場合、要求は不正な形式と見なされます。

- 第 3 回:com.test.client.TestService

  サービスクラス名

- 第 4 回: testMethod

  サービスメソッド名

- 5 回目以降:java.lang.String|java.lang.Integer|myInput1|java.lang.Integer/3438268394

  データ型とデータ。非プリミティブデータ型は、

  <container>.<sub-cntnr>.name/<integer><identifier>

c)Payload: 1|2|3|4|2|5|6|7|8|1|

ペイロードは、文字列テーブル内の要素への参照で構成されます。これらの整数値は、文字列テーブル内の要素数より大きくすることはできません。

GWT アプリケーション用の Web アプリケーションファイアウォール保護

Web App Firewall は、GWT RPC 要求を理解して解釈し、ペイロードにセキュリティチェック違反がないか検査し、指定されたアクションを実行します。

GWT リクエストの Web App Firewall ヘッダーと Cookie チェックは、他のリクエスト形式のものと似ています。適切な URL デコードと文字セット変換の後、文字列テーブル内のすべてのパラメータが検査されます。GWT リクエスト本体には、フィールド名だけフィールド値が含まれていません。入力値は、Web App Firewall Field Format チェックを使用して、指定した形式に対して検証できます。このチェックは、入力の長さの制御にも使用できます。入力における クロスサイトスクリプティング攻撃と **SQL** インジェクション攻撃は、Web App Firewall によって容易に検出され、阻止されます。

学習と緩和のルール：緩和ルールの学習と展開は、GWT 要求でサポートされています。Web App Firewall ルールは、<actionURL> <fieldName> マッピングの形式です。GWT 要求フォーマットは、フィールド名を持っていないため、特別な処理が必要です。Web App Firewall は、緩和ルールとして展開できる学習ルールにダミーのフィールド名を挿入します。-isRegex フラグは、GWT 以外のルールと同じように機能します。

- アクション URL:

  1 つの RPC に応答する複数のサービスを同じ Web サーバ上に構成できます。HTTP 要求には、RPC を処理する実際のサービスではなく、Web サーバーの URL があります。したがって、HTTP リクエスト URL に基づいて緩和が適用されることはありません。これは、ターゲットフィールドの URL のすべてのサービスが緩和されるためです。GWT 要求の場合、Web App Firewall は文字列テーブルの 4 番目のフィールドにある GWT ペイロードで見つかった実際のサービスの URL を使用します。

- フィールド名:

  GWT リクエスト本文にはフィールド値のみが含まれているため、Web App Firewall は、学習したルールを推奨するときに、1、2 などのダミーのフィールド名を挿入します。

  GWT 学習ルールの例

```
POST /abcd/def/gh HTTP/1.1
Content-type: text/x-gwt-rpc
Host: 10.217.222.75
Content-length: 157

5|0|8|http://localhost:8080/acdtest/|16878339
   F02Baf83818D264AE430C20468|
com.test.client.TestService|testMethod|java.lang.String%3b|java.
   lang.Integer|onblur|

  The learn data will be as follows:
> sh learningdata pr1 crossSiteScripting
Profile: pr1    SecurityCheck: crossSiteScripting
```



```
1)  Url:    http://localhost:8080/acdtest/  >> From GWT Payload.
    Field:  10
    Hits:   1
 Done
<!--NeedCopy-->
```

GWT 緩和規則の例

```
bind appfw profile pr1 -crossSiteScripting 1 abcd -isregex NOTREGEX
```

ログメッセージ：Web App Firewall は、GWT 要求で検出されたセキュリティチェック違反のログメッセージを生成します。不正な形式の GWT 要求によって生成されたログメッセージには、簡単に識別するための文字列「GWT」が含まれています。

不正な形式の **GWT** 要求のログメッセージの例：

```
Dec 5 21:48:02 <local0.notice> 10.217.31.247 12/05/2014:21:48:02 GMT ns
0-PPE-0 : APPFW Message 696 0 : "GWT RPC request with malformed payload. <
blocked>"
```

GWT と非 **GWT** リクエストの処理における違い：

同じペイロードによって、異なるコンテンツタイプに対して異なる Web App Firewall セキュリティチェック違反が発生する可能性があります。次の例を考えてみましょう。

```
5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468|com
.test.client.TestService|testMethod|java.lang.String%3b|java.lang.Integer|
select|
```

Content-type: application/x-www-form-urlencoded:

SQL インジェクションタイプが利用可能な 4 つのオプション (SQLSplCharandKeyword、SQLSplCharOR キーワード、SQLKeyword、または SQLSplChar) のいずれかを使用するように構成されている場合、このコンテンツタイプとともに送信された要求は SQL 違反になります。Web App Firewall では、上記のペイロードを処理するときに '&' がフィールドセパレータであり、'=' が名前と値のセパレータであると見なされます。これらの文字はいずれもポスト本体のどこにも表示されないため、コンテンツ全体が単一のフィールド名として扱われます。このリクエストのフィールド名には、SQL 特殊文字 (;) と SQL キーワード (選択) の両方が含まれています。したがって、4 つの SQL インジェクションタイプのオプションすべてに対して違反がキャッチされます。

Content-type: text/x-gwt-rpc:

このコンテンツタイプで送信された要求は、SQL インジェクションの種類が SqlSplCharorKeyword、SqlKeyword、または SqlSplchar の 3 つのオプションのいずれかに設定されている場合にのみ、SQL 違反をトリガーします。SQL インジェクションタイプがデフォルトオプションの SQLSplCharANDKeyword に設定されると、違反はトリガーされません。Web App Firewall は、垂直バー | を GWT リクエストで上記のペイロードのフィールド区切り文字とみなします。そのため、ポストボディはさまざまな form-field 値に分割され、(上記の規則に従って) form-field 名が追加されます。このスプリットによって、SQL 特殊文字と SQL キーワードが個別のフォームフィールドの一部になります。

フォームフィールド 8:`java.lang.String%3b -\> %3b is the (;)char`

フォームフィールド 10:`select`

その結果、SQL インジェクションタイプが **SQLSplChar** に設定されている場合、フィールド 8 は SQL 違反を示します。**SQLKeyword** の場合、フィールド 10 は違反を示します。SQL インジェクションタイプが **SQLSplCharORKeyword** オプションを使用して構成されている場合、これら 2 つのフィールドのいずれかが違反を示している可能性があります。このオプションでは、キーワードまたは特殊文字の存在が検索されます。デフォルトの **SQLSplCharANDKeyword** オプションに対して違反は捕捉されません。これは、**SQLSplChar** と **SQLKeyword** の両方を含む値を持つ単一のフィールドがないためです。

チップ:

- GWT サポートを有効にするために、特別な Web App Firewall 設定は必要ありません。
- コンテンツタイプはテキスト/x-gwt-rpc である必要があります。
- GWT ペイロードに適用されるすべての関連する Web App Firewall セキュリティチェックの緩和ルールの学習と展開は、サポートされている他のコンテンツタイプの場合と同様に機能します。
- 唯一の POST 要求は、GWT のために有効とみなされます。コンテンツタイプが text/x-gwt-rpc の場合、他のすべての要求メソッドはブロックされます。
- GWT 要求は、プロファイルの設定済みの POST 本文制限に従います。
- セキュリティチェックのセッションレス設定は適用されず、無視されます。
- CEF ログフォーマットは、GWT ログメッセージでサポートされています。

クッキー保護

October 7, 2021

Cookie は、Web サーバーからクライアントブラウザに送信される小さなパケットデータです。Cookie は、パスワード、ユーザー認証の詳細、資格情報などの機密データを HTTP 接続を介して伝送し、Web ブラウザーに保存します。したがって、情報を盗む攻撃者から Cookie を保護することは非常に重要です。

Cookie の整合性チェック：ユーザーリクエストで返された Cookie を調べて、Web サーバーがそのユーザーに設定した Cookie と一致することを確認します。変更された Cookie が見つかった場合、要求が Web サーバーに転送される前に、要求から取り除かれます。詳細については、「Cookie の一貫性チェック」トピックを参照してください。

Cookie ハイジャック保護：ハイジャックとは、攻撃者が Cookie への不正アクセスを取得する状況を指します。許可されたアクセスから Cookie を保護するために、Citrix ADC Web App Firewall（WAF）は、WAFCookie の整合性検証とともにクライアントからの TLS 接続にチャレンジします。新しいクライアント要求ごとに、アプライアンスは TLS 接続を検証し、要求内のアプリケーションとセッション Cookie の一貫性を検証します。詳細については、「Cookie ハイジャック保護」のトピックを参照してください。

SameSite cookie 属性： Set-Cookie HTTP 応答の `SameSite` 属性を使用すると、Cookie をファーストパーティまたは同じサイトのコンテキストに制限する必要があるかどうかを宣言できます。Cookie 設定は攻撃を軽減し、安全な Web 通信を提供します。詳細については、SameSite クッキー属性のトピックを参照してください 。

クッキーの整合性チェック

October 7, 2021

Cookie の整合性チェックでは、ユーザーから返された Cookie を調べて、Web サイトがそのユーザーに設定した Cookie と一致することを確認します。変更された Cookie が見つかった場合、要求が Web サーバーに転送される前に、要求から取り除かれます。また、クッキーの暗号化、クッキーのプロキシ、またはフラグの追加によって、処理するすべてのサーバークッキーを変換するように、クッキーの整合性チェックを構成することもできます。このチェックは、要求と応答に適用されます。

攻撃者は通常、Cookie を変更して、以前に認証されたユーザとして機密性の高い個人情報にアクセスしたり、バッファオーバーフローを引き起こしたりします。バッファオーバーフローチェックは、長い Cookie を使用してバッファオーバーフローを引き起こそうとする試みから保護します。Cookie の一貫性チェックでは、最初のシナリオに重点を置きます。

ウィザードまたは GUI を使用する場合は、
[Cookie 整合性チェックの変更] ダイアログボックスの [
全般] タブで、次の操作を有効または無効にできます。

- ブロック
- ログ
- 使い方
- 統計
- Transform. 有効にすると、Transform アクションは、次の設定で指定されているように、すべての Cookie を変更します。
  - サーバークッキーを暗号化します。応答をクライアントに転送する前に、Web サーバーによって設定された Cookie を暗号化します。ただし、Cookie 整合性チェック緩和リストにリストされているものを除きます。暗号化された Cookie は、クライアントが後続の要求を送信するときに復号化され、復号化された Cookie は保護された Web サーバーに転送される前に要求に再挿入されます。次のいずれかの暗号化タイプを指定します。
    * なし。クッキーを暗号化または復号化しないでください。デフォルトです。
    * 復号化のみ。暗号化されたクッキーのみを復号します。クッキーを暗号化しないでください。
    * セッションのみを暗号化します。セッションクッキーのみを暗号化します。永続的なクッキーを暗号化しないでください。暗号化されたクッキーを復号します。
    * すべてを暗号化します。セッションクッキーと永続クッキーの両方を暗号化します。暗号化されたクッキーを復号します。

      注：クッキーを暗号化する場合、Web App Firewall は、クッキーに
      HttpOnly フラグを追加します。このフラグは、スクリプトが Cookie にアクセスして解析するのを防ぎます。したがって、フラグは、スクリプトベースのウイルスやトロイの木馬が復号化された Cookie にアクセスし、その情報を使用してセキュリティを侵害することを防ぎます。これは、Encrypt ServerCookies パラメーター設定とは独立して処理される Flagsto Add inCookies パ

ラメーター設定に関係なく実行されます。

- プロキシサーバーのクッキー。Web サーバーによって設定されたすべての非永続的 (セッション) Cookie をプロキシします。ただし、Cookie 整合性チェック緩和リストにリストされているものは除きます。Cookie は、既存の Web App Firewall セッション Cookie を使用してプロキシされます。Web App Firewall は、保護された Web サーバーによって設定されたセッション Cookie を取り除き、クライアントに応答を転送する前にローカルに保存します。クライアントが後続の要求を送信すると、Web App Firewall はセッション Cookie をリクエストに再挿入してから、保護された Web サーバーに転送します。次のいずれかの設定を指定します。
  - なし。クッキーをプロキシしないでください。デフォルトです。
  - セッションのみ。プロキシ・セッション・クッキーのみ。永続的な Cookie をプロキシしない
    注: Cookie のプロキシを有効にした後で無効にすると（[セッションのみ] に設定した後でこの値を [なし] に設定）、Cookie のプロキシは無効にする前に確立されたセッションに対して維持されます。したがって、Web App Firewall がユーザーセッションを処理している間、この機能を安全に無効にすることができます。
- クッキーに追加するフラグ。変換中にクッキーにフラグを追加します。次のいずれかの設定を指定します。
  - なし。クッキーにフラグを追加しないでください。デフォルトです。
  - **HTTP** のみ。すべてのクッキーに HttpOnly フラグを追加します。HttpOnly フラグをサポートするブラウザでは、スクリプトがこのフラグが設定されている Cookie にアクセスできません。
  - 安全。SSL 接続でのみ送信される Cookie に Secure フラグを追加します。Secure フラグをサポートするブラウザーは、セキュリティで保護されていない接続を介してフラグ付きの Cookie を送信しません。
  - すべて。すべての Cookie に HttpOnly フラグを追加し、SSL 接続を介してのみ送信される Cookie に Secure フラグを追加します。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して Cookie 整合性チェックを設定できます。

- `set appfw profile <name> -cookieConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`
- `set appfw profile <name> -cookieTransforms ([**ON**] | [**OFF**])`
- `set appfw profile <name> -cookieEncryption ([**none**] | [**decryptOnly**] | [**encryptSession**] | [**encryptAll**])`
- `set appfw profile <name> -cookieProxying ([**none**] | [**sessionOnly**])`
- `set appfw profile <name> -addCookieFlags ([**none**] | [**httpOnly**] | [**secure**] | [**all**])`

Cookie 整合性チェックのリラックスを指定するには、GUI を使用する必要があります。「Cookie 整合性チェックの変更」ダイアログボックスの「チェック」タブで、「追加」をクリックして「Cookie 整合性チェック緩和の追加」ダイアログボックスを開くか、既存のリラクゼーションを選択して「開く」をクリックして「Cookie 整合性チェック緩和の変更」ダイアログボックスを開きます。どちらのダイアログボックスも、緩和の設定に同じオプションを提供します。

Cookie 整合性チェックの緩和の例を次に示します。

- [ログオンフィールド]。次の式は、文字列 logon_ で始まり、2 文字以上 15 文字以下の文字または数字の文字列が続くすべての Cookie 名を免除します。

```
^logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- ログオンフィールド (特殊文字)。次の式は、文字列 türkçe-logon_ で始まり、その後に 2 文字以上 15 文字以下の文字または数字の文字列が続くすべての Cookie 名を免除します。

```
^txC3xBCrkxC3xA7e-logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- 任意の文字列。sc-item_、ユーザーがショッピングカートに追加したアイテムの ID（[0-9a-za-Z]+）、2 番目のアンダースコア（_）、最後に彼が望むアイテムの数（[1-9][0-9]?）を含むクッキーを許可します。ユーザーによる変更が可能になるには:

```
^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$
<!--NeedCopy-->
```

注意: 正規表現は強力です。特に、PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク（.*）メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、意図しない Web コンテンツへのアクセスをブロックしたり、Cookie の一貫性チェックでそうしない攻撃を許可したりするなど、望ましくない結果が得られる可能性があります。ブロックされています。

> 重要
>
> リリース 10.5.e（59.13xx.e ビルドより前のいくつかの暫定拡張ビルド）および 11.0 リリース（65.x より前のビルド）では、Cookie ヘッダーの Web App Firewall 処理が変更されました。これらのリリースでは、すべての Cookie が個別に評価され、Cookie ヘッダーで受信された 1 つの Cookie の長さが設定された BufferOverflowMaxCookieLength を超えると、バッファオーバーフロー違反がトリガーされます。この変更の結果、10.5 以前のリリースビルドでブロックされた要求が許可される可能性があります。これは、cookie の長さを決定するために cookie ヘッダー全体の長さが計算されないためです。状況によっては、サーバーに転送される Cookie の合計サイズが受け入れられた値よりも大きく、サーバーが「400BadRequest」で応答

> する場合があります。
>
> この変更は元に戻されたことに注意してください。10.5.e->59.13xx.e およびそれ以降の 10.5.e 拡張ビルド、および 11.0 リリース 65.x 以降のビルドでの動作は、リリース 10.5 の非拡張ビルドの動作と似ています。クッキーの長さを計算する際に、生の Cookie ヘッダー全体が考慮されるようになりました。クッキーの長さの決定には、スペースと、名前と値のペアを区切るセミコロン (;) 文字も含まれます。**
>
> 注
>
> セッションレス **Cookie** の一貫性**:** リリース 11.0 では、Cookie の一貫性の動作が変更されました。以前のリリースでは、Cookie 整合性チェックによってセッション化が呼び出されます。クッキーはセッションに保存され、署名されます。一時的な Cookie には「wlt_」というサフィックスが付加され、クライアントに転送される前に永続的な Cookie に「wlf_」というサフィックスが付加されます。クライアントがこれらの署名付き wlf/wlt クッキーを返さない場合でも、Web App Firewall はセッションに格納された Cookie を使用して Cookie の一貫性チェックを実行します。
>
> リリース 11.0 では、クッキーの整合性チェックはセッションレスです。Web App Firewall は、Web App Firewall によって追跡されるすべての Cookie のハッシュである Cookie を追加するようになりました。このハッシュ Cookie またはその他の追跡された Cookie が見つからないか改ざんされた場合、Web App Firewall は要求をバックエンドサーバーに転送する前にクッキーを取り除き、Cookie 整合性違反をトリガーします。サーバーは、新しい要求として要求を扱い、新しい Set-Cookie ヘッダー（複数可）を送信します。Citrix ADC バージョン 13.0、12.1、および NetScaler 12.0 および 11.1 のクッキーの整合性チェックには、セッションレスオプションはありません。

クッキーのハイジャック保護

October 13, 2021

Cookie ハイジャック保護は、ハッカーからの Cookie 盗み攻撃を緩和します。セキュリティ攻撃では、攻撃者がユーザーセッションを引き継ぎ、Web アプリケーションへの不正アクセスを取得しています。ユーザーが銀行アプリケーションなどのウェブサイトを閲覧すると、そのウェブサイトはブラウザとのセッションを確立します。セッション中、アプリケーションは、ログイン資格情報、ページ訪問などのユーザーの詳細をクッキーファイルに保存します。クッキーファイルは、応答でクライアントブラウザに送信されます。ブラウザは、アクティブなセッションを維持するためにクッキーを保存します。攻撃者は、ブラウザのクッキーストアから、またはいくつかのルージュブラウザ拡張を介して手動でこれらのクッキーを盗むことができます。その後、攻撃者はこれらの Cookie を使用して、ユーザーのウェブアプリケーションセッションにアクセスします。

Cookie 攻撃を緩和するために、Citrix ADC Web App Firewall（WAF）は、WAF Cookie の一貫性検証とともにクライアントからの TLS 接続に挑戦します。新しいクライアント要求ごとに、アプライアンスは TLS 接続を検証し、要求内のアプリケーションとセッション Cookie の一貫性を検証します。攻撃者が犠牲者から盗まれたアプリケーション Cookie とセッション Cookie を混在させようとすると、Cookie の一貫性の検証は失敗し、設定された Cookie ハイジャックアクションが適用されます。Cookie の一貫性の詳細については、「Cookie の一貫性チェック」トピッ

クを参照してください。

> 注:
>
> Cookie ハイジャック機能は、ロギングおよび SNMP トラップをサポートします。ロギングの詳細については、ADM のトピックを、SNMP の構成の詳細については、SNMP のトピックを参照してください。

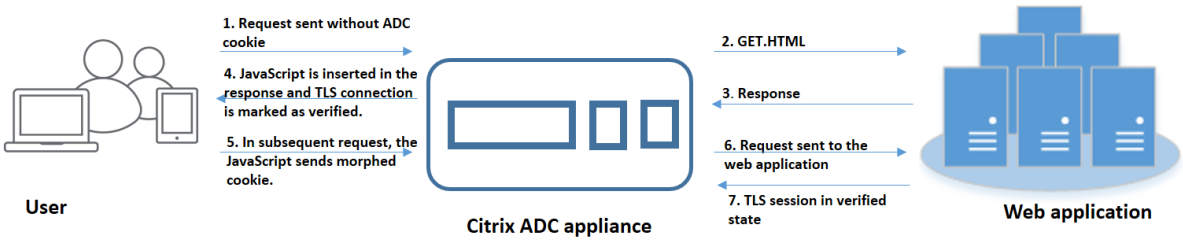
制限事項

- クライアントブラウザで JavaScript を有効にする必要があります。
- クッキーハイジャック保護は、TLS バージョン 1.3 ではサポートされていません。
- ブラウザは SSL 接続を再利用しないため、Internet Explorer (IE) ブラウザのサポートが制限されています。リクエストに対して複数のリダイレクトが送信され、最終的に IE ブラウザで「最大リダイレクトを超えました」というエラーが発生します。

クッキーハイジャック保護の仕組み

以下のシナリオでは、Citrix ADC アプライアンスでクッキーハイジャック保護がどのように機能するかを説明します。

シナリオ **1:** セッション **Cookie** を使用せずに最初の **Web** ページにアクセスするユーザー



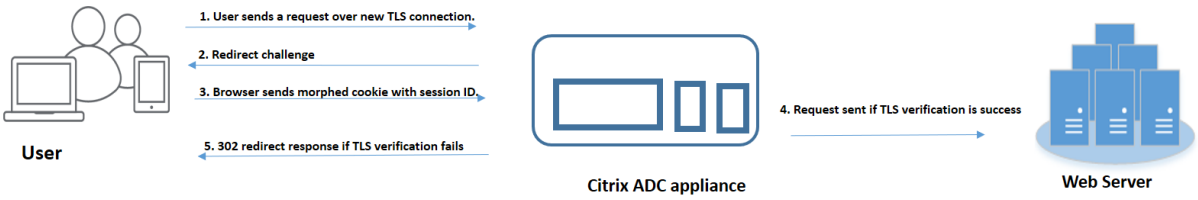
1. ユーザーは、Web アプリケーションへの認証を試み、要求に ADC セッション Cookie を使用せずに最初の Web ページへのアクセスを開始します。
2. 要求を受信すると、アプライアンスはセッション Cookie ID を持つアプリケーションファイアウォールセッションを作成します。
3. これにより、セッションの TLS 接続が開始されます。JavaScript はクライアントブラウザ上で送信および実行されないため、アプライアンスは TLS 接続を検証済みとしてマークし、チャレンジは必要ありません。

   > 注:
   >
   > 攻撃者がセッション Cookie を送信せずに犠牲者からすべてのアプリ Cookie ID を送信しようとした場合でも、アプライアンスは問題を検出し、要求をバックエンドサーバーに転送する前に、要求内のすべ

> てのアプリ Cookie を取り除きます。バックエンドサーバーは、アプリケーションクッキーなしでこの要求を考慮し、その設定に従って必要になります。

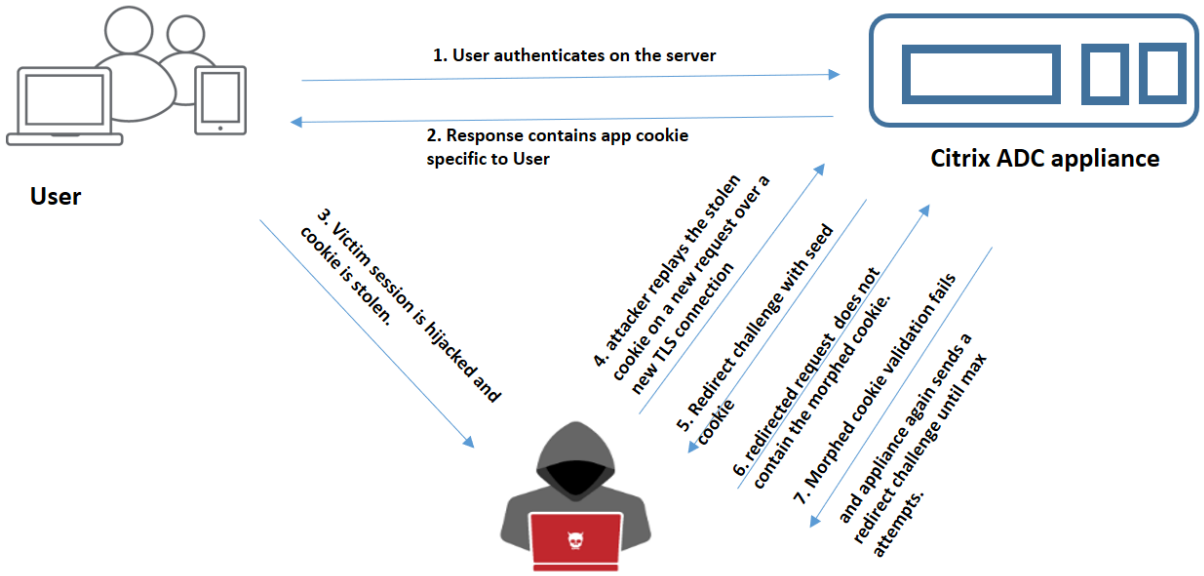
4. バックエンドサーバーが応答を送信すると、アプライアンスは応答を受信し、JavaScript セッショントークンとシード Cookie を使用して応答を転送します。その後、アプライアンスは TLS 接続を検証済みとしてマークします。
5. クライアントブラウザが応答を受信すると、ブラウザは JavaScript を実行し、セッショントークンとシードクッキーを使用してモーフィングされたクッキー ID を生成します。
6. ユーザーが TLS 接続を介して後続の要求を送信すると、アプライアンスはモーフィングされた Cookie 検証をバイパスします。これは、TLS 接続がすでに検証されているためです。

シナリオ **2:** セッション **Cookie** を使用して新しい **TLS** 接続を介して連続する **Web** ページにアクセスするユーザー



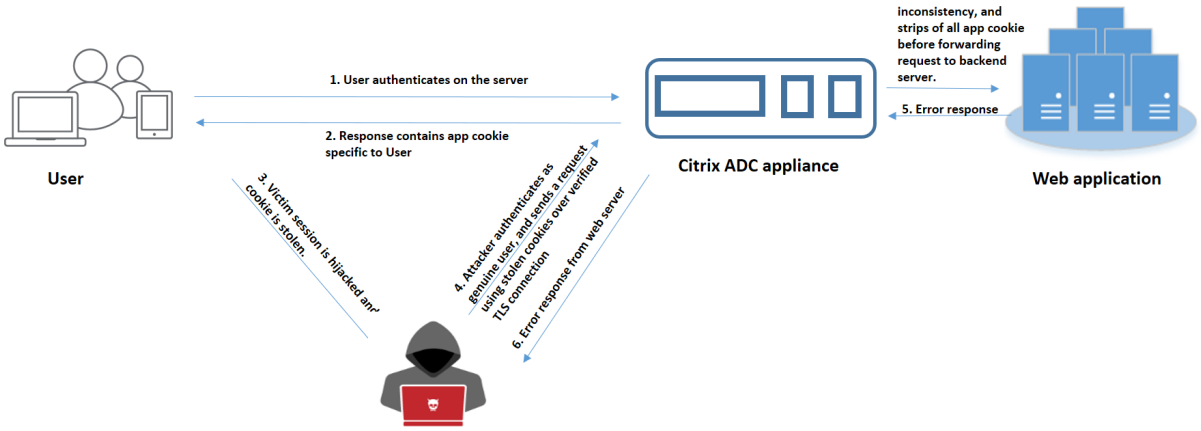
1. ユーザーが新しい TLS 接続を介して連続するページに対する HTTP 要求を送信すると、ブラウザはセッションクッキー ID とモーフィングされたクッキー ID を送信します。
2. これは新しい TLS 接続であるため、アプライアンスは TLS 接続を検出し、シード Cookie によるリダイレクト応答でクライアントにチャレンジします。
3. クライアントは、ADC からの応答を受信すると、セッションのトークンと新しいシードクッキーを使用してモーフィングされたクッキーを計算します。
4. クライアントは、この新しく計算されたモーフィングされたクッキーをセッション ID とともに送信します。
5. ADC アプライアンス内で計算されたモーフィング Cookie と、要求を介して送信されたクッキーが一致する場合、TLS 接続は検証済みとしてマークされます。
6. 計算されたモーフィングされたクッキーがクライアント要求に存在するクッキーと異なる場合、検証は失敗します。その後、アプライアンスはチャレンジをクライアントに送り返し、適切なモーフィングされた Cookie を送信します。

シナリオ **3:** 認証されていないユーザーになりすます攻撃者



1. ユーザーが Web アプリケーションに認証されると、攻撃者はさまざまな手法を使用して Cookie を盗み、再生します。
2. これは攻撃者からの新しい TLS 接続であるため、ADC は新しいシード Cookie とともにリダイレクトチャレンジを送信します。
3. 攻撃者は JavaScript を実行していないため、リダイレクトされたリクエストに対する攻撃者からの応答にはモーフィングされたクッキーは含まれていません。
4. その結果、ADC アプライアンス側でモーフィングされた Cookie 検証が失敗します。アプライアンスは再度、リダイレクトチャレンジをクライアントに送信します。
5. モーフィングされた Cookie 検証の試行回数がしきい値の制限を超えた場合、アプライアンスはステータスに Cookie ハイジャックとしてフラグを付けます。
6. 攻撃者が犠牲者から盗まれたアプリケーション Cookie とセッション Cookie を混在させようとすると、Cookie の一貫性チェックは失敗し、アプライアンスは設定された Cookie ハイジャックアクションを適用します。

シナリオ **4:** 認証されたユーザーになりすます攻撃者



1. また、攻撃者は本物のユーザーとしてウェブアプリケーションに認証し、被害者のクッキーを再生してウェブセッションにアクセスすることもできます。
2. ADC アプライアンスは、このような偽装された攻撃者も検出します。検証された TLS 接続は、攻撃者が被害者のクッキーを再生する際に使用されますが、ADC アプライアンスは要求内のセッション Cookie とアプリケーション Cookie が一貫しているかどうかを確認します。アプライアンスは、要求内のセッション Cookie を使用して、アプリケーション Cookie の一貫性を検証します。リクエストには攻撃者のセッション Cookie と被害者のアプリ Cookie が含まれているため、Cookie の一貫性の検証は失敗します。
3. その結果、アプライアンスは設定された Cookie ハイジャックアクションを適用します。構成済みのアクションが「ブロック」に設定されている場合、アプライアンスはすべてのアプリケーション Cookie を取り除き、要求をバックエンドサーバーに送信します。
4. バックエンドサーバーはアプリケーションクッキーなしで要求を受け取り、「ユーザーがログインしていません」などのエラー応答を攻撃者に応答します。

CLI を使用した Cookie ハイジャックの設定

特定のアプリケーションファイアウォールプロファイルを選択し、Cookie のハイジャックを防止する 1 つ以上のアクションを設定できます。

コマンドプロンプトで入力します。

```
set appfw profile <name> [-cookieHijackingAction <action-name> <block | log
 | stats | none>]
```

> 注:
>
> デフォルトでは、アクションは「none」に設定されています。

例:

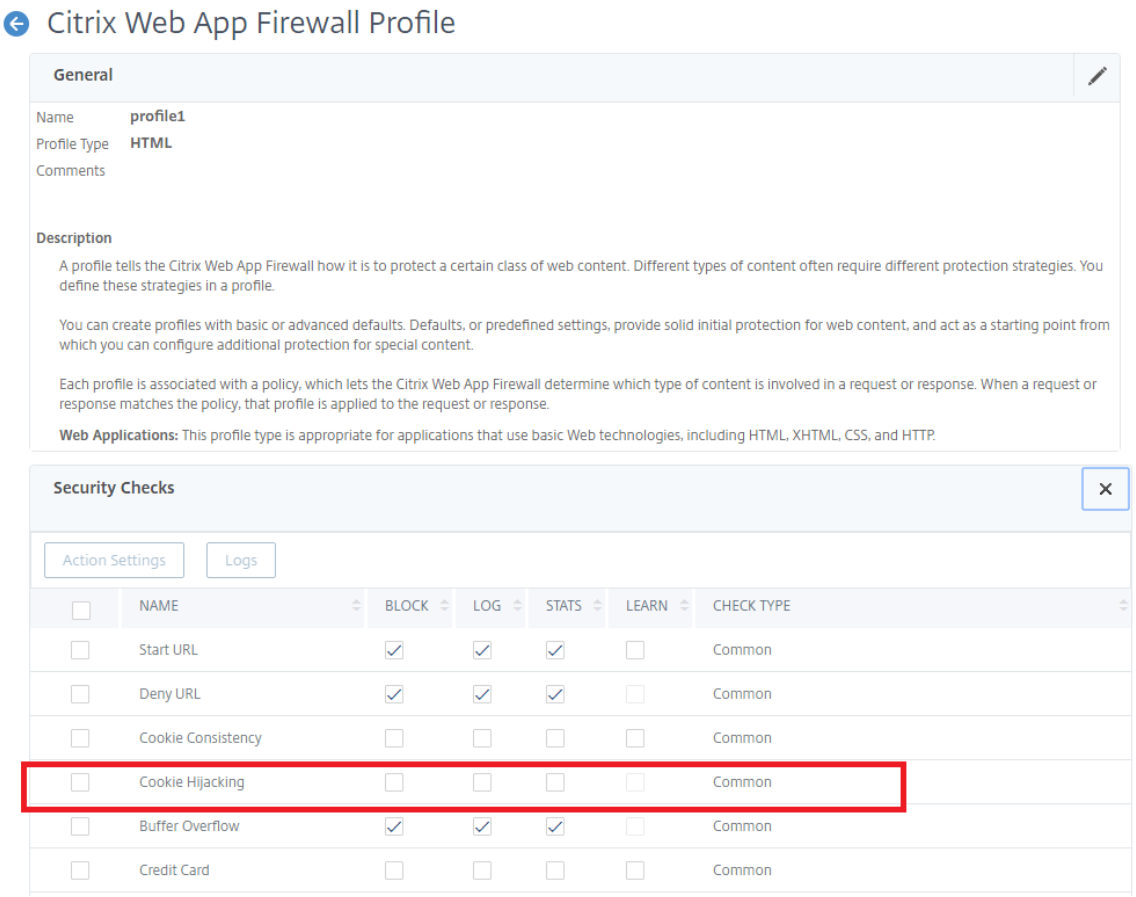
```
set appfw profile profile1 - cookieHijackingAction Block
```

ここで、アクションタイプは次のとおりです。

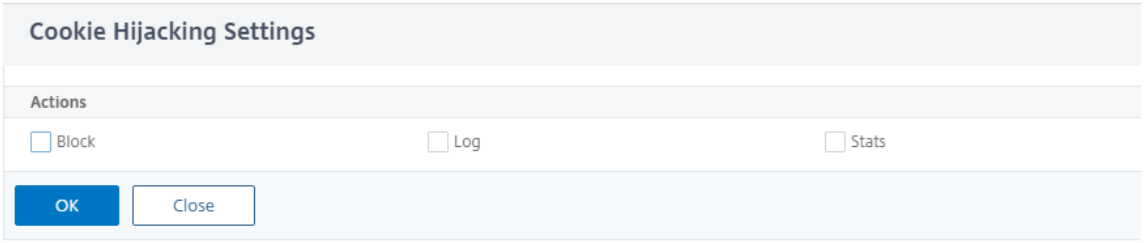
ブロック: このセキュリティチェックに違反する接続をブロックします。
Log: このセキュリティ・チェックの違反をログに記録します。
統計: このセキュリティチェックの統計情報を生成します。
[なし]: このセキュリティチェックのすべてのアクションを無効にします。

Citrix ADC GUI を使用してクッキーのハイジャックを構成する

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. [プロファイル] ページで、プロファイルを選択し、[編集] をクリックします。
3. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] セクションに移動し、[セキュリティチェック] をクリックします。



4. [セキュリティチェック] セクションで、[**Cookie** ハイジャック] を選択し、[アクションの設定] をクリックします。
5. [**Cookie** ハイジャック設定] ページで、Cookie ハイジャックを防止するアクションを 1 つ以上選択します。
6. [**OK**] をクリックします。



Citrix ADC GUI を使用してクッキーの一貫性を検証するための緩和ルールを追加する

クッキーの一貫性検証で誤検出を処理するには、クッキーの検証から除外できる Cookie のリラクゼーションルールを追加できます。

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. [プロファイル] ページで、プロファイルを選択し、[編集] をクリックします。
3. [**Citrix Web App Firewall** プロファイル］ページで、［詳細設定］セクションに移動し、［緩和ルール］をクリックします。
4. 「緩和ルール」セクションで、「クッキーの一貫性」を選択し、「アクション」をクリックします。
5. [**Cookie** 整合性緩和ルール] ページで、次のパラメータを設定します。
   a) 有効。緩和ルールを有効にする場合に選択します。
   b) クッキー名正規表現です。クッキー名が正規表現である場合に選択します。
   c) クッキー名。クッキーの検証から除外できる Cookie の名前を入力します。
   d) 正規表現エディタ。正規表現の詳細を指定する場合は、このオプションをクリックします。
   e) コメント。クッキーに関する簡単な説明。
6. **[作成]** して **[閉じる]** をクリックします。

CLI を使用した Cookie ハイジャックトラフィックおよび違反の統計情報の表示

セキュリティトラフィックとセキュリティ違反の詳細を表形式またはグラフ形式で表示します。

セキュリティ統計を表示するには、次の手順に従います。

コマンドプロンプトで入力します。

```
stat appfw profile profile1
```

Appfw プロファイルトラフィック

統計	レート (/s)	合計
要求	0	0
要求バイト数	0	0

Appfw プロファイルトラフィック統計	レート (/s)	合計
応答	0	0
送信バイト数	0	0
中止します	0	0
リダイレクトします	0	0
長期平均応答時間（ミリ秒）	–	0
最近の平均応答時間（ミリ秒）	–	0

HTML/XML/JSON 違反の統計	レート (/s)	合計
開始 URL	0	0
URL を拒否する	0	0
リファラヘッダー	0	0
バッファオーバーフロー	0	0
クッキーの整合性	0	0
クッキーのハイジャック	0	0
CSRF フォームタグ	0	0
HTML クロスサイトスクリプティング	0	0
HTML SQL インジェクション	0	0
フィールドの書式	0	0
フィールドの一貫性	0	0
クレジットカード	0	0
セーフオブジェクト	0	0
署名の違反	0	0
コンテンツの種類	0	0
JSON サービス拒否攻撃	0	0
JSON SQL injection	0	0
JSON Cross-Site Scripting	0	0
ファイルのアップロードの種類	0	0

HTML/XML/JSON 違反の統計	レート (/s)	合計
コンテンツタイプ XML ペイロードを推定	0	0
HTML CMD インジェクション	0	0
XML 形式	0	0
XML サービス拒否 (XDoS)	0	0
XML メッセージの検証	0	0
Web サービスの相互運用性	0	0
XML SQL Injection	0	0
XML Cross-Site Scripting	0	0
XML 添付ファイル	0	0
SOAP 障害違反	0	0
XML 汎用違反	0	0
違反合計	0	0

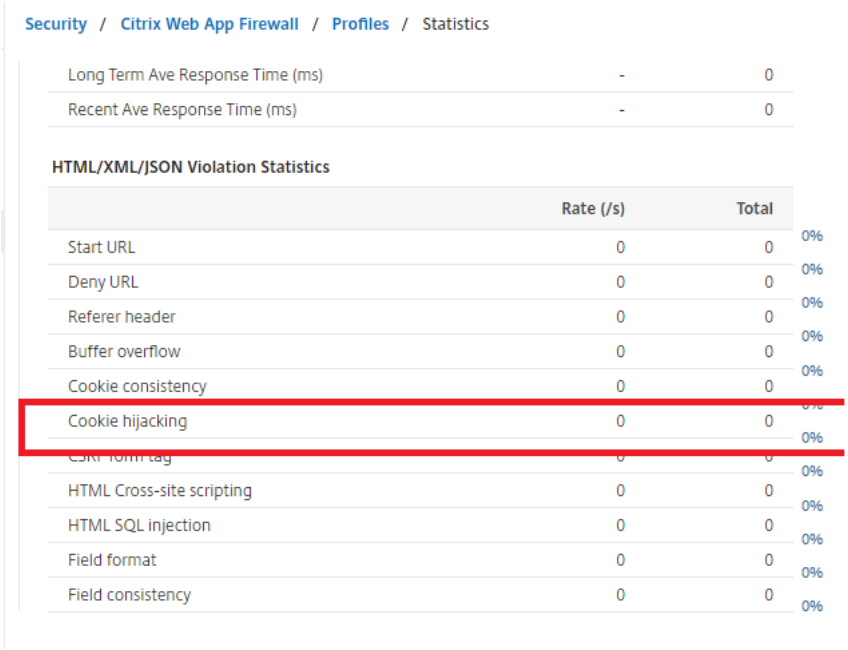
HTML/XML/JSON ログ統計情報	レート (/s)	合計
開始 URL ログ	0	0
URL ログを拒否する	0	0
リファラーヘッダーログ	0	0
バッファオーバーフローログ	0	0
バッファオーバーフローログ	0	0
クッキーの整合性ログ	0	0
クッキーのハイジャックログ	0	0
CSRF フォームタグログ	0	0
HTML クロスサイトスクリプティングログ	0	0
HTML クロスサイトスクリプティング変換ログ	0	0
HTML SQL インジェクションのログ	0	0
HTML SQL 変換ログ	0	0

HTML/XML/JSON ログ統計情報	レート (/s)	合計
フィールド形式のログ	0	0
フィールド整合性ログ	0	0
クレジットカード	0	0
クレジットカード変換ログ	0	0
セーフオブジェクトログ	0	0
シグニチャログ	0	0
コンテンツタイプログ	0	0
JSON サービス拒否ログ	0	0
JSON SQL インジェクションログ	0	0
JSON クロスサイトスクリプティングログ	0	0
ファイルアップロードタイプのログ	0	0
コンテンツタイプ XML ペイロード L を推測する	0	0
HTML コマンドインジェクションログ	0	0
XML 形式のログ	0	0
XML サービス拒否 (XDoS) ログ	0	0
XML メッセージ検証ログ	0	0
WSI ログ	0	0
XML SQL インジェクションログ	0	0
XML クロスサイトスクリプティングログ	0	0
XML 添付ファイルログ	0	0
SOAP 障害ログ	0	0
XML 汎用ログ	0	0
ログメッセージの合計	0	0

サーバエラー応答の統計情報	レート (/s)	合計
HTTP クライアントエラー (4xx Resp)	0	0
HTTP サーバエラー (5xx)	0	0

GUI を使用して Cookie ハイジャックトラフィックと違反の統計情報を表示する

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ウィンドウで、**Web App Firewall** プロファイルを選択し、[統計] をクリックします。
3. **Citrix Web App Firewall** 統計ページには、Cookie のハイジャックトラフィックと違反の詳細が表示されます。
4. 表形式ビューを選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィック形式で表示できます。



SameSite cookie 属性

October 7, 2021

安全なウェブ通信のために、Google は `SameSite` cookie 属性の使用を義務付けています。Google Chrome の新しい遵守することにより `SameSite` 、ポリシーは、Citrix ADC アプライアンスはとサードパーティの Cookie を管理することができ `SameSite` 、内の属性セット `set-cookie` ヘッダー。Cookie 設定は攻撃を軽減し、安全な

Web 通信を提供します。

2020 年 2 月まで、SameSite 属性は Cookie に明示的に設定されていませんでした。ブラウザはデフォルト値を「なし」としました。ただし、Google Chrome 80 などの特定のブラウザのアップグレードでは、Cookie のデフォルトのクロスドメイン動作に変更があります。

Cookie 属性値の設定

SameSite 属性は次のいずれかの値に設定されており、GoogleChrome ブラウザの場合のデフォルト値は「Lax」に設定されています。

なし。安全な接続でのみクロスサイトコンテキストでリクエストに Cookie を使用するようにブラウザを示します。

緩い。同じサイトのコンテキストでリクエストに Cookie を使用するようにブラウザを示します。クロスサイトコンテキストでは、GET リクエストなどの安全な HTTP メソッドのみが Cookie を使用できます。

厳格。ユーザーがドメインを明示的に要求している場合にのみ Cookie を使用してください。

> 注:
>
> set-cookies(ファイアウォールセッション Cookie を含む)に SameSite 属性があり、addcookiesamesite 属性フラグが Web アプリケーションファイアウォールプロファイルで有効になっている場合、SameSite 属性はプロファイルで設定された値に従って上書きされます。

CLI を使用して、Web AppFirewall プロファイルで SameSite 属性を構成します

SameSite 属性を構成するには、次の手順を実行する必要があります。

1. SameSite cookie 属性を有効にします。
2. appfw セッション Cookie の cookie 属性を設定します。

有効にする **'Samesite'** クッキー属性

コマンドプロンプトで入力します。

```
set appfw profile <profile-name> -insertCookieSameSiteAttribute ( ON | OFF)
```

例**:**

```
set appfw profile p1 -insertCookieSameSiteAttribute ON
```

Web アプリケーションファイアウォールセッション **Cookie** に同じサイト **Cookie** 属性値を設定する

コマンドプロンプトで入力します。

```
set appfw profile <profile-name> - cookieSameSiteAttribute ( LAX | NONE |
STRICT )
```


例**:**

```
set appfw profile p1 - cookieSameSiteAttribute LAX
```

属性タイプがある場合、

なし。Cookie 属性 SameSite は「なし」に設定され、すべての WAF およびアプリケーション Cookie に対して安全とマークされます。

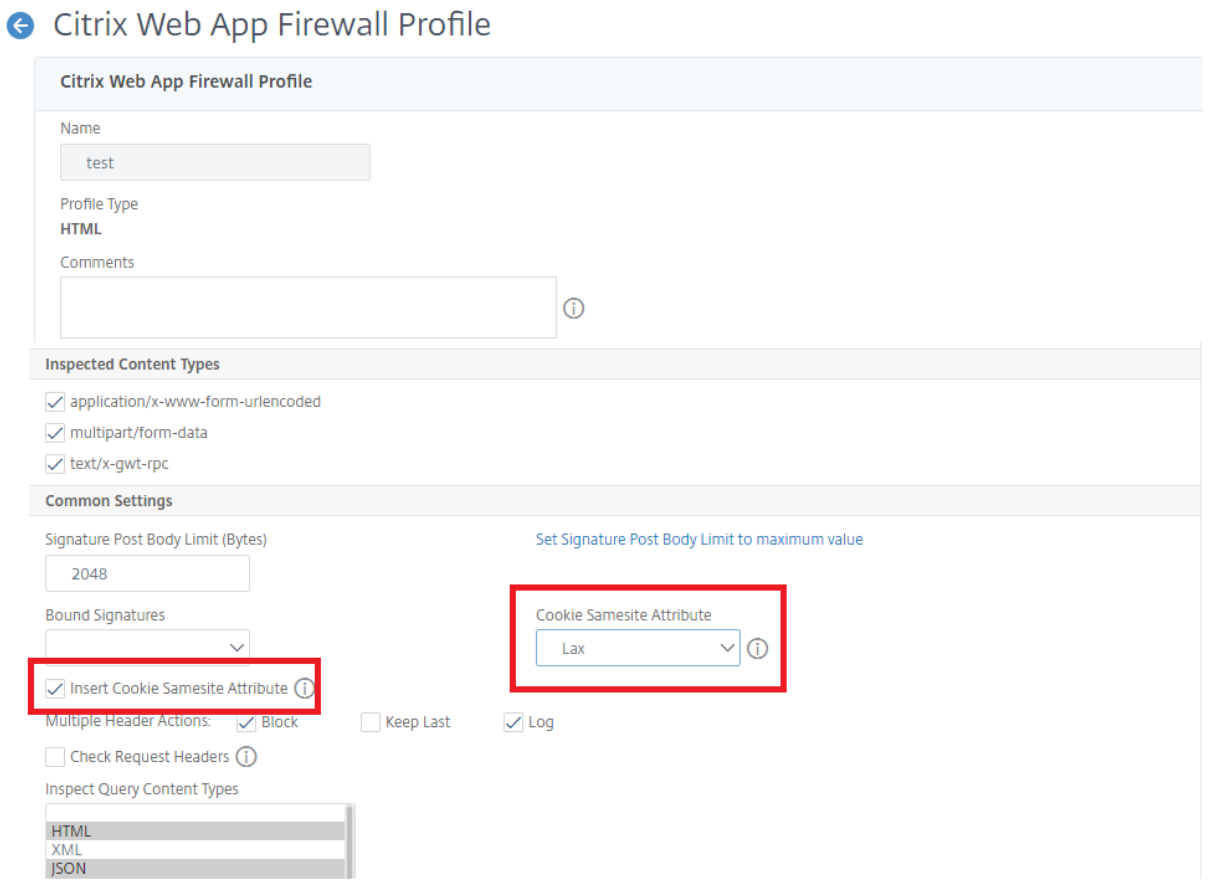
緩い。Cookie 属性 SameSite は、すべての WAF およびアプリケーション Cookie に対して「Lax」に設定されています。

厳格。Cookie 属性 SameSite は、すべての WAF およびアプリケーション Cookie に対して「Lax」に設定されています。

GUI を使用して、**Web AppFirewall** プロファイルで **SameSitecookie** 属性を構成します

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ペインでプロファイルを選択し、[編集] をクリックします。
3. [**Citrix Web App Firewall** プロファイル］ページで、［詳細設定］の下の［プロファイル設定］をクリックします。
4. **[**プロファイル設定**]** セクションで、次のパラメーターを設定します。

   a. cookie `Samesite` 属性を挿入します。チェックボックスを選択して、cookie `Samesite` 属性を有効にします。
   b。CookieSamesite 属性。ドロップダウンリストからオプションを選択して、`Samesite` Cookie 値を設定します。
5. [**OK**] をクリックし、[完了] をクリックします。



データ漏洩防止チェック

October 7, 2021

データ漏洩防止は、フィルタ応答をチェックし、クレジットカード番号や社会保障番号などの機密情報が許可されていない受信者に漏洩しないようにします。

クレジットカード確認

October 7, 2021

クレジットカードを受け入れるアプリケーションがある場合、または Web サイトがクレジットカード番号を格納するデータベースサーバーにアクセスできる場合は、データ漏えい防止 (DLP) 対策を使用し、受け入れるクレジットカードの種類ごとに保護を構成する必要があります。

Citrix Web App Firewall のクレジットカードチェックは、攻撃者がデータ漏洩防止の欠陥を悪用して顧客のクレジットカード番号を取得するのを防ぎます。簡単な設定手順に従うことで、1) ビザ、2) マスターカード、3) ディスカバ

ー、4) アメリカンエキスプレス (アメックス)、5) JCB、6) ダイナースクラブ。

クレジット・カード・セキュリティ・チェックは、サーバーの応答を調べ、ターゲットのクレジット・カード番号のインスタンスを識別し、そのような番号が見つかったときに指定された処理を適用します。アクションは、クレジットカード番号の最後の桁を除くすべての桁数を X'ing して応答を変換したり、指定した数以上のクレジットカード番号が含まれている場合に応答をブロックしたりできます。両方を指定した場合、ブロックアクションが優先されます。[ページごとに許可される最大クレジットカード数] の設定により、ブロックアクションがいつ呼び出されるかが決まります。デフォルト設定である 0 (ページ上のクレジットカード番号は許可されていません) が最も安全ですが、最大 255 まで許可できます。応答で違反が検出され、ブロックアクションがトリガーされる場所によっては、応答で許可されるクレジットカードの最大数よりも少なくなる場合があります。

誤検出を避けるために、リラクゼーションを適用して、クレジットカードの小切手から特定の番号を免除することができます。たとえば、社会保障番号、発注番号、Google アカウント番号は、クレジットカード番号に似ています。個々の番号を指定するか、正規表現を使用して、クレジットカード検査の応答 URL を処理するときにバイパスする桁数を指定します。

免除するクレジットカード番号がわからない場合は、学習機能を使用して、学習したデータに基づいてレコメンデーションを生成できます。パフォーマンスを損なうことなく最適なメリットを得るには、このオプションを短時間有効にしてルールの代表的なサンプルを取得し、緩和を展開して学習を無効にします。

ログ機能を有効にすると、クレジット・カード・チェックによって実行されるアクションを示すログ・メッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、アクセスが阻止された可能性があります。デフォルトでは、doSecureCreditCardLogging パラメータはオンになっているため、安全な商取引（クレジットカード）違反によって生成されたログメッセージにはクレジットカード番号は含まれません。

統計機能は、違反とログに関する統計を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。

アプリケーションを保護するためのクレジット・カード・セキュリティ・チェックを構成するには、このアプリケーションとの間で送受信されるトラフィックの検査を管理するプロファイルを設定します。

> 注:
>
> SQL データベースにアクセスしない Web サイトは、通常、クレジットカード番号などの機密性の高い個人情報にアクセスできません。

コマンドラインを使用したクレジットカードチェックの設定

コマンドラインインターフェイスでは、set appfw profile コマンドまたは add appfw profile コマンドのいずれかを使用して、クレジットカードチェックを有効にし、実行するアクションを指定できます。unset appfw profile コマンドを使用して、デフォルト設定に戻すことができます。緩和を指定するには、bind appfw コマンドを使用して、クレジットカード番号をプロファイルにバインドします。

コマンドラインを使用してクレジットカードチェックを構成するには

次のように、set appfw プロファイルコマンドまたは add appfw プロファイルコマンドを使用します。

- set appfw profile <name> -creditCardAction (([block][learn] [log][stats])| [none])
- set appfw profile <name> -creditCard (VISA | MASTERCARD | DISCOVER | AMEX | JCB | DINERSCLUB)
- set appfw profile <name> -creditCardMaxAllowed <integer>
- set appfw profile <name> -creditCardXOut ([ON] | [OFF])<name> -doSecureCreditCard ([ON] | [OFF])
- コマンドラインを使用してクレジットカード緩和ルールを構成するには

  bind コマンドを使用して、クレジットカード番号をプロファイルにバインドします。プロファイルからクレジットカード番号を削除するには、bind コマンドで使用したものと同じ引数を指定して unbind コマンドを使用します。show コマンドを使用して、プロファイルにバインドされたクレジットカード番号を表示できます。

- クレジットカード番号をプロファイルにバインドするには

  bind appfw profile <profile-name> -creditCardNumber <any number/regex> “<url>”

  例: bind appfw profile test_profile -creditCardNumber 378282246310005 http://www.example.com/credit_card_test.html

  - プロファイルからクレジットカード番号のバインドを解除するには

    unbind appfw profile <profile-name> -creditCardNumber <credit card number / regex> <url>

  - プロファイルにバインドされたクレジットカード番号のリストを表示します。

    show appfw profile <profile>

GUI を使用したクレジットカードチェックの設定

GUI では、アプリケーションに関連付けられたプロファイルのペインでクレジットカードのセキュリティチェックを設定します。

GUI を使用してクレジットカードのセキュリティチェックを追加または変更するには

1. [**Web App Firewall**] > [プロファイル] に移動し、ターゲットプロファイルをハイライト表示して、[編集] をクリックします。
2. [詳細設定] ウィンドウで、[セキュリティチェック] をクリックします。

   セキュリティチェックテーブルには、すべてのセキュリティチェックに対して現在構成されているアクション設定が表示されます。設定には 2 つのオプションがあります。

a) クレジットカードのブロック、ログ、統計、学習の各アクションを有効または無効にするだけの場合は、表のチェックボックスをオンまたはオフにして [**OK**] をクリックし、[保存して 閉じる] をクリックして [セキュリティチェック] ウィンドウを閉じます。

b) このセキュリティー検査の追加オプションを構成する場合は、「クレジット・カード」をダブルクリックするか、行を選択して「アクション設定」をクリックして、次のように追加オプションを表示します。

- [Out]：応答で検出されたクレジットカード番号をマスクします。最後のグループの数字を除く各数字を「X」に置き換えます。
- [ページごとに許可されるクレジットカードの最大数]：ブロックアクションを起動せずにクライアントに転送できるクレジットカードの数を指定します。
- 保護されたクレジットカード。チェックボックスをオンまたはオフにして、クレジットカードの種類ごとに保護を有効または無効にします。
- [クレジットカード設定] ペインで、ブロック、ログ、統計、学習の各アクションを編集することもできます。

  上記の変更を行った後、[OK] をクリックして変更を保存し、[セキュリティチェック] テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。[OK] をクリックして [セキュリティチェック] セクションで行った変更をすべて保存し、[保存して閉じる] をクリックして [セキュリティチェック] ウィンドウを閉じます。

3. [詳細設定] ウィンドウで、[プロファイルの設定] をクリックします。クレジットカード番号のセキュリティで保護されたログを有効または無効にするには、[クレジットカードのログを保護する] チェックボックスをオンまたはオフにします。（デフォルトでは選択されています）。

   [**OK**] をクリックして変更を保存します。

- GUI を使用してクレジットカード緩和ルールを構成するには

  1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
  2. [詳細設定] ペインで、[緩和規則] をクリックします。リラクゼーション・ルール・テーブルには、クレジット・カード・エントリがあります。ダブルクリックするか、この行を選択して「編集」をクリックすると、「クレジット・カード緩和ルール」ダイアログにアクセスできます。リラクゼーションルールの追加、編集、削除、有効化、または無効化の操作を実行できます。

クレジットカード小切手での学習機能の使用

学習アクションが有効の場合、Web App Firewall 学習エンジンはトラフィックを監視し、トリガーされた違反を学習します。これらの学習済みルールを定期的に検査できます。期日を考慮した後、特定の桁数をクレジット・カード・セキュリティ・チェックから除外する場合は、学習したルールを緩和ルールとして展開します。

- コマンドラインインターフェイスを使用して学習データを表示または使用するには

```
show appfw learningdata <profilename> creditCardNumber
```



```
rm appfw learningdata <profilename> -creditcardNumber <credit card
number> "<url>"

export appfw learningdata <profilename> creditCardNumber
```

- GUI を使用して学習したデータを表示または使用するには

  1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
  2. [詳細設定] ウィンドウで、[学習ルール] をクリックします。「学習済みルール」(Learned Rules) テーブルで「クレジットカード」(Credit Card) エントリを選択し、ダブルクリックすると、学習済みルールにアクセスできます。学習したルールを展開するか、緩和ルールとして展開する前にルールを編集できます。ルールを破棄するには、ルールを選択して [スキップ] ボタンをクリックします。一度に編集できるルールは 1 つだけですが、複数のルールを選択して展開またはスキップできます。

  また、「学習済みルール」(Learned Rules) テーブルで「クレジットカード」(Credit Card) エントリを選択し、「ビジュアライザー」(Visualizer) をクリックして、学習済み緩和の要約ビューを表示するオプションもあります。ビジュアライザーを使用すると、学習したルールを簡単に管理できます。データの包括的なビューを 1 つの画面に表示し、1 回のクリックでルールのグループに対するアクションを簡単に実行できます。ビジュアライザーの最大の利点は、複数のルールを統合するために正規表現を推奨することです。区切り文字とアクション URL に基づいて、これらのルールのサブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 のルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。

クレジット・カード・チェックでのログ機能の使用

ログ処理が有効な場合、クレジット・カード・セキュリティ・チェック違反は、APPFW_SAFECOMMERCE_XFORM 違反または APPFW_SAFECOMMERCE_XFORM 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

クレジットカードロギングのデフォルト設定は ON です。OFF に変更すると、クレジットカード番号と種類の両方がログメッセージに含まれます。

クレジット・カード・チェック用に構成された設定によっては、アプリケーションファイアウォールで生成されたログ・メッセージに次の情報が含まれる場合があります。

- 応答がブロックされたか、ブロックされていません。
- クレジットカード番号が変換されました (X'd アウト). 変換されたクレジットカード番号ごとに個別のログメッセージが生成されるため、1 つの応答の処理中に複数のログメッセージが生成される場合があります。
- 応答には、潜在的なクレジットカード番号の最大数が含まれていました。
- クレジットカード番号とそれに対応するタイプ。

- コマンドラインを使用してログメッセージにアクセスするには

  シェルに切り替えて、/var/log/ フォルダの ns.logs を末尾に付けて、クレジットカード違反に関するログメッセージにアクセスします。

  - Shell
  - tail -f /var/log/ns.log | grep SAFECOMMERCE

- GUI を使用してログメッセージにアクセスするには

  1. Citrix GUI には、ログメッセージを分析するための非常に便利なツール（Syslog ビューア）が含まれています。Syslog Viewer にアクセスするには、いくつかのオプションがあります。ターゲットプロファイル > [セキュリティチェック] に移動します。[クレジットカード] 行を強調表示し、[ログ] をクリックします。プロファイルのクレジット・カード・セキュリティ・チェックから直接ログにアクセスすると、ログ・メッセージが除外され、これらのセキュリティ・チェック違反に関連するログのみが表示されます。

  2. また、「**NetScaler** 」>「システム」>「監査」に移動して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティー検査違反がトリガーされる可能性がある場合のデバッグに役立ちます。

     HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。クレジットカードのセキュリティチェック違反ログメッセージにアクセスするには、モジュールのドロップダウン・オプションで「APPFW」を選択してフィルタリングします。[イベントタイプ] には、選択内容をさらに絞り込むための豊富なオプションが表示されます。たとえば、「APPFW_SAFECOMMERCE」および「APPFW_SAFECOMMERCE_XFORM」チェック・ボックスを選択し、「適用」ボタンをクリックすると、クレジット・カードのセキュリティチェック違反に関するログ・メッセージのみが Syslog ビューアに表示されます。

     特定のログメッセージの行にカーソルを置くと、Module や EventType などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログ内の対応する情報を強調表示することができます。

応答がブロックされていない場合のネイティブ形式のログメッセージの例

```
1 May 29 01:26:31 <local0.info> 10.217.31.98 05/29/2015:01:26:31 GMT ns
     0-PPE-0 :
2 default APPFW APPFW_SAFECOMMERCE 2181 0 :  10.217.253.62 1098-PPE0
3 4erNfkaHy0IeGP+nv2S9Rsdu77I0000 pr_ffc http://aaron.stratum8.net/FFC/
     CreditCardMind.html
4 Maximum number of potential credit card numbers seen <not blocked>
5 <!--NeedCopy-->
```


応答の変換時の CEF 形式のログメッセージの例

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE_XFORM|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Transformed (xout) potential credit card numbers seen in server
    response
cn1=66 cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002
cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

応答がブロックされた場合の CEF 形式のログメッセージの例。doSecureCreditCardLogging パラメーターが無効になっているため、クレジットカードの番号と種類はログに記録されます。

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Credit Card number 4505050504030302 of type Visa is seen in
    response cn1=68
cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002 cs4=
    ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

クレジットカード違反の統計

stats アクションが有効な場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、クレジットカードチェックに対応するカウンターが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分は、構成された設定によって異なります。たとえば、ブロックアクションが有効で、[最大許容クレジットカード] 設定が 0 の場合、20 のクレジットカード番号を含むページの要求では、最初のクレジットカード番号が検出されるとすぐに、ページがブロックされると、統計カウンタが 1 ずつ増加します。ただし、ブロックが無効で、トランスフォームが有効な場合、同じ要求を処理すると、ログの統計カウンタが 20 ずつ増加します。これは、クレジットカード変換ごとに個別のログメッセージが生成されるためです。

- コマンドラインを使用してクレジットカード統計を表示するには

  コマンドプロンプトで入力します。


```
sh appfw stats
```

特定のプロファイルの統計情報を表示するには、次のコマンドを使用します。

```
stat appfw profile <profile name>
```

GUI を使用してクレジット・カード統計を表示するには

1. [システム] > [セキュリティ] > [**Web App Firewall**] に移動します。
2. 右側のペインで、[統計リンク] にアクセスします。
3. スクロール・バーを使用して、クレジット・カード違反およびログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

ハイライト

クレジットカードのセキュリティチェックについては、次の点に注意してください。

- Web App Firewall を使用すると、クレジットカード情報を保護し、この機密データにアクセスしようとする試みを検出できます。
- クレジットカード保護チェックを使用するには、クレジットカードの種類とアクションを少なくとも 1 つ指定する必要があります。チェックは、HTML、XML、および Web 2.0 プロファイルに適用されます。
- あなたはすべてのクレジットカード固有の設定を見るために、CreditCard の sh の appfw プロファイルコマンドと grep の出力をパイプすることができます。たとえば、sh appfw プロファイル my_profile | grep CreditCard では、さまざまなパラメータの設定と、my_profile という名前の Web App Firewall プロファイルのクレジットカードチェックに関する緩和ルールが表示されます。
- 残りのクレジットカード番号のセキュリティチェック検査をバイパスすることなく、クレジットカード検査から特定の番号を除外できます。
- リラクゼーションは、すべての Web App Firewall で保護されたクレジットカードパターンでご利用いただけます。GUI では、ビジュアライザを使用して、緩和ルールに対する追加、編集、削除、有効化、無効化操作を指定できます。
- Web App Firewall 学習エンジンは、発信トラフィックを監視して、観察された違反に基づいてルールを推奨できます。ビジュアライザのサポートは、GUI で学習したクレジットカードのルールを管理するためにも利用できます。学習したルールを編集して展開することも、慎重な検査の後にスキップすることもできます。
- 許可されるクレジットカード数の設定は、各応答に適用されます。ユーザーセッション全体で確認されたクレジットカード番号の累積合計には適用されません。
- X 出力桁数は、クレジットカード番号の長さによって異なります。10 桁は、13〜15 桁のクレジットカードの場合は X'd です。16 桁のクレジットカードの場合、12 桁は X'd です。アプリケーションで応答にクレジットカード番号全体を送信する必要がない場合は、この操作を有効にして、クレジットカード番号の数字を隠すこのアクションを有効にすることをお勧めします。
- X-out 操作では、すべてのクレジットカードが変換され、許可されるクレジットカードの最大数に対して構成された設定とは独立して動作します。たとえば、応答に 4 枚のクレジットカードがあり、creditCardMaxAllowed パラメーターが 10 に設定されている場合、4 枚のクレジットカードはすべて X'd-out になりますが、ブロッ

クされません。クレジットカード番号が文書に広がっている場合、応答がブロックされる前に、X'd-out 番号の部分的な応答がクライアントに送信されることがあります。

- 考慮する前に、doSecureCreditCardLogging パラメータを無効にしないでください。このパラメータをオフにすると、クレジットカード番号が表示され、ログメッセージでアクセスできます。X-out アクションが有効になっている場合でも、これらの数字はログにマスクされません。ログをリモート syslog サーバに送信し、ログが危険にさらされると、クレジットカード番号が公開される可能性があります。
- クレジットカード違反のため応答ページがブロックされると、Web App Firewall はエラーページにリダイレクトされません。

セーフオブジェクトチェック

October 7, 2021

セーフオブジェクトチェックでは、顧客番号、注文番号、国固有または地域固有の電話番号または郵便番号などの重要なビジネス情報をユーザーが構成可能な保護を提供します。ユーザー定義の正規表現またはカスタムプラグインは、Web App Firewall にこの情報の形式を伝え、保護に使用するルールを定義します。ユーザー要求内の文字列が安全なオブジェクト定義と一致する場合、Web App Firewall は、特定のセーフオブジェクトルールの設定方法に応じて、応答をブロックするか、保護された情報をマスクするか、または応答から保護された情報を削除してからユーザーに送信します。

セーフオブジェクトチェックは、攻撃者が Web サーバーソフトウェアまたは Web サイトのセキュリティ上の欠陥を悪用して、会社のクレジットカード番号や社会保障番号などの機密性の高い個人情報を取得することを防ぎます。Web サイトがこれらのタイプの情報にアクセスできない場合は、このチェックを構成する必要はありません。そのような情報にアクセスできるショッピングカートやその他のアプリケーションがある場合、または Web サイトがそのような情報を含むデータベースサーバーにアクセスできる場合は、処理および保存する機密性の高い個人情報の種類ごとに保護を構成する必要があります。

> 注:
>
> SQL データベースにアクセスしない Web サイトは、通常、機密性の高い個人情報にアクセスできません。

[セーフオブジェクトチェック] ダイアログボックスは、他のチェックの場合とは異なります。作成する各セーフオブジェクト式は、クレジットカードチェックと同様に、その種類の情報に対する個別のセキュリティチェックに相当します。ウィザードまたは GUI を使用する場合は、[追加] をクリックし、[セーフオブジェクトの追加] ダイアログボックスで式を構成して、新しい式を追加します。既存の式を変更するには、その式を選択して [開く] をクリックし、[セーフオブジェクトの修正] ダイアログボックスで式を設定します。

各セーフオブジェクト式の [セーフオブジェクト] ダイアログボックスで、次の項目を設定できます。

- セーフオブジェクト名。新しいセーフオブジェクトの名前。名前は、文字、数字、またはアンダースコア記号で始まり、1 ～255 文字の文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号で構成されます。

- アクション。ブロック、ログ、統計のアクション、および次のアクションを有効または無効にします。
  - **X-Out**。セーフオブジェクト式に一致する情報を文字「X」でマスクします。
  - **Remove-** セーフオブジェクト式に一致する情報をすべて削除します。
- 正規表現。セーフオブジェクトを定義する PCRE 互換の正規表現を入力します。正規表現を作成するには、テキストボックスに正規表現を直接入力する方法、[正規表現 トークン] メニューを使用して正規表現の要素とシンボルをテキストボックスに直接入力する方法、または [正規表現エディタ] を開いて式を作成します。正規表現は ASCII 文字のみで構成する必要があります。基本的な 128 文字の ASCII セットに含まれていない文字は、切り取って貼り付けないでください。非 ASCII 文字を含める場合は、これらの文字を PCRE 16 進文字エンコード形式で手動で入力する必要があります。
  メモ: セーフオブジェクト式の先頭に開始アンカー (^)、セーフオブジェクト式の末尾に終了アンカー ($) を使用しないでください。これらの PCRE エンティティは Safe Object 式ではサポートされていません。使用すると、式が一致するものと一致しなくなります。
- **[**最大一致長**]**。照合する文字列の最大長を表す正の整数を入力します。たとえば、米国の社会保障番号を照合する場合は、このフィールドに番号 11 (11) を入力します。これにより、正規表現は 9 つの数字と 2 つのハイフンを持つ文字列にマッチすることができます。カリフォルニア州の運転免許証番号を照合する場合は、8 と入力します。

  > 注意:
  >
  > このフィールドに最大一致長を入力しない場合、Web App Firewall では、安全なオブジェクト式に一致する文字列をフィルタリングするときに、デフォルト値の 1 が使用されます。その結果、ほとんどの安全なオブジェクト式はターゲット文字列と一致しません。

コマンドラインインターフェイスを使用して、セーフオブジェクトチェックを構成することはできません。Web App Firewall ウィザードまたは GUI を使用して構成する必要があります。

以下は、セーフオブジェクトチェック正規表現の例です。

- 米国の社会保障番号であると思われる文字列を探します。3 つの数字（最初の数字はゼロであってはいけません）、ハイフン、さらに 2 つの数字、2 番目のハイフン、さらに 4 つの数字で終わる文字列で終わります。

```
[1-9][0-9]{
3,3 }
-[0-9]{
2,2 }
-[0-9]{
4,4 }

<!--NeedCopy-->
```

- カリフォルニア州の運転免許証 ID のように見える文字列を探します。文字列は文字で始まり、その後に正確に 7 つの数字の文字列が続きます。


```
[A-Za-z][0-9]{
7,7 }

<!--NeedCopy-->
```

- 顧客 ID と思われる文字列を探します。これらの文字列は、5 つの 16 進文字 (すべての数字と A から F までの文字)、ハイフン、3 文字のコード、2 番目のハイフン、10 個の数字の文字列で終わる文字列で構成されます。

```
[0-9A-Fa-f]{
5,5 }
-[A-Za-z]{
3,3 }
-[0-9]{
10,10 }

<!--NeedCopy-->
```

> 注意:
>
> 正規表現は強力です。特に PCRE-format の正規表現に精通していない場合は、記述する正規表現をダブルチェックして、安全なオブジェクト定義として追加する文字列のタイプを正確に定義していることを確認してください。それ以外は何も定義しません。ワイルドカード、特にドットとアスタリスク (.*) メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、ブロックする意図がなかった Web コンテンツへのアクセスをブロックするなど、望ましくない結果が生じる可能性があります。

高度なフォーム保護チェック

October 7, 2021

高度なフォーム保護チェックは、Web フォームデータを調べて、攻撃者が Web サイトの Web フォームを変更したり、予期しない種類と量のデータをフォームで Web サイトに送信したりして、システムを危険にさらすことを防ぎます。

> 注
>
> [セキュリティチェックからアップロードファイルを除外する] が設定されていない場合、SQL、クロスサイトスクリプティング、FFC、および FieldFormat 保護チェックが適用されます。

ファイルのアップロードは、フォーム送信の一部として送信されるコントロール 名フィールドを持つフォーム要素でもあります。

詳細については、このページを参照してください。Forms

フィールド形式のチェック

October 7, 2021

[フィールド形式] チェックは、ユーザーが Web フォームで Web サイトに送信するデータを確認します。データの長さと型の両方を調べて、データが表示されるフォームフィールドに適していることを確認します。Web App Firewall は、ユーザーリクエストで不適切な Web フォームデータを検出すると、リクエストをブロックします。

攻撃者が不適切な Web フォームデータを Web サイトに送信するのを防ぐことにより、フィールドフォーマットチェックは Web サイトおよびデータベースサーバーに対する特定の種類の攻撃を防ぎます。たとえば、特定のフィールドでユーザが電話番号を入力することが予想される場合、[Field Formats] チェックは、ユーザが送信した入力を調べ、データが電話番号の形式と一致することを確認します。特定のフィールドにファーストネームが必要な場合は、[Field Formats] チェックによって、そのフィールドのデータがファーストネームに適したタイプと長さであることが確認されます。これは、保護するように構成するフォームフィールドごとに同じことを行います。

このチェックは HTML リクエストにのみ適用されます。XML リクエストには適用されません。HTML プロファイルまたは Web 2.0 プロファイルでフィールド形式チェックを構成して、アプリケーションを保護するための HTML ペイロードを検査できます。Web App Firewall は、Google Web Toolkit (GWT) アプリケーションのフィールドフォーマットチェック保護もサポートしています。

[フィールド形式] チェックでは、1 つ以上のアクションを有効にする必要があります。Web App Firewall は、送信された入力を調べ、指定されたアクションを適用します。

> 注
>
> フィールド書式ルールは締め付けルールです。学習したデータから緩和リストに追加すると、ブロック規則として機能します。
>
> フィールド形式の規則を緩和するには、フィールド形式の緩和リストから特定の「フィールド名」を削除してください。

デフォルトのフィールド形式を設定して、保護する各 Web フォームの各フォームフィールドで想定されるデータの最小長と最大長を指定します。緩和ルールを展開して、特定のフォームの個々のフィールドに対して [フィールド形式] を設定できます。複数のルールを追加して、フィールド名、アクション URL、およびフィールド形式を指定できます。異なるフォームフィールドに異なるタイプの入力を受け入れるには、フィールドフォーマットを指定します。学習機能では、緩和規則の推奨事項を提供できます。

フィールド形式のアクション-ブロック、ログ、統計、学習の各アクションを有効にできます。フィールド形式チェック保護を使用するには、これらのアクションの少なくとも 1 つを有効にする必要があります。

- ブロック。ブロックを有効にすると、入力が指定されたフィールド形式に準拠していない場合に、ブロックアクションがトリガされます。ターゲットフィールドにルールが設定されている場合は、指定されたルールに対

して入力がチェックされます。それ以外の場合は、デフォルトのフィールド形式指定と照らしてチェックされます。フィールドタイプまたは最小/最大長の指定に不一致があると、要求がブロックされます。

- ログ。ログ機能を有効にすると、Field Format チェックによって実行されるアクションを示すログメッセージが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとする悪意のある試みを示している可能性があります。
- 統計。有効にすると、統計機能によって違反とログに関する統計情報が収集されます。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示しているか、指定されたフィールド形式が過度に制限されているかどうかを確認するために設定を再確認する必要がある可能性があります。
- 学ぶ。どのフィールドタイプまたは最小および最大長の値がアプリケーションに最適であるかが不明な場合は、学習機能を使用して、学習データに基づいて推奨事項を生成できます。Web App Firewall 学習エンジンは、トラフィックを監視し、観測値に基づいてフィールド形式の推奨事項を提供します。パフォーマンスを損なうことなく最適なメリットを得るには、ラーニングオプションを短時間有効にして、ルールの代表的なサンプルを取得してから、ルールを展開してラーニングを無効にします。
  注:Web App Firewall の学習エンジンは、名前の最初の 128 バイトのみを区別できます。フォームに、最初の 128 バイトに一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別できないことがあります。同様に、展開された緩和ルールは、そのようなフィールドをすべて誤って緩和する可能性があります。

[**Default Field Format**]— アクションの設定に加えて、デフォルトの [Field Format] を設定して、アプリケーションのすべてのフォームフィールドで想定されるデータのタイプを指定できます。[フィールドタイプ] は、[フィールドフォーマット] タイプとして選択できます。[最小長] および [最大長] パラメータを使用して、許容される入力の長さを指定できます。[フィールドタイプ] の代わりに、キャラクタマップを使用して、フィールドで許可される内容を指定することもできます (クラスタ展開を除く)。

- **[Field Type]**: フィールドタイプは名前付き式で、割り当てられたプライオリティ値を割り当てます。Field Type 式は、許可された入力を指定し、送信されたデータに対して照合され、受信した値が許可された値と一致しているかどうかを判断します。[フィールドタイプ] は、優先順位番号順にチェックされます。数値が小さいほど、プライオリティが高くなります。Web App Firewall には、独自のフィールドタイプを追加し、必要な優先順位を割り当てるオプションがあります。プライオリティ値の範囲は 0 ～64000 です。構成プロセスを簡素化するために、次の組み込みのフィールドタイプが用意されています。

```
1  > sh appfw fieldtype
2  1)      Name:  integer           Regex:  "^[+-]?[0-9]+$"
3                  Priority:  30            Comment:  Integer
4                  Builtin:  IMMUTABLE
5  2)      Name:  alpha             Regex:  "^[a-zA-Z]+$"
6                  Priority:  40            Comment:  "Alpha
                      characters"
7               Builtin:  IMMUTABLE
8  3)      Name:  alphanum          Regex:  "^[a-zA-Z0-9]+$"
9                  Priority:  50            Comment:  "Alpha-numeric
```



```
                        characters”
10                   Builtin:  IMMUTABLE
11   4)      Name:  nohtml            Regex:  ”^[^&<>]\*$”
12                   Priority:  60            Comment:  ”Not HTML”
13                   Builtin:  IMMUTABLE
14   5)      Name:  any               Regex:  ”^.\*$”
15                Priority:  70            Comment:  Anything
16                   Builtin:  IMMUTABLE
17       Done
18   >
19   <!--NeedCopy-->
```

> 注：組み込みのフィールドタイプは不変です。変更や削除はできません。追加したフィールドタイプはすべて編集可能です。これらの項目は編集または削除できます。

アプリケーションのフォームフィールドのすべてまたは大部分の有効な入力を識別し、無効な入力を除外できる PCRE 式がある場合に、Field Type をデフォルトのフィールド形式として設定すると便利です。例えば、申請書のすべての入力に数字と文字のみが含まれていることが予想される場合、デフォルトのフィールドタイプとして組み込みのフィールドタイプの英数字を使用することができます。バックスラッシュ（）やセミコロンなどの英数字以外の文字。入力では、違反が発生します。また、独自にカスタマイズしたフィールドタイプを追加し、それを使用してデフォルトのフィールド形式を構成することもできます。たとえば、小文字の「x」、「y」、および「z」を唯一のアルファ文字にする場合は、正規表現「^[x-z]+$」を使用して、カスタマイズされたフィールドタイプを設定できます。組み込みのフィールドタイプよりも高い優先度（低い優先度番号）を割り当てて、デフォルトのフィールドタイプとして使用できます。

- 「最小長」— 明示的な設定を持たない Web フォームのフォームフィールドに割り当てられるデフォルトの最小データ長。このパラメータはデフォルトで 0 に設定されており、ユーザはフィールドを空白のままにできます。高い設定を行うと、ユーザーは強制的にフィールドに入力されます。

  注意: 最小長の値が 0 で、[フィールドタイプ] が整数、アルファベット、または英数字の場合、最小長の設定にもかかわらず、入力フィールドが空のままであると、要求はブロックされます。これは、これらのフィールド型の regEx には、1 つ以上の文字を意味する + 文字が含まれているためです。整数をアルファ文字と区別するには、少なくとも 1 文字が必要です。

- 最大長— 明示的な設定を持たない Web フォームのフォームフィールドに割り当てられるデフォルトの最大データ長。このパラメータは、デフォルトで 65535 に設定されています。

  > 注：文字対バイト。フィールド形式の最小長と最大長は、文字数ではなく、バイト数を表します。1 バイト文字表現よりも大きい言語では、最大値に設定された文字数よりも少ない文字数で制限を超えることがあります。たとえば、2 バイト文字表現の場合、最大値 9 では 4 文字以下を使用できます。
  > ヒント： GUI を使用すると、UTF-8 文字を 16 進数に変換することなく、直接 GUI にカットアンドペーストすることができます。

- 文字マップ: フィールドタイプの推奨に加えて、Web App Firewall 学習エンジンには、書式チェックルールを展開するための [文字マップの使用] という追加オプションがあります。文字コード表は、特定のフォームフ

ィールドで使用できるすべての文字のセットです。文字マップを使用して、特定の文字を許可または禁止するように、フィールド形式の仕様を微調整できます。フォームフィールドごとに個別の文字コード表が生成されます。文字マップでは、英文字と数字の扱いが異なります。入力にアルファ文字がある場合、すべてのアルファ文字 [A-za-z] がキャラクタマップで推奨される PCRE 式によって許可されます。同様に、数字が含まれている場合は、[0 ～9] の数字がすべて許可されます。印刷できない文字は、x 構造体を使用して指定します。0 ～255 の値を持つ 1 バイト文字のみが、文字コード表推奨の対象となります。

文字コード表は、対応するフィールドタイプの推奨事項よりも具体的に指定できます。状況によっては、キャラクタマップを使用すると、入力として許可されるキャラクタのセットをより厳密に制御できるため、より適切なオプションになる場合があります。展開された文字マップは、接頭辞「CM」で始まり、数字が続く文字列として表示されます。文字マップの優先順位は 10000 から始まります。ユーザーが追加したフィールドタイプと同様に、文字コード表を追加、編集、または削除できます。デプロイされたルールで現在使用されている文字マップは、変更または削除できません。

注：キャラクタマップは、クラスタ配置ではサポートされていません。

> 注
>
> 組み込みのフィールドタイプを含むフィールド書式ルールを追加し、フィールドタイプの代わりに文字マップを使用して保存すると、変更は保存されず、ルールが [フィールドタイプ] に表示されます。
>
> 文字マップが組み込み型の 1 つと一致すると、新しい文字マップを作成する代わりにフィールド型が再利用されます。

コマンドラインを使用したフィールド形式のチェックの設定

コマンドラインインターフェイスでは、add appfw fieldtype コマンドを使用して、新しいフィールドタイプを追加できます。set appfw profile コマンドまたは add appfw profile コマンドのいずれかを使用して、フィールド形式のチェックを設定し、実行するアクションを指定できます。unset appfw profile コマンドを使用して、構成済みの設定をデフォルトに戻すことができます。フィールド形式ルールを指定するには、bind appfw コマンドを使用して、フィールドの型をフォームの Field およびアクション URL に、最小長と最大長の指定とともにバインドします。

コマンドラインを使用して、フィールドの種類を追加、削除、または表示するには、次の操作を行います。

add コマンドを使用して、フィールドタイプを追加します。新しいフィールドタイプを追加するときは、名前、正規表現、優先度を指定する必要があります。コメントを追加するオプションもあります。show コマンドを使用して、設定されたフィールドタイプを表示できます。また、remove コマンドを使用して、フィールドの種類を削除することもできます。このコマンドでは、フィールドの種類の名前のみが必要です。

```
add [appfw] fieldType <name> <regex> <priority> [-comment <string>]
```

各項目の意味は次の通りです：

<regex> は正規表現です

<priority> は正の整数です

例:


```
add fieldtype ”Cust_Zipcode” ”^[0-9]{
 5 }
 [-][0-9]{
 4 }
 $” 4

-  show [appfw] fieldType [<name>]

    Example: sh fieldtype

    sh appfw fieldtype

    sh appfw fieldtype cust_zipcode

-  `rm [appfw] fieldType <name>`

    Example: rm fieldtype cusT_ziPcode

    `rm appfw fieldtype cusT_ziPcode`
<!--NeedCopy-->
```

> 注：上記のように、コマンドで「appfw」の使用はオプションです。たとえば、「フィールドタイプの追加」または「appfw フィールドタイプの追加」はどちらも有効なオプションです。フィールドタイプの名前は、正規化のために大文字と小文字を区別しません。上記の例に示すように、[郵便番号]、[郵便番号]、[郵便番号]、および [郵便番号] は、同じフィールドタイプを参照しています。

コマンドラインを使用してフィールド形式のチェックを構成するには

次のように、set appfw プロファイルコマンドまたは add appfw プロファイルコマンドを使用します。

- `set appfw profile <name> -fieldFormatAction (([block] [learn] [log] [stats])| [none])`
- `set appfw profile <name>-defaultFieldFormatType <string>`
- `set appfw profile <name> -defaultFieldFormatMinLength <integer>`
- `set appfw profile <name> -defaultFieldFormatMaxLength <integer>`

コマンドラインを使用してフィールド形式緩和ルールを構成するには

```
bind appfw profile <name> (-fieldFormat <string> <formActionURL>  <
    fieldType>
[-fieldFormatMinLength <positive_integer>]  [-fieldFormatMaxLength <
    positive_integer>]
[-isRegex ( REGEX | NOTREGEX )])
```



```
<!--NeedCopy-->
```

例:

```
bind appfw profile pr_ffc -fieldFormat "login_name" ".*/login.php"
    integer -fieldformatMinLength 3 -FieldformatMaxlength 6
<!--NeedCopy-->
```

GUI を使用したフィールドフォーマットのセキュリティチェックの設定

GUI では、フィールドタイプを管理できます。また、アプリケーションに関連付けられたプロファイルのペインで [Field Formats] セキュリティチェックを構成することもできます。

GUI を使用してフィールドタイプを追加、変更、または削除するには

1. 「アプリケーションファイアウォール」ノードに移動します。「設定」で「フィールドタイプの管理」をクリックして、「アプリケーションファイアウォールのフィールドタイプの設定」ダイアログを表示します。
2. [追加] をクリックして、新しいフィールドタイプを追加します。このペインの指示に従って、[Create] をクリックします。展開済みのルールで現在使用されていない場合は、ユーザーが追加したフィールドタイプを編集または削除することもできます。

GUI を使用してフィールド形式のセキュリティチェックを追加または変更するには

1. アプリケーションファイアウォールに移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ウィンドウで、[セキュリティチェック] をクリックします。

   セキュリティチェックテーブルには、すべてのセキュリティチェックに対して現在構成されているアクション設定が表示されます。設定には 2 つのオプションがあります。

   a) [フィールド書式] の **[**ブロック**]**、[ログ]、[統計]、および [学習] アクションを有効または無効にするだけの場合は、テーブルのチェックボックスをオンまたはオフにして [**OK**] をクリックし、[保存して閉じる] をクリックして [セキュリティ][チェック] ペイン。

   b) このセキュリティー検査の追加オプションを構成する場合は、「フィールド書式」をダブルクリックするか、行を選択して「アクション設定」をクリックして、「デフォルトのフィールド書式」に次のオプションを表示します。

   - [**Field Type**]: デフォルトのフィールドタイプとして設定するフィールドタイプを選択します。組み込みのフィールド型とユーザー定義フィールド型を選択できます。展開された文字マップもリストに含まれており、選択することができます。
   - **[Minimum Length]**: 各フィールドに入力する必要がある最小文字数を指定します。指定可能な値は 0 ～65535 です。

- 「最大長」— 各フィールドに入力する必要がある最大文字数を指定します。指定可能な値は 1 ～ 65535 です。
  [フィールド形式設定] ペインで、ブロック、ログ、統計、および 学習アクションを編集することもできます。

上記の変更を行った後、[**OK**] をクリックして変更を保存し、[セキュリティチェック] テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。［**OK**］をクリックして［セキュリティチェック］セクションで行ったすべての変更を保存し、［保存して閉じる］をクリックして［セキュリティチェック］ウィンドウを閉じます。

GUI を使用してフィールドフォーマット緩和ルールを構成するには

1. アプリケーションファイアウォールに移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定**]** ペインで、[緩和規則] をクリックします。[緩和規則] テーブルには、[フィールド形式] エントリがあります。[フィールド形式] [緩和規則] ダイアログにアクセスするには、ダブルクリックするか、この行を選択して [編集] ボタンをクリックします。リラクゼーションルールの追加、編集、削除、有効化、または無効化の操作を実行できます。

すべての緩和規則をまとめて表示するには、[フィールド形式] 行をハイライト表示して [ビジュアライザー] をクリックします。展開されたリラクゼーションのビジュアライザーには、新しいルールを追加するか、既存のルールを編集するオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループを有効または無効にすることもできます。

フィールド形式チェックでの学習機能の使用

学習アクションが有効の場合、Web App Firewall 学習エンジンはトラフィックを監視し、トリガーされた違反を学習します。これらの学習済みルールを定期的に検査できます。十分に検討した後、学習したルールをフィールド形式緩和ルールとして展開できます。

フィールド形式の学習機能拡張：リリース 11.0 で、Web App Firewall 学習機能拡張が導入されました。以前のリリースでは、学習されたフィールド形式の推奨事項がデプロイされると、Web App Firewall 学習エンジンは、新しいデータポイントに基づいて新しいルールを推奨する目的で、有効な要求の監視を停止します。これにより、設定済みのセキュリティ保護が制限されます。これは、学習データベースには、セキュリティチェックによって処理された有効な要求に見られる新しいデータの表現が含まれないためです。

違反は学習と結びつくことはなくなりました。学習エンジンは、違反に関係なく、フィールド形式の推奨事項を学習し、提示します。ブロックされたリクエストをチェックして現在のフィールド形式が制限的すぎて緩和する必要があるかどうかを判断するだけでなく、学習エンジンは許可されたリクエストを監視して、現在のフィールド形式が過度に許可されているかどうかを判断し、より多くの制限ルール。

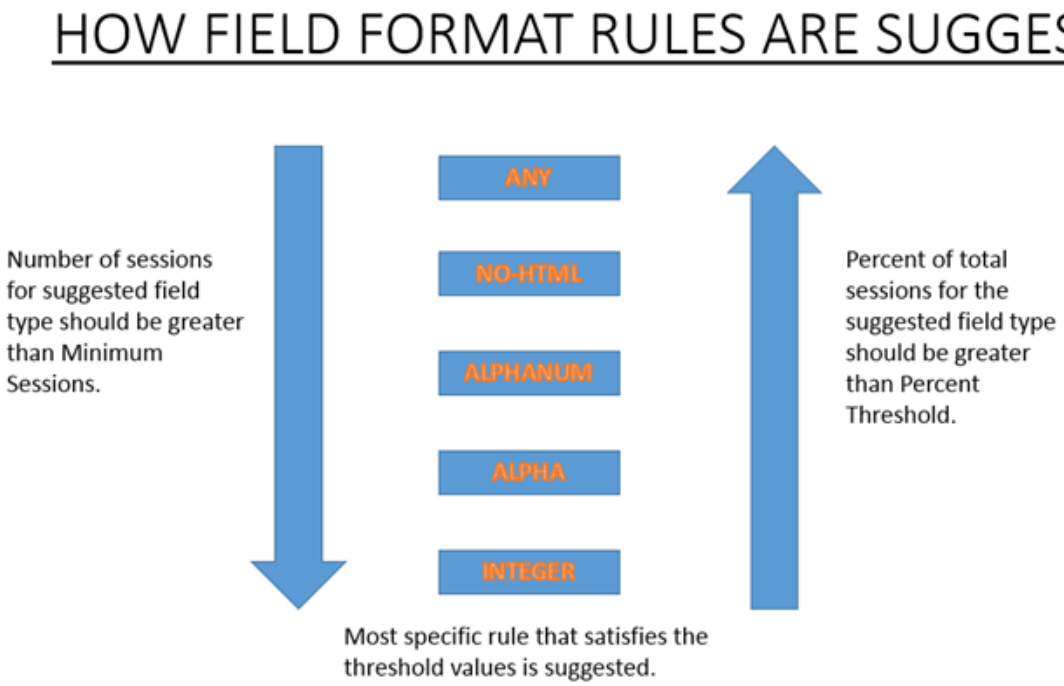
次に、フィールド形式の学習動作の概要を示します。

[フィールド形式がバインドされていません]: このシナリオでは、動作は変更されません。すべての学習データは aslearn エンジンに送信されます。学習エンジンは、データセットに基づいてフィールド形式ルールを提案します。

フィールド形式がバインドされている: 以前のリリースでは、観察されたデータは違反の場合にのみ aslearin エンジンに送信されます。学習エンジンは、データセットに基づいてフィールド形式ルールを提案します。11.0 リリースでは、違反がトリガーされなくても、すべてのデータが aslearn エンジンに送信されます。学習エンジンは、受信したすべての入力のデータセット全体に基づいて、フィールド形式ルールを提案します。

学習強化のためのユースケース:

学習した初期フィールド形式のルールがデータの小さなサンプルに基づいている場合、非典型的な値がいくつかあると、ターゲットフィールドに対して余りな勧告になることがあります。進行中の学習により、Web App Firewall は各リクエストのデータポイントを観察し、学習した推奨事項の代表的なサンプルを収集できます。これは、適切な範囲値で最適な入力フォーマットを展開するためのセキュリティをさらに強化するのに役立ちます。



フィールド形式の学習では、フィールドタイプのプライオリティと、次の学習しきい値の構成済み設定が使用されます。

- **FieldFormatMinThreshold**— 学習された緩和が生成されるまでに、特定のフォームフィールドが観察されなければならない最小回数。デフォルトは 1 です。
- **FieldFormatPercentThreshold**—学習された緩和が生成される前に、フォームフィールドが特定のフィールドタイプに一致した回数の割合。デフォルトは 0 です。

フィールド形式規則の推奨事項は、次の基準に基づいています。

- フィールドタイプの推奨事項: フィールドタイプの推奨は、既存のフィールドタイプの割り当てられた優先順位と、指定されたフィールドフォーマットのしきい値によって決まります。優先度によって、フィールドタイプが入力に対して照合される順序が決まります。数値が小さいほど、プライオリティが高くなります。たとえば、フィールドタイプ整数の方が優先度 (30) が高いため、フィールドタイプの英数字 (50) よりも先に評価されます。しきい値は、データポイントの代表的なサンプルを収集するために評価される入力の数を決定します。

設定済みのフィールドタイプに適切な優先順位を割り当てて、**fieldFormatPercentThreshold** パラメータおよび **fieldFormatMinThreshold** パラメータに適切な 学習設定値を設定することは、正しいフィールドフォーマットの推奨事項を取得するために不可欠です。設定されたしきい値に基づいて、プライオリティが最も高いフィールドタイプが最初に入力と照合されます。一致がある場合は、他のフィールドタイプを考慮せずにこのフィールドタイプが提案されます。たとえば、3 つの既定のフィールドタイプ (整数、英数字、および any) は、すべての入力に数字のみが含まれている場合に一致します。ただし、整数は優先順位が最も高いため推奨されます。

- 最小長と最大長の推奨：フィールド・フォーマットの最小長と最大長の計算は、フィールド・タイプの決定とは無関係に行われます。フィールド形式の長さの計算は、観測されたすべての入力の平均長さに基づいています。この計算された平均値の半分が最小値として示唆され、この平均値の 2 倍が最大値として提案されます。[最小長] の範囲は 0 ～65535 で、[最大長] の範囲は 1 ～65535 です。最小長に設定した値は、最大長を超えることはできません。
- スペース文字の処理：フィールド形式のチェックでは、フィールド形式の長さをチェックするときに、すべてのスペース文字をカウントします。先頭または末尾のスペースは削除されず、入力文字列の中央にある複数の連続したスペースは、入力処理中に 1 つのスペースに統合されなくなりました。

フィールド形式の推奨事項を説明する例:

```
Total requests: 100
Number of Req with Field Type:
Int : 22             (22 int values)  -  22%
Alpha : 44                    (44 alpha values)  -  44%
Alphanum: 14             (14 + 44 + 22 = 80 alphanum values) = 80%
noHTML: 10               (80 + 10 = 90 noHTML values) = 90%
any : 10                      (90 + 10 = 100 any values) = 100%

% threshold                                  Suggested Field Type
0-22                 int
23-44                alpha
45-80                alphanum
81-90                noHTML
91-100               any
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して学習データを表示または使用するには

```
show appfw learningdata <profilename> FieldFormat
rm appfw learningdata <profilename> -fieldFormat <string>  <
    formActionURL>
export appfw learningdata <profilename> FieldFormat
<!--NeedCopy-->
```


GUI を使用して学習したデータを表示または使用するには

1. アプリケーションファイアウォールに移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ウィンドウで、[学習ルール] をクリックします。「学習済みルール」(Learned Rules) テーブルの「フィールドフォーマット」(Field Formats) エントリを選択し、ダブルクリックすると、学習済みルールにアクセスできます。学習したルールを展開するか、緩和ルールとして展開する前にルールを編集できます。ルールを破棄するには、ルールを選択して [スキップ] ボタンをクリックします。一度に編集できるルールは 1 つだけですが、複数のルールを選択して展開またはスキップできます。

   また、「学習済みルール」(Learned Rules) テーブルの「フィールドフォーマット」(Field Formats) エントリを選択し、「ビジュアライザー」(Visualizer) をクリックして、学習済みのすべての違反の統合ビューを表示することもできます。ビジュアライザーを使用すると、学習したルールを簡単に管理できます。データの包括的なビューを 1 つの画面に表示し、1 回のクリックでルールのグループに対するアクションを簡単に実行できます。ビジュアライザーの最大の利点は、複数のルールを統合するために正規表現を推奨することです。区切り文字とアクション URL に基づいて、これらのルールのサブセットを選択できます。ドロップダウンリストから番号を選択すると、ビジュアライザーに 25、50、または 75 のルールを表示できます。学習したルールのビジュアライザーには、ルールを編集して緩和として展開するオプションがあります。または、ルールをスキップして無視することもできます。

フィールド形式チェックでのログ機能の使用

ログアクションを有効にすると、フィールド形式のセキュリティチェック違反が APPFW_FIELDFORMAT 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、/var/log/ フォルダ内の ns.logs を末尾に付けて、フィールド形式違反に関連するログメッセージにアクセスします。

- `Shell`
- `tail -f /var/log/ns.log | grep APPFW_FIELDFORMAT`

GUI を使用してログメッセージにアクセスするには

Citrix GUI には、ログメッセージを分析するための非常に便利なツール（Syslog ビューア）が含まれています。Syslog ビューアには、次の複数のオプションがあります。

- [アプリケーションファイアウォール] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。[フィールド形式] 行をハイライト表示し、[ログ] をクリックします。プロファイルの **Field Formats** セキュリティチェックから直接ログにアクセスすると、ログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。

- 「**Citrix ADC** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして **Syslog Viewer** を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティー検査違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- [アプリケーションファイアウォール] > [ポリシー] > [監査] に移動します。[監査メッセージ] セクションで、[Syslog メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

  HTML ベースの Syslog ビューアには、関心のあるログメッセージのみを選択するためのさまざまなフィルタオプションがあります。[フィールド形式] セキュリティチェック違反ログメッセージにアクセスするには、[モジュール] のドロップダウンオプションで [APPFW] を選択してフィルタリングします。[イベントタイプ] には、選択内容をさらに絞り込むための豊富なオプションが表示されます。たとえば、[**APPFW_FIELDFORMAT**] チェックボックスをオンにして [適用] ボタンをクリックすると、[フィールド形式] セキュリティチェック違反に関連するログメッセージのみが Syslog Viewer に表示されます。

特定のログメッセージの行にカーソルを置くと、Module や EventType などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログ内の対応する情報を強調表示することができます。

要求がブロックされていない場合のネイティブ形式のログメッセージの例

```
Jun 10 22:32:26 <local0.info> 10.217.31.98 06/10/2015:22:32:26 GMT ns
    0-PPE-0 :
default APPFW APPFW_FIELDFORMAT 97 0 :  10.217.253.62 562-PPE0
x1MV+YnNGzQFM3Bsy2wti4bhXio0001 pr_ffc http://aaron.stratum8.net/FFC/
    login_post.php
Field format check failed for field passwd=”65568888sz-*_” <not blocked
    >
Example of a CEF format log message when the request is blocked
Jun 11 00:03:51 <local0.info> 10.217.31.98
CEF:0|Citrix|Citrix ADC|NS11.0|APPFW|APPFW_FIELDFORMAT|6|src
    =10.217.253.62 spt=27076
method=POST requet=http://aaron.stratum8.net/FFC/maxlen_post.php msg=
    Field format check
failed for field text_area=”” cn1=108 cn2=644 cs1=pr_ffc cs2=PPE0
cs3=GaUROfl1Nx1jJTvja5twH5BBqI0000 cs4=ALERT cs5=2015 act=blocked
<!--NeedCopy-->
```


フィールド形式違反の統計情報

stats アクションが有効な場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、Field Formats チェックに対応するカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分は、構成された設定によって異なります。たとえば、ブロックアクションが有効になっている場合、3 つのフィールド形式違反を含むページに対する要求では、最初のフィールド形式違反が検出されるとすぐにページがブロックされるため、stats カウンタが 1 つ増えます。ただし、ブロックが無効になっている場合、同じ要求を処理すると、違反とログの統計カウンタが 3 ずつ増加します。これは、フィールド形式違反ごとに個別のログメッセージが生成されるためです。

コマンドラインを使用してフィールドフォーマットの統計を表示するには

コマンドプロンプトで入力します。

```
sh appfw stats
```

特定のプロファイルの統計情報を表示するには、次のコマンドを使用します。

```
stat appfw profile <profile name>
```

GUI を使用してフィールド形式の統計を表示するには

1. [システム] > [セキュリティ] > [アプリケーションファイアウォール] に移動します。
2. 右側のペインで、[統計リンク] にアクセスします。
3. スクロールバーを使用して、フィールド形式の違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

展開のヒント

- フィールド形式のアクションログ、学習および統計を有効にします。
- アプリケーションへのトラフィックの代表的なサンプルを実行した後、学習した推奨事項を確認します。
- 学習したルールの大部分でフィールドタイプが推奨される場合は、そのフィールドタイプをデフォルトのフィールドタイプとして設定します。最小長と最大長については、これらの規則で示されている最も広い範囲を使用してください。
- 異なるフィールドタイプまたは異なる最小/最大長が適している他のフィールドのルールを展開します。
- ブロッキングを有効にし、学習を無効にします。
- 統計とログを監視します。かなりの数の違反がまだトリガーされている場合は、ログメッセージを確認して、違反がブロックされているはずの悪意のある要求を表していることを確認することをお勧めします。有効なリクエストに違反のフラグが付けられている場合は、設定済みのフィールド形式ルールを編集してさらに緩和するか、再度学習を有効にして、新しいデータポイントに基づく推奨事項を取得できます。

> **注:** 新しい推奨ラーニングを取得することで、設定を微調整できます。

ハイライト

フィールド形式のセキュリティチェックについては、次の点に注意してください。

- 保護：最適なフィールド形式ルールを設定することで、多くの攻撃から保護できます。たとえば、フィールドに整数のみを含めることができるように指定した場合、ハッカーはこのフィールドを使用して SQL インジェクションまたはクロスサイトスクリプティング攻撃を開始できません。このような攻撃を開始するために必要な入力は、構成されたフィールド形式の要件を満たさないためです。。
- パフォーマンス：フィールド形式ルールで、入力に許容される最小長と最大長を制限できます。これにより、悪意のあるユーザーがサーバーに処理オーバーヘッドを追加しようとする際に過度に大きな入力文字列を入力するのを防ぎ、さらに悪いことに、スタックオーバーフローのためにサーバーがコアをダンプするのを防ぐことができます。入力サイズを制限することで、正当な要求の処理に要する時間を短縮できます。
- フィールド形式の設定：フィールド形式の保護を使用するには、いずれかのアクション（ブロック、ログ、統計、学習）を有効にする必要があります。また、フィールド書式ルールを指定して、フォームフィールドで許可される入力を識別することもできます。
- 文字マップの選択とフィールドタイプ：文字マップとフィールドタイプの両方で正規表現を使用します。ただし、文字コード表では、使用できる文字のリストを絞り込むことで、より具体的な式が得られます。たとえば、janedoe@citrix.com などの入力の場合、学習エンジンは Field Type nohtml ではなく、文字マップを推奨する場合があります [。@-za-Z] は、許可された非アルファ文字のセットを狭めるため、より具体的な場合があります。文字コード表オプションでは、アルファ文字の他に、ピリオド (.) とアットマーク (@) の 2 つの非英文字のみを使用できます。
- 継続的な学習：Web App Firewall は、すべての受信データ（違反および許可された入力）を監視して考慮し、ルールを推奨するための学習テーブルを構築します。新しい受信データが到着すると、ルールが改訂され、更新されます。バインドされたフィールド書式ルールがすでにある場合でも、新しいフィールド書式ルールがフィールドに対して提案されます。設定されたフィールド形式が制限的すぎて、有効な要求をブロックしている場合は、よりリラックスしたフィールド形式を展開できます。同様に、現在のフィールド形式が一般的すぎる場合は、より制限の厳しいフィールド形式を展開することで、セキュリティをさらに強化し、強化することができます。
- ルールの上書き：フィールドと URL の組み合わせに対してルールがすでに展開されている場合、GUI を使用してフィールド形式を更新できます。既存のルールを置き換えるかどうかを確認するダイアログボックスが表示されます。コマンドラインインターフェイスを使用している場合は、以前のバインドを明示的にバインド解除してから、新しいルールをバインドする必要があります。
- 複数一致：複数のフィールド形式が特定のフィールド名とそのアクション URL と一致する場合、Web App Firewall は、適用対象の 1 つを任意に選択します。
- バッファ境界：フィールド値が複数のストリーミングバッファにまたがり、フィールド値のこれら 2 つの部分の形式が異なる場合、「any」に対応するフィールド形式が学習データベースに送信されます。
- フィールド形式とフィールドコンシステンシチェック：フィールドフォーマットチェックとフィールドコンシステンシチェックはいずれも、フォームベースの保護チェックです。[フィールド形式] チェックでは、フォームフィールドの一貫性チェックとは異なる種類の保護が提供されます。フォームフィールドの一貫性チェックは、ユーザーから返される Web フォームの構造が損なわれていないこと、HTML で設定されたデータ形式の制限が尊重されていること、および非表示フィールドのデータが変更されていないことを確認します。これは、Web フォーム自体から派生したもの以外の Web フォームに関する特定の知識なしに行うことができます。[Field Formats] チェックでは、各フォームフィールドのデータが、手動で設定した特定の書式制限と一致し

ているか、学習機能が生成され、承認したことを確認します。言い換えれば、フォームフィールドの一貫性チェックは一般的な Web フォームのセキュリティを強化し、フィールドフォーマットチェックは Web フォームの許可された入力に特定のルールを適用します。

フォームフィールドの一貫性チェック

October 7, 2021

フォームフィールドの整合性チェックは、Web サイトのユーザーから返された Web フォームを調べ、Web フォームがクライアントによって不適切に変更されていないことを確認します。このチェックは、データの有無にかかわらず、Web フォームを含む HTML リクエストにのみ適用されます。XML リクエストには適用されません。

フォームフィールドの整合性チェックは、クライアントがフォームに入力して送信するときに、Web サイト上の Web フォームの構造に不正な変更を加えることを防ぎます。また、ユーザーが送信するデータが長さとタイプに関する HTML の制限を満たし、非表示フィールドのデータが変更されないようにします。これにより、攻撃者が Web フォームを改ざんし、変更されたフォームを使用して Web サイトへの不正アクセスを取得したり、安全でないスクリプトを使用する連絡フォームの出力をリダイレクトして一方的な大量の電子メールを送信したり、Web サーバーソフトウェアの脆弱性を悪用したりすることを防ぎます。Web サーバーまたは基盤となるオペレーティングシステムの制御を取得します。Web フォームは、多くの Web サイトの弱いリンクであり、さまざまな攻撃を引き付けます。

フォームフィールドの一貫性チェックでは、次のすべてが検証されます。

- フィールドがユーザーに送信された場合、チェックによってそのフィールドがユーザーによって返されたかどうかが確認されます。
- このチェックでは、HTML フィールドの長さと型が強制されます。

  > 注:
  >
  > - フォームフィールドの一貫性チェックでは、データ型と長さに HTML の制限が適用されますが、Web フォーム内のデータの検証は行われません。[フィールド形式] チェックを使用すると、Web フォーム上の特定のフォームフィールドに返されたデータを検証するルールを設定できます。
  > - フォームフィールドの整合性保護により、非表示のフィールドが挿入されます “as_fid” クライアントに送信される応答フォームで。クライアントがフォームを送信すると、同じ非表示フィールドが ADC によって削除されます。フォームフィールドでチェックサム計算を実行し、バックエンドで同じチェックサムを検証するクライアント側の JavaScript がある場合、アプリケーションが破損する可能性があります。このシナリオでは、アプリケーションファイアウォールのフォームフィールドの整合性の非表示フィールドを緩和することをお勧めします “as_fid” クライアント側の JavaScript チェックサム計算から。

- Web サーバーがユーザーにフィールドを送信しない場合、チェックではユーザーがそのフィールドを追加してデータを返すことはできません。

- フィールドが読み取り専用フィールドまたは非表示フィールドの場合、データが変更されていないことを確認します。
- フィールドがリストボックスまたはラジオボタンフィールドの場合、このチェックは、応答内のデータがそのフィールドの値の 1 つに対応しているかどうかを検証します。

ユーザーから返された Web フォームが 1 つ以上のフォームフィールドの一貫性チェックに違反し、その Web フォームがフォームフィールドの一貫性チェックに違反することを許可するように Web アプリケーションファイアウォールを設定していない場合、リクエストはブロックされます。

ウィザードまたは GUI を使用する場合は、[フォームフィールドの一貫性チェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、[学習]、および [統計] の各アクションを有効または無効にできます。

また、[全般] タブでセッションレスフィールドの整合性も構成します。セッションレスフィールドの一貫性が有効になっている場合、Web App Firewall は Web フォーム構造のみをチェックし、セッション情報の維持に依存するフォームフィールドの一貫性チェックの部分を分配します。これにより、多くのフォームを使用する Web サイトのセキュリティペナルティをほとんど伴わずに、フォームフィールドの整合性チェックを高速化できます。すべての Web フォームでセッションレス項目の整合性を使用するには、[オン] を選択します。HTTP POST メソッドで送信されたフォームにのみ使用するには、[postOnly] を選択します。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して、フォームフィールドの一貫性チェックを設定できます。

- `set appfw profile <name> -fieldConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`

フォームフィールドの一貫性チェックの緩和を指定するには、GUI を使用する必要があります。フォームフィールドの一貫性チェックを修正ダイアログボックスの「チェック」タブで、「追加」をクリックして「フォームフィールドの一貫性チェックリラクゼーションを追加」ダイアログボックスを開くか、既存のリラクゼーションを選択して「開く」をクリックして、フォームフィールドの一貫性チェックリラクゼーションの変更ダイアログボックスを開きます。どちらのダイアログボックスでも、「GUI を使用した手動設定」で説明されているように、緩和の設定に同じオプションが用意されています。

フォームフィールドの一貫性チェック緩和の例を次に示します。

フォームフィールド名:

- UserType という名前のフォームフィールドを選択します。

```
1 ^UserType$
2 <!--NeedCopy-->
```

- 名前が UserType_ で始まり、文字または数字で始まり、1 ～21 の文字、数字、またはアポストロフィまたはハイフン記号で構成されるストリングが続くフォームフィールドを選択します。


```
^UserType_[0-9A-Za-z][0-9A-Za-z'-]{
0,20 }
$
<!--NeedCopy-->
```

- 名前が Turkish-UserType_ で始まり、それ以外の場合は前の式と同じであるフォームフィールドを選択します。ただし、全体としてトルコ語の特殊文字を含めることができます。

```
^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa
   -f])+$
<!--NeedCopy-->
```

> 注:
>
> サポートされている特殊文字の完全な説明と、それらを適切にエンコードする方法については、PCRE 文字エンコーディング形式を参照してください 。

- 文字または数字で始まり、文字または数字のみの組み合わせで構成され、文字列内の任意の場所に文字列 Num を含むフォームフィールド名を選択します。

```
^[0-9A-Za-z]\*Num[0-9A-Za-z]\*$
<!--NeedCopy-->
```

フォームフィールドアクション **URL**:

- http://www.example.com/search.pl?で始まり、新しいクエリを除き、クエリの後に任意の文字列を含む URL を選択します。

```
^http://www[.]example[.]com/search[.]pl?[^?]*$
<!--NeedCopy-->
```

- http://www.example-español.comで始まる URL を選択し、大文字と小文字、数字、非 ASCII 特殊文字、および選択された記号で構成されるパスとファイル名を含めます。ñ 文字およびその他の特殊文字は、UTF-8 文字セットの各特殊文字に割り当てられた 16 進コードを含むエンコードされた UTF-8 文字列として表されます。

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\x[0-9A-
   Fa-f][0-9A-Fa-f])
```



```
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- /search.cgi という文字列を含むすべての URL を選択してください。:

```
^[^?<>]\*/search[.]cgi?[^?<>]\*$
<!--NeedCopy-->
```

> 注意:
>
> 正規表現は強力です。特に、PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク（.*）メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、意図しない Web コンテンツへのアクセスをブロックしたり、Cookie の一貫性チェックでそうしない攻撃を許可したりするなど、望ましくない結果が得られる可能性があります。ブロックされています。

CSRF フォームのタグ付けチェック

October 7, 2021

クロスサイトリクエストフォージェリ (CSRF) フォームのタグ付けチェックでは、保護された Web サイトからユーザーに送信された各 Web フォームに、一意で予測できない FormID のタグが付けられます。その後、ユーザーが返した Web フォームを調べて、提供された FormID が正しいことを確認します。このチェックは、クロスサイトリクエストフォージェリ攻撃から保護します。このチェックは、データの有無にかかわらず、Web フォームを含む HTML リクエストにのみ適用されます。XML リクエストには適用されません。

CSRF フォームのタグ付けチェックは、攻撃者が独自の Web フォームを使用して、保護された Web サイトにデータを含む大量のフォーム応答を送信することを防ぎます。このチェックでは、Web フォームを詳細に分析する他のセキュリティー検査と比較して、CPU 処理能力が比較的少なくて済みます。したがって、保護された Web サイトまたは Web App Firewall 自体のパフォーマンスを大幅に低下させることなく、大量の攻撃を処理できます。

CSRF フォームのタグ付けチェックを有効にする前に、次の点に注意する必要があります。

- フォームのタグ付けを有効にする必要があります。CSRF チェックはフォームのタグ付けに依存し、それがなければ動作しません。
- そのプロファイルによって保護されているフォームを含むすべての Web ページに対して、Citrix ADC 統合キャッシュ機能を無効にする必要があります。統合キャッシュ機能と CSRF フォームのタグ付けには互換性がありません。

- リファラーチェックを有効にすることを検討する必要があります。リファラーチェックは開始 URL チェックの一部ですが、開始 URL 違反ではなく、クロスサイトリクエストフォージェリを防ぐことができます。リファラーチェックは、CSRF フォームのタグ付けチェックよりも CPU の負荷を軽減します。リクエストが Referer チェックに違反した場合、リクエストはすぐにブロックされるため、CSRF フォームタグチェックは呼び出されません。
- CSRF フォームのタグ付けチェックは、フォームオリジン URL とフォームアクション URL で異なるドメインを使用する Web フォームでは機能しません。たとえば、CSRF フォームタグ付けでは、example.com と example.org が異なるドメインであるため、`http://www.example.com`のフォーム生成元 URL と`http://www.example.org/form`.pl のフォームアクション URL を持つウェブフォームを保護できません。

ウィザードまたは GUI を使用する場合は、[CSRF フォームのタグ付けチェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、[ラーニング]、および [統計] アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して CSRF フォームのタグ付けチェックを設定できます。

- `set appfw profile <name> -CSRFtagAction [**block**] [**log**] [**learn**] [**stats**] [**none**]`

CSRF フォームのタグ付けチェックの緩和を指定するには、GUI を使用する必要があります。CSRF フォームのタグ付けチェックダイアログボックスの「チェック」タブで、「追加」をクリックして「CSRF フォームのタグ付けの追加」リラクゼーションチェックダイアログボックスを開くか、既存のリラクゼーションを選択して「開く」をクリックして「CSRF フォームのタグ付けチェックリラクゼーションを変更」ダイアログボックスを開きます。どちらのダイアログボックスも、緩和の設定に同じオプションを提供します。

Citrix Web App Firewall セッション制限を 0 以下の値に設定すると、アラートが生成されます。これは、Web App Firewall セッションが正しく機能する必要がある高度な保護チェック機能に影響する設定です。

CSRF フォームタグ付けチェック緩和の例を以下に示します。

注: 次の式は、フォームオリジン URL ロールとフォームアクション URL ロールの両方で使用できる URL 式です。

- 新しいクエリを除き、`http://www.example.com/search.pl?`で始まるクエリの後に任意の文字列を含む URL を選択します。

```
^http://www[.]example[.]com/search[.]pl?[^?]*$
<!--NeedCopy-->
```

- `http://www.example-español.com`で始まる URL を選択し、大文字と小文字、数字、非 ASCII 特殊文字、および選択された記号で構成されるパスとファイル名を含めます。ñ 文字およびその他の特殊文字は、UTF-8 文字セットの各特殊文字に割り当てられた 16 進コードを含むエンコードされた UTF-8 文字列として表されます。


```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- /search.cgi という文字列を含むすべての URL を選択してください。：

```
^[^?<>]\*/search[.]cgi?[^?<>]\*$
<!--NeedCopy-->
```

> 重要
>
> 正規表現は強力です。PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク (.*) メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、ブロックする意図がない Web コンテンツへのアクセスをブロックしたり、チェックによってブロックされた攻撃を許可したりするなど、望ましくない結果が生じる可能性があります。
>
> ヒント
>
> [URL の開始] アクションで [リファラーヘッダーを有効にする] が有効になっている場合は、リファラーヘッダー URL が StartURL にも追加されていることを確認します。
>
> 注
>
> Citrix ADC が appfw_session_limit に達し、CSRF チェックが有効になっていると、Web アプリケーションがフリーズします。
>
> Web アプリケーションがフリーズしないようにするには、次のコマンドを使用してセッションタイムアウトを減らし、セッション制限を増やします。
>
> CLI: > set appfw settings –sessiontimeout 300
> From shell: root@ns# nsapimgr_wr.sh -s appfw_session_limit=200000
>
> appfw_session_limit に達した場合に SNMP アラームをログに記録して生成すると、問題のトラブルシューティングとデバッグに役立ちます。

CSRF フォームのタグ付けチェック緩和の管理

October 7, 2021

CSRF フォームのタグ付けセキュリティチェックに例外 (またはリラクゼーション) を構成します。[クロスサイト要求偽造タグ付けチェック緩和の追加] ダイアログボックスまたは [クロスサイト要求偽造タグ付けチェック緩和の変更] ダイアログボックス。

GUI を使用して **CSRF** フォームのタグ付けチェック緩和を設定するには

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. [プロファイル] ウィンドウで、構成するプロファイルを選択し、[開く] をクリックします。
3. [**Web App Firewall** プロファイルの構成] ダイアログボックスで、[セキュリティチェック] タブをクリックします。[セキュリティチェック] タブには、Web App Firewall セキュリティチェックの一覧が表示されます。
4. CSRF 緩和を追加または変更するには、次のいずれかの操作を行います。
   - 新しい緩和を追加するには、[追加] をクリックします。
   - 既存の緩和を変更するには、変更する緩和を選択し、[開く] をクリックします。

   [クロスサイト要求偽造タグの追加**] [**緩和チェック**]** または **[**クロスサイト要求偽造タグの変更] ダイアログボックスが表示されます。タイトルを除き、これらのダイアログボックスは同じです。
5. 次の説明に従って、ダイアログボックスに入力します。
   - [**Enabled]** チェックボックス: この緩和または規則をアクティブに使用する場合に選択します。オフにすると、緩和または規則が非アクティブになります。
   - [**Form Origin URL**]: テキスト領域で、フォームをホストする URL を定義する PCRE-形式の正規表現を入力します。
   - [**Form Action URL**]: テキスト領域に、フォームに入力されたデータの配信先となる URL を定義する PCRE 形式の正規表現を入力します。
   - コメント: テキスト領域にコメントを入力します。オプションです。

   > 注:
   >
   > 正規表現を必要とする要素については、正規表現を入力するか、[正規表現 トークン] メニューを使用して正規表現の要素とシンボルをテキストボックスに直接挿入するか、[正規表現 エディタ] をクリックして [正規表現の追加] ダイアログボックスを開きます。を作成し、それを使用して式を構築します。
6. [**OK**] をクリックします。「クロスサイトリクエスト偽造タグの追加」「リラクゼーションの確認」または「クロスサイトリクエスト偽造タグの 修正」ダイアログが閉じ、「クロスサイトリクエスト偽造タグの確認」ダイアログボックスに戻ります。
7. 緩和または規則を削除するには、緩和または規則を選択し、[削除] をクリックします。
8. 緩和または規則を有効にするには、緩和または規則を選択し、[有効にする] をクリックします。
9. 緩和または規則を無効にするには、緩和または規則を選択し、[無効] をクリックします。

10. 統合された対話型グラフィック表示で、既存のすべての緩和の設定と関係を構成するには、[ビジュアライザー] をクリックし、表示ツールを使用します。
11. CSRF チェックの学習ルールを確認して設定するには、[学習] をクリックし、学習機能を設定して使用するにはの手順を実行します。
12. [**OK**] をクリックします。

URL 保護チェック

October 7, 2021

URL 保護チェックは、要求 URL を調べて、攻撃者が複数の URL に積極的にアクセスしようとしたり（強制的な参照）、URL を使用して Web サーバーソフトウェアまたは Web サイトスクリプトの既知のセキュリティ脆弱性をトリガーしたりするのを防ぎます。

URL チェックの開始

October 7, 2021

開始 URL チェックでは、受信要求の URL が検査され、URL が指定された基準を満たしていない場合は接続試行がブロックされます。条件を満たすには、[URL クロージャの強制] パラメータが有効でない限り、URL は開始 URL リストのエントリと一致する必要があります。このパラメーターを有効にすると、Web サイト上のリンクをクリックしたユーザーは、そのリンクのターゲットに接続されます。

開始 URL チェックの主な目的は、ブックマークや外部リンクを介して Web サイト上のランダムな URL に繰り返しアクセスしようとしたり、URL を手動で入力してその部分に到達するために必要なページをスキップしたりして、ページにジャンプするのを防ぐことです。ウェブサイトの。強制ブラウジングを使用すると、バッファオーバーフローを引き起こしたり、ユーザーが直接アクセスすることを意図していないコンテンツを見つけたり、Web サーバーの安全な領域へのバックドアを見つけたりすることができます。Web App Firewall は、開始 URL として設定されている URL のみへのアクセスを許可することで、Web サイトの特定のトラバーサルまたはロジックパスを強制します。

ウィザードまたは GUI を使用する場合は、[開始 URL チェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、[統計]、[ラーニングアクション]、および次のパラメータを有効または無効にできます。

- **URL** の閉鎖を強制します。Web サイトの他のページのハイパーリンクをクリックして、ユーザーが Web サイトの任意の Web ページにアクセスできるようにします。ユーザーは、ハイパーリンクをクリックして、ホームページまたは指定された開始ページからアクセスできる Web サイト上の任意のページに移動できます。注:URL クロージャ機能を使用すると、任意のクエリ文字列を HTTP GET メソッドを使用して送信された Web フォームのアクション URL に追加および送信できます。保護された Web サイトがフォームを使用して

SQL データベースにアクセスする場合は、SQL インジェクションチェックが有効になっていて、適切に構成されていることを確認してください。

- セッションレス **URL** の閉鎖。クライアントの観点からは、このタイプの URL クロージャは、標準のセッション対応 URL クロージャとまったく同じように機能しますが、Cookie の代わりに URL に埋め込まれたトークンを使用してユーザーのアクティビティを追跡します。セッションレス URL 閉鎖が有効になっている場合、Web App Firewall は、URL 閉鎖内にあるすべての URL に「as_url_id」タグを追加します。
  注：セッションレス（セッションレス URL 閉鎖）を有効にする場合は、通常の URL 閉鎖（URL 閉鎖の強制）を有効にする必要があります。そうしないと、セッションレス URL 閉鎖が機能しません。
- リファラーヘッダーを検証します。別の Web サイトではなく保護された Web サイトからの Web フォームデータを含むリクエストの Referer ヘッダーを確認します。このアクションは、外部の攻撃者ではなく、Web サイトが Web フォームのソースであることを確認します。これにより、ヘッダーチェックよりも CPU を大量に消費するフォームのタグ付けを必要とせずに、クロスサイト要求フォージェリ (CSRF) から保護されます。Web App Firewall は、ドロップダウンリストで選択するオプションに応じて、次の 4 つの方法のいずれかで HTTP Referer ヘッダーを処理できます。
  - **Off**：Referer ヘッダーを検証しません。
  - **If-Present**—Referer ヘッダーが存在する場合は、Referer ヘッダーを検証します。無効な Referer ヘッダーが見つかった場合、リクエストはリファラーヘッダー違反を生成します。Referer ヘッダーが存在しない場合、リクエストはリファラーヘッダー違反を生成しません。このオプションを有効にすると、Web App Firewall は Referer ヘッダーを含むリクエストに対して Referer ヘッダーの検証を実行できますが、ブラウザーが Referer ヘッダーを設定していないユーザーや、そのヘッダーを削除する Web プロキシやフィルターを使用するユーザーからのリクエストをブロックすることはできません。
  - 常に開始 **URL** を除く— 常にリファラーヘッダーを検証します。Referer ヘッダーがなく、要求された URL が startURL 緩和ルールによって免除されていない場合、リクエストはリファラーヘッダー違反を生成します。Referer ヘッダーが存在しても無効な場合、リクエストはリファラーヘッダー違反を生成します。
  -「最初の要求を除く常に必ず」—常にリファラーヘッダーを検証します。リファラーヘッダーがない場合は、最初にアクセスされた URL のみが許可されます。他のすべての URL は、有効なリファラーヘッダーなしでブロックされます。Referer ヘッダーが存在しても無効な場合、リクエストはリファラーヘッダー違反を生成します。

[開始 URL チェックの変更]** ダイアログボックスでは、[開始 URL チェックの終了 URL を免除 **] の設定は構成されていませんが、プロファイルの [設定] タブで構成されています。この設定を有効にすると、Web App Firewall が URL クロージャの条件を満たす URL に対してフォームベースのチェック（クロスサイトスクリプティングや SQL インジェクションインスペクションなど）を実行しないように指示します。

> 注
>
> リファラーヘッダーチェックと開始 URL セキュリティチェックは同じアクション設定を共有しますが、開始 URL チェックに違反することなくリファラーヘッダーチェックに違反する可能性があります。差異は、開始 URL チェック違反とは別にリファラーヘッダーチェック違反をログに記録するログに表示されます。

リファラーヘッダーの設定（オフ、存在する場合、常に例外開始 URL、および AlwaysExceptFirstRequest）は、

最も制限の低い順に配置され、次のように動作します。

OFF:

- 参照元ヘッダーはチェックされていません。

If-Present:

- リクエストにはリファラーヘッダーがありません-> リクエストは許可されています。
- リクエストにはリファラーヘッダーがあり、リファラー URL は URL クロージャにあります-> リクエストは許可されています。
- リクエストにはリファラーヘッダーがあり、リファラー URL が URL クロージャに ありません -> リクエストがブロックされています。

AlwaysExceptStartURLs:

- リクエストにはリファラーヘッダーがなく、リクエスト URL が開始 URL です-> リクエストが許可されています。
- リクエストにはリファラーヘッダーがなく、リクエスト URL が開始 URL ではありません-> リクエストはブロックされています。
- リクエストにはリファラーヘッダーがあり、リファラー URL は URL クロージャにあります-> リクエストは許可されています。
- リクエストにはリファラーヘッダーがあり、リファラー URL が URL クロージャに ありません -> リクエストがブロックされています。

AlwaysExceptFirstRequest:

- リクエストにはリファラーヘッダーがなく、セッションの最初のリクエスト URL です-> リクエストが許可されています。
- リクエストにはリファラーヘッダーがなく、セッションの最初のリクエスト URL ではありません -> リクエストがブロックされています。
- リクエストにはリファラーヘッダーがあり、セッションの最初のリクエスト URL であるか、URL クロージャにあります-> リクエストが許可されています。
- リクエストにはリファラーヘッダーがあり、セッションの最初のリクエスト URL でもなく、URL クロージャーにもありません-> リクエストはブロックされています。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して、開始 URL チェックを設定できます。

- `set appfw profile <name> -startURLAction [block] [learn] [log] [stats] [none]`
- `set appfw profile <name> -startURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -sessionlessURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -exemptClosureURLsFromSecurityChecks ([ON] | [OFF)`
- `set appfw profile <name> -RefererHeaderCheck ([OFF] | [if-present] | [AlwaysExceptStartURLs] | [AlwaysExceptFirstRequest])`

開始 URL チェックのリラックスを指定するには、GUI を使用する必要があります。開始 URL チェックを修正ダイアログボックスの「チェック」タブで、「追加」をクリックして「開始 URL チェック緩和を追加」ダイアログボックスを開くか、既存のリラクゼーションを選択して「開く」をクリックして「開始 URL チェック緩和を修正」ダイアログボックスを開きます。どちらのダイアログボックスも、緩和の設定に同じオプションを提供します。

開始 URL チェック緩和の例を次に示します。

- ユーザーに www.example.com のホームページへのアクセスを許可します。

```
^http://www[.]example[.]com$
<!--NeedCopy-->
```

- ユーザーがすべての静的 HTML (.htm および.html)、サーバー解析された HTML (.htp および.shtml)、PHP (.php)、Microsoft の ASP (.asp) 形式のウェブページにアクセスすることを許可します。

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- www.example-español.com で非 ASCII 文字を含むパス名またはファイル名で Web ページにアクセスすることを許可します。

```
^http://www[.]example-espaxC3xB1ol[.]com/(([0-9A-Za-z]|x[0-9A-Fa-
   f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*
([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f
   ][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

注：上記の式では、各文字クラスは文字列
x[0-9a-FA-F でグループ化されています][0-9A-Fa-f]。これは、適切に構築されたすべての文字エンコーディング文字列と一致しますが、UTF-8 文字エンコーディング文字列に関連付けられていない浮遊バックスラッシュ文字は使用できません。二重バックスラッシュ () はエスケープされたバックスラッシュで、Web App Firewall がリテラルバックスラッシュとして解釈するように指示します。バックスラッシュを 1 つだけ含めた場合、Web App Firewall では代わりに次の左角括弧 ([) が文字クラスの開始ではなくリテラル文字として解釈され、式が壊れます。

- www.example.com ですべての GIF (.png)、JPEG (.jpg、.jpeg)、および PNG (.png) 形式のグラフィックへのアクセスを許可します。


```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](gif|jpe?g|png)$
<!--NeedCopy-->
```

- CGI (.cgi) および PERL (.pl) スクリプトへのアクセスを許可しますが、CGI-BIN ディレクトリでのみ許可します。

```
^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_
   .-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- ユーザーが docsarchive ディレクトリにある Microsoft Office およびその他のドキュメントファイルにアクセスすることを許可します。

```
^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_
   -.]*[.](doc|xls|pdf|ppt)$
<!--NeedCopy-->
```

> 注
>
> デフォルトでは、すべての Web App Firewall URL は正規表現と見なされます。

注意: 正規表現は強力です。特に、PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。例外として追加する URL を正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク (
.*) メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、意図しないウェブコンテンツへのアクセスをブロックしたり、開始 URL チェックでブロックされた攻撃を許可したりするなど、望ましくない結果になる可能性があります。

> ヒント
>
> URL 命名スキームの SQL キーワードの許可リストに *-and-* を追加できます。たとえば https://FQDN/bread-and-butterの例。

URL チェックを拒否

October 7, 2021

[URL の拒否] チェックは、ハッカーや悪意のあるコードによってよくアクセスされる URL への接続を調べてブロックします。このチェックには、ハッカーや悪意のあるコードの共通ターゲットであり、正当なリクエストに現れる

ことはほとんどない URL のリストが含まれます。リストに URL または URL パターンを追加することもできます。URL の拒否チェックは、Web サーバーソフトウェアまたは多くの Web サイトに存在することが知られているさまざまなセキュリティの弱点に対する攻撃を防ぎます。

[URL の拒否] チェックは、開始 URL チェックよりも優先されるため、通常、開始 URL の緩和によって要求が続行される場合でも、悪意のある接続の試行を拒否します。

[拒否 URL チェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、[統計] の各アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して、拒否 URL チェックを設定できます。

- `set appfw profile <name> -denyURLAction [**block**] [**log**] [**stats**] [**none**]`

独自の拒否 URL を作成して設定するには、GUI を使用する必要があります。[拒否 URL チェックの変更] ダイアログボックスの [チェック] タブで、[追加] をクリックして [拒否 URL の追加] ダイアログボックスを開くか、既存のユーザー定義の拒否 URL を選択して [開く] をクリックして [拒否 URL の変更] ダイアログボックスを開きます。どちらのダイアログボックスも、拒否 URL の作成と構成に同じオプションを提供します。

URL の拒否 (Deny URL) 式の例を次に示します。

- images.example.com のイメージサーバに直接アクセスすることをユーザに許可しないでください。

```
^http://images[.]example[.]com$
<!--NeedCopy-->
```

- ユーザーが CGI（.cgi）または PERL（.pl）スクリプトに直接アクセスすることを許可しないでください。

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- 非 ASCII 文字をサポートするように変更された、同じ拒否 URL は次のとおりです。

```
^http://www[.]example[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f
   ])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|x[0-9A-
   Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$
<!--NeedCopy-->
```


> 注意:
>
> 正規表現は強力です。特に、PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。ブロックする URL やパターンを正確に定義していることを確認し、それ以外は何も定義しないでください。ワイルドカード、特にドットとアスタリスク (.*) メタキャラクタとワイルドカードの組み合わせを不注意に使用すると、ブロックする意図がない Web コンテンツへのアクセスをブロックするなど、望ましくない結果が生じる可能性があります。

XML 保護チェック

October 7, 2021

XML 保護チェックでは、すべての種類の XML ベースの攻撃に対する要求を調べます。

> 注意:
>
> XML セキュリティチェックは、text/xml の HTTP コンテンツタイプヘッダーで送信されるコンテンツにのみ適用されます。content-type ヘッダーが見つからないか、別の値に設定されている場合、すべての XML セキュリティチェックはバイパスされます。XML または Web2.0 Web アプリケーションを保護する場合は、それらのアプリケーションをホストする各 Web サーバーの Web マスターが、適切な HTTP コンテンツタイプヘッダーが送信されるようにする必要があります。

XML 形式のチェック

October 7, 2021

XML 形式チェックでは、受信要求の XML 形式が検査され、適切な形式の XML ドキュメントに対する XML 仕様の基準を満たさない要求がブロックされます。これらの基準のいくつかは、次のとおりです。

- XML ドキュメントには、Unicode 仕様に一致する適切にエンコードされた Unicode 文字のみが含まれている必要があります。
- XML マークアップで使用される場合を除き、<、>、& などの特殊な XML 構文文字を文書に含めることはできません。
- すべての開始タグ、終了タグ、および空要素タグは、正しくネストされなければならず、欠落したり重複したりしません。
- XML エレメントタグでは、大文字と小文字が区別されます。すべての開始タグと終了タグは正確に一致する必要があります。
- 単一のルート要素は、XML 文書内の他のすべての要素を含む必要があります。

整形式の XML の条件を満たさないドキュメントが、XML ドキュメントの定義を満たしていません。厳密に言えば、XML ではありません。ただし、すべての XML アプリケーションおよび Web サービスが XML 整形式標準を適用する

わけではなく、形式が不適切な XML や無効な XML を正しく処理するわけではありません。形式が不適切な XML ドキュメントを適切に処理すると、セキュリティ侵害が発生する可能性があります。XML 形式チェックの目的は、悪意のあるユーザーが形式が不適切な XML 要求を使用して XML アプリケーションまたは Web サービスのセキュリティを侵害することを防ぐことです。

ウィザードまたは GUI を使用する場合は、[XML 形式のチェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、および [統計] の各アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML フォーマットチェックを設定できます。

- `set appfw profile <name> -xmlFormatAction [**block**] [**log**] [**stats**] [**none**]`

XML 形式のチェックに例外を設定することはできません。有効化または無効化のみが可能です。

XML サービス拒否チェック

October 7, 2021

XML サービス拒否（XML DoS または XDoS）チェックは、着信 XML 要求を調べて、サービス拒否（DoS）攻撃の特性と一致するかどうかを判断します。一致する場合は、それらの要求をブロックします。XML DoS チェックの目的は、攻撃者が XML 要求を使用して Web サーバーまたは Web サイトに対してサービス拒否攻撃を開始するのを防ぐことです。

ウィザードまたは GUI を使用する場合、[XML サービス拒否チェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、[統計]、および [学習] アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML サービス拒否チェックを設定できます。

- `set appfw profile <name> -xmlDoSAction [**block**] [**log**] [**learn**] [**stats**] [**none**]`

個々の XML サービス拒否ルールを設定するには、GUI を使用する必要があります。**[XML** サービス拒否チェックの変更] ダイアログボックスの [チェック] タブで、ルールを選択し、[開く] をクリックして、そのルールの **[XML** サービス拒否の変更] ダイアログボックスを開きます。個々のダイアログボックスはルールごとに異なりますが、単純です。ルールを有効または無効にできるものもあれば、テキストボックスに新しい値を入力して数値を変更できるものもあります。

> 注:
>
> サービス拒否攻撃に対する LearningEngine の予想される動作は、構成されたアクションに基づいています。アクションが「ブロック」に設定されている場合、エンジンは構成されたバインド値を学習します +1 違反があると、XML 解析は停止します。設定されたアクションが「ブロック」として設定されていない場合、エンジン

> は実際の着信違反長の値を学習します。

個々の XML サービス拒否規則は次のとおりです。

- 最大要素の深さ。個々の要素のネストされたレベルの最大数を 256 に制限します。このルールが有効で、Web App Firewall が、最大許容レベル数を超える要素を持つ XML 要求を検出すると、要求をブロックします。レベルの最大数を 1 ～65,535 の値に変更できます。
- 要素名の最大長。各要素名の最大長を 128 文字に制限します。これには、展開された名前空間内の名前が含まれます。この名前には、XML パスと要素名が含まれます。

```
{
http://prefix.example.com/path/ }
target_page.xml
<!--NeedCopy-->
```

ユーザーは、名前の最大長を 1 文字から 65,535 までの任意の値に変更できます。

- 最大 # 要素。XML ドキュメントごとに 1 つのタイプのエレメントの最大数を 65,535 に制限します。要素の最大数を、1 ～65,535 の間の任意の値に変更できます。
- 最大 # エレメントチルドレン。各要素に許可される子の最大数 (他の要素、文字情報、コメントを含む) を 65,535 に制限します。エレメントの子の最大数は、1 ～65,535 の間の任意の値に変更できます。
- 最大 # 属性。各要素に許可される属性の最大数を 256 に制限します。属性の最大数は、1 ～256 の間の任意の値に変更できます。
- 属性名の最大長。各属性名の最大長を 128 文字に制限します。属性名の最大長は、1 ～2,048 の間の任意の値に変更できます。
- 最大属性値の長さ。各属性値の最大長を 2048 文字に制限します。属性名の最大長は、1 ～2,048 の間の任意の値に変更できます。
- 最大文字データ長。各要素の最大文字データ長を 65,535 に制限します。長さは、1 ～65,535 の間の任意の値に変更できます。
- 最大ファイルサイズ。各ファイルのサイズを 20 MB に制限します。最大ファイルサイズは任意の値に変更できます。
- 最小ファイルサイズ。各ファイルの長さが 9 バイト以上である必要があります。最小ファイルサイズは、さまざまなバイトを表す任意の正の整数に変更できます。
- 最大 # エンティティの拡張。エンティティ展開の数を指定された数に制限します。デフォルトは 1024 です。
- 最大エンティティ拡張深度。ネストされたエンティティ拡張の最大数を、指定した数以下に制限します。デフォルトは 32 です。
- 最大 # 名前空間。XML ドキュメント内の名前空間宣言の数を、指定した数以下に制限します。デフォルトは 16 です。

- 名前空間 URI の最大長。各名前空間宣言の URL の長さを、指定した文字数以下に制限します。デフォルトは 256 です。
- ブロック処理命令。リクエストに含まれる特別な処理命令をブロックします。このルールには、ユーザーが変更できる値はありません。
- DTD をブロックします。リクエストに含まれる文書型定義 (DTD) をブロックします。このルールには、ユーザーが変更できる値はありません。
- 外部エンティティをブロックします。リクエスト内の外部エンティティへの参照をすべてブロックします。このルールには、ユーザーが変更できる値はありません。
- SOAP 配列チェック。次の SOAP 配列チェックを有効または無効にします。
  - **SOAP** アレイの最大サイズ。接続がブロックされる前の XML 要求内のすべての SOAP 配列の最大合計サイズ。この値は変更できます。デフォルトは 20000000 です。
  - **SOAP** アレイの最大ランク。接続がブロックされる前の XML 要求内の単一の SOAP 配列の最大ランクまたは次元。この値は変更できます。デフォルトは 16 です。

XML クロスサイトスクリプティングチェック

October 7, 2021

XML クロスサイトスクリプティングチェックでは、XML ペイロードで発生する可能性のあるクロスサイトスクリプティング攻撃に対するユーザ要求を検証します。クロスサイトスクリプティング攻撃の可能性が見つかると、リクエストをブロックします。

保護された Web サービスのスクリプトが悪用されて、Web サービスのセキュリティが侵害されるのを防ぐために、XML クロスサイトスクリプティングチェックは、同じオリジンルールに違反するスクリプトをブロックします。このルールでは、スクリプトはサーバー上のコンテンツにアクセスしたり変更したりしてはなりません。それらが配置されているサーバー。同じオリジン規則に違反するスクリプトはクロスサイトスクリプトと呼ばれ、スクリプトを使用して別のサーバー上のコンテンツにアクセスまたは変更する習慣をクロスサイトスクリプティングと呼びます。クロスサイトスクリプティングがセキュリティ上の問題である理由は、クロスサイトスクリプティングを許可する Web サーバーは、その Web サーバー上ではなく、攻撃者が所有し制御しているサーバーなど、異なる Web サーバー上で攻撃される可能性があるためです。

Web App Firewall は、XML クロスサイトスクリプティング保護を実装するためのさまざまなアクションオプションを提供します。ブロック、ログ、および 統計の各アクションを設定するオプションがあります。

Web App Firewall XML クロスサイトスクリプティングチェックは、着信要求のペイロードに対して実行され、攻撃文字列が複数の行に分散している場合でも識別されます。このチェックでは、要素と 属性値でクロスサイトスクリプティング攻撃文字列を探します。緩和を適用して、指定した条件下でセキュリティチェック検査をバイパスできます。ログと統計情報は、必要な緩和を特定するのに役立ちます。

XML ペイロードの CDATA セクションは、スクリプトが CDATA セクションの外で実行できないため、ハッカーにとって魅力的な領域になる可能性があります。CDATA セクションは、文字データとして完全に扱われるコンテンツに使用されます。HTML マークアップタグ区切り文字 **<**、**>**、および **/>** では、パーサはコードを HTML 要素として解釈しません。次の例は、クロスサイトスクリプティング攻撃文字列を含む CDATA セクションを示しています。

```
1     <![CDATA[rn
2     <script language=”Javascript” type=”text/javascript”>alert (”Got
          you”)</script>rn
3     ]]>
4 <!--NeedCopy-->
```

アクションオプション

XML クロスサイトスクリプティングチェックがリクエストでクロスサイトスクリプティング攻撃を検出すると、アクションが適用されます。アプリケーションの XML クロスサイトスクリプティング保護を最適化するには、次のオプションを使用できます。

- ブロック：クロスサイトスクリプティングタグがリクエストで検出された場合、ブロックアクションがトリガーされます。
- [**Log**]：XML クロスサイトスクリプティングチェックで実行されたアクションを示すログメッセージを生成します。ブロックが無効になっている場合、クロスサイトスクリプティング違反が検出された場所（ELEMENT、ATTRIBUTE）ごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとした可能性があります。
- [**Stats**]：違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされている場合は、設定を再度参照して、新しい緩和ルールを設定するか、既存の緩和ルールを変更する必要があるかを確認する必要があります。

リラクゼーションルール

アプリケーションで XML ペイロード内の特定の ELEMENT または ATTRIBUTE のクロスサイトスクリプティングチェックをバイパスする必要がある場合は、緩和ルールを設定できます。XML クロスサイトスクリプティングチェック緩和規則には、次のパラメータがあります。

- **Name**：リテラル文字列または正規表現を使用して、ELEMENT または属性の名前を設定できます。次の式は、文字列 name_ で始まり、その後に大文字または小文字、または数字の文字列 (2 文字以上 15 文字以下) が続くすべての ELEMENTS を免除します。

  ```
  ^name_[0-9A-Za-z]{ 2,15 } $
  ```


> 注
>
> 大文字と小文字は区別されます。重複するエントリは許可されませんが、名前や場所の違いを大文字にすることで、類似したエントリを作成できます。たとえば、次の緩和規則はそれぞれ一意です。
>
> 1. `XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ATTRIBUTE State: ENABLED`
> 2. `XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ELEMENT State: ENABLED`
> 3. `XMLcross-site scripting: abc IsRegex: NOTREGEX Location: ELEMENT State: ENABLED`
> 4. `XMLcross-site scripting: abc IsRegex: NOTREGEX Location: ATTRIBUTE State: ENABLED`

- **Location**: XML ペイロードで、クロスサイトスクリプティングチェック例外の場所を指定できます。オプション ELEMENT はデフォルトで選択されています。ATTRIBUTE に変更できます。
- [**Comment**]: これはオプションのフィールドです。この緩和規則の目的を説明するために、最大 255 文字の文字列を使用できます。

> 警告
>
> 正規表現は強力です。特に、PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。例外として追加する名前を正確に定義していることを確認し、それ以外は何も定義しないでください。正規表現を不注意に使用すると、意図しない Web コンテンツへのアクセスをブロックしたり、XML クロスサイトスクリプティングチェックでブロックされた攻撃を許可するなど、望ましくない結果が生じる可能性があります。

コマンドラインを使用した **XML** クロスサイトスクリプティングチェックの設定

コマンドラインを使用して XML クロスサイトスクリプティングチェックアクションおよびその他のパラメーターを構成するには

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML クロスサイトスクリプティングチェックを設定できます。

```
> set appfw profile <name> -XMLcross-site scriptingAction (([block] [log] [stats])| [none])
```

コマンドラインを使用して XML クロスサイトスクリプティングチェック緩和ルールを構成するには

リラクゼーションルールを追加して、特定の場所でのクロスサイトスクリプティングスクリプト攻撃検査の検査をバイパスできます。次のように、bind または unbind コマンドを使用して、緩和ルールのバインドを追加または削除します。


```
> bind appfw profile <name> -XMLcross-site scripting <string> [isRegex ( REGEX | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )] -comment <string> [-state ( ENABLED | DISABLED )]

> unbind appfw profile <name> -XMLcross-site scripting <String>
```

例:

```
> bind appfw profile test_pr -XMLcross-site scripting ABC
```

上記のコマンドを実行した後、次の緩和ルールが設定されます。ルールが有効になり、名前はリテラル (NOTREGEX) として扱われ、ELEMENT がデフォルトの場所として選択されます。

```
1)      XMLcross-site scripting:  ABC             IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

`> unbind appfw profile test_pr -XMLcross-site scripting abc`

ERROR: No such XMLcross-site scripting check

`> unbind appfw profile test_pr -XMLcross-site scripting ABC`

 Done
<!--NeedCopy-->
```

GUI を使用した XML クロスサイトスクリプティングチェックの設定

GUI では、アプリケーションに関連付けられたプロファイルのペインで XML クロスサイトスクリプティングチェックを設定できます。

GUI を使用して XML クロスサイトスクリプティングチェックを構成または変更するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[セキュリティチェック] をクリックします。

セキュリティチェックテーブルには、すべてのセキュリティチェックに対して現在構成されているアクション設定が表示されます。設定には 2 つのオプションがあります。

a) **XML** クロスサイトスクリプティングチェックのブロック、ログ、および統計アクションを有効または無効にするだけの場合は、テーブルのチェックボックスをオンまたはオフにして [**OK**] をクリックし、[保存して閉じる] をクリックして [セキュリティ] を閉じます。[チェック] ペイン。

b) [**XML** クロスサイトスクリプティング] をダブルクリックするか、行を選択して [アクション設定] をクリックすると、アクションオプションが表示されます。いずれかのアクション設定を変更したら、[**OK**] をクリックして変更を保存し、[セキュリティチェック] テーブルに戻ります。

必要に応じて、他のセキュリティー検査の設定に進むことができます。[**OK**] をクリックして［セキュリティチェック］セクションで行ったすべての変更を保存し、［保存して閉じる］をクリックして［セキュリティチェック］ウィンドウを閉じます。

GUI を使用して XML クロスサイトスクリプティング緩和ルールを構成するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定**]** ペインで、[緩和規則] をクリックします。
3. 「リラクゼーション規則」テーブルで、「**XML** クロスサイトスクリプティング」エントリをダブルクリックするか、エントリを選択して「編集」をクリックします。
4. 「**XML** クロスサイトスクリプティング緩和規則」ダイアログで、緩和規則の「追加」、「編集」、「削除」、「有効化」、または「無効化」の操作を実行します。

ビジュアライザーを使用して XML クロスサイトスクリプティング緩和ルールを管理するには

すべての緩和規則をまとめて表示するには、[緩和規則] テーブルの [**XML** クロスサイトスクリプティング] 行をハイライト表示し、[ビジュアライザー] をクリックします。展開されたリラクゼーションのビジュアライザーには、新しいルールを 追加するか、既存のルールを 編集するオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効または 無効にすることもできます。

GUI を使用してクロスサイトスクリプティングパターンを表示またはカスタマイズするには

GUI を使用して、クロスサイトスクリプティングが許可される属性または許可されるタグのデフォルトのリストを表示またはカスタマイズできます。クロスサイトスクリプティング拒否パターンのデフォルトリストを表示またはカスタマイズすることもできます。

デフォルトリストは、[**Web App Firewall**] > [署名] > [デフォルトの署名] で指定されます。署名オブジェクトをプロファイルにバインドしない場合、デフォルトの署名オブジェクトで指定されたデフォルトのクロスサイトスクリプティングの許可および拒否リストが、クロスサイトスクリプティングのセキュリティチェック処理のためにプロファイルによって使用されます。デフォルトのシグネチャオブジェクトで指定されているタグ、属性、およびパターンは読み取り専用です。編集や修正はできません。これらを変更または変更する場合は、Default Signatures オブジェクトのコピーを作成して、ユーザー定義シグニチャオブジェクトを作成します。新しいユーザ定義シグニチャオブジェクトの Allowed リストまたは Denied リストを変更し、カスタマイズされた許可リストおよび拒否リストを使用するトラフィックを処理するプロファイルでこのシグニチャオブジェクトを使用します。

シグニチャの詳細については、http://support.citrix.com/proddocs/topic/ns-security-10-map/appfw-signatures-con.htmlを参照してください。

デフォルトのクロスサイトスクリプティングパターンを表示するには:

1. [**Web App Firewall**] > [署名] に移動し、[***** デフォルトの署名] を選択して [編集] をクリックします。次に、[管理] をクリックします SQL/cross-site スクリプトパターン。

管理 SQL/cross-site スクリプトパステーブルは、クロスサイトスクリプティングに関連する次の 3 行を示しています:

```
            xss/allowed/attribute

            xss/allowed/tag

            xss/denied/pattern
<!--NeedCopy-->
```

行を選択し、**[**要素の管理] をクリックして、Web App Firewall のクロスサイトスクリプティングチェックで使用される対応するクロスサイトスクリプティング要素（タグ、属性、パターン）を表示します。

クロスサイトスクリプティング要素をカスタマイズするには: ユーザー定義の署名オブジェクトを編集して、許可されたタグ、許可された属性、および拒否されたパターンをカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除したりできます。

1. [**Web App Firewall] > [**** 署名] に移動し、ターゲットユーザー定義の署名をハイライト表示して、**[**** 編集] をクリックします。[管理**]** をクリックします SQL/cross-site 管理を表示するためのスクリプトパターン SQL/cross-site スクリプトパステーブル。
2. ターゲットのクロスサイトスクリプティング行を選択します。

a）[要素の管理**]** をクリックして、対応するクロスサイトスクリプティング要素を追加、編集、または削除します。

b) [削除] をクリックして、選択した行を削除します。

> 警告
>
> デフォルトのクロスサイトスクリプティング要素を削除または変更する場合、またはクロスサイトスクリプティングパスを削除して行全体を削除する場合は、十分に注意してください。シグニチャ、HTML クロスサイトスクリプティングセキュリティチェック、および XML クロスサイトスクリプティングセキュリティチェックは、これらの要素に依存して攻撃を検出し、アプリケーションを保護します。クロスサイトスクリプティング要素をカスタマイズすると、編集中に必要なパターンが削除された場合、アプリケーションがクロスサイトスクリプティング攻撃に対して脆弱になる可能性があります。

XML クロスサイトスクリプティングチェックでのログ機能の使用

ログアクションを有効にすると、XML クロスサイトスクリプティングのセキュリティチェック違反が監査ログに次のように記録されます。APPFW_XML_cross-site スクリプト違反。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポートしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、/var/log/ フォルダ内の ns.logs を末尾に付けて、XML クロスサイトスクリプティング違反に関連するログメッセージにアクセスします。


```
> \*\*Shell\*\*

> \*\*tail -f /var/log/ns.log | grep APPFW_XML_cross-site scripting\*\*
<!--NeedCopy-->
```

<blocked>アクションを示すネイティブログ形式の **XML** クロスサイトスクリプティングセキュリティチェック違反ログメッセージの例

```
Oct  7 01:44:34 <local0.warn> 10.217.31.98 10/07/2015:01:44:34 GMT ns
    0-PPE-1 : default APPFW APPFW_XML_cross-site scripting 1154 0 :
    10.217.253.69 3466-PPE1 - owa_profile http://10.217.31.101/FFC/login
    .html Cross-site script check failed for field script=”Bad tag:
    script” <\*\*blocked\*\*>
<!--NeedCopy-->
```

<not blocked>アクションを示す CEF ログ形式の XML クロスサイトスクリプティングセキュリティチェック違反ログメッセージの例

```
Oct  7 01:46:52 <local0.warn> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
    .0|APPFW|APPFW_XML_cross-site scripting|4|src=10.217.30.17
    geolocation=Unknown spt=33141 method=GET request=http://
    10.217.31.101/FFC/login.html msg=Cross-site script check failed for
    field script=”Bad tag: script” cn1=1607 cn2=3538 cs1=owa_profile cs2
    =PPE0 cs4=ERROR cs5=2015 act=\*\*not blocked\*\*
<!--NeedCopy-->
```

GUI を使用してログメッセージにアクセスするには

Citrix GUI には、ログメッセージを分析するための便利なツール（**Syslog Viewer**）が含まれています。Syslog ビューアには、次の複数のオプションがあります。

- [**Web App Firewall**] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。[**XML** クロスサイトスクリプティング] 行を強調表示し、[ログ] をクリックします。プロファイルの XML クロスサイトスクリプティングチェックから直接ログにアクセスすると、GUI によってログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
- 「**Citrix ADC** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[Syslog メッセージ] リンクをクリックして Syslog ビューアを表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティー検査違反がトリガーされる可能性がある場合のデバッグに役立ちます。

- [**Web App Firewall**] > [ポリシー] > [監査] に移動します。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

XML ベースの Syslog ビューアには、関心のあるログメッセージだけを選択するためのさまざまなフィルタオプションがあります。**XML** クロスサイトスクリプティングチェックのログメッセージを選択するには、[**Module**] のドロップダウンオプションで [**APPFW**] を選択してフィルタリングします。**[Event Type]** リストには、選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、APPFW_XML_cross-site [スクリプト] チェックボックスをオンにして [適用] ボタンをクリックすると、XML クロスサイトスクリプティングのセキュリティチェック違反に関連するログメッセージのみが Syslog ビューアに表示されます。

特定のログメッセージの行にカーソルを置くと、モジュール、イベントタイプ、イベント **ID**、クライアント **IP** などの複数のオプションが表示されます。ログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログメッセージ内の対応する情報を強調表示することができます。

XML クロスサイトスクリプティング違反の統計

stats アクションが有効な場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、XML クロスサイトスクリプティングチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロックアクションが有効になっている場合、3 つの XML クロスサイトスクリプティング違反を含むページに対するリクエストは、最初の違反が検出されるとすぐにページがブロックされるため、stats カウンタを 1 つ増やします。ただし、ブロックが無効になっている場合、同じ要求を処理すると、違反とログの統計カウンタが 3 ずつ増加します。これは、違反ごとに個別のログメッセージが生成されるためです。

コマンドラインを使用して XML クロスサイトスクリプティングチェック統計を表示するには

コマンドプロンプトで入力します。

```
> **sh appfw stats**
```

特定のプロファイルの統計情報を表示するには、次のコマンドを使用します。

```
> **stat appfw profile** <profile name>
```

GUI を使用して XML クロスサイトスクリプティング統計を表示するには

1. [システム] > [セキュリティ] > [**Web App Firewall**] に移動します。
2. 右側のペインで、[統計リンク] にアクセスします。
3. スクロールバーを使用して、XML クロスサイトスクリプティング違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

XML SQL インジェクションチェック

October 7, 2021

XML SQL インジェクションチェックでは、XML SQL インジェクション攻撃の可能性があるユーザー要求を調べます。XML ペイロードに挿入された SQL が見つかると、要求がブロックされます。

XML SQL 攻撃は、ソースコードを Web アプリケーションに挿入して、悪意を持ってデータベース操作を実行するための有効な SQL クエリとして解釈および実行できるようにする可能性があります。たとえば、XML SQL 攻撃を起動して、データベースのコンテンツへの不正アクセスを取得したり、格納されたデータを操作したりできます。XML SQL インジェクション攻撃は一般的であるだけでなく、非常に有害でコストがかかる可能性があります。

データベース・ユーザーの権限を区分化すると、ある程度データベースを保護するのに役立ちます。すべてのデータベースユーザーには、目的のタスクを完了するために必要な特権のみを付与する必要があります。これにより、SQL クエリを実行して他のタスクを実行できなくなります。たとえば、読み取り専用ユーザーにデータテーブルの書き込みや操作を許可してはなりません。Web App Firewall XML SQL Injection チェックは、すべての XML 要求を検査し、セキュリティを損なう可能性のある不正な SQL コードの挿入に対する特別な防御を提供します。Web App Firewall は、ユーザーの XML 要求で不正な SQL コードを検出すると、要求をブロックできます。

Citrix Web App Firewall は、SQL キーワードと特殊文字の存在を検査して、XMLSQL インジェクション攻撃を識別します。キーワードと特殊文字のデフォルトのセットは、XML SQL 攻撃の起動に一般的に使用される既知のキーワードと特殊文字を提供します。Web App Firewall では、一重引用符 (‘)、バックスラッシュ ()、セミコロン (;) の 3 文字を SQL セキュリティチェック処理の特殊文字と見なします。新しいパターンを追加したり、デフォルトのセットを編集して XML SQL チェック検査をカスタマイズしたりできます。

Web App Firewall は、XML SQL インジェクション保護を実装するためのさまざまなアクションオプションを提供します。リクエストを ブロックし、観察された違反に関する詳細を ns.log ファイルにメッセージを 記録し、統計を収集して観測された攻撃の数を追跡することができます。

アクションに加えて、XML SQL インジェクション処理用に構成できるパラメータがいくつかあります。**SQL** ワイルドカード文字をチェックできます。XML の SQL インジェクションタイプを変更し、4 つのオプション (**SQL** キーワード、**SQLSplChar**、**SQLSplCharandKeyword**、**SQLSplCharOR** キーワード) のいずれかを選択して、XML の処理時に SQL キーワードと SQL 特殊文字の評価方法を指定できます。ペイロード。「XML**SQL** コメント処理」パラメータには、XML SQL インジェクションの検出時に検査または免除する必要のあるコメントのタイプを指定するオプションがあります。

リラクゼーションを展開して、誤検出を回避できます。Web App Firewall XML SQL チェックは、着信要求のペイロードに対して実行され、攻撃文字列が複数行に分散している場合でも識別されます。このチェックでは、要素内の SQL インジェクション文字列と 属性値が検索されます。緩和を適用して、指定した条件下でセキュリティチェック検査をバイパスできます。ログと統計情報は、必要な緩和を特定するのに役立ちます。

アクションオプション

アクションは、XML SQL インジェクションチェックがリクエストで SQL インジェクション攻撃文字列を検出したときに適用されます。アプリケーションに最適化された XML SQL インジェクション保護を構成するには、次のアクションを使用できます。

Block: ブロックを有効にすると、入力が XML SQL インジェクションタイプの指定と一致した場合にのみ、ブロックアクションがトリガーされます。たとえば、**SQLSplCharANDKeyword** が XML SQL インジェクションタイプとして設定されている場合、リクエストにキーワードが含まれていない場合、ペイロードで SQL 特殊文字が検出された場合でも、リクエストはブロックされません。XML SQL インジェクションタイプが **SQLSplChar** または **SQLSplCharOR** キーワードのいずれかに設定されている場合、このような要求はブロックされます。

Log: ログ機能を有効にすると、XML SQL インジェクションチェックによって実行されるアクションを示すログメッセージが生成されます。ブロックが無効になっている場合は、XML SQL 違反が検出された場所 (要素、属性) ごとに個別のログメッセージが生成されます。ただし、要求がブロックされると、1 つのメッセージだけが生成されます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとした可能性があります。

Stats: 有効の場合、stats 機能は違反とログに関する統計情報を収集します。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされている場合は、設定を再度参照して、新しい緩和ルールを設定するか、既存の緩和ルールを変更する必要があるかを確認する必要があります。

XML SQL パラメーター

ブロック、ログ、および統計アクションに加えて、XML SQL インジェクションチェックの次のパラメータを設定できます。

XML SQL ワイルドカード文字のチェック: ワイルドカード文字を使用して、構造化クエリ言語 (SQL-SELECT) ステートメントの選択範囲を広げることができます。これらのワイルドカード演算子は、**LIKE** および **NOT LIKE** 演算子と組み合わせて使用して、値を同様の値と比較できます。パーセント (%) およびアンダースコア (_) 文字は、ワイルドカードとしてよく使用されます。パーセント記号は、MS-DOS で使用されるアスタリスク (*) ワイルドカード文字に似ており、フィールド内の 0、1、または複数の文字に一致します。アンダースコアは、MS-DOS の疑問符 (?) に似ています。ワイルドカード文字。これは、式内の単一の数字または文字と一致します。

たとえば、次のクエリを使用して文字列検索を実行し、名前に D 文字が含まれているすべての顧客を検索できます。

```
SELECT * from customer WHERE name like "%D%"
```

次の例では、演算子を組み合わせて、2 番目と 3 番目の文字として 0 を持つ給与値を検索します。

```
SELECT * from customer WHERE salary like '_00%
```

別の DBMS ベンダーは、余分な演算子を追加することにより、ワイルドカード文字を拡張しました。Citrix Web App Firewall は、これらのワイルドカード文字を挿入することによって開始される攻撃から保護できます。デフォ

ルトのワイルドカード文字は、パーセント（%）、アンダースコア（_）、キャレット（^）、開始角括弧（[）、および閉じ角括弧（]）です。この保護は、HTML プロファイルと XML プロファイルの両方に適用されます。

デフォルトのワイルドカード文字は、***Default Signatures** で指定されたリテラルのリストです。

```
1 -  <wildchar type=" LITERAL" >%</wildchar>
2 -  <wildchar type=" LITERAL" >_</wildchar>
3 -  <wildchar type=" LITERAL" >^</wildchar>
4 -  <wildchar type=" LITERAL" >[</wildchar>
5 -  <wildchar type=" LITERAL" >]</wildchar>
6 <!--NeedCopy-->
```

攻撃のワイルドカード文字は [^A-F] のように PCRE にすることができます。Web App Firewall は PCRE ワイルドカードもサポートしていますが、ほとんどの攻撃をブロックするには、上記のリテラルワイルドカード文字で十分です。

> 注
>
> XML SQL ワイルドカード文字チェックは、XML SQL の特殊文字チェックとは異なります。このオプションは、誤検出を避けるために注意して使用する必要があります。

SQL インジェクションタイプを含むリクエストのチェック：Web App Firewall には、アプリケーションの個々のニーズに基づいて、SQL インジェクションインスペクションに必要なレベルの厳密さを実装するための 4 つのオプションがあります。要求は、SQL 違反を検出するためのインジェクションタイプの指定と照らし合わせてチェックされます。4 つの SQL インジェクションタイプのオプションは次のとおりです。

- **SQL** 特殊文字とキーワード：SQL 違反をトリガーするには、検査場所に SQL キーワードと SQL 特殊文字の両方が存在する必要があります。この制限の最小限の設定もデフォルトの設定です。
- [**SQL** 特殊文字]：SQL 違反をトリガーするには、処理されたペイロード文字列に少なくとも 1 つの特殊文字が含まれている必要があります。
- **SQL** キーワード：SQL 違反をトリガーするには、処理されたペイロードストリングに、指定された SQL キーワードの少なくとも 1 つが存在する必要があります。このオプションは考慮せずに選択しないでください。誤検出を回避するには、入力にキーワードがないことを確認してください。
- **SQL** 特殊文字またはキーワード：セキュリティチェック違反をトリガーするには、キーワードまたは特殊文字列がペイロードに存在している必要があります。

> ヒント
>
> [SQL 特殊文字] オプションを選択すると、Web App Firewall は特殊文字を含まない文字列をスキップします。ほとんどの SQL サーバーは特殊文字が前に付いていない SQL コマンドを処理しないため、このオプションを有効にすると、保護された Web サイトを危険にさらすことなく、Web App Firewall の負荷を大幅に減らし、処理を高速化できます。

SQL コメント処理：デフォルトでは、Web App Firewall は XML データ内のすべてのコメントを解析し、挿入された SQL コマンドがないかチェックします。多くの SQL サーバーは、SQL 特殊文字が先行していても、コメント内の

すべてを無視します。処理を高速化するために、XML SQL サーバーがコメントを無視する場合、挿入された SQL のリクエストを調べるときにコメントをスキップするように Web App Firewall を設定できます。XML SQL コメント処理オプションは次のとおりです。

- **ANSI**—UNIX ベースの SQL データベースで通常使用される ANSI 形式の SQL コメントをスキップします。
- ネスト: ネストされた SQL コメントをスキップします。このコメントは、Microsoft SQL Server で通常使用されます。
- [**ANSI/**ネスト]: ANSI およびネストされた SQL コメント標準の両方に準拠するコメントをスキップします。ANSI 標準のみ、またはネストされた標準のみに一致するコメントは、挿入された SQL についてチェックされます。
- すべてのコメントをチェック: 何もスキップせずに、注入された SQL のリクエスト全体をチェックします。これがデフォルトの設定です。

> ヒント
>
> ほとんどの場合、ネストまたは ANSI/Nested バックエンドデータベースが MicrosoftSQL Server で実行されていない限り、オプション。他のほとんどの種類の SQL Server ソフトウェアは、ネストされたコメントを認識しません。ネストされたコメントが別の種類の SQL サーバーに送信された要求に表示された場合、そのサーバーのセキュリティ侵害の試みを示している可能性があります。

リラクゼーションルール

アプリケーションで、XML ペイロード内の特定の ELEMENT または ATTRIBUTE に対して XML SQL インジェクション検査をバイパスする必要がある場合は、緩和規則を設定できます。XML SQL インジェクションインスペクションのリラクゼーション規則には、次のパラメータがあります。

- **[名前]:** リテラル文字列または正規表現を使用して、要素の名前または **ATTRIBUTE** の名前を設定できます。次の式は、文字列 **PurchaseOrder_** で始まり、2 文字以上 10 文字以下の数値の文字列が続くすべての 要素を除外します。

  コメント: 「発注要素の XML SQL チェックを免除」

```
1     XMLSQLInjection:  ”PurchaseOrder_[0-9A-Za-z]{
2  2,10 }
3  ”
4
5     IsRegex:  REGEX          Location:  ELEMENT
6
7     State:  ENABLED
8  <!--NeedCopy-->
```

注: 名前では大文字と小文字が区別されます。重複するエントリは許可されませんが、名前や場所の違いを大文字にすることで、類似したエントリを作成できます。たとえば、次の緩和規則はそれぞれ一意です。


```
1)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

2)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

3)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ATTRIBUTE     State:  ENABLED

4)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ATTRIBUTE     State:  ENABLED
<!--NeedCopy-->
```

- 場所: XML ペイロードで XML SQL インスペクション例外の場所を指定できます。オプション **ELEMENT** はデフォルトで選択されています。**ATTRIBUTE** に変更できます。
- コメント: これはオプションのフィールドです。この緩和規則の目的を説明するために、最大 255 文字の文字列を使用できます。

> 警告
>
> 正規表現は強力です。特に、PCRE 形式の正規表現に精通していない場合は、記述した正規表現を再確認してください。例外として追加する名前を正確に定義していることを確認し、それ以外は何も定義しないでください。正規表現を不注意に使用すると、意図しない Web コンテンツへのアクセスをブロックしたり、XML SQL インジェクション検査でブロックされた攻撃を許可するなど、望ましくない結果が生じる可能性があります。

コマンドラインを使用した **XML SQL** インジェクションチェックの設定

コマンドラインを使用して **XML SQL** インジェクションアクションおよびその他のパラメーターを構成するには、次の手順に従います。

コマンドラインインターフェイスでは、**set appfw profile** コマンドまたは add appfw profile** コマンドのいずれかを使用して、**XML SQL** インジェクション保護を設定できます。ブロック、ログ、および統計アクションを有効にできます。ペイロードで検出する **SQL** 攻撃パターンのタイプ（キーワード、ワイルドカード文字、特殊文字列）を選択します。設定済みの設定をデフォルトに戻すには、****unset appfw profile** コマンドを使用します。次のコマンドはそれぞれ 1 つのパラメータのみを設定しますが、1 つのコマンドに複数のパラメータを含めることができます。

- `set appfw profile <name> **-XMLSQLInjectionAction** (([block] [log] [stats])| [none])`
- `set appfw profile <name> -XMLSQLInjectionCheckSQLWildChars (ON |OFF)`

- set appfw profile <name> -XMLSQLInjectionType ([SQLKeyword] | [SQLSplChar] | [SQLSplCharANDKeyword] | [SQLSplCharORKeyword])
- set appfw profile <name> -XMLSQLInjectionParseComments ([checkall] | [ansi|nested] | [ansinested])

コマンドラインを使用して SQL インジェクション緩和ルールを構成するには

次のように、緩和ルールを追加または削除するには、bind または unbind コマンドを使用します。

```
1 -  bind appfw profile <name> -XMLSQLInjection <string> [isRegex (REGEX
     | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )]  - comment <string>
     [-state ( ENABLED | DISABLED )]
2 -  unbind appfw profile <name> -XMLSQLInjection <String>
3 <!--NeedCopy-->
```

例:

```
1 > bind appfw profile test_profile -XMLSQLInjection ”PurchaseOrder_[0-9A
     -Za-z]{
2  2,15 }
3  ” -isregex REGEX -location ATTRIBUTE
4
5 > unbind appfw profile test_profile  - XMLSQLInjection ”PurchaseOrder_
     [0-9A-Za-z]{
6  2,15 }
7  ” -location ATTRIBUTE
8 <!--NeedCopy-->
```

GUI を使用した XMLSQL インジェクションセキュリティチェックの設定

GUI では、アプリケーションに関連付けられたプロファイルの [XML SQL Injection] セキュリティー検査を区画で設定できます。

GUI を使用して XML SQL インジェクションチェックを構成または変更するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定] ペインで、[セキュリティチェック] をクリックします。

セキュリティチェックテーブルには、すべてのセキュリティチェックに対して現在構成されているアクション設定が表示されます。設定には 2 つのオプションがあります。

a. XML SQL インジェクションのブロック、ログ、および統計アクションを有効または無効にするだけの場合は、テーブルのチェックボックスをオンまたはオフにして [OK] をクリックし、[保存して閉じる] をクリックして [セキュリティチェック] ウィンドウを閉じます。

b. このセキュリティー検査の追加オプションを構成する場合は、「XML SQL Injection」をダブルクリックするか、行を選択して「アクションの設定」をクリックして、次のオプションを表示します。

SQL ワイルドカード文字のチェック: ペイロード内の SQL ワイルドカード文字を攻撃パターンと見なします。

チェック要求: チェックする SQL インジェクションのタイプ (SQL キーワード、SQLSplchar、SQLSplcharand キーワード、または SQLSplcharorKeyword)。

SQL コメント処理: チェックするコメントのタイプ ([すべてのコメント]、[ANSI]、[ネストされた]、または [ANSI/ネスト])。

上記の設定のいずれかを変更したら、[**OK**] をクリックして変更を保存し、[セキュリティチェック] テーブルに戻ります。必要に応じて、他のセキュリティー検査の設定に進むことができます。[**OK]** をクリックして [セキュリティチェック] セクションで行ったすべての変更を保存し、[保存して 閉じる] をクリックして [セキュリティチェック] ペインを閉じます。

GUI を使用して XML SQL インジェクション緩和ルールを構成するには

1. **Web** AppFirewall に移動します > プロファイルをクリックし、ターゲットプロファイルを強調表示して、[編集] をクリックします。
2. [詳細設定**]** ペインで、[緩和規則] をクリックします。
3. 「緩和規則」テーブルで、「**XML SQL** インジェクション」エントリをダブルクリックするか、エントリを選択して「編集」をクリックします。
4. 「**XML SQL** インジェクション緩和規則」ダイアログで、緩和規則の「追加」、「編集」、「削除」、「有効化」、または「無効化」の操作を実行します。

ビジュアライザーを使用して XML SQL インジェクション緩和ルールを管理するには

すべての緩和ルールをまとめて表示するには、[緩和ルール] テーブルの [**XML SQL Injection**] 行をハイライト表示し、[ビジュアライザー] をクリックします。展開されたリラクゼーションのビジュアライザーには、新しいルールを追加するか、既存のルールを 編集するオプションがあります。ノードを選択し、緩和ビジュアライザの対応するボタンをクリックして、ルールのグループ を有効または 無効にすることもできます。

GUI を使用して **SQL** インジェクション・パターンを表示またはカスタマイズするには、次のステップを実行します。

GUI を使用して、SQL パターンを表示またはカスタマイズできます。

デフォルトの SQL パターンは、[**Web App Firewall**] > [署名] > [***** デフォルトの署名] で指定されます。シグネチャオブジェクトをプロファイルにバインドしない場合、Default Signatures オブジェクトで指定されたデフォルトの SQL パターンが、プロファイルによって XML SQL Injection セキュリティチェック処理に使用されます。デフォルトシグニチャオブジェクトの規則とパターンは読み取り専用です。編集や修正はできません。これらのパターンを変更または変更する場合は、Default Signatures オブジェクトのコピーを作成し、SQL パターンを変更して、ユーザー定義のシグニチャオブジェクトを作成します。カスタマイズされた SQL パターンを使用するトラフィックを処理するプロファイルで、ユーザ定義のシグニチャオブジェクトを使用します。

詳細については、「署名」を参照してください。

デフォルトの **SQL** パターンを表示するには、次のステップを実行します。

a. [**Web App Firewall**] > [署名] に移動し、[***** デフォルトの署名] を選択して [編集] をクリックします。次に、[管理] をクリックします SQL/cross-site スクリプトパターン。

管理 SQL/cross-site スクリプトパステーブルは、SQL インジェクションに関連する次の 4 行を示しています。

```
Injection (not_alphanum, SQL)/ Keyword

Injection (not_alphanum, SQL)/ specialstring

Injection (not_alphanum, SQL)/ transformrules/transform

Injection (not_alphanum, SQL)/ wildchar
<!--NeedCopy-->
```

b. 行を選択して「エレメントの管理」をクリックすると、Web App Firewall SQL インジェクションチェックで使用される対応する SQL パターン (キーワード、特殊文字列、変換ルール、ワイルドカード文字) が表示されます。

SQL パターンをカスタマイズするには**:** ユーザー定義の署名オブジェクトを編集して、SQL キーワード、特殊文字列、およびワイルドカード文字をカスタマイズできます。新しいエントリを追加したり、既存のエントリを削除したりできます。SQL 特殊文字列の変換ルールを変更できます。

a. [**Web App Firewall] > [**署名**]** に移動し、ターゲットのユーザー定義署名をハイライト表示して [編集] をクリックします。[管理**]** をクリックします SQL/cross-site 管理を表示するためのスクリプトパターン SQL/cross-site スクリプトパステーブル。

b. ターゲット SQL 行を選択します。

i. [要素の管理] をクリックして、対応する SQL 要素を 追加、編集、または削除します。

ii. [削除**]** をクリックして、選択した行を削除します。

> 警告
>
> デフォルトの SQL 要素を削除または変更する場合、または SQL パスを削除して行全体を削除する場合は、十分に注意する必要があります。シグニチャルールと XML SQL Injection セキュリティチェックは、これらの要素に依存して SQL Injection 攻撃を検出し、アプリケーションを保護します。SQL パターンをカスタマイズすると、編集中に必要なパターンが削除されると、アプリケーションが XML SQL 攻撃に対して脆弱になる可能性があります。

XML SQL インジェクションチェックでのログ機能の使用

ログアクションを有効にすると、**XML SQL** インジェクションのセキュリティチェック違反が **APPFW_XML_SQL** 違反として監査ログに記録されます。Web App Firewall は、ネイティブログ形式と CEF ログ形式の両方をサポー

トしています。ログをリモート syslog サーバに送信することもできます。

コマンドラインを使用してログメッセージにアクセスするには、次の手順を実行します。

シェルに切り替えて、/var/log/ フォルダ内の ns.logs を末尾に付けて、XML クロスサイトスクリプティング違反に関連するログメッセージにアクセスします。

```
1 > Shell
2
3 > tail -f /var/log/ns.log | grep APPFW_XML_SQL
4 <!--NeedCopy-->
```

GUI を使用してログメッセージにアクセスするには

Citrix GUI には、ログメッセージを分析するための便利なツール（Syslog Viewer）が含まれています。Syslog ビューアには、次の複数のオプションがあります。

- [**Web App Firewall**] > [プロファイル] に移動し、ターゲットプロファイルを選択して [セキュリティチェック] をクリックします。[**XML SQL** インジェクション] 行をハイライト表示し、[ログ] をクリックします。プロファイルの XML SQL Injection チェックから直接ログにアクセスすると、GUI によってログメッセージが除外され、これらのセキュリティチェック違反に関連するログのみが表示されます。
- [システム] > [監査] に移動して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして Syslog Viewer を表示します。このビューアーには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。これは、要求処理中に複数のセキュリティー検査違反がトリガーされる可能性がある場合のデバッグに役立ちます。
- [**Web App Firewall**] > [ポリシー] > [監査] に移動します。[監査メッセージ] セクションで、[**Syslog** メッセージ] リンクをクリックして **Syslog** ビューアを表示します。このビューアには、他のセキュリティチェック違反ログを含むすべてのログメッセージが表示されます。

XML ベースの Syslog ビューアには、関心のあるログメッセージだけを選択するためのさまざまなフィルタオプションがあります。**XML SQL** インジェクションチェックのログメッセージを選択するには、[モジュール] のドロップダウンオプションで [**APPFW**] を選択してフィルタリングします。**[Event Type]** リストには、選択内容をさらに絞り込むための豊富なオプションが用意されています。たとえば、[**APPFW_XML_SQL**] チェックボックスをオンにして [適用] ボタンをクリックすると、**XML SQL Injection** セキュリティチェック違反に関するログメッセージだけが Syslog ビューアに表示されます。

特定のログメッセージの行にカーソルを置くと、モジュール、イベントタイプ、イベント **ID**、クライアント **IP** などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログメッセージ内の対応する情報を強調表示することができます。

XML SQL インジェクション違反の統計

stats アクションが有効な場合、Web App Firewall がこのセキュリティチェックに対して何らかのアクションを実行すると、**XML SQL** インジェクションチェックのカウンタが増加します。統計は、トラフィック、違反、およびログのレートと合計数について収集されます。ログカウンタの増分サイズは、構成された設定によって異なります。たとえば、ブロック・アクションが有効になっている場合、3 つの **XML SQL Injection** 違反を含むページに対するリクエストは、最初の違反が検出されるとすぐにページがブロックされるため、stats カウンタを 1 つずつインクリメントします。ただし、ブロックが無効になっている場合、同じ要求を処理すると、違反とログの統計カウンタが 3 ずつ増加します。これは、違反ごとに個別のログメッセージが生成されるためです。

コマンドラインを使用して XML SQL インジェクションチェック統計を表示するには

コマンドプロンプトで入力します。

```
> sh appfw stats
```

特定のプロファイルの統計情報を表示するには、次のコマンドを使用します。

```
> stat appfw profile <profile name>
```

GUI を使用して XML SQL インジェクション統計を表示するには

1. [システム] > [セキュリティ] > [**Web App Firewall**] に移動します。
2. 右側のペインで、[統計リンク] にアクセスします。
3. スクロールバーを使用して、**XML SQL** インジェクションの違反とログに関する統計を表示します。統計テーブルはリアルタイムデータを提供し、7 秒ごとに更新されます。

XML 添付ファイルのチェック

October 7, 2021

XML 添付ファイルのチェックは、受信した要求に悪意のある添付ファイルを検査し、アプリケーションのセキュリティに違反する可能性のある添付ファイルを含む要求をブロックします。XML 添付ファイルのチェックの目的は、攻撃者が XML 添付ファイルを使用してサーバーのセキュリティを侵害するのを防ぐことです。

ウィザードまたは GUI を使用する場合は、[XML 添付ファイルのチェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ラーニング]、[ログ]、[統計]、および [ラーニング] の各アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML 添付ファイルのチェックを構成できます。

- `set appfw profile <name> -xmlAttachmentAction [block] [learn] [log] [stats] [none]`

GUI で他の XML 添付ファイルのチェック設定を構成する必要があります。[`Modify XML Attachment`チェック] ダイアログボックスの [チェック] タブで、次の設定を構成できます。

- 添付ファイルの最大サイズ。指定した最大添付ファイルサイズを超えない添付ファイルを許可します。このオプションを有効にするには、まず [有効] チェックボックスをオンにし、`Size` テキストボックスに添付ファイルの最大サイズをバイト単位で入力します。
- 添付ファイルのコンテンツタイプ。指定したコンテンツタイプの添付ファイルを許可します。このオプションを有効にするには、最初に [有効] チェックボックスをオンにし、許可する添付ファイルの [Content-Type] 属性と一致する正規表現を入力します。
  - URL 式は、テキストウィンドウに直接入力できます。その場合は、`Regex Tokens` メニューを使用して、手動で入力する代わりに、カーソル位置に便利な正規表現をいくつか入力できます。
  - [Regex Editor] をクリックして`Add Regular Expression` ダイアログボックスを開き、それを使用して URL 式を作成できます。

Web サービスの相互運用性チェック

October 7, 2021

Web サービス相互運用性 (WS-I) チェックでは、要求と応答の両方が WS-I 標準に準拠しているかどうかが検査され、この標準に準拠していない要求と応答がブロックされます。WS-I チェックの目的は、他の XML と適切にやり取りしない可能性のある要求をブロックすることです。攻撃者は、相互運用性の不整合を使用して、XML アプリケーションに対する攻撃を開始する可能性があります。

ウィザードまたは GUI を使用する場合は、[Web サービスの相互運用性チェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、[統計]、および [ラーニング] の各アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して Web サービスの相互運用性チェックを構成できます。

- `set appfw profile <name> -xmlWSIAction [block] ][log] [learn] [stats] [ none]`

個別の Web サービス相互運用性ルールを構成するには、GUI を使用する必要があります。[Web サービスの相互運用性チェックの変更] ダイアログボックスの [チェック] タブで、ルールを選択し、[有効] または [無効] をクリックしてルールを有効または無効にします。[開く] をクリックして、そのルールの [Web サービス相互運用性の詳細] メッセージボックスを開くこともできます。メッセージボックスには、ルールに関する読み取り専用情報が表示されます。これらのルールを変更したり、その他の設定を変更したりすることはできません。

WS-I チェックでは、WS-I 基本プロファイル 1.0 にリストされているルールが使用されます。WS-I は、相互運用可能な Web サービスソリューションを開発するためのベストプラクティスを提供します。WS-I チェックは SOAP メッセージに対してのみ実行されます。

各 WSI 標準規則の説明を以下に示します。

規則	説明
BP1201	メッセージ本文は、名前空間を持つ soap:envelope である必要があります。
R1000	ENVELOPE が Fault である場合、soap:Fault 要素は、faultcode、faultstring、faultactor、detail 以外の要素の子を持つことはできません。
R1001	ENVELOPE が Fault である場合、soap:Fault 要素の子は、修飾されなければなりません。
R1003	受信者が詳細要素に表示される任意の数の修飾属性または非修飾属性（ゼロを含む）を持つフォルト・メッセージを受け入れる必要があります。修飾された属性の名前空間は、修飾されたドキュメントエレメントの Envelope の名前空間以外の任意のものにすることができます。
R1004	ENVELOPE に faultcode 要素が含まれている場合、その要素のコンテンツは、SOAP 1.1 で定義されている障害コード（詳細要素で必要に応じて追加情報を提供）のいずれか、または名前空間が障害の指定機関によって制御されている Qname（優先順に）。
R1005	ENVELOPE は、その名前空間が修飾された文書要素 Envelope の名前空間と同じである要素のいずれかの soap を含んではいけません。
R1006	EnVELOPE は、soap:Body の子である任意の要素の soap:encodingStyle 属性を含んではいけません。
R1007	rpc リテラルバインディングで記述された ENVELOPE は、soap:Body の孫である任意の要素の soap:encodingStyle 属性を含んではいけません。
R1011	EnVELOPE には、SOAPE: Body 要素に続く SOAPE: Envelope の要素の子があってはなりません。
R1012	メッセージを UTF-8 または UTF-16 としてシリアル化する必要があります。
R1013	soap:mustUnderstand 属性を含む ENVELOPE は、字句形式 0 と 1 のみを使用しなければなりません。
R1014	ENVELOPE の soap:Body 要素の子は、名前空間修飾でなければなりません。

規則	説明
R1015	受信者は、ドキュメント要素が soap:Envelope でないエンベロープに遭遇した場合、フォルトを生成する必要があります。
R1031	ENVELOPE に faultcode 要素が含まれている場合、その要素のコンテンツは、障害の意味を絞り込むために SOAP1.1 ドット表記を使用してはなりません。
R1032	The soap:Envelope, soap:Header, and soap:ENVELOPE 内のボディ要素は、修飾された文書要素のエンベロープのそれと同じ名前空間内の属性を持つことはできません
R1033	EnVELOPE は名前空間宣言を含んではいけません: `xmlns:xml=http://www.w3.org/XML/1998/namespace.`
R1109	HTTP 要求メッセージの SOAPAction HTTP ヘッダーフィールドの値は、引用符で囲まれた文字列でなければなりません。
R1111	INSTANCE は、障害ではないエンベロープを含む応答メッセージに対して 200 OK HTTP ステータスコードを使用すべきです。
R1126	応答エンベロープが Fault の場合、INSTANCE は 500 内部サーバーエラー HTTP ステータスコードを返す必要があります。
R1132	HTTP リクエストメッセージでは、HTTP POST メソッドを使用する必要があります。
R1140	メッセージは HTTP/1.1 を使用して送信する必要があります。
R1141	メッセージは、HTTP/1.1 または HTTP/1.0 のいずれかを使用して送信する必要があります。
R2113	エンベロープには、soapenc:arrayType 属性を含めることはできません。
R2211	rpc-literal バインディングで記述された ENVELOPE は、パーツアクセサの値が 1 または true の xsi: nil 属性を持つことはできません。

規則	説明
R2714	一方向操作の場合、INSTANCE はエンベロープを含む HTTP 応答を返してはいけません。具体的には、HTTP 応答エンティティ本体は空である必要があります。
R2729	応答である rpc リテラルバインディングで記述された ENVELOPE は、対応する wsdl：操作名に接尾辞 stringResponse であるラッパー要素を持たなければなりません。
R2735	rpc-literal バインディングで記述された ENVELOPE は、パラメータと戻り値のパートアクセサ要素を名前空間なしに配置しなければなりません。
R2738	ENVELOPE には、wsdl：input または wsdl：それを記述する wsdl：バインディングの wsdl：操作の wsdl：出力で指定されたすべての soapbind を含める必要があります。
R2740	説明の wsdl：バインディングには、それぞれの既知の障害を記述する soapbind を含める必要があります。
R2744	HTTP 要求メッセージには、対応する WSDL 記述内に存在する場合、soapbind:operation の soapAction 属性の値と等しい引用符で囲まれた値を持つ SOAPAction HTTP ヘッダーフィールドが含まれていなければなりません。

XML メッセージの検証チェック

October 7, 2021

XML メッセージ検証チェックは、XML メッセージを含む要求を検証して、それらの要求が有効であることを確認します。要求に無効な XML メッセージが含まれている場合、Web App Firewall は要求をブロックします。XML 検証チェックの目的は、攻撃者が特別に構築された無効な XML メッセージを使用してアプリケーションのセキュリティを侵害することを防ぐことです。

ウィザードまたは GUI を使用する場合は、[XML メッセージ検証チェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、および [統計] の各アクションを有効または無効にできます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML メッセージ検証チェックを設定できます。

- `set appfw profile <name> -xmlValidationAction [**block**] [**log**] [**stats**] [**none**]`

GUI を使用して、他の XML 検証チェック設定を構成する必要があります。
[XML メッセージ検証チェックの変更] ダイアログボックスの [
チェック] タブで、次の設定を構成できます。

- **XML** メッセージの検証。次のいずれかのオプションを使用して、XML メッセージを検証します。
  - **SOAP** エンベロープ。XML メッセージの SOAP エンベロープのみを検証します。
  - **WSDL**。XML SOAP WSDL を使用して XML メッセージを検証します。「WSDL 検証」を選択した場合は、「WSDL オブジェクト」ドロップダウンリストで WSDL を選択する必要があります。Web App Firewall にまだインポートされていない WSDL に対して検証する場合は、「インポート」ボタンをクリックして「WSDL インポートの管理」ダイアログを開き、WSDL をインポートします。詳細については、WSDL を参照してください。
    * URL 全体を検証する場合は、[エンドポイントチェック] ボタン配列の [絶対] ラジオボタンを選択したままにします。ホストの後の URL の一部のみを検証する場合は、[相対] ラジオボタンを選択します。
    * Web App Firewall で WSDL を厳密に適用し、WSDL で定義されていない追加の XML ヘッダーを許可しない場合は、[WSDL で定義されていない追加のヘッダーを許可する] チェックボックスをオフにする必要があります。
      注意:「WSDL で定義されていないヘッダーを
      許可する」チェックボックスのチェックを外し、保護された XML アプリケーションまたは Web 2.0 アプリケーションが期待する XML ヘッダーや、クライアントが送信する XML ヘッダーが WSDL で定義されていない場合、保護されたサービスへの正当なアクセスをブロックする可能性があります。
  - **XML** スキーマ。XML スキーマを使用して XML メッセージを検証します。XML スキーマの検証を選択した場合は、「XML スキーマオブジェクト」ドロップダウンリストで XML スキーマを選択する必要があります。Web App Firewall にまだインポートされていない XML スキーマに対して検証する場合は、「インポート」ボタンをクリックして「XML スキーマのインポートの管理」ダイアログを開き、WSDL をインポートします。詳細については、WSDL を参照してください。
- 応答の検証。デフォルトでは、Web App Firewall は応答を検証しません。保護されたアプリケーションまたは Web 2.0 サイトからの応答を検証する場合は、[応答の検証] チェックボックスをオンにします。この操作を行うと、[要求の検証で指定された XML スキーマを再利用する] チェックボックスと [XML スキーマオブジェクト] ドロップダウンリストがアクティブになります。
  - [Reuse XML Schema] チェックボックスをオンにして、要求の検証に指定したスキーマを使用して応答の検証も行います。
    注: このチェックボックスをオンにすると、「
    XML スキーマオブジェクト」ドロップダウンリストはグレー表示されます。
  - 応答の検証に別の XML スキーマを使用する場合は、「XML スキーマオブジェクト」ドロップダウンリストを使用して、その XML スキーマを選択またはアップロードします。

XML SOAP 障害フィルタリングチェック

October 7, 2021

XML SOAP 障害フィルタリングチェックは、保護された Web サービスからの応答を検査し、XML SOAP 障害を除外します。これにより、攻撃者に機密情報が漏洩するのを防ぎます。

ウィザードまたは GUI を使用する場合は、[XML SOAP フォールトフィルタリングチェックの変更] ダイアログボックスの [全般] タブで、[ブロック]、[ログ]、および [統計] の各アクション、および [削除] アクションを有効または無効にできます。削除すると、応答がユーザーに転送される前に SOAP 障害が削除されます。

コマンドラインインターフェイスを使用する場合は、次のコマンドを入力して XML SOAP 障害フィルタリングチェックを設定できます。

```
set appfw profile <name> -XMLSOAPFaultAction [block] [log] [stats] [none]
```

XML SOAP フォルトフィルタリングチェックに例外を設定することはできません。有効化または無効化のみが可能です。

JSON 保護チェック

October 7, 2021

Citrix Web App Firewall は、コンテンツレベルの DoS、SQL、またはクロスサイトスクリプティング攻撃から JSON アプリケーションを保護します。JSON リクエストに DoS 攻撃、SQL 攻撃、またはクロスサイトスクリプティング攻撃がある場合、配列や文字列などの JSON 構造に制限を設定してアプリケーションを保護する必要があります。

> 注:
>
> JSON セキュリティチェックは、JSON コンテンツタイプヘッダーとともに送信されるコンテンツにのみ適用されます。content-type ヘッダーが見つからないか、別の値に設定されている場合、すべての JSON セキュリティチェックはバイパスされます。JSON アプリケーションを保護する場合、それらのアプリケーションをホストする各 Web サーバーのウェブマスターは、適切な JSON コンテンツタイプのヘッダーが送信されるようにする必要があります。
>
> 学習機能は、JSON SQL、クロスサイトスクリプティング、DOS コンテンツタイプをサポートしていません。

JSON サービス拒否保護チェック

October 7, 2021

JSON サービス拒否 (DoS) チェックは、受信 JSON リクエストを調べ、DoS 攻撃の特性と一致するデータがあるかどうかを検証します。リクエストに JSON 違反があった場合、アプライアンスはリクエストをブロックし、データをログに記録し、SNMP アラートを送信し、JSON エラーページも表示します。JSON DoS チェックの目的は、攻撃者が JSON リクエストを送信して JSON アプリケーションまたは Web サイトに対して DoS 攻撃を起動するのを防ぐことです。

クライアントが Citrix ADC アプライアンスにリクエストを送信すると、JSON パーサーはリクエストペイロードを解析し、違反が観察された場合、アプライアンスは JSON 構造に制約を適用します。この制約は、JSON リクエストのサイズ制限を強制します。その結果、JSON 違反が発生した場合、アプライアンスはアクションを適用し、JSON エラーページに応答します。

JSON DoS 規則

アプライアンスが JSON リクエストを受信すると、JSON DOS 保護により、リクエストペイロードの次の DoS パラメータにサイズ制限が適用されます。

1. 最大深さ：JSON ドキュメントの最大ネスト（深さ）。このチェックは、階層の深さが過剰なドキュメントに対して保護します。
2. 最大ドキュメント長：JSON ドキュメントの最大ドキュメント長。
3. 最大配列長：JSON オブジェクトのいずれかの最大配列長。このチェックは、長い長さの配列から保護します。
4. 最大文字列長：JSON の最大文字列長。このチェックは、長い長さの文字列から保護します。
5. 最大オブジェクトキー数：JSON オブジェクトの最大キー数。このチェックは、多数のキーを持つオブジェクトに対して保護します。
6. オブジェクトキーの最大長：JSON オブジェクトの最大キー長。このチェックは、大きなキーを持つオブジェクトに対して保護します。

JSON 解析中に検証された JSON DoS ルールのリストを次に示します。

1. JSONMaxContainerDepth. このチェックは、JSONMaxContainerDepth チェックを設定することで有効にすることができます。デフォルトでは、このオプションはオフです。
2. JSONMaxContainerDepth. このチェックは enabled/disabled 構成可能なオプション JSONMaxContainerDepthCheck によって、デフォルト値はオプション JSONMaxContainerDepth によって変更できます。ただし、最大レベルを 1 ～127 の範囲の値に変更できます。デフォルト値:5, 最小値:1, 最大値:127
3. JSONMaxDocumentLength. このチェックを有効にするには、JSONMaxDocumentLength チェックを設定し、デフォルトのオプションは OFF です。
4. JSONMaxDocumentLength. このチェックを有効にするには、JSONMaxDocumentLength チェックを設定し、デフォルトの長さを 20000000 バイトに設定します。最小値：1、最大値：2147483647
5. JSONMaxObjectKeyCount. このルールは、JSON の最大オブジェクトキー数のチェックがオンまたはオフになっているかどうかを検証します。設定可能な値：ON、OFF、デフォルト値：OFF
6. JSONMaxObjectKeyCount. このチェックは、JSONMaxObjectKeyCount チェックを設定することで有

効にできます。このチェックでは、多数のキーを持つオブジェクトに対して保護され、デフォルト値は 1000 バイトに設定されます。最小値:0, 最大値:2147483647

7. JSONMaxObjectKeyLength. このチェックは、JSONMaxObjectKeyLength チェックを設定することで有効にすることができます。このルールは、JSON オブジェクトキーの最大長チェックがオンまたはオフになっているかどうかを検証します。デフォルトではオフになっています。
8. JSONMaxObjectKeyLength. このチェックでは、キーの長さが大きいオブジェクトに対して保護されます。デフォルト値は 128 です。最小値：1、最大値：2147483647
9. JSONMaxArrayLength. このルールは、JSON の最大配列長チェックが ON または OFF かどうかを検証します。デフォルトではオフになっています。
10. JSONMaxArrayLength. このチェックは、長さが大きい配列に対して保護します。デフォルトでは、値は 10000 に設定されています。最小値：1、最大値：2147483647
11. JSONMaxStringLength。このチェックは、JSONMaxStringLength チェックを設定することで有効にすることができます。このチェックは、JSON の最大文字列長が ON または OFF かどうかを検証します。デフォルトではオフになっています。
12. JSONMaxStringLength。チェックは、長い長さの文字列から保護します。デフォルトでは 1000000 に設定されています。最小値：1、最大値：2147483647

JSON DoS 保護チェックの設定

JSON DoS 保護を設定するには、次の手順を完了する必要があります。

1. JSON のアプリケーションファイアウォールプロファイルを追加します。
2. JSON DoS 設定のアプリケーションファイアウォールプロファイルを設定します。
3. アプリケーションのファイアウォールプロファイルをバインドして JSON DoS 変数を設定します。

JSON DoS 保護用のアプリケーションファイアウォールプロファイルの追加

まず、アプリケーションファイアウォールが JSON の DoS 攻撃から JSON Web コンテンツを保護する方法を指定するプロファイルを作成する必要があります。
コマンドプロンプトで入力します。

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> 注:
>
> プロファイルタイプを JSON に設定すると、HTML や XML などの他のチェックは適用されません。

例

```
add appfw profile profile1 -type JSON
```


JSON DoS 保護用のアプリケーションファイアウォールプロファイルの設定

1 つ以上の JSON DoS アクションおよび JSON DoS エラーオブジェクトをアプリケーションファイアウォールプロファイルに設定するには、プロファイルを設定する必要があります。
コマンドプロンプトで入力します。

```
set appfw profile <name> -JSONDoSAction [block] | [log] | [stats] | [none]
```

Block -このセキュリティー検査に違反する接続をブロックします。
Log - このセキュリティチェックの違反を記録します。
Stats -このセキュリティー検査の統計を生成します。
None -このセキュリティー検査のすべてのアクションを無効にします。

> 注:
>
> 1 つ以上のアクションを有効にするには、「appfw プロファイルの設定-JSONDoSAction」と入力し、次に有効にするアクションを入力します。

例

```
set appfw profile profile1 -JSONDoSAction block log stat
```

アプリケーションファイアウォールプロファイルをバインドして **DoS** 変数を設定する

JSON DoS 保護を提供するには、アプリケーションファイアウォールプロファイルを JSON DoS 設定でバインドする必要があります。
コマンドプロンプトで入力します。

```
bind appfw profile <name> -JSONDoSURL <expression> [-JSONMaxContainerDepthCheck
 ( ON | OFF )[-JSONMaxContainerDepth <positive_integer>]] [-JSONMaxDocumentLengthChec
 ( ON | OFF )[-JSONMaxDocumentLength <positive_integer>]] [-JSONMaxObjectKeyCountChec
 ( ON | OFF )[-SONMaxObjectKeyCount <positive_integer>]] [-JSONMaxObjectKeyLengthChec
 ( ON | OFF )[-JSONMaxObjectKeyLength <positive_integer>]] [-JSONMaxArrayLengthCheck
 ( ON | OFF )[-JSONMaxArrayLength <positive_integer>]] [-JSONMaxStringLengthCheck
 ( ON | OFF )[-JSONMaxStringLength <positive_integer>]]
```

例

```
bind appfw profile profile1 -JSONDoSURL  ".*"  -JSONMaxContainerDepthCheck ON
```

> 注:
>
> JSON DoS チェックは、プロファイルタイプが JSON として選択されている場合にのみ適用されます。また、JSON プロファイルの場合、SQL、クロスサイトスクリプティング、フィールド形式、およびフォームフィールドの署名がクエリパラメーターに適用されます。

JSON エラーページのインポート

着信要求に DoS 攻撃があった場合、その要求をブロックすると、アプライアンスはエラーメッセージを表示します。これを行うには、JSON エラーページをインポートする必要があります。
コマンドプロンプトで入力します。

```
import appfw jsonerrorpage <src> <name> [-comment <string>] [-overwrite]
```

各項目の意味は次のとおりです。

src インポートされた JSON エラーオブジェクトを格納する場所の URL (プロトコル、ホスト、パス、および名前)。

> 注:
>
> インポートするオブジェクトが、アクセスのためにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。これは必須の引数です。最大長さ:2047。

Name: Citrix ADC 上の JSON エラーオブジェクトに割り当てる名前。これは必須の引数です。最大長さ: 31
コメント。JSON エラーオブジェクトに関する情報を保持するためのコメント。最大長:255
上書き。同じ名前の既存の JSON エラーオブジェクトを上書きします。

構成例

```
1 Add appfw prof profjson - type JSON
2 Bind appfw prof profjson - JSONDoSURL ".*" -
      JSONMaxDocumentLengthCheck ON -JSONMaxDocumentLength 30 -
      JSONMaxContainerDepthCheck ON -JSONMaxContainerDepth 3
      JSONMaxObjectKeyCountCheck ON -JSONMaxObjectKeyCount 4 -
      JSONMaxObjectKeyLengthCheck ON -JSONMaxObjectKeyLength 10 -
      JSONMaxArrayLengthCheck ON -JSONMaxArrayLength 5 -
      JSONMaxStringLengthCheck ON -JSONMaxStringLength 30
3 <!--NeedCopy-->
```

ペイロード、ログメッセージ、カウンタの例**:**

JSONMaxDocumentLength Violation

JSONMaxDocumentLength: 30
Payload: {"a":"A","b":"B","c":"C","d":"D","e":"E"}

ログメッセージ**:**


```
Document Length exceeds 20000000 May 29 20:23:32 <local0.info>
    10.217.31.243 05/29/2019:20:23:32 GMT 0-PPE-0 : default APPFW
    APPFW_JSON_DOS_MAX_DOCUMENT_LENGTH 136 0 : 10.217.32.134 114-PPE0 -
    profjson http://10.217.30.120/forms/login.html Document exceeds
    maximum document length (30). cn1=30467 cn2=115 cs1=profjson cs2=
    PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

カウンタ:

```
1 0 6 as_viol_json_dos
2 0 3 as_viol_json_dos_max_document_length
3 0 6 as_log_json_dos
4 0 3 as_log_json_dos_max_document_length
5 0 6 as_viol_json_dos_profile appfw__(profile1)
6 0 3 as_viol_json_dos_max_document_length_profile appfw__(profile1)
7 0 6 as_log_json_dos_profile appfw__(profile1)
8 0 3 as_log_json_dos_max_document_length_profile appfw__(profile1)
<!--NeedCopy-->
```

JSONMaxContainerDepth Violation

JSONMaxContainerDepth: 3
Payload: {“a”: {”b”: {“c”: {“d”: { ”e” : “f” }}}}}

ログメッセージ:

```
May 29 19:33:59 <local0.info> 10.217.31.243 05/29/2019:19:33:59 GMT 0-
    PPE-1 : default APPFW APPFW_JSON_DOS_MAX_CONTAINER_DEPTH 4626 0 :
    10.217.31.247 22-PPE1 - profjson http://10.217.30.120/forms/login.
    html Document at offset (15) exceeds maximum container depth (3).
    cn1=30466 cn2=113 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=
    blocked
<!--NeedCopy-->
```

カウンタ:

```
36 20999 7 1 0 as_viol_json_dos
37 0 6 1 0 as_viol_json_dos_max_container_depth
38 0 7 1 0 as_log_json_dos
```



```
39 0 6 1 0 as_log_json_dos_max_container_depth
40 0 7 1 0 as_viol_json_dos_profile appfw__(profile1)
41 0 6 1 0 as_viol_json_dos_max_container_depth_profile appfw__(
   profile1)
42 0 7 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 6 1 0 as_log_json_dos_max_container_depth_profile appfw__(profile1
   )
<!--NeedCopy-->
```

JSONMaxObjectKeyCount Violation

JSONMaxObjectKeyCount: 4

Payload: {“a”: ”A”, “b”: “B”, “c”:“C”, “d” :“D”, ”e” : “E” }

ログメッセージ**:**

```
May 30 19:42:41 <local0.info> 10.217.31.243 05/30/2019:19:42:41 GMT 0-
   PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_COUNT 457 0 :
   10.217.32.134 219-PPE1 - profjson http://10.217.30.120/forms/login.
   html Object at offset (41) that exceeds maximum key count (4). cn1
   =30468 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

カウンタ**:**

```
94 119105 15 1 0 as_viol_json_dos
95 0 4 1 0 as_viol_json_dos_max_object_key_count
96 0 15 1 0 as_log_json_dos
97 0 4 1 0 as_log_json_dos_max_object_key_count
98 0 15 1 0 as_viol_json_dos_profile appfw__(profile1)
99 0 4 1 0 as_viol_json_dos_max_object_key_count_profile appfw__(
   profile1)
100 0 15 1 0 as_log_json_dos_profile appfw__(profile1)
101 0 4 1 0 as_log_json_dos_max_object_key_count_profile appfw__(
   profile1)
<!--NeedCopy-->
```

JSONMaxObjectKeyLength Violation

JSONMaxObjectKeyLength: 10

Payload: {“a”: ”A”, “b1234567890”: “B”, “c”:“C”, “d” :“D”, ”e” : “E” }

ログメッセージ:

```
May 31 20:26:10 <local0.info> 10.217.31.243 05/31/2019:20:26:10 GMT 0-PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_LENGTH 102 0 : 10.217.32.134 89-PPE1 - profjson http://10.217.30.120/forms/login.html Object key(b1234567890) at offset (12) exceeds maximum key length (10). cn1=30469 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

カウンタ:

```
242172 6 1 0 as_viol_json_dos
0 1 1 0 as_viol_json_dos_max_object_key_length
10 0 5 1 0 as_log_json_dos
11 0 1 1 0 as_log_json_dos_max_object_key_length
12 0 6 1 0 as_viol_json_dos_profile appfw__(profile1)
13 0 1 1 0 as_viol_json_dos_max_object_key_length_profile appfw__(profile1)
14 0 5 1 0 as_log_json_dos_profile appfw__(profile1)
15 0 1 1 0 as_log_json_dos_max_object_key_length_profile appfw__(profile1)
<!--NeedCopy-->
```

JSONMaxArrayLength Violation

JSONMaxArrayLength: 5
ペイロード: {“a”: ”A”, “c”:[”d”,”e”,”f”,”g”,”h”,”i”],”e”:[“E”,”e”]}

ログメッセージ:

```
May 29 20:58:39 <local0.info> 10.217.31.243 05/29/2019:20:58:39 GMT 0-PPE-1 : default APPFW APPFW_JSON_DOS_MAX_ARRAY_LENGTH 4650 0 : 10.217.32.134 153-PPE1 -profjson http://10.217.30.120/forms/login.html Array at offset (37) that exceeds maximum array length (5). cn1=30469 cn2=120 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

カウンタ:


```
36 182293 10 1 0 as_viol_json_dos
37 0 1 1 0 as_viol_json_dos_max_array_length
38 0 10 1 0 as_log_json_dos 39 0 1 1 0 as_log_json_dos_max_array_length
40 0 10 1 0 as_viol_json_dos_profile appfw__(profile1)
41 0 1 1 0 as_viol_json_dos_max_array_length_profile appfw__(profile1)
42 0 10 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 1 1 0 as_log_json_dos_max_array_length_profile appfw__(profile1))
<!--NeedCopy-->
```

JSONMaxStringLength Violation

JSONMaxStringLength: 10

Payload: {“a”: ”A”, “c”:”CcCcCcCcCcCcCcCcCc”,”e”:[“E”,”e”]}

ログメッセージ**:**

```
May 29 20:05:02 <local0.info> 10.217.31.243 05/29/2019:20:05:02 GMT 0-
    PPE-0 : default APPFW APPFW_JSON_DOS_MAX_STRING_LENGTH 134 0 :
    10.217.32.134 80-PPE0 - profjson http://10.217.30.120/forms/login.
    html String(CcCcCcCcCcCcCc) at offset (27) that exceeds maximum
    string length (10). n1=30470 cn2=122 cs1=profjson cs2=PPE0 cs4=ALERT
     cs5=2019 act=blocked
<!--NeedCopy-->
```

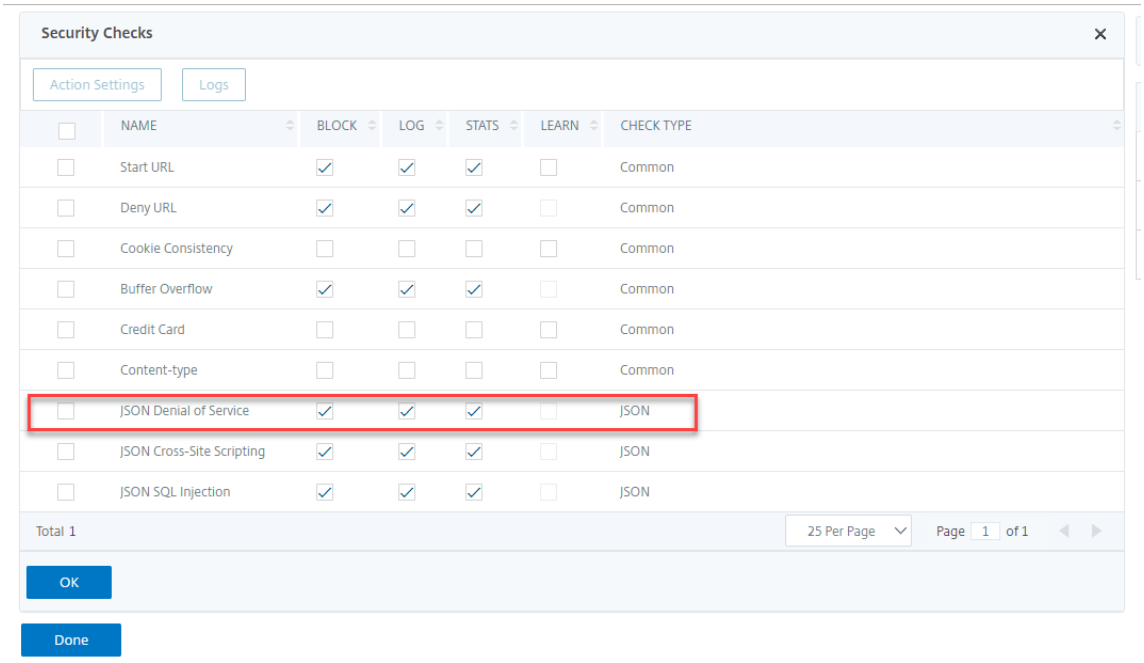
カウンタ**:**

```
44 91079 3 1 0 as_viol_json_dos
45 0 1 1 0 as_viol_json_dos_max_string_length
46 0 3 1 0 as_log_json_dos
47 0 1 1 0 as_log_json_dos_max_string_length
48 0 3 1 0 as_viol_json_dos_profile appfw__(profile1)
49 0 1 1 0 as_viol_json_dos_max_string_length_profile appfw__(profile1)
50 0 3 1 0 as_log_json_dos_profile appfw__(profile1)
51 0 1 1 0 as_log_json_dos_max_string_length_profile appfw__(profile1
<!--NeedCopy-->
```

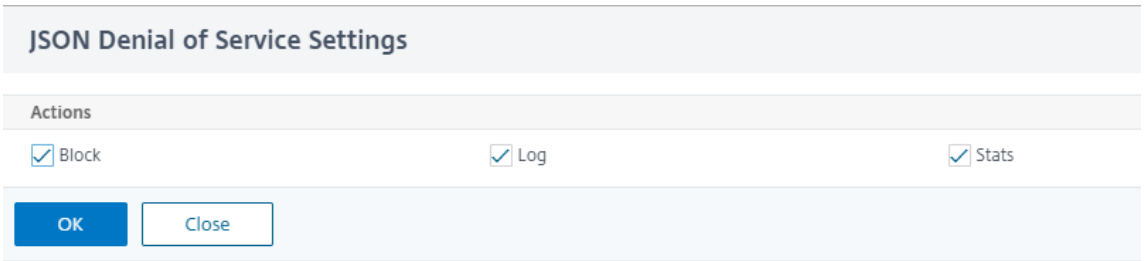
Citrix の **GUI** を使用して **JSON** の **DoS** 保護を構成する

JSON DoS 保護設定を設定するには、以下の手順に従います。

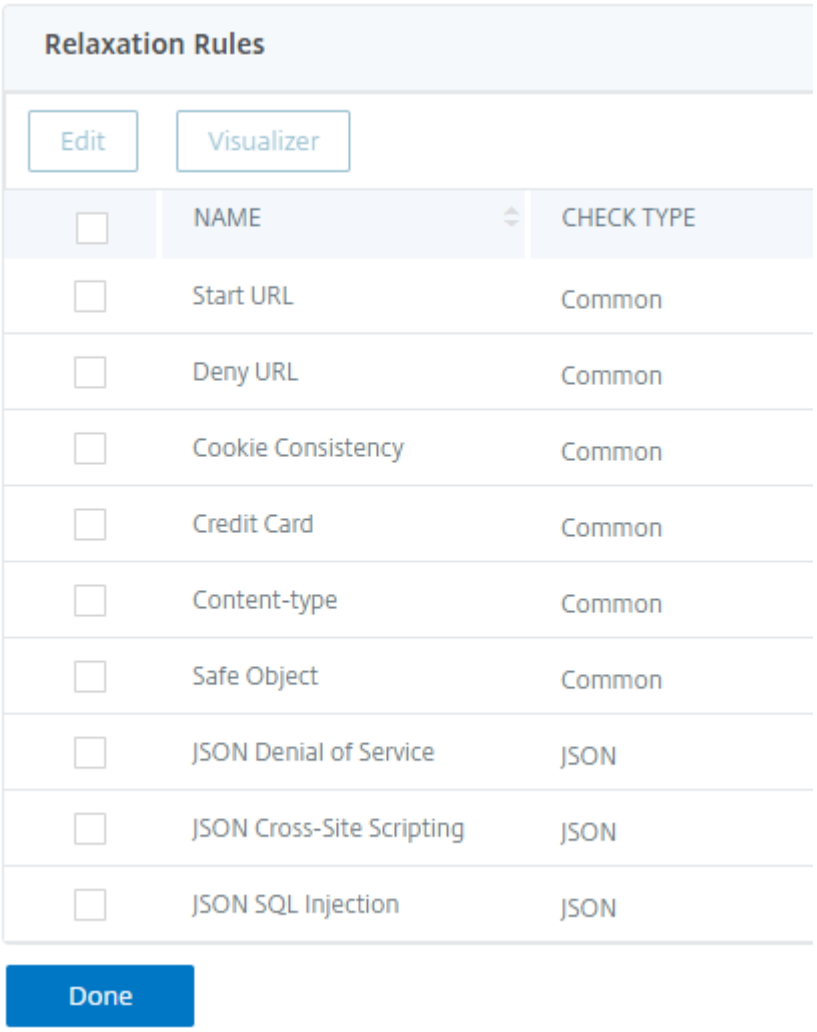
1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] の下の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで、[**JSON** サービス拒否設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。



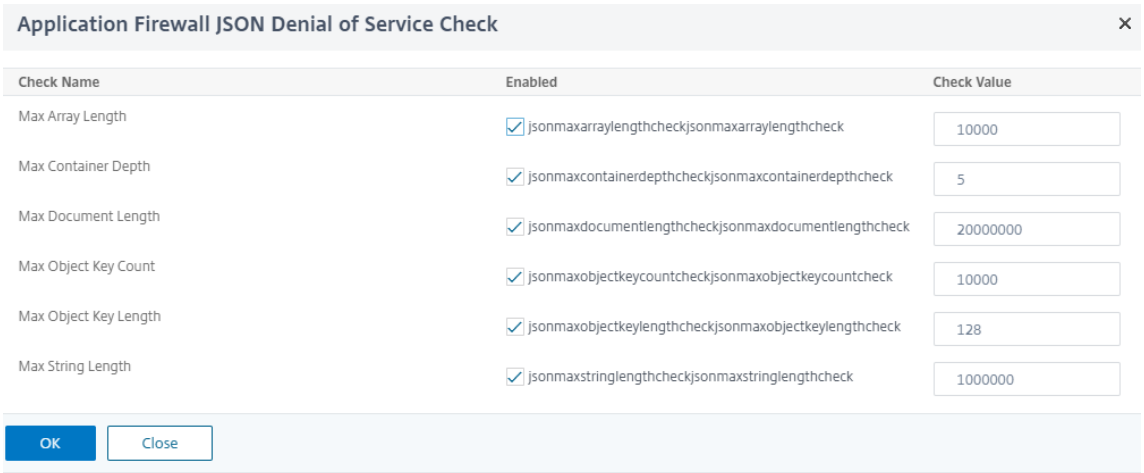
6. 「アクション設定」をクリックして、「**JSON** サービス拒否設定」ページにアクセスします。
7. JSON DoS アクションを選択します。
8. [**OK**] をクリックします。



9. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] の下の [リラクゼーションルール] をクリックします。
10. [緩和ルール] セクションで、[**JSON** サービス拒否設定] を選択し、[編集] をクリックします。



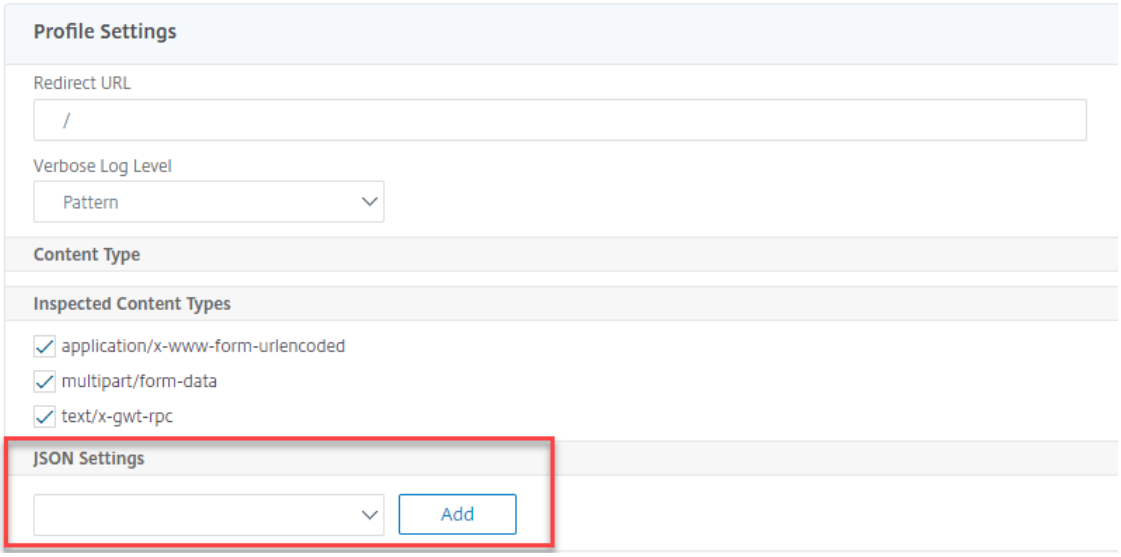
11. アプリケーションファイアウォールの **JSON** サービス拒否チェックで、JSON DoS 検証値を設定します。
12. [**OK**] をクリックします。



13. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] の下の [プロファイル設定] をクリック

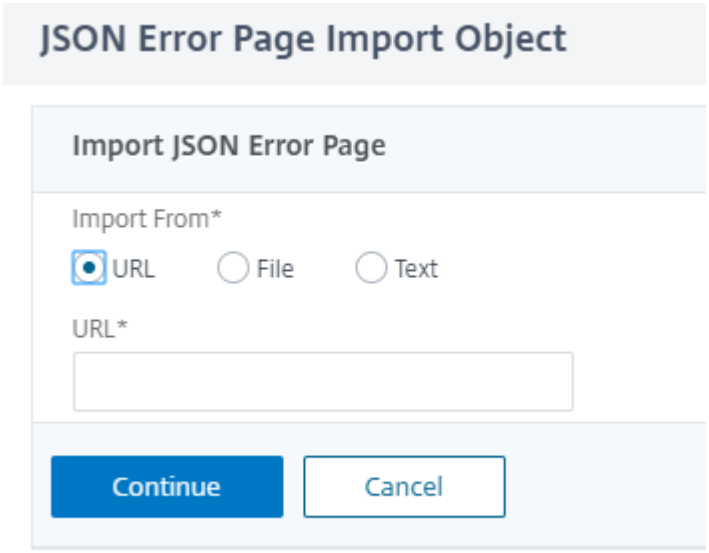
します。

14. 「プロファイル設定」セクションで「**JSON** エラー設定」サブセクションに移動し、**JSON DoS** エラーページを設定します。



15. **JSON** エラーページのオブジェクトのインポートページで、次のパラメータを設定します。

a) インポート元。エラーページをテキスト、ファイル、または URL としてインポートします。
b) URL。ユーザーをエラーページにリダイレクトする URL。
1 ファイル。JSON DoS エラーファイルとしてインポートするファイルを選択します。
c) Text。JSON ファイルの内容を入力します。
d) [続行] をクリックします。
e) File。ファイル名を入力します。
f) ファイルの内容。エラーファイルの内容を追加します。
g) [**OK**] をクリックします。



16. [**OK**] をクリックします。
17. [完了] をクリックします。

JSON SQL インジェクション保護チェック

October 7, 2021

受信 JSON リクエストは、部分的な SQL クエリ文字列またはコード内の不正なコマンドの形で SQL インジェクションを持つことができます。これは、あなたのウェブサーバーの JSON データベースからデータを盗むにつながります。このような要求を受信すると、アプライアンスはお客様のデータを保護するための要求をブロックします。

クライアントが JSON SQL リクエストを Citrix ADC アプライアンスに送信し、JSON パーサーがリクエストペイロードを解析し、SQL インジェクションが観察された場合、アプライアンスは JSON SQL コンテンツに制約を適用します。この制約は、JSON SQL リクエストのサイズ制限を強制します。その結果、JSON SQL インジェクションが検出されると、アプライアンスはアクションを適用し、JSON SQL エラーページに応答します。

JSON SQL インジェクション保護の設定

JSON SQL 保護を構成するには、次の手順を完了する必要があります。

1. アプリケーションのファイアウォールプロファイルを JSON として追加します。
2. JSON SQL インジェクション設定のアプリケーションファイアウォールプロファイルの設定
3. アプリケーションのファイアウォールプロファイルをバインドして JSON SQL アクションを設定します。

タイプ **JSON** のアプリケーションファイアウォールプロファイルを追加します

まず、アプリケーションファイアウォールで JSON Web コンテンツを JSON SQL Injection 攻撃から保護する方法を指定するプロファイルを作成する必要があります。
コマンドプロンプトで入力します。

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> 注:
>
> プロファイルタイプを JSON に設定すると、HTML や XML などの他のチェックは適用されません。

例

```
add appfw profile profile1 -type JSON
```

JSON SQL インジェクションアクションの設定

JSON SQL インジェクション攻撃からアプリケーションを保護するには、1 つ以上の JSON SQL インジェクションアクションを設定する必要があります。
コマンドプロンプトで入力します。

```
set appfw profile <name> - JSONSQLInjectionAction [block] [log] [stats] [
none]
```

SQL インジェクションのアクションは次のとおりです。
Block-このセキュリティー検査に違反する接続をブロックします。
Log - このセキュリティチェックの違反を記録します。
Stats -このセキュリティー検査の統計を生成します。
None -このセキュリティー検査のすべてのアクションを無効にします。

JSON SQL インジェクションタイプの設定

アプリケーションファイアウォールプロファイルで JSON SQL Injection タイプを設定するには、コマンドプロンプトで次のように入力します。

```
set appfw profile <name> - JSONSQLInjectionType <JSONSQLInjectionType>
```

例

```
set appfw profile profile1 -JSONSQLInjectionType SQLKeyword
```

使用可能な SQL インジェクションタイプは次のとおりです。
使用可能な SQL インジェクションタイプ。
sqlSplchar。SQL 特殊文字、

SQL キーワードをチェックします。SQL キーワードをチェックします。
SQLSplCharand キーワードです。見つかった場合は、ブロックとブロックの両方をチェックします。
SQLSplCharOR キーワードです。SQL 特殊文字または spl キーワードが見つかった場合はブロックします。
指定可能な値:SQL 文字、SQL キーワード、SQL スプライン文字またはキーワード、SQLSplcharand キーワードです。

> 注:
> 1 つ以上のアクションを有効にするには、「appfw プロファイルの設定-JSONSQLInjectionAction」と入力し、次に有効にするアクションを入力します。

例

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```

次に、ペイロード、対応するログメッセージ、および統計カウンタの例を示します。

```
Payload:
=======
{

  ”test”: ”data”,
  ”username”: ”waf”,
  ”password”: ”select * from t1;”,
  ”details”: {

    ”surname”: ”test”,
    ”age”: ”23”
   }

 }


Log Message:
============
08/19/2019:08:49:46 GMT pegasus121 Informational 0-PPE-0 : default
   APPFW APPFW_JSON_SQL 6656 0 :  10.217.32.165 18402-PPE0 - profjson
   http://10.217.32.147/test.html SQL Keyword check failed for object
   value(with violation=”select(;)”) starting at offset(52) <blocked>
Counters:
========
   1  441083             1 as_viol_json_sql
   3       0             1 as_log_json_sql
   5       0             1 as_viol_json_sql_profile appfw__(profjson)
```

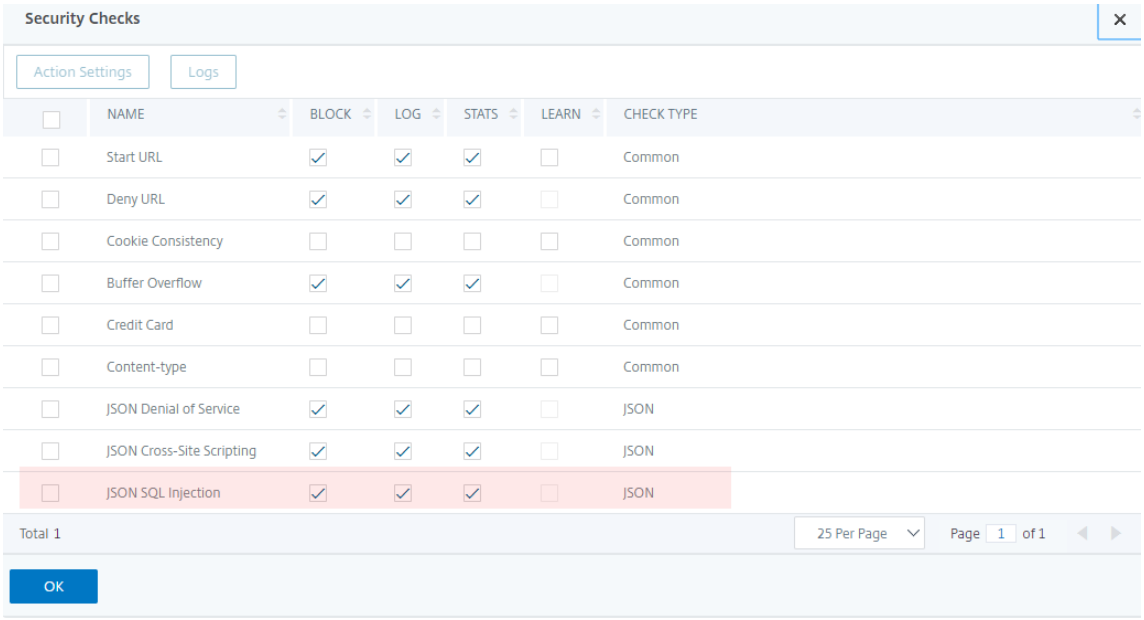


```
25     7        0              1 as_log_json_sql_profile appfw__(profjson)
26 <!--NeedCopy-->
```

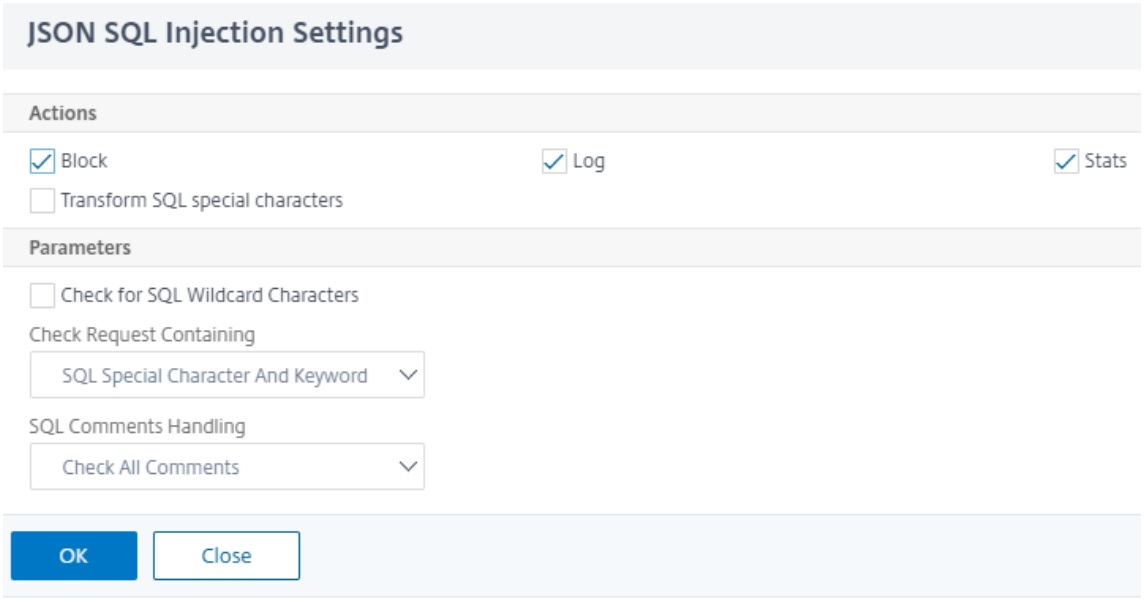
Citrix GUI を使用して **JSON SQL** インジェクション保護を構成する

JSON SQL インジェクション保護設定を設定するには、以下の手順に従います。

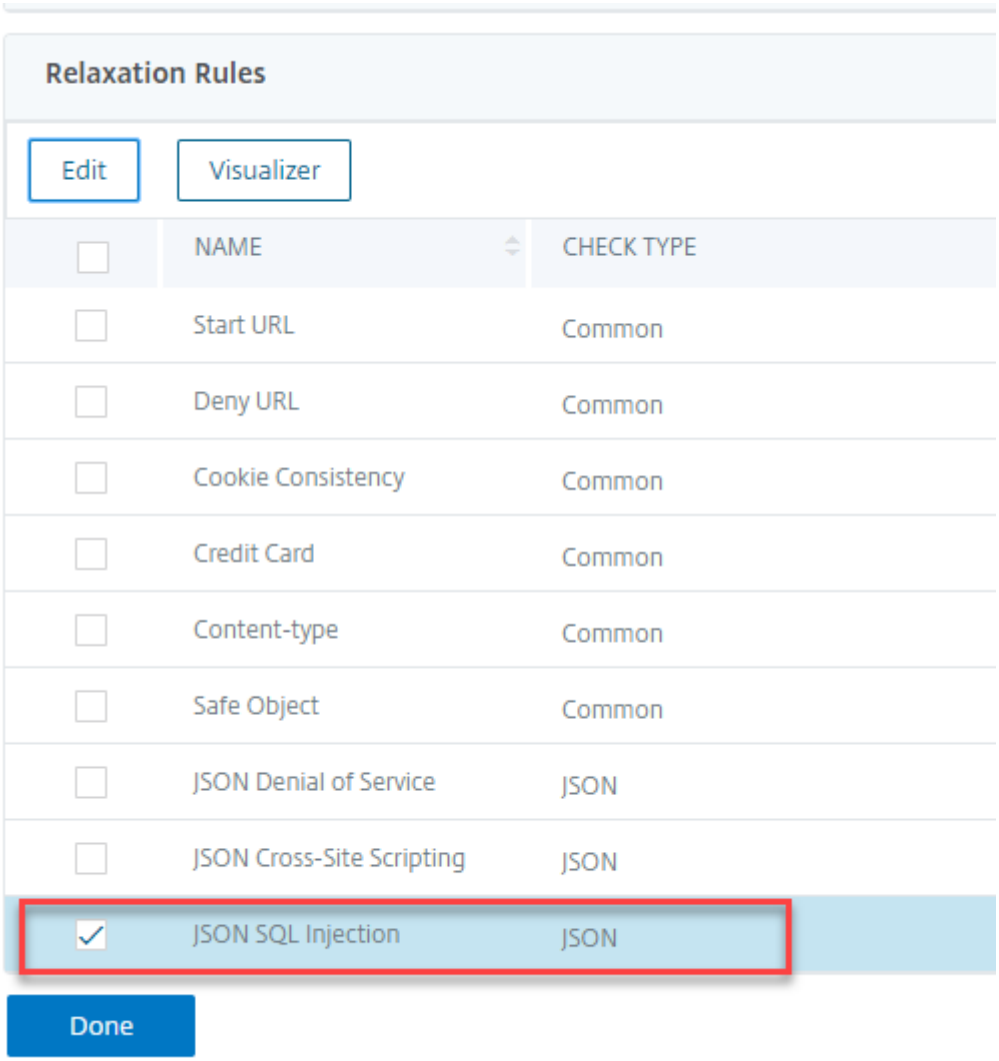
1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. **Citrix Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティーチェック] セクションで、[**JSON SQL** インジェクション設定] に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。



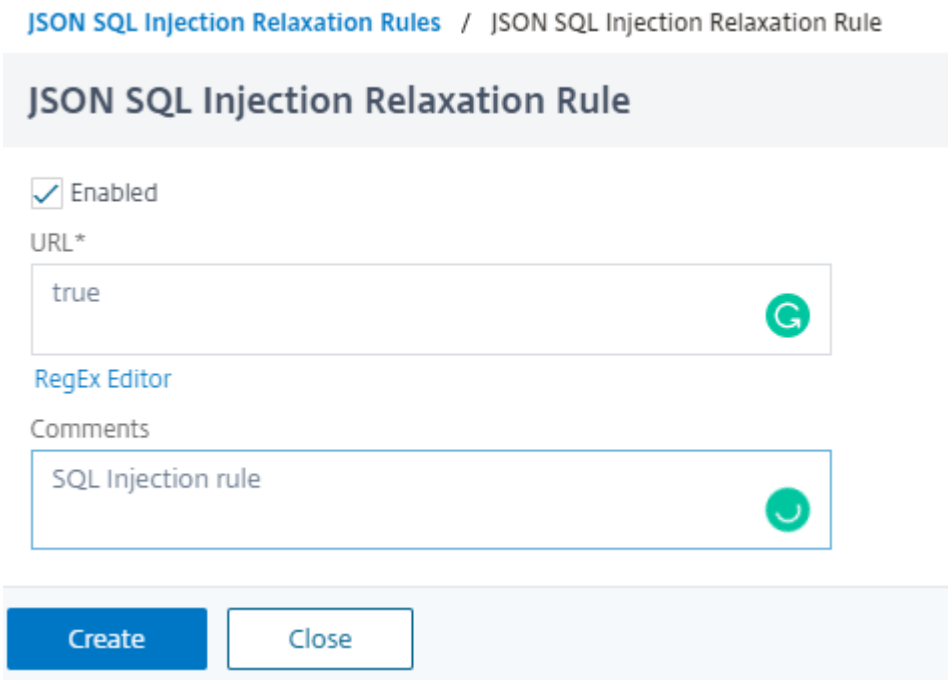
6. [アクション設定] をクリックして、[**JSON SQL** インジェクション設定] ページにアクセスします。
7. **JSON SQL** インジェクションアクションを選択します。
8. [**OK**] をクリックします。



9. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] の下の [リラクゼーションルール] をクリックします。
10. [リラクゼーションルール] セクションで、[**JSON SQL** インジェクション設定] を選択し、[編集] をクリックします。



11. [JSON SQL インジェクション緩和ルール] ページで、リクエストの送信先の URL を入力します。この URL に送信されたすべてのリクエストはブロックされません。
12. ［作成］をクリックします。



JSON クロスサイトスクリプティング保護チェック

October 7, 2021

着信 JSON ペイロードに悪意のあるクロスサイトスクリプティングデータがある場合、WAF はリクエストをブロックします。次の手順では、CLI および GUI インターフェイスを使用してこれを設定する方法について説明します。

JSON クロスサイトスクリプティング保護の設定

JSON クロスサイトスクリプティング保護を構成するには、次の手順を実行する必要があります。

1. アプリケーションのファイアウォールプロファイルを JSON として追加します。
2. JSON クロスサイトスクリプティングアクションを構成して、クロスサイトスクリプティングの悪意のあるペイロードをブロックします

タイプ **JSON** のアプリケーションファイアウォールプロファイルを追加します

最初に、アプリケーションファイアウォールが JSONWeb コンテンツを JSON クロスサイトスクリプティング攻撃から保護する方法を指定するプロファイルを作成する必要があります。

コマンドプロンプトで入力します。

```
add appfw profile <name> -type (HTML | XML | JSON)
```


> 注:
>
> プロファイルタイプを JSON に設定すると、HTML や XML などの他のチェックは適用されません。

例

```
add appfw profile profile1 -type JSON
```

JSON クロスサイトスクリプティング違反のサンプル出力

```
JSONcross-site scriptingAction: block log stats
Payload: {
 ”username”:”<a href=”jAvAsCrIpT:alert(1)”>X</a>”,”password”:”xyz” }


Log message: Aug 19 06:57:33 <local0.info> 10.106.102.21
    08/19/2019:06:57:33 GMT  0-PPE-0 : default APPFW APPFW_JSON_cross-
    site scripting 58 0 :  10.102.1.98 12-PPE0 - profjson http://
    10.106.102.24/ Cross-site script check failed for object value(with
    violation=”Bad URL: jAvAsCrIpT:alert(1)”) starting at offset(12). <
    blocked>

Counters
    1  357000                  1 as_viol_json_xss
    3  0                       1 as_log_json_xss
    5  0                       1 as_viol_json_xss_profile appfw__(
       profjson)
    7  0                       1 as_log_json_xss_profile appfw__(
       profjson)

<!--NeedCopy-->
```

JSON クロスサイトスクリプティングアクションの設定

JSON クロスサイトスクリプティング攻撃からアプリケーションを保護するには、1 つ以上の JSON クロスサイトスクリプティングアクションを構成する必要があります。
コマンドプロンプトで入力します。

```
set appfw profile <name> - JSONcross-site scriptingAction [block] [log] [
stats] [none]
```


例

```
set appfw profile profile1 -JSONcross-site scriptingAction block
```

使用できるクロスサイトスクリプティングのアクションは次のとおりです。
Block-このセキュリティチェックに違反する接続をブロックします。
Log - このセキュリティチェックの違反を記録します。
Stats -このセキュリティー検査の統計を生成します。
None -このセキュリティー検査のすべてのアクションを無効にします。

> 注意:
> 1 つ以上のアクションを有効にするには、「set appfw profile –JSONcross-site scriptingAction」と入力してから、有効にするアクションを入力します。

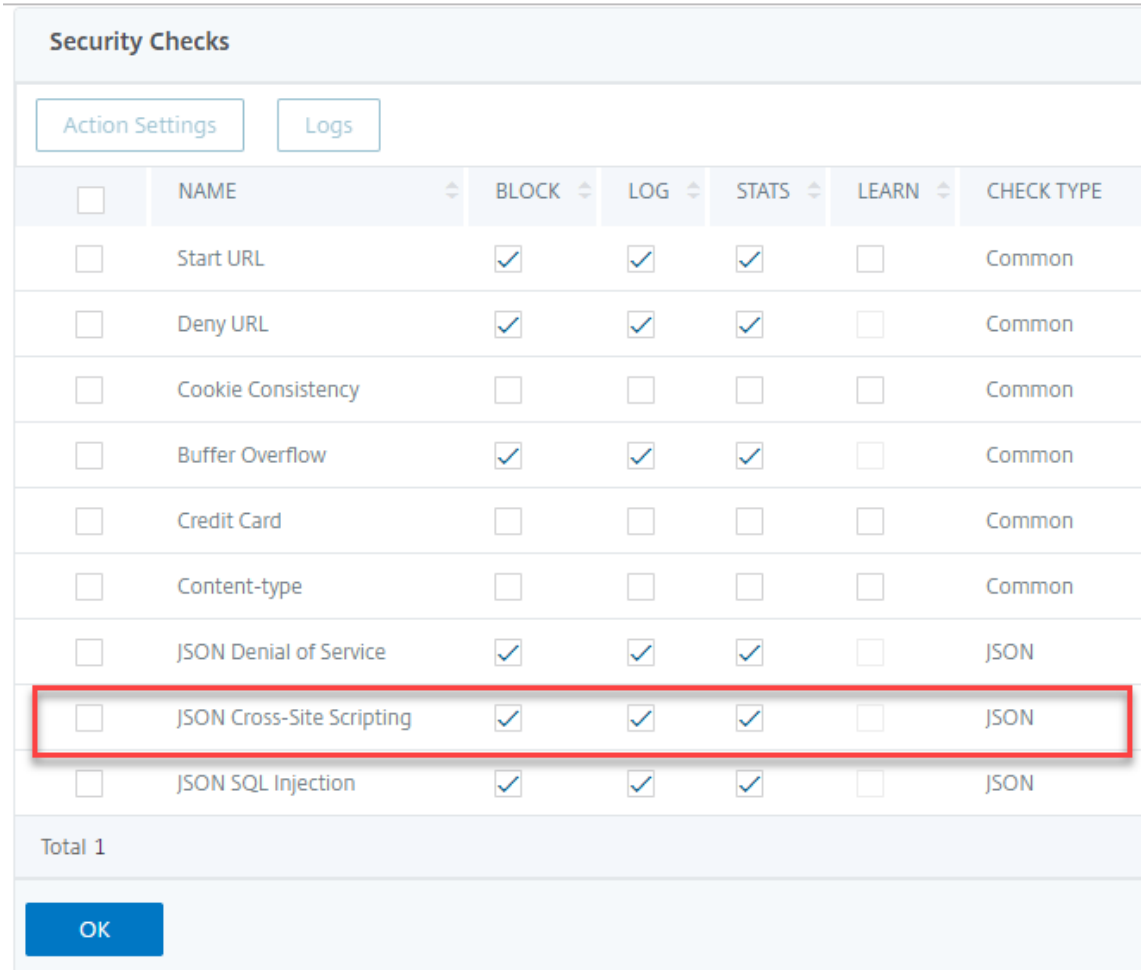
例

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```

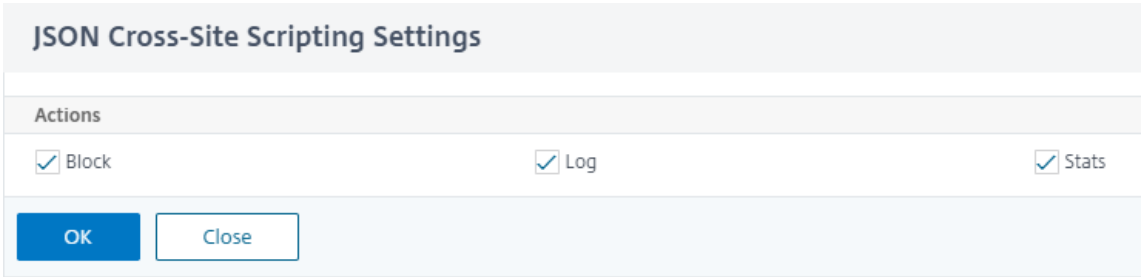
Citrix GUI を使用して、**JSON** クロスサイトスクリプティング（クロスサイトスクリプティング）保護を構成します

以下の手順に従って、クロスサイトスクリプティング（クロスサイトスクリプティング）保護設定を設定します。

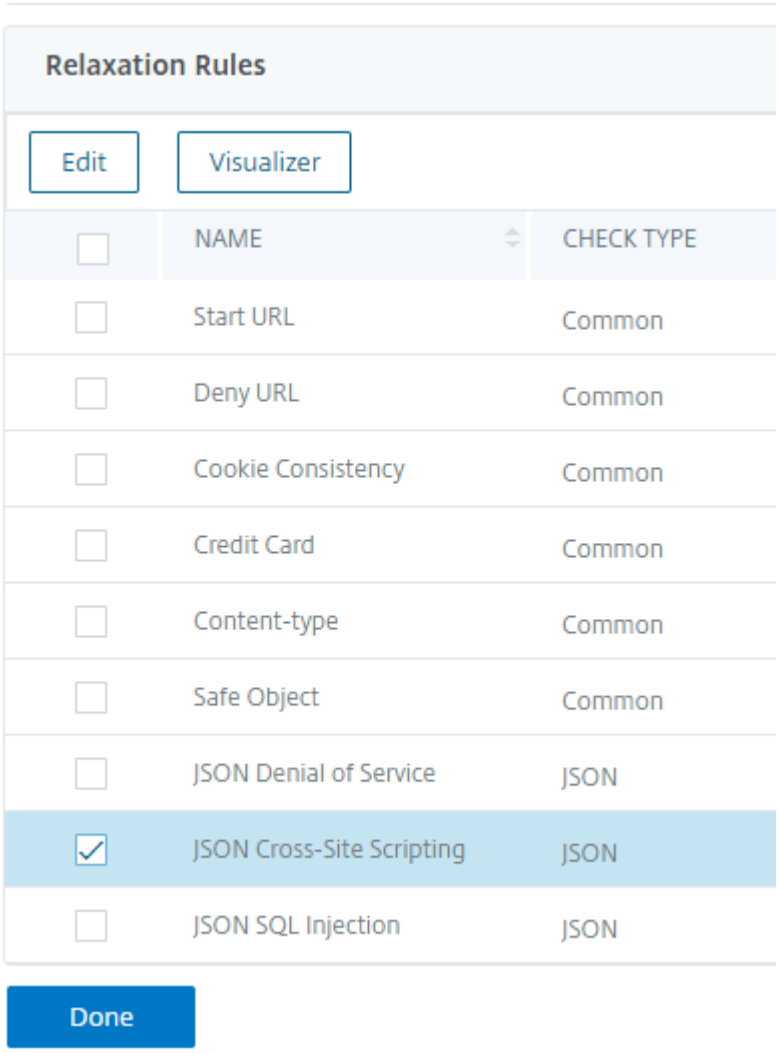
1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. **Citrix Web App Firewall** プロファイルページで、［詳細設定］の［セキュリティチェック］をクリックします。
4. [セキュリティチェック] セクションで、[**JSON** クロスサイトスクリプティング（クロスサイトスクリプティング）] 設定に移動します。
5. チェックボックスの近くにある実行可能アイコンをクリックします。



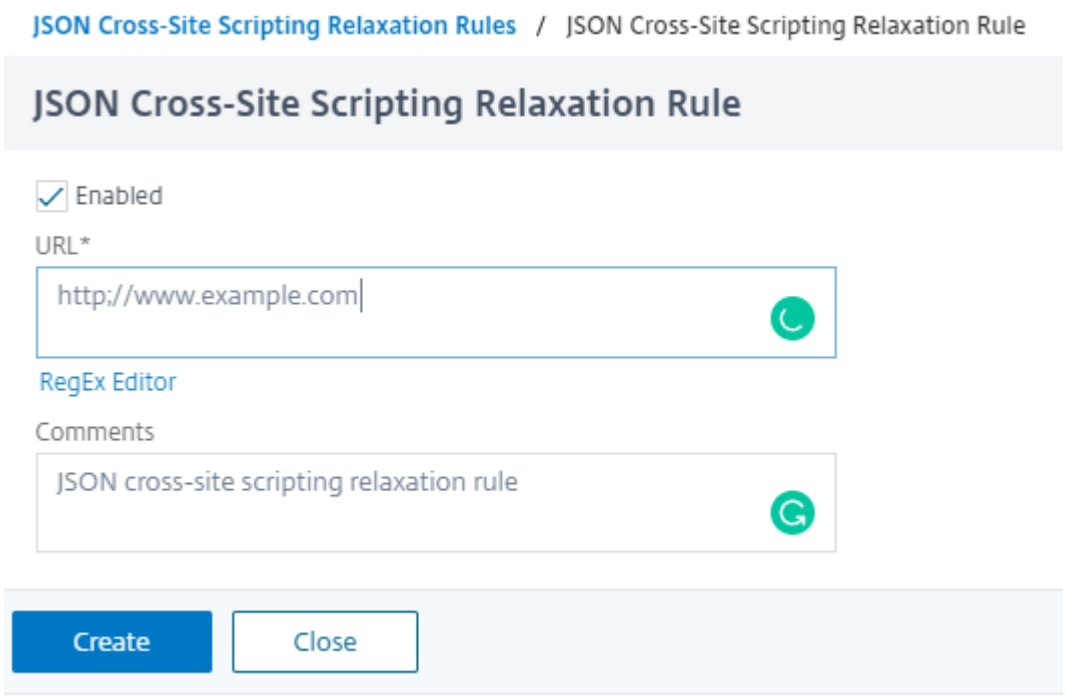
6. [アクション設定] をクリックして、[**JSON** クロスサイトスクリプティング設定] ページにアクセスします。
7. JSON クロスサイトスクリプティングアクションを選択します。
8. [**OK**] をクリックします。



9. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] の下の [リラクゼーションルール] をクリックします。
10. [リラクゼーションルール] セクションで、[JSON クロスサイトスクリプティング設定] を選択し、[編集] をクリックします。



11. [**JSON** クロスサイトスクリプティング緩和規則] ページで、[追加] をクリックして、JSON クロスサイトスクリプティング緩和規則を追加します。
12. 要求の送信先の URL を入力します。この URL に送信されたすべてのリクエストはブロックされません。
13. ［作成］をクリックします。



JSON コマンドインジェクション保護

January 25, 2022

JSON コマンドインジェクションチェックは、受信した JSON トラフィックを調べて、システムセキュリティを侵害したり、システムを変更したりする不正なコマンドがないか調べます。トラフィックを調べる際に、悪意のあるコマンドが検出されると、アプライアンスは要求をブロックするか、設定されたアクションを実行します。

コマンドインジェクション攻撃では、攻撃者は Citrix ADC オペレーティングシステムまたはバックエンドサーバーで不正なコマンドを実行しようとします。これを実現するために、攻撃者は脆弱なアプリケーションを使用してオペレーティングシステムコマンドを注入します。アプライアンスがセキュリティチェックを行わずに要求を転送するだけの場合、バックエンドアプリケーションはインジェクション攻撃に対して脆弱です。したがって、Citrix ADC アプライアンスが安全でないデータをブロックして Web アプリケーションを保護できるように、セキュリティチェックを構成することが非常に重要です。

コマンドインジェクション保護のしくみ

1. 受信した JSON リクエストに対して、WAF はトラフィックにキーワードや特殊文字がないか調べます。JSON リクエストに、拒否されたキーワードまたは特殊文字のいずれにも一致するパターンがない場合、リクエストは許可されます。それ以外の場合、要求は設定されたアクションに基づいてブロック、ドロップ、またはリダイレクトされます。

2. リストからキーワードまたは特殊文字を除外する場合は、特定の条件下でセキュリティチェックをバイパスする緩和ルールを作成できます。
3. ロギングを有効にすると、ログメッセージを生成できます。ログを監視して、正当な要求に対する応答がブロックされているかどうかを判断できます。ログメッセージの数が大幅に増加すると、攻撃を開始しようとしたことを示している可能性があります。
4. また、統計機能を有効にして、違反やログに関する統計データを収集することもできます。stats カウンタの予期しない急増は、アプリケーションが攻撃を受けていることを示している可能性があります。正当な要求がブロックされた場合は、新しい緩和ルールを構成する必要があるか、既存の緩和ルールを変更する必要があるかを再確認するために、構成を再確認する必要があります。

コマンドインジェクションチェックで拒否されたキーワードと特殊文字

JSON コマンドインジェクション攻撃を検出してブロックするために、アプライアンスではデフォルトのシグニチャファイルに一連のパターン（キーワードと特殊文字）が定義されています。コマンドインジェクションの検出中にブロックされるキーワードの一覧を次に示します。

```
<commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
…
</commandinjection>

<!--NeedCopy-->
```

シグネチャファイルに定義されている特殊文字は次のとおりです。

```
| ; & $ > < '\ ! >> ##
```

CLI を使用した JSON コマンドインジェクションチェックの設定

コマンドラインインターフェイスでは、set appfw profile コマンドを使用するか、appfw profile コマンドを追加して JSON コマンドインジェクション設定を構成できます。ブロック、ログ、統計の各アクションを有効にできます。また、ペイロードで検出するキーワードや文字列などのコマンドインジェクションタイプも設定する必要があります。

コマンドプロンプトで入力します。

```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -CMDInjectionType
 <CMDInjectionType>]
```

> 注:
>
> デフォルトでは、コマンドインジェクションアクションは「block log stats」に設定されています。また、デ

> フォルトのコマンドインジェクションタイプはCmdSplCharANDKeyWordとして設定されています。アップグレード後、既存の Web App Firewall プロファイルのアクションは「なし」に設定されます。

例:

```
set appfw profile profile1 -JSONCMDInjectionAction block -JSONCMDInjectionType
 CmdSplChar
```

ここで、使用できる JSON コマンドインジェクションアクションは次のとおりです。

None-コマンドインジェクション保護を無効にします。
Log: セキュリティー検査のコマンドインジェクション違反をログに記録します。
Block-コマンドインジェクションセキュリティー検査に違反するトラフィックをブロックします。
Stats-コマンドインジェクションのセキュリティ違反に関する統計を生成します。

ここで、使用可能な JSON コマンドインジェクションタイプは次のとおりです。

Cmd SplChar -特殊文字をチェックする
CmdKeyWord -コマンドインジェクションをチェックするキーワード
CmdSplCharANDKeyWord -これはデフォルトのアクションです。アクションは特殊文字とコマンドインジェクションをチェックします。キーワードとブロックは、両方が存在する場合に限ります。
CmdSplCharORKeyWord -特殊文字とコマンドインジェクションキーワードとブロックのいずれかが見つかった場合にチェックします。

JSON コマンドインジェクション保護チェックのための緩和ルールの設定

アプリケーションでペイロード内の特定の ELEMENT または ATTRIBUTE に対する JSON コマンドインジェクションインスペクションをバイパスする必要がある場合は、緩和ルールを設定できます。

JSON コマンドのインジェクションインスペクション緩和規則の構文は次のとおりです。

```
bind appfw profile <profile name> -JSONCMDURL <expression> -comment <string
> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED )-state ( ENABLED |
DISABLED )
```

ヘッダーの正規表現の緩和ルールの例

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/hello.html
```

一方、以下では 1.1.1.1 でホストされているすべての URL からのリクエストが緩和されます。

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/*”
```

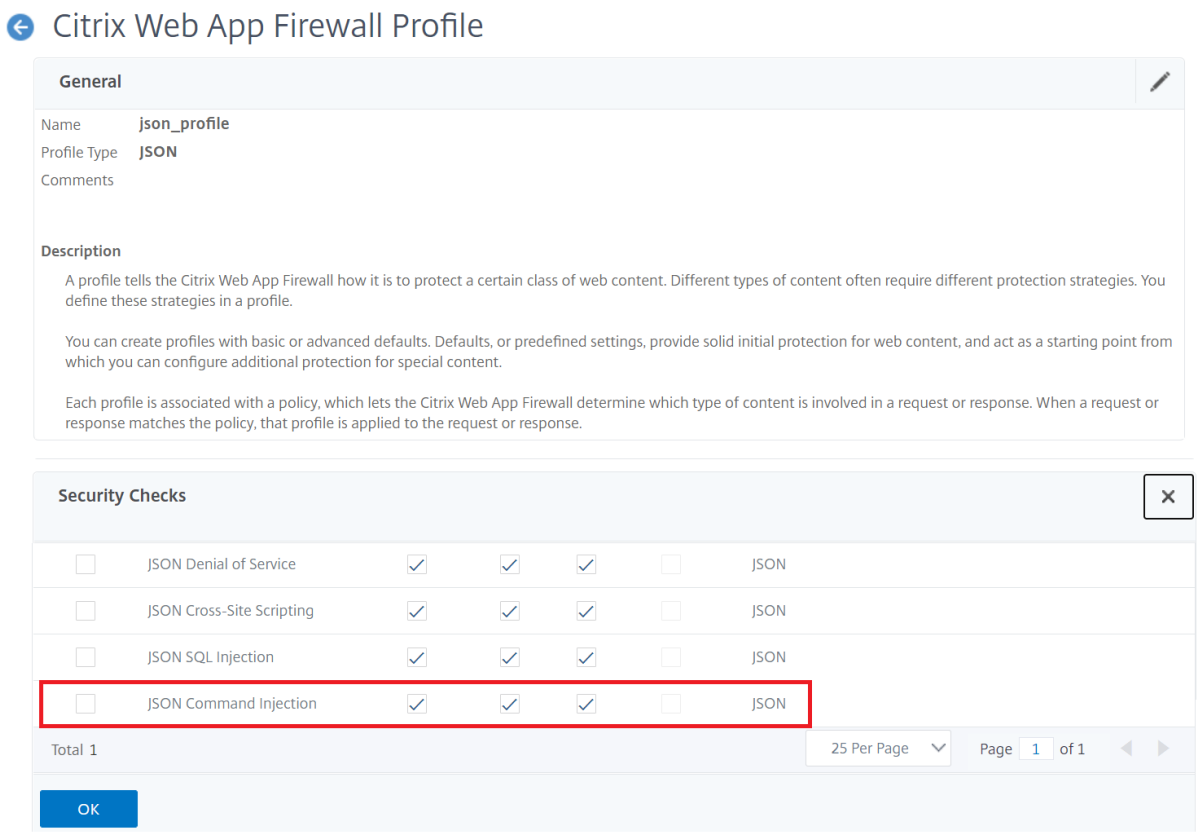
リラクゼーションを削除するには、’unbind’ を使います。

```
unbind appfw profile abc_json -jsoncmDURL  “ http://1.1.1.1/*”
```

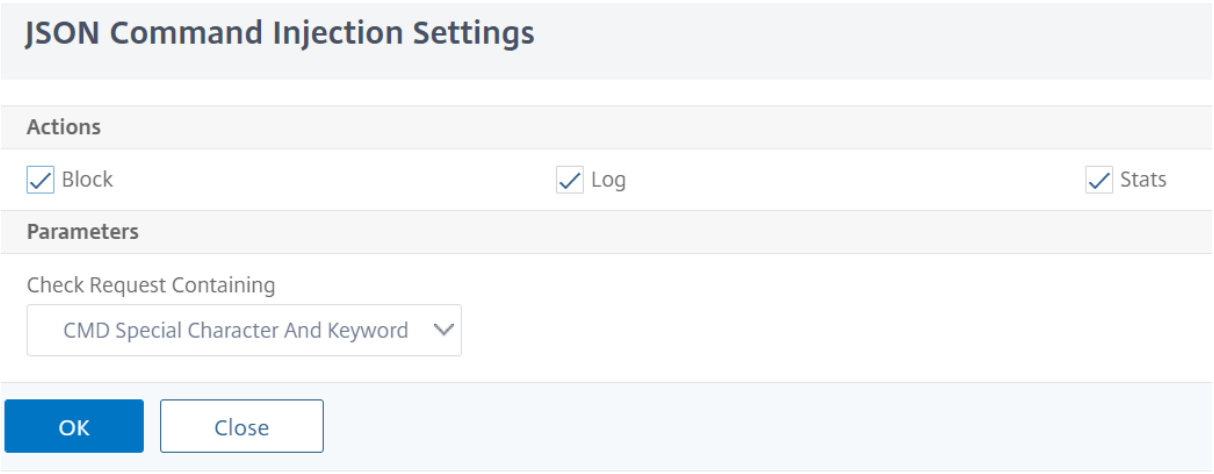

GUI を使用して JSON コマンドインジェクションチェックを設定する

JSON コマンドインジェクションチェックを設定するには、次の手順を実行します。

1. ［セキュリティ］**> Citrix Web App Firewall** とプロファイル］に移動します。
2. [プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. ［**Citrix Web App Firewall** プロファイル］ページで、［詳細設定］セクションに移動し、［セキュリティチェック］をクリックします。



1. [セキュリティチェック] セクションで、[**JSON** コマンドインジェクション] を選択し、[アクションの設定
2. [**JSON** コマンドインジェクションの設定] ページで、次のパラメータを設定します。
   a) アクション。JSON コマンドインジェクションのセキュリティチェックに対して実行するアクションを 1 つ以上選択します。
   b) リクエストに含まれるものをチェックしてください。コマンドインジェクションパターンを選択して、受信リクエストにパターンがあるかどうかを確認します。
3. ［**OK**］をクリックします。



コマンドインジェクショントラフィックおよび違反統計情報の表示

Citrix Web App Firewall Statistics] ページには、セキュリティトラフィックとセキュリティ違反の詳細が表形式またはグラフ形式で表示されます。

コマンドインターフェイスを使用してセキュリティ統計情報を表示するには。

コマンドプロンプトで入力します。

```
stat appfw profile profile1
```

Appfw プロファイルのトラフィック統計	レート (/s)	合計
リクエスト	0	0
要求バイト数	0	0
レスポンス	0	0
送信バイト数	0	0
中止する	0	0
リダイレクト	0	0
長期平均応答時間 (ミリ秒)	–	0
最近の平均応答時間 (ミリ秒)	–	0

HTML/XML/JSON 違反の統計情報	レート (/s)	合計
開始 URL	0	0

HTML/XML/JSON 違反の統計情報	レート (/s)	合計
URL を拒否する	0	0
リファラーヘッダー	0	0
バッファオーバーフロー	0	0
クッキーの整合性	0	0
クッキーハイジャック	0	0
CSRF フォームタグ	0	0
HTML クロスサイトスクリプティング	0	0
HTML SQL インジェクション	0	0
フィールド形式	0	0
フィールドの一貫性	0	0
クレジットカード	0	0
セーフオブジェクト	0	0
シグネチャ違反	0	0
コンテンツの種類	0	0
JSON サービス拒否	0	0
SQL インジェクション	0	0
JSON クロスサイトスクリプティング	0	0
ファイルアップロードの種類	0	0
コンテンツタイプ XML ペイロードを推測	0	0
HTML CMD インジェクション	0	0
XML 形式	0	0
XML サービス拒否 (XDoS)	0	0
XML メッセージ検証	0	0
Web サービスの相互運用性	0	0
XML SQL インジェクション	0	0
XML クロスサイトスクリプティング	0	0

HTML/XML/JSON 違反の統計情報	レート (/s)	合計
XML 添付ファイル	0	0
SOAP フォールト違反	0	0
XML ジェネリック違反	0	0
違反総数	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
URL ログの開始	0	0
URL ログの拒否	0	0
リファラーヘッダーログ	0	0
バッファオーバーフローログ	0	0
クッキー整合性ログ	0	0
クッキーハイジャックのログ	0	0
タグログからの CSRF	0	0
HTML クロスサイトスクリプティングログ	0	0
HTML クロスサイトスクリプティング変換ログ	0	0
HTML SQL インジェクションログ	0	0
HTML SQL 変換ログ	0	0
フィールド形式ログ	0	0
フィールド整合性ログ	0	0
クレジットカード	0	0
クレジットカード変換ログ	0	0
セーフオブジェクトログ	0	0
シグネチャログ	0	0
コンテンツタイプログ	0	0
JSON サービス拒否ログ	0	0
JSON SQL injection logs	0	0

HTML/XML/JSON ログ統計	レート (/s)	合計
JSON クロスサイトスクリプティングログ	0	0
ファイルアップロードタイプログ	0	0
コンテンツタイプ XML ペイロードを推測 L	0	0
JSONCMD インジェクション	0	0
HTML コマンドインジェクションログ	0	0
XML 形式ログ	0	0
XML サービス拒否 (XDoS) ログ	0	0
XML メッセージ検証ログ	0	0
WSI ログ	0	0
XML SQL インジェクションログ	0	0
XML クロスサイトスクリプティングログ	0	0
XML 添付ファイルログ	0	0
SOAP フォールトログ	0	0
XML 汎用ログ	0	0
ログメッセージの総数	0	0

サーバエラーレスポンス統計レート (/s) | 合計 |
|—|–|–|
HTTP クライアントエラー (4xx Resp) | 0 | 0|
HTTP サーバエラー (5xx Resp) | 0 |

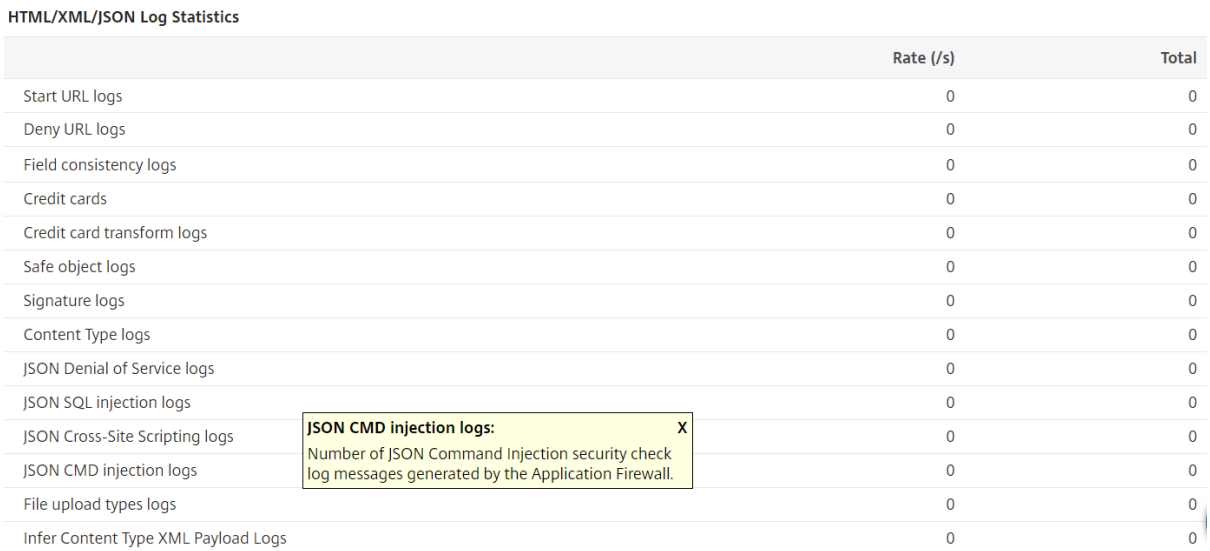
HTML/XML/JSON ログ統計	レート (/s)	合計
JSON コマンド注入ログ	0	0
XML 形式のログ	0	0

Citrix ADC GUI を使用した **JSON** コマンドインジェクションの統計情報の表示

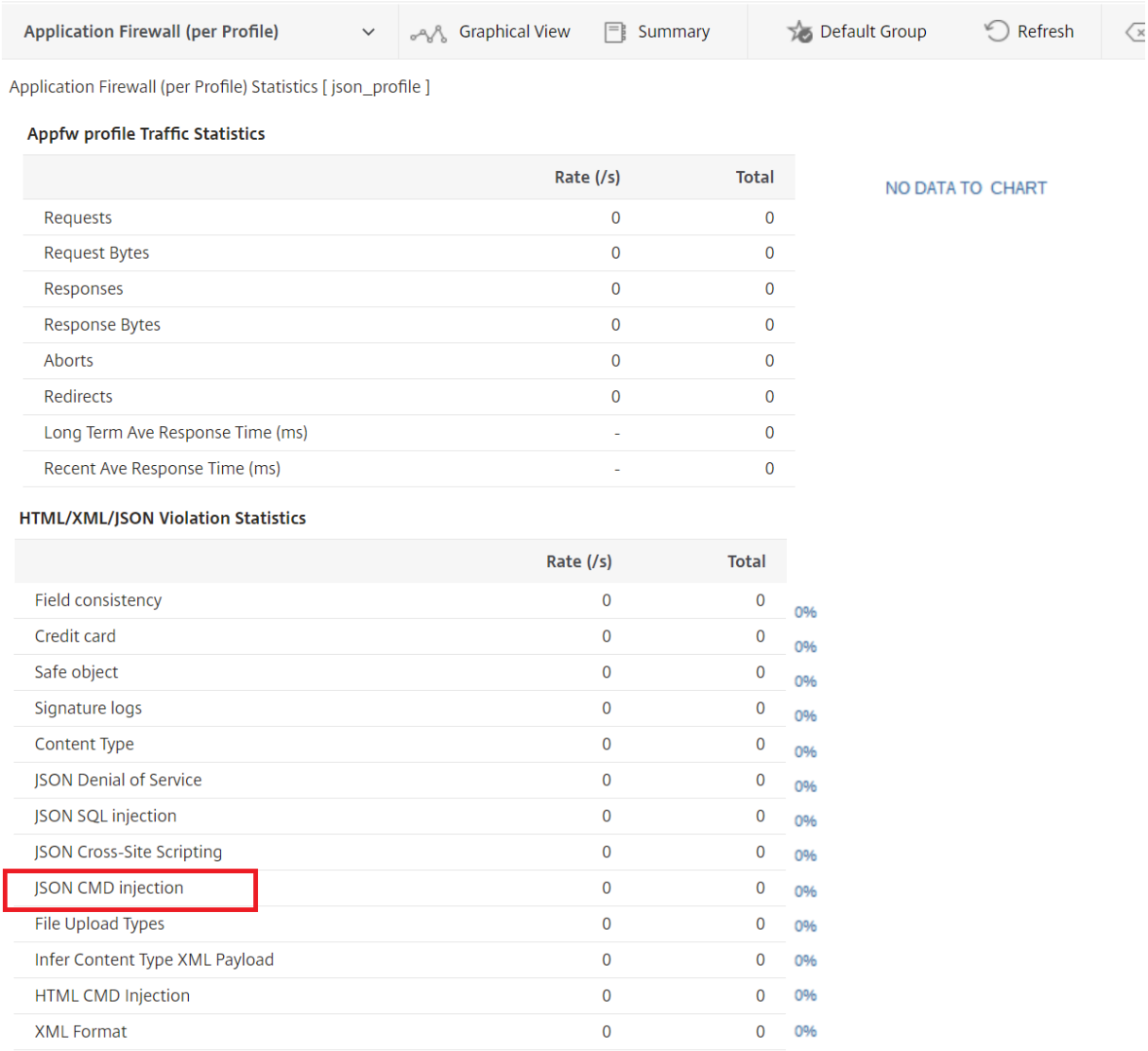
コマンドインジェクションの統計情報を表示するには、次の手順を実行します。

1. ［セキュリティ］ **> Citrix Web App Firewall**］ **>** ［プロファイル］ に移動します。
2. 詳細ウィンドウで Web App Firewall プロファイルを選択し、[統計] をクリックします。
3. **Citrix Web App Firewall** 統計ページには、JSON コマンドインジェクショントラフィックと違反の詳細が表示されます。
4. [表形式] を選択するか、[グラフィカル表示] に切り替えて、データを表形式またはグラフ形式で表示できます。

JSON コマンドのインジェクション



JSON コマンド注入違反統計



コンテンツタイプの管理

October 7, 2021

Web サーバーは、各コンテンツタイプの MIME/タイプ定義を持つコンテンツタイプヘッダーを追加します。Web サーバーは、さまざまな種類のコンテンツを提供します。たとえば、標準 HTML には「テキスト/html」MIME タイプが割り当てられます。JPG 画像には、「画像/JPEG」または「画像/JPG」のコンテンツタイプが割り当てられます。通常の Web サーバーは、割り当てられた MIME /タイプによってコンテンツタイプヘッダーに定義されたコンテンツの異なるタイプを提供することができます。

多くの Web App Firewall フィルタリングルールは、特定のコンテンツタイプをフィルタリングするように設計されています。フィルタ規則は、HTML など、ある種類のコンテンツに適用され、異なる種類のコンテンツ (画像など) を

フィルタリングする場合には不適切であることがよくあります。その結果、Web App Firewall は、要求と応答のコンテンツタイプをフィルタリングする前に判断しようとします。Web サーバーまたはブラウザーが要求または応答に Content-Type ヘッダーを追加しない場合、Web App Firewall はデフォルトのコンテンツタイプを適用し、それに応じてコンテンツをフィルタリングします。

デフォルトのコンテンツタイプは、通常、最も一般的な MIME/タイプ定義を持つ「アプリケーション/オクテットストリーム」です。MIME/タイプは、Web サーバーがサービスを提供する可能性が高いすべてのコンテンツタイプに適しています。しかし、適切なフィルタリングを選択できるようにするために、Web App Firewall に多くの情報を提供しません。保護された Web サーバーが正確なコンテンツタイプヘッダーを追加するように構成されている場合、Web サーバーのプロファイルを作成し、そのサーバーに既定のコンテンツタイプを割り当てることができます。これは、フィルタリングの速度と精度の両方を向上させるために行われます。

また、特定のプロファイルに対して許可された要求コンテンツタイプのリストを構成することもできます。この機能が構成されている場合、Web App Firewall が、許可されているコンテンツタイプのいずれにも一致しない要求をフィルタリングすると、要求がブロックされます。リリース 10.5 から 11.0 へのアップグレード後、デフォルトの許可コンテンツタイプリストにない不明なコンテンツタイプはバインドされません。あなたは、リラックスしたルールに許可したい他のコンテンツタイプを追加することができます。

リクエストは、常に「アプリケーション/x-www-form-urlencoded」、「マルチパート/フォームデータ」、または「テキスト/x-gwt-rpc」のいずれかのタイプでなければなりません。Web App Firewall は、他のコンテンツタイプが指定されているリクエストをブロックします。

> 注
>
> 許可された応答コンテンツタイプのリストには、「application/x-www-form-urlencoded」または「マルチパート/フォームデータ」コンテンツタイプを含めることはできません。

コマンドラインインターフェイスを使用して既定の要求コンテンツタイプを設定するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> -requestContentType <type>`
- `save ns config`

例

次の例では、指定したプロファイルのデフォルトとして「text/html」コンテンツタイプを設定します。

```
set appfw profile profile1 -requestContentType ”text/html”
save ns config
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用してユーザー定義の既定の要求コンテンツタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `unset appfw profile <name> -requestContentType <type>`
- `save ns config`

例

次の例では、指定されたプロファイルのデフォルトのコンテンツタイプ「text/html」を設定解除し、タイプを「application/octet-stream」に戻すことができます。

```
unset appfw profile profile1 -requestContentType ”text/html”
save ns config
<!--NeedCopy-->
```

> 注
>
> 常に最後のコンテンツタイプヘッダーを処理に使用し、バックエンドサーバーが 1 つのコンテンツタイプのみで要求を受信することを保証する場合は、残りのコンテンツタイプヘッダーを削除します。
>
> バイパスできるリクエストをブロックするには、ルールを HTTP.REQ.HEADER (「コンテンツタイプ」) .COUNT.GT (1) ‘として追加し、プロファイルを *appfw_block* として追加します。
>
> 要求が Content-Type ヘッダーなしで受信された場合、または要求に値がない Content-Type ヘッダーがある場合、Web App Firewall は設定された **RequestContentType** 値を適用し、それに応じて要求を処理します。

コマンドラインインターフェイスを使用してデフォルトの応答コンテンツタイプを設定するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> -responseContentType <type>`
- `save ns config`

例

次の例では、指定したプロファイルのデフォルトとして「text/html」コンテンツタイプを設定します。

```
set appfw profile profile1 -responseContentType ”text/html”
save ns config
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用してユーザー定義の既定の応答コンテンツタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `unset appfw profile <name> -responseContentType <type>`
- `save ns config`

例

次の例では、指定されたプロファイルのデフォルトのコンテンツタイプ「text/html」を設定解除し、タイプを「application/octet-stream」に戻すことができます。

```
1 unset appfw profile profile1 -responseContentType ”text/html”
2 save ns config
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、許可されたコンテンツタイプの一覧にコンテンツタイプを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

例

次の例では、指定したプロファイルの許可されたコンテンツタイプリストに「text/shtml」コンテンツタイプを追加します。

```
1 bind appfw profile profile1 -contentType ”text/shtml”
2 save ns config
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、許可されたコンテンツタイプの一覧からコンテンツタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `unbind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

例

次の例では、指定したプロファイルの許可されたコンテンツタイプリストから「text/shtml」コンテンツタイプを削除します。

```
unbind appfw profile profile1 -contentType ”text/shtml”
save ns config
<!--NeedCopy-->
```

URL エンコードされたコンテンツタイプとマルチパートフォームのコンテンツタイプの管理

Citrix ADC Web App Firewall では、フォームに対して URL エンコードされたコンテンツタイプとマルチパートフォームのコンテンツタイプを構成できるようになりました。コンテンツタイプの設定は、XML および JSON リストに似ています。設定に基づいて、Web App Firewall はリクエストを分類し、URL エンコードまたはマルチパートフォームのコンテンツタイプを検査します。

Web App Firewall プロファイルに Urlencoded コンテンツタイプと Multipart-Form コンテンツタイプを構成するにはコマンドプロンプトで次のように入力します。

```
bind appfw profile p2 -contentType <string>
```

例**:**

```
bind appfw profile p2 -contentType UrlencodedFormContentType

bind appfw profile p2 -ContentType appfwmultipartform
```

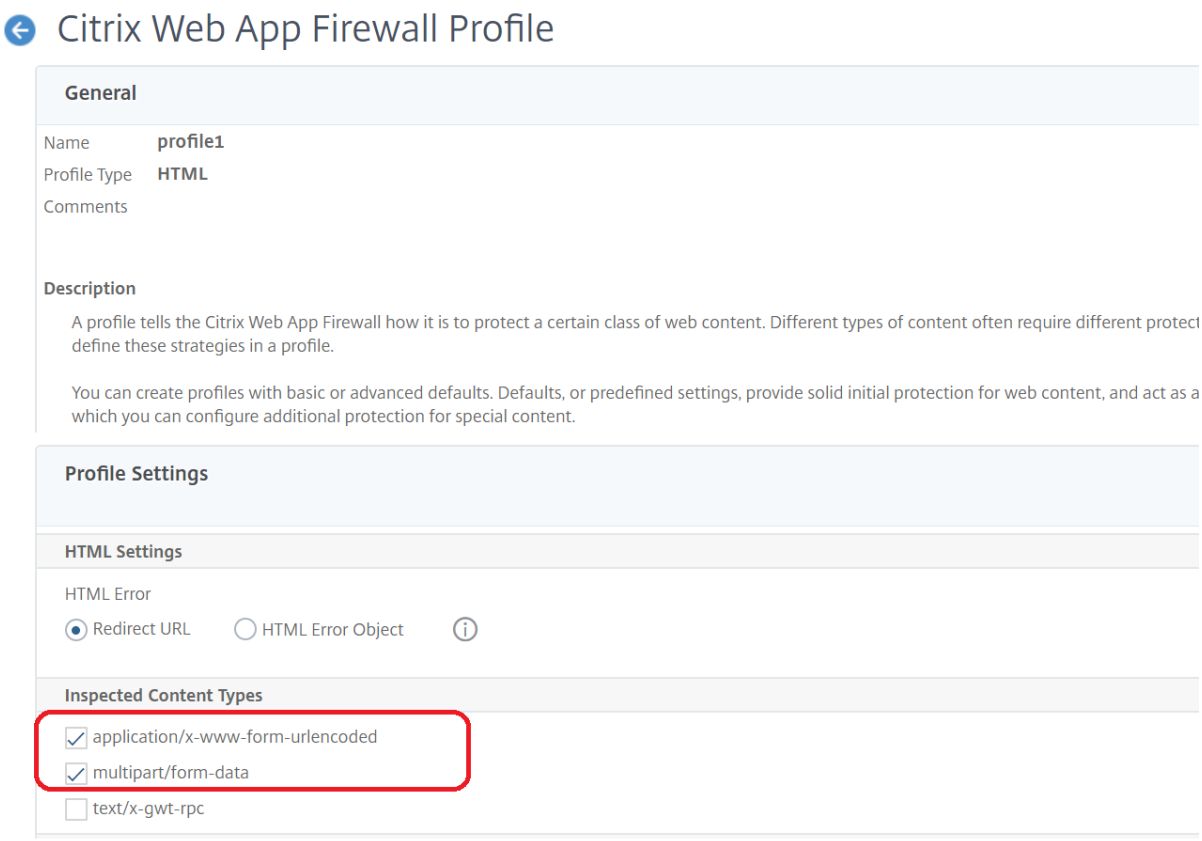
GUI を使用して既定のコンテンツタイプと許可されたコンテンツタイプを管理するには

1. [セキュリティ] > [**Web App Firewall**] > [プロファイル] に移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。**[Web App Firewall** プロファイルの設定] ダイアログボックスが表示されます。
3. [**Web App Firewall** プロファイルの設定] ダイアログボックスで、[設定] タブをクリックします。
4. [設定] タブで、[コンテンツタイプ] 領域まで半分ほど下にスクロールします。
5. [Content Type] 領域で、デフォルトの要求または応答コンテンツタイプを設定します。
   - 既定の要求コンテンツタイプを構成するには、使用するコンテンツタイプの MIME /タイプの定義を [既定の要求] テキストボックスに入力します。
   - 既定の応答コンテンツタイプを構成するには、使用するコンテンツタイプの MIME /タイプの定義を [既定の応答] テキストボックスに入力します。
   - 新しい許可されたコンテンツタイプを作成するには、[追加] をクリックします。**[**許可されたコンテンツタイプの追加] ダイアログボックスが表示されます。
   - 既存の許可されたコンテンツタイプを編集するには、そのコンテンツタイプを選択し、[開く] をクリックします。**[**許可されたコンテンツタイプの変更] ダイアログボックスが表示されます。

6. 許可されたコンテンツタイプを管理するには、[許可されたコンテンツタイプの管理] をクリックします。
7. 新しいコンテンツタイプを追加するか、既存のコンテンツタイプを変更するには、[追加] または [開く] をクリックし、[許可されたコンテンツタイプの追加**]** または **[**許可されたコンテンツタイプの変更] ダイアログボックスで、次の手順を実行します。
   a) 許可されたコンテンツタイプのリストにコンテンツタイプを含めるか、除外するには、[有効] チェックボックスをオンまたはオフにします。
   b) [Content Type] テキストボックスに、追加するコンテンツタイプを説明する正規表現を入力するか、既存のコンテンツタイプの正規表現を変更します。
   コンテンツタイプは、MIME タイプの説明と同じようにフォーマットされます。
   > 注:
   > 許可されたコンテンツタイプリストには、任意の有効な MIME タイプを含めることができます。多くの種類のドキュメントにはアクティブコンテンツが含まれている可能性があり、悪意のあるコンテンツが含まれている可能性があるため、このリストに MIME タイプを追加する場合は注意が必要です。
   c) この特定の MIME タイプを許可されたコンテンツタイプリストに追加する理由を説明する簡単な説明を入力します。
   d) [作成**]** または **[OK]** をクリックして変更を保存します。
8. [閉じる] をクリックして [許可されたコンテンツタイプの管理] ダイアログボックスを閉じ、[設定] タブに戻ります。
9. [**OK**] をクリックして変更を保存します。

Citrix ADC GUI を使用して **URL** エンコードされたコンテンツタイプとマルチパートフォームのコンテンツタイプを管理するには

1. [セキュリティ] > [**Web App Firewall**] > [プロファイル] に移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。
3. [**Web App Firewall** プロファイルの設定] ページで、[詳細 ****** 設定**]** セクションの **[**プロファイル設定]** を選択します。
4. [検査済みコンテンツタイプ**]** セクションで、次のパラメータを設定します。
   a) application/x-www-form-urlencoded. URL エンコードされたコンテンツタイプを検査するには、このチェックボックスをオンにします。
   b) multipart/form-data. マルチパートフォームのコンテンツタイプを検査するチェックを選択します。
5. [**OK**] をクリックします。



プロファイル

October 7, 2021

プロファイルは、特定の種類の Web コンテンツまたは Web サイトの特定の部分を保護するために使用されるセキュリティ設定のコレクションです。プロファイルでは、Web App Firewall が各フィルター（またはチェック）を Web サイトへの要求とそれらからの応答にどのように適用するかを決定します。Web App Firewall は、2 種類のプロファイルをサポートしています。4 つの組み込みプロファイル (デフォルト) で、それ以上の設定を必要としないユーザー定義プロファイルです。

組み込みプロファイル

4 つの WebApp Firewall 組み込みプロファイルは、保護を必要としない、またはユーザーが直接アクセスしてはならないアプリケーションと Web サイトに簡単な保護を提供します。次のプロファイルタイプがあります。

- **APPFW_BYPASS.** すべての WebApp Firewall フィルタリングをスキップし、変更されていないトラフィックを保護されたアプリケーションまたは Web サイト、あるいはクライアントに送信します。

- **APPFW_RESET**。接続をリセットします。クライアントは、指定された開始ページにアクセスしてセッションを再確立する必要があります。
- **APPFW_DROP**。保護されたアプリケーションまたは Web サイトとの間のすべてのトラフィックをドロップし、クライアントにいかなる種類の応答も送信しません。
- **APPFW_BLOCK**。保護されたアプリケーションまたは Web サイトとの間のトラフィックをブロックします。

組み込みプロファイルは、プロファイルを適用するトラフィックを選択し、そのプロファイルとポリシーを関連付けるポリシーを設定することで、ユーザ定義プロファイルの場合とまったく同じように使用します。組み込みのポリシーを構成する必要がないため、特定の種類のトラフィックまたは特定のアプリケーションや Web サイトに送信されるトラフィックを許可またはブロックするための迅速な方法を提供します。

ユーザー定義プロファイル

ユーザー定義プロファイルは、ユーザーによって構築および構成されるプロファイルです。デフォルトのプロファイルとは異なり、保護アプリケーションとの間で送受信されるトラフィックのフィルタリングを使用する前に、ユーザ定義のプロファイルを設定する必要があります。

ユーザー定義プロファイルには、次の 3 つのタイプがあります。

- **HTML**。HTML ベースの Web ページを保護します。
- **XML**。XML ベースの Web サービスと Web サイトを保護します。
- **Web 2.0**。ATOM フィード、ブログ、RSS フィードなどの HTML コンテンツと XML コンテンツを組み合わせた Web 2.0 コンテンツを保護します。

Web App Firewall には多数のセキュリティチェックがあり、これらはすべて有効または無効にでき、各プロファイルでさまざまな方法で構成できます。各プロファイルには、さまざまな種類のコンテンツを処理する方法を制御する設定も多数あります。最後に、すべてのセキュリティチェックを手動で設定するのではなく、学習機能を有効にして設定できます。この機能は、保護された Web サイトへの通常のトラフィックを一定期間監視し、それらの監視を使用して、一部のセキュリティチェックの推奨例外（緩和）の調整済みリスト、およびその他のセキュリティチェックの追加ルールを提供します。

初期構成では、Web App Firewall Wizard を使用するか手動で行うかにかかわらず、通常、1 つの汎用プロファイルを作成して、より具体的なプロファイルでカバーされていない Web サイト上のすべてのコンテンツを保護します。その後、特定のプロファイルをいくつでも作成して、より特殊なコンテンツを保護することができます。

「プロファイル」区画は、次の要素を含むテーブルで構成されます。

名称。アプライアンスで設定されているすべての Web App Firewall プロファイルを表示します。

バインドされた署名。前のカラムにプロファイルにバインドされているシグニチャオブジェクトが表示されます（存在する場合）。

ポリシー。その行の左端の列にプロファイルを起動する Web App Firewall ポリシーを表示します（存在する場合）。

コメント。プロファイルに関連付けられたコメントが、その行の左端の列に表示されます (存在する場合)。

プロファイルタイプ。プロファイルのタイプが表示されます。型は、組み込み型、HTML、XML、および Web 2.0 です。

表の上には、プロファイルに関する情報を作成、設定、削除、表示できるボタンとドロップダウンリストがあります。

- **Add** 新しいプロファイルをリストに追加します。
- 編集。選択したプロファイルを編集します。
- **[削除]**。選択したプロファイルをリストから削除します。
- 統計情報。選択したプロファイルの統計情報を表示します。
- アクション。追加のコマンドを含むドロップダウンリスト。現在、別の Web App Firewall 設定からエクスポートされたプロファイルをインポートできます。

Web App Firewall プロファイルの作成

October 7, 2021

Web App Firewall プロファイルは、コマンドラインを使用する方法と GUI を使用する方法の 2 つの方法のいずれかで作成できます。コマンドラインを使用してプロファイルを作成するには、コマンドラインでオプションを指定する必要があります。このプロセスは、プロファイルの設定と同様であり、いくつかの例外を除いて、2 つのコマンドが同じパラメータを取ります。

GUI を使用してプロファイルを作成するには、2 つのオプションのみを指定する必要があります。* 基本デフォルトまたは詳細デフォルト、プロファイルに含まれるさまざまなセキュリティチェックおよび設定のデフォルト設定を指定し、プロファイルが保護するコンテンツの種類と一致するプロファイルの種類を選択します。**** オプションで、コメントを追加することもできます。プロファイルを作成したら、データペインでプロファイルを選択し、**[**** 編集] をクリックしてプロファイルを構成する必要があります。

学習機能を使用するか、多くの高度な保護を有効にして構成する場合は、高度なデフォルトを選択する必要があります。特に、SQL インジェクションチェック、クロスサイトスクリプティングチェック、Web フォーム攻撃に対する保護を提供するチェック、または Cookie の一貫性チェックのいずれかを構成する場合は、学習機能の使用を計画する必要があります。これらのチェックを設定するときに、保護された Web サイトの適切な例外を含めない限り、正当なトラフィックをブロックする可能性があります。広すぎる例外を作成せずにすべての例外を予測することは困難です。学習機能を使用すると、このタスクがはるかに簡単になります。それ以外の場合、基本的なデフォルト設定は迅速で、Web アプリケーションが必要とする保護を提供する必要があります。

次の 3 つのプロファイルタイプがあります。

- **HTML**。標準的な HTML ベースの Web サイトを保護します。
- **XML**。XML ベースの Web サービスおよび Web サイトを保護します。
- **Web 2.0 (HTML XML)**。ATOM フィード、ブログ、RSS フィードなど、HTML 要素と XML 要素の両方を含む Web サイトを保護します。

また、プロフィールに付けることができる名前にはいくつかの制限があります。プロファイル名は、NetScaler アプライアンスのどの機能でも他のプロファイルまたはアクションに割り当てられた名前と同じにすることはできません。

特定のアクションまたはプロファイル名は、組み込みのアクションまたはプロファイルに割り当てられ、ユーザープロファイルには使用できません。許可されていない名前の完全なリストは、Web App Firewall プロファイルの補足情報を参照してください。アクションまたはプロファイルに既に使用されている名前のプロファイルを作成しようとすると、エラーメッセージが表示され、プロファイルは作成されません。

コマンドラインインターフェイスを使用して **Web App Firewall** プロファイルを作成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw profile <name> [-defaults ( **basic** | **advanced** )]`
- `set appfw profile <name> -type ( **HTML** | **XML** | **HTML XML** )`
- `set appfw profile <name> -comment "<comment>"`
- `save ns config`

例

次の例では、基本デフォルトで pr-basic という名前のプロファイルを追加し、プロファイルタイプに HTML を割り当てます。これは、HTML Web サイトを保護するためのプロファイルの適切な初期設定です。

```
1 add appfw profile pr-basic -defaults basic -comment "Simple profile for
      websites."
2 set appfw profile pr-basic -type HTML
3 save ns config
4 <!--NeedCopy-->
```

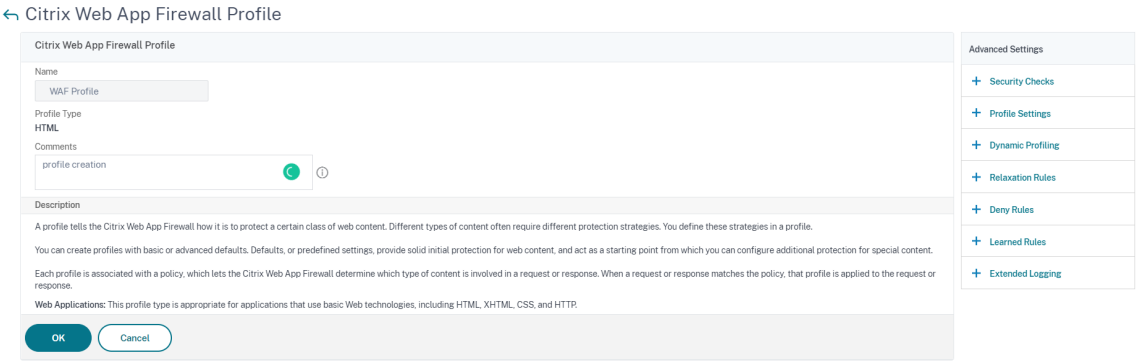
GUI を使用して **Web App Firewall** プロファイルを作成するには

Web App Firewall プロファイルを作成するには、以下の手順を実行します。

1. ［セキュリティ］**> Citrix Web App Firewall**］**>** ［プロファイル］に移動します。
2. 詳細ペインで、［**Add**］をクリックします。
3. **[Web App Firewall** プロファイルの作成] ページで、次の基本パラメータを設定します。
   a) 名前
   b) プロファイルの種類
   c) コメント
   d) デフォルト
   e) 説明
4. 「**OK**」をクリックします。

5. [詳細設定] セクションで、次の設定を行います。

   a) セキュリティチェック
   b) プロファイル設定
   c) 動的プロファイリング
   d) リラクゼーションルール
   e) ルールを拒否する
   f) 学習ルール
   g) 拡張ロギング

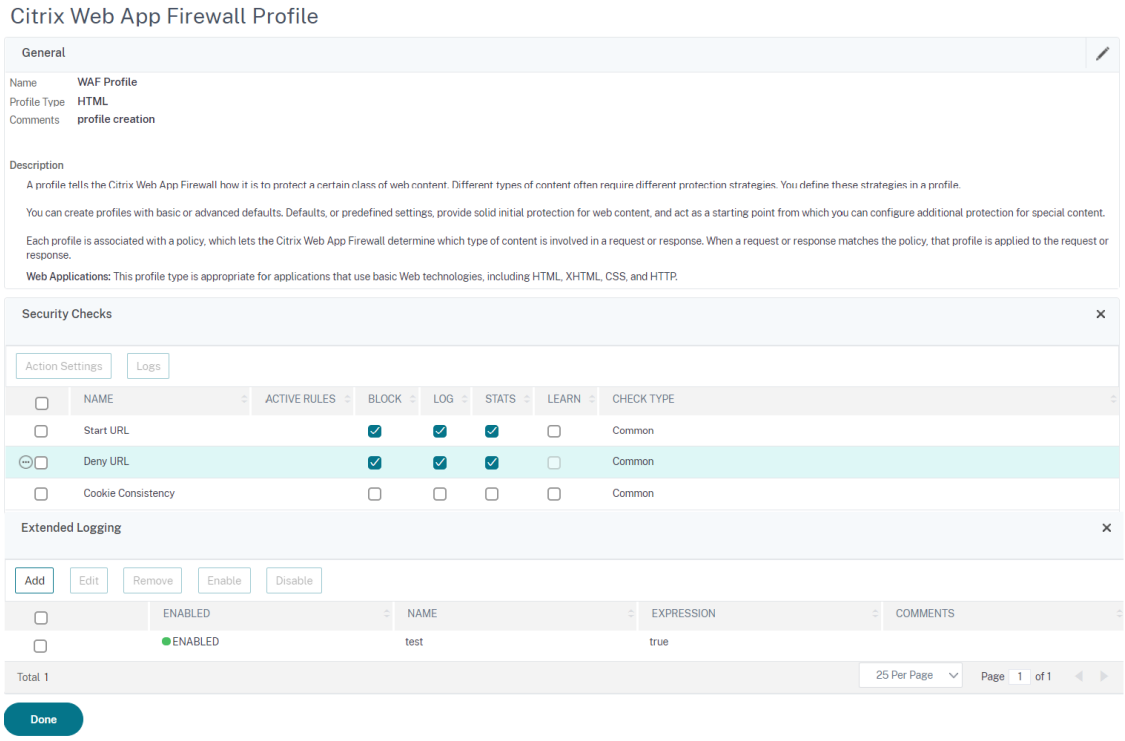


6. [セキュリティチェック] セクションで、セキュリティ保護を選択し、[アクションの設定] をクリックします。
7. [セキュリティチェック] ページで、パラメータを設定します。

   > **注:**
   > アクティブルール設定は、署名ルールを許可または拒否する **HTML SQL** インジェクションチェックでのみ使用できます。

8. [**OK**] をクリックして閉じます。
9. [プロファイル設定] セクションで、プロファイルパラメータを設定します。詳細については、「Web App Firewall プロファイル設定の構成 」トピックを参照してください。
10. [動的プロファイリング] セクションで、セキュリティチェックを選択して動的プロファイル設定を追加します。詳細については、「動的プロファイル 」トピックを参照してください。
11. [緩和ルール] セクションで、[編集] をクリックして、セキュリティチェックの緩和ルールを追加します。詳細については、緩和ルールを参照してください 。
12. [拒否ルール] セクションで、HTML SQL インジェクションチェックの拒否ルールを追加します。詳細については、「HTML 拒否ルール 」トピックを参照してください。
13. [学習ルール] セクションで、学習設定を設定します。詳細については、「Web App Firewall 学習 」トピックを参照してください。
14. [拡張ログ] セクションで、[追加] をクリックして機密データをマスキングします。詳細については、「拡張ロギングのトピック 」を参照してください。

15. [完了] をクリックし、[閉じる] をクリックします。



HTTP RFC コンプライアンスを強制する

June 24, 2022

Citrix Web App Firewall は、HTTP RFC 準拠の着信トラフィックを検査し、デフォルトで RFC 違反のある要求をドロップします。ただし、特定のシナリオがあり、アプライアンスが非 RFC コンプライアンス要求をバイパスまたはブロックする必要がある場合があります。このような場合、グローバルレベルまたはプロファイルレベルでこのような要求をバイパスまたはブロックするようにアプライアンスを設定できます。

グローバルレベルで非 RFC 準拠要求をブロックまたはバイパスする

HTTP モジュールは、リクエストが不完全で、WAF で処理できないリクエストが無効であると識別します。たとえば、受信 HTTP リクエストにホストヘッダーがないとします。このような無効な要求をブロックまたはバイパスするには、アプリケーションファイアウォールのグローバル設定で`malformedReqAction`オプションを構成する必要があります。

‘malformedReqAction’ パラメーターは、受信したリクエストのコンテンツ長が無効か、チャンクされたリクエストが無効か、HTTP バージョンがないか、ヘッダーが不完全かどうかを検証します。

> 注:
>
> malformedReqAction パラメータでブロックオプションを無効にすると、アプライアンスはすべての非 RFC コンプライアンス要求に対してアプリケーションファイアウォール処理全体をバイパスし、要求を次のモジュールに転送します。

コマンドラインインターフェイスを使用して無効な非 **RFC** 苦情 **HTTP** 要求をブロックまたはバイパスするには

無効な要求をブロックまたはバイパスするには、次のコマンドを入力します。

```
set appfw settings -malformedreqaction <action>
```

例:

```
set appfw settings -malformedReqAction block
```

不正なリクエストアクション設定を表示するには

不正なリクエストアクション設定を表示するには、次のコマンドを入力します。

```
show appfw settings
```

出力:

```
1 DefaultProfile:  APPFW_BYPASS UndefAction:  APPFW_BLOCK SessionTimeout:
      900     LearnRateLimit:  400     SessionLifetime:  0
    SessionCookieName:  citrix_ns_id ImportSizeLimit:  134217728
    SignatureAutoUpdate:  OFF SignatureUrl:"https://s3.amazonaws.com/
    NSAppFwSignatures/SignaturesMapping.xml" CookiePostEncryptPrefix:
    ENC GeoLocationLogging:  OFF CEFLogging:  OFF       EntityDecoding:
     OFF     UseConfigurableSecretKey:  OFF SessionLimit:  100000
    MalformedReqAction:  block log stats
2  Done
3 <!--NeedCopy-->
```

Citrix ADC GUI を使用して無効な非 **RFC** 苦情 **HTTP** リクエストをブロックまたはバイパスするには

1. [セキュリティ] **>** [**Citrix Web App Firewall**] に移動します。
2. [**Citrix Web App Firewall**] ページで、[設定] の下の [エンジン設定の変更] をクリックします。
3. [**Citrix Web App Firewall** 設定の構成] ページで、[不正な形式の要求をログに記録する] オプションを選択します。[ブロック]、[ログ]、または [統計] を選択します。
4. [**OK**] をクリックして閉じます。

> 注:
>
> ブロックアクションの選択を解除するか、不正な形式のリクエストアクションを選択しなかった場合、アプライアンスはユーザーを脅かすことなくリクエストをバイパスします。

プロファイルレベルで非 **RFC** 準拠要求をブロックまたはバイパスする

他の非 RFC 準拠要求は、プロファイルレベルでブロックまたはバイパスするように設定できます。RFC プロファイルは、ブロックモードまたはバイパスモードのいずれかで設定する必要があります。この設定を行うと、Web App Firewall プロファイルに一致する無効なトラフィックはバイパスされるか、それに応じてブロックされます。RFC プロファイルは、次のセキュリティチェックを検証します。

- 無効な GWT-RPC リクエスト
- 無効なコンテンツタイプヘッダー
- 無効なマルチパートリクエスト
- 無効な JSON リクエスト
- 重複する Cookie 名と値のペアチェック

> 注:
>
> RFC プロファイルを「バイパス」モードで設定する場合は、「**HTML** クロスサイトスクリプティング設定」および「HTML SQL インジェクション設定 **」セクションで変換オプションを無効にする必要があります。バイパスモードでRFCプロファイルを有効にして設定すると、アプライアンスには「クロスサイトスクリプトの変換」と「SQL 特殊文字の変換」の両方が現在オンになっているという警告メッセージが表示されます。APPFW_RFC_BYPASSと一緒に使用する場合は、オフにすることをお勧めします。
>
> 重要:
>
> また、アプライアンスには、「Appfw セキュリティチェックは、このプロファイルが設定されている場合、RFC チェックに違反する要求には適用されない可能性があります。RFC 違反を含むリクエストが部分的に変換される可能性があるため、変換設定を有効にすることは推奨されません。」

コマンドラインインターフェイスを使用して **Web App Firewall** プロファイルで **RFC** プロファイルを構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
set appfw profile <profile_name> -rfcprofile <rfcprofile_name
```

例

```
set appfw profile P1 -rfcprofile APPFW_RFC_BLOCK
```


> 注:
>
> デフォルトでは、RFC プロファイルはブロックモードの Web App Firewall プロファイルにバインドされます。

GUI を使用して Web App Firewall プロファイルで RFC プロファイルを構成するには

1. [セキュリティ] **> Citrix Web App Firewall**] **>** [プロファイル] に移動します。
2. [プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. **Web App Firewall** プロファイルページで、[詳細設定] セクションの **[**プロファイル設定**]** をクリックします。
4. [**HTML** 設定] セクションで、RFC プロファイルを APPFW_RFC_BYPASS mode に設定します。
   警告メッセージが表示されます。「Appfw Security Check enabled は、このプロファイルが設定されている場合、RFC チェックに違反する要求には適用できない可能性があります。RFC 違反を含むリクエストが部分的に変換される可能性があるため、変換設定を有効にすることはお勧めしません。」

Web App Firewall プロファイルの構成

February 21, 2022

ユーザー定義の Web App Firewall プロファイルを構成するには、まずセキュリティチェックを構成します。セキュリティチェックは、Web App Firewall ウィザードでは「ディーププロテクション **」または「高度な保護」と呼ばれます。一部のチェックでは、使用するためには設定が必要です。安全ではあるが範囲が限定された既定の設定があるものもあります。Web サイトでは、特定のセキュリティチェックのより多くの機能を利用する別の設定が必要か、その恩恵を受ける可能性があります。

セキュリティチェックを構成したら、1 つのセキュリティチェックではなく、Web App Firewall 機能の動作を制御するその他の設定も構成できます。ほとんどの Web サイトを保護するには既定の設定で十分ですが、保護された Web サイトに適しているかどうかを確認するには、これらを確認する必要があります。

> 注:
>
> プロファイル名の長さとすべてのインポートオブジェクト名の長さは、最大 127 文字まで設定できます。

Web App Firewall のセキュリティチェックの詳細については、「高度な保護」を参照してください。

コマンドラインを使用して Web App Firewall プロファイルを構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> <arg1> [<arg2> ...]`

  各項目の意味は次の通りです:

- <arg1> = パラメータと関連するオプション。
- <arg2> = 2 番目のパラメータと関連するオプション。
- ... = 追加のパラメータとオプション。

特定のセキュリティチェックを設定するときに使用するパラメータの詳細については、「高度な保護」を参照してください。

- save ns config

例

次の例は、pr-basic という名前のプロファイルで HTML SQL インジェクションおよび HTML クロスサイトスクリプティングチェックのブロックを有効にする方法を示しています。このコマンドは、プロファイルに他の変更を加えずにこれらのアクションをブロックできるようにします。

```
1 set appfw profile pr-basic -crossSiteScriptingAction block -
     SQLInjectionAction block
2 <!--NeedCopy-->
```

緩和ルールを **Web App Firewall** プロファイルにバインドする

Web App Firewall が違反を検出すると、ユーザーは緩和ルールによって適用されたアクションをバイパスできます。緩和ルールは、検出されたセキュリティ違反に適用される例外です。たとえば、開始 URL 緩和ルールは、強制的なブラウジングから保護します。ハッカーによって悪用される既知の Web サーバーの脆弱性は、デフォルトの URL の拒否ルールのセットを有効にすることで検出およびブロックできます。バッファオーバーフロー、SQL、クロスサイトスクリプティングなど、一般的に起動される攻撃も簡単に検出できます。

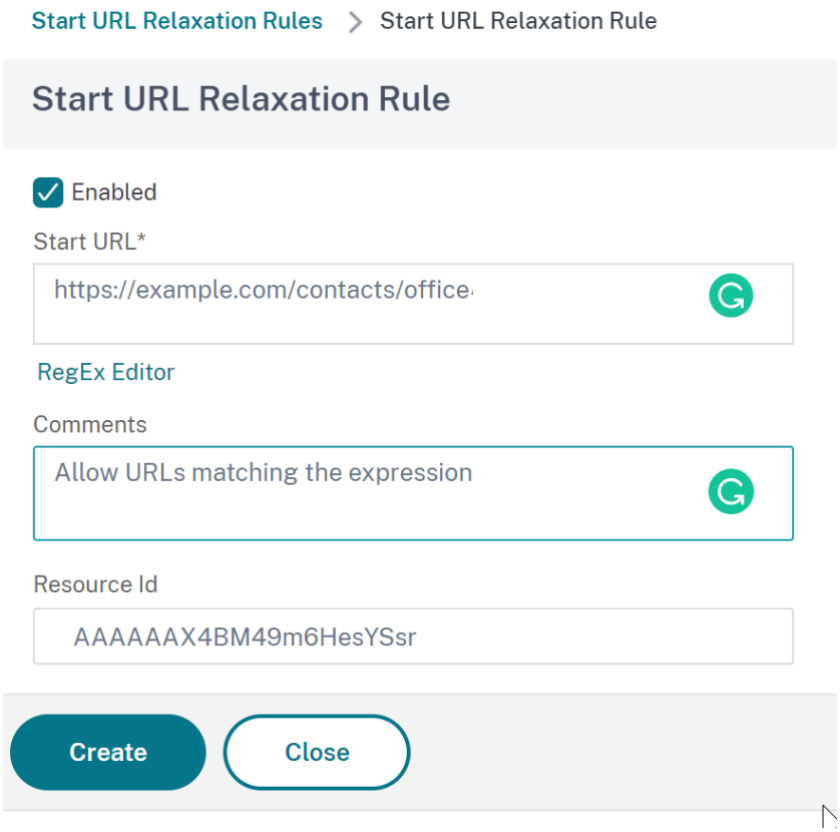
CLI を使用してセキュリティ免除ルールまたは緩和ルールをバインドするには

コマンドプロンプトで入力します。

```
1 bind appfw profile <name> ((-startURL <expression> [-resourceId  <
     string>]) | -denyURL <expression> | (-fieldConsistency <string>   <
     formActionURL> [-isRegex ( REGEX | NOTREGEX )]) | (-
     cookieConsistency <string> [-isRegex ( REGEX | NOTREGEX )]) | (-
     SQLInjection <string> <formActionURL> [-isRegex ( REGEX | NOTREGEX )
     ] [-location <location>] [-valueType <valueType> <valueExpression
     >....
2 <!--NeedCopy-->
```


GUI を使用してセキュリティ免除ルールまたは緩和ルールをバインドするには

1. ［セキュリティ］ **> Citrix Web App Firewall**］ **>** ［プロファイル］ に移動します。
2. 詳細ペインで、プロファイルを選択し、[編集] をクリックします。
3. ［**Citrix Web App Firewall** プロファイル］ ページで、［詳細設定］ セクションの ［緩和ルール］ をクリックします。
4. [緩和ルール] セクションで、[**startURL**] をクリックし、[編集] をクリックします。
5. [開始 **URL** 緩和ルール] ページで、[追加] をクリックします。
6. [**URL** 緩和ルールの開始] ページで、次のパラメータを設定します。
   a) 有効。このチェックボックスを選択して、緩和ルールを有効にします。
   b) 開始 URL。正規表現の値を入力します。
   c) コメント。緩和ルールについて簡単に説明してください。
7. **[**作成**]** して **[**閉じる**]** をクリックします。



GUI を使用して **Web App Firewall** プロファイルを構成するには

1. ［セキュリティ］ **> Citrix Web App Firewall**］ **>** ［プロファイル］ に移動します。
2. 詳細ウィンドウで、構成するプロファイルを選択し、[編集] をクリックします。

3. [**Web App Firewall** プロファイルの構成] ダイアログボックスの [セキュリティチェック] タブで、セキュリティチェックを構成します。

   - チェックに対するアクションを有効または無効にするには、一覧でそのアクションのチェックボックスをオンまたはオフにします。
   - そのチェックに他のパラメータを設定するには、一覧でそのチェックの右端にある青色の山形をクリックします。表示されるダイアログボックスで、パラメータを設定します。これらはチェックごとに異なります。

     チェックを選択し、ダイアログボックスの下部にある [開く] をクリックして、そのチェックの [緩和の構成] ダイアログボックスまたは [規則の構成] ダイアログボックスを表示することもできます。これらのダイアログボックスもチェックごとに異なります。ほとんどの場合、[チェック] タブと [一般] タブがあります。チェックが緩和またはユーザー定義規則をサポートしている場合、「チェック」(Checks) タブには「追加」(Add) ボタンが表示されます。このボタンでは、チェックのリラクゼーションまたは規則を指定できる別のダイアログボックスが開きます。(緩和とは、特定のトラフィックを小切手から除外するルールです)。緩和が既に設定されている場合は、緩和を選択して [開く] をクリックして変更できます。
   - チェックの学習済み例外またはルールを確認するには、チェックを選択し、[学習済み違反] をクリックします。[学習済み規則の管理] ダイアログボックスで、学習した例外または規則を順に選択します。
     - 例外またはルールを編集してリストに追加するには、[**Edit & Deploy**] をクリックします。
     - 例外またはルールを変更せずに受け入れるには、[**Deploy**] をクリックします。
     - リストから例外または規則を削除するには、[スキップ] をクリックします。
   - 確認する例外または規則の一覧を更新するには、[**Refresh**] をクリックします。
   - ラーニングビジュアライザーを開き、それを使用して学習済みルールを確認するには、[ビジュアライザー] をクリックします。
   - チェックに一致する接続のログエントリを確認し、チェックを選択して [ログ] をクリックします。この情報を使用して、どのチェックがマッチング攻撃であるかを判断し、それらのチェックのブロックを有効にすることができます。この情報を使用して、正当なトラフィックと一致するチェックを特定することもできます。これにより、正当な接続を許可するように適切な免除を構成できます。ログの詳細については、ログ、統計、およびレポートを参照してください。
   - チェックを完全に無効にするには、リストで、そのチェックの右側にあるすべてのチェックボックスをオフにします。

4. [設定] タブで、プロファイル設定を構成します。

   - 以前に作成および設定した一連のシグニチャにプロファイルを関連付けるには、[Common Settings] の [Signatures] ドロップダウンリストでシグニチャのセットを選択します。

> 注:
>
> [共通設定] セクションを表示するには、ダイアログボックスの右側のスクロールバーを使用して下にスクロールする必要があります。

- HTML または XML エラーオブジェクトを設定するには、該当するドロップダウンリストからオブジェクトを選択します。

  > 注:
  >
  > まず、[インポート] ペインで使用するエラーオブジェクトをアップロードする必要があります。エラーオブジェクトのインポートの詳細については、「インポート」を参照してください。

- デフォルトの XML コンテンツタイプを構成するには、コンテンツタイプの文字列を [既定の要求] および [既定の応答] テキストボックスに直接入力するか、[許可されたコンテンツタイプの管理] をクリックして許可されたコンテンツタイプのリストを管理します。» もっと...。

5. 学習機能を使用する場合は、「ラーニング」をクリックし、「ラーニング機能の構成と使用」の説明に従って、プロファイルの学習設定を構成します。
6. **OK** をクリックして変更を保存し、[プロファイル] ペインに戻ります。

Web アプリケーションファイアウォールプロファイルの設定

October 7, 2021

アプライアンスで構成する必要があるプロファイル設定は次のとおりです。

コマンドプロンプトで入力します。

```
add appfw profile <name> [-invalidPercentHandling <invalidPercentHandling>]
 [-checkRequestHeaders ( ON | OFF )] [-URLDecodeRequestCookies ( ON | OFF )
] [-optimizePartialReqs ( ON | OFF )] [-errorURL <expression>]
```

例**:**

```
add appfw profile profile1 [-invalidPercentHandling secure_mode] [-checkRequestHeaders
 ON] [-URLDecodeRequestCookies OFF] [-optimizePartialReqs OFF]
```

各項目の意味は次のとおりです。

無効なパーセント処理。パーセントエンコードされた名前と値を処理する方法を設定します。

使用可能な設定は次のように機能します。

asp_mode-ストリップと解析の無効なパーセント。例: `-curl -v "http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)` が取り除かれ、残りのコンテンツが検査され、SQLInjection チェックのアクションが実行されます。

secure_mode-無効なパーセントコード値を検出し、それを無視します。例: `curl -v "http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)`-が検出され、カウンタが増分され、コンテンツがそのままサーバに渡されます。
apache_mode-このモードは、セキュアモードと同様に動作します。
設定可能な値:apache_mode, asp_mode, secure_mode
デフォルト値:secure_mode

optimizePartialReqs。オフ/オン(セーフオブジェクトなし)の場合、Citrix ADC アプライアンスは部分的な要求をバックエンドサーバーに送信します。この部分的な応答は、クライアントに送り返されます。OptimizePartialReqs は、セーフオブジェクトが構成されている場合に意味があります。アプライアンスは、OFF のときにサーバーからの完全な応答要求を送信し、ON のときに部分的な応答のみを要求します。

使用可能な設定は次のとおりです。

ON -クライアントによる部分的な要求は、バックエンドサーバーへの部分的な要求になります。
OFF-クライアントによる部分的な要求は、バックエンドサーバーへの完全な要求に変更されます。
可能な値:ON、OFF
デフォルト値:ON

URL デコード要求クッキー。URL デコード要求クッキーを SQL およびクロスサイトスクリプティングチェックの対象にする前にクッキーをデコードします。

可能な値:ON、OFF
デフォルト値:OFF

署名ポスト本文の制限 **(**バイト**)**。'HTTP_POST_BODY' と指定された場所を持つシグニチャのために検査されるリクエストペイロード(バイト単位)を制限します。

デフォルト値:8096
最小値:0
最大値:4294967295

ポスト本文制限 **(**バイト**)**。Web アプリケーションファイアウォールによって検査される要求ペイロード (バイト単位) を制限します。

デフォルト値:20000000
最小値:0
最大値:10 ギガバイト

セキュリティ設定とその GUI 手順の詳細については、「Web App Firewall プロファイルの構成」トピックを参照してください。

ポストボディ制限アクション PostBodyLimit は、許可する HTTP 本体の最大サイズを指定するときに、エラー設定を反映します。エラー設定を反映するには、1 つまたは複数のボディ制限後のアクションを設定する必要があります。この設定は、転送エンコーディングヘッダーがチャンクされる要求にも適用されます。

```
set appfw profile <profile_name> -PostBodyLimitAction block log stats
```


Where,
Block-このアクションは、セキュリティチェックに違反する接続をブロックします。これは、設定された HTTP 本文の最大サイズ（本文後制限）に基づいています。このオプションは常に有効にする必要があります。

Log - このセキュリティチェックの違反を記録します。

Stats -このセキュリティー検査の統計を生成します。

> 注:
>
> ポストボディ制限アクションのログ形式は、
> 標準の監査ログ形式に従うように変更されました。例:
> ns.log.4.gz:Jun 25 1.1.1.1. <local0.info> 10.101.10.100 06/25/2020:10:10:28 GMT 0-PPE-0 : **default** APPFW APPFW_POSTBODYLIMIT 1506 0 : <Netscaler IP> 4234-PPE0 - testprof ><URL> Request post body length(<Post Body Length >)exceeds post body limit.

InspectQueryContentTypes 次のコンテンツタイプのインジェクションされた SQL およびクロスサイトスクリプトの要求クエリと Web フォームを検査します。

```
set appfw profile p1 -inspectQueryContentTypes HTML XML JSON OTHER
```

可能な値: HTML、XML、JSON、その他

デフォルトでは、このパラメータは「inspectQueryContentTypes: HTML JSON Other」として基本プロファイルと高度な appfw プロファイルの両方に対して設定されます。

クエリコンテンツタイプを **XML** として検査する例:

```
1 > set appfw profile p1 -type XML
2 Warning: HTML, JSON checks except “InspectQueryContentTypes” & “
     Infer Content-Type XML Payload Action” will not be applicable when
     profile type is not HTML or JSON respectively.
3 <!--NeedCopy-->
```

クエリコンテンツタイプを **HTML** として検査する例:

```
1 > set appfw profile p1 -type HTML
2 Warning: XML, JSON checks except “InspectQueryContentTypes” & “Infer
     Content-Type XML Payload Action” will not be applicable when
     profile type is not XML or JSON respectively
3 Done
4 <!--NeedCopy-->
```

クエリコンテンツタイプを **JSON** として検査する例:


```
1 > set appfw profile p1 -type JSON
2 Warning: HTML, XML checks except “InspectQueryContentTypes” & “Infer
      Content-Type XML Payload Action will not be applicable when profile
      type is not HTML or XML respectively
3 Done
4 <!--NeedCopy-->
```

エラー **URL** 式です。Citrix Web App Firewall がエラー URL として使用する URL。最大長さ:2047。

> 注:
>
> 要求された URL の違反をブロックする場合、エラー URL が署名 URL に似ている場合、アプライアンスは接続をリセットします。

Web App Firewall プロファイルタイプの変更

October 7, 2021

Web App Firewall プロファイルに間違ったプロファイルタイプを選択した場合、または保護された Web サイトのコンテンツのタイプが変更された場合は、プロファイルタイプを変更できます。

> 注記プロファイルタイプを変更すると、新しいプロファイルタイプがサポートしていないフィーチャのすべての構成設定と学習済み緩和またはルールが失われます。たとえば、プロファイルの種類を Web 2.0 から XML に変更すると、開始 URL、フォームフィールドの一貫性チェック、その他の HTML 固有のセキュリティチェックの構成オプションが失われます。古いプロファイルタイプと新しいプロファイルタイプの両方でサポートされているオプションの設定は変更されません。

コマンドラインインターフェイスを使用して **Web App Firewall** プロファイルの種類を変更するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw profile <name> -type ( **HTML** | **XML** | **HTML XML** )`
- `save ns config`

例

次の例では、pr-basic という名前のプロファイルのタイプを HTML から HTML XML に変更します。これは、GUI の Web 2.0 タイプと同じです。


```
set appfw profile pr-basic -type HTML XML
save ns config
<!--NeedCopy-->
```

GUI を使用して **Web App Firewall** プロファイルの種類を変更するには

1. **Security > Citrix Web App Firewall > Policies** に移動します。
2. 詳細ウィンドウで、[操作] をクリックし、[プロファイルの種類の変更] をクリックします。
3. [**Web App Firewall** プロファイルの種類の変更] ダイアログボックスの [プロファイルの種類] ドロップダウンリストで、新しいプロファイルの種類を選択します。
4. [**OK]** をクリックして変更を保存し、[プロファイル**]** ペインに戻ります。

Web App Firewall プロファイルのエクスポートとインポート

October 7, 2021

Web App Firewall プロファイルの設定全体（HTML エラーオブジェクト、XML エラーオブジェクト、WSDL スキーマ、XML スキーマ、シグニチャなど、バインドされたすべてのオブジェクトを含む）を複数のアプライアンス間で複製できます。ターゲットプロファイルを選択し、構成をエクスポートしてコンピュータのローカルファイルシステムに保存するか、アーカイブされた構成を転送してサーバに保存することができます。同様に、コンピューターのローカルファイルシステムを参照するか、サーバーからアーカイブをインポートして、以前にエクスポートしたプロファイルを選択し、NetScaler アプライアンスにインポートできます。

プロファイル設定全体をエクスポートし、別のアプライアンスにインポートするオプションは、さまざまなユースケースで役立ちます。たとえば、テストベッドのセットアップで Web App Firewall プロファイルを構成して、期待どおりに動作していることをテストおよび検証できます。問題がなければ、プロファイルをエクスポートし、本番 NetScaler アプライアンスにプロファイル構成をインポートできます。この機能は、設定のバックアップにも役立ちます。変更を行う前にプロファイルをエクスポートできるため、必要に応じて設定を既知の状態に簡単にロールバックできます。

> 注
>
> あるビルドからエクスポートおよびアーカイブされた Web App Firewall プロファイルは、新しいリリースで導入された変更が互換性の問題を引き起こす可能性があるため、別のビルドを実行しているシステムに復元できません。アーカイブされたプロファイルをエクスポート元のビルドとは異なるビルドに復元しようとすると、ns.log にエラーメッセージが記録されます。

プロファイルのエクスポートおよびインポート機能は、GUI（GUI）とコマンドラインインターフェイス（CLI）の両方で使用できます。GUI は使いやすい **Action** オプションを提供するので、推奨されます。ボタンをクリックするだけで、プロファイルの構成全体を エクスポートまたは インポートできます。

CLI を使用した Web App Firewall プロファイルのエクスポート

CLI を使用してプロファイルを エクスポートする場合は、構成を アーカイブしてから エクスポートする必要があります。プロファイルを インポートするには、アーカイブを NetScaler アプライアンスに インポートしてから、**restore** コマンドを実行して構成を抽出する必要があります。プロファイル設定のエクスポート、インポート、および管理には、次の CLI コマンドセットを使用できます。

アーカイブをエクスポートする **CLI** コマンド:

- `archive appfw profile <name> <archivename> [-comment <string>]`
- `export appfw archive <name> <target>`

アーカイブをインポートする **CLI** コマンド:

- `import appfw archive <src> <name> [-comment <string>]`
- `restore appfw profile <archivename>`

アーカイブを管理するための **CLI** コマンド:

- `show appfw archive`
- `rm appfw archive <name>`

あるアプライアンスからプロファイルをエクスポートし、別のアプライアンスへインポートするには、CLI で 5 つの手順が必要です。最初の 3 つのステップは、プロファイル構成が最初に作成されたソースアプライアンスで実行され、次の 2 つのステップは、プロファイル構成がレプリケートされるターゲットアプライアンスで実行されます。

ソース **NetScaler** アプライアンスからプロファイルをエクスポートします。

ステップ **1:** 構成されたプロファイルのアーカイブを作成します。

手順 **2**: アーカイブを NetScaler ファイルシステムにエクスポートします。

手順 **3**: scp などのファイル転送ユーティリティを使用して、エクスポートされたアーカイブファイルを NetScaler アプライアンス A からターゲットの NetScaler アプライアンスに転送します。

ターゲットの **NetScaler** アプライアンスにプロファイルをインポートします。

ステップ 4: import コマンドを実行して、アーカイブファイルをインポートします。NetScaler のローカルファイルシステムからアーカイブをインポートするか、HTTP プロトコルまたは HTTPS プロトコルを使用して URL を使用してサーバーからアーカイブをインポートできます。

ステップ 5: インポートされたアーカイブからプロファイル構成を復元するには、restore コマンドを実行します

コマンドラインインターフェイスを使用して **Web App Firewall** プロファイルをエクスポートするには:

まず、プロファイルの設定を アーカイブし、アーカイブをターゲットの場所に エクスポートします。コマンドプロンプトで、次のコマンドを入力します。

```
archive appfw profile <profileName> <archiveName>
```

各項目の意味は次の通りです:

- <profileName> は、アーカイブするプロファイルの名前です。
- <archiveName> は、作成するアーカイブファイルの名前です。

上記のコマンドを実行すると、アーカイブファイルのインスタンスが 2 つ作成されます。1 つは /var/tmp フォルダにあり、もう 1 つは /var/archive/appfw フォルダにあります。

```
export appfw archive <archiveName> <target>
```

各項目の意味は次の通りです:

- <archiveName> は、エクスポートするアーカイブの名前です。(前のコマンドと同じ名前。
- <target> は、接頭辞として local: で始まるファイルパスで、その後に<archiveName>が続きます。

エクスポートコマンドを実行すると、エクスポートされたアーカイブファイルが NetScaler アプライアンスのファイルシステムの/var/tmp フォルダーに保存されます。

例:

```
> archive appfw profile test_pr archived_test_pr

> export appfw archive archived_test_pr local:dutA_test_pr
```

上記の 2 つのコマンドを実行した後、/var/tmp フォルダには、archived_test_pr ファイルとエクスポートされたコピー、dutA_test_pr, サイズは同じです。CLI から、シェルにドロップしてフォルダに移動し、これらのファイルが存在することを確認できます。

アーカイブファイルをエクスポートした後、**scp** またはその他のファイル転送ユーティリティを使用して、アーカイブファイルのコピーを作成した NetScaler アプライアンスからターゲットの NetScaler アプライアンスに転送できます。

CLI を使用した **Web App Firewall** プロファイルのインポート

アーカイブファイルをソースアプライアンスからターゲットアプライアンスに正常に scp した後、プロファイルのアーカイブを インポートする準備が整いました。次に、**restore** コマンドを実行して、ターゲットアプライアンスでプロファイルの構成を複製します。

ターゲットアプライアンスにログオンします。シェルと cd を /var/tmp フォルダにドロップし、このアプライアンス上の scp ファイルのサイズがソースアプライアンス上の元のアーカイブファイルのサイズと一致していることを確認します。シェルを終了してコマンドラインに戻ります。

CLI を使用してプロファイルをインポートするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

```
import appfw archive <src> <name> [-comment <string>]
```

各項目の意味は次のとおりです。

- <src> は、そのファイルが作成されたソースアプライアンスから転送された後のアーカイブファイルの場所です。ローカルファイルシステムとファイル名を使用できます。アーカイブをサーバーに配置した場合は、

URL を使用してアーカイブされたファイルをインポートできます。パスまたはファイル名にスペースが含まれている場合は、URL を二重引用符で囲みます。

- <name> は、インポートするアーカイブファイルの名前です。
- <string> は、アーカイブの目的に関するオプションの説明です。

```
restore appfw profile <archiveName>
```

例:

A. ローカルファイルからインポートし、次にリストアします。

```
> import appfw archive local:dutA_test_pr dut2_test_pr
```

```
> restore appfw profile dut2_test_pr
```

B. URL からインポートし、次にリストアします。

```
import appfw archive http://10.217.30.16/FFC/Profile_ImportExport/
dutA_test_pr.tgz my_archive
restore appfw profile my_archive
```

この例では、ターゲット NetScaler アプライアンス上のすべてのバインドされたオブジェクト（署名、HTML エラーページ、緩和ルールなど）とともに test_pr プロファイルを復元します。

次の CLI コマンドを使用して、詳細についての man ページにアクセスできます。

- man archive appfw profile
- man export appfw archive
- man import appfw archive
- man restore appfw profile
- man show appfw archive
- man rm appfw archive

GUI を使用した Web App Firewall プロファイルのエクスポートとインポート

GUI は CLI よりも使いやすくなっています。[**Export**] をクリックすると、アーカイブ操作とエクスポート操作の両方が実行されます。同様に、[インポート] をクリックすると、インポートと復元の両方が実行されます。GUI は、ユーティリティにアクセスするコンピュータのローカルファイルシステムにアクセスできます。アーカイブのコピーをエクスポートして、ローカルコンピュータに保存できます。その後、アーカイブファイルをあるアプライアンスから別のアプライアンスに手動で転送しなくても、このコピーをターゲットアプライアンスに直接インポートできます。

GUI を使用して **Web App Firewall** プロファイルをエクスポートするには、次の手順を実行します。

1. [設定] > [セキュリティ] > [**Web App Firewall**] > [プロファイル] に移動します。
2. 詳細ウィンドウで、エクスポートするプロファイルを選択します。[**Actions**] をクリックし、[**Export**] を選択して、コンピュータのローカルファイルシステムにコピーをダウンロードして保存します。

GUI を使用して **Web App Firewall** プロファイルをインポートするには、次の手順を実行します。

1. [設定] > [セキュリティ] > [**Web App Firewall**] > [プロファイル] に移動します。
2. 詳細ペインで、[操作] をクリックし、[インポート] を選択します。[Web App Firewall プロファイルのインポート] ペインの [インポート元 *] 選択ボックスには、次の 2 つのオプションがあります。

URL: **URL** を指定することで、アーカイブのインポートを選択できます。このオプションを選択すると、**URL** 入力ボックスにアーカイブされたファイルの絶対パスを指定する必要があります。

[ファイル**]:** ローカルの [ファイル] からアーカイブをインポートできます。このオプションを選択すると、**[**ローカルファイル**]** 選択フィールドが表示されます。コンピュータのローカルファイルを参照して、ターゲットアーカイブファイルを選択できます。

[**Create**] をクリックして、指定したアーカイブをインポートします。インポート操作が正常に完了すると、ターゲットアプライアンスにプロファイル構成が作成されます。

ハイライト

- プロファイルのエクスポートおよびインポート機能を使用すると、構成手順を繰り返す必要なく、構成全体（プロファイルのすべてのインポートオブジェクトおよび構成済み緩和ルールを含む）を複数のアプライアンスに複製できます。
- 署名、WSDL、スキーマ、エラーページなどのインポートされたオブジェクトは、アーカイブされた tar ファイルに含まれ、ターゲットアプライアンスにレプリケートされます。
- カスタマイズされたフィールドタイプは、アーカイブされた tar ファイルに含まれ、ターゲットアプライアンスにレプリケートされます。
- アーカイブされたプロファイルのポリシーバインディングは、構成が復元されるときにレプリケートされません。アプライアンスにプロファイルをインポートした後、ポリシーを設定してプロファイルにバインドする必要があります。
- アーカイブファイルの名前は 31 文字までです。プロファイル名と同様に、アーカイブ名は英数字またはアンダースコアで始まり、英数字とアンダースコア（_）、数字（#）、ピリオド（.）、スペース（）、コロン（:）、アットマーク（@）、等号（=）、ハイフン（-）のみを含む必要があります。
- アーカイブに関連付けられたコメントは、アーカイブされた構成の目的を伝えるのに十分な説明が必要です。コメントの最大長は 255 文字です。
- clear config –force basic“ このコマンドでは、アーカイブされたプロファイルは削除されません。
- インポートおよびエクスポートプロファイル機能は、高可用性（HA）配置でサポートされます。

デバッグに関するヒント

- コマンド実行中に /var/log/ns.log を監視し、エラーメッセージがあるかどうかを確認します。
- 追加のログ（_restore.log、削除.log、インポート.log）は、/var/tmp/フォルダ内に生成されます。これらは、対応する操作中に問題をデバッグするのに役立ちます。これらのログのサイズが 1 MB に達すると、ログメッセージがパージされ、ログファイルが元のサイズの 4 分の 1 に縮小されます。
- ローカルファイルシステムではなく URL オプションを使用しているときに import コマンドが失敗した場合は、DNS ネームサーバとルート設定が正確に構成されていることを確認します。

- HTTPS プロトコルを使用してアーカイブをインポートする場合、HTTPS サーバーがクライアント証明書認証を要求すると、コマンドが失敗することがあります。

Web アプリケーションファイアウォールログによるトラブルシューティングの容易さ

October 7, 2021

セキュリティ攻撃が発生した場合は、アプライアンスで詳細な WAF ロギングをキャプチャすることが重要です。このために、アプリケーションファイアウォールプロファイルで「VerboseLogLevel」パラメーターを構成できます。

ウェブトラフィックにセキュリティ攻撃があるとします。アプライアンスがトラフィックを受信すると、HTTP ヘッダーの詳細、ログパターン、パターンペイロード情報などの違反の詳細がログに記録され、ADM サーバに送信されます。ADM サーバーは、詳細なログを監視し、監視および追跡を目的として [Security Insight] ページに表示します。

コマンドインターフェイスを使用した詳細ログレベルの構成

詳細な WAF ログをキャプチャするには、次のコマンドを設定します。
コマンドインターフェイスで、次のように入力します。

```
set appfw profile <profile_name> -VerboseLogLevel (pattern|patternPayload|
patternPayloadHeader)
```

例

```
set appfw profile profile1 -VerboseLogLevel patternPayloadHeader
```

使用可能なログレベルは次のとおりです。

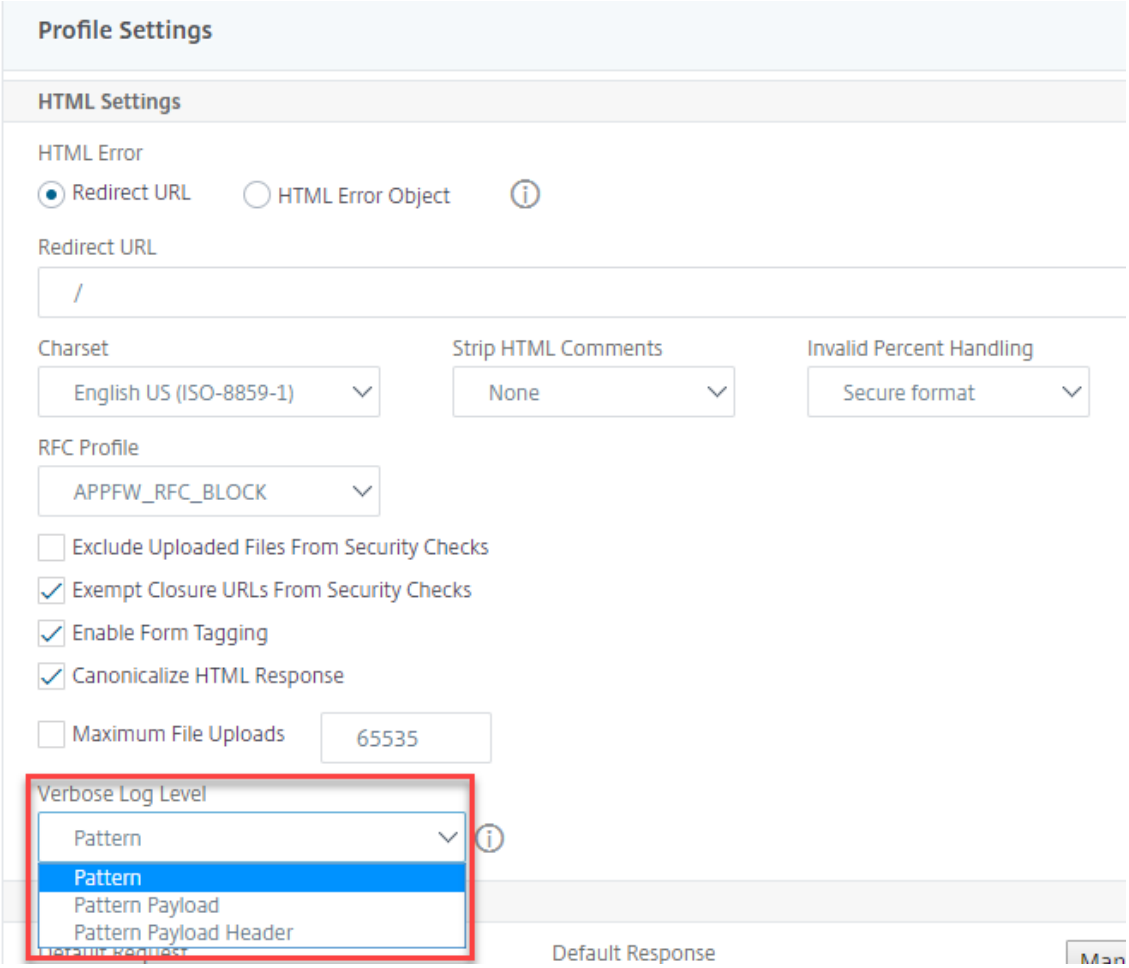
1. Pattern- 違反パターンのみをログに記録します。
2. パターンペイロード。違反パターンと 150 バイトの余分なフィールド要素ペイロードを記録します。
3. パターンペイロードヘッダー。違反パターン、150 バイトの余分なフィールド要素ペイロードおよび HTTP ヘッダー情報をログに記録します。

Citrix ADC GUI を使用した詳細なログレベルの構成

次の手順に従って、WAF プロファイルで詳細ログレベルを設定します。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] の下の [プロファイル設定] をクリックします。

4. [プロファイル設定] セクションで、[詳細ログレベル] フィールドで詳細な WAF ログレベルを選択します。
5. [**OK**] をクリックし、[完了] をクリックします。



ファイルアップロードの保護

October 7, 2021

多くの攻撃者は、マルチフォーム送信中に悪意のあるコード、ウイルス、またはマルウェアを添付ファイルとしてアップロードしようとします。私たちのネットワークを保護し、そのような脅威を克服することが重要です。このような悪意のあるファイルのアップロードを防ぐために、Citrix ADC 管理者は WAF プロファイルで許可される一連のファイルアップロード形式を構成できるようになりました。これにより、ファイルのアップロードを特定の形式に制限し、悪意のあるファイルのアップロードからアプライアンスを保護します。ただし、保護は、WAF プロファイルの「ExcludeFileUploadFormChecks」オプションを無効にする場合にのみ機能します。

ファイルのアップロードの仕組み

許可されるファイルアップロード形式を設定する場合、コンポーネントの相互作用は次のようになります。

- クライアントリクエストには、ファイルアップロードタイプ（PDF など）のフォーム送信があります。
- セキュリティチェックの一環として、WAF はリクエストペイロードを検査し、（マジックシグネチャ番号に基づいて）ファイルタイプを検証します。
- ファイルの種類が許可されるファイル形式の場合、ファイルの種類バインディングに基づく対応するアクションが適用されます。
- ファイルタイプを検証するために、アプライアンスはペイロードを検査し、既知のオフセットで既知のマジックナンバーをチェックします。各ファイルタイプには、ファイルタイプを検証する一連のマジック番号があります。
- 検証に合格した場合のみ、WAF はファイルを許可形式として識別し、関連付けられたアクションが適用されます。

Citrix ADC CLI を使用してファイルタイプのアップロードを構成する

許可されるファイル形式を構成するために、アプライアンスはファイルアップロードパラメータにバインドされた WAF プロファイルを使用します。

1. Web アプリケーションファイアウォールプロファイルの構成

Web アプリケーションのファイアウォールプロファイルを構成するには、次のように入力します。

```
set appfw profile <profile_name> [-fileUploadTypesAction <fileUploadTypesAction
>]<fileUploadTypesAction> = ( none | block | log | stats )
```

例

```
set appfw profile profile1 -fileUploadTypesAction block
```

1. Web アプリケーションファイアウォールプロファイルをファイルアップロードパラメータでバインドします。

ファイルアップロードパラメータを使用してプロファイルをバインドするには、次のように入力します。

```
bind appfw profile <profile_name> - fileUploadType <form_field > <form_action_url
> -fileType <fileType> ( pdf | msdoc | text | image | any)
```

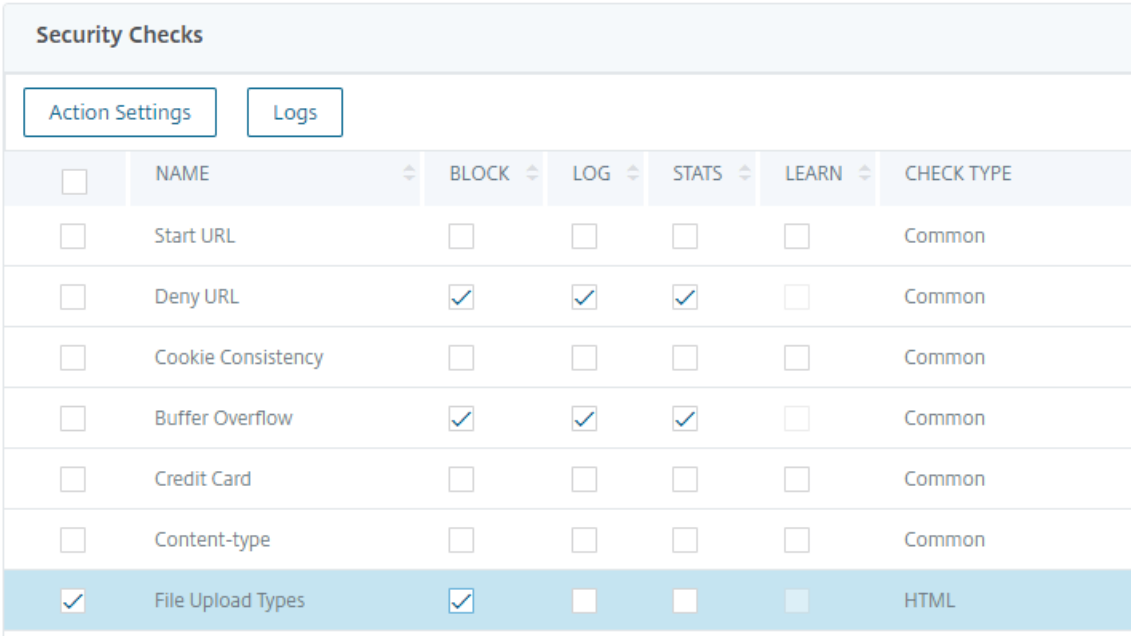
例

```
bind appfw profile profile1 -fileuploadType image action_url -fileType
image
```

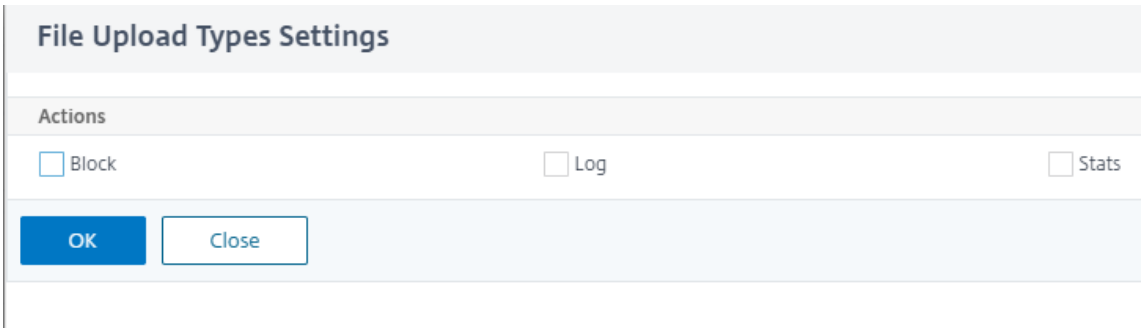

Citrix ADC GUI を使用してファイルアップロードのセキュリティ保護を構成する

ファイルのアップロード設定は以下の手順で行います。

1. ナビゲーションペインで、[セキュリティ] > [プロファイル] に移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. **Citrix Web App Firewall** プロファイルページで、[詳細設定] の [セキュリティチェック] をクリックします。
4. [セキュリティチェック] セクションで、**[**ファイルのアップロードの種類] 設定に移動します。



5. チェックボックスをオンにして、[アクションの設定] をクリックします。
6. 「ファイルアップロードタイプ設定」ページで、ファイルアップロードアクションを設定します。
7. [**OK**] をクリックします。
8. [**Citrix Web App Firewall** プロファイル] ページで、[**OK**] をクリックし、[完了] をクリックします。

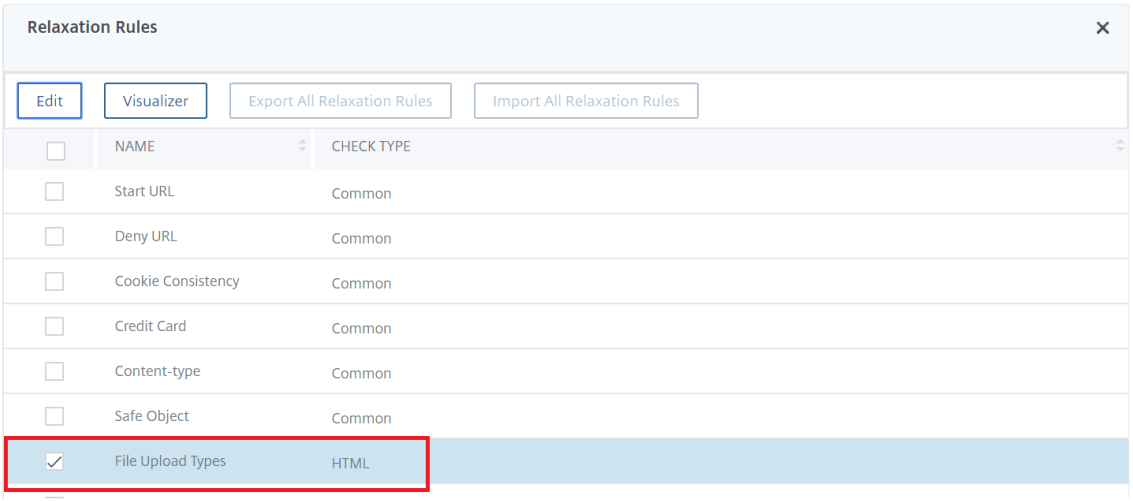


Citrix ADC GUI を使用してファイルアップロード緩和ルールを構成する

ファイルアップロードのセキュリティ保護を緩和して、誤検出を回避できます。たとえば、アプライアンスはファイルのアップロードをブロックしますが、緩和ルールを追加して、特定の Web サイトからのファイルのアップロードを許可できます。これにより、アプライアンスは指定されたフォームフィールドのセキュリティ検査をバイパスし、ユーザーがアクション URL に記載されている Web サイトからファイルをアップロードできるようにします。

緩和ルールを作成するには、以下の手順に従います。

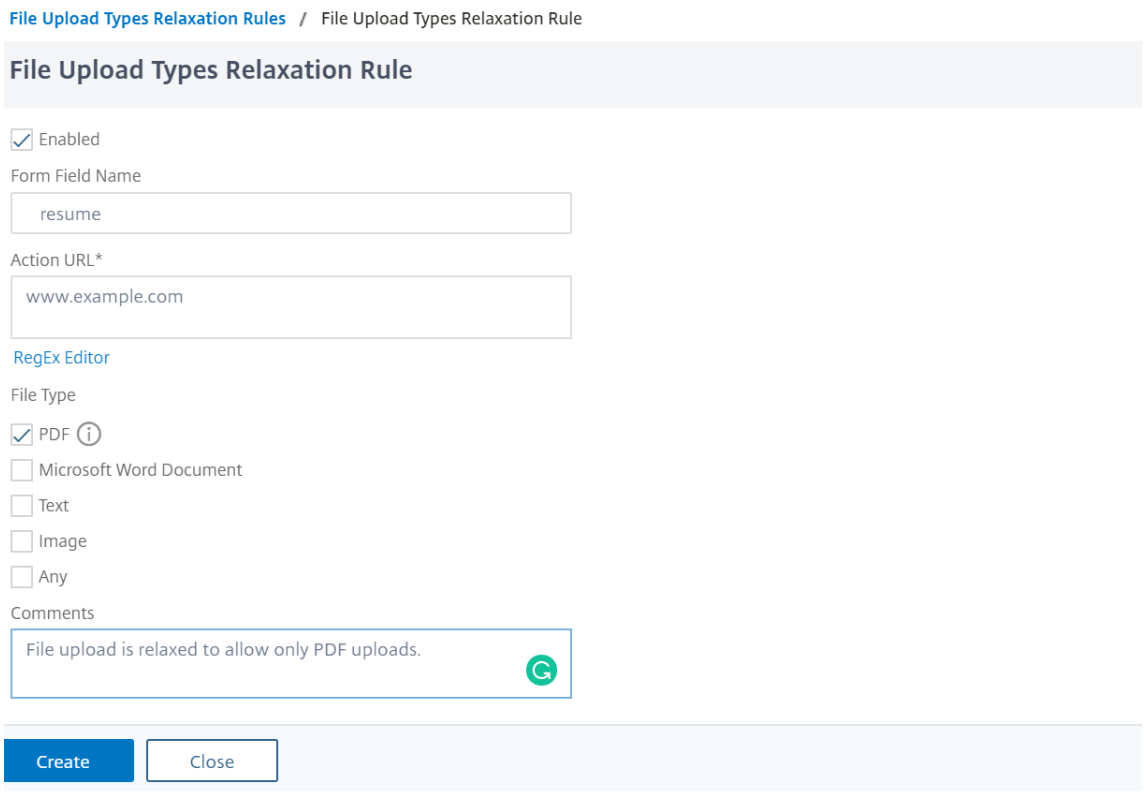
1. ナビゲーションペインで、［セキュリティ］> ［**Citrix Web App Firewall**］< プロファイルに移動します。
2. [プロファイル] ページで、[追加] をクリックします。
3. ［**Citrix Web App Firewall** プロファイル］ページで、［詳細設定］の下の［リラクゼーションルール］をクリックします。
4. 「緩和ルール」セクションで、「ファイル」「アップロードタイプ」を選択し、「編集」をクリックします。



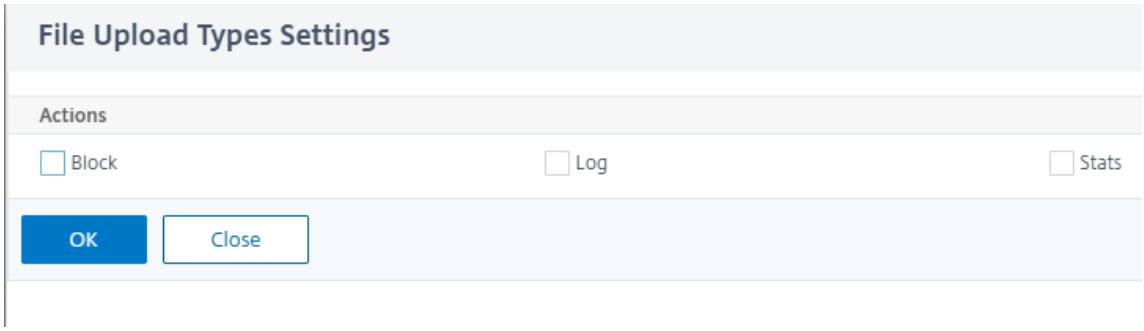
5. [ファイルのアップロードの種類変更ルール] ページで、[追加] をクリックします。
6. 「ファイルのアップロードの種類」緩和ルールページで、次のパラメータを設定します。

   a) 有効。緩和ルールを有効にするには、このチェックボックスをオンにします。
   b) [フォームフィールド名]: セキュリティチェックを必要としないフィールド名を入力します。
   c) アクションの URL。セキュリティチェックから除外する必要があるフォーム送信 URL。
   d) [ファイルタイプ]: ユーザーがアップロードできるようにする必要があるファイルタイプ。
   e) コメント。ファイルのアップロードに関する簡単な説明。

7. ［作成］をクリックします。



8. [**Citrix Web App Firewall** プロファイル] ページで、[**OK**] をクリックし、[完了] をクリックします。



ラーニング機能の設定と使用

December 7, 2021

学習機能は、Web App Firewall で保護された Web サイトまたはアプリケーションのアクティビティを監視して、その Web サイトまたはアプリケーションでの通常のアクティビティを構成するものを特定する反復パターンフィルターです。次に、学習機能のサポートを含むセキュリティチェックごとに、最大 2,000 の推奨ルールまたは例外 (緩和) のリストを生成します。ユーザーは通常、必要な緩和を手動で入力するよりも、学習機能を使用する方が、緩和を設定するほうが簡単です。

ラーニング機能をサポートするセキュリティー検査は、次のとおりです。

- URL チェックを開始
- クッキーの整合性チェック
- フォームフィールドの一貫性チェック
- [フィールド形式] チェック
- CSRF フォームタグ付けチェック
- HTML SQL インジェクションチェック
- HTML クロスサイトスクリプティングチェック
- XML サービス拒否チェック
- XML 添付ファイルのチェック
- Web サービスの相互運用性チェック

学習機能の使用時には、2 つの異なるタイプのアクティビティを実行します。まず、この機能を使用できるように機能を有効にして設定します。保護された Web アプリケーションへのすべてのトラフィックでラーニングを使用することも、ラーニング機能で推奨を生成する必要がある IP のリスト（*[*信頼できるラーニングクライアントの追加（*Add Trusted* Learning Clients）] リスト）を設定することもできます。次に、この機能を有効にして、保護された Web サイトへの一定量のトラフィックを処理したら、推奨されるルールと緩和 (学習済みルール) の一覧を確認し、それぞれに次のいずれかの指定をマークします。

- 編集とデプロイ。ルールは [編集] ダイアログボックスに取り込まれ、変更可能になり、変更されたフォームが展開されます。
- デプロイ。変更されていない学習済みルールは、このセキュリティチェックのルールまたは緩和のリストに配置されます。
- スキップ。学習したルールは、展開されていないルールまたは緩和のリストに配置されます。学習したルールは、スキップすると削除されます。ただし、それらはリラクゼーションに追加されないため、再び学習される可能性があります。

学習は、フィールド形式のルールを除き、緩和が設定されている場合にのみ実行されるわけではありません。ルールをスキップすると、学習したデータベースからのみ削除されます。リラクゼーションは追加されないので、再び学習される可能性があります。ルールが展開されると、学習済みデータベースから削除され、緩和もルールに追加されます。リラクゼーションが追加されると、再び学習されることはありません。フィールド形式の保護では、緩和に関係なく学習が実行されます。

学習機能の基本的な構成にはコマンドラインインターフェイスを使用できますが、この機能は主に Web App Firewall ウィザードまたは GUI を使用した構成のために設計されています。コマンドラインを使用すると、限定されたラーニング機能の設定しか実行できません。

このウィザードは、学習機能の構成と Web App Firewall 全体の構成を統合するため、新しい Citrix ADC アプライアンスでこの機能を構成したり、単純な Web App Firewall 構成を管理したりする場合に最も簡単な方法です。GUI ビジュアライザーと手動インターフェイスはいずれも、すべてのセキュリティチェックの学習済みルールに直接アクセスできるため、多くのセキュリティチェックで学習済みルールを確認する必要がある場合に適しています。

ラーニングデータベースのサイズは 20 MB に制限されており、ラーニングが有効になっているセキュリティチェッ

クごとに、学習済みルールまたは緩和が約 2,000 個生成された後に到達します。学習済みルールを定期的に確認せず、承認または無視しても、この制限に達すると、NetScaler ログにエラーが記録され、既存の学習済みルールと緩和を確認するまで学習済みルールは生成されません。

データベースのサイズ制限に達したために学習が停止した場合は、既存の学習済みルールと緩和を確認するか、学習データをリセットすることで、学習を再開できます。学習したルールまたは緩和が承認または無視されると、データベースから削除されます。学習データをリセットすると、既存の学習データはすべてデータベースから削除され、最小サイズにリセットされます。データベースのサイズが 20 MB 未満になると、学習が自動的に再開されます。

コマンドラインインターフェイスを使用して学習設定を構成するには

構成する Web App Firewall プロファイルを指定し、そのプロファイルに含めるセキュリティチェックごとに、最小しきい値またはパーセントしきい値を指定します。最小しきい値は、Web App Firewall がルールまたは緩和を学習する前に処理しなければならないユーザーセッションの最小数を表す整数です (デフォルト:1)。パーセントしきい値は、Web App Firewall がルールまたは緩和を学習する前に特定のパターン（URL、Cookie、フィールド、添付ファイル、またはルール違反）を観察しなければならないユーザーセッションの割合を表す整数です（デフォルト： 0)。次のコマンドを使用します。

- `set appfw learningsettings <profileName> [-startURLMinThreshold <positive_integer>] [-startURLPercentThreshold <positive_integer>] [-cookieConsistencyMinThreshold <positive_integer>] [-cookieConsistencyPercentThres <positive_integer>] [-CSRFtagMinThreshold <positive_integer>] [-CSRFtagPercentThreshold <positive_integer>] [-fieldConsistencyMinThreshold <positive_integer>] [-fieldConsistencyPercentThreshold <positive_integer>] [-crossSiteScriptingMinThreshold <positive_integer>] [-crossSiteScriptingPerce <positive_integer>] [-SQLInjectionMinThreshold <positive_integer>] [-SQLInjectionPercentThreshold <positive_integer>] [-fieldFormatMinThreshold <positive_integer>] [-fieldFormatPercentThreshold <positive_integer>] [-XMLWSIMinThreshold <positive_integer>] [-XMLWSIPercentThreshold <positive_integer>] [-XMLAttachmentMinThreshold <positive_integer>] [-XMLAttachmentPercentThreshold <positive_integer>]`
- `save ns config`

例

次の例では、プロファイルまたは HTML SQL Injection セキュリティー検査で学習設定を有効にして構成します。これは、Web App Firewall に送信されるトラフィックを完全に制御できる、適切な初期テストベッド学習設定です。

```
1 set appfw learningsettings pr-basic -SQLInjectionMinThreshold 10
2 set appfw learningsettings pr-basic -SQLInjectionPercentThreshold 70
```



```
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して学習設定を既定値にリセットするには

指定したプロファイルとセキュリティチェックの学習設定のカスタム設定を削除し、学習設定をデフォルトに戻すには、コマンドプロンプトで次のコマンドを入力します。

- `unset appfw learningsettings <profileName> [-startURLMinThreshold] [-startURLPercentThreshold] [-cookieConsistencyMinThreshold] [-cookieConsistencyPercentThreshold] [-CSRFtagMinThreshold] [-CSRFtagPercentThresh] [-fieldConsistencyMinThreshold] [-fieldConsistencyPercentThreshold] [-crossSiteScriptingMinThreshold] [-crossSiteScriptingPercentThreshold] [-SQLInjectionMinThreshold] [-SQLInjectionPercentThreshold] [-fieldFormatMinThreshold] [-fieldFormatPercentThreshold] [-XMLWSIMinThreshold] [-XMLWSIPercentThreshold] [-XMLAttachmentMinThreshold] [-XMLAttachmentPercen]`
- `save ns config`

コマンドラインインターフェイスを使用してプロファイルの学習設定を表示するには

コマンドプロンプトで、次のコマンドを入力します。

```
show appfw learningsettings <profileName>
```

コマンドラインインタフェースを使用して、プロファイルの未確認の学習済みルールまたは緩和を表示するには

コマンドプロンプトで、次のコマンドを入力します。

```
show appfw learningdata <profileName> <securityCheck>
```

コマンドラインインターフェイスを使用して、未確認の学習済みルールまたは緩和を学習データベースから削除するには

コマンドプロンプトで、次のコマンドを入力します。

```
rm appfw learningdata <profileName> (-startURL <expression> | -cookieConsistency <string> | (-fieldConsistency <string> <formActionURL>)| (-crossSiteScripting <string> <formActionURL>)| (-SQLInjection <string> <formActionURL>)| (-fieldFormat <string><formActionURL>)| (-CSRFTag <expression> <CSRFFormOriginURL
```



```
>)| -XMLDoSCheck <expression> | -XMLWSICheck <expression> | -XMLAttachmentCheck
 <expression>)[-TotalXMLRequests]
```

例

次の例では、姓フォームフィールドに適用されるプロファイル HTML SQL インジェクションセキュリティチェックの未確認学習緩和をすべて削除します。

```
1 rm appfw learningdata pr-basic -SQLInjection LastName
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、未確認の学習済みデータをすべて削除するには

コマンドプロンプトで、次のコマンドを入力します。

```
reset appfw learningdata
```

コマンドラインインターフェイスを使用して学習データをエクスポートするには

コマンドプロンプトで、次のコマンドを入力します。

```
export appfw learningdata <profileName> <securitycheck>[-target <string>]
```

例

次の例では、プロファイルと HTML SQL インジェクションセキュリティチェックの学習済み緩和を、-target パラメーターで指定されたファイル名の /var/learnt_data/ ディレクトリにあるカンマ区切り値 (CSV) 形式のファイルにエクスポートします。

```
1 export appfw learningdata pr-basic SQLInjection -target sqli_ld
2 <!--NeedCopy-->
```

GUI を使用してラーニング機能を設定するには

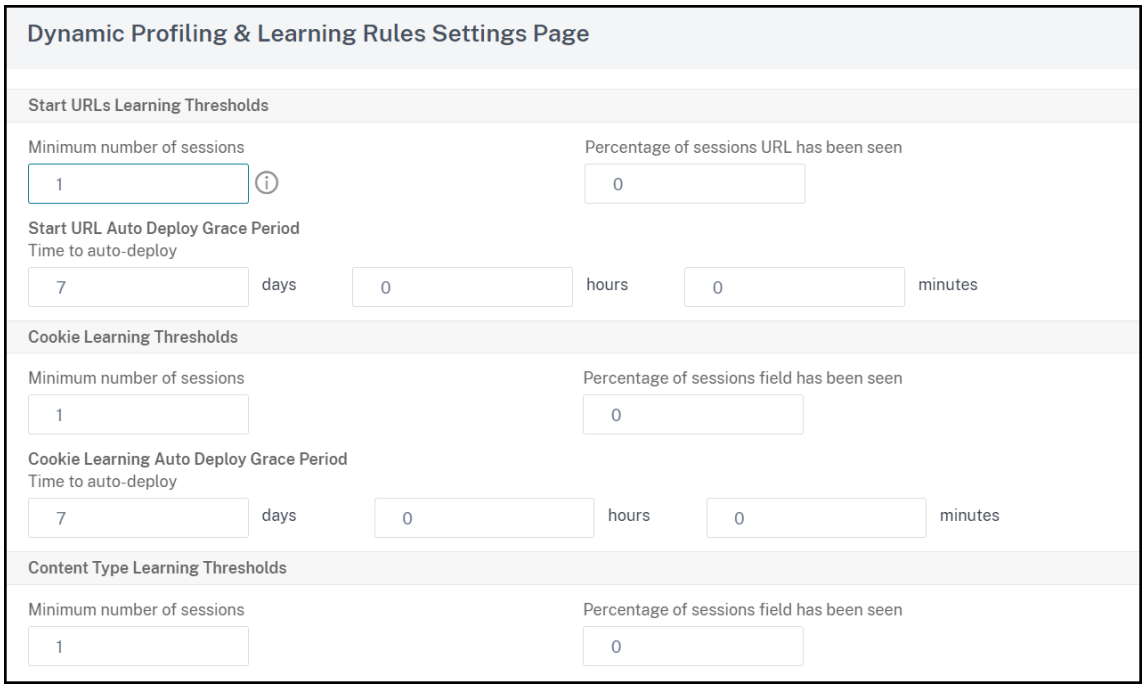
1. セキュリティ > **Web App Firewall** > プロファイルに移動します。
2. [プロファイル] ペインでプロファイルを選択し、[編集] をクリックします。
3. [詳細設定] セクションの **[**学習ルール] をクリックします。
4. [学習したルール] セクションで、セキュリティチェックを選択し、[設定] をクリックします。

5. [セキュリティチェックの設定] ページで、次のパラメータを設定します。

   a) 最小数のしきい値。設定しているセキュリティチェックの学習設定に応じて、最小数のしきい値は、監視する必要のあるユーザーセッションの合計の最小数、遵守する必要のあるリクエストの最小数、または特定のフォームフィールドを監視する必要がある最小回数を指す場合があります。学習したリラクゼーションが生成される前に。デフォルト:1

   b) 回数のしきい値のパーセンテージ。設定しているセキュリティチェックの学習設定に応じて、しきい値の割合は、セキュリティチェックに違反した観察されたユーザーセッションの合計の割合、リクエストの割合、またはフォームフィールドが特定のフィールドタイプと一致した回数の割合を指す場合があります。学習したリラクゼーションが生成されます。デフォルト:0

6. [**OK**] をクリックして閉じます。



7. 学習済みデータをすべて削除して学習機能をリセットするには、[すべての学習済みデータを削除] をクリックします。これにより、最初から観測を再開する必要があります。

   > 注:
   >
   > このボタンは、レビューされておらず、承認またはスキップされた学習済みの推奨事項のみを削除します。受け入れられて展開された学習済み緩和は削除されません。

8. 学習エンジンを特定の IP セットからのトラフィックに制限するには、[信頼できる学習クライアント] をクリックし、使用する IP アドレスをリストに追加します。

   a) [信頼できる学習クライアント] リストに IP アドレスまたは IP アドレス範囲を追加するには、[追加] をクリックします。

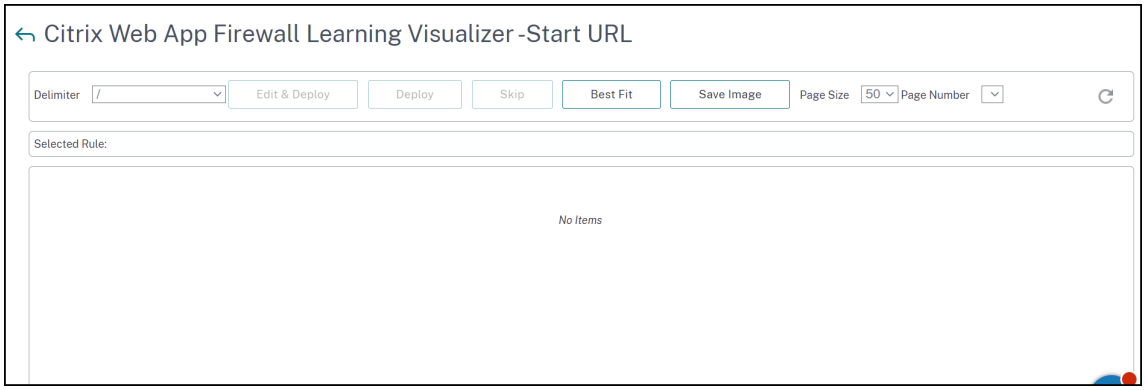
   b) [信頼できるラーニングクライアントの追加] ダイアログボックスの [信頼できるクライアント IP] リストボックスに、IP アドレスまたは IP アドレスの範囲を CIDR 形式で入力します。

c) [コメント] テキスト領域に、この IP アドレスまたは IP アドレス範囲を説明するコメントを入力します。
d) [**Create**] をクリックして、新しい IP アドレスまたは範囲をリストに追加します。
e) 既存の IP アドレスまたは範囲を変更するには、IP アドレスまたは範囲をクリックし、[開く] をクリックします。名前以外は、表示されるダイアログボックスは [信頼できる学習クライアントの追加] ダイアログボックスと同じです。
f) IP アドレスまたは範囲を無効または有効にしても、一覧に残すには、IP アドレスまたは範囲をクリックし、必要に応じて [無効**]** または **[**有効にする] をクリックします。
g) IP アドレスまたは IP 範囲を完全に削除するには、IP アドレスまたは範囲をクリックし、[削除] をクリックします。

9. [閉じる] をクリックして [Web App Firewall プロファイルの構成] ページに戻ります。
10. [完了] をクリックします。

GUI を使用して、学習したルールまたは緩和を確認するには

1. セキュリティ > **Web App Firewall** > プロファイルに移動します。
2. [プロファイル] ペインでプロファイルを選択し、[編集] をクリックします。
3. [詳細設定**]** セクションの **[**学習ルール] をクリックします。
4. [学習したルール**]** セクションで、セキュリティチェックを選択し、[設定] をクリックします。
5. 学習したデータを分岐ツリーとして階層的に確認し、学習したパターンの多くに一致する一般的なパターンを選択できるようにするには、[**Visualizer**] をクリックします。
6. 実際に学習したパターンを確認する場合は、次の手順を実行します。
7. 最初に学習したリラクゼーションを選択し、その処理方法を選択します。
   a) 緩和を変更して承諾するには、「**Edit & Deploy**」をクリックし、緩和の正規表現を編集して「**OK**」をクリックします。
   b) 変更せずに緩和を適用するには、[**Deploy**] をクリックします。
   c) 展開せずに緩和をリストから削除するには、[スキップ] をクリックします。
   d) 前のステップを繰り返して、学習したリラクゼーションを追加するたびに確認します。
8. [閉じる] をクリックして [学習済みルールの管理] ダイアログボックスに戻ります。
9. [完了] をクリックします。



動的プロファイリング

October 7, 2021

ラーニング機能は、バックエンドサーバ上のアクティビティを観察して学習するパターンフィルタです。観察に基づいて、学習エンジンは、セキュリティチェックごとに最大 2000 個のルールまたは例外 (緩和) を生成します。プロセスを自動化し、緩和ルールを自動的に展開するために、Citrix ADC アプライアンスは動的プロファイリングを使用します。

動的プロファイリングでは、アプライアンスは事前に定義されたしきい値について学習したデータを記録し、SNMP アラートをユーザーに送信します。猶予期間内にユーザーがデータをスキップしない場合、アプライアンスはそれを緩和ルールとして自動的にデプロイします。以前は、ユーザーは緩和ルールを手動で展開する必要がありました。現在、動的プロファイリングは、次のセキュリティチェックでのみ使用できます。

1. HTML SQL インジェクション
2. HTML クロスサイトスクリプティング
3. フィールドの書式
4. 開始 URL
5. コンテンツタイプ
6. フィールド書式
7. CSRF フォームのタグ付け
8. クッキーの整合性
9. URL を拒否する
10. バッファオーバーフロー
11. クレジットカード

たとえば、動的プロファイリングで有効化された HTML SQL Injection セキュリティチェックを考えます。学習機能が推奨事項を生成する必要がある IP のリスト（信頼できる学習クライアントリストと呼ばれる）に学習を使用できます。信頼できるクライアントのリストを構成するには、「信頼できるクライアントを学習する」トピックを参照してください。着信トラフィックに違反がある場合、そのトラフィックは学習データとして記録されます。学習したデータが学習エンジンに記録されると、アプライアンスはユーザーに SNMP アラートを送信します。ユーザーが誤検知

を認識せず、猶予期間内に学習したデータをスキップしない場合、アプライアンスはそのデータを緩和ルールとして自動的に展開します。

> 注:
> ダイナミックプロファイルを設定したら、緩和ルールの自動展開についてアプライアンス設定を定期的に確認し、アプライアンスに保存する必要があります。

Citrix ADC コマンドインターフェイスを使用して動的プロファイリングを構成する

動的プロファイリングは、開始 URL、HTML クロスサイトスクリプティング、フィールド形式、または HTML SQL インジェクションのセキュリティチェックで使用できます。動的プロファイリングを設定するには、次の手順を完了する必要があります。

1. 動的学習の設定
2. 自動展開の猶予期間の設定

動的学習の設定

最初のステップとして、アプライアンスに動的学習を設定する必要があります。コマンドプロンプトで入力します。

```
set appfw profile <profile_name> dynamicLearning <security_checks>
```

例

```
set appfw profile test1 dynamicLearning SQLInjection CrossSiteScripting
fieldFormat startURL
```

自動展開の猶予期間の設定

特定のセキュリティチェックで機能を有効にしたら、自動展開の猶予期間を設定する必要があります。

```
set appfw learningsettings <profile name> -crossSiteScriptingAutoDeployGracePeriod
 <seconds>

set appfw learningsettings <profile name> fieldFormatAutoDeploymentGracePeriod
 <seconds>

set appfw learningsettings <profile name> SQLInjectionAutoDeploymentGracePeriod
 <seconds>

set appfw learningsettings <profile name> -startURLAutoDeployGracePeriod <
seconds>
```

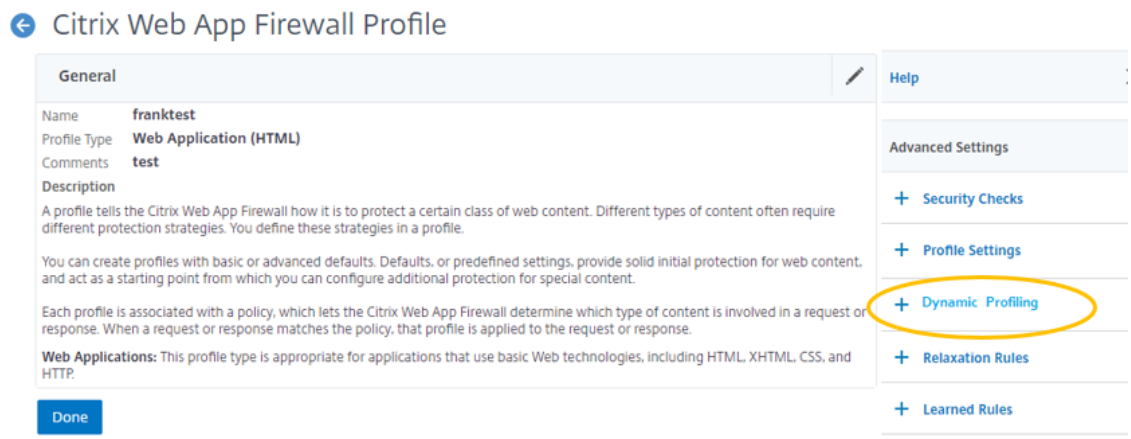

例

```
set appfw learningsettings test1 -crossSiteScriptingAutoDeployGracePeriod 30
set appfw learningsettings test1 -startURLAutoDeployGracePeriod 7
set appfw learningsettings test1 -fieldFormatAutoDeploymentGracePeriod 10
set appfw learning settings test1 -SQLInjectionAutoDeploymentGracePeriod 12
```

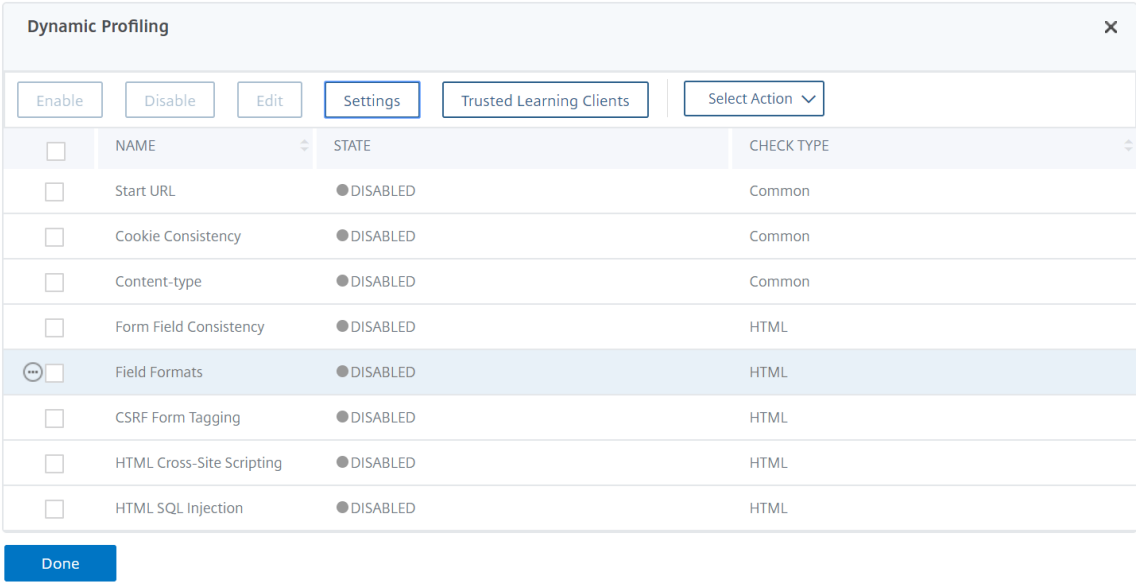
> 注:
>
> ここでは、自動展開の猶予期間は分単位で表示されます。

Citrix ADC GUI を使用した動的プロファイリングの構成

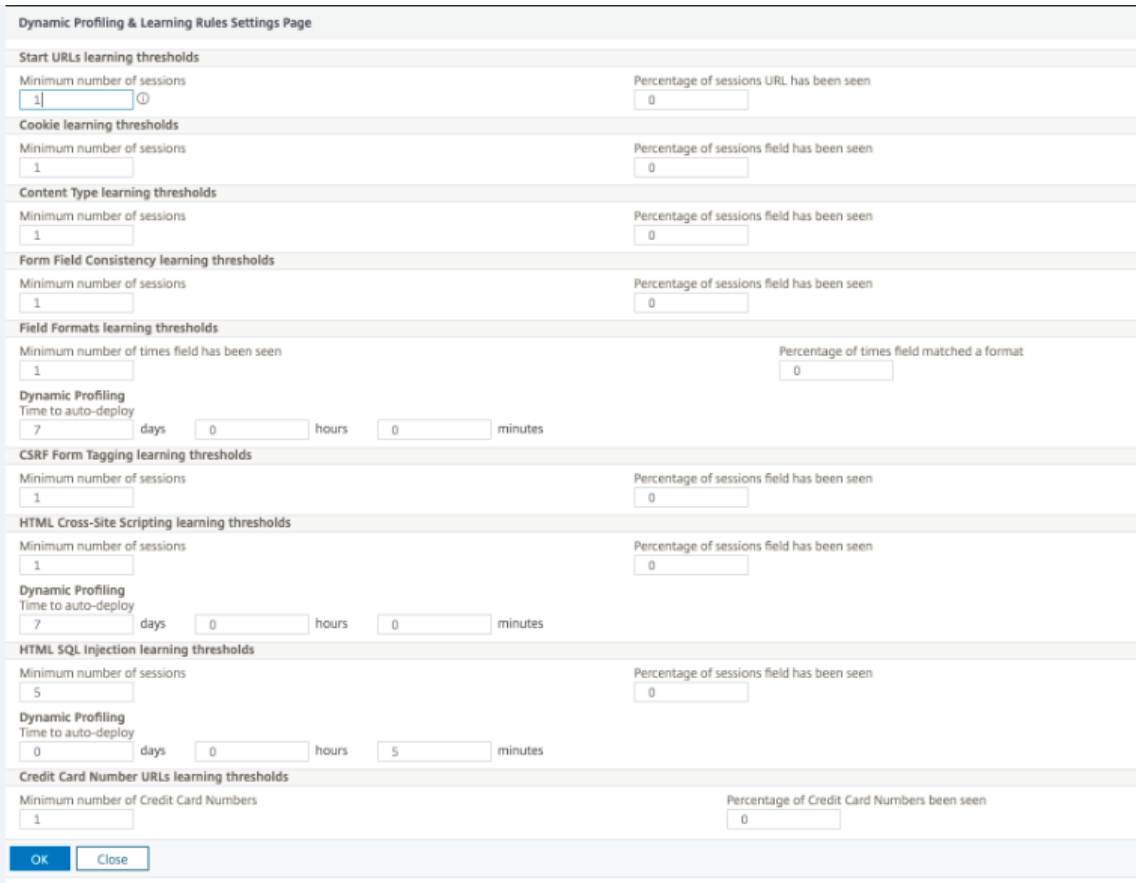
1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ペインでプロファイルを選択し、[編集] をクリックします。
3. [**Citrix Web App** プロファイル] ページで、[詳細設定] の下の [動的プロファイル] をクリックします。



4. 「動的プロファイリング」セクションで、セキュリティーチェックを選択し、「編集」をクリックします。



5. [動的プロファイリングおよび学習設定] ページで、セキュリティチェックの猶予期間を設定します。



6. [**OK**] をクリックし、[完了] をクリックします。

緩和ルールのエクスポートとインポート

動的プロファイリングを有効にすると、学習したデータは緩和ルールとして自動的に展開されます。これに加えて、アプライアンスでは、動的プロファイリングベースの緩和ルールと通常の緩和ルールをエクスポートすることもできます。ステージング環境からルールをエクスポートし、実稼働環境にインポートできます。

> 注:
>
> ルールを実稼働環境にインポートするときは、プロセスが追加され、既存の構成が上書きされないようにする必要があります。

緩和ルールをエクスポートおよびインポートする方法

緩和ルールをエクスポートおよびインポートするには、次の手順を完了する必要があります。

1. まず、動的プロファイルベースのデータをエクスポートする必要があります。このためには、WAF プロファイルの緩和ルールでエクスポートオプションを使用できます。このオプションを選択すると、動的プロファイル緩和ルールと通常の緩和ルールがエクスポートされます。エクスポートオプションを使用すると、構成を圧縮バンドルとしてアプライアンス上にダウンロードできます。
2. ステージング環境からデータをエクスポートしたら、別の Citrix ADC アプライアンスにインポートする必要があります。このためには、WAF プロファイルの緩和ルールで使用可能なインポートオプションを使用する必要があります。このオプションを選択すると、アプライアンスはバンドルされた指定された緩和ルールをインポートし、選択したアプライアンスの WAF プロファイルに復元します。

> 注:
>
> WAF プロファイルで緩和ルールをインポートする場合、次の 2 種類のアクションがあります。
> 補強 — このアクションにより、インポートが付加的になり、既存の設定が上書きされないようになります。
> Overwrite — このアクションは、圧縮エクスポートバンドルに存在する構成で既存の構成を上書きします。

CLI を使用してアーカイブされた緩和ルール・ファイルをインポートする

緩和ルールをインポートするには、アーカイブを Citrix ADC アプライアンスにインポートし、復元コマンドを実行して構成を抽出する必要があります。次の CLI コマンドのセットは、設定のエクスポート、インポート、および管理に使用できます。

特定の場所からアーカイブファイルをインポートして復元するには、コマンドプロンプトで次のように入力します。

```
import appfw archive <src> <name> [-comment <string>]
```

ここで、
「src」: 形式で tar アーカイブファイルのソースを示し、`<protocol>://<host>[:<port>][/<path>]`
「name」: アーカイブ名を示します。
「comment」: このアーカイブに関連するコメント。


```
restore appfw profile <archivename> [-relaxationRules] [-importProfileName
 <string>] [-matchUrlString <string>] [-replaceUrlString <string>] [-
overwrite] [-augment]
```

ここで、
archivename: は tar アーカイブのソースを示します。これは必須の引数です。
「relaxationRules」：すべての appfw 緩和ルールをインポートするオプション。

importProfileName: リストア操作中に緩和ルールを関連付けるために作成または更新されたプロファイル名を示します。
「MatchurlString」：アーカイブされた緩和ルールで照合するアクション URL 文字列を示します。

replaceUrlString: 緩和ルールの復元中にアクション URL で置き換える文字列を示します。

overwrite: 既存の緩和ルールをパージし、インポート中に置き換えるための既存のルールアクション。

augment: インポート中に緩和ルールを強化する既存のルールアクション。

例:

```
import appfw archive local: dutA_test_pr.tgz demo
restore appfw profile dutA_test_pr
```

CLI を使用して、アーカイブされたファイルを選択したアプライアンスにエクスポートします

CLI を使用して appfw 緩和ルールをエクスポートする場合は、設定をアーカイブしてからエクスポートする必要があります。
アーカイブされたファイルをアーカイブおよびエクスポートするには、コマンドプロンプトで次のように入力します。

```
archive appfw profile <name> <archivename> [-comment <string>]
```

ここで、
archive name: は tar アーカイブのソースを示します。これは必須の引数です。
name: エクスポートする緩和ルールを含む appfw プロファイル名を示します。

```
export appfw archive <name> <target>
```

どこで、
名前。tar アーカイブの名前。これは必須の引数です。最大長さ:31
ターゲット。エクスポートするファイルへのパス。これは必須の引数です。最大長：2047

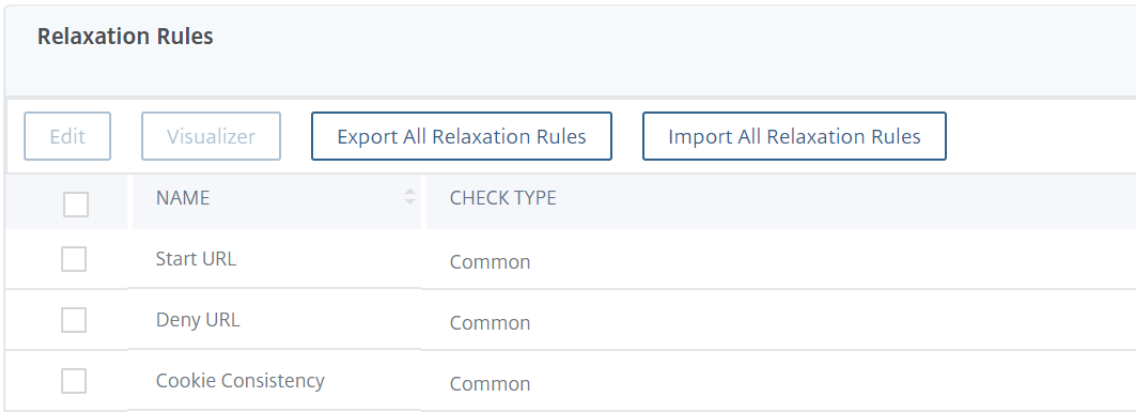
例:

```
> archive appfw profile test_pr archived_test_pr
> export appfw archive archived_test_pr local:dutA_test_pr
```

Citrix ADC GUI を使用して緩和ルールをエクスポートするには

緩和ルールをエクスポートするには、以下の手順に従ってください。

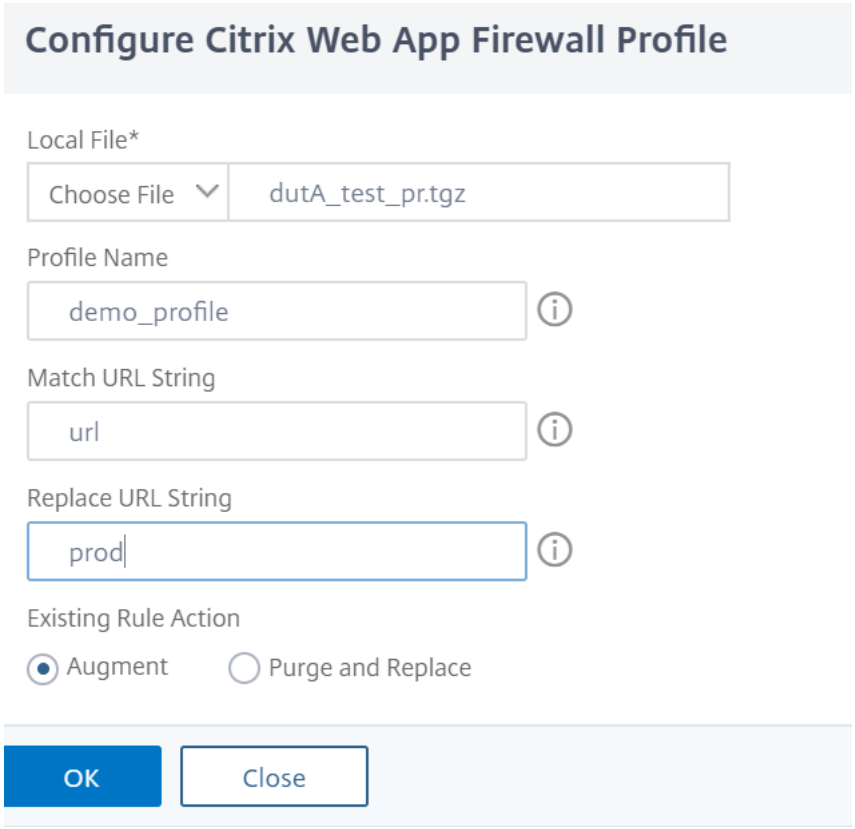
1. ［セキュリティ］**>**［**Citrix Web App Firewall**］に移動します。
2. 詳細ページで、［構成の概要］セクション **Citrix Web App Firewall** プロファイル］リンクをクリックします。
3. ［**Citrix Web App Firewall** プロファイル］ページで、［詳細設定］セクションの［緩和ルール］リンクをクリックします。
4. [緩和ルール] セクションで、[すべての緩和ルールを書き出し] をクリックします。アクションは、すべてのセキュリティチェックと、そのプロファイルで動的学習が有効になっているセキュリティチェックに適用されます。



Citrix ADC GUI を使用して緩和ルールをインポートするには

緩和ルールをインポートする手順を完了します。

1. ［セキュリティ］**>**［**Citrix Web App Firewall**］に移動します。
2. 詳細ページで、［構成の概要］セクション **Citrix Web App Firewall** プロファイル］リンクをクリックします。
3. ［**Citrix Web App Firewall** プロファイル］ページで、［詳細設定］セクションの［緩和ルール］リンクをクリックします。
4. [緩和ルール] セクションで、[すべての緩和ルールを読み込む] をクリックします。
5. ［**Citrix Web App Firewall** プロファイルの構成］ページで、次のパラメーターを設定します。
   a) ローカルファイル。緩和ルールを含む圧縮アーカイブファイルの名前。
   b) プロファイル名。緩和ルールがバインドされているプロファイルの名前。
   c) URL 文字列に一致します。一致する URL の部分。
   d) URL 文字列を置換します。URL 文字列を置き換える URL の部分。
   e) 既存のルール処理ルールで既存のルールを上書きするか、既存のルールを補強するかを選択します。
6. ［**OK**］をクリックします。



プロファイルに関する補足情報

April 7, 2022

Web App Firewall プロファイルの特定の側面に関する補足情報を次に示します。ここでは、セキュリティチェックルールまたは緩和に特殊文字を含める方法と、プロファイルを構成するときに変数を使用する方法について説明します。

設定変数のサポート

静的な値を使用する代わりに、Web App Firewall のセキュリティチェックと設定を構成するために、標準の Citrix ADC 名前付き変数を使用できるようになりました。変数を作成することで、新しい Citrix ADC アプライアンスに構成をエクスポートしてインポートしたり、単一の構成ファイルから既存の Citrix ADC アプライアンスを更新したりすることがより簡単になります。これにより、テストベッドセットアップを使用してローカルネットワークとサーバー用に調整された複雑な Web App Firewall 構成を開発し、その構成を本番環境の Citrix ADC アプライアンスに転送する際の更新が簡単になります。

Web App Firewall 構成変数は、Citrix ADC の標準的な規則に従って、ほかの Citrix ADC の名前付き変数と同じ方法で作成します。GUI と Citrix ADC コマンドラインインターフェイスを使用して、名前付き式変数を作成できます。

次の URL と式は、静的な値の代わりに変数を使用して設定できます。

- 開始 **URL** (-starurl)
- 拒否 **URL** (-denyurl)
- フォームフィールドの一貫性チェック用のフォームアクション **URL** (-fieldconsistency)
- *XML SQL* インジェクションチェックのアクション **URL** (-xmlSQlInjection)
- X*ML* クロスサイトスクリプティングチェックのアクション **URL** (-xmlcross-site スクリプティング)
- *HTML SQL* インジェクションチェック用のフォームアクション **URL** (-sqlInjection)
- フィールドフォーマットチェック用のフォームアクション **URL** (-FieldFormat)
- クロスサイトリクエストフォージェリ *(CSRF)* チェック用のフォームオリジン **URL** とフォームアクション **URL** (-csrfTag)
- *HTML* クロスサイトスクリプティングチェック用のフォームアクション **URL** (-CrossSiteScripting)
- セーフオブジェクト (-safeObject)
- *XML* サービス拒否 *(xDoS)* チェック *(-xmlDos*) のアクション **URL**
- *Web* サービスの相互運用性チェックの URL (-XMLWSIURL)
- < *XML* 検証チェックの **URL** (-xmlValidationURL)
- **XML** 添付ファイルチェック用の *URL* (-XML 添付ファイル URL)

詳細については、「ポリシーと式」を参照してください。

構成で変数を使用するには、変数名を 2 つのアット (@) 記号で囲み、置き換える静的な値と同じように使用します。たとえば、GUI を使用して URL の拒否チェックを構成し、名前付き式変数 myDenyURL を構成に追加する場合は、[拒否 URL の追加] ダイアログボックスの [拒否 URL の拒否] テキスト領域に @myDenyURL @ と入力します。Citrix ADC コマンドラインを使用して同じタスクを実行するには、appfw プロファイルの追加 <name> -denyURLAction @myDenyURL @ と入力します。

PCRE 文字エンコード形式

Citrix ADC オペレーティングシステムは、印刷可能な ASCII 文字セット（16 進 20（ASCII 32）から HEX 7E（ASCII 127）までの 16 進コードを持つ文字のみの直接入力をサポートします。Web App Firewall 構成にその範囲外のコードを持つ文字を含めるには、UTF-8 16 進コードを PCRE 正規表現として入力する必要があります。

Web App Firewall 構成に URL、フォームフィールド名、またはセーフオブジェクト式として含める場合、多くの文字タイプで PCRE 正規表現を使用したエンコードが必要です。それらには以下が含まれます。

- 上部 **ASCII** 文字。16 進数 7F (ASCII 128) から 16 進数 FF (ASCII 255) までのエンコーディングを持つ文字。使用される文字コードによっては、制御コード、アクセントまたはその他の変更を加えた ASCII 文字、非ラテンアルファベット文字、基本 ASCII セットに含まれていない記号がこれらのエンコーディングで参照される場合があります。これらの文字は、URL、フォームフィールド名、およびセーフオブジェクト式で使用できます。
- **2** バイト文字。8 バイトの単語を 2 つ使用するエンコーディングの文字。2 バイト文字は、主に中国語、日本語、韓国語のテキストを電子形式で表すために使用されます。これらの文字は、URL、フォームフィールド名、およびセーフオブジェクト式で使用できます。

- **ASCII** 制御文字。プリンタにコマンドを送信するときに使用される、印刷不能な文字です。16 進数コードが HEX 20 (ASCII 32) 未満の ASCII 文字はすべて、このカテゴリに分類されます。ただし、これらの文字は URL やフォームフィールド名には絶対に使用しないでください。セーフオブジェクト式にはほとんど使用されません。

Citrix ADC アプライアンスは、UTF-8 文字セット全体をサポートするのではなく、次の 8 つの文字セットに含まれる文字のみをサポートします。

- 英語 **(**米国**) (ISO-8859-1)**。ラベルは「英語 US」と表示されますが、Web App Firewall は ISO-8859-1 文字セット（Latin-1 文字セットとも呼ばれる）のすべての文字をサポートしています。この文字セットは、ほとんどの現代西ヨーロッパ言語を完全に表し、残りのいくつかの珍しい文字を除いてすべてを表します。
- 繁体字中国語 **(Big5)**。Web App Firewall は BIG5 文字セットのすべての文字をサポートします。これには、香港、マカオ、台湾、および中国本土以外に住む多くの中国民族遺産の人々によって話され、書き込まれる現代中国語で一般的に使用されているすべての繁体字中国語（表意文字）が含まれます。
- 簡体字中国語 **(GB2312)**。Web App Firewall は GB2312 文字セットのすべての文字をサポートします。これには、中国本土で話され、書き込まれる現代中国語で一般的に使用されるすべての簡体字中国語（表意文字）が含まれます。
- 日本語 **(SJIS)**。Web App Firewall は Shift-JIS (SJIS) 文字セットのすべての文字をサポートします。この文字セットには、現代日本語で一般的に使用されるほとんどの文字 (表意文字) が含まれます。
- 日本語 **(EUC-JP)**。Web App Firewall は EUC-JP 文字セットのすべての文字をサポートします。これには、現代日本語で一般的に使用されるすべての文字（表意文字）が含まれます。
- 韓国語 **(EUC-KR)**。Web App Firewall は EUC-KR 文字セットのすべての文字をサポートします。これには、現代韓国語で一般的に使用されるすべての文字（表意文字）が含まれます。
- トルコ語 **(ISO-8859-9)**。Web App Firewall は、ISO-8859-9 文字セットのすべての文字をサポートします。これには、現代トルコ語で使用されるすべての文字が含まれます。
- ユニコード **(UTF-8)**。Web App Firewall は、現代ロシア語で使用されている文字を含む、UTF-8 文字セットの特定の追加文字をサポートしています。

Web App Firewall を構成するときは、UTF-8 仕様でその文字に割り当てられた 16 進コードを使用して、ASCII 以外のすべての文字を PCRE 形式の正規表現として入力します。通常の ASCII 文字セット内の記号と文字には、その文字セットで 1 桁の 2 桁のコードが割り当てられ、UTF-8 文字セットでも同じコードが割り当てられます。たとえば、感嘆符 (!)ASCII 文字セットでは 16 進コード 21 が割り当てられ、UTF-8 文字セットでは 16 進数 21 にもなります。サポートされている別の文字セットのシンボルと文字には、UTF-8 文字セットの 16 進コードのペアセットが割り当てられます。たとえば、鋭アクセント（á）の付いた文字 a には、UTF-8 コード C3 A1 が割り当てられます。

Web App Firewall 構成でこれらの UTF-8 コードを表すために使用する構文は、ASCII 文字の場合は「xnN」、英語、ロシア語、トルコ語で使用される非 ASCII 文字の場合は「\ xnN\ xNN」、中国語、日本語、韓国語で使用される文字の場合は「\ xnN\ xnN」です。たとえば、!Web App Firewall 正規表現で UTF-8 文字として入力する場合は、「\ x21」と入力します。á を含める場合は、\ xC3\ xA1 と入力します。

> 注:
>
> 通常、ASCII 文字を UTF-8 形式で表現する必要はありませんが、これらの文字が Web ブラウザや基盤となるオペレーティングシステムを混乱させる可能性がある場合は、文字の UTF-8 表現を使用してこの混乱を避けることができます。たとえば、URL にスペースが含まれている場合、特定のブラウザや Web サーバソフトウェアの混乱を避けるために、スペースを x20 としてエンコードできます。

Web App Firewall 構成に含めるには、PCRE 形式の正規表現として入力する必要がある ASCII 以外の文字を含む URL、フォームフィールド名、およびセーフオブジェクト式の例を次に示します。各例は、実際の URL、フィールド名、または式文字列を最初に示し、その後に PCRE 形式の正規表現が続きます。

- 拡張 ASCII 文字を含む URL。

  実際の URL: `http://www.josénuñez.com`

  エンコードされた URL: `^http://www\[.\]jos\xC3\xA9nu\xC3\xB1ez\[.\]com$`

- 拡張 ASCII 文字を含む別の URL。

  実際の URL: `http://www.example.de/trömso.html`

  エンコードされた URL: `^http://www[.]example\[.]de/tr\xC3\xB6mso[.]html$`

- 拡張 ASCII 文字を含むフォームフィールド名。

  `Actual Name: nome_do_usuário`

  エンコードされた名前: `^nome_do_usu\xC3\xA1rio$`

- 拡張 ASCII 文字を含むセーフオブジェクト式。

  エンコードされていない式 [A-Z] {3,6} ¥[1-9][0-9]{6,6}

  エンコードされた式:[A-Z] {3,6}\ xC2\ xA5[1-9][0-9]{6,6}

Unicode 文字セット全体と一致する UTF-8 エンコーディングを含む多くのテーブルがインターネット上で見つけることができます。この情報を含む便利な Web サイトは、次の URL にあります。

`http://www.utf8-chartable.de/unicode-utf8-table.pl`

この Web サイトの表に記載されている文字を正しく表示するには、適切な Unicode フォントがコンピュータにインストールされている必要があります。そうしないと、キャラクターの視覚的な表示に誤りがある可能性があります。ただし、文字を表示するための適切なフォントがインストールされていない場合でも、この Web ページの説明と UTF-8 および UTF-16 コードは正しいものになります。

反転 **PCRE** 式

パターンを含むコンテンツの照合に加えて、反転 PCRE 式を使用して、パターンを含まないコンテンツも照合できます。式を反転するには、感嘆符 (!) を含めるだけです。式の最初の文字に空白文字が続きます。

> 注**:** 式が感嘆符だけで構成され、その後に何もない場合、感嘆符は反転した式を示す構文ではなく、リテラル文字として扱われます。

次の Web App Firewall コマンドは、反転 PCRE 式をサポートしています。

- 開始 URL (URL)
- 拒否 URL (URL)
- フォームフィールドの一貫性 (フォームアクション URL)
- Cookie の一貫性 (フォームアクション URL)
- クロスサイトリクエストフォージェリ (CSRF) (フォームアクション URL)
- HTML クロスサイトスクリプティング (フォームアクション URL)
- フィールド形式 (フォームアクション URL)
- フィールドタイプ (タイプ)
- 機密フィールド (URL)

注: セキュリティチェックに isRegex フラグまたはチェックボックスが含まれている場合は、[YES] に設定するか、オンにしてフィールドで正規表現を有効にする必要があります。それ以外の場合、そのフィールドの内容はリテラルとして扱われ、正規表現 (反転または非反転) は解析されません。

Web App Firewall プロファイルで許可されていない名前

次の名前は、Citrix ADC アプライアンスの組み込みアクションとプロファイルに割り当てられ、ユーザーが作成した Web App Firewall プロファイルの名前として使用することはできません。

- AGRESSIVE
- ALLOW
- BASIC
- CLIENTAUTH
- COMPRESS
- CSSMINIFY
- DEFLATE
- DENY
- DNS-NOP
- DROP
- GZIP
- HTMLMINIFY
- IMGOPTIMIZE
- JSMINIFY
- MODERATE
- NOCLIENTAUTH
- NOCOMPRESS
- NONE

- NOOP
- NOREWRITE
- RESET
- SETASLEARNNSLOG_ACT
- SETNSLOGPARAMS_ACT
- SETSYSLOGPARAMS_ACT
- SETTMSESSSPARAMS_ACT
- SETVPNPARAMS_ACT
- PREAUTHPARAMS_ACT
- default_DNS64_action
- Dns_default_act_cacheBypass
- dns_default_act_drop
- nshttp_default_profile
- nshttp_default_strict_validation
- nstcp_default_mobile_profile
- nstcp_default_xd_profile
- nstcp_default_profile
- nstcp_default_tcp_interactive_stream
- nstcp_default_tcp_lan
- nstcp_default_tcp_lan_thin_stream
- nstcp_default_tcp_lfp
- nstcp_default_tcp_lfp_thin_stream
- nstcp_default_tcp_lnp
- nstcp_default_tcp_lnp_thin_stream
- nstcp_internal_apps

HTML、**XML**、および **JSON** エラーオブジェクトのカスタムエラーステータスとメッセージ

October 7, 2021

Citrix Web App Firewall が違反を検出すると、アプライアンスはリダイレクト URL またはエラーオブジェクト（プロファイルにインポートされ、有効化）を使用してエラーシナリオを処理します。シナリオがエラーオブジェクト設定を使用して処理される場合、WAF プロファイルはカスタム応答ステータスコードとメッセージを提供します。WAF プロファイルの HTML、XML、または JSON エラーオブジェクトの応答エラーの詳細をカスタマイズできます。

> 注:
>
> デフォルトでは、エラーオブジェクトの設定が構成されている場合、エラーコードとエラーメッセージは「200」

> および「OK」に設定されます。

エラーシナリオを処理する場合、問題を解決するために、アプライアンスが適切な HTTP 応答ステータスコードとメッセージで応答することが重要です。カスタムのエラーステータスメッセージとカスタムエラーステータスコードを提供することにより、アプライアンスは、違反が発生したときに問題を解決するためのユーザー介入を向上させることができます。たとえば、応答エラーコードを「404」に設定し、ステータスメッセージを「Not Found」に設定すると、ユーザーは応答ステータスコードとメッセージを検査して、違反が発生したかどうかを確認できます。これにより、エラーオブジェクトを含む応答をフィルタリングできます。

CLI を使用して、**WAF** プロファイル内の **HTML** エラーオブジェクトのカスタムステータスコードとメッセージを設定します

コマンドプロンプトで入力します。

```
1 set appfw profile <profile-name> -HTMLErrorStatusCode <value> -
     HTMLErrorStatusMessage <value> -useHTMLErrorObject ON
2 <!--NeedCopy-->
```

例:

```
set appfw profile profile_1 -HTMLErrorStatusCode 404 -HTMLErrorStatusMessage
  “Not Found”  -useHTMLErrorObject ON
```

CLI を使用して、**WAF** プロファイルの **XML** エラーオブジェクトのカスタムステータスコードとメッセージを設定する

コマンドプロンプトで入力します。

```
1 set appfw profile <profile-name> -XMLErrorStatusCode <value> -
     XMLErrorStatusMessage <value>
2 <!--NeedCopy-->
```

例:

```
set appfw profile profile_1 -XMLErrorStatusCode 406 - XMLErrorStatusMessage
  “Not Acceptable”
```

CLI を使用して **WAF** プロファイルの **JSON** エラーオブジェクトのカスタムステータスコードとメッセージを構成する

コマンドプロンプトで入力します。


```
set appfw profile <profile-name> -JSONErrorStatusCode <value> -
    JSONErrorStatusMessage <value>
<!--NeedCopy-->
```

例:

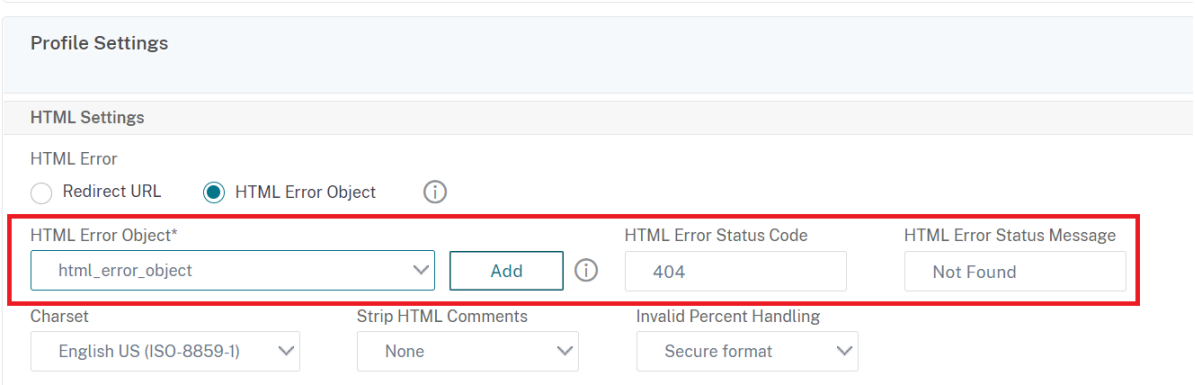
```
set appfw profile profile_1 -JSONErrorStatusCode 500 - JSONErrorStatusMessage
  “Internal Server Error”
```

GUI を使用して **WAF** プロファイルの **HTML**、**JSON**、または **XML** エラーオブジェクトのカスタムステータスコードとメッセージを構成します

1. **Security > Citrix Web App Firewall > Profiles** に移動します。
2. 詳細ウィンドウで、[編集] をクリックします。
3. [**Web App Firewall** プロファイルの作成] ページで、[詳細設定] セクションの **[**プロファイル設定**]** をクリックします。
4. [プロファイル設定] セクションで、次のパラメータを設定します。

   a. HTML エラーオブジェクト。HTML エラーオブジェクトを使用してエラーシナリオを処理するオプションを選択します。URL、ファイル、またはテキストからエラーオブジェクトをインポートします。
   b. HTML エラーステータスコード。カスタムエラーステータスコードを指定します。
   c. HTML エラーステータスメッセージ。カスタマーのエラーメッセージを入力します。
5. [**OK**] をクリックし、[完了] をクリックします。

> 注:
>
> JSON および XML カスタムエラーオブジェクト設定にも同じ手順が適用されます。



ポリシーラベル

October 13, 2021

ポリシーラベルは、一連のポリシー、その他のポリシーラベル、および仮想サーバ固有のポリシーバンクで構成されます。Web App Firewall は、ポリシーラベルにバインドされた各ポリシーを優先度順に評価します。ポリシーが一致すると、関連付けられたプロファイルに指定されているとおりに接続がフィルタリングされます。次に、Goto パラメータが指定するものは何でも行います。このパラメータは、ポリシー評価を終了するか、次のポリシーに移動するか、指定された優先度を持つポリシーに移動できます。Invoke パラメータが設定されている場合、現在のポリシーラベルの処理が終了し、指定したポリシーラベルまたは仮想サーバの処理が開始されます。

コマンドラインを使用して **Web App Firewall** ポリシーラベルを作成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw policylabel <labelName> http_req`
- `save ns config`

例

次の例では、policylbl1 という名前のポリシーラベルを作成します。

```
1 add appfw policylabel policylbl1 http_req
2 save ns config
3 <!--NeedCopy-->
```

コマンドラインを使用してポリシーをポリシーラベルにバインドするには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw policylabel <labelName> <policyName> <priority> [<gotoPriorityExpressi >] [-invoke (<labelType> <labelName>)]`
- `save ns config`

例

次の例では、ポリシー policy1 を、プライオリティ 1 のポリシーラベル policylbl1 にバインドします。


```
bind appfw policylabel policylbl1 policy1 1
save ns config
<!--NeedCopy-->
```

GUI を使用して Web App Firewall ポリシーラベルを構成するには

1. **Security > Citrix Web App Firewall > Policy Labels** に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいポリシーラベルを追加するには、[**Add**] をクリックします。
   - 既存のポリシーラベルを設定するには、ポリシーラベルを選択し、[**Open**] をクリックします。

   [Web App Firewall ポリシーラベルの作成] ダイアログボックスまたは [**Web App Firewall** ポリシーラベルの設定] ダイアログボックスが開きます。ダイアログボックスはほぼ同じです。
3. 新しいポリシーラベルを作成する場合は、[Web App Firewall ポリシーラベルの作成] ダイアログボックスで、新しいポリシーラベルの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～127 の文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、およびアンダースコア (_) 記号で構成できます。
4. [ポリシーの挿入**]** を選択して新しい行を挿入し、既存のすべての Web AppFirewall ポリシーを含むドロップダウンリストを表示します。
5. ポリシーラベルにバインドするポリシーを選択するか、[New Policy] を選択して新しいポリシーを作成し、GUI を使用してポリシーを作成および構成するにはの手順に従います。選択または作成したポリシーが、グローバルにバインドされた Web App Firewall ポリシーのリストに挿入されます。
6. 追加の調整を行います。
   - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。[優先度を再生成] を選択して、優先度を均等に再番号付けすることもできます。
   - ポリシー式を変更するには、そのフィールドをダブルクリックして [Web App Firewall Policy] ダイアログボックスを開き、ポリシー式を編集できます。
   - [Goto Expression] を設定するには、[Goto Expression] 列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。
   - 「呼び出し」オプションを設定するには、「呼び出し」列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。
7. 手順 5 ～7 を繰り返して、ポリシーラベルに追加する Web App Firewall ポリシーをバインドします。
8. 「作成」または「OK」をクリックし、「閉じる」をクリックします。ポリシーラベルが正常に作成または変更されたことを示すメッセージがステータスバーに表示されます。

ポリシー

October 7, 2021

Web App Firewall では、ファイアウォールポリシーと監査ポリシーの 2 種類のポリシーが使用されます。ファイアウォールポリシーは、Web App Firewall に送信されるトラフィックを制御します。監査ポリシーは、Web App Firewall ログの送信先となるログサーバーを制御します。

ポリシー規則は、Citrix ADC 式言語の複数の式で構成できるため、ファイアウォールポリシーは複雑になる場合があります。これは、本格的なオブジェクト指向プログラミング言語で、どの接続をフィルタリングするかを正確に正確に定義できるためです。ファイアウォールポリシーは Web App Firewall のコンテキスト内で動作するため、Web App Firewall の機能と、Web App Firewall によって適切にフィルタリングされるトラフィックに関連する特定の基準を満たす必要があります。ただし、これらの条件を念頭に置いておく限り、ファイアウォールポリシーは Citrix ADC の他の機能のポリシーと似ています。ここで説明する手順は、ファイアウォールポリシーを記述するすべての側面をカバーするものではなく、ポリシーの概要を説明し、Web App Firewall に固有の基準をカバーするものです。

ポリシー・ルールは常に ns_true であるため、監査ポリシーは単純です。必要なのは、ログの送信先となるログサーバー、使用するログレベル、および詳細に説明するその他の条件だけです。

Web App Firewall ポリシー

October 7, 2021

ファイアウォールポリシーは、プロファイルに関連付けられた規則です。ルールは、Web App Firewall がプロファイルを適用してフィルタリングする要求/応答ペアのタイプを定義する式または式のグループです。ファイアウォールポリシー式は、Citrix ADC 式言語で記述されます。この言語は、特定の Citrix ADC 機能をサポートするための特別な機能を備えたオブジェクト指向プログラミング言語です。プロファイルは、ルールに一致する要求/応答のペアをフィルタリングするために、Web App Firewall が使用する一連のアクションです。

ファイアウォールポリシーを使用すると、さまざまな種類の Web コンテンツに異なるフィルタリングルールを割り当てることができます。すべての Web コンテンツが似ているわけではありません。複雑なスクリプトを使用せず、個人データにアクセスして処理しない単純な Web サイトでは、基本的なデフォルトで作成されたプロファイルによって提供される保護レベルのみが必要になる場合があります。JavaScript で拡張された Web フォームを含む Web コンテンツや SQL データベースにアクセスする Web コンテンツには、おそらくよりカスタマイズされた保護が必要です。別のプロファイルを作成してそのコンテンツをフィルタリングし、そのコンテンツにアクセスしようとしている要求を決定する別のファイアウォールポリシーを作成できます。次に、作成したプロファイルにポリシー式を関連付け、ポリシーをグローバルにバインドして有効にします。

Web App Firewall は HTTP 接続のみを処理するため、Citrix ADC 表現言語全体のサブセットを使用します。ここでの情報は、Web App Firewall 構成時に役立つ可能性のあるトピックと例に限定されています。ファイアウォールポリシーの追加情報と手順へのリンクを次に示します。

- ポリシーを作成して構成する手順については、「Web App Firewall ポリシーの作成と構成」を参照してください。
- ポリシールール (式) の作成方法を詳しく説明する手順については、「Web App Firewall ルール (式) を作成または構成するには」を参照してください。
- [式の追加] ダイアログボックスを使用してポリシールールを作成する方法については、「[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには、「式の追加」を参照してください。
- ポリシーの現在のバインディングを表示する方法を説明する手順については、ファイアウォールポリシーのバインディングの表示を参照してください。
- Web App Firewall ポリシーをバインドする手順については、「Web App Firewall ポリシーのバインド」を参照してください。
- Citrix ADC 式の言語の詳細については、「ポリシーと式」を参照してください。

> 注
>
> Web App Firewall は、設定された優先順位と goto 式に基づいてポリシーを評価します。ポリシー評価の最後に、true と評価される最後のポリシーが使用され、対応するプロファイルのセキュリティ設定が要求を処理するために呼び出されます。
>
> たとえば、ポリシーが 2 つあるシナリオを考えてみましょう。
>
> - ポリシー _1 は、式が ns_true の汎用ポリシーであり、基本プロファイルである対応するプロファイル _1 を持ちます。プライオリティは 100 に設定されます。
> - ポリシー _2 は、式 =HTTP.REQ.URL.CONTAINS（「XYZ」）でより具体的に指定され、対応するプロファイル _2（事前プロファイルです）があります。移動式は NEXT に設定され、優先度は 95 に設定されます。これは Policy_1 と比較して高い優先度です。
>
> このシナリオでは、処理された要求の URL でターゲット文字列「XYZ」が検出された場合、Policy_1 も一致しているにもかかわらず、高い優先順位を持つように Policy_2 の一致がトリガーされます。ただし、Policy_2 の GoTo 式設定に従って、ポリシー評価は続行され、次のポリシー Policy_1 も処理されます。ポリシー評価の最後に、Policy_1 が true と評価され、Profile_1 で設定された基本的なセキュリティー検査が呼び出されます。
>
> Policy_2 が変更され、GoTo 式が **NEXT** から **END** に変更された場合、ターゲット文字列「XYZ」を持つ処理された要求は、優先順位を考慮して Policy_2 の一致をトリガーし、GoTo 式の設定に従って、ポリシー評価は終了します。この時点で。Policy_2 は true と評価され、Profile_2 で設定された高度なセキュリティー検査が呼び出されます。
>
> **NEXT**
> **END**
>
> ポリシー評価は 1 回のパスで完了します。リクエストに対するポリシー評価が完了し、対応するプロファイルアクションが呼び出されると、リクエストはポリシー評価の別のラウンドを通過しません。

Web App Firewall ポリシーの作成と構成

October 7, 2021

ファイアウォールポリシーは、ルールと関連するプロファイルの 2 つの要素で構成されます。ルールは、設定した条件に一致する HTTP トラフィックを選択し、そのトラフィックを Web App Firewall に送信してフィルタリングします。プロファイルには、Web App Firewall が使用するフィルタリング基準が含まれています。

ポリシールールは、Citrix ADC 式言語の 1 つ以上の式で構成されます。Citrix ADC 式の構文は、特定のプロファイルで処理するトラフィックを正確に指定できる強力なオブジェクト指向プログラミング言語です。Citrix ADC 式の言語構文に慣れていないユーザーや、Web ベースのインターフェイスを使用して Citrix ADC アプライアンスを構成したいユーザーには、****GUI** には接頭辞メニューと式の追加ダイアログボックスの **2** つのツールが用意されています** 。どちらも、処理するトラフィックを正確に選択する式を記述するのに役立ちます。構文に精通している経験豊富なユーザーは、Citrix ADC コマンドラインを使用して Citrix ADC アプライアンスを構成することをお勧めします。

> 注**:**
>
> デフォルトの式の構文に加えて、下位互換性のために、Citrix ADC オペレーティングシステムは、Citrix ADC Classic、nCore アプライアンスと仮想アプライアンスでの Citrix ADC クラシック式構文をサポートしています。クラシック式は、Citrix ADC Cluster アプライアンスおよび仮想アプライアンスではサポートされていません。既存の構成を Citrix ADC クラスターに移行する現在の Citrix ADC ユーザーは、従来の式を含むポリシーをデフォルトの式の構文に移行する必要があります。

Citrix ADC 式の言語の詳細については、「ポリシーと式」を参照してください。

ファイアウォールポリシーは、GUI または Citrix ADC コマンドラインを使用して作成できます。

コマンドラインインターフェイスを使用してポリシーを作成して構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw policy <name><rule> <profileName>`
- `save ns config`

例

次の例では、ホスト blog.example.com との間で送受信されるすべてのトラフィックをインターセプトするルールを使用して、pl-blog という名前のポリシーを追加し、そのポリシーをプロファイル pr-blog に関連付けます。これは、特定のホスト名でホストされているブログを保護するための適切なポリシーです。

```
1 add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com
     ")" pr-blog
2 <!--NeedCopy-->
```


GUI を使用してポリシーを作成して構成するには

1. [セキュリティ] > [**Web App Firewall**] > [ポリシー] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - ファイアウォールポリシーを作成するには、[追加] をクリックします。**[Web App Firewall** ポリシーの作成] が表示されます。
   - 既存のファイアウォールポリシーを編集するには、ポリシーを選択し、[編集] をクリックします。

   [Web App Firewall ポリシーの作成**]** または **[Web App Firewall** ポリシーの構成] が表示されます。
3. ファイアウォールポリシーを作成する場合は、[**Web App Firewall** ポリシーの作成] ダイアログボックスの [ポリシー名] テキストボックスに、新しいポリシーの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～128 文字、数字、ハイフン（-）、ピリオド（.）ポンド（#）、スペース（）、アットマーク（@）、等号（=）、コロン（:）、およびアンダースコア（_）記号で構成されます。

   既存のファイアウォールポリシーを構成している場合、このフィールドは読み取り専用です。これは変更できません。
4. [プロファイル（Profile）] ドロップダウンリストから、このポリシーに関連付けるプロファイルを選択します。[新規] をクリックすると、ポリシーに関連付けるプロファイルを作成でき、[Modify] をクリックして既存のプロファイルを変更できます。
5. [式] テキスト領域で、ポリシーのルールを作成します。
   - テキスト領域に直接ルールを入力できます。
   - [Prefix] をクリックしてルールの最初の用語を選択し、プロンプトに従って操作できます。
   - [追加] をクリックして [式の追加] ダイアログボックスを開き、それを使用してルールを作成できます。
6. 「作成」または「**OK**」をクリックし、「閉じる」をクリックします。

Web App Firewall ルール **(**式**)** を作成または構成するには

ポリシールールは、式とも呼ばれ、ポリシーに関連付けられたプロファイルを使用して Web App Firewall がフィルタリングする Web トラフィックを定義します。他の Citrix ADC ポリシールール（または式）と同様に、Web App Firewall ルールは Citrix ADC 式の構文を使用します。この構文は強力で柔軟性があり、拡張可能です。この一連の命令で完全に記述するには複雑すぎます。次の手順を使用して単純なファイアウォールポリシールールを作成するか、ポリシー作成プロセスの概要として読むことができます。

1. まだ行っていない場合は、**Web App Firewall** ウィザードまたは **Citrix** ADC GUI で適切な場所に移動し、ポリシールールを作成します。
   - **Web App Firewall** ウィザードでポリシーを構成する場合は、ナビゲーションペインで [**Web App Firewall**] をクリックし、詳細ウィンドウで [**Web App Firewall** ウィザード] をクリックし、[ルールの指定] 画面に移動します。

- ポリシーを手動で構成する場合は、ナビゲーションペインで、[**Web App Firewall]**、**[**** ポリシー]、[ファイアウォール**]** の順に展開します。詳細ウィンドウで、ポリシーを作成するには、**[**** 追加] をクリックします。既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。

2. [ルールの指定] 画面、[**Web App Firewall** プロファイルの作成] ダイアログボックス、または [**Web App Firewall** プロファイルの構成] ダイアログボックスで、[接頭辞] をクリックし、ドロップダウンリストから式のプレフィックスを選択します。選択肢は次のとおりです:

   - **HTTP** プロトコルに関連するリクエストの側面を調べる場合は、HTTP プロトコルを選択します。
   - **SYS**。リクエストの受信者に関連するリクエストの側面を調べたい場合は、保護された Web サイトを選択します。
   - クライアント。リクエストを送信したクライアントを選択します。リクエストの送信者の側面を調べる場合は、これを選択します。
   - サーバー。リクエストの送信先クライアントを選択し、リクエストの受信者の側面を調べるかどうかを指定します。

   プレフィックスを選択すると、Web App Firewall に 2 つの部分からなるプロンプトウィンドウが表示され、次に選択可能な選択肢が一番下部に表示されます。

3. 次の用語を選択してください。

   HTTP プロトコルをプレフィクスとして選択した場合は、ReQ しか選択できません。REQ は Request/Response のペアを指定します。(Web App Firewall は、リクエストとレスポンスをそれぞれ別々ではなくユニットとして動作します)。別のプレフィックスを選択した場合、選択肢はより多様になります。特定の選択肢に関するヘルプを表示するには、その選択肢を 1 回クリックすると、その選択に関する情報が下のプロンプトウィンドウに表示されます。

   使用する用語が決まったら、ダブルクリックして [式] ウィンドウに挿入します。

4. 選択した期間の後にピリオドを入力します。次に、前の手順で説明したように、次の用語を選択するように求められます。用語で値を入力する必要がある場合は、適切な値を入力します。たとえば、HTTP.REQ.HEADER (「」) を選択した場合、引用符の間にヘッダー名を入力します。

5. 式が終了するまで、プロンプトから用語を選択し、必要な値を入力します。

   特定の目的のための式の例をいくつか挙げます。

   - 特定の **Web** ホスト。特定の Web ホストからのトラフィックを照合するには、次の手順を実行します。

     ```
     1  HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
     2  <!--NeedCopy-->
     ```

     `shopping.example.com`の場合は、一致させる Web ホストの名前に置き換えます。

   - 特定の **Web** フォルダまたはディレクトリ。Web ホスト上の特定のフォルダまたはディレクトリからのトラフィックを照合するには、次の手順を実行します。


```
1 HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
2 <!--NeedCopy-->
```

www.example.com の場合は、ウェブホストの名前を代用してください。フォルダの場合は、一致させるコンテンツのフォルダまたはパスを置き換えます。たとえば、ショッピングカートが /solutions/orders というフォルダにある場合、その文字列をフォルダに置き換えます。

- 特定の種類のコンテンツ**:GIF** 画像。GIF 形式の画像を一致させるには:

```
1 HTTP.REQ.URL.ENDSWITH(".png")
2 <!--NeedCopy-->
```

他のフォーマットイメージを一致させるには、.png の代わりに別の文字列を置き換えます。

- 特定の種類のコンテンツ**:** スクリプト。CGI-BIN ディレクトリにあるすべての CGI スクリプトを一致させるには、次の手順を実行します。

```
1 HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
2 <!--NeedCopy-->
```

すべての JavaScript を.js 拡張子で一致させるには:

```
1 HTTP.REQ.URL.ENDSWITH(".js")
2 <!--NeedCopy-->
```

ポリシー式の作成の詳細については、「ポリシーと式」を参照してください。

> 注**:**
>
> コマンドラインを使用してポリシーを構成する場合は、Citrix ADC 式内の二重引用符をエスケープしてください。たとえば、GUI で入力すると、次の式は正しいです。

```
1 HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
2 <!--NeedCopy-->
```

ただし、コマンドラインで入力した場合は、代わりに次のコマンドを入力する必要があります。


```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

[式の追加] ダイアログボックスを使用してファイアウォールルール (式) を追加するには

[式の追加] ダイアログボックス（式エディタとも呼ばれる）は、Citrix ADC 式の言語に慣れていないユーザーが、フィルタリングするトラフィックに一致するポリシーを構築するのに役立ちます。

1. まだ行っていない場合は、**Web App Firewall** ウィザードまたは **Citrix** ADC GUI で適切な場所に移動します。
   - **Web App Firewall** ウィザードでポリシーを構成する場合は、ナビゲーションペインで [**Web App Firewall**] をクリックし、詳細ウィンドウで [**Web App Firewall** ウィザード] をクリックし、[ルールの指定] 画面に移動します。
   - ポリシーを手動で構成する場合は、ナビゲーションペインで、[**Web App Firewall]**、**[**** ポリシー]、[ファイアウォール**]** の順に展開します。詳細ウィンドウで、ポリシーを作成するには、**[**** 追加] をクリックします。既存のポリシーを変更するには、ポリシーを選択し、[開く] をクリックします。
2. [ルールの指定] 画面の [**Web App Firewall** プロファイルの作成] ダイアログボックス、または [**Web App Firewall** プロファイルの構成] ダイアログボックスで、[追加] をクリックします。
3. **[**式の追加**]** ダイアログボックスの **[** 式の構成] 領域の最初のリストボックスで、次のいずれかの接頭辞を選択します。
   -**HTTP**HTTP プロトコルに関連するリクエストの側面を調べる場合は、[HTTP プロトコル] を選択します。デフォルトの選択肢。
   -**SYS**。リクエストの受信者に関連するリクエストの側面を調べたい場合は、保護された Web サイトを選択します。
   -クライアント。リクエストの送信者の側面を調べる場合は、要求を送信したコンピュータを選択します。
   -サーバー。要求の送信先コンピュータを選択し、要求の受信者の側面を調べます。
4. 2 番目のリストボックスで、次の用語を選択します。ダイアログボックスでは、コンテキストに有効な用語のみが含まれるようにリストが自動的に調整されるため、使用可能な用語は、前の手順で行った選択によって異なります。たとえば、前のリストボックスで [HTTP] を選択した場合、リクエストに対する唯一の選択肢は [REQ] です。Web App Firewall は、リクエストと関連する応答を 1 つのユニットとして扱い、両方をフィルタリングするため、個別にレスポンスを指定する必要はありません。2 番目の用語を選択すると、第 2 項の右に 3 番目のリストボックスが表示されます。[ヘルプ] ウィンドウには 2 番目の項の説明が表示され、[式のプレビュー] ウィンドウにはエクスプレッションが表示されます。
5. 3 番目のリストボックスで、次の用語を選択します。右側に新しいリストボックスが表示され、ヘルプウィンドウが新しい用語の説明が表示されます。エクスプレッションのプレビュー (Preview Expression) ウィンドウが更新され、指定したポイントまでのエクスプレッションが表示されます。
6. 条件の選択を続け、引数の入力を求められたら、式が完成します。既に用語を選択した後に間違えたり、式を変更したい場合は、単に別の用語を選択できます。式が変更され、変更した用語の後に追加した引数またはそれ以上の用語がクリアされます。

7. 式の作成が完了したら、「**OK**」をクリックして「式の追加」ダイアログボックスを閉じます。式が [式] テキスト領域に挿入されます。

Web App Firewall ポリシーのバインド

October 7, 2021

Web App Firewall ポリシーを設定したら、それらをグローバルまたはバインドポイントにバインドして有効にします。バインド後、Web App Firewall ポリシーに一致する要求または応答は、そのポリシーに関連付けられたプロファイルによって変換されます。

ポリシーをバインドするときは、そのポリシーに優先度を割り当てます。優先順位によって、定義したポリシーが評価される順序が決まります。プライオリティは任意の正の整数に設定できます。Citrix ADC OS では、ポリシーの優先順位は逆の順序で機能します。値が大きいほど、優先順位は低くなります。

Web App Firewall 機能では、リクエストが一致する最初のポリシーだけが実装され、一致する可能性のある追加のポリシーは実装されないため、意図した結果を得るにはポリシーの優先順位が重要です。最初のポリシーに低い優先度（1000 など）を設定した場合、優先度の高い他のポリシーが要求と一致しない場合にのみ、Web App Firewall を実行するように設定します。最初のポリシーに高い優先度（1 など）を与える場合は、Web App Firewall を最初に実行するように設定し、一致する可能性のあるその他のポリシーをスキップします。ポリシーをバインドするときに、各ポリシー間に 50 または 100 の間隔で優先順位を設定することで、優先順位を再割り当てすることなく、他のポリシーを任意の順序で追加できる十分な余裕を残すことができます。

Citrix ADC アプライアンスでのポリシーのバインドの詳細については、「ポリシーと式」を参照してください。

コマンドラインインターフェイスを使用して **Web App Firewall** ポリシーをバインドするには

コマンドプロンプトで、次のコマンドを入力します。

- `bind appfw global <policyName>`
- `bind appfw profile <profile_name> -crossSiteScripting data`

例

次の例では、pl-blog という名前のポリシーをバインドし、プライオリティ 10 を割り当てます。

```
1 bind appfw global pl-blog 10
2 save ns config
3 <!--NeedCopy-->
```


ログ式の構成

Web App Firewall をバインドするためのログ式サポートは、違反が発生したときに HTTP ヘッダー情報をログに記録するために追加されます。

ログ式はアプリケーションプロファイルにバインドされ、バインディングには、違反が発生したときに評価され、ロギングフレームワークに送信される必要がある式が含まれています。

http ヘッダー情報を持つ Web App Firewall 違反ログレコードが記録されます。カスタムログ式を指定することができ、現在のフロー（要求/応答）に対して違反が生成されたときの分析と診断に役立ちます。

設定例

```
bind appfw profile <profile> -logexpression <string> <expression>
add policy expression headers "" HEADERS(100):"+HTTP.REQ.FULL_HEADER"
add policy expression body_100 ""BODY:"+HTTP.REQ.BODY(100)"
bind appfw profile test -logExpression log_body body_100
bind appfw profile test -logExpression log_headers  headers
bind appfw profile test -logExpression ""URL:"+HTTP.REQ.URL+" IP:"+
    CLIENT.IP.SRC"
<!--NeedCopy-->
```

ログの例

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg= HEADERS(100)
    :POST /test/credit.html HTTP/1.1^M User-Agent: curl/7.24.0 (amd64-
    portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Host:
     10.217.222.44^M Accept: /^M Content-Length: 33^M Content-Type:
    application/x-www-form-urlencoded^M ^M  cn1=58 cn2=174 cs1=test cs2=
    PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=BODY:ata=
    asdadasdasdasddddddddddddddddd cn1=59 cn2=174 cs1=test cs2=PPE1 cs4=
    ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=URL:/test/
    credit.html IP:10.217.222.128 cn1=60 cn2=174 cs1=test cs2=PPE1 cs4=
    ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Other violation logs
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_STARTURL|6|src=10.217.222.128 spt=26409 method=POST
    request=http://10.217.222.44/test/credit.html msg=Disallow Illegal
    URL. cn1=61 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not
    blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_SAFECOMMERCE|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=Maximum
    number of potential credit card numbers seen cn1=62 cn2=174 cs1=test
     cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

> 注
>
> 1. 監査ログのサポートのみ利用可能です。ログストリームのサポートとセキュリティ洞察の可視性は、将来のリリースバージョンで追加される予定です。
> 2. 監査ログが生成される場合、ログメッセージごとに 1024 バイトのデータしか生成できません。
> 3. ログストリーミングを使用する場合、制限は、サポートされるログストリーム/ipfix プロトコルのサイズ制限の最大サイズに基づきます。ログストリームの最大サポートサイズは 1024 バイトを超えています。

GUI を使用して Web App Firewall ポリシーをバインドするには

1. 次のいずれかを行います:
   - [セキュリティ] > [**Web App Firewall**] に移動し、詳細ペインで [Web アプリケーションファイアウォールポリシーマネージャー] をクリックします。
   - **Security** > **Web App Firewall** > **Policies** > **Firewall Policies** に移動し、詳細ペインで **Policy Manager** を選択します。

2. [**Web App Firewall Policy Manager**] ダイアログで、ポリシーをバインドするバインドポイントをドロップダウンリストから選択します。選択肢は次のとおりです。
   - [グローバルをオーバーライド**]**。このバインドポイントにバインドされたポリシーは、Citrix ADC アプライアンスのすべてのインターフェイスからのすべてのトラフィックを処理し、他のポリシーの前に適用されます。
   - **LB** 仮想サーバー。負荷分散仮想サーバーにバインドされたポリシーは、その負荷分散仮想サーバーによって処理されるトラフィックにのみ適用され、Default Global ポリシーの前に適用されます。[LB Virtual Server] を選択した後、このポリシーをバインドする特定の負荷分散仮想サーバーも選択する必要があります。
   - **CS** 仮想サーバ。コンテンツスイッチ仮想サーバーにバインドされたポリシーは、そのコンテンツスイッチ仮想サーバーによって処理されるトラフィックにのみ適用され、デフォルトグローバルポリシーの前に適用されます。[CS Virtual Server] を選択した後、このポリシーをバインドする特定のコンテンツスイッチ仮想サーバーも選択する必要があります。
   - デフォルトグローバル。このバインドポイントにバインドされたポリシーは、Citrix ADC アプライアンスのすべてのインターフェイスからのすべてのトラフィックを処理します。
   - ポリシーラベル。ポリシーラベルにバインドされたポリシーは、ポリシーラベルがルーティングするトラフィックを処理します。ポリシーラベルは、このトラフィックにポリシーを適用する順序を制御します。
   - なし。ポリシーをバインドポイントにバインドしないでください。
3. ［続行］をクリックします。既存の Web App Firewall ポリシーのリストが表示されます。
4. バインドするポリシーをクリックして選択します。
5. バインドに追加の調整を行います。
   - ポリシーの優先度を変更するには、フィールドをクリックして有効にし、新しい優先度を入力します。[優先度を再生成] を選択して、優先度を均等に再番号付けすることもできます。
   - ポリシー式を変更するには、そのフィールドをダブルクリックして [**Web App Firewall Policy**] ダイアログボックスを開き、ポリシー式を編集できます。
   - [Goto Expression] を設定するには、[Goto Expression] 列見出しの フィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。
   - 「呼び出し」オプションを設定するには、「呼び出し」列見出しのフィールドをダブルクリックしてドロップダウンリストを表示し、式を選択できます。
6. 手順 3 ～6 を繰り返して、グローバルにバインドする Web App Firewall ポリシーを追加します。
7. ［**OK**］をクリックします。ポリシーが正常にバインドされたことを示すメッセージがステータスバーに表示されます。

ポリシーバインディングの表示

October 7, 2021

GUI でバインディングを表示することで、ファイアウォールポリシーにどのバインディングが設定されているかをす

ばやく確認することができます。

Web App Firewall ポリシーのバインディングを表示するには

1. **Security > Citrix Web App Firewall > Policies > Firewall Policies** に移動します
2. 詳細ウィンドウで、確認するポリシーを選択し、[バインドの表示] をクリックします。[ポリシーのバインドの詳細: ポリシー] メッセージボックスが表示され、選択したポリシーのバインドの一覧が表示されます。
3. ［閉じる］をクリックします。

Web App Firewall ポリシーに関する補足情報

October 7, 2021

以下は、Web App Firewall を管理するシステム管理者が知っておく必要のある Web App Firewall ポリシーの特定の側面に関する補足情報です。

正しいが予期しない動作

Web アプリケーションのセキュリティと最新の Web サイトは複雑です。多くの場合、Citrix ADC ポリシーにより、ポリシーに精通しているユーザーとは異なる状況で、Web App Firewall の動作が通常どおり異なる場合があります。Web App Firewall が予期せぬ方法で動作する場合を次に示します。

- 欠落した **HTTP** ホストヘッダーと絶対 **URL** を持つリクエスト。ユーザーがリクエストを送信すると、ほとんどの場合、リクエスト URL は相対 URL です。つまり、リファラー URL（リクエストを送信するときにユーザーのブラウザが配置されている URL）を起点として取ります。リクエストがホストヘッダーなしで相対 URL を使用して送信された場合、リクエストは HTTP 仕様に違反しているため、またホストを指定できないリクエストが状況によっては攻撃を構成する可能性があるため、通常はブロックされます。ただし、要求が絶対 URL で送信された場合、Host ヘッダーが欠落している場合でも、要求は Web App Firewall をバイパスし、Web サーバーに転送されます。このような要求は HTTP 仕様に違反しますが、絶対 URL にはホストが含まれているため、脅威は発生しません。

監査ポリシー

October 7, 2021

監査ポリシーによって、Web App Firewall セッション中に生成および記録されるメッセージが決まります。メッセージは SYSLOG 形式でローカル NSLOG サーバまたは外部ロギングサーバに記録されます。選択したロギングのレベルに基づいて、さまざまなタイプのメッセージがログに記録されます。

監査ポリシーを作成するには、まず NSLOG サーバまたは SYSLOG サーバを作成する必要があります。次に、ポリシーを作成し、ログの種類とログの送信先サーバーを指定します。

コマンドラインインターフェイスを使用して監査サーバーを作成するには

NSLOG サーバと SYSLOG サーバの 2 種類の監査サーバを作成できます。コマンド名は異なりますが、コマンドのパラメータは同じです。

監査サーバーを作成するには、コマンドプロンプトで次のコマンドを入力します。

- `add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat (MMDDYYYY | DDMMYYYY)] [-logFacility <logFacility>] [-tcp (NONE | ALL)] [-acl (ENABLED | DISABLED)] [-timeZone (GMT_TIME | LOCAL_TIME)] [-userDefinedAuditlog (YES | NO)] [-appflowExport (ENABLED | DISABLED)]`
- `save ns config`

例

次の例では、IP 10.124.67.91 に syslog1 という名前の syslog サーバを作成し、ログファシリティを緊急、重大、および警告のログファシリティを LOCAL1 に設定し、すべての TCP 接続をログに記録します。

```
1 add audit syslogAction syslog1 10.124.67.91 -logLevel emergency
     critical warning -logFacility
2 LOCAL1 -tcp ALL
3 save ns config
4 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して監査サーバーを変更または削除するには

- 監査サーバーを変更するには、set audit<type> コマンド、監査サーバーの名前、変更するパラメーターを新しい値とともに入力します。
- 監査サーバーを削除するには、rm audit<type> コマンドと監査サーバーの名前を入力します。

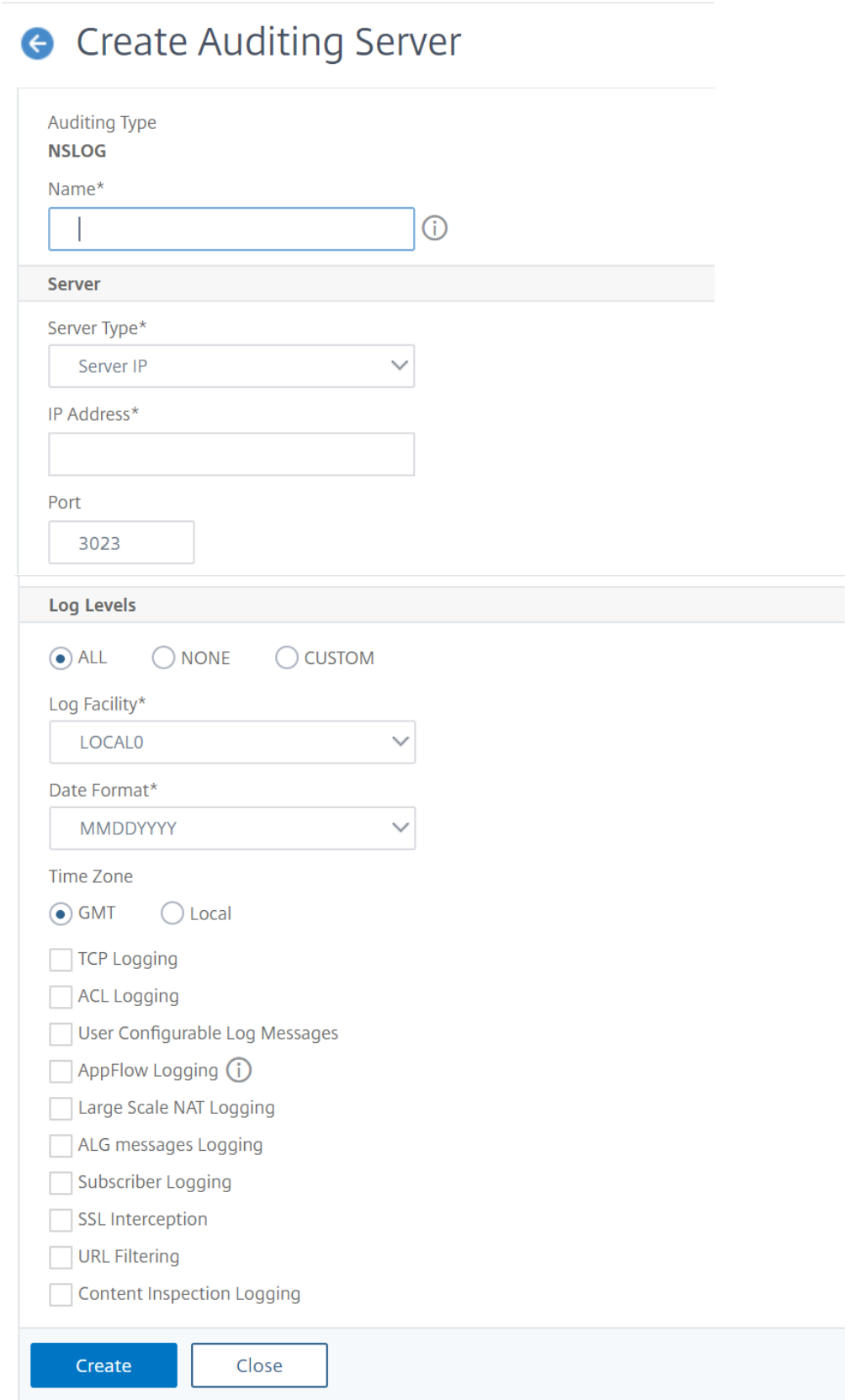
例

次の例では、syslog1 という名前の syslog サーバを変更して、エラーとアラートをログレベルに追加しています。


```
set audit syslogAction syslog1 10.124.67.91 -logLevel emergency
    critical warning alert error
-logFacility LOCAL1 -tcp ALL
save ns config
<!--NeedCopy-->
```

GUI を使用して監査サーバーを作成または構成するには

1. ［セキュリティ］ > ［**Citrix Web App Firewall**］ > ［ポリシー］ > ［監査］ > ［**Nslog**］の順に選択します。
2. [Nslog 監査] ページで、[サーバー] タブをクリックします。
3. 次のいずれかを行います：
   - 新しい監査サーバーを追加するには、[追加] をクリックします。
   - 既存の監査サーバーを変更するには、サーバーを選択し、[編集] をクリックします。
4. 「監査サーバーの作成」ページで、次のパラメータを設定します。
   - 名前
   - サーバーの種類
   - IP アドレス
   - ポート
   - ログレベル
   - ログ機能
   - 日付の書式
   - タイムゾーン
   - TCP ロギング
   - ACL ロギング
   - ユーザー設定可能なログ・メッセージ
   - AppFlow ログ
   - 大規模な NAT ロギング
   - ALG メッセージロギング
   - サブスクライバロギング
   - SSL インターセプション
   - URL フィルタリング
   - コンテンツ検査ロギング
5. **[**作成**]** して **[**閉じる**]** をクリックします。



コマンドラインインターフェイスを使用して監査ポリシーを作成するには

NSLOG ポリシーまたは SYSLOG ポリシーを作成できます。ポリシーのタイプは、サーバのタイプと一致する必要があります。2 つのタイプのポリシーのコマンド名は異なりますが、コマンドのパラメータは同じです。

コマンドプロンプトで、次のコマンドを入力します。

- `add audit syslogPolicy <name> <-rule > <action>`
- `save ns config`

例

次の例では、syslogP1 という名前のポリシーを作成して、Web App Firewall トラフィックを syslog1 という名前の syslog サーバにログに記録します。

```
add audit syslogPolicy syslogP1 rule ”ns_true”action syslog1
save ns config
```

コマンドラインインターフェイスを使用して監査ポリシーを構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `set audit syslogPolicy <name> [-rule <expression>] [-action <string>]`
- `save ns config`

例

次の例では、syslogP1 という名前のポリシーを変更して、Web App Firewall トラフィックを syslog2 という名前の syslog サーバにログに記録します。

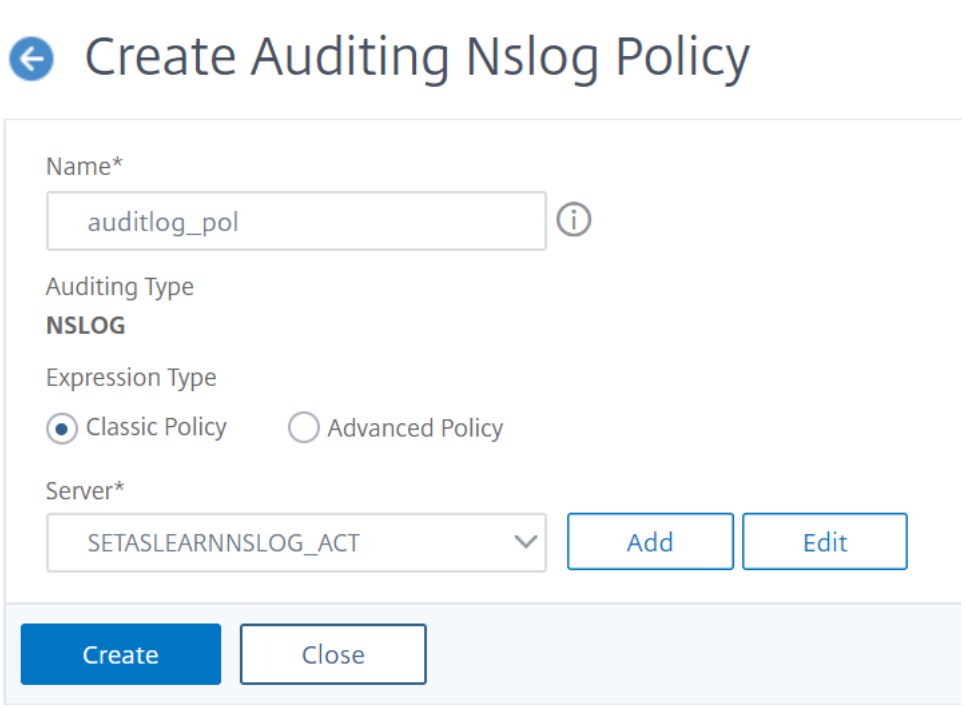
```
set audit syslogPolicy syslogP1 rule ”ns_true”action syslog2
save ns config
```

GUI を使用して監査ポリシーを構成するには

1. セキュリティ > **Citrix Web App Firewall** > ポリシーに移動します。
2. 詳細ウィンドウで、[**Nslog** ポリシーの監査] をクリックします。
3. [Nslog 監査] ページで、[ポリシー] タブをクリックし、次のいずれかの操作を行います。
   - 新しいポリシーを追加するには、[**Add**] をクリックします。
   - 既存のポリシーを変更するには、ポリシーを選択し、[**Edit**] をクリックします。
4. [監査の **Nslog** ポリシーの作成] ページで、次のパラメータを設定します。
   - 名前

- 監査タイプ
- 式タイプ
- サーバー

5. ［作成］をクリックします。



インポート

October 7, 2021

Web App Firewall の一部の機能では、Web App Firewall の設定時にアップロードする外部ファイルが利用されます。GUI を使用して、「インポート」区画でこれらのファイルを管理します。この区画には、HTML エラーオブジェクト、XML エラーオブジェクト、XML スキーマおよび Web サービス記述言語 (WSDL) ファイルの 4 種類のファイルに対応する 4 つのタブがあります。Citrix ADC コマンドラインを使用すると、これらの種類のファイルをインポートできますが、エクスポートすることはできません。

HTML エラーオブジェクト

HTML または Web 2.0 ページへのユーザーの接続がブロックされた場合、またはユーザーが存在しない HTML または Web 2.0 ページを要求した場合、Web App Firewall は HTML ベースのエラー応答をユーザーのブラウザーに送信します。Web App Firewall が使用する必要のあるエラー応答を構成する場合、2 つの選択肢があります。

- リダイレクト URL を設定できます。リダイレクト URL は、ユーザーがアクセスできる任意の Web サーバーでホストできます。たとえば、Web サーバーに 404.html というカスタムエラーページがある場合、接続がブロックされたときにユーザーをそのページにリダイレクトするように Web App Firewall を構成できます。
- HTML エラーオブジェクトを設定できます。HTML エラーオブジェクトは、Web アプリケーションファイアウォール自体でホストされる HTML ベースの Web ページです。このオプションを選択した場合は、HTML エラーオブジェクトを Web App Firewall にアップロードする必要があります。この操作は、[インポート] ペインの [HTML エラーオブジェクト] タブで行います。

エラーオブジェクトは、Web App Firewall エラーオブジェクトのカスタマイズ変数を除き、HTML 以外の構文を含まない標準の HTML ファイルである必要があります。CGI スクリプト、サーバー解析コード、または PHP コードを含めることはできません。カスタマイズ変数を使用すると、要求がブロックされたときにユーザーが受け取るエラーオブジェクトにトラブルシューティング情報を埋め込むことができます。Web App Firewall がブロックするほとんどのリクエストは不正ですが、適切に構成された Web App Firewall でさえ、特に最初にデプロイしたとき、または保護された Web サイトに大幅な変更を加えた後、正当なリクエストをブロックすることがあります。エラーページに情報を埋め込むことで、問題を修正できるように、テクニカルサポート担当者に提供する必要がある情報をユーザーに提供します。

Web App Firewall エラーページのカスタマイズ変数は次のとおりです。

- ${NS_TRANSACTION_ID}. Web App Firewall がこのトランザクションに割り当てたトランザクション ID。
- ${NS_APPFW_SESSION_ID}. Web App Firewall セッション ID。
- ${NS_APPFW_VIOLATION_CATEGORY}. 違反した特定の Web App Firewall セキュリティチェックまたはルール。
- ${NS_APPFW_VIOLATION_LOG}. 違反に関連する詳細なエラーメッセージ。
- ${COOKIE 指定の cookie のコンテンツ。<CookieName>には、エラーページに表示する特定の Cookie の名前を置き換えます。トラブルシューティングのために内容を表示したい Cookie が複数ある場合は、このカスタマイズ変数の複数のインスタンスを、それぞれ適切な Cookie 名とともに使用できます。
  注: Cookie 整合性チェックでブロックが有効になっている場合、Web App Firewall がブロックするため、ブロックされた Cookie はエラーページに表示されません。

これらの変数を使用するには、通常のテキスト文字列であるかのように、エラーページオブジェクトの HTML または XML に変数を埋め込みます。エラーオブジェクトがユーザーに表示される場合、カスタマイズ変数ごとに、変数が参照する情報が Web App Firewall に置き換えられます。カスタム変数を使用する HTML エラーページの例を次に示します。

```
1 <!doctype html public "-//w3c//dtd html 4.0//en"> <html> <head> <
    title>Page Not Accessible</title> </head> <body> <h1>Page Not
    Accessible</h1> <p>The page that you accessed is not available. You
     can:</p> <ul> <li>return to the <b><a href="[homePage]">home page
    </a></b>, re-establish your session, and try again, or,</li> <li>
    report this incident to the help desk via <b><a href="mailto:[
```



```
      helpDeskEmailAddress]”>email</a></b> or by calling [
      helpDeskPhoneNumber].</li>  </ul>  <p>If you contact the help desk,
      please provide the following information:</p>  <table cellpadding=8
      width=80%>  <tr><th align=”right” width=30%>Transaction ID:</th><td
      align=”left” valign=”top” width=70%>${
  NS_TRANSACTION_ID }
  </td></tr>  <tr><th align=”right” width=30%>Session ID:</th><td align=
      ”left” valign=”top” width=70%>${
  NS_APPFW_SESSION_ID }
  </td></tr>  <tr><th align=”right” width=30%>Violation Category:</th><
      td align=”left” valign=”top” width=70%>${
  NS_APPFW_VIOLATION_CATEGORY }
  </td></tr>  <tr><th align=”right” width=30%>Violation Log:</th><td
      align=”left” valign=”top” width=70%>${
  NS_APPFW_VIOLATION_LOG }
  </td></tr>  <tr><th align=”right” width=30%>Cookie Name:</th><td align
      =”left” valign=”top” width=70%>${
  COOKIE(”[cookieName]”) }
  </td></tr>  </table>  <body>  <html>
<!--NeedCopy-->
```

このエラーページを使用するには、テキストエディタまたは HTML エディタにコピーします。以下の変数は、Citrix ADC 変数と区別するために角括弧で囲まれた適切なローカル情報に置き換えます。(変更しないままにしておきます。):

- [homePage]。Web サイトのホームページの URL。
- [helpDeskEmailAddress]。ブロッキングインシデントのレポートに使用する電子メールアドレス。
- [helpDeskPhoneNumber]。ブロッキングインシデントを報告するためにユーザにコールする電話番号。
- [cookieName]。エラーページに表示する内容を持つクッキーの名前。

XML エラーオブジェクト

XML ページへのユーザーの接続がブロックされた場合、またはユーザーが存在しない XML アプリケーションを要求した場合、Web App Firewall は XML ベースのエラー応答をユーザーのブラウザーに送信します。エラー応答を構成するには、[インポート] ウィンドウの [XML エラーオブジェクト] タブで、XML ベースのエラーページを Web App Firewall にアップロードします。すべての XML エラー応答は、Web App Firewall でホストされます。XML アプリケーションのリダイレクト URL を設定することはできません。

> メモ:
> HTML エラーオブジェクトと同じカスタマイズ変数を XML エラーオブジェクトで使用できます。

XML スキーマ

Web App Firewall は、XML または Web 2.0 アプリケーションに対するユーザーの要求に対して検証チェックを実行すると、そのアプリケーションの XML スキーマまたはデザイン型ドキュメント (DTD) に対して要求を検証し、スキーマまたは DTD に従わない要求を拒否できます。XML スキーマと DTD はいずれも、特定の XML ドキュメントの構造を記述する標準 XML 設定ファイルです。

WSDL

Web App Firewall は、XML SOAP ベースの Web サービスに対するユーザーの要求に対して検証チェックを実行するときに、その Web サービスの Web サービスタイプ定義 (WSDL) ファイルに対して要求を検証できます。WSDL ファイルは、特定の XML SOAP Web サービスの要素を定義する標準の XML SOAP 設定ファイルです。

ファイルのインポートとエクスポート

October 7, 2021

GUI またはコマンドラインを使用して、HTML または XML エラーオブジェクト、XML スキーマ、DTD、および WSDL を Web App Firewall にインポートできます。これらのファイルをインポートした後、Web ベースのテキスト領域で編集すると、Citrix ADC で直接小さな変更を加えることができます。コンピュータ上でファイルを変更してから再度インポートする必要はありません。最後に、GUI を使用して、これらのファイルをコンピュータにエクスポートしたり、これらのファイルを削除したりできます。

> 注:
>
> コマンドラインを使用して、インポートしたファイルを削除またはエクスポートすることはできません。

コマンドラインインターフェイスを使用してファイルをインポートするには

コマンドプロンプトで、次のコマンドを入力します。

- `import appfw htmlerrorpage <src> <name>`
- `<save> ns config`

例

次の例では、error.html という名前のファイルから HTML エラーオブジェクトをインポートし、HTMLError という名前を割り当てます。

```
1 import htmlerrorpage error.html HTMLError
2 save ns config
```



```
<!--NeedCopy-->
```

GUI を使用してファイルをインポートするには

XML スキーマ、DTD、WSDL ファイル、または HTML または XML エラーオブジェクトをネットワーク上の場所からインポートする前に、Citrix ADC がファイルがあるインターネットまたは LAN コンピューターに接続できることを確認します。そうしないと、ファイルまたはオブジェクトをインポートできません。

1. **Security > Citrix Web App Firewall > Imports** に移動します。
2. 「アプリケーションファイアウォール」>「インポート」に移動します。
3. [アプリケーションファイアウォールのインポート] ウィンドウで、インポートするファイルの種類に対応するタブを選択し、[追加] をクリックします。

   タブは、HTML エラーページ、XML エラーページ、XML スキーマまたは WSDL です。アップロードプロセスは、ユーザーの視点から 4 つのタブすべてで同じです。
4. ダイアログフィールドに入力します。
   - 「名前」**(Name)**-インポートされたオブジェクトの名前。
   - 「インポート元」(**Import From**)-インポートする HTML ファイル、XML ファイル、XML スキーマ、または WSDL の場所をドロップダウンリストから選択します。
     - **URL:** アプライアンスからアクセス可能な Web サイト上の Web URL。
     - ファイル**:** ローカルまたはネットワーク接続されたハードディスク、またはその他のストレージデバイス上のファイル。
     - テキスト**:** カスタム応答のテキストを GUI のテキストフィールドに直接入力するか、貼り付けます。

     3 番目のテキストボックスが適切な値に変わります。次の 3 つの値を指定できます。
   - [**URL**]: テキストボックスに URL を入力します。
   - **[File]**: HTML ファイルへのパスとファイル名を直接入力するか、[Browse] をクリックして HTML ファイルを参照します。
   - **Text**: 3 番目のフィールドが削除され、空白が残ります。
5. [続行] をクリックします。[ファイルの内容] ダイアログボックスが表示されます。「URL」または「ファイル」を選択した場合は、「ファイルの内容」テキストボックスに、指定した HTML ファイルが表示されます。「テキスト」を選択した場合、「ファイル内容」テキストボックスは空になります。
6. 「テキスト」を選択した場合は、インポートするカスタム応答 HTML を入力またはコピーして貼り付けます。
7. [完了] をクリックします。
8. オブジェクトを削除するには、オブジェクトを選択し、[削除] をクリックします。

GUI を使用してファイルをエクスポートするには

XML スキーマ、DTD、WSDL ファイル、または HTML または XML エラーオブジェクトをエクスポートする前に、Web App Firewall アプライアンスがファイルを保存するコンピューターにアクセスできることを確認します。そうしないと、ファイルをエクスポートできません。

1. [セキュリティ] > [**Web App Firewall**] > [インポート] に移動します。
2. [**Web App Firewall** インポート] ペインで、エクスポートするファイルの種類に対応するタブを選択します。

   エクスポートプロセスは、ユーザーの視点から見ると、4 つのタブすべてで同じです。
3. エクスポートするファイルを選択します。
4. 「アクション」ドロップダウンリストを展開し、「エクスポート」を選択します。
5. ダイアログボックスで、「ファイルを保存」を選択し、「**OK**」をクリックします。
6. [参照] ダイアログボックスで、エクスポートしたファイルを保存するローカルファイルシステムおよびディレクトリに移動し、[保存] をクリックします。

GUI で HTML または XML エラーオブジェクトを編集するには

HTML および XML エラーオブジェクトのテキストは、エクスポートしてから再インポートせずに GUI で編集します。

1. **Security > Citrix Web App Firewall > Imports** に移動し、変更するファイルの種類のタブを選択します。
2. [アプリケーションファイアウォール] > [インポート] に移動し、変更するファイルの種類に対応するタブを選択します。
3. 変更するファイルを選択し、[編集] をクリックします。

   HTML または XML エラーオブジェクトのテキストが、ブラウザのテキスト領域に表示されます。テキストは、ブラウザ用の標準のブラウザベースの編集ツールと方法を使用して変更できます。

   メモ: 編集ウィンドウは，HTML または XML エラーオブジェクトを少し変更できるように設計されています。大規模な変更を行うには、エラーオブジェクトをローカルコンピュータにエクスポートし、標準の HTML または XML Web ページ編集ツールを使用します。
4. [**OK**] をクリックし、[**Close**] をクリックします。

グローバル設定

October 7, 2021

Web App Firewall のグローバル設定は、すべてのプロファイルとポリシーに影響します。グローバル構成項目は次のとおりです。

- エンジン設定。特定の接続のサブセットではなく、Web App Firewall が処理するすべての接続に関係するグローバル設定（セッション Cookie 名、セッションタイムアウト、最大セッションライフタイム、ロギングヘッダー名、未定義のプロファイル、デフォルトプロファイル、インポートサイズ制限）。
- 機密フィールド。Web AppFirewall ログに記録してはならない機密情報を含む Web フォームのフォームフィールドのセット。ログオンページのパスワードフィールドやショッピングカートのチェックアウトフォームのクレジットカード情報などのフォームフィールドは、通常、機密フィールドとして指定されます。
- **[**フィールドタイプ**]**。フィールド形式のセキュリティー検査で使用される Web フォームのフィールドタイプのリスト。これらの各フィールドタイプは、データのタイプを定義する PCRE 準拠の正規表現によって定義されます。minimum/maximum そのタイプのフォームフィールドで許可する必要があるデータの長さ。
- **XML** コンテンツタイプ。XML として認識され、XML 固有のセキュリティチェックの対象となるコンテンツタイプのリスト。これらのコンテンツタイプは、そのコンテンツに割り当てられる正確な MIME タイプを定義する PCRE 準拠の正規表現によって定義されます。
- **JSON** コンテンツタイプ。JSON として認識され、JSON 固有のセキュリティチェックの対象となるコンテンツタイプのリスト。これらのコンテンツタイプは、そのコンテンツに割り当てられる正確な MIME タイプを定義する PCRE 準拠の正規表現によって定義されます。

エンジン設定

October 7, 2021

エンジンの設定は、Citrix Web App Firewall が処理するすべての要求と応答に影響します。設定は次のとおりです。

- クッキー名：Citrix ADC セッション ID を格納するクッキーの名前。
- セッションタイムアウト：許容される最大非アクティブ期間。この時間の間、ユーザー・セッションにアクティビティが表示されない場合、セッションは終了し、ユーザーは指定された開始ページにアクセスしてセッションを再確立する必要があります。
- **cookie** 暗号化後のプレフィクス：暗号化された cookie の暗号化部分の前にあるストリング。
- [**Maximum session lifetime**]：セッションがライブ状態を維持できる最大時間（秒単位）。この期間に達すると、セッションは終了し、ユーザーは指定された開始ページにアクセスしてセッションを再確立する必要があります。この設定は、セッションのタイムアウトより小さくすることはできません。セッションの最大有効期間がないようにこの設定を無効にするには、値をゼロ (0) に設定します。
- ロギングヘッダー名：クライアント IP を保持する HTTP ヘッダーの名前（ロギング用）。
- [**Undefined profile**]：対応するポリシーアクションが undefined として評価されたときに適用されるプロファイル。
- デフォルトプロファイル：ポリシーに一致しない接続に適用されるプロファイル。
- **Import size limit**：シグネチャ、WSDL、スキーマ、HTML、XML エラーページなど、アプライアンスにインポートされたすべてのファイルの最大バイト数。インポート中に、インポートされたオブジェクトのサイズによって、インポートされたすべてのファイルの累積数が構成された制限を超えると、インポート操作は失敗します。アプライアンスは、次のエラーメッセージを表示します。「エラー: インポートに失敗しました-インポ

ートされたオブジェクトに設定された合計サイズ制限を超えています」。

- 学習メッセージレート制限：ラーニングエンジンが処理する 1 秒あたりの要求および応答の最大数。この制限を超える追加の要求または応答は、学習エンジンに送信されません。
- エンティティのデコーディング：Web App Firewall チェックの実行時に HTML エンティティをデコードします。
- 不正な形式の要求のログ：不正な形式の HTTP 要求のロギングを有効にします。
- 設定可能な秘密キーの使用：Web App Firewall 操作には、設定可能な秘密キーを使用します。この秘密鍵は、データの署名と検証に使用されます。「UseConfigurableSecretKey」がオンになっている場合は、「set ns EncryptionParams」パラメータで有効になっているキーを使用する必要があります。
- 学習したデータをリセット：学習したデータをすべて Web App Firewall から削除します。新しいデータを収集して学習プロセスを再開します。

［学習したデータのリセット］と［署名の自動更新］の 2 つの設定は、コマンドインターフェイスと Citrix ADC GUI のどちらを使用して Citrix Web App Firewall を構成するかによって異なります。コマンドインターフェイスを使用する場合は、reset appfw learning data コマンドを使用して、学習済みデータのリセットを設定します。これはパラメータを取らず、他の機能を持ちません。シグニチャの自動更新は、set appfw settings コマンドで設定できます。-SignatureAutoUpdate パラメータは、シグニチャの自動更新を有効または無効にします。-signatureURL は、更新されたシグニチャファイルをホストする URL を設定します。

Citrix ADC GUI を使用する場合は、［セキュリティ］>［**Citrix Web App Firewall**］**>**［エンジン設定］で学習データをリセットします。[学習済みデータをリセット] オプションは、ダイアログボックスの下部にあります。署名の自動更新を設定するには、［セキュリティ］>［**Citrix Web App Firewall**］>［署名］で署名ファイルを選択し、マウスの右ボタンをクリックして［自動更新設定］を選択します。

通常、**Web App Firewall** 設定のデフォルト値は正しいです。ただし、既定の設定によって他のサーバーとの競合が発生したり、ユーザーの接続が早すぎる場合は、変更する必要があります。

Web App Firewall セッション制限は、次のコマンドを使用して設定できます。

```
1 > set appfw settings -sessionLimit 500000
2
3 Done
4
5 Default value:100000   Max value:500000 per PE
6 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用してエンジン設定を構成するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw settings [-sessionCookieName <name>] [-sessionTimeout <positiveInteger> ] [-sessionLifetime <positiveInteger>][-clientIPLoggingHeader`


```
 <headerName> ] [-undefaction <profileName>] [-defaultProfile <profileName
>] [-importSizeLimit <positiveInteger>] [-logMalformedReq ( ON | OFF
)] [-signatureAutoUpdate ( ON | OFF )] [-signatureUrl <expression>]
[-cookiePostEncryptPrefix <string>] [-entityDecoding ( ON | OFF )] [-
useConfigurableSecretKey ( ON | OFF )][-learnRateLimit <positiveInteger
>]
```

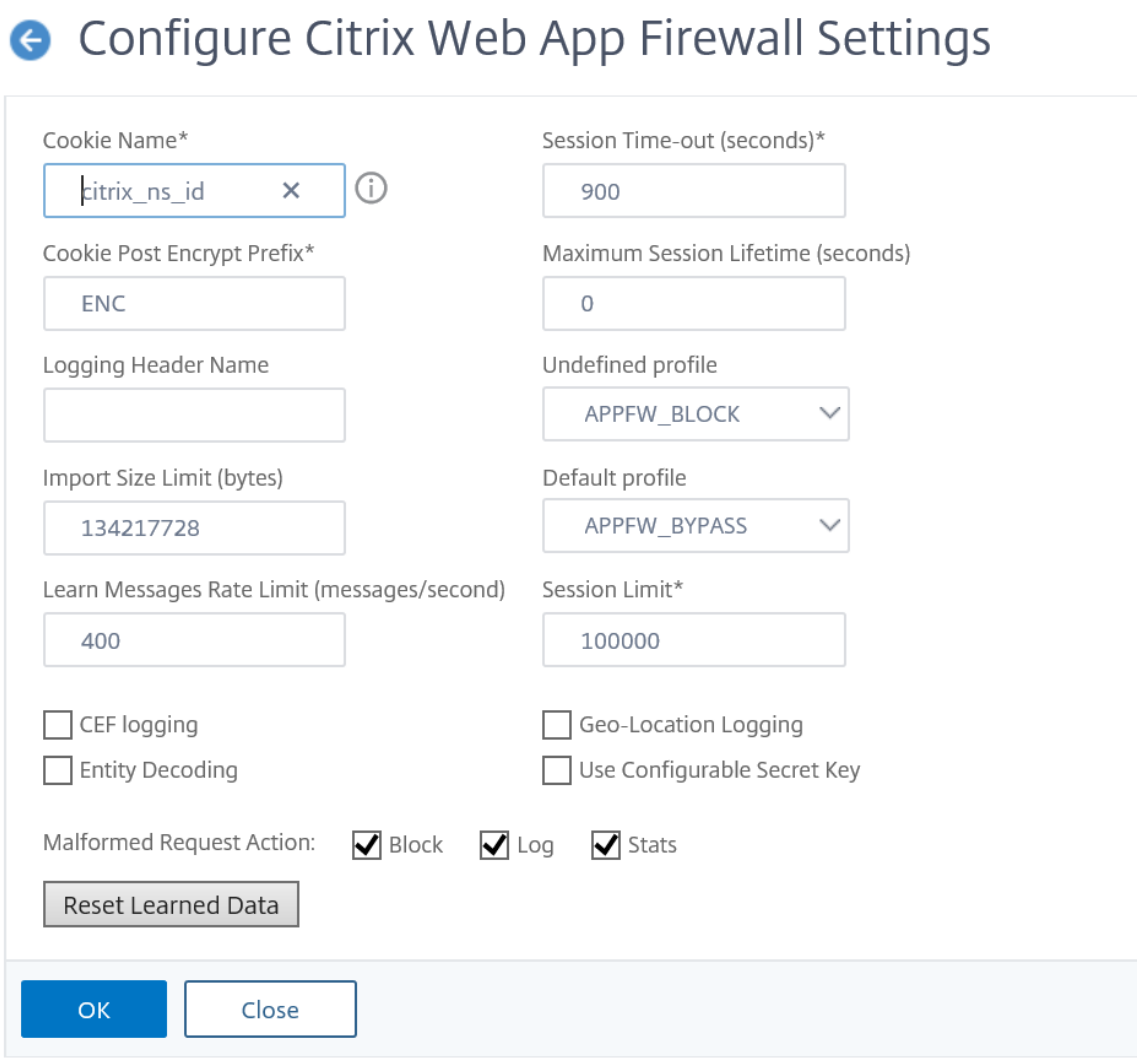
- save ns config

例

```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout
    3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -
    undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096
save ns config
<!--NeedCopy-->
```

Citrix ADC GUI を使用してエンジン設定を構成するには

1. ［セキュリティ］> ［**Citrix Web App Firewall**］に移動します。
2. 詳細ウィンドウで、**[設定]** の **[エンジン設定の変更]** をクリックします。
3. [**Web App Firewall** エンジンの設定] ダイアログボックスで、次のパラメータを設定します。
   - クッキー名
   - セッションのタイムアウト
   - クッキーポスト暗号化プレフィックス
   - 最大セッション寿命
   - ロギングヘッダー名
   - 未定義のプロファイル
   - デフォルトプロファイル
   - 読み込みサイズ制限
   - ラーニングメッセージのレート制限
   - エンティティのデコード
   - 不正な形式の要求をログに記録する
   - シークレットキーを使用
   - ラーニングメッセージレート制限
   - シグニチャの自動更新
4. ［**OK**］をクリックします。



機密フィールド

October 7, 2021

Web フォームフィールドを機密情報として指定して、ユーザーが入力する情報を保護できます。通常、保護された Web サーバーの 1 つ上の Web フォームにユーザーが入力した情報は、Citrix ADC ログに記録されます。ただし、機密として指定された Web フォームのフィールドに入力された情報はログに記録されません。その情報は、Web サイトがそのようなデータを保存するように構成されている場合にのみ、通常は安全なデータベースに保存されます。

機密フィールド指定で保護する必要がある一般的な情報には、次のものがあります。

- パスワード
- クレジットカード番号、検証コード、有効期限
- 社会保障番号

- 納税者番号
- 自宅住所
- 非公開電話番号

e コマースサーバーでの PCI-DSS コンプライアンス、米国内の医療情報を管理するサーバーでの HIPAA コンプライアンス、その他のデータ保護基準への準拠には、適切な機密フィールド指定の使用が必要になる場合があります。

> 重要:
>
> 次の 2 つのケースでは、機密フィールドの指定が期待どおりに機能しません。

- Web フォームに機密フィールドまたはアクション URL が 256 文字を超える場合、Citrix ADC ログでフィールドまたはアクション URL がトランケートされます。
- 特定の SSL トランザクションでは、機密フィールドまたはアクション URL のいずれかが 127 文字を超えると、ログはトランケートされます。

いずれの場合も、Web App Firewall は、通常の 8 文字の文字列ではなく、文字「x」で 15 文字の文字列をマスクします。機密情報を確実に削除するには、最初の 256 文字に一致するフォームフィールド名とアクション URL 式、または (SSL が使用されている場合) 最初の 127 文字を使用する必要があります。

保護された Web サイトの Web フォームフィールドを機密として扱うように WebApp Firewall を設定するには、そのフィールドを [機密フィールド] リストに追加します。フィールド名を文字列として入力することも、1 つ以上のフィールドを指定する PCRE 互換の正規表現を入力することもできます。フィールドを追加するときに機密フィールド指定を有効にすることも、後で指定を変更することもできます。

コマンドラインインターフェイスを使用して機密フィールドを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX )] [-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

例

次の例では、名前が「パスワード」で始まるすべての Web フォームフィールドを機密フィールドの一覧に追加します。

```
1 add appfw confidField Password "https?://www[.]example[.]com/[^<>]\*[^a
    -z]password[0-9a-z._-]\*[.](asp|cgi|htm|html|htp|js|php)" -isRegex
    REGEX -state ENABLED
2 save ns config
3 <!--NeedCopy-->
```


コマンドラインインターフェイスを使用して機密フィールドを変更するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX ) ][-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

例

次の例では、機密フィールドの指定を変更して、コメントを追加します。

```
set appfw confidField Password "https?://www[.]example[.]com/[^<>]\*[^a-z]password[0-9a-z._-]\*[.](asp|cgi|htm|html|htp|js|php)" -comment "Protect password fields." -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して機密フィールドを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `rm appfw confidField <fieldName> <url>`
- `save ns config`

GUI を使用して機密フィールドを構成するには

1. **Security** > **Application Firewall** に移動します。
2. 詳細ウィンドウの [設定] で、[機密フィールドの管理] をクリックします。
3. [機密フィールドの管理] ダイアログボックスで、次のいずれかの操作を行います。
   - 新しいフォームフィールドを一覧に追加するには、[追加] をクリックします。
   - 既存の機密フィールドの指定を変更するには、フィールドを選択し、[編集] をクリックします。
     [**Web App Firewall** 機密フィールド] ダイアログボックスが表示されます。

     > 注:
     > 既存の機密フィールドの指定を選択し、[追加] をクリックすると、[機密フォームフィールドの作成] ダイアログボックスに、その機密フィールドの情報が表示されます。この情報を修正して、新しい機密フィールドを作成できます。
4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。
   - **[有効]** チェックボックス。この機密フィールド指定を有効または無効にする場合は、選択または選択解除します。

- フォームフィールド名は正規表現チェックボックスです。フォームフィールド名で PCRE-format 正規表現を有効にする場合は、選択または選択解除します。
- フィールド名。特定のフィールド名を表すリテラル文字列または PCRE-format 正規表現を入力します。この正規表現は、パターンに続く名前を持つ複数のフィールドと一致します。
- アクションの **URL**。機密フィールドを含む Web フォームが配置されている Web ページの 1 つ以上の URL を定義するリテラル URL または正規表現を入力します。
- コメント。コメントを入力します。オプションです。

5. [作成] または [**OK]** をクリックします。
6. 機密フィールドの一覧から機密フィールドの指定を削除するには、削除する機密フィールドの一覧を選択し、[削除] をクリックして削除し、[**OK**] をクリックして選択を確定します。
7. 機密フィールド指定の追加、変更、削除が完了したら、[閉じる] をクリックします。

例

以下に、便利なフォームフィールド名を定義する正規表現をいくつか示します。

- `^passwd_ (Applies confidential-field status to all field names that begin with the “passwd_” string.)`
- `^((\[0-9a-zA-Z._-]*||\\x\[0-9A-Fa-f][0-9A-Fa-f])+-)?passwd_ (Applies confidential-field status to all field names that begin with the string passwd_, or that contain the string -passwd_ after another string that might contain non-ASCII special characters.)`

有用な特定の URL タイプを定義する正規表現を次に示します。例では、独自のウェブホストとドメインを置き換えます。

- Web フォームが Web ホスト www.example.com の複数の Web ページに表示されているが、それらの Web ページすべてに logon.pl? という名前が付けられている場合は、次の正規表現を使用できます。

```
1  https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)*logon
       [.]pl?
2  <!--NeedCopy-->
```

- Web フォームが Web ホスト www.example-español.com の複数の Web ページに表示され、n-tilde（ñ）特殊文字が含まれている場合は、次の正規表現を使用できます。これは、n-tilde 特殊文字を次のように表します。C3 B1 を含むエンコードされた UTF-8 文字列、UTF-8 文字セットのその文字に割り当てられた 16 進コード：

```
1  https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-
       z_.-]\*/)\* logon[.]pl?
```



```
<!--NeedCopy-->
```

- query.pl を含む Web フォームが example.com ドメイン内の異なるホスト上の複数の Web ページに表示される場合は、次の正規表現を使用できます。

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])\*example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]\*/)*logon[.]pl?
<!--NeedCopy-->
```

- query.pl を含む Web フォームが、異なるドメインの異なるホスト上の複数の Web ページに表示される場合は、次の正規表現を使用できます。

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]\*[.])\*[0-9A-Za-z][0-9A-Za-z_-.]+[.][a-z]{
2,6 }
/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl?
<!--NeedCopy-->
```

- Web フォームが Web ホスト www.example.com の複数の Web ページに表示されているが、それらの Web ページすべてに logon.pl? という名前が付けられている場合は、次の正規表現を使用できます。

```
https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl?
<!--NeedCopy-->
```

フィールドタイプ

October 7, 2021

フィールド型は、Web フォームのフォームフィールドの特定のデータ形式と最小/最大データ長を定義する PCRE 形式の正規表現です。フィールドタイプは、[フィールドフォーマット] チェックで使用されます。

Web App Firewall には、次のようなデフォルトのフィールドタイプがいくつか用意されています。

- 整数. 数字のみで構成される任意の長さの文字列。小数点は除く、オプションの前にマイナス記号 (-) を付けます。
- アルファ。文字のみで構成される任意の長さの文字列。

- 英数字。文字または数字で構成される任意の長さの文字列。
- html にありません。句読点やスペースを含む文字で構成される任意の長さの文字列。HTML 記号やクエリは含まれません。
- 任意。何でもいい

> 重要:
>
> 任意のフィールドタイプをデフォルトのフィールドタイプとして、またはフィールドに割り当てると、アクティブスクリプト、SQL コマンド、およびその他の危険な可能性のあるコンテンツを、保護された Web サイトおよびアプリケーションにそのフォームフィールドで送信できます。使用する場合は、任意のタイプを慎重に使用する必要があります。

[フィールドの種類] リストに独自のフィールドの種類を追加することもできます。たとえば、社会保障番号、郵便番号、または電話番号のフィールドタイプを追加するとします。顧客 ID 番号またはクレジットカード番号のフィールドタイプを追加することもできます。

フィールドタイプを [フィールドタイプ] リストに追加するには、フィールド名をリテラル文字列または PCRE-形式の正規表現として入力します。

コマンドラインインターフェイスを使用してフィールドタイプを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

例

次の例では、米国社会保障番号に一致する SSN という名前のフィールドタイプを [フィールドタイプ] リストに追加し、優先度を 1 に設定します。

```
add appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1
save ns config
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用してフィールドタイプを変更するには

コマンドプロンプトで、次のコマンドを入力します。

- `set appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

例

次の例では、フィールドタイプを変更してコメントを追加します。

```
set appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1 -comment "US Social Security Number"
save ns config
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してフィールドタイプを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `>rm appfw fieldType <name>`
- `save ns config`

GUI を使用してフィールドタイプを構成するには

1. Security > Application Firewall に移動します。
2. 詳細ウィンドウの [設定] で、[フィールドの種類の管理] をクリックします。
3. [フィールド型の管理] ダイアログボックスで、次のいずれかの操作を行います。
   - 新しいフィールドタイプをリストに追加するには、[追加] をクリックします。
   - 既存のフィールドの種類を変更するには、フィールドの種類を選択し、[編集] をクリックします。
     [フィールドの種類の構成] ダイアログボックスが表示されます。

> 注:
>
> 既存のフィールドタイプの指定を選択し、[追加] をクリックすると、そのフィールドタイプの情報がダイアログボックスに表示されます。この情報を修正して、新しいフィールドタイプを作成できます。

4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。
   - Name
   - 正規表現
   - 優先度
   - コメント
5. [作成] または [OK] をクリックします。
6. [フィールドの種類] ボックスの一覧からフィールドの種類を削除するには、削除するフィールドの種類の一覧を選択し、[削除] をクリックして削除し、[**OK**] をクリックして選択を確定します。
7. フィールドタイプの追加、変更、および削除が完了したら、[閉じる] をクリックします。

例

次に示すのは、便利なフィールドタイプの正規表現です。

`^[1-9][0-9]{ 2,2 } -[0-9 } { 2,2 } -[0-9]{ 4,4 } $` 米国社会保障番号

`^\[A-C\]\[0-9\]{ 7,7 } $` カリフォルニア州運転免許証番号

`^+[0-9]{ 1,3 } [0-9()-]{ 1,40 } $`国番号付き国際電話番号

`^[0-9]{ 5,5 } -[0-9]{ 4,4 } $` 郵便番号

`^[0-9A-Za-z][0-9A-Za-z.+_-]{ 0,25 } @([0-9A-Za-z][0-9A-Za-z_-]*[.]){ 1,4 } [A-Za-z]{ 2,6 } $` メールアドレス

XML コンテンツタイプ

October 7, 2021

デフォルトでは、Web App Firewall は特定の命名規則に従うファイルを XML として扱います。Web コンテンツで、これらのファイルが XML ファイルであることを示す追加の文字列またはパターンがないか調べるように Web アプリケーションファイアウォールを設定できます。これにより、特定の XML コンテンツが通常の XML 命名規則に従っていない場合でも、Web App Firewall がサイト上のすべての XML コンテンツを確実に認識し、XML コンテンツが XML セキュリティチェックの対象となるようにすることができます。

XML コンテンツタイプを設定するには、[XML コンテンツタイプ] リストに適切なパターンを追加します。コンテンツタイプを文字列として入力することも、1 つ以上の文字列を指定する PCRE-互換の正規表現を入力することもできます。既存の XML コンテンツタイプのパターンを変更することもできます。

コマンドラインインターフェイスを使用して **XML** コンテンツタイプパターンを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw XMLContentType <XMLContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

例

次の例では、パターンを追加します。*/xml を XML コンテンツタイプリストに追加し、正規表現として指定します。

```
1 add appfw XMLContentType ".*/xml" -isRegex REGEX
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **XML** コンテンツタイプパターンを削除するには

コマンドプロンプトで、次のコマンドを入力します。

- `rm appfw XMLContentType <XMLContenttypevalue>`
- `save ns config`

GUI を使用して **XML** コンテンツタイプリストを構成するには

1. [セキュリティ] > [**Web App Firewall**] に移動します。
2. 詳細ウィンドウで、[設定] の [**XML** コンテンツタイプの管理] をクリックします。
3. [**XML** コンテンツタイプの管理] ダイアログボックスで、次のいずれかの操作を行います。
   - 新しい XML コンテンツタイプを追加するには、[追加] をクリックします。
   - 既存の XML コンテンツタイプを変更するには、そのタイプを選択して [編集] をクリックします。[Web App Firewall の XML コンテンツタイプの設定] ダイアログが表示されます。注：既存の XML コンテンツタイプパターンを選択して [追加] をクリックすると、ダイアログボックスにその XML コンテンツタイプパターンの情報が表示されます。この情報を変更して、新しい XML コンテンツタイプパターンを作成できます。
4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。
   - **IsRegex.** フォームフィールド名で PCRE-format 正規表現を有効にする場合は、選択または選択解除します。
   - [**XML** コンテンツタイプ**]** 追加する XML コンテンツタイプパターンに一致するリテラル文字列または PCRE 形式の正規表現を入力します。
5. [作成] をクリックします。
6. XML コンテンツタイプパターンをリストから削除するには、そのパターンを選択し、[削除] をクリックして削除し、[**OK**] をクリックして選択を確定します。

7. XML コンテンツタイプパターンの追加と削除が完了したら、[閉じる] をクリックします。

JSON コンテンツタイプ

October 7, 2021

デフォルトでは、Web App Firewall はコンテンツタイプ「application/json」のファイルを JSON ファイルとして処理します。デフォルト設定では、Web App ファイアウォールはリクエストと応答の JSON コンテンツを認識し、そのコンテンツを適切に処理できます。

Web コンテンツで、これらのファイルが JSON ファイルであることを示す追加の文字列またはパターンがないか調べるように Web アプリケーションファイアウォールを設定できます。これにより、特定の JSON コンテンツが通常の JSON 命名規則に従っていない場合でも、Web App Firewall がサイト上のすべての JSON コンテンツを認識し、JSON コンテンツが JSON セキュリティチェックの対象となるようにすることができます。

JSON コンテンツタイプを設定するには、[JSON コンテンツタイプ] リストに適切なパターンを追加します。コンテンツタイプを文字列として入力することも、1 つ以上の文字列を指定する PCRE-互換の正規表現を入力することもできます。既存の JSON コンテンツタイプのパターンを変更することもできます。

コマンドラインインターフェイスを使用して JSON コンテンツタイプパターンを追加するには

コマンドプロンプトで、次のコマンドを入力します。

- `add appfw JSONContentType <JSONContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

例

次の例では、パターンを追加します。*/json を JSON コンテンツタイプリストに追加し、正規表現として指定します。

```
1 add appfw JSONContentType ".*/json" -isRegex REGEX
2 <!--NeedCopy-->
```

GUI を使用して JSON コンテンツタイプリストを構成するには

1. **Security** > **Application Firewall** に移動します。
2. 詳細ペインの [設定] で、[**JSON** コンテンツタイプの管理] をクリックします。

3. [JSON コンテンツタイプの管理] ダイアログボックスで、次のいずれかの操作を行います。
   - 新しい JSON コンテンツタイプを追加するには、[追加] をクリックします。
   - 既存の JSON コンテンツタイプを変更するには、そのタイプを選択して [編集] をクリックします。
     [Web App Firewall の JSON コンテンツタイプの設定] ダイアログが表示されます。
     注: 既存の JSON コンテンツタイプパターンを選択して [追加] をクリックすると、ダイアログボックスにその JSON コンテンツタイプパターンの情報が表示されます。この情報を変更して、新しい JSON コンテンツタイプパターンを作成できます。
4. ダイアログボックスで、要素を入力します。これには、次の種類のアカウントがあります。
   - **IsRegex.** フォームフィールド名で PCRE-format 正規表現を有効にする場合は、選択または選択解除します。
   - **JSON** コンテンツタイプ追加する JSON コンテンツタイプパターンと一致するリテラル文字列または PCRE 形式の正規表現を入力します。
5. [作成**]** または [**OK]** をクリックします。
6. JSON コンテンツタイプパターンをリストから削除するには、そのパターンを選択し、[削除] をクリックして削除し、[**OK**] をクリックして選択を確定します。
7. XML コンテンツタイプパターンの追加と削除が完了したら、[閉じる] をクリックします。

統計とレポート

October 7, 2021

ログと統計に保持され、レポートに表示される情報は、Web App Firewall の構成と保守に関する重要なガイダンスを提供します。

Web App Firewall 統計情報

Web App Firewall シグニチャまたはセキュリティチェックの統計アクションを有効にすると、Web App Firewall は、そのシグニチャまたはセキュリティチェックに一致する接続に関する情報を保持します。[グループの選択] リストボックスで次のいずれかのオプションを選択すると、[
監視] タブで累積統計情報を表示できます。

- **Web App Firewall**。Web App Firewall アプライアンスがすべてのプロファイルについて収集したすべての統計情報のサマリー。
- **Web App Firewall (**プロファイルごと**)**。同じ情報ですが、要約ではなくプロファイルごとに表示されます。

この情報を使用して、Web App Firewall の動作を監視し、シグニチャまたはセキュリティチェックに異常なアクティビティまたは異常なヒット数があるかどうかを判断できます。このような異常なアクティビティのパターンが見られる場合は、ログでそのシグニチャまたはセキュリティチェックをチェックして、診断して是正措置を講じることができます。

リラクゼーションヒット統計カウンター

違反したトラフィックに適用された緩和に基づいて、アプライアンス上で違反が発生した回数、違反時に適用された緩和ルールの数、最後に適用されたタイムスタンプなどの統計的な詳細を表示することもできます。これを実行することにより、集中学習エンジンは未使用または冗長な緩和バインディングを自動的に削除できます。詳細については、「WAF Learn Engine 」トピックを参照してください。

リラクゼーションヒット統計カウンターは、以下のセキュリティチェックでのみ使用できます。

- Starturl
- Denyurl
- クロスサイトスクリプティング
- SQL インジェクション

CLI を使用して緩和ルールヒットカウンタの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat appfw profile p1
```

例**:**

```
stat appfw profile p1 -fullvalues
```

スタートルール統計

規則	ヒット	レート	最終ヒット時間
87a4…51177	0	0	木… 1970 年
5b83…dc12a	0	0	木… 1970 年
12345	0	0	木… 1970 年

GUI を使用して緩和ルールヒットカウンタの統計情報を表示するには

緩和ルールヒットカウンタの統計情報を表示するには、次の手順を実行します。

1. ［セキュリティ］ **> Citrix Web App Firewall**］ **>** ［プロファイル］ に移動します。
2. 詳細ウィンドウで、**Web App Firewall** プロファイルを選択し、[統計] をクリックします。
3. **Citrix Web App Firewall** の統計情報ページには、統計の詳細が表示されます。
4. 表形式表示を選択するか、グラフィカルビューに切り替えて、データを表形式またはグラフィカル形式で表示できます。

Web App Firewall レポート

Web App Firewall レポートは、Web App Firewall の設定と、保護された Web サイトのトラフィック処理方法に関する情報を提供します。

PCI DSS レポート

ペイメントカード業界（PCI）データセキュリティスタンダード（DSS）バージョン 1.2 は、12 のセキュリティ基準で構成されており、ほとんどのクレジットカード会社は、クレジットカードおよびデビットカードによるオンライン決済を受け入れる企業が満たすために要求しています。これらの基準は、個人情報の盗難、ハッキング、その他の不正行為を防止するように設計されています。インターネットサービスプロバイダーが PCI DSS 基準を満たしていない場合、その ISP または加盟店が Web サイトを通じたクレジットカード決済の承認を失う可能性があります。

ISP とオンライン加盟店は、PCI DSS 認定セキュリティ評価者 (QSA) 会社による監査を実施することにより、PCI DSS に準拠していることを証明します。PCI DSS レポートは、監査の前と監査中の両方を支援するように設計されています。監査の前に、PCI DSS に関連するウェブアプリケーションファイアウォールの設定、それらの構成方法、および現在の Web App Firewall 構成が標準を満たしているかどうか（最も重要なこと）が示されます。監査中、レポートを使用して、関連する PCI DSS 基準への準拠を示すことができます。

PCI DSS レポートは、Web App Firewall 構成に関連する基準のリストで構成されます。各基準の下には、現在の構成オプション、現在の構成が PCI DSS 基準に準拠しているかどうかを示し、保護された Web サイトが基準に準拠するように Web App Firewall を構成する方法について説明します。

PCI DSS レポートは、[システム**] > [**レポート] の下にあります。レポートを Adobe PDF ファイルとして生成するには、「PCI DSS レポートの生成」をクリックします。ブラウザの設定に応じて、レポートがポップアップウィンドウに表示されるか、ハードディスクに保存するように求められます。

> 注**:**
>
> このレポートやその他のレポートを表示するには、コンピューターに Adobe Reader プログラムをインストールする必要があります。

PCI DSS レポートは、次のセクションで構成されています。

- **[**説明**]**。PCI DSS コンプライアンス概要レポートの説明。
- ファイアウォールのライセンスと機能のステータス。Web App Firewall が Citrix ADC アプライアンスでライセンスされ、有効になっているかどうかを示します。
- エグゼクティブサマリー。PCI DSS 基準をリストし、これらの基準のうちどれが Web App Firewall に関連しているかを示す表。
- **PCI DSS** 基準の詳細情報。Web App Firewall 構成に関連する各 PCI DSS 基準について、PCI DSS レポートには、構成が準拠しているかどうか、および準拠していない場合はそれを準拠させる方法に関する情報を含むセクションが表示されます。

- **Configuration -** 個々のプロファイルのデータ。レポートの上部にある [Web App Firewall 構成] をクリックするか、[レポート] ペインから直接アクセスします。Web App Firewall 構成レポートは PCI DSS レポートと同じで、PCI DSS 固有の概要は省略されています。

Web App Firewall 構成レポート

Web App Firewall 構成レポートは、[システム**] > [**レポート] の下にあります。これを表示するには、[**Web App Firewall** 構成レポートの生成] をクリックします。ブラウザの設定に応じて、レポートがポップアップウィンドウに表示されるか、ハードディスクに保存するように求められます。

Web App Firewall 構成レポートは、次のセクションで構成される [概要] ページから始まります。

- **Web App Firewall** ポリシー。現在の Web App Firewall ポリシーをリストした表。ポリシー名、ポリシーの内容、関連付けられているアクション（またはプロファイル）、およびグローバルバインド情報を示します。
- **Web App Firewall** プロファイル。現在の Web App Firewall プロファイルを一覧表示し、各プロファイルが関連付けられているポリシーを示す表。プロファイルがポリシーに関連付けられていない場合、テーブルには、その場所に INACTIVE と表示されます。

すべてのポリシーのすべてのレポートページをダウンロードするには、[プロファイルの概要] ページの上部にある [すべてのプロファイルのダウンロード] をクリックします。各プロファイルのレポートページを表示するには、画面下部の表でそのプロファイルを選択します。個々のプロファイルの [プロファイル（Profile）] ページには、各チェックに対して各チェックアクションが有効か無効になっているか、およびチェックの他の構成設定が表示されます。

現在のプロファイルの PCI DSS レポートページを含む PDF ファイルをダウンロードするには、ページの上部にある [現在のプロファイルのダウンロード] をクリックします。[プロファイルの概要] ページに戻るには、[**Web App Firewall** プロファイル] をクリックします。メインページに戻るには、[ホーム] をクリックします。PCI DSS レポートは、ブラウザの右上隅にある [**Refresh**] をクリックしていつでも更新できます。

Web App Firewall ログ

June 24, 2022

Web App Firewall は、構成、ポリシー呼び出し、セキュリティチェック違反の詳細を追跡するためのログメッセージを生成します。

セキュリティチェックまたは署名のログアクションを有効にすると、Web App Firewall が Web サイトやアプリケーションを保護する際に監視した要求と応答に関する情報がログメッセージに表示されます。最も重要な情報は、署名またはセキュリティチェック違反が観察されたときに Web App Firewall によって実行されるアクションです。一部のセキュリティチェックでは、ユーザーの場所や違反をトリガーした検出パターンなど、ログメッセージに有用な情報が記録されることがあります。ログ内の違反メッセージの数が過度に増加すると、悪意のある要求が急増している可能性があります。このメッセージは、Web App Firewall 保護によって検出および阻止された特定の脆弱性を悪用するために、アプリケーションが攻撃を受けている可能性があることを警告します。

> 注:
>
> Citrix Web App Firewall ログは、外部の SYSLOG サーバーでのみ使用する必要があります。

Citrix ADC（ネイティブ）形式のログ

Web App Firewall は、デフォルトで Citrix ADC 形式のログ（ネイティブ形式のログとも呼ばれます）を使用します。これらのログは、他の Citrix ADC 機能で生成されるログと同じ形式です。各ログには、次のフィールドが含まれます。

- タイムスタンプ。接続が発生した日時。
- 重大度。ログの重大度レベル。
- モジュール。ログエントリを生成した Citrix ADC モジュール。
- イベントタイプ。署名違反やセキュリティチェック違反などのイベントのタイプ。
- イベント ID。イベントに割り当てられた ID。
- クライアント IP。接続がログに記録されたユーザーの IP アドレス。
- トランザクション ID。ログの原因となったトランザクションに割り当てられた ID。
- セッション ID。ログの原因となったユーザーセッションに割り当てられた ID。
- メッセージ。ログメッセージ。ログエントリをトリガーした署名またはセキュリティチェックを識別する情報が含まれます。

これらのフィールドのいずれか、または異なるフィールドの情報を組み合わせて検索できます。選択できるのは、ログの表示に使用するツールの機能によってのみ制限されます。Citrix ADC syslog ビューアにアクセスして GUI で Web App Firewall ログメッセージを確認したり、Citrix ADC アプライアンスに手動で接続してコマンドラインインターフェイスからログにアクセスしたり、シェルにドロップして/var/log/folderから直接ログを追跡したりできます。

ネイティブ形式のログメッセージの例

```
1 Jun 22 19:14:37 <local0.info> 10.217.31.98 06/22/2015:19:14:37 GMT ns
     0-PPE-1 :
2 default APPFW APPFW_cross-site scripting 60 0 :  10.217.253.62 616-PPE1
      y/3upt2K8ySWWId3Kavbxyni7Rw0000
3 pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
4 12345&drinking_pref=on&text_area=%3Cscript%3E%0D%0A&loginButton=
     ClickToLogin&as_sfid=
5 AAAAAAWEXcNQLlSokNmqaYF6dvfqlChNzSMsdyO9JXOJomm2v
6 BwAMOqZIChv21EcgBc3rexIUcfm0vckKlsgoOeC_BArx1Ic4NLxxkWMtrJe4H7SOfkiv9NL7AG4juPIan

7 %3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c Cross-site script
     check failed for
8 field text_area=”Bad tag: script” <blocked>
9 <!--NeedCopy-->
```


共通イベントフォーマット **(CEF)** ログ

Web App Firewall は CEF ログもサポートします。CEF は、さまざまなセキュリティ、ネットワークデバイス、アプリケーションからのセキュリティ関連情報の相互運用性を向上させるオープンログ管理標準です。CEF を使用すると、お客様は共通のイベントログ形式を使用できるため、エンタープライズ管理システムによる分析のためにデータを簡単に収集および集約できます。ログメッセージはさまざまなフィールドに分割されるため、メッセージを容易に解析し、重要な情報を識別するためのスクリプトを記述できます。

CEF ログメッセージの分析

Web App Firewall CEF ログメッセージには、日付、タイムスタンプ、クライアント IP、ログ形式、アプライアンス、会社、ビルドバージョン、モジュール、セキュリティチェック情報に加えて、次の詳細が含まれます。

- src — 送信元 IP アドレス
- spt — 送信元ポート番号
- リクエスト — リクエスト URL
- act — アクション (ブロックされた、変換されたなど)
- msg — 監視されたセキュリティー検査違反に関するメッセージ
- cn1 — イベント ID
- cn2 — HTTP トランザクション ID
- cs1 — プロファイル名
- cs2 — PPE ID (PPE1 など)
- cs3-セッション ID
- cs4：重大度（情報、アラートなど）
- cs5 — イベント年
- cs6-署名違反カテゴリ
- method — メソッド (GET/POST など)

たとえば、開始 URL 違反がトリガーされたときに生成された次の CEF 形式のログメッセージを考えてみましょう。

```
Jun 12 23:37:17 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0
|APPFW|APPFW_STARTURL|6|src=10.217.253.62 spt=47606 method=GET
request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal URL. cn1=1340
cn2=653 cs1=pr_ffc cs2=PPE1 cs3=EsdGd3VD0OaaURLcZnj05Y6DOmE0002 cs4=ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```


上記のメッセージは、異なるコンポーネントに分解することができます。CEP ログコンポーネントの表を参照してください 。

CEF ログ形式の要求チェック違反の例：要求がブロックされない

```
Jun 13 00:21:28 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_FIELDCONSISTENCY|6|src=10.217.253.62 spt=761 method=GET request=
http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
123456789234&drinking_pref=on&text_area=&loginButton=ClickToLogin&as_sfid
=
AAAAAAWIahZuYoIFbjBhYMP05mJLTwEfIY0a7AKGMg3jIBaKmwtK4t7M7lNxOgj7Gmd3SZc8KUj6CP

7W5kIWDRHN8PtK1Zc-txHkHNx1WknuG9DzTuM7t1THhluevXu9I4kp8%3D&as_fid=feeec8758b4174
0eedeeb6b35b85dfd3d5def30c msg=Field consistency check failed for field passwd cn1=1401
cn2=707 cs1=pr_ffc cs2=PPE1 cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=
not blocked
<!--NeedCopy-->
```

CEF 形式の応答チェック違反の例: 応答が変換される

```
Jun 13 00:25:31 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_SAFECOMMERCE|6|src=10.217.253.62 spt=34041 method=GET request=
http://aaron.stratum8.net/FFC/CreditCardMind.html msg=Maximum number of potential credit
card numbers seen cn1=1470 cn2=708 cs1=pr_ffc cs2=PPE1
cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

CEF 形式のリクエスト側署名違反の例: リクエストはブロックされています

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request=
http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature violation rule ID 807:
```



```
web-cgi /wwwboard/passwd.txt access  cn1=140 cn2=841 cs1=pr_ffc cs2=
    PPE0
cs3=OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
    blocked
<!--NeedCopy-->
```

Web App Firewall 違反メッセージにジオロケーションを記録する

ログの詳細は、リクエストの発信元を特定し、Web App Firewall を最適なセキュリティレベルに設定するのに役立ちます。クライアントの IP アドレスに依存するレート制限などのセキュリティ実装を回避するために、マルウェアまたは不正なコンピュータは、要求の送信元 IP アドレスを変更し続けることができます。リクエストが送信される特定のリージョンを特定すると、リクエストが有効なユーザーからのものか、サイバー攻撃を開始しようとしているデバイスからのものかを判断するのに役立ちます。たとえば、特定のエリアから過度に多数の要求を受信した場合、それらがユーザーによって送信されているのか、不正なマシンによって送信されているのかを簡単に判断できます。受信したトラフィックのジオロケーション分析は、DoS（DoS; サービス拒否）攻撃などの攻撃を回避するのに役立ちます。

Web App Firewall は、組み込みの Citrix ADC データベースを使用して、悪意のある要求の発信元の IP アドレスに対応する場所を特定する利便性を提供します。その後、これらの場所からのリクエストに対して、より高いレベルのセキュリティを適用できます。Citrix のデフォルト構文（PI）式を使用すると、組み込みの場所データベースで使用できる場所ベースのポリシーを柔軟に構成してファイアウォール保護をカスタマイズし、特定の地域の不正なクライアントから攻撃される協調攻撃に対する防御を強化できます。

Citrix ADC 組み込みデータベースを使用することも、他のデータベースを使用することもできます。データベースに特定のクライアント IP アドレスのロケーション情報がない場合、CEF ログにはジオロケーションが不明なジオロケーションとして表示されます。

> 注:
>
> 位置情報ロギングでは、共通イベント形式 (CEF) が使用されます。デフォルトでは、CEF loggingおよびGeoLocationLoggingはオフになっています。両方のパラメータを明示的に有効にする必要があります。

位置情報を示す CEF ログメッセージの例

```
June 8 00:21:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_STARTURL|6|src=10.217.253.62 geolocation=NorthAmerica.US.Arizona.
    Tucson.\*.\*
spt=18655 method=GET request=http://aaron.stratum8.net/FFC/login.html
msg=Disallow Illegal URL. cn1=77 cn2=1547 cs1=test_pr_adv cs2=PPE1
cs3=KDynjg1pbFtfhC/nt0rBU1o/Tyg0001 cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

ジオロケーション = 不明を示すログメッセージの例


```
June 9 23:50:53 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|
APPFW|APPFW_STARTURL|6|src=10.217.30.251 geolocation=Unknown spt=5086
method=GET request=http://aaron.stratum8.net/FFC/login.html msg=
    Disallow Illegal URL.
cn1=74 cn2=1576 cs1=test_pr_adv cs2=PPE2 cs3=
    PyR0eOEM4gf6GJiTyauiHByL88E0002
cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

コマンドインターフェイスを使用してログアクションとその他のログパラメータを設定する

コマンドラインを使用してプロファイルのセキュリティー検査のログアクションを構成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `set appfw profile <name> SecurityCheckAction ([log] | [none])`
- `unset appfw profile <name> SecurityCheckAction`

例

```
set appfw profile pr_ffc StartURLAction log
```

```
unset appfw profile pr_ffc StartURLAction
```

コマンドラインを使用して CEF ロギングを設定するには

CEF ロギングは、デフォルトでは無効になっています。コマンドプロンプトで、次のいずれかのコマンドを入力して、現在の設定を変更または表示します。

- `set appfw settings CEFLogging on`
- `unset appfw settings CEFLogging`
- `sh appfw settings | grep CEFLogging`

コマンドラインを使用してクレジットカード番号のロギングを構成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `set appfw profile <name> -doSecureCreditCardLogging ([ON] | [OFF])`
- `unset appfw profile <name> -doSecureCreditCardLogging`

コマンドラインを使用して位置情報ログを構成するには

1. set コマンドを使用して GeoLocationLogging を有効にします。CEF ロギングは同時に有効にできます。位置情報ロギングを無効にするには、unset コマンドを使用します。show コマンドは、すべての Web App Firewall パラメータの現在の設定を表示します。ただし、grep コマンドを指定して特定のパラメータの設定を表示する場合を除きます。

- set appfw settings GeoLocationLogging ON [CEFLogging ON]
- unset appfw settings GeoLocationLogging
- sh appfw settings | grep GeoLocationLogging

2. データベースを指定する

```
add locationfile /var/netscaler/inbuilt_db/Citrix_netscaler_InBuilt_GeoIP_DB
.csv
```

または

```
add locationfile <path to database file>
```

Web App Firewall ログをカスタマイズする

デフォルトフォーマット (PI) 式を使用すると、ログに含まれる情報を柔軟にカスタマイズできます。Web App Firewall で生成されたログメッセージに、キャプチャする特定のデータを含めるオプションがあります。たとえば、Web App Firewall セキュリティチェックとともに AAA-TM 認証を使用していて、セキュリティチェック違反をトリガーしたアクセスされた URL、URL を要求したユーザーの名前、送信元 IP アドレス、およびユーザーが要求を送信した送信元ポートを知りたい場合、次のようになります。では、次のコマンドを使用して、すべてのデータを含むカスタマイズされたログメッセージを指定できます。

```
1 > sh version
2 NetScaler NS12.1: Build 50.0013.nc, Date: Aug 28 2018, 10:51:08   (64-
     bit)
3  Done
4 <!--NeedCopy-->
```

```
1 > add audit messageaction custom1 ALERT 'HTTP.REQ.URL + " " + HTTP.REQ.
     USER.NAME + " " + CLIENT.IP.SRC + ":" + CLIENT.TCP.SRCPORT'
2 Warning: HTTP.REQ.USER has been deprecated. Use AAA.USER instead.
3  Done
4 <!--NeedCopy-->
```

```
1 > add appfw profile test_profile
2  Done
3 <!--NeedCopy-->
```



```
> add appfw policy appfw_pol true test_profile -logAction custom1
 Done
<!--NeedCopy-->
```

Web App Firewall ログを分離するように Syslog ポリシーを構成する

Web App Firewall には、Web App Firewall セキュリティログメッセージを分離して別のログファイルにリダイレクトするオプションがあります。これは、Web App Firewall が多数のログを生成し、他の Citrix ADC ログメッセージを表示しにくい場合に適しています。このオプションは、Web App Firewall ログメッセージのみを表示し、他のログメッセージを表示したくない場合にも使用できます。

Web App Firewall ログを別のログファイルにリダイレクトするには、Web App Firewall ログを別のログファシリティに送信するように syslog アクションを構成します。このアクションは、syslog ポリシーを構成するときに使用し、Web App Firewall で使用するためにグローバルにバインドできます。

例**:**

1. シェルに切り替え、vi などのエディタを使用して /etc/syslog.conf ファイルを編集します。次の例に示すように、local2.* を使用してログを別のファイルに送信する新しいエントリを追加します。

   ```
   local2.\* /var/log/ns.log.appfw
   ```

2. syslog プロセスを再起動します。次の例に示すように、grep コマンドを使用して syslog プロセス ID（PID）を識別できます。

   ```
   root@ns\## **ps -A | grep syslog**

   1063 ?? Ss 0:03.00 /usr/sbin/syslogd -b 127.0.0.1 -n -v -v -8 -C

   root@ns## **kill -HUP** 1063
   ```

3. コマンドラインインターフェイスから、アクションを含む詳細またはクラシック SYSLOG ポリシーを構成し、グローバル Web App Firewall ポリシーとしてバインドします。高度な SYSLOG ポリシーを構成 Citrix。

   SYSLOG ポリシーの詳細設定

   ```
   add audit syslogAction sysact1 1.1.1.1 -logLevel ALL -logFacility
   LOCAL2

   add audit syslogPolicy syspol1 true sysact1

   bind audit syslogGlobal -policyName syspol1 -priority 100 -globalBindType
    APPFW_GLOBAL
   ```

   クラシック SYSLOG ポリシー設定

   ```
   add audit syslogAction sysact1 1.1.1.1 -logLevel ALL -logFacility
   LOCAL2
   ```



```
add audit syslogPolicy syspol1 true sysact1

bind audit syslogGlobal -policyName syspol1 -priority 100 -globalBindType
 APPFW_GLOBAL
```

4. Web App Firewall のセキュリティチェック違反はすべて、/var/log/ns.log.appfwファイルにリダイレクトされます。このファイルを末尾に付けて、進行中のトラフィックの処理中にトリガーされる Web App Firewall 違反を表示できます。

   ```
   root@ns## tail -f ns.log.appfw
   ```

警告: ログを別のログファシリティにリダイレクトするように syslog ポリシーを構成した場合、Web App Firewall のログメッセージは/var/log/ns.logファイルに表示されなくなります。

> 注:
>
> ローカルの Citrix ADC アプライアンス上の別のログファイルにログを送信する場合は、そのローカル Citrix ADC アプライアンスに syslog サーバーを作成できます。独自の IP にsyslogactionを追加し、外部サーバーの設定と同様に ADC を構成します。ADC はログを保存するサーバーとして機能します。同じ IP とポートで 2 つのアクションを追加することはできません。syslogactionでは、デフォルトでは、IP の値は127.0.0.1に設定され、port の値は514に設定されます。

Web App Firewall ログを表示する

syslog ビューアを使用するか、Citrix ADC アプライアンスにログオンして UNIX シェルを開き、任意の UNIX テキストエディタを使用してログを表示できます。

コマンドラインを使用してログメッセージにアクセスするには

シェルに切り替えて、**/var/log/** フォルダーの ns.logs を末尾に付けて、**Web App Firewall** のセキュリティチェック違反に関するログメッセージにアクセスします。

- Shell
- tail -f /var/log/ns.log

vi エディタ、または任意の Unix テキストエディタまたはテキスト検索ツールを使用して、特定のエントリのログを表示およびフィルタリングできます。たとえば、grep コマンドを使用して、クレジットカード違反に関するログメッセージにアクセスできます。

- tail -f /var/log/ns.log | grep SAFECOMMERCE

GUI を使用してログメッセージにアクセスするには

Citrix GUI には、ログメッセージを分析するための便利なツール（Syslog Viewer）が含まれています。Syslog ビューアにアクセスするには、複数のオプションがあります。

- プロファイルの特定のセキュリティチェックのログメッセージを表示するには、[**Web App Firewall**] > [プロファイル] に移動し、ターゲットプロファイルを選択し、[セキュリティチェック] をクリックします。ターゲットセキュリティチェックの行を強調表示し、[ログ] をクリックします。プロファイルの選択したセキュ

リティチェックから直接ログにアクセスすると、ログメッセージが除外され、選択したセキュリティチェックの違反に関連するログのみが表示されます。Syslog Viewer では、Web App Firewall ログをネイティブ形式および CEF 形式で表示できます。ただし、Syslog Viewer がターゲットプロファイル固有のログメッセージを除外するには、プロファイルからアクセスしたときに、ログが CEF ログ形式である必要があります。

- 「**Citrix ADC** 」>「システム」>「監査」の順に選択して、Syslog ビューアにアクセスすることもできます。[監査メッセージ] セクションで、[Syslog メッセージ] リンクをクリックして Syslog Viewer を表示します。Syslog Viewer には、すべてのプロファイルのすべての Web App Firewall セキュリティチェック違反ログを含む、すべてのログメッセージが表示されます。このログメッセージは、要求処理中に複数のセキュリティー検査違反がトリガされる可能性がある場合のデバッグに役立ちます。
- [**Web App Firewall**] > [ポリシー] > [監査] に移動します。[Audit Messages] セクションで、[Syslog messages] リンクをクリックして Syslog Viewer を表示します。このビューアには、すべてのプロファイルのすべてのセキュリティチェック違反ログを含む、すべてのログメッセージが表示されます。

HTML ベースの Syslog Viewer には、関心のあるログメッセージのみを選択するための次のフィルタオプションが用意されています。

- ファイル：現在の/var/log/ns.logファイルがデフォルトで選択され、対応するメッセージが Syslog Viewer に表示されます。/var/log ディレクトリにある他のログファイルの一覧が、圧縮された.gz 形式で表示されます。アーカイブされたログファイルをダウンロードして解凍するには、ドロップダウンリストのオプションからログファイルを選択します。選択したファイルに関連するログメッセージが syslog ビューアに表示されます。表示を更新するには、[Refresh] アイコン (2 つの矢印の円) をクリックします。
- [モジュール（**Module**）**]** リストボックス：ログを表示する Citrix ADC モジュールを選択できます。Web App Firewall ログの APPFW に設定できます。
- [**Event Type]** リストボックス：このボックスには、関心のあるイベントのタイプを選択するための一連のチェックボックスがあります。たとえば、署名違反に関するログメッセージを表示するには、[**APPFW_SIGNATURE_MATCH**] チェックボックスをオンにします。同様に、チェックボックスをオンにして、関心のある特定のセキュリティチェックを有効にすることができます。複数のオプションを選択できます。
- [**Severity**]：特定の重大度レベルを選択して、その重大度のログだけを表示できます。すべてのログを表示する場合は、チェックボックスをすべて空白のままにします。

  特定のセキュリティ検査の Web App Firewall セキュリティ検査違反ログメッセージにアクセスするには、[モジュール] のドロップダウンリストオプションで [**APPFW**] を選択してフィルタリングします。[イベントタイプ] には、選択をさらに絞り込むための豊富なオプションセットが表示されます。たとえば、[**APPFW_FIELDFORMAT]** チェックボックスをオンにして **[**適用**]** ボタンをクリックすると、フィールド形式のセキュリティチェック違反に関するログメッセージのみが Syslog Viewer に表示されます。同様に、[**APPFW_SQL**] と [**APPFW_STARTURL**] チェックボックスをオンにして [適用] ボタンをクリックすると、これら 2 つのセキュリティー検査違反に関するログメッセージのみが syslog ビューアに表示されます。

特定のログメッセージの行にカーソルを置くと、[**Module**]、[**EventType]**、**[**EventID]**、**[**** メッセージ] などの複数のオプションがログメッセージの下に表示されます。これらのオプションのいずれかを選択して、ログ内の対応する情報を強調表示できます。

ハイライト

- **CEF** ログ形式のサポート：CEF ログ形式オプションは、Web App Firewall ログメッセージを監視、解析、分析して攻撃を特定し、構成された設定を微調整して誤検出を減らし、統計情報を収集するための便利なオプションを提供します。
- ログメッセージをカスタマイズするオプション-高度な PI 式を使用してログメッセージをカスタマイズし、ログに表示するデータを含めることができます。
- **Web App Firewall** 固有のログを分離する-アプリケーションファイアウォール固有のログをフィルタリングし、別のログファイルにリダイレクトするオプションがあります。
- リモートロギング：ログメッセージをリモート syslog サーバにリダイレクトできます。
- 位置情報ロギング：Web App Firewall を構成して、要求を受信したエリアの地理的位置情報を含めることができます。組み込みの地理位置情報データベースを使用できますが、外部位置情報データベースを使用するオプションもあります。Citrix ADC アプライアンスは、IPv4 と IPv6 の両方の静的地理位置情報データベースをサポートしています。
- 情報リッチログメッセージ：設定に応じて、ログに含めることができる情報のタイプの例をいくつか示します。
  - Web App Firewall ポリシーがトリガーされました。
  - セキュリティチェック違反がトリガーされました。
  - リクエストは形式に誤りがあると考えられました。
  - リクエストまたはレスポンスがブロックされたか、ブロックされませんでした。
  - リクエストデータ（SQL またはクロスサイトスクリプティングの特殊文字など）またはレスポンスデータ（クレジットカード番号やセーフオブジェクト文字列など）が変換されました。
  - 応答のクレジットカードの数が、設定された制限を超えました。
  - クレジットカード番号とタイプ。
  - 署名ルールで設定されたログ文字列、および署名 ID。
  - リクエストのソースに関する地理位置情報。
  - 保護された機密フィールドに対するユーザ入力のマスク (X アウト)。

正規表現パターンを使用して機密データをマスクする

Web アプリケーションファイアウォール (WAF) プロファイルにバインドされたログ式のREGEX_REPLACE詳細ポリシー (PI) 関数を使用すると、WAF ログ内の機密データをマスクできます。オプションを使用して、正規表現パターンを使用してデータをマスクし、データをマスクする文字または文字列パターンを指定できます。また、PI 関数を設定して、正規表現パターンの最初のオカレンスまたはすべてのオカレンスを置き換えることができます。

デフォルトでは、Citrix GUI インターフェイスは次のマスクを提供します。

- SSN
- クレジットカード
- パスワード
- ユーザー名

Web アプリケーションファイアウォールのログで機密データをマスクする

WAF プロファイルにバインドされたログ式でREGEX_REPLACE詳細ポリシー式を設定することで、WAF ログ内の機密データをマスクできます。
機密データをマスクするには、次の手順を完了する必要があります。

1. Web アプリケーションファイアウォールプロファイルを追加する
2. ログ式を WAF プロファイルにバインドする

Web アプリケーションファイアウォールプロファイルを追加する

コマンドプロンプトで入力します。

```
add appfw profile <name>
```

例**:**

```
Add appfw profile testprofile1
```

Web アプリケーションファイアウォールプロファイルでログ式をバインドする

コマンドプロンプトで入力します。

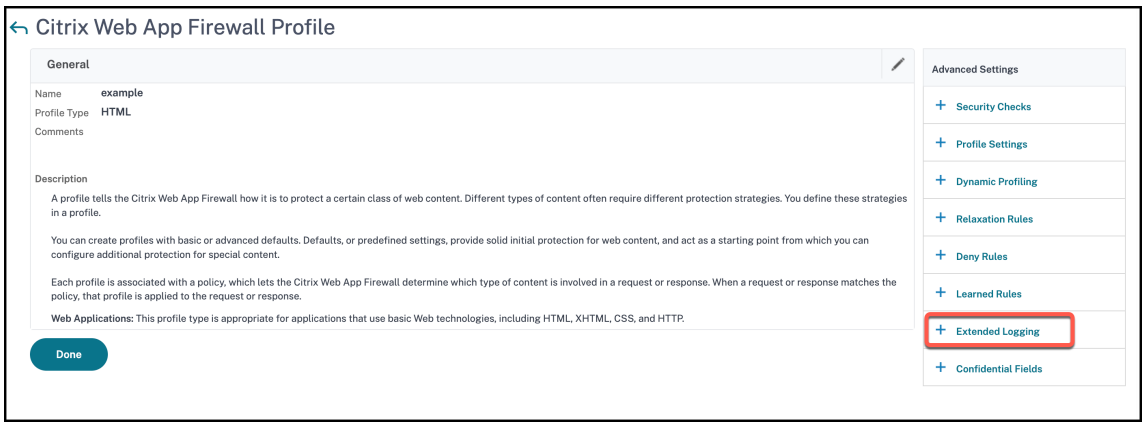
```
bind appfw profile <name> -logExpression <string> <expression> -comment <
string>
```

例**:**

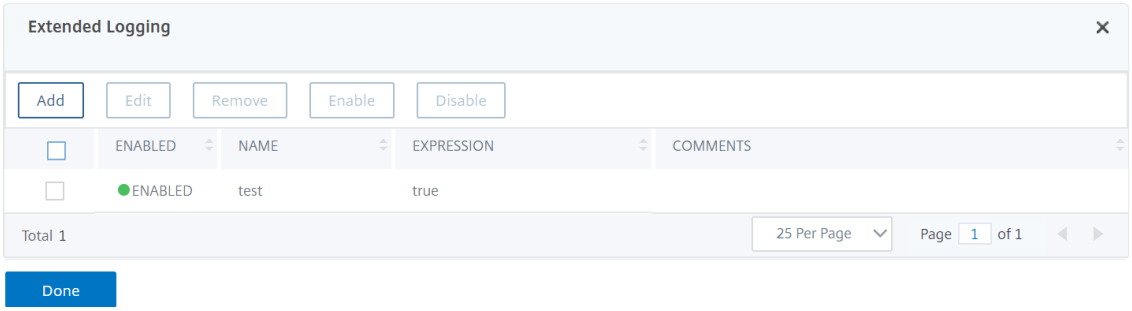
```
bind appfw profile testProfile -logExpression ”MaskSSN””HTTP.REQ.BODY
(10000).REGEX_REPLACE(re!\b\d{ 3 } -\d{ 2 } -\d{ 4 } \b!, “xxx” , ALL)”-
comment ”SSN Masked”
```

Citrix ADC GUI を使用して **Web** アプリケーションファイアウォールログの機密データをマスクする

1. ナビゲーションペインで、[セキュリティ] > [**Citrix Web App Firewall**] > [プロファイル] の順に展開します。
2. [プロファイル] ページで、[編集] をクリックします。
3. [**Citrix Web App Firewall** プロファイル] ページで、[詳細設定] セクションに移動し、[拡張ログ] をクリックします。



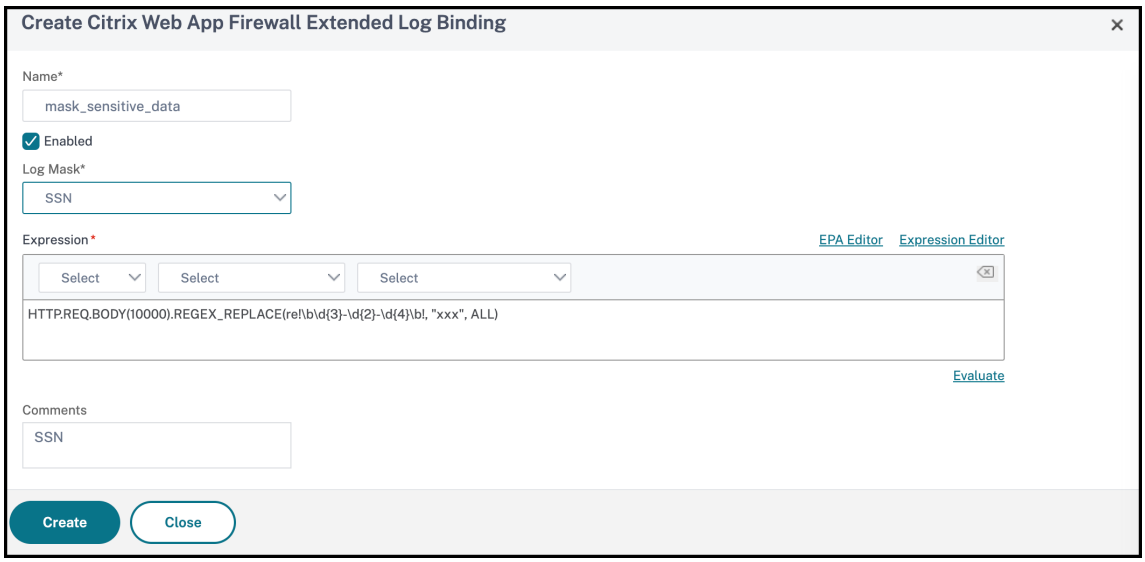
4. [拡張ログ] セクションで、[追加] をクリックします。



5. [**Citrix Web App Firewall** 拡張ログバインドの作成] ページで、次のパラメーターを設定します。

   a) Name：ログ式の名前。
   b) 有効。機密データをマスクするには、このオプションを選択します。
   c) ログマスク。マスクするデータを選択します。
   d) 式。WAF ログの機密データをマスクできるようにする高度なポリシー式を入力します。
   e) コメント。機密データのマスキングに関する簡単な説明。

6. **[作成]** して **[閉じる]** をクリックします。



付録

October 7, 2021

次の補足資料は、複雑または周辺の Web App Firewall タスクに関する追加情報を提供します。

PCRE 文字エンコーディング形式

October 7, 2021

Citrix ADC オペレーティングシステムでは、印刷可能な **ASCII** 文字セット内の文字の直接入力のみをサポートしています。16 進コードを持つ文字は、16 進数コード（ASCII 32）と 16 進数 7E（ASCII 127）です。この範囲外のコードを含む文字を Web App Firewall 設定に含めるには、その UTF-8 の 16 進コードを PCRE 正規表現として入力する必要があります。

Web App Firewall 設定に URL、フォームフィールド名、またはセーフオブジェクト式として含める場合、多くの文字タイプでは PCRE 正規表現を使用してエンコードする必要があります。これには、次のものがあります。

- 上限 **ASCII** 文字。16 進数の 7 F (ASCII 128) から 16 進数の FF (ASCII 255) までのエンコーディングを持つ文字。使用される文字マップに応じて、これらのエンコーディングは、制御コード、アクセントやその他の修正を加えた ASCII 文字、非ラテンアルファベット、および基本的な ASCII セットに含まれない記号を参照できます。これらの文字は、URL、フォームフィールド名、および安全なオブジェクト式で使用できます。
- **2** バイト文字。2 つの 8 バイトワードを使用するエンコード文字です。2 バイト文字は、主に中国語、日本語、韓国語のテキストを電子形式で表現するために使用されます。これらの文字は、URL、フォームフィールド名、および安全なオブジェクト式で使用できます。

ASCII 制御文字。プリンタにコマンドを送信するために使用される、印刷できない文字。16 進コードが 16 進数 20 (ASCII 32) 未満の ASCII 文字はすべて、このカテゴリに分類されます。ただし、これらの文字は URL またはフォームフィールド名に表示されてはならず、安全なオブジェクト式に表示されることはめったにありません。

Citrix ADC アプライアンスは、UTF-8 文字セット全体をサポートするのではなく、次の 8 文字セットに含まれる文字のみをサポートします。

- 英語 **(**米国**) (ISO-8859-1)**。ラベルは「英語 US」と表示されますが、Web App Firewall は ISO-8859-1 文字セット (Latin-1 文字セットとも呼ばれます) のすべての文字をサポートします。この文字セットは、ほとんどの現代の西ヨーロッパ言語を完全に表し、残りの部分ではいくつかの珍しい文字を除くすべてを表します。
- 繁体字中国語 **(Big5)**。Web App Firewall は、BIG5 文字セットのすべての文字をサポートします。この文字セットには、香港、マカオ、台湾で話され書かれた現代中国語で一般的に使用される繁体字中国語 (表意文字)、および中国本土以外に住む多くの中国民族遺産の人々によって含まれています。
- 簡体字中国語 **(GB2312)**。Web App Firewall では、GB2312 文字セットのすべての文字がサポートされています。この文字セットには、中国本土で話され、書かれている現代中国語で一般的に使用される簡体字中国語 (表記) がすべて含まれています。
- 日本語 **(SJIS)**。Web App Firewall では、Shift-JIS (SJIS) 文字セットのすべての文字がサポートされています。この文字セットには、現代日本語でよく使用されるほとんどの文字 (表意文字) が含まれます。
- 日本語 **(EUC-JP)**。Web App Firewall は、EUC-JP 文字セット内のすべての文字をサポートします。この文字セットには、現代日本語でよく使用されるすべての文字 (表意文字) が含まれます。
- 韓国語 **(EUC-KR)**。Web App Firewall は、EUC-KR 文字セットのすべての文字をサポートします。この文字セットには、現代韓国語で一般的に使用されるすべての文字 (表意文字) が含まれます。
- トルコ語 **(ISO-8859-9)**。Web App Firewall は、現代トルコ語で使用されるすべての文字を含む ISO-8859-9 文字セットのすべての文字をサポートしています。
- ユニコード **(UTF-8)**。Web App Firewall では、現代ロシア語で使用されている文字を含め、UTF-8 文字セットの特定の文字がサポートされています。

Web App Firewall を設定する場合、UTF-8 仕様でその文字に割り当てられた 16 進コードを使用して、すべての ASCII 文字を PCRE 形式の正規表現として入力します。通常の ASCII 文字セット内の記号と文字は、その文字セットで 1 桁の 2 桁のコードが割り当てられ、UTF-8 文字セットで同じコードが割り当てられます。たとえば、感嘆符 (!) は ASCII 文字セットで 16 進コード 21 が割り当てられ、UTF-8 文字セットでも 16 進 21 になります。サポートされている別の文字セットのシンボルと文字には、UTF-8 文字セットでそれらに割り当てられた 16 進コードのペアセットがあります。たとえば、急性アクセント（á）の文字 a には、UTF-8 コード C3 A1 が割り当てられます。

Web App Firewall 構成でこれらの UTF-8 コードを表す構文は、ASCII 文字の場合は「\ xNN」、英語、ロシア語、トルコ語で使用される非 ASCII 文字の場合は「\ xNN\ xNN」、中国語、日本語、韓国語で使用される文字の場合は「\ xNN\ xNN」です。たとえば、！を表す場合は、！を UTF-8 文字として Web App Firewall 正規表現に入力する場合は、「\ x21」と入力します。á を含める場合は、xC3xA1 と入力します。

> 注:
>
> 通常、ASCII 文字を UTF-8 形式で表現する必要はありませんが、これらの文字が Web ブラウザや基礎となるオペレーティングシステムを混乱させる可能性がある場合には、文字の UTF-8 表現を使用してこの混乱を避けることができます。たとえば、URL にスペースが含まれている場合、特定のブラウザや Web サーバーソフトウェアを混乱させないように、スペースを x20 としてエンコードできます。

次に、URL、フォームフィールド名、および ASCII 以外の文字を含むセーフオブジェクト式の例を示します。これらの文字は、Web App Firewall 設定に含めるには PCRE 形式の正規表現として入力する必要があります。各例は、実際の URL、フィールド名、または式文字列を最初に示し、その後に PCRE-形式の正規表現を示します。

- 拡張 ASCII 文字を含む URL。

  実際の URL:`http://www.josénuñez.com`
  エンコードされた URL:`^http://www\[.\]jos\\xC3\\xA9nu\\xC3\\xB1ez\[.\]com$`

- 拡張 ASCII 文字を含む別の URL。

  実際の URL:`http://www.example.de/trömso.html`
  エンコードされた URL:`^http://www[.]example[.\]de/tr\xC3\xB6mso[.]html$`

  拡張 ASCII 文字を含むフォームフィールド名。

  実際の名前:nome_do_usuário
  エンコードされた名前:^nome_do_usu\ xC3\ xA1rio$

- 拡張 ASCII 文字を含むセーフオブジェクト式。

  エンコードされていない式 [A-Z]{3,6} ¥[1-9\][0-9]{6,6}
  エンコードされた式: [A-Z]{3,6}\ xC2\ xA5 [1-9] [0-9] {6,6}

Unicode 文字セット全体と一致する UTF-8 エンコーディングを含むいくつかのテーブルがあります。次の表に、この情報を含む便利な Web サイトを示します。

`http://www.utf8-chartable.de/unicode-utf8-table.pl`

この Web サイトの表の文字が正しく表示されるようにするには、コンピュータに適切な Unicode フォントがインストールされている必要があります。そうしないと、文字の視覚的な表示に誤りがある可能性があります。ただし、文字を表示するための適切なフォントがインストールされていない場合でも、この Web ページの説明と UTF-8 および UTF-16 コードは正しいです。

WAF 使用のためのホワイトハット WASC シグニチャタイプ

October 7, 2021

Citrix Web App Firewall は、Whitehat スキャナーが生成するすべての脆弱性タイプのブロックルールを受け入れて生成します。ただし、特定の脆弱性は Web AppFirewall に最も当てはまります。これらの脆弱性のリストは、

WASC 1.0、WASC 2.0、またはベストプラクティスのシグネチャの種類によって分類されます。

WASC 1.0 シグニチャタイプ

- HTTP リクエストの密輸中
- HTTP 応答の分割
- HTTP 応答の密輸
- ヌル・バイト・インジェクション
- リモートファイルの包含
- URL リダイレクタの不正使用

WASC 2.0 シグニチャタイプ

- 機能的濫用
- ブルートフォース
- コンテンツのなりすまし
- サービス拒否
- ディレクトリのインデックス作成
- 情報漏えい
- アンチオートメーションが不十分
- 認証が不十分
- 承認が不十分
- セッションの有効期限が不十分
- LDAP インジェクション
- セッション固定

ベストプラクティス

- オートコンプリート属性
- クッキーのアクセス制御が不十分
- パスワード強度が不十分
- 無効な HTTP メソッドの使用法
- 非 HTTP のみのセッションクッキー
- パーシステント・セッション・クッキー
- 個人を特定できる情報
- セキュアなキャッシュ可能な HTTP メッセージ
- セキュリティで保護されていないセッションクッキー

リクエスト処理のストリーミングサポート

October 7, 2021

Citrix Web App Firewall は、要求側のストリーミングをサポートし、パフォーマンスを大幅に向上させます。アプライアンスは、要求をバッファリングする代わりに、SQL、クロスサイトスクリプティング、フィールドの一貫性、フィールド形式などのセキュリティ違反について着信トラフィックを調べます。アプライアンスがフィールドのデータの処理を完了すると、要求はバックエンドサーバーに転送され、アプライアンスは他のフィールドの評価を続行します。このデータ処理により、多くのフィールドを持つフォームを処理する処理時間が大幅に短縮されます。

> 注:
>
> Citrix Web App Firewall は、ストリーミングなしで最大ポストサイズ 20MB をサポートします。リソースの使用率を高めるには、20 MB を超えるペイロードに対してのみストリーミングを有効にすることをお勧めします。また、ストリーミングが有効になっている場合、バックエンドサーバーはチャンクされた要求を受け入れる必要があります。

ストリーミングプロセスはユーザーに対して透過的ですが、以下の変更により、若干の構成調整が必要になります。

正規表現パターンマッチ**:** 正規表現パターンマッチは、連続する文字列マッチで 4K に制限されるようになりました。

フィールド名の一致： Web App Firewall 学習エンジンは、名前の最初の 128 バイトしか区別できません。フォームに、最初の 128 バイトと同じ文字列が一致する名前のフィールドが複数ある場合、学習エンジンはそれらを区別しません。同様に、展開された緩和ルールは、そのようなフィールドをすべて誤って緩和する可能性があります。

空白の削除、パーセントデコード、Unicode デコード、および文字セット変換は、セキュリティチェック検査を提供するために正規化中に実行されます。128 バイトの制限は、UTF-8 文字形式でのフィールド名の正規化された表現に適用されます。ASCII 文字の長さは 1 バイトですが、一部の国際言語の文字の UTF-8 表現は 1 バイトから 4 バイトの範囲です。名前内の各文字が UTF-8 形式に変換するのに 4 バイトかかる場合、学習したルールによって名前の最初の 32 文字だけが区別されます。

フィールド整合性チェック：フィールド整合性を有効にすると、セッション内のすべてのフォームは、「action_url」を考慮せずに、Web App Firewall によって挿入された「as_fid」タグに基づいて保存されます。

- フォームフィールドの一貫性のための必須フォームタグ付け：フィールドの一貫性チェックが有効になっている場合、フォームタグも有効にする必要があります。フォームのタグ付けがオフになっていると、フィールドの一貫性保護が機能しないことがあります。
- セッションレスフォームフィールドの整合性：セッションレスフィールドの整合性パラメーターが有効になっている場合、Web AppFirewall はフォームの「GET」から「POST」への変換を実行しなくなりました。フォームタグは、セッションレスフィールドの一貫性にも必要です。
- **as_fid** の改ざん： as_fid を改ざんした後にフォームが送信されると、フィールドが改ざんされていなくてもフィールドの一貫性違反が発生します。非ストリーミング要求では、セッションに格納された「action_url」を使用してフォームを検証できるため、これは許可されました。

署名**:** 署名の仕様は次のとおりです。

- 場所**:** 各パターンに場所を指定する必要があることは必須要件になりました。ルール内のすべてのパターンには**<Location>** タグが必要です。
- 高速マッチ：すべてのシグニチャルールに高速マッチパターンが必要です。高速一致パターンがない場合は、可能であれば選択が試みられます。高速一致はリテラル文字列ですが、使用可能な文字列が含まれている場合は、PCRE を使用して高速一致を行うことができます。
- 非推奨のロケーション**:** 署名ルールで次のロケーションがサポートされなくなりました。
  - HTTP_ANY
  - HTTP_RAW_COOKIE
  - HTTP_RAW_HEADER
  - HTTP_RAW_RESP_HEADER
  - HTTP_RAW_SET_COOKIE

クロスサイト scripting/SQL 変換：一重引用符（'）、円記号（\）、セミコロンなどの SQL 特殊文字のため、生データが変換に使用されます (;)), クロスサイトスクリプティングタグは同じであり、データの正規化を必要としません。HTML エンティティエンコーディング、パーセントエンコーディング、ASCII などの特殊文字の表現は、変換操作に対して評価されます。

Web App Firewall は、クロスサイトスクリプティング変換操作の属性名と値の両方を検査しなくなりました。ストリーミング時にクロスサイトスクリプティング属性名のみが変換されるようになりました。

クロスサイトスクリプティングタグの処理： NetScaler 10.5.e ビルド以降でのストリーミングの変更の一環として、クロスサイトスクリプティングタグの処理が変更されました。以前のリリースでは、いずれかの開き角かっこ（< ）、または角かっこを閉じる (>), または開き括弧と閉じ括弧の両方（< >) クロスサイトスクリプティング違反としてフラグが立てられました。この動作は、10.5.e ビルド以降で変更されました。開き括弧文字（<）のみ、または閉じ括弧文字（>）のみが存在することは、もはや攻撃とはみなされません。これは、開かっこ（<) is followed by a close bracket character (>)、クロスサイトスクリプティング攻撃にフラグが付けられる場合です。両方の文字が正しい順序で存在する必要があります（< に続く >) クロスサイトスクリプティング違反をトリガーします。

> 注：
>
> **SQL** 違反ログの変更メッセージ**:** 10.5.e ビルド以降のストリーミング変更の一環として、入力データをブロックで処理するようになりました。RegEx パターンマッチングは、連続した文字列マッチングで 4K に制限されるようになりました。この変更により、SQL 違反ログメッセージには、以前のビルドとは異なる情報が含まれる場合があります。入力内のキーワードと特殊文字は、多くのバイトで区切られています。アプライアンスは、データの処理時に、入力値全体をバッファリングするのではなく、SQL キーワードと特殊文字列を追跡します。ログメッセージには、フィールド名に加えて、SQL キーワード、SQL 特殊文字、または SQL キーワードと SQL 特殊文字の両方が含まれます。次の例に示すように、残りの入力はログメッセージに含まれなくなります。
>
> 例**:**
>
> 10.5 では、Web App Firewall が SQL 違反を検出すると、入力文字列全体が次のログメッセージに含まれることがあります。

> フィールドの SQL キーワードチェックに失敗しました **text=”select a name from testbed1;\(;\)”.*<blocked>**
>
> 11.0 では、次のログメッセージにフィールド名、キーワード、特殊文字（該当する場合）のみを記録します。
>
> フィールド`text=”select(;)”<blocked>`
> の SQL キーワードチェックに失敗しました
> この変更は、アプリケーション/**x-www-form-urlencoded**、マルチパート/フォームデータ、またはテキスト/**x-gwt-rpc** コンテンツタイプを含むリクエストに適用されます。**JSON** または **XML** ペイロードの処理中に生成されるログメッセージは、この変更の影響を受けません。

RAW POST ボディ： セキュリティチェック検査は、常に RAW POST ボディで行われます。

フォーム **ID**: Web App Firewall が挿入した「as_fid」タグは、フォームの計算されたハッシュで、ユーザーセッションで一意の長くなります。これは、ユーザーまたはセッションに関係なく、特定のフォームに対して同じ値です。

Charset: リクエストに文字セットがない場合、リクエストの処理時にアプリケーションプロファイルに指定されたデフォルトの文字セットが使用されます。

カウンタ**:**

ストリーミングエンジンとストリーミングエンジンのリクエストカウンターを追跡するために、プレフィックスが「se」と「appfwreq」のカウンターが追加されています。

```
nsconsmg -d statswt0 -g se_err_
```

```
nsconsmg -d statswt0 -g se_tot_
```

```
nsconsmg -d statswt0 -g se_cur_
```

```
nsconsmg -d statswt0 -g appfwreq_err_
```

```
nsconsmg -d statswt0 -g appfwreq_tot_
```

```
nsconsmg -d statswt0 -g appfwreq_cur_
```

`_err counters`: 成功したはずなのに、メモリ割り当ての問題またはその他のリソース不足のために失敗したまれなイベントを示します。

`_tot counters`: 増え続けるカウンター。

`_cur counters`: 現在のトランザクションからの使用量に基づいて変化し続ける現在の値を示すカウンター。

ヒント：

- Web App Firewall セキュリティチェックは、以前と同じように機能する必要があります。
- セキュリティー検査の処理には順序が設定されていません。
- 応答側の処理は影響を受けず、変更されません。
- クライアントレス VPN が使用されている場合、ストリーミングは使用されません。

> 重要：
>
> **Cookie** の長さの計算**:** リリース 10.5.e では、11.0 リリース (65.x より前のビルド) に加えて、Cookie ヘッ

> ダーを処理する Web App Firewall の方法が変更されました。アプライアンスは Cookie を個別に評価し、Cookie ヘッダー内の Cookie の長さが設定された長さを超えた場合、バッファオーバーフロー違反がトリガーされました。その結果、NetScaler 10.5 以前のリリースでブロックされたリクエストが許可される可能性があります。cookie ヘッダー全体の長さは、cookie の長さを決定するために計算されません。場合によっては、クッキーの合計サイズが受け入れられた値よりも大きくなり、サーバーが「400 Bad Request」で応答することがあります。
>
> 注**:**
> 変更は元に戻されました。NetScaler バージョン 10.5.e からバージョン 59.13xx.e およびそれ以降のビルドでの動作は、リリース 10.5 の非拡張ビルドに似ています。クッキーの長さを計算する際に、生の Cookie ヘッダー全体が考慮されるようになりました。クッキーの長さの決定には、スペースと、名前と値のペアを区切るセミコロン (;) 文字も含まれます。

セキュリティログを使用して **HTML** 要求をトレースする

October 7, 2021

> 注:
>
> この機能は、Citrix ADC リリース 10.5.e で使用できます。

トラブルシューティングには、クライアント要求で受信したデータの分析が必要であり、困難な場合があります。特に、アプライアンスを通過するトラフィックが多い場合。問題を診断すると機能に影響したり、アプリケーションのセキュリティに迅速な対応が必要な場合があります。

Citrix ADC は、Web App Firewall プロファイルのトラフィックを分離し、HTML リクエストの `nstrace` を収集します。appfw モードで収集された `nstrace` には、ログメッセージとともにリクエストの詳細が含まれます。トレースで「FollowTCPstream」を使用すると、ヘッダー、ペイロード、対応するログメッセージなど、個々のトランザクションの詳細を同じ画面に表示できます。

これは、あなたのトラフィックに関する包括的な概要を提供します。要求、ペイロード、および関連するログレコードの詳細を表示すると、セキュリティチェック違反を分析するのに役立ちます。違反を引き起こしているパターンを簡単に特定できます。パターンを許可する必要がある場合は、構成を変更するか、緩和ルールを追加するかを決定できます。

長所

1. 特定のプロファイルに対するトラフィックの分離：この拡張機能は、トラブルシューティングのために 1 つのプロファイルまたはプロファイルの特定のトランザクションのトラフィックを分離する場合に役立ちます。あなたは、もはやトレースで収集されたデータ全体をスキムしたり、大量のトラフィックで退屈することができ、要求の関心を分離するための特別なフィルタを必要とする必要はありません。好みのデータを表示できます。

2. 特定の要求のデータを収集する: トレースは、指定した期間だけ収集できます。必要に応じて、特定のトランザクションを分離、分析、デバッグするために、2 つのリクエストに対してのみトレースを収集できます。
3. リセットまたは中止を特定する: 予期しない接続の切断は簡単にはわかりません。—appfw モードで収集されたトレースは、Web App Firewall によってトリガーされたリセットまたは中止をキャプチャします。これにより、セキュリティチェック違反メッセージが表示されない場合に問題をすばやく特定できます。不正な要求や、Web App Firewall によって終了された RFC 準拠以外の要求が、識別しやすくなりました。
4. 復号化された **SSL** トラフィックを表示する: HTTPS トラフィックはプレーンテキストでキャプチャされるため、トラブルシューティングが容易になります。
5. 包括的なビューを提供します: 要求全体をパケットレベルで確認したり、ペイロードをチェックしたり、ログを調べて、どのセキュリティチェック違反がトリガーされているかを確認したり、ペイロード内の一致パターンを特定したりできます。ペイロードが予期しないデータ、ジャンク文字列、または印刷不可能な文字（ヌル文字、\r または \n など）、それらはトレースで簡単に見つけることができます。
6. 構成を変更する: デバッグは、観察された動作が正しい動作であるか、構成を変更する必要があるかを判断するために役立つ情報を提供できます。
7. 応答時間を短縮します: ターゲットトラフィックのデバッグを高速化すると、応答時間を改善して、Citrix エンジニアリングおよびサポートチームによる説明または根本原因分析を提供できます。

詳細については、「コマンドラインインターフェイスを使用した手動設定 」を参照してください。

コマンドラインインターフェイスを使用してプロファイルのデバッグトレースを構成するには

手順 1. ns トレースを有効にします。

show コマンドを使用して、設定された設定を確認できます。

- `set appfw profile <profile> -trace ON`

手順 2. トレースを収集します。`nstrace` コマンドに適用可能なすべてのオプションを引き続き使用できます。

- `start nstrace -mode APPFW`

手順 3. トレースを停止します。

- `stop nstrace`

トレースの場所: `nstrace` は、に作成されたタイムスタンプ付きのフォルダーに保存されます。/var/nstrace ディレクトリであり、`wireshark`を使用して表示できます。`/var/log/ns.log`を末尾に付けて、新しいトレースの場所に関する詳細を提供するログメッセージを表示できます。

ヒント:

- appfw モードオプションを使用すると、`nstrace` は「nstrace」が有効になっている 1 つ以上のプロファイルのデータのみを収集します。
- プロファイルでトレースを有効にしても、「start ns trace」コマンドを明示的に実行してトレースを収集するまで、トレースの収集は自動的に開始されません。
- プロファイルでトレースを有効にしても、Web App Firewall のパフォーマンスに悪影響はないかもしれませんが、データを収集する期間のみこの機能を有効にすることをお勧めします。トレースを収集した後、—trace

フラグをオフにすることをお勧めします。このオプションは、過去にこのフラグを有効にしたプロファイルから誤ってデータを取得するリスクを防ぎます。

- トランザクションレコードのセキュリティチェックを nstraceに含めるには、ブロックアクションまたはログアクションを有効にする必要があります。
- プロファイルのトレースが「オン」の場合、セキュリティチェックアクションとは関係なく、リセットとアボートがログに記録されます。
- この機能は、クライアントから受信した要求のトラブルシューティングにのみ適用できます。—appfw モードのトレースには、サーバから受信した応答は含まれません。
- nstrace コマンドに適用可能なすべてのオプションを引き続き使用できます。例:

```
start nstrace -tcpdump enabled -size 0 -mode appFW
```

- リクエストが複数の違反をトリガーした場合、そのレコードの nstrace には、対応するすべてのログメッセージが含まれます。
- この機能では、CEF ログメッセージ形式がサポートされています。
- リクエスト側のチェックのブロックまたはログアクションをトリガーする署名違反もトレースに含まれます。
- HTML (非 XML) 要求のみがトレースに収集されます。

クラスター構成の **Web App Firewall** サポート

October 7, 2021

> 注:
>
> ストライプ構成と部分的にストライプ構成のための Citrix Web App Firewall は、Citrix ADC 11.0 バージョンで導入されました。

クラスターは、単一のシステムとして構成および管理される Citrix ADC アプライアンスのグループです。クラスタ内の各アプライアンスをノードと呼びます。構成がアクティブになっているノードの数に応じて、クラスター構成は、ストライプ構成、部分ストライプ構成、またはスポット構成と呼ばれます。Web App Firewall は、すべての構成で完全にサポートされています。

クラスタ構成におけるストライピング仮想サーバと部分的にストライピングされた仮想サーバのサポートの主なメリットは、次のとおりです。

1. セッションフェールオーバーのサポート: ストライピングされた仮想サーバ構成および部分的にストライプされた仮想サーバ構成では、セッションフェールオーバーがサポートされます。Web App Firewall の高度なセキュリティ機能 (開始 URL の閉鎖、フォームフィールドの一貫性のチェック、維持、トランザクション処理中のセッションの使用など)。高可用性構成またはスポットされたクラスター構成では、Web App Firewall トラフィックを処理しているノードに障害が発生すると、すべてのセッション情報が失われ、ユーザーはセッシ

ョンを再確立する必要があります。ストライプ仮想サーバ構成では、ユーザー・セッションは複数のノードにまたがってレプリケートされます。ノードがダウンすると、レプリカを実行しているノードが所有者になります。セッション情報は、ユーザーに目に見える影響を与えることなく維持されます。

2. スケーラビリティ：クラスタ内のすべてのノードがトラフィックを処理できます。クラスタの複数のノードが、ストライピングされた仮想サーバによって処理される着信要求を処理できます。これにより、Web App Firewall の複数の同時要求を処理する能力が向上し、全体的なパフォーマンスが向上します。

セキュリティチェックと署名保護は、追加のクラスター固有の Web App Firewall 構成を必要とせずに展開できます。構成コーディネータ (CCO) ノードで通常の Web App Firewall 構成を行い、すべてのノードに伝播できます。

> 注:
>
> セッション情報は複数のノードにわたってレプリケートされますが、ストライプ構成のすべてのノードにわたってレプリケートされるわけではありません。したがって、フェールオーバーサポートでは、同時に発生する障害の数に制限があります。複数のノードで同時に障害が発生した場合、セッションが別のノードに複製される前に障害が発生すると、Web App Firewall でセッション情報が失われる可能性があります。

ハイライト

- Web App Firewall は、クラスターデプロイメントにおけるスケーラビリティ、高スループット、およびセッションフェイルオーバーのサポートを提供します。
- すべての Web App Firewall セキュリティチェックとシグニチャ保護は、すべてのクラスター構成でサポートされています。
- キャラクタマップは、クラスタではまだサポートされていません。学習エンジンは、フィールド形式のセキュリティチェックの学習ルールでフィールドタイプを推奨しています。
- 統計と学習されたルールは、クラスタ内のすべてのノードから集約されます。
- 分散ハッシュテーブル (DHT) は、セッションのキャッシュを提供し、複数のノード間でセッション情報をレプリケートする機能を提供します。仮想サーバーに要求が送信されると、Citrix ADC アプライアンスは DHT に Web App Firewall セッションを作成し、DHT からセッション情報を取得することもできます。
- クラスタリングは Advanced および Premium ライセンスでライセンスされます。この機能は、Standard ライセンスでは使用できません。

デバッグとトラブルシューティング

October 7, 2021

Web App Firewall の各機能に関連する次のトラブルシューティングおよびデバッグ情報を参照してください。

- アプリケーションファイアウォール-CPU が高い
- メモリ
- 大容量ファイルのアップロードの失敗

- トレーニング
- 署名
- トレースログ
- その他

高い CPU

October 7, 2021

次に、機能と CPU 使用率が高い CPU 関連のデバッグに関する問題のいくつかと、Web App Firewall を使用する際に従うべきベストプラクティスを示します。

ポリシーヒット、バインディング、ネットワーク設定、**Web App Firewall** 設定を確認します。

- 構成ミスの特定
- 影響を受けるトラフィックを処理している *vserver* の特定

次のログファイルのログで、セキュリティ違反と最近の設定変更がないか調べます。

- /var/log/ns.log
- /var/nslog/**import**.log
- /var/nslog/aslearn.log
- tail -f /var/log/ns.log | grep APPFW_SIGNATURE_MATCH

例:

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW| APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method
    =GET request= http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=
    Signature violation rule ID 807: web-cgi /wwwboard/passwd.txt access
     cn1=140 cn2=841 cs1=pr_ffc cs2=PPE0 cs3=
    OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
    not blocked
<!--NeedCopy-->
```

影響を受けるトラフィックを分離します。

- プロファイルを分離する
- セキュリティチェックを分離する
- URL、仮想サーバー、トラフィックパラメータを分離します

条件付きプロファイルレベルのトレースは、トラフィックおよび違反レコードを識別するのに役立ちます。

- set appfw profile <profile> -trace ON

- `start nstrace -mode APPFW -size 0`
- `stop nstrace`

注：トレースは、-size 0 オプションを使用して収集されることを確認してください。

appfw、**dht**、**IP** レピュテーションアクティビティカウンタを確認します。

- `nsconmsg -g as_ -g appfwreq_ -g iprep -d current`

接続中のリセットのモニタ・ウィンドウ・サイズ：

http メッセージが無効なために Citrix ADC が接続をリセットすると、Appfw はウィンドウサイズを 9845 に設定します。

例：

- 不正な形式の要求を受信しました-接続のリセット
- 高い CPU 関連の問題
- システムの制限については、データシートを確認してください。
- CPU の使用状況、appfw、DHT、およびメモリ関連のアクティビティを検査します。appfw セッションの監視
- nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -g mem_AS_OBJ -g mem_AS_COMPONENT -d current

ターゲット期間中に割り当てられ、**Web App Firewall** コンポーネントおよびオブジェクトから解放されたメモリを監視します。これは、高い CPU 使用率につながる保護を分離するのに役立ちます。

- プロファイラの出力
- ログの観察

高 **CPU** につながる **appfw** チェックを分離する：

- startURLClosure
- Formfiledconsistency
- CSRF
- Cookie protections
- リファラヘッダーチェック

シグニチャの自動更新によって **CPU** が高くならないことをご確認ください（確認するには無効にしてください）。

メモリ

October 7, 2021

以下は、Web App Firewall の使用メモリ関連の問題が発生した場合に従うべきベストプラクティスの一部です。

nsconmsg コマンドの使用方法**:**

- 次のコマンドを実行して、システム内に十分なメモリがあり、メモリ割り当てに失敗していないことを確認するために、グローバルメモリ統計情報を調べます。

  ```
  * *- nsconmsg -d memstats
  ```

- 次のコマンドを実行して、appsecure、IP レピュテーション、キャッシュ、および圧縮の現在割り当てられているメモリ制限と最大メモリ制限を確認します。

  ```
  nsconmsg -d memstats | egrep -i APPSECURE|IPREP|CACHE|CMP
  ```

- 次のコマンドを実行して、appfw、DHT、IP レピュテーションアクティビティカウンタを確認します。

  ```
  nsconmsg -g as -g appfwreq_ -g iprep -d current
  ```

- 次のコマンドを実行して、すべての Web App Firewall エラーカウンタを確認します。

  ```
  nsconmsg -g as_ -g appfwreq_ -g iprep_ -d stats | grep err
  ```

- 次のコマンドを実行して、すべてのシステムエラーカウンタをチェックします。

  ```
  nsconmsg -g err -d current
  ```

- 次のコマンドを実行して、CPU、APPFWREQ、AS、および DHT カウンタを調べます。

  ```
  nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -d current
  ```

- 次のコマンドを実行して、設定済みのキャッシュメモリを確認します。
- `show cacheparameter`
- 次のコマンドを実行して、設定されたメモリを確認します。

  ```
  nsconmsg -d memstats | egrep -i CACHE
  ```

- Web App Firewall コンポーネントとオブジェクトでのメモリの分散を特定します。

AS_OBJ_ memory 表示:

```
nsconmsg -K newnslog -d stats | grep AS_OBJ | egrep -v AppFW_cpu0|total |
sort -k3
```

AS_COMPONENT_ memory 表示:

```
nsconmsg -K newnslog -d stats | grep AS_COMPONENT | egrep -v AppFW_cpu0|
total | sort -k3
```

次のコマンドを実行して、アライブセッション数を確認します。

アクティブセッション数をモニタ/プロット:

```
nsconmsg -g as_alive_sessions -d current
```

割り当てられたセッションの合計、空き時間、更新されたセッションの監視/プロット:

- `nsconmsg -g as_tot_alloc_sessions -g as_tot_free_sessions -d current`
- `nsconmsg -g as_tot_update_sessions -d current`

必要に応じて、次のコマンドを実行して、セッションのタイムアウトを減らし、セッション制限が使用されないようにします。

```
set appfwsettings -sessionTimeout <300>
```

必要に応じて、次のコマンドを実行して、セッションの最大有効期間を設定します。

```
set appfwsettings -sessionLifetime <7200>
```

割り当て済みメモリと使用済みメモリの確認

割り当て済みメモリと使用済みメモリの合計を確認するには、次の手順に従います。

- **nsconmsg —d memstats** コマンドを使用します。[**MEM_APPSECURE**] フィールドを確認します。
- **stat appfw** コマンドを使用して、消費量の情報を取得します。

Web App Firewall は、一定期間またはサイズが経過してもログを自動的に削除しません。

- `All AppFw logs are archived in the */var/log/ns.log*` ファイル。*ns.log* ファイルは、ロールオーバータスクを実行します。

詳細については、次のリンクを参照してください。<http://support.citrix.com/article/CTX121898>

Web App Firewall メモリを増やす:

- Web App Firewall メモリを増やすための CLI オプションはありません。Web App Firewall メモリは、プラットフォーム固有です。
- *nsapimgr* オプションを使用してメモリを増やすことはできますが、推奨しません。

Web App Firewall の最大許容メモリはプラットフォームによって決まり、IC を無効にしてもメモリ割り当てには影響しません。

サイズの大きいファイルのアップロードの失敗

October 7, 2021

サイズの大きいファイルのアップロードに失敗した場合は、次の点を確認してください。

— アプリケーションファイアウォールのポストボディ制限の設定の誤り
— ファイルアップロードスキャンが有効になり、処理時間が長くなります。
— システムの制限に達する。

20 MB を超えるペイロードの場合は、アプリケーションファイアウォールプロファイルでストリーミングを有効にすることをお勧めします。また、ストリーミングを有効にする前に、バックエンドサーバーがチャンク要求をサポートしていることを確認する必要があります。

リリース 11.0 以降では、次のコマンドを実行してバッファリングを回避するために、プロファイルごとにストリーミングフラグを有効にできます。

```
set appfw profile <profile name> -streaming on
```

トレーニング

October 7, 2021

ラーニング機能の問題に遭遇した場合に推奨されるベストプラクティスのいくつかを次に示します。

学習プロセス:

- プロセス *aslear* y が実行中であることを確認します。
- トップコマンドの出力をチェックする
- 次のコマンドを実行して、ps コマンドの出力を確認します。

  ```
  ps -ax | grep aslearn | grep -v "grep"
  ```

  例:

  ```
  root@ns# ps -ax | grep aslearn | grep -v "grep"
  1439  ??  Ss     0:03.86 /netscaler/aslearn -start -f /netscaler/
     aslearn.conf
  <!--NeedCopy-->
  ```

- *ns.log* ファイルを確認して、観察された問題の前に実行された最近の構成コマンドを特定します。

  ```
  /var/log/ns.log
  ```

- aslearn ログを調べて、aslearn メッセージを確認します。

  ```
  /var/log/aslearn.log
  ```

- 影響を受けるプロファイルとセキュリティチェックを分離する
- 次のコマンドを実行して、失敗した GUI コマンドと CLI コマンドを特定します。

  ```
  show appfw learningdata <profileName> <securityCheck>
  ```

  例:

  - `show learningdata test_profile starturl`
  - `show learningdata test_profile crosssiteScripting`
  - `show learningdata test_profile sqLInjection`
  - `show learningdata test_profile csRFtag`

  - show learningdata test_profile fieldformat
  - show learningdata test_profile fieldconsistency

- bsd シェルプロンプトから sqlite の整合性チェックを実行します。

  ```
  nsshell ## sqlite3 /var/nslog/asl/<profile_name_in_lowercase>.db '
  pragma integrity_check;
  ```

  例:

  ```
  1  root@ns# sqlite3 /var/nslog/asl/tsk0247284.db 'pragma
     integrity_check;'
  2  ok
  3  <!--NeedCopy-->
  ```

- ルールを展開または削除して、学習を再開します。
  - 保護ごとに 2000 個の学習項目に達した場合、その保護の学習はこれ以上開始できません。
  - データベースのサイズが 20 MB に達した場合、すべての保護の学習を停止します。
  - 学習プロセスを再起動

  ```
  */netscaler/aslearn -start -f/netscaler/aslearn.conf*
  ```

- 次のコマンドを実行して、/var フォルダ内のスペースを確認します。

  ```
  du -h /var
  ```

- 次のコマンドを実行して、学習しきい値制限を確認します。

  ```
  show appfwlearningsettings <profile_name> <securityCheck>
  ```

- 次のコマンドを実行して、学習したデータを収集します。

  ```
  export appfwlearningdata <profile_name> <securityCheck>
  ```

- 学習したデータがコレクタにアップロードされていることを認識します。

署名

October 7, 2021

署名の概要

署名を追加するには、次の手順に従います。

1. 「デフォルト」署名を選択し、「追加」をクリックしてコピーを作成します。

2. わかりやすい名前を付けます。新しい sig オブジェクトがユーザ定義オブジェクトとして追加されます。
3. 特定のニーズに関連するターゲットルールを有効にします。
    - ルールはデフォルトで無効になっています。
    - より多くのルールにより多くの処理が必要
4. アクションを設定します。

    ブロックアクションとログアクションはデフォルトで有効になっています。統計は別のオプションです
5. プロファイルで使用する署名を設定します。

署名を使用する際のヒント

- アプリケーションの保護に適用可能なシグネチャのみを有効にして、処理のオーバーヘッドを最適化します。
- シグニチャの一致をトリガーするには、ルール内のすべてのパターンが一致する必要があります。
- 独自のカスタマイズされたルールを追加して、着信要求を検査し、SQL インジェクション攻撃やクロスサイトスクリプティング攻撃など、さまざまな種類の攻撃を検出できます。
- また、応答を検査するルールを追加して、クレジットカード番号などの機密情報の漏洩を検出してブロックすることもできます。
- 複数のセキュリティチェック条件を追加して、独自のカスタマイズされたチェックを作成します。

署名を使用するためのベストプラクティス

次に、署名に関連する問題に遭遇したときに従うことができるベストプラクティスのいくつかを示します。

- プライマリとセカンダリの両方で import コマンドが成功したことを確認します。
- CLI と GUI の出力が一貫していることを確認します。
- ns.log をチェックして、シグニチャのインポートおよび自動更新中に発生したエラーを特定します。
- DNS ネームサーバが正しく設定されていることを確認します。
- スキーマバージョンの非互換性をチェックします。
- デバイスが自動更新のために AWS でホストされている署名更新 URL にアクセスできないかどうかを確認します。
- デフォルトシグニチャとユーザが追加したシグニチャのバージョンの不一致をチェックします。
- プライマリノードとセカンダリノードのシグニチャオブジェクトのバージョン不一致をチェックします。
- 高い CPU 使用率を監視します（シグニチャ更新の問題を除外するには、自動更新を無効にします）。

トレースログ

October 7, 2021

トレース・ログを記録するには、次の手順に従います。

1. プロファイルのトレースを有効にします。show コマンドを使用して、設定された設定を確認できます。

```
set appfw profile <profile> -trace ON
```

1. トレースの収集を開始します。nstrace コマンドに適用可能なすべてのオプションを引き続き使用できます。

```
start nstrace -mode APPFW
```

1. トレースの収集を停止する

```
stop nstrace
```

トレースの場所：nstrace は、に作成されたタイムスタンプ付きのフォルダーに保存されます。/var/nstrace ディレクトリであり、wireshark を使用して表示できます。**/var/log/ns.log* を末尾に付けて、新しいトレースの場所に関する詳細を提供するログメッセージを表示できます。

トレースログの利点:

- 特定のプロファイルのトラフィックを分離
- 特定のリクエストのデータを収集する
- リセットまたは中止の識別
- 復号化された SSL トラフィックを表示する: HTTPS トラフィックはプレーンテキストでキャプチャされるため、トラブルシューティングが容易になります。
- 包括的なビューを提供します。パケットレベルでの要求全体の表示、ペイロードの確認、ログの表示によるセキュリティチェック違反の確認、およびペイロード内の一致パターンの識別が可能です。ペイロードが予期しないデータ、ジャンク文字列、または印刷不可能な文字（ヌル文字、r または n など）で構成されている場合、トレース内で簡単に検出できます。
- 応答時間の短縮：ターゲットトラフィックのデバッグを高速化し、根本原因分析を行います。

その他

October 7, 2021

Web App Firewall の使用時に発生する可能性のある問題の解決方法は次のとおりです。

- Web App Firewall は、無効な http メッセージの接続をリセットするときに、ウィンドウサイズを 9845 に設定します。

— 不正な形式の要求を受信しました- [接続のリセット無効なコンテンツ長ヘッダーを送信するクライアント/サーバ]
— 要求ヘッダーに不明なコンテンツタイプ

- システム制限：アプリケーションがフリーズしているように見える

  — 最大セッション制限に達したときに発生します。(100K)
  — 操作に必要なシステムメモリが少なくなります。

  - IP レピュテーション機能が機能しない
    — レピュテーション機能を有効にした後、iprep プロセスの開始に約 5 分かかります。その間、IP レピュテーション機能が機能しない可能性があります。

- 予期しない Web App Firewall 違反がトリガーされています

  — セッションのタイムアウトのデフォルト値は 900 秒です。セッションタイムアウトが低い値に設定されている場合、ブラウザはセッション化（CSRF、FFC など）に依存するチェックに対して誤検出を引き起こす可能性があります。セッションタイムアウトを確認し、セッション ID （CEF ログの cs3）を確認します。sessionID が異なる場合は、セッションタイムアウトが原因である可能性があります。
  — フォームが javascript によって動的に生成される場合、偽の FFC 違反を引き起こす可能性があります。

- FFC 違反ログに空のフィールド名 (11.0 より前)

  これは、セッションでフォームに含まれていないフォームフィールドに出くわすシナリオで見ることができます。

  この問題が発生するシナリオ:

  — フォームがクライアントに送信されてから受信されたときからセッションがタイムアウトしました。
  — フォームは Java スクリプトを使用してクライアント側で生成されました。

参照ドキュメント

January 25, 2022

Web App Firewall の機能については、次のリソースを参照してください。

- Citrix Web App Firewall がアプリケーションデータトラフィックをどのように変更するか。
- Citrix ADC アプライアンスでの Web App Firewall セキュリティ違反による HTML リクエストのトレースとログの仕組み
- トップレベルの保護
- セキュリティ緩和
- アプリケーションの構成とデプロイに関する情報:

  - Application
  - ファイアウォール
  - ログ
- シグネチャ更新記事
- ボット管理

署名アラートの記事

October 7, 2021

Citrix Web アプリケーションファイアウォール（WAF）は、アプライアンスにダウンロードして適用できるシグニチャの更新をアナウンスします。セキュリティ攻撃を検出すると、新しいシグニチャの更新に関する電子メール通知が送信されます。署名をダウンロードして、アプライアンスに適用できます。

署名アラート通知の受信方法

October 7, 2021

この記事では、新しい署名の更新に関する電子メール通知を受信するように署名アラート設定を構成する方法について説明します。

概要

ネットワーク管理者は、新しい Web アプリケーションファイアウォールの署名の更新と通知に関する電子メール通知を受信することを希望しています。

問題

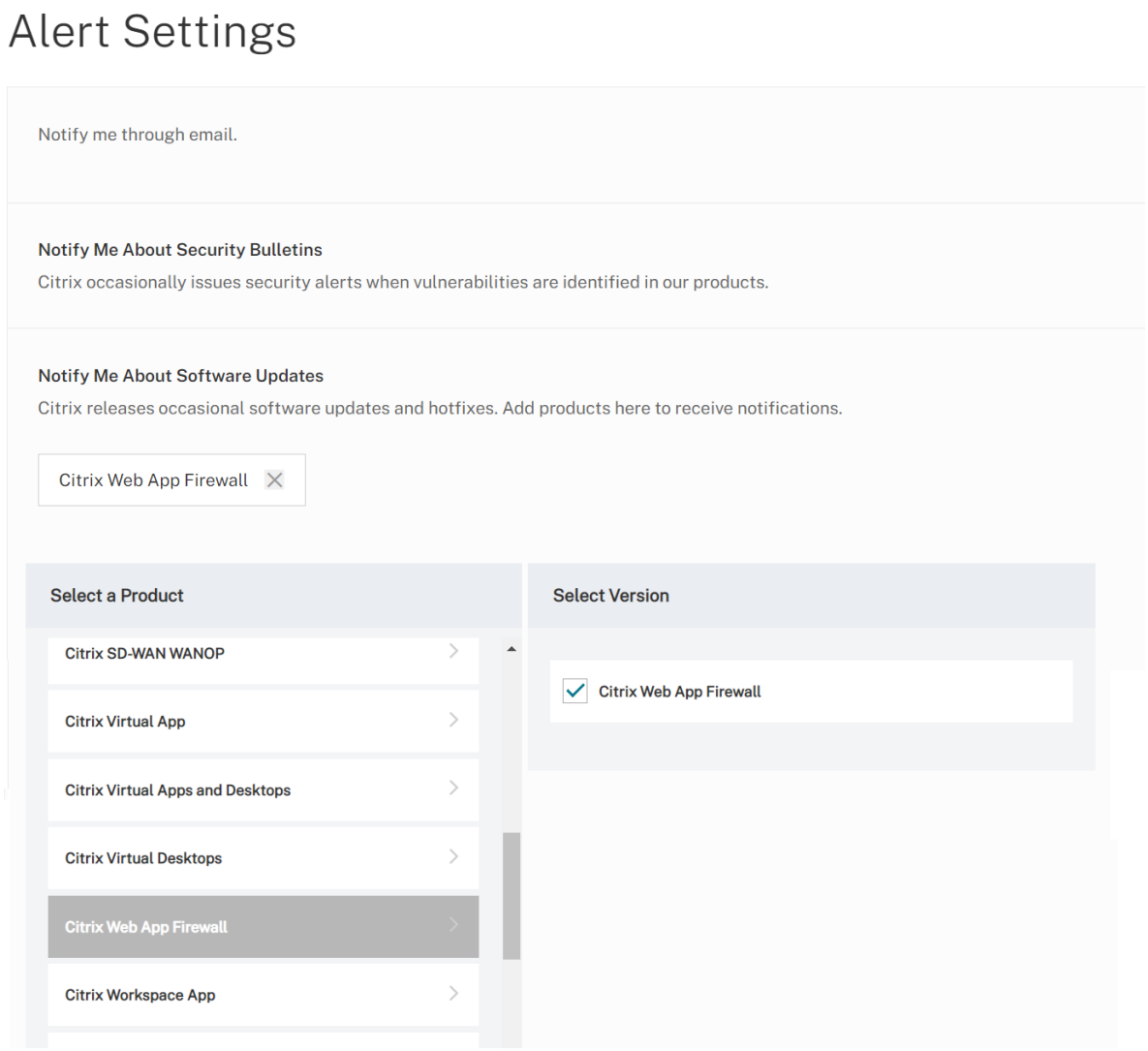
Web アプリケーションファイアウォールで新しい署名が使用可能になったときに通知を受けるネットワーク管理者は、電子メールによる通知を選択できます。新しい署名がダウンロード可能になると、管理者に電子メール通知が送信されます。
ネットワーク管理者は、新しいシグニチャ更新の電子メール通知を受信できます。

解決策

新しい署名の更新に関する電子メール通知を受信するには、次の手順に従います。

1. Citrix サポート Web サイト`https://support.citrix.com/user/alerts`にサインインします。

2. [警告の設定] セクションで、[電子メールで通知する] オプションを有効にします。
3. [製品の追加] を選択して、製品カタログを表示します。
4. [**Citrix Web App Firewall]** をクリックし、[**Citrix Web AppFirewall]** チェックボックスをオンにします。
5. [設定を保存] をクリックします。



1. [警告の設定] セクションで、[電子メールで通知する] オプションを有効にします。
2. [製品の追加] を選択して、製品カタログを表示します。
3. [アプリケーションファイアウォール] をクリックし、[署名] チェックボックスをオンにします。
4. [設定を保存] をクリックします。

署名更新バージョン **27**

January 25, 2022

バージョン 27 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999921	cve-2018-1002000	Web-miscWordPress Arigato オートレスポンダーとニュースレター SQL インジェクションの脆弱性。
999920		WEB-MISCWordPress プラグインコーナー広告 1.0.7-ストアドクロスサイトスクリプティング
999919	cve-2018-1002009	WEB-MISCWordPress Arigato 自動応答とニュースレター bft_unsubscribe クロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002002	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。

署名ルール	**CVE ID**	説明
999918	cve-2018-1002003	WEB-MISCWordPressArigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002004	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002005	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002006	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999918	cve-2018-1002007	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999917	cve-2018-1002001	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999917	cve-2018-1002008	WEB-MISCWordPress Arigato 自動応答とニュースレターの複数のクロスサイトスクリプティングの脆弱性。
999916	cve-2018-8719	WEB-MISCWordPress プラグイン WP セキュリティ監査ログ-wp-content/uploads/wp-security 監査ログ/* 無制限アクセス
999915	cve-2019-7743	WEB-MISC-Joomla phar://ストリームラッパーオブジェクトインジェクションの脆弱性アップロードされた非 phar ファイルの実行

署名ルール	**CVE ID**	説明
999914		WEB-MISCWordPress プラグイン電子メール購読者とニュースレター 3.4.7 情報漏えいの脆弱性
999913		WEB-MISCWordPress プラグイン AD マネージャー WD v1.0.11-wd_ads_admin_class.php 任意のファイルのダウンロード
999912		Web-IISMicrosoft IIS-ショートファイル/フォルダ名の開示

署名更新バージョン **28**

January 25, 2022

バージョン 28 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名バージョン 28 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999898	CVE-2018-12895	4.9.7 より前の WEB-MISC WordPress ディレクトリトラバーサルの脆弱性。

署名ルール	**CVE ID**	説明
999899	CVE-2019-9618	Web-MISC-GraceMedia メディアプレーヤー WordPress プラグイン 1.0 任意のローカルファイルインクルード脆弱性
999900	CVE-2018-20714	WEB-MISC WordPress プラグイン WooCommerce 3.4.6 より前-ファイル削除の脆弱性。
999901	CVE-2018-11868	2.3.7 より前の WEB-MISC FlowPaper FlexPaper では、リモートでのコード実行-設定ファイルのリセットを許可できます。
999902	CVE-2018-11868	2.3.7 より前の WEB-MISC FlowPaper FlexPaper では、リモートコード実行を許可できます
999903	CVE-2019-9184	ウェブその他 Joomla! 3.3.7 より前の J2Store プラグイン 3.x は SQL インジェクションを可能にします。
999904	CVE-2019-9168	WEB-MISC WordPress プラグイン WooCommerce Photoswipe キャプションを介して 3.5.5 クロスサイトスクリプティングの前に。
999905		WEB-MISC WordPress プラグイン WooCommerce で保存されたクロスサイトスクリプティング用の 5.1.3 より前の放棄されたカート。
999906	CVE-2019-8942	4.9.9 より前の WEB-MISCWordPress と 5.0.1 のリモートコード実行前の 5.x。
999907	CVE-2019-8942	4.9.9 より前の WEB-MISC WordPress と 5.0.1 のリモートコード実行前の 5.x。

署名ルール	**CVE ID**	説明
999908	CVE-2019-8942	WEB-MISC 4.9.9 より前の WordPress と 5.0.1 より前の 5.x のリモートコード実行
999909	CVE-2017-16562	web-misc-デラックステーマ UserPro WordPress プラグイン up_auto_log=true パラメータによるセキュリティバイパスの脆弱性
999910	CVE-2018-20782	WEB-MISC WordPress プラグイン GloBee 1.1.2 より前 WooCommerce IPN メッセージスプーフィング
999911	CVE-2019-6340	Drupal-Drupal Core 8 RESTful Web サービスでの任意のリモートコード実行

署名更新バージョン **29**

January 25, 2022

バージョン 29 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 29 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999896	CVE-2019-2725	Weblogic 10.3.6 リモートでのコード実行
999897	CVE-2019-2725	Weblogic 10.3.6 リモートでのコード実行

署名更新バージョン **30**

January 25, 2022

バージョン 30 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 30 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999879	<>	WEB-MISC WordPress プラグイン WooCommerce チェックアウトマネージャー-任意のファイルアップロードの脆弱性
999880	<>	WEB-MISC WordPress プラグインアドバンスお問い合わせフォーム 7 DB 1.6.1 より前-SQL インジェクションの脆弱性

署名ルール	**CVE ID**	説明
999881	<>	WEB-MISC WordPress プラグイン 1.0.67 より前のお問い合わせフォームビルダー-ローカルファイルインクルードの脆弱性
999882	<>	SQL HTTP URI ブラインドインジェクション試行
999883	<>	WEB-MISC WEB-MISC Loco Translate WordPress プラグイン 2.1.1 以前-ローカルファイルインクルージョンの脆弱性
999884	<>	3.4 より前の WEB-MISC WordPress プラグインの重複ページ-SQL インジェクションの脆弱性
999885	CVE-2019-0232	MS Windows で enableCmd-LineArguments=true の場合 WEB-MISC Apache Tomcat RCE Via .CMD CGI スクリプト
999886	CVE-2019-0232	WEB-MISC Apache Tomcat RCE 経由.BAT 経由 CGI スクリプト有効にすると CMdlineArguments=True (MS Windows の場合)
999887	CVE-2019-10692	WWEB-MISC WordPress プラグイン wp-google-maps 7.11.18 より前-SQL インジェクションの脆弱性。
999888	CVE-2019-10946	WEB-MISC Joomla! 3.9.5 より前-セキュリティバイパスの脆弱性
999889	CVE-2019-10945	WEB-MISC Joomla! 3.9.5 より前：ディレクトリトラバーサルの脆弱性

署名ルール	**CVE ID**	説明
999890	CVE-2019-9912	WEB-MISC WPGoogleMaps 7.10.41 より前の WordPress プラグイン反射型クロスサイトスクリプティングの脆弱性
999890	CVE-2019-9912	WEB-MISC WPGoogleMaps 7.10.41 より前の WordPress プラグイン反射型クロスサイトスクリプティングの脆弱性
999891	CVE-2019-9911	WEB-MISC WordPress プラグインソーシャルネットワーク 4.2.8 より前の自動ポスター-反射型クロスサイトスクリプティングの脆弱性
999892	CVE-2019-9908	WEB-MISC WordPress プラグイン Font_Organizer 2.1.1-反射型クロスサイトスクリプティング
999893	CVE-2019-9787	4.9.7 より前の WEB-MISC WordPress-リモートでコードが実行される脆弱性
999894	CVE-2019-9568	WEB-MISC Forminator お問い合わせフォーム、投票、クイズビルダー WordPress プラグイン 1.6 より前のブラインド SQLi 脆弱性
999895	CVE-2019-9567	WEB-MISC Forminator お問い合わせフォーム、投票、クイズビルダー 1.6 より前の WP プラグイン永続的なクロスサイトスクリプティングの脆弱性
999877	CVE-2018-20062	WEB-MISC noneCMS V1.3-ThinkPHP フィルタ任意の PHP コード実行の脆弱性
999878	CVE-2019-9082	5.1.32 より前の ThinkPHP 5.x での WEB-MISC リモートコード実行の脆弱性

署名更新バージョン **32**

January 25, 2022

バージョン 32 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 32 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999875	CVE-2016-4438, CVE-2016-3087	WEB-STRUTS アパッチストラット 2.3.20 から 2.3.28.1 URL 経由でのリモート実行の脆弱性
999876	CVE-2019-10867	5.7.1 より前の WEB-MISC Pimcore-デシリアライズの脆弱性（CVE-2019-10867）

署名更新バージョン **33**

January 25, 2022

バージョン 33 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 33 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

規則	**CVE**	説明	脆弱性リファレンス
999860		WordPress プラグイン Yuzo 関連記事クロスサイトスクリプティング脆弱性	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild
999861	CVE-2019-12099		cve,2019-12099
999862		WordPress plug-in Database Backup <= 5.2 - Remote Code Execution	https://www.wordfence.com/blog/2019/05/os-command-injection-vulnerability-patched-in-wp-database-backup-plug-in

規則	**CVE**	説明	脆弱性リファレンス
999863		WordPress plug-in Slick Popup - Privilege Escalation	https://www.wordfence.com/blog/2019/05/privilege-escalation-flaw-present-in-slick-popup-plug-in
999864	CVE-2019-10866	WordPress plug-in Form Maker 1.13.3 - SQL Injection	cve,2019-10866
999865		WordPress plug-in Give – Stored cross-site scripting for Donors	https://blog.sucuri.net/2019/05/wordpress-plug-in-give-stored-xss-for-donors.html
999866		WordPress plug-in My Calendar <= 3.1.9 - Unauthenticated cross-site scripting Vulnerability	https://wpvulndb.com/vulnerabilities/9267
999867		WordPress plug-in Slimstat <= 4.8 - Unauthenticated Stored cross-site scripting	https://blog.sucuri.net/2019/05/slimstat-stored-xss-from-visitors.html
999868	CVE-2019-2618	WebLogic Arbitrary Upload Vulnerability	cve,2019-2618
999869	CVE-2019-11871	2.5.15 より前の WEB-WORDPRESS WordPress プラグインのカスタムフィールドスイート-クロスサイトスクリプティングの脆弱性	cve,2019-11871

規則	**CVE**	説明	脆弱性リファレンス
999870		WEB-WORDPRESS WordPress ライブチャットサポートプラグイン永続的なクロスサイトスクリプティング 8.0.27 より前の wplc_custom_js パラメータによる脆弱性	https://blog.sucuri.net/2019/05/persistent-cross-site-scripting-in-wp-live-chat-support-plug-in.html
999871		0.9.7.4 より前の WEB-WORDPRESS WordPress プラグイン W3 合計キャッシュ-PHAR リモートでコードが実行される脆弱性	https://wpvulndb.com/vulnerabilities/9270
999872		0.9.7.4 より前の WEB-WORDPRESS WordPress プラグイン W3 合計キャッシュ-PHAR リモートでコードが実行される脆弱性	https://wpvulndb.com/vulnerabilities/9269
999873	CVE-2019-0604	WEB-MISC Microsoft ソフト Windows Sharepoint サーバー-リモートでコードが実行される脆弱性	cve,2019-0604
999874		WEB-WORDPRESS 雄三関連記事 5.12.91 における認証されていないストアドクロスサイトスクリプティングの脆弱性	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild

署名更新バージョン **34**

January 25, 2022

バージョン 34 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 34 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999843		WEB-WORDPRESS WordPress プラグインバージョン 2.0.46 より前のアルティメットメンバー-任意のファイルを読み込む設定
999844		WEB-WORDPRESS WordPress プラグインバージョン 2.0.46 より前のアルティメットメンバー-任意のファイルを読み
999845		WEB-WORDPRESS WordPress プラグインバージョン 2.0.46 より前の究極のメンバー-ファイル置換によるファイル削除
999846		WEB-WORDPRESS WordPress のプラグインバージョン 2.0.46 より前の究極のメンバー-ファイルの削除

署名ルール	**CVE ID**	説明
999847		2.1.10 より前の WEB-WORDPRESS WordPress プラグインショートリンク-CSV インジェクションの脆弱性
999848		WEB-WORDPRESS WordPress プラグイン 2.1.10 より前のショートリンク-認証されていないストアドクロスサイトスクリプティングの脆弱性
999849		7.3.13.727 より前の WEB-WORDPRESS WordPress プラグイン FV Flowplayer ビデオプレーヤー-認証されていないストアドクロスサイトスクリプティングの脆弱性
999850		WEB-WORDPRESS WordPress プラグイン 2.9.16 より前の簡単なデジタルダウンロード-認証されていないストアドクロスサイトスクリプティングの脆弱性
999851		WEB-WORDPRESS WordPress プラグイン Crelly スライダーバージョン 1.3.5 より前-任意のファイルアップロードの脆弱性
999853	CVE-2019-2615	WEB-MISC Oracle WebLogic サーバ情報漏えいの脆弱性
999854	CVE-2019-11872	WordPress プラグイン Hustle 6.0.8.1 より前-CSV インジェクションの脆弱性
999855	CVE-2019-11231	WEB-MISC GetSimple CMS バージョン 3.3.15 以前-任意のファイルアップロードの脆弱性
999856	CVE-2019-11231	WEB-MISC GetSimple CMS バージョン 3.3.15 およびそれ以前-API キー情報の開示

署名ルール	**CVE ID**	説明
999857		5.2 より前の WEB-WORDPRESS WordPress プラグイン WP データベースバックアップ-コマンドインジェクションの脆弱性
999858		WEB-WORDPRESS WordPress プラグインスリックポップアップ 1.7.1 まで-権限昇格の脆弱性
999859	CVE-2019-12099	WEB-MISC PHP Fusion CMS バージョン 9.03.00 以前でリモートでコードが実行される脆弱性

署名更新バージョン **35**

January 25, 2022

バージョン 35 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 35 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999834	CVE-2019-13024	WEB-MISC Centreon バージョン 19.04 以前-コマンドインジェクションの脆弱性
999835	CVE-2019-5420	WEB-MISC Rails 開発モード-シークレットトークン漏洩の脆弱性
999836	CVE-2019-5418	WEB-MISC Rails アクションビュー-ファイルコンテンツ漏洩の脆弱性
999837	CVE-2018-12426, CVE-2019-11185	WEB-WORDPRESS WP ライブチャットサポート Pro プラグイン 8.0.26 より前-任意のファイルのアップロード
999838	CVE-2019-10270	WEB-WORDPRESS WordPress プラグインバージョン 2.0.40 より前のアルティメットメンバー-任意のパスワード
999839	CVE-2019-12826	5.10.2 より前の WEB-WORDPRESS WordPress プラグインウィジェットロジック-CSRF の脆弱性
999840		WEB-WORDPRESS WordPress プラグイン 2.5.39 より前のオールインワンイベントカレンダー-クロスサイトスクリプティングの脆弱性
999841	CVE-2019-11565	WEB-WORDPRESS WordPress プラグイン 1.6.7 より前のブログを印刷する-認証されていない SSRF の脆弱性
999842		WEB-WORDPRESS WordPress のプラグインバージョン 2.0.46 より前の究極のメンバー-複数 `cross-site scripting</LogString`

署名更新バージョン **36**

January 25, 2022

バージョン 36 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。シグニチャルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 36 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999817		バージョン 2.4.22 より前の WEB-WORDPRESS WordPress 広告インサータープラグイン-リモートコード実行
999818	CVE-2019-7839	WEB-MISC Adobe ColdFusion 複数バージョン-HTTP/SOAP ドットネットから Java へのリモートコード実行の脆弱性 (CVE-2019-7839)
999819	CVE-2019-7839	WEB-MISC Adobe ColdFusion 複数バージョン-HTTP/SOAP 経由でのリモートコード実行の脆弱性 Java-to-dotNet (CVE-2019-7839)

署名ルール	**CVE ID**	説明
999820	CVE-2019-11469	WEB-MISC 14150 ビルドより前の Zoho ManageEngine アプリケーションマネージャーで resourceid パラメーター経由で SQLi を許可する (CVE-2019-11469)
999821	CVE-2019-11448	WEB-MISC Zoho ManageEngine アプリケーションマネージャー 11.0 から 14.0-非認証 SQL インジェクション (CVE-2019-11448)
999822	CVE-2019-1003000	WEB-MISC Jenkins スクリプトセキュリティプラグイン 1.49 まで：サンドボックスバイパスの脆弱性（CVE-2019-1003000）
999823		WEB-WORDPRESS WordPress Cforms2 プラグイン 15.0.1 まで-認証されていない HTML インジェクションの脆弱性
999824	CVE-2019-0193	8.2 より前の WEB-MISC Apache Solr-DataConfig パラメーターを介した DIH リモートコード実行の脆弱性（CVE-2019-0193）
999825	CVE-2019-11580	WEB-MISC Atlassian Crowd Pdkinstall 開発プラグインが有効-認証されていない RCE (CVE-2019-11580)
999826	CVE-2019-0192	WEB-MISC Apache Solr 5.5.5/6.6.5 まで-設定 API リモートでコードが実行される脆弱性（CVE-2019-0192）
999827		WEB-WORDPRESS WooCommerce バリエーションスウォッチプラグイン 1.0.61 まで-反射型クロスサイトスクリプティングの脆弱性

署名ルール	**CVE ID**	説明
999828	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy プラグイン最大 2.61-ジョブ作成によるサンドボックスバイパスの脆弱性 (CVE-2019-1003001)
999829	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy プラグイン最大 2.61-サンドボックスバイパスの脆弱性 (CVE-2019-1003001)
999830		2.3.2 より前の WEB-WORDPRESS WordPress 太字ページビルダープラグイン-セキュリティバイパスの脆弱
999831	CVE-2019-15107	1.930 より前の WEB-MISC Webmin：認証されていないリモートコード実行の脆弱性 (CVE-2019-15107)
999832	CVE-2019-2767	ウェブその他 Oracle BI Publisher 11.1.1.9.0 および 12.2.1.4-XXE 脆弱性 (CVE-2019-2767)
999833	CVE-2019-15106	WEB-MISC Zoho ManageEngine OpManager から 12.4x まで-認証バイパスの脆弱性（CVE-2019-15106）
999948	CVE-2014-0114	Apache Struts 1 から 1.3.10 までは、HTTP_FORM_FIELD 経由で任意のコードを実行できるように ClassLoader の操作が可能です
999949	CVE-2013-4316	2.3.15.2 より前の Apache Struts 2 では、機密性、整合性、可用性に影響を与えることで動的メソッド呼び出しが可能になります

署名ルール	**CVE ID**	説明
999950	CVE-2013-4316	2.3.15.2 より前の Apache Struts 2 では、機密性、整合性、可用性に影響を与えることで動的メソッド呼び出しが可能になります

> 注:
>
> パフォーマンスの問題により、シグニチャルール 999947 が削除されました。

署名更新バージョン **37**

January 25, 2022

バージョン 37 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 37 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999806	CVE-2019-3394	WEB-MISC Atlassian Confluence またはデータセンター-ローカルファイル開示の脆弱性 (CVE-2019-3394)

署名ルール	**CVE ID**	説明
999807	CVE-2019-13569	WEB-WORDPRESS Icegram 4.1.8 より前のメール購読者とニュースレタープラグイン-ESFPX_lists パラメータ経由の SQLi (CVE-2019-13569)
999808	CVE-2019-13569	WEB-WORDPRESS Icegram メール購読者＆ニュースレタープラグイン 4.1.8 より前-SQLi Via Order Param (CVE-2019-13569)
999809	CVE-2019-2768	WEB-MISC Oracle BI Publisher -予測可能なセッショントークンの脆弱性（CVE-2019-2768）
999810	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy プラグイン最大 2.61-ジョブ更新によるサンドボックスバイパスの脆弱性 (CVE-2019-1003001)
999811	CVE-2019-13575	WEB-WORDPRESS wpEverest Everest Forms プラグイン 1.5.0 より前-SQL インジェクション (CVE-2019-13575)
999812	CVE-2019-15896	WEB-WORDPRESS LifterLMS プラグイン 3.34.5 まで-セキュリティバイパスの脆弱性 (CVE-2019-15896)
999813	CVE-2019-3396	WEB-MISC アトラシアン Confluence またはデータセンター-リモートでコードが実行される脆弱性（CVE-2019-3396）
999814	CVE-2019-5475	2.14.14 より前の WEB-MISC Sonatype Nexus リポジトリマネージャー-Createrepo パスを介したリモートコード実行 (CVE-2019-5475)

署名ルール	**CVE ID**	説明
999815	CVE-2019-5475	WEB-MISC 2.14.14 より前の Sonatype Nexus リポジトリマネージャー-Mergerepo パスを介したリモートコード実行 (CVE-2019-5475)
999816	CVE-2019-15104	WEB-MISC Zoho ManageEngine 12.4 より前のバージョンの OpManager-SQL インジェクションの脆弱性 (CVE-2019-15104)

署名更新バージョン **38**

January 25, 2022

バージョン 38 で特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 38 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999800	CVE-2019-12517	WEB-WORDPRESS SlickQuiz プラグインバージョン 1.3.7.1 以前-クロスサイトスクリプティングの脆弱性（CVE-2019-12517）
999801	CVE-2019-10392	WEB-MISC Jenkins Git クライアントプラグイン 2.8.4 以前-OS コマンドインジェクションの脆弱性（CVE-2019-10392）
999802	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Filedata フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999803	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Image フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999804	CVE-2019-12516	WEB-WORDPRESS SlickQuiz プラグインバージョン 1.3.7.1 以前-SQL インジェクションの脆弱性（CVE-2019-12516）
999805	CVE-2019-1262	WEB-MISC Microsoft SharePoint サーバー-クロスサイトスクリプティングの脆弱性 (CVE-2019-1262)

2019 年 **12** 月のシグネチャアップデート

January 25, 2022

2019-12-19 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 39 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999760		4.4.7 および 4.5.5 より前のバージョンの WEB-MISC FusionPBX-/app/exec/exec.php を介したリモートでのコード実行の脆弱性
999761	CVE-2019-12747	8.7.27 および 9.5.8 より前の WEB-MISC Typo3-信頼できないデータのデシリアライズ (CVE-2019-12747)
999762	CVE-2019-13608	WEB-MISC Citrix StoreFront サーバー-XML 外部エンティティインジェクションの脆弱性 (CVE-2019-13608)
999763		5.2.4 より前の WEB-WORDPRESS WordPress-フォームを介したプライベートまたはドラフト投稿/ページの脆弱性の認証されていない表示
999764		5.2.4 より前の WEB-WORDPRESS WordPress-URL を介したプライベートまたはドラフト投稿/ページの脆弱性の認証されていない表示

署名ルール	**CVE ID**	説明
999765	CVE-2019-15954	ウェブその他 Total.js CMS 12.0.0-JSON を介したウィジェット JavaScript コードインジェクションの脆弱性 (CVE-2019-15954)
999766	CVE-2019-15954	WEB-MISC Total.js CMS 12.0.0-フォームを介したウィジェット JavaScript コードインジェクションの脆弱性 (CVE-2019-15954)
999767		5.3.1 より前の WEB-WORDPRESS SyntaxHighlighter Evolved プラグイン-コメントによる保存型クロスサイトスクリプティング脆弱性
999768		5.3.1 より前の WEB-WORDPRESS SyntaxHighlighter 進化型プラグイン-POST 経由でのストアドクロスサイトスクリプティング
999769		5.3.1 より前の WEB-WORDPRESS SyntaxHighlighter 進化型プラグイン-JSON 経由で保存されたクロスサイトスクリプティング脆弱性
999770	CVE-2019-16120	4.10.7.2 より前の WEB-WORDPRESS イベントチケットプラグイン：CSV インジェクションの脆弱性 (CVE-2019-16120)
999771	CVE-2019-15029	4.4.8 より前の WEB-MISC FusionPBX：リモートでコードが実行される脆弱性 (CVE-2019-15029)

署名ルール	**CVE ID**	説明
999772		WEB-WORDPRESS Sassy ソーシャル共有プラグイン 3.3.4 より前-認証されていないクロスサイトスクリプティングの脆弱性
999773		WEB-WORDPRESS メール購読者およびニュースレタープラグインバージョン 4.3.1 以前-認証されていないブラインド SQLi の脆弱性
999774	CVE-2019-3398	WEB-MISC Atlassian Confluence またはデータセンター-すべての添付ファイルをダウンロードするパストラバーサルの脆弱性（CVE-2019-3398）
999775	CVE-2019-15952	WEB-MISC Total.js CMS 12.0.0: ページテンプレートパストラバーサルの脆弱性（CVE-2019-15952）
999776	CVE-2019-17236	WEB-WORDPRESS IgniteUp 近日公開およびメンテナンスモードプラグイン 3.4.0 まで-ストアドクロスサイトスクリプティング (CVE-2019-17236)
999777	CVE-2019-10475	WEB-MISC Jenkins ビルドメトリクスプラグイン 1.3-反射型クロスサイトスクリプティングの脆弱性（CVE-2019-10475）
999778	CVE-2019-17132	5.5.4 より前の WEB-MISC vBulletin パッチレベル 2-UpdateAvatar API エンドポイントのリモートコード実行の脆弱性（CVE-2019-17132）
999779	CVE-2019-14994	WEB-MISC Atlassian Jira サービスデスク-パストラバーサルの脆弱性 (CVE-2019-14994)

署名ルール	**CVE ID**	説明
999780	CVE-2019-19367	WEB-MISC FusionPBX 4.4.1 およびそれ以前-クロスサイトスクリプティングの脆弱性 (CVE-2019-19367)
999781	CVE-2019-18668	2.11.2 より前の WEB-WORDPRESS 通貨スイッチャープラグイン-POST による通貨設定バイパスの脆弱性 (CVE-2019-18668)
999782	CVE-2019-18668	2.11.2 より前の WEB-WORDPRESS 通貨スイッチャープラグイン-GET 経由での通貨設定バイパスの脆弱性 (CVE-2019-18668)
999783	CVE-2019-16663	WEB-MISC rConfig 3.9.2 以前-Search.crud.php 経由でリモートでコードが実行される脆弱性 (CVE-2019-16663)
999784		WEB-MISC Apache Solr 8.3.0 まで-VelocityResponseWriter カスタムテンプレートによる認証されていないリモートコードの実行
999785	CVE-2019-17235	WEB-WORDPRESS IgniteUp 近日公開とメンテナンスモードプラグイン 3.4.0 まで-Csv による情報開示 (CVE-2019-17235)
999786	CVE-2019-17235	WEB-WORDPRESS IgniteUp 近日公開とメンテナンスモードプラグイン 3.4.0 まで-BCC を介した情報開示 (CVE-2019-17235)
999787	CVE-2019-12276	WEB-MISC GrandNode 4.40-letsEncryptController パストラバーサルの脆弱性 (CVE-2019-12276)

署名ルール	**CVE ID**	説明
999788		バージョン 4.2.3 より前の WEB-WORDPRESS メール購読者とニュースレタープラグイン-認証されていない情報開示
999789	CVE-2019-4013	WEB-MISC IBM BigFix プラットフォーム 9.5-root 権限による認証済み任意のファイルアップロード (CVE-2019-4013)
999790	CVE-2019-11409	WEB-MISC FusionPBX バージョン 4.4.3 およびそれ以前-/app/basic_operator_panel/exec.php を介したリモートコードの実行 (CVE-2019-11409)
999791	CVE-2019-11409	WEB-MISC FusionPBX バージョン 4.4.3 およびそれ以前-/app/operator_panel/exec.php を介したリモートコードの実行 (CVE-2019-11409)
999792	CVE-2019-16662	WEB-MISC rConfig 3.9.2 およびそれ以前-AjaxServerSettingsChk.php を介した認証されていないリモートコードの実行 (CVE-2019-16662)
999793	CVE-2019-7609	5.6.15 および 6.6.1 より前の WEB-MISC Elastic Kibana：プロトタイプの汚染の脆弱性により認証されていない RCE が許可される（CVE-2019-7609）
999794	CVE-2019-10092	WEB-MISC Apache HTTP サーバー 2.4.39 まで-mod_proxy リミテッドクロスサイトスクリプティング (CVE-2019-10092)

署名ルール	**CVE ID**	説明
999795	CVE-2019-16520	WEB-WORDPRESS 3.2.7 より前のオールインワン SEO パックプラグイン-ストアドクロスサイトスクリプティングの脆弱性 (CVE-2019-16520)
999796	CVE-2019-17234	WEB-WORDPRESS IgniteUp 近日公開とメンテナンスモードプラグイン 3.4.0 まで-任意のファイルの削除 (CVE-2019-17234)
999797	CVE-2019-16525	バージョン 1.1.9 より前の WEB-WORDPRESS チェックリストプラグイン-クロスサイトスクリプティングの脆弱性 (CVE-2019-16525)
999798		1.9.6 より前の WEB-WORDPRESS Safe SVG プラグイン-クロスサイトスクリプティングの脆弱性
999799		バージョン 4.2.3 より前の WEB-WORDPRESS メール購読者とニュースレターのプラグイン-認証されていない任意のオプションの作成

署名更新バージョン **40**

January 25, 2022

2020-01-14 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名バージョン 40 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> シグニチャアップデートバージョン 40 には、不正なシグニチャルール 1861 に対する修正が含まれています。投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999732	CVE-2019-1620	WEB-MISC 11.2 (1) より前の Cisco データセンターネットワークマネージャ：任意のファイルアップロードの脆弱性 (CVE-2019-1620)
999733	CVE-2019-16702	WEB-MISC Integard Pro 2.2.0.9026：NoJS バッファオーバーフローの脆弱性 (CVE-2019-16702)
999734	CVE-2019-1621	WEB-MISC 11.2 (1) より前の Cisco データセンターネットワークマネージャ：任意のファイルダウンロードの脆弱性 (CVE-2019-1621)
999735	CVE-2019-8451	WEB-MISC 8.4.0 より前のアトラシアン Jira サーバー-サーバー側リクエストフォージェリの脆弱性 (CVE-2019-8451)
999736		4.0.3 より前の WEB-WORDPRESS GDPR クッキーコンプライアンスプラグイン-認証された任意の設定の削除の脆弱性

署名ルール	**CVE ID**	説明
999737	CVE-2019-11287	3.7.21 より前の WEB-MISC Pivotal RabbitMQ 3.7.x および 3.8.1 より前の 3.8.x-サービス拒否の脆弱性（CVE-2019-11287）
999738		WEB-WORDPRESS 1.20.1 より前の Elementor の究極のアドオン-Facebook ログインによる認証バイパスの脆弱性
999739		WEB-WORDPRESS 1.20.1 より前の Elementor の究極のアドオン-Google ログインによる認証バイパスの脆弱性
999740	CVE-2019-19366	4.4.10 より前の WEB-MISC FusionPBX：リダイレクトパラメータによる xml_cdr_search.php のクロスサイトスクリプティングの脆弱性（CVE-2019-19366）
999741	CVE-2019-16931	バージョン 3.3.1 より前の WEB-WORDPRESS ビジュアライザープラグイン-認証されていないクロスサイトスクリプティングの脆弱性（CVE-2019-16931）
999742	CVE-2019-16932	バージョン 3.3.1 より前の WEB-WORDPRESS ビジュアライザープラグイン-非認証 SSRF (CVE-2019-16932)
999743	CVE-2019-1619	WEB-MISC 11.1 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性（CVE-2019-1619）
999744	CVE-2019-12562	WEB-MISC 9.4.0 より前の DotnetNuke：ストアドクロスサイトスクリプティングの脆弱性（CVE-2019-12562）

署名ルール	**CVE ID**	説明
999745	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Filedata フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999746	CVE-2019-8371	5.0.2 より前の WEB-MISC OpenEMR-Form_Image フィールドを介したリモートコード実行の脆弱性（CVE-2019-8371）
999747		WEB-WORDPRESS Beaver Builder Ultimate Addons 1.24.1 より前-Facebook ログインによる認証バイパスの脆弱性
999748		WEB-WORDPRESS Beaver Builder Ultimate Addons 1.24.1 より前-Google ログインによる認証バイパスの脆弱性
999749	CVE-2019-19650	WEB-MISC Zoho ManageEngine AM ビルド前 13640-エージェントサーブレット経由の SQLi (CVE-2019-19650)
999750		WEB-MISC Zoho ManageEngine AM ビルド前 13620-opmRequestHandlerServlet サーブレットを介した API キーの開示
999751	CVE-2019-1622	WEB-MISC Cisco データセンターネットワークマネージャ 11.0 (1): 情報開示の脆弱性 (CVE-2019-1622)
999752	CVE-2019-16759	5.5.4 パッチレベル 1 より前の WEB-MISC vBulletin-リモートでコードが実行される脆弱性 (CVE-2019-16759)

署名ルール	**CVE ID**	説明
999753		2.7.8 より前の URL プラグインからの WEB-WORDPRESS の注目画像-REST API の脆弱性でアクセス制御が行われ
999754	CVE-2019-10098	ウェブその他 Apache HTTP サーバ 2.4.39 まで-mod_rewrite 自己参照リダイレクトの脆弱性（CVE-2019-10098）
999755	CVE-2019-1936	WEB-MISC Cisco UCS Director 6.0 ～6.6.1.0 および 6.7.0.0 ～ 6.7.1.0：コマンドインジェクションの脆弱性（CVE-2019-1936）
999756	CVE-2019-19649	WEB-MISC ビルド前の Zoho ManageEngine AM 13620-EventID パラメーターによる認証されていない SQLi (CVE-2019-19649)
999757	CVE-2019-19649	WEB-MISC ビルド 13620 より前の Zoho ManageEngine AM-エンティティパラメータによる認証されていない SQLi（CVE-2019-19649）
999758	CVE-2019-15036	WEB-MISC JetBrains 2019.1 より前のチームシティ： OS コマンドインジェクションの脆弱性（CVE-2019-15036）
999759	CVE-2019-17239	WEB-WORDPRESS ダッシュボードプラグインからプラグインとテーマをダウンロードする最大 1.5-ストアドクロスサイトスクリプティングの脆弱性（CVE-2019-17239）

署名更新バージョン **41**

January 25, 2022

2020-02-04 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。シグニチャアップデートには、シグニチャ ID、シグニチャバージョン、およびアドレス指定された CVE のリストが含まれます。

署名バージョン

署名バージョン 41 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> シグニチャアップデートバージョン 41 には、不正なシグニチャルール 1861 に対する修正が含まれています。投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999717		WEB-WORDPRESS WordPress バージョン 5.3.x 以前-xmlrpc.php pingback.ping メソッドによるサービス拒否の脆弱性
999718		1.21.16 より前の WP Time Capsule プラグインによる WEB-WORDPRESS バックアップとステージング-認証バイパスの脆弱性
999719	CVE-2019-19731	WEB-MISC .NET 1.4.5 用ロキシーファイルマン：RENAMEFILE によるパストラバーサルの脆弱性 (CVE-2019-19731)

署名ルール	**CVE ID**	説明
999720	CVE-2019-19915	WEB-WORDPRESS 301 リダイレクト — 2.4.0 までの簡単リダイレクトマネージャープラグイン-複数の脆弱性 (CVE-2019-19915)
999721	CVE-2019-17662	バージョン 1.0b1 より前の WEB-MISC Cybele ソフトウェア ThinVNC：ディレクトリトラバーサルの脆弱性（CVE-2019-17662）
999722	CVE-2020-6168	WEB-WORDPRESS 最小 2.17 より前のメンテナンスモードプラグイン-メンテナンス設定の脆弱性（CVE-2020-6168）
999723	CVE-2020-6166	WEB-WORDPRESS 最小 2.17 より前のメンテナンスモードプラグイン-テーマ変更の脆弱性 (CVE-2020-6166)
999724	CVE-2020-6166	WEB-WORDPRESS 最小 2.17 より前のメンテナンスモードプラグイン-エクスポート設定の脆弱性（CVE-2020-6166）
999725		1.9.4.5 より前の WEB-WORDPRESS iniFiniteWP クライアントプラグイン-認証バイパスの脆弱性
999726	CVE-2019-16773	5.3.1 より前のバージョンの WEB-WORDPRESS WordPress-JSON オブジェクトを使用した REST API を介したクロスサイトスクリプティングの脆弱性（CVE-2019-16773）

署名ルール	**CVE ID**	説明
999727	CVE-2019-16773	5.3.1 より前のバージョンの WEB-WORDPRESS WordPress-フォームフィールドを使用した REST API を介したクロスサイトスクリプティングの脆弱性（CVE-2019-16773）
999728	CVE-2019-16773	5.3.1 より前の WEB-WORDPRESS WordPress のバージョン：user-edit.php を介したクロスサイトスクリプティングの脆弱性（CVE-2019-16773）
999729	CVE-2019-16773	5.3.1 より前の WEB-WORDPRESS WordPress のバージョン：profile.php を介したクロスサイトスクリプティングの脆弱性（CVE-2019-16773）
999730	CVE-2019-16113	WEB-MISC Bludit 3.9.2：uuid を介したリモートでのイメージアップロードによるコードの実行の脆弱性（CVE-2019-16113）
999731	CVE-2019-16113	WEB-MISC Bludit 3.9.2：ファイル名を介したイメージアップロードのリモートコード実行の脆弱性（CVE-2019-16113）

2020 年 **2** 月の署名更新

January 25, 2022

2020-02-11 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 42 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999707		バージョン 1.4.8 より前の WEB-WORDPRESS WPCentral プラグイン-権限昇格の脆弱性
999708	CVE-2019-15979	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：コマンドインジェクションの脆弱性 (CVE-2019-15979)
999709	CVE-2019-15978	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：コマンドインジェクションの脆弱性 (CVE-2019-15978)
999710	CVE-2019-15975	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性（CVE-2019-15975）
999711	CVE-2019-15976	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性（CVE-2019-15976）

署名ルール	**CVE ID**	説明
999712	CVE-2019-16405	バージョン 19.10.2 より前の WEB-MISC Centreon：リモートでコードが実行される脆弱性（CVE-2019-16405）
999713	CVE-2020-7048	WEB-WORDPRESS WP データベースリセットプラグイン 3.1 まで-認証されていないデータベーステーブルのリセットの脆弱性（CVE-2020-7048）
999714	CVE-2020-7108	バージョン 3.1.2 より前の WEB-WORDPRESS LearnDash プラグイン-反射型クロスサイトスクリプティングの脆弱性（CVE-2020-7108）
999715	CVE-2019-15977	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：認証バイパスの脆弱性（CVE-2019-15977）
999716	CVE-2020-2096	WEB-MISC Jenkins Gitlab Hook プラグインバージョン 1.4.2 以前-クロスサイトスクリプティングの脆弱性（CVE-2020-2096）

2020 年 **2** 月の署名更新

January 25, 2022

2020-02-27 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 43 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999696	CVE-2019-15983	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ：CablePlan 経由の XML 外部エンティティの脆弱性（CVE-2019-15983）
999697	CVE-2019-20197	WEB-MISC Nagios XI 5.6.9：認証された任意のコマンド実行の脆弱性（CVE-2019-20197）
999698	CVE-2020-8417	2.14.0 より前の WEB-WORDPRESS コードスニペットプラグイン-CSRF の脆弱性（CVE-2020-8417）
999699		バージョン 1.4.8 より前の WEB-WORDPRESS WPCentral プラグイン-権限昇格の脆弱性
999700	CVE-2020-8596	1.9.5.6 より前の WEB-WORDPRESS 参加者データベースプラグイン-認証された SQL インジェクションの脆弱性（CVE-2020-8596）
999701	CVE-2020-8426	2.8.5 より前の WEB-WORDPRESS Elementor ページビルダープラグイン-認証済みリフレクションクロスサイトスクリプティングの脆弱性（CVE-2020-8426）

署名ルール	**CVE ID**	説明
999702	CVE-2019-19509	ウェブその他 rConfig 3.9.3: ajaxArchiveFiles.php 経由でリモートでコードが実行される脆弱性（CVE-2019-19509）
999703	CVE-2019-8449	WEB-MISC 8.4.0 より前の Atlassian Jira サーバー-情報漏えいの脆弱性 (CVE-2019-8449)
999704	CVE-2019-9194	2.1.48 より前の WEB-MISC elFinder-PHP コネクタコマンドインジェクションの脆弱性（CVE-2019-9194）
999705	CVE-2019-15985	WEB-MISC 11.3 (1) より前の Cisco データセンターネットワークマネージャ-GetvmHostData を介した SQL インジェクションの脆弱性（CVE-2019-15985）
999706	CVE-2020-8549	2.40.1 より前の WEB-WORDPRESS 強力なお客様の声プラグイン-ストアドクロスサイトスクリプティングの脆弱性（CVE-2020-8549）

2020 年 **4** 月の署名更新

January 25, 2022

2020-04-27 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 44 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999683	CVE-2020-9043	1.5.1 より前の WEB-WORDPRESS WPCentral プラグイン-接続キー漏洩の脆弱性 (CVE-2020-9043)
999684		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング
999685		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング
999686	CVE-2020-0618	WEB-MISC Microsoft SQL Server Reporting Services-リモートでコードが実行される脆弱性 (CVE-2020-0618)
999687	CVE-2019-16278	WEB-MISC 1.3.7 より前のノストロモ Nhttpd-strcutl 関数が認証されていないリモートコード実行を許可する (CVE-2019-16278)
999688	CVE-2019-1937	WEB-MISC Cisco UCS Director 6.6.0.0 ～6.6.1.0 および 6.7.0.0 ～6.7.1.0：認証バイパスの脆弱性 (CVE-2019-1937)
999689		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング

署名ルール	**CVE ID**	説明
999690	CVE-2020-9006	3.0 より前の WEB-WORDPRESS ポップアップビルダープラグイン-PHP デシリアライゼーションによる SQL インジェクションの脆弱性（CVE-2020-9006）
999691		WEB-WORDPRESS Duplicate-Post プラグインバージョン 3.2.3 以前-永続的なクロスサイトスクリプティング
999692		WEB-MISC はコンテンツ長と転送エンコーディングヘッダーによるリクエストの密輸を防止
999693		1.6.3 より前の WEB-WORDPRESS ThemeGrill デモインポータープラグイン-認証バイパスとデータベースワイプの脆弱性
999694	CVE-2019-17237	WEB-WORDPRESS IgniteUp 近日公開およびメンテナンスモードプラグイン 3.4.1 より前-メッセージを介した CSRF の脆弱性（CVE-2019-17237）
999695	CVE-2019-17237	WEB-WORDPRESS IgniteUp 近日公開、3.4.1 より前のメンテナンスモードプラグイン-サブジェクト経由の CSRF の脆弱性 (CVE-2019-17237)

2020 年 **5** 月のシグネチャアップデート

January 25, 2022

2020-05-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 45 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。最新の Snort リリースによると、ID が 1258、1306、2520、2661、5695、10996、11817、12056、15471、17049、21634 のシグニチャルールは削除されました。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999666		1.3.28 より前の WEB-WORDPRESS デュプリケータプラグイン-認証されていない任意のファイルダウンロードの脆弱性
999667	CVE-2020-10220	WEB-MISC rConfig から 3.94 まで-SQL インジェクションの脆弱性（CVE-2020-10220）
999668	CVE-2020-5844	WEB-MISC Artica Pandora FMS 7.0-/attachment/files_repo/ を介した危険なタイプの任意のファイルの実行 (CVE-2020-5844)
999669	CVE-2020-8813	1.2.10 より前の WEB-MISC Cacti-graph_realtime.php 経由でリモートでコードが実行される脆弱性（CVE-2020-8813）
999670	CVE-2020-8654	WEB-MISC EyesofNetwork 5.3: リモートでコードが実行される脆弱性（CVE-2020-8654）

署名ルール	**CVE ID**	説明
999671	CVE-2020-10196	WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン 3.64.1 より前：認証されていないクロスサイトスクリプティングの脆弱性（CVE-2020-10196）
999672	CVE-2019-15949	5.6.6 より前の WEB-MISC Nagios XI：ルートとしてリモートでコードが実行される脆弱性（CVE-2019-15949）
999673	CVE-2020-10879	WEB-MISC rConfig 3.9.5 以前-search.crud.php 経由でリモートでコードが実行される脆弱性（CVE-2020-10879）
999674	CVE-2020-8656	WEB-MISC EyesOfNetwork 5.3 -EyesOfNetwork API 2.4.2 SQL インジェクションの脆弱性 (CVE-2020-8656)
999675	CVE-2020-10195	3.64.1 より前の WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン-認証済みシステム情報開示 (CVE-2020-10195)
999676	CVE-2020-10195	3.64.1 より前の WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン-認証済みサブスクライバー情報開示 (CVE-2020-10195)
999677	CVE-2020-10195	3.64.1 より前の WEB-WORDPRESS Sygnoos ポップアップビルダープラグイン-認証済み設定の変更 (CVE-2020-10195)

署名ルール	**CVE ID**	説明
999678	CVE-2020-0646	Microsoft SharePoint Server-.NET フレームワークワークフローで SOAP 1.2 経由でリモートでコードが実行される脆弱性 (CVE-2020-0646)
999679	CVE-2020-0646	Microsoft SharePoint Server-.NET フレームワークワークフローで SOAP 1.1 経由でリモートでコードが実行される脆弱性 (CVE-2020-0646)
999680	CVE-2020-10221	WEB-MISC rConfig から 3.94 まで-リモートでコードが実行される脆弱性（CVE-2020-10221）
999681	CVE-2019-19134	WEB-WORDPRESS ヒーローマッププレミアム 2.2.3 より前-認証されていないリフレクションクロスサイトスクリプティングの脆弱性（CVE-2019-19134）
999682	CVE-2020-10385	1.5.9 より前の WEB-WORDPRESS WPForms プラグイン-ストアドクロスサイトスクリプティングの脆弱性（CVE-2020-10385）

2020 年 **6** 月のシグネチャアップデート

January 25, 2022

2020-06-03 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 46 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999643		WEB-WORDPRESS 1010.0.64 より前の Google マッププラグイン用ウェブマップビルダー-gmwd_setup ページを介した認証されていないクロスサイトスクリプティングの脆弱性
999644		WEB-WORDPRESS 10Google マッププラグイン 10.0.64 およびそれ以前のウェブマップビルダー-options_gmwd ページを介したクロスサイトスクリプティングの脆弱性
999645	CVE-2020-5187	WEB-MISC DNN 9.4.4 まで: URL 経由のパストラバーサルの脆弱性（CVE-2020-5187）
999646	CVE-2020-5187	WEB-MISC DNN 9.4.4 まで: ローカル経由のパストラバーサルの脆弱性（CVE-2020-5187）
999647	CVE-2020-9335	1.5.46 より前の WEB-WORDPRESS フォトギャラリープラグイン-image_alt_text_ フィールドを介したクロスサイトスクリプティングの脆弱性 (CVE-2020-9335)

署名ルール	**CVE ID**	説明
999648	CVE-2020-9335	1.5.46 より前の WEB-WORDPRESS フォトギャラリープラグイン-名前フィールドを介したクロスサイトスクリプティングの脆弱性（CVE-2020-9335）
999649	CVE-2020-9335	1.5.46 より前の WEB-WORDPRESS フォトギャラリープラグイン-説明フィールドを介したクロスサイトスクリプティングの脆弱性（CVE-2020-9335）
999650	CVE-2020-10189	WEB-MISC 10.0.479 より前の Zoho ManageEngine デスクトップセントラル-認証されていないリモートコード実行 Vuln (CVE-2020-10189)
999651	CVE-2020-10189	WEB-MISC 10.0.479 より前の Zoho ManageEngine デスクトップセントラル-認証されていない任意のファイルアップロードの Vuln (CVE-2020-10189)
999652		WEB-WORDPRESS 2.3.2 より前の WooCommerce プラグインの柔軟なチェックアウトフィールド-認証されていない設定変更 Vuln
999653	CVE-2020-0688	WEB-MISC Microsoft Exchange Server-検証キーのリモートコード実行の脆弱性 (CVE-2020-0688)
999654	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0：ip_src パラメータを介したリモートコード実行の脆弱性 (CVE-2020-8947、CVE-2019-20224)

署名ルール	**CVE ID**	説明
999655	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0-dst_port パラメータを介したリモートコード実行の脆弱性（CVE-2020-8947、CVE-2019-20224）
999656	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0：src_port パラメータを介したリモートコード実行の脆弱性（CVE-2020-8947、CVE-2019-20224）
999657	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0：ip_dst パラメータによるリモートコード実行の脆弱性（CVE-2020-8947、CVE-2019-20224）
999658	CVE-2020-5186	WEB-MISC DNN 9.5.0 まで：ジャーナル XML アップロードによるクロスサイトスクリプティングの脆弱性（CVE-2020-5186）
999659		WEB-WORDPRESS WP サイトマップページプラグイン 1.6.2 以前-クロスサイトスクリプティング wsp_exclude_pages 経由の脆弱性
999660	CVE-2020-5188	WEB-MISC DNN 9.5.0 まで-UploadFromURL を介したセキュアでない権限の脆弱性（CVE-2020-5188）
999661	CVE-2020-5188	WEB-MISC DNN 9.5.0 まで-UploadFromLocal を介したセキュアでない権限の脆弱性（CVE-2020-5188）
999662	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：API テーマを介したリモートコード実行の脆弱性（CVE-2020-7799）

署名ルール	**CVE ID**	説明
999663	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：API 電子メールテンプレートを介したリモートコード実行の脆弱性（CVE-2020-7799）
999664	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：GUI テーマを介したリモートコード実行の脆弱性（CVE-2020-7799）
999665	CVE-2020-7799	1.11.0 より前の WEB-MISC FusionAuth：GUI 電子メールテンプレートを介したリモートコード実行の脆弱性（CVE-2020-7799）

2020 年 **6** 月のシグネチャアップデート

January 25, 2022

2020-06-12 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 47 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999580	CVE-2020-6010	3.2.6.9 より前の WEB-WORDPRESS LearnPress LMS プラグイン-SQL インジェクションの脆弱性 (CVE-2020-6010)
999581		WEB-MISC Nagios XI 5.6.13 まで-サービスコマンド _Test 任意のコマンド実行の脆弱性
999582	CVE-2020-0932	Microsoft SharePoint サーバー-SOAP 1.2 を介した WebPart ソースマークアップのリモートコード実行の脆弱性 (CVE-2020-0932)
999583	CVE-2020-0932	Microsoft SharePoint サーバー-Web パーツソースマークアップ SOAP 1.1 経由でリモートでコードが実行される脆弱性 (CVE-2020-0932)
999584	CVE-2020-12642	3.4.24.2 より前の WEB-WORDPRESS Ninja Forms プラグイン-インポートフィールドを介したクロスサイトリクエストフォージェリの脆弱性 (CVE-2020-12642)
999585	CVE-2020-12642	WEB-WORDPRESS 3.4.24.2 より前の Ninja Forms プラグイン-インポートフォームを介したクロスサイトリクエストフォージェリの脆弱性（CVE-2020-12642）
999586	CVE-2020-11450	WEB-MISC マイクロストラテジー Web 10.4-情報開示の脆弱性 (CVE-2020-11450)

署名ルール	**CVE ID**	説明
999587	CVE-2020-7935	WEB-MISC Artica Pandora FMS 7.0-危険なタイプの脆弱性を持つファイルを無制限にアップロードすると RCE が許可される (CVE-2020-7935)
999588	CVE-2020-12116	WEB-MISC ビルド 125125 より前の Zoho ManageEngine OpManager-情報漏えいの脆弱性 (CVE-2020-12116)
999589		2.9.6 より前の WEB-WORDPRESS エレメントまたはページビルダー-権限昇格の脆弱性
999590	CVE-2020-11738	WEB-WORDPRESS-1.3.28 より前のスナップクリークデュプリケータープラグイン-パストラバーサルの脆弱性（CVE-2020-11738）
999591	CVE-2020-10389	WEB-MISC Chadha PHPKB 標準多言語 9-リモートでコードが実行される脆弱性 (CVE-2020-10389)
999592	CVE-2020-11516	WEB-WORDPRESS お問い合わせフォーム 7 Datepicker プラグイン 2.6.0 まで-ストアドクロスサイトスクリプティングの脆弱性 (CVE-2020-11516)
999593		WEB-MISC Nagios XI 5.6.13 まで-ステップ経由でのエクスポート RRD 任意のコマンド実行の脆弱性
999594		WEB-MISC Nagios XI 5.6.13 まで-エンド経由でのエクスポート RRD 任意のコマンド実行の脆弱性
999595		WEB-MISC Nagios XI 5.6.13 まで-エクスポート RRD 任意のコマンドが起動時に実行される脆弱性

署名ルール	**CVE ID**	説明
999596	CVE-2019-19799	Zoho ManageEngine アプリケーションマネージャー 14600 より前-情報漏えいの脆弱性 (CVE-2019-19799)
999597	CVE-2020-10458	WEB-MISC Chadha PHPKB 標準多言語 9-任意のフォルダを削除する脆弱性（CVE-2020-10458）
999598	CVE-2017-9822	9.1.1 より前の WEB-MISC DNN: DNNPersonalization クッキーを介したリモートコード実行の脆弱性（CVE-2017-9822）
999599	CVE-2020-7953	WEB-MISC opServices opMon 9.3.2：nmap_options パラメータを介した認証されていない情報漏洩の脆弱性（CVE-2020-7953）
999600	CVE-2020-7953	WEB-MISC OpServices opMon 9.3.2：ホストパラメータを介した認証されていない情報漏洩の脆弱性（CVE-2020-7953）
999601		WEB-MISC Bolt CMS 3.7.0-newname パラメータによる危険なタイプの脆弱性へのファイル名変更
999602		WEB-MISC Bolt CMS 3.7.0-newname パラメータによるパストラバーサルの脆弱性
999603		WEB-MISC Bolt CMS 3.7.0-oldname パラメータによるパストラバーサルの脆弱性
999604		WEB-MISC Bolt CMS 3.7.0-親パラメータによるパストラバーサルの脆弱性
999605		WEB-MISC Bolt CMS 3.7.0-表示名パラメータの不適切なフィールド検証の脆弱性

署名ルール	**CVE ID**	説明
999606	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-ビューログに不正な認証の脆弱性 (CVE-2020-9004)
999607	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-メディアキャッシュ設定における不正な認証の脆弱性（CVE-2020-9004）
999608	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-アプリケーション設定での不正な認証の脆弱性 (CVE-2020-9004)
999609	CVE-2020-9004	WEB-MISC-Wowza ストリーミングエンジン 4.7.8-サーバー設定での不正な認証の脆弱性 (CVE-2020-9004)
999610		WEB-MISC PrestaShop 1.7.6.5-ファイルマネージャを介した CSRF の脆弱性
999611	CVE-2020-10238	WEB-MISC Joomla! 3.9.16 より前：com_templates を介したセキュリティバイパスの脆弱性 (CVE-2020-10238)
999612	CVE-2020-11510	3.2.6.9 より前の WEB-WORDPRESS LearnPress LMS プラグイン-learnpress_create_page を介した権限昇格 (CVE-2020-11510)
999613	CVE-2020-11510	WEB-WORDPRESS LearnPress LMS プラグイン 3.2.6.9 より前-learnpress_update_order_status による権限昇格 (CVE-2020-11510)

署名ルール	**CVE ID**	説明
999614	CVE-2020-8636	WEB-MISC OpServices opMon 9.3.2-nmap_options パラメーターによる認証されていないリモートコード実行の脆弱性 (CVE-2020-8636)
999615	CVE-2020-8636	WEB-MISC OpServices opMon 9.3.2：ホストパラメータによる認証されていないリモートコード実行の脆弱性（CVE-2020-8636）
999616	CVE-2020-11511	3.2.6.9 より前の WEB-WORDPRESS LearnPress LMS プラグイン-教師になる受け入れによる権限昇格 (CVE-2020-11511)
999617	CVE-2020-11451	WEB-MISC Microstrategy Web-JSP 経由でのセキュアでないファイルタイプのアップロードの脆弱性（CVE-2020-11451）
999618	CVE-2020-11451	WEB-MISC Microstrategy Web-ASP 経由でのセキュアでないファイルタイプのアップロードの脆弱性（CVE-2020-11451）
999619	CVE-2020-11515	1.0.41 より前の WEB-WORDPRESS WP SEO プラグインのランク数学-URL を介した REST API を介したリダイレクトの脆弱性（CVE-2020-11515）
999620	CVE-2020-11515	1.0.41 より前の WEB-WORDPRESS WP SEO プラグインのランク数学-REST API rest_route パラメータを介したリダイレクトの脆弱性 (CVE-2020-11515)

署名ルール	**CVE ID**	説明
999621	CVE-2020-10457	WEB-MISC Chadha PHPKB 標準多言語 9-imgName を介した任意のファイル名変更の脆弱性 (CVE-2020-10457)
999622	CVE-2020-10457	WEB-MISC Chadha PHPKB 標準多言語 9-imgURL を介した任意のファイル名変更の脆弱性 (CVE-2020-10457)
999623	CVE-2019-1821	WEB-MISC Cisco プライムインフラストラクチャ：リモートでコードが実行される脆弱性 (CVE-2019-1821)
999624		2.10.16 より前の WEB-WORDPRESS ページビルダープラグイン-Ajax action_builder_content を介した CSRF の脆弱性
999625		2.10.16 より前の WEB-WORDPRESS ページビルダープラグイン-ライブエディターによる CSRF の脆弱性
999626	CVE-2020-11514	WEB-WORDPRESS WP SEO プラグイン Rank Math 1.0.41 より前-URL 経由の REST API による権限昇格 (CVE-2020-11514)
999627	CVE-2020-11514	WEB-WORDPRESS WP SEO プラグイン Rank Math 1.0.41 より前-REST API を介した権限昇格 rest_route パラメーター (CVE-2020-11514)
999628	CVE-2019-6713	5.0.190312 より前の WEB-MISC ThinkCMF-/route/editpost.html を介したコードインジェクションの脆弱性 (CVE-2019-6713)

署名ルール	**CVE ID**	説明
999629	CVE-2019-6713	5.0.190312 より前の WEB-MISC ThinkCMF-/route/addpost.html を介したコードインジェクションの脆弱性 (CVE-2019-6713)
999630		1.8.0 より前の WEB-WORDPRESS Google サイトキットプラグイン-保護されていない検証の脆弱性
999631	CVE-2020-9315	WEB-MISC Oracle iPlanet ウェブサーバー 7.0.x-不正なアクセス制御の脆弱性（CVE-2020-9315）
999632	CVE-2020-1947	WEB-MISC Apache ShardingSphere 4.0.0-RC3 および 4.0.0-SnakeYAML リモートでコードが実行される脆弱性 (CVE-2020-1947)
999633	CVE-2020-7961	7.2.1 CE GA2 より前の Liferay ポータル-JSON-RPC を介した JSONWS デシリアライゼーション RCE の脆弱性 (CVE-2020-7961)
999634	CVE-2020-7961	7.2.1 CE GA2 より前の Liferay ポータル-URL パス経由の JSONWS デシリアライゼーション RCE の脆弱性（CVE-2020-7961）
999635	CVE-2020-7961	7.2.1 CE GA2 より前の Liferay ポータル-フォームおよび URI クエリによる JSONWS デシリアライゼーション RCE の脆弱性 (CVE-2020-7961)
999636	CVE-2020-8518	WEB-MISC Horde グループウェアウェブメール版 5.2.22：リモートでコードが実行される脆弱性 (CVE-2020-8518)

署名ルール	**CVE ID**	説明
999637	CVE-2020-7351	WEB-MISC Fonality Trixbox CE 2.8.0.4 およびそれ以前-リモートでコードが実行される脆弱性 (CVE-2020-7351)
999638	CVE-2020-12720	5.6.1 より前の WEB-MISC vBulletin パッチレベル 1-認証されていない SQL インジェクションの脆弱性（CVE-2020-12720）
999639	CVE-2019-19800	Zoho ManageEngine アプリケーションマネージャー 14520 より前：パストラバーサルの脆弱性 (CVE-2019-19800)
999640	CVE-2020-10386	WEB-MISC Chadha PHPKB 標準多言語 9-リモートコード実行 (CVE-2020-10386)
999641	CVE-2020-8497	WEB-MISC Artica Pandora FMS 7.0：認証されていない情報漏洩の脆弱性（CVE-2020-8497）
999642	CVE-2020-6009	WEB-WORDPRESS LearnDash LMS プラグイン 3.1.6 より前-認証されていない SQL インジェクションの脆弱性 (CVE-2020-6009)

2020 年 **7** 月の署名更新

January 25, 2022

2020-07-01 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 48 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999563		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-クロスサイトスクリプティング pagelayer_cf_to_email 経由の脆弱性
999564		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-ページレイヤ電話を介したクロスサイトスクリプティングの脆弱性
999565		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-ページレイヤアドレスを介したクロスサイトスクリプティングの脆弱性
999566	CVE-2020-1961	WEB-MISC Apache Syncope -サーバー側テンプレートインジェクションの脆弱性 (CVE-2020-1961)
999567	CVE-2019-18935	ASP.NET AJAX の WEB-MISC 進捗テレリック UI-RadAsyncUpload .NET デシリアライゼーションの脆弱性 (CVE-2019-18935)
999568	CVE-2020-9463	WEB-MISC Centreon 19.10：OS コマンドインジェクションの脆弱性（CVE-2020-9463）

署名ルール	**CVE ID**	説明
999569		3.7.6 より前の WEB-WORDPRESS サポートレビュープラグイン-認証されていないストアドクロスサイトスクリプティングの脆弱性
999570		WEB-WORDPRESS ページビルダー PageLayer プラグイン 1.1.2 より前-pagelayer_save_template 経由での不適切なアクセス制御 Vuln
999571		WEB-WORDPRESS ペスページビルダー PageLayer プラグイン 1.1.2 より前-pagelayer_update_site_title 経由での不適切なアクセス制御 Vuln
999572		WEB-WORDPRESS ペスページビルダー PageLayer プラグイン 1.1.2 より前-pagelayer_save_content 経由での不適切なアクセス制御 Vuln
999573		1.3.3.3 より前のお問い合わせフォーム 7 の WEB-WORDPRESS ドラッグアンドドロップアップロード-任意のファイル拡張子のアップロードの脆弱性
999574	CVE-2020-9314	WEB-MISC Oracle iPlanet ウェブサーバー 7.0.x-イメージインジェクションの脆弱性 (CVE-2020-9314)
999575	CVE-2020-9484	WEB-MISC Apache Tomcat 複数バージョン-信頼できないデータのデシリアライズ (CVE-2020-9484)

署名ルール	**CVE ID**	説明
999576	CVE-2020-13252	19.04.15 より前の WEB-MISC セントレオン：リモートでコードが実行される脆弱性 (CVE-2020-13252)
999577	CVE-2020-11453	WEB-MISC マイクロストラテジーウェブ-SOAP 経由の CSRF 脆弱性 (CVE-2020-11453)
999578	CVE-2020-11453	WEB-MISC マイクロストラテジーウェブ-CSRF 脆弱性 (CVE-2020-11453)
999579	CVE-2020-7237	1.2.8 より前の WEB-MISC Cacti-リモートでコードが実行される脆弱性（CVE-2020-7237）

2020 年 **8** 月のシグネチャアップデート

January 25, 2022

2020-08-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 49 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999556	CVE-2020-13241	WEB-MISC Microweber 1.1.18-危険なタイプの脆弱性を持つファイルの無制限アップロード (CVE-2020-13241)
999557	CVE-2020-3250	WEB-MISC Cisco UCS Director-UserapiDownloadFile を介した REST API パストラバーサルの脆弱性（CVE-2020-3250）
999558		WEB-WORDPRESS PageBuilder KingComposer プラグイン 2.9.4 より前-アクションによるディレクトリの任意の削除 = 一括削除
999559		WEB-WORDPRESS PageBuilder KingComposer プラグイン 2.9.4 より前-アクション = アップロードによるリモートコード実行の脆弱性
999560	CVE-2018-1999024	WEB-MISC Moodle-MathJax Unicode クロスサイトスクリプティングの脆弱性 (CVE-2018-1999024)
999561	CVE-2020-13693	2.6.5 より前の WEB-WORDPRESS bbPress プラグイン：認証されていない権限昇格の脆弱性（CVE-2020-13693）
999562	CVE-2020-12847	2.0.7 より前の WEB-MISCPydio セル-リモートコード実行の脆弱性 (CVE-2020-12847)

2020 年 **9** 月の署名更新

January 25, 2022

2020-09-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 50 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999532	CVE-2020-1956	WEB-MISC Apache Kylin-キューブ dest-config 経由でのリモートコード実行の移行 (CVE-2020-1956)
999533	CVE-2020-1956	WEB-MISC Apache Kylin-キューブは src-config 経由でリモートコード実行を移行します (CVE-2020-1956)
999534	CVE-2020-1956	WEB-MISC Apache Kylin-キューブは ProjectName 経由でリモートコード実行を移行します (CVE-2020-1956)
999535	CVE-2020-3247	WEB-MISC Cisco UCS Director: CopyFileRunnable 任意のシンボリックリンク作成の脆弱性（CVE-2020-3247）
999536	CVE-2019-16872	1.22.1 より前の WEB-MISC Portainer-アップデートスタックを介した不正なアクセス制御の脆弱性（CVE-2019-16872）

署名ルール	**CVE ID**	説明
999537	CVE-2019-16872	1.22.1 より前の WEB-MISC Portainer-スタックの作成による不正なアクセス制御の脆弱性 (CVE-2019-16872)
999538	CVE-2020-13855	WEB-MISC Artica Pandora FMS 7.44: ファイルリポジトリマネージャを介した任意のファイルアップロードの脆弱性 (CVE-2020-13855)
999539	CVE-2020-5902	WEB-MISC F5 BIG-IP-/hsqldb 経由でのトラフィック管理ユーザーインターフェイス RCE の脆弱性 (CVE-2020-5902)
999540	CVE-2020-5902	WEB-MISC F5 BIG-IP-/tmui 経由のトラフィック管理ユーザーインターフェイス RCE の脆弱性 (CVE-2020-5902)
999541		WEB-MISC WebERP 4.15.1 以前-認証されていない情報漏えいの脆弱性
999542	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-timeline.php およびタイムスタンプパラメータによる認証されていない RCE の脆弱性 (CVE-2020-7209)
999543	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kivis.php および ts パラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)
999544	CVE-2020-7209	6.0-2 より前の WEB-MISC HP LinuxKI-kivis.php およびエンドパラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)

署名ルール	**CVE ID**	説明
999545	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kivis.php および起動パラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)
999546	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kivis.php および pid パラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)
999547	CVE-2020-7209	6.0-2 より前の WEB-MISC HP LinuxKI-kidsk_trace_view.php およびエンドパラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)
999548	CVE-2020-7209	WEB-MISC 6.0-2 より前の HP LinuxKI-kidsk_trace_view.php および起動パラメータを介した認証されていない RCE の脆弱性 (CVE-2020-7209)
999549		WEB-MISC 9.03.70 より前の PHP フュージョン-PHP オブジェクトインジェクションの脆弱性
999550	CVE-2020-1181	WEB-MISC Microsoft SharePoint サーバー-Web パーツを介したリモートコード実行 (CVE-2020-1181)
999551	CVE-2020-10547	3.9.5 より前の WEB-MISC rConfig-SearchColumn を介したポリシー要素における認証されていない SQLi の脆弱性 (CVE-2020-10547)

署名ルール	**CVE ID**	説明
999552	CVE-2020-10547	3.9.5 より前の WEB-MISC rConfig: SearchField を介したポリシー要素における認証されていない SQLi の脆弱性 (CVE-2020-10547)
999553	CVE-2020-8605	WEB-MISC Trend Micro の InterScan Web Security 仮想アプライアンス 6.5 SP2 パッチ 4 より前-RCE の脆弱性 (CVE-2020-8605)
999554	CVE-2019-10068	WEB-MISC Kentico CMS 複数バージョン-認証されていないリモートコード実行の脆弱性 (CVE-2019-10068)
999555	CVE-2020-11108	4.4 までの WEB-MISCPi-hole-Authenticated RCE Vulnerability (CVE-2020-11108)

2020 年 **10** 月のシグネチャアップデート

January 25, 2022

2020-10-13 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 51 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999505		WEB-WORDPRESS WordPress プラグイン wpDiscuz 7.0.0 7.0.4 まで-認証されていない任意のファイルアップロードの脆弱性
999506		WEB-WORDPRESS クイズ＆アンケートマスター-クロスサイトスクリプティング問題における脆弱性機能
999507	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/log_search と cf パラメータを介したパストラバーサルバルン (CVE-2020-8604)
999508	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/コレクションと cf パラメータを介したパストラバーサル Vuln (CVE-2020-8604)
999509	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/log_search とファイルパラメータを介したパストラバーサル Vuln (CVE-2020-8604)
999510	CVE-2020-8604	WEB-MISC Trend Micro IWS VA 6.5 SP2 より前のパッチ 4-/collection とファイルパラメータによるパストラバーサル Vuln (CVE-2020-8604)
999511	CVE-2020-7361	WEB-MISC ZentAo エンタープライズ 8.8.3 以前-リポジトリ編集によるリモートコード実行の脆弱性 (CVE-2020-7361)

署名ルール	**CVE ID**	説明
999512	CVE-2020-7361	WEB-MISC ZentAo Pro 8.8.3 以前-リポジトリ編集によるリモートコード実行の脆弱性 (CVE-2020-7361)
999513	CVE-2020-7361	WEB-MISC ZentAo エンタープライズ 8.8.3 以前-リポジトリ作成によるリモートコード実行の脆弱性 (CVE-2020-7361)
999514	CVE-2020-7361	WEB-MISC ZentAo Pro 8.8.3 以前-リポジトリ作成によるリモートコード実行の脆弱性 (CVE-2020-7361)
999515	CVE-2020-5768	WEB-WORDPRESS Icegram 4.5.1 より前のメール購読者およびニュースレタープラグイン-SQL インジェクションの脆弱性 (CVE-2020-5768)
999516	CVE-2020-5767	WEB-WORDPRESS Icegram 4.5.1 より前のメール購読者およびニュースレタープラグイン-CSRF の脆弱性（CVE-2020-5767）
999517	CVE-2020-15299	WEB-WORDPRESS KingComposer プラグイン 2.9.5 より前-クロスサイトスクリプティングの脆弱性 (CVE-2020-15299)
999518	CVE-2020-13854	WEB-MISC アーティカ Pandora FMS：権限昇格の脆弱性 (CVE-2020-13854)
999519	CVE-2020-13852	WEB-MISC Artica Pandora FMS：ファイルマネージャを介した任意のファイルアップロードの脆弱性（CVE-2020-13852）

署名ルール	**CVE ID**	説明
999520	CVE-2020-13700	WEB-WORDPRESS WordPress プラグイン acf-to-rest-api 3.3.0 より前-URI を介した情報開示の脆弱性（CVE-2020-13700）
999521	CVE-2020-13700	WEB-WORDPRESS WordPress プラグイン acf-to-rest-api 3.3.0 より前-URL を介した情報開示の脆弱性（CVE-2020-13700）
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 から 7.0.1-DOS の脆弱性につながる CSRF バイパス（CVE-2020-13379）
999523	CVE-2020-12851	2.0.7 より前の WEB-MISC Pydio セル：任意のファイル書き込みの脆弱性（CVE-2020-12851）
999524	CVE-2020-12848	2.0.7 より前の WEB-MISC Pydio セル-一時的な共有ユーザーとしてログインする脆弱性（CVE-2020-12848）
999525	CVE-2020-11749	7.47 より前の WEB-Misc Artica Pandora FMS-SNMP ブラウザを介したクロスサイトスクリプティングの脆弱性（CVE-2020-11749）
999526	CVE-2020-11579	WEB-MISC PHPKBV9：ファイル漏洩の脆弱性（CVE-2020-11579）
999527	CVE-2020-10546	3.9.5 より前の WEB-MISC rConfig-SearchColumn を介したコンプライアンスポリシーにおける認証されていない SQLi の脆弱性（CVE-2020-10546）
999528	CVE-2020-10546	3.9.5 より前の WEB-MISC rConfig-SearchField を介したコンプライアンスポリシーにおける認証されていない SQLi の脆弱性（CVE-2020-10546）

署名ルール	**CVE ID**	説明
999529	CVE-2019-16876	1.22.1 より前の WEB-MISC ポータナー：ディレクトリトラバーサルの脆弱性（CVE-2019-16876）
999530		WEB-WORDPRESS-1.5.6 より前の ADNing プラグイン-認証されていない任意のファイル削除の脆弱性
999531		WEB-WORDPRESS-1.5.6 より前の ADning プラグイン-認証されていない任意のファイルアップロードの脆弱性

2020 年 **10** 月のシグネチャアップデート

January 25, 2022

2020-10-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 52 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。また、シグニチャルールログ文字列の一部に、脆弱なバージョンが記載されています。それに応じて有効にする必要があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999500	CVE-2018-14667	WEB-MISC RichFaces フレームワーク 3.X から 3.3.4-ユーザーリソース経由の EL インジェクション (CVE-2018-14667)
999501	CVE-2018-12533	WEB-MISC RichFaces フレームワーク 3.1.0 から 3.3.4-Paint2Resource 経由の EL インジェクション (CVE-2018-12533)
999502	CVE-2015-0279, CVE-2018-12532	WEB-MISC RichFaces フレームワーク 4.X から 4.5.17-メディア出力リソースを介した EL インジェクション (CVE-2015-0279、CVE-2018-12532)
999503	CVE-2013-2165	4.3.3 より前の WEB-MISC RichFaces v4：Java オブジェクトのデシリアライゼーションの脆弱性（CVE-2013-2165）
999504	CVE-2013-2165	3.3.4 より前の WEB-MISCRichFaces v3-Java オブジェクトの逆シリアル化の脆弱性（CVE-2013-2165）

2020 年 **11** 月のシグネチャアップデート

January 25, 2022

2020-11-10 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 53 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999411		WEB-WORDPRESS WordPress プラグイン wpDiscuz 7.0.0 7.0.4 まで-認証されていない任意のファイルアップロードの脆弱性
999412		WEB-WORDPRESS クイズ＆アンケートマスター-クロスサイトスクリプティング問題における脆弱性機能
999413		6.9 より前の WEB-WORDPRESS WordPress プラグインファイルマネージャー-認証されていない ElFinder コマンドの実行脆弱性
999414	CVE-2020-11700	WEB-MISC Titan SpamTitan 7.08 より前-情報開示の脆弱性 (CVE-2020-11700)
999415	CVE-2020-9446	WEB-MISC Apache ofBiz 17.12.03-XML-RPC の安全でないデシリアライゼーションの脆弱性 (CVE-2020-9446)
999416	CVE-2020-9446	WEB-MISC Apache ofBiz 17.12.03-XML-RPC クロスサイトスクリプティングの脆弱性 (CVE-2020-9446)
999417	CVE-2020-9047	WEB-MISC exacqVision ウェブサービス 20.06.3.0 まで-OS コマンドインジェクションの脆弱性 (CVE-2020-9047)

署名ルール	**CVE ID**	説明
999418	CVE-2020-8866	WEB-MISC Horde グループウェアウェブメール版 5.2.22-edit.php を介したファイル脆弱性の無制限アップロード (CVE-2020-8866)
999419	CVE-2020-8866	WEB-MISC Horde グループウェアウェブメール版 5.2.22-add.php を介したファイル脆弱性の無制限アップロード (CVE-2020-8866)
999420	CVE-2020-8865	WEB-MISC Horde グループウェアウェブメール版 5.2.22: edit.php を介した任意のファイルインクルージョンの脆弱性 (CVE-2020-8865)
999421	CVE-2020-8816	4.3.2 より前の WEB-MISC パイホール-removeStatic を介したリモートコード実行の脆弱性 (CVE-2020-8816)
999422	CVE-2020-8816	4.3.2 より前の WEB-MISC パイホール-AddMac 経由でリモートでコードが実行される脆弱性 (CVE-2020-8816)
999423	CVE-2020-8243	9.1R8.2 より前の WEB-MISC パルス接続セキュア：リモートでコードが実行される脆弱性 (CVE-2020-8243)
999424	CVE-2020-8218	9.1R8 より前の WEB-MISC パルス接続セキュア：リモートでコードが実行される脆弱性 (CVE-2020-8218)
999425	CVE-2020-6143, CVE-2020-6144	ウェブその他 OS4ED OpenSIS-/install/Ins1.php を介したコードインジェクションの脆弱性（CVE-2020-6143、CVE-2020-6144）

署名ルール	**CVE ID**	説明
999426	CVE-2020-6142	WEB-MISC OS4ED OpenSIS-modname を介したパストラバーサルの脆弱性 (CVE-2020-6142)
999427	CVE-2020-6141	7.4 より前の WEB-MISC OS4ED OpenSIS-ユーザー名を介した認証されていない SQLi の脆弱性 (CVE-2020-6141)
999428	CVE-2020-6140	7.5 より前の WEB-MISC OS4ED OpenSIS-username_stn_id を介した認証されていない SQLi の脆弱性（CVE-2020-6140）
999429	CVE-2020-6139	7.5 より前の WEB-MISC OS4ED OpenSIS-username_stf_email を介した認証されていない SQLi の脆弱性 (CVE-2020-6139)
999430	CVE-2020-6138	7.5 より前の WEB-MISC OS4ED OpenSIS-uname を介した認証されていない SQLi の脆弱性 (CVE-2020-6138)
999431	CVE-2020-6137	7.5 より前の WEB-MISC OS4ED OpenSIS-password_stf_email を介した認証されていない SQLi の脆弱性 (CVE-2020-6137)
999432	CVE-2020-6125	7.5 より前の WEB-MISC OS4ED OpenSIS：GetSchool.php および u パラメータを介した SQLi の脆弱性（CVE-2020-6125）
999433	CVE-2020-6124	7.5 より前の WEB-MISC OS4ED openSIS-EmailCheckOthers.php を介した SQLi の脆弱性 (CVE-2020-6124)

署名ルール	**CVE ID**	説明
999434	CVE-2020-6123	7.5 より前の WEB-MISC OS4ED OpenSIS-EmailCheck.php および p_id パラメータを介した SQLi の脆弱性（CVE-2020-6123）
999435	CVE-2020-6123	7.5 より前の WEB-MISC OS4ED OpenSIS-EmailCheck.php および電子メールパラメータを介した SQLi の脆弱性（CVE-2020-6123）
999436	CVE-2020-6122	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および mn パラメータを介した SQLi の脆弱性（CVE-2020-6122）
999437	CVE-2020-6121	7.5 より前の WEB-MISC OS4ED OpenSIS-CheckDuplicateStudent.php および ln パラメータを介した SQLi の脆弱性（CVE-2020-6121）
999438	CVE-2020-6120	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および fn パラメータを介した SQLi の脆弱性（CVE-2020-6120）
999439	CVE-2020-6119	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および年ごとのパラメータを介した SQLi の脆弱性（CVE-2020-6119）

署名ルール	**CVE ID**	説明
999440	CVE-2020-6118	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および bmonth パラメータを介した SQLi の脆弱性 (CVE-2020-6118)
999441	CVE-2020-6117	7.5 より前の WEB-MISC OS4ED OpenSIS: CheckDuplicateStudent.php および bday パラメータを介した SQLi の脆弱性（CVE-2020-6117）
999442	CVE-2020-5780	WEB-WORDPRESS WordPress プラグイン 4.5.6 より前のメール購読者とニュースレター-メール偽造の脆弱性（CVE-2020-5780）
999443	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-JSON-RPC を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999444	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4: リモートメソッドによるセキュアでない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999445	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-RemoteJavaScript を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)

署名ルール	**CVE ID**	説明
999446	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-JSON-RPC を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999447	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4：リモートメソッドによるセキュアでない Java デシリアライゼーションの脆弱性（CVE-2020-4280）
999448	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 および 7.4-RemoteJavaScript を介した安全でない Java デシリアライゼーションの脆弱性 (CVE-2020-4280)
999449	CVE-2020-24786	WEB-MISC Zoho ManageEngine AdManager Plus 7.0 ビルド 55 より前-不適切な認証の脆弱性（CVE-2020-24786）
999450	CVE-2020-24389	WEB-WORDPRESS 1.3.5.5 より前の複数ファイルアップローダープラグインをドラッグアンドドロップする：セキュリティバイパスの脆弱性（CVE-2020-24389）
999451	CVE-2020-24046	WEB-MISC TitanHQ SpamTitan Gateway 7.08 - 権限昇格の脆弱性（CVE-2020-24046）
999452	CVE-2020-17506	WEB-MISC Artica ウェブプロキシ 4.30.000000-Apikey パラメータによる事前認証 SQL インジェクションの脆弱性（CVE-2020-17506）

署名ルール	**CVE ID**	説明
999453	CVE-2020-17505	WEB-MISC Artica Web プロキシ 4.30.000000-サービス-cmds-peform パラメータを介した OS コマンドインジェクションの脆弱性（CVE-2020-17505）
999454	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ユーザー/アイテム経由の SQLi 脆弱性 (CVE-2020-17463)
999455	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/サイト変数/アイテムを介した SQLi の脆弱性（CVE-2020-17463）
999456	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/パーミッション/アイテムを介した SQLi の脆弱性（CVE-2020-17463）
999457	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ページ/アイテム経由の SQLi 脆弱性 (CVE-2020-17463)
999458	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ナビゲーション/アイテム経由の SQLi 脆弱性（CVE-2020-17463）
999459	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/logs/items 経由の SQLi 脆弱性 (CVE-2020-17463)
999460	CVE-2020-17463	WEB-MISC 燃料 CMS 1.4.8-/fuel/ブロック/アイテム経由の SQLi 脆弱性 (CVE-2020-17463)
999461	CVE-2020-16875	WEB-MISC Microsoft Exchange Server-DLP ポリシーリモートでコードが実行される脆弱性（CVE-2020-16875）

署名ルール	**CVE ID**	説明
999462	CVE-2020-16171	WEB-MISC 12.5 ビルド 16342 より前のアクロニスサイバーバックアップ-シャードヘッダー経由の SSRF の脆弱性 (CVE-2020-16171)
999463	CVE-2020-14947	2.8 より前の WEB-MISC OCS インベントリ-SNMP_MIB_DIRECTORY を介した OS コマンドインジェクションの脆弱性（CVE-2020-14947）
999464	CVE-2020-14947	2.8 より前の WEB-MISC OCS インベントリ：mib_file を介した OS コマンドインジェクションの脆弱性（CVE-2020-14947）
999465	CVE-2020-14008	WEB-MISC Zoho ManageEngine アプリケーションマネージャー 14710 まで-リモートでコードが実行される脆弱性 (CVE-2020-14008)
999466	CVE-2020-13925	WEB-MISC 3.1.0 より前の Apache Kylin-ジョブを介したリモートコード実行の脆弱性 (CVE-2020-13925)
999467	CVE-2020-13925	WEB-MISC 3.1.0 より前の Apache Kylin-プロジェクト経由でリモートでコードが実行される脆弱性（CVE-2020-13925）
999468	CVE-2020-13854	WEB-MISC Artica Pandora FMS：権限昇格の脆弱性 (CVE-2020-13854)
999469	CVE-2020-13405	1.1.20 より前の WEB-MISC マイクロウェーバー：認証されていない情報漏えいの脆弱性 (CVE-2020-13405)

署名ルール	**CVE ID**	説明
999470	CVE-2020-13376	WEB-MISC SecureEnvoy SecurMail 9.3.503-SecureEnvoyReply クッキーパストラバーサルの脆弱性 (CVE-2020-13376)
999471	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-ドメイン経由の OS コマンドインジェクションの脆弱性 (CVE-2020-13159)
999472	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica ウェブプロキシ：netbiosname を介した OS コマンドインジェクションの脆弱性（CVE-2020-13159）
999473	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-エイリアス経由の OS コマンドインジェクションの脆弱性 (CVE-2020-13159)
999474	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ：ホスト名を介した OS コマンドインジェクションの脆弱性 (CVE-2020-13159)
999475	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-dhclient_server を介した OS コマンドインジェクションの脆弱性（CVE-2020-13159）
999476	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica Web プロキシ-dhclient_interface を介した OS コマンドインジェクションの脆弱性（CVE-2020-13159）

署名ルール	**CVE ID**	説明
999477	CVE-2020-13159	4.30.000000 より前の WEB-MISC Artica ウェブプロキシ：dhclient_mac を介した OS コマンドインジェクションの脆弱性（CVE-2020-13159）
999478	CVE-2020-13158	4.30.000000 より前の WEB-MISC Artica Web プロキシ：ポップアップによるパストラバーサルの脆弱性（CVE-2020-13158）
999479	CVE-2020-12851	2.0.7 より前の WEB-MISC Pydio セル：任意のファイル書き込みの脆弱性（CVE-2020-12851）
999480	CVE-2020-12848	2.0.7 より前の WEB-MISC Pydio セル-一時的な共有ユーザーとしてログインする脆弱性（CVE-2020-12848）
999481	CVE-2020-11699	WEB-MISC Titan SpamTitan 7.08 より前-リモートでコードが実行される脆弱性（CVE-2020-11699）
999482	CVE-2020-11579	WEB-MISC PHPKBV9：ファイル漏洩の脆弱性（CVE-2020-11579）
999483	CVE-2020-10818	WEB-MISC Artica Web Proxy 4.26-fw.system.info.php を介した OS コマンドインジェクションの脆弱性（CVE-2020-10818）
999484	CVE-2020-10228	バージョン 20 より前の WEB-MISC Vtenext CE-危険なタイプの脆弱性を持つファイルの無制限アップロード（CVE-2020-10228）

署名ルール	**CVE ID**	説明
999485	CVE-2020-10204	3.21.2 より前の WEB-MISC ソナタイプネクサスリポジトリマネージャー-CoreUI_user ロールを介した RCE の脆弱性 (CVE-2020-10204)
999486	CVE-2020-10204	3.21.2 より前の WEB-MISC ソナタイプネクサスリポジトリマネージャー-CoreUI_Role 権限による RCE の脆弱性 (CVE-2020-10204)
999487	CVE-2020-10204	3.21.2 より前の WEB-MISC ソナタイプネクサスリポジトリマネージャー-CoreUI_Role ロールを介した RCE の脆弱性 (CVE-2020-10204)
999488	CVE-2020-10199	3.21.2 より前の WEB-MISC Sonatype ネクサスリポジトリマネージャー-REST エンドポイント/bower/グループを介した RCE の脆弱性 (CVE-2020-10199)
999489	CVE-2020-10199	3.21.2 より前の WEB-MISC Sonatype ネクサスリポジトリマネージャー-REST エンドポイント/go/group を介した RCE の脆弱性 (CVE-2020-10199)
999490	CVE-2020-10199	3.21.2 より前の WEB-MISC Sonatype ネクサスリポジトリマネージャー-REST エンドポイント/docker/グループを介した RCE の脆弱性 (CVE-2020-10199)
999491	CVE-2019-19699	WEB-MISC Centreon 19.10 まで：リモートでコードが実行される脆弱性（CVE-2019-19699）
999492	CVE-2019-19499	WEB-MISC Apache Grafana 6.4.3 まで：任意のファイル読み取りの脆弱性（CVE-2019-19499）

署名ルール	**CVE ID**	説明
999493	CVE-2019-18394	WEB-MISC 4.4.2 までのリアルタイムオープンファイアに火をつける-FaviconServlet サーバー側リクエストフォージェリの脆弱性 (CVE-2019-18394)
999494	CVE-2019-18393	WEB-MISC Ignite リアルタイム Openfire 4.4.2 まで-プラグインサーブレットディレクトリトラバーサルの脆弱性 (CVE-2019-18393)
999495	CVE-2019-16759	5.6.2 より前の WEB-MISC vBulletin：ネストされたテンプレートを経由してリモートでコードが実行される脆弱性 (CVE-2019-16759)
999496	CVE-2019-15715	1.3.20 および 2.22.1 より前の WEB-Misc mantisBT: neato_tool を介したリモートでのコード実行の脆弱性 (CVE-2019-15715)
999497	CVE-2019-15715	1.3.20 および 2.22.1 より前の WEB-Misc mantisBT：dot_tool を介したリモートでのコード実行の脆弱性（CVE-2019-15715）
999498	CVE-2019-11043	WEB-MISC PHP-FPM 複数バージョン-領域外書き込み脆弱性により任意のコードが実行される (CVE-2019-11043)
999499		WEB-WORDPRESSWordPress プラグインが最大 2.7.6 まで自動最適化-認証された任意のファイルアップロードの脆弱性

2020 年 12 月のシグネチャーアップデート

January 25, 2022

2020-12-02 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 54 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。シグニチャアップデートバージョン 54 の一部として、シグニチャ 999720 のログ文字列が変更され、ASCI 文字だけが含まれるようになりました。

一般的な脆弱性エントリ (CVE) の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999394	CVE-2020-8255	9.1R9 より前の WEB-MISC パルスコネクトセキュア-情報漏えいの脆弱性（CVE-2020-8255）
999395	CVE-2020-6128	7.5 より前の WEB-MISC OS4ED openSIS-CoursePeriodModal.php を介した SQLi の脆弱性（CVE-2020-6128）
999396	CVE-2020-6126, CVE-2020-6127	7.5 より前のウェブその他 OS4ED openSIS-CoursePeriodModal.php を介した SQLi の脆弱性（CVE-2020-6126、CVE-2020-6127）

署名ルール	**CVE ID**	説明
999397	CVE-2020-28328	7.11.16 より前の WEB-MISC SuiteCRM：リモートでコードが実行される脆弱性（CVE-2020-28328）
999398	CVE-2020-27995	WEB-MISC Zoho ManageEngine アプリケーションマネージャー 14 ビルド前 14560-SQL インジェクションの脆弱性（CVE-2020-27995）
999399	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバ-/servic/ を介した認可バイパスの脆弱性 (CVE-2020-26879)
999400	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバー-/reboot による認証バイパスの脆弱性 (CVE-2020-26879)
999401	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバー-/patch/ を介した認証バイパスの脆弱性 (CVE-2020-26879)
999402	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバー-/upgrade/による認証バイパスの脆弱性 (CVE-2020-26879)
999403	CVE-2020-26879	1.6.0 より前の WEB-MISC Ruckus vRiot サーバ-/module/による認可バイパスの脆弱性 (CVE-2020-26879)
999404	CVE-2020-26878	1.6.0 より前の WEB-MISC Ruckus vRiot サーバ：任意の OS コマンドインジェクションの脆弱性（CVE-2020-26878）

署名ルール	**CVE ID**	説明
999405	CVE-2020-25790	WEB-MISC タイプセッター CMS 5.x から 5.1：セキュアでないファイルアップロードの脆弱性（CVE-2020-25790）
999406	CVE-2020-25540	WEB-MISC ThinkAdmin v6-ディレクトリトラバーサルの脆弱性（CVE-2020-25540）
999407	CVE-2020-14883	WEB-MISC Oracle WebLogic サーバー-認証済みリモートコード実行の脆弱性（CVE-2020-14883）
999408	CVE-2020-14882, CVE-2020-14750	WEB-MISC Oracle WebLogic サーバー-認証バイパスの脆弱性（CVE-2020-14882、CVE-2020-14750）
999409	CVE-2020-11975, CVE-2020-13942	1.5.2 より前の WEB-MISC Apache Uonomi-リモートでコードが実行される脆弱性（CVE-2020-11975、CVE-2020-13942）
999410	CVE-2020-11803	WEB-MISC Titan SpamTitan 7.08 より前-リモートでコードが実行される脆弱性（CVE-2020-11803）

2020 年 **12** 月のシグネチャーアップデート

January 25, 2022

2020-12-17 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 55 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999377		1.21.11 より前の WEB-WORDPRESS TI WooCommerce ウィッシュリストプラグイン-tinvwl_export_settings を介した情報開示の脆弱性
999378		1.21.11 より前の WEB-WORDPRESS TI WooCommerce ウィッシュリストプラグイン-WP オプションは tinvwl_import_settings 経由で脆弱性を変更します
999379	CVE-2020-6134	7.5 より前の Web-MISC OS4ED openSIS: MassDropModal.php を介した SQLi の脆弱性（CVE-2020-6134）
999380	CVE-2020-6133	7.5 より前の WEB-MISC OS4ED openSIS: CourseMoreInfo.php を介した SQLi の脆弱性（CVE-2020-6133）
999381	CVE-2020-6132	7.5 より前の WEB-MISC OS4ED openSIS: ChooseCP.php を介した SQLi の脆弱性（CVE-2020-6132）

署名ルール	**CVE ID**	説明
999382	CVE-2020-6131	7.5 より前の WEB-MISC OS4ED openSIS-MassScheduleSessionSet.php を介した SQLi の脆弱性 (CVE-2020-6131)
999383	CVE-2020-6130	7.5 より前の Web-MISC OS4ED openSIS: MassDropSessionSet.php を介した SQLi の脆弱性 (CVE-2020-6130)
999384	CVE-2020-6129	7.5 より前の Web-MISC OS4ED openSIS-CpSessionSet.php を介した SQLi の脆弱性 (CVE-2020-6129)
999385	CVE-2020-35234	1.4.4 より前の WEB-WORDPRES Easy WP SMTP プラグイン-情報漏えいの脆弱性 (CVE-2020-35234)
999386	CVE-2020-25042	WEB-MISC Mara CMS 7.5: 任意のファイルアップロードの脆弱性 (CVE-2020-25042)
999387	CVE-2020-13526	WEB-MISC ProcessMaker-クライアントセットアップ Ajax を介した SQL インジェクションの脆弱性 (CVE-2020-13526)
999388	CVE-2020-13525	WEB-MISC ProcessMaker-ReportTables_AJAX を介した SQL インジェクションの脆弱性 (CVE-2020-13525)
999389	CVE-2020-12147	WEB-MISC Silver Peak Unity Orchestrator -SQLExecution REST API を介した任意の MySQL クエリの脆弱性 (CVE-2020-12147)

署名ルール	**CVE ID**	説明
999390	CVE-2020-12146	WEB-MISC Silver Peak Unity Orchestrator -デバッグファイル REST API を介したパストラバーサルの脆弱性（CVE-2020-12146）
999391	CVE-2020-12145	WEB-MISC Silver Peak Unity Orchestrator -認証バイパスの脆弱性（CVE-2020-12145）
999392	CVE-2019-8394	WEB-MISC 10.0 ビルド 10012 より前の Zoho ManageEngine ServiceDesk Plus-任意のファイルアップロードの脆弱性（CVE-2019-8394）
999393	CVE-2019-11447	WEB-Misc CutePHP CuteNews 2.1.2-リモートでコードが実行される脆弱性（CVE-2019-11447）

2021 年 **1** 月のシグネチャーアップデート

January 25, 2022

2021-01-18 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 56 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999366	CVE-2020-8466	WEB-MISC ビルド 1919 より前の Trend Micro IWSSVA 6.5 SP2：認証されていない OS コマンドインジェクションの脆弱性（CVE-2020-8466）
999367	CVE-2020-6135	7.5 より前の WEB-MISC OS4ED openSIS：Validator.php を介した SQLi の脆弱性（CVE-2020-6135）
999368	CVE-2020-4001	WEB-MISC VMware SD-WAN Orchestrator-パスザハッシュの脆弱性（CVE-2020-4001）
999369	CVE-2020-4000	WEB-MISC VMware SD-WAN Orchestrator-パストラバーサルの脆弱性（CVE-2020-4000）
999370	CVE-2020-3984	WEB-MISC VMware SD-WAN Orchestrator-モジュラスを介した SQL インジェクションの脆弱性 (CVE-2020-3984)
999371	CVE-2020-35606	1.962 までの WEB-MISC Webmin：リモートでコードが実行される脆弱性（CVE-2020-35606）
999372	CVE-2020-17143	WEB-MISC Microsoft Exchange Server-情報開示の脆弱性 (CVE-2020-17143)
999373	CVE-2020-17141	WEB-MISC Microsoft Exchange Server-ルート苦情によるリモートコード実行の脆弱性 (CVE-2020-17141)

署名ルール	**CVE ID**	説明
999374	CVE-2020-10816	WEB-MISC ビルド 14790 より前の Zoho ManageEngine アプリケーションマネージャー 14-不適切な認証の脆弱性（CVE-2020-10816）
999375	CVE-2019-5533	WEB-MISC VMware SD-WAN Orchestrator-情報開示の脆弱性（CVE-2019-5533）
999376	CVE-2018-15961	WEB-MISC アップデート 6 または 14 より前の Adobe ColdFusion 12-任意のファイルアップロードの脆弱性（CVE-2018-15961）

2021 年 **2** 月の署名の更新

January 25, 2022

2021-02-03 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 57 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999339		WEB-MISC Zoom ミーティングコネクタ 4.6.348.20201217-ProxyPasswd を介したリモートでのコード実行の脆弱性
999340		WEB-MISC Zoom ミーティングコネクタ 4.6.348.20201217-ProxyName 経由でリモートでコードが実行される脆弱性
999341	CVE-2021-3129	2.5.2 より前の WEB-MISC イグニッション-認証されていないリモートコード実行の脆弱性 (CVE-2021-3129)
999342	CVE-2021-3025	4.5.4.2 より前の WEB-MISC インビジョンコミュニティ IPS コミュニティスイート-SortDir を介した SQL インジェクションの脆弱性 (CVE-2021-3025)
999343	CVE-2021-2109	WEB-MISC Oracle WebLogic サーバー-JNDI インジェクションによるリモートコード実行の脆弱性 (CVE-2021-2109)
999344	CVE-2020-7200	WEB-MISC HPE システムインサイトマネージャー 7.6.x-AMF セキュアでないデシリアライゼーションの脆弱性 (CVE-2020-7200)
999345	CVE-2020-7199	WEB-MISC HPE EIM 1.21 より前-/プライベート/EIMApplianceIP に不適切な認証の脆弱性 (CVE-2020-7199)
999346	CVE-2020-7199	WEB-MISC HPE EIM 1.21 より前-/プライベート/adminPassReset に不適切な認証の脆弱性 (CVE-2020-7199)

署名ルール	**CVE ID**	説明
999347	CVE-2020-7199	WEB-MISC HPE EIM 1.21 より前-/プライベート/リセットアプライアンスに不適切な認証の脆弱性 (CVE-2020-7199)
999348	CVE-2020-6136	7.5 より前の Web-MISC OS4ED openSIS: DownloadWindow.php を介した SQLi の脆弱性 (CVE-2020-6136)
999349	CVE-2020-35729	WEB-MISC ログサーバー 2.4.1 以前-OS コマンドインジェクションの脆弱性（CVE-2020-35729）
999350	CVE-2020-35701	WEB-MISC Cacti 1.2.16 およびそれ以前-site_id を介した SQL インジェクションの脆弱性 (CVE-2020-35701)
999351	CVE-2020-35489	5.3.2 より前の WEB-WORDPRESS お問い合わせフォーム 7：無制限のファイルアップロードの脆弱性 (CVE-2020-35489)
999352	CVE-2020-27615	1.6.4 より前の WEB-WORDPRESS Loginizer プラグイン-SQL インジェクションの脆弱性（CVE-2020-27615）
999353	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/sitevariables/create を介した XSS の脆弱性 (CVE-2020-26046)
999354	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/sitevariables/edit を介した XSS の脆弱性 (CVE-2020-26046)

署名ルール	**CVE ID**	説明
999355	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/ナビゲーション/作成を介した XSS の脆弱性（CVE-2020-26046）
999356	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/ナビゲーション/編集による XSS の脆弱性（CVE-2020-26046）
999357	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/blocks/create を介した XSS の脆弱性（CVE-2020-26046）
999358	CVE-2020-26046	WEB-MISC 燃料 CMS 1.4.11 およびそれ以前-/fuel/blocks/edit を介した XSS の脆弱性（CVE-2020-26046）
999359	CVE-2020-26045	WEB-MISC 燃料 CMS 1.4.11-/fuel/パーミッション/作成による SQLi の脆弱性（CVE-2020-26045）
999360	CVE-2020-17519	1.11.3 より前の WEB-MISC Apache Flink：任意のファイル漏洩の脆弱性（CVE-2020-17519）
999361	CVE-2020-17518	WEB-MISC Apache Flink 1.5.1 から 1.11.2：任意のロケーションのファイルアップロードの脆弱性（CVE-2020-17518）
999362	CVE-2019-16010	19.2.2 より前の WEB-MISC Cisco SD-WAN vManage-格納された XSS の脆弱性（CVE-2019-16010）
999363	CVE-2019-15000	WEB-MISC VMware Bitbucket サーバおよびデータセンター-At 経由の Git コマンドインジェクションの脆弱性（CVE-2019-15000）

署名ルール	**CVE ID**	説明
999364	CVE-2019-15000	WEB-MISC VMware Bitbucket サーバおよびデータセンター-Until/untilId を介した Git コマンドインジェクションの脆弱性 (CVE-2019-15000)
999365	CVE-2019-15000	WEB-MISC VMware Bitbucket サーバおよびデータセンター-since/sinceID を介した Git コマンドインジェクションの脆弱性 (CVE-2019-15000)

2021 年 **2** 月の署名の更新

January 25, 2022

2021-02-17 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 58 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999328	CVE-2021-3317	WEB-MISC KLog サーバ 2.4.1 およびそれ以前-OS コマンドインジェクションの脆弱性 (CVE-2021-3317)
999329	CVE-2021-3110	1.7.7.1 より前の WEB-MISC PrestaShop-id_products (CVE-2021-3110) を介した SQL インジェクションの脆弱性
999330	CVE-2021-3110	1.7.7.1 より前の WEB-MISC PrestaShop-/module/ProductComments/commentGrade (CVE-2021-3110) を介した SQL インジェクションの脆弱性
999331	CVE-2021-25646	WEB-MISC 0.20.1 より前の Apache Druid-リモートコード実行の脆弱性 (CVE-2021-25646)
999332	CVE-2020-36171	3.0.14 より前の WEB-WORDPRESS・エレメンターページ・ビルダー・プラグイン-XSS の脆弱性 (CVE-2020-36171)
999333	CVE-2020-35765	WEB-MISC Zoho ManageEngine アプリケーションマネージャビルド前の 15000-SQL インジェクションの脆弱性 (CVE-2020-35765)
999334	CVE-2020-35589	2.15.2 より前にリロードされた WEB WORDPRESS 制限ログイン試行-クロスサイトスクリプティングの脆弱性の反映 (CVE-2020-35589)

署名ルール	**CVE ID**	説明
999335	CVE-2020-26282	2.1.2 より前の WEB-MISC BrowserUp プロキシ-mostRecentEntry を介した RCE の脆弱性につながるテンプレートインジェクション（CVE-2020-26282）
999336	CVE-2020-26282	2.1.2 より前の WEB-MISC BrowserUp プロキシ-エントリを介した RCE の脆弱性につながるテンプレートインジェクション（CVE-2020-26282）
999337	CVE-2020-14815	WEB-MISC オラクル・ビジネス・インテリジェンス・エンタープライズ・エディション-クロスサイト・スクリプティングの脆弱性の反映（CVE-2020-14815）
999338		1.2.5.4 より前の WEB-WORDPRESS のお問い合わせフォーム 7 データベースアドオン-一括削除による SQLi 脆弱性

2021 年 **3** 月のシグネチャアップデート

January 25, 2022

2021-03-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 59 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999313	CVE-2021-25299	WEB-MISC NagiosXi 5.7.5 まで-URL 経由の XSS 脆弱性 (CVE-2021-25299)
999314	CVE-2021-25298	WEB-MISC NagiosXi 5.7.5 まで-DigitalOcean Wizard を介したリモートコード実行の脆弱性 (CVE-2021-25298)
999315	CVE-2021-25297	WEB-MISC NagiOSXI 5.7.5 まで-スイッチウィザードによるリモートコード実行の脆弱性 (CVE-2021-25297)
999316	CVE-2021-25296	WEB-MISC NagiOSXI 5.7.5 まで-WindowsWMI ウィザードによるリモートコード実行の脆弱性 (CVE-2021-25296)
999317	CVE-2021-24164	3.4.34.1 より前の WEB-WORDPRESS 忍者フォームプラグイン-情報開示の脆弱性 (CVE-2021-24164)
999318	CVE-2021-24163	3.4.34 より前の WEB-WORDPRESS 忍者フォームプラグイン-認可バイパスの脆弱性 (CVE-2021-24163)
999319	CVE-2021-21972	WEB-MISC VMware vCenter Server プラグイン-リモートコード実行の脆弱性 (CVE-2021-21972)

署名ルール	**CVE ID**	説明
999320	CVE-2020-35129	3.2.4 より前の WEB-MISC Mautic-新しいソーシャルモニタリングフォームによる XSS の脆弱性（CVE-2020-35129）
999321	CVE-2020-35129	3.2.4 より前の WEB-MISC Mautic-ソーシャルモニタリングフォームの編集による XSS の脆弱性（CVE-2020-35129）
999322	CVE-2020-35128	3.2.4 より前の WEB-MISC Mautic-新しい企業フォームを介した XSS 脆弱性 (CVE-2020-35128)
999323	CVE-2020-35128	3.2.4 より前の WEB-MISC Mautic-会社の編集フォームによる XSS 脆弱性 (CVE-2020-35128)
999324	CVE-2020-35125	3.2.4 より前の WEB-MISC Mautic-リファラーヘッダーを介した XSS の脆弱性 (CVE-2020-35125)
999325	CVE-2020-35125	3.2.4 より前の WEB-MISC Mautic-mauticform[return] による XSS 脆弱性 (CVE-2020-35125)
999326	CVE-2020-13933	WEB-MISC 1.6.0 より前の Apache Shiro-セミコロンによる認証バイパスの脆弱性 (CVE-2020-13933)
999327	CVE-2020-13921, CVE-2020-9483	WEB-MISC Apache SkyWalking 8.4.0 より前-QueryLogs 機能による SQL インジェクションの脆弱性 (CVE-2020-13921、CVE-2020-9483)

2021 年 3 月のシグネチャアップデート

January 25, 2022

2021-03-09 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 60 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ (CVE) の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999311	CVE-2021-26855	WEB-MISC Microsoft Exchange Server-X-anonResource-Backend (CVE-2021-26855) を介したリモートコード実行の脆弱性
999312	CVE-2021-26855	WEB-MISC Microsoft Exchange Server-X-berource を介したリモートコード実行の脆弱性 (CVE-2021-26855)

2021 年 3 月のシグネチャアップデート

January 25, 2022

2021-03-11 週目に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウ

ンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 61 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999308	CVE-2021-21302	1.7.7.2 より前の WEB-MISC PrestaShop-CSV インジェクションの脆弱性（CVE-2021-21302）
999309	CVE-2020-35749	WEB-WORDPRESS 2.9.4 より前のシンプルなジョブボード-任意のファイル開示の脆弱性（CVE-2020-35749）
999310	CVE-2019-16012	WEB-MISC 19.2.2 より前の Cisco SD-WAN vManage-SQL インジェクションの脆弱性（CVE-2019-16012）

2021 年 **3** 月のシグネチャアップデート

January 25, 2022

2021-03-11 週目に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 62 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999307	CVE-2021-27065	WEB-MISC Microsoft Exchange Server-リモートコード実行の脆弱性（CVE-2021-27065）

2021 年 **4** 月のシグネチャアップデート

January 25, 2022

2021-04-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 63 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999294	CVE-2021-3273	WEB-MISC NagiOSXI 5.7 より前-コードインジェクションの脆弱性（CVE-2021-3273）
999295	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-ssh_priv (CVE-2021-3197) を介したリモートコード実行の脆弱性
999296	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-ssh_port を介したリモートコード実行の脆弱性（CVE-2021-3197）
999297	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-ssh_options によるリモートコード実行の脆弱性（CVE-2021-3197）
999298	CVE-2021-3197	3002.3 より前の WEB-MISC SaltStack-JSON オブジェクトの proxyCommand によるリモートコード実行の脆弱性（CVE-2021-3197）
999299	CVE-2021-25282	3002.3 より前の WEB-MISC SaltStack-pillar_roots.write を介したパストラバーサルの脆弱性 (CVE-2021-25282)
999300	CVE-2021-24166	3.4.34 より前の WEB-WORDPRESS 忍者フォームプラグイン-CSRF の脆弱性（CVE-2021-24166）
999301	CVE-2021-24085	WEB-MISC Microsoft Exchange Server-なりすましの脆弱性 (CVE-2021-24085)
999302	CVE-2021-22986	WEB-MISC F5 iControl REST API-リモートコード実行の脆弱性（CVE-2021-22986）

署名ルール	**CVE ID**	説明
999303	CVE-2021-21978	WEB-MISC VMware View Planner Harness 4.6 より前のセキュリティパッチ 1-リモートコード実行の脆弱性 (CVE-2021-21978)
999304	CVE-2020-23132	WEB-MISC Joomla! 3.9.25 より前-file_path を介した安全でない com_media アップロードパスの脆弱性 (CVE-2020-23132)
999305	CVE-2020-23132	WEB-MISC Joomla! 3.9.25 より前-安全でない com_media アップロードパスの脆弱性 image_path (CVE-2020-23132)
999306	CVE-2020-22425	20.10.4 より前の WEB-MISC Centreon-SQL インジェクションの脆弱性（CVE-2020-22425）

2021 年 **4** 月のシグネチャアップデート

January 25, 2022

2021-04-22 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 64 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999275	CVE-2021-3378	WEB-MISC FortiLogger 4.4.2.2-認証されていない任意のファイルアップロードの脆弱性 (CVE-2021-3378)
999276	CVE-2021-28925	2.4.3 より前の WEB-MISC Nagios ネットワークアナライザ-SQL インジェクションの脆弱性（CVE-2021-28925）
999277	CVE-2021-28924	2.4.3 より前の WEB-MISC Nagios ネットワークアナライザ-XSS の脆弱性 (CVE-2021-28924)
999278	CVE-2021-27927	WEB-MISC Zabbix-CSRF 脆弱性によるアクション =authentication.update (CVE-2021-27927)
999279	CVE-2021-26295	WEB-MISC Apache ofBiz 17.12.06-認証されていない任意の逆シリアル化の脆弱性 (CVE-2021-26295)
999280	CVE-2021-25770	WEB-MISC JetBrains YouTrack 以前の 2020.5.3123-サーバーサイドテンプレートインジェクションの脆弱性（CVE-2021-25770）
999281	CVE-2021-25283	3002.5 より前の WEB-MISC SaltStack-リモートコード実行の脆弱性（CVE-2021-25283）
999282	CVE-2021-25283	3002.5 より前の WEB-MISC SaltStack-JSON オブジェクトを介したリモートコード実行の脆弱性（CVE-2021-25283）

署名ルール	**CVE ID**	説明
999283	CVE-2021-24218	3.0.4 より前の WordPress プラグインのための WEB-WORDPRESS フェイスブック-保存されたクロスサイトスクリプティングの脆弱性 (CVE-2021-24218)
999284	CVE-2021-24217	3.0.2 より前の WordPress プラグインのための WEB-WORDPRESS フェイスブック-PHP オブジェクトインジェクションの脆弱性 (CVE-2021-24217)
999285	CVE-2021-24209	1.7.2 より前の WEB-WORDPRESS WP スーパーキャッシュプラグイン-wp-cache-config.php のリモートコード実行の脆弱性 (CVE-2021-24209)
999286	CVE-2021-24209	1.7.2 より前の WEB-WORDPRESS WP スーパーキャッシュプラグイン-任意のコードインジェクションの脆弱性 (CVE-2021-24209)
999287	CVE-2021-24165	3.4.34 より前の WEB-WORDPRESS 忍者フォームプラグイン-オープンリダイレクトの脆弱性（CVE-2021-24165）
999288	CVE-2021-21975	WEB-MISC vRealize オペレーションマネージャ-認証されていないサーバサイド要求偽造の脆弱性 (CVE-2021-21975)
999289	CVE-2020-35578	WEB-MISC Nagios XI 5.8.0 より前のリモートコード実行の脆弱性 (CVE-2020-35578)

署名ルール	**CVE ID**	説明
999290	CVE-2020-2766	WEB-MISC Oracle WebLogic Server-認証されていない SSRF の脆弱性（CVE-2020-2766）
999291	CVE-2020-17523	WEB-MISC 1.7.1 より前の Apache Shiro-スペースを介した認証バイパスの脆弱性（CVE-2020-17523）
999292	CVE-2020-17523	WEB-MISC 1.7.1 より前の Apache Shiro-ドットによる認証バイパスの脆弱性（CVE-2020-17523）
999293	CVE-2020-15160	1.7.6.8 より前の WEB-MISC PrestaShop-SQL インジェクションの脆弱性（CVE-2020-15160）

2021 年 **6** 月のシグネチャアップデート

January 25, 2022

2021-06-02 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 65 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999243	CVE-2021-31761	1.974 より前の WEB-MISC Webmin-/servers/link.cgi/ (CVE-2021-31761) を介した XSS 脆弱性
999244	CVE-2021-31761	1.974 より前の WEB-MISC Webmin-/tunnel/link.cgi/ (CVE-2021-31761) を介した XSS 脆弱性
999245	CVE-2021-31166	WEB-IIS Microsoft HTTP プロトコルスタック-リモートコード実行の脆弱性 (CVE-2021-31166)
999246	CVE-2021-29447	5.7.1 より前の WEB-WORDPRESS WordPress-メディアライブラリ XXE の脆弱性（CVE-2021-29447）
999247	CVE-2021-28157	2021.1 および 2020.3.18 より前の WEB-MISC デボリューションズサーバ-ユーザ削除による SQL インジェクションの脆弱性 (CVE-2021-28157)
999248	CVE-2021-27905	WEB-MISC Apache Solr 8.2.2 より前のバージョン-leaderURL を介したレプリケーションハンドラー SSRF の脆弱性 (CVE-2021-27905)
999249	CVE-2021-27905	WEB-MISC Apache Solr 8.2.2 より前のバージョン-MasterURL を介したレプリケーションハンドラー SSRF 脆弱性 (CVE-2021-27905)
999250	CVE-2021-27890	1.8.26 より前の WEB-MISC MyBB-テーマプロパティ SQL インジェクションの脆弱性 (CVE-2021-27890)

署名ルール	**CVE ID**	説明
999251	CVE-2021-27850, CVE-2019-0195	WEB-MISC Apache Tapestry-認証されていない情報開示の脆弱性 (CVE-2021-27850 および CVE-2019-0195)
999252	CVE-2021-27183	20.0.4 より前の WEB-MISC MDaemon-任意のファイル書き込みの脆弱性 (CVE-2021-27183)
999253	CVE-2021-27181	20.0.4 より前の WEB-MISC MDaemon-アンチ CSRF トークン固定の脆弱性 (CVE-2021-27181)
999254	CVE-2021-27180	20.0.4 より前の WEB-MISC MDaemon-反射 XSS 脆弱性 (CVE-2021-27180)
999255	CVE-2021-24340	13.0.8 より前の WEB-WORDPRESSWP 統計-認証されていない SQL インジェクションの脆弱性 (CVE-2021-24340)
999256	CVE-2021-24171	WEB-WORDPRESS WooCommerce アップロードファイルプラグイン 59.4-パストラバーサルの脆弱性 (CVE-2021-24171)
999257	CVE-2021-24171	WEB-WORDPRESS WooCommerce アップロードファイルプラグイン 59.4-任意のファイルアップロードの脆弱性 (CVE-2021-24171)
999258	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前-UserServlet と user_password による SQLi 脆弱性 (CVE-2021-22658)

署名ルール	**CVE ID**	説明
999259	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前-UserServlet と user_name を介した SQLi 脆弱性（CVE-2021-22658）
999260	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前のバージョン-CommandServlet と user_password による SQLi 脆弱性（CVE-2021-22658）
999261	CVE-2021-22658	WEB-MISC Advantech iView 5.7.03.6112 より前-CommandServlet と user_name による SQLi 脆弱性 (CVE-2021-22658)
999262	CVE-2021-21983	WEB-MISC 8.4 より前の VMware vRealize Operations Manager-任意のファイル書き込みの脆弱性（CVE-2021-21983）
999263	CVE-2020-6754	5.2.4 より前の WEB-MISC dotCMS-アセットを介したディレクトリトラバーサルの脆弱性（CVE-2020-6754）
999264	CVE-2020-27128	WEB-MISC 20.3.1 より前の Cisco SD-WAN vManage-リモート処理による任意のファイル書き込みの脆弱性（CVE-2020-27128）
999265	CVE-2020-27128	WEB-MISC 20.3.1 より前の Cisco SD-WAN vManage-dr を介した任意のファイル書き込みの脆弱性（CVE-2020-27128）
999266	CVE-2020-15714	WEB-MISC RConfig 3.9.5 およびそれ以前-SQL インジェクションの脆弱性（CVE-2020-15714）

署名ルール	**CVE ID**	説明
999267	CVE-2020-15713	3.9.6 より前の WEB-MISC RConfig-SQL インジェクションの脆弱性（CVE-2020-15713）
999268	CVE-2020-14295	1.2.13 より前の WEB-MISC Cacti-SQL インジェクションの脆弱性（CVE-2020-14295）
999269	CVE-2020-13778	3.9.5 より前の WEB-MISC RConfig-ajaxEditTemplate.php を介したリモートコード実行の脆弱性（CVE-2020-13778）
999270	CVE-2020-13778	3.9.5 より前の WEB-MISC RConfig-ajaxAddTemplate.php を介したリモートコード実行の脆弱性（CVE-2020-13778）
999271	CVE-2020-13592	WEB-MISC Rukovoditel プロジェクト管理アプリ-selected_fields を介した SQL インジェクションの脆弱性 (CVE-2020-13592)
999272	CVE-2020-13592	WEB-MISC Rukovoditel プロジェクト管理アプリ-lists_id を介した SQL インジェクションの脆弱性 (CVE-2020-13592)
999273	CVE-2020-13591	WEB-MISC Rukovoditel プロジェクト管理アプリ-SQL インジェクションの脆弱性（CVE-2020-13591）
999274	CVE-2020-13550	WEB-MISC Advantech WebAccess/SCADA-ファイル名によるパストラバーサルの脆弱性（CVE-2020-13550）

2021 年 7 月のシグネチャアップデート

January 25, 2022

2021-07-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 66 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ (CVE) の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999231	CVE-2021-34074	ウェブ-MISC Artica Pandora FMS 7.54 まで-相対パスを介した任意のファイルアップロードの脆弱性（CVE-2021-34074）
999232	CVE-2021-32633	WEB-MISC Plone CMS-Zope ページテンプレートアップロードによるリモートコード実行の脆弱性 (CVE-2021-32633)
999233	CVE-2021-32633	WEB-MISC Plone CMS-Zope ページテンプレート新しい経由のリモートコード実行の脆弱性 (CVE-2021-32633)
999234	CVE-2021-31181	WEB-MISC Microsoft SharePoint Server-リモートコード実行の脆弱性 (CVE-2021-31181)

署名ルール	**CVE ID**	説明
999235	CVE-2021-24370	5.6.9 より前の WEB-WORDPRESS ファンシープロダクトデザイナープラグイン-fpd_custom_uplod_file（CVE-2021-24370）を介した RCE の脆弱性
999236	CVE-2021-24370	5.6.9 より前の WEB-WORDPRESS ファンシープロダクトデザイナープラグイン-custom-image-handler.php (CVE-2021-24370) を介して RCE の脆弱性
999237	CVE-2021-24354	WEB-WORDPRESS Simple 301 が 2.0.4 より前にプラグインをリダイレクトする-任意のプラグインインストールの脆弱性（CVE-2021-24354）
999238	CVE-2021-24352	WEB-WORDPRESS シンプル 301 は 2.0.4 より前のプラグインをリダイレクトする-エクスポートの脆弱性をリダイレクトする（CVE-2021-24352）
999239	CVE-2021-1497, CVE-2021-1498	WEB-MISC 4.0 (2e) より前の Cisco ハイパーフレックス HX-リモートコード実行の脆弱性（CVE-2021-1497、CVE-2021-1498）
999240	CVE-2020-21057	WEB-MISC FusionPBX 4.5.7-フォルダ削除機能によるパストラバーサルの脆弱性（CVE-2020-21057）
999241	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前のバージョン-BackupDatabase を介したパストラバーサルの脆弱性 (CVE-2020-16245)

署名ルール	**CVE ID**	説明
999242	CVE-2020-10148	WEB-MISC SolarWinds オリオン複数バージョン-認証バイパスの脆弱性（CVE-2020-10148）

2021 年 **8** 月のシグネチャアップデート

January 25, 2022

2021-08-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 67 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999183	CVE-2021-37557	WEB-MISC Centreon 複数のバージョン-SQL インジェクションの脆弱性（CVE-2021-3757）
999184	CVE-2021-35501	ウェブ-MISC Artica Pandora FMS 7.54 まで-ビジュアルコンソールが保存された XSS の脆弱性（CVE-2021-35501）

署名ルール	**CVE ID**	説明
999185	CVE-2021-35464	WEB-MISC ForgeRock アクセス管理と OpenAM-リモートコード実行の脆弱性（CVE-2021-35464）
999186	CVE-2021-34523	WEB-MISC Microsoft Exchange Server-権限昇格の脆弱性 (CVE-2021-34523)
999187	CVE-2021-34473	WEB-MISC Microsoft Exchange Server-クエリによるサーバー側の要求偽造認証バイパスの脆弱性 (CVE-2021-34473)
999188	CVE-2021-34473	WEB-MISC Microsoft Exchange Server-クッキーを介したサーバー側要求偽造認証バイパスの脆弱性 (CVE-2021-34473)
999189	CVE-2021-33203	WEB-MISC Django-絶対パスを介した TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)
999190	CVE-2021-33203	WEB-MISC Django-パストラバーサルによる TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)
999191	CVE-2021-33203	WEB-MISC Django-バックスラッシュによる TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)
999192	CVE-2021-33203	WEB-MISC Django-スラッシュによる TemplateDetailView ファイルの存在開示の脆弱性 (CVE-2021-33203)

署名ルール	**CVE ID**	説明
999193	CVE-2021-3287, CVE-2020-28653	WEB-MISC Zoho ManageEngine opManager 12.5.329 より前のバージョン-認証されていない RCE の脆弱性 (CVE-2021-3287、CVE-2020-28653)
999194	CVE-2021-32789	WEB-WORDPRESSWooCommerce プラグイン 5.5.0 まで-SQL インジェクションの脆弱性分類と rest_route（CVE-2021-32789）
999195	CVE-2021-32789	WEB-WORDPRESS WooCommerce プラグイン 5.5.0-タクソノミによる SQL インジェクションの脆弱性 (CVE-2021-32789)
999196	CVE-2021-32604	ウェブ-MISC SolarWinds Serv-U 15.2.3 より前-SenderEmail パラメータを介したクロスサイトスクリプティングの脆弱性 (CVE-2021-3260)
999197	CVE-2021-32093	WEB-MISC 国家安全保障局使者 5.9.0-任意のファイル読み取りの脆弱性 (CVE-2021-32093)
999198	CVE-2021-31760	1.974 より前の WEB-MISC Webmin-CSRF の脆弱性が run.cgi 経由で RCE につながる (CVE-2021-31760)
999199	CVE-2021-31207	WEB-MISC Microsoft Exchange Server-セキュリティ機能のバイパスの脆弱性（CVE-2021-31207）
999200	CVE-2021-31195	WEB-MISC Microsoft Exchange Server-リモートコード実行脆弱性 (CVE-2021-31195)

署名ルール	**CVE ID**	説明
999201	CVE-2021-28474	WEB-MISC Microsoft SharePoint Server-リモートコード実行の脆弱性 (CVE-2021-28474)
999202	CVE-2021-24385	WEB-WORDPRESS FileBird プラグイン 4.7.3-SelectedFolder パラメータと rest_route を介した SQL インジェクションの脆弱性 (CVE-2021-24385)
999203	CVE-2021-24385	WEB-WORDPRESS FileBird プラグイン 4.7.3-SelectedFolder パラメータを介した SQL インジェクションの脆弱性 (CVE-2021-24385)
999204	CVE-2021-24385	WEB-WORDPRESS FileBird プラグイン 4.7.3-JSON エンコードされたボディを介した SQL インジェクションの脆弱性 (CVE-2021-24385)
999205	CVE-2021-24356	WEB-WORDPRESS Simple 301 が 2.0.4 より前にプラグインをリダイレクトする-任意のプラグインアクティベーションの脆弱性 (CVE-2021-24356)
999206	CVE-2021-23024	WEB-MISC F5 BIG-IQ 複数のバージョン-リモートコード実行の脆弱性（CVE-2021-23024）
999207	CVE-2021-22911	WEB-MISC Rocket.Chat Server 3.11、3.12、3.13-ブラインド NOSQL インジェクションの脆弱性（CVE-2021-22911）
999208	CVE-2021-22900	WEB-MISC パルス接続 9.1R11.4 より前のセキュアな接続-smimeCert.cgi を介したリモートコード実行の脆弱性 (CVE-2021-22900)

署名ルール	**CVE ID**	説明
999209	CVE-2021-22900	WEB-MISC パルス接続 9.1R11.4 より前のセキュアな接続-admincert.cgi を介したリモートコード実行の脆弱性 (CVE-2021-22900)
999210	CVE-2021-22900	WEB-MISC パルス接続 9.1R11.4 より前のセキュアな接続-clientauthcert.cgi を介したリモートコード実行の脆弱性 (CVE-2021-22900)
999211	CVE-2021-22160	WEB-MISC Apache Pulsar-JSON ウェブトークン認証バイパスの脆弱性 (CVE-2021-22160)
999212	CVE-2021-21809	WEB-MISC Moodle-スペルチェッカプラグインと getSoggends メソッドによるリモートコード実行の脆弱性（CVE-2021-21809）
999213	CVE-2021-21809	WEB-MISC Moodle-スペルチェッカープラグインとチェックワードメソッドによるリモートコード実行の脆弱性（CVE-2021-21809）
999214	CVE-2021-21809	WEB-MISC Moodle-s__aspellpath (CVE-2021-21809）を介したリモートコード実行の脆弱性
999215	CVE-2021-21805	WEB-MISC Advantech R-seenet-認証されていないリモートコード実行の脆弱性 (CVE-2021-21805)
999216	CVE-2021-21804	WEB-MISC Advantech R-seenet-sub_opt を介したローカルファイル包含の脆弱性 (CVE-2021-21804)

署名ルール	**CVE ID**	説明
999217	CVE-2021-21587	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/os/listfiles によるパストラバーサルの脆弱性 (CVE-2021-21587)
999218	CVE-2021-21587	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/app/rsp/listfiles によるパストラバーサルの脆弱性 (CVE-2021-21587)
999219	CVE-2021-21586	3.3 より前の Web-MISC Dell Wyse 管理スイート-/イメージ/アプリケーションとファイル名によるパストラバーサルの脆弱性 (CVE-2021-21586)
999220	CVE-2021-21586	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/os およびファイル名によるパストラバーサルの脆弱性 (CVE-2021-21586)
999221	CVE-2021-21586	3.3 より前の Web-MISC Dell Wyse 管理スイート-/image/os および FilePath を介したパストラバーサルの脆弱性 (CVE-2021-21586)
999222	CVE-2020-25223	WEB-MISC Sophos SG UTM-SID および /var によるリモートコード実行 (CVE-2020-25223)
999223	CVE-2020-25223	WEB-MISC Sophos SG UTM-SID および /webadmin.plx 経由のリモートコード実行 (CVE-2020-25223)
999224	CVE-2020-21056	WEB-MISC FusionPBX 4.5.7-フォルダを介したパストラバーサルの脆弱性（CVE-2020-21056）

署名ルール	**CVE ID**	説明
999225	CVE-2020-21055	WEB-MISC FusionPBX 4.5.7-ファイル名の変更機能によるパストラバーサルの脆弱性 (CVE-2020-21055)
999226	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前-findSummaryUpdateDeviceListExpo のパストラバーサルの脆弱性 (CVE-2020-16245)
999227	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前のパストラバーサルの脆弱性 findCFGDeviceListExport (CVE-2020-16245)
999228	CVE-2020-14181	ウェブ-MISC アトラシアン Jira サーバー-ViewUserHover.jspa を介した情報開示の脆弱性 (CVE-2020-14181)
999229	CVE-2020-14005	WEB-MISC SolarWinds Orion 2020.2.1 HF 2-ExecuteVBScript によるリモートコード実行 Action Type (CVE-2020-14005)
999230	CVE-2020-14005	WEB-MISC SolarWinds Orion 2020.2.1 HF 2-ExecuteExternalProgram Action Type によるリモートコード実行 (CVE-2020-14005)

2021 年 **9** 月の署名更新

January 25, 2022

2021-09-11 週に特定された脆弱性について、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 68 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999163	CVE-2021-37556	WEB-MISC Centreon 複数バージョン-終了パラメータを介した SQL インジェクションの脆弱性 (CVE-2021-37556)
999164	CVE-2021-37556	WEB-MISC Centreon 複数バージョン-開始パラメータを介した SQL インジェクションの脆弱性 (CVE-2021-37556)
999165	CVE-2021-37353	WEB-MISC Nagios XI Docker ウィザード 1.1.3 より前-URI スキームなしのホストパラメータ経由の SSRF 脆弱性 (CVE-2021-37353)
999166	CVE-2021-37353	WEB-MISC Nagios XI Docker ウィザード 1.1.3 より前-URI スキームによるホストパラメータ経由の SSRF 脆弱性（CVE-2021-37353）
999167	CVE-2021-34638	3.1.25 より前の WEB-WORDPRESS のダウンロードマネージャプラグイン-ディレクトリトラバーサルの脆弱性 (CVE-2021-34638)
999168	CVE-2021-33766	WEB-MISC Microsoft Exchange Server-情報漏えいの脆弱性 (CVE-2021-33766)

署名ルール	**CVE ID**	説明
999169	CVE-2021-32682	2.1.59 より前の WEB-MISC elFinder-アーカイブ経由のコマンドインジェクションの脆弱性 (CVE-2021-32682)
999170	CVE-2021-26084	WEB-MISC Confluence サーバーとデータセンター-doenterpagevariables 経由の OGNL インジェクションの脆弱性 (CVE-2021-26084)
999171	CVE-2021-26084	WEB-MISC Confluence サーバーとデータセンター-作成ページ入力変数経由の OGNL インジェクションの脆弱性（CVE-2021-26084）
999172	CVE-2021-23394	2.1.59 より前の WEB-MISC elFinder-Phar Makefile 経由のリモートコード実行の脆弱性 (CVE-2021-23394)
999173	CVE-2021-23394	2.1.59 より前の WEB-MISC elFinder-Phar の名前変更によるリモートコード実行の脆弱性 (CVE-2021-23394)
999174	CVE-2021-23394	2.1.59 より前の WEB-MISC elFinder-Phar アップロードによるリモートコード実行の脆弱性 (CVE-2021-23394)
999175	CVE-2020-36289	WEB-MISC アトラシアン Jira サーバー-QueryComponentRendererValue による情報開示の脆弱性 (CVE-2020-36289)
999176	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前-FindSummaryCFGDeviceListExport によるパストラバーサルの脆弱性 (CVE-2020-16245)

署名ルール	**CVE ID**	説明
999177	CVE-2020-16245	WEB-MISC Advantech iView 5.7.03.6112 より前-FindUpdateDeviceListExport によるパストラバーサルの脆弱性 (CVE-2020-16245)
999178	CVE-2020-13774	WEB-MISC Ivanti エンドポイントマネージャ複数バージョン-EditLaunchPadDialog.aspx 経由の RCE の脆弱性 (CVE-2020-13774)
999179	CVE-2020-1147	WEB-MISC Microsoft SharePoint サーバー-カスタムページを介したリモートコード実行の脆弱性 (CVE-2020-1147)
999180	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server-quicklinksdialogform.aspx 経由でリモートでコードが実行される脆弱性 (CVE-2020-1147)
999181	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server-quicklinks.aspx 経由でリモートでコードが実行される脆弱性 (CVE-2020-1147)
999182	CVE-2020-11110	WEB-MISC Apache Grafana 6.7.1 まで-XSS 脆弱性 (CVE-2020-11110)
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 から 7.0.1-DOS の脆弱性につながる CSRF バイパス (CVE-2020-13379)

2021 年 10 月のシグネチャーアップデート

January 25, 2022

2021-10-09 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 69 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ (CVE) の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999149	CVE-2021-38312	WEB-WORDPRESS グーテンベルクテンプレートライブラリと 4.2.12 より前の Redux フレームワークプラグイン-REST_ROUTE の脆弱性（CVE-2021-38312）
999150	CVE-2021-38312	WEB-WORDPRESS グーテンベルクテンプレートライブラリと 4.2.12 より前の Redux フレームワークプラグイン-REST API の脆弱性（CVE-2021-38312）
999151	CVE-2021-34639	3.1.25 より前の WEB-WORDPRESS ダウンロードマネージャープラグイン-二重拡張機能のアップロードの脆弱性（CVE-2021-34639）

署名ルール	**CVE ID**	説明
999152	CVE-2021-34621	3.1.3 より前の WEB-WORDPRESS ProfilePress プラグイン-wp_capabilities による権限昇格の脆弱性（CVE-2021-34621）
999153	CVE-2021-32682	WEB-MISC elFinder 2.1.59 より前-名前の変更コマンドによるパストラバーサルの脆弱性（CVE-2021-32682）
999154	CVE-2021-32682	WEB-MISC elFinder 2.1.59 より前-中止コマンドによるパストラバーサルの脆弱性（CVE-2021-32682）
999155	CVE-2021-26086	WEB-MISC アトラシアン Jira サーバーとデータセンター-WEB-INF を介した情報開示の脆弱性 (CVE-2021-26086)
999156	CVE-2021-26086	WEB-MISC アトラシアン Jira サーバーとデータセンター-META-INF を介した情報開示の脆弱性 (CVE-2021-26086)
999157	CVE-2021-22005	WEB-MISC VMware vCenter-データアプリを介したファイルアップロードの脆弱性（CVE-2021-22005）
999158	CVE-2021-22005	WEB-MISC VMware vCenter-テレメトリステージログによるファイルアップロードの脆弱性（CVE-2021-22005）
999159	CVE-2021-22005	WEB-MISC VMware vCenter-テレメトリ製品ログによるファイルアップロードの脆弱性（CVE-2021-22005）

署名ルール	**CVE ID**	説明
999160	CVE-2021-20081	WEB-MISC 11.2.0.5 より前の Zoho ManageEngine サービスデスク-リモートでコードが実行される脆弱性（CVE-2021-20081）
999161	CVE-2020-29453	WEB-MISC アトラシアン Jira サーバーとデータセンター-WEB INF を介した情報開示の脆弱性 (CVE-2020-29453)
999162	CVE-2020-29453	WEB-MISC アトラシアン Jira サーバーとデータセンター-META-INF を介した情報開示の脆弱性 (CVE-2020-29453)

2021 年 **10** 月のシグネチャーアップデート

January 25, 2022

2021-10-26 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 70 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999127	CVE-2021-42013	WEB-MISC Apache HTTP サーバー 2.4.49 および 2.4.50-%%32 経由でのパストラバーサルの脆弱性 (CVE-2021-42013)
999128	CVE-2021-42013	WEB-MISC Apache HTTP サーバー 2.4.49 および 2.4.50-% 2% 経由でのパストラバーサルの脆弱性（CVE-2021-42013）
999129	CVE-2021-41773	WEB-MISC Apache HTTP サーバ 2.4.49-%2e%2e を介したパストラバーサルの脆弱性 (CVE-2021-41773)
999130	CVE-2021-41773	WEB-MISC Apache HTTP サーバー 2.4.49-.% 2e を介したパストラバーサルの脆弱性 (CVE-2021-41773)
999131	CVE-2021-40539	WEB-MISC Zoho ManageEngine AdSelfService Plus 6.1 ビルド 6114 より前-認証バイパスの脆弱性 (CVE-2021-40539)
999132	CVE-2021-34648	WEB-WORDPRESS 忍者フォームプラグイン 3.5.7 まで-送信メールアクションによる REST_ROUTE 脆弱性 (CVE-2021-34648)
999133	CVE-2021-34648	WEB-WORDPRESS 忍者フォームプラグイン最大 3.5.7-送信メールアクションによる REST API 脆弱性 (CVE-2021-34648)
999134	CVE-2021-34647	WEB-WORDPRESS 忍者フォームプラグイン 3.5.7 まで-提出物のエクスポートによる REST_ROUTE 脆弱性（CVE-2021-34647）

署名ルール	**CVE ID**	説明
999135	CVE-2021-34647	WEB-WORDPRESS 忍者フォームプラグイン 3.5.7 まで-提出物のエクスポートによる REST API の脆弱性（CVE-2021-34647）
999136	CVE-2021-34623	3.1.4 より前の WEB-WORDPRESS ProfilePress プラグイン-eup_cover_image を介した任意のファイルアップロードの脆弱性（CVE-2021-34623）
999137	CVE-2021-34623	3.1.4 より前の WEB-WORDPRESS ProfilePress プラグイン-eup_avatar を介した任意のファイルアップロードの脆弱性（CVE-2021-34623）
999138	CVE-2021-2400	WEB-MISC Oracle ル BI パブリッシャー-モバイル X 経由の SAXParser XXE 脆弱性 ReportTemplateService (CVE-2021-2400)
999139	CVE-2021-2400	WEB-MISC Oracle ル BI パブリッシャー-SaxParser XXE モバイルレポートテンプレートサービス経由の脆弱性 (CVE-2021-2400)
999140	CVE-2021-2400	WEB-MISC Oracle BI パブリッシャー-xmlpservice X ReportTemplateService 経由の SAXParser XXE 脆弱性 (CVE-2021-2400)
999141	CVE-2021-2400	WEB-MISC Oracle BI パブリッシャー-xmlpService 経由の SAXParser XXE 脆弱性 ReportTemplateService (CVE-2021-2400)

署名ルール	**CVE ID**	説明
999142	CVE-2021-21985	WEB-MISC VMware vCenter-vSAN ヘルスチェックプラグインのリモートコード実行の脆弱性（CVE-2021-21985）
999143	CVE-2021-20078	WEB-MISC Zoho ManageEngine opManager 12.5 ビルド 125362 より前-パストラバーサルの脆弱性（CVE-2021-20078）
999144	CVE-2020-29448	WEB-MISC アトラシアン Confluence サーバーとデータセンター-WEB INF を介した情報開示の脆弱性 (CVE-2020-29448)
999145	CVE-2020-29448	WEB-MISC アトラシアン Confluence サーバーとデータセンター-META-INF を介した情報開示の脆弱性 (CVE-2020-29448)
999146	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3-osupdate エンドポイントを介した認証されていない SQL インジェクションの脆弱性（CVE-2020-12442）
999147	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3-壁エンドポイントを介した認証されていない SQL インジェクションの脆弱性（CVE-2020-12442）
999148		9.1.1 より前の WEB-WORDPRESS バディプレスプラグイン-bp-メンバー招待状を介した SQL インジェクションの脆弱性機能

2021 年 11 月の署名アップデート

January 25, 2022

2021-11-18 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 71 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ (CVE) の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999098	CVE-2021-41765	WEB-MISC リビジョン 18274 より前の ResourceSpace 9.5 および 9.6-SQL インジェクションの脆弱性（CVE-2021-41765）
999099	CVE-2021-41288	WEB-MISC Zoho マネージエンジン運用マネージャー 125467 ビルド前-GetReportData API による SQL インジェクションの脆弱性（CVE-2021-41288）
999100	CVE-2021-40493	WEB-MISC Zoho マネージエンジン運用マネージャー 125437 ビルド前-デバイス名による SQL インジェクションの脆弱性（CVE-2021-40493）

署名ルール	**CVE ID**	説明
999101	CVE-2021-40493	WEB-MISC Zoho ビルド前 125437 の管理エンジン OpManager-ポーリングオブジェクトによる SQL インジェクションの脆弱性（CVE-2021-40493）
999102	CVE-2021-40438	WEB-MISC Apache HTTP サーバー-mod_proxy リクエスト転送の脆弱性（CVE-2021-40438）
999103	CVE-2021-39341	WEB-WORDPRESS OptinMonster プラグイン 2.6.4 まで-REST_ROUTE 権限バイパスの脆弱性（CVE-2021-39341）
999104	CVE-2021-39341	WEB-WORDPRESS OptinMonster プラグイン 2.6.4 まで-REST API パーミッションバイパスの脆弱性 (CVE-2021-39341)
999105	CVE-2021-37344	WEB-MISC Nagios XI スイッチウィザード 2.5.7 より前-ip_address パラメータによるリモートコード実行の脆弱性 (CVE-2021-37344)
999106	CVE-2021-35218	WEB-MISC SolarWinds オリオン 2020.2.6 より前-Chart.ashx によるデシリアライゼーションの脆弱性（CVE-2021-35218）
999107	CVE-2021-35215	WEB-MISC SolarWinds Orion プラットフォーム 2020.2.6 より前-レポートによるリモートコード実行の脆弱性（CVE-2021-35215）
999108	CVE-2021-35215	WEB-MISC SolarWinds Orion プラットフォーム 2020.2.6 より前-アラートによるリモートコード実行の脆弱性（CVE-2021-35215）

署名ルール	**CVE ID**	説明
999109	CVE-2021-24889	3.6.4 より前の WEB-WORDPRESS 忍者フォームプラグイン-SQL インジェクションの脆弱性（CVE-2021-24889）
999110	CVE-2021-24381	WEB-WORDPRESS 忍者フォームプラグイン 3.5.8.2 より前-カスタムクラス名が格納されたクロスサイトスクリプティングの脆弱性 (CVE-2021-24381)
999111	CVE-2021-2401	WEB-MISC Oracle ル BI パブリッシャー-DOMParser XXE モバイル X レポートテンプレートサービス経由の脆弱性 (CVE-2021-2401)
999112	CVE-2021-2401	WEB-MISC Oracle ル BI パブリッシャー-DOMParser XXE モバイルレポートテンプレートサービス経由の脆弱性 (CVE-2021-2401)
999113	CVE-2021-2401	WEB-MISC Oracle BI パブリッシャー-xmlpservice X ReportTemplateService 経由の DOMParser XXE 脆弱性 (CVE-2021-2401)
999114	CVE-2021-2401	WEB-MISC Oracle BI パブリッシャー-xmlpservice 経由の DOMParser XXE 脆弱性 ReportTemplateService (CVE-2021-2401)
999115	CVE-2021-2392	WEB-MISC Oracle BI パブリッシャー-任意ファイルのアップロードの脆弱性（CVE-2021-2392）
999116	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase 分析プロバイダ・サービス-Essbase 経由のリモート・コード実行の脆弱性 (CVE-2021-2244)

署名ルール	**CVE ID**	説明
999117	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase 分析プロバイダ・サービス-管理者経由のリモート・コード実行の脆弱性 (CVE-2021-2244)
999118	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase アナリティック・プロバイダ・サービス-JAPI 経由のリモート・コード実行の脆弱性 (CVE-2021-2244)
999119	CVE-2021-22205	WEB-MISC GitLab CE/EE-悪意を持って作成された JPEG/TIFF ファイルによるリモートコード実行の脆弱性（CVE-2021-22205）
999120	CVE-2021-22017	WEB-MISC VMware vCenter-rhhtproxy を介したパストラバーサルの脆弱性 (CVE-2021-22017)
999121	CVE-2021-20837	WEB-MISC r.5003 より前の可動タイプ-mt.handler_to_coderef を介したリモートコード実行（CVE-2021-20837）
999122	CVE-2021-20131	WEB-MISC Zoho ManageEngine AdManager ビルド 7115 より前-ファイルのアップロードによるリモートでのコード実行の脆弱性（CVE-2021-20131）
999123	CVE-2021-20130	WEB-MISC Zoho ManageEngine AdManager ビルド 7115 より前-ファイルのアップロードによるリモートでのコード実行の脆弱性（CVE-2021-20130）

署名ルール	**CVE ID**	説明
999124	CVE-2021-20034	WEB-MISC SonicWall セキュアモバイルアクセス-パストラバーサルの脆弱性（CVE-2021-20034）
999125		9.1.1 より前の WEB-WORDPRESS のバディプレスプラグイン-サインアップ REST API と rest_route による情報開示の脆弱性
999126		9.1.1 より前の WEB-WORDPRESS BuddyPress プラグイン-サインアップ REST API による情報開示の脆弱性

2021 年 **12** 月のシグネチャーアップデート

January 25, 2022

2021 年 12 月 11 日に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 72 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999077	CVE-2021-44228	WEB-MISC Apache Log4j：フォーム経由でリモートでコードが実行される脆弱性 (CVE-2021-44228)
999078	CVE-2021-44228	WEB-MISC Apache Log4j：本文を介してリモートでコードが実行される脆弱性（CVE-2021-44228）
999079	CVE-2021-44228	WEB-MISC Apache Log4j：ヘッダーを介したリモートコード実行の脆弱性（CVE-2021-44228）
999080	CVE-2021-44228	WEB-MISC Apache Log4j：URL経由でリモートでコードが実行される脆弱性（CVE-2021-44228）
999081	CVE-2021-42847	WEB-MISC Zoho ManageEngine ADAudit Plus 7006 より前：認証されていない任意のファイル書き込みの脆弱性 (CVE-2021-42847)
999082	CVE-2021-42321	WEB-MISC Microsoft Exchange Server-リモートでコードが実行される脆弱性（CVE-2021-42321）
999083	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2021-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）
999084	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2020-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）

署名ルール	**CVE ID**	説明
999085	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2019-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）
999086	CVE-2021-42258	WEB-MISC BQE BillQuick Web スイート 2018-txTID を介した認証されていない SQL インジェクションの脆弱性（CVE-2021-42258）
999087	CVE-2021-42237	WEB-MISC サイトコア 7.5.0 から 8.2.7 へ：リモートでコードが実行される脆弱性（CVE-2021-42237）
999088	CVE-2021-41950	リビジョン 18277 より前の WEB-MISC ResourceSpace 9.6：バリアントを介した非認証パストラバーサルの脆弱性（CVE-2021-41950）
999089	CVE-2021-41950	リビジョン 18277 より前の WEB-MISC ResourceSpace 9.6：プロバイダーを介した非認証パストラバーサルの脆弱性（CVE-2021-41950）
999090	CVE-2021-41349	WEB-MISC Microsoft Exchange Server-クロスサイトスクリプティングの脆弱性 (CVE-2021-41349)
999091	CVE-2021-35217	2020.2.6 HF1 より前の WEB-MISC SolarWinds オリオン-WSAsyncExecuteTasks.aspx を介したデシリアライゼーションの脆弱性（CVE-2021-35217）

署名ルール	**CVE ID**	説明
999092	CVE-2021-34416	WEB-MISC Zoom ミーティングコネクタ 4.6.360.20210325：リモートでコードが実行される脆弱性（CVE-2021-34416）
999093	CVE-2021-22941	WEB-MISC 5.11.20 より前の Citrix ShareFile ストレージ：不適切なアクセス制御の脆弱性（CVE-2021-22941）
999094	CVE-2020-35136	12.0.4 より前の WEB-MISC Dolibarr: zipfilename_template および bz を介したリモートコード実行の脆弱性（CVE-2020-35136）
999095	CVE-2020-35136	12.0.4 より前の WEB-MISC Dolibarr: zipfilename_template および gz を介したリモートコード実行の脆弱性（CVE-2020-35136）
999096	CVE-2020-2950, CVE-2021-2456	ウェブその他 Oracle BI Publisher -任意のファイルアップロードの脆弱性（CVE-2020-2950、CVE-2021-2456）
999097	CVE-2020-2950, CVE-2021-2456	ウェブその他 Oracle BI Publisher -リモートでコードが実行される脆弱性（CVE-2020-2950、CVE-2021-2456）

2021 年 **12** 月のシグネチャーアップデート

January 25, 2022

2021 年 12 月 13 日に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 73 は、Citrix Application Delivery Controller（ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください 。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、更新されたシグニチャルール、CVE ID、およびその説明の一覧を示します。

> 注:
>
> 以下のシグニチャルール（999077、999078、999079、999080）は、両方の CVE（CVE-2021-44228 および CVE-2021-45046）に対応しています。

署名ルール	**CVE ID**	説明
999077	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：フォームを介したリモートコード実行の脆弱性（CVE-2021-44228、CVE-2021-45046）
999078	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：本文を介してリモートでコードが実行される脆弱性（CVE-2021-44228、CVE-2021-45046）
999079	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：ヘッダーを介したリモートコード実行の脆弱性（CVE-2021-44228、CVE-2021-45046）
999080	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j：URL を介したリモートコード実行の脆弱性（CVE-2021-44228、CVE-2021-45046）

2021 年 12 月のシグネチャーアップデート

January 25, 2022

2021-12-21 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名は、Citrix Application Delivery Controller （ADC）11.1、12.0、12.1、13.0、13.1 の次のソフトウェアバージョンと互換性があります。

Citrix ADC バージョン 12.0 はサポート終了（EOL）に達しました。詳細については、リリースライフサイクルのページを参照してください。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、シグニチャルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999073	CVE-2021-44077	WEB-MISC Zoho ManageEngine サービスデスクプラス 11306 より前-インポート技術者による事前認証 RCE の脆弱性（CVE-2021-44077）
999074	CVE-2021-43798	WEB-MISC Apache Grafana 8.0.0 8.3.0 まで：パストラバーサルの脆弱性（CVE-2021-43798）
999075	CVE-2021-35216	2020.2.6 より前の WEB-MISC SolarWinds オリオン: EditTopXX.aspx を介したデシリアライゼーションの脆弱性（CVE-2021-35216）

署名ルール	**CVE ID**	説明
999076	CVE-2021-34993	WEB-MISC Commvault CommCell-CvSearchService 認証バイパスの脆弱性 (CVE-2021-34993)

2022 年 **1** 月の署名更新

January 31, 2022

2022-01-20 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 75 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999055	CVE-2021-44224	WEB-MISC Apache HTTP サーバー-フォワードプロキシとリバースプロキシを介した不正な形式の UDS の脆弱性 (CVE-2021-44224)
999056	CVE-2021-43815	WEB-MISC Apache Grafana-TestData DB データソースパストラバーサルの脆弱性 (CVE-2021-43815)

署名ルール	**CVE ID**	説明
999057	CVE-2021-43813	WEB-MISC Apache Grafana：マークダウンによるパストラバーサルの脆弱性（CVE-2021-43813）
999058	CVE-2021-43405	4.5.30 より前の WEB-MISC FusionPBX-fax_extension を介した OS コマンドインジェクション（CVE-2021-43405）
999059	CVE-2021-42392	2.0.206 より前の WEB-MISC H2 コンソール：リモートでコードが実行される脆弱性（CVE-2021-42392）
999060	CVE-2021-42362	5.3.3 より前の WEB-WORDPRESS ポピュラーな投稿プラグイン：任意のファイルアップロードの脆弱性（CVE-2021-42362）
999061	CVE-2021-42129	6.3.3 より前の WEB-MISC Ivanti Avalanche：TxtuPass を介した OS コマンドインジェクションの脆弱性（CVE-2021-42129）
999062	CVE-2021-42129	6.3.3 より前の WEB-MISC Ivanti Avalanche：TxtUname を介した OS コマンドインジェクションの脆弱性（CVE-2021-42129）
999063	CVE-2021-42129	6.3.3 より前の WEB-MISC Ivanti Avalanche：TxTuncPath を介した OS コマンドインジェクションの脆弱性（CVE-2021-42129）
999064	CVE-2021-40345	5.8.6 より前の WEB-MISC Nagios XI：悪意を持って細工された ZIP ファイルを介した OS コマンドインジェクションの脆弱性（CVE-2021-40345）

署名ルール	**CVE ID**	説明
999065	CVE-2021-37928	WEB-MISC Zoho ManageEngine アドマネージャープラス 7110 より前-無制限のファイルアップロードの脆弱性 (CVE-2021-37928)
999066	CVE-2021-25037	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-オブジェクト REST API と rest_route を介した SQL インジェクションの脆弱性
999067	CVE-2021-25037	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-オブジェクト REST API を介した SQL インジェクションの脆弱性
999068	CVE-2021-25036	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-REST API と rest_route を介した権限昇格の脆弱性
999069	CVE-2021-25036	4.1.5.3 より前の WEB-WORDPRESS All In One SEO Plugin-REST API を介した権限昇格の脆弱性
999070	CVE-2021-21917	2.4.17 より前の WEB-MISC アドバンテック R-SeeNet-ord 経由の SQL インジェクションの脆弱性 (CVE-2021-21917)
999071	CVE-2021-20040	WEB-MISC SonicWall セキュアモバイルアクセス-任意のファイル書き込みの脆弱性 (CVE-2021-20040)

署名ルール	**CVE ID**	説明
999072	CVE-2021-20039	WEB-MISC SonicWall セキュアモバイルアクセス-コマンドインジェクションの脆弱性（CVE-2021-20039）

2022 年 **2** 月のシグネチャアップデート

February 21, 2022

2022-02-20 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 76 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999047	CVE-2022-23863	4.5.30 より前の WEB-MISC FusionPBX-fax_page_size を介した OS コマンドインジェクション (CVE-2021-43406)
999048	CVE-2021-44515	WEB-MISC JetBrains TeamCity: エージェントプッシュによるリモートコード実行の脆弱性（CVE-2021-43193）

署名ルール	**CVE ID**	説明
999049	CVE-2021-43406	5.1.5 より前の WEB-MISC ゴーアヘッド：CGI 環境変数インジェクションの脆弱性 (CVE-2021-42342)
999050	CVE-2021-43193	WEB-MISC SonicWall セキュアモバイルアクセス-リモートでコードが実行される脆弱性 (CVE-2021-20045)
999051	CVE-2021-42342	5.1.5 より前の WEB-MISC ゴーアヘッド：CGI 環境変数インジェクションの脆弱性 (CVE-2021-42342)
999052	CVE-2021-20045	WEB-MISC SonicWall セキュアモバイルアクセス-リモートでコードが実行される脆弱性 (CVE-2021-20045)
999053	CVE-2021-20044	WEB-MISC SonicWall セキュアモバイルアクセス-コマンドインジェクションの脆弱性 (CVE-2021-20044)
999054		Web-WORDPRESS AdSanity プラグイン-HTML5 ファイルアップロードによるリモートコード実行の脆弱性

2022 年 **2** 月のシグネチャアップデート

March 8, 2022

2022-02-25 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 77 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999034		WEB-WORDPRESS WordPress 5.9-Json オブジェクトのページの抜粋を介して格納された XSS の脆弱性
999035		WEB-WORDPRESS WordPress 5.9-フォーム内のページの抜粋を介して保存された XSS の脆弱性
999036		WEB-WORDPRESS WordPress 5.9-post.php 経由で保存された XSS の脆弱性
999037		WEB-WORDPRESS WordPress 5.9-Json オブジェクトにポスト抜粋を介して格納された XSS の脆弱性
999038		WEB-WORDPRESS WordPress 5.9-フォーム内のポストの抜粋を介して保存された XSS の脆弱性
999039		フォームフィールド値による WEB-MISC パストラバーサルの脆弱性
999040		URI 経由の WEB-MISC パストラバーサルの脆弱性
999041	CVE-2022-23221	2.1.210 より前の WEB-MISC H2 コンソール：test.do を介したリモートコード実行の脆弱性 (CVE-2022-23221)

署名ルール	**CVE ID**	説明
999042	CVE-2022-23221	2.1.210 より前の WEB-MISC H2 コンソール：login.do を介したリモートコード実行の脆弱性 (CVE-2022-23221)
999043	CVE-2022-21662	5.8.3 より前の WEB-WORDPRESS WordPress: ストアドクロスサイトスクリプティングの脆弱性 (CVE-2022-21662)
999044	CVE-2022-0320	WEB-WORDPRESS 5.0.5 より前の Elementor プラグインに不可欠なアドオン-LFI eael_product_gallery (CVE-2022-0320)
999045	CVE-2022-0320	WEB-WORDPRESS 5.0.5 より前の Elementor プラグインに必須のアドオン-woo_product_pagination_product 経由の LFI (CVE-2022-0320)
999046	CVE-2022-0320	WEB-WORDPRESS 5.0.5 より前の Elementor プラグインに必須のアドオン-load_more 経由の LFI (CVE-2022-0320)

2022 年 **3** 月のシグネチャ更新

April 7, 2022

2022-03-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。セキュリティに脆弱な攻撃からアプライアンスを保護するために、これらのシグニチャルールをダウンロードして設定できます。

署名バージョン

署名バージョン 78 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999006		WEB-MISC Zabbix 複数バージョン-items.php を介したリモートコード実行の脆弱性
999007	CVE-2022-24266	WEB-MISC Cuppa CMS v1.0-order_orientation 経由の SQL インジェクションの脆弱性 (CVE-2022-24266)
999008	CVE-2022-24266	WEB-MISC Cuppa CMS v1.0-order_by 経由の SQL インジェクションの脆弱性 (CVE-2022-24266)
999009	CVE-2022-22005	WEB-MISC Microsoft SharePoint-信頼できないデータのデシリアライゼーションによる RCE の脆弱性 (CVE-2022-22005)
999010	CVE-2022-21705	WEB-MISC OctoberCMS ビルド 474 および v1.1.10 より前-リモートでコードが実行される脆弱性 (CVE-2022-21705)
999011	CVE-2022-0557	1.2.11 より前の WEB-MISC マイクロウェーバー-リモートでコードが実行される脆弱性 (CVE-2022-0557)
999012	CVE-2022-0513	13.1.5 より前の WEB-WORDPRESS WP 統計プラグイン-ブラインド SQL インジェクションの脆弱性 (CVE-2022-0513)

署名ルール	**CVE ID**	説明
999013	CVE-2022-0332	WEB-MISC ムードル 3.11.0 から 3.11.4-H5P アクティビティ SQL インジェクションの脆弱性 (CVE-2022-0332)
999014	CVE-2021-46088	WEB-MISC Zabbix 複数バージョン-リモートでコードが実行される脆弱性（CVE-2021-46088）
999015	CVE-2021-43789	WEB-MISC PrestaShop 1.7.8.2 より前-並べ替え順序による SQL インジェクションの脆弱性 (CVE-2021-43789)
999016	CVE-2021-43789	WEB-MISC PrestaShop 1.7.8.2 より前-orderBy による SQL インジェクションの脆弱性 (CVE-2021-43789)
999017	CVE-2021-43408	WEB-WORDPRESS 1.1.9 より前の重複投稿プラグイン-SQL インジェクションの脆弱性 (CVE-2021-43408)
999018	CVE-2021-43319	WEB-MISC Zoho ManageEngine NCM 125488 より前-OS コマンドインジェクションの脆弱性（CVE-2021-43319）
999019	CVE-2021-41282	WEB-MISC pfSense 2.5.2：リモートでコードが実行される脆弱性 (CVE-2021-41282)
999020	CVE-2021-39115, CVE-2021-43947	WEB-MISC アトラシアン Jira サーバーおよびデータセンター-サーバー側テンプレートインジェクションの脆弱性（CVE-2021-39115、CVE-2021-43947）
999021	CVE-2021-38452	WEB-MISC Moxa MxView 3.2.2 より前のネットワーク管理-パストラバーサルの脆弱性 (CVE-2021-38452)

署名ルール	**CVE ID**	説明
999022	CVE-2021-37918	WEB-MISC Zoho ManageEngine AdManager Plus 7111 より前-ドメイン名を介したパストラバーサルの脆弱性 (CVE-2021-37918)
999023	CVE-2021-37918	Web-MISC Zoho ManageEngine AdManager Plus 7111 より前-BM_OperationID を介したパストラバーサルの脆弱性 (CVE-2021-37918)
999024	CVE-2021-37918	Web-MISC Zoho ManageEngine AdManager Plus 7111 より前-任意のファイルアップロードによる RCE の脆弱性 (CVE-2021-37918)
999025	CVE-2021-32649	WEB-MISC OctoberCMS ビルド 473 および v1.1.6 より前-Twig を介したリモートコード実行の脆弱性（CVE-2021-32649）
999026	CVE-2021-32648	WEB-MISC OctoberCMS ビルド 472 および v1.1.5 より前のバージョン-パスワードリセットの脆弱性 (CVE-2021-32648)
999027	CVE-2021-32099, CVE-2020-26518	WEB-MISC 記事 743 より前のパンドラ-chart_generator を介した SQL インジェクションの脆弱性 (CVE-2021-32099、CVE-2020-26518)
999028	CVE-2021-32098	WEB-MISC 記事 743 より前のパンドラ-progressbubble を介した Phar デシリアライゼーションの脆弱性 (CVE-2021-32098)

署名ルール	**CVE ID**	説明
999029	CVE-2021-32098	WEB-MISC 記事 743 より前のパンドラ-プログレスバー経由の Phar デシリアライゼーションの脆弱性（CVE-2021-32098）
999030	CVE-2021-30149	WEB-MISC コンポーザ 10.0.36: リモートでコードが実行される脆弱性（CVE-2021-30149）
999031	CVE-2021-25114	WEB-WORDPRESS 有料会員プロプラグイン 2.6.7 より前-rest_route と discount_code による SQLi の脆弱性 (CVE-2021-25114)
999032	CVE-2021-25114	WEB-WORDPRESS 有料会員プロプラグイン 2.6.7 より前-wp-json と discount_code 経由の SQLi の脆弱性 (CVE-2021-25114)
999033	CVE-2021-21984	WEB-MISC VMware vRealize ビジネス 7.6.0 より前のクラウド 7.x-リモートでコードが実行される脆弱性（CVE-2021-21984）

2022 年 **3** 月のシグネチャ更新

April 7, 2022

2022-03-29 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。セキュリティに脆弱な攻撃からアプライアンスを保護するために、これらのシグニチャルールをダウンロードして設定できます。

署名バージョン

署名バージョン 79 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
18959 (更新されたルール)	CVE-2022-22965	WEB-MISC VMware Spring4Shell、SpringSource Spring Framework クラス.classloader RCE 試行
999005	CVE-2022-22963	WEB-MISC Spring クラウド関数-コードインジェクションの脆弱性（CVE-2022-22963）

2022 年 **4** 月のシグネチャ更新

April 25, 2022

2022-04-04 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 80 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文とレスポンス本文署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999004	CVE-2022-22965	WEB-MISC Spring4Shell スプリングコアフレームワーク-RCE 脆弱性（CVE-2022-22965）

2022 年 4 月のシグネチャ更新

April 25, 2022

2022-04-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 81 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
999001	CVE-2022-0479	4.1.1 より前の WEB-WORDPRESS ポップアップビルダープラグイン-SQL インジェクションの脆弱性（CVE-2022-0479）
999002	CVE-2021-36393	WEB-MISC Moodle 3.11.1 より前-SQL インジェクションの脆弱性（CVE-2021-36393）
999003	CVE-2021-26599	WEB-MISC ImpressCMS 1.4.3 より前-SQL インジェクションの脆弱性（CVE-2021-26599）

2022 年 4 月のシグネチャ更新

May 9, 2022

2022-04-23 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロー

ドして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 82 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998997	CVE-2022-27924	WEB-MISC Zimbra コラボレーションジュール-キャッシュポイズニングの脆弱性（CVE-2022-27924）
998998	CVE-2022-21907	WEB-MISC Microsoft HTTP プロトコルスタック-リモートでコードが実行される脆弱性（CVE-2022-21907）
998999	CVE-2021-37930	WEB-MISC 7111 より前の ManageEngine adManager プラス-SM_DomainName を介した任意のファイルアップロードの脆弱性（CVE-2021-37930）
999000	CVE-2021-37930	WEB-MISC 7111 より前の ManageEngine adManager プラス-SM_OperationID を介した任意のファイルアップロードの脆弱性（CVE-2021-37930）

2022 年 **5** 月の署名更新

June 1, 2022

2022-05-04 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 83 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998993	CVE-2022-29464	WEB-MISC WSO2 複数製品-無制限のファイルアップロードの脆弱性（CVE-2022-29464）
998994	CVE-2022-22954	WEB-MISC VMware Workspace ONE Access および ID マネージャー-デバイスタイプを介したリモートコード実行の脆弱性（CVE-2022-22954）
998995	CVE-2022-22954	WEB-MISC VMware Workspace ONE アクセスおよびアイデンティティマネージャー-deviceUDID を介したリモートコード実行の脆弱性（CVE-2022-22954）
998996	CVE-2022-1329	WEB-WORDPRESS ワードプレスエレメンター 3.6.3 より前のウェブサイトビルダー-不正な AJAX アクションの脆弱性（CVE-2022-1329）

2022 年 5 月の署名更新

June 1, 2022

2022-05-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 84 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ (CVE) の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998988	CVE-2022-26986	WEB-MISC 1.4.3 より前の ImpressCMS-mimetypeid を介した SQL インジェクションの脆弱性（CVE-2022-26986）
998989	CVE-2022-24112	WEB-MISC Apache APIIX バッチリクエストプラグイン-IP 制限バイパスの脆弱性（CVE-2022-24112）
998990	CVE-2021-37558	20.04.14、20.10.8、および 21.04.2 より前の WEB-MISC Centreon-service_description を介した SQL インジェクションの脆弱性（CVE-2021-37558）
998991	CVE-2021-37558	20.04.14、20.10.8、21.04.2 より前の WEB-MISC Centreon-ホスト名を介した SQL インジェクションの脆弱性（CVE-2021-37558）

署名ルール	**CVE ID**	説明
998992	CVE-2021-22056	WEB-MISC VMware Workspace ONE Access および ID マネージャー-サーバー側リクエストフォージェリの脆弱性（CVE-2021-22056）

2022 年 **5** 月の署名更新

June 1, 2022

2022-05-13 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 85 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998982	CVE-2022-26352	WEB-MISC dotCMS-PUT を介した任意のファイルアップロードの脆弱性（CVE-2022-26352）
998983	CVE-2022-26352	WEB-MISC dotCMS-POST を介した任意のファイルアップロードの脆弱性（CVE-2022-26352）
998984	CVE-2022-1388	WEB-MISC F5 BIG-IP-iControl REST 認証バイパスの脆弱性（CVE-2022-1388）

署名ルール	**CVE ID**	説明
998985	CVE-2022-1162	WEB-MISC Gitlab CE/EE 複数バージョン-ハードコードされた資格情報の脆弱性（CVE-2022-1162）
998986	CVE-2022-0888	WEB-WORDPRESS プラグイン忍者フォームファイルアップロード 3.3.1 以前-任意のファイルアップロードの脆弱性（CVE-2022-0888）
998987	CVE-2021-35244	WEB-MISC 2020.2.6 より前の SolarWinds Orion HF3-WriteToFile アクションによる任意のファイルアップロードの脆弱性（CVE-2021-35244）

2022 年 **5** 月の署名更新

June 1, 2022

2022-05-20 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 86 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998980	CVE-2022-30525	WEB-MISC Zyxel ファイアウォール複数バージョン-SetwanPortST における認証されていない OS コマンドインジェクションの脆弱性 (CVE-2022-30525)
998981	CVE-2021-25094	WEB-WORDPRESS プラグイン 3.3.12 より前のタツビルダー-リモートでコードが実行される脆弱性 (CVE-2021-25094)

2022 年 **6** 月のシグネチャーアップデート

June 24, 2022

2022-06-07 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

シグネチャバージョン 87 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998964	CVE-2022-30525	WEB-MISC Zyxel ファイアウォール複数バージョン-SetwanPortST における認証されていない OS コマンドインジェクションの脆弱性 (CVE-2022-30525)
998965	CVE-2022-29108	WEB-MISC Microsoft SharePoint-信頼できないデータのデシリアライズによる RCE の脆弱性（CVE-2022-29108）
998966	CVE-2022-26134	WEB-MISC アトラシアン Confluence 複数バージョン-認証されていない OGNL インジェクションの脆弱性（CVE-2022-26134）
998967	CVE-2022-26019	WEB-MISC pfSense CE < 2.6.0-services_ntpd_gps.php および gpsport を介したリモートコード実行の脆弱性 (CVE-2022-26019)
998968	CVE-2022-26019	WEB-MISC pfSense CE < 2.6.0-services_ntpd.php および gpsport を介したリモートコード実行の脆弱性 (CVE-2022-26019)
998969	CVE-2022-24288	WEB-MISC Apache エアフロー 2.2.3 まで-DAG 例 my_param を介したリモートコード実行の脆弱性（CVE-2022-24288）
998970	CVE-2022-24288	WEB-MISC Apache エアフロー 2.2.3 まで-DAG 例 foo または miff を介したリモートコード実行の脆弱性（CVE-2022-24288）

署名ルール	**CVE ID**	説明
998971	CVE-2022-22978	WEB-MISC スプリングセキュリティ 5.5.6 および 5.6.3 まで-RegexRequestMatcher ラインフィードによる脆弱性のバイパス（CVE-2022-22978）
998972	CVE-2022-22978	WEB-MISC スプリングセキュリティ 5.5.6 および 5.6.3 まで-RegexRequestMatcher キャリッジリターンによる脆弱性のバイパス（CVE-2022-22978）
998973	CVE-2022-22957	WEB-MISC VMware 複数製品-リモートでコードが実行される脆弱性（CVE-2022-22957）
998974	CVE-2021-45232	WEB-MISC 2.10.1 より前の Apache APIIX ダッシュボード-エクスポートによる認証バイパスの脆弱性（CVE-2021-45232）
998975	CVE-2021-45232	WEB-MISC 2.10.1 より前の Apache APIIX ダッシュボード-インポートによる認証バイパスの脆弱性（CVE-2021-45232）
998976	CVE-2021-41739	WEB-MISC アーティカプロキシ-cyrus.events.php を介した OS コマンドインジェクションの脆弱性（CVE-2021-41739）
998977	CVE-2021-37927	WEB-MISC 7111 より前の ManageEngine アドマネージャープラス-認証バイパスの脆弱性（CVE-2021-37927）
998978	CVE-2021-36356	VSM サーバ経由の WEB-MISC クレイマー-writeBrowseFilePathAjax での認証されていないリモートコード実行の脆弱性（CVE-2021-36356）

署名ルール	**CVE ID**	説明
998979	CVE-2021-25094	WEB-WORDPRESS プラグイン 3.3.12 より前のタツビルダー-リモートでコードが実行される脆弱性 (CVE-2021-25094)

2022 年 **6** 月のシグネチャーアップデート

July 15, 2022

2022-06-16 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用可能な署名バージョン 88。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998958	CVE-2022-28810	WEB-MISC Zoho ManageEngine AdSelfService 6122 より前-ロック解除スクリプトによる OS コマンドインジェクションの脆弱性 (CVE-2022-28810)

署名ルール	**CVE ID**	説明
998959	CVE-2022-28810	WEB-MISC Zoho ManageEngine 6122 より前の AdSelfService-リセットスクリプトによる OS コマンドインジェクションの脆弱性 (CVE-2022-28810)
998960	CVE-2022-25237	WEB-MISC Bonita ウェブ 7.14.0 より前-i18ntranslation/../による認証バイパスの脆弱性 (CVE-2022-25237)
998961	CVE-2022-25237	WEB-MISC Bonita ウェブ 7.14.0 より前-認証バイパスの脆弱性経由; i18ntranslation (CVE-2022-25237)
998962	CVE-2022-0540	WEB-MISC アトラシアン Jira サーバーとデータセンター-Jira Seraph 認証バイパスの脆弱性 (CVE-2022-0540)
998963	CVE-2021-44548	WEB-MISC Apache Solr 8.11.1 より前-データインポートハンドラー SMB 攻撃の脆弱性 (CVE-2021-44548)

2022 年 **7** 月の署名更新

July 19, 2022

2022-07-08 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用可能な署名バージョン 89。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998942	CVE-2022-32532	WEB-MISC Apache Shiro 1.9.1 より前-RegexRequestMatcher ラインフィードによる脆弱性のバイパス (CVE-2022-32532)
998943	CVE-2022-32532	WEB-MISC Apache Shiro 1.9.1 より前-RegexRequestMatcher キャリッジリターンによる脆弱性のバイパス (CVE-2022-32532)
998944	CVE-2022-30157	WEB-MISC Microsoft SharePoint-信頼できないデータの逆シリアル化による RCE の脆弱性（CVE-2022-30157）
998945	CVE-2022-29847	WEB-MISC 進行中 Ipswitch WhatsUp Gold-認証されていないサーバー側リクエストフォージェリの脆弱性（CVE-2022-29847）
998946	CVE-2022-29535	WEB-MISC Zoho ManageEngine OPManager 複数のバージョン-ビュー経由の SQL インジェクションの脆弱性 (CVE-2022-29535)
998947	CVE-2022-29535	WEB-MISC Zoho ManageEngine OpManager 複数のバージョン-カテゴリを介した SQL インジェクションの脆弱性（CVE-2022-29535）

署名ルール	**CVE ID**	説明
998948	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus 7060 より前-リモートでコードが実行される脆弱性 (CVE-2022-28219)
998949	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus 7060 より前-タスクによる XXE インジェクションの脆弱性新しいコンテンツ（CVE-2022-28219）
998950	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus 7060 より前-タスクコンテンツを介した XXE インジェクションの脆弱性（CVE-2022-28219）
998951	CVE-2022-23642	WEB-MISC ソースグラフ 3.37 より前-gitserver サービスのリモートコード実行の脆弱性 (CVE-2022-23642)
998952	CVE-2022-23206	WEB-MISC Apache トラフィック制御 5.1.6 および 6.1.0 より前のトラフィック操作-SSRF 脆弱性 (CVE-2022-23206)
998953	CVE-2022-1609	WEB-WORDPRESS Weblizar 9.9.7 より前の学校管理プロプラグイン-リモートでコードが実行される脆弱性（CVE-2022-1609）
998954	CVE-2022-1209	WEB-WORDPRESS ワードプレスプラグインの最終メンバー 2.3.2 より前-オープンリダイレクトの脆弱性（CVE-2022-1209）
998955	CVE-2021-46360	WEB-MISC Composr-CMS-リモートでコードが実行される脆弱性 (CVE-2021-46360)

署名ルール	**CVE ID**	説明
998956	CVE-2021-43350	WEB-MISC Apache トラフィック制御トラフィック操作 5.1.4 および 6.0.1 より前-LDAP インジェクションの脆弱性（CVE-2021-43350）
998957	CVE-2017-9248	WEB-MISC R2 2017 SP1 より前の ASP.NET AJAX 用の Telerik UI-暗号化キー開示の脆弱性 (CVE-2017-9248)

2022 年 **7** 月の署名更新

August 12, 2022

2022-07-30 週に特定された脆弱性に対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用可能な署名バージョン 90。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998929	CVE-2022-34871	WEB-MISC Centreon 21.10.6 より前-SQL インジェクションの脆弱性（CVE-2022-34871）
998930	CVE-2022-29846	WEB-MISC 進行中 Ipswitch WhatsUp Gold-情報漏えいの脆弱性（CVE-2022-29846）

署名ルール	**CVE ID**	説明
998931	CVE-2022-29845	WEB-MISC 進行中 Ipswitch WhatsUp Gold-パストラバーサルの脆弱性（CVE-2022-29845）
998932	CVE-2022-28055	WEB-MISC FusionPBX 5.0.1 より前-リモートでコードが実行される脆弱性（CVE-2022-28055）
998933	CVE-2022-26138	WEB-MISC Confluence アプリに関するアトラシアンの質問-REST API を介したハードコードされた認証情報の脆弱性（CVE-2022-26138）
998934	CVE-2022-26138	WEB-MISC Confluence アプリに関するアトラシアンの質問-ログインフォームによるハードコードされた認証情報の脆弱性（CVE-2022-26138）
998935	CVE-2022-26135	WEB-MISC Jira サーバーとデータセンター-モバイルプラグインのサーバー側リクエストフォージェリの脆弱性（CVE-2022-26135）
998936	CVE-2022-21445	WEB-MISC Oracle OBIEE ADF Faces-信頼できないデータのデシリアライズの脆弱性（CVE-2022-21445）
998937	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-ネットワーク経由の RCE の脆弱性サーブレット URI と fwfilename (CVE-2022-2143)
998938	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-コマンドサーブレット URI および fwfilename による RCE の脆弱性 (CVE-2022-2143)

署名ルール	**CVE ID**	説明
998939	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-ネットワーク経由の RCE の脆弱性サーブレット URI と backup_filename (CVE-2022-2143)
998940	CVE-2022-2143	WEB-MISC アドバンテック iView 5.7.04.6469 より前-コマンドサーブレット URI と backup_filename による RCE の脆弱性 (CVE-2022-2143)
998941	CVE-2022-2099	6.6.0 より前の WEB-WORDPRESS WooCommerce プラグイン-支払いゲートウェイ HTML インジェクションの脆弱性 (CVE-2022-2099)

2022 年 **8** 月の署名更新

September 2, 2022

2022-08-23 週に特定された脆弱性に対して、新しいシグニチャルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用可能な署名バージョン 91。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998909	CVE-2022-38129	2.4.1 より前の WEB-MISC キーサイト SMS-パストラバーサルの脆弱性により RCE が許される (CVE-2022-38129)
998910	CVE-2022-37042, CVE-2022-27925	WEB-MISC Zimbra コラボレーションスイート-MailboxImportServlet 複数の脆弱性（CVE-2022-37042、CVE-2022-27925）
998911	CVE-2022-36446	WEB-MISC Webmin 複数バージョン-HTML インジェクションとリモートコード実行の脆弱性 (CVE-2022-36446)
998912	CVE-2022-35405	WEB-MISC Zoho ManageEngine パスワードマネージャープロ 12101 より前-Java デシリアライゼーションの脆弱性 (CVE-2022-35405)
998913	CVE-2022-34872	WEB-MISC Centreon 21.10.7 より前-vhidden 経由の SQL インジェクションの脆弱性 (CVE-2022-34872)
998914	CVE-2022-34872	21.10.7 より前の WEB-MISC Centreon-rpn_function を介した SQL インジェクションの脆弱性 (CVE-2022-34872)
998915	CVE-2022-34872	21.10.7 より前の WEB-MISC Centreon-unit_name を介した SQL インジェクションの脆弱性 (CVE-2022-34872)
998916	CVE-2022-34872	WEB-MISC Centreon 21.10.7 より前-警告による SQL インジェクションの脆弱性 (CVE-2022-34872)

署名ルール	**CVE ID**	説明
998917	CVE-2022-34872	WEB-MISC Centreon 21.10.7 より前-クリティカル経由の SQL インジェクションの脆弱性 (CVE-2022-34872)
998918	CVE-2022-34872	21.10.7 より前の WEB-MISC Centreon-def_type を介した SQL インジェクションの脆弱性 (CVE-2022-34872)
998919	CVE-2022-31813	WEB-MISC Apache HTTP サーバ最大 2.4.53-mod_proxy X-Forwarded-* ヘッダ削除の脆弱性（CVE-2022-31813）
998920	CVE-2022-31125	WEB-MISC Roxy-WI 6.1.0 より前-alert_consumer を介した認証バイパスの脆弱性 (CVE-2022-31125)
998921	CVE-2022-31101	WEB-MISC PrestaShop 2.1.1 より前のブロックウィッシュリスト-SQL インジェクションの脆弱性（CVE-2022-31101）
998922	CVE-2022-26137	WEB-MISC アトラシアン製品の複数バージョン-クロスオリジンリソース共有バイパスの脆弱性 (CVE-2022-26137)
998923	CVE-2022-24299	WEB-MISC pfSense CE 2.6.0 より前-vpn_openvpn_client.php を介したリモートコード実行の脆弱性（CVE-2022-24299）
998924	CVE-2022-24299	WEB-MISC pfSense CE 2.6.0 より前-vpn_openvpn_server.php を介したリモートコード実行の脆弱性（CVE-2022-24299）

署名ルール	**CVE ID**	説明
998925	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS BadgeOS プラグイン-取得実績とユーザー ID を介した SQL インジェクションの脆弱性（CVE-2022-0817）
998926	CVE-2021-36749	WEB-MISC Apache ドルイド-任意のローカルファイル漏えいの脆弱性（CVE-2021-36749）
998927	CVE-2021-26919	WEB-MISC Apache Druid 0.20.2 より前-autoDeserialize=True による信頼できないデシリアライゼーションの脆弱性（CVE-2021-26919）
998928	CVE-2021-26919	WEB-MISC Apache ドルイド 0.20.2 より前-DetectCustomCollations=True による信頼できないデシリアライゼーションの脆弱性（CVE-2021-26919）

2022 年 **9** 月の署名更新

September 27, 2022

2022-09-22 の週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 92 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注
>
> 投稿本文とレスポンス本文の署名ルールを有効にすると、Citrix ADC CPU に影響する場合があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998884	CVE-2022-38130	2.4.1 より前の WEB-MISC Keysight SMS-任意のファイルアップロードの脆弱性により SQL インジェクションが可能になる (CVE-2022-38130)
998885	CVE-2022-35741	4.16.1.1 より前の WEB-MISC Apache クラウドスタック-SAMLResponse による XML 外部エンティティインジェクションの脆弱性 (CVE-2022-35741)
998886	CVE-2022-35650	WEB-MISC Moodle 複数バージョン-黒板質問によるパストラバーサルの脆弱性 (CVE-2022-35650)
998887	CVE-2022-32551	WEB-MISC Zoho ManageEngine ServiceDesk MSP 10604 以前-/WEB-INF による認証されていない情報開示 (CVE-2022-32551)
998888	CVE-2022-31675	WEB-MISC VMware vRealize Operations Manager-認証バイパスの脆弱性 (CVE-2022-31675)
998889	CVE-2022-31674	WEB-MISC VMware vRealize Operations Manager-情報開示の脆弱性 (CVE-2022-31674)
998890	CVE-2022-31656	WEB-MISC VMware Workspace ONE Access-認証バイパスの脆弱性 (CVE-2022-31656)
998891	CVE-2022-31474	8.7.5 以前の WEB-WORDPRESS BackupBuddy プラグイン-backupbuddy_local_download による情報開示 (CVE-2022-31474)

署名ルール	**CVE ID**	説明
998892	CVE-2022-31137, CVE-2022-31126	6.1.1.0 より前の WEB-MISC Roxy-WI-複数のコマンドインジェクションの脆弱性 (CVE-2022-31137、CVE-2022-31126)
998893	CVE-2022-28731	WEB-MISC 2.11.3 より前の Apache JSPWiki-サーバー側リクエスト偽造の脆弱性 (CVE-2022-28731)
998894	CVE-2022-2551	1.4.7.1 より前の WEB WORDPRESS デュプリケータプラグイン-認証されていないバックアップダウンロードの脆弱性 (CVE-2022-2551)
998895	CVE-2022-2546	7.63 以前の WEB WORDPRESS オールインワン WP 移行プラグイン-ai1wm_export 経由で XSS の脆弱性を反映 (CVE-2022-2546)
998896	CVE-2022-2546	7.63 以前の WEB WORDPRESS オールインワン WP 移行プラグイン-ai1wm_import 経由で XSS の脆弱性を反映 (CVE-2022-2546)
998897	CVE-2022-24948	ウェブその他 2.11.2 より前の Apache jspWiki-XSS 脆弱性 (CVE-2022-24948)
998898	CVE-2022-2139	WEB-MISC 5.7.04.6469 より前のアドバンテック iView-menUservlet URI とページを介したパストラバーサルの脆弱性 (CVE-2022-2139)
998899	CVE-2022-2139	WEB-MISC 5.7.04.6469 より前のアドバンテック iView-CommandServlet URI とページを介したパストラバーサルの脆弱性 (CVE-2022-2139)

署名ルール	**CVE ID**	説明
998900	CVE-2022-2139	WEB-MISC 5.7.04.6469 より前のアドバンテック iView-CommandServlet URI とファイル名によるパストラバーサルの脆弱性 (CVE-2022-2139)
998901	CVE-2022-2139	WEB-MISC Advantech iView 5.7.04.6469 より前-NetworkServlet URI とファイル名によるパストラバーサルの脆弱性 (CVE-2022-2139)
998902	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得アチーブメントと除外による SQLi の脆弱性 (CVE-2022-0817)
998903	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得アチーブメントとインクルードによる SQLi の脆弱性 (CVE-2022-0817)
998904	CVE-2022-0817	3.7.1 以前の WEB-WORDPRESS badgeOS プラグイン-獲得した成果と注文による SQLi の脆弱性 (CVE-2022-0817)
998905	CVE-2022-0817	3.7.1 以前の WEB-WORDPRESS badgeOS プラグイン-獲得実績と注文による SQLi の脆弱性 (CVE-2022-0817)
998906	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得アチーブメントとオフセットによる SQLi の脆弱性 (CVE-2022-0817)

署名ルール	**CVE ID**	説明
998907	CVE-2022-0817	3.7.1 より前の WEB-WORDPRESS badgeOS プラグイン-獲得実績と制限による SQLi の脆弱性 (CVE-2022-0817)
998908	CVE-2018-20062, CVE-2019-9082	ウェブその他 5.1.32 より前の ThinkPHP 5.x-認証されていないリモートコード実行の脆弱性 (CVE-2018-20062、CVE-2019-9082)

2022 年 **10** 月の署名更新

October 7, 2022

2022-10-02 週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 93 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998871	CVE-2022-41082, CVE-2022-41040	WEB-MISC Microsoft Exchange Server-RCE 脆弱性 (CVE-2022-41082、CVE-2022-41040)

署名ルール	**CVE ID**	説明
998872	CVE-2022-37299	WEB-MISC Shirne CMS 1.2.0-/static/ueditor/php/controller.php 経由のパストラバーサルの脆弱性 (CVE-2022-37299)
998873	CVE-2022-36923	WEB-MISC Zoho ManageEngine 複数製品複数バージョン-認証バイパスの脆弱性 (CVE-2022-36923)
998874	CVE-2022-33891	WEB-MISC Apache Spark UI 複数バージョン-DoAS パラメータによるリモートコード実行の脆弱性（CVE-2022-33891）
998875	CVE-2022-3184, CVE-2022-3183	1.42.06162022 より前の WEB-MISC DataProbe iBoot-PDU-リモートコード実行の脆弱性 (CVE-2022-3184、CVE-2022-3183)
998876	CVE-2022-31814	2.1.4_26 より前の WEB-MISC pfSense pfBlockerNG-リモートコード実行の脆弱性 (CVE-2022-31814)
998877	CVE-2022-31097	WEB-MISC Apache Grafana-統合アラート保存 XSS 脆弱性 (CVE-2022-31097)
998878	CVE-2022-2903	3.6.13 より前の WEB-WORDPRESS NinjaForms プラグイン-PHP オブジェクトインジェクションの脆弱性 (CVE-2022-2903)
998879	CVE-2022-2552	1.4.7.1 より前の WEB WORDPRESS デュプリケータプラグイン-認証されていない情報漏えいの脆弱性（CVE-2022-2552）

署名ルール	**CVE ID**	説明
998880	CVE-2022-23854	WEB-MISC AVEVA InTouch どこからでもアクセスできる Secure Gateway-SG URI 経由のパストラバーサルの脆弱性 (CVE-2022-23854)
998881	CVE-2022-23854	WEB-MISC AVEVA InTouch どこからでもアクセスできる Secure Gateway-Blaze URI 経由のパストラバーサルの脆弱性 (CVE-2022-23854)
998882	CVE-2022-23854	WEB-MISC AVEVA InTouch どこでもアクセス Secure Gateway-AccessAnywhere URI 経由のパストラバーサルの脆弱性 (CVE-2022-23854)
998883	CVE-2017-9841	WEB-MISC PHPUnit 4.8.28 以前と 5.x 5.6.3 より前の 5.x-eval-stdin.php 経由でのリモートコード実行の脆弱性 (CVE-2017-9841)

2022 年 10 月の署名更新

October 16, 2022

2022-10-06 の週に特定された脆弱性について、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして設定すると、セキュリティに脆弱な攻撃からアプライアンスを保護できます。

署名バージョン

署名バージョン 94 は、NetScaler VPX 11.1、NetScaler 12.0、Citrix ADC 12.1、Citrix ADC 13.0、Citrix ADC 13.1 プラットフォームに適用されます。

> 注:
>
> 投稿本文と応答本文の署名ルールを有効にすると、Citrix ADC CPU に影響する可能性があります。

一般的な脆弱性エントリ **(CVE)** の洞察

次に、署名ルール、CVE ID、およびその説明の一覧を示します。

署名ルール	**CVE ID**	説明
998870	CVE-2022-41082, CVE-2022-41040	WEB-MISC Microsoft Exchange Server-RCE 脆弱性 (CVE-2022-41082、CVE-2022-41040)

ボット管理

October 7, 2021

着信ウェブトラフィックはボットで構成され、ほとんどの組織はボット攻撃に苦しむことがあります。Web およびモバイルアプリケーションはビジネスの重要な収益ドライバーであり、ほとんどの企業はボットなどの高度なサイバー攻撃の脅威にさらされています。

ボットは、人間よりもはるかに速い速度で特定のアクションを繰り返し自動的に実行するソフトウェアプログラムです。ボットは、Web ページの操作、フォームの送信、アクションの実行、テキストのスキャン、またはコンテンツのダウンロードを行うことができます。ソーシャルメディアプラットフォームで動画にアクセスしたり、コメントを投稿したり、ツイートしたりできます。チャットボットと呼ばれるいくつかのボットは、人間のユーザーとの基本的な会話を保持することができます。

カスタマーサービス、自動チャット、検索エンジンクローラなどの便利なサービスを実行するボットは良いボットです。同時に、ウェブサイトからコンテンツをスクラップまたはダウンロードしたり、ユーザーの資格情報、スパムコンテンツを盗んだり、他の種類のサイバー攻撃を実行したりできるボットは、悪いボットです。

悪意のあるタスクを実行する悪意のあるボットが多数存在するため、ボットトラフィックを管理し、Web アプリケーションをボット攻撃から保護することが不可欠です。Citrix ボット管理を使用すると、着信ボットトラフィックを検出し、ボット攻撃を軽減して Web アプリケーションを保護できます。

Citrix ボット管理は、不良ボットを特定し、高度なセキュリティ攻撃からアプライアンスを保護するのに役立ちます。良いボットと悪いボットを検出し、着信トラフィックがボット攻撃であるかどうかを識別します。ボット管理を使用することで、攻撃を軽減し、Web アプリケーションを保護できます。

Citrix ADC ボット管理には、次のような利点があります。

- ボット、スクリプト、ツールキットから守ります。静的なシグニチャベースの防御とデバイスフィンガープリントを使用して、リアルタイムの脅威を軽減します。
- 自動化された基本攻撃と高度な攻撃を中和します。アプリ層 DDoS、パスワードスプレー、パスワードスタッフィング、価格スクレーパー、コンテンツスクレーパーなどの攻撃を防ぎます。
- **API** と投資を保護します。不当な誤用から API を保護し、自動化されたトラフィックからインフラストラクチ

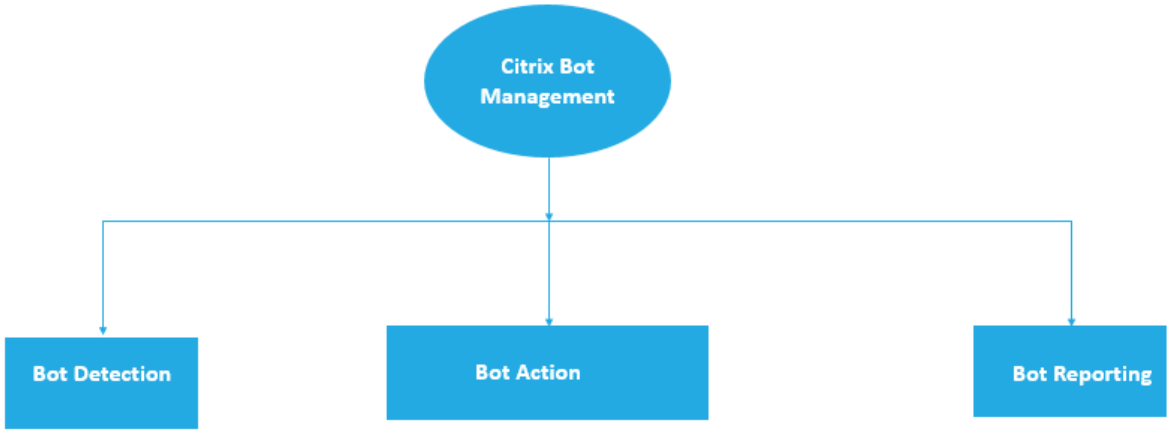
ャへの投資を保護します。

Citrix ボット管理システムを使用することでメリットが得られるユースケースには、次のようなものがあります。

- ブルートフォースログイン。政府の Web ポータルは、ユーザーのログインをブルートフォースしようとするボットによって常に攻撃を受けています。組織は、Web ログを調べ、辞書攻撃アプローチを使用して迅速なログイン試行とパスワードの増分で特定のユーザーが何度も選択されていることを確認することで、攻撃を発見しました。法律により、彼らは彼ら自身と彼らのユーザーを保護しなければなりません。Citrix ボット管理を展開することで、デバイスのフィンガープリントとレート制限技術を使用して、ブルートフォースログインを停止できます。
- 不正なボットやデバイスの指紋不明ボットをブロックします。1 つの Web エンティティは、1 日あたり 100,000 人の訪問者を取得します。彼らは基礎となるフットプリントをアップグレードする必要があり、彼らは大金を費やしています。最近の監査では、トラフィックの 40% がボット、コンテンツのスクレイピング、ニュースの選択、ユーザープロファイルのチェックなどから来ていることが判明しました。このトラフィックをブロックして、ユーザーを保護し、ホスティングコストを削減したいと考えています。ボット管理を使用すると、既知の悪いボットをブロックし、サイトを叩いている未知のボットを指紋することができます。これらのボットをブロックすることで、ボットのトラフィックを 90% 削減できます。

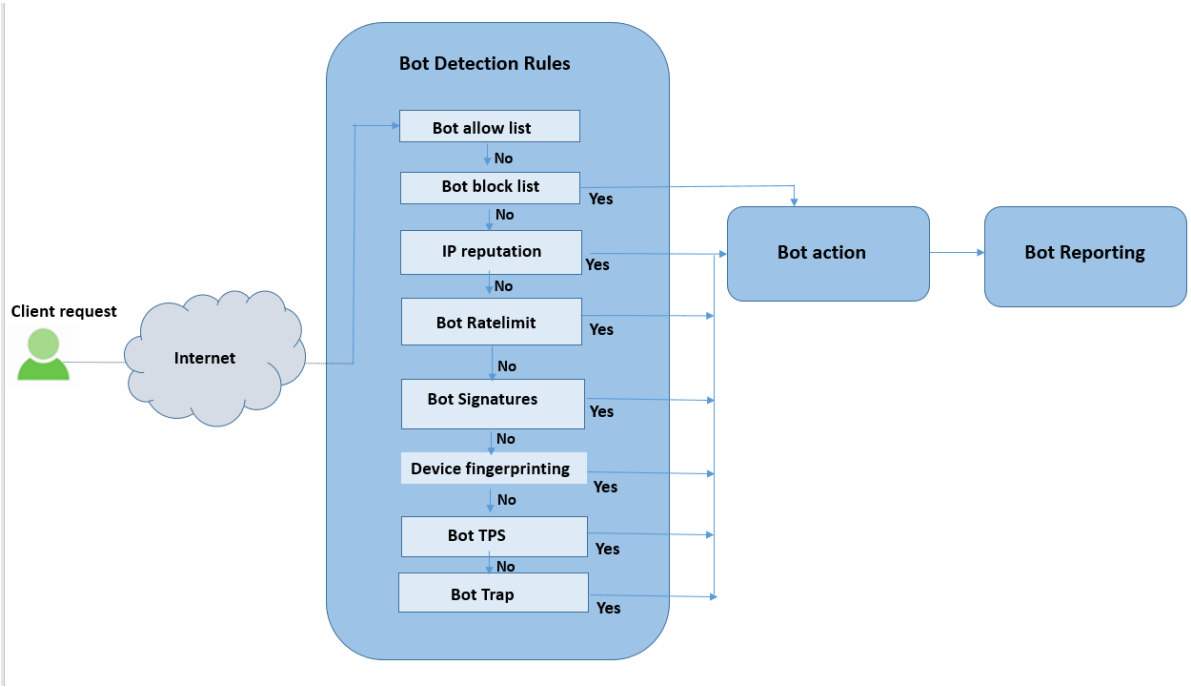
Citrix ボット管理の機能

Citrix ボット管理は、組織が高度なセキュリティ攻撃から Web アプリケーションと公共資産を保護するのに役立ちます。着信トラフィックがボットの場合、次の図に示すように、ボット管理システムはボットタイプを検出し、アクションを割り当てて、ボットインサイトを生成します。



Citrix ADC ボット管理の仕組み

次の図は、Citrix ADC ボット管理の仕組みを示しています。このプロセスには、着信トラフィックを良好または悪いボットとして検出するのに役立つ 8 つの検出技術が含まれます。デフォルトでは、シグニチャによって検出された良好なボットは許可され、シグニチャによって検出された不良ボットはドロップされます。



1. このプロセスは、アプライアンスでボット管理機能を有効にすることから始まります。
2. クライアントがリクエストを送信すると、アプライアンスはボットポリシー規則を使用してトラフィックを評価します。着信要求がボットとして識別された場合、アプライアンスはボット検出プロファイルを適用します。
3. デフォルトまたはカスタムボット署名ファイルは、ボット検出プロファイルにバインドする必要があります。ボット署名ファイルには、入ってくるボットタイプを識別するためのボット署名ルールのリストがあります。
4. ボット検出ルールは、シグニチャファイルの 8 つの検出カテゴリで使用できます。カテゴリは、許可リスト、ブロックリスト、静的シグニチャ、IP レピュテーション、デバイスフィンガープリント、およびレート制限です。ボットのトラフィックに基づいて、システムはトラフィックに検出ルールを適用します。
5. 着信ボットトラフィックがボット許可リストのエントリと一致する場合、システムは他の検出手法をバイパスし、関連するアクションがデータをログに記録します。
6. ボット許可リスト以外の検出手法では、着信要求が設定されたルールと一致する場合、対応するアクションが適用されます。可能なアクションは、ドロップ、リダイレクト、リセット、軽減、およびログです。CAPTCHA は、IP レピュテーション、デバイスフィンガープリント、および TPS 検出技術でサポートされている緩和アクションです。

ボット検出

April 25, 2022

Citrix ADC ボット管理システムは、6 つの異なる手法を使用して、着信ボットトラフィックを検出します。この手法は、ボットタイプを検出するための検出ルールとして使用されます。この技術は、ボット許可リスト、ボットブロックリスト、IP レピュテーション、デバイスフィンガープリント、レート制限、ボットトラップ、TPS、および

CAPTCHA です。

> 注:
>
> ボット管理では、ブロックリスト、許可リスト、およびレート制限の手法について、最大 32 個の構成エンティティがサポートされています。

ボットホワイトリスト。許可リストとしてバイパスできる IP アドレス、サブネット、およびポリシー式のカスタマイズされたリスト。

ボットのブラックリスト。Web アプリケーションへのアクセスをブロックする必要のある IP アドレス、サブネット、およびポリシー式のカスタマイズされたリスト。

IP レピュテーション。このルールは、着信ボットトラフィックが悪意のある IP アドレスからのものかどうかを検出します。

デバイスのフィンガープリント。このルールは、着信ボットトラフィックの着信要求ヘッダーおよび着信クライアントボットトラフィックのブラウザ属性にデバイスフィンガープリント ID があるかどうかを検出します。

> 制限事項:
>
> 1. クライアントブラウザで JavaScript を有効にする必要があります。
> 2. XML 応答では機能しません。

ボットログ式。検出技術を使用すると、追加情報をログメッセージとしてキャプチャできます。データには、URL を要求したユーザーの名前、送信元 IP アドレス、およびユーザーが要求を送信した送信元ポート、または式から生成されたデータを指定できます。

レート制限。このルールは、同じクライアントから送信される複数の要求をレート制限します。

ボットトラップ。クライアント応答でトラップ URL をアドバタイズすることで、自動ボットを検出してブロックします。クライアントが人間のユーザーの場合、URL は見えず、アクセスできないように見えます。この検出技術は、自動ボットからの攻撃をブロックするのに効果的です。

TPS。最大要求数および要求の増加率が設定された時間間隔を超えた場合に、着信トラフィックをボットとして検出します。

CAPTCHA。このルールは、ボット攻撃を軽減するために CAPTCHA を使用します。CAPTCHA は、受信トラフィックが人間のユーザーからのものか、自動化されたボットからのものかを判断するためのチャレンジ/レスポンスの検証です。この検証は、Web アプリケーションにセキュリティ違反を引き起こす自動ボットをブロックするのに役立ちます。CAPTCHA は、IP レピュテーションおよびデバイスフィンガープリント検出技術のボットアクションとして設定できます。

それでは、ボットトラフィックを検出して管理するための各手法をどのように構成できるかを見てみましょう。

アプライアンスを **Citrix ADC CLI** ベースのボット管理構成にアップグレードする方法

アプライアンスを古いバージョン（Citrix ADC リリース 13.0 ビルド 58.32 以前）からアップグレードする場合は、最初に既存のボット管理構成を Citrix ADC CLI ベースのボット管理構成に手動で変換する必要があります。ボット

管理設定を手動で変換するには、次の手順を実行します。

1. 最新バージョンにアップグレードした後、次のコマンドを使用してアップグレードツール「upgrade_bot_config.py」に接続します。

   コマンドプロンプトで入力します。

   ```
   shell "/var/python/bin/python /netscaler/upgrade_bot_config.py > /var/bot_upgrade_commands.txt"
   ```

2. 次のコマンドを使用して、構成を実行します。

   コマンドプロンプトで入力します。

   ```
   batch -f /var/bot_upgrade_commands.txt
   ```

3. アップグレードした設定を保存します。

```
save ns config
```

Citrix ADC CLI ベースのボット管理を構成する

ボット管理設定では、1 つ以上のボット検出手法を特定のボットプロファイルにバインドできます。このプロセスを開始するには、アプライアンスでボット管理機能を有効にします。有効にしたら、ボット署名ファイルをアプライアンスにインポートします。インポート後、ボットプロファイルを作成する必要があります。次に、ボットプロファイルをバインドしたボットポリシーを作成し、着信トラフィックをボットとして評価し、ポリシーをグローバルまたは仮想サーバーにバインドします。

> 注:
>
> アプライアンスを古いバージョンからアップグレードする場合は、最初に既存のボット管理構成を手動で変換する必要があります。詳細については、「Citrix ADC CLI ベースのボット管理構成にアップグレードする方法」セクションを参照してください。

Citrix ADC ベースのボット管理を構成するには、以下の手順を実行する必要があります。

1. ボット管理を有効にする
2. ボット署名のインポート
3. ボットプロファイルの追加
4. ボットプロファイルのバインド
5. ボットポリシーの追加
6. バインドボットポリシー
7. ボット設定の構成

ボット管理を有効にする

開始する前に、アプライアンスでボット管理機能が有効になっていることを確認します。新しい Citrix ADC または VPX をお持ちの場合は、構成する前にその機能を有効にする必要があります。Citrix ADC または VPX アプライアン

スを以前のバージョンの Citrix ADC ソフトウェアバージョンから現在のバージョンにアップグレードする場合は、構成する前に機能を有効にする必要があります。コマンドプロンプトで入力します。

```
enable ns feature Bot
```

ボット署名のインポート

デフォルトのシグニチャボットファイルをインポートし、ボットプロファイルにバインドできます。コマンドプロンプトで入力します。

```
import bot signature [<src>] <name> [-comment <string>] [-overwrite]
```

src。インポートしたシグニチャファイルを格納するファイルのローカルパスとその名前、または URL（プロトコル、ホスト、パス、ファイル名）。注: インポートするオブジェクトが、アクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。最大長:2047
名前。Citrix ADC 上のボット署名ファイルオブジェクトに割り当てる名前。これは必須の議論です。最大長:31
コメント。シグネチャファイルオブジェクトに関する情報を保持するためのコメント。最大長:255
上書き。既存のファイルを上書きします。
注: 署名ファイル内のコンテンツを更新するには、このoverwriteオプションを使用します。または、update bot signature <name> コマンドを使用して、Citrix ADC アプライアンスの署名ファイルを更新します。

例

```
import bot signature http://www.example.com/signature.json signaturefile -comment commentsforbot -overwrite
```

> 注:
>
> 上書きオプションを使用して、署名ファイル内のコンテンツを更新できます。また、update bot signature <name>コマンドを使用して、Citrix ADC アプライアンスの署名ファイルを更新することもできます。

ボットプロファイルの追加

ボットプロファイルは、アプライアンスでボット管理を設定するためのプロファイル設定の集まりです。ボット検出を実行するように設定を構成できます。

コマンドプロンプトで入力します。

```
add bot profile <name> [-signature <string>] [-errorURL <string>] [-trapURL <string>] [-comment <string>] [-whiteList ( ON | OFF )] [-blackList ( ON | OFF )] [-rateLimit ( ON | OFF )] [-deviceFingerprint ( ON | OFF )] [-deviceFingerprintAction ( none | log | drop | redirect | reset | mitigation
```



```
 )] [-ipReputation ( ON | OFF )] [-trap ( ON | OFF )] [-trapAction ( none |
 log | drop | redirect | reset )] [-tps ( ON | OFF )]
```

例:

```
add bot profile profile1 -signature signature -errorURL http://www.example
.com/error.html -trapURL /trap.html -whitelist ON -blacklist ON -ratelimit
 ON -deviceFingerprint ON -deviceFingerprintAction drop -ipReputation ON -
trap ON
```

ボットプロファイルのバインド

ボットプロファイルを作成したら、ボット検出メカニズムをプロファイルにバインドする必要があります。

コマンドプロンプトで入力します。

```
bind bot profile <name> ((-blackList [-type ( IPv4 | Subnet | Expression
 )] [-enabled ( ON | OFF )] [-value <string>] [-action ( log | drop |
reset )] [-logMessage <string>] [-comment <string>])| (-whiteList [-type
 ( IPv4 | Subnet | Expression )] [-enabled ( ON | OFF )] [-value <string
>] [-log ( ON | OFF )] [-logMessage <string>] [-comment <string>]))|
(-rateLimit [-type ( session |SOURCE_IP | url )] [-enabled ( ON | OFF
)] [-url <string>] [-cookieName <string>] [-rate <positive_integer>] [-
timeslice <positive_integer>] [-action ( none | log | drop | redirect |
reset )] [-logMessage <string>] [-comment <string>])| (-ipReputation [-
category <ipReputationCategory>] [-enabled ( ON | OFF )] [-action ( none
 | log | drop | redirect | reset | mitigation )] [-logMessage <string>]
[-comment <string>])| (-captchaResource [-url <string>] [-enabled ( ON |
 OFF )] [-waitTime <positive_integer>] [-gracePeriod <positive_integer>]
 [-mutePeriod <positive_integer>] [-requestLengthLimit <positive_integer
>] [-retryAttempts <positive_integer>] [-action ( none | log | drop |
redirect | reset )] [-logMessage <string>] [-comment <string>])| (-tps
 [-type ( SOURCE_IP | GeoLocation | REQUEST_URL | Host )] [-threshold <
positive_integer>] [-percentage <positive_integer>] [-action ( none | log |
 drop | redirect | reset | mitigation )] [-logMessage <string>] [-comment <
string>])
```

例:

次に、IP レピュテーション検出手法を特定のボットプロファイルにバインドする例を示します。

```
bind bot profile profile5 -ipReputation -category BOTNET -enabled ON -
action drop -logMessage message
```


ボットポリシーの追加

ボットトラフィックを評価するためのボットポリシーを追加する必要があります。

コマンドプロンプトで入力します。

```
add bot policy <name> -rule <expression> -profileName <string> [-undefAction
 <string>] [-comment <string>] [-logAction <string>]
```

各項目の意味は次のとおりです。

Name：ボットポリシーの名前。文字、数字、またはアンダースコア文字（_）で始まり、文字、数字、ハイフン（-）、ピリオド（.）ポンド（#）、スペース（）、アットマーク（@）、等号（=）、コロン（:）、およびアンダースコア文字のみを含める必要があります。ボットポリシーの追加後に変更できます。

ルール。指定されたリクエストにボットプロファイルを適用するかどうかを決定するためにポリシーが使用する式。これは必須の議論です。最大長：1499

ProfileName。リクエストがこのボットポリシーと一致した場合に適用するボットプロファイルの名前。これは必須の議論です。最大長：127

unDefaction だポリシー評価の結果が未定義 (UNDEF) の場合に実行するアクション。UNDEF イベントは、内部エラー状態を示します。最大長：127

[コメント]。このボットポリシーに関するあらゆる種類の情報。最大長：255

logAction。このポリシーに一致するリクエストに使用するログアクションの名前。最大長：127

例:

```
add bot policy pol1 -rule "HTTP.REQ.HEADER(\"header\").CONTAINS(\"custom
\")"- profileName profile1 -undefAction drop -comment commentforbotpolicy -
logAction log1
```

ボットポリシーをグローバルにバインドする

コマンドプロンプトで入力します。

```
bind bot global -policyName <string> -priority <positive_integer> [-gotoPriorityExpres
 <expression>][-type ( REQ_OVERRIDE | REQ_DEFAULT )] [-invoke (-labelType (
 vserver | policylabel )-labelName <string>)]
```

例:

```
bind bot global -policyName pol1 -priority 100 -gotoPriorityExpression NEXT
-type REQ_OVERRIDE
```

ボットポリシーを仮想サーバーにバインドする

コマンドプロンプトで入力します。


```
bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] )| <
serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>]
 [-gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )] [-
invoke (<labelType> <labelName>)] )| -analyticsProfile <string>@)
```

例:

```
bind lb vserver lb-server1 -policyName pol1 -priority 100 -gotoPriorityExpression
 NEXT -type REQ_OVERRIDE
```

ボット設定の構成

必要に応じて、デフォルト設定をカスタマイズできます。
コマンドプロンプトで入力します。

```
set bot settings [-defaultProfile <string>] [-javaScriptName <string>]
 [-sessionTimeout <positive_integer>] [-sessionCookieName <string>] [-
dfpRequestLimit <positive_integer>] [-signatureAutoUpdate ( ON | OFF )]
[-signatureUrl <URL>] [-proxyServer <ip_addr|ipv6_addr|*>] [-proxyPort <
port|*>]
```

各項目の意味は次のとおりです。

DefaultProfile。接続がどのポリシーにも一致しない場合に使用するプロファイル。デフォルト設定は”で、一致しない接続をさらにフィルタリングすることなく、Citrix ADC に返信します。最大長: 31

JavaScriptName。ボットネット機能がレスポンスで使用する JavaScript の名前です。文字または数字で始まり、1～31 の英字、数字、およびハイフン（-）とアンダースコア（_）記号で構成する必要があります。次の要件は、Citrix ADC CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます（たとえば、「私のクッキー名」または「私のクッキー名」）。最大長: 31

セッションタイムアウト。セッションがタイムアウトします（秒単位）。この時間が経過すると、ユーザセッションは終了します。

最小値:1、最大値:65535

セッションクッキー名。ボットネット機能がトラッキングに使用する SessionCookie の名前です。文字または数字で始まり、1～31 の英字、数字、およびハイフン（-）とアンダースコア（_）記号で構成する必要があります。次の要件は、Citrix ADC CLI にのみ適用されます。名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符（「私の Cookie 名」や「Cookie 名」など）で囲みます。最大長: 31

dfPrequestLimit。デバイスフィンガープリントが有効になっている場合に、ボットセッション Cookie なしで許可するリクエストの数。

最小値:1、最大値:4294967295

SignatureAutoUpdate。ボットの自動更新シグネチャを有効/無効にするために使用されるフラグ。

指定可能な値: オン、オフ

デフォルト値:OFF

シグネチャ URL。サーバーからボット署名マッピングファイルをダウンロードするための URL。

デフォルト値: https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json。

最大長：2047

ProxyServer。AWS から更新された署名を取得するためのプロキシサーバー IP。

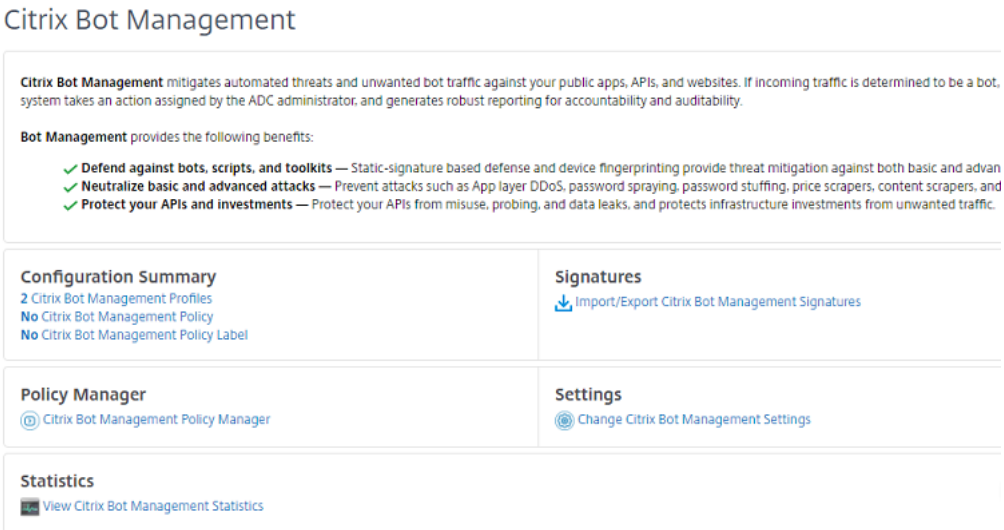
ProxyPort。AWS から更新された署名を取得するためのプロキシサーバーポート。デフォルト値:8080

例:

```
set bot settings -defaultProfile profile1 -javaScriptName json.js -sessionTimeout
 1000 -sessionCookieName session
```

Citrix ADC GUI を使用したボット管理の構成

Citrix ADC ボット管理を構成するには、まずアプライアンスで機能を有効にします。有効にすると、ボットポリシーを作成して、着信トラフィックをボットとして評価し、そのトラフィックをボットプロファイルに送信できます。次に、ボットプロファイルを作成し、そのプロファイルをボット署名にバインドします。別の方法として、デフォルトのボットシグネチャファイルのクローンを作成し、シグネチャファイルを使用して検出手法を設定することもできます。署名ファイルを作成したら、それをボットプロファイルにインポートできます。



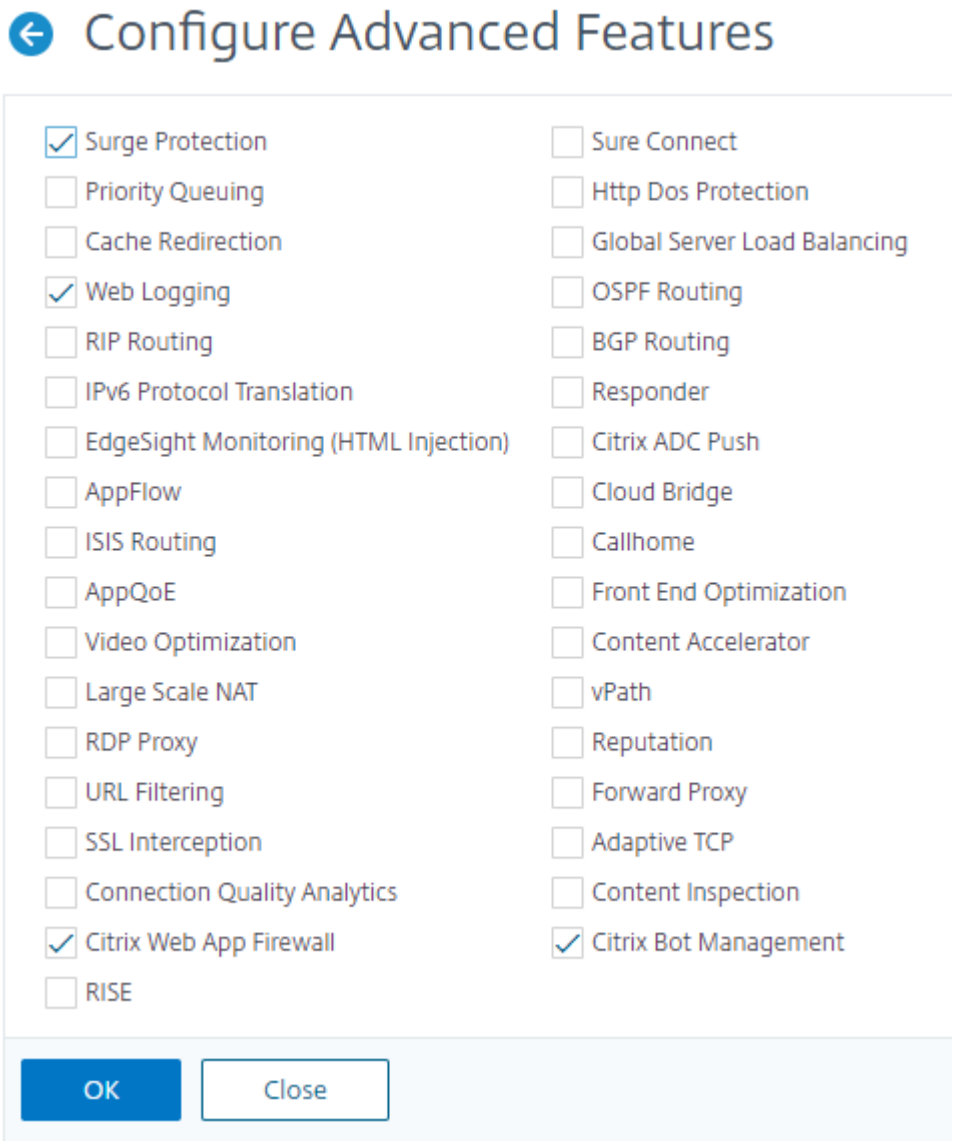
1. ボット管理機能を有効にする
2. ボット管理設定を構成する
3. Citrix ボットのデフォルト署名のクローン作成
4. Citrix ボット署名をインポートする
5. ボット署名設定を構成する
6. ボットプロファイルの作成

7. ボットポリシーの作成

ボット管理機能を有効にする

ボット管理を有効にするには、次の手順を実行します。

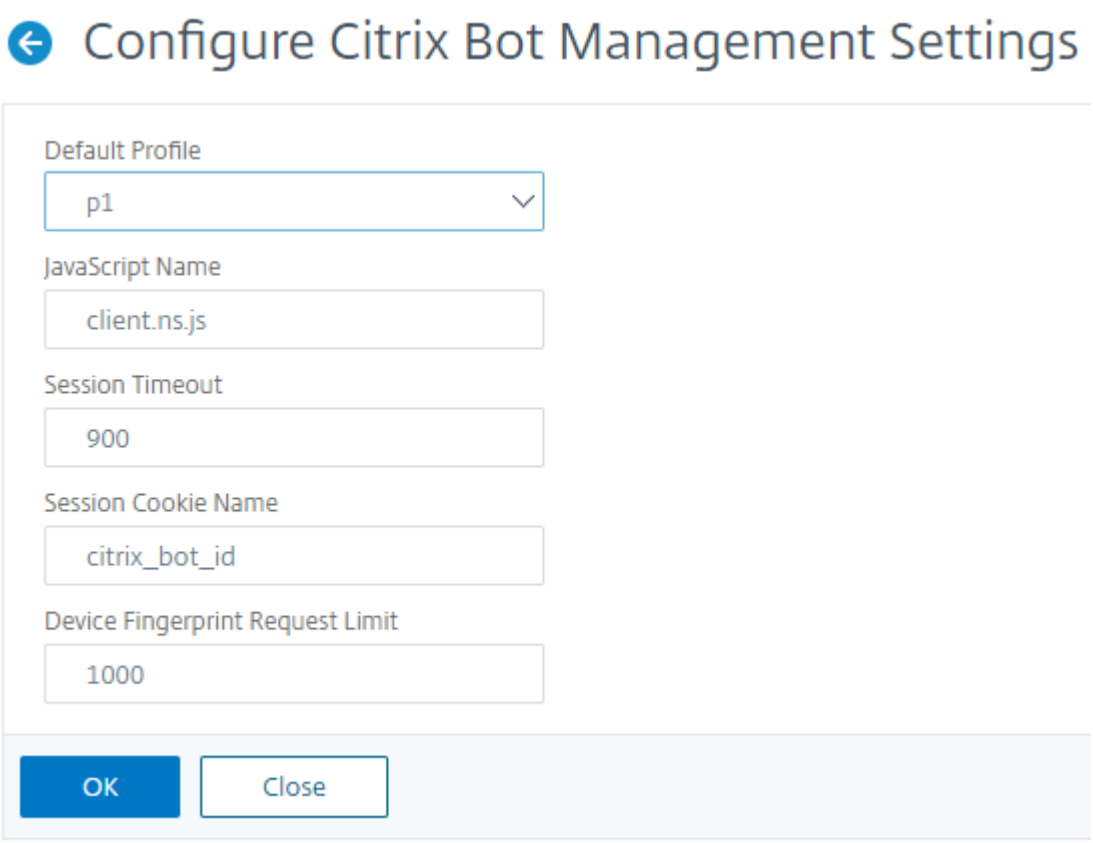
1. ナビゲーションウィンドウで、[システム] を展開し、[設定] をクリックします。
2. [拡張機能の設定] ページで、[ボット管理] チェックボックスをオンにします。
3. 「**OK**」をクリックし、「閉じる」をクリックします。



デバイスフィンガープリント技術のボット管理設定を構成する

次の手順を実行して、デバイスフィンガープリント技術を設定します。

1. ［セキュリティ］>［**Citrix Bot** 管理］に移動します。
2. 詳細ペインの［設定］で、［**Citrix Bot** 管理設定の変更］をクリックします。
3. ［**Citrix Bot** 管理設定の構成］で、次のパラメーターを設定します。
   a) デフォルトプロファイル。ボットプロファイルを選択します。
   b) JavaScript の名前。ボット管理がクライアントへの応答で使用する JavaScript ファイルの名前です。
   c) セッションタイムアウト。ユーザーセッションが終了するまでのタイムアウト（秒単位）。
   d) セッションクッキー。ボット管理システムが追跡に使用するセッション Cookie の名前。
   e) デバイスフィンガープリント要求制限。デバイスフィンガープリントが有効になっている場合に、ボットセッション Cookie なしで許可するリクエストの数



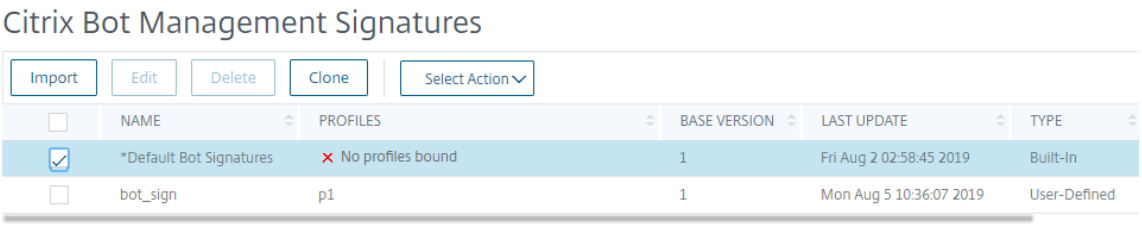
4. **OK** をクリックします。

ボット署名ファイルの複製

次の手順を実行して、ボット署名ファイルのクローンを作成します。

1. [セキュリティ] > [**Citrix **Bot** の管理と署名 **] に移動します。

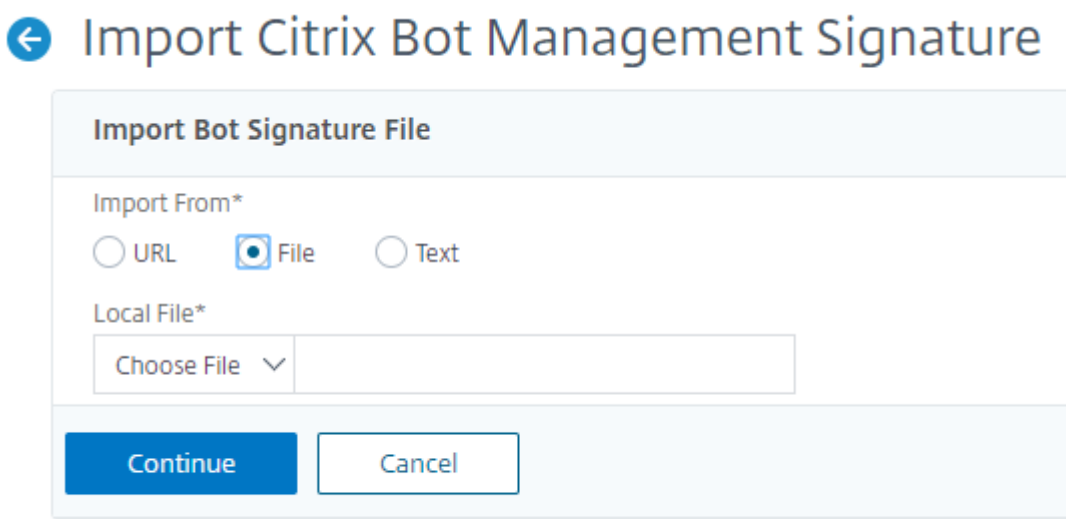
2. [**Citrix Bot Management** の署名] ページで、デフォルトのボット署名レコードを選択し、[複製] をクリックします。
3. [ボット署名の複製] ページで、名前を入力し、署名データを編集します。
4. [**Create**] をクリックします。



ボット署名ファイルをインポートする

独自の署名ファイルがある場合は、ファイル、テキスト、または URL としてインポートできます。次の手順を実行して、ボット署名ファイルをインポートします。

1. [セキュリティ] > [**Citrix **Bot** の管理と署名 **] に移動します。
2. **Citrix Bot Management** の署名ページで、ファイルを URL、ファイル、またはテキストとしてインポートします。
3. [続行] をクリックします。

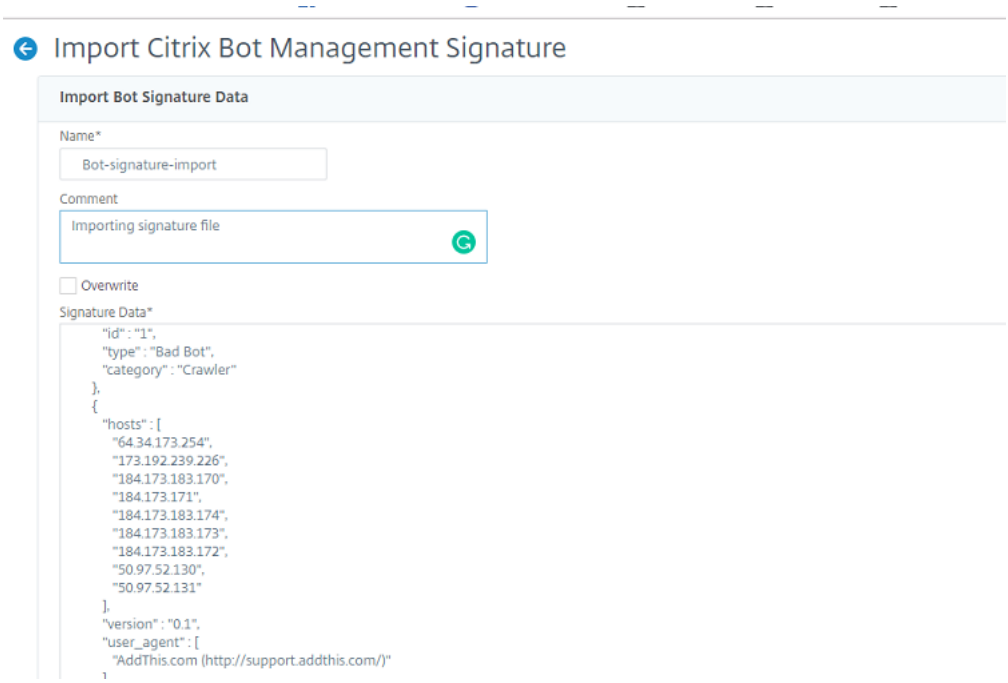


4. [Citrix Bot 管理署名のインポート] ページで、次のパラメーターを設定します。
   a) Name: ボット署名ファイルの名前。
   b) [コメント]。インポートしたファイルに関する簡単な説明。

c) 上書き。ファイルの更新中にデータの上書きを許可するには、このチェックボックスをオンにします。

d) 署名データ。シグニチャパラメータの変更

5. [完了] をクリックします。



Citrix ADC GUI を使用してボット許可リストを構成する

この検出技術により、許可されたリストの URL を設定した URL をバイパスできます。許可リストの URL を設定するには、次の手順を実行します。

1. [セキュリティ] > [**Citrix**Bot の管理とプロファイル **] に移動します。
2. [**Citrix Bot** 管理プロファイル] ページでファイルを選択し、[編集] をクリックします。
3. [**Citrix Bot** 管理プロファイル] ページで、[署名の設定] セクションに移動し、[ホワイトリスト] をクリックします。
4. [ホワイトリスト] セクションで、次のパラメータを設定します。

   a) 有効。このチェックボックスをオンにすると、検出プロセスの一環として許可リストの URL が検証されます。

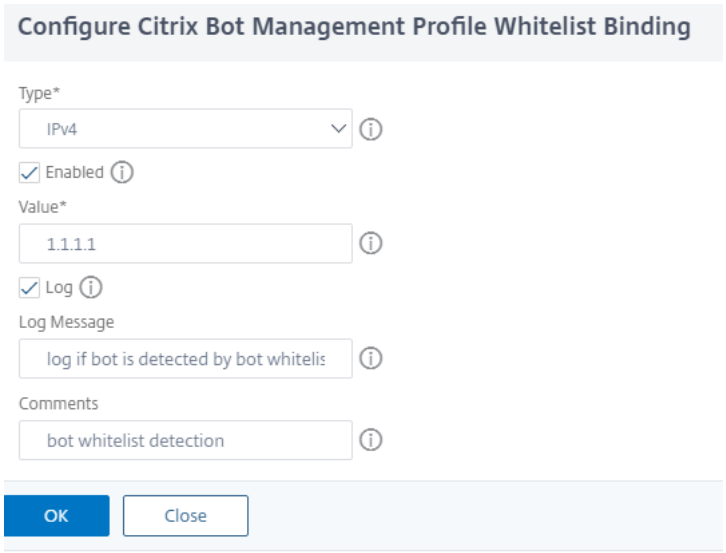
   b) [タイプ] を設定します。許可リスト URL を構成します。URL は、ボットの検出中にバイパスされます。[追加] をクリックして、ボット許可リストに URL を追加します。

   c) [**Citrix Bot** 管理プロファイルのホワイトリストバインドの構成] ページで、次のパラメーターを設定します。

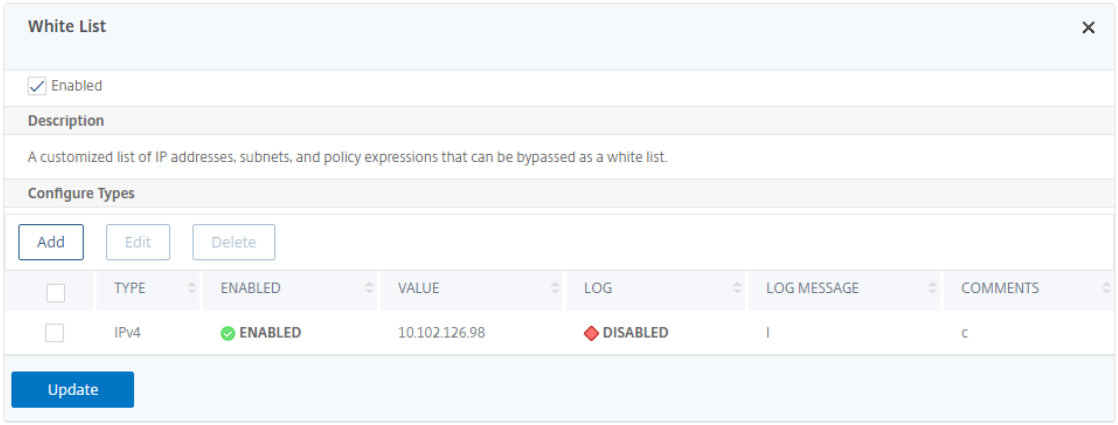
      i. [タイプ]。URL タイプは、IPv4 アドレス、サブネット IP アドレス、またはポリシー式に一致する IP アドレスです。

      ii. 有効。URL を検証するには、このチェックボックスをオンにします。

iii. [値]。URL アドレス。
iv. ログ。ログエントリを保存するには、このチェックボックスをオンにします。
v. ログメッセージ。ログの簡単な説明。
vi. コメント。許可リスト URL に関する簡単な説明。
vii. **OK** をクリックします。



5. [更新] をクリックします。
6. [完了] をクリックします。

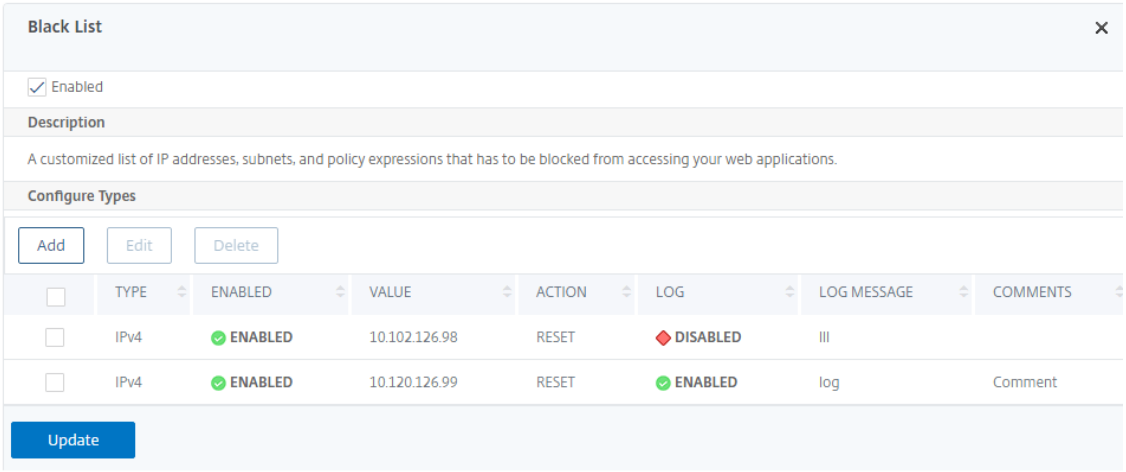


Citrix ADC GUI を使用してボットブロックリストを構成する

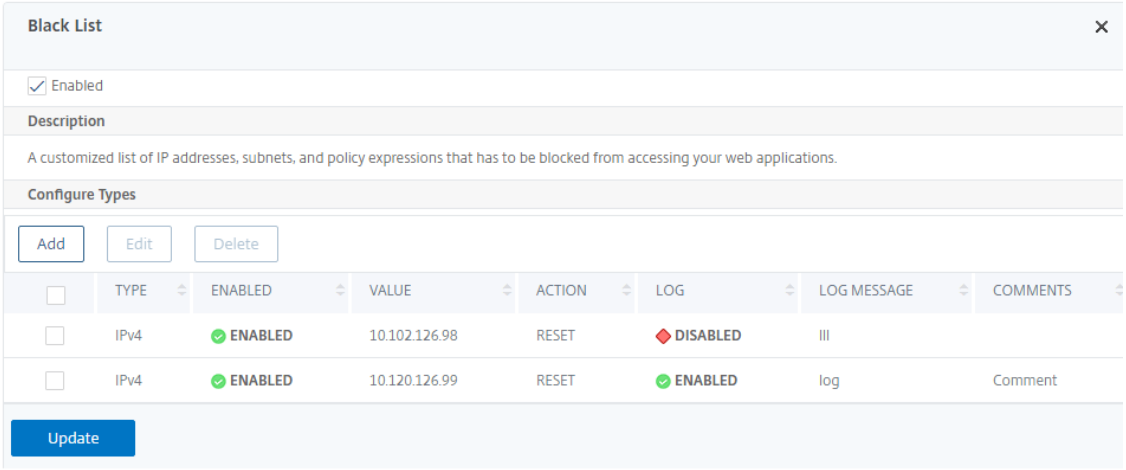
この検出技術により、ブロックリストとして設定した URL を削除できます。ブロックリスト URL を設定するには、次の手順を実行します。

1. [セキュリティ] > [**Citrix**Bot の管理とプロファイル **] に移動します。
2. [**Citrix Bot** 管理プロファイル] ページで、署名ファイルを選択して [編集] をクリックします。

3. [**Citrix Bot** 管理プロファイル] ページで、[署名の設定] セクションに移動し、[ブラックリスト] をクリックします。
4. [ブラックリスト] セクションで、次のパラメータを設定します。
   a) 有効。このチェックボックスをオンにすると、検出プロセスの一環としてブロックリストの URL が検証されます。
   b) [タイプ] を設定します。URL をボットブロックリスト検出プロセスの一部として設定します。これらの URL は、ボット検出中にドロップされます。[追加] をクリックして、ボットブロックリストに URL を追加します。
   c) [**Citrix Bot** 管理プロファイルのブラックリストバインドの構成] ページで、次のパラメーターを設定します。
      i. [タイプ]。URL タイプは、IPv4 アドレス、サブネット IP アドレス、または IP アドレスです。
      ii. 有効。URL を検証するには、このチェックボックスをオンにします。
      iii. [値]。URL アドレス。
      iv. ログ。ログエントリを保存するには、このチェックボックスをオンにします。
      v. ログメッセージ。ログインの簡単な説明。
      vi. コメント。ブロックリスト URL に関する簡単な説明。
      vii. **OK** をクリックします。



5. [更新] をクリックします。
6. [完了] をクリックします。



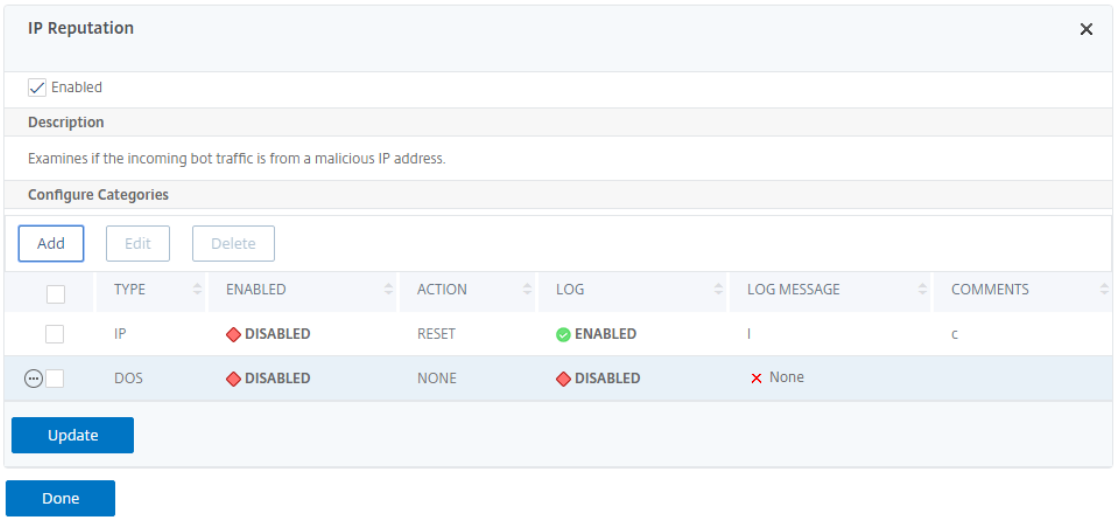
Citrix ADC GUI を使用して IP レピュテーションを構成する

この設定は、ボット IP レピュテーション機能の前提条件です。この検出技術により、着信 IP アドレスからの悪意のあるアクティビティがあるかどうかを識別できます。設定の一環として、さまざまな悪意のあるボットカテゴリを設定し、それぞれにボットアクションを関連付けます。IP レピュテーション手法を設定するには、次の手順を実行します。

1. [セキュリティ] > [**Citrix **Bot** の管理とプロファイル **] に移動します。
2. [**Citrix Bot** 管理プロファイル］ページで、署名ファイルを選択して［編集］をクリックします。
3. [**Citrix Bot** 管理プロファイル］ページで、［署名の設定］セクションに移動し、［**IP** レピュテーション］をクリックします。
4. [**IP** レピュテーション] セクションで、次のパラメータを設定します。
   a) 有効。検出プロセスの一環として受信ボットトラフィックを検証するには、このチェックボックスをオンにします。
   b) カテゴリを設定します。IP レピュテーション技術は、さまざまなカテゴリの着信ボットトラフィックに使用できます。設定されたカテゴリに基づいて、ボットトラフィックをドロップまたはリダイレクトできます。[追加] をクリックして、悪意のあるボットカテゴリを設定します。
   c) ［**Citrix Bot** 管理プロファイルの **IP** レピュテーションバインドの構成］ページで、次のパラメーターを設定します。
      i. カテゴリー。リストから悪意のあるボットのカテゴリを選択します。カテゴリに基づいてボットアクションを関連付けます。
      ii. 有効。IP レピュテーションシグニチャ検出を検証するには、このチェックボックスをオンにします。
      iii. ボットアクション。設定されたカテゴリに基づいて、アクション、ドロップ、リダイレクト、緩和、または CAPTCHA アクションを割り当てることはできません。
      iv. ログ。ログエントリを保存するには、このチェックボックスをオンにします。

v. ログメッセージ。ログの簡単な説明。
vi. コメント。ボットカテゴリに関する簡単な説明。

5. **OK** をクリックします。
6. ［更新］をクリックします。
7. ［完了］をクリックします。

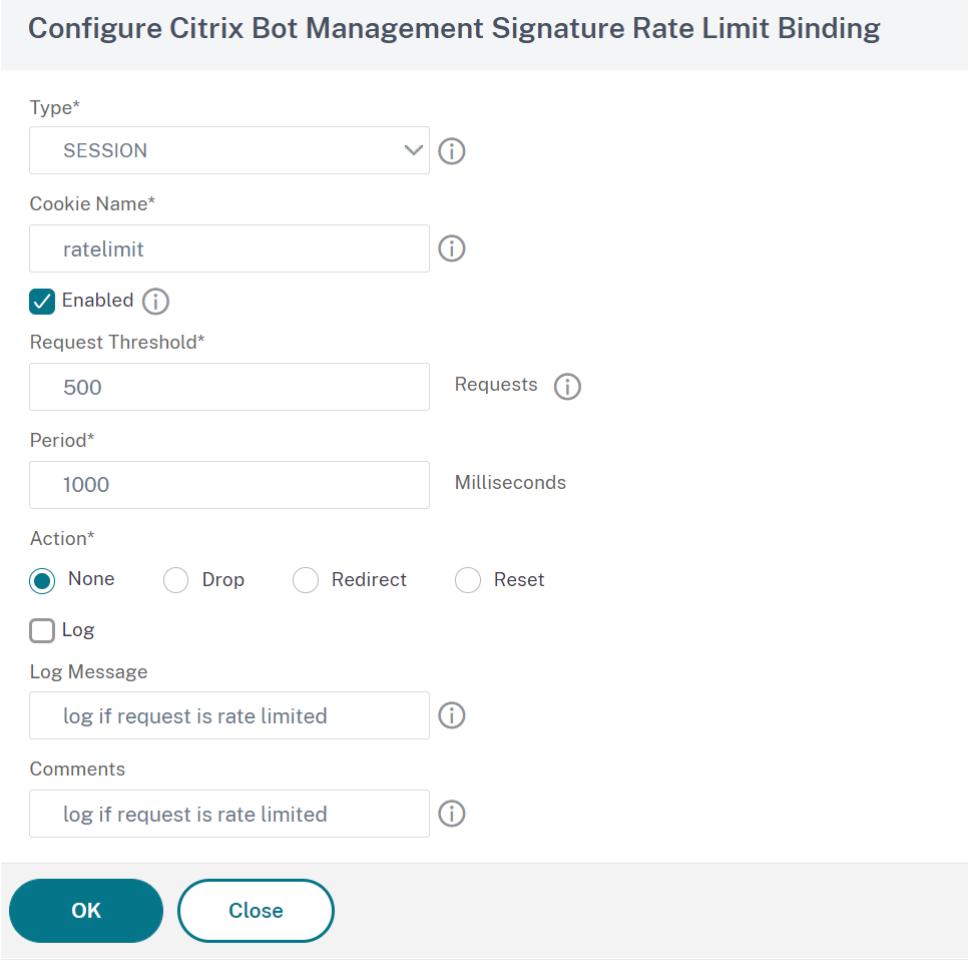


Citrix ADC GUI を使用してボットレート制限を構成する

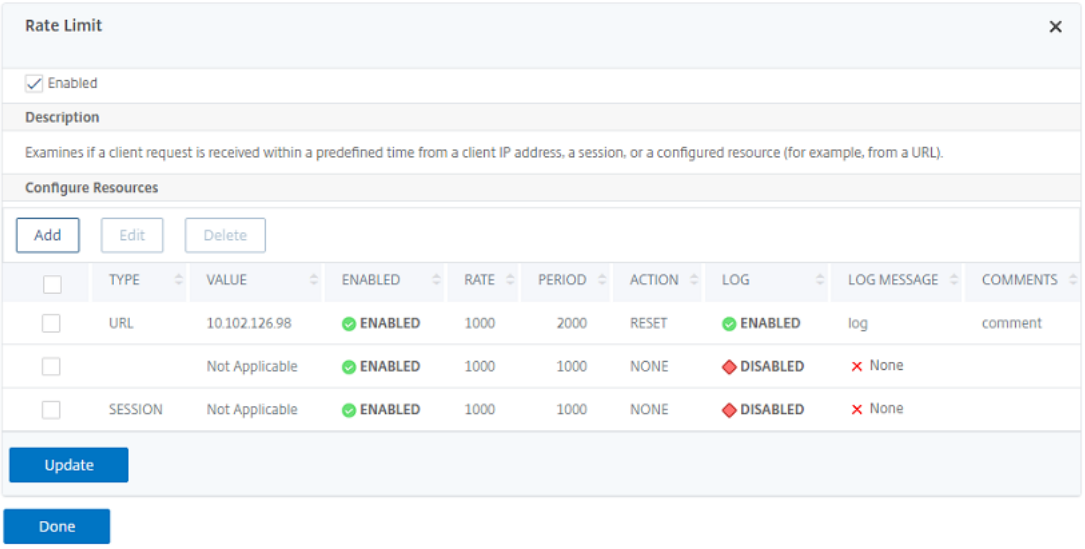
この検出技術により、クライアント IP アドレス、セッション、または設定されたリソース（URL など）から事前に定義された時間内に受信した要求の数に基づいて、ボットをブロックできます。レートリミット手法を設定するには、次の手順を実行します。

1. ［セキュリティ］>［**Citrix**Bot の管理とプロファイル **］に移動します。
2. ［**Citrix Bot** 管理プロファイル］ページで、署名ファイルを選択して［編集］をクリックします。
3. ［**Citrix Bot** 管理プロファイル］ページで、［署名の設定］セクションに移動し、［レート制限］をクリックします。
4. ［ レート制限（**Rate Limit** ）］セクションで、次のパラメータを設定します。
   a) 有効。検出プロセスの一環として受信ボットトラフィックを検証するには、このチェックボックスをオンにします。
   b) セッション。セッションに基づいて要求をレート制限します。セッションに基づいてレート制限要求を設定するには、[Add] をクリックします。
   c) ［**Citrix Bot Management** 署名レート制限の構成］ページで、次のパラメーターを設定します。
      i. カテゴリー。リストから悪意のあるボットのカテゴリを選択します。カテゴリに基づいてアクションを関連付けます。
      ii. 有効。受信ボットトラフィックを検証するには、このチェックボックスをオンにします。

iii. ボットアクション。選択したカテゴリのボットアクションを選択します。
iv. ログ。ログエントリを保存するには、このチェックボックスをオンにします。
v. ログメッセージ。ログの簡単な説明。
vi. コメント。ボットカテゴリに関する簡単な説明。
vii. **OK** をクリックします。



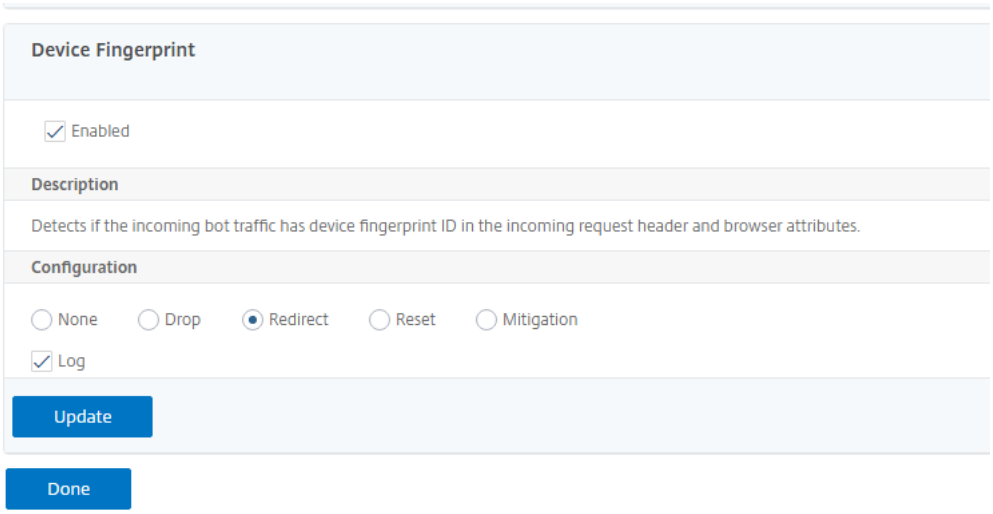
5. ［更新］をクリックします。
6. ［完了］をクリックします。



Citrix ADC GUI を使用してデバイスフィンガープリント技術を構成する

この検出技術は、Java スクリプトチャレンジをクライアントに送信し、デバイス情報を抽出します。デバイス情報に基づいて、この技術はボットトラフィックをドロップまたはバイパスします。手順に従って、検出手法を設定します。

1. ［セキュリティ］> ［**Citrix**Bot の管理とプロファイル **］に移動します。
2. ［**Citrix Bot** 管理プロファイル］ページで、署名ファイルを選択して［編集］をクリックします。
3. ［**Citrix Bot** 管理プロファイル］ページで、［署名の設定］セクションに移動し、［デバイスフィンガープリント］をクリックします。
4. [デバイスフィンガープリント] セクションで、次のパラメータを設定します。
   a) 有効。このオプションを設定すると、ルールが有効になります。
   b) Configuration - 指定されたデバイスフィンガープリントに対して、アクション、ドロップ、リダイレクト、緩和、または CAPTCHA アクションを割り当てません。
   c) ログ。ログエントリを保存するには、このチェックボックスをオンにします。
5. ［更新］をクリックします。
6. ［完了］をクリックします。



モバイル **(Android)** アプリケーション用のデバイスフィンガープリント技術を構成する

デバイスフィンガープリント技術は、クライアントへの HTML 応答に JavaScript スクリプトを挿入することにより、着信トラフィックをボットとして検出します。JavaScript スクリプトは、ブラウザによって呼び出されると、ブラウザとクライアントの属性を収集し、アプライアンスに要求を送信します。これらの属性を調べて、トラフィックが Bot か人間かを判断します。

検出技術はさらに拡張され、モバイル (Android) プラットフォームでボットを検出できます。ウェブアプリケーションとは異なり、モバイル (Android) トラフィックでは、JavaScript スクリプトに基づくボット検出は適用されません。モバイルネットワーク内のボットを検出するために、この手法では、クライアント側のモバイルアプリケーションと統合されたボットモバイル SDK を使用します。SDK は、モバイルトラフィックをインターセプトし、デバイスの詳細を収集し、データをアプライアンスに送信します。アプライアンス側では、検出技術はデータを調べ、接続が Bot または人間からのものかどうかを判断します。

モバイルアプリケーション用のデバイス指紋技術の仕組み

次の手順では、モバイルデバイスからのリクエストが人間かボットかを検出するボット検出ワークフローについて説明します。

1. ユーザーがモバイルアプリケーションを操作すると、デバイスの動作はボットのモバイル SDK によって記録されます。
2. クライアントは Citrix ADC アプライアンスに要求を送信します。
3. 応答を送信すると、アプライアンスは、セッションの詳細とクライアントパラメータを収集するためのパラメータを含むボットセッションクッキーを挿入します。
4. モバイルアプリケーションが応答を受信すると、モバイルアプリケーションと統合された Citrix Bot SDK が応答を検証し、記録されたデバイスフィンガープリントパラメータを取得してアプライアンスに送信します。
5. アプライアンス側のデバイスフィンガープリント検出技術は、デバイスの詳細を検証し、ボットセッションクッキーが疑わしいボットであるかどうかにかかわらず、ボットセッションクッキーを更新します。

6. クッキーの有効期限が切れた場合、またはデバイスの指紋保護がデバイスパラメータを定期的に検証して収集することを好む場合は、すべての手順またはチャレンジが繰り返されます。

前提要件

モバイルアプリケーション用の Citrix ADC デバイスの指紋検出技術を開始するには、モバイルアプリケーションにボットモバイル SDK をダウンロードしてインストールする必要があります。

CLI を使用してモバイル（**Android**）アプリケーションの指紋検出技術を構成する

コマンドプロンプトで入力します。

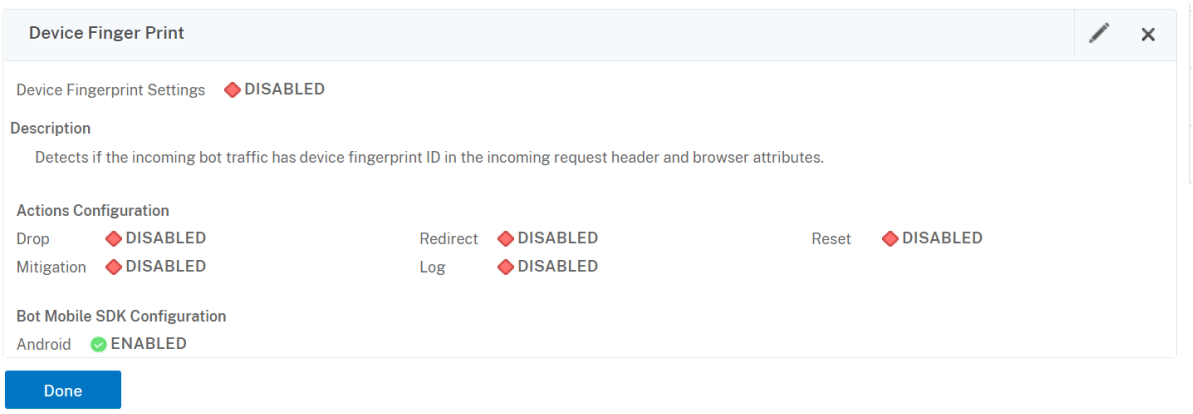
```
set bot profile <profile name> -deviceFingerprintMobile ( NONE | Android )
```

例**:**

```
set bot profile profile 1 -deviceFingerprintMobile Android
```

GUI を使用してモバイル（**Android**）アプリケーションのデバイス指紋検出技術を構成する

1. [セキュリティ] > [**Citrix**Bot の管理とプロファイル **] に移動します。
2. [**Citrix Bot** 管理プロファイル] ページでファイルを選択し、[編集] をクリックします。
3. [**Citrix Bot** 管理プロファイル] ページで、[プロファイル設定] の [デバイスフィンガープリント] をクリックします。
4. [**Bot Mobile SDK** の設定] セクションで、モバイルクライアントの種類を選択します。
5. [更新して完了] をクリックします。



ボットログ式の設定

クライアントがボットとして識別された場合、Citrix ボット管理により、追加情報をログメッセージとしてキャプチャできます。データには、URL を要求したユーザーの名前、送信元 IP アドレス、およびユーザーが要求を送信した

送信元ポート、または式から生成されたデータを指定できます。カスタムログを実行するには、ボット管理プロファイルでログ式を設定する必要があります。

CLI を使用してボットプロファイルのログ式をバインドします

コマンドプロンプトで入力します。

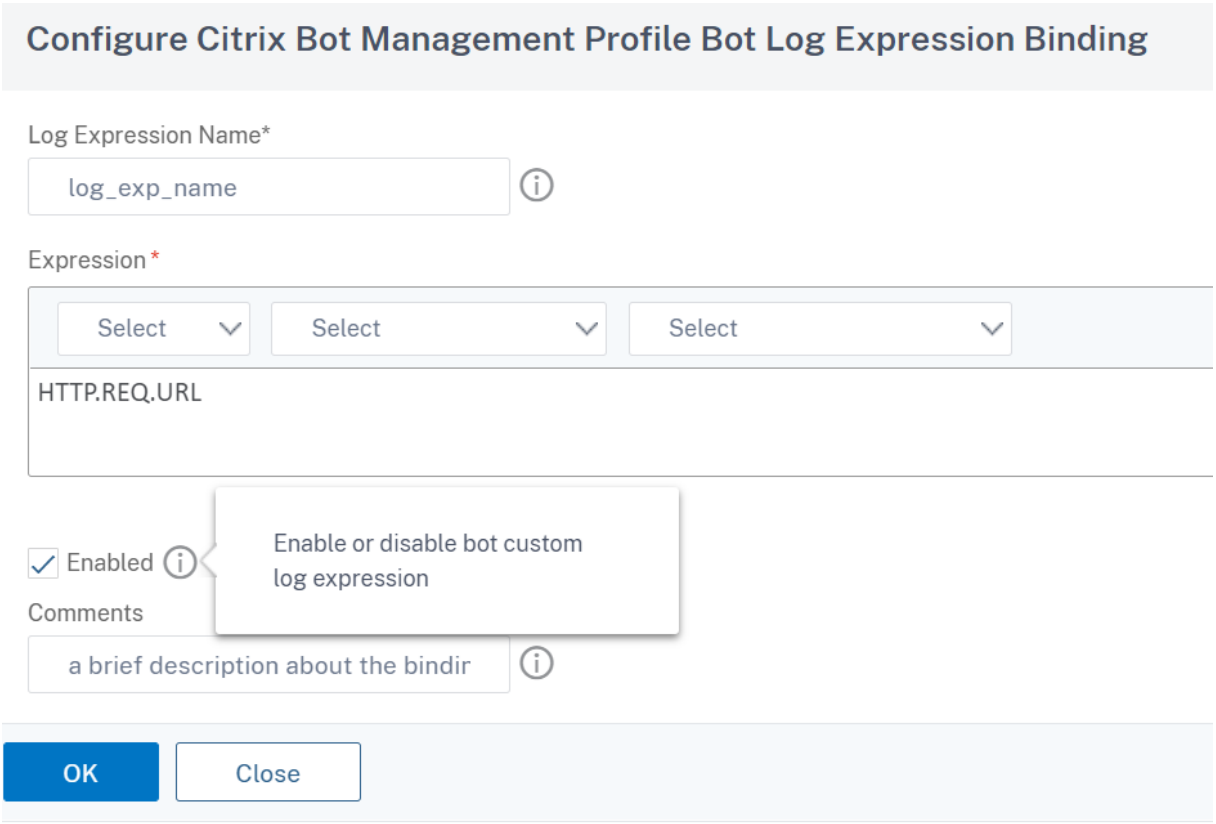
```
1 bind bot profile <name> (-logExpression -name <string> -expression <
     expression> [-enabled ( ON | OFF )]) -comment <string>
2 <!--NeedCopy-->
```

例**:**

```
bind bot profile profile1 -logExpression exp1 -expression HTTP.REQ.URL -
enabled ON -comment ”testing log expression”
```

GUI を使用してログ式をボットプロファイルにバインドする

1. ［セキュリティ］**>**［**Citrix Bot** 管理］**>**［プロファイル］に移動します。
2. ［**Citrix Bot Management Profile**］ページで、［プロファイル設定 ****** ］セクションから［****** ボットログ式］を選択します。
3. [ボットログ式の設定 ***]** セクションで、**[**** 追加] をクリックします。
4. ［**Citrix Bot Management** プロファイルのボットログ式バインドの構成］ページで、次のパラメータを設定します。
   a) ログ式名。ログ式の名前。
   b) 式。ログ式を入力します。
   c) 有効。ログ式バインディングを有効または無効にします。
   d) コメント。ボットログ式バインディングについての簡単な説明。
5.「**OK**」をクリックして「完了」をクリックします。



ボットトラップ手法を構成する

Citrix ボットトラップ技術は、クライアント応答にトラップ URL をランダムまたは定期的に挿入します。トラップ URL リストを作成し、その URL を追加することもできます。クライアントが人間のユーザーの場合、URL は見えず、アクセスできないように見えます。ただし、クライアントが自動化されたボットの場合、URL はアクセス可能であり、アクセスされると、攻撃者はボットに分類され、ボットからの後続のリクエストはブロックされます。このトラップ技術は、ボットからの攻撃をブロックするのに効果的です。

トラップ URL は、設定可能な長さの英数字の URL で、設定可能な間隔で自動生成されます。また、この方法では、よくアクセスする Web サイトまたは頻繁にアクセスする Web サイトのトラップ挿入 URL を設定することもできます。これにより、トラップ挿入 URL に一致するリクエストに対してボットトラップ URL を挿入する目的を指示できます。

> 注:
>
> ボットトラップ URL は自動生成されますが、Citrix ADC ボット管理では、ボットプロファイルでカスタマイズされたトラップ URL を構成できます。これは、
> ボット検出技術を強化し、攻撃者がトラップ URL にアクセスしにくくするために行われます。

ボットトラップの設定を完了するには、次の手順を完了する必要があります。

1. ボットトラップ URL を有効にする
2. ボットプロファイルでのボットトラップ URL の設定
3. ボットトラップ挿入 URL をボットプロファイルにバインドする
4. ボット設定でボットトラップの URL の長さと間隔を構成する

ボットトラップ **URL** 保護を有効にする

開始する前に、アプライアンスでボットトラップ URL 保護が有効になっていることを確認する必要があります。コマンドプロンプトで入力します。

```
enable ns feature Bot
```

ボットプロファイルでのボットトラップ **URL** の設定

ボットトラップ URL を設定し、ボットプロファイルでトラップアクションを指定できます。
コマンドプロンプトで入力します。

```
add bot profile <name> -trapURL <string> -trap ( ON | OFF )-trapAction <
trapAction>
```

各項目の意味は次のとおりです。

trapURL。ボット保護がトラップ URL として使用する URL。最大長: 127

trap。ボットトラップ検出を有効にします。可能な値: オン、オフ、デフォルト値:OFF

trapAction。ボット検出に基づいて実行するアクション。可能な値: なし、ログ、ドロップ、リダイレクト、リセット、緩和。デフォルト値: なし

例**:**

```
add bot profile profile1 -trapURL www.bottrap1.com trap ON -trapAction
RESET
```

ボットトラップ挿入 **URL** をボットプロファイルにバインドする

ボットトラップ挿入 URL を設定し、ボットプロファイルにバインドできます。
コマンドプロンプトで入力します。

```
bind bot profile <profile_name> trapInsertionURL -url <url> -enabled ON|OFF
-comment <comment>
```

各項目の意味は次のとおりです。

URL。ボットトラップ URL が挿入される要求 URL 正規表現パターン。最大長: 127

例**:**


```
bind bot profile profile1 trapInsertionURL -url www.example.com -enabled ON
-comment insert a trap URL randomly
```

ボット設定でボットトラップの **URL** の長さと間隔を構成する

ボットトラップの URL の長さを設定し、ボットトラップ URL を自動生成する間隔を設定することもできます。
コマンドプロンプトで入力します。

```
set bot settings -trapURLAutoGenerate ( ON | OFF )-trapURLInterval <positive_integer
> -trapURLLength <positive_integer>
```

各項目の意味は次のとおりです。

trapURLInterval。ボットトラップ URL が更新されるまでの時間（秒単位）。デフォルト値:3600、最小値:300、最大値:86400

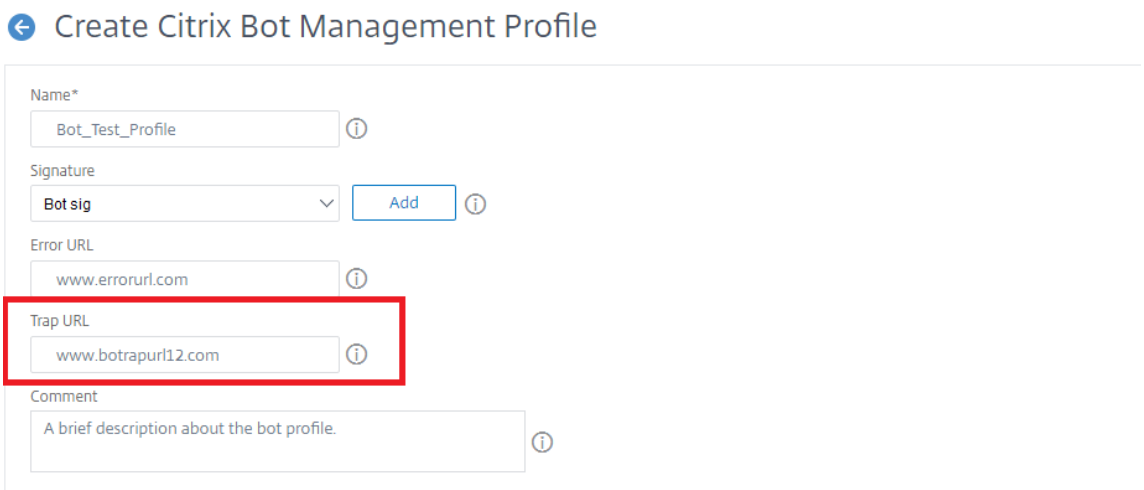
trapURLLength。自動生成されたボットトラップ URL の長さ。デフォルト値:32、最小値:10、最大値:255

例**:**

```
set bot settings -trapURLAutoGenerate ON -trapURLInterval 300 -trapURLLength
 60
```

GUI を使用したボットトラップ **URL** の設定

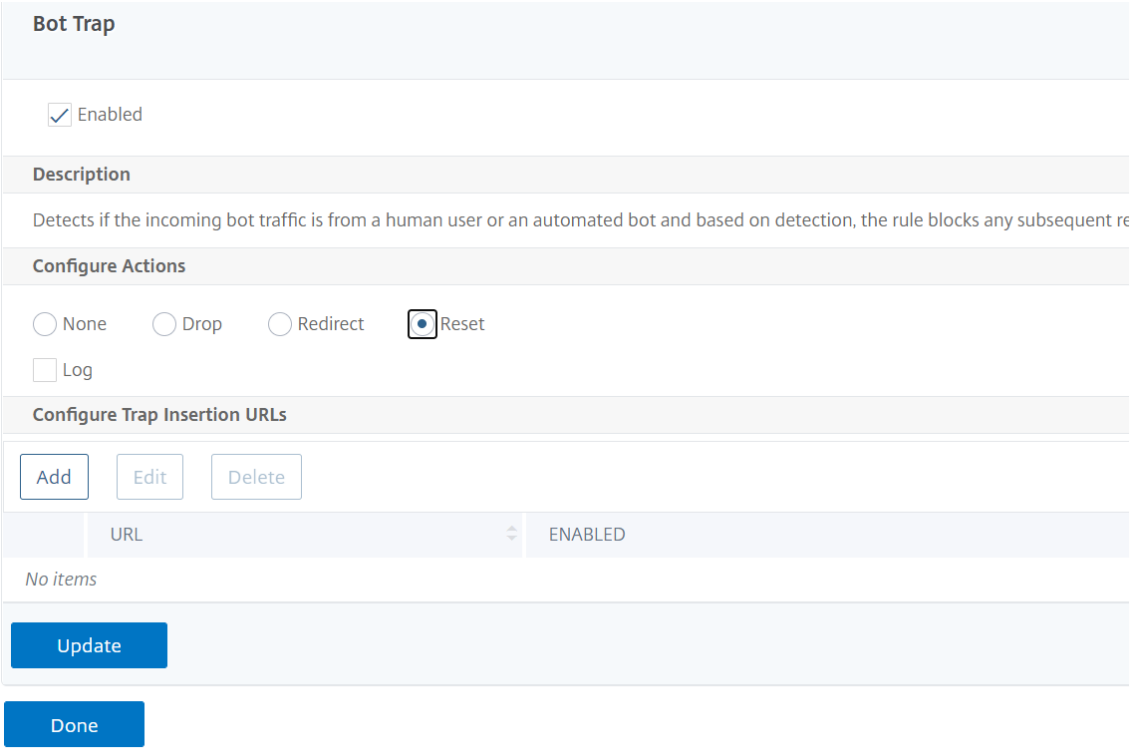
1. ［セキュリティ］**>**［**Citrix Bot** 管理］**>**［プロファイル］に移動します。
2. ［**Citrix Bot** 管理プロファイル］ページで、［編集］をクリックしてボットトラップ URL 手法を構成します。
3. ［**Citrix Bot** 管理プロファイルの作成］ページで、［全般］セクションにボットトラップ URL を入力します。



4. ［**Citrix Bot** 管理プロファイルの作成］ページで、［プロファイル設定 ****** ］の［****** ボットトラップ］をクリックします。

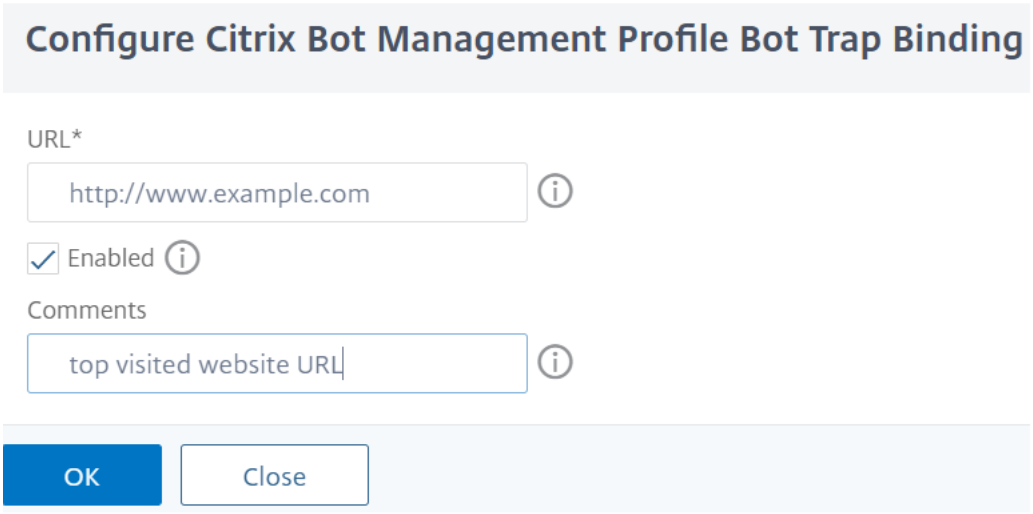
5. [**Bot Trap**] セクションで、次のパラメータを設定します。

   a. 有効。ボットトラップ検出を有効にするには、このチェックボックスをオンにします。
   b. Description。URL に関する簡単な説明。
   c. アクションの設定。ボットトラップアクセスによって検出されたボットに対して実行されるアクション。



6. [トラップ挿入 **URL** の設定] セクションで、[追加] をクリックします。
7. [**Citrix Bot** 管理プロファイルのボットトラップバインドの構成］ページで、次のパラメーターを設定します。

   a) トラップ URL。ボットトラップ挿入 URL として確認する URL を入力します。
   b) 有効。ボットトラップ挿入 URL を有効または無効にします。
   c) [コメント]。トラップ挿入 URL に関する簡単な説明。

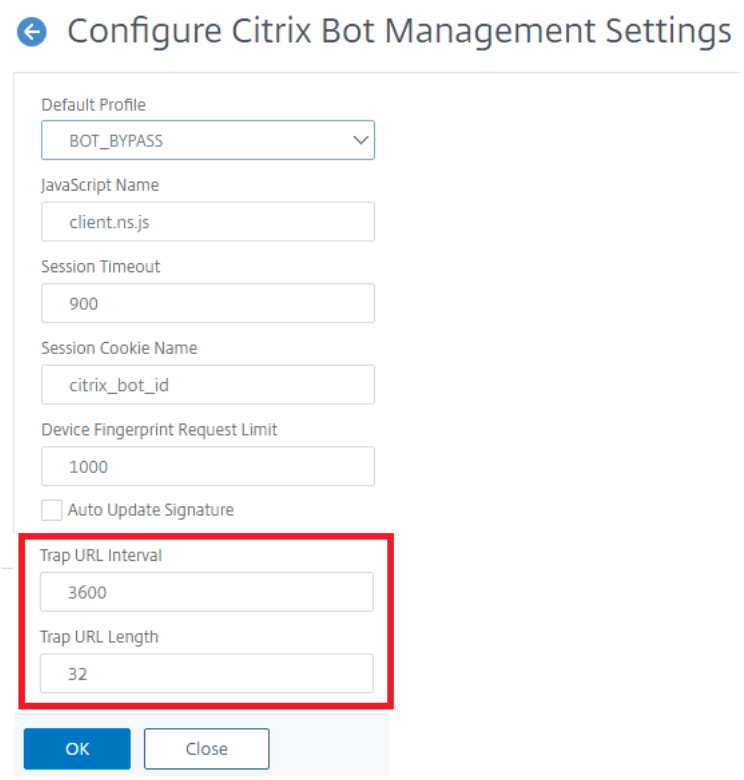


8. [署名の設定] セクションで、[ボットトラップ] をクリックします。
9. [**Bot Trap**] セクションで、次のパラメータを設定します。
   a) 有効。このチェックボックスをオンにすると、ボットトラップ検出が有効になります。
   b) [構成] セクションで、次のパラメータを設定します。
      i. 操作。ボットトラップアクセスによって検出されたボットに対して実行されるアクション。
      ii. ログ。ボットトラップバインディングのロギングを有効または無効にします。
10. [更新して完了] をクリックします。

ボットトラップ **URL** 設定の構成

ボットトラップ URL 設定を構成するには、次の手順を実行します。

1. ［セキュリティ］ **>** ［**Citrix Bot** 管理］に移動します。
2. 詳細ペインの［設定］で、［**Citrix Bot** 管理設定の変更］をクリックします。
3. ［**Citrix Bot** 管理設定の構成］で、次のパラメーターを設定します。
   a) トラップ URL 間隔。ボットトラップ URL が更新されるまでの時間（秒単位）。
   b) トラップ URL の長さ。自動生成されたボットトラップ URL の長さ。
4. [**OK**] をクリックし、[完了] をクリックします。



ボット検出用のクライアント **IP** ポリシー式

Citrix ボット管理では、HTTP リクエストヘッダー、HTTP リクエスト本文、HTTP リクエスト URL、または高度なポリシー式を使用してクライアント IP アドレスを抽出する高度なポリシー式を構成できるようになりました。抽出された値は、ボット検出メカニズム（TPS、ボットトラップ、レート制限など）で使用して、着信要求がボットかどうかを検出できます。

> 注:
>
> クライアント IP 式を設定していない場合は、デフォルトまたは既存の送信元クライアント IP アドレスがボットの検出に使用されます。式が設定されている場合、評価結果には、ボット検出に使用できるクライアント IP アドレスが提供されます。

受信要求がプロキシサーバを介して送信され、クライアントの IP アドレスがヘッダーに存在する場合は、クライアント IP 式を設定して使用して、実際のクライアント IP アドレスを抽出できます。この設定を追加することで、アプライアンスはボット検出メカニズムを使用して、ソフトウェアクライアントとサーバーにより多くのセキュリティを提供できます。

CLI を使用して、ボットプロファイルでクライアント **IP** ポリシー式を設定します

コマンドプロンプトで入力します。

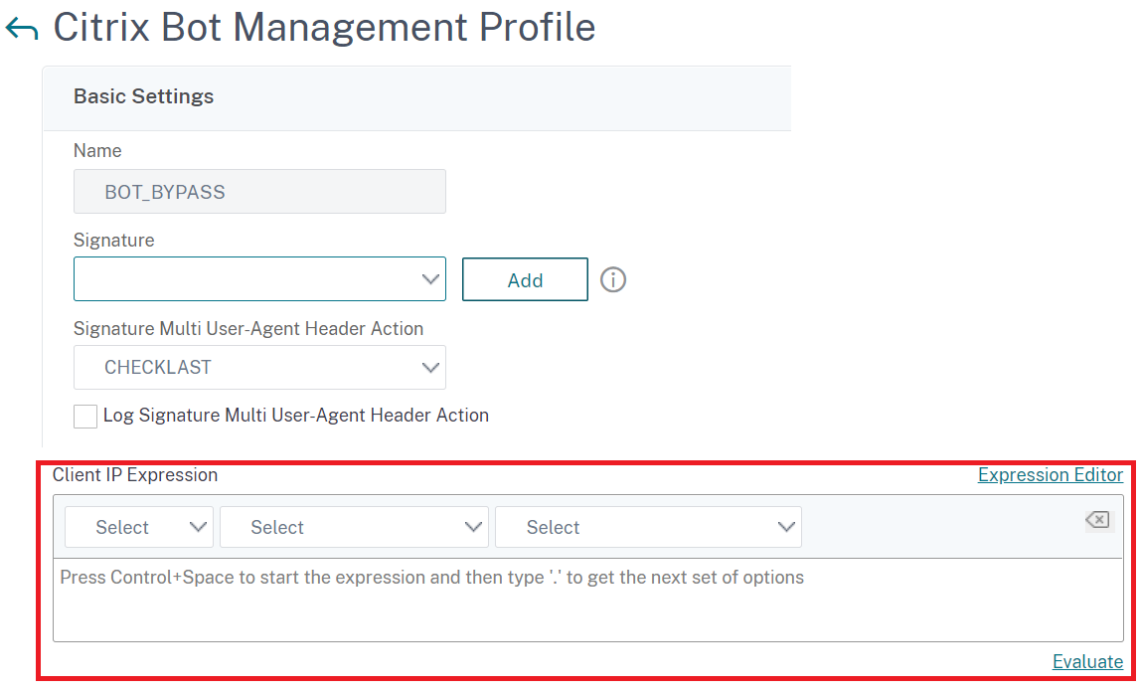

```
add bot profile <name> [-clientIPExpression <expression>]
<!--NeedCopy-->
```

例:

```
add bot profile profile1 -clientIPExpression 'HTTP.REQ.HEADER(”X-Forwarded-For”)ALT CLIENT.IP.SRC.TYPECAST_TEXT_T'
```

GUI を使用してボットプロファイルでクライアント **IP** ポリシー式を構成する

1. ［セキュリティ］ **>** ［**Citrix Bot** 管理］ **>** ［プロファイル］に移動します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. ［**Citrix Bot** 管理プロファイルの作成］ページで、［クライアント IP 式］を設定します。
4. **[**作成**]** して **[**閉じる**]** をクリックします。



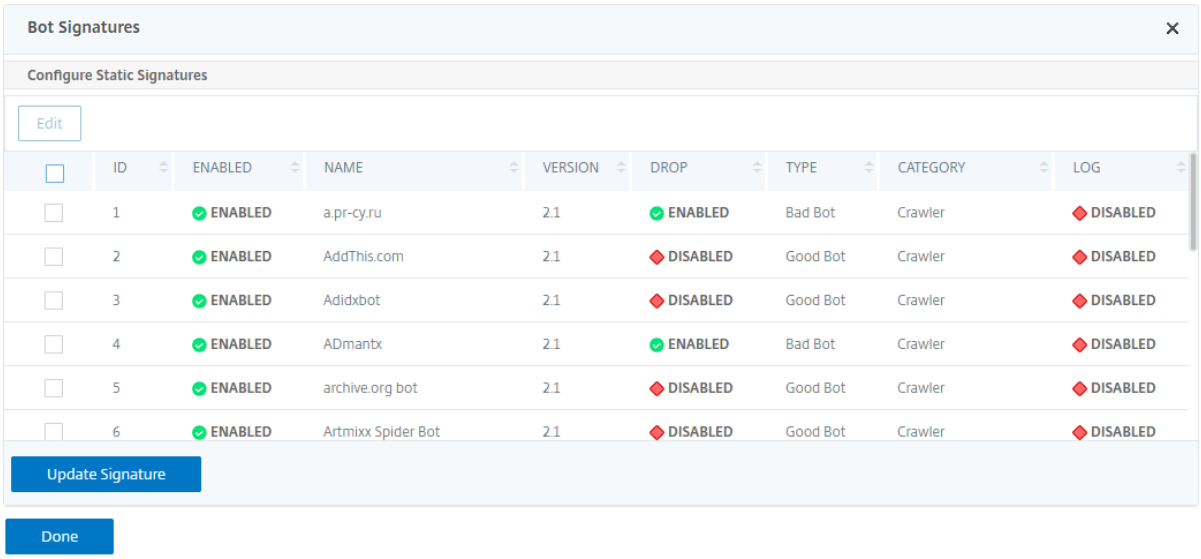
IP レピュテーションとデバイスフィンガープリント検出のための **CAPTCHA** の構成

CAPTCHA は、「Computers and Humans Apart を伝える完全に自動化された Public Turing test」の略の頭字語です。CAPTCHA は、受信トラフィックが人間のユーザーまたは自動ボットからのものかどうかをテストするように設計されています。CAPTCHA は、ウェブアプリケーションにセキュリティ違反を引き起こす自動ボットをブロックするのに役立ちます。ADC アプライアンスでは、CAPTCHA はチャレンジレスポンスモジュールを使用して、着信トラフィックが自動ボットではなく人間のユーザーからのものであるかどうかを識別します。

ボットの静的シグネチャの設定

この検出手法により、ブラウザの詳細からユーザエージェント情報を識別できます。ユーザーエージェント情報に基づいて、ボットは不良または良好なボットとして識別され、ボットアクションを割り当てます。スタティックシグニチャ手法を設定するには、次の手順に従います。

1. ナビゲーションペインで、［セキュリティ］ > ［**Citrix** ボット管理］ > ［署名］の順に展開します。
2. **Citrix Bot** 管理の［署名］ページで、署名ファイルを選択し、［編集］をクリックします。
3. ［**Citrix Bot Management** の署名］ページで、［署名の設定］セクションに移動し、［ボット署名］をクリックします。
4. [ボット署名] セクションで、次のパラメータを設定します。
   a) スタティックシグニチャを設定します。このセクションには、ボットの静的署名レコードのリストがあります。レコードを選択し、[編集] をクリックして、そのレコードにボットアクションを割り当てることができます。
   b) **OK** をクリックします。
5. [署名の更新] をクリックします。
6. ［完了］をクリックします。



ボットの静的シグニチャの描写

Citrix ADC ボット管理は、Web アプリケーションをボットから保護します。ボットの静的シグネチャは、着信リクエストのユーザーエージェントなどのリクエストパラメータに基づいて、良いボットと悪いボットを識別するのに役立ちます。

ファイル内のシグネチャのリストは巨大で、新しいルールが追加され、古いルールが定期的に削除されます。管理者は、カテゴリで特定のシグニチャまたはシグニチャのリストを検索したい場合があります。署名を簡単にフィルタリングするために、ボットシグネチャページには拡張された検索機能があります。検索機能を使用すると、署名のルールを検索し、アクション、署名 ID、開発者、署名名などの 1 つ以上のシグニチャパラメータに基づいてプロパティを構成できます。

操作。シグニチャルールの特定のカテゴリに設定するボットアクションを選択します。使用可能なアクションタイプは次のとおりです。

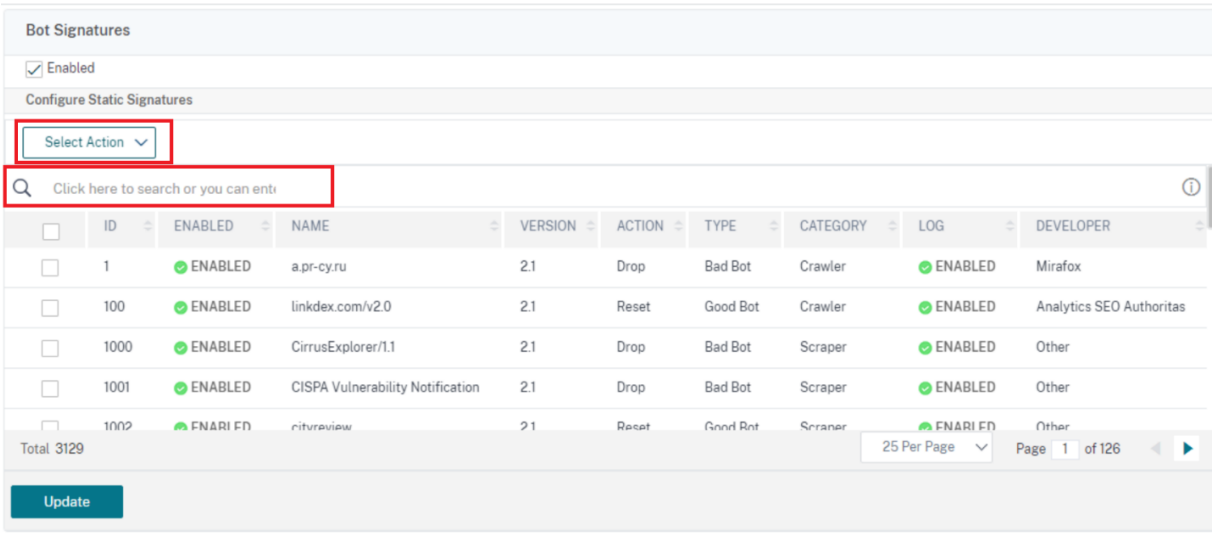
- [選択済み] を有効にします。選択したすべてのシグニチャルールを有効にします。
- [選択を無効にする]。選択したすべてのシグニチャルールを無効にします。
- [選択項目をドロップ]。選択したすべてのシグニチャルールに、アクションを「ドロップ」として適用します。
- 選択をリダイレクト。選択したすべてのシグニチャルールに「リダイレクト」としてアクションを適用します。
- [選択をリセット]。選択したすべてのシグニチャルールに「リセット」としてアクションを適用します。
- ログ選択済み。選択したすべてのシグニチャルールに「Log」としてアクションを適用します。
- 選択したドロップを削除。選択したすべてのシグニチャルールに対するドロップアクションの設定を解除します。
- 選択したリダイレクトを削除。選択したすべてのシグニチャルールへのリダイレクトアクションの設定を解除します。
- [選択をリセット] を削除します。選択したすべてのシグニチャルールに対するリセットアクションの設定を解除します。
- 選択したログを削除。選択したすべてのシグニチャルールに対するログアクションの設定を解除します。

カテゴリー。カテゴリを選択して、それに応じてシグニチャルールをフィルタリングします。次に、シグニチャルールのソートに使用できるカテゴリの一覧を示します。

- 操作。ボットアクションに基づいてソートします。
- カテゴリー。ボットカテゴリに基づいてソートします。
- 開発者。ホスト会社のパブリッシャーに基づいてソートします。
- 有効。有効になっているシグニチャルールに基づいてソートします。
- Id。シグニチャルール ID に基づいてソートします。
- ログ。ロギングが有効になっているシグニチャルールに基づいてソートします。
- Name：シグニチャルール名に基づいてソートします。
- [タイプ]。署名タイプに基づいてソートします。
- バージョン。シグニチャルールのバージョンに基づいてソートします。

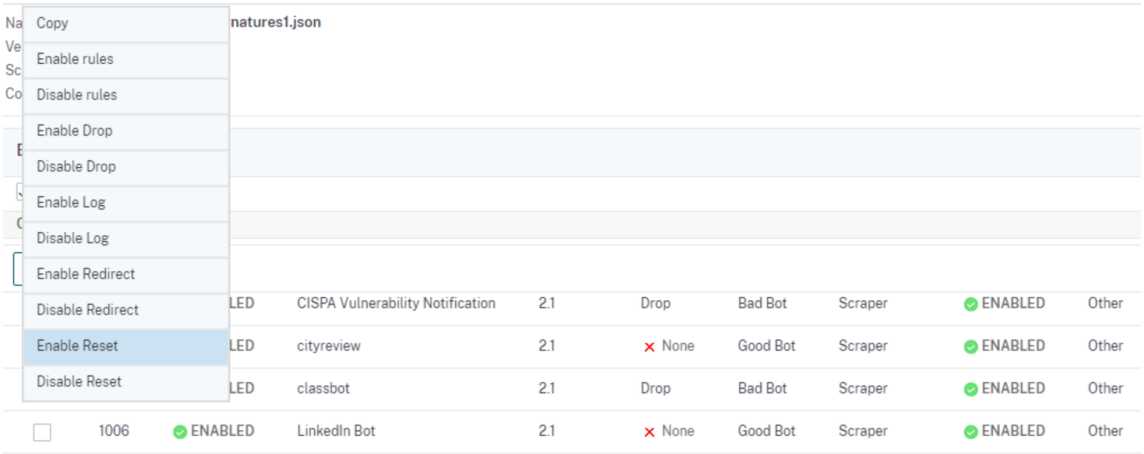
Citrix ADC GUI を使用して、アクションとカテゴリタイプに基づいてボット静的署名ルールを検索します

1. [セキュリティ**] > [Citrix** ボット管理**] > [**署名] に移動します。
2. 詳細ページで、[追加] をクリックします。
3. ［**Citrix Bot Management** 署名］ページで、［静的署名］セクションの［編集］をクリックします。
4. [静的署名の構成] セクションで、ドロップダウンリストからシグニチャアクションを選択します。
5. 検索機能を使用してカテゴリを選択し、それに応じてルールをフィルタリングします。
6. ［更新］をクリックします。



Citrix ADC GUI を使用してボットの静的署名ルールのプロパティを編集する

1. [セキュリティ] **> [Citrix** ボット管理] **> [**署名] に移動します。
2. 詳細ページで、[追加] をクリックします。
3. [**Citrix Bot Management** 署名] ページで、[静的署名] セクションの [編集] をクリックします。
4. [静的署名の構成] セクションで、ドロップダウンリストからアクションを選択します。
5. 検索機能を使用して、カテゴリを選択し、それに応じてルールをフィルタします。
6. 静的署名リストから、シグニチャを選択してプロパティを変更します。



7. [**OK**] をクリックして確定します。

Citrix ADC ボット管理における **CAPTCHA** の仕組み

Citrix ADC ボット管理では、CAPTCHA 検証は、ボットポリシーが評価された後に実行されるポリシーアクションとして構成されます。CAPTCHA アクションは、IP レピュテーションおよびデバイスフィンガープリント検出技術でのみ使用できます。以下に、CAPTCHA の仕組みを理解する手順を示します。

1. IP レピュテーションまたはデバイスフィンガープリントボットの検出中にセキュリティ違反が観察された場合、ADC アプライアンスは CAPTCHA チャレンジを送信します。
2. クライアントは CAPTCHA レスポンスを送信します。
3. アプライアンスは CAPTCHA 応答を検証し、CAPTCHA が有効な場合、要求は許可され、バックエンドサーバーに転送されます。
4. CATCHA 応答が無効な場合、アプライアンスは最大試行回数に達するまで新しい CAPTCHA チャレンジを送信します。
5. 最大試行回数の後でも CAPTCHA 応答が無効である場合、アプライアンスは要求をドロップするか、設定されたエラー URL にリダイレクトします。
6. ログアクションを設定した場合、アプライアンスは要求の詳細を ns.log ファイルに保存します。

Citrix ADC GUI を使用してキャプチャ設定を構成する

ボット管理 CAPTCHA アクションは、IP レピュテーションおよびデバイスフィンガープリント検出技術でのみサポートされます。**CAPTCHA** 設定を構成するには、次の手順を実行します。

1. ［セキュリティ］**>**［**Citrix Bot** の管理とプロファイル］に移動します。
2. ［**Citrix Bot** 管理プロファイル］ページでプロファイルを選択し、［編集］をクリックします。
3. ［**Citrix Bot** 管理プロファイル］ページで、［署名の設定］セクションに移動し、［**CAPTCHA**］をクリックします。
4. 「**CAPTCHA** 設定」セクションで、「追加」をクリックして、プロファイルに **CAPTCHA** 設定を構成します。
5. ［**Citrix Bot** 管理 **CAPTCHA** の構成］ページで、次のパラメーターを設定します。
    a) URL。IP レピュテーションおよびデバイスフィンガープリント検出技術中に CAPTCHA アクションが適用されるボット URL。
    b) 有効。CAPTCHA サポートを有効にするには、このオプションを設定します。
    c) 猶予時間。現在の有効な CAPTCHA 応答を受信した後、新しい CAPTCHA チャレンジが送信されないまでの期間。
    d) 待ち時間。ADC アプライアンスがクライアントが CAPTCHA 応答を送信するまで待機するまでに要した時間。
    e) ミュート期間。不正な CAPTCHA 応答を送信したクライアントが、次の試行を許可されるまで待つ必要がある期間。このミュート期間中、ADC アプライアンスは要求を許可しません。範囲:60 ～900 秒、推奨:300 秒

f) リクエストの長さ制限。CAPTCHA チャレンジがクライアントに送信される要求の長さ。長さがしきい値より大きい場合、要求はドロップされます。デフォルト値は 10 ～3000 バイトです。

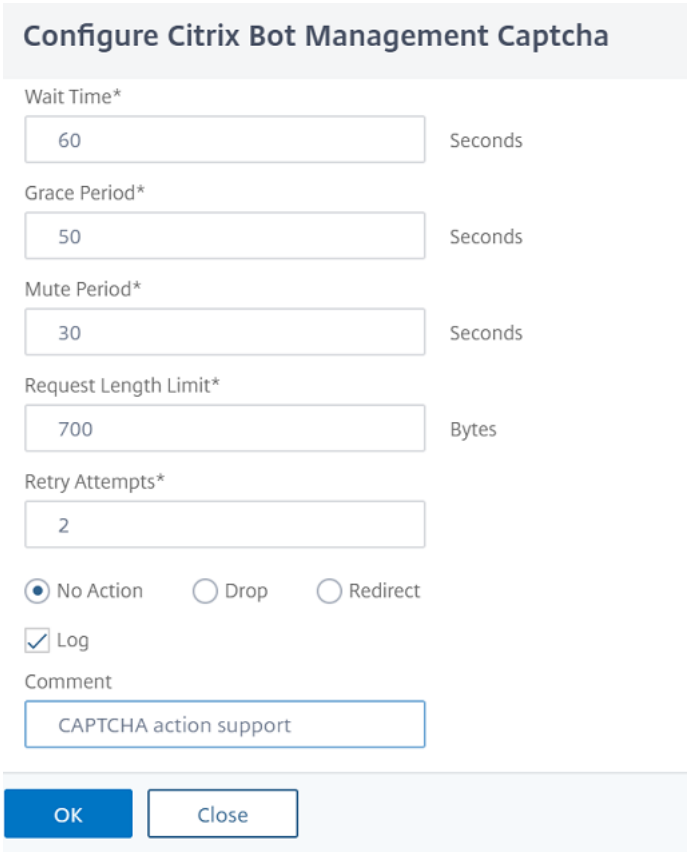
g) 再試行回数。クライアントが CAPTCHA チャレンジを解決するために再試行できる試行回数。範囲:1—10、推奨:5。

h) クライアントが CAPTCHA 検証に失敗した場合、アクション/ドロップ/リダイレクトアクションは実行されません。

i) ログ。このオプションを設定すると、レスポンス CAPTCHA が失敗したときにクライアントからのリクエスト情報を保存できます。データは`ns.log`ファイルに格納されます。

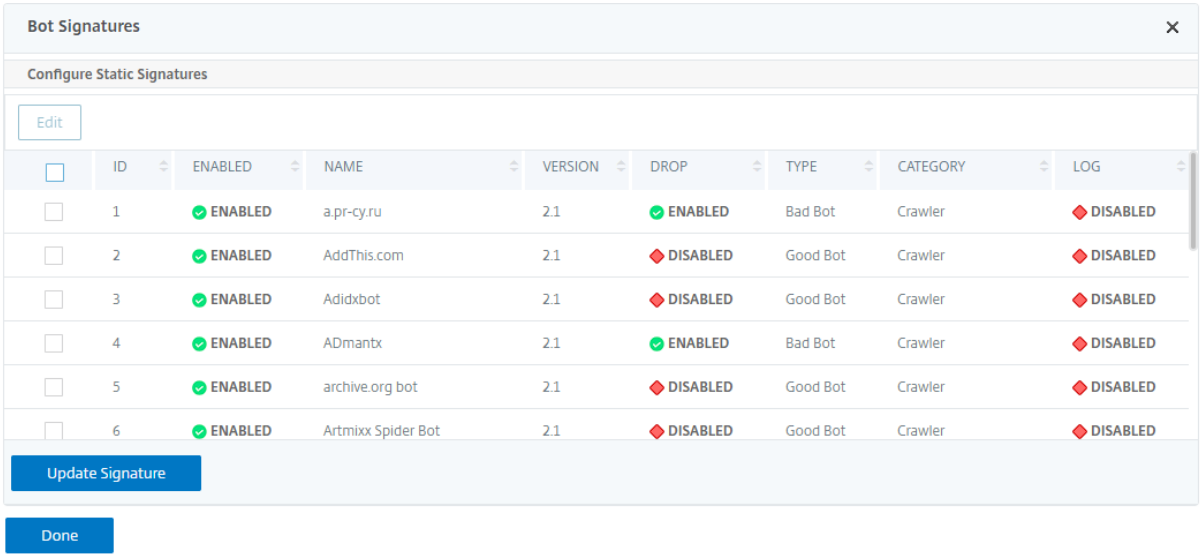
j) [コメント]。CAPTCHA 設定に関する簡単な説明。

6. [**OK**] をクリックし、[完了] をクリックします。



7. [セキュリティ] **> [Citrix Bot** 管理] **> [**署名] に移動します。
8. **Citrix Bot** 管理の［署名］ページで、署名ファイルを選択し、［編集］をクリックします。
9. [**Citrix Bot Management** の署名］ページで、［署名の設定］セクションに移動し、［ボット署名］をクリックします。
10. [ボット署名] セクションで、次のパラメータを設定します。

11. スタティックシグニチャを設定します。ボット静的署名レコードを選択し、[編集] をクリックして、ボットアクションをそのレコードに割り当てます。
12. **OK** をクリックします。
13. [署名の更新] をクリックします。
14. [完了] をクリックします。



ボット署名の自動更新

ボット静的シグニチャ手法では、シグニチャルックアップテーブルと良いボットと不良ボットのリストを使用します。ボットは、ユーザーエージェント文字列とドメイン名に基づいて分類されます。着信ボットトラフィックのユーザーエージェント文字列とドメイン名がルックアップテーブルの値と一致する場合、設定されたボットアクションが適用されます。
ボット署名の更新は AWS クラウドでホストされ、署名ルックアップテーブルは、署名の更新のために AWS データベースと通信します。自動署名更新スケジューラーは 1 時間ごとに実行され、**AWS** データベースをチェックし、Citrix ADC アプライアンスの署名テーブルを更新します。

設定するシグニチャ自動更新 URL は、`https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`

> 注:
>
> また、プロキシサーバーを設定し、プロキシを介して AWS クラウドからアプライアンスにシグネチャを定期的に更新することもできます。プロキシ設定の場合、ボット設定でプロキシ IP アドレスとポートアドレスを設定する必要があります。

ボット署名の自動更新の仕組み

次の図は、ボット署名が AWS クラウドから取得され、Citrix ADC で更新され、Citrix ADM で署名更新の概要が表示される方法を示しています。

ボットシグネチャ自動更新スケジューラは、次のことを行います。

1. AWS URI からマッピングファイルを取得します。
2. マッピングファイル内の最新のシグニチャを、ADC アプライアンスの既存のシグニチャでチェックします。
3. AWS から新しい署名をダウンロードし、署名の整合性を検証します。
4. 既存のボット署名を、ボット署名ファイル内の新しい署名で更新します。
5. SNMP アラートを生成し、署名の更新の概要を Citrix ADM に送信します。

ボット署名の自動更新を設定する

ボット署名の自動更新を設定するには、次の手順を実行します。

ボット署名の自動更新を有効にする

ADC アプライアンスのボット設定で自動更新オプションを有効にする必要があります。
コマンドプロンプトで入力します。

```
set bot settings -signatureAutoUpdate ON
```

プロキシサーバーの設定を構成する (オプション)

プロキシサーバーを介して AWS 署名データベースにアクセスする場合は、プロキシサーバーとポートを設定する必要があります。

```
set bot settings -proxyserver -proxyport
```

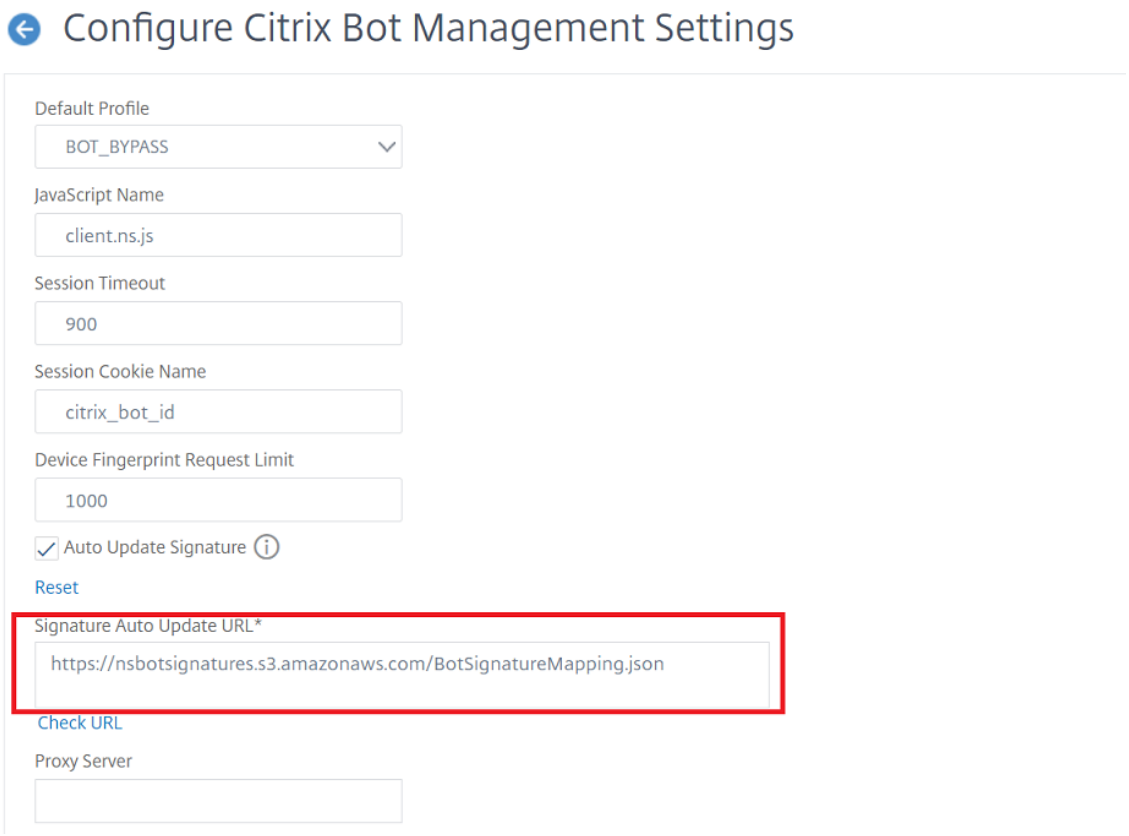
例:

```
set bot settings -proxy server 1.1.1.1 -proxyport 1356
```


Citrix ADC GUI を使用してボット署名の自動更新を構成する

ボット署名の自動更新を設定するには、次の手順を実行します。

1. ［セキュリティ］**>**［**Citrix Bot** 管理］に移動します。
2. 詳細ペインの［設定］で、［**Citrix Bot** 管理設定の変更］をクリックします。
3. ［**Citrix Bot** 管理設定の構成］で、［署名の自動更新］チェックボックスをオンにします。



4. ［**OK**］をクリックして閉じます。

ボット管理プロファイルの作成

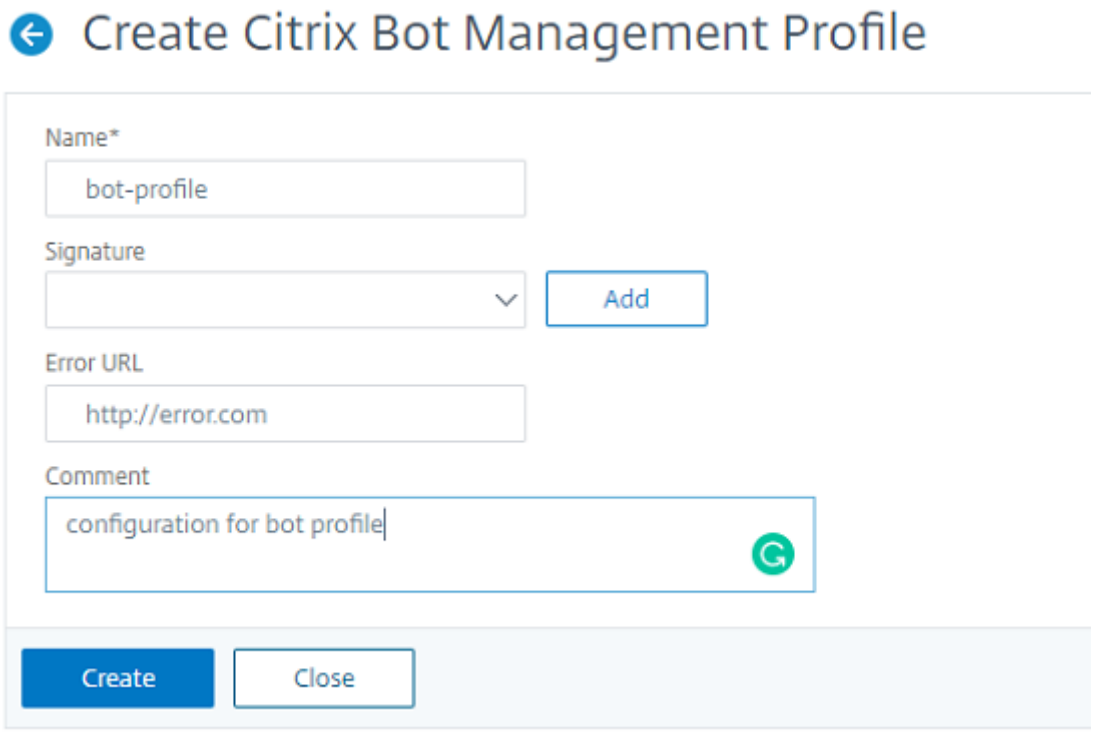
ボットプロファイルは、ボットタイプの検出に使用されるボット管理設定の集まりです。プロファイルでは、Web App Firewall が各フィルタ（またはチェック）を Web サイトへのボットトラフィックに適用する方法と、それらからの応答を決定します。

ボットプロファイルを設定するには、次の手順を実行します。

1. ［セキュリティ］>［**Citrix Bot** 管理］>［プロファイル］に移動します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. ［**Citrix Bot** 管理プロファイルの作成］ページで、次のパラメーターを設定します。

a) Name：ボットプロファイル名。
b) 署名。ボット署名ファイルの名前。
c) エラー URL。リダイレクトの URL。
d) [コメント]。プロファイルに関する簡単な説明。

4. **[**作成**]** して **[**閉じる**]** をクリックします。

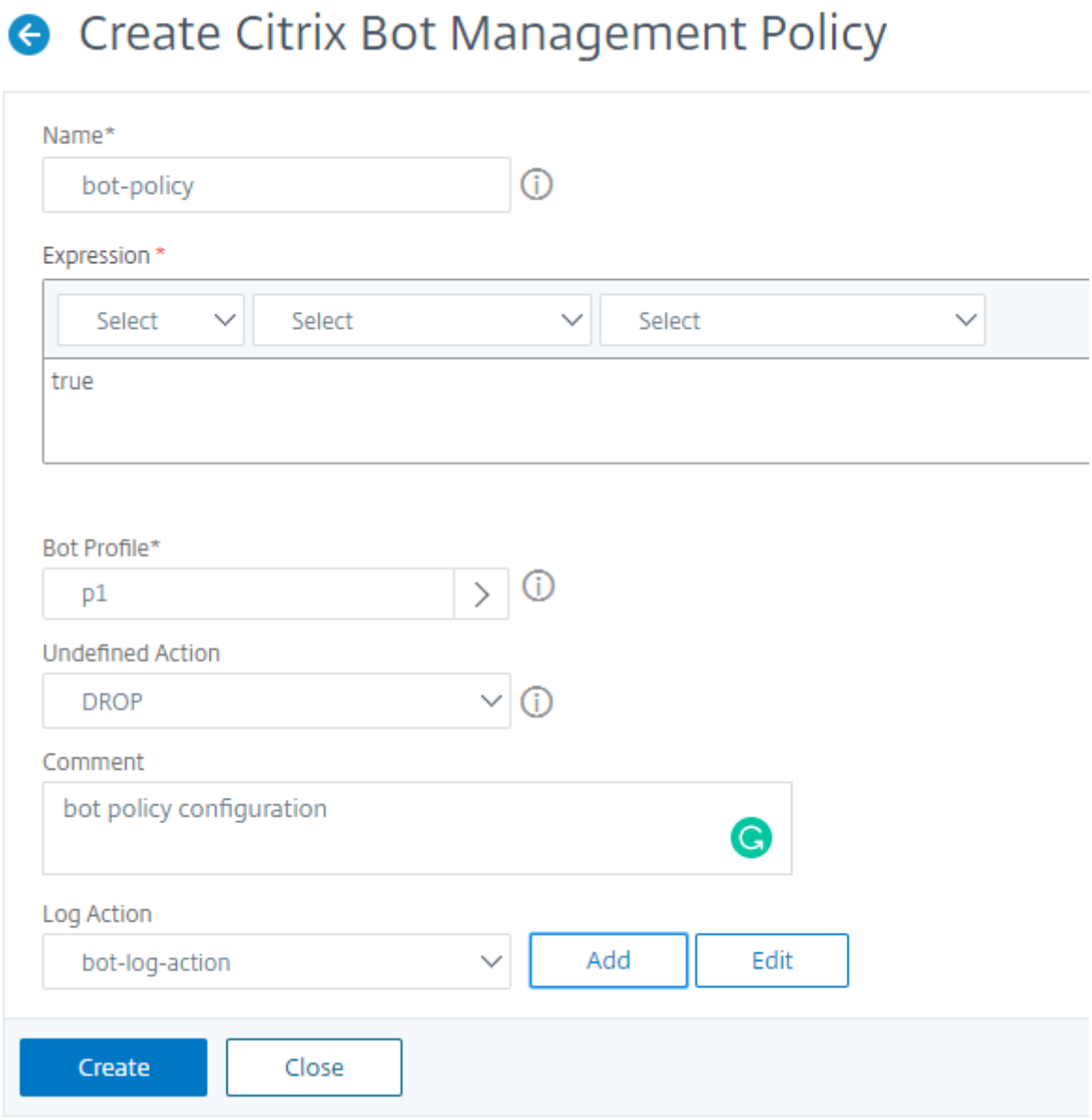


ボットポリシーの作成

ボットポリシーは、ボット管理システムへのトラフィックを制御し、監査ログサーバーに送信されるボットログも制御します。手順に従って、ボットポリシーを設定します。

1. [セキュリティ] > [**Citrix** ボット管理] > [ボットポリシー] に移動します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. [**Citrix Bot** 管理ポリシーの作成］ページで、次のパラメーターを設定します。
   a) Name：ボットポリシーの名前。
   b) 式。ポリシー式またはルールをテキスト領域に直接入力します。
   c) ボットプロファイル。ボットポリシーを適用するボットプロファイル。
   d) 未定義のアクション。割り当てるアクションを選択します。
   e) [コメント]。ポリシーに関する簡単な説明。
   f) アクションをログに記録します。ボットトラフィックを記録するための監査ログメッセージアクション。監査ログアクションの詳細については、「監査ログ」のトピックを参照してください。

4. [作成] して [閉じる] をクリックします。



1 秒あたりのボットトランザクション **(TPS)**

1 秒あたりのトランザクション数 (TPS) ボット技術は、1 秒あたりの要求数 (RPS) と RPS の増加率が設定されたしきい値を超えた場合に、着信トラフィックをボットとして検出します。この検出技術は、ウェブスクレイピングアクティビティ、ブルートフォースログイン、その他の悪意のある攻撃を引き起こす可能性のある自動ボットから Web アプリケーションを保護します。

> 注:
>
> ボット技術は、両方のパラメータが設定されていて、両方の値がしきい値制限を超えて増加した場合にのみ、着

> 信トラフィックをボットとして検出します。
> アプライアンスが特定の URL から多数の要求を受信し、Citrix ADC ボット管理がボット攻撃があるかどうかを検出するシナリオを考えてみましょう。TPS 検出技術は、1 秒以内に URL から送信された要求の数（設定値）と 30 分以内に受信された要求数の増加率（設定値）を調べます。値がしきい値制限を超えると、トラフィックはボットと見なされ、アプライアンスは設定されたアクションを実行します。

1 秒あたりのボットトランザクション **(TPS)** 手法の設定

TPS を設定するには、次の手順を完了する必要があります。

1. ボットの TPS を有効にする
2. TPS 設定をボット管理プロファイルにバインドする

TPS 設定をボット管理プロファイルにバインドする

ボット TPS 機能を有効にしたら、TPS 設定をボット管理プロファイルにバインドする必要があります。

コマンドプロンプトで入力します。

```
bind bot profile <name>… (-tps [-type ( SourceIP | GeoLocation | RequestURL
 | Host )] [-threshold <positive_integer>] [-percentage <positive_integer
>] [-action ( none | log | drop | redirect | reset | mitigation )] [-
logMessage <string>])
```

例**:**

```
bind bot profile profile1 -tps -type RequestURL -threshold 1 -percentage
100000 -action drop -logMessage log
```

1 秒あたりのボットトランザクション **(TPS)** を有効にする

開始する前に、アプライアンスでボット TPS 機能が有効になっていることを確認する必要があります。コマンドプロンプトで入力します。

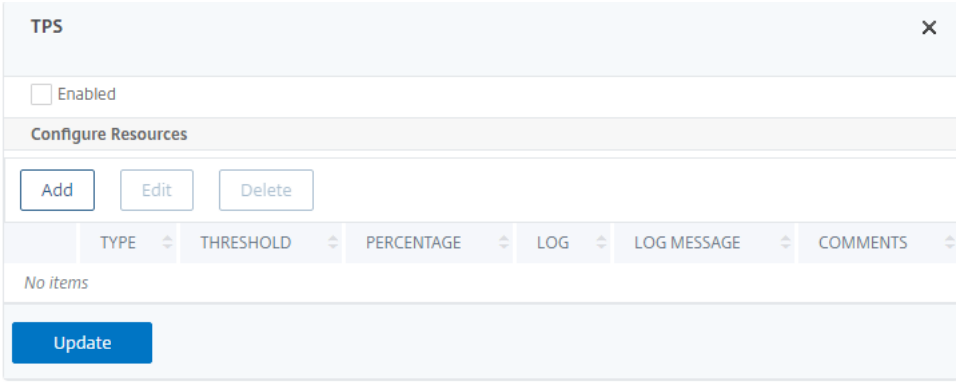
```
set bot profile profile1 -enableTPS ON
```

Citrix ADC GUI を使用してボットトランザクション**/**秒（**TPS**）を構成する

1 秒あたりのボットトランザクションを設定するには、次の手順を実行します。

1. ［セキュリティ］**>**［**Citrix Bot** 管理］**>**［プロファイル］に移動します。
2. ［**Citrix Bot** 管理プロファイル］ページで、プロファイルを選択して［編集］をクリックします。
3. ［**Citrix Bot** 管理プロファイルの作成］ページで、［署名の設定］セクションの［**TPS** ］をクリックします。

4. [**TPS**] セクションで、機能を有効にし、[追加] をクリックします。



5. [**Citrix Bot** 管理プロファイルの **TPS** バインドの構成] ページで、次のパラメーターを設定します。

   a) [タイプ]。検出手法で許可される入力タイプ。指定可能な値: ソース IP、ジオロケーション、ホスト、URL。

      SOURCE_IP — クライアント IP アドレスに基づいた TPS。

      GEOLOCATION — クライアントの地理的位置に基づく TPS。

      HOST-特定のバックエンドサーバーの IP アドレスに転送されたクライアント要求に基づく TPS。

      URL — 特定の URL からのクライアント要求に基づく TPS。

   b) 固定スレッショルド。1 秒の間隔内に TPS 入力タイプから許可される要求の最大数。

   c) パーセンテージしきい値。30 分以内の TPS 入力タイプからのリクエストの最大増加率。

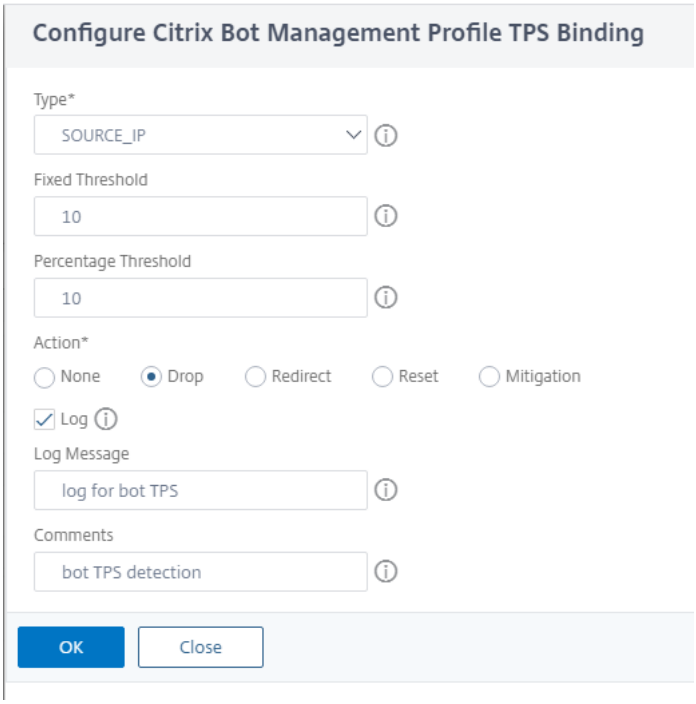
   d) 操作。TPS バインディングによって検出されたボットに対して実行されるアクション。

   e) ログ。TPS バインディングのログを有効または無効にします。

   f) ログメッセージ。TPS バインディングによって検出されたボットのログを記録するメッセージです。最大長:255

   g) コメント。TPS 設定に関する簡単な説明。最大長: 255

6. **OK**] をクリックし、[閉じる] をクリックします。



マウスとキーボードのダイナミクスに基づくボット検出

ボットを検出し、ウェブスクレイピングの異常を軽減するために、Citrix ADC ボット管理では、マウスとキーボードの動作に基づいて強化されたボット検出技術を使用します。人間の直接的なインタラクション（CAPTCHA 検証など）を必要とする従来のボット手法とは異なり、強化された手法はマウスとキーボードのダイナミクスを受動的に監視します。Citrix ADC アプライアンスは、リアルタイムのユーザーデータを収集し、人間とボットの間の行動を分析します。

マウスとキーボードのダイナミクスを用いたパッシブボット検出は、既存のボット検出メカニズムに比べて次のような利点があります。

- ユーザー・セッション全体にわたって継続的な監視を提供し、単一のチェックポイントを排除します。
- 人間のやりとりを必要とせず、ユーザーに対して完全に透過的です。

マウスとキーボードのダイナミクスを使用したボット検出の仕組み

キーボードとマウスのダイナミクスを使用したボット検出技術は、ウェブページロガーとボット検出器の 2 つのコンポーネントで構成されています。ウェブページロガーは、ユーザーがウェブページでタスクを実行しているときのキーボードとマウスの動きを記録する JavaScript です（登録フォームへの入力など）。その後、ロガーはデータをバッチで Citrix ADC アプライアンスに送信します。その後、アプライアンスはデータを KM レコードとして保存し、Citrix ADM サーバーのボットディテクタに送信し、ユーザーが人間かボットかを分析します。

次の手順では、コンポーネントがどのように相互作用するかを説明します。

1. Citrix ADC 管理者は、ADM StyleBook、CLI、または NITRO、またはその他の方法でポリシー式を構成します。
2. URL は、管理者がアプライアンスで機能を有効にすると、ボットプロファイルで設定されます。
3. クライアントが要求を送信すると、Citrix ADC アプライアンスはセッションとセッション内のすべての要求を追跡します。
4. リクエストがボットプロファイルで設定された式と一致する場合、アプライアンスはレスポンスに JavaScript（ウェブページロガー）を挿入します。
5. JavaScript はすべてのキーボード、マウスアクティビティを収集し、KM データを POST URL (一時的) に送信します。
6. Citrix ADC アプライアンスはデータを格納し、セッションの最後に Citrix ADM サーバーに送信します。アプライアンスが POST 要求の完全なデータを受信すると、そのデータは ADM サーバに送信されます。
7. Citrix ADM サービスはデータを分析し、その分析に基づいて Citrix ADM サービス GUI で結果を利用できます。

JavaScript ロガーは、次のマウスとキーボードの動きを記録します。

- キーボードイベント — すべてのイベント
- マウスイベント-マウスの移動、マウスアップ、マウスダウン
- クリップボードイベント-貼り付け
- カスタムイベント-オートフィル、オートフィルキャンセル
- 各イベントのタイムスタンプ

マウスとキーボードのダイナミクスを使用してボット検出を構成する

Citrix ADC ボット管理構成には、キーボードおよびマウスベースの検出機能の有効化または無効化が含まれ、ボットプロファイルで JavaScript URL を構成します。
マウスとキーボードのダイナミクスを使用してボットの検出を設定するには、次の手順を実行します。

1. キーボードとマウスベースの検出を有効にする
2. JavaScript を HTTP レスポンスに挿入できるタイミングを決定する式を設定します。

キーボードマウスベースのボット検出を有効にする

設定を開始する前に、アプライアンスでキーボードおよびマウスベースのボット検出機能が有効になっていることを確認します。

コマンドプロンプトで入力します。

```
add bot profile <name> -KMDetection ( ON | OFF )
<!--NeedCopy-->
```

例:


```
add bot profile profile1 -KMDetection ON
```

JavaScript 挿入用のボット式を構成する

トラフィックを評価し、JavaScript を挿入するボット式を設定します。JavaScript は、式が true と評価された場合にのみ挿入されます。

コマンドプロンプトで入力します。

```
1 bind bot profile <name> -KMDetectionExpr -name <string> -expression <
     expression> -enabled ( ON | OFF ) - comment <string>
2 <!--NeedCopy-->
```

例:

```
bind bot profile profile1 -KMDetectionExpr -name test -expression http.req.
url.startswith(”/testsite”)-enabled ON
```

キーボードマウスベースのボット検出のために **HTTP** レスポンスに挿入された **JavaScript** ファイル名を設定する

ユーザーアクションの詳細を収集するために、アプライアンスは HTTP レスポンスに JavaScript ファイル名を送信します。JavaScript ファイルは KM レコードのすべてのデータを収集し、アプライアンスに送信します。

コマンドプロンプトで入力します。

```
1 set bot profile profile1 - KMJavaScriptName <string>
2 <!--NeedCopy-->
```

例:

```
set bot profile profile1 -KMJavaScriptName script1
```

ビヘイビアバイオメトリクスサイズの構成

KM レコードとしてアプライアンスに送信し、ADM サーバで処理できるマウスおよびキーボードの動作データの最大サイズを設定できます。

コマンドプロンプトで入力します。

```
1 set bot profile profile1 -KMEventsPostBodyLimit <positive_integer>
2 <!--NeedCopy-->
```

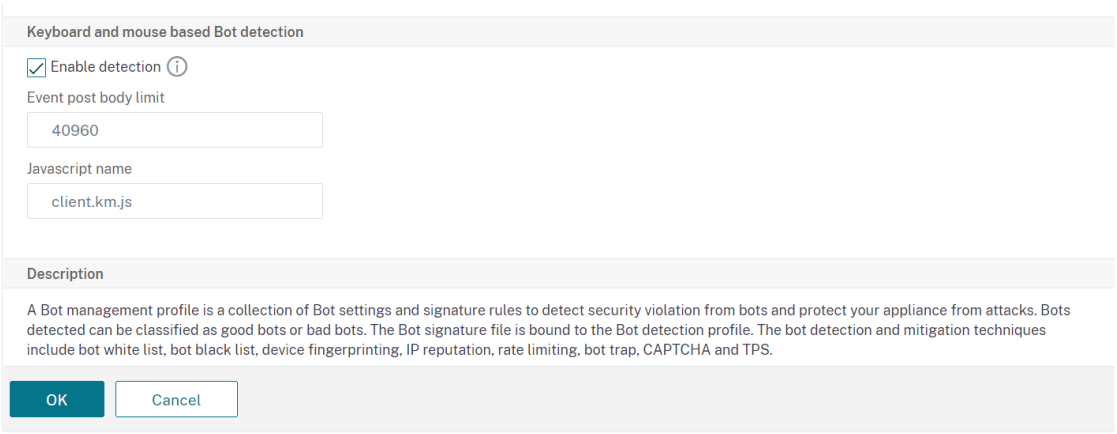

例:

```
set bot profile profile1 - KMEventsPostBodyLimit 25
```

JavaScript を構成し、キーボードとマウスの動作バイオメトリクスを収集するように Citrix ADC アプライアンスを構成すると、アプライアンスはデータを Citrix ADM サーバーに送信します。Citrix ADM サーバーが行動バイオメトリクスからボットを検出する方法の詳細については、「ボット違反」トピックを参照してください。

GUI を使用してキーボードとマウスのボットの式設定を構成する

1. [セキュリティ] > [**Citrix Bot** の管理とプロファイル] に移動します。
2. [**Citrix Bot** 管理プロファイル] ページでプロファイルを選択し、[編集] をクリックします。
3. [キーボードとマウスベースのボットの検出] セクションで、次のパラメータを設定します。
   a) 検出を有効にします。このチェックボックスをオンにすると、ボットベースのキーボードとマウスのダイナミクスの動作が検出されます。
   b) イベント投稿本体の制限。Citrix ADC アプライアンスによって処理されるブラウザから送信されるキーボードおよびマウスのダイナミクスデータのサイズ。
4. **OK** をクリックします。

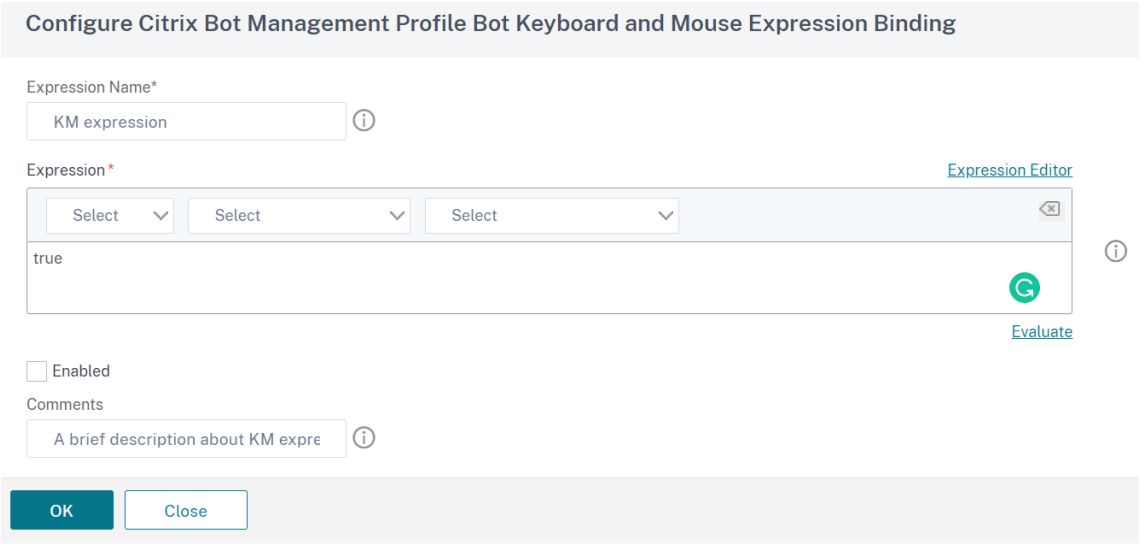


5. [**Citrix Bot Management** プロファイル] ページで、[プロファイル設定] セクションに移動し、[キーボードおよびマウスベースのボット式の設定] をクリックします。
6. [キーボードとマウスベースのボットの式の設定] セクションで、[追加] をクリックします。
7. [**Citrix Bot Management** プロファイルのボットのキーボードとマウス式のバインドの構成] ページで、次のパラメータを設定します。
   a) エクスプレッション名。検出キーボードとマウスのダイナミクスのボットポリシー式の名前。
   b) 式。ボットポリシー式。
   c) 有効。このチェックボックスをオンにすると、キーボードとボットのキーボードとマウスの式のバインドが有効になります。

d) コメント。ボットポリシー式とボットプロファイルへのバインドについての簡単な説明。

e) [**OK**] をクリックして閉じます。

8. [キーボードとマウスベースのボット式の設定] セクションで、[更新] をクリックします。



Citrix Bot Management によってドロップされたリクエストヘッダー

キャッシュに関連するリクエストヘッダーの多くは、セッションのコンテキスト内のすべてのリクエストを表示するためにドロップされます。同様に、Web サーバーが圧縮された応答を送信できるようにするエンコードヘッダーが要求に含まれている場合、ボット管理はこのヘッダーを削除して、圧縮されていないサーバー応答の内容がボット管理者によって検査されて JavaScript が挿入されるようにします。

ボット管理者は以下のリクエストヘッダーを削除します。

[範囲]。失敗したファイル転送または部分的なファイル転送からの回復に使用されます。

If-Range。クライアントは、そのオブジェクトの一部が既にキャッシュに含まれている場合に、そのオブジェクトの一部を取得できます (条件付き GET)。

修正された場合-以降。要求されたオブジェクトがこのフィールドに指定された時間以降に変更されない場合、エンティティはサーバーから返されません。HTTP 304 変更されていないというエラーが表示されます。

一致しない場合。最小限のオーバーヘッドで、キャッシュされた情報を効率的に更新できます。

受け入れ-エンコーディング。特定のオブジェクト (gzip など) に許可されるエンコーディング方法。

ボット管理

October 7, 2021

Citrix ADC ボットの管理で説明されているトラブルシューティングのシナリオの一部を以下に示します。

1. 偽陽性のケースを処理する方法は?

   ボットの許可リスト機能を使用して誤検知のケースを管理でき、これらのトランザクションをバイパスできます。

2. どのように悪いボットトラフィックの詳細を見つけるには?

   監査ログ機能を使用して、不良ボットとして分類されたトラフィックの詳細を取得できます。

3. デフォルトのシグニチャ名を変更する必要があるのはなぜですか。

   Citrix ADC アプライアンスが提供するエンドポイントリソースで競合が検出された場合は、デフォルトの署名名を変更できます。

ボット管理

October 7, 2021

1. Citrix ADC ボット管理とは何ですか?

   Citrix ADC ボット管理は、トラフィックを検出し、良いボット、不良ボット、および人間のクライアントと区別します。ボット管理機能は、不良ボットからの接続を切断することにより、Web アプリケーションを保護します。

2. Citrix ADC が Web アプリケーションのボットを管理する必要があるのはなぜですか?

   悪意のあるボットはあなたのインターネットトラフィックの 30% を構成します。悪意のあるボットは、DoS 攻撃の開始、電子メールアドレスのスパム、ダウンローダプログラムを使用したアプリケーションの速度低下、Web サイトからのコンテンツのダウンロードなど、さまざまな方法で Web アプリケーションに影響を与えます。さらに、ボットは、よく知られた検出メカニズムの一部を簡単に迂回し、データの損失、収益、および組織への評判につながります。

3. 着信ボットを検出するために使用される技術は何ですか?

   アプライアンスは、IP レピュテーション、レート制限、デバイスフィンガープリント技術などの検出技術を使用します。さらに、Citrix ADC GUI でカスタマイズされたブロックリストを構成して、組織固有の不良ボットを分類できます。

4. ボット署名ファイルとその目的は何ですか?

   ボット署名ファイルには、既知の良いボットと悪いボットのフットプリントが含まれています。署名ファイルは、ボットの保護を強化するために、最新のボット署名を含めるように定期的に更新することができます。

5. どのような種類の Citrix ADC ライセンスを購入する必要がありますか?

   ボット管理は ADC Premium ライセンスで利用できます。

6. トラブルシューティング用のボットログはどこで確認できますか?

   Citrix ADC 監査ログには、検出されたボットの詳細が表示されます。詳細については、「監査ログ 」トピックを参照してください。

7. ボット署名ファイルの自動更新機能はありますか?

   現在、Citrix ADC は自動更新機能をサポートしていません。

8. ボット IP レピュテーション手法を使用するための前提条件はありますか?

   ボットプロファイルで IP レピュテーションを有効にして構成する前に、IP レピュテーション機能を有効にします。

ボット署名の自動更新

April 7, 2022

Bot シグニチャの自動更新機能により、最新のシグネチャを取得して、善良なボットと悪いボットの両方からより優れた保護とトラフィック管理を実現できます。

署名は 1 時間ごとに自動更新されるため、最新の更新が利用可能かどうかを常に確認する必要がなくなります。署名の自動更新機能を有効にした場合、Citrix ADC アプライアンスは署名をホストしているサーバーに接続して、新しいバージョンが利用可能かどうかを確認します。

Amazon クラウドでホストされている最新のボット署名は、最新の更新を確認するためのデフォルトの署名 URL として設定されます。自動更新機能を使用するには、DNS サーバーが外部サイトにアクセスするように構成する必要もあります。

署名の更新

ボットのデフォルト署名オブジェクトを使用して作成されたすべてのユーザー定義署名オブジェクトには、0 より大きいバージョンがあります。署名の自動更新を有効にすると、すべての署名が自動的に更新されます。Citrix ADC ボット管理 GUI の検索機能を使用して署名または署名のグループを選択することで、ボット署名のデフォルトアクションを更新できます。

ボット署名更新 URL: `https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`

シグニチャ自動更新を構成する

シグニチャ自動更新機能を有効にするには、次のコマンドを実行する必要があります。

コマンドプロンプトで入力します。


```
set bot settings SignatureAutoUpdate ON
<!--NeedCopy-->
```

ボット署名アラート記事

October 7, 2021

Citrix ボット管理は、アプライアンスにダウンロードして適用できる署名の更新を発表します。ボット攻撃を検出すると、新しい署名の更新に関する電子メール通知を受け取ります。署名をダウンロードして、アプライアンスに適用できます。

2020 年 11 月のボット署名の更新

October 7, 2021

2020-11-11 週に特定されたボットに対して、新しい署名ルールが生成されます。これらの署名ルールをダウンロードして構成し、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

Citrix ADC13.0 プラットフォームに適用可能な署名バージョン 5。

新しいボットインサイト

以下は、ボット署名ルール、カテゴリ、およびそのタイプのリストです。

カテゴリ	ボットタイプ	署名数
スクレーパー	グッドボット	3
マーケティング	グッドボット	23
フィードフェッチャー	グッドボット	2
ツール	悪いボット	3
検索エンジン	グッドボット	34
昇降補助具	グッドボット	6
未分類	悪いボット	6

カテゴリ	ボットタイプ	署名数
ウイルススキャナー	グッドボット	1
スクリーンショットクリエーター	グッドボット	7
スクレーパー	悪いボット	1
ツール	グッドボット	7

Bot signature update for January 2021

December 7, 2021

Some of existing bot signatures are updated. You can download and configure these signature rules to protect your appliance from bot attacks.

Bot signature version

Signature version 6 is applicable for Citrix ADC platforms with 13.0 61.x builds or later.

Updated bot signatures in this version

Following is a list of bot signature rule IDs, category and its type.

Bot signature rule	ID	Description
143	Crawler	Good Bot
561	Scraper	Good Bot
857	Site Monitor	Good Bot
892	Site Monitor	Bad Bot
894	Site Monitor	Bad Bot
980	Scraper	Bad Bot
1025	Site Monitor	Bad Bot
1029	Feed Fetcher	Bad Bot
1030	Screenshot Creator	Bad Bot
1034	Tool	Bad Bot

Bot signature rule	ID	Description
1039	Marketing	Bad Bot
1042	Site Monitor	Bad Bot
1047	Site Monitor	Bad Bot
1053	Site Monitor	Bad Bot
1072	Search Engine	Bad Bot
1073	Feed Fetcher	Bad Bot
1074	Uncategorized	Bad Bot
1078	Screenshot Creator	Bad Bot
1109	Marketing	Bad Bot
1132	Feed Fetcher	Bad Bot
1138	Marketing	Bad Bot
1150	Search Engine	Bad Bot
1164	Search Engine	Bad Bot
1167	Marketing	Bad Bot
1173	Tool	Bad Bot
1174	Marketing	Bad Bot
1176	Search Engine	Bad Bot
1178	Speed Tester	Bad Bot
1185	Screenshot Creator	Bad Bot
1209	Uncategorized	Bad Bot
1244	Site Monitor	Bad Bot
1251	Search Engine	Bad Bot
1254	Site Monitor	Bad Bot
1256	Uncategorized	Bad Bot
1259	Tool	Bad Bot
1287	Search Engine	Bad Bot
1296	Search Engine	Bad Bot
1312	Uncategorized	Bad Bot
1316	Marketing	Bad Bot

Bot signature rule	ID	Description
1322	Site Monitor	Bad Bot
1325	Screenshot Creator	Bad Bot
1328	Search Engine	Bad Bot
1330	Marketing	Bad Bot
1337	Tool	Bad Bot
1360	Search Engine	Bad Bot
1367	Search Engine	Bad Bot
1374	Tool	Bad Bot
1380	Uncategorized	Bad Bot
1388	Search Engine	Bad Bot
1400	Feed Fetcher	Bad Bot
1413	Uncategorized	Bad Bot
1420	Feed Fetcher	Bad Bot
1422	Site Monitor	Bad Bot
1442	Uncategorized	Bad Bot
1447	Search Engine	Bad Bot
1460	Marketing	Bad Bot
1467	Tool	Bad Bot
1469	Tool	Bad Bot
1471	Search Engine	Bad Bot
1484	Uncategorized	Bad Bot
1493	Marketing	Bad Bot
1502	Site Monitor	Bad Bot
1504	Uncategorized	Bad Bot
1506	Uncategorized	Bad Bot
1518	Uncategorized	Bad Bot
1520	Search Engine	Bad Bot
1531	Feed Fetcher	Bad Bot
1533	Uncategorized	Bad Bot

Bot signature rule	ID	Description
1540	Search Engine	Bad Bot
1556	Marketing	Bad Bot
1560	Uncategorized	Bad Bot
1564	Tool	Bad Bot
1570	Site Monitor	Bad Bot
1575	Search Engine	Bad Bot
1586	Virus Scanner	Bad Bot
1588	Uncategorized	Bad Bot
1594	Tool	Bad Bot
1619	Marketing	Bad Bot
1623	Tool	Bad Bot
1626	Search Engine	Bad Bot
1632	Feed Fetcher	Bad Bot
1648	Search Engine	Bad Bot
1652	Marketing	Bad Bot
1660	Marketing	Bad Bot
1713	Tool	Bad Bot
1719	Search Engine	Bad Bot
1722	Uncategorized	Bad Bot
1744	Uncategorized	Bad Bot
1754	Uncategorized	Bad Bot
1757	Uncategorized	Bad Bot
1762	Uncategorized	Bad Bot
1769	Uncategorized	Bad Bot
1771	Marketing	Bad Bot
1779	Tool	Bad Bot
1782	Tool	Bad Bot
1785	Speed Tester	Bad Bot
1786	Tool	Bad Bot

Bot signature rule	ID	Description
1792	Site Monitor	Bad Bot
1869	Tool	Bad Bot
1928	Marketing	Bad Bot
1942	Site Monitor	Bad Bot
1949	Marketing	Bad Bot
1954	Marketing	Bad Bot
1964	Uncategorized	Bad Bot
1969	Search Engine	Bad Bot
2294	Search Engine	Bad Bot
2303	Uncategorized	Bad Bot
2308	Scraper	Bad Bot
2335	Marketing	Bad Bot
2374	Uncategorized	Bad Bot
2377	Uncategorized	Bad Bot
2385	Tool	Bad Bot
2389	Uncategorized	Bad Bot
2414	Uncategorized	Bad Bot
2421	Uncategorized	Bad Bot
2424	Uncategorized	Bad Bot
2427	Uncategorized	Bad Bot
2429	Search Engine	Bad Bot
2437	Uncategorized	Bad Bot
2440	Search Engine	Bad Bot
2443	Uncategorized	Bad Bot
2453	Marketing	Bad Bot
2472	Marketing	Bad Bot
2474	Feed Fetcher	Bad Bot
2482	Uncategorized	Bad Bot
2500	Screenshot Creator	Bad Bot

Bot signature rule	ID	Description
2503	Uncategorized	Bad Bot
2507	Uncategorized	Bad Bot
2516	Tool	Bad Bot
2536	Marketing	Bad Bot
2543	Tool	Bad Bot
2548	Tool	Bad Bot
2557	Marketing	Bad Bot
2561	Uncategorized	Bad Bot
2572	Uncategorized	Bad Bot
2578	Uncategorized	Bad Bot
2584	Uncategorized	Bad Bot
2588	Uncategorized	Bad Bot
2592	Search Engine	Bad Bot
2600	Tool	Bad Bot
2606	Uncategorized	Bad Bot
2611	Uncategorized	Bad Bot
2622	Tool	Bad Bot
2625	Tool	Bad Bot
2631	Tool	Bad Bot
2635	Tool	Bad Bot
2637	Screenshot Creator	Bad Bot
2641	Search Engine	Bad Bot
2655	Uncategorized	Bad Bot
2657	Marketing	Bad Bot
2663	Uncategorized	Bad Bot
2666	Tool	Bad Bot
2672	Feed Fetcher	Bad Bot
2674	Tool	Bad Bot
2681	Search Engine	Bad Bot

Bot signature rule	ID	Description
2684	Marketing	Bad Bot
2690	Uncategorized	Bad Bot
2704	Uncategorized	Bad Bot
2707	Uncategorized	Bad Bot
2714	Feed Fetcher	Bad Bot
2722	Uncategorized	Bad Bot
2726	Feed Fetcher	Bad Bot
2730	Screenshot Creator	Bad Bot
2736	Uncategorized	Bad Bot
2749	Uncategorized	Bad Bot
2753	Tool	Bad Bot
2756	Tool	Bad Bot
2760	Speed Tester	Bad Bot
2780	Tool	Bad Bot
2785	Site Monitor	Bad Bot
2789	Uncategorized	Bad Bot
2797	Tool	Bad Bot
2801	Tool	Bad Bot
2808	Tool	Bad Bot
2810	Uncategorized	Bad Bot
2813	Uncategorized	Bad Bot
2816	Uncategorized	Bad Bot
2820	Link Checker	Bad Bot
2824	Link Checker	Bad Bot
2831	Screenshot Creator	Bad Bot
2843	Tool	Bad Bot
2846	Tool	Bad Bot
2849	Marketing	Bad Bot
2851	Uncategorized	Bad Bot

Bot signature rule	ID	Description
2855	Uncategorized	Bad Bot
2859	Tool	Bad Bot
2873	Uncategorized	Bad Bot
2875	Screenshot Creator	Bad Bot
2879	Uncategorized	Bad Bot
2881	Uncategorized	Bad Bot
2886	Site Monitor	Bad Bot
2899	Uncategorized	Bad Bot
2916	Uncategorized	Bad Bot
2924	Tool	Bad Bot
2932	Marketing	Bad Bot
2935	Link Checker	Bad Bot
2939	Marketing	Bad Bot
2942	Uncategorized	Bad Bot
2955	Search Engine	Bad Bot
2960	Tool	Bad Bot
2964	Uncategorized	Bad Bot
2972	Marketing	Bad Bot
2978	Vulnerability Scanner	Bad Bot
2980	Tool	Bad Bot
2985	Marketing	Bad Bot
2993	Uncategorized	Bad Bot
2999	Screenshot Creator	Bad Bot
3003	Feed Fetcher	Bad Bot
3005	Uncategorized	Bad Bot
3013	Uncategorized	Bad Bot
3016	Uncategorized	Bad Bot
3021	Search Engine	Bad Bot
3026	Uncategorized	Bad Bot

Bot signature rule	ID	Description
3030	Marketing	Bad Bot
3065	Marketing	Bad Bot
3068	Uncategorized	Bad Bot
3072	Marketing	Bad Bot
3077	Marketing	Bad Bot
3080	Uncategorized	Bad Bot
3086	Scraper	Bad Bot
3092	Search Engine	Bad Bot
3100	Uncategorized	Bad Bot
3104	Tool	Bad Bot
3111	Uncategorized	Bad Bot
3116	Site Monitor	Bad Bot
3118	Tool	Bad Bot
3120	Marketing	Bad Bot
3122	Search Engine	Bad Bot
3126	Marketing	Bad Bot
3141	Tool	Bad Bot
3143	Uncategorized	Bad Bot
3145	Scraper	Bad Bot
3150	Uncategorized	Bad Bot
3173	Link Checker	Bad Bot
3176	Uncategorized	Bad Bot
3186	Speed Tester	Bad Bot
3190	Scraper	Bad Bot
3203	Search Engine	Bad Bot
3216	Uncategorized	Bad Bot
3220	Tool	Bad Bot
3223	Link Checker	Bad Bot
3241	Uncategorized	Bad Bot

Bot signature rule	ID	Description
3245	Site Monitor	Bad Bot
3285	Uncategorized	Bad Bot
3304	Marketing	Bad Bot
3307	Link Checker	Bad Bot
3316	Tool	Bad Bot
3326	Marketing	Bad Bot
3333	Search Engine	Bad Bot
3340	Search Engine	Bad Bot
3344	Marketing	Bad Bot
3350	Uncategorized	Bad Bot
3355	Marketing	Bad Bot
3365	Uncategorized	Bad Bot
3378	Uncategorized	Bad Bot
3388	Tool	Bad Bot
3396	Uncategorized	Bad Bot
3400	Uncategorized	Bad Bot
3421	Uncategorized	Bad Bot
3439	Uncategorized	Bad Bot
3447	Feed Fetcher	Bad Bot
3451	Tool	Bad Bot
3459	Screenshot Creator	Bad Bot
3469	Vulnerability Scanner	Bad Bot
3475	Uncategorized	Bad Bot
3485	Search Engine	Bad Bot
3493	Tool	Bad Bot
3502	Marketing	Bad Bot
3507	Search Engine	Bad Bot
3523	Uncategorized	Bad Bot
3535	Speed Tester	Bad Bot

Bot signature rule	ID	Description
3549	Uncategorized	Bad Bot
3556	Uncategorized	Bad Bot
3561	Uncategorized	Bad Bot
3565	Uncategorized	Bad Bot
3572	Search Engine	Bad Bot
3578	Uncategorized	Bad Bot
3610	Search Engine	Bad Bot
3617	Uncategorized	Bad Bot
3621	Marketing	Bad Bot
3632	Tool	Bad Bot
3635	Marketing	Bad Bot
3653	Uncategorized	Bad Bot
3661	Search Engine	Bad Bot
3704	Uncategorized	Bad Bot
3707	Uncategorized	Bad Bot
3711	Uncategorized	Bad Bot
3730	Search Engine	Bad Bot
3740	Site Monitor	Bad Bot
3759	Search Engine	Bad Bot
3764	Uncategorized	Bad Bot
3770	Uncategorized	Bad Bot

Bot signature update for March 2021

December 7, 2021

Some of existing bot signatures are updated. You can download and configure these signature rules to protect your appliance from bot attacks.

Bot signature version

Signature version 7 is applicable for Citrix ADC platforms with 13.0 61.x builds or later.

Updated bot signatures in this version

Following is a list of bot signature rule IDs, category, and its type.

Bot signature rule	ID	Description
278	Scraper	Good Bot
378	Scraper	Good Bot
379	Scraper	Good Bot
380	Scraper	Good Bot
381	Scraper	Good Bot
382	Scraper	Good Bot
383	Scraper	Good Bot
384	Scraper	Good Bot
385	Scraper	Good Bot
386	Scraper	Good Bot
387	Scraper	Good Bot
389	Scraper	Good Bot
390	Scraper	Good Bot
391	Scraper	Good Bot
494	Scraper	Good Bot
627	Search Engine	Good Bot
660	Search Engine	Good Bot
3840	Crawler	Good Bot

Bot signature update for August 2021

December 7, 2021

New signatures are added and some of existing bot signatures are updated. You can download and

configure these signature rules to protect your appliance from bot attacks.

Bot signature version

Signature version 8 is applicable for Citrix ADC platforms with 13.0 61.x builds or later.

Updated bot signatures in this version

Following is a list of bot signature rule IDs, category, and its type.

Bot signature rule	ID	Description
236	Scraper	Good Bot
378	Scraper	Good Bot
381	Scraper	Good Bot
382	Scraper	Good Bot
390	Scraper	Good Bot
544	Scraper	Good Bot
702	Search Engine	Good Bot
979	Scraper	Bad Bot
3791	Speed Tester	Good Bot
3797	Marketing	Good Bot
3800	Marketing	Good Bot
3824	Crawler	Bad Bot
3833	Search Engine	Good Bot
3849	Crawler	Good Bot
3871	Marketing	Good Bot
3963	Marketing	Good Bot
4027	Search Engine	Good Bot

New bot signatures in this version

Bot signature rule	ID	Description
4028	Marketing	Good Bot
4029	Tool	Good Bot
4030	Scraper	Good Bot
4031	Scraper	Good Bot
4032	Uncategorized	Bad Bot
4033	Crawler	Good Bot
4034	Crawler	Good Bot
4035	Marketing	Good Bot
4036	Vulnerability Scanner	Good Bot
4037	Vulnerability Scanner	Good Bot
4038	Uncategorized	Bad Bot
4039	Tool	Good Bot
4040	Crawler	Good Bot
4041	Tool	Good Bot
4042	Crawler	Good Bot
4043	Screenshot Creator	Good Bot
4044	Scraper	Bad Bot
4045	Scraper	Bad Bot
4046	Scraper	Bad Bot
4047	Uncategorized	Bad Bot
4048	Feed Fetcher	Good Bot
4049	Uncategorized	Bad Bot
4050	Crawler	Good Bot
4051	Crawler	Good Bot
4052	Tool	Good Bot
4053	Tool	Good Bot
4054	Scraper	Bad Bot
4055	Uncategorized	Good Bot
4056	Marketing	Good Bot

Bot signature rule	ID	Description
4057	Screenshot Creator	Good Bot
4058	Crawler	Good Bot
4059	Uncategorized	Bad Bot
4060	Search Engine	Good Bot
4061	Search Engine	Good Bot
4062	Search Engine	Good Bot
4063	Search Engine	Good Bot
4064	Tool	Good Bot
4065	Scraper	Good Bot
4066	Marketing	Good Bot
4067	Marketing	Good Bot
4068	Uncategorized	Bad Bot
4069	Uncategorized	Bad Bot
4070	Uncategorized	Bad Bot
4071	Tool	Good Bot
4072	Tool	Bad Bot
4073	Uncategorized	Bad Bot
4074	Uncategorized	Bad Bot
4075	Tool	Bad Bot
4076	Marketing	Good Bot
4077	Scraper	Good Bot
4078	Crawler	Good Bot
4079	Crawler	Good Bot
4080	Tool	Bad Bot
4081	Search Engine	Good Bot
4082	Tool	Good Bot
4083	Uncategorized	Bad Bot
4084	Uncategorized	Bad Bot
4085	Tool	Good Bot

Bot signature rule	ID	Description
4086	Tool	Good Bot
4087	Tool	Bad Bot
4088	Search Engine	Good Bot
4089	Marketing	Good Bot
4090	Tool	Good Bot
4091	Tool	Good Bot
4092	Tool	Good Bot
4093	Tool	Good Bot
4094	Uncategorized	Good Bot
4095	Site Monitor	Good Bot
4096	Site Monitor	Good Bot
4097	Site Monitor	Good Bot
4098	Crawler	Good Bot
4099	Search Engine	Good Bot
4100	Search Engine	Good Bot
4101	Search Engine	Good Bot
4102	Search Engine	Good Bot
4103	Marketing	Good Bot
4104	Marketing	Good Bot
4105	Marketing	Good Bot
4106	Marketing	Good Bot
4107	Marketing	Good Bot
4108	Marketing	Good Bot
4109	Search Engine	Good Bot
4110	Crawler	Good Bot
4111	Crawler	Good Bot
4112	Crawler	Good Bot
4113	Vulnerability Scanner	Good Bot
4114	Crawler	Good Bot

Bot signature rule	ID	Description
4115	Tool	Good Bot
4116	Uncategorized	Bad Bot
4117	Uncategorized	Bad Bot
4118	Uncategorized	Bad Bot
4119	Uncategorized	Bad Bot
4120	Marketing	Good Bot
4121	Marketing	Good Bot
4122	Marketing	Good Bot
4123	Marketing	Good Bot
4124	Marketing	Good Bot
4125	Marketing	Good Bot
4126	Marketing	Good Bot
4127	Marketing	Good Bot
4128	Marketing	Good Bot
4129	Marketing	Good Bot
4130	Marketing	Good Bot
4131	Tool	Good Bot
4132	Marketing	Good Bot
4133	Marketing	Good Bot
4134	Tool	Good Bot
4135	Marketing	Good Bot
4136	Marketing	Good Bot
4137	Marketing	Good Bot
4138	Marketing	Good Bot
4139	Marketing	Good Bot
4140	Marketing	Good Bot
4141	Marketing	Good Bot
4142	Marketing	Good Bot
4143	Marketing	Good Bot

Bot signature rule	ID	Description
4144	Marketing	Good Bot
4145	Search Engine	Good Bot
4146	Search Engine	Good Bot
4147	Search Engine	Good Bot
4148	Search Engine	Good Bot
4149	Search Engine	Good Bot
4150	Search Engine	Good Bot
4151	Search Engine	Good Bot
4152	Search Engine	Good Bot
4153	Search Engine	Good Bot
4154	Search Engine	Good Bot
4155	Search Engine	Good Bot
4156	Screenshot Creator	Good Bot
4157	Search Engine	Good Bot
4158	Search Engine	Good Bot
4159	Search Engine	Good Bot
4160	Screenshot Creator	Good Bot
4161	Search Engine	Good Bot
4162	Search Engine	Good Bot
4163	Tool	Good Bot
4164	Search Engine	Good Bot
4165	Marketing	Good Bot
4166	Uncategorized	Bad Bot
4167	Tool	Bad Bot
4168	Speed Tester	Good Bot
4169	Scraper	Bad Bot
4170	Tool	Good Bot
4171	Scraper	Bad Bot
4172	Web Crawler	Good Bot

Bot signature rule	ID	Description
4173	Tool	Good Bot
4174	Crawler	Good Bot
4175	Crawler	Good Bot
4176	Tool	Good Bot
4177	Search Engine	Good Bot
4178	Tool	Good Bot
4179	Web Crawler	Good Bot
4180	Tool	Good Bot
4181	Site Monitor	Good Bot
4182	Site Monitor	Good Bot
4183	Site Monitor	Good Bot
4184	Site Monitor	Good Bot
4185	Search Engine	Good Bot
4186	Tool	Good Bot
4187	Tool	Good Bot
4188	Screenshot Creator	Good Bot
4189	Marketing	Good Bot
4190	Search Engine	Good Bot
4191	Search Engine	Good Bot
4192	Search Engine	Good Bot
4193	Search Engine	Good Bot
4194	Tool	Good Bot
4195	Search Engine	Bad Bot
4196	Tool	Good Bot
4197	Tool	Good Bot
4198	Marketing	Good Bot
4199	Marketing	Good Bot
4200	Vulnerability Scanner	Good Bot
4201	Tool	Good Bot

Bot signature rule	ID	Description
4202	Tool	Good Bot
4203	Uncategorized	Bad Bot
4204	Uncategorized	Bad Bot
4205	Search Engine	Good Bot
4206	Marketing	Good Bot
4207	Marketing	Good Bot
4208	Search Engine	Good Bot
4209	Search Engine	Good Bot
4210	Speed Tester	Good Bot
4211	Tool	Good Bot
4212	Feed Fetcher	Good Bot
4213	Feed Fetcher	Good Bot
4214	Scraper	Bad Bot
4215	Tool	Good Bot
4216	Tool	Good Bot
4217	Tool	Bad Bot
4218	Scraper	Bad Bot
4219	Marketing	Good Bot
4220	Tool	Good Bot
4221	Tool	Bad Bot
4222	Site Monitor	Good Bot
4223	Marketing	Good Bot
4224	Search Engine	Good Bot
4225	Search Engine	Good Bot
4226	Search Engine	Good Bot
4227	Marketing	Good Bot
4228	Marketing	Good Bot
4229	Tool	Good Bot
4230	Uncategorized	Bad Bot

Bot signature rule	ID	Description
4231	Screenshot Creator	Good Bot
4232	Tool	Good Bot
4233	Site Monitor	Good Bot
4234	Site Monitor	Good Bot
4235	Site Monitor	Good Bot
4236	Site Monitor	Good Bot
4237	Site Monitor	Good Bot
4238	Site Monitor	Good Bot
4239	Uncategorized	Bad Bot
4240	Marketing	Good Bot
4241	Marketing	Good Bot
4242	Marketing	Good Bot
4243	Marketing	Good Bot
4244	Marketing	Good Bot
4245	Marketing	Good Bot
4246	Marketing	Good Bot
4247	Search Engine	Good Bot
4248	Search Engine	Good Bot
4249	Screenshot Creator	Good Bot
4250	Search Engine	Good Bot
4251	Search Engine	Good Bot
4252	Crawler	Good Bot
4253	Crawler	Good Bot
4254	Crawler	Good Bot
4255	Tool	Good Bot
4256	Uncategorized	Good Bot
4257	Tool	Good Bot
4258	Crawler	Good Bot
4259	Crawler	Good Bot

Bot signature rule	ID	Description
4260	Tool	Good Bot
4261	Tool	Good Bot
4262	Tool	Good Bot
4263	Marketing	Good Bot
4264	Crawler	Bad Bot
4265	Search Engine	Good Bot
4266	Uncategorized	Good Bot
4267	Tool	Good Bot
4268	Tool	Good Bot
4269	Search Engine	Good Bot
4270	Search Engine	Good Bot
4271	Search Engine	Good Bot
4272	Search Engine	Good Bot
4273	Search Engine	Good Bot
4274	Search Engine	Good Bot
4275	Search Engine	Good Bot
4276	Uncategorized	Bad Bot
4277	Uncategorized	Bad Bot
4278	Uncategorized	Bad Bot
4279	Marketing	Good Bot
4280	Crawler	Good Bot
4281	Uncategorized	Bad Bot
4282	Marketing	Good Bot
4283	Marketing	Good Bot
4284	Marketing	Good Bot
4285	Marketing	Good Bot
4286	Marketing	Good Bot
4287	Marketing	Good Bot
4288	Marketing	Good Bot

Bot signature rule	ID	Description
4289	Marketing	Good Bot
4290	Marketing	Good Bot
4291	Marketing	Good Bot
4292	Marketing	Good Bot
4293	Marketing	Good Bot
4294	Marketing	Good Bot
4295	Search Engine	Good Bot
4296	Search Engine	Good Bot
4297	Search Engine	Good Bot
4298	Search Engine	Good Bot
4299	Search Engine	Good Bot
4300	Search Engine	Good Bot
4301	Search Engine	Good Bot
4302	Search Engine	Good Bot
4303	Search Engine	Good Bot
4304	Search Engine	Good Bot
4305	Search Engine	Good Bot
4306	Screenshot Creator	Good Bot
4307	Search Engine	Good Bot
4308	Search Engine	Good Bot
4309	Search Engine	Good Bot
4310	Search Engine	Good Bot
4311	Screenshot Creator	Good Bot
4312	Search Engine	Good Bot
4313	Search Engine	Good Bot
4314	Search Engine	Good Bot
4315	Search Engine	Good Bot
4316	Search Engine	Good Bot
4317	Search Engine	Good Bot

Bot signature rule	ID	Description
4318	Screenshot Creator	Good Bot
4319	Screenshot Creator	Good Bot
4320	Uncategorized	Bad Bot
4321	Uncategorized	Good Bot
4322	Crawler	Good Bot
4323	Tool	Good Bot
4324	Tool	Good Bot
4325	Tool	Good Bot
4326	Scraper	Bad Bot
4327	Search Engine	Good Bot
4328	Marketing	Good Bot
4329	Uncategorized	Bad Bot
4330	Site Monitor	Good Bot
4331	Search Engine	Good Bot
4332	Search Engine	Good Bot
4333	Uncategorized	Bad Bot
4334	Scraper	Good Bot
4335	Marketing	Good Bot
4336	Marketing	Good Bot
4337	Tool	Good Bot
4338	Tool	Good Bot
4339	Tool	Good Bot
4340	Crawler	Good Bot
4341	Crawler	Good Bot
4342	Vulnerability Scanner	Good Bot
4343	Vulnerability Scanner	Good Bot
4344	Scraper	Good Bot
4345	Marketing	Good Bot
4346	Marketing	Good Bot

Bot signature rule	ID	Description
4347	Marketing	Good Bot
4348	Marketing	Good Bot
4349	Marketing	Good Bot
4350	Marketing	Good Bot
4351	Marketing	Good Bot
4352	Marketing	Good Bot
4353	Marketing	Good Bot
4354	Marketing	Good Bot
4355	Search Engine	Good Bot
4356	Search Engine	Good Bot
4357	Search Engine	Good Bot
4358	Search Engine	Good Bot
4359	Search Engine	Good Bot
4360	Search Engine	Good Bot
4361	Search Engine	Good Bot
4362	Search Engine	Good Bot
4363	Search Engine	Good Bot
4364	Search Engine	Good Bot
4365	Screenshot Creator	Good Bot
4366	Search Engine	Good Bot
4367	Search Engine	Good Bot
4368	Search Engine	Good Bot
4369	Search Engine	Good Bot
4370	Screenshot Creator	Good Bot
4371	Search Engine	Good Bot
4372	Search Engine	Good Bot
4373	Search Engine	Good Bot
4374	Search Engine	Good Bot
4375	Search Engine	Good Bot

Bot signature rule	ID	Description
4376	Screenshot Creator	Good Bot
4377	Crawler	Good Bot
4378	Crawler	Good Bot
4379	Search Engine	Good Bot
4380	Search Engine	Good Bot
4381	Search Engine	Good Bot
4382	Search Engine	Good Bot
4383	Crawler	Good Bot
4384	Search Engine	Good Bot
4385	Tool	Good Bot
4386	Uncategorized	Good Bot
4387	Crawler	Good Bot
4388	Crawler	Good Bot
4389	Tool	Good Bot
4390	Tool	Good Bot
4391	Tool	Good Bot
4392	Tool	Good Bot
4393	Tool	Good Bot
4394	Uncategorized	Good Bot
4395	Tool	Good Bot
4396	Site Monitor	Good Bot
4397	Site Monitor	Good Bot
4398	Tool	Bad Bot
4399	Tool	Bad Bot
4400	Tool	Bad Bot
4401	Tool	Bad Bot
4402	Tool	Bad Bot
4403	Tool	Bad Bot
4404	Search Engine	Good Bot

Bot signature rule	ID	Description
4405	Search Engine	Good Bot
4406	Search Engine	Good Bot
4407	Uncategorized	Good Bot

2021 年 9 月のボットシグネチャアップデート

January 31, 2022

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 9 は、13.0 61.48 以降のビルドを持つ Citrix ADC プラットフォームに適用されます。

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	ID	説明
2	昇降補助具	いいボット
5	昇降補助具	いいボット
9	昇降補助具	いいボット
45	昇降補助具	いいボット
46	昇降補助具	いいボット
48	昇降補助具	いいボット
52	昇降補助具	いいボット
60	昇降補助具	いいボット
61	昇降補助具	いいボット
63	昇降補助具	いいボット
67	昇降補助具	いいボット
71	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
74	昇降補助具	いいボット
75	昇降補助具	いいボット
76	昇降補助具	いいボット
78	昇降補助具	いいボット
79	昇降補助具	いいボット
80	昇降補助具	いいボット
81	昇降補助具	いいボット
82	昇降補助具	いいボット
83	昇降補助具	いいボット
84	昇降補助具	いいボット
87	昇降補助具	いいボット
90	昇降補助具	いいボット
95	昇降補助具	いいボット
96	昇降補助具	いいボット
97	昇降補助具	いいボット
100	昇降補助具	いいボット
101	昇降補助具	いいボット
102	昇降補助具	いいボット
103	昇降補助具	いいボット
104	昇降補助具	いいボット
107	昇降補助具	いいボット
108	昇降補助具	いいボット
110	昇降補助具	いいボット
111	昇降補助具	いいボット
114	昇降補助具	いいボット
115	昇降補助具	いいボット
123	昇降補助具	いいボット
135	昇降補助具	いいボット
136	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
137	昇降補助具	いいボット
140	昇降補助具	いいボット
141	昇降補助具	いいボット
143	昇降補助具	いいボット
144	昇降補助具	いいボット
145	昇降補助具	いいボット
146	昇降補助具	いいボット
147	昇降補助具	いいボット
149	昇降補助具	いいボット
152	昇降補助具	いいボット
155	昇降補助具	いいボット
156	昇降補助具	いいボット
157	昇降補助具	いいボット
158	昇降補助具	いいボット
159	昇降補助具	いいボット
160	昇降補助具	いいボット
161	昇降補助具	いいボット
162	昇降補助具	いいボット
163	昇降補助具	いいボット
164	昇降補助具	いいボット
165	昇降補助具	いいボット
166	昇降補助具	いいボット
167	昇降補助具	いいボット
172	昇降補助具	いいボット
173	昇降補助具	いいボット
174	昇降補助具	いいボット
176	昇降補助具	いいボット
177	昇降補助具	いいボット
180	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
187	昇降補助具	いいボット
197	昇降補助具	いいボット
201	昇降補助具	いいボット
202	昇降補助具	いいボット
203	昇降補助具	いいボット
206	昇降補助具	いいボット
211	フィードフェッチャー	悪いボット
217	フィードフェッチャー	いいボット
219	フィードフェッチャー	いいボット
229	スクレーパー	いいボット
235	スクレーパー	いいボット
236	スクレーパー	いいボット
237	スクレーパー	いいボット
248	スクレーパー	いいボット
250	スクレーパー	いいボット
260	スクレーパー	いいボット
263	スクレーパー	いいボット
265	スクレーパー	いいボット
267	スクレーパー	いいボット
268	スクレーパー	いいボット
271	スクレーパー	いいボット
272	スクレーパー	いいボット
276	スクレーパー	いいボット
277	スクレーパー	いいボット
278	スクレーパー	いいボット
279	スクレーパー	いいボット
280	スクレーパー	いいボット
281	スクレーパー	いいボット
283	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
285	スクレーパー	いいボット
286	スクレーパー	いいボット
287	スクレーパー	いいボット
290	スクレーパー	いいボット
292	スクレーパー	いいボット
293	スクレーパー	いいボット
342	スクレーパー	いいボット
343	スクレーパー	いいボット
344	スクレーパー	いいボット
355	スクレーパー	いいボット
357	スクレーパー	いいボット
360	スクレーパー	いいボット
362	スクレーパー	いいボット
366	スクレーパー	いいボット
370	スクレーパー	いいボット
371	スクレーパー	いいボット
372	スクレーパー	いいボット
373	スクレーパー	いいボット
374	スクレーパー	いいボット
376	スクレーパー	いいボット
377	スクレーパー	いいボット
380	スクレーパー	いいボット
392	スクレーパー	いいボット
393	スクレーパー	いいボット
394	スクレーパー	いいボット
396	スクレーパー	いいボット
397	スクレーパー	いいボット
414	スクレーパー	いいボット
418	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
419	スクレーパー	いいボット
421	スクレーパー	いいボット
422	スクレーパー	いいボット
423	スクレーパー	いいボット
424	スクレーパー	いいボット
425	スクレーパー	いいボット
426	スクレーパー	いいボット
427	スクレーパー	いいボット
428	スクレーパー	いいボット
430	スクレーパー	いいボット
432	スクレーパー	いいボット
433	スクレーパー	いいボット
434	スクレーパー	いいボット
435	スクレーパー	いいボット
441	スクレーパー	いいボット
445	スクレーパー	いいボット
446	スクレーパー	いいボット
451	スクレーパー	いいボット
452	スクレーパー	いいボット
454	スクレーパー	いいボット
455	スクレーパー	いいボット
456	スクレーパー	いいボット
457	スクレーパー	いいボット
458	スクレーパー	いいボット
461	スクレーパー	いいボット
465	スクレーパー	いいボット
466	スクレーパー	いいボット
469	スクレーパー	いいボット
473	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
474	スクレーパー	いいボット
476	スクレーパー	いいボット
477	スクレーパー	いいボット
484	スクレーパー	いいボット
485	スクレーパー	いいボット
487	スクレーパー	いいボット
488	スクレーパー	いいボット
489	スクレーパー	いいボット
490	スクレーパー	いいボット
493	スクレーパー	いいボット
494	スクレーパー	いいボット
495	スクレーパー	いいボット
497	スクレーパー	いいボット
498	スクレーパー	いいボット
499	スクレーパー	いいボット
500	スクレーパー	いいボット
505	スクレーパー	いいボット
506	スクレーパー	いいボット
507	スクレーパー	いいボット
512	スクレーパー	いいボット
513	スクレーパー	いいボット
514	スクレーパー	いいボット
527	スクレーパー	いいボット
533	スクレーパー	いいボット
539	スクレーパー	いいボット
540	スクレーパー	いいボット
542	スクレーパー	いいボット
544	スクレーパー	いいボット
545	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
546	スクレーパー	いいボット
547	スクレーパー	いいボット
548	スクレーパー	いいボット
551	スクレーパー	いいボット
552	スクレーパー	いいボット
554	スクレーパー	いいボット
556	スクレーパー	いいボット
558	スクレーパー	いいボット
560	スクレーパー	いいボット
561	スクレーパー	いいボット
566	スクレーパー	いいボット
575	スクレーパー	いいボット
578	スクレーパー	いいボット
581	スクレーパー	いいボット
591	スクレーパー	いいボット
593	スクレーパー	いいボット
595	スクレーパー	いいボット
600	スクレーパー	いいボット
601	スクレーパー	いいボット
602	スクレーパー	いいボット
604	スクレーパー	いいボット
605	スクレーパー	いいボット
609	スクレーパー	いいボット
610	スクレーパー	いいボット
611	スクレーパー	いいボット
612	スクレーパー	いいボット
613	スクレーパー	いいボット
615	スクレーパー	いいボット
620	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
622	検索エンジン	いいボット
623	検索エンジン	いいボット
624	検索エンジン	いいボット
626	検索エンジン	いいボット
627	検索エンジン	いいボット
628	検索エンジン	いいボット
629	検索エンジン	いいボット
633	検索エンジン	いいボット
634	検索エンジン	いいボット
636	検索エンジン	いいボット
637	検索エンジン	いいボット
639	検索エンジン	いいボット
640	検索エンジン	いいボット
641	検索エンジン	いいボット
642	検索エンジン	いいボット
643	検索エンジン	いいボット
647	検索エンジン	いいボット
649	検索エンジン	いいボット
650	検索エンジン	いいボット
651	検索エンジン	いいボット
654	検索エンジン	いいボット
656	検索エンジン	いいボット
657	検索エンジン	いいボット
658	検索エンジン	いいボット
659	検索エンジン	いいボット
660	検索エンジン	いいボット
663	検索エンジン	いいボット
664	検索エンジン	いいボット
665	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
666	検索エンジン	いいボット
667	検索エンジン	いいボット
669	検索エンジン	いいボット
670	検索エンジン	いいボット
671	検索エンジン	いいボット
672	検索エンジン	いいボット
673	検索エンジン	いいボット
674	検索エンジン	いいボット
675	検索エンジン	いいボット
676	検索エンジン	いいボット
677	検索エンジン	いいボット
679	検索エンジン	いいボット
680	検索エンジン	いいボット
690	検索エンジン	いいボット
693	検索エンジン	いいボット
694	検索エンジン	いいボット
697	検索エンジン	いいボット
698	検索エンジン	いいボット
703	検索エンジン	いいボット
706	検索エンジン	いいボット
712	検索エンジン	いいボット
714	検索エンジン	いいボット
715	検索エンジン	いいボット
716	検索エンジン	いいボット
721	検索エンジン	いいボット
723	検索エンジン	いいボット
725	検索エンジン	いいボット
727	検索エンジン	いいボット
728	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
729	検索エンジン	いいボット
730	検索エンジン	いいボット
731	検索エンジン	いいボット
732	検索エンジン	いいボット
735	検索エンジン	いいボット
736	検索エンジン	いいボット
740	検索エンジン	いいボット
748	検索エンジン	いいボット
749	検索エンジン	いいボット
750	検索エンジン	いいボット
751	検索エンジン	いいボット
756	検索エンジン	いいボット
757	検索エンジン	いいボット
758	検索エンジン	いいボット
759	検索エンジン	いいボット
760	検索エンジン	いいボット
761	検索エンジン	いいボット
762	検索エンジン	いいボット
763	検索エンジン	いいボット
764	検索エンジン	いいボット
765	検索エンジン	いいボット
766	検索エンジン	いいボット
767	検索エンジン	いいボット
768	検索エンジン	いいボット
769	検索エンジン	いいボット
770	検索エンジン	いいボット
771	検索エンジン	いいボット
772	検索エンジン	いいボット
773	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
776	検索エンジン	いいボット
777	検索エンジン	いいボット
780	検索エンジン	いいボット
781	検索エンジン	いいボット
784	検索エンジン	いいボット
786	検索エンジン	いいボット
787	検索エンジン	いいボット
788	検索エンジン	いいボット
789	検索エンジン	いいボット
790	検索エンジン	いいボット
791	検索エンジン	いいボット
792	検索エンジン	いいボット
795	検索エンジン	いいボット
796	検索エンジン	いいボット
798	検索エンジン	いいボット
800	検索エンジン	いいボット
801	検索エンジン	いいボット
802	検索エンジン	いいボット
803	検索エンジン	いいボット
805	検索エンジン	いいボット
806	検索エンジン	いいボット
807	検索エンジン	いいボット
809	検索エンジン	いいボット
810	検索エンジン	いいボット
811	検索エンジン	いいボット
812	検索エンジン	いいボット
814	検索エンジン	いいボット
815	検索エンジン	いいボット
816	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
817	検索エンジン	いいボット
818	検索エンジン	いいボット
819	検索エンジン	いいボット
820	検索エンジン	いいボット
821	検索エンジン	いいボット
822	検索エンジン	いいボット
823	検索エンジン	いいボット
825	検索エンジン	いいボット
827	検索エンジン	いいボット
830	検索エンジン	いいボット
831	検索エンジン	いいボット
834	検索エンジン	いいボット
837	検索エンジン	いいボット
838	検索エンジン	いいボット
849	サイトモニター	いいボット
850	サイトモニター	いいボット
851	サイトモニター	いいボット
853	サイトモニター	いいボット
857	サイトモニター	いいボット
858	サイトモニター	いいボット
859	サイトモニター	いいボット
860	サイトモニター	いいボット
861	サイトモニター	いいボット
862	サイトモニター	いいボット
863	サイトモニター	いいボット
864	サイトモニター	いいボット
865	サイトモニター	いいボット
866	サイトモニター	いいボット
867	サイトモニター	いいボット

ボットシグネチャルール	**ID**	説明
868	サイトモニター	いいボット
869	サイトモニター	いいボット
870	サイトモニター	いいボット
871	サイトモニター	いいボット
872	サイトモニター	いいボット
873	サイトモニター	いいボット
874	サイトモニター	いいボット
875	サイトモニター	いいボット
876	サイトモニター	いいボット
877	サイトモニター	いいボット
880	サイトモニター	いいボット
883	サイトモニター	いいボット
885	サイトモニター	いいボット
886	サイトモニター	いいボット
888	サイトモニター	いいボット
889	サイトモニター	いいボット
895	サイトモニター	いいボット
896	サイトモニター	いいボット
897	サイトモニター	いいボット
898	サイトモニター	いいボット
900	サイトモニター	いいボット
901	サイトモニター	いいボット
904	サイトモニター	いいボット
906	サイトモニター	いいボット
908	サイトモニター	いいボット
909	サイトモニター	いいボット
910	サイトモニター	いいボット
911	サイトモニター	いいボット
912	サイトモニター	いいボット

ボットシグネチャルール	**ID**	説明
913	サイトモニター	いいボット
917	サイトモニター	いいボット
918	サイトモニター	いいボット
919	サイトモニター	いいボット
920	サイトモニター	いいボット
921	サイトモニター	いいボット
924	サイトモニター	いいボット
926	サイトモニター	いいボット
927	サイトモニター	いいボット
928	サイトモニター	いいボット
929	サイトモニター	いいボット
930	サイトモニター	いいボット
931	サイトモニター	いいボット
938	サイトモニター	いいボット
939	サイトモニター	いいボット
943	サイトモニター	悪いボット
958	サイトモニター	いいボット
959	サイトモニター	いいボット
960	サイトモニター	いいボット
963	サイトモニター	いいボット
984	スクレーパー	いいボット
996	スクレーパー	いいボット
997	スクレーパー	いいボット
998	スクレーパー	いいボット
1002	スクレーパー	いいボット
1006	スクレーパー	いいボット
1588	未分類	悪いボット
2561	スクレーパー	悪いボット
2810	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
3782	マーケティング	いいボット
3783	検索エンジン	いいボット
3788	ツール	いいボット
3789	ツール	いいボット
3790	昇降補助具	いいボット
3792	ツール	いいボット
3793	ツール	いいボット
3794	昇降補助具	いいボット
3796	スクレーパー	いいボット
3798	マーケティング	いいボット
3799	マーケティング	いいボット
3801	マーケティング	いいボット
3802	スクリーンショットクリエーター	いいボット
3803	検索エンジン	いいボット
3804	スクリーンショットクリエーター	いいボット
3805	検索エンジン	いいボット
3806	ツール	いいボット
3807	昇降補助具	いいボット
3808	昇降補助具	いいボット
3809	ツール	いいボット
3810	スクレーパー	いいボット
3811	ツール	いいボット
3813	ツール	いいボット
3814	昇降補助具	いいボット
3815	未分類	いいボット
3817	ツール	いいボット
3818	ツール	いいボット
3819	ツール	いいボット
3820	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
3821	検索エンジン	いいボット
3822	マーケティング	いいボット
3823	未分類	いいボット
3831	スクレーパー	いいボット
3834	検索エンジン	いいボット
3835	検索エンジン	いいボット
3836	未分類	いいボット
3837	未分類	いいボット
3838	未分類	いいボット
3839	マーケティング	いいボット
3840	昇降補助具	いいボット
3842	昇降補助具	いいボット
3843	昇降補助具	いいボット
3844	マーケティング	いいボット
3845	マーケティング	いいボット
3846	マーケティング	いいボット
3847	マーケティング	いいボット
3848	未分類	いいボット
3850	ツール	いいボット
3851	未分類	いいボット
3852	ツール	いいボット
3853	脆弱性スキャナ	いいボット
3854	昇降補助具	いいボット
3855	昇降補助具	いいボット
3856	ツール	いいボット
3861	マーケティング	いいボット
3862	マーケティング	いいボット
3863	マーケティング	いいボット
3864	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
3865	マーケティング	いいボット
3866	マーケティング	いいボット
3867	マーケティング	いいボット
3868	マーケティング	いいボット
3869	ツール	いいボット
3870	マーケティング	いいボット
3872	マーケティング	いいボット
3873	検索エンジン	いいボット
3874	検索エンジン	いいボット
3875	検索エンジン	いいボット
3876	検索エンジン	いいボット
3877	スクリーンショットクリエーター	いいボット
3878	検索エンジン	いいボット
3879	検索エンジン	いいボット
3880	スクリーンショットクリエーター	いいボット
3881	スクリーンショットクリエーター	いいボット
3882	検索エンジン	いいボット
3883	検索エンジン	いいボット
3884	検索エンジン	いいボット
3885	検索エンジン	いいボット
3886	ツール	いいボット
3887	昇降補助具	いいボット
3888	昇降補助具	いいボット
3889	未分類	いいボット
3890	マーケティング	いいボット
3893	昇降補助具	いいボット
3894	ツール	いいボット
3895	ツール	いいボット
3896	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
3897	ツール	いいボット
3898	ツール	いいボット
3899	未分類	いいボット
3901	昇降補助具	いいボット
3903	ツール	いいボット
3904	検索エンジン	いいボット
3905	検索エンジン	いいボット
3906	検索エンジン	いいボット
3912	昇降補助具	いいボット
3918	昇降補助具	いいボット
3919	未分類	いいボット
3920	未分類	いいボット
3921	未分類	いいボット
3922	未分類	いいボット
3923	未分類	いいボット
3924	未分類	いいボット
3925	未分類	いいボット
3926	マーケティング	いいボット
3927	マーケティング	いいボット
3928	マーケティング	いいボット
3929	ツール	いいボット
3930	マーケティング	いいボット
3931	未分類	いいボット
3932	昇降補助具	いいボット
3933	マーケティング	いいボット
3934	マーケティング	いいボット
3935	スクレーパー	いいボット
3936	マーケティング	いいボット
3937	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
3938	フィードフェッチャー	いいボット
3940	検索エンジン	いいボット
3941	昇降補助具	いいボット
3942	スクレーパー	いいボット
3946	フィードフェッチャー	いいボット
3947	昇降補助具	いいボット
3950	ウイルススキャナー	いいボット
3951	マーケティング	いいボット
3952	マーケティング	いいボット
3953	マーケティング	いいボット
3954	マーケティング	いいボット
3955	マーケティング	いいボット
3956	マーケティング	いいボット
3957	マーケティング	いいボット
3958	マーケティング	いいボット
3959	マーケティング	いいボット
3960	マーケティング	いいボット
3961	マーケティング	いいボット
3962	マーケティング	いいボット
3964	マーケティング	いいボット
3965	マーケティング	いいボット
3966	マーケティング	いいボット
3967	マーケティング	いいボット
3968	マーケティング	いいボット
3969	マーケティング	いいボット
3970	検索エンジン	いいボット
3971	スクリーンショットクリエーター	いいボット
3972	スクリーンショットクリエーター	いいボット
3973	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
3974	検索エンジン	いいボット
3975	検索エンジン	いいボット
3976	検索エンジン	いいボット
3977	検索エンジン	いいボット
3978	スクリーンショットクリエーター	いいボット
3979	検索エンジン	いいボット
3980	スクリーンショットクリエーター	いいボット
3981	検索エンジン	いいボット
3982	検索エンジン	いいボット
3983	検索エンジン	いいボット
3984	検索エンジン	いいボット
3985	検索エンジン	いいボット
3986	検索エンジン	いいボット
3987	スクリーンショットクリエーター	いいボット
3988	検索エンジン	いいボット
3989	検索エンジン	いいボット
3990	検索エンジン	いいボット
3991	検索エンジン	いいボット
3992	検索エンジン	いいボット
3993	検索エンジン	いいボット
3994	検索エンジン	いいボット
3995	検索エンジン	いいボット
3996	検索エンジン	いいボット
3997	検索エンジン	いいボット
3998	検索エンジン	いいボット
3999	検索エンジン	いいボット
4000	スクリーンショットクリエーター	いいボット
4001	検索エンジン	いいボット
4002	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4003	検索エンジン	いいボット
4004	検索エンジン	いいボット
4005	スクリーンショットクリエーター	いいボット
4006	昇降補助具	いいボット
4007	マーケティング	いいボット
4008	マーケティング	いいボット
4011	ツール	いいボット
4012	昇降補助具	いいボット
4013	検索エンジン	いいボット
4014	ツール	いいボット
4015	昇降補助具	いいボット
4016	昇降補助具	いいボット
4017	ツール	いいボット
4018	ツール	いいボット
4019	ツール	いいボット
4020	ツール	いいボット
4021	マーケティング	いいボット
4024	ツール	いいボット
4025	検索エンジン	いいボット
4026	検索エンジン	いいボット
4028	マーケティング	いいボット
4029	ツール	いいボット
4030	スクレーパー	いいボット
4031	スクレーパー	いいボット
4035	マーケティング	いいボット
4037	脆弱性スキャナ	いいボット
4042	昇降補助具	いいボット
4043	スクリーンショットクリエーター	いいボット
4048	フィードフェッチャー	いいボット

ボットシグネチャルール	**ID**	説明
4052	ツール	いいボット
4055	未分類	いいボット
4056	マーケティング	いいボット
4057	スクリーンショットクリエーター	いいボット
4058	昇降補助具	いいボット
4060	検索エンジン	いいボット
4061	検索エンジン	いいボット
4062	検索エンジン	いいボット
4063	検索エンジン	いいボット
4065	スクレーパー	いいボット
4066	マーケティング	いいボット
4067	マーケティング	いいボット
4071	ツール	いいボット
4076	マーケティング	いいボット
4078	昇降補助具	いいボット
4079	昇降補助具	いいボット
4081	検索エンジン	いいボット
4082	ツール	いいボット
4085	ツール	いいボット
4086	ツール	いいボット
4090	ツール	いいボット
4091	ツール	いいボット
4092	ツール	いいボット
4093	ツール	いいボット
4094	未分類	いいボット
4095	サイトモニター	いいボット
4096	サイトモニター	いいボット
4097	サイトモニター	いいボット
4098	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
4099	検索エンジン	いいボット
4100	検索エンジン	いいボット
4101	検索エンジン	いいボット
4102	検索エンジン	いいボット
4103	マーケティング	いいボット
4104	マーケティング	いいボット
4105	マーケティング	いいボット
4106	マーケティング	いいボット
4107	マーケティング	いいボット
4108	マーケティング	いいボット
4109	検索エンジン	いいボット
4110	昇降補助具	いいボット
4111	昇降補助具	いいボット
4112	昇降補助具	いいボット
4113	脆弱性スキャナ	いいボット
4114	昇降補助具	いいボット
4115	ツール	いいボット
4120	マーケティング	いいボット
4121	マーケティング	いいボット
4122	マーケティング	いいボット
4123	マーケティング	いいボット
4124	マーケティング	いいボット
4125	マーケティング	いいボット
4126	マーケティング	いいボット
4127	マーケティング	いいボット
4128	マーケティング	いいボット
4129	マーケティング	いいボット
4130	マーケティング	いいボット
4131	ツール	いいボット

ボットシグネチャルール	**ID**	説明
4132	マーケティング	いいボット
4133	マーケティング	いいボット
4134	ツール	いいボット
4135	マーケティング	いいボット
4136	マーケティング	いいボット
4137	マーケティング	いいボット
4138	マーケティング	いいボット
4139	マーケティング	いいボット
4140	マーケティング	いいボット
4141	マーケティング	いいボット
4142	マーケティング	いいボット
4143	マーケティング	いいボット
4144	マーケティング	いいボット
4147	検索エンジン	いいボット
4148	検索エンジン	いいボット
4149	検索エンジン	いいボット
4150	検索エンジン	いいボット
4151	検索エンジン	いいボット
4152	検索エンジン	いいボット
4153	検索エンジン	いいボット
4154	検索エンジン	いいボット
4155	検索エンジン	いいボット
4156	スクリーンショットクリエーター	いいボット
4157	検索エンジン	いいボット
4158	検索エンジン	いいボット
4159	検索エンジン	いいボット
4160	スクリーンショットクリエーター	いいボット
4161	検索エンジン	いいボット
4162	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4163	ツール	いいボット
4164	検索エンジン	いいボット
4168	スピードテスター	いいボット
4170	ツール	いいボット
4172	昇降補助具	いいボット
4173	ツール	いいボット
4174	昇降補助具	いいボット
4175	昇降補助具	いいボット
4176	ツール	いいボット
4177	検索エンジン	いいボット
4178	ツール	いいボット
4179	昇降補助具	いいボット
4180	ツール	いいボット
4181	サイトモニター	いいボット
4182	サイトモニター	いいボット
4183	サイトモニター	いいボット
4184	サイトモニター	いいボット
4185	検索エンジン	いいボット
4186	ツール	いいボット
4187	ツール	いいボット
4190	検索エンジン	いいボット
4191	検索エンジン	いいボット
4192	検索エンジン	いいボット
4193	検索エンジン	いいボット
4194	ツール	いいボット
4196	ツール	いいボット
4197	ツール	いいボット
4198	マーケティング	いいボット
4199	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
4200	脆弱性スキャナ	いいボット
4201	ツール	いいボット
4202	ツール	いいボット
4205	検索エンジン	いいボット
4206	マーケティング	いいボット
4207	マーケティング	いいボット
4208	検索エンジン	いいボット
4209	検索エンジン	いいボット
4210	スピードテスター	いいボット
4211	ツール	いいボット
4212	フィードフェッチャー	いいボット
4213	フィードフェッチャー	いいボット
4215	ツール	いいボット
4216	ツール	いいボット
4219	マーケティング	いいボット
4220	ツール	いいボット
4222	サイトモニター	いいボット
4223	マーケティング	いいボット
4224	検索エンジン	いいボット
4225	検索エンジン	いいボット
4226	検索エンジン	いいボット
4227	マーケティング	いいボット
4228	マーケティング	いいボット
4229	ツール	いいボット
4231	スクリーンショットクリエーター	いいボット
4232	ツール	いいボット
4233	サイトモニター	いいボット
4234	サイトモニター	いいボット
4235	サイトモニター	いいボット

ボットシグネチャルール	**ID**	説明
4236	サイトモニター	いいボット
4237	サイトモニター	いいボット
4238	サイトモニター	いいボット
4240	マーケティング	いいボット
4241	マーケティング	いいボット
4242	マーケティング	いいボット
4243	マーケティング	いいボット
4244	マーケティング	いいボット
4245	マーケティング	いいボット
4246	マーケティング	いいボット
4247	検索エンジン	いいボット
4248	検索エンジン	いいボット
4249	スクリーンショットクリエーター	いいボット
4250	検索エンジン	いいボット
4251	検索エンジン	いいボット
4252	昇降補助具	いいボット
4253	昇降補助具	いいボット
4254	昇降補助具	いいボット
4255	ツール	いいボット
4256	未分類	いいボット
4257	ツール	いいボット
4258	昇降補助具	いいボット
4259	昇降補助具	いいボット
4260	ツール	いいボット
4261	ツール	いいボット
4262	ツール	いいボット
4265	検索エンジン	いいボット
4266	未分類	いいボット
4267	ツール	いいボット

ボットシグネチャルール	**ID**	説明
4268	ツール	いいボット
4269	検索エンジン	いいボット
4270	検索エンジン	いいボット
4271	検索エンジン	いいボット
4272	検索エンジン	いいボット
4273	検索エンジン	いいボット
4274	検索エンジン	いいボット
4275	検索エンジン	いいボット
4279	マーケティング	いいボット
4280	昇降補助具	いいボット
4282	マーケティング	いいボット
4283	マーケティング	いいボット
4284	マーケティング	いいボット
4285	マーケティング	いいボット
4286	マーケティング	いいボット
4287	マーケティング	いいボット
4288	マーケティング	いいボット
4289	マーケティング	いいボット
4290	マーケティング	いいボット
4291	マーケティング	いいボット
4292	マーケティング	いいボット
4293	マーケティング	いいボット
4294	マーケティング	いいボット
4295	検索エンジン	いいボット
4296	検索エンジン	いいボット
4297	検索エンジン	いいボット
4298	検索エンジン	いいボット
4299	検索エンジン	いいボット
4300	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4301	検索エンジン	いいボット
4302	検索エンジン	いいボット
4303	検索エンジン	いいボット
4304	検索エンジン	いいボット
4305	検索エンジン	いいボット
4306	スクリーンショットクリエーター	いいボット
4307	検索エンジン	いいボット
4308	検索エンジン	いいボット
4309	検索エンジン	いいボット
4310	検索エンジン	いいボット
4311	スクリーンショットクリエーター	いいボット
4312	検索エンジン	いいボット
4313	検索エンジン	いいボット
4314	検索エンジン	いいボット
4315	検索エンジン	いいボット
4316	検索エンジン	いいボット
4317	検索エンジン	いいボット
4318	スクリーンショットクリエーター	いいボット
4319	スクリーンショットクリエーター	いいボット
4321	未分類	いいボット
4322	昇降補助具	いいボット
4323	ツール	いいボット
4324	ツール	いいボット
4325	ツール	いいボット
4328	マーケティング	いいボット
4330	サイトモニター	いいボット
4331	検索エンジン	いいボット
4332	検索エンジン	いいボット
4335	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
4336	マーケティング	いいボット
4337	ツール	いいボット
4338	ツール	いいボット
4339	ツール	いいボット
4340	昇降補助具	いいボット
4341	昇降補助具	いいボット
4342	脆弱性スキャナ	いいボット
4343	脆弱性スキャナ	いいボット
4344	スクレーパー	いいボット
4345	マーケティング	いいボット
4346	マーケティング	いいボット
4347	マーケティング	いいボット
4348	マーケティング	いいボット
4349	マーケティング	いいボット
4350	マーケティング	いいボット
4351	マーケティング	いいボット
4352	マーケティング	いいボット
4353	マーケティング	いいボット
4354	マーケティング	いいボット
4355	検索エンジン	いいボット
4356	検索エンジン	いいボット
4357	検索エンジン	いいボット
4358	検索エンジン	いいボット
4359	検索エンジン	いいボット
4360	検索エンジン	いいボット
4361	検索エンジン	いいボット
4362	検索エンジン	いいボット
4363	検索エンジン	いいボット
4364	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4365	スクリーンショットクリエーター	いいボット
4366	検索エンジン	いいボット
4367	検索エンジン	いいボット
4368	検索エンジン	いいボット
4369	検索エンジン	いいボット
4370	スクリーンショットクリエーター	いいボット
4371	検索エンジン	いいボット
4372	検索エンジン	いいボット
4373	検索エンジン	いいボット
4374	検索エンジン	いいボット
4375	検索エンジン	いいボット
4376	スクリーンショットクリエーター	いいボット
4377	昇降補助具	いいボット
4378	昇降補助具	いいボット
4379	検索エンジン	いいボット
4380	検索エンジン	いいボット
4381	検索エンジン	いいボット
4382	検索エンジン	いいボット
4383	昇降補助具	いいボット
4384	検索エンジン	いいボット
4385	ツール	いいボット
4386	未分類	いいボット
4387	昇降補助具	いいボット
4388	昇降補助具	いいボット
4389	ツール	いいボット
4390	ツール	いいボット
4391	ツール	いいボット
4392	ツール	いいボット
4393	ツール	いいボット

ボットシグネチャルール	**ID**	説明
4394	未分類	いいボット
4395	ツール	いいボット
4396	サイトモニター	いいボット
4397	サイトモニター	いいボット
4404	検索エンジン	いいボット
4405	検索エンジン	いいボット
4406	検索エンジン	いいボット
4407	未分類	いいボット

2021 年 **10** 月のボット署名の更新

January 31, 2022

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 10 は、ビルドが 13.0 76.31 以降の NetScaler Citrix ADC プラットフォームに適用されます。

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	**ID**	説明
71	昇降補助具	いいボット
74	昇降補助具	いいボット
75	昇降補助具	いいボット
372	スクレーパー	いいボット
373	スクレーパー	いいボット
374	スクレーパー	いいボット
375	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
376	スクレーパー	いいボット
377	スクレーパー	いいボット
378	スクレーパー	いいボット
379	スクレーパー	いいボット
380	スクレーパー	いいボット
381	スクレーパー	いいボット
382	スクレーパー	いいボット
383	スクレーパー	いいボット
384	スクレーパー	いいボット
385	スクレーパー	いいボット
386	スクレーパー	いいボット
387	スクレーパー	いいボット
389	スクレーパー	いいボット
390	スクレーパー	いいボット
391	スクレーパー	いいボット
639	検索エンジン	いいボット
702	検索エンジン	いいボット
703	検索エンジン	いいボット
1173	ツール	いいボット
1174	マーケティング	いいボット
1176	検索エンジン	いいボット
1178	スピードテスター	いいボット
1185	スクリーンショットクリエーター	いいボット
1209	未分類	いいボット
1531	フィードフェッチャー	いいボット
2586	未分類	いいボット
2674	ツール	いいボット
2756	ツール	いいボット
2758	未分類	いいボット

ボットシグネチャルール	**ID**	説明
2759	ツール	いいボット
2784	ツール	いいボット
2952	ツール	いいボット
3163	ツール	いいボット
3554	ツール	いいボット
3782	マーケティング	いいボット
3788	ツール	いいボット
3789	ツール	いいボット
3797	マーケティング	いいボット
3798	マーケティング	いいボット
3799	マーケティング	いいボット
3800	マーケティング	いいボット
3801	マーケティング	いいボット
3802	スクリーンショットクリエーター	いいボット
3803	検索エンジン	いいボット
3804	スクリーンショットクリエーター	いいボット
3805	検索エンジン	いいボット
3861	マーケティング	いいボット
3862	マーケティング	いいボット
3863	マーケティング	いいボット
3864	マーケティング	いいボット
3865	マーケティング	いいボット
3866	マーケティング	いいボット
3867	マーケティング	いいボット
3868	マーケティング	いいボット
3869	ツール	いいボット
3871	マーケティング	いいボット
3872	マーケティング	いいボット
3873	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
3874	検索エンジン	いいボット
3875	検索エンジン	いいボット
3876	検索エンジン	いいボット
3877	スクリーンショットクリエーター	いいボット
3878	検索エンジン	いいボット
3879	検索エンジン	いいボット
3880	スクリーンショットクリエーター	いいボット
3881	スクリーンショットクリエーター	いいボット
3882	検索エンジン	いいボット
3883	検索エンジン	いいボット
3884	検索エンジン	いいボット
3885	検索エンジン	いいボット
3963	マーケティング	いいボット
4040	昇降補助具	いいボット
4041	ツール	いいボット
4120	マーケティング	いいボット
4122	マーケティング	いいボット
4123	マーケティング	いいボット
4124	マーケティング	いいボット
4125	マーケティング	いいボット
4133	マーケティング	いいボット
4134	ツール	いいボット
4135	マーケティング	いいボット
4136	マーケティング	いいボット
4137	マーケティング	いいボット
4138	マーケティング	いいボット
4139	マーケティング	いいボット
4140	マーケティング	いいボット
4141	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
4142	マーケティング	いいボット
4143	マーケティング	いいボット
4144	マーケティング	いいボット
4145	検索エンジン	いいボット
4146	検索エンジン	いいボット
4147	検索エンジン	いいボット
4148	検索エンジン	いいボット
4149	検索エンジン	いいボット
4150	検索エンジン	いいボット
4151	検索エンジン	いいボット
4152	検索エンジン	いいボット
4153	検索エンジン	いいボット
4154	検索エンジン	いいボット
4155	検索エンジン	いいボット
4156	スクリーンショットクリエーター	いいボット
4157	検索エンジン	いいボット
4158	検索エンジン	いいボット
4159	検索エンジン	いいボット
4160	スクリーンショットクリエーター	いいボット
4161	検索エンジン	いいボット
4162	検索エンジン	いいボット
4163	ツール	いいボット
4164	検索エンジン	いいボット
4209	検索エンジン	いいボット
4240	マーケティング	いいボット
4241	マーケティング	いいボット
4248	検索エンジン	いいボット
4249	スクリーンショットクリエーター	いいボット
4250	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4251	検索エンジン	いいボット
4282	マーケティング	いいボット
4283	マーケティング	いいボット
4284	マーケティング	いいボット
4285	マーケティング	いいボット
4286	マーケティング	いいボット
4287	マーケティング	いいボット
4288	マーケティング	いいボット
4289	マーケティング	いいボット
4290	マーケティング	いいボット
4291	マーケティング	いいボット
4292	マーケティング	いいボット
4293	マーケティング	いいボット
4294	マーケティング	いいボット
4295	検索エンジン	いいボット
4296	検索エンジン	いいボット
4297	検索エンジン	いいボット
4298	検索エンジン	いいボット
4299	検索エンジン	いいボット
4300	検索エンジン	いいボット
4301	検索エンジン	いいボット
4302	検索エンジン	いいボット
4303	検索エンジン	いいボット
4304	検索エンジン	いいボット
4305	検索エンジン	いいボット
4306	スクリーンショットクリエーター	いいボット
4307	検索エンジン	いいボット
4308	検索エンジン	いいボット
4309	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4310	検索エンジン	いいボット
4311	スクリーンショットクリエーター	いいボット
4312	検索エンジン	いいボット
4313	検索エンジン	いいボット
4314	検索エンジン	いいボット
4315	検索エンジン	いいボット
4316	検索エンジン	いいボット
4317	検索エンジン	いいボット
4318	スクリーンショットクリエーター	いいボット
4319	スクリーンショットクリエーター	いいボット
4337	ツール	いいボット
4338	ツール	いいボット
4345	マーケティング	いいボット
4346	マーケティング	いいボット
4347	マーケティング	いいボット
4348	マーケティング	いいボット
4349	マーケティング	いいボット
4350	マーケティング	いいボット
4351	マーケティング	いいボット
4352	マーケティング	いいボット
4353	マーケティング	いいボット
4354	マーケティング	いいボット
4355	検索エンジン	いいボット
4356	検索エンジン	いいボット
4357	検索エンジン	いいボット
4358	検索エンジン	いいボット
4359	検索エンジン	いいボット
4360	検索エンジン	いいボット
4361	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4362	検索エンジン	いいボット
4363	検索エンジン	いいボット
4364	検索エンジン	いいボット
4365	スクリーンショットクリエーター	いいボット
4366	検索エンジン	いいボット
4367	検索エンジン	いいボット
4368	検索エンジン	いいボット
4369	検索エンジン	いいボット
4370	スクリーンショットクリエーター	いいボット
4371	検索エンジン	いいボット
4372	検索エンジン	いいボット
4373	検索エンジン	いいボット
4374	検索エンジン	いいボット
4375	検索エンジン	いいボット
4376	スクリーンショットクリエーター	いいボット

2021 年 **11** 月のボット署名の更新

July 15, 2022

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 11 は、ビルドが 13.0 76.31 以降の NetScaler Citrix ADC プラットフォームに適用されます。

新しいボット署名

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	**ID**	説明
4408	スクレーパー	いいボット
4409	昇降補助具	悪いボット
4411	マーケティング	いいボット
4412	マーケティング	いいボット
4413	マーケティング	いいボット
4421	スクリーンショットクリエーター	いいボット
4422	昇降補助具	いいボット
4423	ツール	悪いボット
4424	サイトモニター	いいボット
4425	マーケティング	いいボット
4426	昇降補助具	悪いボット
4427	スクレーパー	いいボット
4428	スクレーパー	いいボット
4429	スクリーンショットクリエーター	いいボット
4430	ウイルススキャナー	いいボット
4431	サイトモニター	いいボット
4432	ツール	いいボット
4433	検索エンジン	いいボット
4434	検索エンジン	いいボット
4435	検索エンジン	いいボット
4436	マーケティング	いいボット
4437	マーケティング	いいボット
4438	スクレーパー	いいボット
4439	スクレーパー	いいボット
4440	スクレーパー	いいボット
4441	フィードフェッチャー	いいボット
4442	マーケティング	いいボット
4443	スクレーパー	いいボット
4445	未分類	悪いボット

ボットシグネチャルール	**ID**	説明
4446	スクレーパー	いいボット
4450	スクリーンショットクリエーター	いいボット
4451	スピードテスター	いいボット
4452	検索エンジン	いいボット
4466	未分類	いいボット
4467	スクリーンショットクリエーター	いいボット
4468	ツール	いいボット
4469	未分類	いいボット
4470	ツール	いいボット
4472	スクレーパー	いいボット
4473	未分類	いいボット
4474	マーケティング	いいボット
4476	昇降補助具	いいボット
4477	昇降補助具	いいボット
4478	昇降補助具	いいボット
4479	昇降補助具	いいボット
4480	昇降補助具	いいボット
4481	昇降補助具	いいボット
4482	昇降補助具	いいボット
4483	昇降補助具	いいボット
4484	昇降補助具	いいボット
4485	昇降補助具	いいボット
4486	スクレーパー	いいボット
4487	スクレーパー	いいボット
4488	スクレーパー	いいボット
4489	検索エンジン	いいボット
4491	ツール	いいボット
4492	未分類	悪いボット
4493	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
4494	ツール	いいボット
4496	ツール	いいボット
4497	昇降補助具	いいボット
4498	未分類	悪いボット
4499	未分類	悪いボット
4501	マーケティング	いいボット
4502	マーケティング	いいボット
4503	マーケティング	いいボット
4508	未分類	いいボット
4509	未分類	いいボット
4510	未分類	いいボット
4511	未分類	いいボット
4512	ツール	いいボット
4513	ツール	いいボット
4514	ツール	いいボット
4515	ツール	いいボット
4516	未分類	いいボット
4518	スクレーパー	悪いボット
4519	スクリーンショットクリエーター	いいボット
4520	マーケティング	いいボット
4521	未分類	いいボット
4522	ツール	いいボット
4523	未分類	悪いボット
4524	未分類	悪いボット
4525	昇降補助具	いいボット
4526	昇降補助具	いいボット
4527	昇降補助具	いいボット
4528	昇降補助具	いいボット
4529	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
4530	未分類	悪いボット
4531	マーケティング	いいボット
4532	マーケティング	いいボット
4533	マーケティング	いいボット
4534	マーケティング	いいボット
4535	マーケティング	いいボット
4541	マーケティング	いいボット
4552	未分類	いいボット
4553	ツール	悪いボット
4554	ツール	悪いボット
4555	ツール	いいボット
4556	ツール	いいボット
4558	スクレーパー	いいボット
4559	昇降補助具	いいボット
4560	昇降補助具	いいボット
4561	サイトモニター	いいボット
4562	検索エンジン	いいボット
4563	検索エンジン	いいボット
1000000	Web ブラウザー	いいボット
1000001	スクレーパー	悪いボット
1000002	アプリケーション	悪いボット
1000003	Web ブラウザー	いいボット
1000004	スクレーパー	いいボット
1000005	スクレーパー	いいボット
1000006	昇降補助具	悪いボット
1000007	Web ブラウザー	悪いボット
1000008	未分類	悪いボット
1000009	Web ブラウザー	いいボット
1000010	スクレーパー	悪いボット

ボットシグネチャルール	**ID**	説明
1000011	Web ブラウザー	悪いボット
1000012	Web ブラウザー	いいボット
1000013	Web ブラウザー	悪いボット
1000014	スクレーパー	いいボット
1000015	スクレーパー	悪いボット
1000016	スクレーパー	悪いボット
1000017	Web ブラウザー	いいボット
1000018	Web ブラウザー	悪いボット
1000019	未分類	悪いボット
1000020	スクレーパー	いいボット
1000021	Web ブラウザー	悪いボット
1000022	スクレーパー	いいボット
1000023	スクレーパー	いいボット
1000024	昇降補助具	いいボット
1000025	Web ブラウザー	悪いボット
1000026	Analyzer	いいボット
1000027	Analyzer	いいボット
1000028	Analyzer	いいボット
1000029	Analyzer	いいボット
1000030	Analyzer	いいボット
1000031	Web ブラウザー	いいボット
1000032	Analyzer	いいボット
1000033	Analyzer	いいボット
1000034	Web ブラウザー	悪いボット
1000035	スクレーパー	いいボット
1000036	スクレーパー	いいボット
1000037	Analyzer	いいボット
1000038	Analyzer	いいボット
1000039	Analyzer	いいボット

ボットシグネチャルール	**ID**	説明
1000040	Analyzer	いいボット
1000041	スクレーパー	いいボット
1000042	Analyzer	いいボット
1000043	Analyzer	いいボット
1000044	昇降補助具	いいボット
1000045	Web ブラウザー	悪いボット
1000046	Web ブラウザー	悪いボット
1000047	スクレーパー	いいボット
1000048	Web ブラウザー	悪いボット
1000049	Analyzer	いいボット
1000050	Web ブラウザー	悪いボット
1000051	Web ブラウザー	いいボット
1000052	Web ブラウザー	悪いボット
1000053	スクレーパー	いいボット
1000054	Web ブラウザー	いいボット
1000055	Web ブラウザー	いいボット
1000056	スクレーパー	悪いボット
1000057	昇降補助具	悪いボット
1000058	スクレーパー	悪いボット
1000059	Analyzer	いいボット
1000060	Web ブラウザー	悪いボット
1000061	Web ブラウザー	悪いボット
1000062	Web ブラウザー	悪いボット
1000063	スクレーパー	悪いボット
1000064	スクレーパー	悪いボット
1000065	スクレーパー	悪いボット
1000066	アプリケーション	悪いボット
1000067	スクレーパー	悪いボット
1000068	Web ブラウザー	悪いボット

ボットシグネチャルール	**ID**	説明
1000069	スクレーパー	悪いボット
1000070	スクレーパー	いいボット
1000071	Web ブラウザー	いいボット
1000072	Web ブラウザー	いいボット
1000073	Web ブラウザー	悪いボット
1000074	Web ブラウザー	悪いボット
1000075	アプリケーション	悪いボット
1000076	スクレーパー	悪いボット

ボットの署名を更新しました

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	**ID**	説明
2	昇降補助具	いいボット
5	昇降補助具	いいボット
9	昇降補助具	いいボット
30	昇降補助具	悪いボット
45	昇降補助具	いいボット
46	昇降補助具	いいボット
48	昇降補助具	いいボット
52	昇降補助具	いいボット
60	昇降補助具	いいボット
61	昇降補助具	いいボット
63	昇降補助具	いいボット
67	昇降補助具	いいボット
76	昇降補助具	いいボット
78	昇降補助具	いいボット
79	昇降補助具	いいボット
80	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
81	昇降補助具	いいボット
82	昇降補助具	いいボット
83	昇降補助具	いいボット
84	昇降補助具	いいボット
87	昇降補助具	いいボット
90	昇降補助具	いいボット
95	昇降補助具	いいボット
96	昇降補助具	いいボット
97	昇降補助具	いいボット
100	昇降補助具	いいボット
101	昇降補助具	いいボット
102	昇降補助具	いいボット
103	昇降補助具	いいボット
104	昇降補助具	いいボット
107	昇降補助具	いいボット
108	昇降補助具	いいボット
110	昇降補助具	いいボット
111	昇降補助具	いいボット
114	昇降補助具	いいボット
115	昇降補助具	いいボット
123	昇降補助具	いいボット
135	昇降補助具	いいボット
136	昇降補助具	いいボット
137	昇降補助具	いいボット
140	昇降補助具	いいボット
141	昇降補助具	いいボット
143	昇降補助具	いいボット
144	昇降補助具	いいボット
145	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
146	昇降補助具	いいボット
147	昇降補助具	いいボット
149	昇降補助具	いいボット
152	昇降補助具	いいボット
155	昇降補助具	いいボット
156	昇降補助具	いいボット
157	昇降補助具	いいボット
158	昇降補助具	いいボット
159	昇降補助具	いいボット
160	昇降補助具	いいボット
161	昇降補助具	いいボット
162	昇降補助具	いいボット
163	昇降補助具	いいボット
164	昇降補助具	いいボット
165	昇降補助具	いいボット
166	昇降補助具	いいボット
167	昇降補助具	いいボット
172	昇降補助具	いいボット
173	昇降補助具	いいボット
174	昇降補助具	いいボット
176	昇降補助具	いいボット
177	昇降補助具	いいボット
180	昇降補助具	いいボット
182	昇降補助具	いいボット
187	昇降補助具	いいボット
197	昇降補助具	いいボット
201	昇降補助具	いいボット
202	昇降補助具	いいボット
203	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
206	昇降補助具	いいボット
217	フィードフェッチャー	いいボット
219	フィードフェッチャー	いいボット
229	スクレーパー	いいボット
235	スクレーパー	いいボット
236	スクレーパー	いいボット
237	スクレーパー	いいボット
248	スクレーパー	いいボット
250	スクレーパー	いいボット
252	スクレーパー	いいボット
260	スクレーパー	いいボット
263	スクレーパー	いいボット
265	スクレーパー	いいボット
267	スクレーパー	いいボット
268	スクレーパー	いいボット
271	スクレーパー	いいボット
272	スクレーパー	いいボット
276	スクレーパー	いいボット
277	スクレーパー	いいボット
278	スクレーパー	いいボット
279	スクレーパー	いいボット
280	スクレーパー	いいボット
281	スクレーパー	いいボット
283	スクレーパー	いいボット
285	スクレーパー	いいボット
286	スクレーパー	いいボット
287	スクレーパー	いいボット
290	スクレーパー	いいボット
292	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
293	スクレーパー	いいボット
338	スクレーパー	いいボット
342	スクレーパー	いいボット
343	スクレーパー	いいボット
344	スクレーパー	いいボット
351	スクレーパー	いいボット
352	スクレーパー	いいボット
353	スクレーパー	いいボット
355	スクレーパー	いいボット
357	スクレーパー	いいボット
360	スクレーパー	いいボット
362	スクレーパー	いいボット
366	スクレーパー	いいボット
370	スクレーパー	いいボット
371	スクレーパー	いいボット
392	スクレーパー	いいボット
393	スクレーパー	いいボット
394	スクレーパー	いいボット
396	スクレーパー	いいボット
397	スクレーパー	いいボット
414	スクレーパー	いいボット
418	スクレーパー	いいボット
419	スクレーパー	いいボット
421	スクレーパー	いいボット
422	スクレーパー	いいボット
423	スクレーパー	いいボット
424	スクレーパー	いいボット
425	スクレーパー	いいボット
426	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
427	スクレーパー	いいボット
428	スクレーパー	いいボット
430	スクレーパー	いいボット
432	スクレーパー	いいボット
433	スクレーパー	いいボット
434	スクレーパー	いいボット
435	スクレーパー	いいボット
441	スクレーパー	いいボット
445	スクレーパー	いいボット
446	スクレーパー	いいボット
451	スクレーパー	いいボット
452	スクレーパー	いいボット
454	スクレーパー	いいボット
455	スクレーパー	いいボット
456	スクレーパー	いいボット
457	スクレーパー	いいボット
458	スクレーパー	いいボット
461	スクレーパー	いいボット
465	スクレーパー	いいボット
466	スクレーパー	いいボット
469	スクレーパー	いいボット
473	スクレーパー	いいボット
474	スクレーパー	いいボット
476	スクレーパー	いいボット
477	スクレーパー	いいボット
484	スクレーパー	いいボット
485	スクレーパー	いいボット
487	スクレーパー	いいボット
488	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
489	スクレーパー	いいボット
490	スクレーパー	いいボット
493	スクレーパー	いいボット
494	スクレーパー	いいボット
495	スクレーパー	いいボット
497	スクレーパー	いいボット
498	スクレーパー	いいボット
499	スクレーパー	いいボット
500	スクレーパー	いいボット
505	スクレーパー	いいボット
506	スクレーパー	いいボット
507	スクレーパー	いいボット
512	スクレーパー	いいボット
513	スクレーパー	いいボット
514	スクレーパー	いいボット
527	スクレーパー	いいボット
533	スクレーパー	いいボット
539	スクレーパー	いいボット
540	スクレーパー	いいボット
542	スクレーパー	いいボット
544	スクレーパー	いいボット
545	スクレーパー	いいボット
546	スクレーパー	いいボット
547	スクレーパー	いいボット
548	スクレーパー	いいボット
551	スクレーパー	いいボット
552	スクレーパー	いいボット
554	スクレーパー	いいボット
556	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
558	スクレーパー	いいボット
560	スクレーパー	いいボット
561	スクレーパー	いいボット
566	スクレーパー	いいボット
575	スクレーパー	いいボット
578	スクレーパー	いいボット
581	スクレーパー	いいボット
582	スクレーパー	いいボット
591	スクレーパー	いいボット
593	スクレーパー	いいボット
595	スクレーパー	いいボット
600	スクレーパー	いいボット
601	スクレーパー	いいボット
602	スクレーパー	いいボット
604	スクレーパー	いいボット
605	スクレーパー	いいボット
609	スクレーパー	いいボット
610	スクレーパー	いいボット
611	スクレーパー	いいボット
612	スクレーパー	いいボット
613	スクレーパー	いいボット
615	スクレーパー	いいボット
620	検索エンジン	いいボット
622	検索エンジン	いいボット
623	検索エンジン	いいボット
624	検索エンジン	いいボット
626	検索エンジン	いいボット
627	検索エンジン	いいボット
628	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
629	検索エンジン	いいボット
633	検索エンジン	いいボット
634	検索エンジン	いいボット
636	検索エンジン	いいボット
637	検索エンジン	いいボット
640	検索エンジン	いいボット
641	検索エンジン	いいボット
642	検索エンジン	いいボット
643	検索エンジン	いいボット
647	検索エンジン	いいボット
649	検索エンジン	いいボット
650	検索エンジン	いいボット
651	検索エンジン	いいボット
654	検索エンジン	いいボット
656	検索エンジン	いいボット
657	検索エンジン	いいボット
658	検索エンジン	いいボット
659	検索エンジン	いいボット
660	検索エンジン	いいボット
663	検索エンジン	いいボット
664	検索エンジン	いいボット
665	検索エンジン	いいボット
666	検索エンジン	いいボット
667	検索エンジン	いいボット
669	検索エンジン	いいボット
670	検索エンジン	いいボット
671	検索エンジン	いいボット
672	検索エンジン	いいボット
673	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
674	検索エンジン	いいボット
675	検索エンジン	いいボット
676	検索エンジン	いいボット
677	検索エンジン	いいボット
679	検索エンジン	いいボット
680	検索エンジン	いいボット
690	検索エンジン	いいボット
693	検索エンジン	いいボット
694	検索エンジン	いいボット
697	検索エンジン	いいボット
698	検索エンジン	いいボット
702	検索エンジン	いいボット
706	検索エンジン	いいボット
712	検索エンジン	いいボット
713	検索エンジン	いいボット
714	検索エンジン	いいボット
715	検索エンジン	いいボット
716	検索エンジン	いいボット
721	検索エンジン	いいボット
723	検索エンジン	いいボット
725	検索エンジン	いいボット
727	検索エンジン	いいボット
728	検索エンジン	いいボット
729	検索エンジン	いいボット
730	検索エンジン	いいボット
731	検索エンジン	いいボット
732	検索エンジン	いいボット
735	検索エンジン	いいボット
736	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
740	検索エンジン	いいボット
748	検索エンジン	いいボット
749	検索エンジン	いいボット
750	検索エンジン	いいボット
751	検索エンジン	いいボット
756	検索エンジン	いいボット
757	検索エンジン	いいボット
758	検索エンジン	いいボット
759	検索エンジン	いいボット
760	検索エンジン	いいボット
761	検索エンジン	いいボット
762	検索エンジン	いいボット
763	検索エンジン	いいボット
764	検索エンジン	いいボット
765	検索エンジン	いいボット
766	検索エンジン	いいボット
767	検索エンジン	いいボット
768	検索エンジン	いいボット
769	検索エンジン	いいボット
770	検索エンジン	いいボット
771	検索エンジン	いいボット
772	検索エンジン	いいボット
773	検索エンジン	いいボット
776	検索エンジン	いいボット
777	検索エンジン	いいボット
780	検索エンジン	いいボット
781	検索エンジン	いいボット
784	検索エンジン	いいボット
786	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
787	検索エンジン	いいボット
788	検索エンジン	いいボット
789	検索エンジン	いいボット
790	検索エンジン	いいボット
791	検索エンジン	いいボット
792	検索エンジン	いいボット
795	検索エンジン	いいボット
796	検索エンジン	いいボット
798	検索エンジン	いいボット
800	検索エンジン	いいボット
801	検索エンジン	いいボット
802	検索エンジン	いいボット
803	検索エンジン	いいボット
805	検索エンジン	いいボット
806	検索エンジン	いいボット
807	検索エンジン	いいボット
809	検索エンジン	いいボット
810	検索エンジン	いいボット
811	検索エンジン	いいボット
812	検索エンジン	いいボット
814	検索エンジン	いいボット
815	検索エンジン	いいボット
816	検索エンジン	いいボット
817	検索エンジン	いいボット
818	検索エンジン	いいボット
819	検索エンジン	いいボット
820	検索エンジン	いいボット
821	検索エンジン	いいボット
822	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
823	検索エンジン	いいボット
825	検索エンジン	いいボット
827	検索エンジン	いいボット
830	検索エンジン	いいボット
831	検索エンジン	いいボット
834	検索エンジン	いいボット
837	検索エンジン	いいボット
838	検索エンジン	いいボット
849	サイトモニター	いいボット
850	サイトモニター	いいボット
851	サイトモニター	いいボット
853	サイトモニター	いいボット
857	サイトモニター	いいボット
858	サイトモニター	いいボット
859	サイトモニター	いいボット
860	サイトモニター	いいボット
861	サイトモニター	いいボット
862	サイトモニター	いいボット
863	サイトモニター	いいボット
864	サイトモニター	いいボット
865	サイトモニター	いいボット
866	サイトモニター	いいボット
867	サイトモニター	いいボット
868	サイトモニター	いいボット
869	サイトモニター	いいボット
870	サイトモニター	いいボット
871	サイトモニター	いいボット
872	サイトモニター	いいボット
873	サイトモニター	いいボット

ボットシグネチャルール	**ID**	説明
874	サイトモニター	いいボット
875	サイトモニター	いいボット
876	サイトモニター	いいボット
877	サイトモニター	いいボット
880	サイトモニター	いいボット
881	サイトモニター	いいボット
883	サイトモニター	いいボット
885	サイトモニター	いいボット
886	サイトモニター	いいボット
888	サイトモニター	いいボット
889	サイトモニター	いいボット
895	サイトモニター	いいボット
896	サイトモニター	いいボット
897	サイトモニター	いいボット
898	サイトモニター	いいボット
900	サイトモニター	いいボット
901	サイトモニター	いいボット
904	サイトモニター	いいボット
906	サイトモニター	いいボット
908	サイトモニター	いいボット
909	サイトモニター	いいボット
910	サイトモニター	いいボット
911	サイトモニター	いいボット
912	サイトモニター	いいボット
913	サイトモニター	いいボット
917	サイトモニター	いいボット
918	サイトモニター	いいボット
919	サイトモニター	いいボット
920	サイトモニター	いいボット

ボットシグネチャルール	**ID**	説明
921	サイトモニター	いいボット
924	サイトモニター	いいボット
926	サイトモニター	いいボット
927	サイトモニター	いいボット
928	サイトモニター	いいボット
929	サイトモニター	いいボット
930	サイトモニター	いいボット
931	サイトモニター	いいボット
934	サイトモニター	いいボット
938	サイトモニター	いいボット
939	サイトモニター	いいボット
958	サイトモニター	いいボット
959	サイトモニター	いいボット
960	サイトモニター	いいボット
963	サイトモニター	いいボット
984	スクレーパー	いいボット
991	スクレーパー	悪いボット
996	スクレーパー	いいボット
997	スクレーパー	いいボット
998	スクレーパー	いいボット
1002	スクレーパー	いいボット
1006	スクレーパー	いいボット
1622	スクリーンショットクリエーター	いいボット
2810	昇降補助具	いいボット
3432	未分類	悪いボット
3783	検索エンジン	いいボット
3784	スクレーパー	悪いボット
3788	ツール	いいボット
3790	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
3791	スピードテスター	いいボット
3792	ツール	いいボット
3793	ツール	いいボット
3794	昇降補助具	いいボット
3796	スクレーパー	いいボット
3797	マーケティング	いいボット
3799	マーケティング	いいボット
3800	マーケティング	いいボット
3806	ツール	いいボット
3807	昇降補助具	いいボット
3808	昇降補助具	いいボット
3809	ツール	いいボット
3810	スクレーパー	いいボット
3811	ツール	いいボット
3812	昇降補助具	いいボット
3813	ツール	いいボット
3814	昇降補助具	いいボット
3815	未分類	いいボット
3817	ツール	いいボット
3818	ツール	いいボット
3819	ツール	いいボット
3820	昇降補助具	いいボット
3821	検索エンジン	いいボット
3822	マーケティング	いいボット
3823	未分類	いいボット
3831	スクレーパー	いいボット
3833	検索エンジン	いいボット
3834	検索エンジン	いいボット
3835	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
3836	未分類	いいボット
3838	未分類	いいボット
3839	マーケティング	いいボット
3840	昇降補助具	いいボット
3842	昇降補助具	いいボット
3843	昇降補助具	いいボット
3844	マーケティング	いいボット
3845	マーケティング	いいボット
3846	マーケティング	いいボット
3847	マーケティング	いいボット
3848	未分類	いいボット
3849	昇降補助具	いいボット
3850	ツール	いいボット
3851	未分類	いいボット
3852	ツール	いいボット
3853	脆弱性スキャナ	いいボット
3854	昇降補助具	いいボット
3855	昇降補助具	いいボット
3856	ツール	いいボット
3871	マーケティング	いいボット
3886	ツール	いいボット
3887	昇降補助具	いいボット
3888	昇降補助具	いいボット
3889	未分類	いいボット
3890	マーケティング	いいボット
3893	昇降補助具	いいボット
3894	ツール	いいボット
3895	ツール	いいボット
3896	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
3897	ツール	いいボット
3898	ツール	いいボット
3899	未分類	いいボット
3901	昇降補助具	いいボット
3902	ツール	いいボット
3903	ツール	いいボット
3904	検索エンジン	いいボット
3905	検索エンジン	いいボット
3906	検索エンジン	いいボット
3907	検索エンジン	いいボット
3912	昇降補助具	いいボット
3917	未分類	いいボット
3918	昇降補助具	いいボット
3919	未分類	いいボット
3920	未分類	いいボット
3921	未分類	いいボット
3922	未分類	いいボット
3923	未分類	いいボット
3924	未分類	いいボット
3925	未分類	いいボット
3926	マーケティング	いいボット
3927	マーケティング	いいボット
3928	マーケティング	いいボット
3929	ツール	いいボット
3930	マーケティング	いいボット
3931	未分類	いいボット
3932	昇降補助具	いいボット
3933	マーケティング	いいボット
3934	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
3935	スクレーパー	いいボット
3936	マーケティング	いいボット
3937	スクレーパー	いいボット
3938	フィードフェッチャー	いいボット
3940	検索エンジン	いいボット
3941	昇降補助具	いいボット
3942	スクレーパー	いいボット
3946	フィードフェッチャー	いいボット
3947	昇降補助具	いいボット
3950	ウイルススキャナー	いいボット
3951	マーケティング	いいボット
3952	マーケティング	いいボット
3953	マーケティング	いいボット
3954	マーケティング	いいボット
3955	マーケティング	いいボット
3956	マーケティング	いいボット
3957	マーケティング	いいボット
3958	マーケティング	いいボット
3959	マーケティング	いいボット
3960	マーケティング	いいボット
3961	マーケティング	いいボット
3962	マーケティング	いいボット
3963	マーケティング	いいボット
3964	マーケティング	いいボット
3965	マーケティング	いいボット
3966	マーケティング	いいボット
3967	マーケティング	いいボット
3968	マーケティング	いいボット
3969	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
3970	検索エンジン	いいボット
3971	スクリーンショットクリエーター	いいボット
3972	スクリーンショットクリエーター	いいボット
3973	検索エンジン	いいボット
3974	検索エンジン	いいボット
3975	検索エンジン	いいボット
3976	検索エンジン	いいボット
3977	検索エンジン	いいボット
3978	スクリーンショットクリエーター	いいボット
3979	検索エンジン	いいボット
3980	スクリーンショットクリエーター	いいボット
3981	検索エンジン	いいボット
3982	検索エンジン	いいボット
3983	検索エンジン	いいボット
3984	検索エンジン	いいボット
3985	検索エンジン	いいボット
3986	検索エンジン	いいボット
3987	スクリーンショットクリエーター	いいボット
3988	検索エンジン	いいボット
3989	検索エンジン	いいボット
3990	検索エンジン	いいボット
3991	検索エンジン	いいボット
3992	検索エンジン	いいボット
3993	検索エンジン	いいボット
3994	検索エンジン	いいボット
3995	検索エンジン	いいボット
3996	検索エンジン	いいボット
3997	検索エンジン	いいボット
3998	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
3999	検索エンジン	いいボット
4000	スクリーンショットクリエーター	いいボット
4001	検索エンジン	いいボット
4002	検索エンジン	いいボット
4003	検索エンジン	いいボット
4004	検索エンジン	いいボット
4005	スクリーンショットクリエーター	いいボット
4006	昇降補助具	いいボット
4007	マーケティング	いいボット
4008	マーケティング	いいボット
4011	ツール	いいボット
4012	昇降補助具	いいボット
4013	検索エンジン	いいボット
4014	ツール	いいボット
4015	昇降補助具	いいボット
4016	昇降補助具	いいボット
4017	ツール	いいボット
4018	ツール	いいボット
4019	ツール	いいボット
4020	ツール	いいボット
4021	マーケティング	いいボット
4024	ツール	いいボット
4025	検索エンジン	いいボット
4026	検索エンジン	いいボット
4027	検索エンジン	いいボット
4028	マーケティング	いいボット
4029	ツール	いいボット
4030	スクレーパー	いいボット
4031	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
4033	昇降補助具	いいボット
4034	昇降補助具	いいボット
4035	マーケティング	いいボット
4036	脆弱性スキャナ	いいボット
4037	脆弱性スキャナ	いいボット
4038	未分類	悪いボット
4039	ツール	いいボット
4042	昇降補助具	いいボット
4043	スクリーンショットクリエーター	いいボット
4048	フィードフェッチャー	いいボット
4050	昇降補助具	いいボット
4051	昇降補助具	いいボット
4052	ツール	いいボット
4053	ツール	いいボット
4055	未分類	いいボット
4056	マーケティング	いいボット
4057	スクリーンショットクリエーター	いいボット
4058	昇降補助具	いいボット
4060	検索エンジン	いいボット
4061	検索エンジン	いいボット
4062	検索エンジン	いいボット
4063	検索エンジン	いいボット
4064	ツール	いいボット
4065	スクレーパー	いいボット
4066	マーケティング	いいボット
4067	マーケティング	いいボット
4071	ツール	いいボット
4076	マーケティング	いいボット
4077	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
4078	昇降補助具	いいボット
4079	昇降補助具	いいボット
4081	検索エンジン	いいボット
4082	ツール	いいボット
4085	ツール	いいボット
4086	ツール	いいボット
4087	ツール	悪いボット
4088	検索エンジン	いいボット
4089	マーケティング	いいボット
4090	ツール	いいボット
4091	ツール	いいボット
4092	ツール	いいボット
4093	ツール	いいボット
4094	未分類	いいボット
4095	サイトモニター	いいボット
4096	サイトモニター	いいボット
4097	サイトモニター	いいボット
4098	昇降補助具	いいボット
4099	検索エンジン	いいボット
4100	検索エンジン	いいボット
4101	検索エンジン	いいボット
4102	検索エンジン	いいボット
4103	マーケティング	いいボット
4104	マーケティング	いいボット
4105	マーケティング	いいボット
4106	マーケティング	いいボット
4109	検索エンジン	いいボット
4110	昇降補助具	いいボット
4111	昇降補助具	いいボット

ボットシグネチャルール	**ID**	説明
4112	昇降補助具	いいボット
4113	脆弱性スキャナ	いいボット
4114	昇降補助具	いいボット
4115	ツール	いいボット
4121	マーケティング	いいボット
4126	マーケティング	いいボット
4127	マーケティング	いいボット
4128	マーケティング	いいボット
4129	マーケティング	いいボット
4130	マーケティング	いいボット
4131	ツール	いいボット
4132	マーケティング	いいボット
4165	マーケティング	いいボット
4168	スピードテスター	いいボット
4170	ツール	いいボット
4172	昇降補助具	いいボット
4173	ツール	いいボット
4174	昇降補助具	いいボット
4175	昇降補助具	いいボット
4176	ツール	いいボット
4177	検索エンジン	いいボット
4178	ツール	いいボット
4179	昇降補助具	いいボット
4180	ツール	いいボット
4181	サイトモニター	いいボット
4182	サイトモニター	いいボット
4183	サイトモニター	いいボット
4184	サイトモニター	いいボット
4185	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4186	ツール	いいボット
4187	ツール	いいボット
4188	スクリーンショットクリエーター	いいボット
4189	マーケティング	いいボット
4190	検索エンジン	いいボット
4191	検索エンジン	いいボット
4192	検索エンジン	いいボット
4193	検索エンジン	いいボット
4194	ツール	いいボット
4196	ツール	いいボット
4197	ツール	いいボット
4198	マーケティング	いいボット
4199	マーケティング	いいボット
4200	脆弱性スキャナ	いいボット
4201	ツール	いいボット
4202	ツール	いいボット
4205	検索エンジン	いいボット
4209	検索エンジン	いいボット
4210	スピードテスター	いいボット
4211	ツール	いいボット
4212	フィードフェッチャー	いいボット
4213	フィードフェッチャー	いいボット
4215	ツール	いいボット
4216	ツール	いいボット
4219	マーケティング	いいボット
4220	ツール	いいボット
4222	サイトモニター	いいボット
4223	マーケティング	いいボット
4224	検索エンジン	いいボット

ボットシグネチャルール	**ID**	説明
4225	検索エンジン	いいボット
4226	検索エンジン	いいボット
4227	マーケティング	いいボット
4228	マーケティング	いいボット
4229	ツール	いいボット
4231	スクリーンショットクリエーター	いいボット
4232	ツール	いいボット
4233	サイトモニター	いいボット
4236	サイトモニター	いいボット
4242	マーケティング	いいボット
4243	マーケティング	いいボット
4244	マーケティング	いいボット
4245	マーケティング	いいボット
4246	マーケティング	いいボット
4247	検索エンジン	いいボット
4252	昇降補助具	いいボット
4253	昇降補助具	いいボット
4254	昇降補助具	いいボット
4255	ツール	いいボット
4256	未分類	いいボット
4257	ツール	いいボット
4258	昇降補助具	いいボット
4259	昇降補助具	いいボット
4260	ツール	いいボット
4261	ツール	いいボット
4262	ツール	いいボット
4263	マーケティング	いいボット
4265	検索エンジン	いいボット
4266	未分類	いいボット

ボットシグネチャルール	**ID**	説明
4267	ツール	いいボット
4268	ツール	いいボット
4269	検索エンジン	いいボット
4270	検索エンジン	いいボット
4271	検索エンジン	いいボット
4272	検索エンジン	いいボット
4273	検索エンジン	いいボット
4274	検索エンジン	いいボット
4275	検索エンジン	いいボット
4279	マーケティング	いいボット
4280	昇降補助具	いいボット
4321	未分類	いいボット
4322	昇降補助具	いいボット
4323	ツール	いいボット
4324	ツール	いいボット
4325	ツール	いいボット
4327	検索エンジン	いいボット
4328	マーケティング	いいボット
4330	サイトモニター	いいボット
4331	検索エンジン	いいボット
4334	スクレーパー	いいボット
4335	マーケティング	いいボット
4336	マーケティング	いいボット
4339	ツール	いいボット
4340	昇降補助具	いいボット
4341	昇降補助具	いいボット
4342	脆弱性スキャナ	いいボット
4343	脆弱性スキャナ	いいボット
4344	スクレーパー	いいボット

ボットシグネチャルール	**ID**	説明
4377	昇降補助具	いいボット
4378	昇降補助具	いいボット
4379	検索エンジン	いいボット
4380	検索エンジン	いいボット
4381	検索エンジン	いいボット
4382	検索エンジン	いいボット
4383	昇降補助具	いいボット
4384	検索エンジン	いいボット
4385	ツール	いいボット
4386	未分類	いいボット
4387	昇降補助具	いいボット
4388	昇降補助具	いいボット
4389	ツール	いいボット
4390	ツール	いいボット
4391	ツール	いいボット
4392	ツール	いいボット
4393	ツール	いいボット
4394	未分類	いいボット
4395	ツール	いいボット
4396	サイトモニター	いいボット
4397	サイトモニター	いいボット
4404	検索エンジン	いいボット
4405	検索エンジン	いいボット
4406	検索エンジン	いいボット
4407	未分類	いいボット

2022 年 3 月のボット署名の更新

July 28, 2022

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 12 は、13.0 76.31 以降のビルドの Citrix ADC プラットフォームに適用されます。

新しいボット署名

以下は、ボット署名ルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	**ID**	説明
4564	マーケティング	いいボット
4565	マーケティング	いいボット
4566	マーケティング	いいボット
4567	マーケティング	いいボット
4568	マーケティング	いいボット
4569	未分類	悪いボット
4570	未分類	悪いボット
4571	昇降補助具	いいボット
4572	昇降補助具	いいボット
4573	未分類	悪いボット
4574	未分類	悪いボット
4575	マーケティング	いいボット
4576	マーケティング	いいボット
4577	マーケティング	いいボット
4578	マーケティング	いいボット
4579	マーケティング	いいボット
4580	マーケティング	いいボット
4581	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
4582	マーケティング	いいボット
4583	スクリーンショットクリエーター	いいボット
4584	検索エンジン	いいボット
4585	検索エンジン	いいボット
4586	スクリーンショットクリエーター	いいボット
4587	未分類	いいボット
4588	スピードテスター	いいボット
4589	昇降補助具	いいボット
4590	ツール	いいボット
4591	ツール	いいボット
4592	昇降補助具	悪いボット
4593	検索エンジン	いいボット
4594	検索エンジン	いいボット
4595	検索エンジン	いいボット
4596	マーケティング	いいボット
4597	ツール	いいボット
4598	検索エンジン	いいボット
4599	マーケティング	いいボット
4600	マーケティング	いいボット
4601	マーケティング	いいボット
4602	検索エンジン	いいボット
4603	未分類	いいボット
4604	マーケティング	いいボット
4605	マーケティング	いいボット
4606	未分類	悪いボット
4607	未分類	悪いボット
4608	ツール	いいボット
4609	未分類	悪いボット
4610	ツール	いいボット

ボットシグネチャルール	**ID**	説明
4611	ツール	いいボット
4612	スクレーパー	いいボット
4613	未分類	いいボット
4614	未分類	いいボット
4615	サイトモニター	いいボット
4616	昇降補助具	いいボット
4617	サイトモニター	いいボット
4618	検索エンジン	いいボット
4619	マーケティング	いいボット
4620	マーケティング	いいボット
4621	検索エンジン	いいボット
4622	昇降補助具	いいボット
4623	昇降補助具	いいボット
4624	昇降補助具	いいボット
4625	スクレーパー	いいボット
4626	昇降補助具	いいボット
4627	脆弱性スキャナ	いいボット
4628	ツール	いいボット
4629	未分類	悪いボット
4630	未分類	悪いボット
4631	ツール	いいボット
4632	フィードフェッチャー	いいボット
4633	昇降補助具	悪いボット
4634	未分類	いいボット
4635	フィードフェッチャー	いいボット
4636	未分類	いいボット
4637	ツール	いいボット
4638	ツール	いいボット
4639	スクレーパー	悪いボット

ボットシグネチャルール	**ID**	説明
4640	未分類	悪いボット
4641	ツール	いいボット
4642	昇降補助具	悪いボット
4643	サイトモニター	いいボット
4644	サイトモニター	いいボット
4645	検索エンジン	いいボット
4646	検索エンジン	いいボット
4647	検索エンジン	いいボット
4648	検索エンジン	いいボット
4649	検索エンジン	悪いボット
4650	未分類	いいボット

ボットの署名を更新しました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	**ID**	説明
2554	未分類	悪いボット
3835	検索エンジン	いいボット
4027	検索エンジン	いいボット
4038	未分類	悪いボット
4085	ツール	いいボット
4098	昇降補助具	いいボット
4100	検索エンジン	いいボット
4220	ツール	いいボット
4224	検索エンジン	いいボット
4281	未分類	悪いボット
4412	マーケティング	いいボット
4425	マーケティング	いいボット
4429	スクリーンショットクリエーター	いいボット

ボットシグネチャルール	**ID**	説明
4430	ウイルススキャナー	いいボット
4483	昇降補助具	いいボット
4552	未分類	いいボット
4562	検索エンジン	いいボット
1000000	Web ブラウザー	いいボット
1000003	Web ブラウザー	いいボット
1000004	スクレーパー	いいボット
1000005	Google_Crawler	悪いボット
1000006	Web ブラウザー	悪いボット
1000007	ボット	悪いボット
1000008	Web ブラウザー	悪いボット
1000009	Web ブラウザー	いいボット
1000010	ボット	悪いボット
1000011	Web ブラウザー	悪いボット
1000012	スクレーパー	いいボット
1000013	スクレーパー	悪いボット
1000014	スクレーパー	悪いボット
1000015	Web ブラウザー	いいボット
1000016	ボット	悪いボット
1000017	Web ブラウザー	悪いボット
1000018	Web ブラウザー	いいボット
1000019	スクレーパー	いいボット
1000020	スクレーパー	いいボット
1000021	スクレーパー	いいボット
1000022	Google_Crawler	いいボット
1000023	Web ブラウザー	悪いボット
1000024	Analyzer	いいボット
1000025	Analyzer	いいボット
1000026	Analyzer	いいボット

ボットシグネチャルール	**ID**	説明
1000027	Analyzer	いいボット
1000028	Analyzer	いいボット
1000029	Web ブラウザー	いいボット
1000030	Analyzer	いいボット
1000031	Analyzer	いいボット
1000032	Web ブラウザー	悪いボット
1000033	Analyzer	いいボット
1000034	Web ブラウザー	悪いボット
1000035	スクレーパー	いいボット
1000036	スクレーパー	いいボット
1000037	Web ブラウザー	いいボット
1000038	Analyzer	いいボット
1000039	Analyzer	いいボット
1000040	Analyzer	いいボット
1000041	Analyzer	いいボット
1000042	Analyzer	いいボット
1000043	Analyzer	いいボット
1000044	Analyzer	いいボット
1000045	Google_App_Engine_Software	いいボット
1000046	Google_Crawler	いいボット
1000047	Web ブラウザー	悪いボット
1000048	Web ブラウザー	悪いボット
1000049	Analyzer	いいボット
1000050	Web ブラウザー	悪いボット
1000051	Web ブラウザー	いいボット
1000052	Web ブラウザー	悪いボット
1000053	スクレーパー	いいボット
1000054	Google_Crawler	悪いボット
1000055	スクレーパー	悪いボット

ボットシグネチャルール	**ID**	説明
1000056	Analyzer	いいボット
1000057	Web ブラウザー	悪いボット
1000058	Web ブラウザー	悪いボット
1000059	Web ブラウザー	悪いボット
1000060	スクレーパー	悪いボット
1000061	アプリケーション	悪いボット
1000062	スクレーパー	悪いボット
1000063	スクレーパー	悪いボット
1000064	スクレーパー	いいボット
1000065	スクレーパー	悪いボット
1000066	スクレーパー	悪いボット
1000067	Web ブラウザー	悪いボット
1000068	スクレーパー	悪いボット
1000069	Web ブラウザー	悪いボット
1000070	スクレーパー	悪いボット
1000071	アプリケーション	悪いボット

2022 年 **8** 月のボット署名の更新

September 27, 2022

新しいシグネチャが追加され、既存のボットシグネチャの一部が更新されます。これらの署名ルールをダウンロードして設定すると、アプライアンスをボット攻撃から保護できます。

ボット署名バージョン

署名バージョン 13 は、13.0 76.31 以降のビルドを搭載した Citrix ADC プラットフォームに適用されます。

新しいボット署名

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	**ID**	説明
4651	マーケティング	いいボット
4652	未分類	悪いボット
4653	検索エンジン	いいボット
4654	ツール	いいボット
4655	昇降補助具	いいボット
4656	マーケティング	いいボット
4657	スクレーパー	いいボット
4658	フィードフェッチャー	いいボット
4659	未分類	悪いボット
4660	ツール	いいボット
4661	ツール	いいボット
4662	未分類	悪いボット
4663	未分類	悪いボット
4664	マーケティング	いいボット
4665	未分類	いいボット
4666	未分類	いいボット
4667	フィードフェッチャー	いいボット
4668	未分類	いいボット
4669	ツール	いいボット
4670	ツール	いいボット
4671	検索エンジン	いいボット
4672	ツール	いいボット
4673	未分類	いいボット
4674	未分類	いいボット
4675	未分類	いいボット
4676	マーケティング	いいボット
4677	スクレーパー	いいボット
4678	マーケティング	いいボット
4679	昇降補助具	悪いボット

ボットシグネチャルール	**ID**	説明
4680	未分類	いいボット
4681	未分類	いいボット
4682	サイトモニター	いいボット
4683	サイトモニター	いいボット
4684	検索エンジン	いいボット
4685	検索エンジン	いいボット
4686	検索エンジン	いいボット
4687	検索エンジン	いいボット
4688	検索エンジン	いいボット
4689	検索エンジン	いいボット
4690	検索エンジン	いいボット
4691	検索エンジン	いいボット
4692	検索エンジン	いいボット
4693	未分類	いいボット
4694	未分類	悪いボット
4695	昇降補助具	いいボット
4696	昇降補助具	いいボット
4697	昇降補助具	いいボット
4698	検索エンジン	いいボット
4699	検索エンジン	いいボット
4700	検索エンジン	いいボット
4701	ツール	悪いボット
4702	未分類	いいボット
4703	ツール	いいボット
4704	ツール	いいボット
4705	昇降補助具	いいボット
4706	サイトモニター	いいボット
4707	検索エンジン	いいボット
4708	ツール	いいボット

ボットシグネチャルール	**ID**	説明
4709	脆弱性スキャナ	いいボット
4710	脆弱性スキャナ	いいボット
4711	昇降補助具	いいボット
4712	昇降補助具	いいボット
4713	昇降補助具	いいボット
4714	スクレーパー	いいボット
4715	ツール	いいボット
4716	ツール	いいボット
4717	検索エンジン	悪いボット
4718	未分類	いいボット
4719	ツール	いいボット
4720	マーケティング	いいボット
4721	マーケティング	いいボット
4722	検索エンジン	いいボット
4723	未分類	悪いボット
4724	ツール	いいボット
4725	検索エンジン	いいボット
4726	検索エンジン	いいボット
4727	ツール	いいボット
4728	未分類	悪いボット
4729	サイトモニター	いいボット
4730	検索エンジン	いいボット
4731	検索エンジン	いいボット
4732	検索エンジン	いいボット
4733	検索エンジン	いいボット
4734	ツール	悪いボット
4735	ツール	悪いボット
4736	ツール	いいボット
4737	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
4738	ツール	いいボット
4739	フィードフェッチャー	いいボット
4740	検索エンジン	いいボット
4741	未分類	悪いボット
4742	検索エンジン	いいボット
4743	昇降補助具	いいボット
4744	ツール	いいボット
4745	ツール	いいボット
4746	マーケティング	いいボット
4747	未分類	悪いボット
4748	検索エンジン	いいボット
4749	検索エンジン	いいボット
4750	検索エンジン	いいボット
4751	検索エンジン	いいボット
4752	検索エンジン	いいボット

ボットの署名を更新しました

以下は、ボットシグニチャルール ID、カテゴリ、およびそのタイプの一覧です。

ボットシグネチャルール	**ID**	説明
3796	スクレーパー	いいボット
3835	検索エンジン	いいボット
3935	スクレーパー	いいボット
4027	検索エンジン	いいボット
4061	検索エンジン	いいボット
4100	検索エンジン	いいボット
4451	スピードテスター	いいボット
4562	検索エンジン	いいボット
4575	マーケティング	いいボット

ボットシグネチャルール	**ID**	説明
4577	マーケティング	いいボット
4578	マーケティング	いいボット
4579	マーケティング	いいボット
4580	マーケティング	いいボット
4583	スクリーンショットクリエーター	いいボット
4584	検索エンジン	いいボット
4585	検索エンジン	いいボット
4597	ツール	いいボット
4599	マーケティング	いいボット
4601	マーケティング	いいボット
4623	昇降補助具	いいボット
4630	未分類	悪いボット
4647	検索エンジン	いいボット
1000000	ブラウザー	いいボット
1000001	アプリケーション	悪いボット
1000002	ブラウザー	いいボット
1000003	スクレーパー	いいボット
1000004	ブラウザー	いいボット
1000005	ブラウザー	悪いボット
1000006	Google Crawler	悪いボット
1000007	スクレーパー	悪いボット
1000008	スクレーパー	いいボット
1000009	ブラウザー	悪いボット
1000010	ボット	悪いボット
1000011	ボット	悪いボット
1000012	スクレーパー	悪いボット
1000013	スクレーパー	悪いボット
1000014	ブラウザー	悪いボット
1000015	ブラウザー	いいボット

ボットシグネチャルール	**ID**	説明
1000016	ブラウザー	悪いボット
1000017	スクレーパー	いいボット
1000018	スクレーパー	悪いボット
1000019	スクレーパー	悪いボット
1000020	スクレーパー	悪いボット
1000021	ブラウザー	いいボット
1000022	スクレーパー	いいボット
1000023	ブラウザー	悪いボット
1000024	ボット	悪いボット
1000025	Analyzer	いいボット
1000026	スクレーパー	いいボット
1000027	ブラウザー	悪いボット
1000028	ブラウザー	悪いボット
1000029	スクレーパー	いいボット
1000030	Google Crawler	いいボット
1000031	ブラウザー	悪いボット
1000032	Analyzer	いいボット
1000033	ボット	悪いボット
1000034	Analyzer	いいボット
1000035	Analyzer	いいボット
1000036	Analyzer	いいボット
1000037	Analyzer	いいボット
1000038	スクレーパー	いいボット
1000039	Analyzer	いいボット
1000040	ブラウザー	悪いボット
1000041	ブラウザー	悪いボット
1000042	スクレーパー	いいボット
1000043	ブラウザー	いいボット
1000044	Analyzer	いいボット

ボットシグネチャルール	**ID**	説明
1000045	Analyzer	いいボット
1000046	Analyzer	いいボット
1000047	Analyzer	いいボット
1000048	Analyzer	いいボット
1000049	ブラウザー	悪いボット
1000050	Google Crawler	いいボット
1000051	ブラウザー	悪いボット
1000052	ブラウザー	悪いボット
1000053	Analyzer	いいボット
1000054	ブラウザー	いいボット
1000055	スクレーパー	いいボット
1000056	ブラウザー	いいボット
1000057	Analyzer	いいボット
1000058	Google Crawler	悪いボット
1000059	スクレーパー	悪いボット
1000060	ブラウザー	悪いボット
1000061	ブラウザー	いいボット
1000062	ブラウザー	悪いボット
1000063	ブラウザー	悪いボット
1000064	ブラウザー	悪いボット
1000065	スクレーパー	悪いボット
1000066	アプリケーション	悪いボット
1000067	スクレーパー	悪いボット
1000068	スクレーパー	悪いボット
1000069	ブラウザー	いいボット
1000070	アプリケーション	悪いボット

キャッシュリダイレクト

October 7, 2021

一般的な配置では、異なるクライアントが Web サーバーに同じコンテンツを繰り返し要求します。オリジン Web サーバーが各リクエストの処理を解放するために、キャッシュリダイレクトを有効にした Citrix ADC アプライアンスは、オリジンサーバーではなくキャッシュサーバーからこのコンテンツを提供できます。

Citrix ADC アプライアンスは、着信要求を分析し、キャッシュ可能なデータに対する要求をキャッシュサーバーに送信し、キャッシュ不可能な要求と動的 HTTP 要求をオリジンサーバーに送信します。

キャッシュのリダイレクトは、ポリシーベースの機能です。デフォルトでは、ポリシーに一致するリクエストはオリジンサーバーに送信され、他のすべてのリクエストはキャッシュサーバーに送信されます。テストまたはメンテナンスのために、ポリシー評価をスキップして、すべての要求をキャッシュまたはオリジンサーバーに送信することができます。

コンテンツスイッチングとキャッシュリダイレクトを組み合わせて、選択的なコンテンツをキャッシュし、特定の種類の要求コンテンツに対して特定のキャッシュサーバーからコンテンツを提供できます。

キャッシュリダイレクト用に構成された Citrix ADC アプライアンスは、ネットワークのエッジ、オリジンサーバーの前、またはネットワークバックボーン上の任意の場所に展開できます。インターネットサービスプロバイダー（ISP）、ケーブル会社、コンテンツ配信ネットワーク、およびエンタープライズネットワークで一般的に使用されるエッジ展開では、Citrix ADC アプライアンスはクライアントの前に直接配置されます。サーバー側の展開では、Citrix ADC アプライアンスは元のサーバーに近づきます。

キャッシュのリダイレクトは HTTP サービスタイプで最も一般的に使用されますが、セキュアな HTTPS プロトコルもサポートされます。

キャッシュリダイレクションポリシー

October 7, 2021

キャッシュリダイレクト仮想サーバは、着信要求ごとにキャッシュリダイレクションポリシーを適用します。デフォルトでは、リクエストが構成されたポリシーのいずれかに一致する場合、リクエストはキャッシュ不可能とみなされ、Citrix ADC アプライアンスはこれをオリジンサーバーに送信します。その他の要求はキャッシュサーバーに送信されます。この動作は元に戻すことができ、設定されたキャッシュリダイレクションポリシーと一致する要求がキャッシュサーバに送信されます。

アプライアンスは、キャッシュのリダイレクトのためのポリシーのセットを提供します。これらの組み込みポリシーが展開に適していない場合は、ユーザー定義のキャッシュリダイレクトポリシーを構成できます。

注意**:** 使用する組み込みキャッシュ・リダイレクト・ポリシーを決定するか、ユーザー定義のポリシーを作成したら、キャッシュ・リダイレクトの構成に進みます。この機能を使用するには、少なくとも 1 つのキャッシュリダイレクシ

ョン仮想サーバーを構成する必要があります。通常の動作では、少なくとも 1 つのキャッシュリダイレクションポリシーをその仮想サーバーにバインドする必要があります。

組み込みのキャッシュリダイレクションポリシー

October 7, 2021

Citrix ADC アプライアンスには、一般的なキャッシュ要求を処理する組み込みのキャッシュリダイレクトポリシーが用意されています。これらのポリシーは、HTTP メソッド、受信リクエストの URL または URL トークン、HTTP バージョン、または HTTP ヘッダーとリクエスト内の値に基づいています。

組み込みのキャッシュリダイレクションポリシーは、仮想サーバに直接バインドでき、それ以上の設定は必要ありません。

キャッシュリダイレクションポリシーでは、クラシック構文とデフォルト構文の 2 種類のアプライアンス表現言語を使用します。これらの言語の詳細については、「ポリシーと式」を参照してください。

組み込みのクラシックキャッシュリダイレクトポリシー

従来の式に基づく組み込みのキャッシュリダイレクトポリシーは、クラシックキャッシュリダイレクトポリシーと呼ばれます。クラシック式とその設定方法の詳細については、ポリシーと式を参照してください。

従来のキャッシュリダイレクションポリシーは、トラフィックおよびその他のデータの基本的な特性を評価します。たとえば、従来のキャッシュリダイレクションポリシーでは、HTTP 要求または応答に特定のタイプのヘッダーまたは URL が含まれているかどうかを判断できます。

Citrix ADC アプライアンスには、次の組み込みのクラシックキャッシュリダイレクトポリシーが用意されています。

組み込みポリシー名	説明
bypass-non-get	リクエストが GET 以外の HTTP メソッドを使用する場合は、キャッシュをバイパスします。
bypass-cache-control	リクエストヘッダーに Cache-Control：no-cache または Cache-Control：no-store ヘッダー、または HTTP リクエストにプラグマヘッダーが含まれている場合は、キャッシュをバイパスします。
bypass-dynamic-url	cgi、asp、exe、cfm、ex、shtml、または htx のいずれかの拡張機能が存在することによって示されるように、URL がコンテンツが動的であることを示唆する場合は、キャッシュをバイパスします。URL が /cgi-bin/、/bin/、/exec/ のいずれかで始まる場合も、キャッシュをバイパスします。

組み込みポリシー名	説明
bypass-urltokens	リクエストが動的であるため、キャッシュをバイパスします。URL の次のいずれかのトークンで示されます。? ,! 、または =。
bypass-cookie	Cookie ヘッダーと拡張子が.png または.jpg 以外のすべての URL のキャッシュをバイパスします。

組み込みの既定の構文キャッシュリダイレクションポリシー

デフォルトの構文式に基づく組み込みのキャッシュリダイレクトポリシーは、デフォルトの構文キャッシュリダイレクトポリシーと呼ばれます。デフォルトの構文式とその構成方法の詳細については、「ポリシーと式」を参照してください。

デフォルトの構文キャッシュリダイレクションポリシーで実行される同じタイプの評価に加えて、デフォルトの構文キャッシュリダイレクションポリシーを使用すると、より多くのデータ（HTTP リクエストの本体など）を分析したり、ポリシールールでより多くの操作を構成したりできます（たとえば、キャッシュまたはオリジンサーバー)。

Citrix ADC アプライアンスには、デフォルトの構文キャッシュリダイレクトポリシーに対して、次の 2 つの組み込みアクションが用意されています。

- CACHE
- ORIGIN

名前によって示されるように、要求はそれぞれキャッシュサーバーまたはオリジンサーバーに送られます。

注意**:** 組み込みのデフォルトの構文キャッシュリダイレクトポリシーを使用している場合は、アクションを変更できません。

Citrix ADC アプライアンスには、次の組み込みの構文キャッシュリダイレクトポリシーが用意されています。

組み込みポリシー名	説明
bypass-non-get_adv	リクエストが GET 以外の HTTP メソッドを使用する場合は、キャッシュをバイパスします。
bypass-cache-control_adv	リクエストヘッダーに Cache-Control：no-cache または Cache-Control：no-store ヘッダー、または HTTP リクエストにプラグマヘッダーが含まれている場合は、キャッシュをバイパスします。

組み込みポリシー名	説明
bypass-dynamic-url_adv	cgi、asp、exe、cfm、ex、shtml、または htx のいずれかの拡張機能が存在することによって示されるように、URL がコンテンツが動的であることを示唆する場合は、キャッシュをバイパスします。URL が /cgi-bin/、/bin/、/exec/ のいずれかで始まる場合も、キャッシュをバイパスします。
bypass-urltokens_adv	リクエストが動的であるため、キャッシュをバイパスします。URL の次のいずれかのトークンで示されます。? ,! 、または =。
bypass-cookie_adv	Cookie ヘッダーと拡張子が.png または.jpg 以外のすべての URL のキャッシュをバイパスします。

組み込みのキャッシュリダイレクトポリシーの表示

コマンドラインインターフェイスまたは構成ユーティリティを使用して、使用可能なキャッシュリダイレクションポリシーを表示できます。

CLI を使用した組み込みのキャッシュリダイレクションポリシーの表示

コマンドプロンプトで入力します。

```
show cr policy [<policyName>]
```

例**:**

```
1 > show cr policy
2 1)       Cache-By-Pass RULE: NS_NON_GET          Policy:bypass-non-get
3 2)       Cache-By-Pass RULE: (NS_CACHECONTROL_NOSTORE ||
     NS_CACHECONTROL_NOCACHE || NS_HEADER_PRAGMA)    Policy:bypass-cache-
     control
4 3)       Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE ||
      NS_EXT_CFM || NS_EXT_EX || NS_EXT_SHTML || NS_EXT_HTX) || (
     NS_URL_PATH_CGIBIN || NS_URL_PATH_EXEC || NS_URL_PATH_BIN)
     Policy:bypass-dynamic-url
5 4)       Cache-By-Pass RULE: NS_URL_TOKENS       Policy:bypass-
     urltokens
6 5)       Cache-By-Pass RULE: (NS_HEADER_COOKIE && NS_EXT_NOT_GIF &&
     NS_EXT_NOT_JPEG)     Policy:bypass-cookie
7  Done
```



```
<!--NeedCopy-->
```

GUI を使用した組み込みのキャッシュリダイレクションポリシーの表示

1. [トラフィック管理] > [キャッシュリダイレクト] > [ポリシー] に移動します。構成済みのキャッシュリダイレクションポリシーが詳細ペインに表示されます。
2. 設定済みのポリシーの 1 つを選択して、詳細を表示します。

キャッシュリダイレクションポリシーを構成する

October 7, 2021

キャッシュリダイレクションポリシーには、1 つ以上の式 (規則とも呼ばれます) が含まれます。各式は、クライアント要求がポリシーと比較されたときに評価される条件を表します。

キャッシュリダイレクションポリシーのアクションは明示的に設定しません。デフォルトでは、Citrix ADC アプライアンスは、ポリシーに一致するすべての要求をキャッシュ不可能と見なし、キャッシュではなくオリジンサーバーに要求を送信します。

クラシックポリシー形式に基づくキャッシュリダイレクションポリシーは、クラシックキャッシュリダイレクションポリシーと呼ばれます。このような各ポリシーには名前があり、論理演算子を使用して組み合わせた古典的な式または古典的な式のセットが含まれます。

従来のキャッシュリダイレクションポリシーの場合は、ポリシーのアクションを明示的に設定しません。デフォルトでは、Citrix ADC アプライアンスは、ポリシーに一致するすべての要求をキャッシュ不可能と見なし、キャッシュではなくオリジンサーバーに要求を送信します。

新しいポリシー形式に基づくキャッシュリダイレクトポリシーは、高度なリダイレクトポリシーと呼ばれます。このようなポリシーには、名前があり、既定の構文式、または論理演算子を使用して組み合わされた一連の既定の構文式、および次の組み込みアクションが含まれます。

- CACHE
- ORIGIN

クラシック式とデフォルトの構文式の詳細については、「ポリシーと式」を参照してください。

CLI を使用したキャッシュリダイレクションポリシーの追加

コマンドプロンプトで次のコマンドを入力して、キャッシュリダイレクションポリシーを追加し、構成を確認します。


```
-  add cr policy <policyName> \*\*-rule\*\* <expression>
-  show cr policy [<policyName>]
<!--NeedCopy-->
```

例:

単純な式を持つポリシー:

```
> add cr policy Policy-CRD-1 -rule ”REQ.HTTP.URL != /\*.jpeg”
 Done
> show cr policy Policy-CRD-1
         Cache-By-Pass RULE: REQ.HTTP.URL != '/\*.jpeg'   Policy:Policy
            -CRD-1
 Done
<!--NeedCopy-->
```

複合式を持つポリシー:

```
> add cr policy Policy-CRD-2 -rule ”REQ.HTTP.METHOD == POST && (REQ.
   HTTP.URL == /\*.cgi || REQ.HTTP.URL != /\*.png)”
 Done
> show cr policy Policy-CRD-2
         Cache-By-Pass RULE: REQ.HTTP.METHOD == POST && (REQ.HTTP.URL
            == '/\*.cgi' || REQ.HTTP.URL != '/\*.png')   Policy:Policy-
            CRD-2
 Done
<!--NeedCopy-->
```

ヘッダーを評価するポリシー:

```
> add cr policy Policy-CRD-3 -rule ”REQ.HTTP.HEADER If-Modified-Since
   EXISTS”
 Done
> show cr policy Policy-CRD-3
         Cache-By-Pass RULE: REQ.HTTP.HEADER If-Modified-Since EXISTS
               Policy:Policy-CRD-3
 Done
<!--NeedCopy-->
```


CLI を使用したデフォルトの構文キャッシュリダイレクションポリシーの追加

コマンドプロンプトで次のコマンドを入力して、キャッシュリダイレクションポリシーを追加し、構成を確認します。

```
- add cr policy <policyName> \*\*-rule\*\* <expression> [-action<
   string>] [-logAction<string>]
- show cr policy [<policyName>]
<!--NeedCopy-->
```

例:

単純な式を持つポリシー:

```
> add cr policy crpol1 -rule !(HTTP.REQ.URL.ENDSWITH(".jpeg" )) -action
    origin
 Done
> show cr policy crpoll
        Policy: crpol1 Rule: !(HTTP.REQ.URL.ENDSWITH(".jpeg")) Action:
            ORIGIN
 Done
<!--NeedCopy-->
```

複合式を持つポリシー:

```
> add cr policy crpol11 -rule "http.req.method.eq(post) && (HTTP.REQ.
   URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))"  -action
   cache
 Done
> show cr policy crpol11
        Policy: crpol11    Rule: http.req.method.eq(post) && (HTTP.REQ
           .URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))
           Action: CACHE
 Done
<!--NeedCopy-->
```

ヘッダーを評価するポリシー:

```
> add cr policy crpol12 -rule http.req.header("If-Modified-Since").
   exists -action origin
 Done
```



```
> show cr policy crpol12
          Policy: crpol12    Rule: http.req.header(”If-Modified-Since”).
             exists   Action: ORIGIN
 Done
<!--NeedCopy-->
```

CLI を使用してキャッシュリダイレクションポリシーを変更または削除する

- キャッシュリダイレクションポリシーを変更するには、set cr policy コマンドを使用します。このコマンドは add cr policy コマンドに似ていますが、既存のポリシーの名前を入力する点が異なります。
- ポリシーを削除するには、<name> 引数だけを受け付ける`rm cr policy` コマンドを使用します。ポリシーが仮想サーバにバインドされている場合は、ポリシーを削除する前に、ポリシーをバインド解除する必要があります。

キャッシュリダイレクトポリシーのバインド解除の詳細については、「キャッシュリダイレクト仮想サーバーからポリシーをバインド解除する」を参照してください。

GUI を使用して単純な式を使用してキャッシュリダイレクションポリシーを構成する

1. [トラフィック管理] > [キャッシュリダイレクト] > [ポリシー] に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [キャッシュリダイレクトポリシーの作成] ダイアログボックスの [名前 *] テキストボックスにポリシーの名前を入力し、[式] 領域で [追加] をクリックします。
4. 単純な式を設定するには、式を入力します。次に、URL 内の.jpeg 拡張子をチェックする式の例を示します。

   - 式タイプ-General
   - フロータイプ-REQ
   - プロトコル-HTTP
   - 修飾子-URL
   - 演算子-! =
   - 値-/.jpeg

   次の例の単純な式は、リクエスト内の If-Modified-Since ヘッダーをチェックします。

   - 式タイプ-General
   - フロータイプ-REQ
   - プロトコル-HTTP
   - Qualifier -HEADER
   - オペレータ-EXISTS
   - ヘッダー名-If-Modified-Since
5. 式の入力が完了したら、[OK] または [作成] をクリックし、[閉じる] をクリックします。

GUI を使用して複合式を使用してキャッシュリダイレクションポリシーを構成する

1. [トラフィック管理] > [キャッシュリダイレクト] > [ポリシー] に移動します。
2. 詳細ペインで、［Add］をクリックします。
3. [Name] テキストボックスに、ポリシーの名前を入力します。

   名前は、文字、数字、またはアンダースコア記号で始まり、1 ～127 の文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、アンダースコア (_) の記号で構成されます。このポリシーが検出するために作成されたコンテンツの種類を他のユーザーにとってわかりやすい名前を選択してください。
4. 作成する複合エクスプレッションのタイプを選択します。選択肢は次のとおりです:
   - 任意の式に一致します。ポリシーでは、1 つ以上の個別の式がトラフィックに一致する場合に、トラフィックが照合されます。
   - すべての式に一致します。ポリシーでは、個々の式がトラフィックと一致する場合にのみ、トラフィックが照合されます。
   - 表形式。[式] リストを 3 つの列で構成された表形式に切り替えます。右端の列に、次のいずれかの演算子を配置します。
     - AND [&& 演算子は] 、ポリシーを一致させるには、リクエストが現在の式と次の式の両方と一致する必要があります。

OR [	] 演算子。ポリシーを一致させるには、要求が現在の式または次の式のいずれか、またはその両方と一致する必要があります。リクエストがどちらの式にも一致しない場合にのみ、ポリシーと一致しません。

     -

     既存の式を選択し、次のいずれかの演算子をクリックして、ネストされたサブグループ内の式をグループ化することもできます。
     - BEGIN SUBGROUP [+（演算子）] 。選択した式でネストされたサブグループを開始するように Citrix ADC アプライアンスに指示します。(式からこの演算子を削除するには、-(.) をクリックします。
     - END SUBGROUP [+）] 演算子。選択した式で現在のネストされたサブグループを終了するように Citrix ADC アプライアンスに指示します。(式からこの演算子を削除するには、[-] をクリックします)。

- 高度なフリーフォーム。エクスプレッションエディタ (Expressions Editor) を完全にオフにし、エクスプレッションリストをテキスト領域に変換して、複合エクスプレッションを入力できます。これは、ポリシー式を作成するための最も強力で難しい方法であり、Citrix ADC クラシック式言語に精通している方のみに推奨されます。

  アドバンスト・フリー・フォーム・テキスト・エリアでのクラシック式の作成の詳細については、「クラシックポリシーと式の設定」を参照してください。

  注意**:** アドバンスフリーフォーム式編集モードに切り替えると、他のモードに戻すことはできません。このエクスプレッション編集モードを使用することが確実でない限り、選択しないでください。

5. [任意の式と一致]、[すべての式と一致]、または [表形式の式] を選択した場合は、[追加] をクリックして [式の追加] ダイアログボックスを表示します。

   キャッシュリダイレクションポリシーでは、式タイプを [General] のままにしておいてください。

6. [Flow Type] ドロップダウンリストで、式のフロータイプを選択します。

   フロータイプによって、ポリシーが着信接続と発信接続のどちらを調べるかが決まります。次の 2 つの選択肢があります。

   - **REQ**. 着信接続または要求を調べるように Citrix ADC アプライアンスを構成します。
   - **RES**. 発信接続または応答を検査するようにアプライアンスを設定します。

7. [Protocol] ドロップダウンリストで、式のプロトコルを選択します。

   プロトコルは、要求または応答でポリシーが検査する情報の種類を決定します。前のドロップダウンリストで [REQ] と [RES] のどちらを選択したかに応じて、次の 4 つすべて、または 3 つだけを選択できます。

   - **HTTP**。HTTP ヘッダーを検査するようにアプライアンスを設定します。
   - **SSL**。SSL クライアント証明書を検査するようにアプライアンスを設定します。前のドロップダウンリストで [REQ (requests)] を選択した場合にのみ使用できます。
   - **TCP**。TCP ヘッダーを検査するようにアプライアンスを設定します。
   - 知的財産権です。送信元または宛先 IP アドレスを検査するようにアプライアンスを設定します。

8. 「修飾子」(Qualifier) ドロップダウンリストから式の修飾子を選択します。

   [Qualifier] ドロップダウンリストの内容は、選択したプロトコルによって異なります。次の表に、各プロトコルで使用可能な選択肢を示します。

   表 1. 各プロトコルで使用できるキャッシュリダイレクションポリシー修飾子

プロトコル	修飾子	定義
HTTP	METHOD	リクエストで使用される HTTP メソッド。
-	URL	URL ヘッダーの内容。

プロトコル	修飾子	定義
-	URLTOKENS	HTTP ヘッダー内の URL トークン。
-	バージョン	接続の HTTP バージョン。
-	HEADER	HTTP 要求のヘッダー部分。
-	URLLEN	URL ヘッダーの内容の長さ。
-	URLQUERY	URL ヘッダーの内容の一部を照会します。
-	URLQUERYLEN	URL ヘッダーのクエリ部分の長さ。
SSL	CLIENT.CERT	SSL クライアント証明書全体。
-	CLIENT.CERT.SUBJECT	クライアント証明書のサブジェクトフィールドの内容。
-	CLIENT.CERT.ISSUER	クライアント証明書の発行者。
-	CLIENT.CERT.SIGALGO	クライアント証明書で使用される署名アルゴリズム。
-	CLIENT.CERT.VERSION	クライアント証明書のバージョン。
-	CLIENT.CERT.VALIDFROM	クライアント証明書が有効な日付。(開始日)
-	CLIENT.CERT.VALIDTO	クライアント証明書が無効になる日付。(終了日)
-	CLIENT.CERT.SERIALNUMBER	クライアント証明書のシリアル番号。
-	CLIENT.CIPHER.TYPE	クライアント証明書で使用される暗号化方式。
-	CLIENT.CIPHER.BITS	暗号化キーの有効ビット数。
-	CLIENT.SSL.VERSION	クライアント証明書の SSL バージョン。
TCP	SOURCEPORT	TCP 接続の送信元ポート。
-	DESTPORT	TCP 接続の宛先ポート。
-	MSS	TCP 接続の最大セグメントサイズ (MSS)。

プロトコル	修飾子	定義
IP	SOURCEIP	接続の送信元 IP アドレス。
-	DESTIP	接続の宛先 IP アドレス。

9. 「演算子」(Operator) ドロップダウンリストから式の演算子を選択します。

   選択内容は、前の手順で選択した修飾子によって異なります。このドロップダウンリストに表示できる演算子の完全なリストは次のとおりです。

   - == . 次のテキスト文字列と完全に一致します。
   - != . 次の文字列と一致しません。
     - . 次の整数より大きい。
   - CONTAINS . 次のテキスト文字列が含まれます。
   - CONTENTS . 指定されたヘッダー、URL、または URL クエリの内容。
   - EXISTS . 指定されたヘッダーまたはクエリが存在します。
   - NOTCONTAINS . 次のテキスト文字列は含まれません。
   - NOTEXISTS . 指定されたヘッダーまたはクエリは存在しません。

   特定のホストに送信されたリクエストに対してこのポリシーを動作させる場合は、デフォルトの「等号 (==)」記号のままにしておきます。

10. 「値」(Value) テキストボックスが表示されている場合は、テキストボックスに適切な文字列または数値を入力します。

    たとえば、このポリシーでホスト shopping.example.com に送信されるリクエストを選択する場合は、[Value] テキストボックスにその文字列を入力します。

11. 修飾子として「HEADER」を選択した場合は、「ヘッダー名」テキスト・ボックスに目的のヘッダーを入力します。
12. [OK] をクリックして、式を [式] リストに追加します。
13. 手順 4 ～11 を繰り返して、追加の式を作成します。
14. [閉じる] をクリックして [式の追加] ダイアログボックスを閉じ、[キャッシュリダイレクトポリシーの作成] ダイアログボックスに戻ります。

キャッシュのリダイレクト構成

October 7, 2021

配置とネットワークトポロジに応じて、次のいずれかのタイプのキャッシュリダイレクトを構成できます。

- 透明。透過キャッシュは、ネットワークバックボーン上のさまざまなポイントに配置して、配信ルートに沿ったトラフィックを緩和できます。透過モードでは、キャッシュリダイレクト仮想サーバーは Citrix ADC アプライアンスに流れるすべてのトラフィックを傍受し、キャッシュリダイレクトポリシーを適用して、コンテンツをキャッシュから配信するか、オリジンサーバーから配信するかを決定します。
- フォワードプロキシ。フォワードプロキシキャッシュサーバは、エンタープライズ LAN のエッジ上に存在し、WAN に面しています。転送プロキシモードでは、キャッシュリダイレクト仮想サーバーは、DNS サーバーを使用して着信要求のホスト名を解決し、キャッシュ不可能なコンテンツに対する要求を解決されたオリジナルサーバーに転送します。キャッシュ可能な要求は、設定されたキャッシュサーバに送信されます。
- リバースプロキシ。リバースプロキシキャッシュは、特定のオリジンサーバー用に構成されます。リバースプロキシ宛ての着信トラフィックは、キャッシュサーバーから提供することも、URL の変更の有無にかかわらず、オリジンサーバーに送信することもできます。

トランスペアレントリダイレクトの構成

October 7, 2021

透過キャッシュリダイレクトを構成すると、Citrix ADC アプライアンスは受信したすべてのトラフィックを評価し、キャッシュ可能かどうかを判断します。このモードは、配信ルートに沿ったトラフィックを軽減し、キャッシュサーバが ISP またはキャリアのバックボーン上にある場合によく使用されます。

デフォルトでは、キャッシュ可能なリクエストはキャッシュサーバーに送信され、キャッシュできないリクエストはオリジンサーバーに送信されます。たとえば、Citrix ADC アプライアンスは、Web サーバーに送信される要求を受信すると、要求内の HTTP ヘッダーと一連のポリシー表現を比較します。要求がポリシーと一致しない場合、アプライアンスは要求をキャッシュサーバーに転送します。要求がポリシーと一致する場合、アプライアンスは要求をそのまま Web サーバーに転送します。

このデフォルトの動作を変更する方法の詳細については、「オリジンではなくキャッシュへの直接ポリシーヒット」を参照してください。

トランスペアレントリダイレクションを設定するには、まずキャッシュリダイレクションとロードバランシングを有効にし、エッジモードを設定します。次に、ワイルドカード IP アドレス（*）を使用してキャッシュリダイレクト仮想サーバーを作成します。これにより、この仮想サーバーは、アプライアンスが所有する任意の IP アドレスでアプライアンスに着信するトラフィックを受信できます。この仮想サーバーに、キャッシュされるべきではない要求の種類を記述するキャッシュリダイレクションポリシーをバインドします。次に、キャッシュ可能な要求のキャッシュリダイレクト仮想サーバーからトラフィックを受信する負荷分散仮想サーバーを作成します。最後に、物理キャッシュサーバーを表すサービスを作成し、負荷分散仮想サーバーにバインドします。

キャッシュのリダイレクトと負荷分散を有効にする

October 7, 2021

アプライアンスキャッシュのリダイレクトおよびロードバランシング機能は、デフォルトでは有効になっていません。キャッシュのリダイレクト設定を有効にするには、これらを有効にする必要があります。

CLI を使用したキャッシュリダイレクトとロードバランシングの有効化

コマンドプロンプトで次のコマンドを入力して、キャッシュリダイレクトと負荷分散を有効にし、設定を確認します。

```
-  enable ns feature cr lb
-  show ns feature
<!--NeedCopy-->
```

例**:**

```
> enable ns feature cr lb
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 4)     Content Switching               CS                   ON
 5)     Cache Redirection               CR                   ON
 6)     Sure Connect

            ...
            ...
            ...

 23)    HTML Injection                  HTMLInjection        ON
 24)    appliance Push                  push                 OF
 Done
<!--NeedCopy-->
```


GUI を使用したキャッシュリダイレクトとロードバランシングの有効化

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. キャッシュリダイレクトを有効にするには、詳細ウィンドウの [モードと機能] で、[詳細機能の構成] をクリックします。
   a) [拡張機能の構成] ダイアログボックスで、[キャッシュリダイレクト] の横にあるチェックボックスをオンにし、[OK] をクリックします。
   b) 機能を有効化/無効化しますか? ダイアログボックスで、[はい] をクリックします。
3. 負荷分散を有効にするには、詳細ウィンドウの [モードと機能] で、[基本機能の構成] をクリックします。
   a) [基本機能の構成] ダイアログボックスで、[負荷分散] の横にあるチェックボックスをオンにし、[OK] をクリックします。
   b) 機能を有効化/無効化しますか? ダイアログボックスで、[はい] をクリックします。

エッジモードの設定

October 7, 2021

ネットワークのエッジに展開すると、Citrix ADC アプライアンスは、そのネットワーク上のサーバーについて動的に学習します。エッジモードでは、アプライアンスは最大 40,000 台の HTTP サーバーとこれらのサーバーのプロキシ TCP 接続を動的に学習できます。

このモードでは、動的に学習されたサービスの統計情報の収集が有効になり、通常、キャッシュリダイレクションの透過的な展開で使用されます。

CLI を使用したエッジモードの有効化

コマンドプロンプトで次のコマンドを入力して、エッジモードを有効にし、設定を確認します。

```
-  enable ns mode Edge
-  show ns mode
<!--NeedCopy-->
```

例:

```
> enable ns mode edge
 Done

> show ns mode

```



```
        Mode                            Acronym             Status
        -------                         -------             ------
          ...
          ...
          ...
 6)     MAC-based forwarding            MBF                 ON
 7)     Edge configuration              Edge                ON
 8)     Use Subnet IP                   USNIP               OFF
          ...
          ...
          ...
 16)    Bridge BPDUs                    BridgeBPDUs         OFF
 Done
<!--NeedCopy-->
```

GUI を使用してエッジモードを有効にする

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ペインの [Modes and Features] で [Configure modes] をクリックします。
3. [モードの構成] ダイアログボックスで、[エッジ構成] の横にあるチェックボックスをオンにし、[OK] をクリックします。
4. 機能を有効化/無効化しますか? ダイアログボックスで、[はい] をクリックします。

キャッシュリダイレクト仮想サーバーの構成

October 7, 2021

デフォルトでは、キャッシュリダイレクト仮想サーバーは、キャッシュ可能な要求を負荷分散仮想サーバーに転送し、キャッシュ不可能な要求を元のサーバーに転送します（ただし、キャッシュ不可能な要求が負荷分散仮想サーバーに送信されるリバースプロキシ構成では除きます）。キャッシュリダイレクト仮想サーバーには、トランスペアレント、フォワードプロキシ、リバースプロキシの 3 種類があります。

トランスペアレントキャッシュリダイレクト仮想サーバは、* の IP アドレスとポート番号（通常は 80）を使用します。このポートは、アプライアンスが表す任意の IP アドレスに送信される HTTP トラフィックを受け入れることができます。その結果、トランスペアレントキャッシュリダイレクト仮想サーバーを 1 つだけ構成できます。構成する追加のキャッシュリダイレクト仮想サーバーは、フォワードプロキシリダイレクトサーバーまたはリバースプロキシリダイレクトサーバーである必要があります。

cli を使用して透過モードでキャッシュリダイレクト仮想サーバーを追加する

コマンドプロンプトで次のコマンドを入力して、キャッシュリダイレクト仮想サーバーを追加し、構成を確認します。


```
1 -  add cr vserver <name> <serviceType> [<IPAddress> <port> ] [-
     cacheType <cacheType>] [-redirect <redirect>]
2 -  show cr vserver [<name>]
3 <!--NeedCopy-->
```

例:

```
1 add cr vserver Vserver-CRD-1 HTTP * 80 -cacheType TRANSPARENT -redirect
      POLICY
2 > show cr vserver Vserver-CRD-1
3         Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
4         State: UP  ARP:DISABLED
5         Client Idle Timeout: 180 sec
6         Down state flush: ENABLED
7         Disable Primary Vserver On Down : DISABLED
8         Default:        Content Precedence: RULE        Cache:
             TRANSPARENT
9         On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
10        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
11  Done
12 <!--NeedCopy-->
```

CLI を使用してキャッシュリダイレクト仮想サーバーを変更または削除する

- 仮想サーバを変更するには、set cr vserver コマンドを使用します。このコマンドは、add cr vserver コマンドと同様ですが、既存の仮想サーバの名前を入力する点が異なります。
- 仮想サーバを削除するには、<name> 引数だけを受け付ける`rm cr vserver` コマンドを使用します。

GUI を使用して透過モードでキャッシュリダイレクト仮想サーバーを追加する

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (キャッシュリダイレクト)] ダイアログボックスで、次のパラメータの値を指定します。
   - 名前 *—name
   - ポート *: port

   * 必須パラメータ

4. [Protocol] ドロップダウンリストで、サポートされているプロトコル (**HTTP** など) を選択します。仮想サーバが、選択したプロトコルの標準ポート以外のポートでトラフィックを受信する場合は、[Port] フィールドに新しい値を入力します。
5. [詳細設定] タブをクリックします。
6. [キャッシュタイプ] が [トランスパレント] に設定され、[リダイレクト] が [ポリシー] に設定されていることを確認します。
7. [Create] をクリックしてから、[Close] をクリックします。[キャッシュリダイレクト仮想サーバー] ペインに、新しい仮想サーバーが表示されます。
8. 新しいキャッシュリダイレクト仮想サーバーを選択して、その構成の詳細を表示します。

キャッシュ・リダイレクト仮想サーバーにポリシーをバインドする

October 7, 2021

キャッシュリダイレクションポリシーは、キャッシュリダイレクション仮想サーバに自動的にはバインドされません。ポリシーベースのキャッシュリダイレクション仮想サーバは、少なくとも 1 つのポリシーをバインドしない限り機能しません。

CLI を使用してポリシーをキャッシュリダイレクト仮想サーバーにバインドする

コマンドプロンプトで入力します。

```
1 -  bind cr vserver <name> -policyName <string>
2 -  show cr vserver [<name>]
3 <!--NeedCopy-->
```

例**:**

```
1 > bind cr vserver Vserver-CRD-1 -policyName bypass-cache-control
2  Done
3 > bind cr vserver Vserver-CRD-1 -policyName bypass-dynamic-url
4  Done
5 > bind cr vserver Vserver-CRD-1 -policyName bypass-urltokens
6  Done
7 > bind cr vserver Vserver-CRD-1 -policyName bypass-cookie
8  Done
9
```



```
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache:
           TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: bypass-dynamic-url
3)      Cache bypass  Policy: bypass-urltokens
4)      Cache bypass  Policy: bypass-cookie
 Done
<!--NeedCopy-->
```

GUI を使用してユーザー定義のポリシーをキャッシュリダイレクト仮想サーバーにバインドする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 構成する仮想サーバをクリックし、[開く] をクリックします。
3. [ポリシー] タブで、ポリシーの種類を選択し、[ポリシーの挿入] をクリックします。
4. [Policy Name] 列で、バインドするポリシーを選択します。
5. [OK] をクリックします。

キャッシュリダイレクト仮想サーバーからポリシーをバインド解除する

October 7, 2021

キャッシュリダイレクト仮想サーバーからポリシーをバインド解除すると、Citrix ADC アプライアンスはクライアント要求の評価時にポリシーを適用しなくなります。

コマンド **CLI** を使用して、キャッシュリダイレクト仮想サーバーからポリシーをバインド解除します

コマンドプロンプトで入力します。

```
-  unbind cr vserver <name> -policyName <string>
-  show cr vserver [<name>]
```



```
<!--NeedCopy-->
```

例:

```
unbind cr vserver Vserver-CR-1 -policyName bypass-non-get
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache:
           TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
 Done
<!--NeedCopy-->
```

GUI を使用してキャッシュリダイレクト仮想サーバーからユーザー定義ポリシーをバインド解除する

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 構成する仮想サーバーをクリックし、[開く] をクリックします。
3. [ポリシー] タブの [ポリシー名] で、バインドを解除するポリシーを選択します。
4. [ポリシーのバインド解除] をクリックし、[OK] をクリックします。

負荷分散仮想サーバーの作成

October 7, 2021

Citrix ADC アプライアンス上のキャッシュリダイレクト仮想サーバーは、要求がキャッシュ可能な場合はキャッシュサーバーファームに、要求がキャッシュ可能でない場合はオリジンサーバーファームに要求を送信できます。

各キャッシュ・サーバーは、サービスによってアプライアンス上で表されます。サービスとは、キャッシュ・リダイレクト仮想サーバーから要求を受け取り、その要求をサーバーに転送する負荷分散仮想サーバーにバインドされます。

負荷分散仮想サーバーおよびその他の構成オプションの設定の詳細については、負荷分散を参照してください。

CLI を使用した負荷分散仮想サーバーの作成

コマンドプロンプトで次のコマンドを入力して、負荷分散仮想サーバーを作成し、構成を確認します。

```
-  add lb vserver <name> <serviceType> [<IPAddress>] [<port>]
-  show lb vserver [<name>]
<!--NeedCopy-->
```

例:

```
> add lb vserver Vserver-LB-CR HTTP 10.102.20.30 80
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:00:08.470
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
 Done
<!--NeedCopy-->
```

GUI を使用した負荷分散仮想サーバーの作成

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次に示すパラメータの値を指定します。
   - 名前 *-name
   - IP アドレス *-IPAddress

- ポート *-port

* 必須パラメータ

4. [プロトコル] ボックスの一覧で、サポートされているプロトコル (**HTTP** など) を選択します。仮想サーバが、選択したプロトコルの既知のポート以外のポートでトラフィックを受信する場合は、[Port] フィールドに新しい値を入力します。
5. [Create] をクリックしてから、[Close] をクリックします。[負荷分散仮想サーバー] ペインに、新しい仮想サーバーが表示されます。

HTTP サービスを構成する

October 7, 2021

Citrix ADC アプライアンスでは、サービスはネットワーク上の物理サーバーを表します。透過キャッシュリダイレクション設定では、サービスはキャッシュサーバを表します。キャッシュ可能な要求は、キャッシュ・リダイレクト仮想サーバから負荷分散仮想サーバに送信され、各要求は正しいサービスに転送され、キャッシュサーバに渡されます。

CLI を使用した **HTTP** サービスの設定

コマンドプロンプトで、次のコマンドを入力して HTTP サービスを作成し、構成を確認します。

```
-  add service <name> <IP> <serviceType> <port> -cacheType <cacheType>
-  show service [<name>]
<!--NeedCopy-->
```

例:

```
> add service Service-HTTP-1 10.102.29.40 HTTP 80 -cacheType
   TRANSPARENT
 Done
> show service Service-HTTP-1
        Service-HTTP-1 (10.102.29.40:80) - HTTP
        State: DOWN
        Last state change was at Fri Jul  2 09:14:17 2010
        Time since last state change: 0 days, 00:00:13.820
        Server Name: 10.102.29.40
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
```



```
12          Client Keepalive(CKA): NO
13          Access Down Service: NO
14          TCP Buffering(TCPB): NO
15          HTTP Compression(CMP): YES
16          Idle timeout: Client: 180 sec   Server: 360 sec
17          Client IP: DISABLED
18          Cache Type: TRANSPARENT Redirect Mode:
19          Cacheable: NO
20          SC: OFF
21          SP: ON
22          Down state flush: ENABLED
23
24  1)      Monitor Name: tcp-default
25                  State: DOWN     Weight: 1
26                  Probes: 3       Failed [Total: 3 Current: 3]
27                  Last response: Failure - Time out during TCP connection
                        establishment stage
28                  Response Time: N/A
29   Done
30  <!--NeedCopy-->
```

CLI を使用したサービスの変更または削除

- サービスを変更するには、set service コマンドを使用します。このコマンドは、add service コマンドと同様ですが、既存のサービスの名前を入力する点が異なります。
- サービスを削除するには、<name> 引数だけを受け付ける`rm service` コマンドを使用します。

GUI を使用して HTTP サービスを追加します

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [サービスの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - サービス名 *—name
   - サーバ *: IP
   - ポート *: port

   * 必須パラメータ
4. [Protocol*] ドロップダウンリストで、サポートされているプロトコル (**HTTP** など) を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。

負荷分散仮想サーバーに対するサービスのバインド/バインド解除

October 7, 2021

負荷分散仮想サーバーにサービスをバインドする必要があります。これにより、ロードバランサーは、サービスが表すサーバーに要求を転送できます。構成が変更された場合は、負荷分散仮想サーバーからサービスをバインド解除できます。

CLI を使用してサービスを負荷分散仮想サーバーにバインドする

コマンドプロンプトで入力します。

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

例**:**

```
> bind lb vserver vserver-LB-CR service-HTTP-1
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:42:25.610
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
```



```
<!--NeedCopy-->
```

CLI を使用して負荷分散仮想サーバーからサービスをバインド解除する

サービスのバインドを解除するには、`bind lb vserver`の代わりに`unbind lb vserver`コマンドを使用します。

GUI を使用した負荷分散仮想サーバーからのサービスのバインド/バインド解除

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します
2. 詳細ウィンドウで、サービスをバインドまたはバインド解除する仮想サーバーを選択し、[開く] をクリックします。
3. [サービス] タブの [アクティブ] 列で、[サービス名] の横にあるチェックボックスをオンまたはオフにします。
4. [OK] をクリックします。

透過キャッシュにプロキシポート設定を使用しない

October 7, 2021

Citrix ADC アプライアンスで構成されたキャッシュサービスで「ソース IP（USIP）の使用」オプションが無効になっている場合、アプライアンスはアプライアンスが所有するサブネット IP（SNIP）アドレスまたはマップ IP（MIP）アドレスをソース IP アドレスとして、ランダムポートをソースポートとして使用して、クライアント要求をキャッシュサービスに転送します。ランダムに選択されたポートは、プロキシポートと呼ばれます。

ただし、完全透過キャッシュ (キャッシュサービスがクライアントの IP アドレスとポート番号を受け取るキャッシュ構成) を構成する場合は、USIP オプションをグローバルまたはキャッシュサービスで有効にするだけでなく、グローバルまたはキャッシュサービスを有効にします。Use Proxy Port 設定を無効にすると、アプライアンスはキャッシュサービスに接続するときにクライアントのソースポートをソースポートとして使用でき、完全に透過的なキャッシュ構成が保証されます。

グローバルまたはサービスで [プロキシポートを使用] オプションの設定の詳細については、サーバー側接続の送信元ポートの構成を参照してください。

Citrix ADC アプライアンスへのポート範囲の割り当て

October 7, 2021

クライアント IP アドレスを共有すると競合が発生し、ルーター、キャッシュサーバー、オリジンサーバー、その他の Citrix ADC アプライアンスなどのネットワークデバイスが、アプライアンス、つまり応答の送信先クライアントを特定できなくなる場合があります。

この問題を解決するには、Citrix ADC アプライアンスにソースポート範囲を割り当てます。この割り当てにより、ネットワークデバイスは、要求を送信した Citrix ADC アプライアンスを明確に識別できます。

CLI を使用して送信元ポート範囲を **Citrix ADC** アプライアンスに割り当てる

コマンドプロンプトで入力します。

```
set ns param -crPortRange <startPortNumber-endPortNumber>
```

アプライアンス **GUI** を使用してソースポート範囲を **Citrix ADC** アプライアンスに割り当てる

1. ナビゲーションウィンドウで、[システム] をクリックし、[設定] をクリックします。
2. [設定] グループで、[グローバルシステム設定の変更] リンクをクリックします。
3. [キャッシュリダイレクトポート範囲] グループで、[開始ポート] にポート番号、[終了ポート] にポート番号を入力して、アプライアンスのポート範囲を指定します。
4. [OK] をクリックします。

仮想サーバーの負荷分散を有効にして要求をキャッシュにリダイレクトする

October 7, 2021

負荷分散仮想サーバーが特定の IP アドレスとポートの組み合わせをリッスンするように構成されている場合、そのアドレスとポートの組み合わせを宛先とする要求については、キャッシュリダイレクト仮想サーバーよりも優先されます。したがって、キャッシュのリダイレクト仮想サーバーはこれらの要求を処理しません。

この機能を無効にし、キャッシュのリダイレクト仮想サーバーに要求をキャッシュから処理するかどうかを決定させる場合は、特定の負荷分散仮想サーバーをキャッシュ可能に構成します。

このような構成は通常、ISP がネットワークのエッジで Citrix ADC アプライアンスを使用し、すべてのトラフィックがアプライアンスを通過する場合に使用されます。

CLI を使用して要求をキャッシュにリダイレクトする負荷分散仮想サーバーを有効にする

コマンドプロンプトで入力します。

```
1 -  set lb vserver <name> [-cacheable ( YES | NO)]
2 -  show lb vserver [<name>]
```



```
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-CR  - cacheable YES
> show lb vserver vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 01:05:51.510
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Cacheable: YES  PQ: OFF SC: OFF
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

トランスペアレントキャッシュリダイレクトの場合、アプライアンスはすべてのトラフィックを代行受信し、すべての要求を評価してキャッシュ可能かどうかを判断します。キャッシュ不可能な要求は、変更されずにオリジンサーバーに送信されます。

トランスペアレントキャッシュリダイレクトを使用する場合、常にトラフィックをオリジンサーバーに送信するロードバランシング仮想サーバーのキャッシュリダイレクトを無効にすることができます。

CLI を使用して負荷分散仮想サーバーのキャッシュを無効にする

ロードバランシング仮想のキャッシュをオフにするには、set lb vserver の代わりに unset lb vserver コマンドを使用します。キャッシュ可能なパラメータに NO 値を指定します。

GUI を使用して要求をキャッシュにリダイレクトする負荷分散仮想サーバーを有効または無効にする

1. ［Traffic Management］ > ［Load Balancing］ > ［Virtual Servers］ の順に選択します。
2. 詳細ウィンドウで、キャッシュを有効または無効にする仮想サーバーを選択し、[開く] をクリックします。
3. [詳細設定] タブで、[キャッシュリダイレクト] チェックボックスをオンまたはオフにします。
4. ［OK］ をクリックします。

フォワードプロキシリダイレクトの構成

October 7, 2021

フォワードプロキシは、クライアントまたはクライアントグループの単一の連絡先です。この構成では、Citrix ADC アプライアンスはキャッシュ不可能な要求をオリジナル・サーバーにリダイレクトし、キャッシュ可能な要求をフォワード・プロキシ・キャッシュまたは透過キャッシュにリダイレクトします。

アプライアンスがフォワード・プロキシとして構成されている場合、ユーザーは、ブラウザが宛先サーバーではなくフォワード・プロキシに要求を送信するようにブラウザを変更する必要があります。

アプライアンス上の転送プロキシキャッシュリダイレクト仮想サーバーは、要求とキャッシュのポリシーを比較します。要求がキャッシュ可能でない場合、アプライアンスは DNS ロード・バランシング仮想サーバーに宛先の解決を照会し、その要求をオリジナル・サーバーに送信します。要求がキャッシュ可能な場合、アプライアンスはキャッシュ用の負荷分散仮想サーバーに要求を転送します。

アプライアンスは、リクエストの HOST ヘッダー内のホストドメイン名または IP アドレスを使用して、リクエストされた宛先を決定します。要求に HOST ヘッダーがない場合、アプライアンスは要求内の宛先 IP アドレスに基づいて HOST ヘッダーを挿入します。

通常、Citrix ADC アプライアンスは、企業 LAN のフォワードプロキシとして機能します。このような構成では、アプライアンスはエンタープライズ LAN のエッジにあり、クライアント要求が WAN にファンアウトされる前に傍受します。アプライアンスをフォワードプロキシモードに設定すると、WAN 上のトラフィックが減少します。

フォワードプロキシキャッシュリダイレクトを構成するには、まずアプライアンスのロードバランシングとキャッシュリダイレクトを有効にします。次に、DNS 負荷分散仮想サーバーおよび関連するサービスを構成します。また、負荷分散仮想サーバーを構成し、キャッシュに適切なサービスをバインドします。フォワードプロキシキャッシュリダイレクト仮想サーバーを構成し、DNS およびロードバランシング仮想サーバーをバインドします。また、キャッシュポリシーを構成し、キャッシュリダイレクト仮想サーバーにバインドする必要があります。セットアップを完了するには、フォワードプロキシを使用するようにクライアントブラウザを設定します。

アプライアンスでキャッシュリダイレクトと負荷分散を有効にする方法の詳細については、「キャッシュのリダイレクトとロードバランシングの有効化」を参照してください。

負荷分散仮想サーバーの作成方法の詳細については、「負荷分散仮想サーバーを作成する」を参照してください。

キャッシュサーバーを表すサービスを構成する方法の詳細については、「HTTP サービスの構成」を参照してください。

サービスを仮想サーバーにバインドする方法の詳細については、「負荷分散仮想サーバーへのサービスのバインド/バインド解除」を参照してください。

フォワードプロキシキャッシュリダイレクトサーバーの作成方法の詳細については、「キャッシュリダイレクト仮想サーバーの構成」および「TRANSTARENT」または「FORWARD」タイプの仮想サーバーの作成を参照してください。

キャッシュリダイレクトポリシーをキャッシュリダイレクト仮想サーバーにバインドする方法については、「キャッシュリダイレクトポリシーを構成する」を参照してください。

DNS サービスを作成する

October 7, 2021

DNS サービスは、Citrix ADC アプライアンス上で、ネットワーク内の物理 DNS サーバーの表現です。DNS 負荷分散仮想サーバーは、そのようなサービスを通じて DNS 要求をネットワーク内の DNS サーバーに送信します。

CLI を使用した DNS サービスの作成

コマンドラインで次のコマンドを入力して、DNS サービスを作成し、構成を確認します。

```
-  add service <name> <IP> <serviceType> <port>
-  show service [<name>]
<!--NeedCopy-->
```

例:

```
add service Service-DNS-1 10.102.29.41 DNS 53
show service Service-DNS-1
        Service-DNS-1 (10.102.29.41:53) - DNS
        State: DOWN
        Last state change was at Fri Jul  2 10:14:32 2010
        Time since last state change: 0 days, 00:00:13.550
        Server Name: 10.102.29.41
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
```



```
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping-default
                State: DOWN     Weight: 1
                Probes: 3       Failed [Total: 3 Current: 3]
                Last response: Failure - Probe timed out.
                Response Time: 2000.0 millisec
 Done
<!--NeedCopy-->
```

GUI を使用して DNS サービスを追加する

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [サービスの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - サービス名 *—name
   - サーバ *: IP
   - ポート *: port

* 必須パラメータ

1. [Protocol*] ドロップダウンリストで、サポートされているプロトコル (**DNS** など) を選択します。
2. [Create] をクリックしてから、[Close] をクリックします。

DNS 負荷分散仮想サーバーを作成する

October 7, 2021

DNS 仮想サーバーは、クライアント要求をオリジンサーバーに転送する前に、フォワードプロキシが DNS 解決を実行できるようにします。DNS 負荷分散仮想サーバーは、ネットワーク上の物理 DNS サーバーを表す DNS サービスに関連付けられます。

CLI を使用して DNS 負荷分散仮想サーバーを作成する

コマンドラインで次のコマンドを入力して、DNS 負荷分散仮想サーバーを作成し、構成を確認します。


```
-  add lb vserver <name> <serviceType>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

例:

```
> add lb vserver Vserver-DNS-1 DNS
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:00:08.10
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
 Done
<!--NeedCopy-->
```

GUI を使用して **DNS** 負荷分散仮想サーバーを作成する

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスの [名前] ボックスに、仮想サーバーの名前を入力します。
4. [Protocol*] ドロップダウンリストで、サポートされているプロトコル (**DNS** など) を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。[DNS 仮想サーバー] ペインに新しい仮想サーバーが表示されます。

DNS サービスを仮想サーバーにバインドする

October 7, 2021

DNS サーバーが DNS 要求に応答するには、DNS サーバーを表すサービスが DNS 仮想サーバーにバインドされている必要があります。

CLI を使用して DNS サービスを負荷分散仮想サーバーにバインドする

コマンドプロンプトで次のコマンドを入力して、DNS サービスを負荷分散仮想サーバーにバインドし、構成を確認します。

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver <name>
<!--NeedCopy-->
```

例:

```
> bind lb vserver Vserver-DNS-1 Service-DNS-1
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:12:16.80
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE

1) Service-DNS-1 (10.102.29.41: 53) - DNS State: DOWN   Weight: 1
 Done
>
<!--NeedCopy-->
```

CLI を使用して負荷分散仮想サーバーから DNS サービスをバインド解除する

`bind lb vserver`の代わりに`unbind lb vserver`コマンドを使用します。

GUI を使用した負荷分散仮想サーバーのバインド/バインド解除

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します
2. 詳細ウィンドウで、DNS サービスをバインドまたはバインド解除する仮想サーバーを選択し、[開く] をクリックします。
3. [サービス] タブの [アクティブ] 列で、[サービス名] の横にあるチェックボックスをオンまたはオフにします。
4. [OK] をクリックします。

フォワードプロキシを使用するようにクライアント **Web** ブラウザを構成する

October 7, 2021

Citrix ADC アプライアンスをネットワークでフォワードプロキシキャッシュリダイレクト仮想サーバーとして構成する場合は、クライアント Web ブラウザーを構成して、フォワードプロキシに要求を送信する必要があります。通常、転送プロキシを使用する場合、ネットワーク内のサーバーへの唯一のルートは、転送プロキシ経由です。

フォワードプロキシを使用するようにブラウザを設定するには、ブラウザのマニュアルを参照してください。この構成のフォワードプロキシキャッシュリダイレクト仮想サーバーの IP アドレスとポート番号を指定します。

リバースプロキシリダイレクトを構成する

October 7, 2021

リバースプロキシは、1 つ以上の Web サーバーの前に存在し、オリジンサーバーをクライアント要求から保護します。多くの場合、リバースプロキシキャッシュは、サーバーへのすべてのクライアント要求のフロントエンドです。管理者は、リバース・プロキシ・キャッシュを特定のオリジナル・サーバーに割り当てます。リバースプロキシキャッシュは、任意のオリジンサーバーへのすべてのリクエストに対して頻繁にリクエストされたコンテンツをキャッシュする透過プロキシキャッシュとフォワードプロキシキャッシュとは異なり、サーバーの選択は要求に基づいて行われます。

トランスペアレントプロキシキャッシュとは異なり、リバースプロキシキャッシュは独自の IP アドレスを持っており、キャッシュ不可能な要求内の宛先ドメインと URL を新しい宛先ドメインおよび URL に置き換えることができます。

リバースプロキシキャッシュリダイレクトは、オリジンサーバー側またはネットワークのエッジに展開できます。オリジンサーバーにデプロイされると、リバースプロキシキャッシュのリダイレクト仮想サーバーは、オリジンサーバーへのすべてのリクエストのフロントエンドになります。

リバース・プロキシ・モードでは、アプライアンスが要求を受信すると、キャッシュ・リダイレクト仮想サーバーが要求を評価し、キャッシュ用の負荷分散仮想サーバーまたはオリジンの負荷分散仮想サーバーのいずれかに転送しま

す。受信リクエストは、バックエンドサーバーに送信される前にホストヘッダーまたはホスト URL を変更することで変換できます。

リバースプロキシキャッシュリダイレクトを構成するには、最初にキャッシュリダイレクトとロードバランシングを有効にします。次に、キャッシュ可能な要求をキャッシュサーバーに送信するように、負荷分散仮想サーバーとサービスを構成します。また、オリジナル・サーバーの負荷分散仮想サーバーおよび関連サービスを構成します。次に、リバースプロキシキャッシュリダイレクト仮想サーバーを構成し、関連するキャッシュリダイレクトポリシーをバインドします。最後に、マッピングポリシーを構成し、リバースプロキシキャッシュリダイレクト仮想サーバーにバインドします。

マッピングポリシーには、キャッシュのリダイレクト仮想サーバーが、キャッシュ不可能な要求をオリジンのロードバランシング仮想サーバーに転送できるようにするアクションが関連付けられています。

必ずデフォルトのキャッシュサーバーの宛先を作成してください。

アプライアンスでキャッシュリダイレクトと負荷分散を有効にする方法の詳細については、「キャッシュのリダイレクトとロードバランシングの有効化」を参照してください。

負荷分散仮想サーバーの作成方法の詳細については、「負荷分散仮想サーバーを作成する」を参照してください。

キャッシュサーバーを表すサービスを構成する方法の詳細については、「HTTP サービスの構成」を参照してください。

サービスを仮想サーバーにバインドする方法の詳細については、「負荷分散仮想サーバーへのサービスのバインド/バインド解除」を参照してください。

リバースプロキシキャッシュリダイレクトサーバーの作成方法の詳細については、「キャッシュリダイレクト仮想サーバーの構成」および「REVERSE」タイプの仮想サーバーの作成を参照してください。

組み込みのキャッシュリダイレクトポリシーをキャッシュリダイレクト仮想サーバーにバインドする方法については、キャッシュリダイレクト仮想サーバーへのポリシーのバインドを参照してください。

マッピングポリシーの構成

着信要求がキャッシュ不可能な場合、リバースプロキシキャッシュリダイレクト仮想サーバーは、要求内のドメインと URL をターゲットオリジンサーバーのドメインと URL に置き換え、リクエストをオリジン用のロードバランシング仮想サーバーに転送します。

マッピングポリシーにより、リバースプロキシキャッシュリダイレクト仮想サーバーは、宛先ドメインと URL を置き換え、要求をオリジンの負荷分散仮想サーバーに転送できます。

マッピングポリシーでは、最初にドメインと URL を変換し、その要求を元のロードバランシング仮想サーバに渡す必要があります。

マッピングポリシーは、次のようにドメイン、URL プレフィックス、および URL サフィックスをマッピングできます。

- ドメインマッピング: プレフィックスまたはサフィックスなしでドメインをマッピングできます。ドメインマッピングは、仮想サーバのデフォルトのマッピングです（たとえば、www.mycompany.com を www.myrealcompany.com にマッピングします）。
- プレフィックスマッピング: URL の一部としてプレフィックスが付く指定されたパターンを置き換えることができます（たとえば、www.mycompany.com/sports/index.html を www.mycompany.com/news/index.html にマッピングします）。
- サフィックスマッピング: URL のファイルサフィックスを置き換えることができます（たとえば、www.mycompany.com/スポーツ/インデックス.html を www.mycompany.com/スポーツ/インデックス.asp にマッピングします）。

マッピングされるソース文字列とデスティネーション文字列は類似している必要があります。送信元ドメインを指定する場合は、宛先ドメインを指定する必要があります。送信元サフィックスを指定する場合は、宛先サフィックスを指定する必要があります。同様に、ソースから完全な URL を指定する場合、ターゲット URL も完全な URL である必要があります。

リバースプロキシモードのマッピングポリシーを構成したら、キャッシュリダイレクト仮想サーバーにバインドする必要があります。

ソース URL、ターゲット URL、ソースドメインとターゲットドメインの組み合わせを使用して、3 種類のドメインマッピングをすべて構成できます。

CLI を使用してリバースプロキシモードのマッピングポリシーを構成する

コマンドプロンプトで次のコマンドを入力して、ポリシーマップを追加し、設定を確認します。

```
1 -  add policy map <mapPolicyName> -sd <string> [-su <string>] [-td <
     string>] [-tu <string>]
2 -  show policy map [<mapPolicyName>]
3 <!--NeedCopy-->
```

例**:**

次のコマンドは、クライアント要求内のドメインをターゲットドメインにマッピングします。

```
1 > add policy map myMappingPolicy -sd www.mycompany.com -td www.
     myrealcompany.com
2  Done
3 > show policy map myMappingPolicy
4 1)      Name: myMappingPolicy
5         Source Domain: www.mycompany.com        Source Url:
6         Target Domain: www.myrealcompany.com    Target Url:
7  Done
```



```
<!--NeedCopy-->
```

次に、URL サフィックスを別の URL サフィックスにマッピングする例を示します。

```
> add policy map myOtherMappingPolicy -sd www.mycompany.com -td www.
    myrealcompany.com -su /news.html -tu /realnews.html
 Done
> show policy map myOtherMappingPolicy
1)      Name: myOtherMappingPolicy
        Source Domain: www.mycompany.com        Source Url: /news.html
        Target Domain: www.myrealcompany.com    Target Url: /realnews.
            html
 Done
<!--NeedCopy-->
```

GUI を使用してリバースプロキシモードのマッピングポリシーを構成する

1. [トラフィック管理] **>** [キャッシュリダイレクト] **>** [マップポリシー] に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [マップポリシーの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - 名前 *-mapPolicyName
   - ソース・ドメイン *-sd
   - ターゲットドメイン *-td
   - ソース URL su
   - ターゲット URL-tu

   * 必須パラメータ
4. [Create] をクリックしてから、[Close] をクリックします。[Map] ペインに新しいマッピングポリシーが表示されます。

CLI を使用してマッピングポリシーをキャッシュリダイレクト仮想サーバーにバインドする

コマンドプロンプトで次のコマンドを入力して、マッピングポリシーをキャッシュリダイレクト仮想サーバーにバインドし、構成を確認します。

```
-  bind cr vserver <name> -policyName <string> [<targetVserver>]
-  show cr vserver <name>
<!--NeedCopy-->
```


例:

```
> bind cr vserver Vserver-CRD-3 -policyName myMappingPolicy Vserver-LB-CR
 Done
> show cr vserver Vserver-CRD-3
        Vserver-CRD-3 (10.102.29.50:88) - HTTP  Type: CONTENT
        State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default: Vserver-LB-CR  Content Precedence: RULE        Cache: REVERSE
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Policy:         Target: Vserver-LB-CR   Priority: 0     Hits: 0
1)      Map: myMappingPolicy Target: Vserver-LB-CR
 Done
<!--NeedCopy-->
```

GUI を使用してマッピング・ポリシーをキャッシュ・リダイレクト仮想サーバーにバインドする

1. [トラフィック管理] **>** [キャッシュリダイレクト] **>** [仮想サーバー] に移動します。
2. 詳細ペインで、マッピングポリシーをバインドする仮想サーバーを選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (キャッシュリダイレクト)] の [ポリシー] タブで、[マップ] を選択し、[ポリシーの挿入] をクリックします。
4. [ポリシー名（Policy Name）] 列で、ドロップダウンリストからポリシーを選択します。
5. [ターゲット] 列で下矢印をクリックし、ドロップダウンリストから仮想サーバーを選択します。
6. [**OK**] をクリックします。

選択的キャッシュリダイレクト

October 7, 2021

選択型キャッシュリダイレクトは、特定の種類のコンテンツ（画像など）に対する要求を 1 つのキャッシュサーバまたはキャッシュサーバのグループに送信し、他の種類のコンテンツを別のキャッシュサーバまたはキャッシュサーバのグループに送信します。高度なキャッシュリダイレクトは、トランスペアレント、リバースプロキシ、またはフォワードプロキシモードで構成できます。

選択的キャッシュリダイレクトでは、Citrix ADC アプライアンスはクライアント要求を傍受し、キャッシュ不可能な要求をクライアント要求の元の宛先に転送します。キャッシュ可能な要求の場合、アプライアンスは特定のコンテンツタイプのコンテンツを提供できる宛先キャッシュサーバーに要求を送信します。

選択的キャッシュリダイレクトには、キャッシュリダイレクトポリシーに加えてコンテンツスイッチングポリシーを構成することが含まれます。アプライアンスは、まず、キャッシュリダイレクション仮想サーバーにバインドされているキャッシュリダイレクションポリシーを評価します。要求がキャッシュリダイレクションポリシーと一致する場合、キャッシュリダイレクション仮想サーバーは、要求をオリジンサーバーまたはオリジンのロードバランシング仮想サーバーに送信します。要求に一致するキャッシュリダイレクトポリシーがない場合、アプライアンスはキャッシュリダイレクション仮想サーバーにバインドされたコンテンツスイッチングポリシーを評価します。コンテンツスイッチングポリシーが要求と一致する場合、キャッシュリダイレクト仮想サーバーは、要求をキャッシュの負荷分散仮想サーバーにリダイレクトします。

選択的キャッシュリダイレクトを構成するには、まず Citrix ADC アプライアンスでキャッシュリダイレクト、負荷分散、コンテンツスイッチングを有効にします。次に、キャッシュと関連する HTTP サービスの負荷分散仮想サーバーを構成します。その後、キャッシュリダイレクト仮想サーバーを構成し、キャッシュリダイレクトとコンテンツスイッチングポリシーの両方をバインドします。ポリシーをバインドしたら、ルールベースまたは URL ベースのコンテンツスイッチングポリシーを優先するように仮想サーバを設定できます。

Edge デプロイトポロジでトランスペアレントモードキャッシュリダイレクト用に構成されている場合、アプライアンスはキャッシュ可能なすべての HTTP トラフィックをトランスペアレントキャッシュファームに送信します。クライアントは、アプライアンスを介してインターネットにアクセスします。アプライアンスは、ポート 80 でトラフィックを受信するレイヤ 4 スイッチとして構成されています。

アプライアンスは、イメージのリクエスト（.png ファイルや.jpg ファイルなど）を透過キャッシュファーム内の 1 つのサーバーに送信し、その他の静的コンテンツのリクエストはすべてファーム内の他のサーバーに送信できます。この構成では、イメージキャッシュにイメージを送信し、キャッシュ可能な他のすべてのコンテンツをデフォルトのキャッシュに送信するように、コンテンツスイッチングポリシーを構成します。

注：ここで説明する設定は、透過的な選択キャッシュのリダイレクト用です。したがって、リバースプロキシ構成と同様に、オリジンに負荷分散仮想サーバーは必要ありません。

このタイプの選択キャッシュリダイレクトを構成するには、最初にキャッシュリダイレクト、ロードバランシング、およびコンテンツスイッチングを有効にします。次に、キャッシュの負荷分散仮想サーバーを構成し、関連付けられた HTTP サービスを構成します。次に、キャッシュリダイレクト仮想サーバーを構成し、キャッシュリダイレクトとコンテンツスイッチングポリシーの両方を作成し、この仮想サーバーにバインドします。

アプライアンスでキャッシュリダイレクトと負荷分散を有効にする方法の詳細については、「キャッシュのリダイレクトとロードバランシングの有効化」を参照してください。

コンテンツスイッチングを有効にする

October 7, 2021

選択キャッシュリダイレクトを構成するには、アプライアンスのロードバランシング機能とキャッシュリダイレクト機能の両方を有効にした後、コンテンツの切り替えを有効にする必要があります。

CLI を使用したコンテンツの切り替えの有効化

コマンドプロンプトで入力します。

```
-  enable ns feature CS

-  show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature cs
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 4)     Content Switching               CS                   ON
 5)     Cache Redirection               CR                   ON
          ...
          ...
          ...
 23)    HTML Injection                  HTMLInjection        ON
 24)    appliance Push                  push                 OFF
 Done
<!--NeedCopy-->
```

GUI を使用したキャッシュリダイレクトとロードバランシングの有効化

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. 詳細ウィンドウの [モードと機能] で、[基本機能の構成] をクリックします。
3. [基本機能の設定] ダイアログボックスで、[コンテンツの切り替え] の横にあるチェックボックスをオンにし、[OK] をクリックします。
4. 機能を有効化/無効化しますか? ダイアログボックスで、[はい] をクリックします。

キャッシュ用の負荷分散仮想サーバーの構成

October 7, 2021

使用するキャッシュサーバーのタイプごとに、ロードバランシング仮想サーバーと HTTP サービスを作成します。たとえば、あるキャッシュサーバーの JPEG ファイルと、別のキャッシュサーバーの GIF ファイルを提供し、残りのコンテンツに 3 つ目のキャッシュサーバーを使用する場合は、3 つのタイプのキャッシュサーバーごとに HTTP サービスと仮想サーバーを作成します。次に、各サービスをそれぞれの仮想サーバーにバインドします。

負荷分散仮想サーバーの作成方法の詳細については、「負荷分散仮想サーバーを作成する」を参照してください。

キャッシュサーバーを表すサービスを構成する方法の詳細については、「HTTP サービスの構成」を参照してください。

サービスを仮想サーバーにバインドする方法の詳細については、「負荷分散仮想サーバーへのサービスのバインド/バインド解除」を参照してください。

透過プロキシキャッシュリダイレクトサーバーの作成方法の詳細については、「キャッシュリダイレクト仮想サーバーの構成」および「TRANSTARENT」タイプの仮想サーバーの作成を参照してください。

組み込みのキャッシュリダイレクトポリシーをキャッシュリダイレクト仮想サーバーにバインドする方法については、キャッシュリダイレクト仮想サーバーへのポリシーのバインドを参照してください。

特定の種類のコンテンツに対してキャッシュリダイレクトポリシーを構成する

.png 拡張子または.jpeg 拡張子を含む要求をキャッシュ可能として識別するには、キャッシュリダイレクトポリシーを構成し、キャッシュリダイレクション仮想サーバーにバインドします。

注：リクエストがポリシーと一致する場合、Citrix ADC アプライアンスはリクエストをオリジンサーバーに転送します。その結果、次の手順では、拡張子が「.png」または「.jpeg」
でない要求を照合するようにポリシーを設定します。

特定のタイプのコンテンツに対してキャッシュリダイレクトを構成するには、「キャッシュリダイレクトポリシーを構成する」の説明に従って、単純な式を使用するポリシーを構成します。

コンテンツスイッチングのポリシーを構成する

October 7, 2021

あるキャッシュサーバーまたはファームでキャッシュされる特定の種類のコンテンツを識別し、別のキャッシュサーバーまたはファームからサービスを提供する他の種類のコンテンツを識別するコンテンツスイッチポリシーを作成する必要があります。たとえば、拡張子が.png および.jpeg のイメージファイルの場所を決定するポリシーを構成できます。

コンテンツスイッチングポリシーを定義したら、それをキャッシュリダイレクト仮想サーバーにバインドし、負荷分散仮想サーバーを指定します。ポリシーに一致する要求は、名前付きロードバランシング仮想サーバに転送されます。コンテンツスイッチングポリシーに一致しない要求は、キャッシュの既定の負荷分散仮想サーバーに転送されます。

コンテンツスイッチング機能とコンテンツスイッチングポリシーの設定の詳細については、「コンテンツ切り替え」を参照してください。

コンテンツスイッチングポリシーを作成してから、キャッシュリダイレクト仮想サーバーにバインドする必要があります。

コマンド **CLI** を使用してコンテンツスイッチングポリシーを作成する

コマンドラインで、次のように入力します。

```
-  add cs policy <policyName> [-url <string> | -rule <expression>]
-  show cs policy [<policyName>]
<!--NeedCopy-->
```

例:

```
> add cs policy Policy-CS-JPEG -rule ”REQ.HTTP.URL == '/\*.jpeg'”
 Done
> show cs policy Policy-CS-JPEG
         Rule: REQ.HTTP.URL == '/\*.jpeg'         Policy: Policy-CS-
            JPEG
         Hits: 0
 Done
>

> add cs policy Policy-CS-GIF -rule ”REQ.HTTP.URL == '/ *.png'”
 Done
> show cs policy Policy-CS-GIF
         Rule: REQ.HTTP.URL == '/ *.png'         Policy: Policy-CS-GIF
         Hits: 0
 Done
>

> add cs policy Policy-CS-JPEG-URL -url /\*.jpg
 Done
> show cs policy Policy-CS-JPEG-URL
         URL: /\*.jpg     Policy: Policy-CS-JPEG-URL
         Hits: 0
```



```
 Done
>

> add cs policy Policy-CS-GIF-URL -url /\*.png
 Done
> show cs policy Policy-CS-GIF-URL
         URL: /\*.png     Policy: Policy-CS-GIF-URL
         Hits: 0
 Done
<!--NeedCopy-->
```

GUI を使用して URL ベースのコンテンツスイッチングポリシーを作成する

1. Traffic Management > Content Switching > Policies に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [コンテンツスイッチングポリシーの作成] ダイアログボックスの [名前] テキストボックスに、ポリシーの名前を入力します。
4. [URL] ラジオボタンを選択します。
5.「値」テキストボックスに、文字列値（**/sports** など）を入力します。
6. [作成] をクリックし、[閉じる] をクリックします。作成したポリシーが [コンテンツスイッチングポリシー] ページに表示されます。

GUI を使用したルールベースのコンテンツスイッチングポリシーの作成

1. Traffic Management > Content Switching > Policies に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [コンテンツスイッチングポリシーの作成] ダイアログボックスの [名前] テキストボックスに、ポリシーの名前を入力します。
4. [式] ラジオボタンを選択し、[構成] をクリックします。
5. [式を作成] ダイアログボックスで、使用する式の構文を選択します。
   - デフォルトの構文を使用する場合は、デフォルトを受け入れ、次の手順に進みます。
   - 従来の構文を使用する場合は、[クラシック構文に切り替える] をクリックします。

   ダイアログボックスの [式] 部分が、選択した内容に合わせて変更されます。デフォルトの構文式ビューでは、従来の構文式ビューよりも要素数が少なくなります。デフォルトの構文の [式] ビューでは、プレビューウィンドウではなく、ボタンによって式エバリュエータにアクセスできます。エバリュエータは、入力した式を評価して有効であることを確認し、式の効果の分析を表示します。
6. ポリシー式を入力します。

   高度な構文の使用の詳細については、「高度なポリシー式の構成: はじめに」を参照してください。

7. [作成] をクリックし、[閉じる] をクリックします。作成したポリシーが [**Content Switchitching Policies**] ペインに表示されます。

CLI を使用してコンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドする

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドし、構成を確認します。

```
-  bind cs vserver <name> <targetVserver> [-policyName <string>]
-  show cs vserver [<name>]
<!--NeedCopy-->
```

例:

```
> bind cs vserver Vserver-CR-1 lbcachejpeg -policyName Policy-CS-JPEG
 Done
> bind cs vserver Vserver-CR-1 lbcachegif -policyName Policy-CS-GIF
 Done
> show cs vserver Vserver-CR-1
        Vserver-CR-1 (10.102.29.60:80) - HTTP   Type: CONTENT
        State: UP
        Last state change was at Fri Jul  2 12:53:45 2010
        Time since last state change: 0 days, 00:00:58.920
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Cacheable: YES
        Vserver IP and Port insertion: OFF
        Case Sensitivity: ON
        Push: DISABLED  Push VServer:
        Push Label Rule: none

1)      Policy: Policy-CS-JPEG  Target: lbcachejpeg     Priority: 0
       Hits: 0
2)      Policy: Policy-CS-GIF   Target: lbcachegif      Priority: 0
       Hits: 0
 Done
>
```



```
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドする

1. **[**トラフィック管理**]** > **[**コンテンツスイッチング**]** > **[**仮想サーバー**]** に移動します。
2. 詳細ウィンドウで、ポリシーをバインドする仮想サーバー (**Vserver-CS-1** など) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (コンテンツの切り替え)] ダイアログボックスの [ポリシー] タブで、[CSW] をクリックし、[ポリシーの挿入] をクリックします。
4. [ポリシー名] 列で、コンテンツスイッチ仮想サーバーに対して構成するポリシーを選択します。
5. [**Target**] 列で緑色の矢印をクリックし、一覧からターゲットの負荷分散仮想サーバーを選択します。
6. [**OK**] をクリックします。

ポリシー評価の優先順位の設定

October 7, 2021

コンテンツスイッチポリシーは、さまざまなコンテンツタイプに対応するための汎用的な構成であるルール、または特定のキャッシュサーバーに送信する必要があるコンテンツタイプをより具体的に定義する URL のいずれかに基づいて構成できます。基本的に、同じコンテンツは、ルールベースのポリシーまたは URL ベースのポリシーのいずれかで定義できます。

いずれかのタイプのコンテンツスイッチングポリシーをキャッシュリダイレクト仮想サーバーにバインドしたら、ルールベースまたは URL ベースのポリシーに優先するように仮想サーバーを構成できます。これにより、特定の要求がどのサーバーに送信されるかが決まります。

ポリシー評価の優先順位を構成するには、precedence パラメータを使用します。このパラメータは、コンテンツリダイレクト仮想サーバーで優先されるポリシーの種類 (URL または RULE) を指定します。

可能な値:RULE、URL

デフォルト値:RULE

CLI を使用したポリシー評価の優先順位の設定

コマンドプロンプトで次のコマンドを入力して、ポリシー評価の優先順位を構成し、構成を確認します。


```
-  set cr vserver <name> [-precedence (RULE | URL)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -precedence URL
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

GUI を使用したポリシー評価の優先順位の設定

1. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、優先順位を構成する仮想サーバー (**Vserver-CS-1** など) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (コンテンツの切り替え)] ダイアログボックスの [詳細設定] タブの [優先順位] の横にある [ルールまたは URL] をクリックし、[OK] をクリックします。

キャッシュリダイレクト仮想サーバーの管理

October 7, 2021

キャッシュリダイレクト仮想サーバーを管理するには、キャッシュリダイレクトの統計情報を表示する必要があります。キャッシュリダイレクションサーバーを有効または無効にしたり、ポリシーヒットをオリジンではなくキャッシ

ュにダイレクトする必要がある場合があります。管理タスクには、キャッシュ・リダイレクト仮想サーバのバックアップとクライアント接続の管理も含まれます。

キャッシュ・リダイレクト仮想サーバの統計情報の表示

October 7, 2021

キャッシュリダイレクション仮想サーバーのプロパティと、キャッシュリダイレクション仮想サーバーを通過したトラフィックの統計情報を表示できます。また、負荷分散仮想サーバーにバインドしたキャッシュリダイレクト仮想サーバーおよびポリシーを表示することもできます。

特定のキャッシュリダイレクション仮想サーバの統計情報を表示するには、name パラメータを使用して、統計情報を表示する仮想サーバの名前を指定します。それ以外の場合、すべてのキャッシュリダイレクション仮想サーバの統計情報が表示されます。最大長：127

CLI を使用したキャッシュリダイレクト仮想サーバーの統計情報の表示

コマンドプロンプトで入力します。

```
stat cr vserver [<name>]
```

例:

```
1  > stat cr vserver Vserver-CRD-1
2
3  Vserver Summary
4                            IP  port     Protocol        State
5  Vser...CRD-1         0.0.0.0    80         HTTP           UP
6
7  VServer Stats:
8                                          Rate (/s)
                                              Total
9  Requests                                       0
                        0
10 Responses                                      0
                        0
11 Request bytes                                  0
                        0
12 Response bytes                                 0
                        0
13
14  Done
```



```
15 >
16 <!--NeedCopy-->
```

GUI を使用したキャッシュリダイレクト仮想サーバーの統計情報の表示

1. 「トラフィック管理」>「キャッシュリダイレクト」>「仮想サーバー」に移動します。
2. 詳細ウィンドウで、統計情報を表示する仮想サーバー (たとえば、**Vserver-CRD-1**) を選択し、[統計] をクリックします。

すべてのキャッシュリダイレクション仮想サーバの基本統計情報を表示するには、サーバ名を省略します。サーバ名を含めて、仮想サーバを通過する要求と応答の数とサイズなど、その仮想サーバの詳細な統計を表示します。

監視ユーティリティとダッシュボードユーティリティを使用して、キャッシュリダイレクト仮想サーバーの統計情報を表示する

1. 監視ユーティリティを使用して統計を表示するには、[監視] タブをクリックします。
2. [グループの選択] ドロップダウンメニューで、[CR 仮想サーバー] を選択します。キャッシュリダイレクション仮想サーバのリストが表示されます。
3. ダッシュボードユーティリティを使用して統計を表示するには、[ダッシュボード] タブをクリックします。
4. [統計ユーティリティ] の横にある [アプレットクライアント] または [Web Start クライアント] をクリックします。
5. [グループの選択] ドロップダウンメニューで、[CR 仮想サーバー] を選択します。ダッシュボードには、キャッシュリダイレクト仮想サーバーの概要統計が表示されます。
6. 仮想サーバのアクティビティのチャートを表示するには、[Chart] をクリックします。仮想サーバの統計情報がグラフィカルに表示されます。

キャッシュのリダイレクト仮想サーバーを有効または無効にする

October 7, 2021

キャッシュリダイレクト仮想サーバーを作成すると、デフォルトで有効になります。キャッシュのリダイレクト仮想サーバーを無効にすると、その状態が OUT OF SERVICE に変わり、キャッシュ可能なクライアント要求のリダイレクトが停止します。ただし、Citrix ADC アプライアンスは、この仮想サーバーの IP アドレスに対する ARP および PING 要求に応答し続けます。

CLI を使用してキャッシュリダイレクト仮想サーバーを有効または無効にする

コマンドラインで、次のいずれかのコマンドを入力します。


```
-  enable cr vserver <name>
-  show cr vserver <name>
-  disable cr vserver <name>
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> enable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>

> disable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: OUT OF SERVICE  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```


GUI を使用してキャッシュリダイレクト仮想サーバーを有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. ナビゲーションウィンドウで、[キャッシュリダイレクト] を展開し、[仮想サーバー] をクリックします。
3. 詳細ウィンドウで、有効または無効にする仮想サーバー (たとえば、**Vserver-CRD-1**) を選択し、[統計] をクリックします。
4. [続行] ダイアログボックスで、[はい] をクリックします。

オリジン Web サーバーではなくキャッシュへのポリシーリクエストの直接的な要求

October 7, 2021

デフォルトでは、リクエストがポリシーに一致すると、Citrix ADC アプライアンスは、キャッシュリダイレクトの構成に応じて、オリジンサーバーに直接送信するか、オリジンの負荷分散仮想サーバーに要求を転送します。

要求がポリシーに一致したときに、要求がキャッシュの負荷分散仮想サーバーに転送されるように、デフォルトの動作を変更できます。

ポリシー要求の送信先をオリジンまたはキャッシュに変更するには、パラメータを使用します。`onPolicyMatch` パラメータを使用して、キャッシュリダイレクトポリシーに一致するリクエストの送信先を指定します。

有効なオプションは次のとおりです。

1. `CACHE` -一致するすべてのリクエストをキャッシュに転送します。
2. `ORIGIN` -一致するすべてのリクエストをオリジンサーバーに送信します。

> 注:
>
> このオプションが機能するには、キャッシュリダイレクトタイプを`POLICY`として選択する必要があります。

可能な値: `CACHE`, `ORIGIN`

デフォルト値: `ORIGIN`

CLI を使用して、ポリシー要求の宛先をオリジンまたはキャッシュに変更する

コマンドプロンプトで次のコマンドを入力して、ポリシーヒットの宛先を変更し、設定を確認します。

```
set cr vserver <name> [-onPolicyMatch (ORIGIN | CACHE)]
<!--NeedCopy-->
```

```
show cr vserver <name>
<!--NeedCopy-->
```


例:

```
> set cr vserver Vserver-CRD-1 -onPolicyMatch CACHE
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用して、ポリシーヒットの送信先をオリジンまたはキャッシュに変更する

1. [トラフィック管理] **>** [キャッシュリダイレクト] **>** [仮想サーバー] に移動します。
2. 詳細ペインで、ポリシー要求の宛先を変更する仮想サーバー (たとえば、**vserver-CRD-1**) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 **(**キャッシュリダイレクト**)**] ダイアログボックスで、[詳細設定] をクリックします。
4. [リダイレクト先] ドロップダウンリストから [キャッシュ] または [オリジン] を選択します。
5. [**OK**] をクリックします。

キャッシュリダイレクト仮想サーバーをバックアップする

October 7, 2021

プライマリ仮想サーバーに障害が発生した場合、または過剰なトラフィックを処理できない場合、キャッシュのリダイレクトが失敗することがあります。プライマリ仮想サーバに障害が発生したときにトラフィックの処理を引き継ぐバックアップ仮想サーバを指定できます。

バックアップキャッシュリダイレクト仮想サーバーを指定するには、backupVServer パラメーターを使用して、バックアップ仮想サーバーを指定します。最大長：127

CLI を使用したバックアップキャッシュリダイレクト仮想サーバの指定

コマンドプロンプトで次のコマンドを入力して、バックアップキャッシュリダイレクト仮想サーバーを指定し、構成を確認します。

```
-  set cr vserver <name> [-backupVServer <string>]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -backupVServer Vserver-CRD-2
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用したバックアップキャッシュリダイレクト仮想サーバの指定

1. [トラフィック管理] **>** [キャッシュリダイレクト] **>** [仮想サーバー] に移動します。
2. 詳細ペインで、ポリシー要求の宛先を変更する仮想サーバー (たとえば、**vserver-CRD-1**) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (キャッシュリダイレクト)] ダイアログボックスで、[詳細設定] タブを選択します。
4. [仮想サーバーのバックアップ] ドロップダウンリストで、仮想サーバーを選択します。
5. [OK] をクリックします。

仮想サーバーのクライアント接続の管理

October 7, 2021

キャッシュのリダイレクト仮想サーバーでタイムアウトを構成して、クライアント接続を無期限に開いたままにしないようにすることができます。また、Via ヘッダーをリクエストに挿入することもできます。ネットワークの輻輳を減らすために、開いている TCP 接続を再利用することができます。キャッシュリダイレクト仮想サーバー接続の遅延クリーンアップを有効または無効にできます。

設定に従って、PING 要求に ICMP 応答を送信するようにアプライアンスを設定できます。仮想サーバに対応する IP アドレスで、ICMP 応答を VSVR_CNTRLD に設定し、仮想サーバ上で ICMP VSERVER 応答を設定します。

仮想サーバーでは、次の設定を行うことができます。

- すべての仮想サーバーで ICMP VSERVER RESPONSE を PASSIVE に設定すると、アプライアンスは常に応答します。
- すべての仮想サーバで ICMP VSERVER RESPONSE を ACTIVE に設定すると、アプライアンスは 1 つの仮想サーバが UP でも応答します。
- ICMP VSERVER RESPONSE を一部で ACTIVE に設定し、他のサーバで PASSIVE に設定した場合、アプライアンスは ACTIVE に設定された 1 つの仮想サーバが UP でも応答します。

このドキュメントでは、次の内容について説明します。

- クライアントタイムアウトの構成
- リクエストに Via ヘッダーを挿入する
- TCP 接続を再利用する
- 遅延接続クリーンアップの構成

クライアントタイムアウトの構成

キャッシュのリダイレクト仮想サーバーのタイムアウト値を設定することで、クライアント要求の有効期限を指定できます。タイムアウト値は、キャッシュリダイレクション仮想サーバがクライアント要求に対する応答を受信するまで待機する秒数です。

タイムアウト値を構成するには、cltTimeout パラメーターを使用します。cltTimeout パラメーターは、Citrix ADC アプライアンスがアイドル状態のクライアント接続を閉じるまでの時間を秒単位で指定します。デフォルト値は、HTTP/SSL ベースのサービスの場合は 180 秒、TCP ベースのサービスでは 9000 秒です。

CLI を使用したクライアントのタイムアウトの設定

コマンドプロンプトで次のコマンドを入力して、クライアントタイムアウトを構成し、構成を確認します。


```
-  set cr vserver <name> [-cltTimeout <secs>]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -cltTimeout 6000
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用したクライアントのタイムアウトの構成

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、クライアントタイムアウトを構成する仮想サーバー (**Vserver-CRD-1** など) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (キャッシュリダイレクト)] ダイアログボックスで、[詳細設定] タブを選択します。
4. [クライアントタイムアウト (秒)] テキストボックスに、タイムアウト値を秒単位で入力します。
5. [OK] をクリックします。

リクエストに **Via** ヘッダーを挿入する

Via ヘッダーは、要求または応答の開始点と終了点の間のプロトコルと受信者をリストし、要求が送信されたプロキシのサーバーに通知します。各 HTTP 要求に Via ヘッダーを挿入するように、キャッシュリダイレクション仮想サーバーを構成できます。via パラメーターは、キャッシュリダイレクト仮想サーバーを作成するときにデフォルトで有効になります。

クライアント要求での Via ヘッダー挿入を有効または無効にするには、via パラメーターを使用します。このパラメーターは、HTTP 要求に Via ヘッダーを挿入する際のシステムの状態を指定します。

設定可能な値: オン、オフ

デフォルト値: オン

CLI を使用して、クライアント要求での **Via** ヘッダー挿入を有効または無効にする

コマンドプロンプトで入力します。

```
-  set cr vserver <name> [-via (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例**:**

```
> set cr vserver Vserver-CRD-1 -via ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

GUI を使用してクライアント要求での **Via** ヘッダー挿入を有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、クライアントタイムアウトを構成する仮想サーバー (**Vserver-CRD-1** など) を選択し、[開く] をクリックします。

3. [仮想サーバーの構成 (キャッシュリダイレクト)] ダイアログボックスで、[詳細設定] タブを選択します。
4. [ビア] チェックボックスをオンにします。
5. [OK] をクリックします。

TCP 接続を再利用する

クライアント接続間でキャッシュサーバーとオリジンサーバーへの TCP 接続を再利用するように、Citrix ADC アプライアンスを構成できます。これにより、サーバとアプライアンス間のセッションを確立するために必要な時間が節約され、パフォーマンスが向上します。キャッシュのリダイレクト仮想サーバーを作成すると、再利用オプションがデフォルトで有効になります。

TCP 接続の再利用を有効または無効にするには、reuse パラメーターを使用します。このパラメーターは、クライアント接続間でキャッシュまたはオリジンサーバーへの TCP 接続の再利用の状態を指定します。

設定可能な値: オン、オフ

デフォルト値: オン

CLI を使用した TCP 接続の再利用の有効化または無効化

コマンドプロンプトで入力します。

```
-  set cr vserver <name> [-reuse (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例:

```
> set cr vserver Vserver-CRD-1 -reuse ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
```



```
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

GUI を使用して **TCP** 接続の再利用を有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、クライアントタイムアウトを構成する仮想サーバー (**Vserver-CRD-1** など) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (キャッシュリダイレクト)] ダイアログボックスで、[詳細設定] タブを選択します。
4. [再利用] チェックボックスをオンにします。
5. [OK] をクリックします。

遅延接続クリーンアップの構成

ダウン状態のフラッシュオプションは、キャッシュリダイレクト仮想サーバー上の接続の遅延クリーンアップを実行します。キャッシュのリダイレクト仮想サーバーを作成すると、ダウン状態のフラッシュオプションがデフォルトで有効になります。

ダウン状態のフラッシュオプションを有効または無効にするには、downStateFlush パラメータを設定します。

設定可能な値: ENABLED, DISABLED

デフォルト値:ENABLED

CLI を使用したダウン状態のフラッシュオプションの無効化

コマンドプロンプトで次のコマンドを入力して、遅延接続のクリーンアップを構成し、構成を確認します。

```
-  set cr vserver <name> [-downStateFlush (ENABLED | DISABLED)]
-  show cr vserver <name>
<!--NeedCopy-->
```

例**:**

```
> set cr vserver Vserver-CRD-1 -downStateFlush ENABLED
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
```



```
7           Down state flush: ENABLED
8           Disable Primary Vserver On Down : DISABLED
9           Default:        Content Precedence: URL Cache: TRANSPARENT
10          On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
11          Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
12          Backup: Vserver-CRD-2
13
14  1)      Cache bypass  Policy: bypass-cache-control
15  2)      Cache bypass  Policy: Policy-CRD
16   Done
17  <!--NeedCopy-->
```

GUI を使用して **TCP** 接続の再利用を有効または無効にする

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、クライアントタイムアウトを構成する仮想サーバー (**Vserver-CRD-1** など) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (キャッシュリダイレクト)] ダイアログボックスで、[詳細設定] タブをクリックします。
4. [ダウン状態のフラッシュ] チェックボックスをオンにします。
5. [OK] をクリックします。

UDP 仮想サーバーの外部 **TCP** ヘルスチェックを有効にする

October 7, 2021

パブリッククラウドでは、ネイティブロードバランサーを第 1 層として使用する場合、Citrix ADC アプライアンスを第 2 層ロードバランサーとして使用できます。ネイティブロードバランサーは、アプリケーションロードバランサー (ALB) またはネットワークロードバランサー (NLB) です。パブリッククラウドのほとんどは、ネイティブのロードバランサで UDP 正常性プローブをサポートしていません。UDP アプリケーションの状態を監視するために、パブリッククラウドでは、サービスに TCP ベースのエンドポイントを追加することをお勧めします。エンドポイントは UDP アプリケーションの正常性を反映します。

Citrix ADC アプライアンスは、UDP 仮想サーバーの外部 TCP ベースのヘルスチェックをサポートしています。この機能では、キャッシュリダイレクト仮想サーバの VIP と、設定されたポートに TCP リスナーが導入されます。TCP リスナーは、仮想サーバーのステータスを反映します。

CLI を使用して **UDP** 仮想サーバーの外部 **TCP** ヘルスチェックを有効にするには

コマンドプロンプトで次のコマンドを入力して、TcpProbeport オプションを指定して外部 TCP ヘルスチェックを有効にします。


```
add cr vserver <name> <serviceType> -tcpProbePort <tcpProbePort>

<!--NeedCopy-->
```

例**:**

```
add cr vserver Vserver-CR-1 HTTP -tcpProbePort 80
<!--NeedCopy-->
```

GUI を使用して **UDP** 仮想サーバーの外部 **TCP** ヘルスチェックを有効にするには

1. [トラフィック管理**] > [**キャッシュリダイレクト**] > [**仮想サーバー] に移動し、仮想サーバーを作成します。
2. [**Add**] をクリックして、仮想サーバーを作成します。
3. [基本設定] ペインで、[**TCP** プローブポート**]** フィールドにポート番号を追加します。
4. [**OK**] をクリックします。

N 層キャッシュのリダイレクト

October 7, 2021

大量のキャッシュデータ (通常は数ギガバイト/秒) を効率的に処理するために、インターネットサービスプロバイダ (ISP) は複数の専用キャッシュサーバーを展開します。Citrix ADC アプライアンスのキャッシュリダイレクト機能は、キャッシュサーバーの負荷分散に役立ちますが、1 台または 2 台のアプライアンスで大量のトラフィックを効率的に処理できない場合があります。

この問題を解決するには、Citrix ADC アプライアンスを 2 つの層（層）に展開します。この場合、上位層のアプライアンスは下位層のアプライアンスを負荷分散し、下位層のアプライアンスはキャッシュサーバーを負荷分散します。この配置を *n* 層キャッシュリダイレクトと呼びます。

監査やセキュリティなどの目的で、ISP は IP アドレス、提供された情報、対話の時間などのクライアントの詳細を追跡する必要があります。したがって、Citrix ADC アプライアンスを介したクライアント接続は完全に透過的である必要があります。ただし、Citrix ADC アプライアンスを並行して展開して透過キャッシュリダイレクトを構成する場合は、クライアントの IP アドレスをすべてのアプライアンスで共有する必要があります。クライアント IP アドレスを共有すると、競合が発生し、ルーター、キャッシュサーバー、オリジンサーバー、その他の Citrix ADC アプライアンスなどのネットワークデバイスが、応答の送信先となるアプライアンス、つまりクライアントを特定できなくなります。

N 層キャッシュリダイレクトの実装方法

この問題を解決するために、アプライアンスの n 層キャッシュリダイレクトは、ソースポート範囲を下層内のアプライアンス間で分割し、クライアント IP アドレスをキャッシュサーバに送信する要求に含めます。上位層の Citrix ADC アプライアンスは、アプライアンスへの不要な負荷を避けるために、セッションレス負荷分散を行うように構成されています。

下位層の Citrix ADC アプライアンスがキャッシュサーバーと通信する場合、マッピングされた IP アドレス（MIP）を使用してソース IP アドレスを表します。したがって、キャッシュサーバは、要求を受信したアプライアンスを識別し、同じアプライアンスに応答を送信できます。

下位層の Citrix ADC アプライアンスは、キャッシュサーバーに送信される要求のヘッダーにクライアント IP アドレスを挿入します。ヘッダーのクライアント IP は、アプライアンスがキャッシュサーバまたはキャッシュミスが発生した場合に、パケットを転送するクライアントを決定するのに役立ちます。オリジンサーバーは、リクエストヘッダーに挿入されたクライアント IP に従って送信される応答を決定します。

オリジナル・サーバーは、オリジナル・サーバーが要求を受信したソース・ポート番号を含む、上位層のアプライアンスに応答を送信します。ソースポート範囲全体（1024～65535）は、下位層の Citrix ADC アプライアンスに分散されます。各下位層のアプライアンスには、範囲内のアドレスのグループのみが割り当てられます。この割り当てにより、上位層のアプライアンスは、要求を元のサーバーに送信した下位層の Citrix ADC アプライアンスを明確に識別できます。したがって、上位層のアプライアンスは、正しい下位層のアプライアンスに応答を転送できます。

上位層の Citrix ADC アプライアンスはポリシーベースのルーティングを行うように構成され、ルーティングポリシーは送信元ポート範囲から宛先アプライアンスの IP アドレスを決定するように定義されます。

N 層 CRD の構成に必要なセットアップ

n 層キャッシュリダイレクトの機能には、次の設定が必要です。

各上位層 Citrix ADC アプライアンスの場合:

- レイヤ 3 モードを有効にします。
- トラフィックが宛先ポートの範囲に従って転送されるように、ポリシーベースルート（PBR）のポリシーを定義します。
- 負荷分散仮想サーバーを構成します。
- クライアントからのすべてのトラフィックをリッスンするように仮想サーバを設定します。サービスタイプ/プロトコルを ANY に設定し、IP アドレスをアスタリスク（*）に設定します。
- MAC ベースのリダイレクトモードでセッションレス負荷分散を有効にして、上位層の Citrix ADC アプライアンスの不要な負荷を回避します。
- [プロキシポートを使用] オプションが有効になっていることを確認します。
- 下位層アプライアンスごとにサービスを作成し、すべてのサービスを仮想サーバにバインドします。

下位層の Citrix ADC アプライアンスごとに、

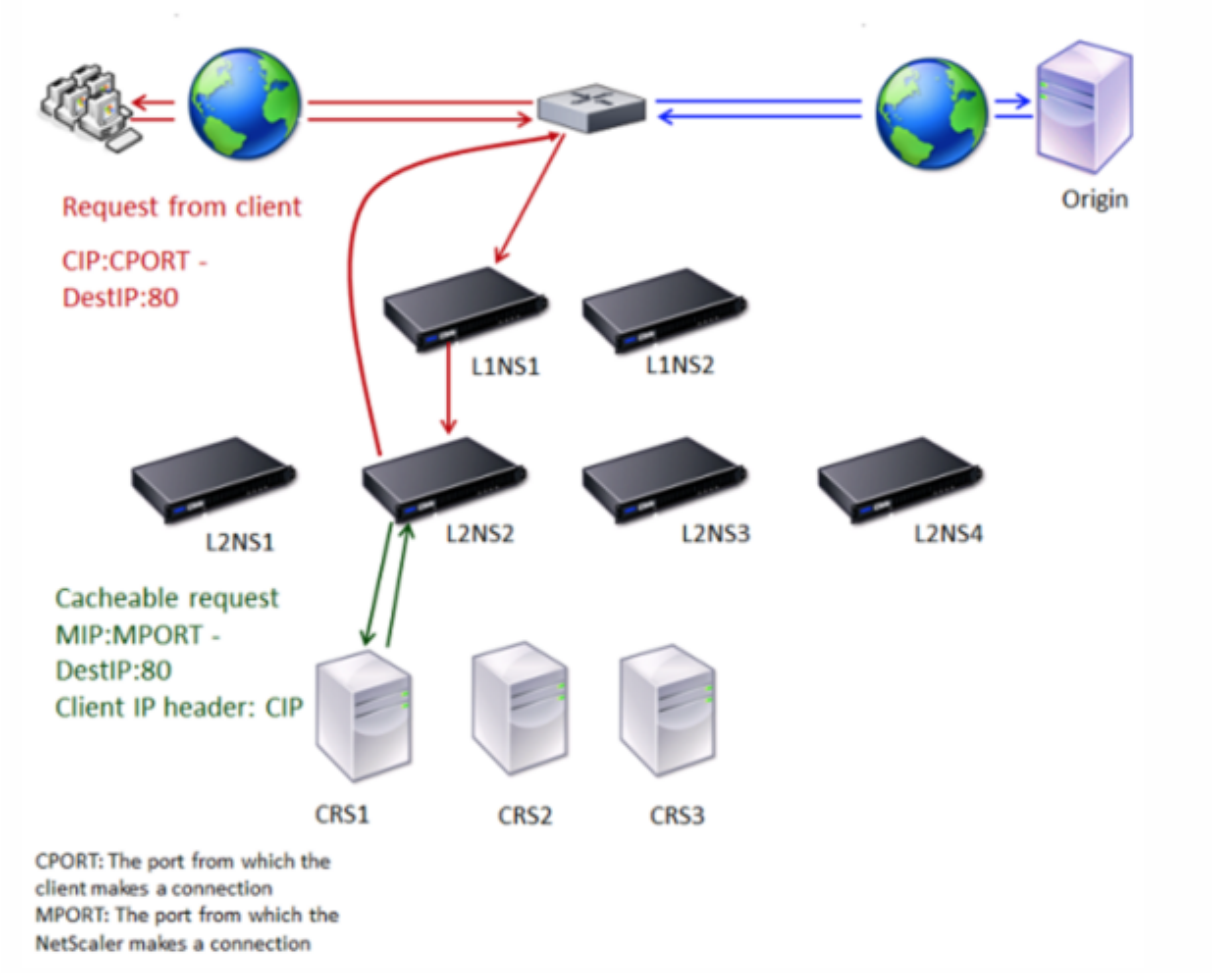
- アプライアンスのキャッシュリダイレクションポート範囲を設定します。下位層のアプライアンスごとに排他的な範囲を割り当てます。

- 負荷分散仮想サーバーを構成し、MAC ベースのリダイレクトを有効にします。
- このアプライアンスによって負荷分散されるキャッシュサーバーごとにサービスを作成します。サービスを作成するときに、ヘッダーにクライアント IP を挿入できるようにします。次に、負荷分散仮想サーバーにすべてのサービスをバインドします。
- トランスペアレントモードキャッシュリダイレクト仮想サーバーを次の設定で構成します。
  - [オリジン USIP] オプションを有効にします。
  - 送信元 IP 式を追加して、クライアント IP をヘッダーに含めます。
  - [ポート範囲を使用] オプションを有効にします。

キャッシュヒット時の **N** 層キャッシュリダイレクトの仕組み

次の図は、クライアント要求がキャッシュ可能で、応答がキャッシュサーバから送信された場合のキャッシュリダイレクトの動作を示しています。

図 1: キャッシュヒット時にキャッシュリダイレクト



2 つの Citrix ADC アプライアンス、L1NS1 と L1NS2 が上位層に展開され、4 つの Citrix ADC アプライアンス、L2NS1、L2NS2、L2NS3 および L2NS4 が下位層に展開されます。クライアント A は要求を送信し、ルータによっ

て転送されます。キャッシュサーバー CRS1、CRS2、および CRS3 は、キャッシュ要求を処理します。オリジナル・サーバー O は、キャッシュされていない要求を処理します。

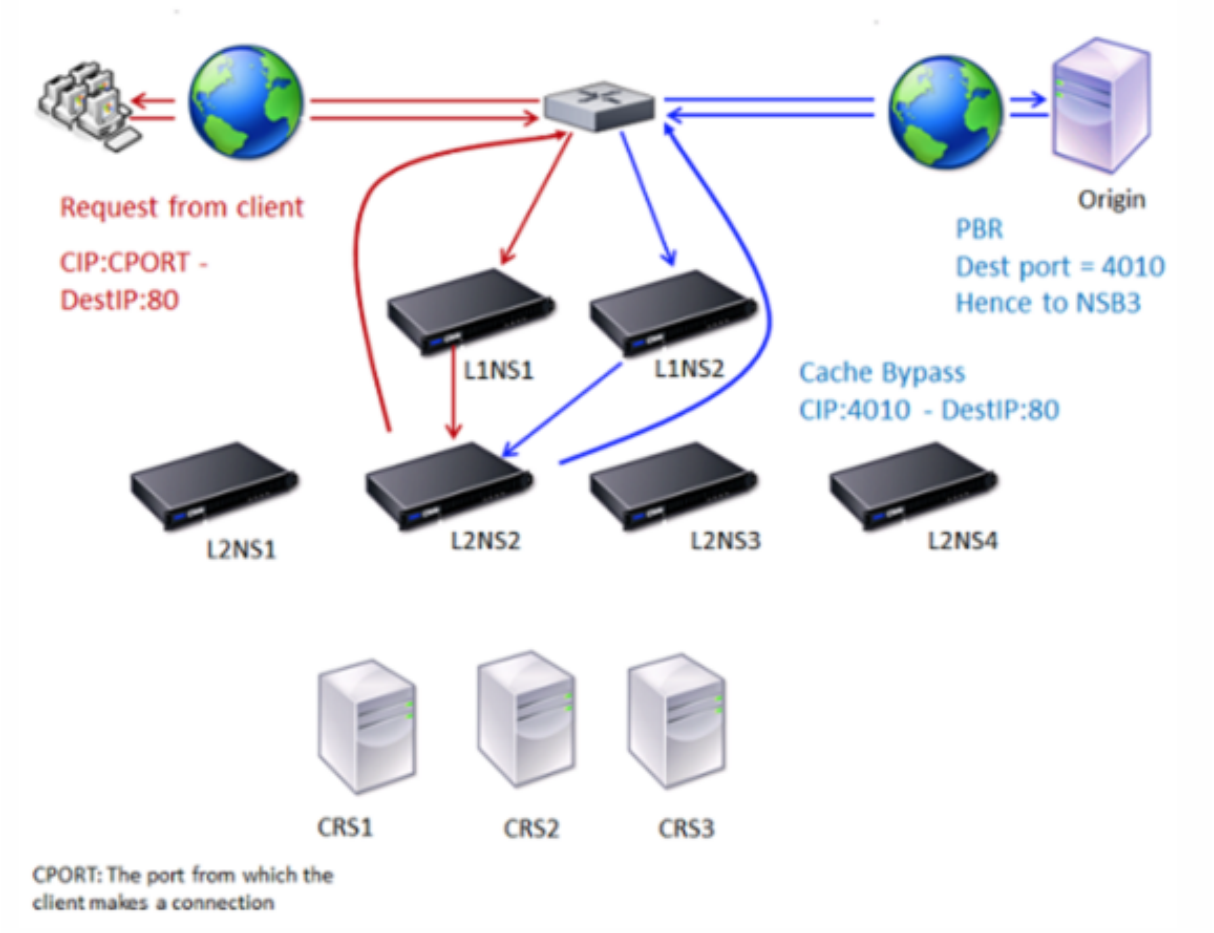
トラフィックフロー

1. クライアントは要求を送信し、ルータは要求を L1NS1 に転送します。
2. L1NS1 は、要求を L2NS2 にロードバランシングします。
3. L2NS2 は要求をキャッシュサーバ CRS1 にロードバランシングし、要求はキャッシュ可能です。L2NS2 は、要求ヘッダーにクライアント IP を含めます。
4. CRS1 は L2NS2 に応答を送信します。これは、L2NS2 が CRS1 に接続するときに MIP を送信元 IP アドレスとして使用するためです。
5. 要求ヘッダー内のクライアント IP アドレスの助けを借りて、L2NS2 は要求が来たクライアントを識別します。L2NS2 は応答を直接ルータに送信し、上位層のアプライアンスへの不要な負荷を回避します。
6. ルータは応答をクライアント A に転送します。

キャッシュバイパス中の **N** 層キャッシュリダイレクトの仕組み

次の図は、クライアント要求が応答のためにオリジンサーバーに送信されたときのキャッシュリダイレクトの動作を示しています。

図 2：キャッシュバイパスの場合、キャッシュリダイレクト



2 つの Citrix ADC アプライアンス、L1NS1 と L1NS2 が上位層に展開され、4 つの Citrix ADC アプライアンス、L2NS1、L2NS2、L2NS3 および L2NS4 が下位層に展開されます。クライアント A は要求を送信し、ルータによって転送されます。キャッシュサーバー CRS1、CRS2、および CRS3 は、キャッシュ要求を処理します。オリジナル・サーバー O は、キャッシュされていない要求を処理します。

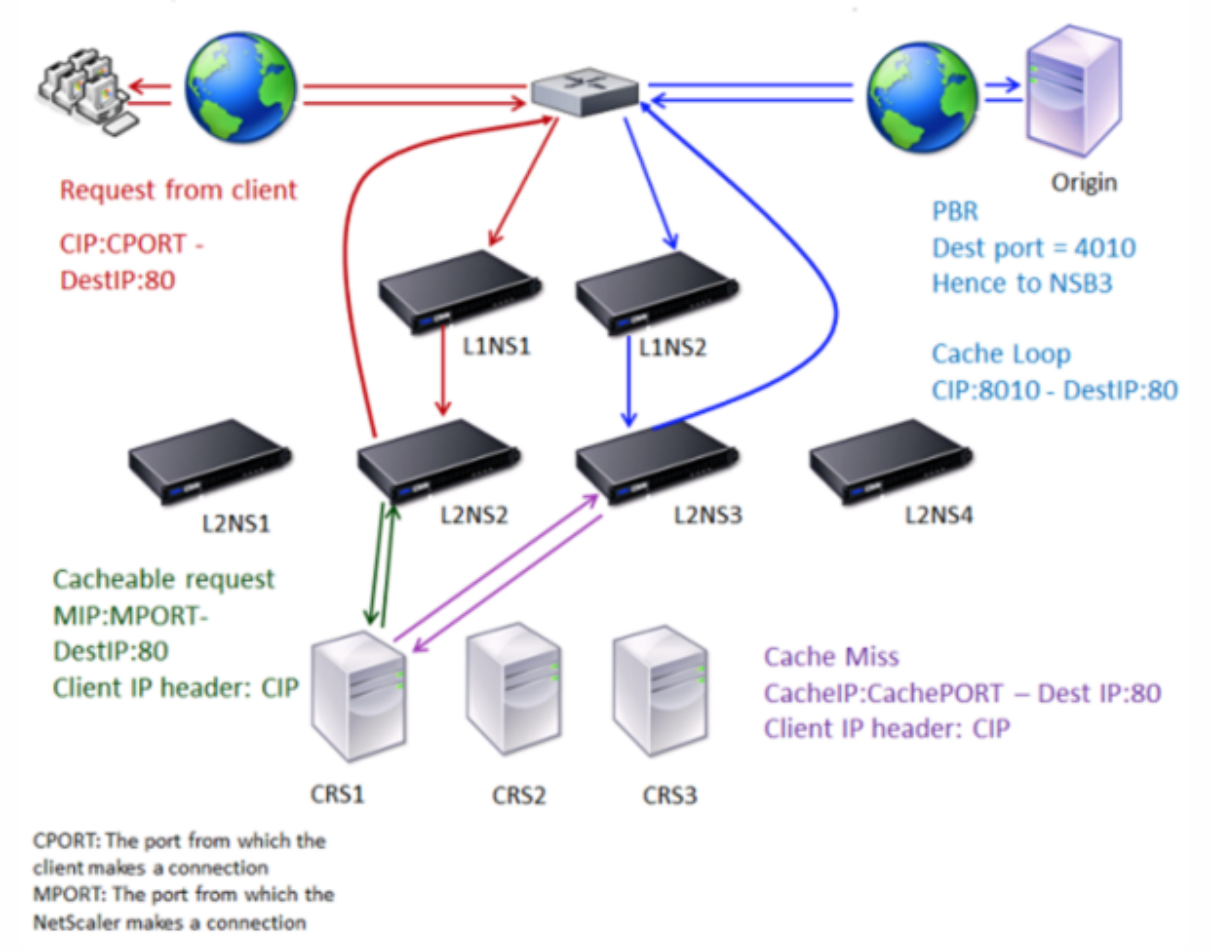
トラフィックフロー

1. クライアントは要求を送信し、ルータは要求を L1NS1 に転送します。
2. L1NS1 は、要求を L2NS2 にロードバランシングします。
3. 要求はキャッシュできません（キャッシュバイパス）。したがって、L2NS2 はルータ経由で送信元サーバに要求を送信します。
4. オリジンサーバーは、上位層アプライアンス L1NS2 に応答を送信します。
5. PBR ポリシーに従って、L1NS2 はトラフィックを下位層 L2NS2 の適切なアプライアンスに転送します。
6. L2NS2 は、要求ヘッダーのクライアント IP アドレスを使用して、要求の送信元クライアントを識別し、応答を直接ルータに送信します。これにより、上位層のアプライアンスへの不要な負荷を回避できます。
7. ルータは応答をクライアント A に転送します。

キャッシュミス時の **N** 層キャッシュリダイレクトの仕組み

次の図は、クライアント要求がキャッシュされていない場合のキャッシュリダイレクトの動作を示しています。

図 3：キャッシュミスによるキャッシュリダイレクト



2 つの Citrix ADC アプライアンス、L1NS1 と L1NS2 が上位層に展開され、4 つの Citrix ADC アプライアンス、L2NS1、L2NS2、L2NS3 および L2NS4 が下位層に展開されます。クライアント A は要求を送信し、ルータによって転送されます。キャッシュサーバー CRS1、CRS2、および CRS3 は、キャッシュ要求を処理します。オリジナル・サーバー O は、キャッシュされていない要求を処理します。

トラフィックフロー

1. クライアントは要求を送信し、ルータは要求を L1NS1 に転送します。
2. L1NS1 は、要求を L2NS2 にロードバランシングします。
3. L2NS2 は、要求がキャッシュ可能であるため、要求をキャッシュサーバ CRS1 にロードバランシングします。
4. CRS1 には応答（キャッシュミス）がありません。CRS1 は、下層のアプライアンスを介してオリジンサーバーに要求を転送します。L2NS3 はトラフィックを代行受信します。

5. L2NS3 は、ヘッダーからクライアント IP を受け取り、要求をオリジンサーバーに転送します。パケットに含まれる送信元ポートは、要求がオリジンサーバに送信される L2NS3 ポートです。
6. オリジンサーバーは、上位層アプライアンス L1NS2 に応答を送信します。
7. PBR ポリシーに従って、L1NS2 はトラフィックを下位層 L2NS3 の適切なアプライアンスに転送します。
8. L2NS3 は応答をルータに転送します。
9. ルータは応答をクライアント A に転送します。

上位層の **Citrix ADC** アプライアンスの構成

October 7, 2021

各上位層の Citrix ADC アプライアンスを次のように構成します。

コマンド **CLI** を使用して **n** 層キャッシュリダイレクト用の上位層アプライアンスを構成する

コマンドプロンプトで、次のコマンドを入力します。

- `add service \<name\>@ \<serviceIP\> \<serviceType\> \<port\>`

  追加するサービスごとにこのコマンドを実行します。

- `add lb vserver \<name\>@ ANY \* \<port\> -persistenceType \<persistenceMethod \> -lbMethod \<lbMethod\> -m MAC -sessionless ENABLED -cltTimeout \< client\\_Timeout\\_Value\>`

- `bind lb vserver \<name\>@ \<serviceName\>`

  バインドする各サービスに対してこのコマンドを実行します。

- `enable ns mode l3`

- `add ns pbr \<name\> \<action\> -srcPort \<sourcePortNumber\> -destPort \<startPortNumber-endPortNumber\> -nexthop \<serviceIpAddress\> -protocol TCP`

- `apply ns pbrs`

  必要なすべての PBR を追加した後、このコマンドを実行します。

GUI を使用して **n** 層キャッシュリダイレクト用の上位層アプライアンスを構成する

1. L3 モードを有効にします。
   a) ナビゲーションウィンドウで、[システム] をクリックし、[設定] をクリックします。
   b) [設定] グループで、[モードの構成] リンクをクリックします。
   c) [レイヤ 3 モード（IP 転送）] チェックボックスをオンにします。

d) ［OK］をクリックします。

2. ポリシーベースルーティング（PBR）を設定します。
   a) System > Network > PBRs に移動します。
   b) [ポリシーベースルーティング (PBR)] ウィンドウで、[追加] をクリックします。
   c) PBR の名前を入力します。
   d) アクションを [許可] として選択します。
   e) [Next Hop] ボックスに、サービスの IP アドレスを入力します。これは、下位層のアプライアンスを表します。
   f) [プロトコル] ドロップダウンリストから [TCP] を選択します。
   g) 送信元ポートと、追加する下位層のアプライアンスに対応する宛先ポートの範囲を入力します。
   h) ［作成］をクリックします。
   i) 詳細ペインで PBR を選択し、[Apply] をクリックします。
   j) 下位層アプライアンスごとに、ステップ（i）からステップ（vii）を繰り返します。
3. 下位層アプライアンスごとにサービスを作成します。
   a) ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
   b) 詳細ペインで、［Add］をクリックします。
   c) 名前、プロトコル、IP アドレス、およびポートを指定します。プロトコルは ANY である必要があります。
   d) ［作成］をクリックします。
4. 負荷分散仮想サーバーを構成します。
   a) ［Traffic Management］> ［Load Balancing］> ［Virtual Servers］の順に選択します。
   b) 詳細ペインで、［Add］をクリックします。
   c) 名前、プロトコル、IP アドレス、およびポートを指定します。プロトコルは ANY、IP アドレスは * である必要があります。
   d) ［サービス］タブで、下位層の Citrix ADC アプライアンスを表すサービスを選択します。
   e) [詳細設定] タブで、[MAC ベース] としてリダイレクションモードを選択し、[セッションレス] チェックボックスをオンにします。
   f) ［作成］をクリックします。

下位層の **Citrix ADC** アプライアンスの構成

October 7, 2021

各下位層の Citrix ADC アプライアンスを次のように構成します。

CLI を使用して **n** 層キャッシュリダイレクト用の下位層アプライアンスを構成する

コマンドプロンプトで、次のコマンドを入力します。

- `add service <name>@ <cacheServiceIP> <serviceType> <port> -cip ENABLED "ClientIP"-cachetype transparent`

各キャッシュサーバーに対して、を繰り返します。

- `add lb vserver <name>@ <serviceType> -m MAC`
- `bind lb vserver <name>@ <cacheServiceName>`

  各キャッシュサーバーに対して、を繰り返します。
- `add cr vserver <name> <serviceType> * <port> -srcIPExpr "HTTP.REQ.HEADER("ClientIP")"-originusip ON -usePortRange ON`
- `set ns param-crPortRange <startPortNumber-endPortNumber>`

GUI を使用して n 層キャッシュリダイレクト用の下位層アプライアンスを構成する

1. キャッシュサーバーごとにサービスを作成します。サービスを作成するには、次の手順に従います。
   a) [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
   b) 詳細ウィンドウで [追加] をクリックし、名前とプロトコルを指定します。[直接アドレス指定可能] チェックボックスをオフにします。
   c) [詳細設定] タブで、[グローバル上書き] チェックボックスと [クライアント IP] チェックボックスをオンにし、[ヘッダー] ボックスに「ClientIP」と入力します。
   d) [キャッシュタイプ] ボックスで、[透過キャッシュ] を選択します。
   e) [作成] をクリックします。
2. 負荷分散仮想サーバーを構成します。
   a) [トラフィック管理] > [負荷分散] > [仮想サービス] に移動します。
   b) 詳細ウィンドウで、[追加] をクリックし、名前、プロトコル、IP アドレス、およびポートを指定します。IP アドレスはアスタリスク (*) である必要があります。
   c) [サービス] タブで、キャッシュサーバーを表すサービスを選択します。
   d) [詳細設定] タブの [リダイレクションモード] で、[MAC ベース] を選択します。
   e) [作成] をクリックします。
3. キャッシュリダイレクト仮想サーバーを構成します。
   a) [トラフィック管理] > [負荷分散] > [仮想サービス] に移動します。
   b) 詳細ウィンドウで、[追加] をクリックし、名前、プロトコル、IP アドレス、およびポートを指定します。IP アドレスは * である必要があります。
   c) [キャッシュタイプ] で、[透明] を選択します。
   d) [詳細設定] タブの [キャッシュサーバー] ボックスで、新しい負荷分散仮想サーバーを選択し、[Origin USIP] および [ポート範囲を使用] チェックボックスをオンにします。[ソース IP 式] ボックスで、HTTP.REQ.HEADER (「クライアント IP」) と入力します。
   e) [作成] をクリックします。
4. アプライアンスの送信元ポート範囲を割り当てます。
   a) ナビゲーションウィンドウで、[システム] をクリックし、[設定] をクリックします。
   b) [設定] グループで、[グローバルシステム設定の変更] リンクをクリックします。

c) [キャッシュリダイレクトポート範囲] グループで、[開始ポート] にポート番号、[終了ポート] にポート番号を入力して、アプライアンスのポート範囲を指定します。
d) [OK] をクリックします。

要求の宛先 **IP** アドレスを発信元 **IP** アドレスに変換する

October 7, 2021

Citrix ADC アプライアンスで転送プロキシキャッシュリダイレクト仮想サーバーを構成して、キャッシュリダイレクト仮想サーバーに着陸した要求の宛先 IP アドレスを元のサーバーの IP アドレスに変換できます。この変換は、要求がキャッシュされたサーバーに送信されたか、オリジンサーバーに送信されたかに関係なく行われます。
以前は、コンテンツスイッチングポリシーを使用したキャッシュリダイレクトの制限により、サービスプロバイダー環境で転送プロキシキャッシュリダイレクト仮想サーバーを効果的に使用できませんでした。キャッシュリダイレクション仮想サーバは、パケットがキャッシュに送信されたときに、発信元 IP アドレスを宛先 IP に変換しませんでした。宛先 IP アドレスは、要求がキャッシュされたサーバーから提供された場合にのみ、オリジンサーバーのアドレスでした。

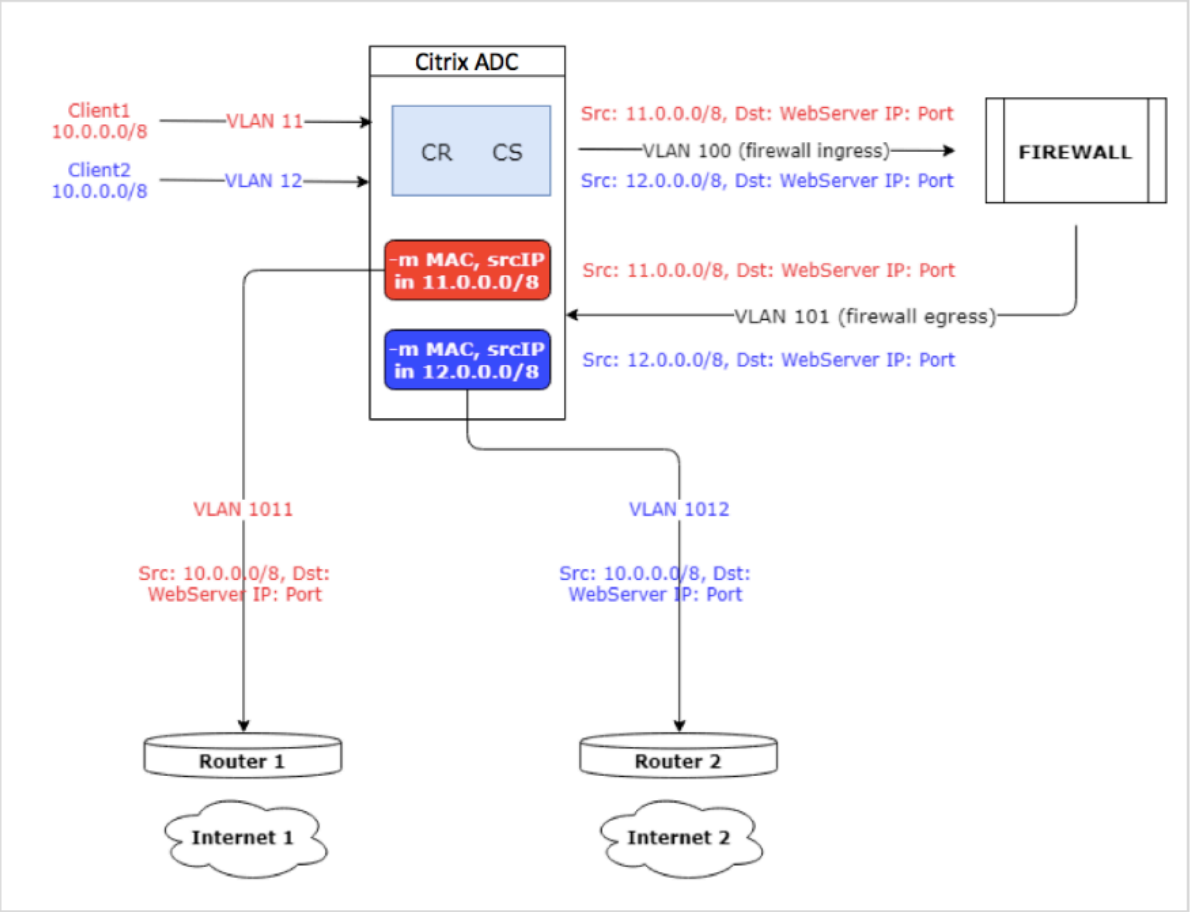
注：要求の宛先 IP アドレスから発信元 IP アドレスへの変換は、透過キャッシュリダイレクト仮想サーバーではサポートされていません。透過キャッシュリダイレクト仮想サーバーの場合、このオプションを OFF に設定する必要があります。

使用例

Citrix ADC アプライアンスがフォワードプロキシキャッシュのリダイレクト、ファイアウォール、および再利用されたクライアント IP アドレス用に構成されている環境では、ファイアウォールは再利用された IP アドレスを区別/使用できません。したがって、これらの再利用された IP アドレスは、異なる IP アドレスに変換する必要があります。再利用された IP アドレスを変換するには、Citrix ADC アプライアンスが以下を実行する必要があります。

1. DNS ロードバランシング仮想サーバーに宛先の解決を照会します。
2. 送信先の発信元 IP アドレスとポート番号を更新します。
3. 要求をファイアウォールに送り返します。

Citrix ADC アプライアンスがフォワードプロキシキャッシュリダイレクト、ファイアウォール、2 つのルーター（ルーター 1 とルーター 2）用に構成されている次の展開を検討してください。ネットワークトラフィックは、それぞれルータ 1 を経由してインターネット 1 に、ルータ 2 を経由してインターネット 2 に流れます。



この例では、クライアントからの入力要求は、VLAN11 または VLAN12 の 2 つの異なる VLAN から送られます。クライアントの IP アドレス（10.0.0.0）が再利用されます。
キャッシュのリダイレクトおよびコンテンツスイッチングポリシーに基づいて、リクエストはオリジンサーバーまたはファイアウォールに直接送信できます。

- 要求がファイアウォールをバイパスしてインターネットにアクセスする必要がある場合、入力要求 VLAN に基づいて、ルータ 1 またはルータ 2 のいずれかが選択され、要求はインターネット 1 またはインターネット 2 に送信されます。
- 要求がファイアウォールを通過する必要がある場合は、要求の送信元 IP を特定の IP アドレスに変換する必要があります。変換された IP アドレスは、要求が通過する VLAN を識別するために使用できます。たとえば、入力要求が VLAN11 から送信される場合、送信元 IP アドレスは 11.x.x.x に変換されます。要求が VLAN12 から送信される場合、送信元 IP アドレスは 12.x.x.x に変換されます。

ファイアウォールが要求を処理した後、要求はアプライアンスに返送されます。アプライアンスは、リスンポリシーとネットプロファイルの組み合わせを使用して、送信元 IP アドレスを元の IP アドレスに変換し、入力 VLAN ID に基づいてルータ 1 またはルータ 2 に要求を送信します。

注：キャッシュにバインドされている負荷分散仮想サーバーのモードは、常に MAC モードに設定する必要があります。この機能の IP モードはブロックされませんが、IP モードに設定すると、予期しない動作が発生します。

CLI を使用して、要求の宛先 **IP** アドレスとポート番号を送信元 **IP** アドレスに変換するには

コマンドプロンプトで次を入力します。

```
set cr vserver <vsname> -useoriginIpPortForCache <YES|NO>
<!--NeedCopy-->
```

例:

```
set cr vserver cvsrv1 -useoriginIpPortForCache YES
<!--NeedCopy-->
```

useoriginIpPortForCache が Yes に設定され、キャッシュされたサーバーから要求を処理する必要がある場合、要求の宛先 IP は元のサーバーの IP アドレスに変換されます。

注：useoriginIpPortForCache が有効になっている場合は、キャッシュにバインドされている負荷分散仮想サーバーを常に MAC モードに設定します。

GUI を使用して、要求の宛先 **IP** アドレスとポートを元の **IP** アドレスに変換するには

1. [トラフィック管理] > [キャッシュリダイレクト] > [仮想サーバー] に移動し、[追加] をクリックします。
2. キャッシュ・リダイレクト仮想サーバーの詳細を指定します。
3. 要求の宛先 **IP** アドレスを発信元 **IP** アドレスに変換するには、**[**キャッシュに発信元 **IP** ポートを使用] を選択します。
4. [**OK**] をクリックします。

クラスタリング

October 7, 2021

> 注
>
> この機能は、Citrix ADC Advanced エディションまたは Premium エディションのライセンスで利用できます。

Citrix ADC クラスタは、1 つのシステムイメージとして連携して動作する nCore アプライアンスのグループです。クラスタの各アプライアンスは、ノードと呼ばれます。クラスタには、1 つのアプライアンス、または最大 32 個の Citrix ADC nCore ハードウェアまたは仮想アプライアンスをノードとして持つことができます。

クライアントトラフィックは、高可用性、高スループット、およびスケーラビリティを提供するために、ノード間で分散されます。

クラスターを作成するには、次の手順を実行する必要があります。

- アプライアンスをクラスターノードとして追加します。
- ノード間の通信を設定します。
- クライアントおよびサーバーネットワークへのリンクを設定します。
- アプライアンスを構成し、クライアントとサーバーのトラフィックの分散を構成します。

Citrix ADC クラスタのサポートマトリックス

October 7, 2021

Citrix ADC アプライアンスのクラスタリングは、Citrix ADC 構成の幅広い機能をサポートします。

次の表に、Citrix ADC の機能と、異なる Citrix ADC リリースのクラスタセットアップにおけるサポート性のステータスを示します。13.0 Citrix ADC BLX クラスターの一部の Citrix ADC 機能のサポート性のステータスは、13.0 Citrix ADC 以外の BLX（MPX、または VPX、SDX ADC）クラスタとは異なります。

> 重要
>
> 表の「Node-level」エントリは、機能が個々のクラスターノードでのみサポートされていることを示します。

Citrix ADC の機能	11.1	12.1	13.0	13.0 Citrix ADC BLX クラスター
SSL FIPS	いいえ	いいえ	いいえ	いいえ
SSL 証明書バンドル	いいえ	いいえ	いいえ	いいえ
SSL インターセプション	-	いいえ	いいえ	いいえ
コンテンツ切り替えアクション	はい	はい	はい	はい
コンテンツスイッチングポリシー用のポリシーベースのロギング	はい	はい	はい	はい
レート制限	はい	はい	はい	はい
アクション分析	はい	はい	はい	いいえ
GSLB か	はい	はい	はい	はい

Citrix ADC の機能	11.1	12.1	13.0	13.0 Citrix ADC BLX クラスター
RTSP	はい	はい	はい	はい
DNSSEC	いいえ	いいえ	いいえ	いいえ
DNS64	いいえ	いいえ	いいえ	いいえ
FTP	はい	はい	はい	いいえ
TFTP	いいえ	はい	はい	はい
接続ミラーリング	いいえ	いいえ	いいえ	いいえ
統合されたキャッシング	Node-Level	Node-Level	Node-Level	いいえ
大容量共有キャッシュ	Node-Level	Node-Level	Node-Level	いいえ
フロントエンドの最適化	Node-Level	Node-Level	Node-Level	いいえ
Application firewall	はい	はい	はい	いいえ
HTTP サービス拒否保護 (HDOSP)	Node-Level	Node-Level	Node-Level	廃止済み
プライオリティキューイング（PQ）	Node-Level	Node-Level	Node-Level	廃止済み
Sure connect (SC)	Node-Level	Node-Level	Node-Level	廃止済み
AppQoE	はい	はい	はい	いいえ
サージ保護	Node-Level	Node-Level	Node-Level	はい
MPTCP	はい	はい	はい	いいえ
ストライピングされた SNIP	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。

Citrix ADC の機能	11.1	12.1	13.0	13.0 Citrix ADC BLX クラスター
MSR	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。
IS-IS（IPv4 および IPv6）	はい	はい	はい	いいえ
ジャンボフレーム	はい	はい	はい	いいえ
IP-IP トンネリング	はい	はい	はい	いいえ
リンク負荷分散	はい	はい	はい	はい
FIS（フェールオーバーインターフェイスセット）	はい	はい	はい	いいえ
リンク冗長性（LR）	はい	はい	はい	いいえ
NAT46	いいえ	いいえ	はい	はい
NAT64	いいえ	いいえ	はい	はい
RNAT6	はい	はい	はい	はい
LSN/CGNAT	いいえ	はい	はい	いいえ
IPv6 レディロゴ	いいえ	はい	はい	いいえ
トラフィックドメイン	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	いいえ
ルートモニタ	はい。DR を使用する場合のみ。	はい。注**:** L2 クラスターでサポートされています。L3 クラスタではサポートされません。	はい 注**:** L2 クラスタでサポートされています。L3 クラスタではサポートされません。	いいえ
GRE トンネリング（CB）	いいえ	いいえ	いいえ	いいえ
レイヤ 2 モード	はい	はい	はい	いいえ

Citrix ADC の機能	11.1	12.1	13.0	13.0 Citrix ADC BLX クラスター
ネットプロファイル	はい	はい	はい	いいえ
HTTPS コールアウト	はい	はい	はい	はい
AAA-TM	Node-level	はい	はい	いいえ
AppFlow	Node-level	Node-Level	Node-Level	いいえ
Web Insight	はい	はい	はい	いいえ
HDX Insight	はい	はい	はい	いいえ
VMAC/VRRP	はい	はい	はい	いいえ
NetScaler Push	いいえ	いいえ	いいえ	いいえ
ステートフル接続のフェールオーバー	いいえ	いいえ	いいえ	いいえ
グレースフルシャットダウン	いいえ	はい	はい	はい
DBS オートスケール	いいえ	いいえ	はい	はい
TOS を使用した DSR	いいえ	いいえ	いいえ	はい
より細かいスタートアップ RR 制御	Node-Level	Node-Level	Node-Level	いいえ
XML XSM	いいえ	いいえ	いいえ	いいえ
DHCP RA	いいえ	いいえ	いいえ	いいえ
ブリッジグループ	はい	はい	はい	いいえ
ネットワークブリッジ	いいえ	いいえ	いいえ	いいえ
Citrix ADC (WIonNS) 上の Web インターフェイス	はい	はい	はい	いいえ
EdgeSight Monitoring	廃止済み	廃止済み	廃止済み	いいえ

Citrix ADC の機能	11.1	12.1	13.0	13.0 Citrix ADC BLX クラスター
メトリックス表-ローカル	いいえ	いいえ	いいえ	いいえ
DNS キャッシュ	Node-Level	Node-Level	Node-Level	Node-Level
Call Home	Node-Level	Node-Level	Node-Level	いいえ
Citrix Gateway ICA プロキシモード	はい	はい	はい	いいえ
Citrix Gateway (SSL VPN /フル VPN およびクライアントレス VPN)	Node-Level	Node-Level	Node-Level	いいえ
Citrix CloudBridge Connector	いいえ	はい	はい	いいえ
ポリシーベースのルーティング (PBR/PBR6)	はい	はい	はい	いいえ
LLB 仮想サーバをネクストホップとする IPv4 ポリシーベースルーティング（PBR）	いいえ	いいえ	はい	いいえ
LLB 仮想サーバをネクストホップとする IPv6 ポリシーベースルーティング（PBR6）	いいえ	いいえ	いいえ	いいえ
加入者認識	いいえ	いいえ	いいえ	いいえ
動的ルーティング	はい、v6 プロトコル（ospfv3、RIPng、ISIS6、BGP6）のサポート	はい、v6 プロトコル（ospfv3、RIPng、ISIS6、BGP6）のサポート	はい、v6 プロトコル（ospfv3、RIPng、ISIS6、BGP6）のサポート	はい

Citrix ADC の機能	11.1	12.1	13.0	13.0 Citrix ADC BLX クラスター
SYSLOG-TCP、Syslog サーバの負荷分散、SNIP サポート、および Syslog の FQDN サポート	はい。注: NetScaler 11.1 ビルド 54.16 以降でサポートされています。	はい	はい	はい
ボットの管理	いいえ	いいえ	はい	いいえ
VXLAN	いいえ	いいえ	いいえ	いいえ

また、次の Citrix ADC 構成がサポートされています。

負荷分散、負荷分散の永続性、DNS 負荷分散、SIP、maxClient、波及効果（接続および動的）。帯域幅、DataStream、圧縮制御、コンテンツフィルタリング、TCP バッファリング、キャッシュリダイレクト、分散型サービス拒否（DDoS）に基づく波及効果。クライアントキープアライブ、基本ネットワーキング（IPv4 および IPv6）、OSPF（IPv4 および IPv6）、RIP（IPv4 および IPv6）、RIP（IPv4 および IPv6）。VLAN、ICMP、フラグメンテーション、MBF、ACL、簡易 ACL、MSR、パス MTU 検出、IP-IP、SNMP、ポリシー（クラシックおよびアドバンスト）。リライト、レスポンダー、HTTP コールアウト、Web サーバーのログ、監査ログ（NSLOG および syslog）。USIP、ロケーションコマンド、HTML インジェクション、NITRO API、AppExpert、KRPC。

前提条件

April 7, 2022

クラスタに追加する Citrix ADC アプライアンス（MPX、VPX、SDX ADC、BLX）は、次の前提条件を満たしている必要があります。

- すべてのアプライアンスのソフトウェアバージョンとビルドが同じである必要があります。
- すべてのアプライアンスは、同じプラットフォームタイプである必要があります。つまり、クラスタには、すべてのハードウェアアプライアンス（Citrix ADC MPX）またはすべての Citrix ADC VPX アプライアンス、またはすべての Citrix BLX アプライアンス、またはすべての Citrix SDX ADC インスタンスが必要です。

  注:

  - ハードウェアアプライアンス（MPX）のクラスタの場合、アプライアンスは同じモデルタイプである必要があります。
  - 異機種クラスタを構成するには、すべてのアプライアンスが MPX プラットフォーム・タイプである必要があります。

  - 仮想アプライアンスのクラスタ（VPX）の場合、アプライアンスは XenServer、Hyper-V、VMware ESX、KVM のハイパーバイザーに展開する必要があります。
  - SDX Citrix ADC インスタンスのクラスターの設定については、Citrix ADC インスタンスのクラスターの設定を参照してください。
  - ジャンボフレームは、Citrix ADC SDX インスタンスで構成される Citrix ADC クラスターでサポートされます。
  - SDX インスタンスの L3 クラスタを作成できます。
  - Citrix ADC BLX クラスターの設定の詳細については、Citrix ADC BLX クラスターを参照してください。
- アプライアンスは異なるネットワークに属することができます。
- 最初に構成され、共通のクライアント側およびサーバー側のネットワークに接続されている。
- 大規模な構成を持つ仮想アプライアンス（Citrix ADC VPX、Citrix ADC BLX、または Citrix SDX ADC インスタンス）のクラスタでは、クラスタの各ノードに 6 GB RAM を使用することをお勧めします。

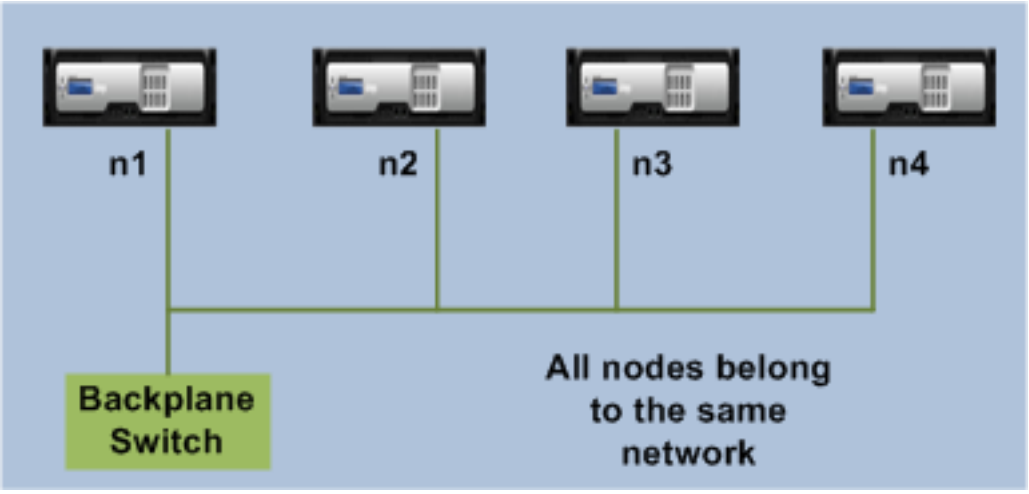
クラスタの概要

October 7, 2021

Citrix ADC クラスタは、Citrix ADC アプライアンスをグループ化することによって形成されます。クラスターを追加する予定の Citrix ADC アプライアンスのネットワーク上の場所に基づいて、次のクラスター設定に注意する必要があります。

> 注
>
> 特に指定がない限り、クラスタの機能と設定は L2 クラスタと L3 クラスタで同じになります。

- **L2** クラスター**:** このクラスター展開では、すべてのクラスターノードが同じネットワークに属します。

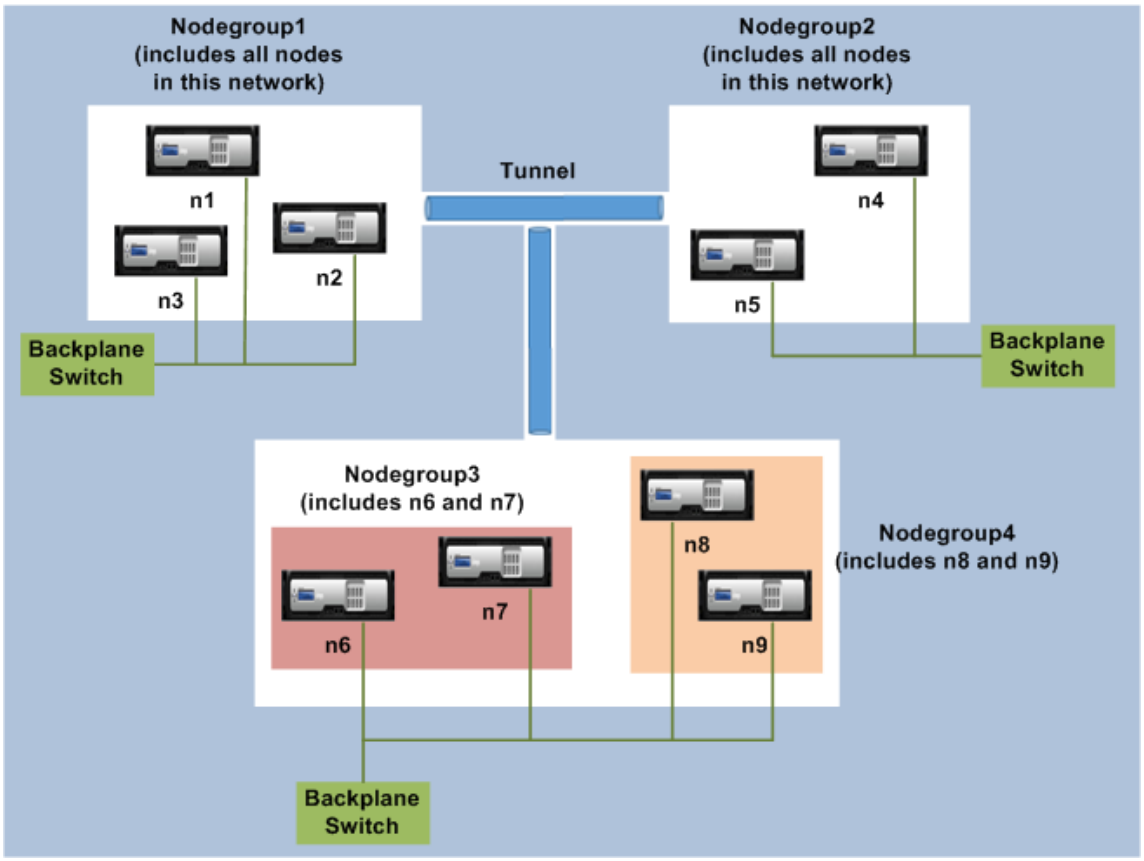


- **L3** クラスタ（「**INC** モードのクラスタ」とも呼ばれます）: このクラスタ展開では、クラスタノードは異なるネットワークに属することができます。特定のネットワークのクラスターノードは、そのネットワークのノー

ドのみを含むノードグループにグループ化する必要があります。次の図から、ノード n1、n2、n3 は同じネットワーク内にあり、Nodegroup1 にグループ化されていることがわかります。

同様に、ノード n4 と n5 の場合、ノードグループ 2 にグループ化されています。3 番目のネットワークには、2 つのノードグループがあります。ノードグループ 3 には n6 と n7 が含まれており、ノードグループ 4 には n8 と n9 が含まれています。

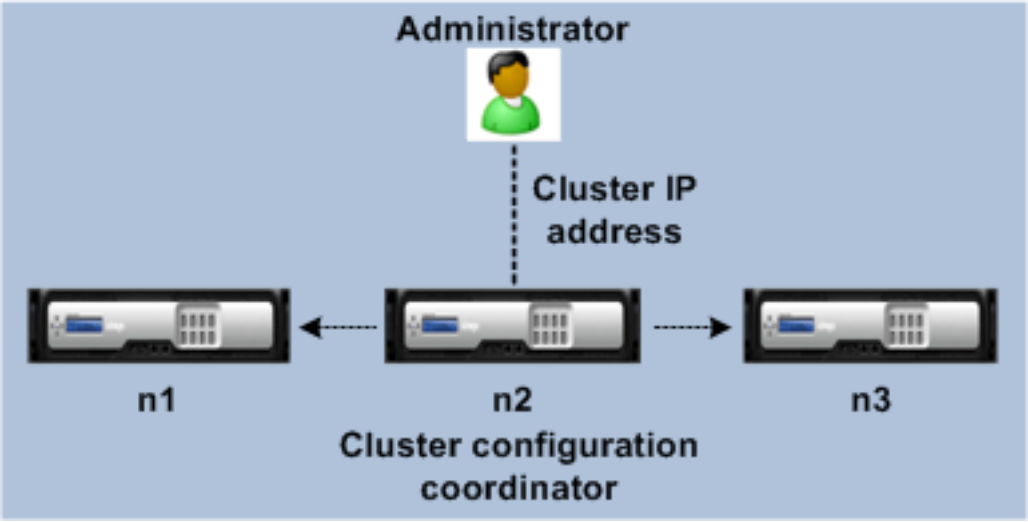
> 注
>
> NetScaler 11.0 以降でサポートされています。



クラスタ・ノード間の同期

October 7, 2021

Citrix ADC クラスタ上のすべての構成は、クラスタの IP アドレス（クラスタの管理アドレス）で実行されます。次の図に示すように、クラスターノードはクラスター構成コーディネーター（CCO）と呼ばれるクラスター IP アドレスを所有します。



CCO で使用可能な構成は、他のクラスターノードに自動的に伝播されるため、すべてのクラスターノードの構成は同じです。

- Citrix ADC では、NSIP アドレスを通じて個々のクラスタノード上で実行できる構成はごくわずかです。このような場合は、クラスタ内のすべてのノードで構成の一貫性を手動で確保する必要があります。これらの構成は、他のクラスタノードに伝播されません。各クラスタノードでサポートされる操作の詳細については、「個々のクラスタノードでサポートされる操作」を参照してください。
- クラスター IP アドレスで実行された場合、次のコマンドは他のクラスターノードに伝達されません。
  - **shutdown.** 構成コーディネータのみをシャットダウンします。
  - **reboot.** 構成コーディネータだけをリブートします。
  - **rm cluster instance.** コマンドを実行しているノードからクラスターインスタンスを削除します。
- 他のクラスタ・ノードに伝播するコマンドの場合:
  - クォーラムは、クラスターインスタンスで構成する必要があります。
  - ほとんどのクラスタークォーラム (n/2 + 1) クラスターが動作可能になるには、クラスターノードの 1 つがアクティブである必要があります。
  - 多数決により、クラスターは最小数のノードで実行できます (n/2 + 1) リラックスしている。

ノードがクラスターに追加されると、CCO で使用可能な構成とファイル（SSL 証明書、ライセンス、DNS など）が、新しく追加されたクラスターノードに同期されます。意図的に無効にされた、または障害が発生した既存のクラスターノードが再度追加されると、クラスターはノードで使用可能な構成を CCO で使用可能な構成と比較します。構成に不一致がある場合、ノードは次のいずれかを使用して同期されます。

- 完全同期。構成間の差異が 255 コマンドを超える場合、CCO のすべての構成が、クラスターに再参加しているノードに適用されます。同期中、ノードは操作上使用できません。
- 差分同期。構成間の差が 255 コマンド以下の場合は、使用できない構成のみがクラスタに再参加するノードに適用されます。ノードの動作状態は影響を受けません。

> 注
>
> 構成とファイルを手動で同期することもできます。詳細については、「クラスタ構成の同期」および「[クラスタ

> ファイルの同期](/ja-jp/citrix-adc/13/clustering/cluster-managing/cluster-file-sync.html)」を参照してください。

ストライプ、部分的にストライプ、およびスポットされた構成

October 7, 2021

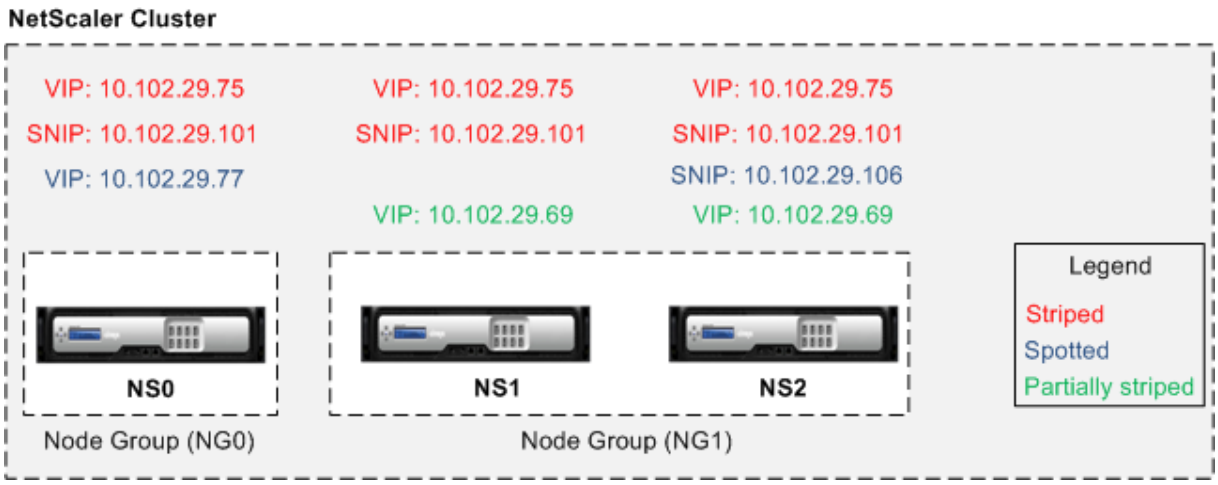
コマンド伝播により、クラスタ内のすべてのノードが同じ構成になります。ただし、一部の構成は、特定のクラスターノードでのみ使用できるようにしたい場合があります。構成が使用可能なノードを制限することはできませんが、構成がアクティブなノードを指定することはできます。

たとえば、次の作業を行えます。

- SNIP アドレスを 1 つのノードでのみアクティブにするように定義する。
- すべてのノードでアクティブにする SNIP アドレスを定義する。
- VIP アドレスを 1 つのノードでのみアクティブにすることを定義する。
- すべてのノードでアクティブな VIP アドレスを定義する。
- 3 ノードクラスタの 2 つのノードでのみアクティブになるように VIP アドレスを定義する

構成がアクティブになっているノードの数に応じて、クラスター構成は、ストライプ構成、部分ストライプ構成、またはスポット構成と呼ばれます。

図 1: ストライプ、部分ストライプ、およびスポット構成を持つ 3 ノードクラスタ



次の表に、構成のタイプの詳細を示します。

構成の種類	アクティブ・オン	適用対象	構成
ストライプ構成	すべてのクラスタノード	すべてのエントリ	エンティティをストライプ化するために特定の構成は必要ありません。デフォルトでは、クラスター IP アドレスに定義されているすべてのエンティティは、すべてのクラスターノードでストライプされます。
部分的にストライプされた構成	クラスタノードのサブセット	クラスタノードグループを参照してください。	部分的にストライプするエンティティをノードグループにバインドします。構成は、ノードグループに属するクラスタノード上でのみアクティブになります。

構成の種類	アクティブ・オン	適用対象	構成
スポッティング	単一クラスタ・ノード	SNIP アドレス、SNMP エンジン ID、クラスタノードのホスト名、ノードグループにバインドできるエンティティ	スポット設定を定義するには、2 つの方法のいずれかを使用します。**SNIP** アドレス SNIP アドレスを作成するときは、SNIP アドレスをアクティブにするノードを所有者ノードとして指定します。例、`add ns ip 10.102.29.106 255.255.255.0 -type SNIP -ownerNode 2`（ノード NS2 ID が 2 であると仮定）。注: スポットされた SNIP アドレスの所有権は、実行時に変更できません。所有権を変更するには、最初に SNIP アドレスを削除し、新しい所有者を指定して再度追加する必要があります。ノードグループにバインドできるエンティティ。エンティティを単一メンバーノードグループにバインドします。

注

- USIP を無効にする場合は、スポットの SNIP アドレスを使用することをお勧めします。ストライプ SNIP アドレスを使用できるのは、IP アドレスが不足している場合だけです。ストライプ IP アドレスを使用すると、ARP 解決のために同じサブネットにスポッティング IP アドレスが存在しない場合、ARP フラックスの問題が発生する可能性があります。
- USIP を有効にする場合、ストライピングされた SNIP アドレスは、サーバーが開始するトラフィックのゲートウェイとして使用することをお勧めします。

ストライプ **IP** に対する **ARP** オーナーのサポート

クラスタのセットアップでは、ストライピング IP の ARP 要求に応答するように特定のノードを設定できます。設定されたノードは ARP トラフィックに応答します。

新しいパラメータ「arpOwner」が「IP の追加、設定、および設定解除」コマンドに導入されました。

CLI を使用して、ノードで ARP 所有者を有効にします。

コマンドプロンプトで入力します。

```
add ns ip <ip_address> -arpOwner <node_id>
```

> 注
>
> ARP 所有者パラメーターは、L2 クラスターでのみサポートされます。

ストライプされた **IPv6** アドレスに対するネイバー探索所有者のサポート

クラスタのセットアップでは、特定のノードをストライプされた IPv6 アドレスのネイバー検出（ND）所有者として構成して、リンク層アドレスを決定できます。クライアントは、クラスタのセットアップ内のすべてのノードにネイバー送信請信（NS）メッセージを送信します。ND 所有者は、ストライプ IPv6 アドレスのリンク層アドレスを持つネイバーアドバタイズメント（NA）メッセージで応答し、トラフィックを処理します。

CLI を使用してノードで **ND** 所有者を有効にするには

コマンドプロンプトで入力します。

```
add ns ip6 <IPv6Address> -ndOwner <node id>

set  ns ip6 <IPv6Address> -ndOwner <node id>
<!--NeedCopy-->
```

例:

```
add ns ip6 2001::21/64 -ndOwner 1

set ns ip6 2001::21/64 -ndOwner 1
<!--NeedCopy-->
```

GUI を使用してノードで **ND** 所有者を有効にするには

1. [システム] **>** [ネットワーク] **>** [**IP**] に移動します。

2. [**IP**] ページで、[**IPv6**] タブに移動し、[追加] をクリックします。
3. [**IPv6** の作成] ページで、[クラスタ**]** ドロップダウンメニューの **[NDowner**] に一覧表示されているノード ID のいずれかを選択します。

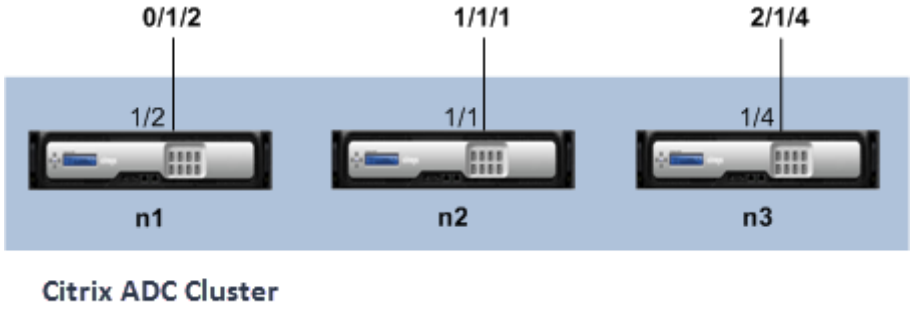
> 注
>
> ND 所有者パラメーターは、L2 クラスターでのみサポートされます。

クラスタ設定での通信

October 7, 2021

クラスターに追加される Citrix ADC アプライアンスのインターフェースには、ノード ID の接頭辞が付きます。これは、インターフェースが属するクラスターノードを識別するのに役立ちます。したがって、インターフェイス識別子 c/u（c は Controller 番号、u はユニット番号）が n/c/u になります（n はノード ID)。たとえば、次の図では、ノード n1 のインタフェース 1/2 は 0/1/2、ノード n2 のインタフェース 1/1 は 1/1/1、ノード n3 のインタフェース 1/4 は 2/1/4 と表されます。

図 1: クラスタ内のインタフェース命名規則



- サーバー通信-
  クラスタは、クラスタノードとサーバー側の接続デバイス間の物理接続を介してサーバーと通信します。これらの物理接続の論理的なグループ化は、サーバデータプレーンと呼ばれます。
- クライアント通信-クラスタは、クラスタノードとクライアント側の接続デバイス間の物理接続を介してクライアントと通信します。これらの物理接続の論理的なグループ化は、クライアントデータプレーンと呼ばれます。
- ノード間通信**-**クラスタノードは、相互に通信することもできます。これらの通信方法は、ノードが同じネットワーク上に存在するか、ネットワーク上に存在するかによって異なります。
  - 同じネットワーク内のクラスタノードは、クラスタバックプレーンを使用して相互に通信します。バックプレーンは、各ノードの 1 つのインターフェイスが共通のスイッチ（クラスタバックプレーンスイッチ）に接続されている一連のインターフェイスです。ノード間通信で使用される、バックプレーンを通過するさまざまなタイプのトラフィックは次のとおりです。
    * ノードからノードへのメッセージング（NNM）
    * トラフィックステアリング

    * 構成の伝播と同期
  - クラスタの各ノードは、特別な MAC クラスタバックプレーンスイッチアドレスを使用して、バックプレーンを介して他のノードと通信します。クラスター特殊 MAC の形式は次のとおりです。`0x02 0x00 0x6F <cluster_id> <node_id> <reserved>`)、`cluster_id` はクラスターインスタンス ID、`node_id` はクラスターに追加される Citrix ADC アプライアンスのノード番号です。

次の図は、L2 クラスタと L3 クラスタの通信インターフェイスを示しています。

図 2: クラスタ通信インタフェース-L2 クラスタ

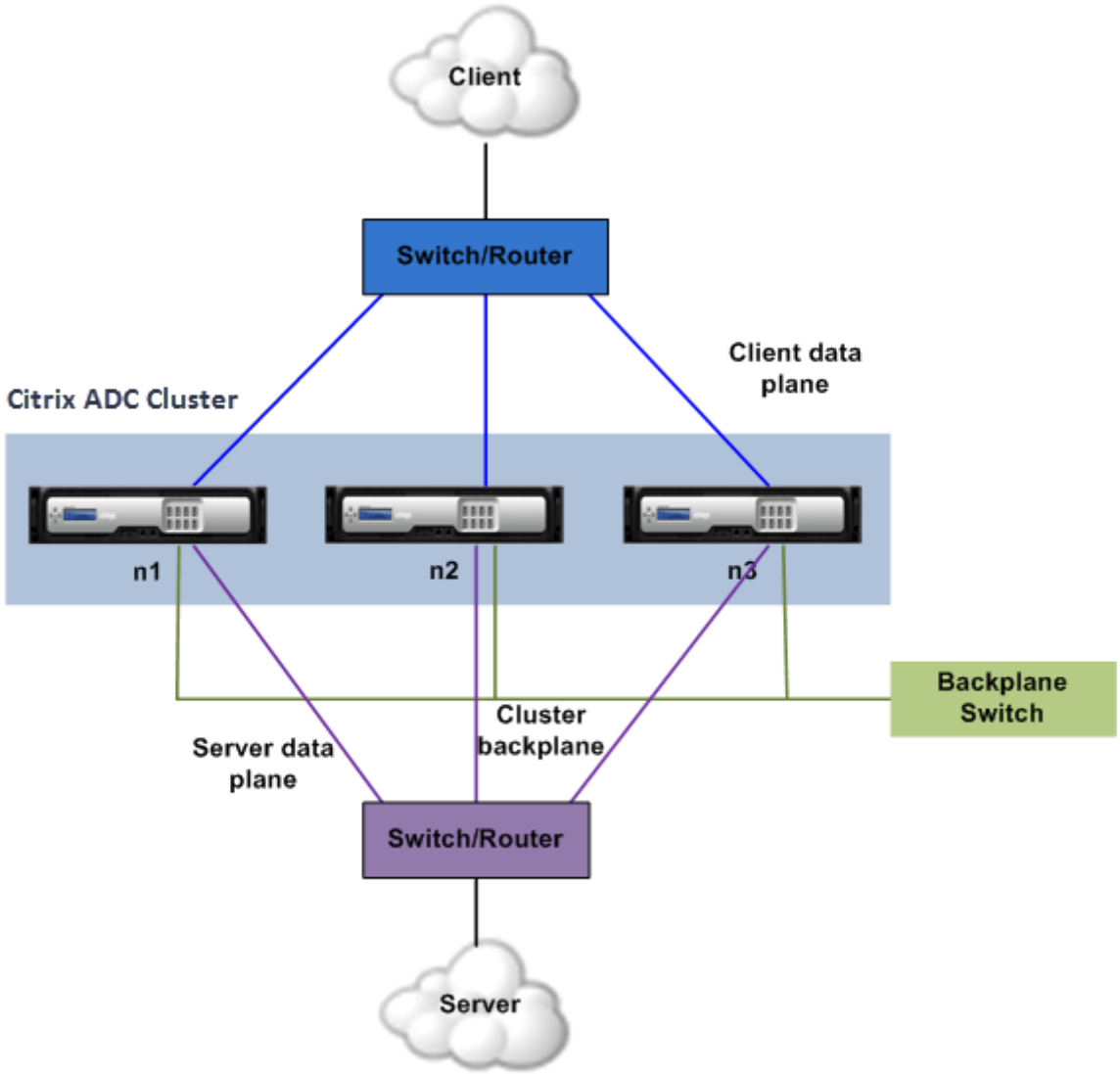
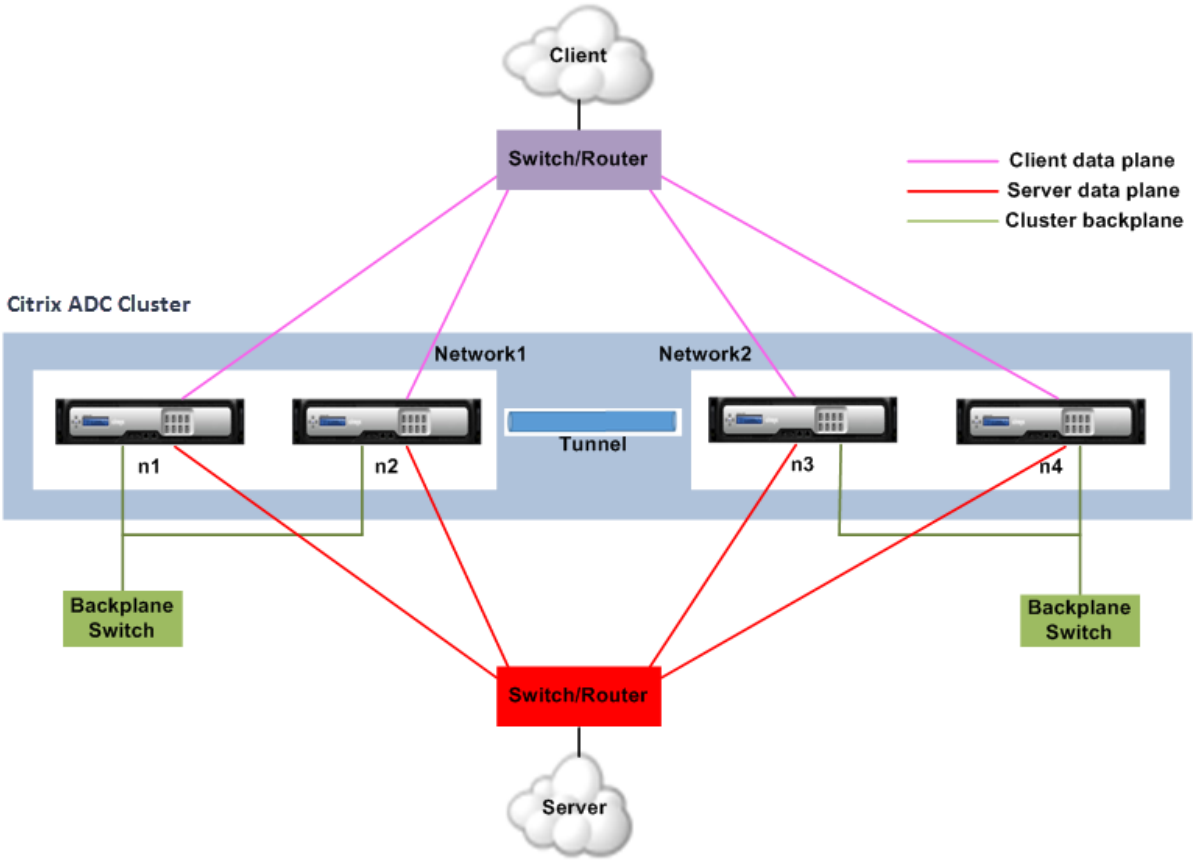


図 3: クラスタ通信インタフェース-L3 クラスタ



クラスタ設定でのトラフィック分散

October 7, 2021

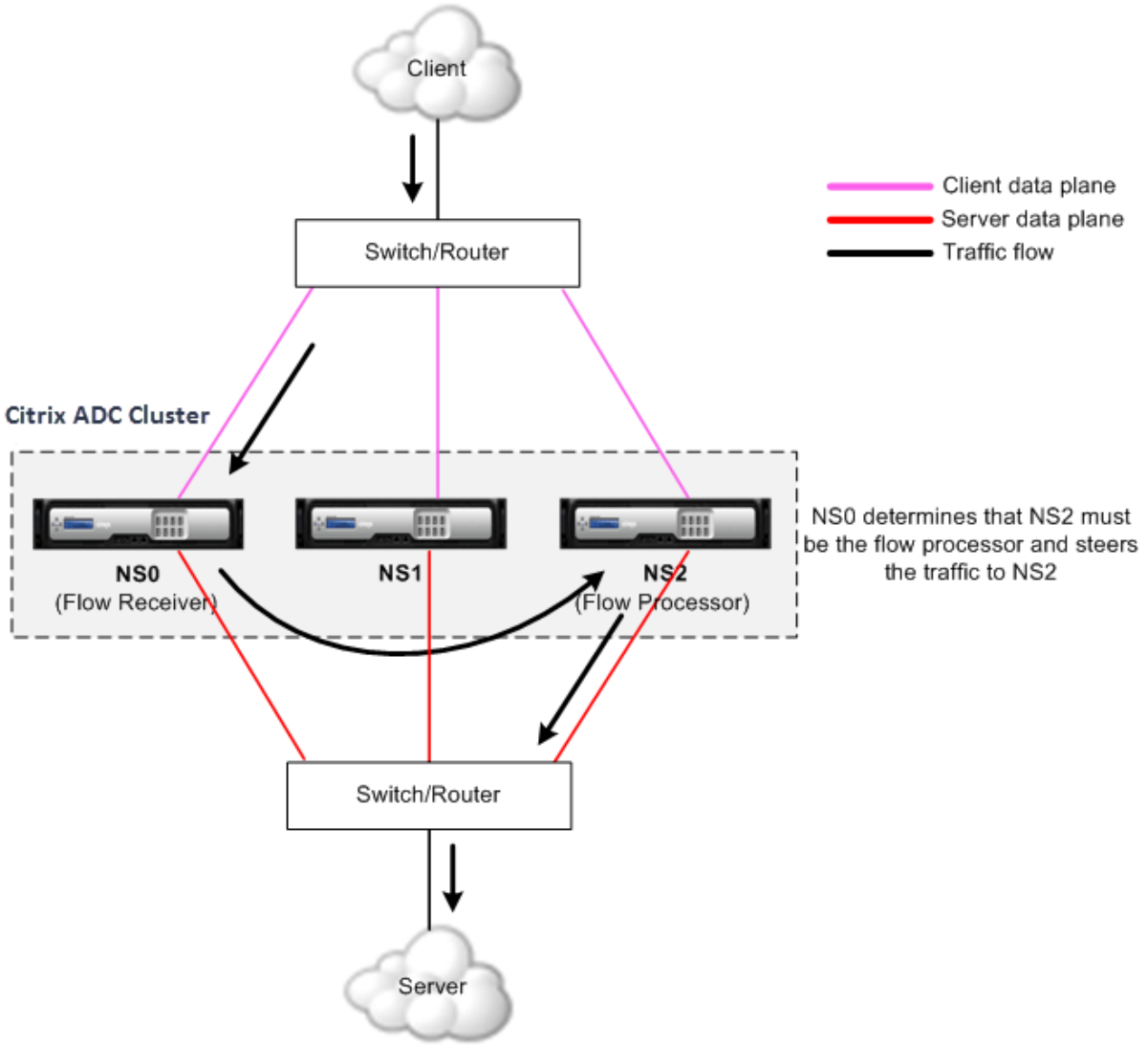
クラスター設定では、外部ネットワークは Citrix ADC アプライアンスのコレクションを単一のエンティティとして表示します。したがって、クラスターは、トラフィックを受信する必要がある単一のノードを選択する必要があります。クラスターは、Equal Cost Multiple Path（ECMP）またはクラスターリンクアグリゲーショントラフィック分散メカニズムを使用してこの選択を行います。選択されたノードをフロー受信機と呼びます。

> 注
>
> L3 クラスター（異なるネットワークにまたがるノード）の場合、ECMP トラフィック分散のみを使用できます。

フローレシーバはトラフィックを取得し、内部クラスタロジックを使用してトラフィックを処理する必要があるノードを決定します。このノードをフロープロセッサと呼びます。フローレシーバーとフロープロセッサーが同じネットワーク上にある場合、フローレシーバーはバックプレーンを介してトラフィックをフロープロセッサーに誘導します。フローレシーバーとフロープロセッサーが異なるネットワーク上にある場合、トラフィックはトンネルを介してステアリングされます。

> 注
>
> - フローレシーバとフロープロセッサは、トラフィックを処理できるノードである必要があります。
> - NetScaler 11 以降では、クラスタバックプレーンでステアリングを無効にできます。詳細については、クラスタバックプレーンでのステアリングの無効化を参照してください。

図 1: クラスタ内のトラフィック分散



前の図は、クラスターを流れるクライアント要求を示しています。クライアントは、仮想 IP（VIP）アドレスに要求を送信します。クライアントデータプレーンに設定されたトラフィック分散メカニズムは、クラスタノードの 1 つをフローレシーバとして選択します。フローレシーバは、トラフィックを受信し、トラフィックを処理する必要があるノードを決定し、そのノードに要求を操縦します（フローレシーバが自身をフロープロセッサとして選択しない限り）。

フロープロセッサは、サーバとの接続を確立します。サーバは要求を処理し、要求をサーバに送信したサブネット IP（SNIP）アドレスに応答を送信します。

- SNIP アドレスがストライピングされた IP アドレスまたは部分的にストライピングされた IP アドレスである場合、サーバデータプレーンに設定されたトラフィック分散メカニズムによって、クラスタノードの 1 つがフローレシーバとして選択されます。フローレシーバは、トラフィックを受信し、フロープロセッサを決定し、クラスタバックプレーンを介してフロープロセッサに要求を誘導します。
- SNIP アドレスがスポッティング IP アドレスの場合、SNIP アドレスを所有するノードはサーバからの応答を受信します。

非対称クラスタトポロジ（すべてのクラスタノードが外部スイッチに接続されていない）では、リンクセットを排他的に使用するか、ECMP またはクラスタリンク集約と組み合わせて使用する必要があります。詳細については、「リンクセットの使用」を参照してください。

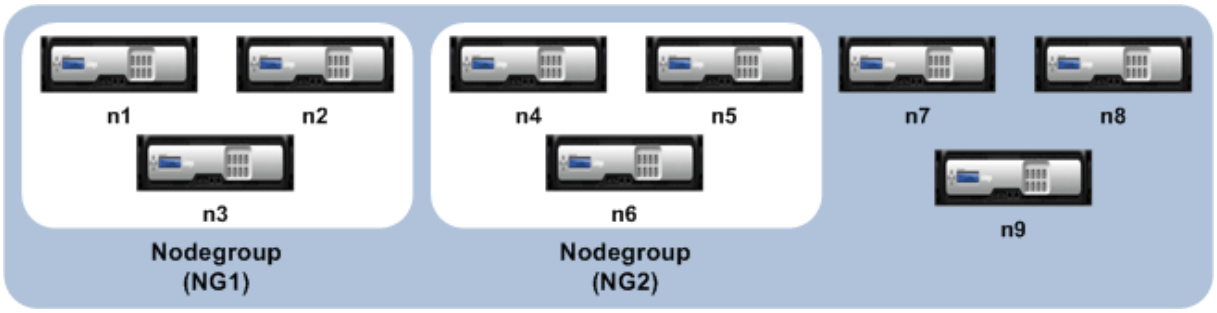
クラスターノードグループ

October 7, 2021

> 注
>
> ノードグループは、NetScaler10.1 以降でサポートされています。

名前が示すように、クラスターノードグループはクラスターノードのグループです。

図 1: ノードグループを備えた Citrix ADC クラスター



前の図は、それぞれ 3 つのクラスターノードを含むノードグループ NG1 と NG2 を持つクラスターを示しています。クラスターには、ノードグループの一部ではない 3 つのノードもあります。

ノードグループは、次のように構成できます。

- スポットおよび部分的にストライプされた構成を定義する。詳細については、「スポッティングされた構成と部分ストライプ構成のノードグループ」を参照してください。
- ノードグループの冗長性を構成します。詳細については、「ノードグループの冗長性の設定」を参照してください。
  注：NetScaler 10.5 ビルド 52.1115.e 以降でサポートされています。
- L3 クラスタ（INC モードではクラスタとも呼ばれる）を定義する。L3 クラスタでは、クラスタノードは異なるネットワークから配置できます。ネットワークに属するノードを単一のノードグループにグループ化する必要があります。たとえば、n1、n2、n3 がネットワーク 1 にあり、n4、n5、n6 がネットワーク 2 にある場合、

NG1 はネットワーク 1 のノードを含み、NG2 はネットワーク 2 のノードを含める必要があります。L3 クラスターの設定については、Citrix ADC クラスターの作成を参照してください。

> 注
>
> - NetScaler 11 以降でサポートされています。
> - ノードグループの前述の機能は相互に排他的です。これは、ノードグループが前述の機能の 1 つのみを提供できることを意味します。

クラスタとノードの状態

October 7, 2021

クラスターが機能するためには、ほとんどのノードが (n/2 + 1) 動作可能である必要があります（動作状態は ACTIVE です)。

> 重要
>
> NetScaler リリース 10.5 からは、過半数の基準が満たされていない場合でもクラスタが機能するように構成できます。この構成は、クラスターの作成時に実行する必要があります。

クラスタノードの状態の詳細については、「クラスタノードの状態」を参照してください。

クラスタ内のルーティング

October 7, 2021

クラスター内のルーティングは、スタンドアロンシステムでのルーティングとほぼ同じように機能します。注意すべきいくつかのポイント:

- すべてのルーティング構成はクラスタ IP アドレスから実行する必要があり、構成は他のクラスタノードに伝播されます。
- ルートは、アップストリームルータでサポートされる ECMP ルートの最大数に制限されます。
- ノード固有のルーティング設定は、次のように owner-node 引数を使用して実行する必要があります。

```
1   router ospf
2     owner-node 0
3       ospf router-id 97.131.0.1
4     exit-owner-node
5   !
6 <!--NeedCopy-->
```


次のコマンドは、VTYSH 内のすべてのノードの統合クラスタ構成を表示します。

```
show cluster-config
```

次のコマンドは、各ノードのクラスターステータスを表示します。

```
show cluser node
```

L2 クラスタでの IPv4 ルーティング

次のセクションには、L2 クラスターで IPv4OSPF および BGP ルーティングを構成するのに役立つ構成例が含まれています。

スポッティング SNIP アドレスの追加と動的ルーティングの有効化

次の設定では、OSPF および BGP ルーティングが有効になっています。また、スポッティング SNIP アドレスが追加され、これらの SNIP アドレスでダイナミックルーティングが有効になります。

```
en ns fea ospf bgp
add vlan 10
add ns ip 10.10.10.1 255.255.255.0 -dynamicrouting enabled -ownernode 1
add ns ip 10.10.10.2 255.255.255.0 -dynamicrouting enabled -ownernode 2
add ns ip 10.10.10.3 255.255.255.0 -dynamicrouting enabled -ownernode 3
bind vlan 10 -ipaddress 10.10.10.1 255.255.255.0
<!--NeedCopy-->
```

VTYSH IPv4 OSPF 構成

L2 クラスターで IPv4OSPF を構成するには、次のことを行う必要があります。

- プライオリティを 0 に設定します。
- ルータ ID をスポット設定として設定します。

> 注
>
> L2 クラスターの OSPF 構成ガイドラインは、OSPFv3 にも適用できます。

次の設定例では、IPv4 OSPF が設定されています。

```
       interface vlan10
        IP OSPF PRIORITY 0
        !
       router ospf
```



```
5            owner-node 1
6              ospf router-id 97.131.0.1
7            exit-owner-node
8            owner-node 2
9              ospf router-id 97.131.0.2
10           exit-owner-node
11           owner-node 3
12             ospf router-id 97.131.0.3
13           exit-owner-node
14       network 10.10.10.0/24 area 0
15       redistribute kernel
16        !
17 <!--NeedCopy-->
```

VTYSH IPv4 BGP 構成

次の VTYSH 設定例では、IPv4 BGP が設定されています。

```
1      router bgp 100
2        neighbor 10.10.10.10 remote-as 200
3       owner-node 1
4        neighbor 10.10.10.10 update-source 10.10.10.1
5       exit-owner-node
6       owner-node 2
7        neighbor 10.10.10.10 update-source 10.10.10.2
8       exit-owner-node
9       owner-node 3
10       neighbor 10.10.10.10 update-source 10.10.10.3
11      exit-owner-node
12     redistribute kernel
13     !
14 <!--NeedCopy-->
```

> 注
>
> 次の設定では、owner-node 引数を使用してネイバーごとに update-source コマンドを使用して、適切な送信元 IP に接続します。

L2 クラスタでの IPv6 ルーティング

次のセクションには、L2 クラスターで IPv6OSPF および BGP ルーティングを構成するのに役立つ構成例が含まれています。

IPv6 ルーティングを有効にする

L2 クラスタで IPv6 ルーティングを設定する前に、IPv6 機能を有効にする必要があります。

CLI を使用して IPv6 ルーティングを有効にするには、

コマンドプロンプトで入力します。

- `enable ns fea ipv6pt`

スポッティング **SNIP6** アドレスの追加と動的ルーティングの有効化

次の設定では、OSPF および BGP ルーティングが有効になっています。また、スポッティング SNIP6 アドレスが追加され、これらの SNIP6 アドレスでダイナミックルーティングが有効になります。

```
add ns ip6 3ffa::1/64 -dynamicrouting enabled -ownernode 1
add ns ip6 3ffa::2/64 -dynamicrouting enabled -ownernode 2
add ns ip6 3ffa::3/64 -dynamicrouting enabled -ownernode 3
add vlan 10
bind vlan 10 -ipaddress 3ffa::1/64
<!--NeedCopy-->
```

VTYSH IPv6 BGP 構成

次の VTYSH 設定例では、IPv6 BGP が設定されています。

```
  router bgp 100
   neighbor 3ffa::10 remote-as 200
     owner-node 1
      neighbor 3ffa::10 update-source 3ffa::1
     exit-owner-node
     owner-node-2
      neighbor 3ffa::10 update-source 3ffa::2
     exit-owner-node
     owner-node-3
      neighbor 3ffa::10 update-source 3ffa::3
     exit-owner-node
   no neighbor 3ffa::10 activate
   address-family ipv6
    redistribute kernel
    neighbor 3ffa::10 activate
   exit-address-family
```



```
17     !
18 <!--NeedCopy-->
```

IPv6 学習ルートのインストール

Citrix ADC クラスタは、Citrix ADC クラスタのルーティングテーブルにルートをインストールした後、さまざまなルーティングプロトコルで学習したルートを使用できます。

CLI を使用して IPv6 学習ルートを内部ルーティングテーブルにインストールするには、次の手順を実行します。

コマンドプロンプトで入力します。

- `ns route-install ipv6 bgp`
- `ns route-install ipv6 ospf`
- `ns route-install default`

> 注
>
> - IPv6 ネイバー上で IPv4 ルートを交換する必要がある場合は、以前の設定から`no neighbor 3ffa::10 active` VTYSH コマンドを削除する必要があります。
> - `update-source` VTYSH コマンドを各所有者ノードに使用して、BGPIPv4 構成で指定されている BGP ピアへの接続中に適切な IPv6 送信元 IP を指定する必要があります。

L3 クラスタでのルーティング

L3 クラスターでのルーティングは、Citrix ADC アプライアンスで次の構成が行われた場合にのみ機能します。

- VLAN のダイナミックルーティングを有効にします。

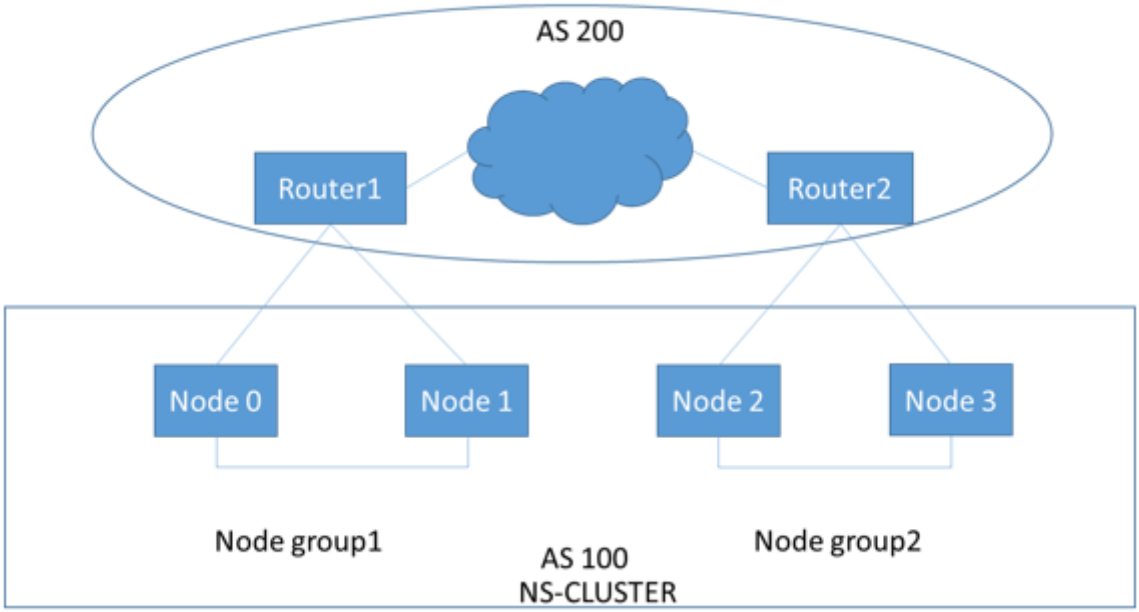
```
1 set vlan <id> -dynamicrouting enabled
2 <!--NeedCopy-->
```

- すべてのクラスターノードに到達するには、VIP、CLIP、および Citrix ADC IP（NSIP）を、`set vlan` コマンドとともにルーティングプロトコルによってアドバタイズする必要があります。

L3 クラスタでの **BGP** の展開シナリオ

すべてのクラスターノードが AS100 ネットワークにグループ化され、アップストリームルーターが別の AS200 にある例を考えてみます。

次の図は、クラスタ設定における AS 100 および AS 200 の配置を示しています。



この展開では、CLIP はアップストリームルータに CCO をアドバタイズします。一部のクラスターノードは、AS ループが検出されると、アドバタイズされたトラフィックをドロップします。

この問題を解決するには、ネイバーごとに VTYSH BGP ルータモードで次のコマンドを設定します。

VTYSH コマンドプロンプトで、次のように入力します。

```
neighbor <peer_ip> allowas-in 1
```

ベストプラクティスとして、以下のいずれかを構成することをお勧めします。

- ルートマップを設定して、クラスタノード上のデフォルトルート、Citrix ADC IP（NSIP）、NSIP サブネットなど、必要なネットワークのみを学習します。
- クラスタ内の CLIP や Citrix ADC IP（NSIP）など、必要なネットワークのみをアドバタイズするようにアップストリームルートを設定します。

クラスタの **IP** アドレス指定

October 7, 2021

Citrix ADC NSIP、仮想 IP（VIP）、およびサブネット IP（SNIP）の標準タイプに加えて、クラスタ化された Citrix ADC アプライアンスは、クラスタ管理 IP（CLIP）アドレスを持つことができます。また、ストライプおよびスポッティング IP アドレスを持つこともできます。

- **CLIP** アドレス。クラスタコーディネータノード (CCO) が所有する IP アドレス。CLIP アドレスは、クラスタ設定の異なるノード間でフローティングできます。CLIP をクラスタの別のノードに移動すると、そのノード

が CCO になります。CCO は、クラスタ内の管理タスクを担当する Citrix ADC アプライアンスです。ネットワーク管理者は、CLIP アドレスを使用してクラスターに接続し、統合 GUI へのアクセス、レポート、パケットフローのトレース、ログの収集などの構成および管理タスクを実行します。同一または異なるネットワーク上のクラスタ内に複数の CLIP アドレスを追加できます。クラスタの IP アドレスを介して CCO で実行された設定だけが、クラスタ内の他のノードに伝播されます。

- ストライピングされた **IP** アドレス。クラスタのすべてのノードで使用可能な論理 IP アドレス。VIP または SNIP アドレスのいずれかになります。
- スポッティング **IP** アドレス。論理 IP（好ましくは SNIP アドレス）は、1 つのノードでのみ使用できます。スポッティング IP アドレスは、そのノードでのみ可視性を持ちます。トラフィックステアリングのオーバーヘッドを最小限に抑えるために、サーバーとのバックエンド通信にスポットされた SNIP アドレスを使用することをお勧めします。

次の表に、構成の詳細を示します。

IP アドレス	NSIP	VIP	SNIP
スポッティング	はい	はい	はい
ストライプ	いいえ	はい	はい

たとえば、4 ノードのクラスターグループでは、各ノードにスポッティング SNIP アドレスを構成する必要があります。スポッティング IP 設定を構成する方法の詳細については、「ストライピング、部分ストライプ、およびスポッティングされた構成」を参照してください。

SNIP アドレスは、1 つのノードでのみアクティブにすることも、すべてのノードでアクティブにすることもできます。仮想 IP アドレスとサブネット IP アドレスが特定のノードでのみ使用可能な場合、それはスポット構成です。サブネット IP アドレスと仮想サーバー IP アドレスがすべてのノードで使用可能な場合、構成はストライプとして定義されます。スポットされた SNIP アドレスは、ステアリングとバックプレーンのトラフィックを減らすのに役立ちます。

ノードをクラスターに参加させる際の **VLAN** バインディングとルート構成のベストプラクティス

VLAN IP バインディング

VLAN をスポット IP アドレスでバインドする場合、Citrix ADC クラスターは、すべてのノードの同じサブネット内のスポット IP アドレスで構成する必要があります。たとえば、ノード 0 とノード 1 の 2 ノードクラスターでは、次の構成を使用できます。

```
1 add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 1
2 add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 0
```



```
add vlan 100
bind vlan 100 -IPAddress 192.254.101.101 255.255.255.0
<!--NeedCopy-->
```

ルーティング構成

スポット IP アドレスをデフォルトゲートウェイとしてルーティング構成が必要な場合は、ADC クラスターをすべてのノードの同じサブネット内のスポット IP アドレスで構成する必要があります。たとえば、ノード 0 とノード 1 の 2 ノードクラスターでは、次の構成を使用できます。

```
add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
    dynamicRouting ENABLED -ownerNode 1
add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
    dynamicRouting ENABLED -ownerNode 0

add route 192.254.102.0 255.255.255.0 192.254.101.103
<!--NeedCopy-->
```

> 注
>
> L3 クラスターのセットアップでは、スポットされた SNIP 構成のみがサポートされます。

レイヤ 3 クラスタリングの構成

January 31, 2022

L3 クラスターを理解する

高可用性の展開を拡張し、L3 クラスターを確立するためにガイドされたさまざまなネットワーク間でクライアントトラフィックのスケーラビリティを向上させるという要求。L3 クラスターを使用すると、Citrix ADC アプライアンスを個々のサブネット間でグループ化できます（L2 クラスター）。

L3 クラスターは、「独立ネットワーク構成（INC）モードのクラスター」とも呼ばれます。L3 クラスター展開では、同じネットワーク内のクラスターノードがグループ化されてノードグループを形成します。L3 クラスタは、GRE トンネリングを使用して、ネットワーク間でパケットを操縦します。L3 クラスター全体のハートビートメッセージがルーティングされます。

このドキュメントでは、次の詳細について説明します。

- アーキテクチャ
- 例

アーキテクチャ

L3 クラスターアーキテクチャは、次のコンポーネントで構成されています。

- ノードグループ。次の図に示すように、各ネットワーク（n1、n2）および（n3、n4）のクラスターノードはグループ化されてノードグループを形成します。これらのノードグループは、ネットワークのいずれかの側のレイヤー 3 スイッチで終端されます。
  - クラスタは、クラスタノードとクライアント側の接続デバイス間の物理接続を介してクライアントと通信します。これらの物理接続の論理的なグループ化は、クライアントデータプレーンと呼ばれます。
  - クラスタは、クラスタノードとサーバ側の接続デバイス間の物理接続を介してサーバと通信します。これらの物理接続の論理的なグループ化は、サーバデータプレーンと呼ばれます。
- バックプレーンスイッチ。同じネットワーク内のクラスタノードは、クラスタバックプレーンを使用して相互に通信します。バックプレーンは、各ノードの 1 つのインターフェイスが共通のスイッチ（クラスタバックプレーンスイッチ）に接続されている一連のインターフェイスです。
- **GRE** トンネル。L3 クラスタ内のノード間のパケットは、ルーティングのために送信元ノードと宛先ノードの NSIP アドレスを使用する暗号化されていない GRE トンネルを介して交換されます。異なるネットワークに属するノードのステアリングメカニズムが変わります。パケットは、MAC を書き換えるのではなく、GRE トンネルを介して他のサブネット上のノードに操縦されます。

例

以下で構成される L3 クラスタ展開の例を考えてみましょう。

- 3 つの Citrix ADC アプライアンス（n1、n2、および n3）ノードが Nodegroup1 にグループ化されます。
- 同様に、ノード n4 と n5 は、ノードグループ 2 にグループ化されています。3 番目のネットワークには、2 つのノードグループがあります。ノードグループ 3 には n6 と n7 が含まれており、ノードグループ 4 には n8 と n9 が含まれています。
- 同じネットワークに属する Citrix ADC アプライアンスは、ノードグループを形成するために結合されます。

L3 クラスターを構成する前に考慮すべきポイント

Citrix ADC アプライアンスで L3 クラスターを構成する前に、次の点を考慮してください。

- L3 サブネットの設定時には、バックプレーンは必須ではありません。バックプレーンが指定されていない場合、ノードはバックプレーン障害状態になりません。

  > 注
  >
  > 同じ L2 ネットワークに複数のノードがある場合は、バックプレーンインターフェイスを定義する必要があります。バックプレーンインターフェイスが言及されていない場合、ノードはバックプレーン障害状態になります。

- L2 機能とストライプ SNIP は、L3 クラスターではサポートされていません。
- L3 クラスターの外部トラフィック分散は、Equal Cost Multiple Path（ECMP）のみをサポートします。

- L3 クラスター展開でステアリングが無効になっている場合、ICMP エラーとフラグメンテーションは処理されません。
- ネットワークエンティティ（route， route6， pbr、および pbr6）は、構成ノードグループにバインドする必要があります。
- VLAN、RNAT、および IP トンネルを構成ノードグループにバインドすることはできません。
- 構成ノードグループには、常にプロパティ STRICT“ YES。
- 「addclusternode」コマンドを使用して、クラスターノードを構成ノードグループに追加しないでください。
- add cluster instance -INC enabled コマンドは、ネットワークエンティティ（route、route6、PBR、pb6、RNAT、IP トンネル、ip6tunnel）をクリアします。
- clear config extended+ コマンドは、L3 クラスター内のエンティティ（route、route6、PBR、pb6、RNAT、IP トンネル、ip6tunnel）をクリアしません。

L3 クラスターの設定

L3 クラスター構成では、cluster コマンドには、ノードおよびノードグループに基づいて構成するさまざまな属性があります。L3 クラスタ構成には、IPv4 プロファイルとは別に IPv6 プロファイルも含まれます。

Citrix ADC アプライアンスでの L3 クラスターの構成は、次のタスクで構成されます。

- クラスターインスタンスの作成
- L3 クラスターにノードグループを作成する
- Citrix ADC アプライアンスをクラスターに追加し、ノードグループ
  でグループ化します- クラスタ IP アドレスをノードに追加します
- クラスターインスタンスの有効化
- 構成を保存します
- 既存のノードグループにノードを追加する
- L3 クラスターにノードグループを作成する
- 新しいノードを新しく作成されたノードグループにグループ化します
- ノードをクラスタに結合する

CLI を使用して以下を構成する

- クラスターインスタンスを作成するには

  ```
  add cluster instance <clid> -inc (<ENABLED|DISABLED>)[-processLocal <
  ENABLED | DISABLED]
  ```

- **L3** クラスタでノードグループを作成するには

  ```
  add cluster nodegroup <name>
  ```


- **Citrix ADC** アプライアンスをクラスタに追加し、ノードグループに関連付けるには

  ```
  add cluster node <nodeid> <nodeip> -backplane <interface_name> node group <ng>
  ```

- このノードでクラスタ **IP** アドレスを追加するには

  ```
  add ns ip <IPAddress> <netmask> -type clip
  ```

- クラスターインスタンスの有効化

  ```
  enable cluster instance <clId>
  ```

- 構成を保存します

  ```
  save ns config
  ```

- アプライアンスのウォームリブート

  ```
  reboot -warm
  ```

- 既存のノードグループに新しいノードを追加するには

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng>
  ```

- **L3** クラスターに新しいノードグループを作成するには

  ```
  add cluster nodegroup <ng>
  ```

- 新規ノードを新規作成したノードグループにグループ化するには

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng>
  ```

- ノードをクラスタに参加させるには

```
1     join cluster  - clip <ip_addr> -password <password>
2
3     add cluster instance 1  - inc ENABLED  - processLocal  ENABLED
4
5        Done
6 <!--NeedCopy-->
```

> 注
>
> L3 クラスターでは、「inc」パラメーターを ENABLED にする必要があります。

```
1     add cluster nodegroup ng1
2
3        Done
4
```



```
5      > add cluster node 0 1.1.1.1 -state ACTIVE -backplane 0/1/1 -
           nodegroup ng1
6
7         Done
8
9      > add ns ip 1.1.1.100 255.255.255.255 -type clip
10
11        Done
12
13     > enable cluster instance 1
14
15        Done
16
17     > save ns config
18
19        Done
20
21     > add cluster node 1 1.1.1.2 -state ACTIVE -nodegroup ng1
22
23        Done
24
25     > add cluster nodegroup ng2
26
27        Done
28
29     > add cluster node 4 2.2.2.1 -state ACTIVE -nodegroup ng2
30
31        Done
32
33     > add cluster node 5 2.2.2.2 -state ACTIVE -nodegroup ng2
34
35        Done
36
37     > join cluster -clip 1.1.1.100 -password nsroot
38 <!--NeedCopy-->
```

L3 クラスターのアドバタイズクラスター **IP** アドレス

アップストリームルーターにアドバタイズされるクラスター IP アドレスを構成して、任意のサブネットからクラスター構成にアクセスできるようにします。クラスタ IP アドレスは、ノードに設定された動的ルーティングプロトコルによってカーネルルートとしてアドバタイズされます。

クラスタ IP アドレスのアドバタイズは、次のタスクで構成されます。

- クラスタ **IP** アドレスのホストルートオプションを有効にします。ホストルートオプションは、動的ルーティングプロトコルを介したカーネルルート再配布のために、クラスター IP アドレスを ZebOS ルーティングテーブルにプッシュします。
- ノードでの動的ルーティングプロトコルの構成。 ダイナミックルーティングプロトコルは、クラスタ IP アドレスをアップストリームルータにアドバタイズします。ダイナミックルーティングプロトコルの設定の詳細については、ダイナミックルートの設定を参照してください。

CLI を使用してクラスター **IP** アドレスのホストルートオプションを有効にするには

コマンドプロンプトで、次のように入力します。

```
1 -  add nsip <IPAddress> <netmask> -hostRoute ENABLED
2
3 -  show nsip <IPAddress>
4
5     > add ns ip 10.102.29.60 255.255.255.255 -hostRoute ENABLED
6
7        Done
8 <!--NeedCopy-->
```

L3 クラスタで部分的にストライピングされた構成にスポッティング

L3 クラスターの斑点および部分的に縞模様の構成は、L2 クラスターとはわずかに異なります。ノードが異なるサブネット上に存在するため、構成はノードごとに異なる場合があります。ネットワーク構成は L3 クラスター内でノード固有である可能性があるため、以下のパラメーターに基づいて、スポット構成または部分的にストライプ化された構成を構成する必要があります。

L3 クラスター上の Citrix ADC アプライアンスで、スポットされた部分的にストライプ化された構成を構成するには、次のタスクを実行します。

- クラスター所有者グループを IPv4 静的ルーティングテーブルに追加します
- クラスター所有者グループを IPv6 静的ルーティングテーブルに追加します
- クラスター所有者グループを IPv4 ポリシーベースルーティング（PBR）に追加する
- クラスター所有者グループを IPv6PBR に追加します
- VLAN の追加
- VLAN をクラスターノードグループの特定の所有者グループにバインドします

CLI を使用して以下を構成する

- **Citrix ADC** アプライアンスの **IPv4** 静的ルートテーブルにクラスター所有者グループを追加するには

  ```
  add route <network> <netmask> <gateway> -owner group <ng>
  ```


- **Citrix ADC** アプライアンスの **IPv6** 静的ルートテーブルにクラスター所有者グループを追加するには

  ```
  add route6 <network> -owner group <ng>
  ```

- クラスター所有者グループを **IPv4PBR** に追加するには

  ```
  add pbr <name> <action> -owner group <ng>
  ```

- クラスター所有者グループを **IPv6PBR** に追加するには

  ```
  add pbr6 <name> <action> -owner group <ng>
  ```

- **VLAN** を追加するには

  ```
  add vlan <id>
  ```

- **VLAN** をクラスタノードグループの特定の所有者グループにバインドするには

  ```
  bind vlan <id> -ifnum - [IPAddress <ip_addr | ipv6_addr> [-owner group
  <ng>]
  ```

  次のコマンドは、CLI を使用して設定できるスポット設定および部分ストライプ設定のサンプル例です。

```
1     > add route 10.102.29.0 255.255.255.0 10.102.29.2  - ownergroup ng2
2
3         Done
4
5     > add route6 fe80::9404:60ff:fedd:a464/64  - ownergroup ng1
6
7         Done
8
9     > add pbr pbr1 allow  - ownergroup ng1
10
11        Done
12
13    > add pbr6 pbr2 allow  - ownergroup ng2
14
15        Done
16
17    > add vlan 2
18
19        Done
20
21    > bind vlan 2  - ifnum 1/2  - [IPAddress 10.102.29.80 | fe80::9404:60
          ff:fedd:a464/64-ownergroup ng1
22
23        Done
24 <!--NeedCopy-->
```


ノードグループを構成する

L3 クラスターでは、同じ構成セットを複数のノードグループに複製するには、次のコマンドを使用します。

CLU を使用して以下を構成する

- **Citrix ADC** アプライアンスのルーティングテーブルに **IPv4** 静的ルートを追加するには

  ```
  add route <network> <netmask> <gateway> -ownerGroup <ng>
  ```

設定例**:**

```
1 add route 0 0 10.102.53.1 -ownerGroup ng1
2
3 add route 0 0 10.102.53.1 -ownerGroup ng2
4 <!--NeedCopy-->
```

上記の構成をサポートするために新しいノードグループ「all」を定義し、次のコマンドを構成する必要があります。

CLI を使用して以下を構成する

- **strict** パラメーターを使用してクラスターに新しいノードグループを追加するには

  ```
  add cluster node group <name> -strict <YES | NO>
  ```

- クラスタノードまたはエンティティを指定されたノードグループにバインドするには

  ```
  bind cluster nodegroup <name> -node <nodeid>
  ```

- すべての所有者グループに **IPv4** 静的ルートを追加するには

  ```
  add route <network> <netmask> <gateway> -ownerGroup <ng>
  ```

設定例:

```
1 add cluster nodegroup all -strict YES
2
3 bind cluster nodegroup all -node 1
4
5 bind cluster nodegroup all -node 2
6
7 add route 0 0 10.102.53.1 -ownerGroup all
8 <!--NeedCopy-->
```


L3 クラスタ内のトラフィック分散

クラスター設定では、外部ネットワークは Citrix ADC アプライアンスのコレクションを単一のエンティティとして表示します。したがって、クラスターは、トラフィックを受信する必要がある単一のノードを選択する必要があります。L3 クラスタでは、この選択は ECMP を使用して行われます。選択されたノードをフロー受信機と呼びます。

> 注
>
> L3 クラスター（異なるネットワークにまたがるノード）の場合、ECMP トラフィック分散のみを使用できます。

フローレシーバはトラフィックを取得し、内部クラスタロジックを使用してトラフィックを処理する必要があるノードを決定します。このノードをフロープロセッサと呼びます。フローレシーバーとフロープロセッサーが同じネットワーク上にある場合、フローレシーバーはバックプレーンを介してトラフィックをフロープロセッサーに誘導します。フローレシーバーとフロープロセッサーが異なるネットワーク上にある場合、トラフィックはトンネルを介してステアリングされます。

> 注
>
> - フローレシーバとフロープロセッサは、トラフィックを処理できるノードである必要があります。
> - NetScaler 11 以降では、クラスタバックプレーンでステアリングを無効にできます。詳細については、クラスタバックプレーンでのステアリングの無効化を参照してください。

前の図は、クラスターを流れるクライアント要求を示しています。クライアントは、仮想 IP（VIP）アドレスに要求を送信します。クライアントデータプレーンに設定されたトラフィック分散メカニズムは、クラスタノードの 1 つをフローレシーバとして選択します。フローレシーバは、トラフィックを受信し、トラフィックを処理する必要があるノードを決定し、そのノードに要求を操縦します（フローレシーバが自身をフロープロセッサとして選択しない限り）。フロープロセッサとフローレシーバが同じノードグループにある場合、パケットはバックプレーン上でステアリングされます。また、フロープロセッサとフローレシーバが異なるノードグループに属している場合、パケットはルーテッドパスを介してトンネルを通過します。

フロープロセッサは、サーバとの接続を確立します。サーバは要求を処理し、要求をサーバに送信したサブネット IP（SNIP）アドレスに応答を送信します。L3 クラスタでは SNIP は常にスポッティングされた SNIP であるため、SNIP アドレスを所有するノードはサーバからの応答を受信します。

Citrix ADC クラスターをセットアップする

October 7, 2021

クラスターに追加する Citrix ADC アプライアンスは、「クラスターノードの前提条件」で指定された基準を満たす必要があります。クラスターを実際にセットアップする前に、クラスターの基本を理解しておく必要があります。詳細については、「クラスターの概要」を参照してください。

クラスターを形成するには、ノード間通信をセットアップし、（最初の Citrix ADC アプライアンスを追加して）クラスターを作成し、他のクラスターノードを追加する必要があります。これらの各手順については、後続のトピックで

関連する詳細とともに説明します。

> 注
>
> L2 および L3 クラスタの設定にはいくつかの違いがありますが、類似点も多数あります。以降のトピックでは、L3 クラスタに固有の設定を強調しながら、両方のクラスタタイプのセットアップについて説明します。

ノード間通信の設定

October 7, 2021

クラスタ設定のノードは、次のノード間通信メカニズムを使用して相互に通信します。

- ネットワーク内のノード（同じサブネット）は、クラスタバックプレーンを介して相互に通信します。バックプレーンは明示的に設定する必要があります。詳細な手順は次のとおりです。
- ネットワーク間で、パケットのステアリングは GRE トンネルを介して行われ、必要に応じて他のノード間通信がノード間でルーティングされます。

> 重要
>
> - リリース 11.0 から、すべてのビルドでは、クラスタに異なるネットワークのノードを含めることができます。
> - リリース 13.0 ビルド 58.3 からは、L3 クラスタの Fortville NIC で GRE ステアリングがサポートされています。

クラスタバックプレーンを設定するには、各ノードで次の操作を行います

1. バックプレーンに使用するネットワークインターフェイスを特定します。
2. 選択したネットワークインターフェイスからクラスタバックプレーンスイッチに、イーサネットケーブルまたは光ケーブルを接続します。

たとえば、ノード 4 のバックプレーンインターフェイスとしてインターフェイス 1/2 を使用するには、ノード 4 の 1/2 インターフェイスからバックプレーンスイッチにケーブルを接続します。

クラスタバックプレーンを設定する際の注意点

- アプライアンスの管理インターフェイス（0/x）をバックプレーンインターフェイスとして使用しないでください。クラスタでは、インターフェイス 0/1/x は次のように読み込まれます。

  0 -> ノード ID 0
  1/x-> Citrix ADC インターフェイス

- クライアントまたはサーバのデータプレーンには、バックプレーンインターフェイスを使用しないでください。
- クラスタバックプレーンには、リンク集約（LA）チャネルを使用することをお勧めします。

- バックプレーンがバックツーバック接続されている 2 ノードクラスタでは、次のいずれかの条件下で、クラスタは動作上 DOWN になります。
  - ノードの 1 つが再起動されます。
  - いずれかのノードのバックプレーンインターフェイスが無効になっています。

  したがって、別のクラスタノードとトラフィックに影響が及ばないように、バックプレーン専用のスイッチの使用をお勧めします。バックツーバックリンクを使用してクラスターをスケールアウトすることはできません。クラスターノードをスケールアウトすると、運用環境でダウンタイムが発生することがあります。
- クラスタのすべてのノードのバックプレーンインターフェイスは、同じスイッチに接続され、同じ L2 VLAN にバインドされている必要があります。
- 同じクラスタインスタンス ID を持つクラスタが複数ある場合は、各クラスタのバックプレーンインターフェイスが異なる VLAN にバインドされていることを確認します。
- バックプレーンインターフェイスは、そのインターフェイスの HA モニタリング設定に関係なく、常にモニタされます。
- 異なる仮想化プラットフォームでの MAC スプーフィングの状態は、クラスタバックプレーンのステアリングメカニズムに影響を与える可能性があります。したがって、適切な状態が設定されていることを確認します。
  - XenServer - MAC スプーフィングを無効にする
  - Hyper-V - MAC スプーフィングを有効にする
  - VMware ESX-MAC スプーフィングを有効にする（「偽造転送」が有効になっていることも確認）
- クラスタバックプレーンの MTU が自動的に更新されます。ただし、クラスタにジャンボフレームが設定されている場合は、クラスタバックプレーンの MTU を明示的に設定する必要があります。この値は 78 + X に設定する必要があります。X は、クライアントおよびサーバのデータプレーンの最大 MTU です。たとえば、サーバデータプレーンの MTU が 7500 で、クライアントデータプレーンの MTU が 8922 であるとします。クラスタバックプレーンの MTU は、78 + 8922 = 9000 に設定する必要があります。この MTU を設定するには、次のコマンドを使用します。

  ```
  > set interface <backplane_interface> -mtu <value>
  ```
- バックプレーンスイッチのインターフェイスの MTU は、1,578 バイト以上に指定する必要があります。クラスタに MBF、L2 ポリシー、ACL、CLAG 展開でのルーティング、および vPath などの機能がある場合に適用できます。

L2 および L3 クラスタに対する UDP ベースのトンネルのサポート

Citrix ADC リリース 13.0 ビルド 36.x 以降では、Citrix ADC L2 および L3 クラスタは UDP ベースのトンネリングを使用してトラフィックを制御できます。これは、クラスタ内の 2 つのノードのノード間通信用に定義されます。「tunnelmode」パラメータを使用すると、add and set clusternode コマンドから GRE または UDP トンネルモードを設定できます。

L3 クラスタ展開では、Citrix ADC ノード間のパケットは、ルーティングのために送信元ノードと宛先ノードの NSIP アドレスを使用する暗号化されていない GRE トンネルを介して交換されます。この交換がインターネット経由で行われる場合、IPsec トンネルがない場合、NSIP はインターネット上に公開され、セキュリティ上の問題が発生する可能性があります。

> 重要
>
> L3 クラスターを使用する場合は、独自の IPsec ソリューションを確立することをお勧めします。

次の表は、さまざまな配置に基づいてトンネルサポートを分類するのに役立ちます。

ステアリングタイプ	AWS	Microsoft Azure	オンプレミスで
MAC	未サポート	未サポート	サポート対象か
GRE トンネル	サポート対象か	未サポート	サポート対象か
UDP トンネル	サポート対象か	サポート済み	サポート対象か

> 重要
>
> L3 クラスタでは、トンネルモードはデフォルトで GRE に設定されています。

UDP ベースのトンネルの設定

ノード ID のパラメータを設定して状態を言及することで、クラスタノードを追加できます。インターフェイス名を指定してバックプレーンを設定し、任意のトンネルモード（GRE または UDP）を選択します。

CLI のプロシージャ

CLI を使用して UDP トンネルモードを有効にします。

コマンドプロンプトで入力します。

- `add cluster node <nodeId>@ [-state <state>] [-backplane <interface_name>] [-tunnelmode <tunnelmode>]`
- `set cluster node <nodeId>@ [-state <state>] [-tunnelmode <tunnelmode>]`

> 注
>
> トンネルモードに設定可能な値は、NONE、GRE、UDP です。

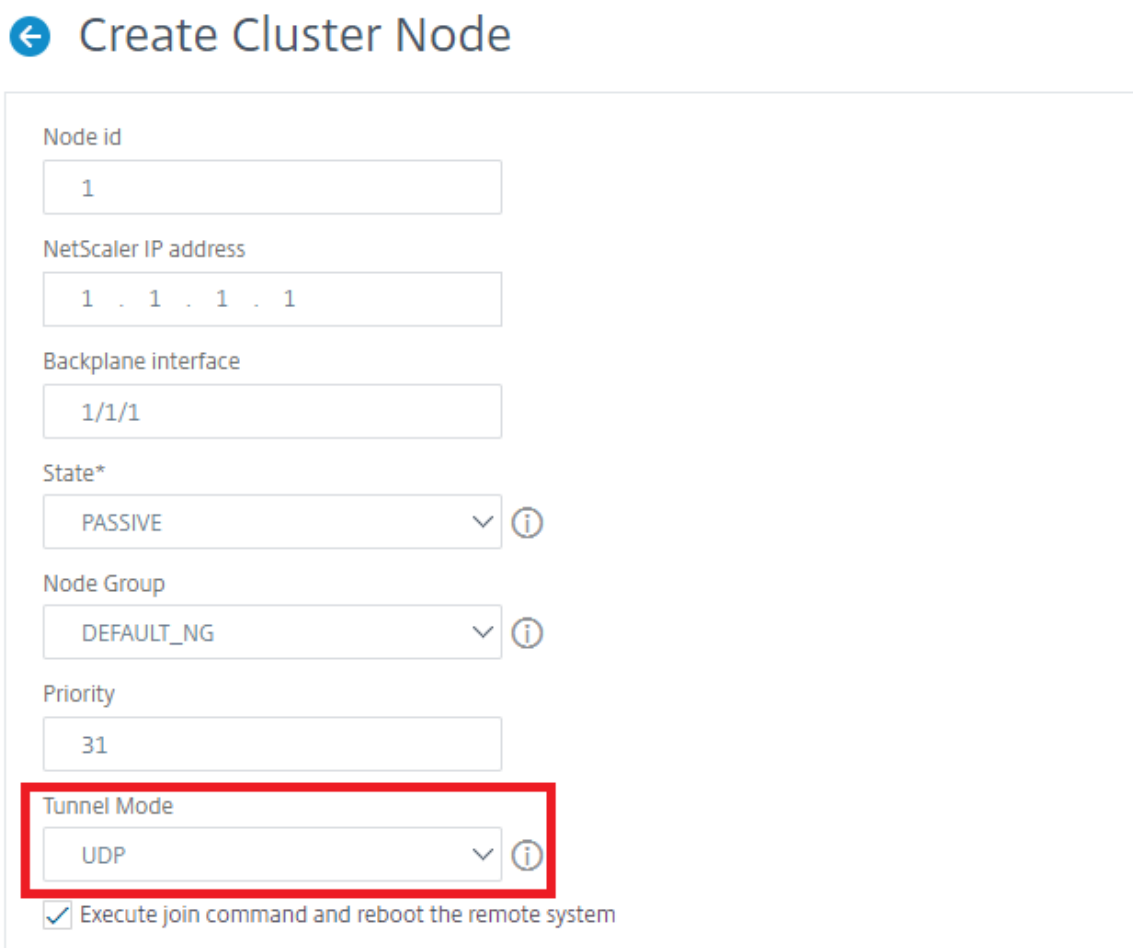
例

- `add cluster node 1 -state ACTIVE -backplane 1/1/1 -tunnelmode UDP`
- `set cluster node 1 -state ACTIVE -tunnelmode UDP`

GUI のプロシージャ

GUI を使用して UDP トンネルモードを有効にします。

1. **System > Cluster > Nodes** に移動します。
2. [クラスタノード] ページで、[追加] をクリックします。
3. [クラスタノードの作成] で、[トンネルモード] パラメータを [UDP] に設定し、[作成] をクリックします。



4. ［閉じる］をクリックします。

Citrix ADC クラスターを作成する

October 7, 2021

クラスターを作成するには、まず、クラスターに追加する Citrix ADC アプライアンスのいずれかを使用します。このノードでは、クラスタインスタンスを作成し、クラスタ IP アドレスを定義する必要があります。このノードは最初のクラスターノードであり、クラスター構成コーディネーター（CCO）と呼ばれます。クラスター IP アドレスで実行されるすべての構成は、このノードに格納され、他のクラスターノードに伝達されます。

クラスター内の CCO の責任は、特定のノードに固定されていません。次の要因によって、時間の経過とともに変化する可能性があります。

- ノードの優先順位。優先度が最も高い（優先度が最も低い）ノードが CCO になります。したがって、既存の CCO よりも低い優先順位番号のノードが追加されると、新しいノードが CCO として引き継がれます。

  > 注
  >
  > ノードの優先順位は、NetScaler 10.1 以降から構成できます。

- 現在の CCO がダウンした場合、次に優先度の低いノードが CCO として引き継ぎます。優先度が設定されていない場合、または優先度番号が最も小さいノードが複数ある場合は、使用可能なノードの 1 つから CCO が選択されます。

> 注
>
> アプライアンスの構成（SNIP アドレスと VLAN を含む）は、`clear ns config extended` コマンドを暗黙的に実行することによってクリアされます。ただし、デフォルトの VLAN と NSVLAN はアプライアンスからはクリアされません。したがって、クラスタに NSVLAN が必要な場合は、アプライアンスをクラスタに追加する前に NSVLAN が作成されていることを確認してください。L3 クラスタ（異なるネットワーク上のクラスタノード）の場合、ネットワーク設定はアプライアンスから消去されません。
>
> 重要
>
> クラスタ設定の HA モニタ (HAMON) は、各ノードのインターフェイスの健全性を監視するために使用されます。インターフェイスの状態を監視するには、各ノードで HAMON パラメータを有効にする必要があります。何らかの理由で HAMON 対応インターフェイスの動作状態がダウンした場合、それぞれのクラスタノードは Unhealthy（NOT UP）としてマークされ、そのノードはトラフィックを処理できません。

コマンドラインインターフェイスを使用してクラスターを作成するには

1. クラスタに追加するアプライアンス（たとえば、NSIP アドレスが 10.102.29.60 のアプライアンス）にログオンします。
2. クラスターインスタンスを追加します。

   ```
   add cluster instance <clId> -quorumType <NONE | MAJORITY> -inc <ENABLED | DISABLED> -backplanebasedview <ENABLED | DISABLED><!--NeedCopy-->
   ```

   > 注
   >
   > ```
   > 1    - クラスタインスタンス ID は、LAN 内で一意である必要があります。
   > ```
   >
   > - 次のシナリオでは、`-quorumType` パラメータを [MAJORITY] に設定し、[なし] に設定する必要

があります。

- クラスタノード間に冗長リンクがないトポロジ。これらのトポロジは、単一障害点のためにネットワークパーティションになりやすい可能性があります。
- ノードの追加や削除などのクラスタ操作中。

- L3 クラスタの場合は、-inc パラメータが ENABLED に設定されていることを確認します。L2 クラスタでは、-inc パラメータを無効にする必要があります。
- -backplanebasedviewパラメータが有効の場合、動作ビュー（トラフィックを処理するノードのセット）は、バックプレーンインターフェイスだけで受信されたハートビートに基づいて決定されます。デフォルトでは、このパラメータは無効になっています。このパラメータが無効の場合、ノードはバックプレーン上のハートビート受信だけに依存しません。

3. [L3 クラスタの場合のみ] 、ノードグループを作成します。次のステップでは、新しく追加されたクラスタノードをこのノードグループに関連付ける必要があります。

> 注
>
> このノードグループには、同じネットワークに属する Citrix ADC アプライアンスのすべてまたはサブセットが含まれます。

```
add cluster nodegroup <name><!--NeedCopy-->
```

4. Citrix ADC アプライアンスをクラスターに追加します。

“‘add cluster node -state -backplane -nodegroup

```
> **注： ** L3 クラスタの場合：
>
>- ノードグループパラメータは、作成されるノードグループの名前に設定する必要があります。
>- ネットワーク内のノードが相互に通信できるように、複数のノードを持つノードグループに関連付けられているノードにはバックプレーンパラメータが必須です。 </span>

例：

L2 クラスターのノードを追加する （すべてのクラスターノードが同じネットワークにある）。
```

add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1

```
各ネットワークの 1 つのノードを含む L3 クラスタのノードを追加します。ここでは、バックプレーンを設定する必要はありません。
```

add cluster node 0 10.102.29.60 -state PASSIVE -nodegroup ng1


```
1    各ネットワークからの複数のノードを含む L3 クラスタのノードを追加
       します。ここでは、ネットワーク内のノードが互いに通信できるよ
       うに、バックプレーンを設定する必要があります。
```

add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1 -nodegroup ng1
‘‘‘

5. このノードでクラスター IP アドレス (10.102.29.61 など) を追加します。

```
1 add ns ip <IPAddress> <netmask> -type clip
2 <!--NeedCopy-->
```

例

```
1 > add ns ip 10.102.29.61 255.255.255.255 -type clip
2 <!--NeedCopy-->
```

6. クラスタインスタンスを有効にします。

   ```
   enable cluster instance <clId><!--NeedCopy-->
   ```

7. 構成を保存します。

   ```
   save ns config<!--NeedCopy-->
   ```

8. アプライアンスをウォーム再起動します。

   ```
   reboot -warm<!--NeedCopy-->
   ```

   show cluster instance コマンドを使用して、クラスタ設定を確認します。コマンドの出力に、クラスタのノードとしてアプライアンスの NSIP アドレスが表示されていることを確認します。

9. ノードが UP になったら、CLIP にログインし、クラスタ IP アドレスとノード IP アドレスの両方の RPC 資格情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。

GUI を使用してクラスターを作成するには

1. クラスタに追加するアプライアンス（たとえば、NSIP アドレスが 10.102.29.60 のアプライアンス）にログオンします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインで、[クラスターの管理] リンクをクリックします。

4. [クラスタ構成] ダイアログボックスで、クラスタの作成に必要なパラメータを設定します。パラメータの説明を参照するには、対応するテキストボックスにマウスカーソルを合わせます。
5. [作成] をクリックします。
6. [クラスターインスタンスの構成] ダイアログボックスで、[クラスターインスタンスの有効化] チェックボックスをオンにします。
7. [クラスタノード] ペインで、ノードを選択し、[開く] をクリックします。
8. [クラスタノードの構成] ダイアログボックスで、[状態] を設定します。
9. [**OK**] をクリックし、[保存] をクリックします。
10. アプライアンスをウォーム再起動します。
11. ノードが UP になったら、CLIP にログインし、クラスタ IP アドレスとノード IP アドレスの両方の RPC 資格情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。

クラスタの同期状態に対する厳密なモードのサポート

構成を適用するときにエラーを表示するようにクラスタノードを構成できるようになりました。クラスター内の各ノードのステータスを追跡するために、新しいパラメーター「syncStatusStrictMode」がクラスターインスタンスの追加コマンドと設定コマンドの両方に導入されました。デフォルトでは、「SyncStatusStrictMode」パラメータは無効になっています。

CLI を使用して厳密モードを有効にするには

コマンドプロンプトで入力します。

```
set cluster instance <clID> [-syncStatusStrictMode (ENABLED | DISABLED)]
```

例**:**

```
set cluster instance 1 -syncStatusStrictMode ENABLED
```

CLI を使用して **strict** モードのステータスを表示するには

```
1      >show cluster instance
2      1) Cluster ID: 1
3          Dead Interval: 3 secs
4          Hello Interval: 200 msecs
5          Preemption: DISABLED
6          Propagation: ENABLED
7          Quorum Type: MAJORITY
8          INC State: DISABLED
9          Process Local: DISABLED
10         Retain Connections: NO
```



```
        Heterogeneous: NO
        Backplane based view: DISABLED
        Cluster sync strict mode: ENABLED
        Cluster Status: ENABLED(admin), ENABLED(operational), UP

      WARNING(s):
        (1) - There are no spotted SNIPs configured on the cluster.
           Spotted SNIPs can help improve cluster performance

        Member Nodes:
        Node ID     Node IP      Health     Admin State  Operational
           State
        -------     -------      ------     -----------
           -----------------
    1)      1           192.0.2.20      UP         ACTIVE       ACTIVE(
       Configuration Coordinator)
    2)      2           192.0.2.21      UP         ACTIVE       ACTIVE
    3)      3           192.0.2.19*     UP         ACTIVE       ACTIVE
<!--NeedCopy-->
```

GUI を使用してクラスターノードの同期失敗の理由を表示するには

1. [システム] **>** [クラスタ] **>** [クラスタノード] に移動します。
2. [クラスターノード] ページで、右端までスクロールして、クラスターノードの同期失敗の理由の詳細を表示します。

クラスタへのノードの追加

October 7, 2021

クラスターのサイズをシームレスにスケーリングして、最大 32 個のノードを含めることができます。Citrix ADC アプライアンスがクラスターに追加されると、そのアプライアンスの構成がクリアされます（clear ns config -extended コマンドを内部的に実行することにより）。SNIP アドレス、バックプレーンインターフェイスの **MTU** 設定、およびすべての VLAN 構成（デフォルトの VLAN と NSVLAN を除く）もアプライアンスからクリアされます。

クラスター構成は、このノード上で同期されます。同期の進行中に、トラフィックが断続的にドロップされる可能性があります。

> 重要
>
> Citrix ADC アプライアンスをクラスターに追加する前に：

- ノードのバックプレーンインターフェイスを設定します。前のトピックを確認してください。
- アプライアンスで使用可能なライセンスが、構成コーディネーターで使用可能なライセンスと一致するかどうかを確認します。アプライアンスは、ライセンスが一致する場合にのみ追加されます。
- NSVLAN をクラスタに追加する場合は、NSVLAN がアプライアンスで作成されていることを確認してから、クラスタに追加してください。
- ノードをパッシブノードとして追加することをお勧めします。次に、ノードをクラスタに参加させた後、クラスタ IP アドレスからノード固有の構成を完了します。クラスタに検出された IP アドレスしかない場合は、force clustersync コマンドを実行します。また、L3 VLAN バインディングがあるか、静的ルートがあります。
- 構成済みのリンク集約（LA）チャネルを持つアプライアンスをクラスタに追加しても、LA チャネルはクラスタ環境に存在し続けます。LA チャネルの名前は LA/x から nodeId/LA/x に変更されます。ここで、LA/x は LA チャネル識別子です。

CLI を使用してクラスターにノードを追加するには

注

クラスター設定にノードを追加するときに、ノードにデフォルトの静的ルートがある場合、そのノードはクラスターコーディネーターノード（CCO）に追加されます。このデフォルトスタティックルートが誤った Gateway を指している場合、サービスのダウンタイムが発生する可能性があります。したがって、クラスタ設定に追加する前に、新しいノードのデフォルトの静的ルートを確認してください。

1. クラスタ IP アドレスにログオンし、コマンドプロンプトで次の操作を行います。
   - アプライアンス（10.102.29.70 など）をクラスタに追加します。

   注

   L3 クラスターの場合：

   ```
   1 - ノードグループパラメータは、同じネットワークのノードを持つ
       ノードグループに設定する必要があります。
   ```

   - このノードが最初に追加されたノードと同じネットワークに属している場合は、そのノードに使用されたノードグループを構成します。
   - このノードが別のネットワークに属している場合は、ノードグループを作成し、このノードをノードグループにバインドします。
   - ネットワーク内のノードが相互に通信できるように、複数のノードを持つノードグループに関連付けられているノードにはバックプレーンパラメータが必須です。

```
1 add cluster node <nodeId> <IPAddress> -state <state> -backplane <
    interface_name> -nodegroup <name>
```



```
2
3 Example:
4
5 add cluster node 1 10.102.29.70 -state PASSIVE -backplane 1/1/1
6 <!--NeedCopy-->
```

- 構成を保存します。

```
1 save ns config
2 <!--NeedCopy-->
```

2. 新しく追加したノード (10.102.29.70 など) にログオンし、ノードをクラスタに参加させます。

```
1 join cluster -clip <ip_addr> -password <password>
2
3 Example:
4
5 join cluster -clip 10.102.29.61 -password nsroot
6 <!--NeedCopy-->
```

3. CLIP で次のコマンドを設定します。
   - VLAN をインターフェースにバインドする

```
1 bind vlan <id> -ifnum <interface_name>
2 <!--NeedCopy-->
```

例:

```
1 bind vlan 1 -ifnum 2/1/2
2 <!--NeedCopy-->
```

   - 新しく追加されたノードにスポットされた IP アドレスを追加します

```
1 add ns ip <IpAddress> <netmask> -ownerNode <positive_interger
     >
2 <!--NeedCopy-->
```


例:

```
add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2
<!--NeedCopy-->
```

- NSIP で VLAN を確認する

```
show vlan <id>
<!--NeedCopy-->
```

例:

```
show vlan 1
<!--NeedCopy-->
```

4. 次の構成を実行します。
   - スポッティング IP のみを持つクラスターにノードが追加された場合、構成は同期されてから、スポッティング IP アドレスがそのノードに割り当てられます。このような場合、L3 VLAN バインディングが失われる可能性があります。この損失を回避するには、ストライプ IP を追加するか、L3 VLAN バインディングを追加します。
   - 必要なスポット設定を定義します。
   - バックプレーンインターフェイスの MTU を設定します。
5. 構成を保存します。

```
save ns config
<!--NeedCopy-->
```

6. アプライアンスをウォーム再起動します。

```
reboot -warm
<!--NeedCopy-->
```

7. ノードが UP 状態になり、同期が成功したら、クラスタの IP アドレスからノードの RPC 認証情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。


```
set rpcNode <node-NSIP> -password <passwd>

Example:

set rpcNode 192.0.2.4 -password mypassword
<!--NeedCopy-->
```

8. クラスタノードを [アクティブ] に設定します。

```
set cluster node <nodeID> -state active.

Example:

set cluster node 1 -state active
<!--NeedCopy-->
```

GUI を使用してクラスターにノードを追加するには

1. クラスタ IP アドレスにログオンします。
2. **System > Cluster > Nodes** に移動します。
3. 詳細ウィンドウで、[追加] をクリックして新しいノードを追加します (10.102.29.70 など)。
4. [クラスタノードの作成] ダイアログボックスで、新しいノードを構成します。パラメータの説明を参照するには、対応するテキストボックスにマウスカーソルを合わせます。
5. [作成] をクリックします。ウォームリブートを実行するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。
6. ノードが UP 状態になり、同期が成功したら、クラスタの IP アドレスからノードの RPC 認証情報を変更します。RPC ノードのパスワードの変更の詳細については、「RPC ノードのパスワードを変更する」を参照してください。
7. [システム] > [クラスタ] > [ノード] > [編集] に移動します。
8. [状態] を [アクティブ] に変更し、確定します。

GUI を使用して、以前に追加したノードをクラスターに参加させる

CLI を使用してノードをクラスターに追加したが、ノードをクラスターに参加させていない場合は、次の手順を使用できます。

> 注
>
> ノードがクラスタに参加すると、クラスタからのトラフィックのシェアを引き継ぐため、既存の接続が終了する可能性があります。

1. クラスタに参加させるノード (10.102.29.70 など) にログオンします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインの [はじめに] で、[クラスターに参加] リンクをクリックします。
4. [既存のクラスターに参加] ダイアログボックスで、構成コーディネーターのクラスター IP アドレスと `nsroot` パスワードを設定します。パラメータの説明を参照するには、対応するテキストボックスにマウスカーソルを合わせます。
5. 「**OK**」をクリックします。

クラスターの詳細の表示

October 7, 2021

クラスターインスタンスとクラスターノードの詳細を表示するには、クラスター IP アドレスにログオンします。

CLI を使用してクラスターインスタンスの詳細を表示するには

クラスタ IP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
1 show cluster instance <clId>
```

> 注
>
> 非 CCO ノードの NSIP アドレスから上記のコマンドを実行すると、このコマンドはこのノードのクラスターの状況を表示します。

CLI を使用してクラスターノードの詳細を表示するには

クラスタ IP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
1 show cluster node <nodeId>
```

GUI を使用してクラスターインスタンスの詳細を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインの [は じめに] で、[クラスターの管理] リンクをクリックして、クラスターの詳細を表示します。

GUI を使用してクラスターノードの詳細を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] **>** [ノード] に移動します。
3. 詳細ペインで、詳細を表示するノードをクリックします。

クラスタノード間でのトラフィックの分散

October 7, 2021

Citrix ADC クラスタを作成し、必要な構成を実行したら、クライアントデータプレーン（クライアントトラフィック用）またはサーバーデータプレーン（サーバートラフィック用）に等価コスト多重パス（ECMP）またはクラスタリンクアグリゲーション（LA）を展開する必要があります。これらのメカニズムは、外部トラフィックをクラスタノード全体に分散します。

ポリシーベースのバックプレーンステアリング

ポリシーベースのバックプレーンステアリング (PBS) は、クラスタ展開のメカニズムで、フローに定義されたハッシュ方式に基づいて、クラスタノード間のトラフィックをステアリングします。フローは、アクセス制御リスト（ACL）と同様の L2 パラメーターと L3 パラメーターの組み合わせによって定義されます。

PBS は、IPv4 トラフィックと IPv6 トラフィックの両方をサポートします。IPv6 展開の場合、ステアリングは追加オプション [`dfdprefix <positive_integer>`]をサポートします。同じ IP プレフィックスに対して同じフロープロセッサを選択する柔軟性を提供します。prefix オプションは、送信元 IP または宛先 IP ハッシュ方式に対してだけサポートされます。

> 注
>
> PBS メカニズムを使用してトラフィックを操縦しない場合、トラフィックはデフォルトの方式で操縦されます。

新しい ACL 属性を設定するには、CLI で次のコマンドを入力します。

IPv4 用の **CLI** コマンド

- `add ns acl <aclname> <aclaction> [-type (classic | dfd)] [-dfdhash <dfdhash>]`
- `set ns acl <aclname> <aclaction> [-dfdhash <dfdhash>`
- `show ns acl [<aclname>][-type (classic | DFD)]`
- `apply ns acls [-type (classic | DFD)]`
- `clear ns acls [-type (classic | DFD)]`
- `renumber ns acls [-type (classic | DFD)]`

IPv6 用の **CLI** コマンド

- `add ns acl6 <acl6name> <acl6action> [-type (classic | dfd)][-dfdhash <dfdhash>][-dfdprefix <positive_interger>]`
- `set ns acl6 <acl6name> <acl6action> [-dfdhash <dfdhash>][-dfdprefix <positive_interger>]`
- `show ns acl6 [<acl6name>][-type (classic | DFD)]`
- `apply ns acls6 [-type (classic | DFD)]`
- `clear ns acls6 [-type (classic | DFD)]`
- `renumber ns acls6 [-type (classic | DFD)]`

次に、パケットをフロープロセッサに操縦するために指定できるさまざまなタイプのハッシュ方式を示します。

- SIP-SPORT-DIP-DPORT
- SIP
- DIP
- SIP-DIP
- SIP-SPORT

制限事項

1. フロープロセッサは管理者が設定したルールによって決定されるため、クラスタノード間のトラフィックフローの分散は保証されません。
2. L2 モードはサポートされていません。
3. 展開シナリオがないため、ノードグループとストライプ SNIP はサポートされていません。
4. MPTCP はサポートされていません。
5. TCP、UDP、および ICMP トラフィックのみをサポートします。
6. Cluster overL3 モードはサポートされていません。
7. サービスレベルでのローカルプロセスはサポートされていません。

等コストマルチパス（**ECMP**）の使用

October 7, 2021

クラスタ展開で等価コストマルチパス (ECMP) メカニズムを使用することで、アクティブなクラスタノードは仮想サーバの IP アドレスをアドバタイズします。アドバタイズされたトラフィックを受信するクラスタノードは、トラフィックを処理する必要があるノードへのトラフィックを操縦します。スポッティングされた仮想サーバーと部分的にストライプされた仮想サーバーでは、冗長なステアリングが可能です。したがって、NetScaler 11 以降では、スポットされた仮想サーバー IP アドレスと部分的にストライプされた仮想サーバーの IP アドレスが所有者ノードをアドバタイズするため、冗長なステアリングが減少します。

ECMP を使用するには、ルーティングプロトコルに関する詳細な知識が必要です。詳細については、ダイナミックルートの設定を参照してください。クラスター内のルーティングの詳細については、「クラスター内のルーティング」を参照してください。

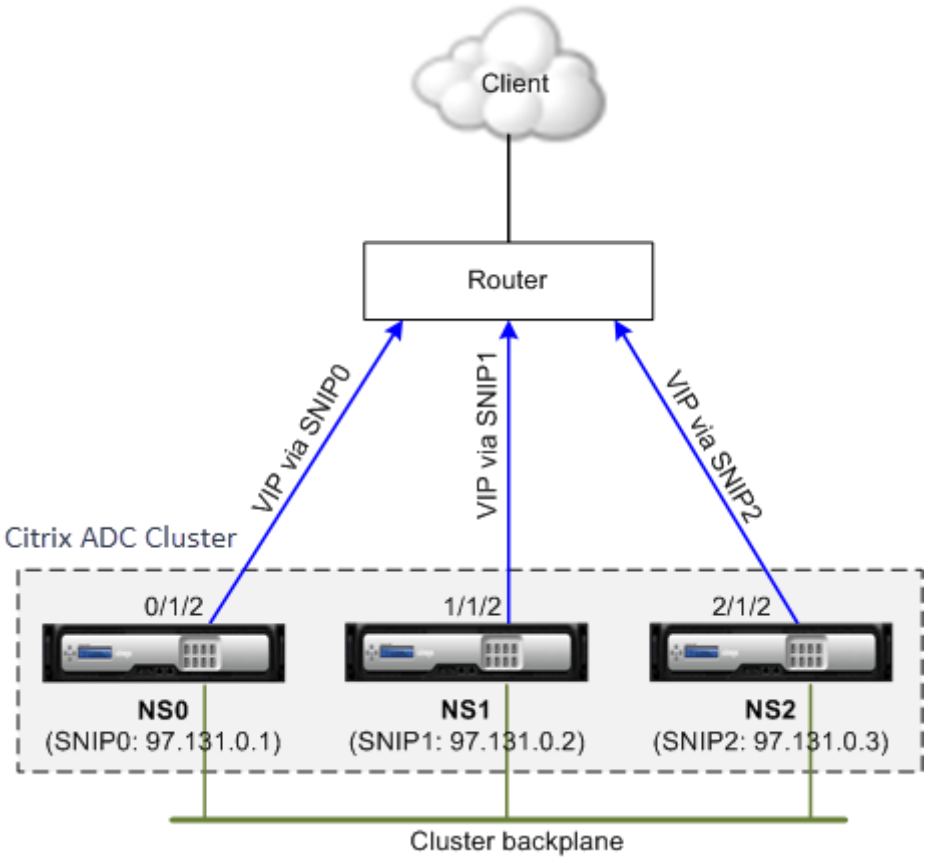
ECMP を使用するには、まず以下を実行する必要があります。

- クラスタ IP アドレスで必要なルーティングプロトコル（OSPF、RIP、BGP、または ISIS）を有効にします。
- インターフェイスとスポッティング IP アドレス（ダイナミックルーティングが有効の場合）を VLAN にバインドします。
- 選択したルーティングプロトコルを構成し、VTYSH シェルを使用して ZebOS でカーネルルートを再配布します。

クラスタ IP アドレスおよび外部接続デバイス上で同様の構成を実行します。

> 注
>
> - クラスタ上のライセンスが動的ルーティングをサポートしていることを確認します。そうでない場合、ECMP は機能しません。
> - RHI はルーターとワイルドカード仮想サーバーにアドバタイズするために VIP アドレスを必要とするため、ECMP はワイルドカード仮想サーバーではサポートされていません。VIP アドレスが関連付けられていないため。

図 1: ECMP トポロジ



クラスター展開でトラフィック分散に ECMP メカニズムを使用する場合、アクティブなクラスターノードは仮想サーバーの IP アドレスをアップストリームルーターにアドバタイズします。ECMP ルータは、SNIP0、SNIP1、または SNIP2 を介して VIP アドレスに到達できます。図 1 のトラフィックフローは、次のように説明されています。

1. クライアントは、クラスターでホストされている VIP に要求を送信します。
2. アップストリームルータは、VIP の学習ルートに基づいて、パケットをいずれかのノードに転送します。NS1 とか言いそうノード NS1 はフローレシーバです。
3. フローレシーバ（NS1）は、フロープロセッサと呼ばれるトラフィックを処理する必要があるノードを決定します。たとえば、ノード NS2 はフロープロセッサです。
4. SNIP1（97.131.0.2）を持つフローレシーバ（NS1）は、SNIP2（97.131.0.3）を持つフロープロセッサ（NS2）に要求を操縦する。
5. フロープロセッサ（NS2）は、サーバとの接続を確立します。
6. サーバーは要求を処理し、要求を送信した SNIP アドレスに応答を送信します。

メモ:

- VIP ルートをアドバタイズするのは、ACTIVE ノードだけです。
- 非アクティブノードは、VIP ルートをアドバタイズしません。
- すべての ACTIVE ノードがストライプされた VIP をアドバタイズします。
- ACTIVE オーナーノードだけが、斑点または部分的にストライプされた VIP をアドバタイズします

コマンドラインインターフェイスを使用してクラスタで **ECMP** を構成するには

1. クラスタ IP アドレスにログオンします。
2. ルーティングプロトコルを有効にします。

```
enable ns feature <feature>
```

例: OSPF ルーティングプロトコルを有効にするには。

```
enable ns feature ospf
```

3. VLAN を追加します。

```
add vlan <id>
```

例


```
1 add vlan 97
```

4. クラスタノードのインターフェイスを VLAN にバインドします。

```
1 bind vlan <id> -ifnum <interface_name>
```

例

```
1 bind vlan 97 -ifnum 0/1/2 1/1/2 2/1/2
```

5. 各ノードにスポッティング SNIP アドレスを追加し、そのノードで動的ルーティングを有効にします。

```
1 add ns ip <SNIP> <netmask> -ownerNode <positive_integer> -
      dynamicRouting ENABLED
```

例

```
1 add ns ip 97.131.0.1 255.0.0.0 -ownerNode 0 -dynamicRouting
      ENABLED -type SNIP
2 add ns ip 97.131.0.2 255.0.0.0 -ownerNode 1 -dynamicRouting
      ENABLED -type SNIP
3 add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2 -dynamicRouting
      ENABLED -type SNIP
```

6. スポットのある SNIP アドレスの 1 つを VLAN にバインドします。スポッティング SNIP アドレスを 1 つの VLAN にバインドすると、そのサブネット内のクラスタで定義されている他のすべてのスポッティング SNIP アドレスは、自動的に VLAN にバインドされます。

```
1 bind vlan <id> -IPAddress <SNIP> <netmask>
```

例

```
1 bind vlan 97 -ipAddress 97.131.0.1 255.0.0.0
```


> 注
>
> SNIP アドレスを追加する代わりに、クラスタノードの NSIP アドレスを使用できます。その場合は、手順 3 ～6 を実行する必要はありません。

7. VTYSH シェルを使用して ZebOS でルーティングプロトコルを構成します。

   例:

   ノード ID 0、1、および 2 に OSPF ルーティングプロトコルを設定します。

```
vtysh
! interface vlan97 !
 router ospf   owner-node 0
ospf router-id 97.131.0.1   exit-owner-node
owner-node 1    ospf router-id 97.131.0.2
exit-owner-node
owner-node 2
ospf router-id 97.131.0.3   exit-owner-node   redistribute kernel
   network 97.0.0.0/8 area 0  !
```

> 注
>
> VIP アドレスがアドバタイズされるためには、次のように vServerRHIlevel パラメータを使用して RHI 設定を行います。

```
add ns ip <IPAddress> <netmask> -type VIP -vserverRHILevel <
   vserverRHILevel>
```

   OSPF 固有の RHI 設定では、次のような設定が可能です。

```
add ns ip <IPAddress> <netmask> -type VIP -ospfLSAType ( TYPE1 |
   TYPE5 ) -ospfArea <positive_integer>
```

   IPv6 アドレスに対して前述のコマンドを実行するには、add ns ip6 コマンドを使用します。

8. 外部スイッチで ECMP を設定します。次に、Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチ用の設定例を示します。他のスイッチでも同様の設定を行う必要があります。

```
//For OSPF (IPv4 addresses) Global config: Configure terminal
   feature ospf    Interface config:  Configure terminal
```



```
      interface Vlan10      no shutdown      ip address 97.131.0.5/8
         Configure terminal  router ospf 1  network 97.0.0.0/8 area
      0.0.0.0  ---------------------------------
2
3 //For OSPFv3 (IPv6 addresses) Global config:  Configure terminal
      feature ospfv3    Configure terminal  interface Vlan10      no
      shutdown      ipv6 address use-link-local-only      ipv6 router
       ospfv3 1 area 0.0.0.0    Configure terminal  router ospfv3 1
```

ECMP 導入におけるクラスタ・ノードを監視するルータ

クラスタセットアップで、スポッティングされた SNIP アドレス構成を持つ所有者ノードで、ownerDownResponse オプションを無効にできるようになりました。デフォルトでは、このオプションは有効になっており、ノードはアップストリームルータからの ICMP/ARP/ICMP6/ND6 要求に応答できます。このオプションを無効にすると、クラスタノードがアクティブか非アクティブかをルータが監視できるようになります。ルータが要求を送信するときに、このオプションが無効になっている場合、ルータは所有者ノードを非アクティブにし、トラフィックの分散に使用できないことを識別します。

コマンドラインインターフェイスを使用して **ECMP** をスタティックルートトラフィック分散に設定するには

```
1 add ns ip <ipddress> <netmask> -ownernode <node-id>  - ownerDownResponse
      disable
```

ユースケース：**BGP** ルーティングを使用した **ECMP**

October 7, 2021

BGP ルーティングプロトコルを使用して ECMP を設定するには、次の手順を実行します。

1. クラスタ IP アドレスにログオンします。
2. BGP ルーティングプロトコルを有効にします。

```
1 > enable ns feature bgp
```

3. VLAN を追加し、必要なインターフェイスをバインドします。


```
1 > add vlan 985
2 > bind vlan 985 -ifnum 0/0/1 1/0/1
```

4. スポッティング IP アドレスを追加し、VLAN にバインドします。

```
1 > add ns ip 10.100.26.14 255.255.255.0 -ownerNode 1 -
     dynamicRouting ENABLED
2 > add ns ip 10.100.26.15 255.255.255.0 -ownerNode 2 -
     dynamicRouting ENABLED
3 > bind vlan 985 -ipAddress 10.100.26.10 255.255.255.0
```

5. VTYSH シェルを使用して ZebOS で BGP ルーティングプロトコルを構成します。

```
1 > vtysh conf t router bgp 65535 neighbor 10.100.26.1 remote-as
     65535
```

6. 外部スイッチで BGP を設定します。次に、Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチ用の設定例を示します。他のスイッチでも同様の設定を行う必要があります。

```
1 > router bgp 65535 no synchronization
2   bgp log-neighbor-changes neighbor 10.100.26.14 remote-as 65535
       neighbor 10.100.26.15 remote-as 65535 no auto-summary
3   dont-capability-negotiate
4   dont-capability-negotiate
5   no dynamic-capability
```

ルーティングプロトコルを使用した **Cisco Nexus 7000** スイッチを使用したクラスタ **ECMP** の設定

October 7, 2021

クラスタ構成上の ECMP を使用すると、Citrix ADC アプライアンスはルーティングプロトコルを介してトラフィックを処理できます。ECMP メカニズムは、すべてのアクティブなクラスタノードを介して仮想サーバの IP アドレスをアドバタイズするのに役立ちます。

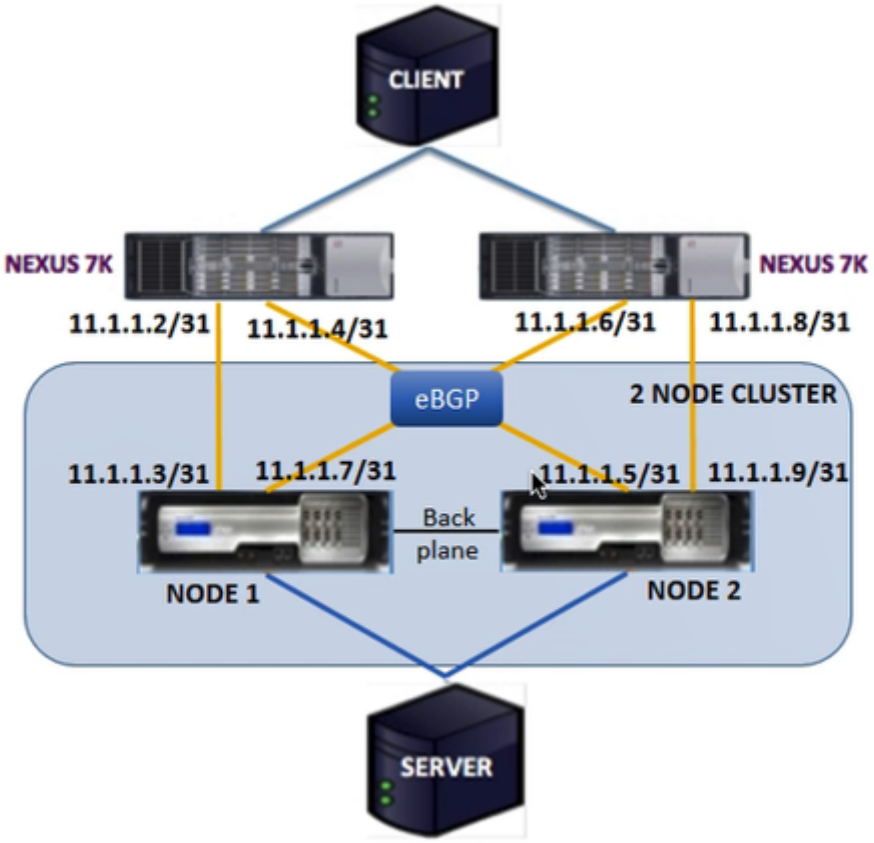
ECMP を使用するには、まずクラスタ IP アドレスで BGP プロトコルを有効にする必要があります。インターフェイスとスポッティング IP アドレス（ダイナミックルーティングが有効の場合）を VLAN にバインドします。選択した

ルーティングプロトコルを構成し、VTYSH シェルを使用して ZebOS でカーネルルートを再配布します。

ユースケース：**Cisco Nexus 7000** スイッチとルーティングプロトコルを使用したクラスタ **ECMP**

Cisco Nexus7000 スイッチを使用したクラスタ展開の例を考えてみましょう。

- Nexus スイッチ（アップストリーム）に接続された 2 つの Citrix ADC アプライアンス（ノード 1 とノード 2）
- 2 つの Cisco Nexus 7000 スイッチ.
- クライアントとサーバ（Nexus スイッチ経由の HTTP トラフィックの描画）クライアント側でホットスタンバイルータプロトコル（HSRP）が有効になっている場合。



前提条件

Citrix ADC アプライアンスでクラスタノードを構成する前に、次の点を考慮してください。

1. すべてのアプライアンスは、同じプラットフォームタイプである必要があります。
2. クラスタノードでボーダーゲートウェイプロトコル（BGP）を有効にする必要があります。

Citrix ADC アプライアンスで **CLI** を使用して構成する

1. アプライアンス（NSIP アドレス 1.1.1.1 のアプライアンスなど）にログオンします。

2. クラスターノードを追加します。

```
add cluster node 0 1.1.1.2 -state ACTIVE -backplane 0/10/8
```

3. クラスター IP アドレスを追加するには

```
add ns ip 1.1.1.10 255.255.255.254 -type clip
```

4. 構成を保存します

```
save ns config
```

5. アプライアンスのウォームリブート

```
reboot -warm
```

6. クリップ (CLIP) を使用してノード 1 を追加するには

```
add cluster node 1 2.2.2.2 -state ACTIVE -backplane 1/10/8
```

7. ノードをクラスタに参加するには

```
join cluster -clip 1.1.1.10 -password nsroot
```

8. CLIP で次の設定を行います。

   - enable ns feature bgp ospf DYNAMICROUTING
   - add ns ip 11.1.1.3 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0
   - add ns ip 11.1.1.7 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0
   - add ns ip 11.1.1.5 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1

- `add ns ip 11.1.1.9 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1`

Cisco Nexus ルーター（11.1.1.2/31 および 11.1.1.4/31）で、コマンドラインを使用して次の構成を実行する必要があります：

- `feature ospf`
- `feature bgp`
- `feature interface-vlan`
- `feature hsrp`

```
> interface vlan100
  no shutdown
  ip address 50.1.1.1/8
   hsrp 50
    ip 50.50.50.50

> interface Ethernet 4/15
   ip address 11.1.1.2/31
   no shutdown

> interface Ethernet 4/19
   ip address 11.1.1.4/31
   no shutdown

> interface Ethernet 4/22
   switchport
    switchport access vlan 100
```

Cisco Nexus ルーター（11.1.1.6/31 および 11.1.1.8/31）で、コマンドラインを使用して次の構成を実行する必要があります：

- `feature ospf`
- `feature bgp`
- `feature interface-vlan`
- `feature hsrp`

```
> interface vlan100
   no shutdown
   no ip redirects
```



```
4       ip address 50.1.1.2/8
5        hsrp 50
6        ip 50.50.50.50
7
8   >  interface Ethernet 4/13
9       ip address 11.1.1.6/31
10      no shutdown
11
12  >  interface Ethernet 4/15
13      ip address 11.1.1.8/31
14      no shutdown
15
16  >  interface Ethernet 4/22
17      switchport
18      switchport access vlan 100
```

BGP プロトコルの場合、Citrix ADC アプライアンスの CLIP で次の構成を実行する必要があります。

```
1  >  vtysh
2  ns# router bgp 1
3   redistribute kernel
4   owner-node 0
5   neighbor 11.1.1.2 remote-as 2
6   neighbor 11.1.1.2 as-origination-interval 1
7   neighbor 11.1.1.2 advertisement-interval 0
8   neighbor 11.1.1.6 remote-as 2
9   neighbor 11.1.1.6 as-origination-interval 1
10  neighbor 11.1.1.6 advertisement-interval 0
11 owner-node 1
12  neighbor 11.1.1.4 remote-as 2
13  neighbor 11.1.1.4 as-origination-interval 1
14  neighbor 11.1.1.4 advertisement-interval 0
15  neighbor 11.1.1.8 remote-as 2
16  neighbor 11.1.1.8 as-origination-interval 1
17  neighbor 11.1.1.8 advertisement-interval 0
18 exit-owner-node
```

Cisco Nexus ルータ（11.1.1.3 および 11.1.1.5）で次の設定を実行します。

```
1  > ip access-list acl1
2    10 permit ip 50.0.0.0/8 any
3    route-map test permit
```



```
4    match ip address acl1
5   router bgp 2
6    address-family ipv4 unicast
7     redistribute direct route-map test
8     maximum-paths 2
9   neighbor 11.1.1.3 remote-as 1
10   address-family ipv4 unicast
11   neighbor 11.1.1.5 remote-as 1
12  address-family ipv4 unicast
```

Cisco Nexus ルータ（11.1.1.7 および 11.1.1.9）で次の設定を実行します。

```
1  > ip access-list acl1
2     10 permit ip 50.0.0.0/8 any
3    route-map test permit 1
4     match ip address acl1
5    router bgp 2
6     address-family ipv4 unicast
7     redistribute direct route-map test
8     maximum-paths 2
9    neighbor 11.1.1.7 remote-as 1
10    address-family ipv4 unicast
11   neighbor 11.1.1.9 remote-as 1
12    address-family ipv4 unicast
```

OSPF プロトコルの場合、Citrix ADC アプライアンスの CLIP で次の構成を実行する必要があります。

```
1  > vtysh
2  ns# router osfp 1
3  redistribute kernel
4  owner-node 0
5   network 15.1.1.2/31 area 0
6   network 15.1.1.6/31 area 0
7  exit-owner-node
8
9  owner-node 1
10  network 15.1.1.4/31 area 0
11  network 15.1.1.8/31 area 0
12 exit-owner-node
13
14 route-map map2 permit 1
15 set metric 10
```


Cisco Nexus ルーター（11.1.1.2/31 および 11.1.1.4/31）で、コマンドラインを使用して次の構成を実行する必要があります：

```
> route-map- map2 permit 1
   set metric 10

   interface Ethernet4/15
     ip address 15.1.1.2/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

   interface Ethernet4/19
     ip address 15.1.1.4/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

   router ospf 1
    router-id 1.1.1.1
    redistribute direct route-map map2
```

Cisco Nexus ルータ（11.1.1.7/31 および 11.1.1.9/31）では、コマンドラインを使用して次の設定を実行する必要があります。

```
> route-map- map2 permit 1
    set metric 10

   interface Ethernet4/13
     ip address 15.1.1.6/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

    interface Ethernet4/15
     ip address 15.1.1.8/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

    router ospf 1
      router-id 1.1.1.2
      redistribute direct route-map map2
```


クラスタリンクアグリゲーションの使用

October 7, 2021

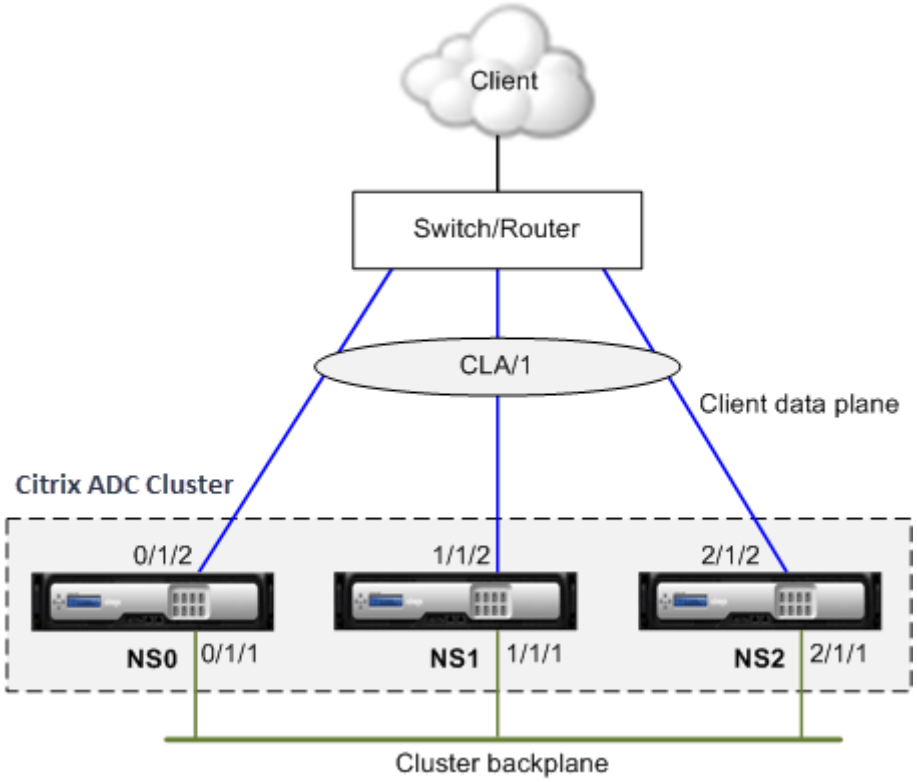
クラスターリンクアグリゲーションは、クラスターノードのインターフェイスのグループです。これは、Citrix ADC リンクアグリゲーションを拡張したものです。唯一の違いは、リンクアグリゲーションでは、同じデバイスからのインターフェイスが必要ですが、クラスタリンクアグリゲーションでは、インターフェイスはクラスタの異なるノードからのものであることです。リンクアグリゲーションの詳細については、「リンク集約の設定」を参照してください。

> 重要
>
> - クラスタリンク集約は、ハードウェア（MPX）アプライアンスのクラスタでサポートされています。
> - クラスタリンクアグリゲーションは、ESX および KVM ハイパーバイザーに展開された仮想（VPX）アプライアンスのクラスタでサポートされますが、以下の制限があります。
> - 専用インターフェースを使用する必要があります。これは、インターフェースを他の仮想マシンと共有してはならないことを意味します。
> - ノードが INACTIVE になると、対応するクラスター LA インターフェースは power DOWN としてマークされるため、データトラフィックは INACTIVE ノードに送信されません。
> - ノードが ACTIVE になると、対応するクラスター LA インターフェースは電源オンとしてマークされます。
> - クラスタリンクアグリゲーションメンバーインターフェイスが手動で無効になっている場合、またはクラスタリンクアグリゲーション自体が手動で無効になっている場合、インターフェイスのパワーダウン機能は LACP タイムアウトメカニズムによってのみ実現されます。
> - ジャンボ MTU は、LACP クラスタリンク集約ではサポートされません。
>
> 注：クラスタリンクアグリゲーションは、XenServer、AWS、および Hyper-V にデプロイされた VPX アプライアンスではサポートされません。
>
> - 12. 0 リリース以降、Citrix ADC SDX アプライアンスでクラスタリンクアグリゲーションがサポートされています。
> - クラスタ LA にバインドできるインターフェイスの数は 16 です（各ノードから）。クラスタ LA のインターフェイスの最大数は（16 * n）、ここで n はクラスター内のノードの数です。クラスタ LA のインターフェイスの総数は、アップストリームスイッチのすべてのポートチャネルのインターフェイスの数によって異なります。
> - Citrix ADC アプライアンスが IntelFortville インターフェースを使用している場合、クラスターノードをパッシブモードに切り替えると、CLAG が数秒間停止する可能性があります。この問題は、CLAG が正しく機能するように LACP が有効になっており、停止時間は NIC LACP タイマーによって異なるために発生します。

たとえば、3 つのノードすべてがアップストリームスイッチに接続されている 3 ノードのクラスタを考えてみます。クラスタ LA チャネル（CLA/1）は、インターフェイス 0/1/2、1/1/2、および 2/1/2 のバインディングによって形成

されます。

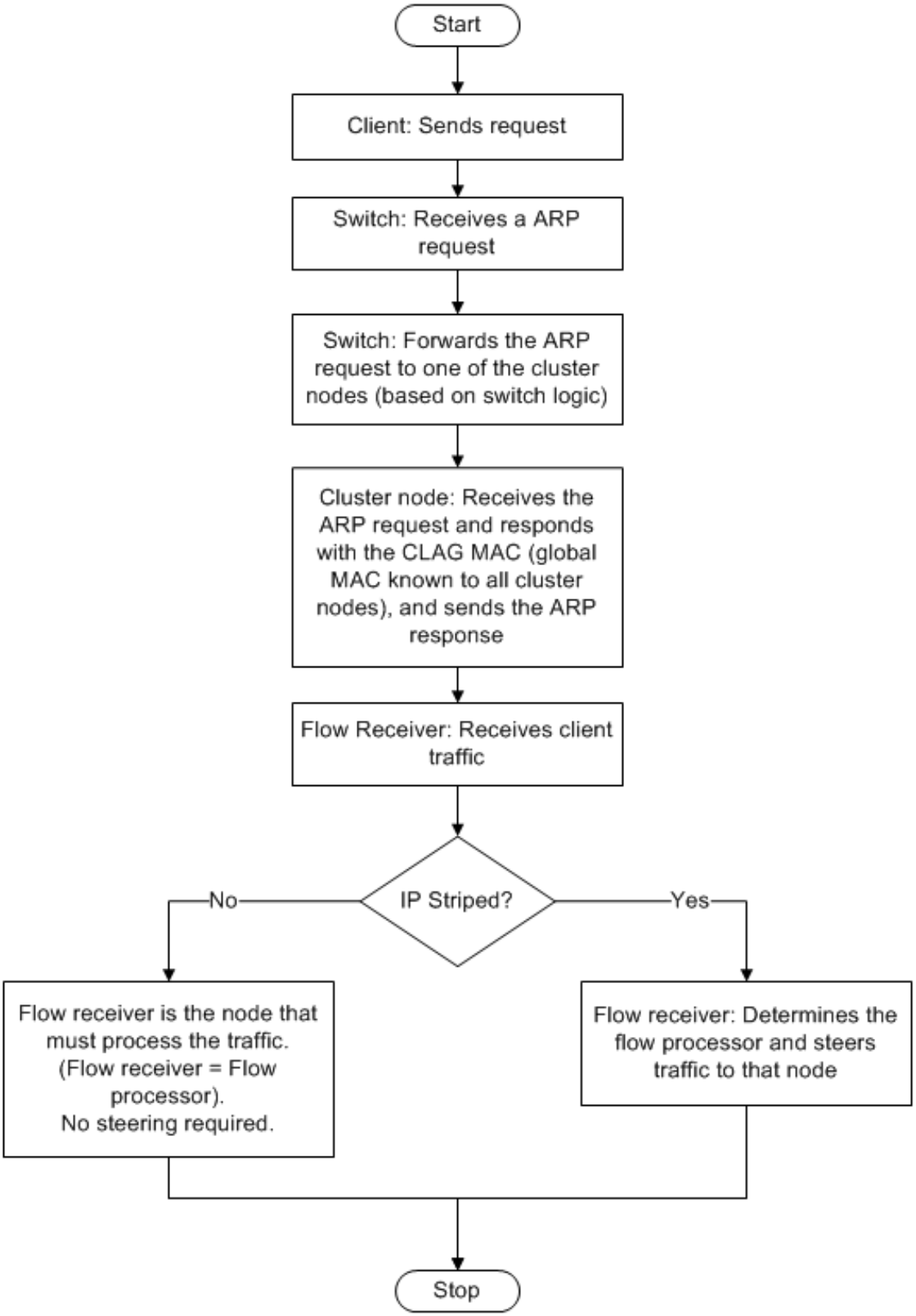
図 1: クラスタリンク集約トポロジ



クラスター LA チャネルには、次の属性があります。

- 各チャネルには、クラスタノードによって合意された一意の MAC があります。
- チャネルは、ローカルノードとリモートノードの両方のインターフェイスをバインドできます。
- クラスタでは、最大 4 つのクラスタ LA チャネルがサポートされます。
- バックプレーンインターフェイスは、クラスタ LA チャネルの一部にはできません。
- インターフェイスがクラスタ LA チャネルにバインドされている場合、チャネルパラメータはネットワークインターフェイスパラメータよりも優先されます。ネットワークインターフェイスは、1 つのチャネルにのみバインドできます。
- クラスタノードへの管理アクセスは、クラスタ LA チャネル（CLA/1 など）またはそのメンバインターフェイス上で設定しないでください。これは、ノードが INACTIVE の場合、対応するクラスター LA インターフェースがパワーダウンとしてマークされ、管理アクセスが失われるためです。

図 2: クラスタ LA を使用したトラフィック分散フロー



Citrix ADC MPX 上のクラスタ LA のバックアップとリストアのサポート

Citrix ADC MPX で LA のクラスター設定をバックアップおよび復元できます。クラスタ LA MAC アドレスは、クラスタノードの物理インターフェイス MAC アドレスとは無関係であり、バックアップおよび復元プロセスの後に変更される可能性があります。クラスタ LA は、クラスタ復元プロセスが完了した後、トラフィックを処理できます。バックアップと復元の詳細については、「クラスタ設定のバックアップと復元」を参照してください。

スタティッククラスタリンク集約

October 7, 2021

スタティッククラスタ LA チャネルは、クラスタ IP アドレスおよび外部接続デバイスで設定する必要があります。可能であれば、MAC アドレスではなく IP アドレスまたはポートに基づいてトラフィックを分散するようにアップストリームスイッチを設定します。

CLI を使用して静的クラスター LA チャネルを構成するには

1. クラスタ IP アドレスにログオンします。

   > 注
   >
   > 外部スイッチでリンク集約を設定する前に、必ずクラスタ IP アドレスにクラスタ LA チャネルを設定してください。それ以外の場合、クラスタ LA チャネルが設定されていなくても、スイッチはトラフィックをクラスタに転送します。トラフィックの損失につながる可能性があります。

2. クラスタ LA チャネルを作成します。

   ```
   add channel <id> -speed <speed>
   ```

   例

   ```
   add channel CLA/1 -speed 1000
   ```

   > 注
   >
   > 速度を AUTO として指定しないでください。代わりに、速度を 10、100、1000、または 10000 に明示的に指定する必要があります。クラスタ LA チャネルの <speed> 属性に一致する速度を持つインターフェイスのみがアクティブな配布リストに追加されます。

3. 必要なインターフェイスをクラスタ LA チャネルにバインドします。これらのインターフェイスがクラスタバックプレーンに使用されていないことを確認します。


```
bind channel <id> <ifnum>
```

例

```
bind channel CLA/1 0/1/2 1/1/2 2/1/2
```

4. 設定を確認します。

```
show channel <id>
```

例

```
show channel CLA/1
```

> 注
>
> `bind vlan` コマンドを使用して、クラスター LA チャネルを VLAN にバインドできます。チャネルのインターフェイスは、自動的に VLAN にバインドされます。

5. 外部スイッチにスタティック LA を設定します。次の設定例は、Cisco® Nexus 7000 C7010 リリース 5.2 (1) 用に提供されています。他のスイッチでも同様の設定を行う必要があります。

```
Global config:
Configure terminal

Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown

interface Ethernet2/48
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown
```


動的クラスタリンク集約

October 7, 2021

動的クラスター LA チャネルは、リンクアグリゲーション制御プロトコル（LACP）を使用します。

クラスタ IP アドレスと外部接続デバイスでも同様の設定を行う必要があります。可能であれば、MAC アドレスではなく IP アドレスまたはポートに基づいてトラフィックを分散するようにアップストリームスイッチを設定します。

> 確認事項
>
> - LACP を有効にします（LACP モードを ACTIVE または PASSIVE に指定します）。

```
>**Note**
>
> Make sure the LACP mode is not set as PASSIVE on both the Citrix ADC
    cluster and the external connecting device.
```

> - チャネルの一部にする各インターフェイスで、同じ LACP キーを指定します。クラスタ LA チャネルを作成する場合、LACP キーには 5 ～8 の値を指定できます。たとえば、インターフェイス 0/1/2、1/1/2、および 2/1/2 に LACP キーを設定すると、CLA/1 が作成されます。インターフェイス 0/1/2、1/1/2、および 2/1/2 は、自動的に CLA/1 にバインドされます。同様に、LACP キーを 6 に設定すると、CLA/2 チャネルが作成されます。
> - LAG タイプをクラスタとして指定します。

CLI を使用して動的クラスター LA チャネルを構成するには

クラスター IP アドレスで、クラスター LA チャネルに追加する各インターフェイスに対して、次のように入力します。

```
set interface <id> -lacpMode <lacpMode> -lacpKey <positive_integer> -
lagType CLUSTER<!--NeedCopy-->
```

例:

3 つのインターフェイスのクラスタ LA チャネル CLA/1 を設定します。

```
> set interface 0/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
```

> 注
>
> オプションで、LACP を使用してクラスタのリンク冗長性を有効にできます。

同様に、外部スイッチでダイナミック LA を設定します。次の設定例は、Cisco® Nexus 7000 C7010 リリース 5.2 (1) 用に提供されています。他のスイッチでも同様の設定を行う必要があります。

```
Global config:
Configure terminal
feature lacp
Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown

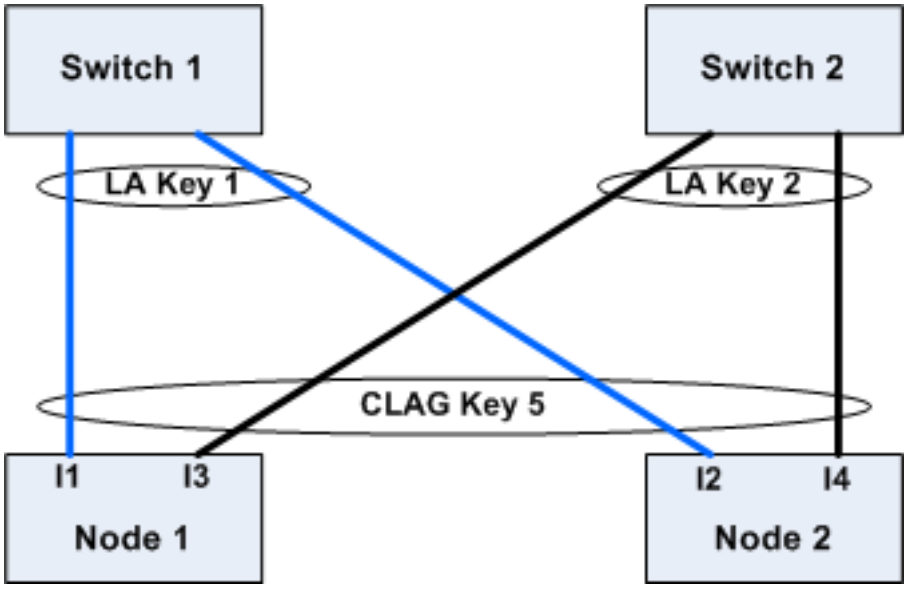
interface Ethernet2/48
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown
```

LACP を使用したクラスタ内のリンク冗長性

October 7, 2021

Citrix ADC クラスタは、LACP のリンク冗長性を提供し、すべてのノードが同じパートナーキーを持つようにします。

リンク冗長性の必要性を理解するために、次のクラスタ設定の例と付随するケース（ケース 3 に注意）を考えてみましょう。



このセットアップでは、インターフェイス I1、I2、I3、および I4 が KEY5 で LACP チャネルにバインドされます。パートナー側では、I1 と I2 がスイッチ 1 に接続され、KEY 1 で単一の LA チャネルが形成されます。同様に、I3 と I4 はスイッチ 2 に接続され、KEY 2 を持つ単一の LA チャネルを形成します。

次に、リンクの冗長性の必要性を理解するために、次のケースを検討してみましょう。

- ケース **1**: スイッチ **1** はアップで、スイッチ **2** はダウンしています。

  この場合、両方のノードのクラスタ LA は、Key2 からの LACPDU の受信を停止し、Key1 からの LACPDU の受信を開始します。両方のノードで、クラスター LA は KEY 1 と I1 に接続され、I2 は UP であり、両方のノードのチャネルは UP になります。

- ケース **2**: スイッチ **1** がダウンし、スイッチ **2** が **UP** になる

  この場合、両方のノードのクラスタ LA は、Key1 からの LACPDU の受信を停止し、Key2 からの LACPDU の受信を開始します。両方のノードで、クラスター LA は Key2 と I3 に接続され、I4 は UP であり、両方のノードのチャネルは UP になります。

- ケース **3**: スイッチ **1** とスイッチ **2** の両方が稼動している

  この場合、ノード 1 のクラスター LA がパートナーとして Key1 を選択し、ノード 2 のクラスター LA がパートナーとして Key2 を選択する可能性があります。これは、ノード 1 の I1 とノード 2 の I4 が望ましくないトラフィックを受信していることを意味します。LACP ステートマシンがノードレベルであり、先着順でパートナーを選択するために発生する可能性があります。

  これらの問題を解決するために、ダイナミッククラスタ LA のリンク冗長性がサポートされています。チャネルまたはインターフェイスでリンク冗長性を設定するには、リンク冗長性を有効にし、オプションでしきい値のスループットを次のように指定する必要があります。

  ```
  set channel CLA/1 -linkRedundancy ON -lrMinThroughput <positive_integer
  >
  ```


パートナーチャネルのスループットは、設定されたしきい値のスループットに対してチェックされます。しきい値スループットを満たすパートナーチャネルは、先入れ先出し（FIFO）方式で選択されます。しきい値を満たすパートナーチャネルがない場合、またはしきい値のスループットが設定されていない場合は、リンク数が最大であるパートナーチャネルが選択されます。

> 注
>
> しきい値のスループットは、NetScaler 11 以降で構成できます。

クラスタでの **USIP** モードの使用

October 7, 2021

ソース IP（USIP）モードでは、クラスタまたは Citrix ADC アプライアンスが各パケットをクライアント IP アドレスで適切なバックエンドサーバーに転送します。

USIP モードのトラフィック分散

USIP モードの動作は、ECMP および CLAG 展開では、クライアントデータプレーンとサーバデータプレーン間でトラフィックの分散が異なります。次のセクションでは、USIP モードの動作について詳しく説明します。USIP モードでの CLAG の詳細については、「クラスタリンク集約の使用」を参照してください。

USIP モード

クラスターは、クライアント IP を使用してサーバー側接続を開きます。送信元ポートは、`useproxyport` 設定に基づいて保持される場合とされない場合があります。

USIP `useproxyport` シナリオ

USIP `useproxyport` がトラフィックフローで ON の場合、送信元ポートはリバーストラフィックがフロープロセッサにハッシュされるように選択されます。サーバー側でシングルステアリングを確実にします。

USIP `useproxyport` はトラフィックフローに対してオフになり、送信元ポートは保持されるため、サーバ側にダブルステアリングがあります。

> 重要
>
> - USIP がオンの場合、クライアント IP はバックエンドサーバー接続で使用され、応答のためのトラフィック分散がクラスタノード間で必要になります。ECMP または CLAG 展開は、サーバ側のトラフィック分散に使用できます。サーバー側でトラフィックが分散されていない場合、リターントラフィック全体が単一のクラスタノードに着陸し、輻輳が発生する可能性があります。
> - `set rsskeytype -rsskey symmetric`コマンドは、`useproxyport`オフ展開でのトラフィッ

> クのダブルステアリングをシングルステアリングに減らすために使用されます。サーバー側とクライアント側で、接続の 4 タプルが同じままです。たとえば、ワイルドカード MAC モード仮想サーバーなどです。

制限事項

プロセスローカルが無効になっている場合、USIP は機能しません。

USIP モードデプロイ

次の図は、クラスタセットアップでの USIP モードの展開を示しています。

CLI を使用して次の設定を行います

1. ルーティングプロトコルを有効にします。

```
enable ns feature <feature>
```

例:

```
enable ns feature ospf
```

2. 各ノードにスポッティング SNIP アドレスを追加し、そのノードで動的ルーティングを有効にします。

```
add ns ip <SNIP> <netmask> -dynamicRouting ( ENABLED | DISABLED )
    - ownerNode <positive_interger>  - ownerdownResponse ( YES | NO
    )
```

例

```
-  add ns ip 192.0.2.1 255.255.255.0 -dynamicRouting ENABLED  -
   ownerNode 0  - ownerDownResponse NO
-  add ns ip 192.0.2.2 255.255.255.0 -dynamicRouting ENABLED  -
   ownerNode 1  - ownerDownResponse NO
-  add ns ip 192.0.2.3 255.255.255.0 -dynamicRouting ENABLED  -
   ownerNode 2  - ownerDownResponse NO
```

3. VLAN を追加します。


```
add vlan <id>
```

例

```
add vlan 300
```

4. クラスタノードのインターフェイスを VLAN にバインドします。

```
bind vlan <id> -ifnum <interface_name>
```

例

```
bind vlan 300 -ifnum 0/1/2 1/1/2 2/1/2
```

5. スポットのある SNIP アドレスの 1 つを VLAN にバインドします。スポッティング SNIP アドレスを 1 つの VLAN にバインドすると、そのサブネット内のクラスタで定義されている他のすべてのスポッティング SNIP アドレスは、自動的に VLAN にバインドされます。

```
bind vlan <id> -IPAddress <ip_addr | ipv6_addr> -netmask
```

例

```
bind vlan 300  - IPAddress 192.0.2.1 255.255.255.0
```

6. VTYSH シェルを使用して ZebOS でルーティングプロトコルを構成します。ノード ID 0、1、および 2 に OSPF ルーティングプロトコルを設定します。

```
vtysh
configure terminal
ns block-sec-rtadv
router ospf
owner -node 0
router-id 192.0.2.1
exit-owner-node
owner-node 1
```



```
router-id 192.0.2.2
exit-owner-node
owner-node 2
router-id 192.0.2.3
exit-owner-node
network 192.0.2.0/24 area 0

default-information originate always
```

7. CLI を使用して、Cisco 3750 ルータで次の設定を実行します。

```
Configure terminal
feature ospf
interface vlan300
no shutdown
ip address 192.0.2.100/24
Configure terminal
router ospf 1
router-id 192.0.2.100
network 192.0.2.0 0.0.0.255 area 0
```

> メモ
>
> - クライアントとサーバーでのトラフィックの分散は同じである必要はありません。たとえば、クライアント側で ECMP を設定し、サーバ側に CLAG を設定したり、反対の方法で設定したりできます。
> - USIP 展開ではステアリングオーバーヘッドが増えるため、バックプレーンの追加容量を計画してください。
> - CLAG およびモニタスタティックルート（MSR）に関連する設定は、サーバ側で同じままにしておく必要があります。
> - USIP モードの展開では、トラフィックステアリングがさらに高くなります。

Citrix ADC クラスタの管理

October 7, 2021

クラスタを作成し、必要なトラフィック分散メカニズムを設定したら、クラスタはトラフィックを処理できます。クラスターの存続期間中、次のクラスタータスクを実行できます。

- ノードグループの構成
- クラスターのノードを無効にする

- Citrix ADC アプライアンスの検出
- 統計の表示
- クラスター構成とクラスターファイルの同期
- ノード間で時間を同期する
- クラスタノードのソフトウェアのアップグレードまたはダウングレード

リンクセットの構成

October 7, 2021

リンクセットは、同じブロードキャストドメインに属するクラスターノードのインターフェイスのグループです。リンクセットでは、各ノードには、他のノードのどのインターフェイスが同じブロードキャストドメインに接続されているかに関する情報があります。

> 注
>
> 次のシナリオでは、リンクセットは必須の設定です。
>
> - MAC ベースフォワーディング（MBF）が必要な配置の場合。
> - 仮想サーバで有効になっている「-m MAC」モードと MBF モードがグローバルに有効になっている。
> - インターフェイスに関係する ACL および L2 ポリシーの管理性を向上させるため。インターフェイスのリンクセットを定義し、リンクセットに基づいて ACL および L2 ポリシーを追加します。

クラスター設定では、次の機能は内部で MBF を使用します。

- 転送セッション
- L2Conn
- MAC モード仮想サーバー
- 透明モニター
- LLB

リンクセットは、クラスター IP アドレスを使用してのみ構成する必要があります。

3 ノードクラスターの例を考えてみましょう。次の図では、インターフェイス 0/1/2, 1/1/2, そして 2/1/2 同じブロードキャストドメインにあるため、リンクセットとして構成できます (LS/1).

図 1: リンクセットトポロジ

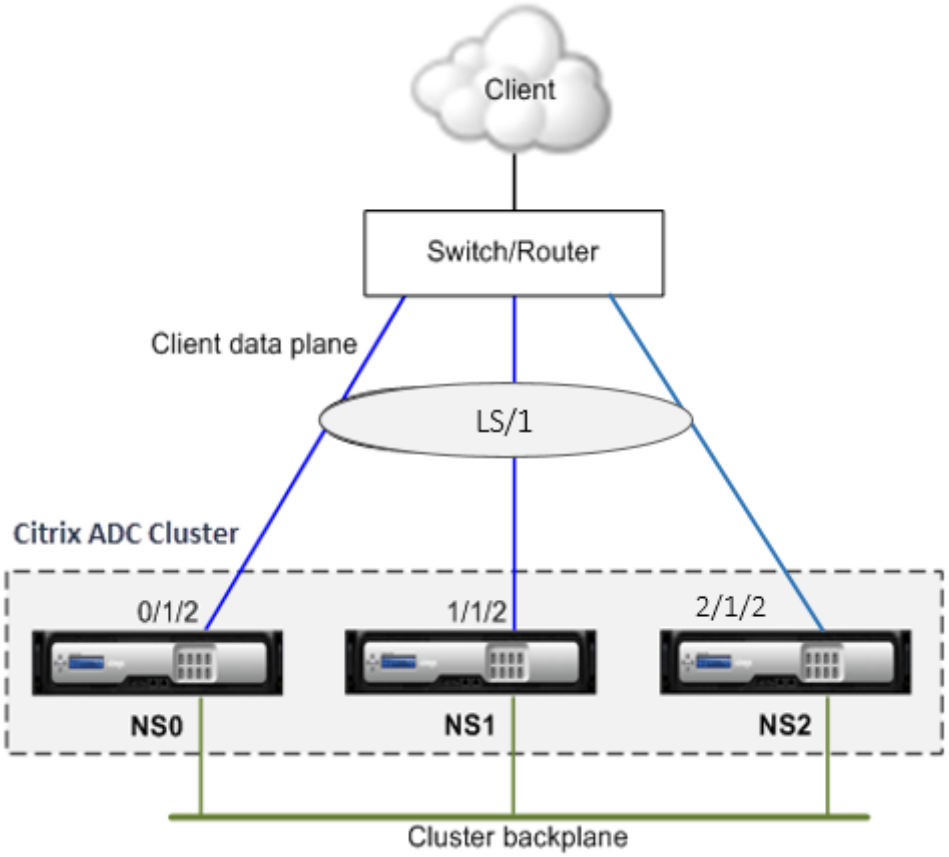
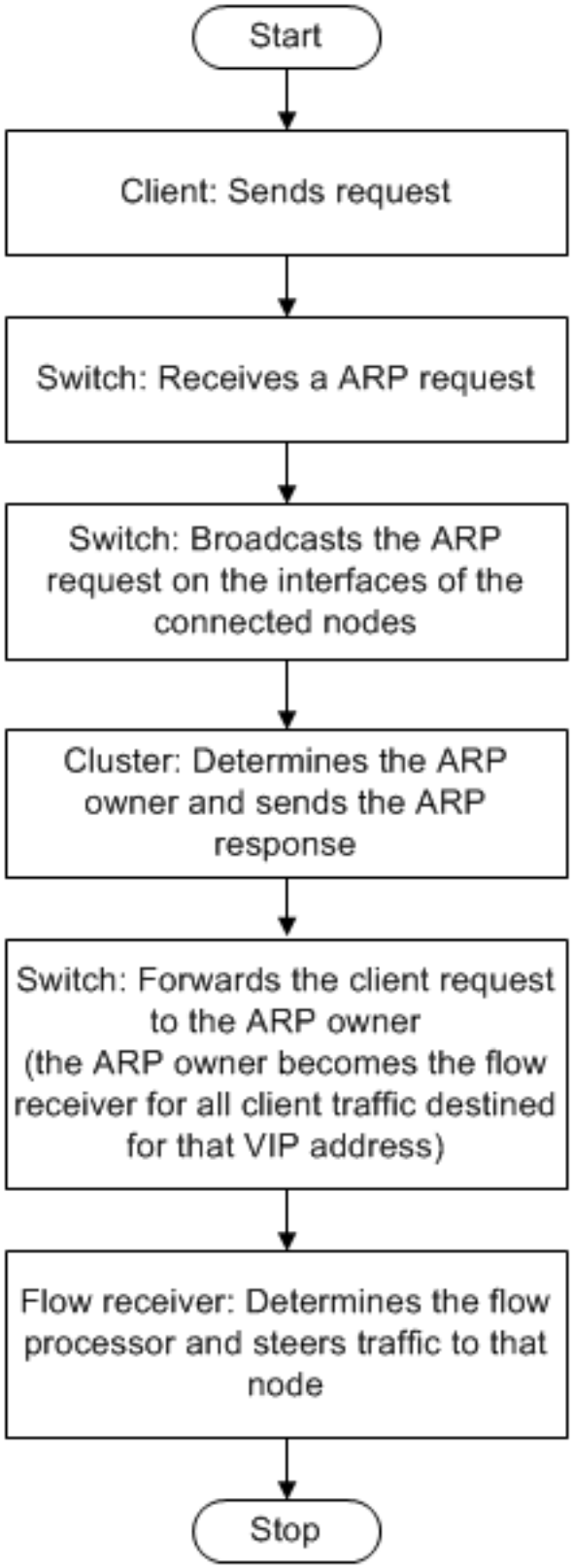


図 2: リンクセットを使用したトラフィック分散フロー



CLI を使用してリンクセットを構成するには

1. クラスタ IP アドレスにログオンします。
2. リンクセットを作成します。

   “‘add linkset

```
** 例 **

```add linkset LS/1<!--NeedCopy-->
```

3. 必要なインターフェースをリンクセットにバインドします。インターフェイスがクラスタバックプレーンに使用されていないことを確認します。

   “‘bind linkset -ifnum ...

```
** 例 **

```bind linkset LS/1 -ifnum 0/1/2 1/1/2 2/1/2<!--NeedCopy-->
```

4. リンクセットの設定を確認します。

   “‘show linkset

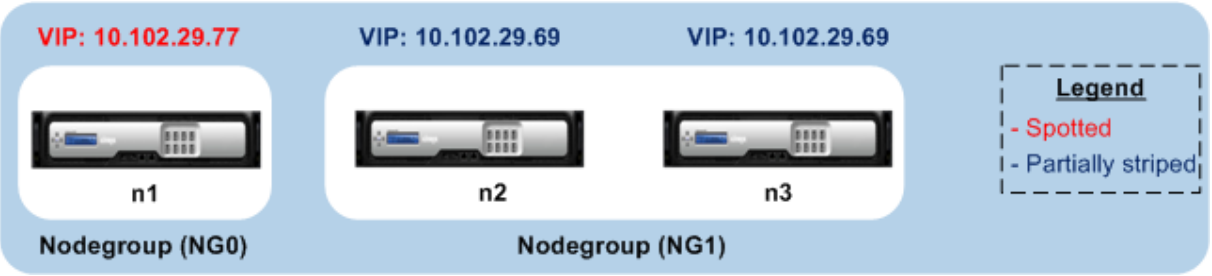
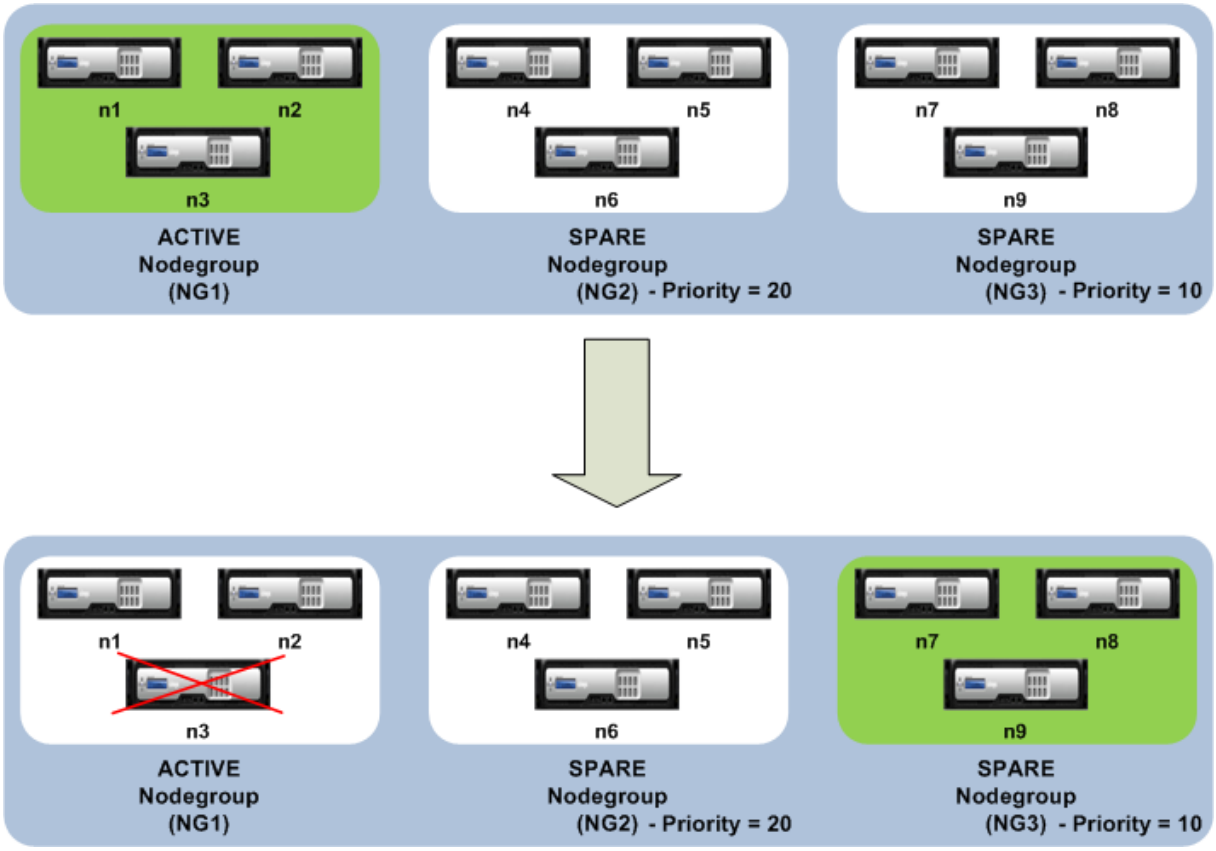
```
** 例 **

```show linkset LS/1<!--NeedCopy-->
```

> 注
>
> `bind vlan` コマンドを使用して、リンクセットを VLAN にバインドできます。リンクセットのインターフェイスは、自動的に VLAN にバインドされます。

GUI を使用してリンクセットを構成するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [ネットワーク] **>** [リンクセット] に移動します。
3. 詳細ペインで、[**Add**] をクリックします。
4. [リンクセットを作成] ダイアログボックスで、次の操作を行います。
   - Link set パラメータを設定して、リンクセットの名前を指定します。


```

- リンクセットに追加するインターフェイスを指定し、[Add] をクリックします。リンクセットに追加するインターフェイスごとに、この手順を繰り返します。

5. [**Create**] をクリックしてから、[**Close**] をクリックします。

## スポット構成および部分ストライプ構成のノードグループ

October 7, 2021

デフォルトのクラスタ動作により、クラスタ IP アドレスで実行されるすべての構成は、クラスタのすべてのノードで使用できます。ただし、特定のクラスターノードでのみ使用できる構成が必要な場合があります。

この要件を達成するには、特定のクラスターノードを含むノードグループを定義してから、そのノードグループに構成をバインドします。これにより、構成がそれらのクラスターノードでのみアクティブになります。これらの構成は、部分的なストライプまたはスポットと呼ばれます（アクティブな場合は単一ノードが 1 つだけ）。詳細については、ストライプ、部分的にストライプ、およびスポッティングされた設定を参照してください。

たとえば、3 つのノードを持つクラスターを考えてみましょう。ノード n1 を含むノードグループ NG0 と、n2 と n3 を含む別のノードグループ NG1 を作成します。負荷分散仮想サーバー 0.77 を NG0 にバインドし、負荷分散仮想サーバー 0.69 を NG1 にバインドします。

これは、仮想サーバー 0.77 が n1 でのみアクティブであるため、n1 のみが 0.77 に向けられたトラフィックを受信することを意味します。同様に、仮想サーバー 0.69 はノード n2 と n3 でのみアクティブであるため、n2 と n3 のみが 0.69 に向けられたトラフィックを受信します。

図 1: スポット構成および部分ストライプ構成用に構成されたノードグループを備えた Citrix ADC クラスター




ノードグループにバインドできるエンティティまたは構成は次のとおりです。

- 負荷分散、コンテンツの切り替え、キャッシュのリダイレクト、認証、承認、および仮想サーバーの監査

  > 注
  >
  > FTP 負荷分散仮想サーバーをノードグループにバインドすることはできません。

- VPN 仮想サーバー（NetScaler 10.5 ビルド 50.10 以降でサポートされています）
- グローバルサーバー負荷分散（GSLB）サイトおよびその他の GSLB エンティティ（NetScaler 10.5 ビルド 52.11 以降でサポートされています）
- 制限識別子とストリーム識別子

 

# ノードグループの動作

October 7, 2021

さまざまな Citrix ADC 機能およびエンティティとのノードグループの相互運用性のため、注意すべきいくつかの動作面があります。ノードグループ内のノードもバックアップできます。詳細については、以下をお読みください。

## クラスタノードグループの一般的な動作

- エンティティがバインドされているノードグループは削除できません。
- エンティティがバインドされているノードグループに属するクラスターノードは削除できません。
- エンティティがバインドされたノードグループを持つクラスターインスタンスは削除できません。
- 別のエンティティに依存しているエンティティを追加することはできません。ノードグループの一部であってはなりません。そうする必要がある場合は、最初に依存関係を削除します。次に、両方のエンティティをノードグループに追加し、エンティティを再度関連付けます。

  例:

  - バックアップが仮想サーバー VS2 である仮想サーバー VS1 があるとします。VS1 をノードグループに追加するには、最初に VS2 が VS1 のバックアップサーバーとして削除されていることを確認します。次に、各サーバーを個別にノードグループにバインドし、VS2 を VS1 のバックアップとして構成します。
  - ターゲットロードバランシング仮想サーバーが LBVS1 であるコンテンツスイッチ仮想サーバー CSVS1 があるとします。CSVS1 をノードグループに追加するには、最初にターゲットとして LBVS1 を削除します。次に、各サーバーを個別にノードグループにバインドし、LBVS1 をターゲットとして構成します。
  - 別の負荷分散仮想サーバー LBVS2 を呼び出すポリシーを持つ、負荷分散仮想サーバー LBVS1 があるとします。いずれかの仮想サーバーを追加するには、まず関連付けを削除します。次に、各サーバーを個別にノードグループにバインドしてから、仮想サーバーを再度関連付けます。
- エンティティをノードグループにバインドすることはできません。ノードがなく、strict オプションが有効になっています。したがって、エンティティがバインドされていて、strict オプションが有効になっているノードグループの最後のノードのバインドを解除することはできません。
- strict オプションは、ノードがないがエンティティがバインドされているノードグループには変更できません。

## ノードグループ内のノードのバックアップ

デフォルトでは、ノードグループは、ノードグループのメンバーにバックアップノードを提供するように設計されています。ノードグループのメンバーがダウンした場合、ノードグループのメンバーではないクラスターノードが、障害が発生したノードを動的に置き換えます。このノードは、置換ノードと呼ばれます。

 

> 注
>
> 単一メンバーのノードグループの場合、エンティティがノードグループにバインドされると、バックアップノードが自動的に事前選択されます。

ノードグループの元のメンバーが起動すると、デフォルトでは、置換ノードが元のメンバーノードに置き換えられます。

ただし、NetScaler 10.5 ビルド 50.10 以降では、Citrix ADC ではこの置換動作を変更できます。sticky オプションを有効にすると、元のメンバーノードが起動した後も置換ノードは保持されます。元のノードは、置換ノードがダウンしたときにのみ引き継がれます。

バックアップ機能を無効にすることもできます。これを行うには、strict オプションを有効にする必要があります。このシナリオでは、ノードグループメンバーがダウンすると、他のクラスターノードはバックアップノードとして選択されません。元のノードは、起動時にノードグループの一部であり続けます。このオプションにより、ノードグループにバインドされたエンティティがノードグループメンバーでのみアクティブになります。

> 注
>
> strict and sticky オプションは、ノードグループを作成するときにのみ設定できます。

## スポット構成および部分ストライプ構成のノードグループの構成

October 7, 2021

スポット構成および部分ストライプ構成のノードグループを構成するには、最初にノードグループを作成してから、必要なノードをノードグループにバインドする必要があります。次に、必要なエンティティをそのノードグループに関連付けます。ノードグループにバインドされているエンティティは次のとおりです。

- **Spotted-** 単一ノードを持つノードグループにバインドされている場合。
- 部分的にストライプ -複数のノードを持つノードグループにバインドされている場合。

覚えておくべきポイント:

- GSLB は、GSLB サイトが単一のクラスターノードを持つノードグループにバインドされている場合にのみ、クラスターでサポートされます。詳細については、「クラスタ内の GSLB の設定」を参照してください。
- Citrix Gateway は、VPN 仮想サーバーが単一のクラスターノードを持つノードグループにバインドされている場合にのみ、クラスターでサポートされます。スティッキーオプションは、ノードグループで有効にする必要があります。
- NetScaler 11 より前のバージョンの場合、アプリケーションファイアウォールは個々のクラスターノードでのみサポートされます（スポット構成）。アプリケーションファイアウォールプロファイルは、単一のクラスターノードを持つノードグループにバインドされている仮想サーバーにのみ関連付けることができます。これは、次のことを許可されていないアプリケーションを意味します。
  - アプリケーションファイアウォールプロファイルをストライピングされた仮想サーバーまたは部分的にストライピングされた仮想サーバーにバインドします。

 

  - グローバルバインドポイントまたはユーザー定義のポリシーラベルにポリシーをバインドします。
  - ノードグループから、アプリケーションファイアウォールプロファイルを持つ仮想サーバーのバインドを解除します。
- NetScaler 11 では、ストライプ構成と部分ストライプ構成のアプリケーションファイアウォールサポートが導入されました。詳細については、「クラスタ構成に対するアプリケーションファイアウォールのサポート」を参照してください。

クラスタでサポートされる Citrix ADC 機能をチェックして、クラスタで GSLB 、Citrix Gateway、およびアプリケーションファイアウォールがサポートされている NetScaler バージョンを確認します。

コマンドラインインターフェイスを使用してノードグループを構成するには

1. クラスタ IP アドレスにログオンします。
2. ノードグループを作成します。タイプ:

   ```
   add cluster nodegroup <name> -strict (YES | NO)<!--NeedCopy-->
   ```

   例

   ```
   1 add cluster nodegroup NG0 -strict YES
   ```

3. 必要なノードをノードグループにバインドします。ノードグループのメンバーごとに次のコマンドを入力します。

   ```
   bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
   ```

   例

   ID が 1、5、および 6 のノードをバインドします。

   ```
   1 > bind cluster nodegroup NG0 -node 1
   2 > bind cluster nodegroup NG0 -node 5
   3 > bind cluster nodegroup NG0 -node 6
   ```

4. エンティティをノードグループにバインドします。バインドするエンティティごとに、次のコマンドを 1 回入力します。

   ```
   bind cluster nodegroup <name> (-vServer <string> | -identifierName <
   string> | -gslbSite <string> -service <string>)<!--NeedCopy-->
   ```

   > 注
   >
   > gslbSite パラメータとサービスパラメータは、NetScaler 10.5 以降で使用できます。

例

仮想サーバ VS1 と VS2 をバインドし、識別子 1 という名前のレート制限識別子をバインドします。

```
1 > bind cluster nodegroup NG0 -vServer VS1
2 > bind cluster nodegroup NG0 -vServer VS2
3 > bind cluster nodegroup NG0 -identifierName identifier1
```

5. ノードグループの詳細を表示して、構成を確認します。タイプ:

   ```
   show cluster nodegroup <name><!--NeedCopy-->
   ```

   例

   ```
   1 > show cluster nodegroup NG0
   ```

構成ユーティリティを使用してノードグループを構成するには

1. クラスタ IP アドレスにログオンします。
2. [システム] > [クラスタ] > [ノードグループ] に移動します。
3. 詳細ウィンドウで、[追加] をクリックします。
4. [ノードグループの作成] ダイアログボックスで、ノードグループを構成します。
   a) [クラスタノード] で、[追加] ボタンをクリックします。
      - 「使用可能」リストには、ノード・グループにバインドできるノードが表示され、「構成済み」リストには、ノード・グループにバインドされているノードが表示されます。
      - [使用可能] リストの [+] 記号をクリックして、ノードをバインドします。同様に、[構成済み] リストの [-] 記号をクリックして、ノードのバインドを解除します。

   b) [仮想サーバー] で、ノードグループにバインドする仮想サーバーのタイプに対応するタブを選択します。[追加] をクリックします。
      - 「使用可能」リストには、ノード・グループにバインドできる仮想サーバーが表示され、「構成済み」リストには、ノード・グループにバインドされている仮想サーバーが表示されます。
      - [使用可能] リストの [+] 記号をクリックして、仮想サーバをバインドします。同様に、[構成済み] リストの [-] 記号をクリックして、仮想サーバのバインドを解除します。

ノードグループの冗長性の構成

October 7, 2021

> 注
>
> NetScaler 10.5 ビルド 52.1115.e 以降でサポートされています。

ノードグループは、1 つのノードグループがダウンしたときに、別のノードグループがトラフィックを引き継いで処理できるように構成できます。たとえば、ノードグループ NG1 がダウンすると、NG2 が引き継ぎます。

> 注
>
> この機能を使用して、各ノードグループがデータセンターとして構成されているデータセンターの冗長性を構成できます。

このユースケースを実現するには、クラスターノードをノードグループに論理的にグループ化する必要があります。この場合、一部のノードグループはアクティブとして構成され、他のノードグループはスペアとして構成されます。優先度が最も高い（つまり、優先度が最も低い）アクティブノードグループは、操作上アクティブになり、トラフィックを処理します。この操作上アクティブなノードグループのノードがダウンすると、このノードグループのノード数が、他のアクティブなノードグループのノード数と優先度の順に比較されます。ノードグループのノード数がそれ以上の場合、そのノードグループは操作上アクティブになります。それ以外の場合は、スペアノードグループがチェックされます。

> 注
>
> - 特定の時点でアクティブにできるのは、1 つの状態固有のノードグループのみです。
> - クラスタノードは、ノードグループの状態を継承します。したがって、「SPARE」状態のノードが「ACTIVE」状態のノードグループに追加されると、そのノードは自動的にアクティブノードとして動作します。
> - クラスタインスタンスに定義されているプリエンプションパラメータは、最初のアクティブノードグループが再び起動したときに制御を取得するかどうかを決定します。
> - スペアノードグループは、アクティブなノードグループがダウンしたときに、ノードグループとホストアクティブなトラフィックを占有できます。

次の図は、ノードグループの冗長性が定義されているノードグループのセットアップを示しています。NG1 は、最初はアクティブノードグループです。ノードの 1 つが失われると、優先度が最も高いスペアノードグループ（NG3）がトラフィックの処理を開始します。

図 1: ノードグループの冗長性が構成された Citrix ADC クラスター。



ノードグループの冗長性の構成

1. クラスタ IP アドレスにログオンします。
2. アクティブノードグループを作成し、必要なクラスターノードをバインドします。

```
1 > add cluster nodegroup NG1 -state ACTIVE
2 > bind cluster nodegroup NG1 -node n1
3 > bind cluster nodegroup NG1 -node n2
4 > bind cluster nodegroup NG1 -node n3
```

3. スペアノードグループを作成し、必要なノードをバインドします。

```
1 > add cluster nodegroup NG2 -state SPARE -priority 20
2 > bind cluster nodegroup NG2 -node n4
3 > bind cluster nodegroup NG2 -node n5
4 > bind cluster nodegroup NG2 -node n6
```

4. 別のスペアノードグループを作成し、必要なノードをバインドします。


```
> add cluster nodegroup NG3 -state SPARE -priority 10
> bind cluster nodegroup NG3 -node n7
> bind cluster nodegroup NG3 -node n8
> bind cluster nodegroup NG3 -node n9
```

クラスタバックプレーンのステアリングを無効にする

October 7, 2021

> 注
>
> NetScaler 11 以降でサポートされています。

Citrix ADC クラスターのデフォルトの動作は、受信したトラフィック（フローレシーバー）を別のノード（フロープロセッサー）に転送することです。次に、フロープロセッサはトラフィックを処理する必要があります。フロープロセッサへのフロー受信機からのトラフィックを指示するこのプロセスは、クラスタバックプレーン上で発生し、ステアリングと呼ばれています。

必要に応じて、ステアリングを無効にして、プロセスがフローレシーバーに対してローカルになり、フローレシーバーをフロープロセッサにすることができます。このような構成設定は、高遅延リンクがある場合に便利です。

> 注
>
> この構成は、ストライピングされた仮想サーバにのみ適用できます。

- 部分的にストライプ化された仮想サーバーの場合、フローレシーバーが非所有者ノードの場合、トラフィックは所有者ノードに誘導されます。ただし、フローレシーバーが所有者ノードである場合、ステアリングは無効になります。
- スポット仮想サーバーの場合、フローレシーバーはフロープロセッサーであるため、ステアリングの必要はありません。

ステアリング機構を無効にするときに覚えておくべきポイント：

- ステアリングが無効になっているため、ストライプ SNIP はサポートされません。
- MPTCP と FTP は機能しません。
- L2 モードは無効にする必要があります。
- USIP が有効になっている場合、ステアリングが無効になっているため、トラフィックが同じノードに戻らない可能性があります。
- クラスタ IP アドレスに送信されるトラフィックは、構成コーディネータに誘導されます。
- ノードがクラスターに参加またはクラスターから脱退する場合、1/N 接続が影響を受けます。これは、使用可能なノードを変更すると、ルートが再ハッシュされる可能性があるためです。その結果、トラフィックは別のノードにルーティングされ、ステアリングが使用できないため、トラフィックは処理されません。

ステアリングは、個々の仮想サーバレベルまたはグローバルレベルで無効にできます。グローバル構成は、仮想サーバ設定よりも優先されます。

- すべてのストライプ仮想サーバのバックプレーンステアリングの無効化

  クラスタインスタンスレベルで設定されます。ストライプ化された仮想サーバー向けのトラフィックは、クラスターバックプレーンでは操作されません。

  ```
  add cluster instance \<clId\> -processLocal ENABLED<!--NeedCopy-->
  ```

- 特定のストライプ仮想サーバのバックプレーンステアリングの無効化

  ストライピングされた仮想サーバ上で構成されます。仮想サーバー向けのトラフィックは、クラスターバックプレーンでは操作されません。

  ```
  add lb vserver <name> <serviceType> -processLocal ENABLED<!--NeedCopy
  -->
  ```

クラスタ構成の同期

October 7, 2021

構成コーディネーターで使用できる Citrix ADC 構成は、以下の場合にクラスターの他のノードと同期されます。

- ノードがクラスタに参加する
- ノードがクラスタに再参加する
- クラスタ IP アドレスを介して新しいコマンドが実行されます

また、構成コーディネーターで使用可能な構成を特定のクラスターノードに強制的に同期（完全同期）することもできます。一度に 1 つのクラスタノードを同期するようにしてください。同期しないと、クラスタが影響を受けます。

CLI を使用してクラスター構成を同期するには

設定を同期するアプライアンスのコマンド・プロンプトで、次のように入力します。

```
1 force cluster sync
```

GUI を使用してクラスター構成を同期するには

1. 設定を同期するアプライアンスにログオンします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインの [ユーティリティ] で、[クラスター同期の強制] をクリックします。
4. 「**OK**」をクリックします。

クラスタノード間で時刻を同期する

October 7, 2021

クラスターは、Precision Time Protocol（PTP）を使用して、クラスターノード間で時間を同期します。PTP は、マルチキャストパケットを使用して時刻を同期させます。時刻同期に問題がある場合は、PTP を無効にして、クラスタでネットワークタイムプロトコル (NTP) を構成する必要があります。

コマンドラインインターフェイスを使用して **PTP** を有効/無効にするには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
set ptp -state disable
```

構成ユーティリティを使用して **PTP** を有効/無効にするには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインの [ユーティリティ] で、[**PTP** 設定の構成] をクリックします。
4. [**PTP** の有効化/無効化] ダイアログボックスで、PTP を有効にするか無効にするかを選択します。
5.「**OK**」をクリックします。

クラスタ・ファイルの同期

October 7, 2021

構成コーディネータで使用可能なファイルは、クラスタファイルと呼ばれます。これらのファイルは、ノードがクラスターに追加されると、他のクラスターノードで自動的に同期され、クラスターの存続期間中に定期的に同期されます。また、クラスターファイルを手動で同期することもできます。

重要：クラスター環境で証明書またはキーファイルを削除すると、ADC アプライアンスでのそれ以上の構成が制限されます。同じ場所にファイルを追加して戻し、構成を変更します。

同期される構成コーディネータのディレクトリとファイルは次のとおりです。

- /nsconfig/ssl/
- /var/netscaler/ssl/
- /var/vpn/bookmark/
- /nsconfig/dns/

- /nsconfig/htmlinjection/
- /netscaler/htmlinjection/ens/
- /nsconfig/monitors/
- /nsconfig/nstemplates/
- /nsconfig/ssh/
- /nsconfig/rc.netscaler
- /nsconfig/resolv.conf
- /nsconfig/inetd.conf
- /nsconfig/syslog.conf
- /nsconfig/nmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/httpd.conf
- /nsconfig/sshd_config
- /nsconfig/hosts
- /nsconfig/enckey
- /var/nslw.bin/etc/krb5.conf
- /var/nslw.bin/etc/krb5.keytab
- /var/lib/likewise/db/
- /var/download/
- /var/wi/tomcat/webapps/
- /var/wi/tomcat/conf/Catalina/localhost/
- /var/wi/java_home/lib/security/cacerts
- /var/wi/java_home/jre/lib/security/cacerts
- /nsconfig/license/
- /nsconf/rc.conf

> ヒント
>
> クラスタ構成コーディネータに手動 (またはシェルを介して) コピーされたファイル (証明書およびキーファイル) は、他のクラスタノードでは自動的には使用できません。これらのファイルに依存するコマンドを実行する前に、クラスター IP アドレスから「syncclusterfiles」コマンドを実行してください。

コマンドラインインターフェイスを使用してクラスタファイルを同期するには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
sync cluster files <mode>
```


構成ユーティリティを使用してクラスタファイルを同期するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] に移動します。
3. 詳細ペインの [ユーティリティ] で、[クラスターファイルの同期] をクリックします。
4. [クラスターファイルの 同期] ダイアログボックスで、[モード] ドロップダウンリストから同期するファイルを選択します。
5. 「**OK**」をクリックします。

クラスターの統計情報の表示

October 7, 2021

クラスターインスタンスとクラスターノードの統計情報を表示して、パフォーマンスを評価したり、クラスターの操作をトラブルシューティングしたりできます。

コマンドラインインターフェイスを使用してクラスターインスタンスの統計を表示するには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
stat cluster instance <clId>
```

コマンドラインインターフェイスを使用してクラスターノードの統計情報を表示するには

クラスタ IP アドレスのコマンドプロンプトで、次のように入力します。

```
stat cluster node <nodeid>
```

> 注
>
> `stat cluster node <nodeid>` コマンドは、クラスター IP アドレスからコマンドを実行すると、クラスターレベルの統計を表示します。ただし、クラスターノードの NSIP アドレスから実行すると、コマンドはノードレベルの統計を表示します。

構成ユーティリティを使用してクラスターインスタンスの統計を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] **>** [クラスタ] に移動します。

3. 詳細ウィンドウで、ページの中央にある [統計] をクリックします。

構成ユーティリティを使用してクラスタノードの統計情報を表示するには

1. クラスタ IP アドレスにログオンします。
2. [システム] > [クラスタ] > [ノード] に移動します。
3. 詳細ペインでノードを選択し、[**Statistics**] をクリックしてノードの統計情報を表示します。すべてのノードの統計情報を表示するには、特定のノードを選択せずに [**Statistics**] をクリックします。

Citrix ADC アプライアンスの検出

October 7, 2021

現在のノードと同じサブネットに存在するアプライアンスを検出できます。検出された必要なアプライアンスは、クラスタに選択的に追加できます。この操作は、クラスターを作成するか、既存のクラスターにノードを追加するために実行できます。

> 注
>
> - 検出操作は、構成ユーティリティを使用してのみ実行できます。
> - この操作では、異なるネットワークから Citrix ADC アプライアンスを検出できません。
> - この操作を実行して既存のクラスターにノードを追加すると、L3VLAN 構成がノードからクリアされます。アプライアンスがクラスターに追加されたら、必ずこれらの構成を定義してください。

GUI を使用してアプライアンスを検出するには

1. クラスタ IP アドレスにログオンします。
2. [システム] > [クラスタ] > [ノード] に移動します。
3. 詳細ペインで、ページの下部にある [**NetScalers** の検出] をクリックします。
4. [**NetScalers** の検出] ダイアログ・ボックスで、次のパラメータを設定します。
   - **IP** アドレスの範囲: アプライアンスを検出する IP アドレスの範囲を指定します。たとえば、10.102.29.4 ～10.102.29.15 の間のすべての NSIP アドレスを検索するには、このオプションを 10.102.29.4 ～15 と指定します。
   - バックプレーンインターフェイス: バックプレーンインターフェイスとして使用するインターフェイスを指定します。これはオプションのパラメーターです。このパラメーターを指定しない場合は、ノードをクラスターに追加した後にパラメーターを更新する必要があります。
5. 「**OK**」をクリックします。
6. クラスタに追加するアプライアンスを選択します。
7. 「**OK**」をクリックします。

クラスタノードの無効化

October 7, 2021

そのノードのクラスタインスタンスを無効にすることで、クラスタからノードを一時的に削除できます。無効になっているノードは、クラスタ構成と同期されません。ノードを再度有効にすると、クラスタ構成は自動的に同期されます。詳細については、「クラスタノード間の同期」を参照してください。

無効化されたノードはトラフィックを処理できず、このノード上の既存の接続はすべて終了します。

> 注
>
> 無効にされた非構成コーディネーターノードの構成が（ノードの NSIP アドレスを使用して）変更された場合、構成はそのノードで自動的に同期されません。クラスタ構成の同期の説明に従って、構成を手動で同期できます。

コマンドラインインターフェイスを使用してクラスタノードを無効にするには

無効にするノードのコマンドプロンプトで、次のように入力します。

```
disable cluster instance <clId>
```

> 注
>
> クラスターを無効にするには、クラスター IP アドレスに対して
> disable cluster インスタンスコマンドを実行します。

構成ユーティリティを使用してクラスタノードを無効にするには

1. 無効にするノードで、[システム] > [クラスタ] に移動し、[クラスタの管理] をクリックします。
2. [クラスターインスタンスの 構成] ダイアログボックスで、[クラスターインスタンスの 有効化] チェックボックスの選択を解除します。

   > 注
   >
   > すべてのノードでクラスターインスタンスを無効にするには、クラスター IP アドレスで前述の手順を実行します。

クラスタノードの削除

October 7, 2021

ノードがクラスターから削除されると、クラスター構成はノードからクリアされます（clear ns config -extended コマンドを内部的に実行することにより）。SNIP アドレス、バックプレーンインターフェイスの **MTU** 設定、およびすべての VLAN 構成（デフォルトの VLAN と NSVLAN を除く）もアプライアンスからクリアされます。

> 注
>
> - 削除されたノードがクラスター構成コーディネーター（CCO）であった場合、別のノードが CCO として自動的に選択され、クラスター IP アドレスがそのノードに割り当てられます。現在のクラスター IP アドレスセッションはすべて無効であり、新しいセッションを開始する必要があります。
> - クラスター全体を削除するには、各ノードを個別に削除する必要があります。最後のノードを削除すると、クラスターの IP アドレスが削除されます。
> - アクティブなノードを削除すると、クラスタのトラフィック処理能力が 1 つのノードだけ減少します。このノード上の既存の接続は終了します。

CLI を使用してクラスターノードを削除するには

NetScaler 10.1 以降のバージョンの場合

1. クラスタ IP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
rm cluster node <nodeId>

save ns config
```

2. 削除されたノード、NSIP アドレスにログオンし、コマンドプロンプトで次のように入力します。

```
save ns config
```

> 注
>
> クラスタ IP アドレスがノードから到達できない場合は、そのノード自体の NSIP アドレスで rm cluster instance コマンドを実行します。

NetScaler 10 の場合

1. クラスターから削除するノードにログオンし、クラスターインスタンスへの参照を削除します。

```
rm cluster instance <clId>

save ns config
```


2. クラスタ IP アドレスにログオンし、クラスタインスタンスを削除したノードを削除します。

```
rm cluster node <nodeId>

save ns config
```

ローカルノードから `rm cluster node` コマンドを実行しないでください。その結果、CCO とノード間の構成が矛盾します。

GUI を使用してクラスタノードを削除するには

クラスタ IP アドレスで、[システム] **>** [クラスタ] **>**[ノード] に移動し、削除するノードを選択して [削除] をクリックします。

クラスタリンクアグリゲーションを使用してデプロイされたクラスタからノードを削除する

October 7, 2021

トラフィック分散メカニズムとしてクラスタリンクアグリゲーションを使用するクラスタからノードを削除するには、トラフィックを受信しないようにノードがパッシブになっていることを確認してから、アップストリームスイッチで対応するインターフェイスをチャネルから削除する必要があります。

クラスタリンク集約の詳細については、「クラスタリンク集約の使用」を参照してください。

クラスタリンクアグリゲーションをトラフィック分散メカニズムとして使用するクラスタからノードを削除するには

1. クラスタ IP アドレスにログオンします。
2. 削除するクラスターノードの状態を PASSIVE に設定します。

```
set cluster node <nodeId> -state PASSIVE
```

3. アップストリームスイッチで、スイッチ固有のコマンドを使用して、対応するインターフェイスをチャネルから削除します。

> 注
>
> クラスタリンク集約チャネル上のノードインターフェイスを手動で削除する必要はありません。次の手順でノードを削除すると、自動的に削除されます。

4. クラスタからノードを削除します。

```
rm cluster node <nodeId>
```

クラスタ上のジャンボプローブの検出

October 7, 2021

クラスタインターフェイスでジャンボフレームが有効になっている場合、バックプレーンインターフェイスは、ジャンボフレーム内のすべてのパケットをサポートするのに十分な大きさである必要があります。これは、バックプレーンの最大伝送ユニット（MTU）を次のように設定することで実現されます。

Backplane_MTU = 最大（すべてのクラスターインターフェイス MTU）+ 78

上記の構成を確認するには、（上記の計算サイズの）ジャンボプローブをクラスターセットアップのすべてのピアノードに送信する必要があります。プローブが成功しなかった場合、アプライアンスは「show cluster instance」コマンドの出力に警告メッセージを表示します。

コマンドインターフェイスモードで、次のコマンドを入力します。

```
> show cluster instance
  Cluster ID: 1
  Dead Interval: 3 secs
  Hello Interval: 200 msecs
  Preemption: DISABLED
  Propagation: ENABLED
  Quorum Type: MAJORITY
  INC State: DISABLED
  Process Local: DISABLED
  Cluster Status: ENABLED(admin),    ENABLED(operational), UP
```

> 警告
>
> バックプレーンインターフェイスの MTU は、フレーム内のすべてのパケットを処理するのに十分な大きさで

> ある必要があります。これは<MTU_VAL>と等しくなければなりません。推奨値がユーザ設定可能でない場合は、ジャンボインターフェイスの MTU 値を確認する必要があります。

Sl 番号	メンバー・ノード	状況	管理者の状態	動作状態
1	ノード ID: 1; ノード IP: 10.102.53.167	上へ	Active	ACTIVE（構成コーディネータ）
2	ノード ID: 2、ノード IP: 10.102.53.168	上へ	Active	Active

クラスタ内の動的ルートのルート監視

October 7, 2021

ルートモニターを使用して、動的に学習されたルートが含まれているかどうかに関係なく、クラスターノードを内部ルーティングテーブルに依存させることができます。各ノードのルートモニターは、内部ルーティングテーブルをチェックして、特定のネットワークに到達するためのルートエントリが常に存在することを確認します。ルートエントリが存在しない場合、ルートモニタの状態は DOWN に変わります。

クラスタ展開では、ノードのクライアント側またはサーバー側リンクがダウンした場合、トラフィックは処理のためにピアノードを介してこのノードに誘導されます。トラフィックのステアリングは、動的ルーティングを設定し、すべてのノードで各ノードの特殊な MAC アドレスを指すスタティック ARP エントリを追加することによって実装されます。クラスタ展開に多数のノードがある場合、すべてのノードに特別な MAC アドレスを持つ静的 ARP エントリを追加して管理することは、面倒な作業です。現在、ノードは暗黙的にパケットをステアリングするために特別な MAC アドレスを使用します。したがって、特別な MAC アドレスを指すスタティック ARP エントリをクラスターノードに追加する必要はなくなりました。

CLI を使用してクラスターノードをバインドするには

コマンドプロンプトで入力します。

```
bind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
unbind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
```


ノード 1 がルートモニタ 1.1.1.0 255.255.255.0 にバインドされているシナリオを考えてみましょう。ダイナミックルートに障害が発生すると、ノード 1 は INACTIVE になります。ヘルスステータスは、次のようにノード ID によって `show cluster node` コマンドで確認できます。

```
Node ID: 1
IP:  10.102.169.96
Backplane:  1/1/2
Health: NOT UP
Reason(s): Route Monitor(s) of the node have failed
Route Monitor -  Network: 1.1.1.0   Netmask: 255.255.255.0   State:
    DOWN
```

SNMP リンクを使用した SNMP MIB を使用したクラスタ設定のモニタリング

October 7, 2021

SNMP MIB は、Citrix ADC アプライアンスを識別するために SNMP エージェントで構成されるデバイス固有の情報です。アプライアンス名、管理者、場所などの情報を識別できます。クラスタセットアップで、set SNMP MIB コマンドに「ownerNode」パラメータを含めることにより、任意のノードで SNMP MIB を設定できるようになりました。このパラメーターがない場合、set SNMP MIB コマンドはクラスターコーディネーター（CCO）ノードにのみ適用されます。

CCO 以外のクラスタノードの MIB 設定を表示するには、show SNMP MIB コマンドに「ownerNode」パラメータを含めます。

CLIP での SNMP MIB の設定

コマンドラインインターフェイスを使用して CLIP の MIB 設定を設定および表示する。

```
set snmp mib [-contact <string>] [-name <string>] [-location <string>]
      [-customID <string>] [-ownerNode <positive_integer>]
Done
show snmp mib [-ownerNode <positive_integer>]

 > set mib -contact John -name NS59 -location San Jose -customID 123 -
    ownerNode 3
Done
> sh mib -ownerNode 3
         --------------------
```



```
10          Cluster Node ID: 3
11         --------------------
12         NetScaler system MIB:
13         sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun  7
               2016, 10:27:29
14         sysUpTime:   124300
15         sysObjectID: .1.3.6.1.4.1.5951.1.1
16         sysContact:  John
17         sysName:     NS59
18         sysLocation: San Jose
19         sysServices: 72
20         Custom ID: 123
21  Done
22
23 > unset mib -contact -name -location -customID -ownerNode 3
24  Done
25 > sh mib -ownerNode 3
26         --------------------
27          Cluster Node ID: 3
28         --------------------
29         NetScaler system MIB:
30         sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun 7
               2016, 10:27:29
31         sysUpTime:   146023
32         sysObjectID: .1.3.6.1.4.1.5951.1.1
33         sysContact:  WebMaster (default)
34         sysName:     NetScaler
35         sysLocation: POP (default)
36         sysServices: 72
37         Custom ID: Default
38  Done
```

クラスター **SNMP** トラップメッセージ

クラスタセットアップでは、SNMP トラップアラームの設定は CLIP から行う必要があります。コマンドは各ノードに伝播されます。

SNMP の構成の詳細については、「SNMP トラップを生成するように Citrix ADC を構成する」を参照してください。

使用可能なクラスター固有のトラップは次のとおりです。

```
1   >sh snmp alarm | grep cluster
2       CLUSTER-BACKPLANE-HB-MISSING N/A N/A 86400   ENABLED - ENABLED
```



```
3         CLUSTER-CCO-CHANGE N/A N/A N/A ENABLED - ENABLED
4         CLUSTER-NODE-HEALTH N/A N/A 86400 ENABLED - ENABLED
5         CLUSTER-NODE-QUORUM N/A N/A 86400 ENABLED - ENABLED
6         CLUSTER-OVS-CHANGE N/A N/A N/A ENABLED - ENABLED
7         CLUSTER-PROP-FAILURE N/A N/A N/A ENABLED - ENABLED
8         CLUSTER-SYNC-FAILURE N/A N/A N/A ENABLED - ENABLED
9         CLUSTER-SYNC-PARTIAL-SUCCESS N/A N/A N/A ENABLED - ENABLED
10        CLUSTER-VERSION-MISMATCH N/A N/A 86400 ENABLED - ENABLED
```

クラスタ展開におけるコマンド伝播の失敗の監視

October 7, 2021

クラスタ展開では、新しいコマンド「show prop status」を使用して、問題の監視とトラブルシューティングを高速化できます。非 CCO ノードでのコマンド伝播の失敗に関連する問題。このコマンドは、すべての CCO ノードで最新のコマンド伝播の失敗を最大 20 件表示します。この操作を実行するには、Citrix ADC アプライアンスの CLI または GUI のいずれかを使用できます。CLIP アドレスまたはクラスター展開内の任意のノードの NSIP アドレスを介してそれらにアクセスできます。

ノードの正常なシャットダウン

October 7, 2021

クラスターセットアップでは、クラスターレベルまたは特定の仮想サーバーレベルでの既存の接続 (1/N 番目の接続、N はクラスターサイズ) の一部が失われます。この動作は、ノードがシステムから脱退または加入した場合に観察されます。損失に対処するには、既存の接続を正常に処理する必要があります。正常な処理は、CLIP アドレスで「クラスターに接続を保持する」オプションを構成し、ノードの NSIP でタイムアウト間隔を指定することによって行われます。

接続の正常な処理は、次の 2 つのシナリオで適用できます。

1. クラスタのアップグレード
2. 新規ノードの追加

クラスタアップグレード時のノードの正常な処理

クラスタをアップグレードするには、ノードを 1 つずつアップグレードする必要があります。ノードをアップグレードする前に、ノードをパッシブ状態に設定し、アップグレード後にアクティブ状態に設定する必要があります。ノー

ドのアップグレード時に既存の接続が終了しないようにするには、設定されたタイムアウト間隔で正常にシャットダウンします。それ以外の場合、クラスタの接続の 1/N (N はクラスタサイズ) が終了します。

> 注
>
> 既存のセッションが設定されたタイムアウト間隔内に完了しなかった場合、猶予時間の後に終了されます。

クラスタのアップグレードシナリオでノードを正常に処理する手順は次のとおりです。

1. 5 つのノード (n0、n1、n2、n3、n4) のクラスタ設定について考えてみます。
2. ノードをシャットダウンする前に、「retainConnectionsOnCluster」オプションを構成する必要があります。このノードの既存のすべての接続をクラスターレベルまたは仮想サーバーレベルで特定の時間間隔に保持するのに役立ちます。

   例

   CLIP の場合

   “‘set cluster instance –retainConnectionsOnCluster YES

   ```
   ま た は

   ```set lb vserver <vserver name>  - retainConnectionsOnCluster Yes
      <!--NeedCopy-->
   ```

3. 次に、ノード n3 の NSIP アドレスにログオンし、ノード n3 を内部タイムアウトで PASSIVE に設定します。

   例

   “‘set cluster node n3 –state PASSIVE –delay 60

   ```
   ```saveconfig<!--NeedCopy-->
   ```

4. 猶予期間が終了したら、すべての接続を閉じて n3 をシャットダウンし、Citrix ADC アプライアンスを再起動します。
5. アプライアンスをアップグレードします。次に、CLI がアプライアンスの NSIP アドレスに接続され、ノードを ACTIVE に設定します。

   例

   “‘set cluster node n3 –state ACTIVE

   ```
   ```saveconfig<!--NeedCopy-->
   ```


   ```

6. クラスタ内のすべてのノードについて、手順 4 ～6 を繰り返します。
7. すべてのノードをアップグレードして ACTIVE に設定したら、CLIP アドレスから retainConnectionsOnCluster オプションをリセットします。

   例

   “‘set cluster instance -retainConnectionsOnCluster NO

```
ま た は

```set lb vserver <vserver name>  - retainConnectionsOnCluster NO
   <!--NeedCopy-->
```

> 注
>
> クラスタのアップグレード時にバージョンの不一致がある場合、クラスタの伝播は自動的に無効になり、CLIP でコマンドは使用できません。

新規ノード追加時のノードの正常な処理

ノードの正常な処理は、新しいノードを既存の Citrix ADC クラスタに追加する方法を示しています。すでにトラフィックを処理している Citrix ADC クラスタがあるとします。また、既存の接続を終了せずに、ノードとしてクラスタに追加のアプライアンスを追加したいと考えています。前述のシナリオを実行するには、グローバルレベルまたは特定の仮想サーバーレベルで既存の接続を保持するオプションを設定します。完了したら、設定を保存します。ここで、接続を NO に保持するオプションを設定し、他のノードからの既存の接続を新しいノードに再割り当てできるようにします。

ノードが新しく追加された場合、正常にノードを処理するための手順は次のとおりです。

1. 「retainConnectionsOnCluster」オプションが有効になっている既存の構成を保存します。そうすることで、このノードの既存のすべての接続をクラスターレベルまたは仮想サーバーレベルで特定の時間間隔で保持できます。

   CLIP の場合

```
set cluster instance x  -  retainConnectionsOnCluster YES
```

   または


```

```
set lb vserver xxxx - retainConnectionsOnCluster Yes
```

2. クラスタセットアップにノード ‘n5’ を追加します。
3. 他のノードから新しく追加したノード n5 に既存の接続を分散するには、「retainConnectionOnCluster」オプションを「NO」に無効にします。

   CLIP の場合

```
set cluster instance x  -  retainConnectionsOnCluster NO
```

   または

```
set lb vserver xxxx  - retainConnectionsOnCluster NO
```

> 注
>
> バックプレーンステアリングは、クラスタセットアップのトラフィック分散メカニズムのタイプ (ECMP、CLAG、および USIP) によって異なります。バックプレーンステアリングの増加は、トラフィックタイプに基づきます。

クラスタ内のノードのグレースフルシャットダウンの構成

クラスタ内のノードのグレースフルシャットダウンを設定するには、次の手順を実行します。

1. グローバル（クラスター）レベルで「retainConnectionsonCluster」オプションを構成します。
2. 仮想サーバーレベルで「retainConnectionsonCluster」オプションを構成します。
3. ノードの NSIP アドレスで指定されたグレースフルタイムアウト間隔で、ノードをパッシブ状態に設定する（システムから離れる）。
4. 既存の接続を監視して、すべてのトランザクションが猶予期間内に完了していることを確認します。

CLI を使用して、既存の接続をグローバル（クラスター）レベルで保持するには

既存の接続は、グローバルレベルまたは特定の仮想サーバーレベルで保持できます。このオプションは、既存のすべての接続をグローバルレベルで保持するように構成されています。デフォルトでは、このオプションは無効になっています。

コマンドプロンプトで次のように入力します。


```
- set cluster instance <clusterID> -retainConnectionsOnCluster YES

- set cluster instance 60 - retainConnectionsOnCluster YES
```

CLI を使用して、クラスター内の特定の仮想サーバーの既存の接続を保持するには

このオプションは、負荷分散仮想サーバーに固有の既存の接続を保持するように構成されています。これらの接続を維持するために、仮想サーバーレベルでこのオプションを有効にします。デフォルトでは、このオプションは無効になっています。

コマンドプロンプトで入力します。

```
- set lb vserver <clusterID> -retainConnectionsOnCluster Yes

- set lb vserver v1 -retainConnectionsOnCluster Yes
```

CLI を使用してクラスターノードをパッシブ状態に設定するには

グレースフルタイムアウト間隔で、クラスターノードをパッシブ状態に設定する。クラスタのアップグレード中に伝播が無効になるため、この設定はノードの NSIP で実行されます。

コマンドプロンプトで入力します。

```
- set cluster node <clusterID> -state passive
-backplane <interface_name>@
-priority <positive_integer>
-delay <mins>

- set cluster node 4 -state PASSIVE -delay 60

- set cluster instance 60 - retainConnectionsOnCluster YES
- set lb vserver v1 -retainConnectionsOnCluster Yes
- set cluster node 4 -state PASSIVE -delay 60
```

> 注
>
> CLIP から構成された遅延オプションを使用してパッシブに設定されている場合、クラスターノードで次の動作が見られる場合があります。
>
> - タイムアウト後、ノードはノードの NSIP からパッシブとして表示されます。

> - CLIP で **show cluster instance** コマンドを実行すると、CLIP からノードがアクティブとして表示されます。一方、CLIP の **show cluster node** コマンドでは、ノードはパッシブとして表示されます。

GUI を使用してノードの正常なシャットダウンを構成するには

1. [構成] **>** [システム] **>** [クラスタ] に移動し、[クラスタの管理] をクリックします
2. [クラスターの管理] ページで、[クラスターの接続を保持] オプションを選択します。
3. [**OK**] をクリックし、[完了] をクリックします。

サービスの正常なシャットダウン

October 7, 2021

NetScaler 12.1 ビルド 49.xx 以降、Citrix ADC クラスターはサービスの正常なシャットダウンをサポートします。サービスを正常にシャットダウンするには、次のいずれかのタスクを実行できます。

- サービスを明示的に無効にし、
  - 遅延を秒単位で設定します。
  - グレースフルシャットダウンを有効にします。
- TROFS コードまたは文字列をモニタに追加します。

詳細については、「サービスのグレースフルシャットダウン」を参照してください。

CLI を使用してサービスのグレースフルシャットダウンを設定するには

グレースフルオプションでのみ無効にします。

コマンドプロンプトで入力します。

```
disable service <name> [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

例

```
disable service svc1 -graceFul YES
Done
sh service svc1
```



```
                svc1 (10.102.225.11:80) - HTTP
                State: GOING OUT OF SERVICE   Graceful (number of
                   active clients: 1)
                Last state change was at Wed Jul 25 10:46:29 2018
                Time since last state change: 0 days, 00:00:02.680
                ... ... ... .
                ... ... ... .
                Traffic Domain: 0

1)            Monitor Name: tcp-default
                                  State: UP              Weight: 1
                                                  Passive: 0
                                  Probes: 26           Failed [Total: 0
                                     Current: 0]
                                  Last response: Success - TCP syn+ack
                                     received.
                                  Response Time: 0.0 millisec
<!--NeedCopy-->
```

タイムアウトとグレースフルオプションで無効にします。

コマンドプロンプトで入力します。

```
disable service <name> [<delay>] [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

例

```
disable service svc1 2000 -graceFul YES

Done
> sh service svc1
                svc1 (10.102.225.11:80) - HTTP
                State: GOING OUT OF SERVICE (Graceful (number of active
                    clients: 1), Out Of Service in 1998 seconds)
                Last state change was at Wed Jul 25 10:49:08 2018
                Time since last state change: 0 days, 00:00:01.710
                ... ... ... .
                ... ... ... .
                Traffic Domain: 0
```



```
12
13 1)            Monitor Name: tcp-default
14                                 State: UP             Weight: 1
                                                  Passive: 0
15                                 Probes: 57            Failed [Total: 0
                                       Current: 0]
16                                 Last response: Success - TCP syn+ack
                                       received.
17                                 Response Time: 0.0 millisec
18 Done
19 <!--NeedCopy-->
```

タイムアウトとグレースフルオプションを使用してサービスグループを無効にします。

コマンドプロンプトで入力します。

```
1 disable serviceGroup <serviceGroupName>@ [<serverName>@ <port>] [-delay
2 <secs>] [-graceFul ( YES | NO )]
3 Show service group <serviceGroupName>
4 <!--NeedCopy-->
```

例:

```
1 disable servicegroup sg -delay 2000 -graceFul yes
2 sh servicegroup sg
3                  sg - HTTP
4                  State: DISABLED                Effective State: OUT OF
                       SERVICE Monitor Threshold : 0
5                  Max Conn: 0       Max Req: 0          Max Bandwidth: 0
                       kbits
6                  Use Source IP: NO
7                  Client Keepalive(CKA): NO
8                   … … … .
9                   … … …
10
11
12                 1)   200.200.10.21:80       Server Name: server3
                       Server ID: None Weight: 1
13                                 State:    GOING OUT OF SERVICE (learnt
                                       from node:2 )     Graceful (number
                                       of active clients: 6), Out Of
                                       Service in 1993 seconds
```



```
14                                    Last state change was at Mon Aug 13
                                         15:15:11 2018
15                                    ... ... ...
16
17                  2)   200.200.10.22:80       Server Name: server4
                        Server ID: None Weight: 1
18                                    State:    GOING OUT OF SERVICE (learnt
                                         from node:2 )     Graceful (number
                                         of active clients: 7), Out Of
                                         Service in 1993 seconds
19                                    Last state change was at Mon Aug 13
                                         15:15:11 2018
20  <!--NeedCopy-->
```

> 注
> CLIP は、すべてのクラスターノードからのすべてのアクティブなクライアント接続の集計値を表示します。

GUI を使用してサービスのグレースフルシャットダウンを構成するには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. サービスを開き、[アクション] リストから [無効にする] をクリックします。待機時間を入力し、[Graceful] を選択します。

CLI を使用してモニターで **TROFS** コードまたは文字列を構成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
1  add lb monitor <monitor-name> HTTP -trofsCode <respcode>
2  add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>
3  add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
4  <!--NeedCopy-->
```

GUI を使用してモニターで **TROFS** コードまたは文字列を構成するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. [モニター] ウィンドウで、[追加] をクリックし、次のいずれかの手順を実行します。
   - 「HTTP」としてタイプを選択し、「TROFS コード」を指定します。
   - 「タイプ」として「HTTP-ECV」または「TCP-ECV」を選択し、「TROFS 文字列」を指定します。

クラスタに対する **IPv6** 対応ロゴのサポート

October 7, 2021

クラスター化されたアプライアンスで IPv6 Ready ロゴ認定をテストできます。ND テストケース、ルーター要請処理、ルートアドバタイズメントおよびルーターリダイレクションメッセージの送信など、IPv6 コアプロトコルをテストするための変更されたコマンドは、クラスタ化されたセットアップで使用できます。IPv6 コアプロトコルをテストするために利用可能な IPv6 の機能は次のとおりです。

以下は、IPv6readylogo フェーズ 2 テストスイートで、ND テストケース、ルーター要請処理とルートアドバタイズメントの送信、ルーターリダイレクトメッセージングなど、IPv6 コアプロトコルを渡すために使用できる変更された機能です。

- ローカル SNIP のリンク
- アドレス解決とネイバー到達不能
- ルータおよびプレフィクス検出
- ルータリダイレクション
- DoDAD

これらの変更されたコマンドでは、クラスタ化されたアプライアンスで次の構成がサポートされます。

IPv6 コアプロトコルをテストするためのサポート可能な構成

クラスター化されたセットアップで IPv6Ready Logo テストケースに合格するには、クラスター管理 IP アドレス（CLIP）で次の構成を実行できます。

- global IP6 configuration
- basic IPv6 configuration
- その他の IPv6 構成

グローバル設定

グローバル IPv6 構成を使用すると、グローバル IPv6 パラメーター（再学習、routeRedirection、ndBasereachTime、nRetransmissionTime、natprefix、td、doodad など）を設定して、基本的な IPv6 構成を実行できます。

コマンドプロンプトで、次のように入力します。

```
set ipv6 [-ralearning ( ENABLED | DISABLED )] [-routerRedirection (
    ENABLED | DISABLED )] [-ndBasereachTime<positive_integer>][-
    ndRetransmissionTime <positive_integer>] [-natprefix <ipv6_addr|*>[-
    td<positive_integer>]] [-doDAD ( ENABLED | DISABLED )]
```


基本的な **IPv6** 構成

基本的な IPv6 設定では、IPv6 アドレスを作成し、VLAN インターフェイスにバインドできます。次の構成を実行して、IPv6 コアプロトコルをテストできます。

CLI を使用してクラスター化されたセットアップに VLAN を追加するには

コマンドプロンプトで入力します。

```
1 add vlan <id>
```

CLI を使用してクラスター化されたセットアップに別の VLAN を追加するには

コマンドプロンプトで入力します。

```
1 add vlan <id>
```

CLI を使用してインターフェイスを VLAN にバインドするには

コマンドプロンプトで入力します。

```
1 bind vlan <id> -ifnum <interface_name>
```

CLI を使用してインターフェイスを VLAN にバインドするには

このコマンドは、グローバルプレフィクスを、後続のルータアドバタイズメントの RA 情報にオンリンクプレフィクスとして追加します。コマンドプロンプトで入力します。

```
1 bind vlan <id> -ifnum <interface_name>
```

CLI を使用して VLAN に IPv6SNIP アドレスを追加するには

コマンドプロンプトで、次のように入力します。

```
1 add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

CLI を使用して VLAN に IPv6 アドレスを追加するには

コマンドプロンプトで、次のように入力します。


```
add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

CLI を使用して IPv6 アドレスを VLAN にバインドするには

コマンドプロンプトで、次のように入力します。

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

CLI を使用して IPv6 アドレスを VLAN にバインドするには

コマンドプロンプトで、次のように入力します。

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

CLI を使用して VLAN に接続されたリンクローカル IPv6 アドレスを表示するには

コマンドプロンプトで、次のように入力します。

```
sh VLAN
```

例 **1**

```
add vlan 2
add vlan 3
bind vlan 2 -ifnum 1/2
bind vlan 3 -ifnum 1/3
add ip6 fe80::9404:60ff:fedd:a464/64 -vlan 2 -scope link-local -type
    SNIP
add ip6 fe80::c0ee:7bff:fede:263f/64 -vlan 3 -scope link-local -type
    SNIP
add ip6 3ffe:501:ffff:100:9404:60ff:fedd:a464/64 -vlan 2
add ip6 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64 -vlan 3
bind vlan 2 -ipAddress 3ffe:501:ffff:100:9404:60ff:fedd:a464/64
bind vlan 3 -ipAddress 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64
```

例 **2**

```
sh vlan
1)      VLAN ID: 2      VLAN Alias Name:
        Interfaces : 1/6
        IPs :
             3ffe:501:ffff:100:2e0:edff:fe15:ea2a/64
3)      VLAN ID: 3      VLAN Alias Name:
        Link-local IPv6 addr: fe80::9404:60ff:fedd:a464/64
        Interfaces : 1/5
        IPs :
             3ffe:501:ffff:101:2e0:edff:fe15:ea2b/64
 Done
```

その他の **IPv6** クラスター構成

IPv6 コアプロトコルをテストするには、次の新規または変更された IPv6 構成を使用できます。

CLI を使用して VLAN 固有のルーターアドバタイズメントパラメーターを設定するには

コマンドプロンプトで入力します。

```
set nd6RAvariables -vlan <positive_integer> [-ceaseRouterAdv ( YES | NO
    )] [-sendRouterAdv ( YES | NO )] [-srcLinkLayerAddrOption ( YES | NO
     )][-onlyUnicastRtAdvResponse ( YES | NO )] [-managedAddrConfig (
    YES | NO)] [-otherAddrConfig ( YES | NO )] [-currHopLimit <
    positive_integer>][-maxRtAdvInterval <positive_integer>] [-
    minRtAdvInterval<positive_integer>] [-linkMTU <positive_integer>] [-
    reachableTime<positive_integer>] [-retransTime <positive_integer>]
    [-defaultLifeTime<integer>]
```

CLI を使用して、リンク上のグローバルプレフィックスの設定可能なパラメータを設定するには

コマンドプロンプトで入力します。

```
set onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-
    autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-
    decrementPrefixLifeTimes ( YES | NO )] [-prefixValideLifeTime   <
    positive_integer>] [-prefixPreferredLifeTime <positive_integer>]
```

CLI を使用して、構成可能なパラメーターをオンリンクグローバルプレフィックスに追加するには

コマンドプロンプトで入力します。


```
add onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-
    autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-
    decrementPrefixLifeTimes ( YES | NO )]-prefixValideLifeTime <
    positive_integer>][-prefixPreferredLifeTime <positive_integer>]
```

CLI を使用して、IPv6 プレフィックスの構成可能なパラメーターへのオンリンクリンクを設定するには

コマンドプロンプトで、次のように入力します。

```
help set onLinkIPv6Prefix
```

CLI を使用して、リンク上のリンクを IPv6 プレフィックスの構成可能なパラメーターにバインドするには

コマンドプロンプトで入力します。

```
help bind nd6RAvariables
```

CLI を使用して nd6RA 変数を表示するには

コマンドプロンプトで入力します。

```
help sh nd6RAvariables
```

例

```
> sh nd6RAvariables
  1) Vlan : 1
     SendAdvert     : NO   CeaseAdv       : NO        SourceLLAddress:
        YES
     UnicastOnly    : NO   ManagedFlag    : NO        OtherConfigFlag:
        NO
     CurHopLimit    : 64   MaxRtrAdvInterv: 600       MinRtrAdvInterv:
        198
     LinkMTU        : 0    ReachableTime  : 0          RetransTimer   :
         0
     DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    :
         0

```



```
  2) Vlan : 2
     SendAdvert      : NO   CeaseAdv        : NO          SourceLLAddress:
         YES
     UnicastOnly     : NO   ManagedFlag     : NO          OtherConfigFlag:
         NO
     CurHopLimit     : 64   MaxRtrAdvInterv: 600         MinRtrAdvInterv:
         198
     LinkMTU         : 0    ReachableTime   : 0           RetransTimer    :
         0
     DefaultLifetime: 1800 LastRAsentTime : 0           NextRAdelay     :
         0
Done
>
> sh nd6Ravariables  -vlan 2
  1) Vlan : 2
     SendAdvert      : NO   CeaseAdv        : NO          SourceLLAddress:
         YES
     UnicastOnly     : NO   ManagedFlag     : NO          OtherConfigFlag:
         NO
     CurHopLimit     : 64   MaxRtrAdvInterv: 600         MinRtrAdvInterv:
         198
     LinkMTU         : 0    ReachableTime   : 0           RetransTimer    :
         0
     DefaultLifetime: 1800 LastRAsentTime : 0           NextRAdelay     :
         0
     Prefix :
3ffe:501:ffff:100::/64
Done
```

クラスタハートビートメッセージの管理

October 7, 2021

クラスターでのハートビートメッセージの管理は、高可用性 (HA) 構成での管理と似ています。ノードは、有効になっているすべてのインターフェイス上で、相互にハートビートメッセージを送受信できます。ハートビートメッセージに起因するトラフィックの増加を回避するために、ノードインターフェイスでハートビートオプションを無効にできるようになりました。ただし、バックプレーンインターフェイスのハートビートオプションは、クラスタノード間の接続を維持するために必要であるため、無効にすることはできません。

ハートメッセージの管理の詳細については、「NetScaler アプライアンスでの高可用性ハートビートメッセージの管理」を参照してください。

Citrix ADC CLI を使用してノードインターフェイスでハートビートメッセージを管理するには

コマンドプロンプトで入力します。

```
set interface <ID> [-HAHeartBeat (ON | OFF)]
Show interface <ID>
```

所有者ノードの応答ステータスの設定

October 7, 2021

スポットされた SNIP アドレスを持つノードで ownerDownResponse オプションを構成できます。デフォルトでは、このオプションは有効になっています。これにより、ノードが非アクティブのときに、検出された IP アドレスが（アップストリームルーターからの）PING または ARP 要求に応答できるようになります。このオプションを無効にすると、所有者ノードが非アクティブのとき、IP アドレスはルーター要求に応答できません。

ECMP 展開でスタティックルートのモニタリングにこの機能がどのように使用されるかについては、「等コストマルチパス (ECMP) の使用」トピックを参照してください。

Citrix ADC CLI を使用して所有者ノードの応答ステータスを設定するには

コマンドプロンプトで入力します。

```
add ns ip <IPAddress> [-ownerNode <positive_integer>] [-
    ownerDownResponse (YES | NO )] [-td <positive_integer>]
```

例

```
add ns ip 2.2.2.2 255.255.255.0 -ownernode 6  - ownerdownResponse YES
```

Citrix ADC GUI を使用して所有者ノードの応答ステータスを設定するには

1. [システム] **>** [ネットワーク] **>** [**IP**] に移動し、[追加] をクリックして、スポッティング SNIP アドレスを作成します。
2. [**IP** アドレスの作成] ページで、[**ownerDownResponse**] チェックボックスをオンまたはオフにします。

Citrix ADC GUI を使用して所有者ノードの応答ステータスを編集するには

[システム] > [ネットワーク] > [IP] に移動し、IP アドレスを選択し、[編集] をクリックして [**ownerDownResponse**] チェックボックスをオンまたはオフにします。

スポッティングクラスタ構成内の非アクティブなノードに対するスタティックルート **(MSR)** サポートの監視

October 7, 2021

ルートで MSR オプションを有効にしてセットアップされたクラスターでは、アクティブノードのみが静的ルートをプローブできます。非アクティブでスペアノードがルートにリンクしておらず、プローブできない間は、ネットワークに到達できます。PING および ARP プローブを IPv4 ルートに送信し、ping6 および nd6 プローブを IPv6 ルートに送信するように、非アクティブまたはスペアノードを設定できるようになりました。これは、SNIP アドレスがアクティブであり、1 つのノードのみが独占的に所有しているスポットクラスター構成でのみ実行できます。

単一ノードのアクティブクラスタでの **VRRP** インターフェイスバインディング

October 7, 2021

高可用性 (HA) セットアップをクラスタセットアップに移行する場合、すべての構成に互換性があり、クラスタ内でサポート可能である必要があります。これを実現するために、ノードインターフェイスで仮想ルータ ID（VRID および VRID6）を設定できるようになりました。

> 重要
>
> 現在、VRID と VRID6 をサポートしているのは単一ノードのアクティブクラスターシステムのみです。

VRID および VRID6 の設定手順については、仮想 MAC アドレスの設定を参照してください。

単一ノードのアクティブクラスタで仮想ルータ ID を設定するには、VRID または VRID6 を追加して、クラスタノードインターフェイスにバインドします。

Citrix ADC CLI を使用して VRID を追加するには

コマンドプロンプトで入力します。

```
1 add vrID <ID>
```

Citrix ADC CLI を使用して VRID をクラスターノードインターフェイスにバインドするには

コマンドプロンプトで入力します。


```
Bind vrid <ID> -ifnum <interface_name> | -trackifNum <interface_name

Add vrID 100
Bind vrid 100  - ifnum 1/1  1/2
done
```

Citrix ADC CLI を使用して VRID6 を追加するには

コマンドプロンプトで入力します。

```
add vrID6 <ID>
```

CLI を使用して VRID6 をクラスターノードインターフェイスにバインドするには

コマンドプロンプトで入力します。

```
bind vrid6 <ID> -ifnum <interface_name> | -trackifNum <interface_name>

Add vrID6 100
Bind vrid6 100  - ifnum 1/1  1/2
Done
```

クラスターのセットアップと使用シナリオ

October 7, 2021

このセクションでは、Citrix ADC クラスターをセットアップし、さまざまな機能とネットワークトポロジー用に構成できるシナリオについて説明します。他のシナリオを文書化する場合は、フィードバックを提供します。

2 ノードクラスターの作成

October 7, 2021

2 ノードクラスタは、クラスタが最低限 (n/2 +1) のノード (n はクラスタノード数) がトラフィックを処理できる場合にのみ機能するという規則の例外です。同じ式を 2 ノードのクラスターに適用すると、1 つのノードがダウンするとクラスターが失敗します (n/2 +1=2).

2 ノードクラスタは、1 つのノードだけがトラフィックを処理できる場合でも機能します。

2 ノードクラスタの作成は、他のクラスタの作成と同じです。1 つのノードを構成コーディネーターとして追加し、もう 1 つのノードを他のクラスターノードとして追加します。

> 注
>
> 差分構成の同期は、2 ノードクラスタではサポートされません。完全同期のみがサポートされています。

HA セットアップのクラスタセットアップへの移行

October 7, 2021

既存の高可用性（HA）セットアップをクラスタセットアップに移行するには、まず高可用性セットアップから Citrix ADC アプライアンスを削除し、高可用性構成ファイルのバックアップを作成する必要があります。次に、2 つのアプライアンスを使用してクラスターを作成し、バックアップされた構成ファイルをクラスターにアップロードできます。

> 注
>
> - バックアップされた HA 設定ファイルをクラスタにアップロードする前に、クラスタ互換になるように変更する必要があります。手順の関連するステップを参照してください。
> - **batch-f <backup_filename> コマンドを使用して、バックアップされた構成ファイルをアップロードします。

前述のアプローチは、デプロイされたアプリケーションのダウンタイムをもたらす基本的な移行ソリューションです。そのため、アプリケーションの可用性を考慮しないデプロイメントでのみ使用する必要があります。

ただし、ほとんどのデプロイメントでは、アプリケーションの可用性が最も重要です。このような場合は、ダウンタイムが発生することなく、HA セットアップをクラスタセットアップに移行できるアプローチを使用する必要があります。この方法では、最初にセカンダリアプライアンスを削除し、そのアプライアンスを使用して単一ノードクラスタを作成することにより、既存の HA セットアップをクラスタセットアップに移行します。クラスタが動作可能になり、トラフィックを処理すると、HA セットアップのプライマリアプライアンスがクラスタに追加されます。

コマンドラインインターフェイスを使用して HA セットアップを（ダウンタイムなしで）クラスタセットアップに変換するには

次に、プライマリアプライアンス（NS1）（10.102.97.131 およびセカンダリアプライアンス（NS2））-10.102.97.132 を使用した HA セットアップの例を考えてみましょう。

1. HA ペアの設定が安定していることを確認します。
2. いずれかの HA アプライアンスにログオンし、シェルに移動し、ns.conf ファイル（ns_backup.conf など）のコピーを作成します。

3. セカンダリアプライアンス NS2 にログオンし、設定をクリアします。この操作により、HA セットアップから NS2 が削除され、スタンドアロンアプライアンスになります。

```
> clear ns config full
```

注

- この手順は、NS2 が VIP アドレスの所有を開始しないように、スタンドアロン・アプライアンスであるようにするために必要です。
- この段階では、プライマリアプライアンス NS1 はアクティブのままであり、引き続きトラフィックを処理します。

4. NS2（セカンダリアプライアンスではなくなった）にクラスタを作成し、PASSIVE ノードとして構成します。

```
> add cluster instance 1

> add cluster node 0 10.102.97.132 -state PASSIVE -backplane
   0/1/1

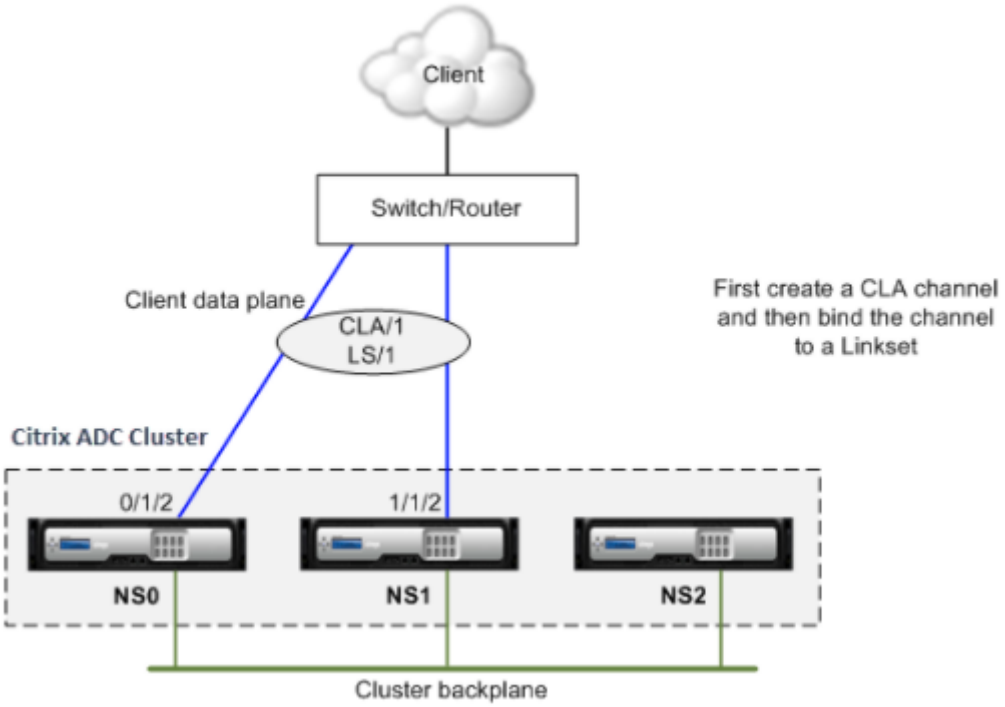
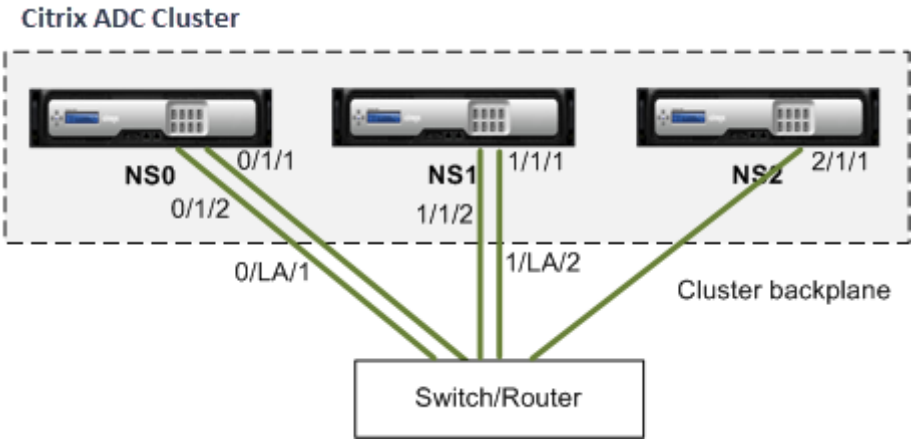
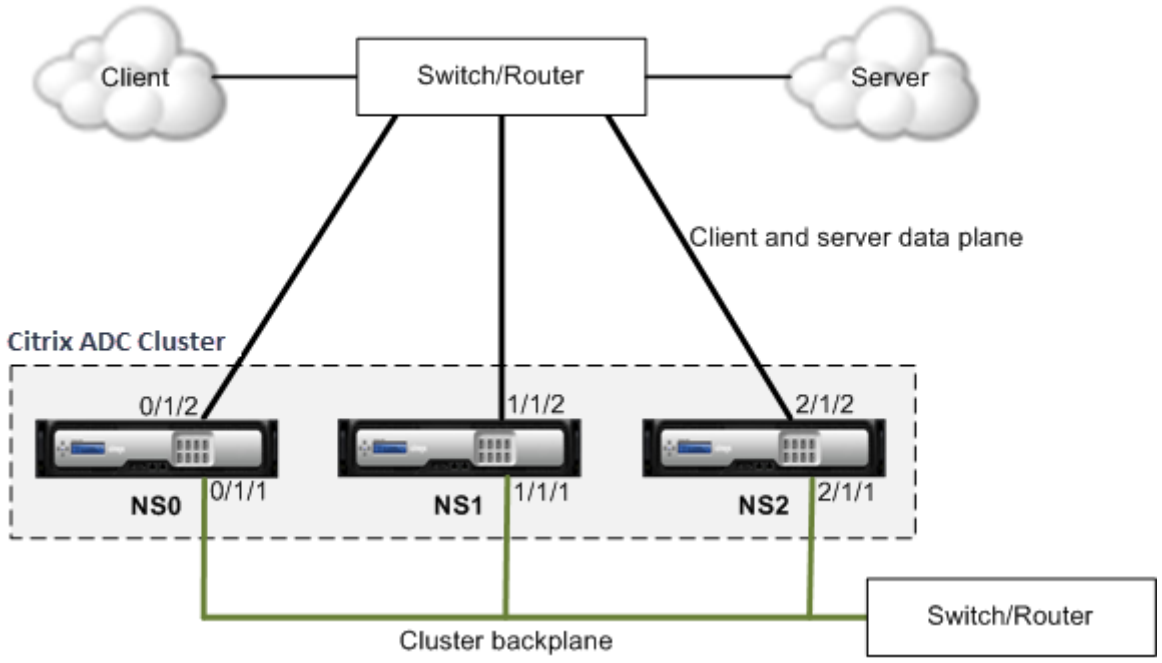
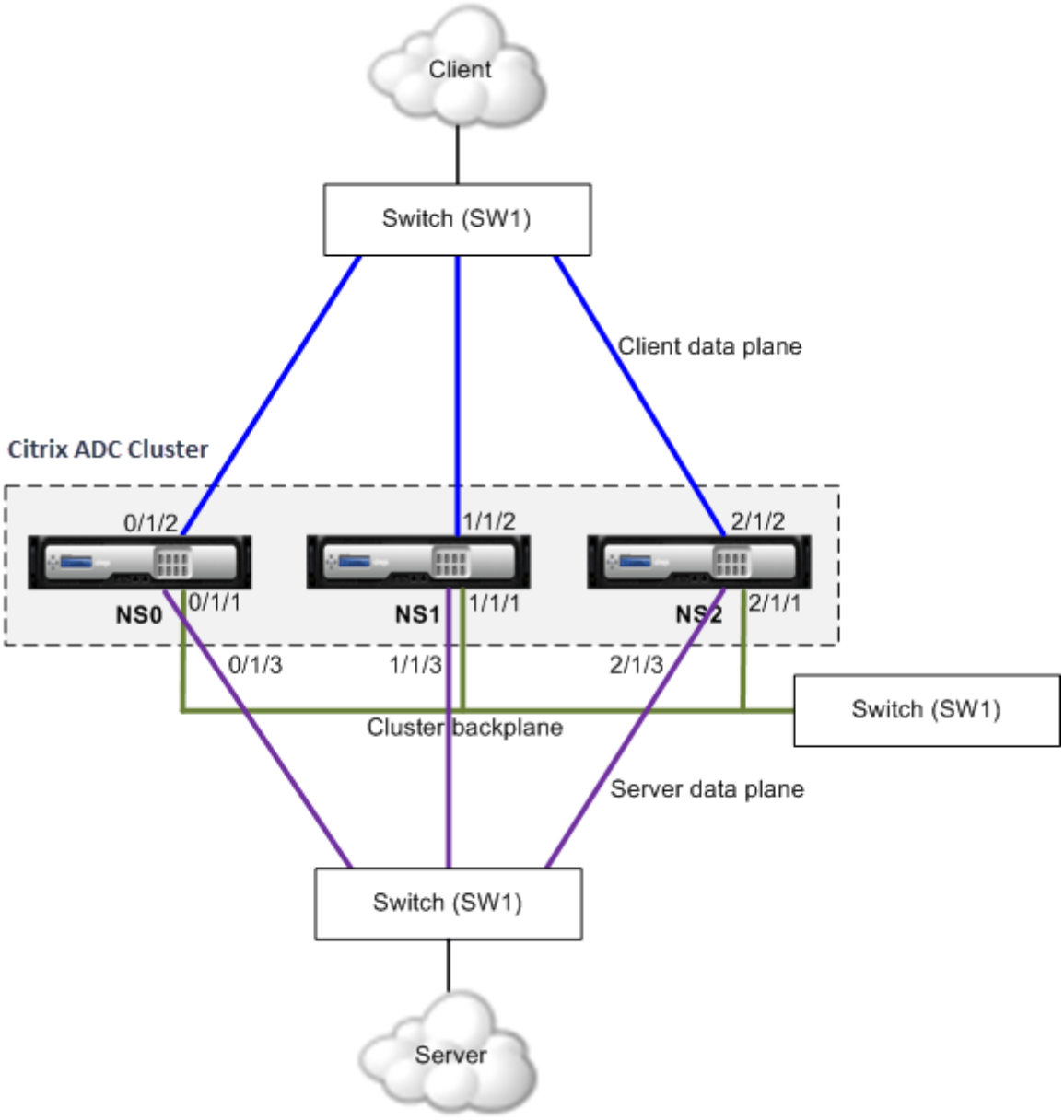
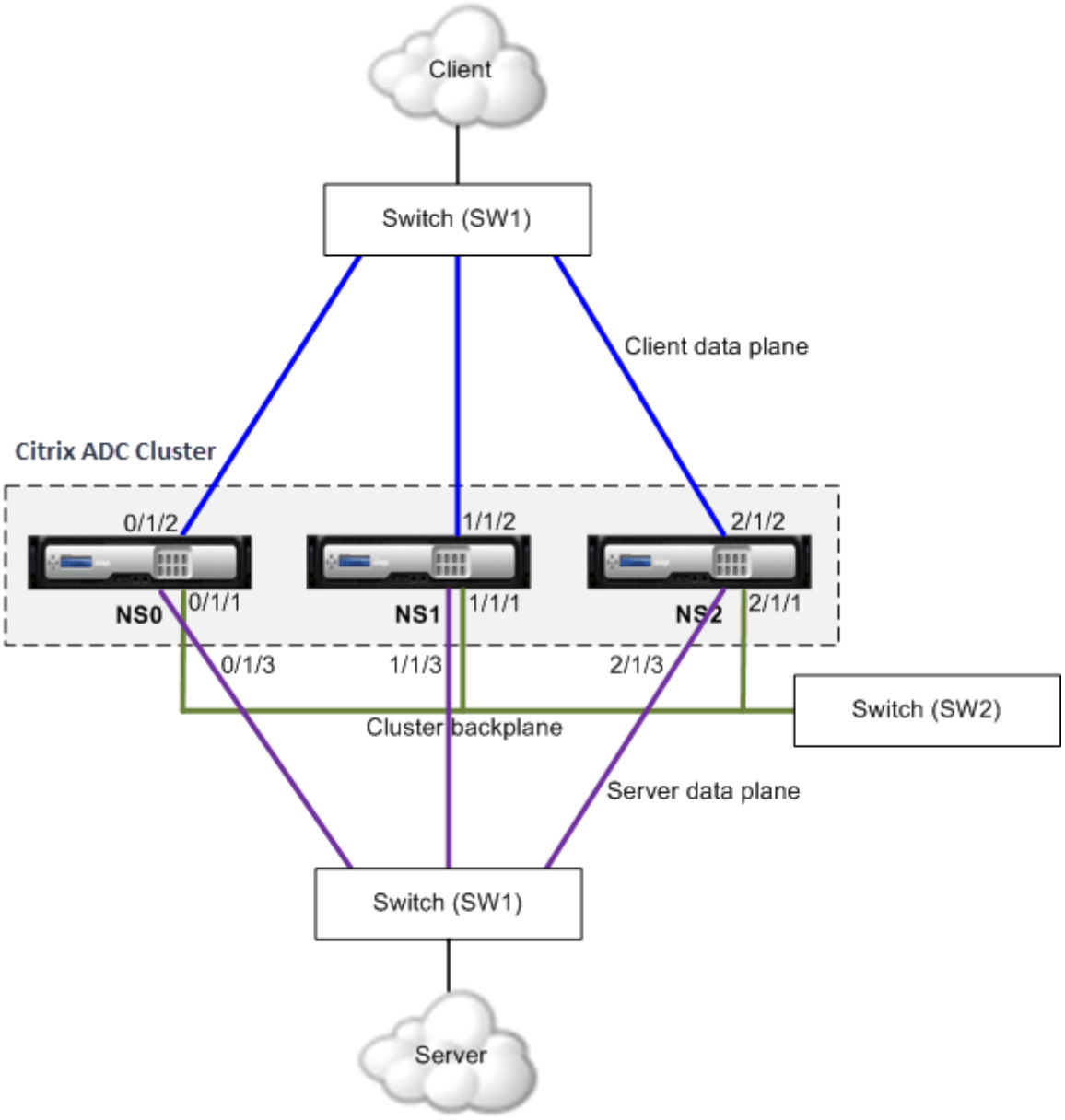
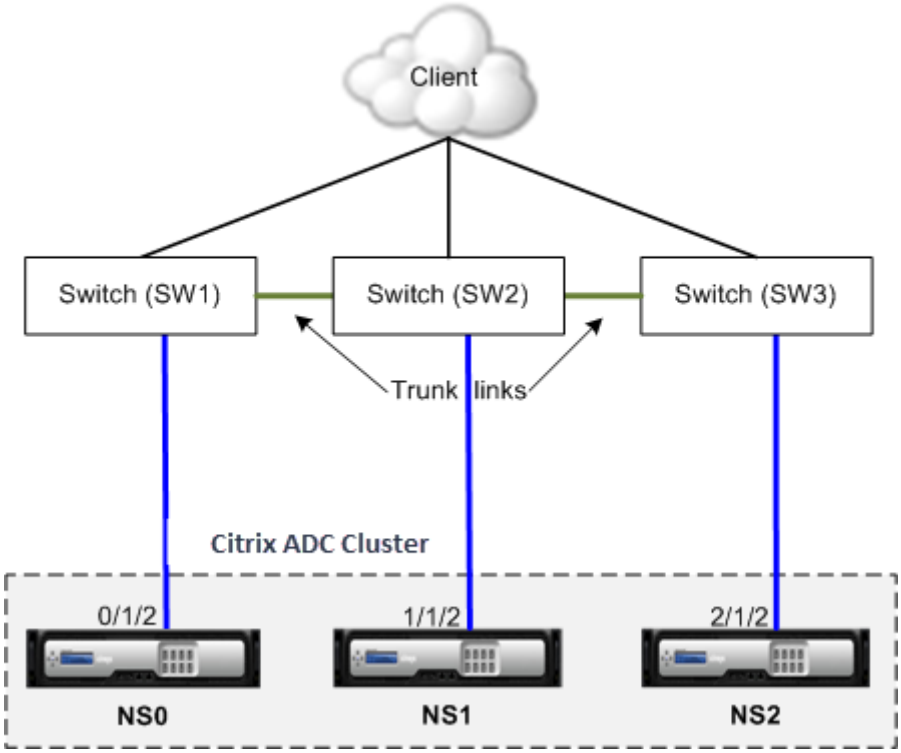
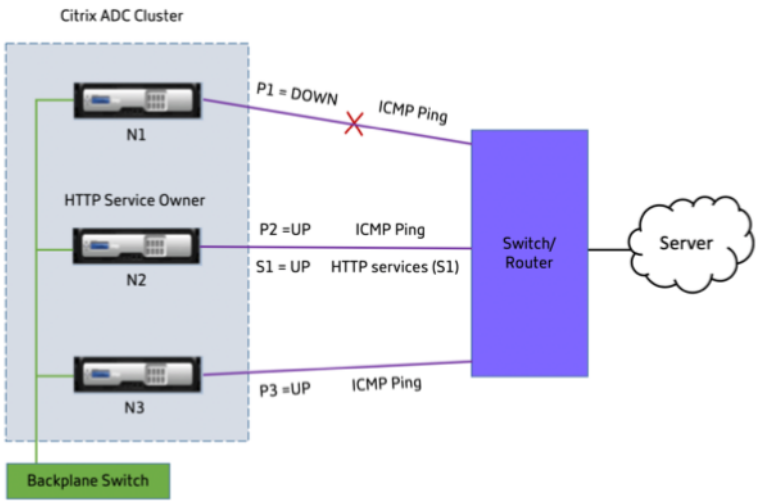
> add ns ip 10.102.97.133 255.255.255.255 -type CLIP

> enable cluster instance 1

> save ns config

> reboot -warm
```

5. バックアップされた構成ファイルを次のように変更します。

   - クラスターでサポートされていない機能を削除します。サポートされていない機能の一覧については、「クラスターでサポートされている Citrix ADC 機能」を参照してください。これはオプションのステップです。この手順を実行しないと、サポートされていないコマンドの実行は失敗します。
   - インターフェイスを持つ設定を削除するか、c/u 規則から n/c/u 規則にインターフェイス名を更新します。

     例

     ```
     > add vlan 10 -ifnum 0/1
     ```

     に変更する必要があります


```
> add vlan 10 -ifnum 0/0/1 1/0/1
```

- バックアップ構成ファイルには、SNIP アドレスを持つことができます。これらのアドレスは、すべてのクラスターノードでストライプされます。各ノードにスポッティング IP アドレスを追加することをお勧めします。

  例

```
> add ns ip 1.1.1.1 255.255.255.0 -ownerNode 0

> add ns ip 1.1.1.2 255.255.255.0 -ownerNode 1
```

- ホスト名を更新して、所有者ノードを指定します。

  例

```
> set ns hostname ns0 -ownerNode 0

> set ns hostname ns1 -ownerNode 1
```

- 検出された IP に依存する他のすべての関連するネットワーク構成を変更します。たとえば、L3 VLAN、NATIP として SNIP を使用する RNAT 設定、SNIP/MIP を参照する INAT ルール)。

6. クラスターで、次の操作を行います。

   - クラスタバックプレーン、クラスタリンクアグリゲーションチャネルなどを接続して、クラスタにトポロジを変更します。
   - バックアップおよび変更された構成ファイルから構成コーディネータに、クラスタ IP アドレスを通じて構成を適用します。

```
> batch -f ns_backup.conf
```

   - ECMP やクラスタリンク集約などの外部トラフィック分散メカニズムを設定します。

7. トラフィックを HA セットアップからクラスタに切り替えます。

   - プライマリアプライアンス NS1 にログオンし、その上にあるすべてのインターフェイスを無効にします。

```
> disable interface <interface_id>
```


- クラスタ IP アドレスにログオンし、NS2 を ACTIVE ノードとして構成します。

```
1   > set cluster node 0 -state ACTIVE
```

> 注
>
> インターフェイスを無効にしてからクラスターノードをアクティブにする間に、わずかなダウンタイムが (秒単位で) 発生することがあります。

8. プライマリアプライアンス NS1 にログオンし、HA セットアップから削除します。
   - すべての設定をクリアします。この操作により、HA セットアップから NS1 が削除され、スタンドアロンアプライアンスになります。

```
1   > clear ns config full
```

   - すべてのインターフェイスを有効にします。

```
1   > enable interface <interface_id>
```

9. NS1 をクラスタに追加します。
   - クラスタの IP アドレスにログオンし、NS1 をクラスタに追加します。

```
1   > add cluster node 1 10.102.97.131 -state PASSIVE -backplane
       1/1/1
```

   - NS1 にログオンし、次のコマンドを順番に実行してクラスターに参加します。

```
1   > join cluster -clip 10.102.97.133 -password nsroot
2
3   > save ns config
4
5   > reboot -warm
```

10. NS1 にログオンし、必要なトポロジーと構成の変更を実行します。
11. クラスタ IP アドレスにログオンし、NS1 を ACTIVE ノードとして設定します。


```
1      > set cluster node 1 -state ACTIVE
```

L2 クラスタと L3 クラスタ間の移行

October 7, 2021

> 注
>
> NetScaler 11 以降でサポートされています。

L2 クラスタは、すべてのノードが同じネットワークに属し、L3 クラスタは異なるネットワークのノードを含むことができるクラスタです。Citrix ADC に展開されているアプリケーションのダウンタイムを発生させずに、あるタイプのクラスターから別のクラスターにシームレスに移行できます。

L2 から L3 へのクラスターの移行

クラスターに他のネットワークのノードを含める場合は、L3 クラスターに移行できます。

クラスタ IP アドレスで、次の操作を行います。

1. ノードグループを作成します。

   例

   ```
   1  > add cluster nodegroup NG0
   ```

   このノードグループは、既存の L2 クラスターからすべてのノードをグループ化するために次のステップで使用されます。

2. L2 クラスタを L3 クラスタに移行。

   例

   ```
   1  > set cluster instance 1 -inc ENABLED -nodegroup NG0
   ```

   このコマンドは、L3 クラスターに移行することと、L2 クラスターのすべてのノードをノードグループに追加することの 2 つの目的を達成します。

3. これで、クラスタへのノードの追加の説明に従って、クラスタにノードを追加できます。

クラスタの **L3** から **L2** への移行

単一のネットワークに属するノードを保持する場合は、L2 クラスターに移行できます。

クラスタ IP アドレスで、次の操作を行います。

1. 保持しないネットワークからクラスターノードを削除します。

   例

```
1 > rm cluster node <nodeId>
```

2. L3 クラスタを L2 クラスタに移行。

   例

```
1 > set cluster instance 1 -inc DISABLED
```

   クラスターに 1 つのネットワークのノードのみが含まれるようになりました。

クラスターでの **GSLB** の設定

October 7, 2021

> 注
>
> NetScaler 10.5 ビルド 52.11 以降でサポートされています。

クラスターで GSLB を設定するには、異なる GSLB エンティティをノードグループにバインドする必要があります。ノードグループには単一のメンバーノードが必要です。

> メモ
>
> - 静的近接 GSLB 方式を構成した場合は、静的近接データベースがすべてのクラスターノードに存在することを確認してください。データベースファイルがデフォルトの場所で利用可能な場合、デフォルトで発生します。ただし、データベースファイルが以外のディレクトリに保持されている場合 /var/netscaler/locdb/, ファイルをすべてのクラスターノードに手動で同期する必要があります。
> - `show gslb domain` コマンドは、クラスター設定ではサポートされていません。

CLI を使用してクラスターに **GSLB** をセットアップするには、次のようにします。

クラスタ IP アドレスにログオンし、コマンドプロンプトで次の操作を実行します。

1. 異なる GSLB エンティティを設定します。詳細については、GSLB 構成エンティティを参照してください。

> 注
>
> GSLB サイトを作成するときは、必ずクラスター IP アドレスとパブリッククラスター IP アドレスを指定してください。パブリッククラスターの IP アドレスは、クラスターが NAT デバイスの背後に展開されている場合にのみ必要です。GSLB サイトを構成するときは、同じサイトのクラスター IP アドレスを使用する必要があります。これらのパラメータは、GSLB 自動同期機能を利用できるようにするために必要です。

```
add gslb site <siteName> <siteType> <siteIPAddress> -publicIP <ip_addr>
 -clip <ip_addr> <publicCLIP><!--NeedCopy-->
```

2. クラスタノードグループを作成します。

```
add cluster nodegroup <name> <name>@ [-strict ( YES | NO )] [-sticky (
YES | NO )] [-state <state>] [-priority <positive_integer>]<!--NeedCopy
-->
```

> 注
>
> VPN ユーザー向けに GSLB を設定する場合は、スティッキオプションを有効にします。

3. 1 つのクラスタノードをノードグループにバインドします。

```
bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
```

4. ローカル GSLB サイトをノードグループにバインドします。

```
bind cluster nodegroup <name> -gslbSite <string><!--NeedCopy-->
```

> 注
>
> ローカル GSLB サイトの IP アドレスの IP アドレスがストライプされていることを確認します (すべてのクラスターノードで利用可能)。

5. ADNS (または ADNS-TCP) サービスまたは DNS (または DNS-TCP) 負荷分散仮想サーバーをノードグループにバインドします。

   ADNS サービスをバインドするには、次の手順を実行します。

   “‘bind cluster nodegroup -service

```
1 **DNS ロードバランシング仮想サーバーをバインドするには、次の手順を
      実行します。**
2
3 ```bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
```

6. GSLB 仮想サーバーをノードグループにバインドします。

```
bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
```


7. [オプション] VPN ユーザーに基づいて GSLB を設定するには、VPN 仮想サーバーを GSLB ノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

8. 設定を確認します。

   ```
   show gslb runningConfig<!--NeedCopy-->
   ```

GUI を使用してクラスターに **GSLB** をセットアップするには、次のようにします。

クラスタの IP アドレスにログオンし、[構成] タブで次の操作を実行します。

1. GSLB エンティティを設定します。

   [トラフィック管理] **>** [**GSLB**] に移動し、必要な設定を実行します。

2. ノードグループを作成し、その他のノードグループ関連の設定を実行します。

   [システム] **>** [クラスタ] **>** [ノードグループ] に移動し、必要な設定を実行します。

実行する詳細な構成については、前の CLI 手順で提供された説明を参照してください。

クラスタ内の **GSLB** 親子トポロジのサポート

NetScaler 12.1 ビルド 49.xx 以降、GSLB 親子トポロジはクラスタでサポートされています。

親子トポロジの詳細については、MEP プロトコルを使用した親子トポロジの展開を参照してください。

CLI を使用してクラスターに **GSLB** 親子トポロジを設定するには

親サイト

次の設定を実行します。

1. クラスタノードグループを作成します。

   ```
   add cluster nodegroup <name>
   ```

   例:

   ```
   add cluster nodegroup parentng
   ```

2. 1 つのクラスタノードをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   例:

   ```
   bind cluster nodegroup parentng -node n2
   ```


3. ローカル GSLB サイトをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   例:

   ```
   bind cluster nodegroup parentng -gslbSite site1
   ```

4. ADNS (または ADNS-TCP) サービスまたは DNS (または DNS-TCP) 負荷分散仮想サーバーをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -service <string>
   ```

   例:

   ```
   bind cluster nodegroup parentng - service ADNS
   ```

5. GSLB 仮想サーバーをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -vServer <string>
   ```

   例:

   ```
   bind cluster nodegroup parentng -vServce gslbvs1
   ```

子サイト

次の設定を実行します。

1. クラスタノードグループを作成します。

   ```
   add cluster nodegroup <name>
   ```

   例:

   ```
   add cluster nodegroup childng
   ```

2. 1 つのクラスタノードをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   例:

   ```
   bind cluster nodegroup childng -node -n3
   ```

3. ローカル GSLB サイトをノードグループにバインドします。

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   例:

   ```
   bind cluster nodegroup childng -gslbSite site1
   ```


> 注
>
> 親サイトと子サイトがメトリックベースの負荷分散メソッドで集約された統計を交換するには、子サイトにローカル GSLB サービスを追加する必要があります。メトリックベースのロードバランシング方式は、最小接続、最小帯域幅、最小パケットです。

GUI を使用してクラスタに **GSLB** 親子トポロジを設定するには

1. GSLB エンティティを設定します。

   [トラフィック管理**]** > **[GSLB]** に移動して、必要な設定を実行します。

2. ノードグループを作成します。

   [システム**]** > **[**クラスタ**]** > **[**ノードグループ**]** に移動して、必要な構成を実行します。

3. [ノードグループ] ページで、ノードをバインドするノードグループを選択し、[編集] をクリックして、次のタスクを実行します。これらのタスクは、ノードグループを追加するときにも実行できます。

   - ノードをノードグループにバインドします。

     [詳細 設定**]** で、[クラスターノード] をクリックして、次のタスクを実行します。

     - クラスタノードのセクションでは、ありませんクラスタノードをクリックします。
     - **[**クラスターノードの選択] で、[> そして、ノードグループにバインドするノードを選択します。クラスタノードを追加することもできます。

   - ローカル GSLB サイトをノードグループにバインドします。

     [詳細設定] で、[GSLB サイト] をクリックし、次のタスクを実行します。

     - **GSLB** サイト**]** セクションでは、ありません GSLB サイト] をクリックします。
     - **[GSLB** サイトの選択] で、[> ノードグループにバインドする GSLB サイトを選択します。GSLB サイトを追加することもできます。

   - GSLB 仮想サーバーをノードグループにバインドします。

     [詳細 設定**]** で、[仮想サーバー] をクリックして、次のタスクを実行します。

     - [仮想サーバー**]** ペインで、[+.
     - [仮想サーバーの選択**]** で、ノードグループにバインドするサーバーを選択します。

   - ADNS (または ADNS-TCP) サービスまたは DNS (または DNS-TCP) 負荷分散仮想サーバーをノードグループにバインドします。

     [詳細 設定**]** で、[サービス] をクリックして、次のタスクを実行します。

     - **[**サービス**]** セクションでは、ありません **Service]** をクリックします。
     - **[**サービスの選択] で、ノードグループにバインドするサービスを選択します。サービスを追加することもできます。

> 注
>
> 子サイトの場合、クラスターノードとローカル GSLB サイトをノードグループにバインドするだけで済みます。

クラスターでのキャッシュリダイレクトの使用

October 7, 2021

クラスター内のキャッシュリダイレクトは、スタンドアロンの Citrix ADC アプライアンスと同じように機能します。唯一の違いは、構成がクラスタ IP アドレスで実行されることです。キャッシュリダイレクトの詳細については、「キャッシュリダイレクト」を参照してください。

クラスターでトランスペアレントモードでキャッシュのリダイレクトを使用する際の注意点**:**

- キャッシュリダイレクトを設定する前に、すべてのノードを外部スイッチに接続し、リンクセットが設定されていることを確認します。それ以外の場合、クライアント要求はドロップされます。
- 負荷分散仮想サーバで MAC モードが有効になっている場合は、enable ns mode MBF コマンドを使用して、クラスタで MBF モードが有効になっていることを確認します。それ以外の場合、要求はキャッシュサーバーに送信されるのではなく、オリジンサーバーに直接送信されます。

クラスターセットアップでの **L2** モードの使用

October 7, 2021

> 注
>
> NetScaler 10.5 以降のリリースでサポートされています。

クラスター設定で L2 モードを使用するには、次のことを確認する必要があります。

- 必要に応じて、すべてのノードでスポッティング IP アドレスを使用できる必要があります。
- 外部ネットワークとの通信には、リンクセットを使用する必要があります。
- 非対称トポロジまたは非対称クラスタ LA グループはサポートされません。
- クラスタ LA グループを推奨します。
- トラフィックは、サービスが存在するデプロイメントに対してのみ、クラスタノード間で分散されます。

リンクセットでのクラスター **LA** チャネルの使用

October 7, 2021

非対称クラスタートポロジでは、一部のクラスターノードがアップストリームネットワークに接続されていません。そのような場合は、リンクセットを使用する必要があります。パフォーマンスを最適化するには、スイッチに接続されているインターフェイスをクラスタ LA チャネルとしてバインドし、チャネルをリンクセットにバインドします。

クラスタ LA チャネルとリンクセットの組み合わせの使用方法を理解するために、アップストリームスイッチで使用できるポートが 2 つしかない 3 ノードクラスタについて考えてみます。2 つのクラスタノードをスイッチに接続し、もう一方のノードを接続しないままにしておくことができます。

> 注
>
> 同様に、非対称トポロジで ECMP とリンクセットの組み合わせを使用することもできます。

図 1: リンクセットとクラスター LA チャネルトポロジ



CLI を使用してクラスター **LA** チャネルとリンクセットを構成するには

1. クラスタ IP アドレスにログオンします。
2. 接続されたインターフェイスをクラスタ LA チャネルにバインドします。

```
1 add channel CLA/1 -ifnum 0/1/2 1/1/2
```

3. クラスタ LA チャネルをリンクセットにバインドします。


```
add linkset LS/1 -ifnum CLA/1
```

LA チャネル上のバックプレーン

October 7, 2021

この配置では、LA チャネルがクラスタバックプレーンに使用されます。



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

バックプレーンインターフェイスを **LA** チャネルとしてクラスタを展開するには

1. ノード NS0、NS1、NS2 のクラスタを作成します。

   a) クラスタに追加する最初のノードにログオンし、次の操作を行います。

```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

   b) クラスタの IP アドレスにログオンし、次の操作を行います。


```
> add cluster node 1 10.102.29.70 -state ACTIVE
> add cluster node 2 10.102.29.80 -state ACTIVE
```

c) ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

2. クラスタの IP アドレスにログオンし、次の操作を行います。

a) ノード NS0 および NS1 の LA チャネルを作成します。

```
> add channel 0/LA/1 -ifnum 0/1/1 0/1/2
> add channel 1/LA/2 -ifnum 1/1/1 1/1/2
```

b) クラスタノードのバックプレーンを設定します。

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/1/1
```

クライアントとサーバ用の共通インタフェース、およびバックプレーン用の専用インタフェース

October 7, 2021

これは、Citrix ADC クラスターのワンアーム展開です。この展開では、クライアントネットワークとサーバーネットワークが同じインターフェイスを使用してクラスタと通信します。クラスタバックプレーンは、ノード間通信専用のインターフェイスを使用します。



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

  クライアントとサーバ用の共通のインターフェイスと、クラスタバックプレーン用の異なるインターフェイスを持つクラスタを展開するには

  1. ノード NS0、NS1、NS2 のクラスタを作成します。
  2. クラスタに追加する最初のノードにログオンし、次の操作を行います。

```
1 > create cluster instance 1
2 > add cluster node 0 10.102.29.60 -state ACTIVE -backplane
     0/1/1
3 > enable cluster instance 1
4 > add ns ip 10.102.29.61 255.255.255.255 -type CLIP
5 > save ns config
6 > reboot -warm
```

  3. クラスタの IP アドレスにログオンし、次の操作を行います。

```
1 > add cluster node 1 10.102.29.70 -state ACTIVE -backplane
     1/1/1
```



```
2  > add cluster node 2 10.102.29.80 -state ACTIVE -backplane
       2/1/1
```

4. ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
1  > join cluster -clip 10.102.29.61 -password nsroot
2  > save ns config
3  > reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

1. クラスタ IP アドレスで、バックプレーンインターフェイスおよびクライアントインターフェイスおよびサーバインターフェイス用の VLAN を作成します。

   //バックプレーンインタフェースの場合

```
1  > add vlan 10
2  > bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //クライアントおよびサーバーネットワークに接続されているインターフェイス用。

```
1  > add vlan 20
2  > bind vlan 20 0/1/2 1/1/2 2/1/2
```

2. スイッチ上で、バックプレーンインターフェイス、およびクライアントおよびサーバインターフェイスに対応するインターフェイスの VLAN を作成します。次に、Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチ用の設定例を示します。他のスイッチでも同様の設定を行う必要があります。

   //バックプレーンインターフェイスの場合。Repeat for each interface…

```
1  > interface Ethernet2/47
2    switchport access vlan 100
3    switchport mode access
4    end
```

   //クライアントおよびサーバーネットワークに接続されたインターフェースの場合。Repeat for each interface…


```
> interface Ethernet2/47
  switchport access vlan 200
  switchport mode access
  end
```

クライアント、サーバ、バックプレーン用の共通スイッチ

October 7, 2021

この展開では、クライアント、サーバ、およびバックプレーンは同じスイッチ上の専用インターフェイスを使用して、Citrix ADC クラスタと通信します。



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

クライアント、サーバ、およびバックプレーン用の共通スイッチを使用してクラスタを展開するには

1. ノード NS0、NS1、NS2 のクラスタを作成します。
2. クラスタに追加する最初のノードにログオンし、次の操作を行います。


```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
   0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

3. クラスタの IP アドレスにログオンし、次の操作を行います。

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

4. ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

1. クラスタ IP アドレスで、バックプレーン、クライアント、およびサーバインターフェイスの VLAN を作成します。

//バックプレーンインタフェースの場合

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

//クライアント側インタフェースの場合

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

//サーバー側インタフェースの場合


```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

2. スイッチ上で、バックプレーンインターフェイス、およびクライアントおよびサーバインターフェイスに対応するインターフェイスの VLAN を作成します。次に、Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチ用の設定例を示します。他のスイッチでも同様の設定を行う必要があります。

   //バックプレーンインターフェイスの場合。Repeat for each interface…

```
> interface Ethernet2/47
  switchport access vlan 100
  switchport mode access
  end
```

   //クライアントインターフェース用。Repeat for each interface…

```
> interface Ethernet2/48
  switchport access vlan 200
  switchport mode access
  end
```

   //サーバーインターフェース用。Repeat for each interface…

```
> interface Ethernet2/49
  switchport access vlan 300
  switchport mode access
  end
```

クライアント/サーバ用共通スイッチ、バックプレーン専用スイッチ

October 7, 2021

この展開では、クライアントとサーバーは同じスイッチ上で異なるインターフェイスを使用して Citrix ADC クラスタと通信します。クラスタバックプレーンは、ノード間通信専用のスイッチを使用します。



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

  クライアントとサーバには同じスイッチを使用し、クラスタバックプレーンには別のスイッチを使用してクラスタを展開するには

1. ノード NS0、NS1、NS2 のクラスタを作成します。
   - クラスタに追加する最初のノードにログオンし、次の操作を行います。


```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
   0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

- クラスタの IP アドレスにログオンし、次の操作を行います。

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

- ノード 10.102.29.70 および 10.102.29.80 にログオンし、ノードをクラスタに参加させます。

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

前のコマンドに見られるように、インターフェース 0/1/1, 1/1/1, そして 2/1/1 3 つのクラスターノードのバックプレーンインターフェイスとして構成されます。

2. クラスタ IP アドレスで、バックプレーン、クライアント、およびサーバインターフェイスの VLAN を作成します。

//バックプレーンインタフェースの場合

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

//クライアント側インタフェースの場合

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

//サーバー側インタフェースの場合


```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

3. スイッチ上で、バックプレーンインターフェイス、およびクライアントおよびサーバインターフェイスに対応するインターフェイスの VLAN を作成します。次に、Cisco® Nexus 7000 C7010 リリース 5.2 (1) スイッチ用の設定例を示します。他のスイッチでも同様の設定を行う必要があります。

   //バックプレーンインターフェイスの場合。Repeat for each interface…

```
> interface Ethernet2/47
> switchport access vlan 100
> switchport mode access
> end
```

   //クライアントインターフェース用。Repeat for each interface…

```
> interface Ethernet2/48
> switchport access vlan 200
> switchport mode access
> end
```

   //サーバーインターフェース用。Repeat for each interface…

```
> interface Ethernet2/49
> switchport access vlan 300
> switchport mode access
> end
```

ノードごとに異なるスイッチ

October 7, 2021

この展開では、各クラスタノードは異なるスイッチに接続され、スイッチ間にトランクリンクが設定されます。



クラスター構成は、他のデプロイメントシナリオと同じです。クライアント側の構成のほとんどは、クライアント側のスイッチで行われます。

クラスタ構成の例

October 7, 2021

次の例は、ECMP、クラスター LA、またはリンクセットを使用して 4 ノードクラスタを構成するために使用できます。

1. クラスターを作成します。
   - 最初のノードにログオンします。
   - クラスターインスタンスを追加します。

```
1   > add cluster instance 1
```

   - クラスタに最初のノードを追加します。

```
1   > add cluster node 0 10.102.33.184 -backplane 0/1/1
```


- クラスタインスタンスを有効にします。

```
1   > enable cluster instance 1
```

- クラスタ IP アドレスを追加します。

```
1   > add ns ip 10.102.33.185 255.255.255.255 -type CLIP
```

- 構成を保存します。

```
1   > save ns config
```

- アプライアンスをウォーム再起動します。

```
1   > reboot -warm
```

2. 他の 3 つのノードをクラスターに追加します。
   - クラスタ IP アドレスにログオンします。
   - 2 番目のノードをクラスターに追加します。

```
1   > add cluster node 1 10.102.33.187 -backplane 1/1/1
```

   - 3 番目のノードをクラスタに追加します。

```
1   > add cluster node 2 10.102.33.188 -backplane 2/1/1
```

   - 4 番目のノードをクラスターに追加します。

```
1   > add cluster node 3 10.102.33.189 -backplane 3/1/1
```

3. 追加されたノードをクラスターに参加させます。この手順は、最初のノードには適用されません。
   - 新しく追加された各ノードにログオンします。

- ノードをクラスタに参加させます。

```
1   > join cluster -clip 10.102.33.185 -password nsroot
```

- 構成を保存します。

```
1   > save ns config
```

- アプライアンスをウォーム再起動します。

```
1   > reboot -warm
```

4. クラスター IP アドレスを使用して Citrix ADC クラスターを構成します。

   // 負荷分散機能を有効にする

```
1      > enable ns feature lb
```

   // 負荷分散仮想サーバーを追加します

```
1      > add lb vserver first_lbvserver http
2      ....
3      ....
```

5. クラスタに対して次のいずれかの（ECMP、クラスタ LA、または Linkset）トラフィック分散メカニズムを設定します。

   ECMP

   - クラスタ IP アドレスにログオンします。
   - OSPF ルーティングプロトコルを有効にします。

```
1   > enable ns feature ospf
```

   - VLAN を追加します。


```
1   > add vlan 97
```

- クラスタノードのインターフェイスを VLAN にバインドします。

```
1   > bind vlan 97 -ifnum 0/1/4 1/1/4 2/1/4 3/1/4
```

- 各ノードにスポッティング SNIP を追加し、そのノードで動的ルーティングを有効にします。

```
1   > add ns ip 1.1.1.10 255.255.255.0 -ownerNode 0 -
       dynamicRouting ENABLED
2   > add ns ip 1.1.1.11 255.255.255.0 -ownerNode 1 -
       dynamicRouting ENABLED
3   > add ns ip 1.1.1.12 255.255.255.0 -ownerNode 2 -
       dynamicRouting ENABLED
4   > add ns ip 1.1.1.13 255.255.255.0 -ownerNode 3 -
       dynamicRouting ENABLED
```

- SNIP アドレスのいずれかを VLAN にバインドします。

```
1   > bind vlan 97 -ipAddress 1.1.1.10 255.255.255.0
```

- VTYSH シェルを使用して、ZebOS でルーティングプロトコルを構成します。

  静的クラスタ **LA**

  - クラスタ IP アドレスにログオンします。
  - クラスタ LA チャネルを追加します。

    ```
    1   > add channel CLA/1 -speed 1000
    ```

  - インターフェイスをクラスタ LA チャネルにバインドします。

    ```
    1   > bind channel CLA/1 0/1/5 1/1/5 2/1/5 3/1/5
    ```

  - スイッチで同等の設定を実行します。

    動的クラスタ **LA**

* クラスタ IP アドレスにログオンします。
* クラスタの LA チャネルにインターフェイスを追加します。

```
> set interface 0/1/5 -lacpmode active -lacpkey 5 -
   lagtype cluster
> set interface 1/1/5 -lacpmode active -lacpkey 5 -
   lagtype cluster
> set interface 2/1/5 -lacpmode active -lacpkey 5 -
   lagtype cluster
> set interface 3/1/5 -lacpmode active -lacpkey 5 -
   lagtype  cluster
```

* スイッチで同等の設定を実行します。

リンクセット。nodeId 3 のノードがスイッチに接続されていないと仮定します。接続されていないノードが他のノードインターフェイスを使用してスイッチと通信できるように、リンクセットを設定する必要があります。

a) クラスタ IP アドレスにログオンします。

b) リンクセットを追加します。

```
> add linkset LS/1
```

c) 接続されたインターフェイスをリンクセットにバインドします。

```
> bind linkset LS/1 -ifnum 0/1/6 1/1/6 2/1/6
```

6. クラスタノードの状態を ACTIVE に更新します。

```
> set cluster node 0 -state ACTIVE
> set cluster node 1 -state ACTIVE
> set cluster node 2 -state ACTIVE
> set cluster node 3 -state ACTIVE
```

クラスタセットアップでの **VRRP** の使用

October 7, 2021

仮想ルータ冗長プロトコル（VRRP）は、IPv4 と IPv6 の両方のクラスタセットアップでサポートされます。クラスタセットアップでサポートされる VRRP 機能は、インターフェイスベースの VRRP と IP ベースの VRRP の 2 つです。

IP ベースの VRRP

IP ベースの VRRP では、同じ VRID にバインドされたストライプ VIP アドレスが、クラスタセットアップのすべてのノードに設定されます。これらの VIP アドレスはすべてのノードでアクティブです。

クラスタノードの 1 つが VRID 所有者として機能し、VRRP アドバタイズメントを他のノードに送信します。VRID 所有者ノードに障害が発生した場合、クラスタ内の別のノードが VRID の所有権を引き受け、VRRP アドバタイズメントの送信を開始します。特定のクラスタノードを VRID の所有者として割り当てることもできます。

> 注
>
> クラスタ内の VRRP 展開には、IP ベースの方法を使用することをお勧めします。

IPv4 用の IP ベースの VRRP の設定

IPv4 用の IP ベースの VRRP を設定するためのクラスタ設定で、次のタスクを実行します。

- **VRID** を追加します。VRID は、クラスタセットアップによって仮想 MAC アドレスを形成するために使用される整数です。汎用 VMAC アドレスは、00:00:5 e: 00:02: の形式 <VRID> になります。
- （任意）仮想 **MAC** アドレスの所有者としてノードを割り当てます。クラスタノードの ID に owner node パラメータ（VRID6 の追加または変更中）を設定し、仮想 MAC アドレスの所有者として割り当てることができます。割り当てられた所有者ノードに障害が発生すると、UP クラスタノードの 1 つが仮想 MAC アドレスの所有者として動的に選択されます。オーナーノードは、`set vrID <id> -ownerNode <positive_interger>` コマンドを使用して設定できます。
- **VRID** をノードの **VIP** アドレスにバインドします。作成した VRID をストライプ VIP アドレスにバインドします。

CLI を使用して VRID を追加するには

コマンドプロンプトで入力します。

```
1 -  add vrid <ID> [-ownerNode <positive_integer>]
2 -  show vrid <ID>
```

CLI を使用して VRID を VIP アドレスにバインドするには

コマンドプロンプトで入力します。

- `set ns ip <IPv4Address> -vrid <ID><!--NeedCopy-->`

- show vrid <ID><!--NeedCopy-->

GUI を使用して **VRID** を追加するには

1. **[システム] > [ネットワーク] > [VMAC]** に移動し、[**VMAC**] タブで [追加] をクリックします。
2. [**VMAC** の作成] ページで、[仮想ルータ **ID**] フィールドに値を指定し、[作成] をクリックします。

GUI を使用して **VRID** を **VIP** アドレスにバインドするには

1. **[システム] > [ネットワーク] > [IP]** に移動し、[**IPV4s**] タブで VIP アドレスを選択し、[編集] をクリックします。
2. VIP 設定の編集中に、仮想ルータ **ID** パラメータを設定します。

```
1 > add vrid 90
2  Done
3 > set ns ip 192.0.2.90  - vrid 90
4  Done
```

IPv6 用の IP ベースの VRRP の設定

IPv6 用の IP ベースの VRRP を設定するためのクラスタ設定で、次の作業を実行します。

- **VRID6** を追加します。VRID6 は、クラスタセットアップによって仮想 MAC6 アドレスを形成するために使用される整数です。一般的な VMAC6 アドレスは、00:00:5 e: 00:02: の形式 <VRID6> になります。
- （オプション）仮想 **MAC6** アドレスの所有者としてノードを割り当てます。クラスタノードの ID に owner node パラメータ（VRID6 の追加または変更中）を設定し、仮想 MAC6 アドレスの所有者として割り当てることができます。割り当てられた所有者ノードに障害が発生すると、UP クラスタノードの 1 つが仮想 MAC6 アドレスの所有者として動的に選択されます。
- **VRID6** をノードの **VIP6** アドレスにバインドします。作成した VRID6 をストライプ VIP6 アドレスにバインドします。

CLI を使用して VRID6 を追加するには

コマンドプロンプトで入力します。

- add vrid6 <ID> [-ownerNode <positive_integer>]<!--NeedCopy-->
- show vrid6 <ID><!--NeedCopy-->

CLI を使用して VRID6 を VIP6 アドレスにバインドするには

コマンドプロンプトで入力します。

- `set ns ip6 <IPv6Address> -vrid6 <ID><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

GUI を使用して VRID6 を追加するには

1. **[システム] > [ネットワーク] > [VMAC]** に移動し、[**VMAC6**] タブで [追加] をクリックします。
2. [仮想 **MAC6** の作成] ページで、[仮想ルータ **ID**] フィールドに値を指定し、[作成] をクリックします。

GUI を使用して VRID6 を VIP6 アドレスにバインドするには

1. **[システム] > [ネットワーク] > [IP]** に移動し、[**IPV6s**] タブで VIP アドレスを選択し、[編集] をクリックします。
2. VIP6 設定の編集中に、仮想ルータ **ID** パラメータを設定します。

```
> add vrid6 90
 Done
> set ns ip6 2001:db8::5001  - vrid6 90
 Done
```

インターフェイスベースの VRRP

インターフェイスベースの VRRP 機能では、クラスタの両方のノードに同じ仮想 MAC アドレスが設定されます。この仮想 MAC アドレスは、ノードで設定された IP アドレスの GARP アドバタイズメントおよび ARP 応答で使用されます。この機能は、GARP アドバタイズメントを受け入れない外部デバイス/ルータがあるアクティブスペアの 2 ノードクラスタ設定に役立ちます。

> 注
>
> インターフェイスベースの VRRP 機能は、1 つのノードがアクティブステートで、もう 1 つのノードがスペアとして機能する 2 ノードクラスタにのみ適用できます。

両方のクラスタノードで同じ仮想 MAC アドレスを使用して、アクティブノードがダウンしてスペアノードがアクティブとして引き継ぐと、新しいアクティブノードの IP アドレスの MAC アドレスは変更されず、外部デバイス/ルータの ARP テーブルを更新する必要はありません。

IPv4 用のインターフェイスベース VRRP の設定

IPv4 のインターフェイスベースの VRRP を設定するには、クラスタセットアップで次の作業を実行します。

- **VRID** を追加します。VRID は、クラスタセットアップによって仮想 MAC アドレスを形成するために使用される整数です。
- **VRID** をノードインターフェイスにバインドします。作成した VRID にインターフェイスをバインドします。（現在のアクティブノードの）バインドされたインターフェイスは、その IPv4 アドレスに対して GARP アドバタイズメントおよび ARP 応答の仮想 MAC アドレスを使用します。VRID は、アクティブスペアクラスタセットアップの両方のノードのインターフェイスに関連付ける必要があります。これは、高可用性セットアップとは異なり、クラスタセットアップではインターフェイス ID が異なるためです。

CLI を使用して **VRID** を追加するには

コマンドプロンプトで入力します。

```
1 -  add vrid <ID>
2 -  show vrid <ID>
```

CLI を使用して **VRID** をインターフェイスにバインドするには

コマンドプロンプトで入力します。

```
1 -  bind vrid <ID> -ifnum <interface_name>
2 -  show vrid <ID>
```

GUI を使用して **VRID** を追加し、インターフェイスにバインドするには

1. **[システム] > [ネットワーク] > [VMAC]** に移動し、[**VMAC**] タブで [追加] をクリックします。
2. [仮想 **MAC** の作成] ページで、[仮想ルータ **ID***] フィールドに値を指定し、[インターフェイスの関連付け] セクションでインターフェイスをバインドして、[作成] をクリックします。

```
1 > add vrid 300
2  Done
3 > bind vrid 300 -ifnum 1/1/2 2/1/3
4  Done
```

IPv6 用のインターフェイスベースの VRRP の設定

IPv6 用のインターフェイスベースの VRRP を設定するには、クラスタセットアップで次の作業を実行します。

- **VRID6** を追加します。VRID6 は、クラスタセットアップによって仮想 MAC6 アドレスを形成するために使用される整数です。一般的な VMAC6 アドレスは、00:00:5 e: 00:01: という形式 <VRID6> になります。
- **VRID6** をノードインターフェイスにバインドします。作成した VRID6 にインターフェイスをバインドします。（現在のアクティブノード内の）バインドされたインターフェイスは、GARP アドバタイズメント内の仮想 MAC6 アドレスと IPv6 アドレスの ARP 応答を使用します。アクティブスペアクラスタセットアップの両方のノードのインターフェイスに VRID6 を関連付ける必要があります。これは、高可用性セットアップとは異なり、クラスタセットアップではインターフェイス ID が異なるためです。

CLI を使用して **VRID6** を追加するには

コマンドプロンプトで入力します。

```
1 -  add vrid6 <ID>
2 -  show vrid6 <ID>
```

CLI を使用して **VRID6** をインターフェイスにバインドするには

コマンドプロンプトで入力します。

- `bind vrid6 <ID> -ifnum <interface_name><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

GUI を使用して **VRID6** を追加し、インターフェイスにバインドするには

1. **[システム] > [ネットワーク] > [VMAC]** に移動し、[**VMAC6**] タブで [追加] をクリックします。
2. [仮想 **MAC6** の作成] ページで、[仮想ルータ **ID**] フィールドに値を指定し、[インターフェイスの関連付け] セクションでインターフェイスをバインドして、[作成] をクリックします。

```
1 > add vrid6 100
2  Done
3 > bind vrid6 100 -ifnum 0/1/1 1/1/2 2/1/3
4  Done
```

パス監視を使用したクラスタ内のサービスの監視

October 7, 2021

クラスタ設定では、モニタリングサービスの所有権がノード間で分散されます。したがって、異なるノードは異なるサービスを監視します。サービスを監視するノードは、サービス所有者と呼ばれます。サービス所有者だけがサーバーをプローブし、サーバーに割り当てられたサービスのステータスを監視します。さらに、クラスタ内の他のすべてのノードにサービスのステータスを伝達します。分散型監視の欠点は、すべてのノードとサーバー間のネットワーク接続とリンク状態が決定されないことです。この欠点を克服するには、パスモニタリングを使用できます。

> 注
>
> サービスを監視するノードを選択することはできません。サービスを監視するノードの選択は、内部メカニズムを介して行われます。`show service <service name>` および `show serviceGroup <service group name>` コマンドを使用すると、サービスを監視する所有者ノードを確認できます。

パスモニタリングは、ノードとサーバーによって提供されるサービスの間のネットワーク接続とリンク状態をチェックします。ノードは ICMP ping を送信して、サーバが到達可能かどうかを検証します。

パス監視の仕組み

3 つのノード N1、N2、および N3 で構成される Citrix ADC クラスタの例を考えてみましょう。N2 は、HTTP サービス（S1）の状態を監視するサービス所有者です。サービス状態をクラスタ内の他のノードにアドバタイズします。パス監視は、クラスタ内のすべてのノードで、すべてのサービスに対して有効になります。各ノードは ICMP ping だけをサーバに送信します。サービスの所有者は、HTTP サービスリクエストと ICMP ping の両方を送信します。各ノードは、パス監視状態をサービス所有者に報告します。



次の 2 つのパラメーターは、ノードのサービス状態を決定します。

- S = サービス所有者によってアドバタイズされたサービス状態
- P = 各ノードのパス監視状態

ノードがサーバに到達できるかどうかは、そのノードのパス監視状態を決定します。

次の表は、pathMonitorIndv パラメータが有効または無効になっている場合の、パス監視状態に基づいて設定されたサービス状態を示しています。

パラメーター	パス監視の状態	サービスの状態
pathMonitorIndv = 番号; デフォルトの構成です。	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = DOWN
	P3 = UP	S1 = DOWN
pathMonitorIndv = YES	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = UP
	P3 = UP	S1 = UP

この例では、サービス所有者は、パス監視状態が DOWN に設定されているノードに基づいて、すべてのノードのサービス状態を決定します。いずれかのノードのパス監視状態が DOWN の場合、サービス所有者はすべてのノードのサービス状態を DOWN に設定します。すべてのノードのサービス状態は、各ノードのパス監視状態が UP の場合にのみ UP に設定されます。

pathMonitorIndv パラメーターを有効にすると、個々のノードのパス監視を使用できます。このパラメータを使用すると、サービス所有者は、各ノードのパス監視状態に基づいて、各ノードのサービス状態を設定できます。

> 注
>
> pathMonitorIndv パラメーターが設定されている場合、永続性などの一部の機能が機能しなくなる可能性があります。

パス監視の設定

パスモニタリングは、すべてのサービスおよびサービスグループに適用できます。パスモニタリングパラメータは、デフォルトで無効になっています。

CLI を使用してサービス/サービスグループのパスモニタリングを有効にするには

コマンドプロンプトで入力します。

```
add service <service name> <IP address> <service type> <port> [-
    pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]

add servicegroup <servicegroup name> <service type> [-pathMonitor <YES
    | NO>] [-pathMonitorIndv <YES | NO>]
```



```
<!--NeedCopy-->
```

例:

```
add service s1 1.1.1.1 HTTP 80 -pathMonitor YES
add servicegroup sg_1 HTTP -pathMonitor YES

add service s1 1.1.1.1 HTTP 80 -pathMonitor YES -pathMonitorIndv YES
add servicegroup sg_1 HTTP -pathMonitor YES -pathMonitorIndv YES
<!--NeedCopy-->
```

次のように、set コマンドからパスモニタリングパラメータを設定することもできます。

```
set service <service name> [-pathMonitor <YES | NO>] [-pathMonitorIndv
    <YES | NO>]
set servicegroup <servicegroup name> [-pathMonitor <YES | NO>] [-
    pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

例:

```
set service s1 -pathMonitor YES
set servicegroup sg_1 -pathMonitor YES


set service s1 -pathMonitorIndv YES
set servicegroup sg_1 -pathMonitorIndv NO
<!--NeedCopy-->
```

GUI を使用してサービス/サービスグループのパスモニタリングを有効にするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。

   サービスグループの場合は、[トラフィック管理] > [ロードバランシング] > [サービスグループ] に移動します。

2. [サービス/サービスグループ] ペインで、リストからサービス/ サービスグループを選択し、ダブルクリックして開きます。

3. [サービスの設定] タブで、[編集] をクリックします。

4. [パス監視] を選択します。

5. 適用する場合は、[個別のパス監視] を選択し、[**OK**] をクリックします。

> 注
>
> 個々のパスモニタリングを有効にできるのは、パスモニタリングを有効にしている場合だけです。

クラスタセットアップのバックアップと復元

October 7, 2021

Citrix ADC クラスターノードの現在の状態をバックアップできます。後で、バックアップされたファイルを使用して、ノードを同じクラスター状態に復元できます。予防措置として、クラスタノードでアップグレードを実行する前に、この機能を使用する必要があります。

クラスターのセットアップをバックアップする

以下に応じて、基本バックアップまたは完全バックアップを作成できます。

- バックアップするデータの種類。
- バックアップを作成する頻度。
- 基本的なバックアップ。構成ファイルのみをバックアップします。バックアップするファイルは常に変更するため、頻繁にこのタイプのバックアップの実行が必要になる場合があります。バックアップされるファイルが表に一覧表示されます。

ディレクトリ

サブディレクトリまたはファイル

/nsconfig/

- ns.conf
- ZebOS.conf
- rc.netscaler
- snmpd.conf
- nsbefore.sh
- nsafter.sh
- inetd.conf
- ntp.conf
- syslog.conf
- newsyslog.conf
- crontab
- host.conf
- hosts

- ttys
- sshd_config
- httpd.conf
- monitrc
- rc.conf
- ssh_config
- ローカル時間
- issue
- issue.net

/var/

- download/*
- log/wicmd.log
- wi/tomcat/webapps/*
- wi/tomcat/logs/*
- wi/tomcat/conf/catalina/localhost/*
- nslw.bin/etc/krb.conf
- nslw.bin/etc/krb.keytab
- netscaler/locdb/*
- lib/likewise/db/*
- vpn/bookmark/*
- netscaler/crl
- nstemplates/*
- learnt_data/*

/netscaler/

- custom.html
- vsr.html
- フル・バックアップ。基本バックアップによってバックアップされるファイルとは別に、完全バックアップは、更新頻度の低いファイルをバックアップします。フルバックアップオプションの使用時にバックアップされるファイルは、表に一覧表示されます。

ディレクトリ

サブディレクトリまたはファイル

/nsconfig/

- ssl/*
- license/*
- fips/*

/var/

- netscaler/ssl/*
- wi/java_home/jre/lib/security/cacerts/*
- Wi/java_home/lib/セキュリティ/カサルト/*

> 重要
>
> CLAG が SDX クラスター設定で構成されている場合、バックアップと復元は機能しません。

バックアップは、圧縮された TAR ファイルとして /var/ns_sys_backup/ ディレクトリに保存されます。ディスクスペース不足による問題を避けるために、このディレクトリに格納できるバックアップファイルは最大 50 個となっています。rm system backup コマンドを使用して、既存のバックアップファイルを削除し、さらにバックアップを作成できます。

クラスター設定の CLIP でバックアップ操作を実行すると、各クラスターノードにバックアップファイルが作成されます。

クラスター設定をバックアップする方法

Citrix ADC CLI を使用して、CLIP でクラスター設定をバックアップします。

コマンドプロンプトで、次の操作を行います。

- 構成を保存します。

  ```
  save ns config<!--NeedCopy-->
  ```

- バックアップファイル (基本または完全) を作成します。

  “‘create system backup [][–level (basic | full)][-comment]

```
** 例 **

```create system backup cluster-backup-1  - level basic<!--
   NeedCopy-->
```

上記のコマンドは、指定されたファイル名で各クラスターノードにバックアップ TAR ファイルを作成します。たとえば、CLUSTER-backup-1.tgz ファイルは各クラスターノードに作成されます。

> 注
>
> ファイル名が指定されていない場合、バックアップ TAR ファイルは、次の命名規則に従って各クラスターノードに作成されます。
>
> - backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp >.tgz<!--NeedCopy-->
> - backup_<level>_<nsip_address of the cluster node 1>_<date-timestamp


```

>.tgz<!--NeedCopy-->

たとえば、3 ノードクラスタのセットアップでは、

- backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp >.tgz<!--NeedCopy-->は node0 に作成されます。
- backup _<level>_<nsip_address of the cluster node 1>_<date-timestamp >.tgz<!--NeedCopy-->ノード 1 に作成されます
- backup_<level>_<nsip_address of the cluster node 2>_<date-timestamp >.tgz<!--NeedCopy-->はノード 2 に作成されます

- 作成したバックアップファイルを CLIP で確認します。

  ```
  show system backup<!--NeedCopy-->
  ```

クラスタのセットアップを復元する

クラスタノードに障害が発生した場合、このノードを新しいノードに置き換えることができます。障害のあるノードのバックアップファイルを使用して、クラスターの新しいノードを設定できます。

たとえば、3 ノードのクラスタ設定で node1 が障害になった場合、この障害のあるノードを node1 という新しいノードに置き換えることができます。復元操作を使用すると、障害のあるノードのバックアップファイルのいずれかを新しいノードに復元できます。

> 注
>
> バックアップファイルの名前が変更された場合、またはファイルの内容が変更された場合、復元操作は成功しません。

クラスタノードを復元する方法

CLI を使用してクラスターノードを復元するには

コマンドプロンプトで、次の操作を行います。

- CLIP で使用可能なバックアップファイルのリストを取得します。

  ```
  show system backup<!--NeedCopy-->
  ```

- バックアップ tar ファイルをにコピーします /var/ns_sys_backup 復元されるクラスターノードのディレクトリ。

- クラスタノードで次のコマンドを実行して、クラスタノードのメモリにバックアップ tar ファイルを追加します。

  “‘add system backup


```
** 例 **

```add system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
```

> 注
>
> このコマンドは、復元するクラスタノードで実行する必要があります。

- バックアップファイルを指定して、クラスタノードを復元します。

  “‘restore system backup

```
** 例 **

```restore system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
```

> 注
>
> このコマンドは、復元するクラスタノードで実行する必要があります。

- クラスタノードを再起動します。

  リブート

> 注
>
> このコマンドは、復元するクラスタノードで実行する必要があります。

Citrix ADC クラスターのアップグレードまたはダウングレード

October 7, 2021

Citrix ADC クラスタのすべてのノードで、同じソフトウェアバージョンが実行されている必要があります。したがって、クラスタをアップグレードまたはダウングレードするには、クラスタの各 Citrix ADC アプライアンスを一度に 1 つずつアップグレードまたはダウングレードする必要があります。

アップグレードまたはダウングレードされているノードは、クラスタから削除されません。ノードはクラスターの一部であり、アップグレードまたはダウングレード後にノードが再起動するときのダウンタイムを除いて、トラフィックを中断することなく提供します。

ただし、クラスターノード間のソフトウェアバージョンの不一致により、クラスターでの構成の伝達は無効になっています。構成の伝播は、すべてのクラスターノードが同じバージョンである場合にのみ有効になります。クラスタのダウングレード時にアップグレード中に設定の伝播が無効になるため、この間はクラスタ IP アドレスを通じて設定を実行できません。

重要

- 最大接続 (maxConn) グローバルパラメータをゼロ以外の値に設定したクラスタセットアップでは、次のいずれかの条件が満たされると、CLIP 接続が失敗することがあります。

```
- Upgrading the setup from Citrix ADC 13.0 76.x build to Citrix ADC 13.0 79.x build.
- Restarting the CCO node in a cluster setup running Citrix ADC 13.0 76.x build.
```

回避方法:

```
- クラスターセットアップをCitrix ADC 13.0 76.xビルドからCitrix ADC 13.0 79.xビルドにアップグレードする前に、最大接続（maxConn）グローバルパラメータをゼロに設定する必要があります。セットアップをアップグレードした後、maxConn パラメータを目的の値に設定し、構成を保存できます。
\- Citrix ADC 13.0 76.xビルドはクラスタセットアップには適していません。Citrix ADC 13.0 76.xビルドをクラスターセットアップに使用しないことをお勧めします。
```

- クラスター設定では、Citrix ADC アプライアンスが次の場合にクラッシュすることがあります。

```
- upgrading the setup from Citrix ADC 13.0 47.x or 13.0 52.x build to a later build, or
- upgrading the setup to Citrix ADC 13.0 47.x or 13.0 52.x build
```

回避策：アップグレードプロセス中に、次の手順を実行します。

```
\- すべてのクラスタノードを無効にし、各クラスタノードをアップグレードします。
\- すべてのノードのアップグレード後に、すべてのクラスタノードを有効にします。
```


クラスタをアップグレードまたはダウングレードする前に注意すべきポイント

- クラスタソフトウェアのバージョンのアップグレードまたはダウングレード中は、クラスタノードを追加できません。
- 個々のノードの NSIP アドレスを介してノードレベルの構成を実行できます。同期を維持するために、必ずすべてのノードで同じ構成を実行してください。
- クラスタのアップグレード中は、クラスタの IP アドレスから `start nstrace` コマンドを実行することはできません。ただし、NSIP アドレスを使用して個々のクラスターノードでこの操作を実行することにより、個々のノードのトレースを取得できます。
- Citrix ADC 13.0 76.x ビルドは、クラスターのセットアップには適していません。Citrix ADC 13.0 76.x ビルドをクラスターセットアップに使用しないことをお勧めします。
- Citrix ADC 13.0 47.x および 13.0 52.x ビルドは、クラスターセットアップには適していません。これは、これらのビルドではノード間通信に互換性がないためです。
- クラスターがアップグレードされている場合、アップグレードされたノードで、まだアップグレードされていないノードでは使用できない追加機能がアクティブになっている可能性があります。クラスタのアップグレード中に、ライセンスの不一致の警告が表示されます。この警告は、すべてのクラスターノードがアップグレードされると自動的に解決されます。

> 重要
>
> - 前のノードがアクティブになるまで待ってから、次のノードをアップグレードまたはダウングレードすることをお勧めします。
> - Citrix は、クラスター構成ノードが upgraded/downgraded 最後に、クラスター IP セッションの複数の切断を回避します。

クラスタノードのソフトウェアをアップグレードまたはダウングレードするには

1. クラスターが安定しており、すべてのノードで構成が同期されていることを確認します。
2. NSIP アドレスを通じて各ノードにアクセスし、次の手順を実行します。
   - クラスタノードをアップグレードまたはダウングレードします。アプライアンスのソフトウェアのアップグレードとダウングレードの詳細については、「NetScaler アプライアンスのアップグレードとダウングレード」を参照してください。
   - 構成を保存します。
   - アプライアンスを再起動します。
3. その他の各クラスターノードについて、手順 2 を繰り返します。

個々のクラスタノードでサポートされる操作

October 7, 2021

原則として、クラスタの一部である Citrix ADC アプライアンスは、NSIP アドレスから個別に構成することはできません。ただし、この規則の例外となる操作もあります。これらの操作は、NSIP アドレスから実行されると、他のクラスターノードに伝播されません。

操作は次のとおりです。

- cluster instance (set | rm | enable | disable)
- cluster node (set | rm)
- ns トレース（開始 | 公演 | やめる）
- interface (set | enable | disable)
- route (add | rm | set | unset)
- ARP（追加 | rm | 全て送る）
- force cluster sync
- sync cluster files
- NTP 同期を無効にする
- save ns config
- リブート
- shutdown

たとえば、クラスタノードの NSIP アドレスからコマンド `disable interface 1/1/1` を実行すると、そのノードでのみインターフェイスが無効になります。このコマンドは伝播されないため、インターフェイス 1/1/1 は他のすべてのクラスタノードで有効のままです。

異機種クラスタのサポート

October 7, 2021

Citrix ADC アプライアンスは、クラスター展開で異種クラスターをサポートします。異種クラスタは、異なる Citrix ADC ハードウェアのノードにまたがり、同じクラスタ内に異なるプラットフォームを組み合わせることができます。

> 重要
>
> 異機種クラスタの形成またはサポートは可能であり、MPX ハードウェアプラットフォームのみに制限されています。

異種クラスタのサポートと形成は、特定の Citrix ADC モデルによって異なります。次の表に、同数のパケットエンジンを持つ異種クラスタの形成でサポートされるプラットフォームを示します。

パケットエンジンの数	MPX ハードウェア・プラットフォーム	異機種混在クラスタを形成するためにサポートされる MPX ハードウェアプラットフォーム
5	MPX 11500	MPX 14020
7	MPX 11515	MPX 14040
9	MPX 11530	MPX 14060

次の表に、異種クラスタの形成でサポートされるプラットフォームを示します。このプラットフォームの数は等しくありません。

ハードウェア・プラットフォーム	異機種混在クラスタを形成するサポート対象のハードウェアプラットフォーム
MPX 150XX	MPX 140XX

異なる SSL チップセットにまたがるパケットエンジン数の異なる Citrix ADC MPX アプライアンスの異種クラスタ展開を形成する方法の詳細については、SSL オフロード構成の 「異機種間クラスタの展開」セクションを参照してください。

> （注）
>
> リリース 13.0 ビルド 47.x より前では、パケットエンジンの数が異なるノードから「joincluster」コマンドを実行すると、「CCO とローカルノード間のアクティブな PPE の数が一致しません」というエラーメッセージが表示されます。

注意事項

1. 追加の管理 CPU 設定は、すべてのクラスターノードで同じである必要があります。
2. 新しく追加されたノードは、既存のクラスタノードと同じデータプレーンとバックプレーン上の容量を持つ必要があります。
3. 異なる暗号をサポートするプラットフォームが混在している場合、クラスタは共通の暗号リストに同意します。

よくある質問

October 7, 2021

クラスタリングに関する FAQ のリスト。

1 つの Citrix ADC クラスタにいくつの Citrix ADC アプライアンスを含めることができますか?

Citrix ADC クラスタには、1 つのアプライアンス、または最大 32 台の Citrix ADC nCore ハードウェアまたは仮想アプライアンスを含めることができます。これらのノードは、クラスタノードの前提条件で指定された基準を満たす必要があります。

Citrix ADC アプライアンスを複数のクラスタの一部にすることはできますか?

いいえ。Citrix ADC アプライアンスは 1 つのクラスターにのみ属することができます。

クラスタ IP アドレスとは何ですか? サブネットマスクとは何ですか?

クラスタ IP アドレスは、Citrix ADC クラスタの管理アドレスです。すべてのクラスタ構成は、このアドレスを通じてクラスタにアクセスすることによって実行する必要があります。クラスタ IP アドレスのサブネットマスクは、255.255.255.255 に固定されています。

特定のクラスタノードをクラスタ構成コーディネータとして作成するにはどうすればよいですか

特定のノードをクラスタ構成コーディネータとして手動で設定するには、そのノードの優先順位を最も低い数値 (最も高い優先度) に設定する必要があります。理解するために、次の優先順位を持つ 3 つのノードを持つクラスタを考えてみましょう。

n1 - 29, n2 - 30, n3 - 31

ここで、n1 は構成コーディネータです。n2 を構成コーディネータにするには、その優先順位を n1 より低い値 (28 など) に設定する必要があります。構成を保存すると、n2 が構成コーディネータになります。

> 注
>
> 元の優先度値が 30 の n2 は、n1 がダウンすると構成コーディネーターになります。構成コーディネーターがダウンした場合に備えて、次に優先度の低いノードが選択されます。

クラスタのネットワークインターフェイスは、通常の 2 タプル（u/c）表記ではなく、3 タプル（n/u/c）表記で表されるのはなぜですか?

Citrix ADC アプライアンスがクラスタの一部である場合、インタフェースが属するノードを特定できる必要があります。したがって、クラスタノードのネットワークインターフェース命名規則は、u/c から n/u/c に変更されます。ここで、n はノード ID を表します。

クラスタノードのホスト名を設定するにはどうすればよいですか

クラスターノードのホスト名は、クラスター IP アドレスを介して **set nshostname** コマンドを実行して指定する必要があります。たとえば、ID 2 のクラスターノードのホスト名を設定するには、コマンドは次のとおりです。

ns ホスト名ホスト名 **1** を設定 -所有者ノード 2

Citrix ADC アプライアンスを自動的に検出してクラスタに追加することはできますか**?**

はい。構成ユーティリティを使用すると、構成コーディネータの NSIP アドレスと同じサブネットに存在するアプライアンスを検出できます。詳細については、「NetScaler アプライアンスの検出」を参照してください。

ノードが削除されたり、無効になったり、再起動またはシャットダウンされたり、非アクティブになったりすると、クラスタのトラフィック処理機能が影響を受けますか

はい。これらの操作のいずれかがクラスターのアクティブノードで実行されると、クラスターにはトラフィックを処理するノードが 1 つ少なくなります。また、このノード上の既存の接続も終了します。

私は複数のスタンドアロンアプライアンスを持っており、それぞれの設定が異なります。**1** つのクラスターに追加できますか

はい。異なる構成のアプライアンスを単一のクラスタに追加できます。ただし、アプライアンスがクラスタに追加されると、既存の設定はクリアされます。各アプライアンスで使用可能な構成を使用するには、次の作業を行う必要があります。

1. すべての構成に対して単一の *.conf ファイルを作成します。
2. 構成ファイルを編集して、クラスタ環境でサポートされていない機能を削除します。
3. インタフェースの命名規則を 2 タプル (u/c) 形式から 3 タプル (n/u/c) 形式に更新します。
4. batch コマンドを使用して、クラスターの構成コーディネーターノードに構成を適用します。

スタンドアロン **Citrix ADC** アプライアンスまたは **HA** セットアップの構成をクラスター化されたセットアップに移行できますか**?**

いいえ。クラスタ化されたセットアップにノードを追加すると、**clear ns config** コマンド（拡張オプション付き）を使用して、ノードの設定が暗黙的にクリアされます。さらに、SNIP アドレスとすべての VLAN 設定（デフォルト VLAN および NSVLAN を除く）がクリアされます。したがって、アプライアンスをクラスタに追加する前に、設定をバックアップすることをお勧めします。クラスタのバックアップ構成ファイルを使用する前に、次の作業を行う必要があります。

1. 構成ファイルを編集して、クラスタ環境でサポートされていない機能を削除します。
2. インタフェースの命名規則を 2 タプル (x/y) 形式から 3 タプル (x/y/z) 形式に更新します。
3. **batch** コマンドを使用して、クラスターの構成コーディネーターノードに構成を適用します。

バックプレーンインターフェイスは **L3VLAN** の一部ですか

はい。デフォルトでは、バックプレーンインターフェイスは、クラスタに設定されているすべての L3 VLAN 上に存在します。

異なるネットワークのノードを含むクラスタを構成するにはどうすればよいですか?

> 注
>
> NetScaler 11.0 以降でサポートされています。

異なるネットワークからのノードを含むクラスタは、L3 クラスタ（INC モードではクラスタとも呼ばれます）と呼ばれます。L3 クラスターでは、単一のネットワークに属するすべてのノードを単一のノードグループにグループ化する必要があります。したがって、クラスターに 3 つの異なるネットワークからそれぞれ 2 つのノードが含まれる場合、3 つのノードグループ（ネットワークごとに 1 つ）を作成し、これらの各ノードグループをそのネットワークに属するノードに関連付ける必要があります。構成情報については、クラスターをセットアップする手順を参照してください。

クラスタで **NSVLAN** を設定/設定解除するにはどうすればよいですか?

次のいずれかの操作を行います。

- NSVLAN をクラスターで使用できるようにするには、クラスターに追加する前に、各アプライアンスに同じ NSVLAN が構成されていることを確認してください。
- クラスタノードから NSVLAN を削除するには、まずクラスタからノードを削除してから、アプライアンスから NSVLAN を削除します。

一部の **Citrix ADC** ノードが外部ネットワークに接続されていないクラスターをセットアップしています。クラスタは正常に機能しますか?

はい。クラスターは linksets と呼ばれるメカニズムをサポートしています。これにより、接続されていないノードは、接続されたノードのインターフェイスを使用してトラフィックを処理できます。接続されていないノードは、クラスタバックプレーンを介して接続されたノードと通信します。詳細については、「リンクセットの使用」を参照してください。

MAC ベースフォワーディング **(MBF)** を必要とする展開は、クラスタ化されたセットアップでどのようにサポートできますか?

MBF を使用する展開では、リンクセットを使用する必要があります。詳細については、「リンクセットの使用」を参照してください。

クラスタノードの **NSIP** アドレスからコマンドを実行できますか

いいえ。NSIP アドレスを通じた個々のクラスタ・ノードへのアクセスは読み取り専用です。したがって、クラスタノードの NSIP アドレスにログオンすると、構成と統計情報のみを表示できます。何も設定できません。ただし、クラスターノードの NSIP アドレスから実行できる操作がいくつかあります。詳細については、個々のノードでサポートされる操作を参照してください。

クラスタ・ノード間の構成の伝播を無効にできますか**?**

いいえ。クラスタノード間のクラスタ構成の伝播を明示的に無効にすることはできません。ただし、ソフトウェアのアップグレードまたはダウングレード中に、バージョンの不一致エラーにより、設定の伝達が自動的に無効になることがあります。

Citrix ADC アプライアンスがクラスタの一部である場合、**NSIP** アドレスの変更や **NSVLAN** の変更は可能ですか**?**

いいえ。このような変更を行うには、まずクラスタからアプライアンスを削除し、変更を実行してから、クラスタにアプライアンスを追加する必要があります。

Citrix ADC クラスタは **L2** および **L3VLAN** をサポートしていますか**?**

はい。クラスタは、クラスタノード間の VLAN をサポートします。VLAN は、クラスタ IP アドレスに設定する必要があります。

- **L2 VLAN**。クラスタの異なるノードに属するインターフェイスをバインドすることによって、レイヤ 2 VLAN を作成できます。
- **L3 VLAN**。クラスタの異なるノードに属する IP アドレスをバインドすることによって、レイヤ 3 VLAN を作成できます。IP アドレスは同じサブネットに属している必要があります。次の基準のいずれかが満たされていることを確認してください。そうしないと、L3 VLAN バインディングが失敗する可能性があります。
  - すべてのノードは、VLAN にバインドされたサブネットと同じサブネット上に IP アドレスを持つ。
  - クラスタにはストライピングされた IP アドレスがあり、その IP アドレスのサブネットは VLAN にバインドされます。

スポット IP のみを持つクラスターにノードを追加すると、スポット IP アドレスがそのノードに割り当てられる前に同期が行われます。このような場合、L3 VLAN バインディングが失われる可能性があります。この損失を回避するには、ストライプ IP を追加するか、新しく追加されたノードの NSIP に L3 VLAN バインディングを追加します。

Citrix ADC クラスターで **SNMP** を構成するにはどうすればよいですか**?**

SNMP は、スタンドアロンアプライアンスを監視するのと同じ方法で、クラスタおよびクラスタのすべてのノードを監視します。唯一の違いは、クラスタ上の SNMP は、クラスタ IP アドレスを使用して構成する必要があることです。ハードウェア固有のトラップを生成する場合、クラスターのノードを識別するために、ノード ID とノードの NSIP アドレスの 2 つの varbind が含まれます。

クラスタ関連の問題についてテクニカルサポートに連絡する場合、どのような詳細情報を入手する必要がありますか**?**

Citrix ADC アプライアンスには、構成データ、統計情報、およびすべての クラスタノードのログを抽出する **show techsupport-scope** cluster コマンドが用意されています。このコマンドは、クラスターの IP アドレスで実行します。

このコマンドの出力は、**collector_cluster_ <nsip_CCO> _P_ <date-timestamp> .tar.gz* という名前のファイルに保存されます。このファイルは、構成コーディネータの **/var/tmp/support/cluster/* ディレクトリにあります。

このアーカイブをテクニカルサポートチームに送信して、問題をデバッグします。

ストライプ **IP** アドレスをサーバーのデフォルト **Gateway** として使用できますか**?**

クラスター展開では、サーバーのデフォルトゲートウェイがストライプ IP アドレスを指していることを確認します (Citrix ADC が所有する IP アドレスを使用している場合)。たとえば、USIP が有効になっている LB 展開の場合、デフォルトゲートウェイはストライプ SNIP アドレスである必要があります。

クラスター **IP** アドレスから特定のクラスターノードのルーティング構成を表示できますか

はい。VTYSH シェルに入るときに所有者ノードを指定することにより、ノードに固有の構成を表示およびクリアできます。

たとえば、ノード 0 と 1 のコマンドの出力を表示するには、コマンドは次のようになります。

```
\> vtysh
ns# owner-node 0 1
ns(node-0 1)\# show cluster state
ns(node-0 1)\# exit-cluster-node
ns\#
```

LACP システムプライオリティを設定するノードはどのように指定できますか**?**

> 注
>
> NetScaler 10.1 以降でサポートされています。

クラスタでは、**set lacp** コマンドを使用して、そのノードを所有者ノードとして設定する必要があります。

例えば: ID を持つノードの LACP システム優先度を設定するには 2:

```
set lacp -sysPriority 5 -ownerNode 2<!--NeedCopy-->
```


クラスタ設定で **IP** トンネルはどのように設定されますか**?**

> 注
>
> NetScaler 10.1 以降でサポートされています。

クラスタでの IP トンネルの設定は、スタンドアロンアプライアンスでの設定と同じです。唯一の違いは、クラスタのセットアップでは、ローカル IP アドレスがストライピングされた SNIP アドレスである必要があることです。

Citrix ADC クラスタのノードにフェイルオーバーインターフェイスセット（**FIS**）を追加するにはどうすればよいですか**?**

> 注
>
> NetScaler 10.5 以降でサポートされています。

クラスタ IP アドレスで、次のコマンドを使用して、FIS を追加するクラスタノードの ID を指定します。

```
add fis <name> -ownerNode <nodeId>
```

> メモ
>
> - 各クラスタノードの FIS 名は一意である必要があります。
> - クラスタ LA チャネルを FIS に追加できます。クラスタ LA チャネルに、メンバーインターフェイスとしてローカルインターフェイスがあることを確認してください。

FIS の詳細については、「フェールオーバーインターフェイスセットの設定」を参照してください。

クラスタ設定でのネットプロファイルの構成方法

> 注
>
> NetScaler 10.5 以降でサポートされています。

スポットされた IP アドレスをネットプロファイルにバインドできます。このネットプロファイルは、スポットされた負荷分散仮想サーバーまたはサービス（ノードグループを使用して定義されている）にバインドできます。次の推奨事項に従う必要がありますが、これに失敗すると、ネットプロファイル構成が尊重されず、USIP/USNIP 使用される設定:

> 注
>
> - ノードグループの **strict** パラメータが **Yes** に設定されている場合、ネットプロファイルには各ノードグループメンバーからの少なくとも 1 つの IP アドレスが含まれている必要があります。
> - ノードグループの **strict** パラメータが **No** に設定されている場合、ネットプロファイルには各クラスタノードからの少なくとも 1 つの IP アドレスが含まれている必要があります。

クラスタ設定で **WIonNS** を設定するにはどうすればよいですか**?**

> 注
>
> NetScaler 11.0 ビルド 62.x 以降でサポートされています。

クラスターで WIonNS を使用するには、次の操作を行う必要があります。

1. Java パッケージと WI パッケージが、すべてのクラスタノード上の同じディレクトリに存在することを確認します。
2. 永続性が構成された負荷分散仮想サーバーを作成します。
3. WI トラフィックを処理する各クラスタノードの NSIP アドレスとして IP アドレスを持つサービスを作成します。この手順は、Citrix ADC CLI を使用してのみ構成できます。
4. サービスを負荷分散仮想サーバーにバインドします。

> 注
>
> VPN 接続で WIonNS を使用している場合は、負荷分散仮想サーバーが WIHOME に設定されていることを確認します。

クラスタ **LA** チャネルを管理アクセスに使用できますか**?**

いいえ。クラスタノードへの管理アクセスは、クラスタ LA チャネル（CLA/1 など）またはそのメンバインターフェイス上で設定しないでください。これは、ノードが INACTIVE の場合、対応するクラスター LA インターフェースがパワーダウンとしてマークされ、管理アクセスが失われるためです。

クラスタノードはどのように相互に通信し、バックプレーンを通過するさまざまなタイプのトラフィックは何ですか

バックプレーンは、各ノードの 1 つのインターフェイスが共通のスイッチ（クラスタバックプレーンスイッチ）に接続されている一連のインターフェイスです。ノード間通信で使用されるバックプレーンを通過するさまざまなタイプのトラフィックは次のとおりです。

- ノードからノードへのメッセージング（NNM）
- トラフィックステアリング
- 構成の伝播と同期

クラスタの各ノードは、特別な MAC クラスタバックプレーンスイッチアドレスを使用して、バックプレーンを介して他のノードと通信します。クラスター特殊 MAC の形式は次のとおりです：**0x02 0x00 0x6F**< cluster_id> < node_id> < 予約済み **>** 、ここで < cluster_id> はクラスターインスタンス ID です。<node_id> は、クラスターに追加される Citrix ADC アプライアンスのノード番号です。

> 注
>
> バックプレーンによって処理されるトラフィックの量には、ごくわずかな CPU オーバーヘッドがあります。

レイヤー **3** クラスターの **GRE** トンネルを介して何がルーティングされますか

操縦されたデータトラフィックだけが GRE トンネルを通過します。パケットは、GRE トンネルを介して他のサブネット上のノードに操縦されます。

ノード間メッセージング（**NNM**）およびハートビートメッセージの交換方法、およびルーティング方法

NNM、ハートビートメッセージ、およびクラスタプロトコルは、非ステアリングトラフィックです。これらのメッセージはトンネルを介して送信されませんが、直接ルーティングされます。

レイヤ **3** クラスタトンネリングトラフィックに対してジャンボフレームが有効になっている場合、**MTU** の推奨事項は何ですか**?**

GRE トンネルを介したジャンボ MTU のレイヤ 3 クラスタの推奨事項を次に示します。

- ジャンボ MTU は、GRE トンネルのオーバーヘッドに対応するために、L3 パス上のクラスタノード間で設定できます。
- 断片化は、ステアリングする必要のあるフルサイズのパケットでは発生しません。
- ジャンボフレームが許可されていない場合でもトラフィックのステアリングは引き続き機能しますが、フラグメンテーションによるオーバーヘッドが増えます。

グローバルハッシュキーがどのように生成され、すべてのノードで共有されますか**?**

スタンドアロンアプライアンスの `rsskey` は、起動時に生成されます。クラスター設定では、最初のノードがクラスターの `rsskey` を保持します。クラスタに参加するすべての新しいノードは、`rsskey`を同期します。

`set rsskeytype -rsskey symmetric` コマンドの必要性は何ですか ***:*, USIP** オン、**`useproxyport`** オフ、トポロジ

これはクラスターに固有のものではなく、スタンドアロンアプライアンスにも適用されます。USIP がオンで、プロキシポートの使用が無効になっている場合、対称 `rsskey` は、コアからコア（C2C）へのステアリングとノードからノードへのステアリングの両方を削減します。

CCO ノードの変更に寄与する要因は何ですか

クラスターセットアップを形成するために追加された最初のノードは、構成コーディネーター（CCO）ノードになります。次の要因が、クラスター設定の CCO ノードの変更に寄与します。

- 現在の CCO ノードがクラスター設定から削除されたとき
- 現在の CCO ノードがクラッシュしたとき
- 非 CCO ノードの優先順位が変更された場合（優先順位が低いほど優先順位が高くなります）
- のような動的な条件では、ノード間のネットワーク到達可能性

- ノードの状態に変化がある場合（アクティブ、スペア、パッシブ）。CCO としてアクティブノードが優先されます。
- 構成に変更があり、最新の構成のノードが CCO として優先される場合。

Citrix ADC クラスタのトラブルシューティング

October 7, 2021

Citrix ADC クラスターで障害が発生した場合、トラブルシューティングの最初のステップは、クラスターインスタンスに関する情報を取得することです。あなたは実行して情報を取得することができます `show cluster instance clId` と `show cluster node nodeId` それぞれのクラスタノード上でコマンドを。

上記の 2 つの方法を使用して問題を見つけることができない場合は、次のいずれかを使用できます。

- 障害の原因を特定します。クラスタをバイパスしてサーバに接続してみます。試行が成功した場合は、おそらくクラスタのセットアップに問題があります。
- 最近実行したコマンドを確認します。history コマンドを実行して、クラスターで実行された最近の設定を確認します。また、ns.conf ファイルを確認して、実装された構成を確認することもできます。
- **ns.log** ファイルを確認します。で利用可能なログファイルを使用します /var/log/ 各ノードのディレクトリ。実行されたコマンド、コマンドのステータス、および状態変化を識別します。
- **newnslog** ファイルを確認します。で利用可能な `newnslog` ファイルを使用します /var/nslog/ 各ノードのディレクトリ。クラスターノードで発生したイベントを識別します。ファイルを単一のディレクトリにコピーしてから次のコマンドを実行することにより、複数の `newnslog` ファイルを単一のファイルとして表示できます。

```
1  nsconmsg -K newnslog-node<id> -K     newnslog.node<id> -d current
```

それでも問題を解決できない場合は、クラスター上のパケットをトレースするか、show `techsupport -scope cluster` コマンドを使用してみてください。このコマンドを使用して、レポートをテクニカルサポートチームに送信できます。

Citrix ADC クラスターのパケットのトレース

October 7, 2021

Citrix の ADC のオペレーティングシステムは、受信したアプライアンスによって送出されたパケットのダンプを取得するためのユーティリティと呼ばれる *NS* のトレースを提供します。このユーティリティは、パケットをトレース

ファイルに格納します。これらのファイルを使用して、クラスタノードへのパケットフローの問題をデバッグできます。トレースファイルは、Wireshark アプリケーションで表示する必要があります。

nstrace ユーティリティのいくつかの重要な側面は次のとおりです。

- 従来の式とデフォルトの式を使用して、パケットを選択的にトレースするように設定できます。
- 複数の形式でトレースをキャプチャできます：ns トレース形式（.cap）および TCP ダンプ形式（.pcap）。
- 構成コーディネータ上のすべてのクラスターノードのトレースファイルを集約できます。
- 複数のトレースファイルを単一のトレースファイルにマージできます（.cap ファイルのみ）。

ns trace ユーティリティは、Citrix ADC コマンドラインまたは Citrix ADC シェルから使用できます。

スタンドアロンアプライアンスのパケットをトレースするには

アプライアンスで startnstrace コマンドを実行します。このコマンドは、/var/nstrace/ <date-timestamp> ディレクトリにトレースファイルを作成します。トレースファイル名の形式は nstrace です。< id \ >.cap.

show ns trace コマンドを実行すると、ステータスを表示できます。stop ns trace コマンドを実行すると、パケットのトレースを停止できます。

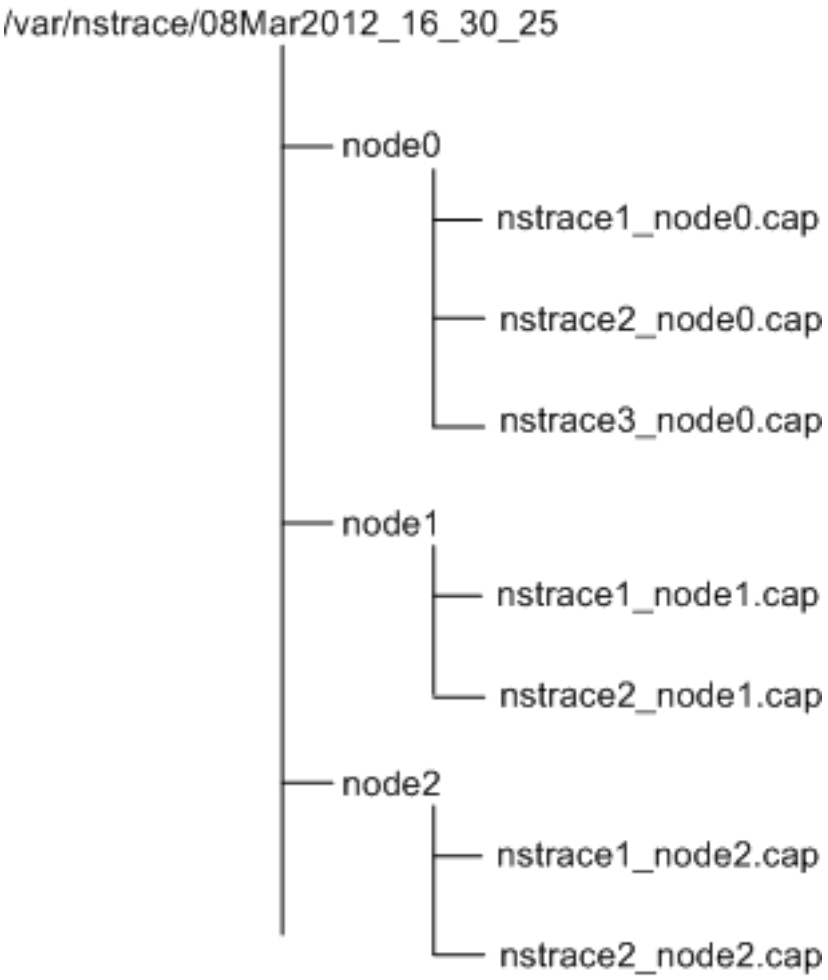
> 注
>
> nstrace.sh ファイルを実行して、Citrix ADC シェルから nstrace ユーティリティを実行することもできます。ただし、Citrix ADC コマンドラインインターフェイスを介して nstrace ユーティリティを使用することをお勧めします。

クラスタのパケットをトレースするには

すべてのクラスターノード上のパケットをトレースし、構成コーディネータ上のすべてのトレースファイルを取得できます。

クラスタ IP アドレスで startnstrace コマンドを実行します。コマンドは伝播され、すべてのクラスターノードで実行されます。トレースファイルは、の個々のクラスターノードに保存されます。/var/nstrace/<date-timestamp> ディレクトリ。トレースファイル名の形式は nstrace です。< id>_node < id \ >.cap.

各ノードのトレースファイルを使用して、ノードの操作をデバッグできます。ただし、すべてのクラスターノードのトレースファイルを 1 つの場所に配置する場合は、クラスターの IP アドレスに対して stop nstrace コマンドを実行する必要があります。すべてのノードのトレースファイルは、のクラスター構成コーディネーターにダウンロードされます。/var/nstrace/<date-timestamp> ディレクトリは次のとおりです。



複数のトレースファイルをマージする

トレースファイルから単一のファイルを準備できます（でのみサポートされます）。キャップファイル）クラスターノードから取得。単一のトレースファイルは、クラスターパケットのトレースの累積的なビューを提供します。単一のトレースファイル内のトレースエントリは、クラスタでパケットが受信された時間に基づいてソートされます。

トレースファイルをマージするには、Citrix ADC シェルで次のように入力します。

```
1 > nstracemerge.sh -srcdir \<DIR\> -dstdir \<DIR\> -filename \<name\> -
    filesize \<num\>
```

各項目の意味は次のとおりです。

- srcdir トレースファイルのマージ元のディレクトリです。このディレクトリ内のすべてのトレースファイルは、単一のファイルにマージされます。
- dstdir マージされたトレースファイルが作成されるディレクトリです。

- Filename 作成されるトレースファイルの名前です。
- Filesize トレースファイルのサイズです。

例

以下は、nstrace ユーティリティを使用してパケットをフィルタリングする例です。

- 3 つのノードのバックプレーンインターフェイス上のパケットをトレースするには、次の手順を実行します。

  クラシックエクスプレッションの使用**:**

```
1   > start nstrace -filter "INTF == 0/1/1 && INTF == 1/1/1 && INTF
      == 2/1/1"
```

  既定のエクスプレッションを使用する**:**

```
1   > start nstrace -filter "CONNECTION.INTF.EQ("0/1/1") &&
      CONNECTION.INTF.EQ("1/1/1") && CONNECTION.INTF.EQ("2/1/1")"
```

- 送信元 IP アドレス 10.102.34.201、または送信元ポートが 80 より大きく、サービス名が"s1" でないシステムからのパケットをトレースするには、次の手順を実行します。

  クラシックエクスプレッションの使用

```
1   > start nstrace -filter "SOURCEIP == 10.102.34.201 || (SVCNAME !=
      s1 && SOURCEPORT > 80)"
```

  既定のエクスプレッションの使用

```
1   > start nstrace -filter "CONNECTION.SRCIP.EQ(10.102.34.201) || (
      CONNECTION.SVCNAME.NE("s1") && CONNECTION.SRCPORT.GT(80))"
```

> 注
>
> ns トレースで使用されるフィルタの詳細については、ns トレースを参照してください。

トレース中の **SSL** セッションキーのキャプチャ

「startnstrace」コマンドを実行すると、新しい capsslkeys パラメーターを設定して、すべての SSL セッションの SSL マスターキーをキャプチャできます。このパラメータを指定すると、nstrace.sslkeys という名前のファイル

がパケットトレースとともに生成されます。このファイルは Wireshark にインポートして、対応するトレースファイル内の SSL トラフィックを復号化できます。

この機能は、後で SSL トラフィックを復号化するために Wireshark にインポートできるセッションキーをエクスポートする Web ブラウザに似ています。

SSL セッションキーを使用する利点

SSL セッションキーを使用する利点は次のとおりです。

1. キャプチャの SSLPLAIN モードによって作成された余分なパケットを含まない小さなトレースファイルを生成します。
2. トレースからプレーンテキスト [SP (1)] を表示し、マスターキーファイルを共有するか、機密データを共有しないことによって保護するかを選択できます。

SSL セッションキーを使用する場合の制限事項

SSL セッションキーを使用する際の制限事項を次に示します。

1. セッションの最初のパケットがキャプチャされていない場合、SSL セッションを復号化することはできません。
2. 連邦情報処理標準 (FIPS) モードが有効になっている場合は、SSL セッションをキャプチャできません。

コマンドラインインターフェイス（**CLI**）を使用して **SSL** セッションキーをキャプチャするには

コマンドプロンプトで次のコマンドを入力して、トレースファイルの SSL セッションキーを有効または無効にし、トレース操作を確認します。

```
1  > start nstrace -capsslkeys ENABLED
2  > show nstrace
3  Example
4  > start nstrace -capsslkeys ENABLED
5  > show nstrace
6        State:  RUNNING          Scope:  LOCAL            TraceLocation:
             ”/var/nstrace/04May2016_17_51_54/...”
7        Nf:  24                  Time:  3600              Size:  164
                          Mode:  TXB NEW_RX
8        Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  04
            May2016_17_51_54 Link:  DISABLED
9        Merge:  ONSTOP           Doruntimecleanup:  ENABLED TraceBuffers:
              5000      SkipRPC:  DISABLED
10       SkipLocalSSH:  DISABLED  Capsslkeys:  ENABLED     InMemoryTrace:
             DISABLED
```



```
11    Done
```

Citrix ADC GUI を使用して **SSL** セッションキーを構成するには

1. アプライアンスの暗号化パケットのトレースを開始するには、[設定] **>** [システム] **>** [診断] **>** [テクニカルサポートツール] に移動し、[新しいトレースの開始] をクリックします。
2. [トレースの開始] ページで、[**SSL** マスターキーのキャプチャ] チェックボックスをオンにします。
3. [**OK**] をクリックし、[完了] をクリックします。

SSL マスターキーを **Wireshark** にインポートするには

Wireshark GUI で、[編集] **>** [環境設定] **>** [プロトコル] **>** [**SSL**] **>** [**(Pre)-Master-Secret** ログファイル名] に移動し、アプライアンスから取得したマスターキーファイルを指定します。

よくある問題のトラブルシューティング

April 25, 2022

クラスタにノードを結合しているときに、次のメッセージが表示されます。**"ERROR: Invalid interface name/number."** このエラーを解決するにはどうすればいいですか**?**

上記のエラーは、add clusternode コマンドを使用してノードを追加しているときに無効または誤ったバックプレーンインターフェイスを指定した場合に発生します。このエラーを解決するには、ノードの追加時に指定したインターフェイスを確認します。アプライアンスの管理インターフェイスをバックプレーンインターフェイスとして指定していないこと、およびインターフェイスの <nodeId> ビットがノードの ID と同じであることを確認します。たとえば、nodeID が 3 の場合、バックプレーンインターフェイスは 3/<c>/<u> でなければなりません。

ノードをクラスタに参加しているときに、次のメッセージが表示されます。**"ERROR: Clustering cannot be enabled, because the local node is not a member of the cluster."** このエラーを解決するにはどうすればいいですか**?**

このエラーは、ノードの NSIP をクラスタに追加せずにノードに参加しようとしたときに発生します。このエラーを解決するには、最初に **add cluster node** コマンドを使用してノードの NSIP アドレスをクラスターに追加してから、joincluster コマンドを実行する必要 があります。

ノードをクラスタに参加しているときに、「エラー：接続が拒否されました。」このエラーを解決するにはどうすればいいですか**?**

このエラーは、次の理由で発生する可能性があります。

- 接続の問題。ノードはクラスタ IP アドレスに接続できません。参加しようとしているノードからクラスター IP アドレスに ping を実行してみます。
- クラスタ **IP** アドレスが重複しています。クラスタ IP アドレスが一部の非クラスタノードに存在しているかどうかを確認します。含まれている場合は、クラスター IP アドレスを作成し、クラスターに再参加してみてください。

クラスタにノードを結合しているときに、次のメッセージが表示されます。**"ERROR: License mismatch between the configuration coordinator and the local node."** このエラーを解決するにはどうすればいいですか**?**

クラスタに参加するアプライアンスには、構成コーディネータと同じライセンスが必要です。このエラーは、参加するノードのライセンスが、構成コーディネータのライセンスと一致しない場合に発生します。このエラーを解決するには、両方のノードで次のコマンドを実行し、出力を比較します。

コマンドラインから：

- `show ns hardware`
- `show ns license`

シェルから：

- `nsconmsg -g feature -d stats`
- `ls /nsconfig/license`
- /var/log/license.log ファイルの内容を表示する

クラスターノードの構成がクラスター構成と同期していない場合はどうすればよいですか

通常、構成はすべてのクラスターノード間で自動的に同期されます。ただし、特定のノードで構成が同期されていないと思われる場合は、同期するノードから force clustersync コマンドを実行して同期を強制する必要があります。詳細については、「クラスター構成の同期」を参照してください。

クラスタノードを構成するときに、「エラー: セッションは読み取り専用です。クラスタの IP アドレスに接続して構成を変更します」

クラスター上のすべての構成は、クラスター IP アドレスを介して実行する必要があり、構成は他のクラスターノードに伝達されます。個々のノードの NSIP アドレスを通じて確立されたすべてのセッションは読み取り専用です。

ノードの健全性が「**UP**」と表示されたときに、ノードの状態が「**INACTIVE**」と表示されるのはなぜですか**?**

正常なノードは、さまざまな理由で INACTIVE 状態になる可能性があります。ns.log またはエラーカウンターをスキャンすると、正確な理由を特定するのに役立ちます。

ノードの健全性が「**NOT UP**」と表示されている場合、ノードの健全性をどのように解決できますか**?**

ノードの健全性 "**Not UP**" は、ノードにいくつかの問題があることを示します。根本的な原因を確認するには、**show cluster node** コマンドを実行する必要があります。このコマンドは、ノードのプロパティとノード障害の理由を表示します。

ノードの健全性が「**NOT UP**」と表示され、その理由がノード上で構成コマンドが失敗したことを示している場合はどうすればよいですか**?**

この問題は、一部のコマンドがクラスターノードで実行されていない場合に発生します。このような場合、次のオプションのいずれかを使用して、設定が同期されていることを確認する必要があります。

- 一部のクラスターノードがこの状態にある場合は、それらのノードで強制クラスター同期操作を実行する必要があります。詳細については、「クラスター構成の同期」を参照してください。
- すべてのクラスターノードがこの状態にある場合は、すべてのクラスターノードでクラスターインスタンスを無効にしてから有効にする必要があります。

set virtual server コマンドを実行すると、「そのようなリソースはありません」というメッセージが表示されます。このエラーを解決するにはどうすればいいですか**?**

set vserver コマンドは、クラスタリングではサポートされていません。vserver の設定解除、**vserver** の有効化、**vserver** の無効化、および **rm vserver** のコマンドもサポートされません。ただし、**show vserver** コマンドはサポートされています。

Telnet セッションでクラスタを構成できません。どうしたらいいですか**?**

Telnet セッションでは、クラスタ IP アドレスには読み取り専用モードでのみアクセスできます。したがって、Telnet セッションを介してクラスタを構成することはできません。

クラスタノード間で大きな時間差があることに気付きました。このエラーを解決するにはどうすればいいですか**?**

バックプレーンスイッチが原因で PTP パケットがドロップされた場合、または仮想環境で物理リソースがオーバーコミットされた場合、時刻は同期されません。

時刻を同期するには、クラスタ IP アドレスで次の操作を行う必要があります。

1. PTP を無効にします。

   set ptp -state disable

2. クラスタのネットワークタイムプロトコル (NTP) を構成します。詳細については、クロック同期の設定を参照してください。

クラスタ **IP** アドレスとクラスタノードの **NSIP** アドレスに接続できない場合、どうすればよいですか**?**

クラスタ IP アドレスまたはクラスタノードの NSIP にアクセスできない場合は、シリアルコンソールからアプライアンスにアクセスする必要があります。NSIP アドレスに到達できる場合は、シェルプロンプトで次のコマンドを実行することにより、シェルからクラスター IP アドレスに SSH で接続できます。

“‘# ssh nsroot@

```
## 接続に問題があるクラスタノードをリカバリするには、どうすればよいですか?

接続に問題があるノードをリカバリするには、次の手順で行います。

1. そのノードでクラスターインスタンスを無効にします（クラスターノードのNSIPからコマンドを実行できないため）。

1. ノードの回復に必要なコマンドを実行します。

1. そのノードでクラスタインスタンスを有効にします。

## クラスタの一部のノードには、2 つのデフォルトルートがあります。クラスタノードから2番目のデフォルトルートを削除するにはどうすればよいですか?

追加のデフォルトルートを削除するには、追加のルートがある各ノードで次の操作を行います。

1. クラスタインスタンスを無効にします。

   ```disable cluster instance <clId><!--NeedCopy-->
```

1. ルートを削除します。

   ```
   rm route <network> <netmask> <gateway><!--NeedCopy-->
   ```

2. クラスタインスタンスを有効にします。

   ```
   enable cluster instance <clId><!--NeedCopy-->
   ```


   ```

既存のクラスターノードがオンラインになると、クラスター機能が影響を受けます。このエラーを解決するにはどうすればいいですか❓

ノードがクラスター外にあるときにノードの RPC パスワードがクラスター IP アドレスから変更された場合、ノードがオンラインになると、RPC 資格情報に不一致が生じ、クラスターの機能に影響を与える可能性があります。この問題を解決するには、set ns rpcNode コマンドを使用して、オンラインになったノードの NSIP のパスワードを更新します。

## コンテンツスイッチ

January 31, 2022

今日の複雑な Web サイトでは、さまざまなコンテンツをさまざまなユーザーに提示したい場合があります。たとえば、顧客またはパートナーの IP 範囲のユーザーが特別な Web ポータルにアクセスできるようにしたい場合があります。特定の地域に関連するコンテンツを、その地域のユーザーに提示したい場合があります。さまざまな言語のコンテンツをそれらの言語の話者に提示することをお勧めします。スマートフォンなどの特定のデバイスに合わせたコンテンツを、デバイスを使用するユーザーに提示することをお勧めします。Citrix ADC コンテンツスイッチング機能を使用すると、アプライアンスは、ユーザーに提示する特定のコンテンツに基づいて、クライアント要求を複数のサーバーに分散できます。

コンテンツスイッチングを構成するには、まず基本的なコンテンツスイッチング設定を作成し、ニーズに合わせてカスタマイズします。これには、コンテンツスイッチング機能の有効化、スイッチ対象のコンテンツの各バージョンをホストするサーバーの負荷分散の設定、コンテンツスイッチング仮想サーバーの作成、負荷分散仮想サーバーへの要求を選択するポリシーの作成、およびコンテンツスイッチ仮想サーバーへのポリシーのバインド。次に、ポリシーの優先順位を設定し、バックアップ仮想サーバーを構成してセットアップを保護し、要求をキャッシュにリダイレクトしてセットアップのパフォーマンスを向上させることで、ニーズに合わせてセットアップをカスタマイズできます。

### コンテンツスイッチングの仕組み

コンテンツスイッチングにより、Citrix ADC アプライアンスは、同じ Web ホストに送信された要求を、異なるコンテンツの異なるサーバーに送信できます。たとえば、動的コンテンツ（拡張子が.asp、.dll、または.exe の URL など）に対する要求をあるサーバーに送信し、静的コンテンツの要求を別のサーバーに送信するようにアプライアンスを構成できます。TCP/IP ヘッダーとペイロードに基づいてコンテンツスイッチングを実行するようにアプライアンスを設定できます。

コンテンツスイッチングを使用して、さまざまなクライアント属性に基づいて、さまざまなコンテンツを持つさまざまなサーバーにリクエストをリダイレクトするようにアプライアンスを構成することもできます。これらのクライアント属性のいくつかは、次のとおりです。

- デバイスタイプ。アプライアンスは、クライアント要求内のユーザーエージェントまたはカスタム HTTP ヘッダーを調べて、要求の発信元のデバイスのタイプを調べます。デバイスタイプに基づいて、特定の Web サー

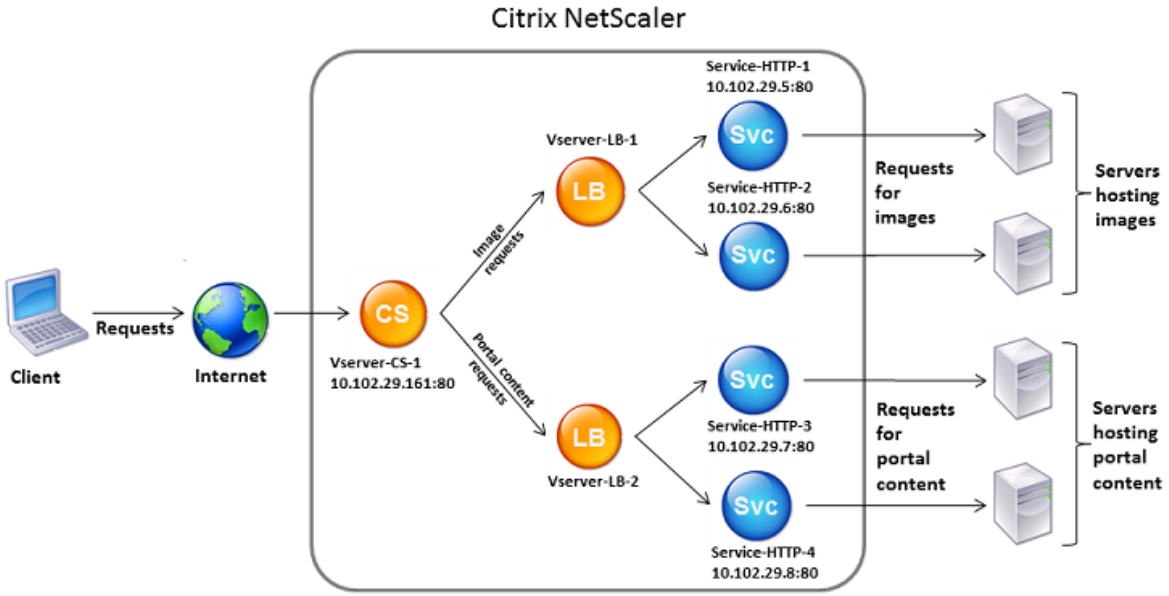
バーに要求を転送します。たとえば、要求が携帯電話からのものである場合、要求はサーバーに送信され、ユーザーが携帯電話で表示できるコンテンツを提供できます。コンピューターからの要求は、コンピューター画面用に設計されたコンテンツを提供できる別のサーバーに送信されます。

- 言語。アプライアンスは、クライアント要求の Accept-Language HTTP ヘッダーを調べ、クライアントのブラウザで使用される言語を決定します。次に、アプライアンスは、その言語でコンテンツを提供するサーバーに要求を送信します。たとえば、言語に基づいたコンテンツスイッチングを使用すると、アプライアンスは、フランス語でコンテンツをリクエストするようにブラウザが設定されているユーザーを、フランス語版の新聞が搭載されたサーバーに送信できます。これは、英語版でコンテンツを要求するようにブラウザが設定されている他の人を、英語版のサーバーに送ることができます。
- クッキー。アプライアンスは、サーバーが以前に設定した Cookie の HTTP 要求ヘッダーを調べます。クッキーを検出すると、カスタムコンテンツをホストする適切なサーバーに要求が送信されます。たとえば、クライアントが顧客ロイヤルティプログラムのメンバーであることを示す Cookie が見つかった場合、要求はより高速なサーバーまたは特別なコンテンツを持つサーバーに送信されます。クッキーが見つからない場合、またはクッキーがユーザーがメンバーではないことを示している場合、要求は一般のサーバーに送信されます。
- **HTTP** メソッド。アプライアンスは、使用されるメソッドの HTTP ヘッダーを調べ、クライアント要求を適切なサーバーに送信します。たとえば、イメージの GET 要求はイメージサーバーに送信でき、POST 要求は動的コンテンツを処理するより高速なサーバーに送信できます。
- レイヤ **3/4** データ。アプライアンスは、送信元または宛先 IP、送信元または宛先ポート、または TCP または UDP ヘッダーに存在するその他の情報に対する要求を調べ、クライアント要求を適切なサーバーに送信します。たとえば、顧客に属するソース IP からのリクエストは、より高速なサーバー上のカスタム Web ポータルや、特別なコンテンツを持つポータル Web ポータルに転送できます。

一般的なコンテンツスイッチ展開は、次の図に示すエンティティで構成されます。

図 1: コンテンツスイッチングアーキテクチャ




コンテンツスイッチの構成には、コンテンツスイッチ仮想サーバー、負荷分散仮想サーバーとサービスから成る負荷分散セットアップ、およびコンテンツスイッチポリシーが含まれます。コンテンツスイッチングを構成するには、コンテンツスイッチング仮想サーバーを構成し、ポリシーおよび負荷分散仮想サーバーに関連付ける必要があります。このプロセスでは、*content group* を作成します。*content group* は、特定のコンテンツスイッチング設定に関連するすべての仮想サーバとポリシーのグループです。

コンテンツスイッチングは、HTTP、HTTPS、TCP、および UDP 接続で使用できます。HTTPS の場合は、SSL オフロードを有効にする必要があります。

要求がコンテンツスイッチ仮想サーバーに到達すると、仮想サーバーはその要求に対して関連するコンテンツスイッチポリシーを適用します。ポリシーの優先順位により、コンテンツスイッチ仮想サーバーにバインドされたポリシーを評価する順序を定義します。デフォルトの構文ポリシーを使用している場合は、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに、そのポリシーに優先度を割り当てる必要があります。Citrix ADC クラシックポリシーを使用している場合は、ポリシーに優先度を割り当てることができますが、優先度を割り当てる必要はありません。優先順位を割り当てると、設定した順序でポリシーが評価されます。そうしないと、Citrix ADC アプライアンスによってポリシーが作成された順序で評価されます。

ポリシープライオリティの設定に加えて、Goto 式とポリシーバンクの呼び出しを使用して、ポリシー評価の順序を操作できます。デフォルトの構文ポリシー設定の詳細については、「デフォルトの構文ポリシーの設定」を参照してください。

ポリシーが評価されると、コンテンツスイッチ仮想サーバーは要求を適切な負荷分散仮想サーバーにルーティングし、適切なサービスに送信します。

コンテンツスイッチング仮想サーバーは、他の仮想サーバーにのみ要求を送信できます。外部ロードバランサーを使用している場合は、その負荷分散仮想サーバーを作成し、その仮想サーバーをサービスとしてコンテンツスイッチ仮

想サーバーにバインドする必要があります。

## 基本的なコンテンツスイッチングの構成

January 31, 2022

コンテンツスイッチングを構成する前に、コンテンツスイッチングの設定方法と、サービスと仮想サーバーの接続方法を理解する必要があります。

基本的な機能的なコンテンツスイッチングの設定を構成するには、まずコンテンツスイッチング機能を有効にします。次に、少なくとも 1 つのコンテンツグループを作成します。コンテンツグループごとに、コンテンツスイッチングを使用する Web サイトのグループへの要求を受け入れるコンテンツスイッチング仮想サーバーを作成します。また、コンテンツスイッチング仮想サーバーが要求を送信する負荷分散仮想サーバーのグループを含む負荷分散セットアップを作成します。どの要求をどの負荷分散仮想サーバーに転送するかを指定するには、リダイレクトする要求の種類ごとに 1 つずつ、少なくとも 2 つのコンテンツスイッチングポリシーを作成します。仮想サーバーとポリシーを作成したら、ポリシーをコンテンツスイッチ仮想サーバーにバインドします。ポリシーを複数のコンテンツスイッチ仮想サーバーにバインドすることもできます。ポリシーをバインドするときは、ポリシーに一致する要求が転送される負荷分散仮想サーバーを指定します。

個々のポリシーをコンテンツスイッチ仮想サーバーにバインドするだけでなく、ポリシーラベルをバインドすることもできます。より多くのコンテンツグループを作成する場合は、ポリシーまたはポリシーラベルを複数のコンテンツスイッチング仮想サーバーにバインドできます。

> 注
>
> コンテンツグループを作成したら、そのコンテンツスイッチ仮想サーバーを変更して、構成をカスタマイズできます。

### コンテンツスイッチングの有効化

コンテンツスイッチング機能を使用するには、コンテンツスイッチングを有効にする必要があります。コンテンツスイッチング機能が無効になっていても、コンテンツスイッチングエンティティを構成できます。ただし、エンティティは機能しません。

#### **CLI** を使用してコンテンツの切り替えを有効にするには

コマンドプロンプトで次のコマンドを入力して、コンテンツの切り替えを有効にし、構成を確認します。

```
enable ns feature CS

show ns feature
```


```
<!--NeedCopy-->
```

例:

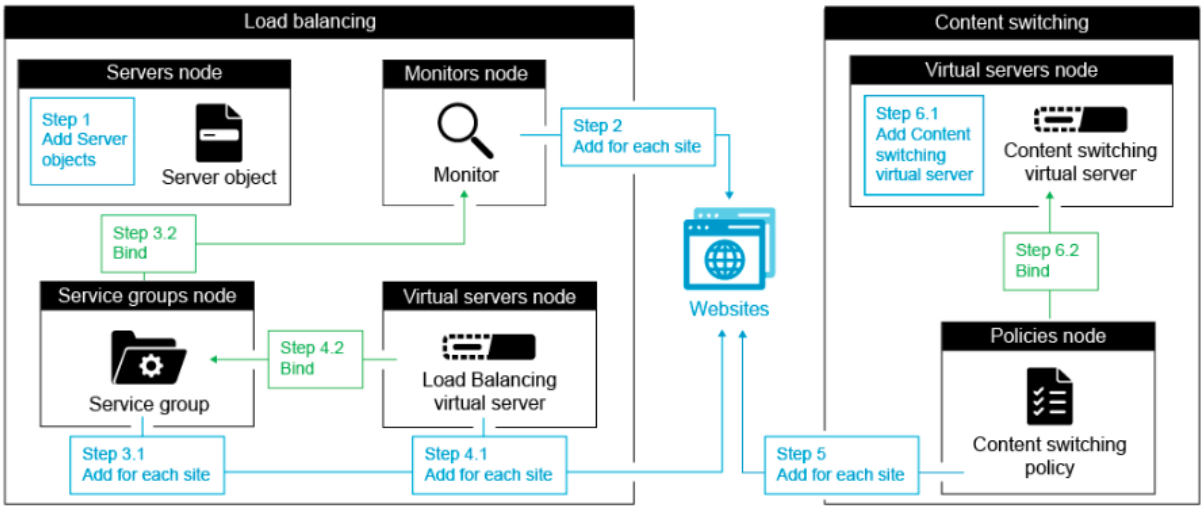
```
> enable  feature ContentSwitch
    Done
> show feature

    Feature                         Acronym              Status
    -------                         -------              ------
1)      Web Logging                     WL                   OFF
2)      Surge Protection                SP                   ON
3)      Load Balancing                  LB                   ON
4)      Content Switching               CS                   ON
.
.
.
22)     Responder                       RESPONDER            ON
23)     HTML Injection                  HTMLInjection        ON
24)     NetScaler Push                  push                 OFF
Done
<!--NeedCopy-->
```

GUI を使用してコンテンツの切り替えを有効にするには

[システム] > [設定] に移動し、[モードと機能] グループで [基本機能の構成] を選択し、[コンテンツの切り替え] を選択します。

次の図は、コンテンツスイッチングの段階的な構成を示しています。



コンテンツスイッチング仮想サーバーの作成

コンテンツスイッチ仮想サーバーを追加、変更、および削除できます。仮想サーバーを作成すると、仮想サーバーの状態は DOWN になります。これは、負荷分散仮想サーバーがまだバインドされていないためです。

CLI を使用して仮想サーバーを作成するには

コマンドプロンプトで入力します。

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

例**:**

```
add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチ仮想サーバーを追加するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを追加します。
2. コンテンツスイッチング仮想サーバーの名前を指定します。

   > 注
   >
   > プロトコルごとに異なるコンテンツスイッチング仮想サーバーがあります。(HTTP や SSL など)。

3. 関連するフィールドに入力し、[**OK**] をクリックします。

コンテンツスイッチング仮想サーバの統計

コンテンツスイッチング仮想サーバーの統計には、仮想サーバーの選択、要求バイト、応答バイト、受信したパケットの合計、送信したパケットの合計、スピルオーバーしきい値、スピルオーバーの選択、現在のクライアントが確立した接続、仮想サーバーのダウンバックアップの選択などの情報が表示されます。

コンテンツスイッチ仮想サーバーの統計情報には、バインドされたデフォルトの負荷分散仮想サーバーの概要の詳細も表示されます。

CLI を使用してコンテンツスイッチ仮想サーバーの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat cs vserver <name>
<!--NeedCopy-->
```

例:

```
stat cs vserver CS_stats
<!--NeedCopy-->
```

```
Vserver Summary
                          IP  port     Protocol        State
CS_stats             1.1.1.1    80         HTTP           UP

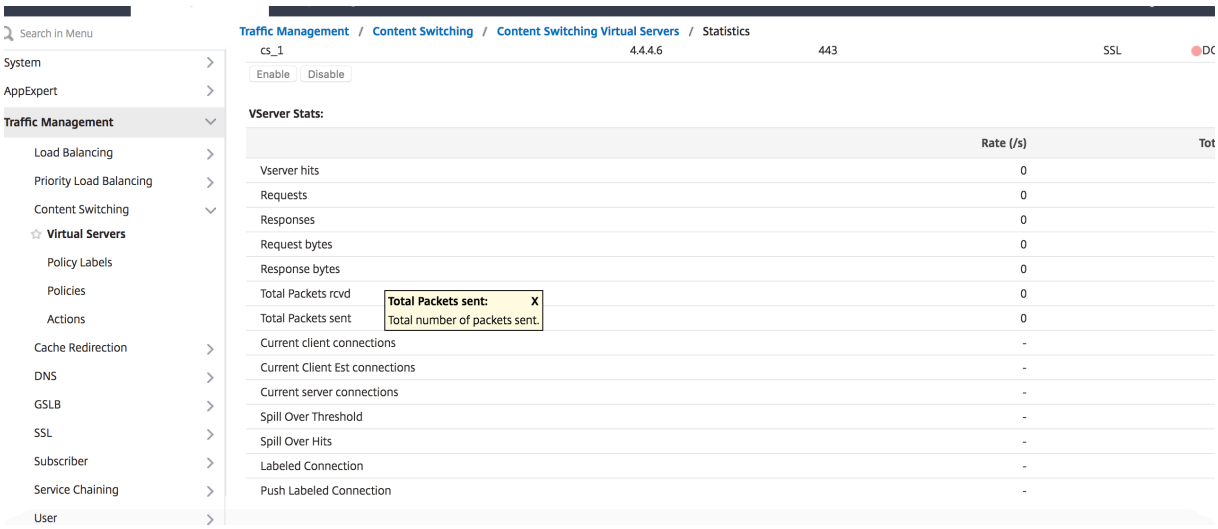
VServer Stats:
                                        Rate (/s)                Total
Vserver hits                                    0                    0
Requests                                        0                    0
Responses                                       0                    0
Request bytes                                   0                    0
Response bytes                                  0                    0
Total Packets rcvd                              0                    0
Total Packets sent                              0                    0
Current client connections                     --                    0
Current Client Est connections                 --                    0
Current server connections                     --                    0
Spill Over Threshold                           --                    0
Spill Over Hits                                --                    0
Labeled Connection                             --                    0
Push Labeled Connection                        --                    0
Deferred Request                                0                    0
Invalid Request/Response                       --                    0
Invalid Request/Response Dropped               --                    0
Vserver Down Backup Hits                       --                    0
Current Multipath TCP sessions                 --                    0
Current Multipath TCP subflows                 --                    0
Apdex for client response times.               --                 1.00
Average client TTLB                            --                    0

 Done
```

GUI を使用してコンテンツスイッチ仮想サーバーの統計情報を表示するには

1.**[**トラフィック管理**]** > **[**コンテンツスイッチング**]** > **[**仮想サーバー**]** に移動します。

2. 仮想サーバを選択し、［統計］をクリックします。



コンテンツスイッチング用の負荷分散設定の構成

コンテンツスイッチング仮想サーバーは、すべての要求を負荷分散仮想サーバーにリダイレクトします。切り替えるコンテンツのバージョンごとに、負荷分散仮想サーバーを 1 つ作成する必要があります。これは、セットアップのコンテンツのバージョンごとにサーバーが 1 つしかないため、それらのサーバーとの負荷分散を行っていない場合でも当てはまります。また、コンテンツの各バージョンをミラーリングする複数の負荷分散サーバーを使用して、実際の負荷分散を構成することもできます。どちらのシナリオでも、コンテンツスイッチング仮想サーバーには、切り替え対象のコンテンツの各バージョンに割り当てられた特定の負荷分散仮想サーバーが必要です。

次に、負荷分散仮想サーバは要求をサービスに転送します。バインドされているサービスが 1 つしかない場合は、そのサービスを選択します。複数のサービスがバインドされている場合は、設定された負荷分散方式を使用して要求のサービスを選択し、その要求を選択したサービスに転送します。

基本的な負荷分散設定を構成するには、次のタスクを実行する必要があります。

- 負荷分散仮想サーバーの作成
- サービスの作成
- 負荷分散仮想サーバへのサービスのバインド

ロードバランシングの詳細については、「ロードバランシングの仕組み」を参照してください。基本的なロードバランシング構成の設定の詳細については、「基本的なロードバランシングの設定」を参照してください。

コンテンツスイッチングアクションの設定

コンテンツを切り替える仮想サーバーにポリシーをバインドするときに、コンテンツスイッチングポリシーのターゲット負荷分散仮想サーバーを指定します。したがって、トラフィックの転送先となる負荷分散仮想サーバごとに 1 つのポリシーを設定する必要があります。

ただし、コンテンツスイッチングポリシーでデフォルトの構文規則が使用されている場合は、ポリシーのアクションを設定できます。アクションでは、ターゲットの負荷分散仮想サーバーの名前を指定するか、実行時に要求の送信先

となる負荷分散仮想サーバーの名前を計算する要求ベースの式を構成できます。アクション式は、デフォルトの構文で指定する必要があります。

expression オプションを使用すると、コンテンツスイッチング仮想サーバーごとに 1 つのポリシーしか必要ないため、コンテンツスイッチング構成のサイズを大幅に削減できます。アクションを使用するコンテンツスイッチングポリシーは、複数のコンテンツスイッチ仮想サーバーにバインドすることもできます。これは、ターゲットの負荷分散仮想サーバーがコンテンツスイッチングポリシーで指定されなくなったためです。単一のポリシーを複数のコンテンツスイッチング仮想サーバーにバインドする機能によって、コンテンツスイッチング構成のサイズをさらに縮小できます。

アクションを作成したら、コンテンツスイッチポリシーを作成し、ポリシーでアクションを指定します。これにより、ポリシーがリクエストに一致したときにアクションが実行されます。

> 注
>
> また、既定の構文規則を使用するコンテンツスイッチングポリシーの場合、別のアクションを使用する代わりに、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに、ターゲットの負荷分散仮想サーバーを指定することもできます。従来の式を使用するドメインベースのポリシー、URL ベースのポリシー、および規則ベースのポリシーの場合、アクションは使用できません。したがって、これらのタイプのポリシーでは、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに、ターゲットの負荷分散仮想サーバーの名前を指定します。

ターゲットの負荷分散仮想サーバーの名前を指定するアクションの構成

コンテンツスイッチング操作でターゲット負荷分散仮想サーバーの名前を指定する場合は、ターゲット負荷分散仮想サーバーと同じ数のコンテンツスイッチングポリシーが必要です。この場合、コンテンツスイッチングの決定は、コンテンツスイッチングポリシーのルールに基づいており、アクションはターゲット負荷分散仮想サーバーを指定するだけです。要求がポリシーに一致すると、要求は指定された負荷分散仮想サーバーに転送されます。

CLI を使用して、ターゲットの負荷分散仮想サーバーの名前を指定するコンテンツスイッチングアクションを作成および検証するには

コマンドプロンプトで入力します。

```
1 add cs action <name> -targetLBVserver <string> [-comment <string>]
2
3 show cs action <name>
4 <!--NeedCopy-->
```

例**:**


```
> add cs action mycsaction -targetLBVserver mylbvserver -comment ”
   Forwards requests to mylbvserver.”
 Done
> show cs action mycsaction
    Name: mycsaction
    Target LB Vserver: mylbvserver
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0
    Comment: ”Forwards requests to mylbvserver.”

 Done
>
<!--NeedCopy-->
```

GUI を使用して、ターゲットの負荷分散仮想サーバーの名前を指定するコンテンツスイッチングアクションを構成するには

1. [トラフィック管理] **>** [コンテンツの切り替え] **>** [アクション] に移動します。
2. コンテンツスイッチングアクションを構成し、ターゲットの負荷分散仮想サーバーの名前を指定します。

実行時にターゲットを選択するための式を指定するアクションの構成

ターゲットの負荷分散仮想サーバーの名前を動的に計算できる要求ベースの式を構成する場合は、適切な仮想サーバーを選択するために 1 つのコンテンツスイッチングポリシーを構成するだけで済みます。ポリシーの規則は、単純な TRUE（ポリシーはすべてのリクエストに一致する）にすることができます。この場合、コンテンツスイッチングの決定はアクションの式に基づいているためです。アクションで式を設定することで、コンテンツスイッチングの設定のサイズを大幅に減らすことができます。

実行時にターゲットの負荷分散仮想サーバーの名前を計算するための要求ベースの式を構成する場合は、構成内の負荷分散仮想サーバーに名前を付ける方法を慎重に検討する必要があります。アクションで要求ベースのポリシー式を使用して、名前を取得できる必要があります。

たとえば、URL サフィックス（要求されたリソースの拡張）に基づいて要求を切り替える場合、負荷分散仮想サーバーに名前を付けるときに、`mylb_`などの所定の文字列に URL サフィックスを追加する規則に従うことができます。たとえば、HTML ページと PDF ファイルの負荷分散仮想サーバーには、それぞれ`mylb_html`と`mylb_pdf`という名前を付けることができます。その場合、適切な負荷分散仮想サーバーを選択するために、コンテンツスイッチングアクションで使用できるルールは`”mylb_”`+`HTTP.REQ.URL.SUFFIX`です。コンテンツスイッチング仮想サーバーが HTML ページの要求を受信すると、式は `mylb_html`を返し、要求は仮想サーバー `mylb_html`に切り替えられます。

CLI を使用して、式を指定するコンテンツスイッチングアクションを作成するには

コマンドラインで次のコマンドを入力して、式を指定し、構成を確認するコンテンツスイッチングアクションを作成します。

```
add cs action <name> -targetVserverExpr <expression>) [-comment <string
    >]

show cs action <name>
<!--NeedCopy-->
```

例**:**

```
> add cs action mycsaction1 -targetvserverExpr '"mylb_" + HTTP.REQ.URL.
    SUFFIX'
 Done
> show cs action mycsaction1
    Name: mycsaction1
    Target Vserver Expression: "mylb_" + HTTP.REQ.URL.SUFFIX
    Target LB Vserver: No_Target
                ...
 Done
>
<!--NeedCopy-->
```

GUI を使用して式を指定するコンテンツスイッチングアクションを構成するには

1. [トラフィック管理] **>** [コンテンツの切り替え] **>** [アクション] に移動します。
2. コンテンツスイッチングアクションを構成し、ターゲットの負荷分散仮想サーバーの名前を動的に計算する式を指定します。

コンテンツスイッチングポリシーの構成

コンテンツスイッチングポリシーは、負荷分散仮想サーバーに送信される要求のタイプを定義します。これらのポリシーは、割り当てられた優先順位の順序、または（Citrix ADC クラシックポリシーを使用し、バインド時に優先順位を割り当てない場合）、ポリシーが作成された順序で適用されます。

ポリシーには、次のものがあります。

- ドメインベースのポリシー。Citrix ADC アプライアンスは、受信 URL のドメインとポリシーで指定されたドメインを比較します。その後、アプライアンスは最適なコンテンツを返します。ドメインベースのポリシーは、

従来のポリシーである必要があります。このタイプのコンテンツスイッチングポリシーでは、デフォルトの構文ポリシーはサポートされていません。

- **URL** ベースのポリシー。アプライアンスは、着信 URL とポリシーで指定された URL を比較します。次に、アプライアンスは、最も適切な URL ベースのコンテンツを返します。これは、通常、最も長く一致する構成された URL です。URL ベースのポリシーは、クラシックポリシーである必要があります。このタイプのコンテンツスイッチングポリシーでは、デフォルトの構文ポリシーはサポートされていません。
- ルールベースのポリシー。アプライアンスは、受信データをポリシーで指定された式と比較します。ルールベースのポリシーは、クラシック式またはデフォルトの構文式を使用して作成します。ルールベースのコンテンツスイッチングポリシーでは、従来の構文ポリシーとデフォルトの構文ポリシーの両方がサポートされています。

  > 注
  >
  > 規則ベースのポリシーは、オプションのアクションで設定できます。アクションを持つポリシーは、複数の仮想サーバまたはポリシーラベルにバインドできます。

  ポリシーをコンテンツスイッチ仮想サーバーにバインドするときに優先度を設定すると、ポリシーが優先度順に評価されます。ポリシーをバインドするときに特定の優先順位を設定しない場合、ポリシーは作成された順序で評価されます。

Citrix ADC クラシックポリシーと式の詳細については、「クラシックポリシーと式の構成」を参照してください。デフォルトの構文ポリシーの詳細については、デフォルトの構文式の構成を参照してください。

CLI を使用してコンテンツスイッチングポリシーを作成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add cs policy <policyName> -domain <domain>

add cs policy <policyName> -url <URLValue>

add cs policy <policyName> -rule <RULEValue>

add cs policy <policyName> -rule <RULEValue> -action <actionName>
<!--NeedCopy-->
```

例:

```
add cs policy Policy-CS-1 -url ”http://example.com”

add cs policy Policy-CS-1 -domain ”example.com”
```



```
4
5 add cs policy Policy-CS-1 -rule ”CLIENT.IP.SRC.SUBNET(24).EQ
     (10.217.84.0)”
6
7 add cs policy Policy-CS-2 -rule ”SYS.TIME.BETWEEN(GMT 2009 Nov,GMT 2009
     Dec)”
8
9 add cs policy Policy-CS-3 -rule ”http.req.method.eq(GET)” -action act1
10 <!--NeedCopy-->
```

CLI を使用してコンテンツスイッチングポリシーの名前を変更するには

コマンドプロンプトで入力します。

```
1 rename cs policy <policyName> <newName>
2 <!--NeedCopy-->
```

例:

```
1 rename cs policy myCSPolicy myCSPolicy1
2 <!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーの名前を変更するには

[トラフィック管理] > [コンテンツの切り替え] > [ポリシー] に移動し、ポリシーを選択し、[アクション] リストで [名前の変更] を選択します。

GUI を使用してコンテンツスイッチングポリシーを作成するには

1. [トラフィック管理] > [コンテンツの切り替え] > [ポリシー] に移動し、[追加] をクリックします。
2. 関連フィールドに入力し、[**Create**] をクリックします。

コンテンツスイッチングポリシーラベルの設定

ポリシーラベルは、ポリシーがバインドされるユーザー定義のバインドポイントです。ポリシーラベルが呼び出されると、そのラベルにバインドされているすべてのポリシーが、割り当てた優先順位に従って評価されます。ポリシー・ラベルには、1 つ以上のポリシーを含めることができ、それぞれに独自の結果を割り当てることができます。ポリシー・ラベル内の 1 つのポリシーが一致すると、次のポリシーに進み、別のポリシー・ラベルまたは適切なリソースを

呼び出したり、ポリシー評価の即時終了を行ったり、ポリシー・ラベルを呼び出したポリシーに制御が戻ったりすることがあります。ポリシーラベルは、デフォルトの構文ポリシーに対してのみ作成できます。

コンテンツスイッチポリシーラベルは、名前、ラベルタイプ、およびポリシーラベルにバインドされたポリシーのリストで構成されます。ポリシーラベルタイプは、ラベルにバインドされたポリシーに割り当てられたプロトコルを指定します。ポリシーラベルを呼び出すポリシーがバインドされているコンテンツスイッチ仮想サーバーのサービスタイプと一致する必要があります。たとえば、TCP ペイロードポリシーを TCP タイプのポリシーラベルだけにバインドできます。TCP ペイロードポリシーを HTTP タイプのポリシーラベルにバインドすることはできません。

コンテンツスイッチングポリシーラベルの各ポリシーは、ターゲット（書き換えポリシーやレスポンダーポリシーなど、他のタイプのポリシーに関連付けられているアクションと同等）または gotoPriorityExpression オプションと invoke オプションのいずれかに関連付けられています。つまり、コンテンツスイッチングポリシーラベル内の特定のポリシーに対して、ターゲットを指定するか、gotoPriorityExpression オプションと invoke オプションを設定できます。また、複数のポリシーが true と評価された場合、true と評価された最後のポリシーのターゲットのみが考慮されます。

Citrix ADC CLI または GUI のいずれかを使用して、コンテンツスイッチングポリシーラベルを構成できます。Citrix ADC CLI では、最初に add cs policylabel コマンドを使用してポリシーラベルを作成します。次に、bind cs policy label コマンドを使用して、一度に 1 つのポリシーをポリシーラベルにバインドします。Citrix ADC GUI では、単一のダイアログボックスで両方のタスクを実行します。

CLI を使用してコンテンツスイッチングポリシーラベルを作成するには

コマンドプロンプトで入力します。

```
add cs policylabel <labelName> <cspolicylabelTypetype>`
<!--NeedCopy-->
```

例**:**

```
add cs policylabel testpollab http
<!--NeedCopy-->
```

CLI を使用してコンテンツスイッチングポリシーラベルの名前を変更するには

コマンドプロンプトで入力します。

```
rename cs policylabel <labelName> <newName>`
<!--NeedCopy-->
```


例:

```
rename cs policylabel oldPolicyLabelName newPolicyLabelName
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーラベルの名前を変更するには

[トラフィック管理] > [コンテンツスイッチング] > [ポリシーラベル] に移動し、ポリシーラベルを選択し、[アクション] リストで [名前の変更] を選択します。

CLI を使用してポリシーをコンテンツスイッチングポリシーラベルにバインドするには

コマンドプロンプトで次のコマンドを入力して、ポリシーをポリシーラベルにバインドし、構成を確認します。

```
bind cs policylabel <labelName> <policyName> <priority>[-targetVserver
    <string>] | [-gotoPriorityExpression <expression>] | [-invoke <
    labeltype> <labelName>] ]

show cs policylabel <labelName>
<!--NeedCopy-->
```

例:

```
bind cs policylabel testpollab test_Pol 100 -targetVserver LBVIP
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 1
        Number of times invoked: 0
        Policy Name: test_Pol
        Priority: 100
        Target Virtual Server: LBVIP
<!--NeedCopy-->
```

> 注
>
> ポリシーがアクションで構成されている場合、ターゲット仮想サーバー（targetVserver）、優先順位式（gotoPriorityExpression）に移動、および呼び出し（呼び出し）パラメーターは必要ありません。ポリシーにアクションが設定されていない場合は、targetVserver、gotoPriorityExpression、および呼び出しのパラメ

> ータの少なくとも 1 つを設定する必要があります。

CLI を使用してポリシーラベルからポリシーのバインドを解除するには

コマンドプロンプトで次のコマンドを入力して、ポリシーラベルからポリシーをバインド解除し、構成を確認します。

```
unbind cs policylabel <labelName> <policyName>

show cs policylabel <labelName>
<!--NeedCopy-->
```

例:

```
unbind cs policylabel testpollab test_Pol
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 0
        Number of times invoked: 0
<!--NeedCopy-->
```

CLI を使用してポリシーラベルを削除するには

コマンドプロンプトで入力します。

```
rm cs policylabel <labelName>
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーラベルを管理するには

[トラフィック管理] > [コンテンツスイッチング] > [ポリシーラベル] に移動し、ポリシーラベルを設定し、ラベルにポリシーをバインドし、オプションで優先順位、gotoPriority 式、呼び出しオプションを指定します。

コンテンツスイッチング仮想サーバーへのポリシーのバインド

コンテンツスイッチ仮想サーバーとコンテンツスイッチポリシーの作成後、各ポリシーをコンテンツスイッチ仮想サーバーにバインドします。ポリシーをコンテンツスイッチ仮想サーバーにバインドするときには、ターゲットとなる負荷分散仮想サーバーを指定します。

> 注
>
> コンテンツスイッチングポリシーでデフォルトの構文規則が使用されている場合は、そのポリシーのコンテンツスイッチングアクションを設定できます。アクションを構成する場合、ポリシーをコンテンツスイッチ仮想サーバーにバインドするときではなく、アクションを構成するときにターゲットの負荷分散仮想サーバーを指定する必要があります。コンテンツスイッチングアクションの設定の詳細については、「コンテンツスイッチングアクションの設定」セクションを参照してください。

ポリシーをコンテンツスイッチング仮想サーバーにバインドし、**CLI** を使用してターゲットの負荷分散仮想サーバーを選択するには

コマンドプロンプトで入力します。

```
1 bind cs vserver <name>[-lbvserver<string> -targetLBVServer<string> -
     policyname <string> -priority <positive_integer>] [-
     gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )]
     [-invoke (<labelType> <labelName>)]
2 <!--NeedCopy-->
```

例:

```
1 bind cs vserver csw-vip2 -policyname csw-ape-policy2 -priority 14 -
     gotoPriorityExpression NEXT
2
3 bind cs vserver csw-vip3 -policyname rewrite-policy1 -priority 17 -
     gotoPriorityExpression
4 'q.header(”a”).count' -flowtype REQUEST -invoke policylabel label1
5
6 bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -
     priority 20
7 <!--NeedCopy-->
```

> 注
>
> ポリシーにアクションがある場合、パラメータ、ターゲット負荷分散仮想サーバ（targetVserver）、優先度式（gotoPriorityExpression）、および呼び出しメソッド（invoke）は使用できません。

ポリシーをコンテンツスイッチング仮想サーバーにバインドし、**GUI** を使用してターゲットの負荷分散仮想サーバーを選択するには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開き、

[コンテンツスイッチングポリシーのバインディング] セクションでポリシーを仮想サーバーにバインドし、ターゲットの負荷分散仮想サーバーを指定します。

コンテンツスイッチングのためのポリシーベースのロギングの構成

コンテンツスイッチングポリシーのポリシーベースのログを設定できます。ポリシーベースのロギングでは、ログメッセージの形式を指定できます。ログメッセージの内容は、コンテンツスイッチングポリシーでデフォルトの構文式を使用して定義されます。ポリシーで指定されているコンテンツスイッチングアクションが実行されると、Citrix ADC アプライアンスは式からログメッセージを作成し、ログファイルにメッセージを書き込みます。ポリシーベースのロギングは、コンテンツスイッチングアクションが実行時にターゲット負荷分散仮想サーバーを識別する構成をテストおよびトラブルシューティングする場合に特に便利です。

> 注
>
> 特定の仮想サーバーにバインドされた複数のポリシーが TRUE と評価され、監査メッセージアクションで構成されている場合、Citrix ADC アプライアンスはすべての監査メッセージアクションを実行しません。コンテンツスイッチングアクションが実行されるポリシーに対して構成されている監査メッセージアクションだけを実行します。

コンテンツスイッチングポリシーのポリシーベースのログを設定するには、まず監査メッセージアクションを構成する必要があります。監査メッセージアクションの構成の詳細については、「監査ログ用の Citrix ADC アプライアンスの構成」を参照してください。監査メッセージアクションを構成したら、コンテンツスイッチングポリシーでアクションを指定します。

CLI を使用して、コンテンツスイッチングポリシーのポリシーベースのログを構成するには

コマンドラインで次のコマンドを入力して、コンテンツスイッチングポリシーのポリシーベースのロギングを設定し、設定を確認します。

```
set cs policy <policyName> -logAction <string>

show cs policy <policyName>
<!--NeedCopy-->
```

例:

```
> set cs policy cspol1 -logAction csLogAction
 Done
> show cs policy cspol1

    Policy: cspol1  Rule: TRUE  Action: csact1
```



```
    LogAction: csLogAction
    Hits: 0

1)  CS Vserver: csvs1
    Priority: 10
 Done
>
<!--NeedCopy-->
```

GUI を使用して、コンテンツスイッチングポリシーのポリシーベースのログを構成するには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [ポリシー] に移動し、ポリシーを開き、[ログアクション] リストでポリシーのログアクションを選択します。

設定の確認

コンテンツスイッチングの設定が正しいことを確認するには、コンテンツスイッチングエンティティを表示する必要があります。コンテンツスイッチ構成の展開後に正しく動作していることを確認するために、サーバーへのアクセス時に生成される統計情報を表示できます。

コンテンツスイッチング仮想サーバーのプロパティの表示

Citrix ADC アプライアンスで構成したコンテンツスイッチ仮想サーバーのプロパティを表示できます。この情報を使用して、仮想サーバーが正しく構成されているかどうかを確認し、必要に応じてトラブルシューティングを行うことができます。名前、IP アドレス、ポートなどの詳細に加えて、仮想サーバーにバインドされたさまざまなポリシーとそのトラフィック管理設定を表示できます。

コンテンツスイッチングポリシーは、優先順位の順に表示されます。複数のポリシーに同じプライオリティが設定されている場合は、仮想サーバにバインドされた順序で表示されます。

> 注
>
> 負荷分散仮想サーバーにトラフィックを転送するようにコンテンツスイッチ仮想サーバーを構成している場合は、負荷分散仮想サーバーのプロパティを表示して、コンテンツスイッチングポリシーを表示することもできます。

CLI を使用してコンテンツスイッチング仮想サーバーのプロパティを表示するには

構成内のすべてのコンテンツスイッチ仮想サーバーの基本プロパティ、または特定のコンテンツスイッチ仮想サーバーの詳細なプロパティを一覧表示するには、コマンドプロンプトで次のコマンドのいずれかを入力します。


```
show cs vserver

show cs vserver <name>
<!--NeedCopy-->
```

例

```
1.
show cs vserver Vserver-CS-1
Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
Last state change was at Thu Jun 30 10:48:59 2011
Time since last state change: 6 days, 20:03:00.760
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none

...
1) Policy : __ESNS_PREBODY_POLICY Priority:0
2) Policy : __ESNS_POSTBODY_POLICY Priority:0

1) Compression Policy Name: __ESNS_CMP_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

2) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

3) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

4) Responder Policy Name: __ESNS_RESPONDER_POLICY Priority: 2147483647
```



```
GotoPriority Expression: END

1) Policy: wiki Target: LBVIP2 Priority: 25 Hits: 0
2) Policy: plain Target: LBVIP1 Priority: 90 Hits: 0
3) Policy: DispOrderTest2 Target: KerbAuthLBVS Priority: 91 Hits: 0
4) Policy: test_Pol Target: LBVIP1 Priority: 92 Hits: 0
5) Policy: PolicyNameTesting Target: LBVIP1 Priority: 100 Hits: 0
 Done
>

show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
…
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
…
Client Idle Timeout: 180 sec
Down state flush: DISABLED
…

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
…
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
…
<!--NeedCopy-->
```

コンテンツスイッチングポリシーの表示

名前、ドメイン、URL または式など、定義したコンテンツスイッチングポリシーのプロパティを表示し、その情報を使用して構成の誤りを見つけたり、何かが必要に応じて機能しない場合のトラブルシューティングを行うことができます。

CLI を使用してコンテンツスイッチングポリシーのプロパティを表示するには

構成内のすべてのコンテンツスイッチングポリシーの基本プロパティまたは特定のコンテンツスイッチングポリシーの詳細プロパティのいずれかを一覧表示するには、コマンドプロンプトで、次のいずれかのコマンドを入力します。

```
show cs policy

show cs policy <PolicyName>
<!--NeedCopy-->
```

例:

```
show cs policy

show cs policy Policy-CS-1
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチングポリシーのプロパティを表示するには

[トラフィック管理] > [コンテンツの切り替え] > [ポリシー] に移動し、ポリシーを選択し、[アクション] リストで [バインディングの表示] を選択します。

ビジュアライザーを使用したコンテンツスイッチング仮想サーバー構成の表示

コンテンツスイッチングビジュアライザは、コンテンツスイッチング設定をグラフィカル形式で表示するために使用できるツールです。ビジュアライザーを使用すると、次の構成項目を表示できます。

- コンテンツスイッチ仮想サーバーがバインドされている負荷分散仮想サーバーの概要。
- 負荷分散仮想サーバーにバインドされているすべてのサービスとサービスグループ、およびサービスにバインドされているすべてのモニター。
- 表示された要素の設定の詳細。
- コンテンツスイッチ仮想サーバーにバインドされたポリシー。これらのポリシーは、コンテンツスイッチングポリシーである必要はありません。書き換えポリシーなど、多くの種類のポリシーは、コンテンツスイッチング仮想サーバーにバインドできます。

コンテンツスイッチングおよび負荷分散のセットアップでさまざまな要素を構成したら、構成全体をアプリケーションテンプレートファイルにエクスポートできます。

> 注
>
> ビジュアライザーにはグラフィカルインターフェイスが必要なため、GUI からのみ使用できます。

GUI のビジュアライザーを使用してコンテンツスイッチング構成を表示するには

1. [トラフィック管理**] > [**コンテンツスイッチング**] > [**仮想サーバー] に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。
3. [コンテンツスイッチングビジュアライザー] ウィンドウでは、次のように表示可能領域を調整できます。
   - [拡大**]** および **[**縮小**]** アイコンをクリックして、表示領域を拡大または縮小します。
   - [イメージを保存**]** アイコンをクリックして、グラフをイメージファイルとして保存します。
   - 「検索場所」テキストフィールドに、検索する項目の名前を入力します。項目を識別するのに十分な文字を入力すると、その位置が強調表示されます。検索を制限するには、ドロップダウンメニューをクリックし、検索する要素のタイプを選択します。
4. この仮想サーバーにバインドされているエンティティの構成の詳細を表示するには、次の操作を行います。
   - 仮想サーバにバインドされているポリシーを表示するには、ダイアログ・ボックスの上部にあるツールバーで、1 つ以上の機能固有のポリシー・アイコンを選択します。ポリシーラベルが設定されている場合は、メインビュー領域に表示されます。
   - バインドされたサービスまたはサービスグループの構成の詳細を表示するには、サービスのアイコンをクリックし、[関連タスク] タブをクリックして、[メンバーサービスの表示] をクリックします。
   - モニタの構成の詳細を表示するには、モニタのアイコンをクリックし、[関連タスク] タブをクリックし、[モニタの表示] をクリックします。
5. コンテンツスイッチ構成内の仮想サーバーの詳細な統計情報を表示するには、統計情報を表示する仮想サーバーをクリックし、[関連タスク] タブをクリックし、[統計] をクリックします。
6. 負荷分散仮想サーバーのサービスコンテナー間で値が異なる、または定義されていないパラメータの比較リストを表示するには、コンテナーのアイコンをクリックし、[関連タスク] タブをクリックして、[サービス属性の相違] をクリックします。
7. コンテナ内のサービスの監視バインドの詳細を表示するには、[サービス属性の相違] ダイアログボックスで、コンテナの [グループ] 列で [詳細] をクリックします。この比較リストは、どのサービスコンテナにすべてのサービスコンテナに適用する設定があるかを判断するのに役立ちます。
8. 構成内の仮想サーバーが特定の時点で受信した 1 秒あたりの要求数、およびリライト、レスポンダー、およびキャッシュポリシーについて、特定の時点での 1 秒あたりの選択数を表示するには、[統計の表示] をクリックします。統計情報は、ビジュアライザーの各ノードに表示されます。この情報はリアルタイムで更新されません。手動で更新されます。情報を更新するには、[統計情報の更新] をクリックします。

   > 注
   >
   > このオプションは、Citrix ADC nCore ビルドでのみ使用できます。

9. 要素の構成の詳細をドキュメントまたはスプレッドシートにコピーするには、その要素のアイコンをクリックし、[関連タスク]、[プロパティのコピー] の順にクリックし、情報をドキュメントに貼り付けます。
10. ビジュアライザーに表示される構成全体をアプリケーションテンプレートファイルにエクスポートするには、コンテンツスイッチ仮想サーバーのアイコンをクリックし、[関連タスク]、[テンプレートの作成] の順にクリ

ックします。アプリケーションテンプレートを作成するときに、いくつかのポリシー式とアクションで変数を設定できます。アプリケーションテンプレートファイルの作成およびテンプレートの変数の構成の詳細については、AppExpertを参照してください。

基本的なコンテンツスイッチング構成のカスタマイズ

October 7, 2021

基本的なコンテンツスイッチの設定を構成したら、必要に応じてカスタマイズする必要があります。Web サーバが UNIX ベースで、大文字と小文字を区別するパス名を使用する場合は、ポリシー評価で大文字と小文字の区別を設定できます。構成したコンテンツスイッチングポリシーの評価の優先順位を設定することもできます。別々の仮想サーバーを作成する代わりに、複数のポートでリッスンするように HTTP および SSL コンテンツスイッチング仮想サーバーを構成できます。特定の仮想 LAN のコンテンツスイッチングを構成する場合は、リッスンポリシーを使用してコンテンツスイッチング仮想サーバーを構成できます。

ポリシー評価のための大文字と小文字の区別の構成

URL ベースのポリシーで大文字と小文字が区別される URL を処理するように、コンテンツスイッチ仮想サーバーを構成できます。大文字と小文字の区別が設定されている場合、Citrix ADC アプライアンスはポリシーを評価するときに大文字と小文字を考慮します。たとえば、大文字と小文字の区別がオフの場合、URL /a/1.htm と /A/1.HTM は同じものとして扱われます。大文字と小文字の区別がオンの場合、これらの URL は別個として扱われ、別のターゲットに切り替えることができます。

コマンドラインインターフェイスを使用して大文字と小文字の区別を構成するには

コマンドプロンプトで入力します。

```
set cs vserver \<name\> -caseSensitive (ON|OFF)
```

例:

```
1 set cs vserver Vserver-CS-1 -caseSensitive ON
2 <!--NeedCopy-->
```

構成ユーティリティを使用して大文字と小文字の区別を構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] を選択し、[大文字と小文字を区別] を選択します。

ポリシー評価の優先順位の設定

優先順位とは、仮想サーバにバインドされているポリシーが評価される順序を指します。precedence を設定する必要はありません。デフォルトの優先順位が正しく動作することがよくあります。

次のシナリオでは、URL ベースの優先順位またはルールベースの優先順位を設定できます。

- 1 つのポリシーまたは一連のポリシーを最初に適用する必要がある
- 別のポリシーまたはポリシーのセットは、最初のセットが要求と一致しない場合にのみ適用されます。

URL ベースのポリシーを使用した優先順位

着信要求に対して一致する URL が複数ある場合、URL ベースのポリシーの優先順位（優先順位）は次のようになります。

1. ドメインと完全な URL
2. ドメイン、プレフィックス、サフィックス
3. ドメインとサフィックス
4. ドメインとプレフィックス
5. ドメインのみ
6. 完全な URL
7. 接頭辞と接尾辞
8. 接尾辞のみ
9. プレフィックスのみ
10. デフォルト

URL に基づいて優先順位を設定する場合、要求 URL が設定された URL と比較されます。設定された URL のいずれも要求 URL と一致しない場合は、ルールベースのポリシーがチェックされます。リクエスト URL がどのルールベースのポリシーとも一致しない場合、またはリクエストに対して選択されたコンテンツグループがダウンしている場合、リクエストは次のように処理されます。

- コンテンツスイッチ仮想サーバーのデフォルトグループを構成すると、要求はデフォルトグループに転送されます。
- 設定されたデフォルトグループがダウンしている場合、またはデフォルトグループが設定されていない場合は、「HTTP 404 Not Found」エラーメッセージがクライアントに送信されます。

> 注
>
> コンテンツタイプ（画像など）がすべてのクライアントで同じである場合は、URL ベースの優先順位を構成する必要があります。ただし、クライアント属性（Accept-Language など）に基づいて異なる種類のコンテンツを提供する必要がある場合は、ルールベースの優先順位を使用する必要があります。

ルールベースのポリシーを使用した優先順位

デフォルトの設定である規則に基づいて優先順位を設定すると、設定した規則ベースのポリシーに基づいて要求がテストされます。要求がどのルールベースのポリシーとも一致しない場合、または着信要求に対して選択されたコンテンツグループがダウンしている場合、要求は次の方法で処理されます。

- 既定のグループがコンテンツスイッチ仮想サーバー用に構成されている場合、要求は既定のグループに転送されます。
- 設定されたデフォルトグループがダウンしている場合、またはデフォルトグループが設定されていない場合は、「HTTP 404 Not Found」エラーメッセージがクライアントに送信されます。

コマンドラインインターフェイスを使用して優先順位を構成するには

コマンドプロンプトで入力します。

```
set cs vserver \<name\> -precedence ( RULE | URL )
```

例:

```
set cs vserver Vserver-CS-1 -precedence RULE
```

構成ユーティリティを使用して優先順位を構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] を選択し、**[**優先順位] を指定します。

HTTP および **SSL** タイプのコンテンツスイッチング仮想サーバに対する複数ポートのサポート

HTTP および SSL コンテンツスイッチング仮想サーバーが複数のポートでリッスンするように Citrix ADC を構成できます。個別の仮想サーバーを構成する必要はありません。この機能は、URL と他の L7 パラメータの一部に基づいてコンテンツスイッチングの決定を行う場合に特に便利です。同じ IP アドレスと異なるポートを持つ複数の仮想サーバを設定する代わりに、1 つの IP アドレスを構成し、ポートを * として指定できます。その結果、構成サイズも縮小されます。

コマンドラインを使用して複数のポートでリッスンするように **HTTP** または **SSL** コンテンツスイッチ仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add cs vserver \<name\> \<serviceType\> \<IPAddress\> Port \*
```

例


```
> add cs vserver cs1 HTTP 10.102.92.215 *
 Done
> sh cs vserver cs1
        cs1 (10.102.92.215:*) - HTTP    Type: CONTENT
        State: UP
        Last state change was at Tue May 20 01:15:49 2014
        Time since last state change: 0 days, 00:00:03.270
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Vserver IP and Port insertion: OFF
        L2Conn: OFF     Case Sensitivity: ON
        Authentication: OFF
        401 Based Authentication: OFF
        Push: DISABLED  Push VServer:
        Push Label Rule: none
        IcmpResponse: PASSIVE
        RHIstate:  PASSIVE
        TD: 0
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して複数のポートでリッスンするように **HTTP** または **SSL** コンテンツスイッチ仮想サーバーを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、HTTP または SSL タイプの仮想サーバーを作成します。
2. ポートを指定するには、アスタリスク（*）を使用します。

VLAN ごとのワイルドカード仮想サーバーの構成

特定の VLAN 上のトラフィックのコンテンツスイッチングを構成する場合は、指定された VLAN でのみトラフィックを処理するように制限するリッスンポリシーを使用してワイルドカード仮想サーバーを作成できます。

コマンドラインインターフェイスを使用して特定の **VLAN** をリッスンするワイルドカード仮想サーバーを構成するには

コマンドプロンプトで入力します。


```
add cs vserver <name> <serviceType> IPAddress `* Port *` -listenpolicy
    <expression> [-listenpriority <positive_integer>]
<!--NeedCopy-->
```

例:

```
add cs vserver Vserver-CS-vlan1 ANY * *
-listenpolicy "CLIENT.VLAN.ID.EQ(2)"  -listenpriority 10
<!--NeedCopy-->
```

構成ユーティリティを使用して、特定の **VLAN** をリッスンするワイルドカード仮想サーバーを構成するには

[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定します。指定した VLAN 上のトラフィックだけを処理するように制限するリッスンポリシーを指定します。

この仮想サーバーを作成したら、基本的な負荷分散の設定の説明に従って、仮想サーバーを 1 つ以上のサービスにバインドします。

SQL サーバのバージョン設定を構成する

MSSQL タイプのコンテンツスイッチング仮想サーバーに対して、Microsoft® SQL Server® のバージョンを指定できます。一部のクライアントが Microsoft SQL Server 製品と同じバージョンを実行していないことが予想される場合は、バージョン設定をお勧めします。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。

コマンドラインインターフェイスを使用して **Microsoft SQL Server** のバージョンパラメーターを設定するには

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチ仮想サーバーの Microsoft SQL Server バージョンパラメーターを設定し、構成を確認します。

- `set cs vserver \<name\> -mssqlServerVersion \<mssqlServerVersion\>`
- `show cs vserver \<name\>`

例


```
1 > set cs vserver myMSSQLcsvip -mssqlServerVersion 2008R2 Done > show cs
      vserver myMSSQLcsvip myMSSQLcsvip (192.0.2.13:1433) - MSSQL Type:
     CONTENT State: UP . . . . . . MSsql Server Version: 2008R2 . . . . .
      . Done >
2 <!--NeedCopy-->
```

構成ユーティリティを使用して **Microsoft SQL Server** のバージョンパラメーターを設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定し、プロトコルを MySQL に指定します。
2. [詳細設定] で [**MySQL**] を選択し、[サーバーのバージョン] を指定します。

UDP 仮想サーバーの外部 **TCP** ヘルスチェックを有効にする

パブリッククラウドでは、ネイティブロードバランサーを第 1 層として使用する場合、Citrix ADC アプライアンスを第 2 層ロードバランサーとして使用できます。ネイティブロードバランサーは、アプリケーションロードバランサー (ALB) またはネットワークロードバランサー (NLB) です。パブリッククラウドのほとんどは、ネイティブのロードバランサで UDP 正常性プローブをサポートしていません。UDP アプリケーションの状態を監視するために、パブリッククラウドでは、サービスに TCP ベースのエンドポイントを追加することをお勧めします。エンドポイントは UDP アプリケーションの正常性を反映します。

Citrix ADC アプライアンスは、UDP 仮想サーバーの外部 TCP ベースのヘルスチェックをサポートします。この機能により、コンテンツスイッチング仮想サーバおよび設定されたポートの VIP に TCP リスナーが導入されます。TCP リスナーは、仮想サーバーのステータスを反映します。

CLI を使用して **UDP** 仮想サーバーの外部 **TCP** ヘルスチェックを有効にするには

コマンドプロンプトで次のコマンドを入力して、TcpProbeport オプションを指定して外部 TCP ヘルスチェックを有効にします。

```
1 add cs vserver <name> <protocol> <IPAddress> <port> -tcpProbePort <
     tcpProbePort>
2 <!--NeedCopy-->
```

例**:**

```
1 add cs vserver Vserver-CS-1 UDP 10.102.29.161 5002 -tcpProbePort 5000
2 <!--NeedCopy-->
```


GUI を使用して **UDP** 仮想サーバーの外部 **TCP** ヘルスチェックを有効にするには

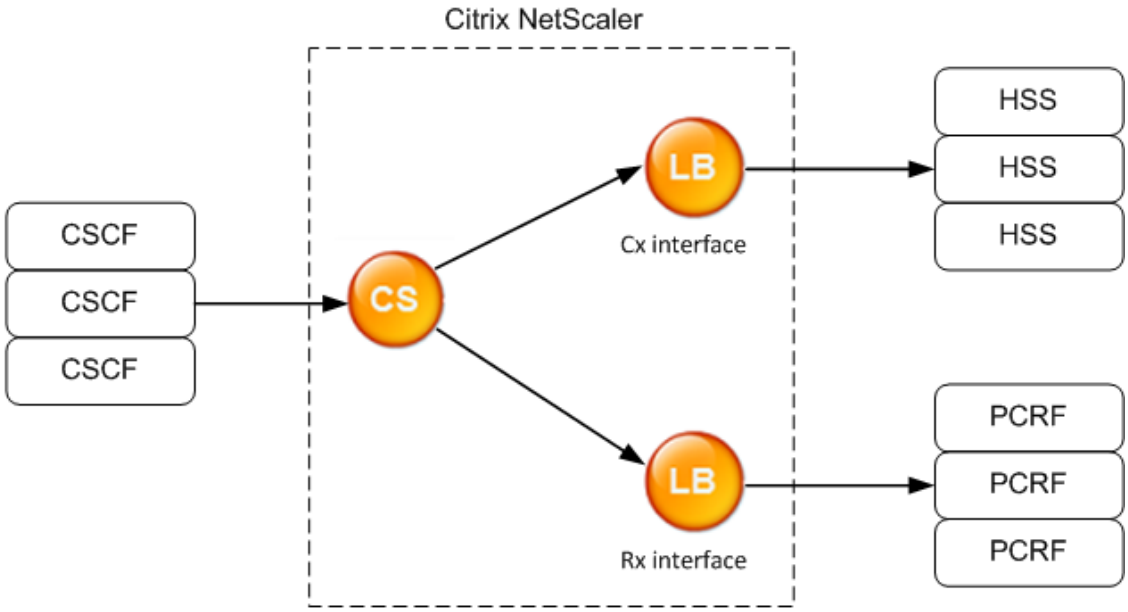
1. [トラフィック管理**] > [**コンテンツスイッチング**] > [**仮想サーバー] に移動し、仮想サーバーを作成します。
2. [**Add**] をクリックして、仮想サーバーを作成します。
3. [基本設定] ペインで、[**TCP** プローブポート**]** フィールドにポート番号を追加します。
4. [**OK**] をクリックします。

直径プロトコルのコンテンツスイッチ

October 7, 2021

Diameter-protocol トラフィックの場合、Citrix ADC アプライアンス（または仮想アプライアンス）が、メッセージの内容（メッセージ内の AVP 値）に基づいてパケットを負荷分散して適切な宛先に転送するリレーエージェントとして機能するように設定できます。アプライアンスはアプリケーションレベルの処理を実行しないため、構成済みのコンテンツスイッチングポリシーで指定されているとおりに、すべての直径アプリケーションにリレーサービスを提供します。したがって、アプライアンスは、クライアントが直径接続を確立するときに、機能交換応答（CEA）メッセージでリレーアプリケーション ID をアドバタイズします。直径ノードを表すコンテンツスイッチ仮想サーバー、負荷分散仮想サーバー、およびサービスを構成する必要があります。要求がコンテンツスイッチ仮想サーバーに到達すると、仮想サーバーはその種類の要求に関連付けられたコンテンツスイッチングポリシーを適用します。ポリシーを評価した後、コンテンツスイッチ仮想サーバーは要求を適切な負荷分散仮想サーバーにルーティングし、適切なサービスに送信します。

直径インタフェースは、異なる直径ノード間の接続を提供します。次の展開例では、Cx および Rx インターフェイスを使用しています。Cx インターフェイスは、CSCF と HSS の間の接続を提供します。Rx インターフェースは、CSCF と PCRF 間の接続を提供します。すべてのメッセージは Citrix ADC アプライアンスに届きます。メッセージが Cx インターフェイスか Rx インターフェイスか、および定義されたコンテンツスイッチポリシーに応じて、Citrix ADC は適切な負荷分散サーバープールを選択します。



CSCF=Call Session Control Function
HSS=Home Subscriber Server
PCRF=Policy and Charging Rules Function

構成例

1. エンティティごとに、サービス、負荷分散サーバーを作成し、サービスを仮想サーバーにバインドします。

```
add service svc_pcrf[1-3] 1.1.1.1[1-3] DIAMETER 3868
add service svc_hss[1-3] 1.1.1.2[1-3] DIAMETER 3868
add lb vserver vs_rx DIAMETER -persistenceType DIAMETER -
    persistavpno 263
add lb vserver vs_cx DIAMETER -persistenceType DIAMETER -
    persistavpno 263
bind lb vserver vs_rx svc_pcrf[1-3]
bind lb vserver vs_cx svc_hss[1-3]
<!--NeedCopy-->
```

2. コンテンツスイッチング仮想サーバーと 2 つのアクション (負荷分散仮想サーバーごとに 1 つ) を作成します。2 つのコンテンツスイッチングポリシーを作成し、これらのポリシーをコンテンツスイッチ仮想サーバーにバインドし、各ポリシーの優先順位を指定します。

```
add cs vserver cs_diameter DIAMETER 10.1.1.10 3868
add cs action cx_action -targetLBVserver vs_cx
```



```
3 add cs action rx_action  - targetLBvserver vs_rx
4 add cs policy cx_policy -rule ”DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
     (16777216)” -action cx_action
5 add cs policy rx_policy -rule ”DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
     (16777236)” -action rx_action
6 bind cs vserver cs_diameter -policyName rx_policy -priority 100
7 bind cs vserver cs_diameter -policyName cx_policy -priority 110
8 <!--NeedCopy-->
```

コンテンツスイッチングセットアップを障害から保護する

October 7, 2021

コンテンツスイッチング仮想サーバーがダウンしたり、過剰なトラフィックを処理できなかったりした場合、またはその他の理由で、コンテンツスイッチングが失敗する場合があります。失敗の可能性を減らすには、次の措置を講じて、コンテンツスイッチの設定を障害から保護します。

バックアップ仮想サーバーの構成

プライマリコンテンツスイッチング仮想サーバーが「ダウン」または「無効」とマークされている場合、Citrix ADC アプライアンスはバックアップコンテンツスイッチング仮想サーバーに要求を送信できます。また、サイトの停止またはメンテナンスに関する通知メッセージをクライアントに送信することもできます。バックアップコンテンツスイッチ仮想サーバは、プロキシであり、クライアントに対して透過的です。

バックアップ仮想サーバを設定する場合、構成パラメータの [Disable Primary When Down] を指定して、プライマリ仮想サーバが復帰したときに、手動でプライマリとして引き継がれるまで、プライマリ仮想サーバがセカンダリとして維持されるようにすることができます。バックアップのためにサーバー上のデータベースへの更新を確実に保持し、プライマリ仮想サーバーを復元する前にデータベースを同期できるようにする場合に役立ちます。

コンテンツスイッチ仮想サーバーを作成するとき、または既存のコンテンツスイッチ仮想サーバーのオプションのパラメーターを変更するときに、バックアップコンテンツスイッチ仮想サーバーを構成できます。既存のバックアップコンテンツスイッチ仮想サーバーのバックアップコンテンツスイッチ仮想サーバーを構成することもできます。これにより、カスケードされたバックアップコンテンツスイッチ仮想サーバーが作成されます。カスケード・バックアップ・コンテンツ・スイッチング仮想サーバの最大深さは 10 です。アプライアンスは、稼働中のバックアップコンテンツスイッチ仮想サーバーを検索し、コンテンツスイッチ仮想サーバーにアクセスしてコンテンツを配信します。

> 注
>
> コンテンツスイッチ仮想サーバーがバックアップコンテンツスイッチ仮想サーバーとリダイレクト URL の両方で構成されている場合、バックアップコンテンツスイッチ仮想サーバーはリダイレクト URL よりも優先されます。リダイレクトは、プライマリおよびバックアップ仮想サーバがダウンしている場合に使用されます。

CLI を使用してバックアップコンテンツスイッチング仮想サーバーをセットアップするには

コマンドプロンプトで入力します。

```
set cs vserver <name> -backupVserver <string> -disablePrimaryOnDown (ON
    |OFF)
<!--NeedCopy-->
```

例

```
set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -
    disablePrimaryOnDown ON
<!--NeedCopy-->
```

GUI を使用してバックアップコンテンツスイッチング仮想サーバーをセットアップするには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定し、プロトコルを MySQL に指定します。
2. [詳細設定] で、[保護] を選択し、バックアップ仮想サーバーを指定します。

余分なトラフィックをバックアップ仮想サーバーに迂回させる

spillover オプションは、コンテンツスイッチング仮想サーバーへの接続数が設定されたしきい値を超えた場合に、コンテンツスイッチング仮想サーバーに着信した新しい接続をバックアップコンテンツスイッチング仮想サーバーに転送します。しきい値は動的に計算されるか、値を設定できます。仮想サーバーで確立された接続の数（TCP）がしきい値と比較されます。接続数がしきい値に達すると、新しい接続はバックアップコンテンツスイッチ仮想サーバーに転送されます。

バックアップコンテンツスイッチング仮想サーバが設定されたしきい値に達し、負荷をかけることができない場合、プライマリコンテンツスイッチング仮想サーバはすべての要求をリダイレクト URL に転送します。プライマリコンテンツスイッチ仮想サーバにリダイレクト URL が構成されていない場合、後続の要求はドロップされます。

CLI を使用して、新しい接続をバックアップ仮想サーバーに転送するようにコンテンツスイッチング仮想サーバーを構成するには

コマンドプロンプトで入力します。


```
set cs vserver <name> -soMethod <methodType> -soThreshold <
    thresholdValue> -soPersistence <persistenceValue> -
    soPersistenceTimeout <timeoutValue>
<!--NeedCopy-->
```

例

```
set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

GUI を使用して、新しい接続をバックアップ仮想サーバーに転送するようにコンテンツスイッチング仮想サーバーを設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定し、プロトコルを MySQL に指定します。
2. [詳細設定] で、[保護] を選択し、スピルオーバーを構成します。

リダイレクト **URL** の構成

タイプ HTTP または HTTPS のコンテンツスイッチング仮想サーバーが DOWN または DISABLED の場合に、Citrix ADC アプライアンスのステータスを伝達するようにリダイレクト URL を構成できます。この URL は、ローカルでもリモートでもかまいません。

リダイレクト URL には、絶対 URL または相対 URL を指定できます。構成したリダイレクト URL に絶対 URL が含まれている場合、着信した HTTP 要求で指定された URL に関係なく、その絶対 URL にリダイレクトされます。構成したリダイレクト URL にドメイン名のみが含まれている場合（相対 URL）、そのドメインに着信 URL を追記した場所にリダイレクトされます。

絶対 URL の使用をお勧めします。つまり、相対 URL ではなく、たとえば www.example.com/で終わる URL です。相対 URL リダイレクトにより、脆弱性スキャナが誤検知を報告する可能性があります。

> 注
>
> コンテンツスイッチ仮想サーバーがバックアップ仮想サーバーとリダイレクト URL の両方で構成されている場合、バックアップ仮想サーバーはリダイレクト URL よりも優先されます。リダイレクト URL は、プライマリおよびバックアップ仮想サーバがダウンしている場合に使用されます。

リダイレクトが構成され、コンテンツスイッチング仮想サーバーが使用できない場合、アプライアンスは HTTP 302 リダイレクトをユーザーのブラウザに発行します。

CLI を使用してコンテンツスイッチング仮想サーバーが利用できない場合のリダイレクト **URL** を構成するには

コマンドプロンプトで入力します。

```
set cs vserver <name> -redirectURL <URLValue>
<!--NeedCopy-->
```

例

```
set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/
   mysite/maintenance
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチング仮想サーバーが利用できない場合のリダイレクト **URL** を構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定し、プロトコルを MySQL に指定します。
2. [詳細設定] で [保護] を選択し、リダイレクト URL を指定します。

状態更新オプションの構成

コンテンツスイッチング機能により、ユーザーに提示された特定のコンテンツに基づいて、クライアント要求を複数のサーバーに分散できます。コンテンツスイッチングを効率的に行うために、コンテンツスイッチング仮想サーバーは、コンテンツの種類に応じてトラフィックを負荷分散仮想サーバーに分散し、負荷分散仮想サーバーは、指定された負荷分散方法に従ってトラフィックを物理サーバーに分散します。

スムーズなトラフィック管理のためには、コンテンツスイッチング仮想サーバーが負荷分散仮想サーバーの状態を知ることが重要です。状態更新オプションは、バインドされている負荷分散仮想サーバーが DOWN とマークされている場合、コンテンツスイッチング仮想サーバーを DOWN とマークするのに役立ちます。負荷分散仮想サーバは、バインドされているすべての物理サーバが「ダウン」とマークされている場合、「ダウン」とマークされます。

状態更新が無効になっている場合**:**

コンテンツスイッチング仮想サーバーのステータスが UP としてマークされます。バインドされた負荷分散仮想サーバーが稼働していなくても、UP のままです。

[状態更新**]** が有効な場合**:**

コンテンツスイッチング仮想サーバーを追加すると、最初はそのステータスが DOWN として表示されます。ステータスが UP の負荷分散仮想サーバーをバインドすると、コンテンツスイッチング仮想サーバーのステータスが UP になります。

複数の負荷分散仮想サーバーがバインドされ、そのうちの 1 つが既定として指定されている場合、コンテンツスイッチ仮想サーバーの状態には、既定の負荷分散仮想サーバーの状態が反映されます。

複数の負荷分散仮想サーバーが既定として指定されていない状態でバインドされている場合、コンテンツスイッチング仮想サーバーのステータスは、バインドされたすべての負荷分散仮想サーバーが UP の場合にのみ UP とマークされます。

CLI を使用して状態更新オプションを構成するには

コマンドプロンプトで入力します。

```
1 add cs vserver <name> <protocol> <ipAddress> <port> -stateUpdate
     ENABLED
2 <!--NeedCopy-->
```

例

```
1  add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED
      -cltTimeout 180
2 <!--NeedCopy-->
```

GUI を使用して状態更新オプションを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定し、プロトコルを MySQL に指定します。
2. [詳細設定] で、[トラフィックの設定] を選択し、[状態の更新] を選択します。

サージキューのフラッシュ

物理サーバが要求の急増を受信すると、現在接続しているクライアントへの応答が遅くなり、ユーザが不満になり、不満や不満が残ります。多くの場合、オーバーロードにより、クライアントはエラーページを受信します。このような過負荷を回避するために、Citrix ADC アプライアンスは、サービスへの新しい接続を確立する速度を制御するサージ保護などの機能を提供します。

アプライアンスは、クライアントと物理サーバ間の接続の多重化を行います。サーバ上のサービスにアクセスするためのクライアント要求を受信すると、アプライアンスはすでに確立されているサーバへの接続を空いているか探します。空き接続が見つかった場合、その接続を使用してクライアントとサーバー間の仮想リンクを確立します。既存の空き接続が見つからない場合、アプライアンスはサーバーとの新しい接続を確立し、クライアントとサーバーの間に

仮想リンクを確立します。ただし、アプライアンスがサーバーとの新しい接続を確立できない場合、クライアント要求をサージキューに送信します。負荷分散またはコンテンツスイッチング仮想サーバーにバインドされているすべての物理サーバーがクライアント接続の上限（最大クライアント値、サージ保護しきい値、またはサービスの最大容量）に達した場合、アプライアンスはどのサーバーとも接続を確立できません。サージ保護機能は、サージキューを使用して、物理サーバとの接続が開かれる速度を調整します。アプライアンスは、仮想サーバにバインドされたサービスごとに異なるサージキューを維持します。

サージキューの長さは、アプライアンスが接続を確立できない要求が来ると増加し、キュー内の要求がサーバーに送信されるか、要求がタイムアウトしてキューから削除されるたびに減少します。

サービスまたはサービスグループのサージキューが長くなりすぎる場合は、それをフラッシュすることをお勧めします。特定のサービスまたはサービスグループ、または負荷分散仮想サーバーにバインドされているすべてのサービスとサービスグループのサージキューをフラッシュできます。サージキューをフラッシュしても、既存の接続には影響しません。サージキューに存在する要求だけが削除されます。これらの要求の場合、クライアントは新しい要求を行う必要があります。

コンテンツスイッチング仮想サーバーのサージキューをフラッシュすることもできます。コンテンツスイッチング仮想サーバーが特定の負荷分散仮想サーバーにいくつかの要求を転送し、負荷分散仮想サーバーが他のいくつかの要求も受信する場合、コンテンツスイッチング仮想サーバーのサージキューをフラッシュすると、このコンテンツスイッチングから受信した要求のみが仮想サーバーがフラッシュされます。負荷分散仮想サーバーのサージキュー内の他の要求はフラッシュされません。

> 注
>
> キャッシュリダイレクト、認証、VPN、または GSLB 仮想サーバーまたは GSLB サービスのサージキューをフラッシュすることはできません。
> [ソース IP の使用（USIP）] が有効になっている場合は、サージ保護機能を使用しないでください。

CLI を使用してサージキューをフラッシュするには

flush ns surgeQ コマンドは、次のように動作します。

- サージキューをフラッシュする必要があるサービス、サービスグループ、または仮想サーバの名前を指定できます。
- コマンドの実行中に名前を指定すると、指定したエンティティのサージキューがフラッシュされます。複数のエンティティが同じ名前を持っている場合、アプライアンスはそれらすべてのエンティティのサージキューをフラッシュします。
- コマンドの実行中にサービスグループの名前、サーバー名、およびポートを指定すると、アプライアンスは指定されたサービスグループメンバーのみのサージキューをフラッシュします。
- サービスグループメンバー（<serverName> および <port>）を直接指定するには、サービスグループの名前（<name>）を指定します。また、を指定 <port> しないとを指定できません <serverName>。特定のサービスグループメンバーのサージキューをフラッシュする場合は、<serverName> および <port> を指定します。
- 名前を指定せずにコマンドを実行すると、アプライアンスはアプライアンスに存在するすべてのエンティティのサージキューをフラッシュします。

- サービスグループメンバーがサーバ名で識別される場合は、このコマンドでサーバ名を指定する必要があります。その IP アドレスを指定することはできません。

コマンドプロンプトで入力します。

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>].
<!--NeedCopy-->
```

例

```
1. flush ns surgeQ  -name SVC1ANZGB  -serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual
    server that is named SVC1ANZGB and has IP address as 10.10.10

2. flush ns surgeQ
The above command flushes all the surge queues on the appliance.
<!--NeedCopy-->
```

GUI を使用してサージキューをフラッシュするには

Traffic Management > Content Switching > Virtual Servers に移動して仮想サーバーを選択し、アクションリストで **Flush Surge Queue** を選択します。

コンテンツスイッチング設定の管理

October 7, 2021

コンテンツスイッチングの設定が構成された後、定期的な変更が必要になる場合があります。オペレーティングシステムまたはソフトウェアが更新された場合、またはハードウェアが摩耗して交換された場合は、セットアップを停止する必要がある場合があります。セットアップの負荷が増加し、より多くのリソースが必要になる場合があります。パフォーマンスを向上させるために構成を変更することもできます。

これらのタスクでは、コンテンツスイッチング仮想サーバーからのポリシーのバインド解除、またはコンテンツスイッチング仮想サーバーの無効化または削除が必要になる場合があります。設定を変更した後、サーバーを再度有効にしてポリシーを再バインドする必要がある場合があります。仮想サーバーの名前を変更することもできます。

コンテンツスイッチング仮想サーバーからのポリシーのバインド解除

コンテンツスイッチポリシーを仮想サーバーからバインド解除すると、要求を送信する場所を決定するときに、仮想サーバーにはそのポリシーが含まれなくなります。

CLI を使用して、コンテンツスイッチング仮想サーバーからポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
unbind cs vserver <name> -policyname <string>
```

例:

```
unbind cs vserver Vserver-CS-1 -policyname Policy-CS-1
```

GUI を使用して、コンテンツスイッチング仮想サーバーからポリシーをバインド解除するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [ポリシー] セクションをクリックし、ポリシーを選択して [バインド解除] をクリックします。

コンテンツスイッチング仮想サーバの削除

通常、コンテンツスイッチング仮想サーバーを削除するのは、仮想サーバーが不要になったときだけです。コンテンツスイッチング仮想サーバーを削除すると、Citrix ADC アプライアンスはまずコンテンツスイッチング仮想サーバーからすべてのポリシーのバインドを解除し、その後削除します。

CLI を使用してコンテンツスイッチング仮想サーバーを削除するには

コマンドプロンプトで入力します。

```
rm cs vserver <name>
```

例:

```
rm cs vserver Vserver-CS-1
```

GUI を使用してコンテンツスイッチング仮想サーバーを削除するには

[トラフィック管理] **>** [コンテンツの切り替え] **>** [仮想サーバー] に移動し、仮想サーバーを選択して [削除] をクリックします。

コンテンツスイッチング仮想サーバーの無効化と再有効化

コンテンツスイッチング仮想サーバーは、作成時にデフォルトで有効になります。メンテナンスのために、コンテンツスイッチング仮想サーバーを無効にすることができます。コンテンツスイッチング仮想サーバーを無効にすると、コンテンツスイッチング仮想サーバーの状態が Out of Service に変わります。アウト・オブ・サービスでは、コンテンツスイッチング仮想サーバーは要求に応答しません。

CLI を使用して仮想サーバーを無効または再度有効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `disable cs vserver <name>`
- `enable cs vserver <name>`

例:

```
disable cs vserver Vserver-CS-1
enable cs vserver Vserver-CS-1
```

GUI を使用して仮想サーバーを無効または再度有効にするには

[トラフィック管理] **>** [コンテンツの切り替え] **>** [仮想サーバー] に移動し、仮想サーバーを選択し、[アクション] リストで [有効] または [無効] を選択します。

コンテンツスイッチング仮想サーバの名前変更

コンテンツスイッチング仮想サーバーの名前は、バインドを解除することなく変更できます。新しい名前は、Citrix ADC 構成のすべての影響を受ける部分に自動的に反映されます。

CLI を使用して仮想サーバーの名前を変更するには

コマンドプロンプトで入力します。

```
rename cs vserver <name> <newName>
```

例:

```
1 `rename cs vserver Vserver-CS-1 Vserver-CS-2`
```

GUI を使用して仮想サーバーの名前を変更するには

[トラフィック管理] **>** [コンテンツの切り替え] **>** [仮想サーバー] に移動し、仮想サーバーを選択し、[アクション] リストで [名前の変更] を選択します。

コンテンツスイッチングポリシーの管理

ルールを構成するか、ポリシーの URL を変更することにより、既存のポリシーを変更するか、ポリシーを削除することができます。既存の高度なコンテンツスイッチングポリシーの名前を変更することもできます。URL に基づいて異なるポリシーを作成できます。URL ベースのポリシーは、次の表で説明するように、異なるタイプにすることができます。

詳細については、「URL ベースのポリシーの例」を参照してください。

> 注
>
> 従来のポリシー式または高度なポリシー式を使用して、ルールベースのコンテンツスイッチングを設定できます。

CLI を使用してポリシーを変更、削除、または名前変更するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `set cs policy <policyName> [-domain <domainValue>] [-rule <ruleValue>] [-url <URLValue>]`
- `rm cs policy <policyName>`
- `rename cs policy <policyName> <newPolicyName>`

例:

```
set cs policy Policy-CS-1 -domain ”www.domainxyz.com”

set cs policy Policy-CS-1 -rule ”CLIENT.IP.SRC.SUBNET(22).EQ
    (10.100.148.0)”

set cs policy Policy-CS-2 -rule ”SYS.TIME.BETWEEN(GMT 2010 Jun,GMT 2010
     Jul)”

set cs policy Policy-CS-1 -url /sports/*

rename cs policy Policy-CS-1 Policy-CS-11

rm cs policy Policy-CS-1
```

GUI を使用してポリシーを変更、削除、または名前変更するには

1. **Traffic Management > Content Switching > Policies** に移動します。
2. ポリシーを選択し、削除または編集するか、[アクション] リストで [名前の変更] をクリックします。

クライアント接続の管理

October 7, 2021

クライアント接続を効率的に管理するために、Citrix ADC アプライアンスのコンテンツスイッチ仮想サーバーを構成して、次の機能を使用できます。

- **ICMP** 応答の設定。Citrix ADC アプライアンスは、設定に従って PING 要求に ICMP 応答を送信するように構成できます。仮想サーバーに対応する IP アドレスで、ICMP RESPONSE を VSVR_CNTRLD に設定します。仮想サーバーで、ICMP 仮想サーバー RESPONSE を設定します。
  仮想サーバーでは、次の設定を行うことができます。
  - すべての仮想サーバーで ICMP 仮想サーバー RESPONSE を PASSIVE に設定すると、Citrix ADC アプライアンスは常に応答します。
  - すべての仮想サーバーで ICMP 仮想サーバー RESPONSE を ACTIVE に設定すると、1 つの仮想サーバーが稼働している場合でも ADC アプライアンスが応答します。
  - ICMP 仮想サーバーの RESPONSE を ACTIVE に設定し、PASSIVE を他のサーバーに設定すると、ACTIVE に設定された 1 つの仮想サーバーが稼働している場合でも ADC アプライアンスが応答します。

クライアント要求をキャッシュにリダイレクトする

Citrix ADC キャッシュリダイレクト機能は、HTTP 要求をキャッシュにリダイレクトします。キャッシュリダイレクト機能を適切に実装することで、HTTP リクエストへの応答の負担を大幅に軽減し、Web サイトのパフォーマンスを向上させることができます。

キャッシュは、頻繁に要求された HTTP コンテンツを格納します。仮想サーバーでキャッシュリダイレクトを構成すると、Citrix ADC アプライアンスはキャッシュ可能な HTTP リクエストをキャッシュに送信し、キャッシュ不可能な HTTP リクエストを元の Web サーバーに送信します。キャッシュリダイレクトの詳細については、「キャッシュリダイレクト」を参照してください。

CLI を使用して仮想サーバーでキャッシュリダイレクトを構成するには

コマンドプロンプトで入力します。

```
set cs vserver \<name\> -cacheable \<Value\>
```

例

```
set cs vserver Vserver-CS-1 -cacheable yes
```

GUI を使用して仮想サーバーでキャッシュリダイレクトを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。

2.「詳細設定」で「トラフィック設定」を選択し、「キャッシュ可能」を選択します。

仮想サーバー接続の遅延クリーンアップの有効化

特定の条件下では、ダウン状態のフラッシュ設定を構成して、サービスまたは仮想サーバーがダウンとマークされたときに既存の接続を終了できます。既存の接続を終了すると、リソースが解放され、場合によっては負荷分散設定の回復が高速化されます。

CLI を使用して仮想サーバーでダウン状態のフラッシュ設定を構成するには

コマンドプロンプトで入力します。

```
set cs vserver \<name\> -downStateFlush \<Value\>
```

例

```
1 set cs vserver Vserver-CS-1 -downStateFlush enabled
2 <!--NeedCopy-->
```

GUI を使用して仮想サーバーでダウン状態のフラッシュ設定を構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] を選択し、[ダウン状態のフラッシュ] を選択します。

リダイレクト用のポートおよびプロトコルの書き換え

仮想サーバーとそれにバインドされているサービスは、異なるポートを使用する場合があります。サービスがリダイレクトを使用して HTTP 接続に応答する場合、リダイレクトが正常に実行されるように、ポートとプロトコルを変更するように Citrix ADC アプライアンスを構成する必要がある場合があります。これを行うには、redirectPortRewrite 設定を有効にして構成します。

CLI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

コマンドプロンプトで入力します。

```
set cs vserver \<name\> -redirectPortRewrite \<Value\>
```


例

```
set cs vserver Vserver-CS-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[書き換え] を選択します。

要求ヘッダーへの仮想サーバーの **IP** アドレスとポートの挿入

同じサービス上で異なるアプリケーションと通信する仮想サーバーが複数ある場合は、Citrix ADC アプライアンスを構成して、そのサービスに送信される HTTP 要求に適切な仮想サーバーの IP アドレスとポート番号を追加する必要があります。この設定により、サービスで実行されているアプリケーションが、要求を送信した仮想サーバーを識別できるようになります。

プライマリ仮想サーバがダウンし、バックアップ仮想サーバが稼働している場合、バックアップ仮想サーバの構成設定がクライアント要求に追加されます。要求がプライマリ仮想サーバからのものかバックアップ仮想サーバからのものかに関係なく、同じヘッダータグを追加する場合は、両方の仮想サーバで必要なヘッダータグを設定する必要があります。

> 注
>
> このオプションは、ワイルドカード仮想サーバーまたはダミー仮想サーバーではサポートされていません。

CLI を使用して仮想サーバの **IP** アドレスとポートをクライアント要求に挿入するには

コマンドプロンプトで入力します。

```
set cs vserver \<name\> -insertVserverIPPort \<vServerIPPORT\>
```

例

```
set cs vserver Vserver-CS-1 -insertVserverIPPort 10.201.25.136:80
<!--NeedCopy-->
```


GUI を使用して、クライアント要求に仮想サーバの **IP** アドレスとポートを挿入するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. **[**詳細設定**]** で [トラフィック設定] を選択し、[仮想サーバーの IP ポート挿入] リストで [VIPADDR] または [V6TOV4MAPPING] を選択し、仮想サーバーの [IP ポート挿入値] でポートヘッダーを指定します。

アイドル状態のクライアント接続のタイムアウト値の設定

構成されたタイムアウト期間が経過した後、アイドル状態のクライアント接続を終了するように仮想サーバーを構成できます。この設定を構成すると、Citrix ADC アプライアンスは指定した時間待機し、その時間以降にクライアントがアイドル状態になると、クライアント接続を閉じます。

CLI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
set cs vserver \<name\> -cltTimeout \<Value\>
```

例

```
1 set cs vserver Vserver-CS-1 -cltTimeout 100
2 <!--NeedCopy-->
```

GUI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、**[**トラフィック設定**]** を選択し、**[**クライアントアイドルタイムアウト] の値を指定します。

4 タプルおよびレイヤ **2** 接続パラメータを使用した接続の識別

コンテンツスイッチング仮想サーバーに L2Conn オプションを設定できるようになりました。L2Conn オプションを設定すると、コンテンツスイッチ仮想サーバーへの接続は、4 タプル (<source IP>::<source port><destination IP>:<destination port>) とレイヤ 2 接続パラメータの組み合わせによって識別されます。レイヤ 2 接続パラメータは、MAC アドレス、VLAN ID、およびチャネル ID です。

CLI を使用してコンテンツスイッチング仮想サーバーの **L2Conn** オプションを設定するには

コマンドラインで次のコマンドを入力して、コンテンツスイッチ仮想サーバーの L2Conn パラメーターを構成し、構成を確認します。


```
-  set cs vserver \<name\> -l2Conn (**ON** | **OFF**)
-  show cs vserver \<name\>
```

例

```
> set cs vserver mycsvserver -l2Conn ON
Done
> show cs vserver mycsvserver
     mycsvserver (192.0.2.56:80) - HTTP   Type: CONTENT
     State: UP
          . . .
          . . .
     L2Conn: ON Case Sensitivity: ON
          . . .
          . . .
Done
>
<!--NeedCopy-->
```

GUI を使用してコンテンツスイッチング仮想サーバーの **L2Conn** オプションを設定するには

1. [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] を選択し、[レイヤ **2** パラメータ] を選択します。

コンテンツスイッチ仮想サーバの永続性サポート

October 7, 2021

アプリケーションは、モノリシックアーキテクチャからマイクロサービスアーキテクチャに移行しています。同じアプリケーションの異なるバージョンがマイクロサービスアーキテクチャに共存できます。Citrix ADC アプライアンスは、アプリケーションの継続的な展開をサポートする必要があります。これは、カナリア展開を実行するプラットフォーム（Spinnaker など）によって実現されます。継続的なデプロイ設定では、アプリケーションの新しいバージョンが自動的にデプロイされ、アプリケーションが安定して完全なトラフィックを取得するまで、段階的にクライアントトラフィックに公開されます。また、クライアントには中断のないサービスが必要です。

Citrix ADC コンテンツスイッチング機能を使用すると、Citrix ADC アプライアンスは、コンテンツスイッチング仮想サーバーにバインドされたポリシーに基づいて、複数の負荷分散仮想サーバーにクライアント要求を分散できます。

継続的な展開では、コンテンツの切り替えを使用して、アプリケーションのさまざまなバージョンを提供する負荷分散仮想サーバーを選択します。

コンテンツスイッチングでは、コンテンツスイッチングポリシーの変更により、特定のアプリケーションバージョンの負荷分散仮想サーバーの選択が実行時に変更されます。この移行中に、一部のセッションが古いバージョンのアプリケーションに存在する場合、そのようなトラフィックは古いバージョンのみで引き続き処理する必要があります。この要件をサポートするために、Citrix ADC アプライアンスは、コンテンツスイッチング仮想サーバーの背後にある複数の負荷分散グループ間で永続性を維持します。コンテンツスイッチング仮想サーバの永続性により、あるバージョンから別のバージョンへのクライアントのシームレスな移行が可能になります。

コンテンツスイッチング仮想サーバーでサポートされる永続性タイプ

コンテンツスイッチング仮想サーバーでは、次の永続タイプがサポートされています。

持続性タイプ	説明
接続元 IP	**SOURCEIP**. 同じクライアント IP アドレスからの接続は、同じ永続性セッションの一部です。詳細については、「送信元 IP アドレスの永続性」を参照してください。
HTTP Cookie	**COOKIEINSERT**. 同じ HTTP Cookie ヘッダーを持つ接続は、同じ永続性セッションの一部です。Citrix ADC アプライアンスが挿入するクッキーの形式は次のとおりです。**NSC_ <vid_str of CSvserver> = <vid_str of Lbvserver>** ここで、**NSC_XXXX** は仮想サーバー名から派生した仮想サーバー ID です。詳細については、「HTTP クッキーの永続性」を参照してください。
SSL Session ID	**SSLSESSION**. 同じ SSL セッション ID を持つ接続は、同じ永続性セッションの一部です。詳細については、「SSL セッション ID の永続性」を参照してください。

HTTP Cookie に基づくパーシステンスに対して、タイムアウト値を設定できます。タイムアウト値を 0 に設定すると、使用されている HTTP Cookie バージョンに関係なく、ADC アプライアンスは有効期限を指定しなくなります。有効期限はクライアントソフトウェアによって異なり、この Cookie はソフトウェアが実行されている場合にのみ有効です。

構成したパーシステンスの種類に応じて、仮想サーバーでは、Citrix ADC アプライアンスのメモリ容量によって課される制限まで、250,000 の同時永続接続または任意の数の永続接続をサポートできます。次の表は、各カテゴリに分

類される永続性のタイプを示しています。

持続性タイプ	サポートされる同時永続接続の数
送信元 IP、SSL セッション ID	250,000
HTTP Cookie	メモリの上限。CookieInsert では、タイムアウトが 0 でない場合、接続の数はメモリによって制限されます。

一部のタイプの永続性は、特定のタイプの仮想サーバに固有のものです。次の表は、各タイプの永続性と、どのタイプの仮想サーバーでサポートされる永続性のタイプを示しています。

持続性タイプ	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEI	はい	はい	はい	はい	はい	はい	いいえ	いいえ
COOKIEINSERT	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
SSLSESSI	いいえ	はい	いいえ	いいえ	はい	はい	いいえ	いいえ

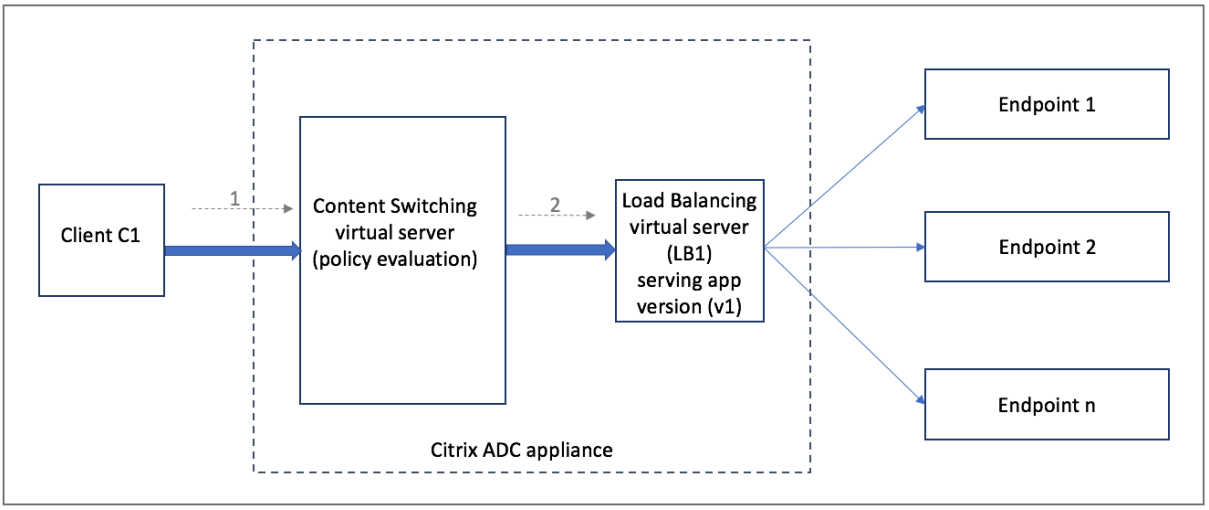
バックアップ永続性のサポート

Cookie の永続性タイプに障害が発生した場合に、バックアップの永続性タイプとしてソース IP の永続性タイプを使用するようにコンテンツスイッチング仮想サーバーを構成できます。これは、マイクロサービスアーキテクチャでのカナリア展開に役立ちます。
Cookie パーシステンスタイプが失敗した場合、アプライアンスはクライアントブラウザが要求に Cookie を返さない場合にのみ、ソース IP ベースのパーシステンスにフォールバックします。ただし、ブラウザがクッキー（必ずしも永続性クッキーではない）を返す場合、ブラウザはクッキーをサポートしているとみなされるため、バックアップパーシステンスはトリガーされません。
バックアップ永続性のタイムアウト値を設定することもできます。「タイムアウト」は、持続性セッションが有効になる期間です。

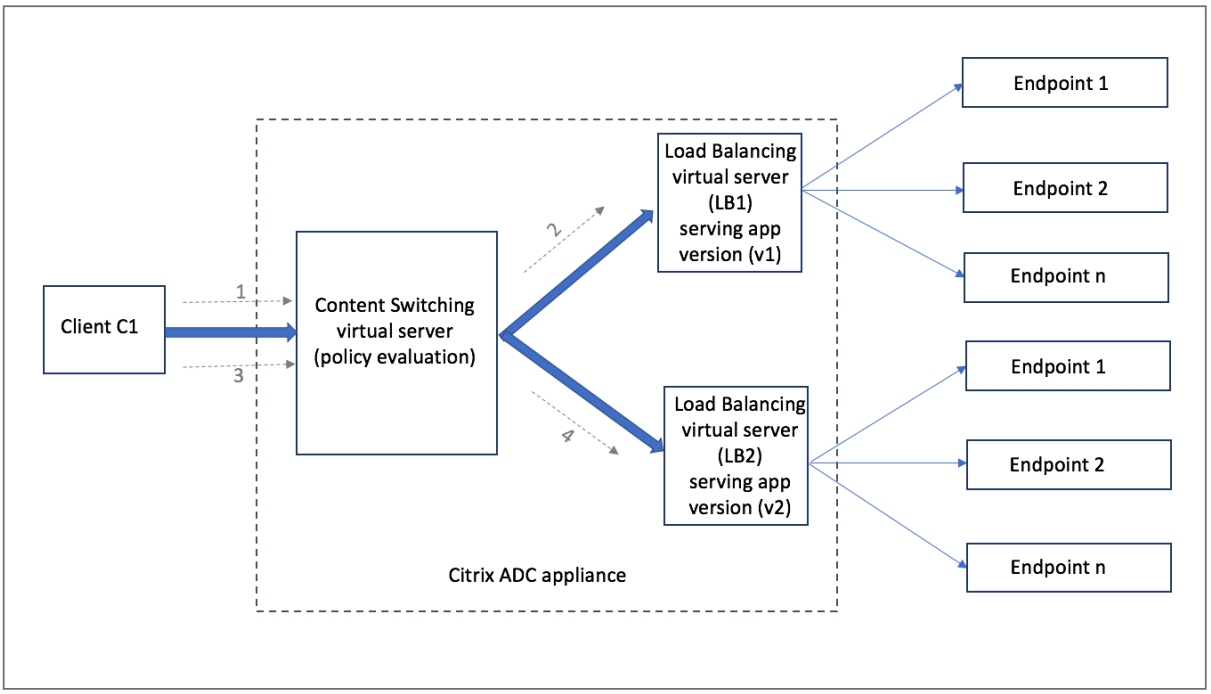
コンテンツスイッチング仮想サーバーの永続性の仕組み

シナリオ **1:** 永続性のないコンテンツスイッチング仮想サーバー

次の例は、永続性のないコンテンツスイッチング仮想サーバーを使用した複数バージョンのアプリケーションのデプロイを示しています。



クライアント C1 がアプリケーションに要求を送信すると、その要求は Citrix ADC アプライアンスのコンテンツスイッチ仮想サーバーに送信されます。コンテンツスイッチ仮想サーバーは、ポリシーを評価し、アプリケーションのバージョン v1 を提供する負荷分散仮想サーバー (LB1) に要求を転送します。

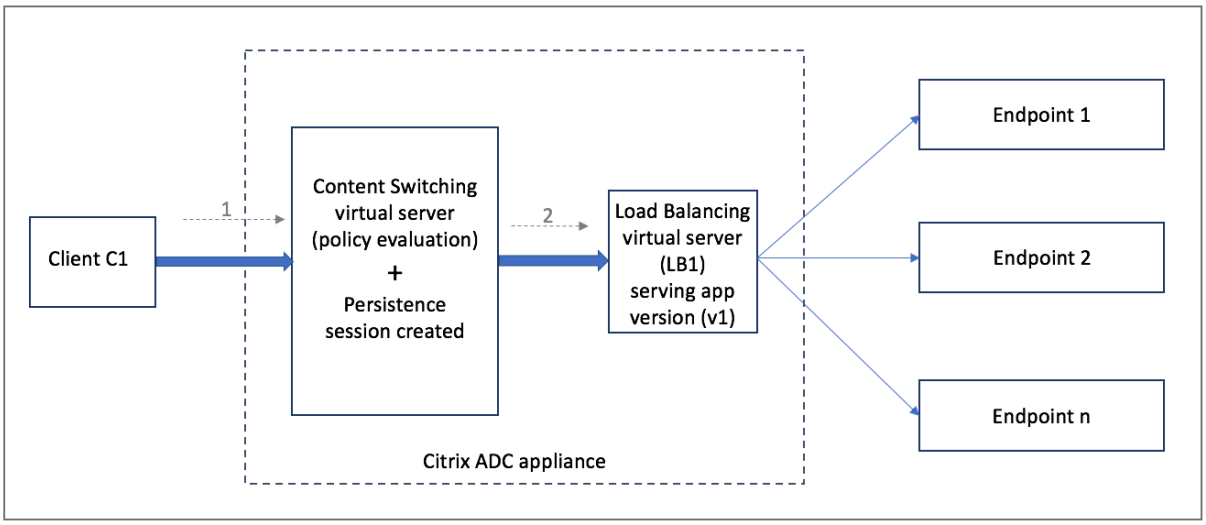


アプリケーションの新しいバージョン v2 がデプロイされ、ユーザーのサブセットに公開する必要があるとします。v2 バージョンを提供する新しい負荷分散仮想サーバー（LB2）は、適切なコンテンツスイッチングポリシーによってコンテンツスイッチング仮想サーバーにバインドされます。

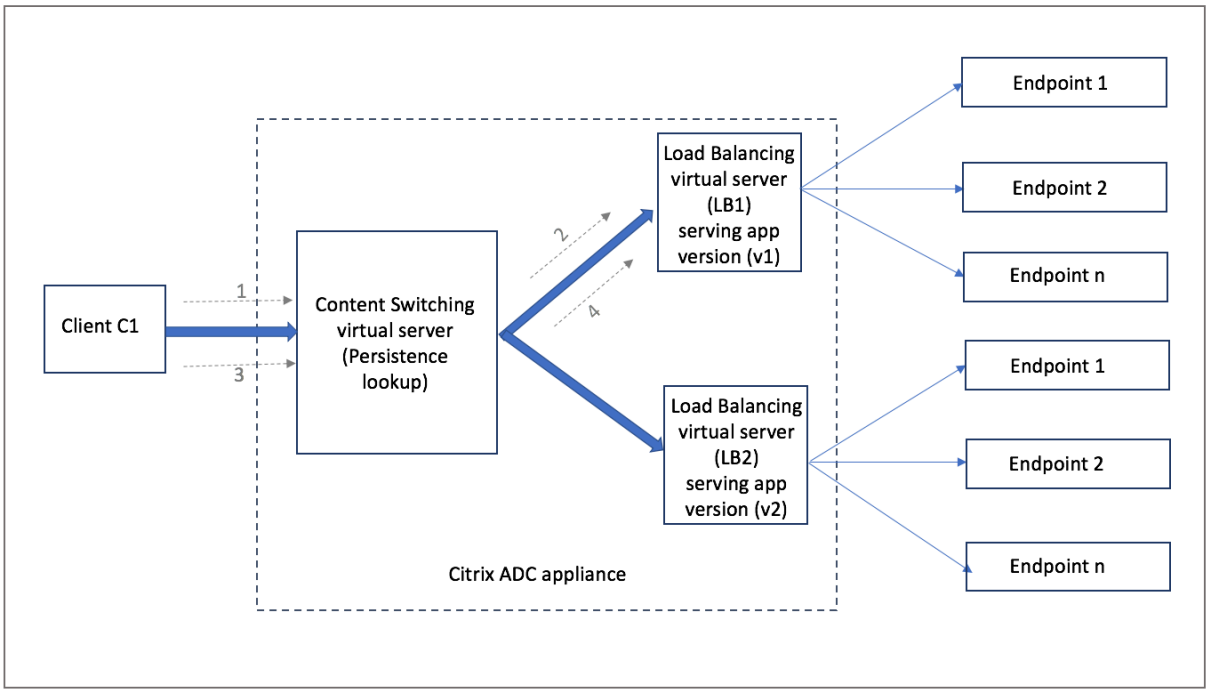
クライアント C1 が新しい要求を送信すると、ポリシーが再度評価され、要求が負荷分散仮想サーバー LB2 に転送されます。したがって、ステートフルアプリケーションのトランザクションは、アプリケーションの複数のバージョンがデプロイされると、失敗します。

シナリオ **2:** 永続性を持つコンテンツスイッチング仮想サーバー

次の例は、永続性を持つコンテンツスイッチング仮想サーバーを使用した複数バージョンのアプリケーションのデプロイを示しています。

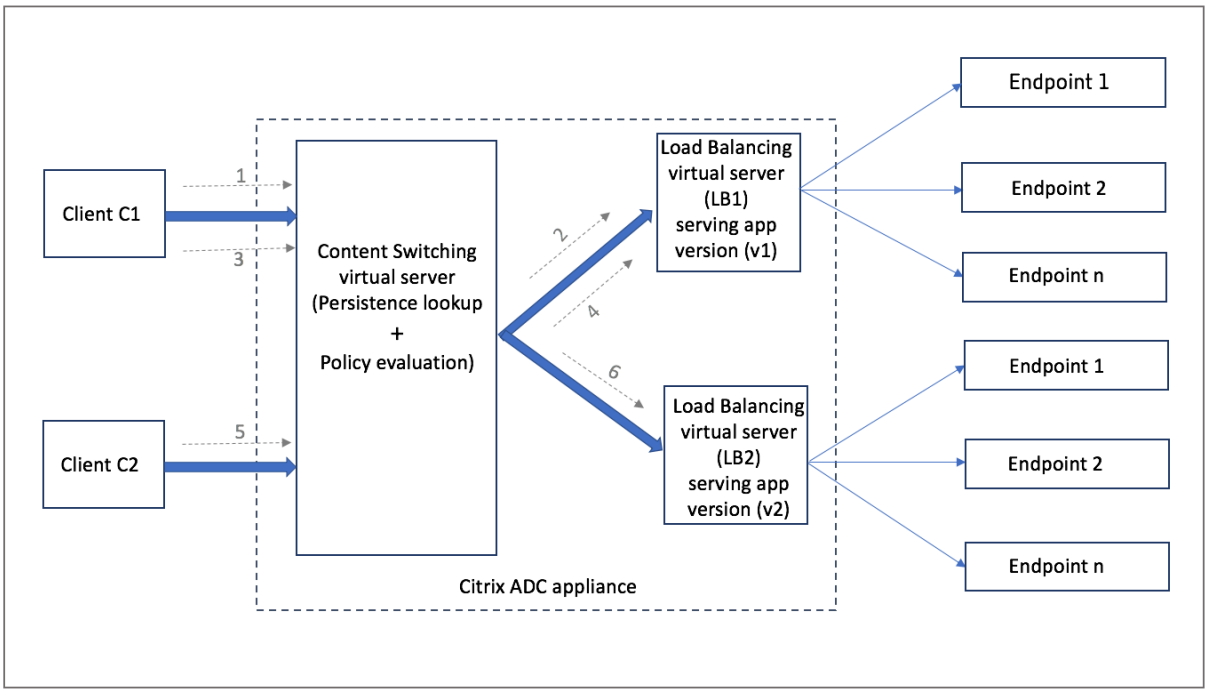


クライアント C1 がアプリケーションに要求を送信すると、その要求は Citrix ADC アプライアンスのコンテンツスイッチ仮想サーバーに送信されます。コンテンツスイッチング仮想サーバーは、ポリシーを評価し、永続セッションエントリを作成して、アプリケーションのバージョン v1 を提供している負荷分散仮想サーバー LB1 に要求を転送します。



同じクライアント C1 がアプリケーションに対して再度要求し、その要求が Citrix ADC アプライアンスのコンテンツスイッチ仮想サーバーに送信されます。永続性セッションのルックアップが行われ、負荷分散仮想サーバー LB1 が既存の永続性セッションから取得され、要求が LB1 に転送されます。このソリューションでは、既存のトランザクショ

ンの破損は発生しません。したがって、アプリケーションのステートフルな性質を維持します。



新しいクライアント C2 を考えてみましょう。このクライアントには既存の永続性セッションが存在しないため、新しい要求 C2 はポリシー評価によって新しいバージョンのアプリケーションに送信されます。その結果、ステートフルネスを損なうことなく、新しいバージョンのアプリケーションのロールアウトが成功します。

永続性サポートにより、お客様は既存のトランザクション（特にステートフルなアプリケーションの場合）に影響を与えることなく、複数のコンテンツまたは異なるバージョンのアプリケーションをシームレスにデプロイできます。絵に固執せずにそれは不可能です。

CLI を使用してコンテンツスイッチ仮想サーバで永続性タイプを構成する

コマンドプロンプトで入力します。

```
set cs vserver <name> -PersistenceType <type> [-timeout <integer>]
<!--NeedCopy-->
```

例:

```
set cs vserver Vserver-CS-1 -persistenceType SOURCEIP -timeout 60
<!--NeedCopy-->
```


GUI を使用したコンテンツスイッチング仮想サーバーのパーシステンスタイプの構成

1. [トラフィック管理] > [コンテンツスイッチング] > [仮想サーバー] に移動し、[追加] をクリックします。
2. [基本設定] で、永続性の詳細を設定します。

トラブルシューティング

October 7, 2021

構成後にコンテンツスイッチング機能が期待どおりに機能しない場合は、一般的なツールを使用して Citrix ADC リソースにアクセスし、問題を診断できます。

コンテンツスイッチングのトラブルシューティングのためのリソース

最適な結果を得るには、次のリソースを使用して、Citrix ADC アプライアンスのコンテンツスイッチングに関する問題のトラブルシューティングを行います。

- 設定ファイル
- 関連する `newnslog` ファイル
- トレースファイル
- 顧客のネットワーク設定のネットワークトポロジ図
- リリースノート、ナレッジセンターの記事、製品ドキュメントなどの Citrix ドキュメント。

上記のリソースに加えて、次のツールはトラブルシューティングを促進します。

- `iehttpheaders` または同様のユーティリティ
- Citrix ADC トレースファイル用にカスタマイズされた Wireshark アプリケーション
- コマンドラインアクセス用の SSH ユーティリティ
- コンソールにアクセスするためのハイパーターミナルユーティリティ

コンテンツスイッチングの問題のトラブルシューティング

最も一般的なコンテンツスイッチングの問題には、コンテンツスイッチング機能がまったく動作しない、または断続的にしか動作しない、および Service Unavailable 応答が含まれます。

- 問題

  コンテンツスイッチング機能が機能していません。

  解像度

  次のように設定を確認します。

  - アプライアンスにコンテンツスイッチングのライセンスが付与されていることを確認します。

- 機能が有効になっていることを確認します。
- 構成ファイルから、有効なコンテンツスイッチングポリシーが、負荷分散仮想サーバーに正しくバインドされていることを確認します。

- 問題

  クライアントが「503-サービス利用不可」という応答を受信します。

  解像度

  - URL とポリシーバインディングを確認します。構成したポリシーのいずれも評価されず、デフォルトの負荷分散仮想サーバーが定義されてコンテンツスイッチ仮想サーバーにバインドされていない場合、クライアントは 503 応答を受信します。
  - 構成から、ポリシーと URL がクライアントによってアクセスされていることを確認します。
  - リクエストのタイプごとに、それぞれのポリシーが評価されることを確認します。ポリシーが評価されない場合は、ポリシー式をチェックし、必要に応じて更新します。
  - URL および HTTP 要求および応答ヘッダーを確認します。これを行うには、`HTTPHeader` トレースを記録し、必要に応じて、アプライアンスとクライアントでパケットトレースを記録します。

- 問題

  断続的に、コンテンツの切り替え機能が期待どおりに動作していません。

  解像度

  - セットアップのネットワークトポロジ図（利用可能な場合）を調べて、クライアントとサーバーの間にインストールされているさまざまなデバイスを理解します。
  - 設定およびポリシーバインディングを確認します。ポリシー式の URL が、クライアント要求の URL と一致していることを確認します。
  - ポリシーに適切な優先順位が割り当てられていることを確認します。ポリシーに不適切な優先順位または優先順位が割り当てられていると、問題が発生する可能性があります。
  - 次のコマンドを実行して、コマンドの出力にあるポリシー選択カウンターのバインディングと値を確認します。

    ```
    show cs vserver \<CS VServer\>

    show cs policy \<CS Policy\>

    stat cs vserver \<CS VServer\>
    ```

  - `iehttpheaders` または同様のユーティリティを使用して、要求または応答の HTTP ヘッダーが問題へのポインターを提供するかどうかを判別します。
  - リリースノートとナレッジセンターの記事を確認してください。
  - それでも問題が解決しない場合は、Citrix テクニカルサポートに連絡して適切なデータを入手し、詳細な調査を行ってください。

DataStream

October 7, 2021

Citrix ADC DataStream 機能は、送信される SQL クエリに基づいて要求を分散することにより、データベース層での要求切り替えのためのインテリジェントなメカニズムを提供します。

Citrix ADC アプライアンスをデータベースサーバーの前に展開すると、アプリケーションサーバーと Web サーバーからのトラフィックの最適な分散が保証されます。管理者は、SQL クエリの情報に従って、データベース名、ユーザー名、文字セット、およびパケットサイズに基づいてトラフィックをセグメント化できます。

負荷分散アルゴリズムに基づいてリクエストを切り替えるように負荷分散を構成できます。または、SQL クエリパラメータに基づいて決定を下すようにコンテンツの切り替えを構成することで、切り替え基準を詳しく説明することもできます。さらに、データベースサーバーの状態を追跡するようにモニターを構成できます。

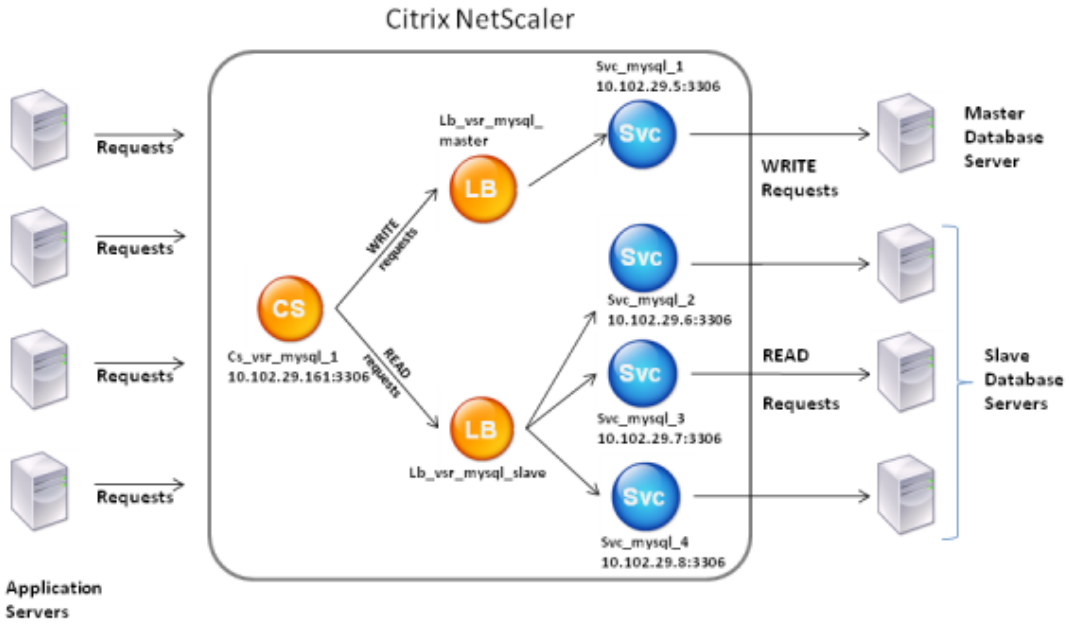
> 注
>
> Citrix ADC DataStream は、MySQL および MS SQL データベースでのみサポートされます。サポートされているプロトコルバージョン、文字セット、特殊クエリ、およびトランザクションの詳細については、「DataStream リファレンス」を参照してください。

DataStream のしくみ

DataStream では、ADC アプライアンスはアプリケーションまたは Web サーバーとデータベースサーバーの間にインラインで配置されます。アプライアンスでは、データベース・サーバはサービスによって表されます。

一般的な DataStream デプロイメントは、次の図で説明するエンティティで構成されます。

図 1: DataStream エンティティモデル



この図に示すように、DataStream 構成は次のもので構成できます。

- オプションのコンテンツスイッチング仮想サーバー（CS）。
- 負荷分散仮想サーバー（LB1 および LB2）で構成される負荷分散セットアップ。
- サービス（Svc1、Svc2、Svc3、および Svc4）。
- コンテンツスイッチングポリシー（オプション）。

クライアント（アプリケーションまたは Web サーバー）は、Citrix ADC アプライアンスで構成されたコンテンツスイッチ仮想サーバー（CS）の IP アドレスに要求を送信します。その後、アプライアンスは、アプライアンスに構成されたデータベース・ユーザー認証情報を使用してクライアントを認証します。コンテンツスイッチ仮想サーバー (CS) は、関連付けられたコンテンツスイッチングポリシーを要求に適用します。ポリシーを評価した後、コンテンツスイッチング仮想サーバー（CS）は、要求を適切な負荷分散仮想サーバー（LB1 または LB2）にルーティングします。次に、負荷分散仮想サーバーは、負荷分散アルゴリズムに基づいて、適切なデータベースサーバー（アプライアンス上のサービスで表される）に要求を分散します。Citrix ADC アプライアンスは、同じデータベースユーザー認証情報を使用して、データベースサーバーとの接続を認証します。

コンテンツスイッチング仮想サーバーがアプライアンスで構成されていない場合、クライアント（アプリケーションまたは Web サーバー）は、アプライアンスで構成された負荷分散仮想サーバーに要求を送信します。Citrix ADC アプライアンスは、アプライアンスで構成されたデータベースユーザーの資格情報を使用してクライアントを認証し、同じ資格情報を使用してデータベースサーバーとの接続を認証します。負荷分散仮想サーバーは、負荷分散アルゴリ

ズムに従って要求をデータベースサーバーに分散します。データベース切り替えのための最も効果的な負荷分散アルゴリズムは、最小接続方法です。

DataStream は、接続多重化を使用して、同じサーバー側接続で複数のクライアント側要求を実行できるようにします。次の接続プロパティが考慮されます。

- ユーザー名
- データベース名
- パケットサイズ
- 文字セット

データベースユーザーを構成する

October 7, 2021

データベースでは、接続は常にステートフルです。つまり、接続が確立されると、認証される必要があります。

NetScaler アプライアンスでデータベースのユーザー名とパスワードを構成します。たとえば、データベースにユーザー John が設定されている場合は、ADC にもユーザー John を設定する必要があります。ADC にデータベースのユーザー名とパスワードを追加すると、それらが `nsconfig` ファイルに追加されます。

> 注
>
> 名前では、大文字と小文字が区別されます。

ADC は、これらのユーザー資格情報を使用してクライアントを認証し、データベースサーバーとのサーバー接続を認証します。

CLI を使用してデータベースユーザーを追加します

コマンドプロンプトで、次のように入力します。

```
add db user <username> - password <password>
```

例:

```
1 add db user nsdbuser -password dd260427edf
2 <!--NeedCopy-->
```

GUI を使用してデータベースユーザーを追加します

[システム] > [ユーザ管理] > [データベースユーザ] に移動し、データベースユーザを設定します。

データベースサーバでデータベースユーザのパスワードを変更した場合は、ADC アプライアンスで設定されている対応するユーザのパスワードをリセットする必要があります。

CLI を使用してデータベースユーザーのパスワードをリセットします

コマンドプロンプトで、次のように入力します。

```
set db user <username> -password <password>
<!--NeedCopy-->
```

例:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーのパスワードをリセットします

[システム] > [ユーザー管理] > [データベースユーザー] に移動し、ユーザーを選択して、パスワードの新しい値を入力します。

データベース・ユーザーがデータベース・サーバ上に存在しなくなった場合は、ADC アプライアンスからユーザーを削除できます。ただし、ユーザがデータベースサーバ上に存在し続け、ADC アプライアンスからそのユーザを削除すると、このユーザ名のクライアントからの要求は認証されません。その結果、要求はデータベースサーバーにルーティングされません。

CLI を使用してデータベースユーザーを削除します

コマンドプロンプトで、次のように入力します。

```
rm db user <username>
<!--NeedCopy-->
```

例:

```
rm db user nsdbuser
<!--NeedCopy-->
```


GUI を使用してデータベースユーザーを削除します

[システム**] > [**ユーザ管理**] > [**データベースユーザ] に移動し、ユーザを選択して [削除] をクリックします。

データベースプロファイルを構成する

October 7, 2021

データベースプロファイルは、一度構成されるパラメーターの名前付き集合で、これらの特定のパラメーター設定を必要とする複数の仮想サーバーに適用されます。データベースプロファイルを作成したら、負荷分散またはコンテンツスイッチ仮想サーバーにバインドします。プロファイルは必要な数だけ作成できます。

CLI を使用してデータベースプロファイルを作成します

コマンド・ラインで次のコマンドを入力して、データベース・プロファイルを作成し、構成を確認します。

```
add db dbProfile <name> [-interpretQuery ( YES | NO )] [-stickiness (
    YES | NO )] [-kcdAccount <string>]

show db dbProfile
<!--NeedCopy-->
```

例:

```
> add dbProfile myDBProfile -interpretQuery YES -stickiness YES -
    kcdAccount mykcdaccnt
 Done
> show dbProfile myDBProfile
    Name: myDBProfile
    Interpret Query: YES
    Stickyness: YES
    KCD Account: mykcdaccnt
    Reference count: 0

 Done
>
<!--NeedCopy-->
```


GUI を使用してデータベースプロファイルを作成する

「システム」**>**「プロファイル」に移動し、「データベース・プロファイル」タブでデータベース・プロファイルを設定します。

CLI を使用して、データベースプロファイルを負荷分散またはコンテンツスイッチング仮想サーバーにバインドします

コマンドラインで、次のように入力します。

```
set (lb | cs) vserver <name> -dbProfileName <string>
<!--NeedCopy-->
```

GUI を使用して、データベースプロファイルを負荷分散またはコンテンツスイッチング仮想サーバーにバインドします

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] または [トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [プロファイル] を選択し、[**DB** プロファイル] リストで、仮想サーバーにバインドするプロファイルを選択します。プロファイルを作成するには、プラスをクリックします (+).

DataStream の負荷分散を構成します

October 7, 2021

負荷分散設定を設定する前に、負荷分散機能を有効にする必要があります。次に、負荷分散グループのデータベースサーバーごとに少なくとも 1 つのサービスを作成します。サービスを構成したら、負荷分散仮想サーバーを作成し、サービスを仮想サーバーにバインドする準備が整いました。

> 注:
>
> データベースの場合、負荷分散は、同種のデータベースサーバー (まったく同じデータベースを含むデータベースサーバー) でのみ実行できます。異なるサーバー上の一意のデータベースを含む構成では、コンテンツの切り替えを使用する必要があります。一部のデータベースサーバーが同一のコンテンツをホストしている場合は、それらのサーバーでのみ負荷分散を使用できます。その後、コンテンツスイッチングポリシーを使用して、それらのデータベースの負荷分散を管理する負荷分散仮想サーバーに要求を送信できます。
>
> Citrix ADC アプライアンスは、現在、データベースセッション中にデータベース名とログイン情報を保存しま

> す。データベースに対してクエリが実行されると、その情報を使用して特定のデータベースサーバーに接続します。

DataStream に固有のパラメータ値

- プロトコル

  仮想サーバーとサービスを構成するときは、MySQL データベースには MYSQL プロトコルタイプを使用し、MS SQL データベースには MSSQL プロトコルタイプを使用します。MySQL および TDS プロトコルは、SQL クエリを使用してそれぞれのデータベースサーバーと通信するためにクライアントによって使用されます。MySQL プロトコルの詳細については、http://dev.mysql.com/doc/internals/en/client-server-protocol.htmlを参照してください TDS プロトコルの詳細については、http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspxを参照してください。

- ポート

  仮想サーバがクライアント接続をリッスンするポート。MySQL データベースサーバーにはポート 3306 を使用します。

- 方法

  負荷分散を改善し、サーバーの負荷を軽減するには、最小接続方式を使用することをお勧めします。ただし、ラウンドロビン、最小応答時間、送信元 IP ハッシュ、送信元 IP 宛先 IP ハッシュ、最小帯域幅、最小パケット、送信元 IP 送信元ポートハッシュなどの他の方法もサポートされています。

  注：URL ハッシュメソッドは、DataStream のためにサポートされていません。

- MS SQLServer のバージョン

  Microsoft SQL Server を使用していて、一部のクライアントが Microsoft SQL Server 製品とは異なるバージョンを実行していると予想される場合は、負荷分散仮想サーバーのサーバーバージョンパラメーターを設定します。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。サーバーバージョンパラメーターの設定の詳細については、「MySQL および Microsoft SQL Server のバージョン設定を構成する」を参照してください。

- MySQL サーバーバージョン

  MySQL サーバーを使用していて、一部のクライアントが MySQL サーバー製品とは異なるバージョンを実行していると予想される場合は、負荷分散仮想サーバーのサーバーバージョンパラメーターを設定します。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。サーバーバージョンパラメーターの設定の詳細については、「MySQL および Microsoft SQL Server のバージョン設定を構成する」を参照してください。

DataStream のコンテンツスイッチングを構成する

October 7, 2021

データベース名、ユーザー名、文字セット、およびパケットサイズに基づいて、SQL クエリの情報に従ってトラフィックをセグメント化できます。

デフォルトの構文式を使用してコンテンツスイッチングポリシーを構成し、接続プロパティに基づいてコンテンツを切り替えることができます。たとえば、ユーザー名とデータベース名、コマンドパラメーター、サーバーを選択するための SQL クエリなどです。

デフォルトの構文式では、MYSQL および MS SQL データベースサーバに関連付けられたトラフィックが評価されます。デフォルトの構文ポリシーで要求ベースの式を使用して、コンテンツスイッチング仮想サーバーのバインドポイントで要求切り替えの決定を行います。応答ベースの式（MYSQL.RES で始まる式）を使用して、ユーザーが構成したヘルスモニターに対するサーバーの応答を評価します。

デフォルトの構文式の詳細については、デフォルトの構文式:DataStreamを参照してください。

> 注:
>
> データベースの場合、負荷分散は、同種のデータベースサーバー (まったく同じデータベースを含むデータベースサーバー) でのみ実行できます。異なるサーバー上の一意のデータベースを含む構成では、コンテンツの切り替えを使用する必要があります。一部のデータベースサーバーが同一のコンテンツをホストしている場合は、それらのサーバーでのみ負荷分散を使用できます。その後、コンテンツスイッチングポリシーを使用して、それらのデータベースの負荷分散を管理する負荷分散仮想サーバーに要求を送信できます。
>
> Citrix ADC アプライアンスは、現在、データベースセッション中にデータベース名とログイン情報を保存します。データベースに対してクエリが実行されると、その情報を使用して特定のデータベースサーバーに接続します。

DataStream に固有のパラメータ値

- プロトコル

  仮想サーバーとサービスを構成するときは、MySQL データベースには MYSQL プロトコルタイプを使用し、MS SQL データベースには MSSQL プロトコルタイプを使用します。MySQL および TDS プロトコルは、SQL クエリを使用してそれぞれのデータベースサーバーと通信するためにクライアントによって使用されます。MySQL プロトコルの詳細については、http://dev.mysql.com/doc/internals/en/client-server-protocol.htmlを参照してください TDS プロトコルの詳細については、http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspxを参照してください。

- ポート

  仮想サーバがクライアント接続をリッスンするポート。MySQL データベースサーバーにはポート 3306 を使用します。

- MS SQL サーバーのバージョン

  Microsoft SQL Server を使用していて、一部のクライアントが Microsoft SQL Server 製品とは異なるバージョンを実行していると予想される場合は、コンテンツスイッチング仮想サーバーのサーバーバージョンパラメーターを設定します。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。サーバーバージョンパラメーターの設定の詳細については、「Microsoft SQL Server のバージョン設定の構成」を参照してください。

DataStream のモニターを構成します

October 7, 2021

負荷分散された各データベースサーバーの状態をリアルタイムで追跡するには、モニターを各サービスにバインドする必要があります。モニターは、定期的なプローブをサービスに送信することによってサービスをテストするように構成されています。これは、ヘルスチェックの実行と呼ばれることもあります。モニターがプローブへのタイムリーな応答を受信すると、サービスを UP としてマークします。指定された数のプローブに対するタイムリーな応答を受信しない場合、サービスは DOWN としてマークされます。

DataStream の場合、組み込みのモニター MYSQL-ECV および MSSQL-ECV を使用する必要があります。このモニターを使用すると、SQL 要求を送信し、文字列の応答を解析できます。

DataStream のモニターを構成する前に、データベースユーザーの資格情報を NetScaler アプライアンスに追加する必要があります。モニタの構成の詳細については、「負荷分散セットアップでのモニタの設定」を参照してください。

モニターを作成すると、データベースサーバーとの TCP 接続が確立され、モニターの作成時に指定されたユーザー名を使用して接続が認証されます。その後、データベースサーバーに対する SQL クエリを実行し、サーバー応答を評価して、構成済みのルールに一致するかどうかを確認できます。

以下の例は、MYSQL サーバー用です。

例:

次の例では、エラーメッセージの値を評価して、サーバーの状態を判別します。

```
1 add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from
2 table2;" -evalrule "mysql.res.error.message.contains("Invalid
3 User")"-database "NS" -userName "user1"
4 <!--NeedCopy-->
```

次の例では、応答のロー数を評価して、サーバーの状態を判断します。


```
add lb monitor lb_mon4 MYSQL-ECV -sqlQuery ”select * from
table4;” -evalrule ”mysql.res.atleast_rows_count(7)” -database ”NS” -
    userName ”user2”
<!--NeedCopy-->
```

次の例では、特定のカラムの値を評価して、サーバーの状態を判断します。

```
add lb monitor lb_mon3 MYSQL-ECV
-sqlQuery ”select * from ABC;” -evalrule ”mysql.res.row(1).double_elem
    (2) == 345.12”
-database ”NS” -userName ”user3”
<!--NeedCopy-->
```

次の例は、MSSQL サーバー用です。

例:

次の例では、エラーメッセージの値を評価して、サーバーの状態を判別します。

```
add lb monitor lb_mon1 MSSQL-ECV -sqlQuery ”select * from
table2;” -evalrule ”mssql.res.error.message.contains(”Invalid
User”)”-database ”NS” -userName ”user1”
<!--NeedCopy-->
```

次の例では、応答のロー数を評価して、サーバーの状態を判断します。

```
add lb monitor lb_mon4 MSSQL-ECV -sqlQuery ”select * from
table4;” -evalrule ”mssql.res.atleast_rows_count(7)” -database ”NS” -
    userName ”user2”
<!--NeedCopy-->
```

次の例では、特定のカラムの値を評価して、サーバーの状態を判断します。

```
add lb monitor lb_mon3 MSSQL-ECV
-sqlQuery ”select * from ABC;” -evalrule ”mssql.res.row(1).double_elem
    (2) == 345.12”
-database ”NS” -userName ”user3”
<!--NeedCopy-->
```


ユースケース 1: プライマリ/セカンダリデータベースアーキテクチャの DataStream を構成する

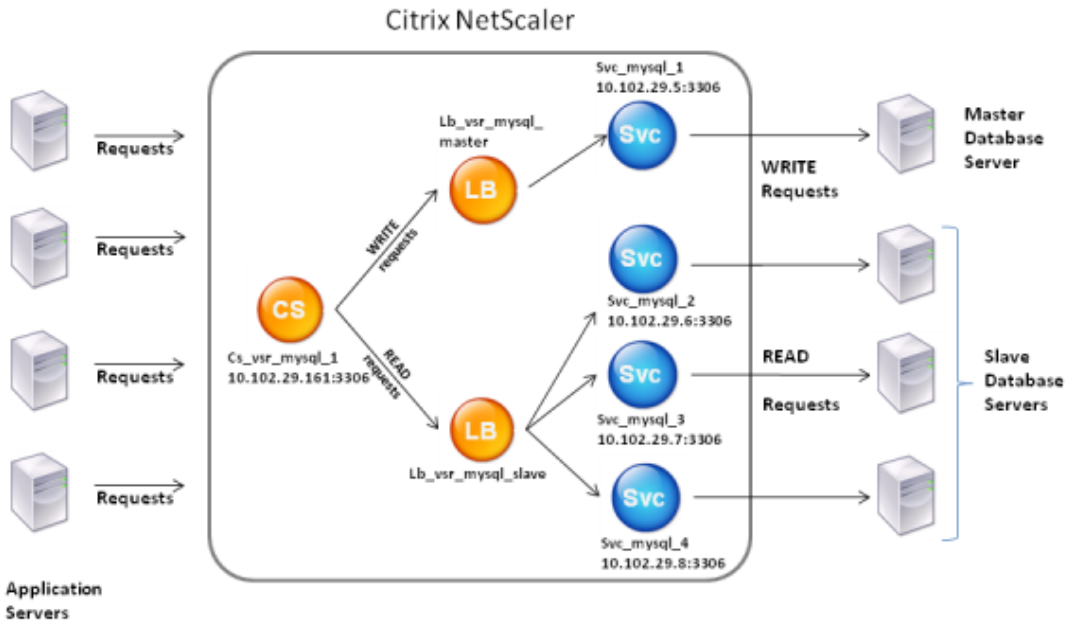
April 25, 2022

一般的に使用される導入シナリオは、プライマリデータベースがすべての情報をセカンダリデータベースにレプリケートするプライマリ/セカンダリデータベースアーキテクチャです。

プライマリ/セカンダリデータベースアーキテクチャでは、すべての WRITE 要求をプライマリデータベースに送信し、すべての READ 要求をセカンダリデータベースに送信できます。

次の図は、アプライアンスに設定する必要のあるエンティティとパラメーターの値を示しています。

図 1: プライマリ/セカンダリデータベースセットアップ用の DataStream エンティティモデル



このシナリオ例では、プライマリデータベースを表すサービス (SVC_MySQL_1) が作成され、負荷分散仮想サーバー (lb_vsr_mysql_Primary) にバインドされます。3 つのセカンダリデータベースを表すためにさらに 3 つのサービス (SVC_MySQL_2、svc_mysql_3、および svc_mysql_4) が作成され、それらは別の負荷分散仮想サーバー (lb_vsr_mysql_Secondary) にバインドされます。

コンテンツスイッチ仮想サーバー (cs_vsr_MySQL_1) は、すべての WRITE 要求を負荷分散仮想サーバー

lb_vsr_mysql_Primary に送信するように、関連付けられたポリシーを使用して構成されます。すべての READ 要求は、負荷分散仮想サーバー lb_vsr_mysql_secondary に送信されます。

要求がコンテンツスイッチ仮想サーバーに到達すると、仮想サーバーはその要求に対して関連するコンテンツスイッチポリシーを適用します。ポリシーを評価すると、コンテンツスイッチ仮想サーバーは要求を適切な負荷分散仮想サーバーにルーティングし、要求は適切なサービスに送信されます。

次の表に、Citrix ADC アプライアンスで構成されているエンティティとポリシーの名前と値を示します。

エンティティの種類	名前	IP アドレス	プロトコル	ポート	式
Services	Svc_mysql_1	198.51.100.5	MYSQL	3306	-
	Svc_mysql_2	198.51.100.6	MYSQL	3306	-
	Svc_mysql_3	198.51.100.7	MYSQL	3306	-
	Svc_mysql_4	198.51.100.8	MYSQL	3306	-
監視	lb_mon1	-	MYSQL-ECV	-	mysql.res.atleast_rows_cou (1)
仮想サーバの負荷分散	lb_vsr_mysql_primary	198.51.100.201	MYSQL	3306	-
	lb_vsr_mysql_s	198.51.100.202	MYSQL	3306	-
コンテンツスイッチ仮想サーバー	Cs_vsr_mysql_1	198.51.100.161	MYSQL	3306	-
コンテンツスイッチポリシー	Cs_select	-	-	-	`MYSQL.REQ.QUERY.COMMAND.contains("select")`

表 1. エンティティとポリシーの名前と値

コマンドラインインターフェイスを使用して **DataStream** をプライマリ/セカンダリデータベース設定用に設定するには

コマンドプロンプトで、次のように入力します。

```
1 add db user user1 -password user1
```



```
2
3  add service Svc_mysql_1 198.51.100.5 mysql 3306
4
5  add service Svc_mysql_2 198.51.100.6 mysql 3306
6
7  add service Svc_mysql_3 198.51.100.7 mysql 3306
8
9  add service Svc_mysql_4 198.51.100.8 mysql 3306
10
11 add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from table1;" -
       evalrule "mysql.res.atleast_rows_count(1)" -database "NS" -userName
        "user1"
12
13 add lb vserver Lb_vsr_mysql_primary mysql 198.51.100.201 3306
14
15 add lb vserver Lb_vsr_mysql_secondary mysql 198.51.100.202 3306
16
17 bind lb vserver Lb_vsr_mysql_primary svc_mysql_1
18
19 bind lb vserver Lb_vsr_mysql_secondary svc_mysql_2
20
21 bind lb vserver Lb_vsr_mysql_secondary svc_mysql_3
22
23 bind lb vserver Lb_vsr_mysql_secondary svc_mysql_4
24
25 add cs vserver Cs_vsr_mysql_1 mysql 198.51.100.161 3306
26
27 add cs policy Cs_select  -rule "MYSQL.REQ.QUERY.COMMAND.contains("
       select")"
28
29 bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_primary
30
31 bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_secondary  -policy
       Cs_select  -priority 10
32
33 bind service Svc_mysql_1 -monitorName lb_mon1
34
35 bind service Svc_mysql_2 -monitorName lb_mon1
36
37 bind service Svc_mysql_3 -monitorName lb_mon1
38
39 bind service Svc_mysql_4 -monitorName lb_mon1
40 <!--NeedCopy-->
```


使用事例 **2: DataStream** の負荷分散のトークン方式を構成します

October 7, 2021

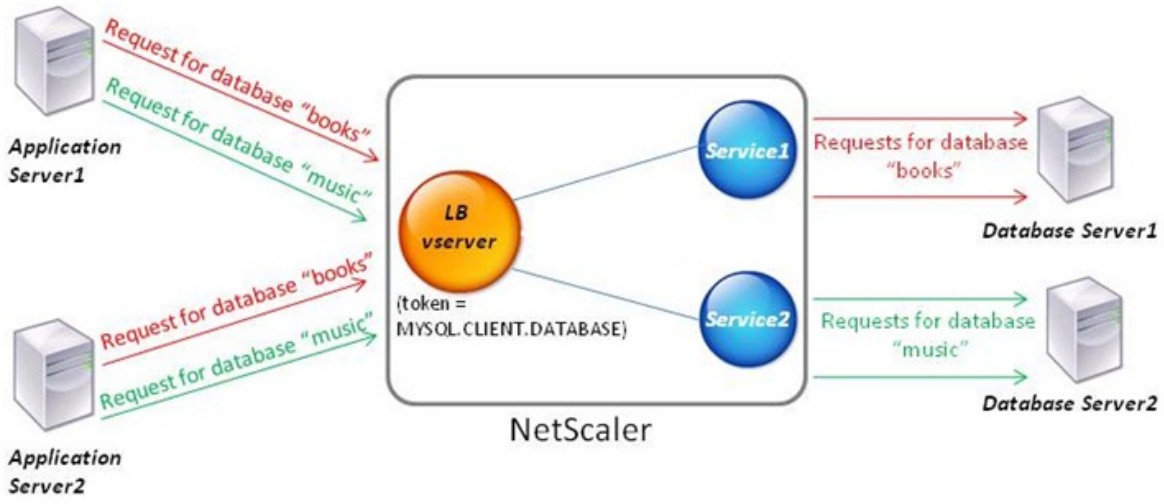
DataStream の負荷分散のトークン方式を構成して、クライアント (アプリケーションまたは Web サーバー) 要求から抽出されたトークンの値に基づいてデータベースサーバーを選択できます。これらのトークンは、SQL 式を使用して定義されます。同じトークンを持つ後続の要求では、Citrix ADC アプライアンスは最初の要求を処理したのと同じデータベースサーバーに要求を送信します。同じトークンを持つ要求は、最大接続制限に達するか、セッションエントリが期限切れになるまで、同じデータベースサーバーに送信されます。

次のサンプル SQL 式を使用して、トークンを定義できます。

MySQL	MS SQL
MYSQL.REQ.QUERY.TEXT	MSSQL.REQ.QUERY.TEXT
MYSQL.REQ.QUERY.TEXT(n)	MSSQL REQ.QUERY.TEXT(n)
MYSQL.REQ.QUERY.COMMAND	MSSQL.REQ.QUERY.COMMAND
MYSQL.CLIENT.USER	MSSQL.CLIENT.USER
MYSQL.CLIENT.DATABASE	MSSQL.CLIENT.DATABASE
MYSQL.CLIENT.CAPABILITIES	

次の例は、負荷分散のトークン方式を構成するときの Citrix ADC DataStream 機能の動作を示しています。

図 1: DataStream と負荷分散のトークン方式



この例では、トークンはデータベースの名前です。トークンブックを含む要求がデータベース Server1 に送信され、

トークンミュージックを含む要求がデータベース Server2 に送信されます。トークン・ブックを含む後続のすべての要求はデータベース・サーバー 1 に送信され、トークン・ミュージックを含む要求はデータベース・サーバー 2 に送信されます。この設定により、データベース・サーバとの擬似永続性が提供されます。

CLI を使用してこの例を構成します

コマンドプロンプトで入力します。

```
add service Service1 192.0.2.9 MYSQL 3306

add service Service2 192.0.2.11 MYSQL 3306

add lb vserver token_lb_vserver MYSQL 192.0.2.15 3306 -lbmethod token -
    rule MYSQL.CLIENT.DATABASE

bind lb vserver token_lb_vserver Service1

bind lb vserver token_lb_vserver Service2
<!--NeedCopy-->
```

GUI を使用してこの例を構成します

1. **[**トラフィック管理**] > [**ロードバランシング**] > [**仮想サーバー**]** に移動し、仮想サーバーを設定し、プロトコルを **MYSQL** として指定します。
2. 「サービス」セクションをクリックし、プロトコルを MYSQL として指定して 2 つのサービスを構成します。これらのサービスを仮想サーバにバインドします。
3. 「詳細設定」で「メソッド」をクリックし、「負荷分散メソッド」リストで「トークン」を選択して、式を **MYSQL.CLIENT.DATABASE** として指定します。

使用事例 **3:** 透過モードで **MSSQL** トランザクションをログに記録する

October 7, 2021

Citrix ADC アプライアンスは、MSSQL クライアントとサーバー間で透過的に動作し、すべてのクライアントとサーバーのトランザクションの詳細のみをログに記録または分析するように構成できます。透過モードは、Citrix ADC アプライアンスが MSSQL 要求のみをサーバーに転送し、サーバーの応答をクライアントに中継するように設計されています。要求と応答がアプライアンスを通過すると、アプライアンスは、監査ログまたは AppFlow 構成で指定されたとおりにそれらから収集された情報をログに記録するか、Action Analytics 構成で指定された統計を収集します。データベースユーザーをアプライアンスに追加する必要はありません。

透過モードで動作している場合、Citrix ADC アプライアンスは要求に対して負荷分散、コンテンツスイッチング、接続多重化を実行しません。ただし、サーバーの代わりにクライアントのログイン前パケットに応答するため、ログイン前ハンドシェイク中に暗号化が合意されないようにします。ログインパケットとその後のパケットは、サーバに転送されます。

構成タスクの要約

トランスペアレントモードで MSSQL 要求をログに記録または分析するには、次の操作を行う必要があります。

- Citrix ADC アプライアンスをクライアントとサーバーの両方のデフォルト Gateway として構成します。
- Citrix ADC アプライアンスで次のいずれかの操作を行います。
  - 送信元 **IP** アドレス（**USIP**）オプションをグローバルに構成します。ワイルドカード IP アドレスと、MSSQL サーバーが要求をリッスンするポート番号（ポート固有のワイルドカード仮想サーバー）を使用して、負荷分散仮想サーバーを作成します。次に、USIP オプションをグローバルに有効にします。ポート固有のワイルドカード仮想サーバーを構成する場合、アプライアンスで MSSQL サービスを作成する必要はありません。アプライアンスは、クライアント要求の宛先 IP アドレスに基づいてサービスを検出します。
  - **USIP** オプションをグローバルに構成しない場合：それぞれで USIP オプションを有効にして MSSQL サービスを作成します。サービスを構成する場合、ポート固有のワイルドカード仮想サーバーを作成する必要はありません。
- 監査ロギング、AppFlow またはアクション分析を構成して、リクエストに関する統計をログに記録または収集します。仮想サーバーを構成する場合は、ポリシーを仮想サーバーまたはグローバルバインドポイントにバインドできます。仮想サーバーを構成しない場合は、ポリシーをグローバルバインドポイントにのみバインドできます。

ワイルドカード仮想サーバーを使用して透過モードを構成します

トランスペアレントモードを設定するには、ポート固有のワイルドカード仮想サーバを設定し、Use Source IP（USIP）モードをグローバルに有効にします。クライアントがデフォルト Gateway（Citrix ADC アプライアンス）に、宛先 IP アドレスヘッダーに MSSQL サーバーの IP アドレスを含む要求を送信すると、アプライアンスは宛先 IP アドレスが使用可能かどうかをチェックします。IP アドレスが使用可能な場合、仮想サーバは要求をサーバに転送します。それ以外の場合は、要求がドロップされます。

CLI を使用してワイルドカード仮想サーバーを作成します

コマンドプロンプトで次のコマンドを入力して、ワイルドカード仮想サーバーを作成し、構成を確認します。

```
add lb vserver <name> <serviceType> <IPAddress> <port>

show lb vserver <name>
```



```
<!--NeedCopy-->
```

例:

```
> add lb vserver wildcardLbVs MSSQL * 1433
 Done
> show lb vserver wildcardLbVs
    wildcardLbVs (*:1433) - MSSQL   Type: ADDRESS
    State: UP
    . . .

 Done
>
<!--NeedCopy-->
```

GUI を使用してワイルドカード仮想サーバーを作成します

[トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動し、仮想サーバーを作成します。プロトコルとして MSSQL を指定し、IP アドレスとして * を指定します。

CLI を使用して、ソース **IP**（**USIP**）モードをグローバルに有効にする

コマンドプロンプトで次のコマンドを入力して USIP モードをグローバルに有効にし、構成を確認します。

```
enable ns mode USIP

show ns mode
<!--NeedCopy-->
```

例:

```
> enable ns mode USIP
 Done
> show ns mode

    Mode                              Acronym
       Status
    -------                           -------
       ------
```



```
                . . .
 3) Use Source IP                  USIP                          ON
                . . .
 Done
>
<!--NeedCopy-->
```

GUI を使用して **USIP** モードをグローバルに有効にする

1. [システム] **>** [設定] に移動し、[モードと機能] で [モードの構成] を選択します。
2. [ソース **IP** を使用] を選択します。

MSSQL サービスを使用して透過モードを構成する

透過モードを設定するには、MSSQL サービスを構成し、各サービスで USIP を有効にします。クライアントがデフォルト Gateway（Citrix ADC アプライアンス）に、宛先 IP アドレスヘッダーに MSSQL サーバーの IP アドレスを含む要求を送信すると、アプライアンスは宛先サーバーに要求を転送します。

MSSQL サービスを作成し、**CLI** を使用してサービスで **USIP** モードを有効にします

コマンドプロンプトで次のコマンドを入力して、USIP を有効にして MSSQL サービスを作成し、構成を確認します。

```
add service <name> (<IP> | <serverName>) <serviceType> <port> -usip YES
    `

show service <name>
<!--NeedCopy-->
```

例

```
> add service myDBservice 192.0.2.0 MSSQL 1433 -usip YES
 Done
> show service myDBservice
    myDBservice (192.0.2.0:1433) - MSSQL
    State: UP
                . . .
    Use Source IP: YES      Use Proxy Port: YES
                . . .
 Done
```



```
10  >
11  <!--NeedCopy-->
```

GUI を使用して、**USIP** を有効にして **MSSQL** サービスを作成します

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを設定します。
2. プロトコルを **MSSQL** として指定し、[設定] で [ソース **IP** を使用] を選択します。

使用事例 **4:** データベース固有の負荷分散

October 7, 2021

データベースサーバーファームは、サーバーの状態だけでなく、各サーバーのデータベースの可用性にも基づいて負荷分散する必要があります。サービスが稼働している可能性があり、負荷分散デバイスでは稼働状態と表示されますが、要求されたデータベースがそのサービスでは利用できない可能性があります。データベースが使用できないサービスにクエリが転送された場合、要求は提供されません。したがって、負荷分散デバイスは、各サービス上のデータベースの可用性を認識する必要があります。また、負荷分散を決定する際には、データベースが使用可能なサービスのみを考慮する必要があります。

たとえば、データベースサーバー server1、server2、および server3 がデータベース mydatabase1 および mydatabase2 をホストしているとします。mydatabase1 が server2 で使用不能になった場合、負荷分散デバイスはその状態の変化を認識する必要があります。mydatabase1 に対する要求を server1 と server3 に対してのみ負荷分散する必要があります。mydatabase1 が server2 で使用可能になったら、負荷分散の決定に server2 を含める必要があります。同様に、mydatabase2 が server3 で使用不能になった場合、デバイスは mydatabase2 に対する要求を server1 と server2 に対してのみ負荷分散する必要があります。mydatabase2 が使用可能になった場合にのみ、負荷分散の決定に server3 を含める必要があります。この負荷分散の動作は、サーバーファームでホストされているすべてのデータベース間で一貫している必要があります。

Citrix ADC アプライアンスは、サービス上でアクティブなすべてのデータベースのリストを取得することによって、この動作を実装します。アクティブなデータベースのリストを取得するには、アプライアンスは適切な SQL クエリで構成されたモニターを使用します。要求されたデータベースがサービス上で利用できない場合、アプライアンスはサービスを使用可能になるまで負荷分散の決定から除外します。この動作により、クライアントへのサービスが中断されることはありません。

> 注
>
> データベース固有の負荷分散は、MSSQL および MySQL サービスタイプでのみサポートされます。このサポートは、Microsoft SQL Server 2012 年の高可用性展開でも利用できます。

データベース固有の負荷分散を設定するには、以下を設定する必要があります。

- 負荷分散機能を有効にし、タイプが MSSQL または MySQL の負荷分散仮想サーバーを構成します。

- データベースをホストするサービスを構成し、サービスを仮想サーバーにバインドします。モニターでデータベースサーバーにログオンするには、有効なユーザー資格情報が必要です。そのため、各サーバーでデータベースユーザーアカウントを構成し、そのユーザーアカウントを Citrix ADC アプライアンスに追加する必要があります。
- 次に、MSSQL-ECV または MYSQL-ECV モニターを構成し、モニターを各サービスにバインドします。
- 最後に、設定をテストして、意図したとおりに動作していることを確認する必要があります。これらの構成タスクを実行する前に、データベース固有の負荷分散の仕組みを理解しておいてください。

データベース固有の負荷分散のしくみ

データベース固有のロード・バランシングでは、各データベース・サーバにあるすべてのアクティブなデータベースの名前を定期的に照会するモニターを設定します。Citrix ADC アプライアンスは結果を格納し、監視によって取得した情報に基づいてレコードを定期的に更新します。クライアントが特定のデータベースに問い合わせると、アプライアンスは設定された負荷分散方式を使用してサービスを選択し、そのレコードをチェックして、そのサービスでデータベースが使用可能かどうかを判断します。レコードでデータベースが使用できないことが示されている場合、構成された負荷分散方法を使用して次に使用可能なサービスを選択し、チェックを繰り返します。アプライアンスは、データベースがアクティブである最初に使用可能なサービスにクエリーを転送します。

負荷分散を有効にする

負荷分散機能が無効になっている場合、サービスや仮想サーバーなどの負荷分散エンティティを構成できます。エンティティは、この機能を有効にするまで機能しません。

CLI を使用して負荷分散を有効にする

コマンドプロンプトで次のコマンドを入力して、負荷分散を有効にし、構成を確認します。

```
enable ns feature LB

show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
```



```
6           -------                         -------              ------
7    1)     Web Logging                     WL                   OFF
8    2)     Surge Protection                SP                   ON
9    3)     Load Balancing                  LB                   ON
10   .
11   .
12   .
13   24)    NetScaler Push                  push                 OFF
14   Done
15  <!--NeedCopy-->
```

GUI を使用して負荷分散を有効にする

[システム] **>** [設定] に移動し、[基本機能の構成] で [負荷分散] を選択します。

データベース固有の負荷分散のために負荷分散仮想サーバーを構成する

可用性に基づいてデータベースを負荷分散するように仮想サーバーを構成するには、仮想サーバーでデータベース固有の負荷分散パラメーターを有効にします。このパラメータを有効にすると、負荷分散ロジックが変更され、Citrix ADC アプライアンスが選択したサービスに送信された監視プローブの結果を参照してから、そのサービスにクエリが転送されます。

CLI を使用して、データベース固有の負荷分散用に負荷分散仮想サーバーを構成します

コマンドプロンプトで次のコマンドを入力して、データベース固有の負荷分散用に負荷分散仮想サーバーを構成し、構成を確認します。

```
1  add lb vserver <name> <serviceType> <ipAddress> <port> -dbsLb ENABLED
2
3  show lb vserver <name>
4  <!--NeedCopy-->
```

サービスの構成

負荷分散機能を有効にした後、負荷分散設定に含めるアプリケーションサーバーごとに少なくとも 1 つのサービスを作成する必要があります。構成するサービスは、Citrix ADC アプライアンスと負荷分散サーバー間の接続を提供します。各サービスには名前があり、IP アドレス、ポート、および提供されるデータの種類を指定します。

最初にサーバオブジェクトを作成せずにサービスを作成する場合、サービスの IP アドレスもサービスをホストするサーバの名前になります。サーバーを IP アドレスではなく名前で識別する場合は、サーバーオブジェクトを作成し、サービスの作成時に IP アドレスの代わりにサーバー名を指定できます。

データベースユーザーを構成する

データベースでは、接続は常にステートフルです。つまり、接続が確立されると、認証される必要があります。

Citrix ADC でデータベースのユーザー名とパスワードを構成します。たとえば、データベースにユーザー John が設定されている場合は、ADC にもユーザー John を設定する必要があります。ADC に追加されたデータベースのユーザー名とパスワードが `nsconfig` ファイルに追加されます。

> 注
>
> 名前では、大文字と小文字が区別されます。

ADC は、これらのユーザー資格情報を使用してクライアントを認証し、データベースサーバーとのサーバー接続を認証します。

CLI を使用してデータベースユーザーを追加します

コマンドプロンプトで、次のように入力します。

```
1 add db user <username> - password <password>
2 <!--NeedCopy-->
```

例:

```
1 add db user nsdbuser -password dd260427edf
2 <!--NeedCopy-->
```

GUI を使用してデータベースユーザーを追加します

[システム] > [ユーザ管理] > [データベースユーザ] に移動し、データベースユーザを設定します。

データベースサーバーでデータベースユーザーのパスワードを変更した場合は、Citrix ADC アプライアンスで構成されている対応するユーザーのパスワードをリセットする必要があります。

CLI を使用してデータベースユーザーのパスワードをリセットします

コマンドプロンプトで、次のように入力します。


```
set db user <username> -password <password>
<!--NeedCopy-->
```

例:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーのパスワードをリセットします

[システム] > [ユーザー管理] > [データベースユーザー] に移動し、ユーザーを選択して、パスワードの新しい値を入力します。

データベースサーバー上にデータベースユーザーが存在しなくなった場合は、Citrix ADC アプライアンスからユーザーを削除できます。ただし、ユーザがデータベースサーバ上に存在し続け、ADC アプライアンスからそのユーザを削除すると、このユーザ名のクライアントからの要求は認証されません。したがって、ユーザー名はデータベースサーバーにルーティングされません。

CLI を使用してデータベースユーザーを削除します

コマンドプロンプトで、次のように入力します。

```
rm db user <username>
<!--NeedCopy-->
```

例:

```
rm db user nsdbuser
<!--NeedCopy-->
```

GUI を使用してデータベースユーザーを削除します

[システム] > [ユーザ管理] > [データベースユーザ] に移動し、ユーザを選択して [削除] をクリックします。

アクティブなデータベースの名前を取得するようにモニターを構成します

モニタを作成して、データベース・インスタンス上のすべてのアクティブ・データベースのリストを取得できます。モニターは、有効なユーザー資格情報を使用してデータベースサーバーにログオンし、適切な SQL クエリを実行します。使用する必要がある SQL クエリは、SQL サーバーのデプロイメントによって異なります。たとえば、MSSQL データベースミラーリングのセットアップでは、次のクエリを使用して、サーバーインスタンスで使用可能なアクティブなデータベースの一覧を取得できます。

```
select name from sys.databases where state=0
<!--NeedCopy-->
```

MySQL データベース設定では、次のクエリを使用して、サーバーインスタンスで使用可能なアクティブなデータベースのリストを取得できます。

データベースの表示:

また、エラー状態の応答を評価し、エラーがない場合は結果を保存するようにモニターを構成します。応答にエラーが含まれている場合、モニタはサービスを DOWN としてマークします。アプライアンスは、エラーが返されなくなるまで、サービスを負荷分散の決定から除外します。

> 注
>
> データベース固有の負荷分散機能は、MSSQL および MySQL サービスタイプでのみサポートされます。したがって、モニタタイプは MSSQL-ECV または MYSQL-ECV である必要があります。

CLI を使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを構成します

コマンドプロンプトで次のコマンドを入力して、サービスでホストされているすべてのアクティブなデータベースの名前を取得し、構成を確認します。

```
add lb monitor <monitorName> <type> -userName <string> -sqlQuery <text>
    -evalRule <expression> -storedb ENABLED

show lb monitor <monitorName>
<!--NeedCopy-->
```

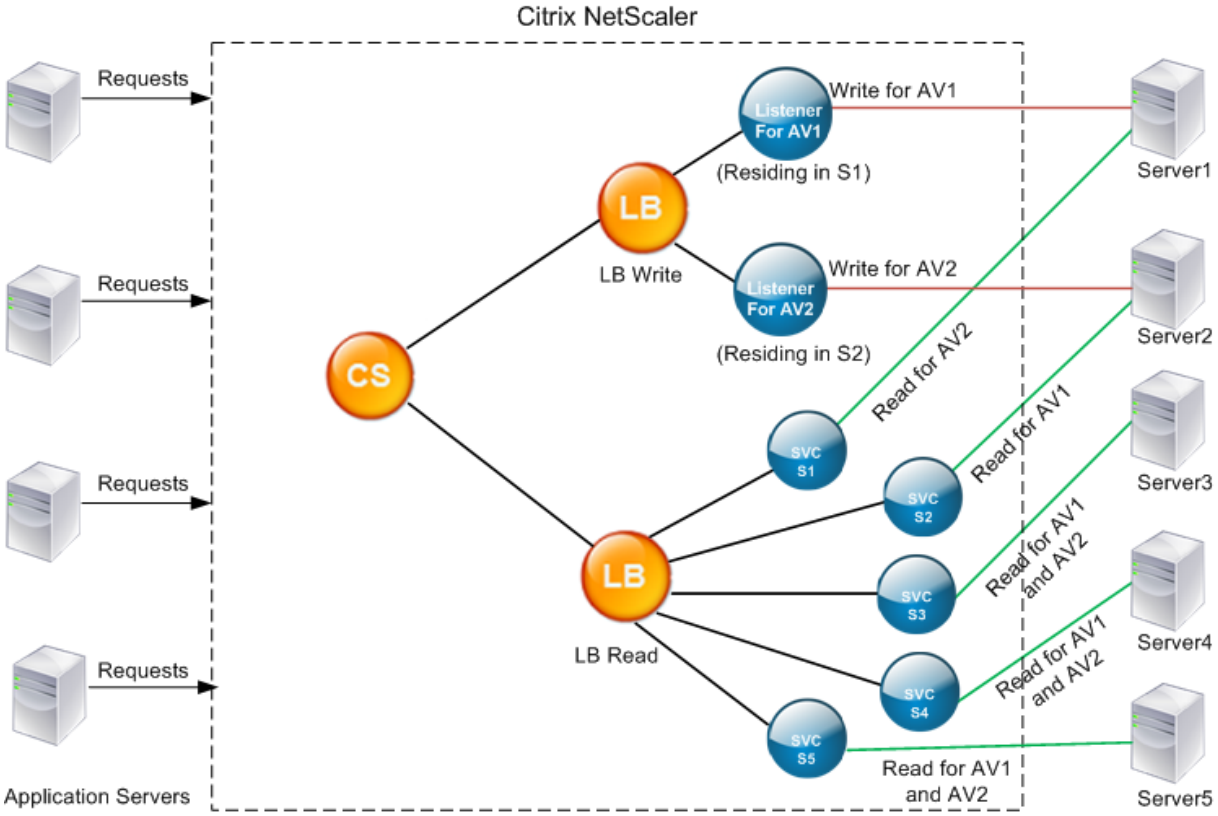
GUI を使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを構成します

1. [トラフィック管理] **>** [負荷分散] **>** [モニター] に移動し、MSSQL-ECV または MYSQL-ECV タイプのモニタを設定します。

2. 「特殊パラメータ」で、ユーザー名、クエリー、およびルールを指定します。たとえば、MSSQL-ECV の場合、クエリは「状態 = 0 の sys.databases から名前を選択」である必要があります)。また、ルールは MSSQL.RES.TYPE.NE (エラー) である必要があります。MYSQL-ECV の場合、クエリは「データベースの表示」でなければならず、ルールは MYSQL.RES.TYPE.NE（エラー）でなければなりません。

MSSQL の可用性グループ展開のサポート

高可用性グループの展開で、データベース固有の負荷分散が構成される次のシナリオを検討します。S1 ～S5 は、ADC アプライアンス上のサービスです。DB1 ～DB4 は、サービス S1 ～S5 で表されるサーバ上のデータベースです。AV1 と AV2 は可用性グループです。各可用性グループには、プライマリデータベースサーバーインスタンスとセカンダリデータベースサーバーインスタンスが 4 つまで含まれます。可用性グループ内のサーバーを表すサービスは、ある可用性グループのプライマリ、別の可用性グループのセカンダリにすることができます。各可用性グループには、異なるデータベースと、サービスである 1 つのリスナーが含まれます。すべての要求は、プライマリ・データベースに存在するリスナー・サービスに到着します。AVI は、データベース DB1 と DB2 が含まれています。AV2 は、データベース DB3 と DB4 が含まれています。L1 と L2 は、それぞれ AV1 と AV2 のリスナーです。S1 は AV1 のプライマリサービスで、S2 は AV2 のプライマリサービスです。



サービス	サービス上のアクティブなデータベースのリスト
S1	DB1, DB2, DB3, DB4
S2	DB3, DB4

サービス	サービス上のアクティブなデータベースのリスト
S3	DB3, DB4
S4	DB1, DB2
S5	DB1, DB2

可用性グループ	データベース	可用性グループのサーバーを表すサービス
AV1	DB1, DB2	S1, S4, S5
AV2	DB3, DB4	S1, S2, S3

クエリは次のようにフローされます。

1. AV1 の READ クエリは、S4 と S5 の間で負荷分散されます。S1 は AV1 のプライマリです。
2. AV1 の WRITE クエリは、L1 に送信されます。
3. AV2 の READ クエリは、S1 と S3 の間で負荷分散されます。S2 は AV2 のプライマリです。
4. AV1 の WRITE クエリは、L2 に送信されます。

構成例

1. 負荷分散とコンテンツスイッチング仮想サーバーを構成します。
   - `add lb vserver lbwrite -dbslb enabled`
   - `add lbvserver lbread MSSQL -dbslb enabled`
   - `add csvserver csv MSSQL 1.1.1.10 1433`
2. 可用性グループごとに 1 つずつ、データベース DB1 から DB4 を表す 5 つのサービス S1 から S5 の 2 つのリスナーサービスを構成します。
   - `add service L1 1.1.1.11 MSSQL 1433`
   - `add service L2 1.1.1.12 MSSQL 1433`
   - `add service s1 1.1.1.13 MSSQL 1433`
   - `add service s2 1.1.1.14 MSSQL 1433`
   - `add service s3 1.1.1.15 MSSQL 1433`
   - `add service s4 1.1.1.16 MSSQL 1433`
   - `add service s5 1.1.1.17 MSSQL 1433`
3. サービスを負荷分散仮想サーバーにバインドします。
   - `bind lbvserver lbwrite L1`
   - `bind lbvserver lbwrite L2`
   - `bind lbvserver lbread s1`

- bind lbvserver lbread s2
- bind lbvserver lbread s3
- bind lbvserver lbread s4
- bind lbvserver lbread s5

4. データベース・ユーザーを構成します。
   - add db user nsdbuser1 -password dd260427edf
   - add db user nsdbuser2 -password ccd1234xyzw
5. リスナーサービスごとに monitor_L1 と monitor_L2 の 2 つのモニターを構成して、その可用性グループ内のアクティブなデータベースの一覧を取得します。モニタ monitor1 を追加して、セカンダリデータベースサーバーインスタンスのデータベースのリストを取得します。
   - add lb monitor monitor_L1 MSSQL-ECV -userName user1 -sqlQuery ” SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_ b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address like '1.1.1.11'”-evalRule ”MSSQL.RES.TYPE.NE(ERROR)” -storedb ENABLED
   - add lb monitor monitor_L2 MSSQL-ECV -userNameuser1 -sqlQuery ” SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replicca b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address like '1.1.1.12'”-evalRule ”MSSQL.RES.TYPE.NE(ERROR)”-storedb ENABLED
   - add lb monitor monitor1 MSSQL-ECV -userNameuser1 -sqlQuery ”SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states b ON a.replica_id=b.replica_id WHERE b.role = 2”-evalRule ”MSSQL. RES.TYPE.NE(ERROR)”-storedb ENABLED
6. 読み取りおよび書き込みポリシーを構成します。
   - add cs policy pol_write -rule ”MSSQL.REQ.QUERY.TEXT.CONTAINS(” insert”)”
   - add cs policy pol_read -rule ”MSSQL.REQ.QUERY.TEXT.CONTAINS(”select ”)”
7. ポリシーをコンテンツスイッチ仮想サーバーにバインドします。
   - bind csvserver csv -targetLBVserver lbwrite -policyName pol_write -priority 11
   - bind csvserver csv -targetLBVserver lbread -policyName pol_read -priority 12
8. モニタをサービスにバインドします。監視をサービス L1 および L2 にバインドして、リスナーである可用性グ

ループのアクティブなデータベースの一覧を取得します。読み取り専用の仮想サーバーにバインドされているすべてのサービスにモニターをバインドします。

- bind service L1 -monitorName monitor_L1
- bind service L2 -monitorName monitor_L2
- bind service s1 -monitorName monitor1
- bind service s2 -monitorName monitor1
- bind service s3 -monitorName monitor1
- bind service s4 -monitorName monitor1
- bind service s5 -monitorName monitor1

MSSQL 仮想サーバーの設定例

データベース固有のロード・バランシング用にロード・バランシング仮想サーバーを構成するには、次の手順を実行します。

```
1  add lb vserver DBSpecificLB1 MSSQL 192.0.2.10 1433 -dbsLb ENABLED
2
3  Done
4
5  show lb vserver DBSpecificLB1
6
7  DBSpecificLB1 (192.0.2.10:1433) - MSSQL Type: ADDRESS
8  . . .
9  DBS_LB: ENABLED
10
11 Done
12 <!--NeedCopy-->
```

サービスを構成するには、次の手順に従います。

add service msservice1 5.5.5.5 MSSQL 1433

コマンドラインを使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを構成するには、次の手順を実行します。

```
1  add lb monitor mssql-monitor1 MSSQL-ECV -userName user1 -sqlQuery ”
       select name from sys.databases where state=0” -evalRule ”MSSQL.RES.
       TYPE.NE(ERROR)” -storedb EN
2
3  Done
4
```



```
show lb monitor mssql-monitor1

1)   Name.......: mssql-monitor1    Type......: MSSQL-ECV

...

Special parameters: Database.....:""

User name.....:"user1"

Query..:select name from sys.databases where state=0 EvalRule...:MSSQL.
   RES.TYPE.NE(ERROR)

Version...:70 STORE_DB...:ENABLED

Done
<!--NeedCopy-->
```

MySQL 仮想サーバーの設定例

データベース固有のロード・バランシング用にロード・バランシング仮想サーバーを構成するには、次の手順を実行します。

```
add lb vserver DBSpecificLB1 MYSQL 192.0.2.10 3306 -dbsLb ENABLED

Done

show lb vserver DBSpecificLB1

DBSpecificLB1 (192.0.2.10:3306) - MYSQL Type: ADDRESS

. . .

DBS_LB: ENABLED

Done
<!--NeedCopy-->
```

サービスを構成するには、次の手順に従います。

```
add service msservice1 5.5.5.5 MYSQL 3306
```



```
<!--NeedCopy-->
```

コマンドラインを使用して、サービスでホストされているすべてのアクティブなデータベースの名前を取得するようにモニターを構成するには、次の手順を実行します。

```
add lb monitor mysql-monitor1 MYSQL-ECV -userName user1 -sqlQuery ”show
    databases” -evalRule ”MYSQL.RES.TYPE.NE(ERROR)” -storedb ENABLED

Done

show lb monitor mysql-monitor1

1)     Name.......: mysql-monitor1  Type......: MYSQL-ECV  State....:
   ENABLED

...

Special parameters: Database.....:””

User name.....:”user1” Query..:show databases

EvalRule...:MYSQL.RES.TYPE.NE(ERROR) STORE_DB...:ENABLED

Done
<!--NeedCopy-->
```

DataStream リファレンス

July 19, 2022

このリファレンスでは、MySQL および TDS プロトコル、データベースのバージョン、認証方法、および DataStream 機能でサポートされる文字セットについて説明します。また、Citrix ADC がトランザクション要求と接続状態を変更する特別なクエリをどのように処理するかについても説明します。

また、Citrix ADC アプライアンスを構成して、DataStream 機能の監査ログメッセージを生成することもできます。

サポートされているデータベースのバージョン、プロトコル、および認証方法

	MySQL Database	MS SQL Database
データベースのバージョン	MySQL データベースのバージョン 4.1, 5.0, 5.1, 5.4, 5.5, 5.6	MS SQL データベースバージョン 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014 (Kerberos 認証サポート)
プロトコル	MySQL プロトコルのバージョン 10。MySQL プロトコルの詳細については、「MySQL クライアント/サーバープロトコル」を参照してください。	Tabular Data Stream（TDS）プロトコルバージョン 7.1 以上。TDS プロトコルの詳細については、表形式データストリームプロトコルを参照してください。
認証方法	MySQL ネイティブ認証がサポートされています。	SQL サーバー認証と Windows 認証 (Kerberos または NTLM) がサポートされています。

文字セット

DataStream 機能は、UTF-8 文字セットのみをサポートします。

要求を送信するときにクライアントが使用する文字セットは、データベースサーバーの応答で使用される文字セットとは異なる場合があります。charset パラメーターは接続の確立中に設定されますが、SQL クエリを送信することでいつでも変更できます。文字セットは接続に関連付けられているため、1 つの文字セットを持つ接続に対する要求は、別の文字セットを持つ接続に多重化できません。

Citrix ADC アプライアンスは、クライアントから送信されたクエリとデータベースサーバーから送信された応答を解析します。

接続に関連付けられた文字セットは、最初のハンドシェイク後に次の 2 つのクエリを使用して変更できます。

```
1 SET NAMES <charset> COLLATION <collation>
2
3 SET CHARACTER SET <charset>
4 <!--NeedCopy-->
```

トランザクション

MySQL では、トランザクションは、接続パラメータ AUTOCOMMIT または BEGIN: COMMIT クエリを使用して識別されます。AUTOCOMMIT パラメータは、最初のハンドシェイク中、またはクエリ SET AUTOCOMMIT を使用して接続が確立された後に設定できます。

Citrix ADC アプライアンスは、各クエリを明示的に解析して、トランザクションの開始と終了を決定します。

MySQL プロトコルでは、応答には、接続がトランザクションであるかどうかを示す 2 つのフラグ（TRANSACTION フラグと AUTOCOMMIT フラグ）が含まれています。

接続がトランザクションの場合は、TRANSACTION フラグが設定されます。または、自動コミットモードがオフの場合、AUTOCOMMIT フラグは設定されません。ADC アプライアンスは応答を解析し、TRANSACTION フラグが設定されているか、AUTOCOMMIT フラグが設定されていない場合、接続の多重化は行われません。これらの条件が満たされなくなると、ADC アプライアンスは接続の多重化を開始します。

> 注
>
> トランザクションは、MS SQL でもサポートされています。

特別なクエリ

SET や PREPARE など、接続の状態を変更し、要求の切り替えを中断する可能性がある特殊なクエリがあります。したがって、これらのクエリは異なる方法で処理する必要があります。

特別なクエリで要求を受信すると、Citrix ADC アプライアンスはクライアントに OK 応答を送信し、接続に要求を保存します。

INSERT や SELECT などの特別でないクエリが、保存されたクエリとともに受信されると、ADC アプライアンスは、保存されたクエリがデータベースサーバーにすでに送信されているサーバー側の接続を探します。このような接続が存在しない場合、ADC アプライアンスは接続を作成し、格納されたクエリーを最初に送信してから、特殊でないクエリーで要求を送信します。

SET、USE db、および INIT_DB 特殊クエリでは、アプライアンスは特殊クエリに対応するサーバー側接続のフィールドを変更します。この変更により、サーバー側の接続の再利用が向上します。

各接続に格納されるクエリは 16 個だけです。

以下は、ADC アプライアンスの動作が変更された特殊なクエリのリストです。

- SET クエリ

  SET SQL クエリは、接続に関連付けられている変数を定義します。これらのクエリーはグローバル変数の定義にも使用されますが、現時点では、ADC アプライアンスはローカル変数とグローバル変数を区別できません。このクエリでは、ADC アプライアンスは「保存と転送」メカニズムを使用します。

- USE<db> クエリ

  このクエリを使用すると、接続に関連付けられたデータベースを変更できます。この場合、ADC アプライアンスは送信された<db> 値を解析し、使用する新しいデータベースを反映するようにサーバー側接続のフィールドを変更します。

- INIT_DB コマンド

  このクエリを使用すると、接続に関連付けられたデータベースを変更できます。この場合、ADC アプライアンスは送信された<init_db> 値を解析し、使用する新しいデータベースを反映するようにサーバー側接続のフィールドを変更します。

- COM_PREPARE

  ADC アプライアンスは、このコマンドの受信時に要求切り替えを停止します。

- PREPARE query

  このクエリは、接続に関連付けられたプリペアドステートメントを作成するために使用されます。このクエリでは、ADC アプライアンスは「保存と転送」メカニズムを使用します。

監査ログメッセージのサポート

これで、DataStream 機能の監査ログメッセージを生成するように Citrix ADC アプライアンスを構成できるようになりました。監査ログメッセージは、クライアント側およびサーバー側の接続が確立、クローズ、またはドロップされたときに生成されます。ログに記録して表示できるメッセージのカテゴリは、ERROR と INFO です。クライアント側接続のエラーメッセージは「CS」で始まり、サーバー側接続のエラーメッセージは「SS」で始まります。必要に応じて、追加情報が提供されます。たとえば、閉じられた接続（CS_CONN_CLOSED）のログメッセージには、接続 ID のみが含まれます。ただし、確立された接続 (CS_CONN_ESTD) のログメッセージには、接続 ID に加えて、ユーザー名、データベース名、クライアント IP アドレスなどの情報が含まれます。

ドメインネームシステム

October 7, 2021

注: リリース 13.0 ビルド 41.x 以降、ADNS およびプロキシモードの Citrix ADC アプライアンスは、2019 年の DNS フラグに完全に準拠しています。

Citrix ADC アプライアンスは、ドメインの権限のあるドメインネームサーバー（ADNS サーバー）として機能するように構成できます。アプライアンスが権限を持つドメインに属する DNS リソースレコードを追加し、リソースレコードパラメータを設定します。ネットワーク内またはネットワーク外にある DNS ネームサーバーのファームをロードバランシングするプロキシ DNS サーバーとしてアプライアンスを構成することもできます。アプライアンスをエンドリゾルバおよびフォワーダとして設定します。完全修飾ドメイン名が構成されていない場合、名前解決を有効にする DNS サフィックスを構成できます。アプライアンスは、ドメインに属するすべてのレコードを取得する DNS ANY クエリもサポートしています。

アプライアンスは、あるドメインの権限のある DNS サーバーとして同時に機能し、別のドメインの DNS プロキシサーバーとして同時に機能するように構成できます。アプライアンスをゾーンの権限のある DNS サーバーまたは DNS プロキシサーバーとして構成すると、ユーザーデータグラムプロトコル（UDP）に指定されたサイズ制限を超える応答サイズに対して TCP を使用できるようになります。

DNS が **Citrix ADC** でどのように動作するか

Citrix ADC アプライアンスは、ADNS サーバー、DNS プロキシサーバー、エンドリゾルバ、フォワーダとして機能するように構成できます。Citrix ADC アプライアンスでは、次のレコードを含む DNS リソースレコードを追加できます。

- サービス（SRV）レコード
- IPv6（AAAA）レコード
- アドレス（A）レコード
- メール交換 (MX) レコード
- 正規名 (CNAME) レコード
- ポインタ (PTR) レコード
- 権限の開始 (SOA) レコード
- テキスト (TXT) レコード

また、外部 DNS ネームサーバーの負荷分散を行うように Citrix ADC を構成することもできます。

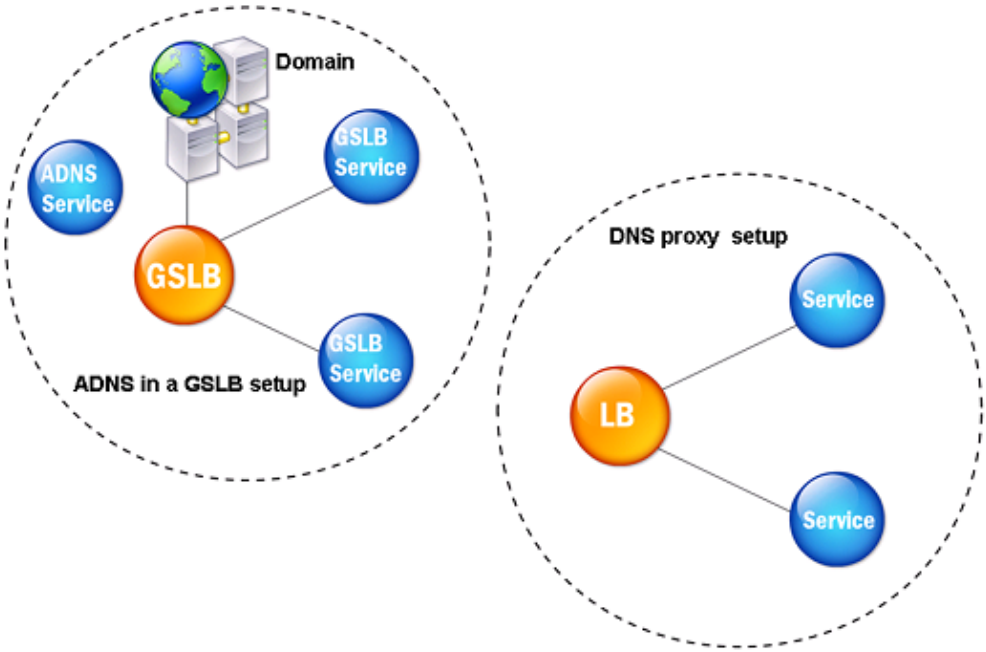
Citrix ADC アプライアンスは、ドメインの権限として構成できます。ドメインの有効な SOA レコードと NS レコードを追加します。

ADNS サーバーは、ゾーンに関する完全な情報を含む DNS サーバーです。

Citrix ADC アプライアンスをゾーンの ADNS サーバーとして構成するには、ADNS サービスを追加し、ゾーンを構成する必要があります。これを行うには、ドメインの有効な SOA レコードと NS レコードを追加します。クライアントが DNS リクエストを送信すると、Citrix ADC アプライアンスは構成されたリソースレコードでドメイン名を検索します。Citrix ADC グローバルサーバー負荷分散 (GSLB) 機能と共に使用するように ADNS サービスを構成できます。

サブドメインの NS レコードを親ドメインのゾーンに追加することで、サブドメインを委任できます。その後、サブドメインネームサーバーごとに「グルーレコード」を追加することで、Citrix ADC にサブドメインに対する権限を付与できます。GSLB が構成されている場合、Citrix ADC はその構成に基づいて GSLB の負荷分散を決定し、選択した仮想サーバーの IP アドレスで応答します。次の図は、ADNS GSLB セットアップと DNS プロキシセットアップのエンティティを示しています。

図 1: DNS プロキシエンティティモデル



Citrix ADC アプライアンスは、DNS プロキシとして機能できます。DNS プロキシの重要な機能である DNS レコードのキャッシュは、Citrix ADC アプライアンスでデフォルトで有効になっています。キャッシングにより、Citrix ADC アプライアンスは、翻訳を繰り返すための迅速な応答を提供できます。負荷分散の DNS 仮想サーバーと DNS サービスを作成し、これらのサービスを仮想サーバーにバインドします。

Citrix ADC には、キャッシュされたデータの存続期間を構成するための最小有効期間（TTL）と最大 TTL の 2 つのオプションがあります。キャッシュされたデータは、これら 2 つのオプションの設定で指定されたとおりにタイムアウトします。Citrix ADC は、サーバーから送信される DNS レコードの TTL をチェックします。TTL が設定された最小 TTL より小さい場合は、設定された最小 TTL に置き換えられます。TTL が設定された最大 TTL より大きい場合は、設定された最大 TTL に置き換えられます。

Citrix ADC では、ドメインに対する否定的な応答をキャッシュすることもできます。否定的な応答は、要求されたドメインに関する情報が存在しないか、サーバーがクエリに対する応答を提供できないことを示します。この情報の保存は、ネガティブキャッシングと呼ばれます。ネガティブキャッシュは、ドメイン上のクエリへの応答を高速化するのに役立ちます。また、オプションでレコードタイプを指定することもできます。

負の応答は、次のいずれかになります。

- NXDOMAIN エラーメッセージ-ローカルキャッシュに負の応答がある場合、Citrix ADC はエラーメッセージ（NXDOMAIN）を返します。応答がローカルキャッシュにない場合、クエリはサーバーに転送され、サーバーは Citrix ADC に NXDOMAIN エラーを返します。Citrix ADC は応答をローカルにキャッシュし、エラーメッ

セージをクライアントに返します。

- NODATA エラーメッセージ-Citrix ADC は、クエリ内のドメイン名が有効であるが、指定された種類のレコードが利用できない場合、NODATA エラーメッセージを送信します。

Citrix ADC は、DNS リクエストの再帰的な解決をサポートしています。再帰的な解決では、リゾルバ（DNS クライアント）はドメイン名の再帰クエリをネーム・サーバに送信します。照会されたネームサーバーがドメインに対して権限を持つ場合、要求されたドメイン名で応答します。それ以外の場合、Citrix ADC は、要求されたドメイン名が見つかるまで、ネームサーバーを再帰的に照会します。

再帰クエリオプションを適用する前に、まず再帰クエリオプションを有効にする必要があります。DNS ルックアップが失敗した場合に、DNS リゾルバーが解決要求 (DNS 再試行) を送信する必要がある回数も設定できます。

Citrix ADC を DNS フォワーダーとして構成できます。フォワーダは DNS 要求を外部ネームサーバに渡します。Citrix ADC では、外部ネームサーバーを追加したり、ネットワーク外のドメインの名前解決を行えます。Citrix ADC では、名前検索の優先順位を DNS または Windows インターネットネームサービス（WINS）に設定することもできます。

ADC アプライアンスが **DNS** を使用してホスト名をそれぞれの **IP** アドレスに解決できるようにする

注：アプライアンスのコマンドラインインターフェイス（CLI）にアクセスするには、SSH ユーティリティが必要です。

デフォルトでは、ADC アプライアンスはホスト名をそれぞれの IP アドレスに解決できません。アプライアンスで名前解決を有効にするには、次のタスクを実行します。

1. ネームサーバを定義します。
2. DNS サフィックスを定義します。

注意事項

CLI から DNS ルックアップを実行します。FreeBSD オペレーティングシステムのシェルプロンプトからの DNS ルックアップは、/etc/resolv.conf ファイルは 127.0.0.2 の IP アドレスを指しています。

次のコマンドは、アプライアンスの CLI では使用できません。

```
- host
- dig
- getent/MIP
- nslookup
<!--NeedCopy-->
```

アプライアンスが SNIP アドレスで DNS サーバに ping を実行できない場合、サーバのステータスは down と表示されます。アプライアンスがファイアウォールの内側にある場合、ping の成功は重要です。

CLI 構成

コマンドプロンプトで入力します。

```
add dns nameServer <Name_Server_IP_Address>
add dns suffix <DNS_Suffix>
<!--NeedCopy-->
```

構成を確認するには、次のように入力します。

```
show dns nameServer
show dns suffix
<!--NeedCopy-->
```

DNS 解決をテストするには、次のように入力します。

```
show dns addrec <Host_Name>
<!--NeedCopy-->
```

GUI 構成

1. [トラフィック管理] **>** **[DNS]** **>** [ネームサーバー] **>** [追加] に移動します。
2. [ネームサーバの作成] ダイアログボックスで、ネームサーバの IP アドレスを入力し、[作成] をクリックします。
3. [トラフィック管理] **>** **[DNS]** **>** **[DNS** サフィックス] **>** [追加] に移動します。
4. [**DNS** サフィックスの作成] ダイアログボックスで、すべてのホストクエリに使用する DNS サフィックス（example.com など）を入力し、[作成] をクリックします。

ラウンドロビン **DNS**

クライアントは DNS リソースレコードを検索する DNS 要求を送信すると、DNS 要求内の名前に解決する IP アドレスの一覧を受信します。次に、クライアントはリスト内の IP アドレスのいずれか（通常は最初のレコードまたは IP アドレス）を使用します。したがって、単一のサーバーがキャッシュの合計 TTL に使用され、多くの要求が到着したときに過負荷になります。

Citrix ADC は、DNS 要求を受信すると、ラウンドロビン方式で DNS リソースレコードのリストの順序を変更することで応答します。この機能は、ラウンドロビン *DNS* と呼ばれます。ラウンドロビンは、トラフィックをデータセンター間で均等に分散します。Citrix ADC はこの機能を自動的に実行します。この動作を構成する必要はありません。

機能の概要

Citrix ADC が ADNS サーバーとして構成されている場合、レコードが構成されている順序で DNS レコードが返されます。Citrix ADC が DNS プロキシとして構成されている場合、サーバーからレコードを受信した順序で DNS レコードを返します。キャッシュに存在するレコードの順序は、サーバーからレコードを受信する順序と一致します。

次に、Citrix ADC は、ラウンドロビン方式で DNS 応答でレコードが送信される順序を変更します。最初の応答にはシーケンス内の最初のレコードが含まれ、2 番目の応答にはシーケンス内の 2 番目のレコードが含まれ、順序は同じシーケンスで継続されます。したがって、同じ名前を要求するクライアントは、異なる IP アドレスに接続できます。

ラウンドロビン **DNS** の例

ラウンドロビン DNS の例として、次のように追加された DNS レコードについて考えます。

```
1   add dns addRec ns1 1.1.1.1  add dns addRec ns1 1.1.1.2  add dns
       addRec ns1 1.1.1.3  add dns addRec ns1 1.1.1.4
2 <!--NeedCopy-->
```

ドメイン abc.com は、次のように NS レコードにリンクされています。

```
1   add dns nsrec abc.com. ns1
2 <!--NeedCopy-->
```

Citrix ADC が ns1 の A レコードのクエリを受信すると、アドレスレコードは次のようにラウンドロビン方式で提供されます。最初の DNS 応答では、1.1.1.1 が最初のレコードとして提供されます。

```
1   ns1.                    1H IN A         1.1.1.1  ns1.
                           1H IN A         1.1.1.2  ns1.
                           1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4
2 <!--NeedCopy-->
```

2 番目の DNS 応答では、2 番目の IP アドレス 1.1.1.2 が最初のレコードとして提供されます。

```
1   ns1.                    1H IN A         1.1.1.2  ns1.
                           1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4  ns1.
                           1H IN A         1.1.1.1
2 <!--NeedCopy-->
```


3 番目の DNS 応答では、3 番目の IP アドレス 1.1.1.2 が最初のレコードとして提供されます。

```
1   ns1.                    1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4  ns1.
                           1H IN A         1.1.1.1  ns1.
                           1H IN A         1.1.1.2
2 <!--NeedCopy-->
```

DNS リソースレコードを構成する

October 7, 2021

Citrix® ADC アプライアンスのリソースレコードは、アプライアンスをゾーンの ADNS サーバーとして構成するときに構成します。アプライアンスが DNS プロキシ・サーバーであるゾーンにリソース・レコードが属している場合は、アプライアンスのリソース・レコードを構成することもできます。アプライアンスでは、次のレコードタイプを設定できます。

- サービス記録
- AAAA レコード
- Address レコード
- Mail Exchange レコード
- Name Server レコード
- Canonical レコード
- Pointer レコード
- NAPTR レコード
- 権限の開始レコード
- Text レコード

次の表に、Citrix ADC アプライアンスでドメイン名レコードに対して構成できるレコードの種類を示します。たとえば、1 つのレコードに対して最大 25 個の IP アドレスを設定できます。

表 1. レコードタイプと番号は設定可能

レコードタイプ	レコード数
アドレス（A）	25
IPv6 (AAAA)	5
Mail exchange (MX)	12
ネームサーバー（NS）	16

レコードタイプ	レコード数
サービス (SRV)	8
ポインタ (PTR)	20
標準名 (CNAME)	1
権限の開始 (SOA)	1
テキスト (TXT)	20
名前付け権限ポインタ (NAPTR)	20

> 注:
>
> 特定のホスト名の IP アドレスの最大数は 25 です。ただし、異なる住所レコードの数は 25 を超える場合があります。

サービスの **SRV** レコードの作成

October 7, 2021

SRV レコードは、Citrix ADC アプライアンスで使用可能なサービスに関する情報を提供します。SRV レコードには、次の情報が含まれています。

- サービスの名前とプロトコル
- ドメイン名
- TTL
- DNS クラス
- ターゲットの優先度
- 同じ優先度のレコードの重み
- サービスのポート
- サービスのホスト名。

Citrix ADC は、優先順位が最も低い SRV レコードを最初に選択します。サービスに、同じ優先順位の複数の SRV レコードがある場合、クライアントは weight フィールドを使用して、使用するホストを決定します。

CLI を使用した **SRV** レコードの追加

コマンドプロンプトで、次のコマンドを入力して SRV レコードを追加し、構成を確認します。


```
-  add dns srvRec <domain> <target> -priority <positive_integer> -
   weight <positive_integer> -port <positive_integer> [-TTL <secs>]
-  sh dns srvRec <domain>
<!--NeedCopy-->
```

例:

```
> add dns srvRec _http._tcp.example.com nameserver1.com -priority 1 -
   weight 1 -port 80
 Done
> show dns srvRec _http._tcp.example.com
1)      Domain Name : _http._tcp.example.com
        Target Host : nameserver1.com
        Priority : 1    Weight : 1
        Port : 80       TTL : 3600 secs
 Done
<!--NeedCopy-->
```

CLI を使用した **SRV** レコードの変更または削除

- SRV レコードを変更するには、次のように入力します。
  - `set dns srvRec` コマンド
  - SRV レコードが構成されているドメインの名前
  - 関連するサービスをホストするターゲットホストの名前
  - 変更するパラメータとその新しい値
- SRV レコードを削除するには、次のように入力します。
  - `rm dns srvRec` コマンド
  - SRV レコードが構成されているドメインの名前
  - 関連するサービスをホストするターゲットホストの名前

GUI を使用した **SRV** レコードの設定

[トラフィック管理**] > [DNS] > [**レコード**] > [SRV** レコード**]** に移動し、SRV レコードを作成します。

ドメイン名の **AAAA** レコードを作成する

October 7, 2021

AAAA リソースレコードには、単一の IPv6 アドレスが格納されます。

CLI を使用した AAAA レコードの追加

コマンドプロンプトで次のコマンドを入力して、AAAA レコードを追加し、設定を確認します。

```
1 -  add dns aaaaRec <hostName> <IPv6Address> ... [-TTL <secs>]
2 -  show dns aaaaRec <hostName>
3 <!--NeedCopy-->
```

例:

```
1 > add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57
     ab
2  Done
3 > show dns aaaaRec www.example.com
4 1)      Host Name : www.example.com
5         Record Type : ADNS              TTL : 5 secs
6         IPV6 Address : 2001:db8::1428:57ab
7  Done
8 <!--NeedCopy-->
```

AAAA レコードとドメイン名に関連付けられているすべての IPv6 アドレスを削除するには、`rm dns aaaaRec` コマンドと AAAA レコードが設定されているドメイン名を入力します。AAAA レコードのドメイン名に関連付けられている IPv6 アドレスのサブセットのみを削除するには、次のように入力します。

- `rm dns aaaaRec` -command
- AAAA レコードが設定されているドメイン名
- 削除する IPv6 アドレス

GUI を使用して AAAA レコードを追加する

[トラフィック管理**] > [DNS] > [**レコード**] > [AAAA** レコード**]** に移動し、AAAA レコードを作成します。

ドメイン名のアドレスレコードを作成する

October 7, 2021

アドレス (A) レコードは、ドメイン名を IPv4 アドレスにマッピングする DNS レコードです。

グローバルサーバ負荷分散 (GSLB) に参加しているホストのアドレスレコードは削除できません。ただし、GSLB 仮想サーバーからドメインのバインドを解除すると、Citrix ADC は GSLB ドメインに追加されたアドレスレコードを

削除します。手動で削除できるのは、ユーザが設定したレコードだけです。NS、MX、CNAME などのレコードによって参照されるホストのレコードは削除できません。

CLI を使用してアドレスレコードを追加する

コマンドプロンプトで次のコマンドを入力して、アドレスレコードを追加し、構成を確認します。

```
1 -  add dns addRec <hostName> <IPAddress> [-TTL <secs>]
2 -  show dns addRec <hostName>
3 <!--NeedCopy-->
```

例:

```
1 > add dns addRec ns.example.com 192.0.2.0
2  Done
3 > show dns addRec ns.example.com
4 1)      Host Name : ns.example.com
5         Record Type : ADNS                    TTL : 5 secs
6         IP Address : 192.0.2.0
7  Done
8 <!--NeedCopy-->
```

アドレスレコードとドメイン名に関連付けられているすべての IP アドレスを削除するには、`rm dns addRec` コマンドとアドレスレコードが構成されているドメイン名を入力します。アドレスレコードのドメイン名に関連付けられている IP アドレスのサブセットのみを削除するには、次のように入力します。

- `rm dns addRec` -command
- アドレスレコードが構成されているドメイン名
- 削除する IP アドレス

GUI を使用してアドレスレコードを追加する

[トラフィック管理**] > [DNS] > [**レコード**] > [**アドレスレコード**]** に移動し、アドレスレコードを作成します。

メール交換サーバーの MX レコードの作成

October 7, 2021

Mail Exchange (MX) レコードは、インターネット経由で電子メールメッセージを転送するために使用されます。MX レコードには、使用する MX サーバーを指定する MX プリファレンスが含まれています。MX プリファレンス値の範囲は 0 ～65536 です。MX レコードには、一意の MX プリファレンス番号が含まれます。MX レコードの MX プリファレンスと TTL 値を設定できます。

電子メールメッセージがインターネット経由で送信されると、メール転送エージェントは、ドメイン名の MX レコードを要求する DNS クエリを送信します。このクエリは、ドメインのメール交換サーバーのホスト名のリストとプリファレンス番号を返します。MX レコードがない場合、そのドメインのアドレスレコードに対する要求が行われます。1 つのドメインに複数のメール交換サーバーを設定できます。

CLI を使用して MX レコードを追加する

コマンドプロンプトで、次のコマンドを入力して MX レコードを追加し、構成を確認します。

```
1 -  add dns mxRec <domain> -mx <string> -pref <positive_integer> [-TTL <
     secs>]
2 -  show dns mxRec <domain>
3 <!--NeedCopy-->
```

例:

```
1 > add dns mxRec example.com -mx mail.example.com -pref 1
2  Done
3 > show dns mxRec example.com
4 1)      Domain : example.com    MX Name : mail.example.com
5         Preference :   1        TTL : 5 secs
6  Done
7 <!--NeedCopy-->
```

CLI を使用して MX レコードを変更または削除する

- MX レコードを変更するには、`set dns mxRec` コマンド、MX レコードが構成されているドメインの名前、MX レコードの名前、変更するパラメーターを新しい値とともに入力します。
- TTL パラメーターをデフォルト値に設定するには、`unset dns mxRec` コマンド、MX レコードが構成されているドメインの名前、MX レコードの名前、および TTL 値なしの-TTL を入力します。`unset dns mxRec` コマンドを使用して、TTL パラメーターのみの設定を解除できます。
- MX レコードを削除するには、`rm dns mxRec` コマンド、MX レコードが構成されているドメインの名前、および MX レコードの名前を入力します。

GUI を使用して MX レコードを追加する

[トラフィック管理**] > [DNS] > [**レコード**] > [**メール交換レコード**]** に移動し、MX レコードを作成します。

オーソリテイティブ・サーバの NS レコードの作成

October 7, 2021

ネームサーバー (NS) レコードは、ドメインの権限のあるサーバーを指定します。最大 16 個の NS レコードを設定できます。NS レコードを使用して、サブドメインの制御を DNS サーバーに委任できます。

CLI を使用した NS レコードの作成

コマンドプロンプトで、次のコマンドを入力して NS レコードを作成し、構成を確認します。

```
1 -  add dns nsRec <domain> <nameServer> [-TTL <secs>]
2 -  show dns nsRec <domain>
3 <!--NeedCopy-->
```

例:

```
1 > add dns nsRec example.com nameserver1.example.com
2  Done
3 > show dns nsRec example.com
4 1)      Domain : example.com    NameServer : nameserver1.example.com
5         TTL : 5 sec
6  Done
7 <!--NeedCopy-->
```

NS レコードを削除するには、`rm dns nsRec` コマンド、NS レコードが属するドメインの名前、およびネームサーバの名前を入力します。

GUI を使用した NS レコードの作成

[トラフィック管理**] > [DNS] > [**レコード**] > [**ネームサーバレコード**]** に移動し、NS レコードを作成します。

サブドメインの **CNAME** レコードの作成

October 7, 2021

正規名レコード (CNAME レコード) は、DNS 名のエイリアスです。これらのレコードは、複数のサービスが DNS サーバーに照会する場合に便利です。アドレス (A) レコードを持つホストは、CNAME レコードを持つことはできません。

プロキシモードの Citrix ADC アプライアンスが、サーバーではなくキャッシュからアドレスレコードを要求する場合があります。

CLI を使用して **CNAME** レコードを追加する

コマンドプロンプトで、次のコマンドを入力して CNAME レコードを作成し、構成を確認します。

```
1 -  add dns cnameRec <aliasName> <canonicalName> [-TTL <secs>]
2 -  show dns cnameRec <aliasName>
3 <!--NeedCopy-->
```

例**:**

```
1 > add dns cnameRec www.example.com www.examplenw.com
2  Done
3 > show dns cnameRec www.example.com
4         Alias Name      Canonical Name  TTL
5 1)      www.example.com        www.examplenw.com       5 secs
6  Done
7 <!--NeedCopy-->
```

特定のドメインの CNAME レコードを削除するには、`rm dns cnameRec` コマンドとドメイン名のエイリアスを入力します。

GUI を使用して **CNAME** レコードを追加する

[トラフィック管理**] > [DNS] > [**レコード**] > [**正規レコード**]** に移動し、CNAME レコードを作成します。

CNAME レコードをキャッシュする

プロキシモードで展開された場合、ADC アプライアンスは常にアドレスレコードのクエリをバックエンドサーバーに送信するとは限りません。この動作は、アドレスレコードのクエリに対する回答の場合、部分的な CNAME チェーン

がキャッシュに存在する場合に発生します。ADC が部分的な CNAME レコードをキャッシュし、キャッシュからクエリを提供する条件はほとんどありません。次に、条件を示します。

- Citrix ADC はプロキシモードで展開する必要があります。
- バックエンドサーバーからの応答には CNAME チェーンが必要です。このチェーンの場合、回答セクションの最後のエントリのレコードタイプは CNAME であり、質問タイプは CNAME ではありません。
- バックエンドサーバーからの応答は、データなしまたは NX ドメインにすることはできません。
- バックエンドサーバーからの応答は、信頼できる応答である必要があります。

通信ドメインの **NAPTR** レコードの作成

October 7, 2021

NAPTR（Naming Address Pointer）は、電気通信ドメインで最も一般的に使用されている DNS レコードの 1 つです。NAPTR レコードは、インターネットテレフォニーアドレススペースをインターネットアドレススペースにマッピングします。したがって、モバイルデバイスが正しいサーバーに要求を送信できるようにします。NAPTR レコードとサービスレコード (SRV) を組み合わせることで、複数のレコードを連鎖して、新しいドメインラベルまたは統一リソース識別子 (URI) を生成する複雑な書き換えルールを形成できます。NAPTR の DNS コードは 35 です。

Citrix ADC は、ADNS モードとプロキシモードの 2 つのモードで NAPTR をサポートします。プロキシモードでは、ADC はサーバーからの応答をキャッシュし、将来のクエリをサーバーにキャッシュされたレコードを使用します。Citrix ADC では、特定のドメインに対して最大 20 個の NAPTR レコードを追加できます。Citrix ADC は、DNS NAPTR レコードのクエリへの応答をキャッシュします。NAPTR レコードに対する後続の要求はすべて、キャッシュから提供されます。

CLI を使用して **NAPTR** レコードを作成する

コマンドプロンプトで次のコマンドを入力して、NAPTR レコードを追加し、構成を確認します。

```
add dns naptrRec <order> <preference>[flags<string>][services<string>](
regexp<expressions>|-replacement<string>)[-TTL<secs>]
```

CLU を使用して **NAPTR** レコードを削除する

```
rm dns naptrRec<domain> (<order> <preference> [-flags <string>] [-services
 <string>] (-regexp <expression> | -replacement <string>))| -recordId <
positive_integer>@)
```

GUI を使用した **NAPTR** レコードの設定

[トラフィック管理**] > [DNS] > [**レコード**] > [NAPTR** レコード**]** に移動し、NAPTR レコードを作成します。

IPv4 アドレスと IPv6 アドレスの PTR レコードの作成

October 7, 2021

ポインタ (PTR) レコードは、IP アドレスをドメイン名に変換します。IPv4 PTR レコードは、IP アドレスのオクテットによって逆の順序で「in-addr.arpa」という文字列で表されます。」が最後に追加されました。たとえば、IP アドレス 1.2.3.4 の PTR レコードは 4.3.2.1.in-addr.arpa です。

IPv6 アドレスは、ドメイン IP6.ARPA の下にリバースマップされます。IPv6 リバースマップは、RFC 3596 で定義されているように、接尾辞「.IP6.ARPA」を持つドットで区切られたニブルのシーケンスを使用します。たとえば、アドレス 4321:0:1:2:3:4:567:89 に対応する逆引き参照ドメイン名は、b.a.9.8.7.6.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.1.0.0.0.0.1.2.3.4.IP6.ARPA になります。

CLI を使用した PTR レコードの追加

コマンドプロンプトで次のコマンドを入力して PTR レコードを追加し、構成を確認します。

```
-  add dns ptrRec <reverseDomain> <domain> [-TTL <secs>]
-  show dns ptrRec <reverseDomain>
<!--NeedCopy-->
```

例:

```
> add dns ptrRec 0.2.0.192.in-addr.arpa example.com
 Done
> show dns ptrRec 0.2.0.192.in-addr.arpa
1)      Reverse Domain Name : 0.2.0.192.in-addr.arpa
        Domain Name : example.com               TTL : 3600 secs
 Done
<!--NeedCopy-->
```

PTR レコードを削除するには、`rm dns ptrRec` コマンドと PTR レコードに関連付けられた逆ドメイン名を入力します。

GUI を使用して PTR レコードを追加する

[トラフィック管理] > [DNS] > [レコード] > [PTR レコード] に移動し、**PTR** レコードを作成します。

権限のある情報の **SOA** レコードの作成

October 7, 2021

SOA (Start of Authorority) レコードは、ゾーン頂点でのみ作成され、ゾーンに関する情報が含まれます。レコードには、他のパラメーターの中でも、プライマリネームサーバー、連絡先情報（電子メール）、およびレコードのデフォルト（最小）存続時間（TTL）値が含まれます。

CLI を使用して **SOA** レコードを作成する

コマンドプロンプトで、次のコマンドを入力して SOA レコードを追加し、構成を確認します。

```
1 -  add dns soaRec <domain> -originServer <originServerName> -contact <
     contactName>
2 -  sh dns soaRec <do main>
3 <!--NeedCopy-->
```

例:

```
1 > add dns soaRec example.com -originServer nameserver1.example.com -
     contact admin.example.com
2  Done
3 > show dns soaRec example.com
4 1)      Domain Name : example.com
5         Origin Server : nameserver1.example.com
6         Contact : admin.example.com
7         Serial No. : 100        Refresh : 3600 secs     Retry : 3 secs
8         Expire : 3600 secs      Minimum : 5 secs        TTL : 3600 secs
9  Done
10 <!--NeedCopy-->
```

CLI を使用して **SOA** レコードを変更または削除する

- SOA レコードを変更するには、`set dns soaRec` コマンド、レコードが構成されているドメインの名前、および変更するパラメータを新しい値とともに入力します。
- SOA レコードを削除するには、`rm dns soaRec` コマンドとレコードが構成されているドメインの名前を入力します。

GUI を使用して SOA レコードを構成する

[トラフィック管理**] > [DNS] > [**レコード**] > [SOA** レコード**]** に移動し、SOA レコードを作成します。

説明テキストを保持するための TXT レコードの作成

October 7, 2021

ドメインホストは、情報提供を目的として TXT レコードを保存します。TXT レコードの RDATA コンポーネントは、可変長の 1 つ以上の文字列で構成されており、受信者がドメインについて知る必要のある事実上すべての情報を格納できます。また、サービスプロバイダー、連絡担当者、電子メールアドレス、および関連する詳細に関する情報を含めることもできます。SPF（送信者ポリシーフレームワーク）保護は、TXT レコードで最も顕著な使用例でした。

Citrix ADC アプライアンスのすべての構成タイプ（権限のある DNS、DNS プロキシ、エンドリゾルバ、フォワーダ構成）は、TXT レコードをサポートします。1 つのドメインには、最大 20 個の TXT リソースレコードを追加できます。各リソースレコードは、内部で生成された一意のレコード ID で格納されます。レコードの ID を表示し、それを使用してレコードを削除できます。ただし、TXT リソースレコードを変更することはできません。

CLI を使用して TXT リソースレコードを作成する

コマンドプロンプトで、次のコマンドを入力して TXT リソースレコードを作成し、構成を確認します。

```
1 -  add dns txtRec <domain> <string> ... [-TTL <secs>]
2 -  show dns txtRec [<domain> | -type <type>]
3 <!--NeedCopy-->
```

例:

```
1 > add dns txtRec www.example.com ”Contact: Mark” ”Email: mark@example.
     com” -TTL 36000
2  Done
3 > show dns txtRec www.example.com
4 1)  Domain : www.example.com    Record id: 13783       TTL : 36000 secs
      Record Type : ADNS
5         ”Contact: Mark”
6         ”Email: mark@example.com”
7  Done
8 <!--NeedCopy-->
```


CLI を使用して TXT リソースレコードを削除する

コマンドプロンプトで次のコマンドを入力して、TXT リソースレコードを削除し、構成を確認します。

```
-  rm dns txtRec <domain> (<string> ... | -recordId <positive_integer>)
-  show dns txtRec [<domain> | -type <type>]
<!--NeedCopy-->
```

例:
最初に show dns txtRec コマンドを使用して、削除する TXT リソースレコードのレコード ID を次のように表示できます。

```
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 36865       TTL : 36000 secs
     Record Type : ADNS
         ”Contact: Evan”
         ”Email: evan@example.com”
2)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
     Record Type : ADNS
         ”Contact: Mark”
         ”Email: mark1@example.com”
 Done
<!--NeedCopy-->
```

TXT レコードを削除する簡単な方法は、レコード ID を使用することです。文字列を指定する場合は、レコードに格納されている順序で文字列を入力します。次の例では、TXT レコードは、レコード ID を使用して削除されます。

```
>rm dns txtRec www.example.com -recordID 36865
 Done
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
     Record Type : ADNS
         ”Contact: Mark”
         ”Email: mark1@example.com”
 Done
<!--NeedCopy-->
```

GUI を使用して TXT レコードを構成する

[トラフィック管理**] > [DNS] > [**レコード**] > [TXT** レコード**]** に移動し、TXT レコードを作成します。

DNS 統計情報の表示

October 7, 2021

Citrix ADC アプライアンスによって生成された DNS 統計情報を表示できます。DNS 統計には、実行時、構成、およびエラーの統計が含まれます。

CLI を使用した DNS レコードの統計情報の表示

コマンドプロンプトで入力します。

```
stat dns
```

例:

```
1  > stat dns
2  DNS Statistics
3
4  Runtime Statistics
5  Dns queries                          21
6  NS queries                            8
7  SOA queries                          18
8  .
9  .
10 .
11 Configuration Statistics
12 AAAA records                         17
13 A records                            36
14 MX records                            9
15 .
16 .
17 .
18 Error Statistics
19 Nonexistent domain                   17
20 No AAAA records                       0
21 No A records                         13
22 .
23 .
24 .
25 Done
26 <!--NeedCopy-->
```


GUI を使用した **DNS** レコードの統計情報の表示

1. [トラフィック管理] **> [DNS**] に移動します。
2. 詳細ペインで、[統計] をクリックします。

DNS ゾーンを構成する

October 7, 2021

Citrix ADC アプライアンス上の DNS ゾーンエンティティは、アプライアンス上のドメインの所有権を容易にします。アプライアンスのゾーンでは、ゾーンの DNS セキュリティ拡張 (DNSSEC) を実装したり、ゾーンの DNSSEC 操作を DNS サーバーからアプライアンスにオフロードしたりすることもできます。DNSSEC 署名操作は、DNS ゾーン内のすべてのリソースレコードに対して実行されます。したがって、ゾーンに署名する場合、またはゾーンの DNSSEC 操作をオフロードする場合は、まず Citrix ADC アプライアンスにゾーンを作成する必要があります。

次のシナリオで、アプライアンスに DNS ゾーンを作成します。

- Citrix ADC アプライアンスは、ゾーン内のすべてのレコードを所有します。つまり、アプライアンスはゾーンの権限のある DNS サーバーとして動作しています。ゾーンは、proxyMode パラメータを NO に設定して作成する必要があります。
- Citrix ADC アプライアンスは、ゾーン内のレコードのサブセットのみを所有します。ゾーン内の他のすべてのリソースレコードは、一連のバックエンドネームサーバーでホストされます。アプライアンスは、これらのバックエンドサーバーの DNS プロキシサーバーとして構成されています。Citrix ADC アプライアンスがゾーン内のリソースレコードのサブセットのみを所有する一般的な構成は、グローバルサーバー負荷分散（GSLB）構成です。Citrix ADC アプライアンスは GSLB ドメイン名のみを所有し、バックエンドネームサーバーは他のすべてのレコードを所有します。ゾーンは、proxyMode パラメータを YES に設定して作成する必要があります。
- ゾーンの DNSSEC 操作を権限のある DNS サーバーからアプライアンスにオフロードする場合。ゾーンは、proxyMode パラメータを YES に設定して作成する必要があります。ゾーンの設定をさらに構成する必要がある場合があります。

現在のトピックでは、最初の 2 つのシナリオでゾーンを作成する方法について説明します。アプライアンスに DNSSEC 操作をオフロードするためのゾーンを構成する方法の詳細については、「Citrix ADC アプライアンスへの DNSSEC 操作のオフロード」を参照してください。

> 注
>
> ADC アプライアンスがゾーンの権限のある DNS サーバーとして動作している場合は、ゾーンを作成する前に、ゾーンの権限の開始（SOA）レコードとネームサーバー（NS）レコードを作成する必要があります。Citrix ADC がゾーンの DNS プロキシサーバーとして動作している場合は、Citrix ADC アプライアンスで SOA レコードと NS レコードを作成しないでください。SOA レコードと NS レコードの作成の詳細については、「DNS リソースレコードの設定」を参照してください。

ゾーンを作成すると、ゾーン名で終わる既存のドメイン名およびリソースレコードはすべて、自動的にゾーンの一部として処理されます。また、ゾーンの名前と一致するサフィックスで作成された新しいリソースレコードは、暗黙的にゾーンに含まれます。

CLI を使用して、Citrix ADC アプライアンスで DNS ゾーンを作成する

コマンドプロンプトで次のコマンドを入力して、Citrix ADC アプライアンスに DNS ゾーンを追加し、構成を確認します。

```
-  add dns zone <zoneName> -proxyMode ( YES | NO )
-  show dns zone [<zoneName> | -type <type>]
<!--NeedCopy-->
```

例:

```
> add dns zone example.com -proxyMode Yes
 Done
> show dns zone example.com
          Zone Name : example.com
          Proxy Mode : YES
Done
<!--NeedCopy-->
```

CLI を使用して DNS ゾーンを変更または削除する

- DNS ゾーンを変更するには、`set dns zone` コマンド、DNS ゾーンの名前、変更するパラメーターを新しい値とともに入力します。
- DNS ゾーンを削除するには、`rm dns zone` コマンドと DNS ゾーンの名前を入力します。

GUI を使用して DNS ゾーンを構成する

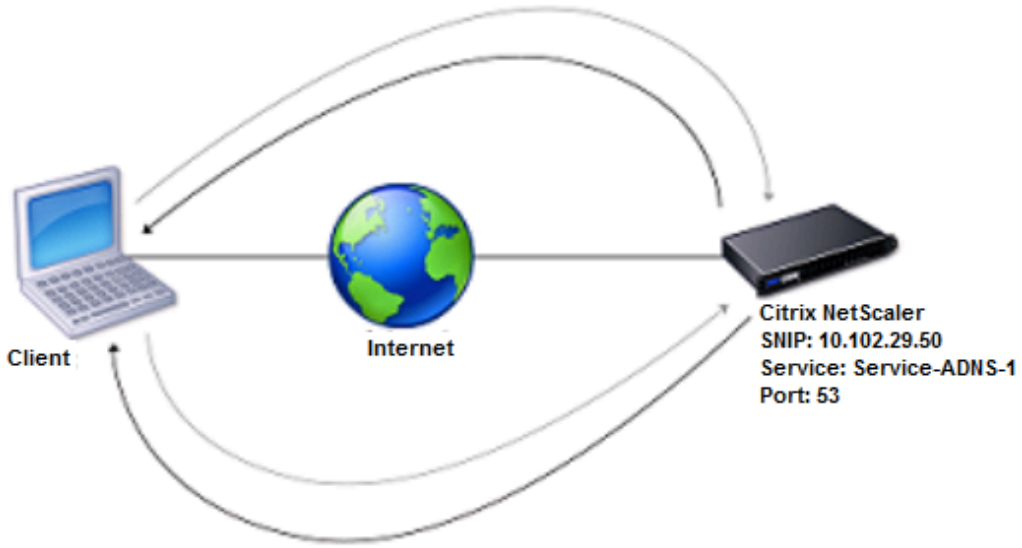
[トラフィック管理**] > [DNS] > [**ゾーン**]** に移動し、DNS ゾーンを作成します。

Citrix ADC を ADNS サーバーとして構成する

October 7, 2021

ADC アプライアンスは、ドメインの権限のあるドメインネームサーバ（ADNS）として機能するように設定できます。ドメインの ADNS サーバーとして、Citrix ADC は、ドメインに属するすべての種類の DNS レコードに対する DNS 要求を解決します。ドメインの ADNS サーバーとして機能するように Citrix ADC を構成するには、ADNS サービスを作成し、Citrix ADC でドメインの NS レコードとアドレスレコードを構成する必要があります。ADNS サービスは、サブネット IP アドレス (SNIP) または別の IP アドレスを使用して構成できます。次のトポロジ図は、設定例と、要求と応答のフローを示しています。

図 1: ADNS としての Citrix ADC



次の表に、前述のトポロジ図に示した ADNS サービス用に構成されるパラメーターを示します。

エンティティタイプ	名前	IP アドレス	種類	ポート
ADNS サービス	Service-ADNS-1	10.102.29.51	ADNS	53

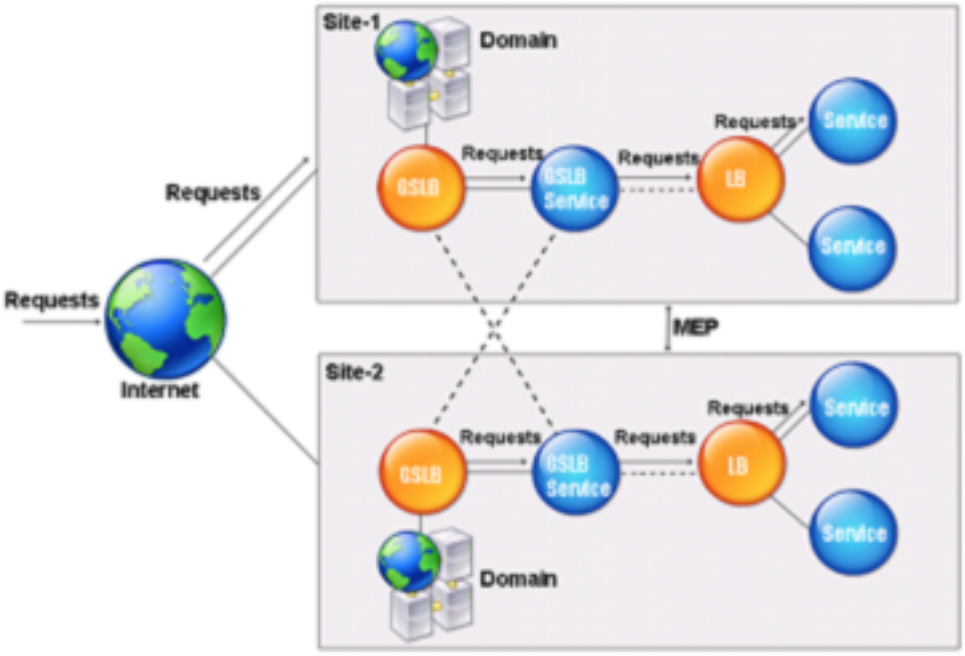
表 1. ADNS サービスの構成例

ADNS セットアップを構成するには、ADNS サービスを構成する必要があります。ADNS サービスの構成手順については、「負荷分散」を参照してください。

DNS 解決中、ADNS サーバーは DNS プロキシまたはローカル DNS サーバーに Citrix ADC にドメインの IP アドレスを問い合わせるよう指示します。Citrix ADC はドメインに対して権限を持つため、IP アドレスは DNS プロキシま

たはローカル DNS サーバーに送信されます。次の図は、GSLB 構成における ADNS サーバーの配置と役割を示しています。

図 2：GSLB エンティティモデル



注：ADNS モードでは、SOA レコードと ADNS レコードを削除すると、Citrix ADC がホストするドメインでは機能しません。ANY クエリ（ANY クエリの詳細については、「DNSANY クエリ」を参照）、および NODATA や NXDOMAIN などのネガティブレスポンス。

ADNS サービスを作成する

ADNS サービスは、グローバルサービスの負荷分散に使用されます。GSLB セットアップの作成の詳細については、「グローバルサーバーの負荷分散」を参照してください。ADNS サービスを追加、変更、有効化、無効化、および削除できます。ADNS サービスの作成手順については、「サービスの構成」を参照してください。

注: SNIP または任意の新しい IP アドレスを使用するように ADNS サービスを構成できます。

ADNS サービスを作成すると、Citrix ADC は、構成された ADNS サービスの IP およびポート上の DNS クエリに応答します。

ADNS サービスのプロパティを表示して、構成を確認できます。名前、状態、IP アドレス、ポート、プロトコル、最大クライアント接続などのプロパティを表示できます。

TCP を使用するように ADNS セットアップを構成します

デフォルトでは、一部のクライアントでは DNS にユーザーデータグラムプロトコル (UDP) を使用します。このプロトコルでは、UDP パケットのペイロード長として 512 バイトの制限が指定されています。サイズが 512 バイトを超えるペイロードを処理するには、クライアントは TCP を使用する必要があります。TCP を介した DNS 通信を有効にするには、DNS 用の TCP プロトコルを使用するように Citrix ADC アプライアンスを構成する必要があります。Citrix ADC は、DNS 応答パケットにトランケーションビットを設定します。切り捨てビットは、応答が UDP に対して大きすぎること、およびクライアントが TCP 接続を介して要求を送信する必要があることを指定します。その後、クライアントはポート 53 で TCP プロトコルを使用し、Citrix ADC への新しい接続を開きます。Citrix ADC は、ADNS サービスの IP アドレスでポート 53 をリッスンし、クライアントからの新しい TCP 接続を受け入れます。

TCP プロトコルを使用するように Citrix ADC を構成するには、ADNS_TCP サービスを構成する必要があります。ADNS_TCP サービスの作成手順については、負荷分散を参照してください。

> 重要
>
> DNS に UDP を使用し、UDP のペイロード長が 512 バイトを超える場合にのみ TCP を使用するように Citrix ADC を構成するには、ADNS および ADNS_TCP サービスを構成する必要があります。の IP アドレス ADNS_TCP service は、ADNS サービスの IP アドレスと同じである必要があります。

DNS リソースレコードを追加する

ADNS サービスを作成したら、DNS レコードを追加できます。DNS レコードの追加手順については、「DNS リソースレコードの設定」を参照してください。

ADNS サービスの削除

サービスの削除手順については、「負荷分散」を参照してください。

ドメインの委任を構成する

ドメイン委任は、ドメイン領域の一部に対する責任を別のネームサーバーに割り当てるプロセスです。したがって、ドメインの委任中に、クエリに応答する責任は別の DNS サーバーに委任されます。委任は NS レコードを使用します。

次の例では、sub1.abc.com が abc.com のサブドメインです。ここでは、サブドメインをネームサーバ ns2.sub1.abc.com に委任し、ns2.sub1.abc.com のアドレスレコードを追加する手順について説明します。

ドメインの委任を構成するには、次のタスクを実行する必要があります。これらのタスクについては、以降のセクションで説明します。

1. ドメインの SOA レコードを作成します。
2. NS レコードを作成して、ドメインのネーム・サーバを追加します。
3. ネームサーバーのアドレスレコードを作成します。
4. サブドメインを委任する NS レコードを作成します。
5. ネームサーバーのグルーレコードを作成します。

SOA レコードの作成

SOA レコードの設定手順については、「信頼できる情報の SOA レコードの作成」を参照してください。

ネーム・サーバの **NS** レコードの作成

NS レコードの構成手順については、「権限のあるサーバーの NS レコードの作成」を参照してください。[ネームサーバー] リストで、プライマリ権限のあるネームサーバー（ns1.abc.com など）を選択します。

住所レコードを作成する

アドレスレコードの設定手順については、「ドメイン名のアドレスレコードを作成する」を参照してください。[ホスト名] および [IP アドレス] テキストボックスに、DNS アドレスレコードのドメイン名と IP アドレス（たとえば、ns1.abc.com と 10.102.11.135）を入力します。

ドメイン委任用の **NS** レコードを作成する

NS レコードの構成手順については、「権限のあるサーバーの NS レコードの作成」を参照してください。[ネームサーバー] リストで、プライマリ権限のあるネームサーバー（ns2.sub1.abc.com など）を選択します。

グルーレコードの作成

NS レコードは通常、SOA レコードの直後に定義されます（制限ではありません）。ドメインには少なくとも 2 つの NS レコードが必要です。NS レコードがドメイン内で定義されている場合は、一致する Address レコードが必要です。このアドレスレコードは、接着レコードと呼ばれます。グルーレコードは、DNS クエリを高速化します。

サブドメインのグルーレコードを追加する手順については、「アドレス (A) レコードの追加手順」、「DNS リソースレコードの設定」を参照してください。

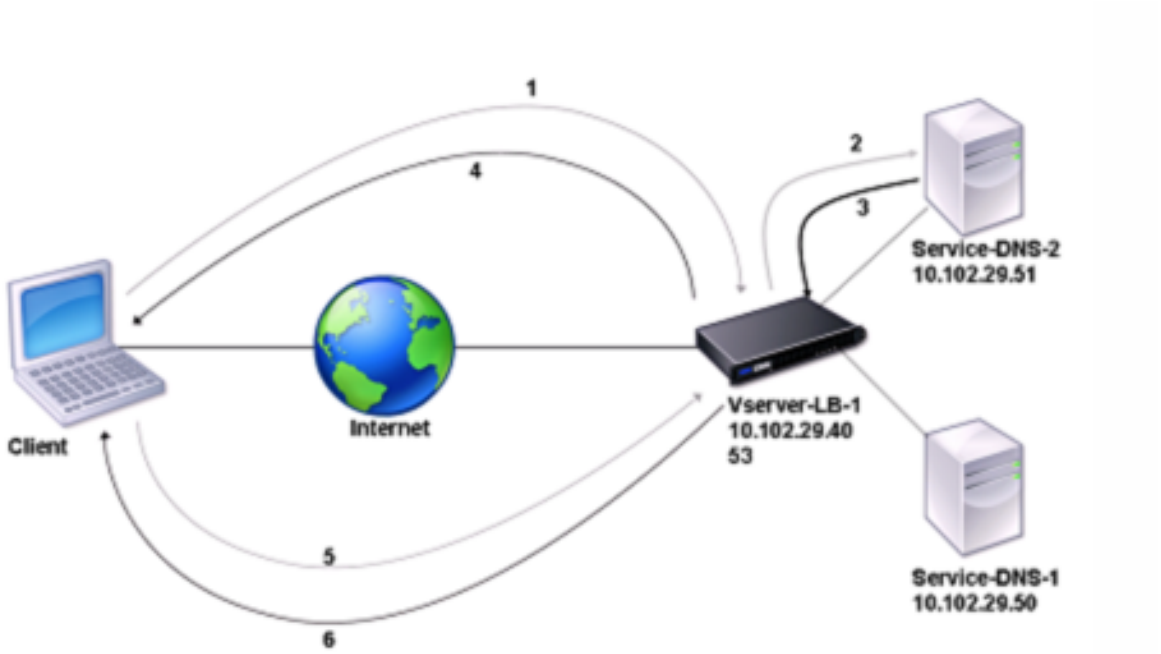
アドレスレコードの設定手順については、「ドメイン名のアドレスレコードを作成する」を参照してください。[ホスト名] および [IP アドレス] テキストボックスに、DNS アドレスレコードのドメイン名と IP アドレスを入力します（例：それぞれ ns2.sub1.abc.com と 10.102.12.135）。

Citrix ADC アプライアンスを **DNS** プロキシサーバーとして構成する

October 7, 2021

ADC アプライアンスは、DNS プロキシ・サーバとして、単一の DNS サーバまたは DNS サーバのグループのプロキシとして機能できます。要求と応答の流れを次のサンプルトポロジ図に示します。

図 1: DNS プロキシとしての Citrix ADC



デフォルトでは、Citrix ADC アプライアンスは DNS ネームサーバーからの応答をキャッシュします。アプライアンスは DNS クエリーを受信すると、クエリーされたドメインがキャッシュ内でチェックされます。照会されたドメインのアドレスがキャッシュに存在する場合、Citrix ADC は対応するアドレスをクライアントに返します。それ以外の場合、DNS ネームサーバーにクエリが転送され、DNS ネームサーバーはアドレスの可用性をチェックし、Citrix ADC に返します。その後、Citrix ADC はクライアントにアドレスを返します。

以前にキャッシュされたドメインに対する要求の場合、Citrix ADC は、構成された DNS サーバーに問い合わせることなく、キャッシュからドメインのアドレスレコードを提供します。

アプライアンスは、レコードの存続時間（TTL）値が設定値に達すると、キャッシュに格納されたレコードを破棄します。期限切れのレコードを要求するクライアントは、Citrix ADC がサーバーからレコードを取得し、キャッシュを更新するまで待機する必要があります。この遅延を回避するために、Citrix ADC は、レコードの有効期限が切れる前にサーバーからレコードを取得することによって、キャッシュを積極的に更新します。

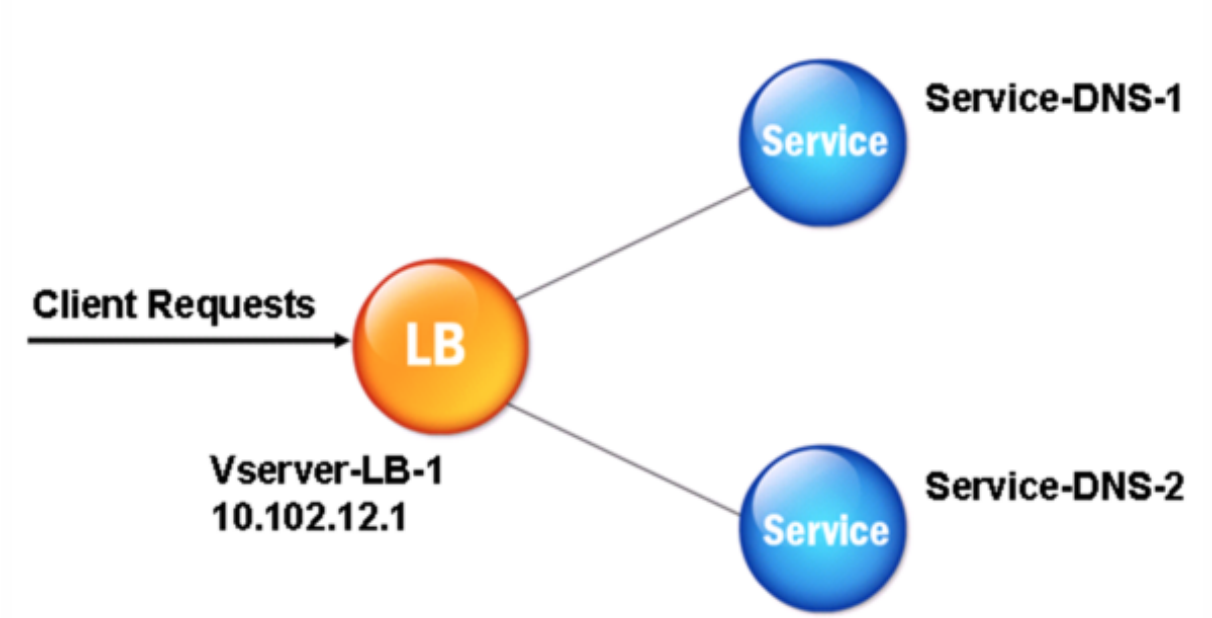
次の表に、Citrix ADC 上で構成する必要のあるエンティティの名前の例と値を示します。

表 1. DNS プロキシエンティティの設定例

エンティティタイプ	名前	IP アドレス	種類	ポート
LB virtual server	Vserver-DNS-1	10.102.29.40	DNS	53
サービス	Service-DNS-1	10.102.29.50	DNS	53
サービス	Service-DNS-2	10.102.29.51	DNS	53

次の図は、DNS プロキシのエンティティと、Citrix ADC 上で構成されるパラメーターの値を示しています。

図 2：DNS プロキシエンティティモデル



> 注
>
> DNS プロキシ機能を構成するには、負荷分散サービスと仮想サーバーを構成する方法を知っている必要があります。

負荷分散仮想サーバーの作成

Citrix ADC で DNS プロキシを構成するには、DNS タイプの負荷分散仮想サーバーを構成します。再帰クエリをサポートする DNS サーバーのセットを負荷分散するように DNS 仮想サーバーを構成するには、[再帰使用可能] オプションを設定する必要があります。このオプションを使用すると、DNS 仮想サーバからの DNS 応答で RA ビットが ON に設定されます。

負荷分散仮想サーバーの作成手順については、負荷分散を参照してください。

DNS サービスの作成

タイプ DNS の負荷分散仮想サーバーを作成したら、DNS サービスを作成する必要があります。DNS サービスを追加、変更、有効化、無効化、および削除できます。DNS サービスの作成手順については、「負荷分散」を参照してください。

負荷分散仮想サーバーを **DNS** サービスにバインドする

DNS プロキシ構成を完了するには、DNS サービスを負荷分散仮想サーバーにバインドする必要があります。サービスを負荷分散仮想サーバーにバインドする手順については、負荷分散を参照してください。

TCP を使用するように DNS プロキシのセットアップを構成する

一部のクライアントは、DNS 通信にユーザーデータグラムプロトコル (UDP) を使用します。ただし、UDP は最大パケットサイズを 512 バイトに指定しています。ペイロードの長さが 512 バイトを超える場合、クライアントは TCP を使用する必要があります。クライアントが Citrix ADC アプライアンスに DNS クエリを送信すると、アプライアンスはいずれかのネームサーバーにクエリを転送します。応答が UDP パケットに対して大きすぎる場合、ネームサーバーは Citrix ADC に対する応答にトランケーションビットを設定します。切り捨てビットは、応答が UDP に対して大きすぎること、クライアントが TCP 接続を介してクエリを送信する必要があることを示します。ADC アプライアンスは、切り捨てビットをそのままにして応答をクライアントに中継します。クライアントがポート 53 で DNS 負荷分散仮想サーバーの IP アドレスとの TCP 接続を開始するのを待ちます。クライアントは、TCP 接続を介して要求を送信します。次に、Citrix ADC アプライアンスは要求をネームサーバーに転送し、応答をクライアントに中継します。

DNS に TCP プロトコルを使用するように Citrix ADC を構成するには、DNS_TCP タイプの負荷分散仮想サーバーとサービスを構成する必要があります。DNS_TCP タイプのモニタを構成して、サービスの状態を確認することができます。DNS_TCP 仮想サーバー、サービス、およびモニターの作成手順については、負荷分散を参照してください。

レコードをプロアクティブに更新するために、Citrix ADC はサーバーへの TCP 接続を使用してレコードを取得します。

> 重要
>
> DNS に UDP を使用し、UDP のペイロード長が 512 バイトを超える場合にのみ TCP を使用するように Citrix ADC を構成するには、DNS と DNS_TCP サービス。の IP アドレス DNS_TCP service は、DNS サービスの IP アドレスと同じである必要があります。

DNS エントリの有効期間値を構成する

TTL は、ドメイン名およびレコードタイプが同じすべての DNS レコードで同じです。レコードの 1 つで TTL 値が変更されると、新しい値が同じドメイン名とタイプのすべてのレコードに反映されます。デフォルトの TTL 値は 3600 秒です。最小値は 0 で、最大値は 604800 です。DNS エントリの TTL 値が最小値よりも小さいか、最大値よりも大きい場合は、それぞれ最小 TTL 値または最大 TTL 値として保存されます。

CLI を使用して最小および最大 TTL を指定します

Citrix ADC コマンドプロンプトで次のコマンドを入力して、TTL の最小値と最大値を指定し、構成を確認します。

```
1 -  set dns parameter [-minTTL <secs>] [-maxTTL <secs>]
2 -  show dns parameter
3 <!--NeedCopy-->
```

例:


```
> set dns parameter -minTTL 1200 -maxTTL 1800
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        Minimum TTL: 1200               Maximum TTL: 1800
            .
            .
            .
 Done
>
<!--NeedCopy-->
```

GUI を使用して最小および最大 **TTL** を指定します

1. [トラフィック管理] **> [DNS]** に移動します。
2. 詳細ウィンドウの [設定] で、[DNS 設定の変更] をクリックします。
3. [DNS パラメーターの構成] ダイアログボックスの [TTL] の [最小] ボックスと [最大] ボックスに、それぞれ最小有効期間 (秒単位) と最大有効期間 (秒単位) を入力し、[OK] をクリックします。

メモ**:** TTL の有効期限が切れると、レコードはキャッシュから削除されます。Citrix ADC は事前にサーバーに接続し、DNS レコードの有効期限が切れる直前に DNS レコードを取得します。

DNS レコードをフラッシュする

キャッシュに存在するすべての DNS レコードを削除できます。たとえば、変更が行われた後にサーバーを再起動したときに、DNS レコードをフラッシュできます。

CLI を使用してすべてのプロキシレコードを削除する

Citrix ADC コマンドプロンプトで、次のように入力します。

```
flush dns proxyRecords
```

GUI を使用してすべてのプロキシレコードを削除する

1. **[**トラフィック管理**] > [DNS] > [**レコード**]** に移動します。
2. 詳細ペインで、[プロキシレコードのフラッシュ] をクリックします。

DNS リソースレコードを追加する

Citrix ADC アプライアンスが DNS プロキシサーバーとして構成されているドメインに DNS レコードを追加できます。DNS レコードの追加の詳細については、「DNS リソースレコードの設定」を参照してください。

負荷分散 DNS 仮想サーバーを削除する

負荷分散仮想サーバーの削除については、負荷分散を参照してください。

クライアント接続での同時 DNS 要求の数を制限する

<clientip:port>-<vserver ip:port> タプルで識別される、単一のクライアント接続での同時 DNS 要求の数を制限できます。同時 DNS リクエストとは、Citrix ADC アプライアンスがネームサーバーに転送し、アプライアンスが応答を待機しているリクエストです。クライアント接続での同時要求の数を制限すると、敵対的なクライアントが大量の DNS 要求を送信して分散型サービス拒否 (DDoS) 攻撃を試みたときに、ネームサーバーを保護できます。クライアント接続の制限に達すると、未処理の要求数が制限を下回るまで、接続での後続の DNS 要求はドロップされます。この制限は、Citrix ADC アプライアンスがキャッシュから送信される要求には適用されません。

このパラメーターのデフォルト値は 255 です。ほとんどのシナリオでは、この既定値で十分です。ネームサーバーが通常の動作条件下で多数の同時 DNS 要求を処理する場合は、大きな値またはゼロ（0）の値を指定できます。値 0 を指定すると、この機能が無効になり、1 つのクライアント接続で許可される DNS 要求の数に制限はありません。このパラメーターはグローバルパラメーターであり、Citrix ADC アプライアンスで構成されているすべての DNS 仮想サーバーに適用されます。
このパラメーターのデフォルト値は 255 です。ほとんどのシナリオでは、この既定値で十分です。ネームサーバーが通常の動作条件下で多数の同時 DNS 要求を処理する場合は、大きな値またはゼロ（0）の値を指定できます。値 0 を指定すると、この機能が無効になり、1 つのクライアント接続で許可される DNS 要求の数に制限はありません。このパラメーターはグローバルパラメーターであり、Citrix ADC アプライアンスで構成されているすべての DNS 仮想サーバーに適用されます。
このパラメーターのデフォルト値は 255 です。ほとんどのシナリオでは、この既定値で十分です。ネームサーバーが通常の動作条件下で多数の同時 DNS 要求を処理する場合は、大きな値またはゼロ（0）の値を指定できます。値 0 を指定すると、この機能が無効になり、1 つのクライアント接続で許可される DNS 要求の数に制限はありません。このパラメーターはグローバルパラメーターであり、Citrix ADC アプライアンスで構成されているすべての DNS 仮想サーバーに適用されます。

CLI を使用して、単一のクライアント接続で許可される同時 DNS 要求の最大数を指定する

コマンドプロンプトで次のコマンドを入力して、単一のクライアント接続で許可される同時 DNS 要求の最大数を指定し、構成を確認します。

```
1 -  set dns parameter -maxPipeline <positive_integer>
```



```
-  show dns parameter
<!--NeedCopy-->
```

例:

```
> set dns parameter -maxPipeline 1000
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        .
        .
        .
        Max DNS Pipeline Requests: 1000
 Done
<!--NeedCopy-->
```

GUI を使用して、**1** つのクライアント接続で許可される同時 **DNS** 要求の最大数を指定する

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細ウィンドウで、[DNS 設定の変更] をクリックします。
3. [DNS パラメーターの構成] ダイアログボックスで、[DNS パイプライン要求の最大数] の値を指定します。
4. [OK] をクリックします。

Citrix ADC をエンドリゾルバとして構成する

October 7, 2021

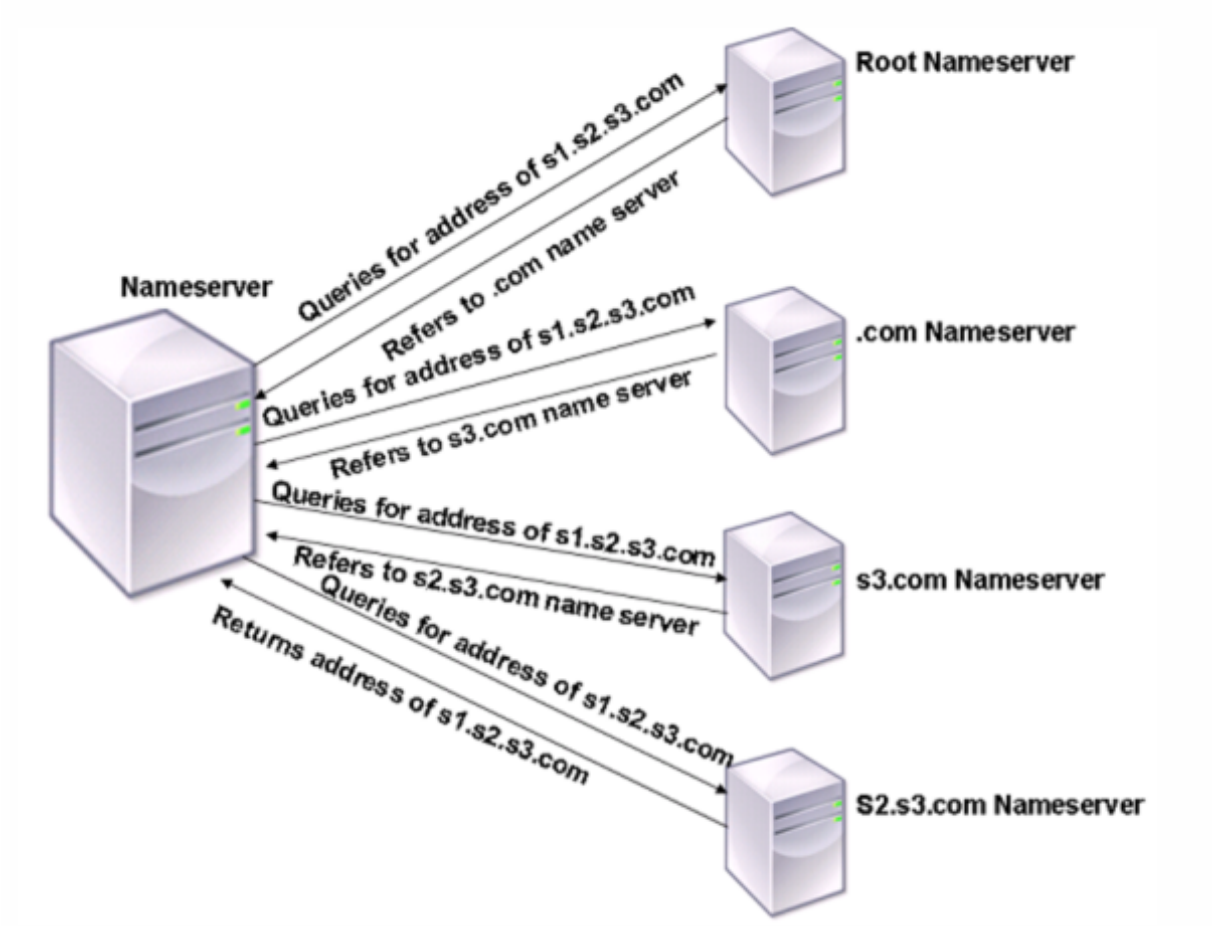
リゾルバは、ドメイン/ホスト名をリソースレコードに変換するアプリケーションプログラムによって呼び出されるプロシージャです。リゾルバは LDNS と対話します。LDNS は、ドメイン名を検索してその IP アドレスを取得します。Citrix ADC は、DNS クエリに対してエンドツーエンドの解決を提供できます。

再帰的な解決では、Citrix ADC アプライアンスは異なるネームサーバーに再帰的にクエリを行い、ドメインの IP アドレスにアクセスします。Citrix ADC が DNS 要求を受信すると、そのキャッシュで DNS レコードがチェックされます。レコードがキャッシュに存在しない場合、ns.conf ファイルに構成されているルートサーバーに問い合わせます。ルートネームサーバーは、第 2 レベルのドメインに関する詳細情報を持つ DNS サーバーのアドレスを報告します。このプロセスは、必要なレコードが見つかるまで繰り返されます。

Citrix ADC アプライアンスを初めて起動すると、13 台のルートネームサーバーが ns.conf ファイルに追加されます。13 台のルートサーバの NS レコードとアドレスレコードも追加されます。ns.conf ファイルは変更できますが、

Citrix ADC では 13 レコードすべてを削除することはできません。アプライアンスが名前解決を実行するには、少なくとも 1 つのネームサーバーエントリが必要です。次の図は、名前解決のプロセスを示しています。

図 1: 再帰的解決



図に示すプロセスでは、ネームサーバーが s1.s2.s3.com のアドレスに対するクエリを受信すると、最初に s1.s2.s3.com のルートネームサーバーをチェックします。ルートネームサーバは、.com ネームサーバのアドレスをレポートします。s1.s2.s3.com のアドレスがネームサーバにある場合は、適切な IP アドレスで応答します。それ以外の場合は、s3.com を他のネームサーバーに照会し、s2.s3.com に対して s1.s2.s3.com のアドレスを取得します。この方法では、解決は常にルートネームサーバーから始まり、ドメインの権限ネームサーバーで終わります。

注: 再帰的解決機能を使用するには、キャッシュを有効にする必要があります。

再帰的な解決を有効にする

Citrix ADC アプライアンスをエンドリゾルバとして機能するように設定するには、アプライアンスで再帰的な解決を有効にする必要があります。

CLI を使用した再帰的な解決の有効化

コマンドプロンプトで次のコマンドを入力して、再帰的な解決を有効にし、構成を確認します。

```
-  set dns parameter -recursion ENABLED
-  show dns parameter
<!--NeedCopy-->
```

例**:**

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
        DNS parameters:
            .
            .
            .
        Recursive Resolution : ENABLED
            .
            .
            .
 Done
<!--NeedCopy-->
```

GUI を使用した再帰的な解決の有効化

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細ウィンドウの [設定] で、[DNS 設定の変更] をクリックします。
3. [DNS パラメーターの構成] ダイアログボックスで、[再帰を有効にする] チェックボックスをオンにし、[OK] をクリックします。

再試行回数の設定

クエリの送信先のサーバーから応答を受信しない場合に、事前に構成された回数の試行（DNS 再試行と呼ばれる）を行うように ADC アプライアンスを構成します。デフォルトでは、DNS リトライ回数は 5 に設定されています。

CLI を使用した **DNS** 再試行回数の設定

コマンドプロンプトで次のコマンドを入力して、再試行回数を設定し、構成を確認します。


```
-  set dns parameter -retries <positive_integer>
-  show dns parameter
<!--NeedCopy-->
```

例:

```
> set DNS parameter -retries 3
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 3
            .
            .
            .
 Done
<!--NeedCopy-->
```

GUI を使用した再試行回数の設定

1. [トラフィック管理] **> [DNS]** に移動します。
2. 詳細ウィンドウの [設定] で、[DNS 設定の変更] をクリックします。
3. [DNS パラメーターの構成] ダイアログボックスの [DNS 再試行] ボックスに、DNS リゾルバー要求の再試行回数を入力し、[OK] をクリックします。

Citrix ADC アプライアンスをフォワーダーとして構成する

June 1, 2022

フォワーダーは、フォワーダーサーバーのネットワーク外にある DNS サーバーに DNS クエリを転送するサーバーです。ローカルで解決できないクエリは、他の DNS サーバーに転送されます。フォワーダーは、DNS クエリを解決するときに外部 DNS 情報をキャッシュに蓄積します。Citrix ADC アプライアンスをフォワーダーとして構成するには、外部ネームサーバーを追加する必要があります。

Citrix ADC アプライアンスでは、ローカルで解決できない名前解決クエリを転送できる外部ネームサーバーを追加できます。Citrix ADC アプライアンスをフォワーダーとして構成するには、名前解決クエリの転送先となるネームサーバーを追加する必要があります。ルックアップの優先順位を指定して、Citrix ADC アプライアンスが名前解決に使用する必要のあるネームサービスを指定できます。

Citrix ADC アプライアンスをフォワーダーとして構成する方法については、CLI を使用したネームサーバーの追加（Citrix ADC アプライアンスがフォワーダーとして機能する場合）を参照してください。

> 注:
>
> フォワーダーモードの Citrix ADC アプライアンスは、TCP、UDP、および UDP-TCP ネームサーバーをサポートします。
>
> - TCP ネームサーバーを構成した場合、Citrix ADC アプライアンスは TCP 経由で DNS 要求を送信します。
> - UDP ネームサーバーを構成した場合、Citrix ADC アプライアンスは UDP 経由で DNS 要求を送信します。
> - UDP-TCP ネームサーバーを構成した場合、Citrix ADC アプライアンスは UDP 経由で DNS 要求を送信します。ただし、切り捨てられたビットが DNS 応答に設定されている場合、アプライアンスはそのような DNS 要求を TCP 経由で送信します。

ネームサーバの追加

October 7, 2021

ネームサーバを作成するには、IP アドレスを指定するか、既存の仮想サーバをネームサーバとして設定します。

- **IP** アドレスベースのネームサーバ -ドメイン名解決のために連絡する外部ネームサーバ。アプライアンス上に複数の IP アドレスベースのネームサーバが設定されていて、いずれにもローカルパラメータが設定されていない場合、着信 DNS クエリーはすべてのネームサーバ間でラウンドロビン方式で負荷分散されます。
- 仮想サーバーベースのネームサーバー -Citrix ADC で構成された DNS 仮想サーバー。外部 DNS ネームサーバーの負荷分散方法をよりきめ細かく制御するには（たとえば、ラウンドロビン以外の負荷分散方法が必要な場合）、次の手順を実行します。
- アプライアンスでの DNS 仮想サーバーの構成
- 外部ネームサーバーをサービスとしてバインドする
- このコマンドで仮想サーバの名前を指定します。

設定を確認するには、`show dns nameServer` コマンドを使用します。

ネームサーバーを削除するには、Citrix ADC CLI で、`rm dns nameServer` コマンドに続けてネームサーバーの IP アドレスを入力します。

DNS ネームサーバーの詳細を表示するには、Citrix ADC CLI で、`show dns nameServer` コマンドに続けてネームサーバーの IP アドレスを入力します。

CLI を使用してネームサーバーを追加します（**Citrix ADC** アプライアンスがフォワーダとして機能する場合）

コマンドプロンプトで次を入力します。


```
add dns nameServer ((<IP>) | <dnsVserverName>)
<!--NeedCopy-->
```

または

```
add dns nameServer ((<IP> | <dnsVserverName>) [-type <type>]
<!--NeedCopy-->
```

例:

```
add dns nameServer dnsVirtualNS

add dns nameServer 192.0.2.11 -type TCP

add dns nameServer 192.0.2.12 -type UDP_TCP


add dns nameServer 192.0.2.10
show dns nameServer 192.0.2.10

1) 192.0.2.10  -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

> 注:
>
> ネームサーバタイプが指定されていない場合は、デフォルトで UDP ネームサーバが作成されます。TCP または UDP_TCP タイプのネームサーバーを作成するには、タイプを指定する必要があります。
>
> タイプを UDP_TCP として指定すると、指定された IP アドレスに対して 2 つのネームサーバ（1 つの UDP ネームサーバと 1 つの TCP ネームサーバ）が作成されます。

CLI を使用してネームサーバーを追加します（**Citrix ADC** アプライアンスがリゾルバとして機能する場合）

コマンドプロンプトで入力します。

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>)
<!--NeedCopy-->
```


例:

```
add dns nameServer 10.102.9.19 -local
show dns nameServer
1)  10.102.9.19 LOCAL -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

ローカル -Citrix ADC アプライアンス上のローカル再帰 DNS サーバーに属する IP アドレスをマークします。アプライアンスは、ローカルとしてマークされた IP アドレスで受信したクエリーを再帰的に解決します。
再帰的な解決が機能するには、グローバル DNS パラメータ`recursion`も設定する必要があります。
ローカルとしてマークされているネーム・サーバがない場合、アプライアンスはスタブ・リゾルバとして機能し、ネーム・サーバのロード・バランシングを行います。

GUI を使用してネームサーバーを追加する

[トラフィック管理] > [**DNS**] > [ネームサーバー] に移動し、ネームサーバーを作成します。

DNS ルックアップの優先順位を設定する

October 7, 2021

検索の優先順位は、DNS または WINS のいずれかに設定できます。このオプションは、SSL VPN モードの動作で使用されます。

CLI を使用してルックアップの優先順位を **DNS** に設定する

コマンドプロンプトで次のコマンドを入力して、検索の優先順位を DNS に設定し、構成を確認します。

```
-  set dns parameter -nameLookupPriority (DNS | WINS)
-  show dns parameter
<!--NeedCopy-->
```

例:

```
> set dns parameter -nameLookupPriority DNS
 Done
```



```
> show dns parameter
                .
                .
                .
          Name lookup priority : DNS
                .
                .
                .
 Done
<!--NeedCopy-->
```

GUI を使用してルックアップの優先順位を DNS に設定する

1. [トラフィック管理] > [DNS] に移動します。
2. 詳細ウィンドウの [設定] で、[DNS 設定の変更] をクリックします。
3. [DNS パラメータの構成] ダイアログボックスの [名前参照の優先度] で、[DNS] または [WINS] を選択し、[OK] をクリックします。

> 注
>
> 構成した DNS 仮想サーバーがダウンしていて、-nameLookupPriorityを DNS に設定した場合、Citrix ADC は WINS ルックアップを試行しません。したがって、DNS 仮想サーバが構成されていない、または無効になっている場合は、-nameLookupPriorityを WINS に設定します。

ネームサーバーの無効化と有効化

October 7, 2021

次の手順では、既存のネームサーバーを有効または無効にする手順について説明します。

CLI を使用したネームサーバーの有効化または無効化

コマンドプロンプトで次のコマンドを入力して、ネームサーバーを有効または無効にし、構成を確認します。

```
-  (enable | disable) dns nameServer <IPAddress>
-  show dns nameServer <IPAddress>
<!--NeedCopy-->
```

例:


```
> disable dns nameServer 10.102.9.19
 Done
> show dns nameServer 10.102.9.19
1)       10.102.9.19: LOCAL  -  State: OUT OF SERVICE
 Done
<!--NeedCopy-->
```

GUI を使用したネームサーバーの有効化または無効化

1. **Traffic Management > DNS > Name Servers** に移動します。
2. 詳細ウィンドウで、有効または無効にするネームサーバーを選択します。
3. [有効] または [無効] をクリックします。ネームサーバが有効になっている場合は、[Disable] オプションを使用できます。ネームサーバが無効になっている場合は、[Enable] オプションを使用できます。

Citrix ADC を非検証セキュリティ対応のスタブリゾルバとして構成する

October 7, 2021

Citrix ADC 12.1 ビルド 49.xx 以降、Citrix ADC は非検証のセキュリティ対応スタブリゾルバーとして機能します。このサポートを有効にするには、DNS ヘッダーに AD ビットが設定され、OPT ヘッダーに DO ビットが設定解除されます。AD ビットが設定されていて DO ビットが設定されていない場合、アップストリーム再帰リゾルバは DNSSEC 応答を検証します。検証が成功すると、再帰リゾルバは DNSSEC RR なしで応答します。DNSSEC 検証が失敗した場合、再帰リゾルバは SERVFAIL 応答を返します。

> 重要:
>
> AD ビットは、ADC フォワーダにデフォルトで設定されます。AD ビットは、DBS によって開始されたクエリには設定されません。

大きなサイズの応答を処理するための **DNS** のジャンボフレームのサポート

October 7, 2021

Citrix ADC 12.1 ビルド 49.xx 以降、DNS は 1,280 バイトを超える UDP 応答を処理するためのジャンボフレームをサポートしています。以前は、Citrix ADC アプライアンスでサポートされる UDP パケットサイズは最大 1,280 バイトまででした。

プロキシモード、ADNS モード、フォワーダモードでアプライアンスが処理できる最大 UDP パケットサイズを設定するには、[Maximum UDP Packet Size] パラメータ値を設定します。たとえば、最大 UDP パケットサイズパラメータ値が 4096 に設定されている場合、アプライアンスはサイズ 4,096 バイトの DNS 応答を処理できます。

> 重要
>
> - プロキシモードでは、DNS クエリをバックエンドに送信するために、クライアント要求の OPT ペイロードサイズと最大 UDP パケットサイズの値の間の最小サイズが考慮されます。たとえば、クライアント要求の OPT ペイロードサイズが 3000 で、最大 UDP パケットサイズの値が 4096 の場合、3,000 バイトの DNS クエリがバックエンドに送信されます。
>
>   また、アプライアンスはバックエンドから、大きなサイズの応答を受信し、大きなサイズの応答を処理できます。
>
> - フォワーダモードでは、アプライアンスは OPT ペイロードサイズを UDP パケットサイズパラメータ値と同じ値に設定します。
> - DNS レコードがアプライアンスに対してローカルである場合、アプライアンスは最大 UDP パケットサイズパラメータ値と同じ大きさの応答サイズを構成できます。この設定は、ADNS、プロキシ、および再帰リゾルバーに適用できます。

CLI を使用して UDP パケットの最大サイズを設定するには

コマンドプロンプトで入力します。

```
set dns parameter [-maxUDPPacketSize <positive_integer>]
<!--NeedCopy-->
```

例:

```
set dns parameter -maxUDPPacketSize 10000
<!--NeedCopy-->
```

> 注: [UDP パケットの最大サイズ] パラメータに設定できる最小値と最大値は、それぞれ 512 と 16384 です。デフォルト値は 1280 です。

CLI を使用して UDP パケットの最大サイズを設定するには

1. **Traffic Management** > **DNS** に移動します。
2. 詳細ウィンドウで、[**DNS** 設定の変更] をクリックします。
3. [最大 UDP パケットサイズ] で、最大 UDP パケットサイズを指定します。
4. [**OK**] をクリックします。

DNS ログを構成する

October 7, 2021

Citrix ADC アプライアンスは、処理する DNS 要求と応答を記録するように構成できます。アプライアンスは、DNS 要求と応答を SYSLOG 形式で記録します。DNS 要求または DNS 応答、またはその両方をログに記録し、syslog メッセージをリモートログサーバに送信するように選択できます。ログメッセージは、次の目的で使用できます。

- クライアントに対する DNS 応答の監査
- DNS クライアントの監査
- DNS 攻撃の検出と防止
- トラブルシューティング

Citrix ADC アプライアンスは、構成に基づいて、DNS 要求または応答に次のセクションを記録できます。

- ヘッダーセクション
- 質問セクション
- 回答セクション
- 権限セクション
- 追加セクション

DNS プロファイル

DNS プロファイルを使用して、DNS エンドポイントが DNS トラフィックに適用するさまざまな DNS パラメーターを構成できます。プロファイルでは、ロギング、キャッシング、およびネガティブキャッシングを有効にできます。

重要： NetScaler 11.0 リリース以降、グローバル DNS パラメーターを使用した DNS キャッシュの有効化は非推奨になりました。DNS プロファイルを使用して、DNS キャッシュを有効または無効にできます。DNS プロファイルで DNS キャッシュを有効にし、DNS プロファイルを個々の仮想サーバーに設定することで、個々の仮想サーバーの DNS キャッシュを有効にできるようになりました。

DNS プロファイルは、次の種類の DNS ログをサポートしています。

- DNS クエリーロギング
- DNS 応答セクションのロギング
- DNS 拡張ロギング
- DNS エラーロギング

DNS クエリーロギング

Citrix ADC アプライアンスは、アプライアンス上の DNS エンドポイントが受信した DNS クエリのみをログに記録するように構成できます。

注: クエリの処理中にエラーが発生した場合、DNS プロファイルでこのオプションが設定されていると、エラーがログに記録されます。

次に、クエリログメッセージの例を示します。

```
DNS DNS_QUERY 143 0 :  U:10.102.27.70#61297:10.102.27.73#53/22142/Q/
(RD)/NO/1/0/0/0#test.com./1#
<!--NeedCopy-->
```

DNS 応答セクションロギング

アプライアンスがクライアントに送信する DNS 応答のすべての 回答セクションをログに記録するように Citrix ADC アプライアンスを構成できます。DNS Answer Section ロギングは、Citrix ADC が DNS リゾルバーとして構成されている場合、または GLSB のユースケースで役立ちます。

DNS 応答セクションログの例を次に示します。

```
DNS DNS_RESPONSE 6678 0 :  U:100.100.100.210#32776:100.100.100.10#
53/61373/Q/(RD,AA,RA,R)/NO/1/1/2/4#n1.citrix.com1./
28#ANS#AAAA/120/1111:2345:6789:ffab:abcd:effa:1234:3212##
<!--NeedCopy-->
```

DNS 拡張ロギング

DNS 応答の権限セクションと 追加セクションをログに記録するように Citrix ADC アプライアンスを構成するには、応答セクションログを使用して拡張ログを有効にします。

注: クエリまたは応答の処理中にエラーが発生した場合、DNS プロファイルでこのオプションが設定されていると、エラーがログに記録されます。

次に、キャッシュルックアップが完了し、応答がパケットに埋め込まれているときにログに記録されるメッセージの例を示します。

```
DNS DNS_RESPONSE 2252 0 :  T:100.100.100.118#21411:100.100.100.10
#53/48537/Q/(RD,AA,CD,RA,R)/NO/1/1/2/6#a1.citrix.com1./1#ANS#A/
120/1.1.1.1##AUTH#citrix.com1/NS/120/n2.citrix.com1#n1.citrix.com1##ADD
    #n1.citrix.com1
/A/120/1.1.1.1#1.1.1.2##n1.citrix.com1/AAAA/120/
1111:2345:6789:ffab:abcd:effa:1234:3212##n2.citrix.com1/A/120/2.1.1.2
##n2.citrix.com1/AAAA/120/2222:faff:3212:8976:123:1241:64:ff9b##OPT
    /0/1280/DO##
<!--NeedCopy-->
```


DNS エラーログ

Citrix ADC アプライアンスは、DNS クエリまたは応答を処理するときに発生するエラーまたは障害をログに記録するように構成できます。これらのエラーの場合、アプライアンスは DNS ヘッダー、質問セクション、および OPT レコードをログに記録します。

次に、DNS 要求または応答の処理中にエラーが発生したときに記録されるメッセージの例を示します。

```
DNS DNS_ERROR 149 0 :  U:10.102.27.70#27832:10.102.27.73#53/61153/Q/
(RD)/NO/1/0/0/0#test.com./1140#Packet Dropped
<!--NeedCopy-->
```

ポリシーベースのロギング

DNS ポリシー、書き換え、またはレスポンダーポリシーで logAction を構成することで、DNS 式に基づいてカスタムログを構成できます。特定の DNS ポリシーが true と評価された場合にのみロギングが行われるように指定できます。詳細については、「DNS のポリシーベースのログを構成する」を参照してください。

Citrix ADC Syslog ログメッセージの形式を理解する

Citrix ADC アプライアンスは、DNS 要求と応答を次の Syslog 形式で記録します。

```
<transport> :<client IP>#<client ephemeral port>:<DNS endpoint IP>#<
   port>
 : <query id> /opcode/header flags/rcode/question section count/answer
    section count
/ auth section count / additional section count #<queried domain name>
/<queried type>#...
<!--NeedCopy-->
```

- **<transport>:**
  - T = TCP
  - U = UDP
- **<client IP>#< client ephemeral port >:** DNS クライアントの IP アドレスとポート番号
- **<DNS endpoint IP># <port>:** Citrix ADC DNS エンドポイントの IP アドレスとポート番号
- **<query id>:**
  クエリ ID
- **<opcode>:** 作業コード。サポートされる値:

  - Q: クエリ
  - I: 逆クエリ
  - S: ステータス
  - X0: 未割り当て
  - N: 通知する
  - U: 更新
  - X1-10: 未割り当ての値

- **<header flags>:** フラグ。サポートされる値:

  - RD：再帰が必要
  - TC：切り捨て
  - AA：信頼できる対応
  - CD：チェックが無効
  - AD：認証されたデータ
  - Z: 未割り当て
  - RA：再帰が利用可能
  - R: 応答

- **<rcode>:** 応答コード。サポートされる値:

  - いいえ：エラーなし
  - F フォーマットエラー
  - S: サーバー障害
  - NX：存在しないドメイン
  - NI：実装されていません
  - R: クエリが拒否されました
  - YX：名前は存在してはならないときに存在します
  - YXR：RR セットは存在してはならないときに存在します
  - NXR：存在しなければならない RR セットは存在しません
  - NAS：サーバーはゾーンに対して権限がありません
  - NA：許可されていません
  - NZ：名前がゾーンに含まれていません
  - X1-5: 未割り当て

- /question セクション count/answer セクション count/auth セクション count/additional セクション数：DNS 要求の質問セクション、権限セクション数、および 追加セクション数

- **<queried domain name>/ <queried type>:**DNS 要求でクエリされたドメインとクエリされたタイプ

- **#ANS # <record type>/<ttl> /.. #AUTH # <domain name>/<record type>/<ttl>.. #ADD #**<domain name>/<record type>/<ttl>…:

  DNS 応答の場合:

DNS プロファイルで応答セクションのログが有効になっている場合は、応答セクションがログに記録されます。DNS プロファイルで拡張ログが有効になっている場合、権限セクションと 追加セクションがログに記録されます。ログ形式は、レコードの種類によって異なります。詳細については、「レコードログ形式について」を参照してください。

- ANS: 回答セクション
- 自動: 権限
- 追加：追加セクション

- **OPT/<edns version>/UDP max payload size/DO:** DNS ログの OPT レコード形式
- **OPT/<EDNS version>/<UDP payload size>/<“DO”or empty based on whether DNSSEC OK bit is set or not>/<value of RDLEN>/ECS/<Q/R>/<option length>/<Family>/<Source Prefix-Length>/<Scope Prefix-Length>/<ECS Address>:**

  DNS クエリまたは応答に EDNS クライアントサブネット (ECS) オプションが含まれている場合は、OPT レコード形式で DNS ログファイルに記録されます。

  IPv4 または IPv6 アドレスのいずれかを含む ECS オプションを使用した DNS クエリが送信されると、ECS オプションは次のいずれかのオプションでログに記録されます。

  - “ECS/Q” ログの値がクエリからのものであることを示します
  - “ECS/R” ログの値が応答からのものであることを示します。

  [スコーププレフィックス-長さ] の値も適切に設定されます。DNS クエリではゼロに設定され、応答の場合は計算値に設定されます。

  次の表では、さまざまなシナリオでログに記録される詳細について説明します。

シナリオ	DNS クエリに設定された ECS オプション	DNS 応答に設定された ECS オプション	ログに記録された詳細
クエリー・ロギングと拡張ロギングの両方が有効になっている	はい	はい	ECS オプションは、文字列「ECS/R/」でログに記録され、スコーププレフィックス-長さが計算された値に設定されます。
クエリー・ロギングと拡張ロギングの両方が有効になっている	はい	いいえ	ECS オプションは、文字列「ECS/Q」でログに記録され、スコーププレフィックス-長さがゼロに設定されます。

シナリオ	DNS クエリに設定された ECS オプション	DNS 応答に設定された ECS オプション	ログに記録された詳細
クエリー・ロギングは有効だが、拡張ロギングは有効にならない	はい	はい	ECS オプションは、文字列「ECS/Q/」でログに記録され、スコーププレフィクス-長さがゼロに設定されます。
クエリログと拡張ログが有効になっていません	はい	はい	ECS オプションはログに記録されません。
クエリー・ロギングは有効だが、拡張ロギングは有効にならない	はい	いいえ	ECS オプションは、文字列「ECS/Q/」でログに記録され、スコーププレフィクス-長さがゼロに設定されます。
クエリー・ロギングは有効になっていないが、拡張ロギングは有効になっている	はい	はい	ECS オプションは、文字列「ECS/R/」でログに記録され、スコーププレフィックス-長さが計算された値に設定されます。
クエリー・ロギングは有効になっていないが、拡張ロギングは有効になっている	はい	いいえ	ECS オプションはログに記録されません。

レコードロギング形式を理解する

次に、Syslog メッセージのレコードロギング形式の例を示します。

```
1 <domainname>/<record type>/ <record ttl> / <resource record data>#<
     resource record data>#......##
2 <!--NeedCopy-->
```

各項目の意味は次のとおりです:

レコードタイプ	サンプル形式	リソースレコードデータ/形式
Address (A) レコード	A/5/1.1.1.1#1.1.1.2#1.1.1.3##	IPv4 アドレス

レコードタイプ	サンプル形式	リソースレコードデータ/形式
AAAA レコード	AAAA/5/1::1#1::2#1::3##	IPv6 アドレス
SOA レコード	SOA/3600/ns1.dnslogging.test./	元のサーバー、連絡先、およびその他の詳細。リソースレコードの形式:<originServer>//<contact>/<serial number>/<refresh rate>/<retry>//<expire>/<minimum>##
NS レコード	NS/5/ns1.dnslogging.test	ネームサーバーのホスト名。
MX レコード	#MX/5/10/host1.dnslogging.test	優先設定に続いてメール交換サーバーのホスト名
CNAME レコードロギング	CNAME/5/host1.dnslogging.test.#	標準名
SRV レコード	SRV/5/1/2/3/host1.dnslogging.te	リソースレコード形式:<priority>//<weight>/<port>/<target>#
TXT レコード	TXT/5/dns+logging##	データは、すべてのテキストを含みます。
NAPTR レコード	NAPTR/5/10/11////dnslogging#2	リソースレコード形式:<order>//<preference>/<glags>//<services>/<expression>/<replacement string> #
DNSKEY レコード	DNSKEY/5/1/3/5/AwEAAanP0K+i5bfv5Sx4	73L76OEjDjnPqI2Ccx6JZgiDBZhSON 形式:<glags>//<protocol>/<algorithm>/<public key in base64 encoding> #
PTR レコード	PTR/3600/test.com.#test4.com	ドメイン名

DNS ログ記録の制限

DNS ログには、次の制限があります。

- 応答ロギングが有効な場合、次のレコードタイプのみがログに記録されます。
  - Address (A) レコード
  - AAAA レコード
  - SOA レコード
  - NS レコード
  - MX レコード

  - CNAME レコード
  - SRV レコード
  - TXT レコード
  - NAPTR レコード
  - DNSKEY レコード
  - PTR レコード

  他のすべてのレコードタイプでは、L3/L4 パラメータ、DNS ヘッダー、および質問セクションのみがログに記録されます。

- RRSIG レコードは、応答ロギングが有効になっている場合でもログに記録されません。
- DNS64 はサポートされていません。
- DNS プロアクティブ更新要求または応答は、既定のプロファイルの設定に従ってログに記録されます。
- 仮想サーバーで、セッションレスオプションおよび応答ロギングが有効になっている場合、応答ではなく L3/L4 パラメータ、DNS ヘッダー、および DNS クエスチョンセクションがログに記録されます。
- syslog メッセージの最大サイズは 1024 バイトです。
- アクションタイプ RewriteResponse で DNS ポリシーの DNS プロファイルを設定した場合、Citrix ADC アプライアンスはクエリまたは操作された応答をログに記録しません。必要な情報をログに記録するには、DNS ポリシーで監査メッセージアクションを使用する必要があります。
- DNS 監視トラフィックが原因の DNS トランザクションはログに記録されません。

DNS ロギングの設定

DNS ロギングの設定の概要を次に示します。

1. Syslog アクションを作成し、アクションで DNS を有効にします。
2. Syslog ポリシーを作成し、ポリシーで Syslog アクションを指定します。
3. Syslog ポリシーをグローバルにバインドして、すべての Citrix ADC システムイベントのログを有効にします。または、特定の負荷分散仮想サーバーに Syslog ポリシーをバインドします。
4. DNS プロファイルを作成し、有効にする次のタイプのログのいずれかを定義します。
   - DNS クエリーロギング
   - DNS 応答セクションのロギング
   - DNS 拡張ロギング
   - DNS エラーロギング
5. 要件に応じて、次のいずれかを設定します。
   - DNS サービスおよび DNS 用の仮想サーバー
   - ADNS サービス
   - フォワーダとしての Citrix ADC
   - リゾルバとしての Citrix ADC
6. 作成した DNS プロファイルを DNS エンティティの 1 つに設定します。

CLI を使用して **DNS** プロキシとして構成された **Citrix ADC DNS** ログを構成する

1. syslog アクションを追加し、アクションで DNS を有効にします。コマンドプロンプトで入力します。

```
add audit syslogAction <name> (<serverIP> | -lbVserverName <string
    >) [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat <
    dateFormat>] [-logFacility <logFacility>] [-tcp ( NONE | ALL )]
     [-acl ( ENABLED | DISABLED )] [-timeZone ( GMT_TIME |
    LOCAL_TIME )] [-userDefinedAuditlog ( YES | NO )] [-
    appflowExport ( ENABLED |DISABLED )] [-lsn ( ENABLED | DISABLED
     )] [-alg ( ENABLED | DISABLED )] [-transport ( TCP | UDP )] [-
    tcpProfileName <string>] [-maxLogDataSizeToHold <
    positive_integer>] [-dns ( ENABLED | DISABLED)]
<!--NeedCopy-->
```

例:

```
add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
LOCAL_TIME -dns ENABLED
```

2. syslog ポリシーを作成し、作成した syslog アクションをポリシー内で指定します。コマンドプロンプトで入力します。

```
add audit syslogPolicy <name> <rule> <action>
```

例:

```
add audit syslogPolicy syslogpol1 ns_true nssyslogact1
```

3. syslog ポリシーをグローバルにバインドします。コマンドプロンプトで入力します。

```
bind system global [<policyName> [-priority <positive_integer>]]
```

例:

```
bind system global syslogpol1
```

4. DNS プロファイルを作成し、構成する次の種類のログのいずれかを有効にします。

   - DNS クエリーロギング
   - DNS 応答セクションのロギング
   - DNS 拡張ロギング
   - DNS エラーロギング

   コマンドプロンプトで入力します。

```
add dns profile <dnsProfileName> [-dnsQueryLogging ( ENABLED | DISABLED
 )] [-dnsAnswerSecLogging ( ENABLED | DISABLED )] [-dnsExtendedLogging
```


(ENABLED | DISABLED)] [-dnsErrorLogging (ENABLED | DISABLED)] [-cacheRecords (ENABLED | DISABLED)] [-cacheNegativeResponses (ENABLED | DISABLED)]

例:

```
add dns profile dnsprofile1 -dnsQueryLogging ENABLED
```

5. DNS タイプのサービスを構成します。コマンドプロンプトで入力します。

```
add service <name> <serverName> <serviceType> <port>
```

例:

```
add service svc1 10.102.84.140 dns 53
```

6. サービスタイプ DNS の負荷分散仮想サーバーを構成します。

```
add lb vserver <name> <serviceType> <ip> <port>
```

例:

```
add lb vserver lb1 dns 100.100.100.10 53
```

7. サービスを仮想サーバーにバインドします。コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
```

例:

```
bind lb vserver lb1 svc1
```

8. 作成した DNS プロファイルを仮想サーバーに設定します。コマンドプロンプトで入力します。

```
set lb vserver <name> [ - dnsProfileName <string>]
```

例:

```
set lb vserver lb1 -dnsProfileName dnsprofile1
```

DNS プロキシとして構成された Citrix ADC アプライアンスの DNS ログ設定例

```
1 > add audit syslogAction nssyslogact1 10.102.151.136 -logLevel
2 CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -
     timeZone
3 LOCAL_TIME -dns ENABLED
4 Done
5 > add audit syslogPolicy syslogpol1 ns_true nssyslogact1
6 Done
7 > bind system global syslogpol1
8 Done
9 > add dns profile dnsprofile1 -dnsqueryLogging ENABLED
```



```
Done
> add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
<!--NeedCopy-->
```

ADNS として構成された **Citrix ADC** アプライアンスの **DNS** ログ構成例

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
<!--NeedCopy-->
```

フォワーダーとして構成された **Citrix ADC** アプライアンスの **DNS** ログ設定の例

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
```



```
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add dns nameserver 8.8.8.8  - dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

リゾルバーとして構成された **Citrix ADC** アプライアンスの **DNS** ロギング構成の例

```
> add audit syslogAction nssyslogact1 10.102.151.136
-logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -
    logFacility LOCAL4
-timeZone LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> set dns parameter -recursion enABLED
Done
> add nameserver 1.1.1.100 -local dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

DNS のポリシーベースのログ記録を構成する

ポリシーベースのロギングでは、ログメッセージの形式を指定できます。ログメッセージの内容は、デフォルトの構文式を使用して定義されます。ポリシーで指定されたメッセージアクションが実行されると、Citrix ADC アプライアンスは式からログメッセージを作成し、ログファイルにメッセージを書き込みます。特定の DNS ポリシーが True と評価された場合にのみログを記録するようにアプライアンスを設定できます。

> 注
>
> 要求側の DNS プロファイルを使用して DNS ポリシーを設定した場合、Citrix ADC アプライアンスはクエリのみをログに記録します。

DNS ポリシーのポリシーベースのロギングを設定するには、まず監査メッセージアクションを設定する必要があります。監査メッセージアクションの構成の詳細については、「監査ログ用の NetScaler アプライアンスの構成」を参照してください。監査メッセージアクションを設定したら、DNS ポリシーでメッセージアクションを指定します。

CLI を使用して **DNS** ポリシーのポリシーベースのログ記録を構成する

コマンドプロンプトで次のコマンドを入力して、DNS ポリシーのポリシーベースのログを構成し、構成を確認します。

```
1 -  add dns action <actionName> <actionType> [-IPAddress <ip_addr|
     ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
     ...] [-TTL <secs>] [-dnsProfileName <string>]
2 -  set dns policy <name> [<rule>] [-actionName <string>] [-logAction <
     string>]
3 -  show dns policy [<name>]
4 <!--NeedCopy-->
```

例 **1**:

GSLB 展開で、特定のサブネットからのクライアント要求に異なる IP アドレスで応答する場合、一般的な目的で使用される IP アドレス (内部ユーザーの IP アドレスなど) で応答するのではなく、アクションタイプを DNS ビューとして DNS ポリシーを設定できます。この場合、特定の応答を記録できるように、指定された DNS アクションで DNS ロギングを設定できます。

```
1  > add dns profile dns_prof1 -dnsqueryLogging enABLED -
       dnsanswerSecLogging enABLED
2   Done
3  > add dns view dns_view1
4   Done
5  > add dns action dns_act1 viewName -view dns_view1  - dnsprofilename
       dns_prof1
6   Done
7  > add dns policy dns_pol1 ”CLIENT.IP.SRC.APPLY_MASK(255.255.255.0).EQ
       (100.100.100.0)”
8  dns_act1
9   Done
10 > bind dns global dns_pol1 100 -gotoPriorityExpression END -type
       REQ_DEFAULT
11  Done
12 > bind gslb service site_1_svc -viewName dns_view1 123.1.1.1
13  Done
14 > bind gslb service site_5_svc -view dns_view1 132.1.1.1
15  Done
16 <!--NeedCopy-->
```

注：前述の構成で、GSLB 仮想サーバー（たとえば、*sampletest.com*）で構成されたドメインを照会すると、サブ

ネットのすべての内部ユーザー 100.100.100.0/24 DNS ビューの IP アドレスが提供され、応答がログに記録されます。他のサブネットに対するクライアント要求はログに記録されません。

例 **2**:

ドメイン *example.com* のクエリのみをログに記録する場合は、クエリログを有効にした DNS プロファイルを作成し、アクションタイプ
NOOP で DNS プロファイルを DNS アクションに設定してから、DNS ポリシーを作成し、DNS アクションを設定します。次に例を示します:

```
1 >add dns profile query_logging -dnsqueryLogging ENABLED
2 Done
3 >add dns action dns_act1 NOOP -dnsprofileName query_logging
4 Done
5 >add dns policy dns_pol1 DNS.REQ.QUESTION.DOMAIN.EQ(”example.com”)
     dns_act1
6 Done
7 <!--NeedCopy-->
```

DNS サフィックスの設定

October 7, 2021

名前解決中に Citrix ADC アプライアンスが完全修飾されていないドメイン名を完成できるようにする DNS サフィックスを構成できます。たとえば、完全修飾されていないドメイン名 abc を解決するときに、DNS サフィックス example.com が構成されている場合、アプライアンスはサフィックスをドメイン名に追加します。次に、ドメイン名を解決します。この場合、abc.example.com を解決します。DNS サフィックスが設定されていない場合、アプライアンスは完全修飾されていないドメイン名にピリオドを追加し、ドメイン名を解決します。

DNS サフィックスの作成

DNS サフィックスは重要であり、Citrix ADC がエンドリゾルバまたはフォワーダとして構成されている場合にのみ有効です。最大 127 文字の接尾辞を指定できます。

注**:** DNS サフィックスの順序は重要です。ADC アプライアンスは、設定されたサフィックスをシリアル順に試み、サフィックスの応答が成功すると停止します。

CLI を使用して DNS サフィックスを作成する

コマンドプロンプトで次のコマンドを入力して、DNS サフィックスを作成し、構成を確認します。


```
-  add dns suffix <dnsSuffix>
-  show dns suffix <dnsSuffix>
<!--NeedCopy-->
```

例:

```
> add dns suffix example.com
 Done
> show dns suffix example.com
1)      Suffix: example.com
Done

<!--NeedCopy-->
```

Citrix ADC コマンドラインを使用して DNS サフィックスを削除するには、コマンドプロンプトで、`rm dns suffix` コマンドと DNS サフィックスの名前を入力します。

GUI を使用して **DNS** サフィックスを作成する

[トラフィック管理] **> [DNS] > [DNS** サフィックス] に移動し、DNS サフィックスを作成します。

DNS ANY クエリ

October 7, 2021

ANY クエリは、ドメイン名に使用できるすべてのレコードを取得する DNS クエリの一種です。ANY クエリは、ドメインに対して権限のあるネームサーバに送信する必要があります。

ADNS モードでの動作

ADNS モードでは、Citrix ADC アプライアンスはローカルキャッシュに保持されているレコードを返します。キャッシュにレコードがない場合、アプライアンスは NXDOMAIN（負）応答を返します。

Citrix ADC がドメイン委任レコードと一致する場合、NS レコードが返されます。それ以外の場合は、ルートドメインの NS レコードを返します。

DNS プロキシモードでの動作

プロキシモードでは、Citrix ADC アプライアンスはローカルキャッシュをチェックします。キャッシュにレコードがない場合、アプライアンスはクエリをサーバーに渡します。

グローバルサーバー負荷分散（**GSLB**）ドメインの動作

ADC アプライアンスで GSLB ドメインが構成されていて、GSLB（サイト）ドメインに対して ANY クエリが送信された場合、アプライアンスは GSLB サービスの IP アドレスを返します。負荷分散の決定を通じてこのサービスを選択します。マルチ IP 応答 (MIR) オプションが有効になっている場合は、すべての GSLB サービスの IP アドレスが送信されます。

Citrix ADC が ANY クエリに応答したときにこれらのレコードを返すようにするには、GSLB ドメインに対応するすべてのレコードが Citrix ADC 上で構成されている必要があります。

> 注
>
> ドメインのレコードが Citrix ADC とサーバー間で分散される場合、Citrix ADC で構成されたレコードのみが返されます。

Citrix ADC には、DNS ビューと DNS ポリシーを構成するオプションがあります。これらのビューとポリシーは、グローバルサーバーの負荷分散を実行するために使用されます。詳細については、「グローバルサーバーの負荷分散」を参照してください。

DNS レコードのネガティブキャッシュを構成する

October 7, 2021

Citrix ADC アプライアンスは、ドメインに対する否定的な応答のキャッシュをサポートしています。否定的な応答は、要求されたドメインに関する情報が存在しないか、サーバーがクエリに対する応答を提供できないことを示します。この情報の保存は、ネガティブキャッシングと呼ばれます。ネガティブキャッシュは、ドメインに関するクエリへの応答を高速化するのに役立ちます。

> 注:
>
> ネガティブキャッシュは、バックエンドサーバーが照会されたドメインの権限がある DNS (ADNS) サーバーとして構成されている場合にのみサポートされます。

負の応答は、次のいずれかになります。

- NXDOMAIN エラーメッセージ：照会されたドメイン名にサーバにレコードが構成されていない場合、権限を持つ DNS サーバが NXDOMAIN エラーメッセージで応答します。このメッセージは、照会されたドメインが無効または存在しないドメイン名であることを示しています。
- NODATA エラーメッセージ — クエリ内のドメイン名は有効で、指定されたタイプのレコードが使用できない場合、アプライアンスは NODATA エラーメッセージを送信します。

ネガティブ・キャッシュが有効な場合、アプライアンスは DNS サーバーからのネガティブ応答をキャッシュし、キャッシュからの今後の要求のみを処理します。このアクションは、クエリへの応答を高速化し、バックエンド DNS トラフィックを削減するのにも役立ちます。ネガティブキャッシュは、すべての展開、つまり Citrix ADC アプライアンスがプロキシ、エンドリゾルバ、またはフォワーダとして機能している場合に使用できます。

DNS プロファイルを使用してネガティブキャッシュを有効または無効にできます。詳細については、「DNS プロファイル」を参照してください。既定では、ネガティブキャッシュは、既定で DNS 仮想サーバーにバインドされている既定の DNS プロファイル (**default**-dns-profile) または新しく作成された DNS プロファイルに有効になっています。

CLI を使用してネガティブキャッシュを有効または無効にする

コマンドプロンプトで次のコマンドを入力して、ネガティブキャッシュを有効または無効にし、構成を確認します。

```
1 -  add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED
     )] [-cacheNegativeResponses (ENABLED | DISABLED )]
2 -  show dns profile [<dnsProfileName>]
3 <!--NeedCopy-->
```

デフォルトの **DNS** プロファイルの例:

```
1 > sh dns profile default-dns-profile
2     1)   default-dns-profile
3         Query logging : DISABLED        Answer section logging :
             DISABLED
4         Extended logging : DISABLED     Error logging : DISABLED
5         Cache Records : ENABLED     Cache Negative Responses: ENABLED
6 Done
7 <!--NeedCopy-->
```

新しく作成された **DNS** プロファイルの例**:**

```
1 > add dnsprofile dns_profile1 -cacheRecords ENABLED -
     cacheNegativeResponses ENABLED
2 Done
3 > show dns profile dns_profile1
4     1)   dns_profile1
5         Query logging : DISABLED        Answer section logging :
             DISABLED
6         Extended logging : DISABLED     Error logging : DISABLED
```



```
7          Cache Records : ENABLED     Cache Negative Responses: ENABLED
8  Done
9  <!--NeedCopy-->
```

CLI を使用してサービスレベルまたは仮想サーバレベルの DNS パラメータを指定する

コマンドプロンプトで、次の手順を実行します。

1. DNS プロファイルを設定します。

   ```
   add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED )]
    [-cacheNegativeResponses (ENABLED | DISABLED )]
   ```

2. DNS プロファイルをサービスまたは仮想サーバーにバインドします。

   DNS プロファイルをサービスにバインドするには、次の手順を実行します。

   ```
   set service <name> [-dnsProfileName <string>]
   ```

例:

```
1  >set service service1 -dnsProfileName dns_profile1
2  Done
3  <!--NeedCopy-->
```

DNS プロファイルを仮想サーバにバインドするには、次の手順を実行します。

```
set lb vserver <name> [-dnsProfileName <string>]
```

例:

```
1  >set lb vserver lbvserver1 -dnsProfileName dns_profile1
2  Done
3  <!--NeedCopy-->
```

GUI を使用してサービスレベルまたは仮想サーバレベルの DNS パラメータを指定する

1. HTTP プロファイルを設定します。

   [システム] > [プロファイル] > [**DNS** プロファイル] に移動し、DNS プロファイルを作成します。

2. HTTP プロファイルをサービスまたは仮想サーバーにバインドします。

   [トラフィック管理] > [負荷分散] > [サービス/仮想サーバー] に移動し、サービスまたは仮想サーバーにバインドする必要がある DNS プロファイルを作成します。

アプライアンスが処理するレート制限負応答

キャッシュから Citrix ADC アプライアンスが処理するネガティブ応答のしきい値を設定できます。しきい値が設定されると、アプライアンスはしきい値に達するまでキャッシュからの応答を提供します。しきい値に達すると、アプライアンスは NXDOMAIN 応答で応答するのではなく、要求を廃棄します。

否定的な応答のレート制限を設定すると、次の利点があります。

- Citrix ADC アプライアンスにリソースを保存します。
- 存在しないドメイン名に対する悪意のあるクエリを防ぎます。

> 注： ADC アプライアンスが権限のあるドメインネームサーバーとして構成されているドメインに対してのみ、否定応答のしきい値を設定できます。権限のあるバックエンドネームサーバーから受信したキャッシュレコードのしきい値を設定することはできません。

CLI を使用してキャッシュによって提供される負の応答のレート制限

コマンドプロンプトで、次のように入力します。

```
set dns parameter -NXDOMainRateLimitThreshold <positive-integer>
<!--NeedCopy-->
```

例**:**

```
set dns parameter -NXDOMainRateLimitThreshold 1000
<!--NeedCopy-->
```

nxDomainRateLimitThreshold：このパラメータが正の整数値に設定されている場合、このしきい値（秒単位）に達するまで応答がキャッシュから提供されます。しきい値を超えると、要求はドロップされます。設定されたしきい値は、パケットエンジン単位です。

GUI を使用したキャッシュによって提供される負応答のレート制限

1. [トラフィック管理**] > [DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2. [**DNS** パラメータの構成] ページの [**NXDOMAIN** レート制限しきい値] フィールドに、キャッシュから応答が処理されるまでのしきい値を入力します。

注：［**NXDOMAIN** しきい値の超過］の値には、しきい値に達した後に要求がドロップされた回数が表示されます。

Citrix ADC アプライアンスがプロキシモードのときに EDNS0 クライアントのサブネットデータをキャッシュする

October 7, 2021

Citrix ADC プロキシモードでは、EDNS0 クライアントサブネット（ECS）をサポートするバックエンドサーバーが ECS オプションを含む応答を送信すると、Citrix ADC アプライアンスは次のことを行います。

- 応答をそのままクライアントに転送し、
- クライアントのサブネット情報とともに、応答をキャッシュに保存します。

同じドメインの同じサブネットからのもので、サーバーが同じ応答を送信する DNS 要求は、キャッシュから処理されます。

注:

- ECS キャッシュはデフォルトで無効になっています。関連する DNS プロファイルで EDNS0 クライアントサブネットデータのキャッシュを有効にします。
- ドメインに対してキャッシュできるサブネットの数は、使用可能なサブネット ID（Citrix ADC アプライアンスでは 1270 個）に制限されます。オプションで、制限を小さい数値に設定できます（最小値:1 ipv4/ipv6）。

CLI を使用した ECS 応答のキャッシュの有効化

コマンドプロンプトで入力します。

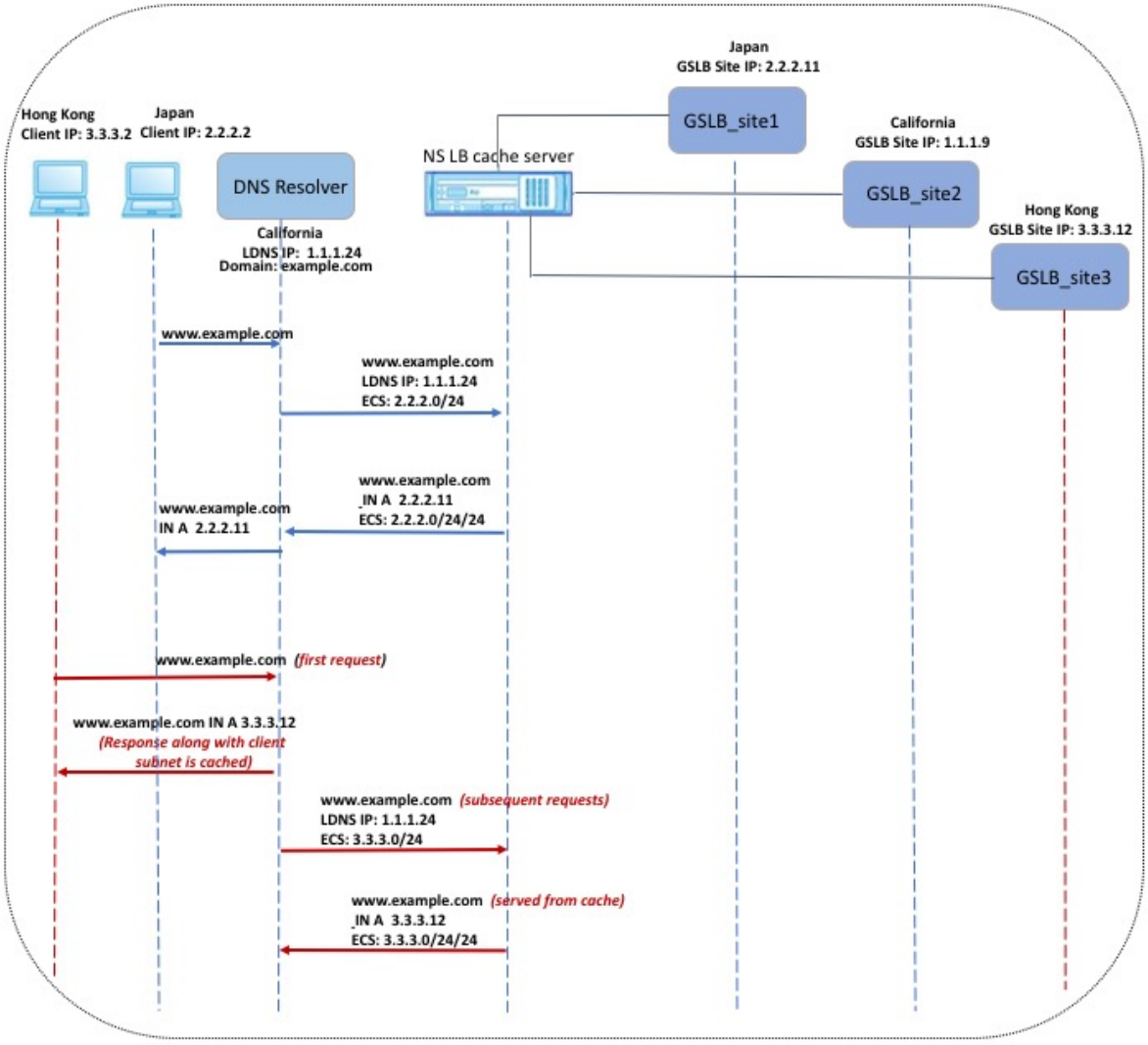
```
set dns profile <dnsProfileName> -cacheECSSubnet ( ENABLED | DISABLED )
```

CLI を使用してドメインごとにキャッシュできるサブネットの数を制限する

コマンドプロンプトで入力します。

```
set dns profile <dnsProfileName> -maxSubnetsPerDomain <positive_integer>
```

例:



前の図に示されている例では、IP アドレス 2.2.2.2 のクライアントが www.example.com のクエリを DNS リゾルバーに送信します。DNS リゾルバーは次の応答を送信します。

www.example.com IN A, IP is 2.2.2.11, and ECS 2.2.2.0/24/24

この時点で、応答とクライアントサブネット識別子 (2.2.2.0/24) がキャッシュされます。同じサブネットおよびドメインからのそれ以降の要求は、キャッシュから処理されます。

たとえば、クライアントの IP アドレスが 2.2.2.100 で、クエリが www.example.com に対するものである場合、クエリはバックエンドサーバーに送信されるのではなく、キャッシュから提供されます。

ドメイン・ネーム・システムのセキュリティ拡張

October 7, 2021

DNS Security Extensions（DNSSEC）は、インターネット技術特別調査委員会（IETF）の標準です。これは、UDP応答をクリアテキストで送信しながら、ネームサーバーとクライアント間の通信でデータの整合性とデータ発信元の認証を提供することを目的としています。DNSSEC は、非対称キー暗号を使用するメカニズムと、その実装に固有の新しいリソースレコードのセットを指定します。

DNSSEC 仕様は次のとおりです。

- RFC 4033、「DNS セキュリティの概要と要件」
- RFC 4034、「DNS セキュリティ拡張機能のリソースレコード」
- RFC 4035、「DNS セキュリティ拡張機能のプロトコル変更」

DNS 内で DNSSEC を実装する際の運用面については、RFC 4641「DNSSEC 運用慣行」で説明しています。

DNSSEC は、Citrix ADC 上で構成することができます。DNS ゾーンに署名するためのキーを生成およびインポートできます。Citrix ADC が権限を持つゾーンに対して DNSSEC を構成できます。バックエンドネームサーバーのファームでホストされている署名付きゾーンの DNS プロキシサーバーとして ADC を構成できます。ADC が DNS プロキシサーバーとして構成されているゾーンに属するレコードのサブセットに対して権限がある場合は、DNSSEC 実装にレコードのサブセットを含めることができます。

DNSSEC の構成

October 7, 2021

DNSSEC を設定するには、次の手順を実行します。

1. Citrix ADC アプライアンスで DNSSEC を有効にします。
2. ゾーンのゾーン署名キーとキー署名キーを作成します。
3. ゾーンに 2 つのキーを追加します。
4. キーでゾーンに署名します。

Citrix ADC アプライアンスは、DNSSEC リゾルバとして機能しません。ADC の DNSSEC は、次の展開シナリオでのみサポートされます。

1. ADNS：Citrix ADC は ADNS であり、署名自体を生成します。
2. プロキシ：Citrix ADC は DNSSEC プロキシとして機能します。Citrix ADC は、ADNS/LDNS サーバーの前にトラステッドモードで配置されることを前提としています。ADC はプロキシ・キャッシング・エンティティとしてのみ動作し、署名を検証しません。

DNSSEC を有効または無効にする

ADC が DNSSEC 対応クライアントに応答するために、Citrix ADC で DNSSEC を有効にします。デフォルトでは、DNSSEC は有効になっています。

Citrix ADC が DNSSEC 固有の情報を使用してクライアントに応答しないようにする場合は、DNSSEC 機能を無効にできます。

CLI を使用して **DNSSEC** を有効または無効にする

コマンドプロンプトで次のコマンドを入力して DNSSEC を有効または無効にし、構成を確認します。

```
-  set dns parameter -dnssec ( ENABLED | DISABLED )
-  show dns parameter
<!--NeedCopy-->
```

例**:**

```
> set dns parameter -dnssec ENABLED
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
         .
         .
         .
         DNSEC Extension: ENABLED
        Max DNS Pipeline Requests: 255
 Done

<!--NeedCopy-->
```

GUI を使用して **DNSSEC** を有効または無効にする

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細ウィンドウで、[DNS 設定の変更] をクリックします。
3. [**DNS** パラメータの構成] ダイアログボックスで、[**DNSSEC** 拡張を有効にする] チェックボックスをオンまたはオフにします。

ゾーンの **DNS** キーの作成

署名する DNS ゾーンごとに、非対称キーのペアを 2 つ作成する必要があります。ゾーン署名キー (ZSK) と呼ばれる 1 つのペアは、ゾーン内のすべてのリソースレコードセットに署名するために使用されます。2 番目のペアは、キー署名キー (KSK) と呼ばれ、ゾーン内の DNSKEY リソースレコードにのみ署名するために使用されます。

ZSK と KSK が作成されると、キーの公開コンポーネントの名前にsuffix.keyが追加されます。プライベートコンポーネントの名前にsuffix.**private** が付加されます。追加は自動的に行われます。

また、Citrix ADC は委任署名者（DS）レコードを作成し、レコードの名前に接尾辞.ds を追加します。親ゾーンが署名付きゾーンの場合、信頼チェーンを確立するには、親ゾーンで DS レコードを発行する必要があります。

キーを作成すると、そのキーは/nsconfig/dns/ ディレクトリに格納されますが、ゾーンに自動的に公開されることはありません。create dns keyコマンドを使用してキーを作成した後、add dns keyコマンドを使用してゾーンでキーを明示的に公開する必要があります。キーを生成するプロセスは、ゾーンでキーを発行するプロセスとは独立しており、代替手段を使用してキーを生成できます。たとえば、Secure FTP（SFTP）を使用して、他のキー生成プログラム（bind-keygenなど）によって生成されたキーをインポートし、ゾーンでキーを発行できます。ゾーンでのキーの公開の詳細については、「ゾーンでの DNS キーの公開」を参照してください。

このトピックで説明されている手順を実行してゾーン署名キーを作成し、その手順を繰り返してキー署名キーを作成します。コマンド構文に続く例では、まず zone example.com のゾーン署名キーペアを作成します。次に、コマンドを使用して、ゾーンのキー署名キーペアを作成します。

リリース 13.0 ビルド 61.x から、Citrix ADC アプライアンスは、DNS ゾーンを認証するために、RSASHA256 や RSASHA512 などの強力な暗号化アルゴリズムをサポートするようになりました。以前は、RSASHA1 アルゴリズムだけがサポートされていました。

CLI を使用して **DNS** キーを作成する

コマンドプロンプトで入力します。

```
create dns key -zoneName <string> -keyType <keyType> -algorithm <algorithm>
 -keySize <positive_integer> -fileNamePrefix <string>
```

例**:**

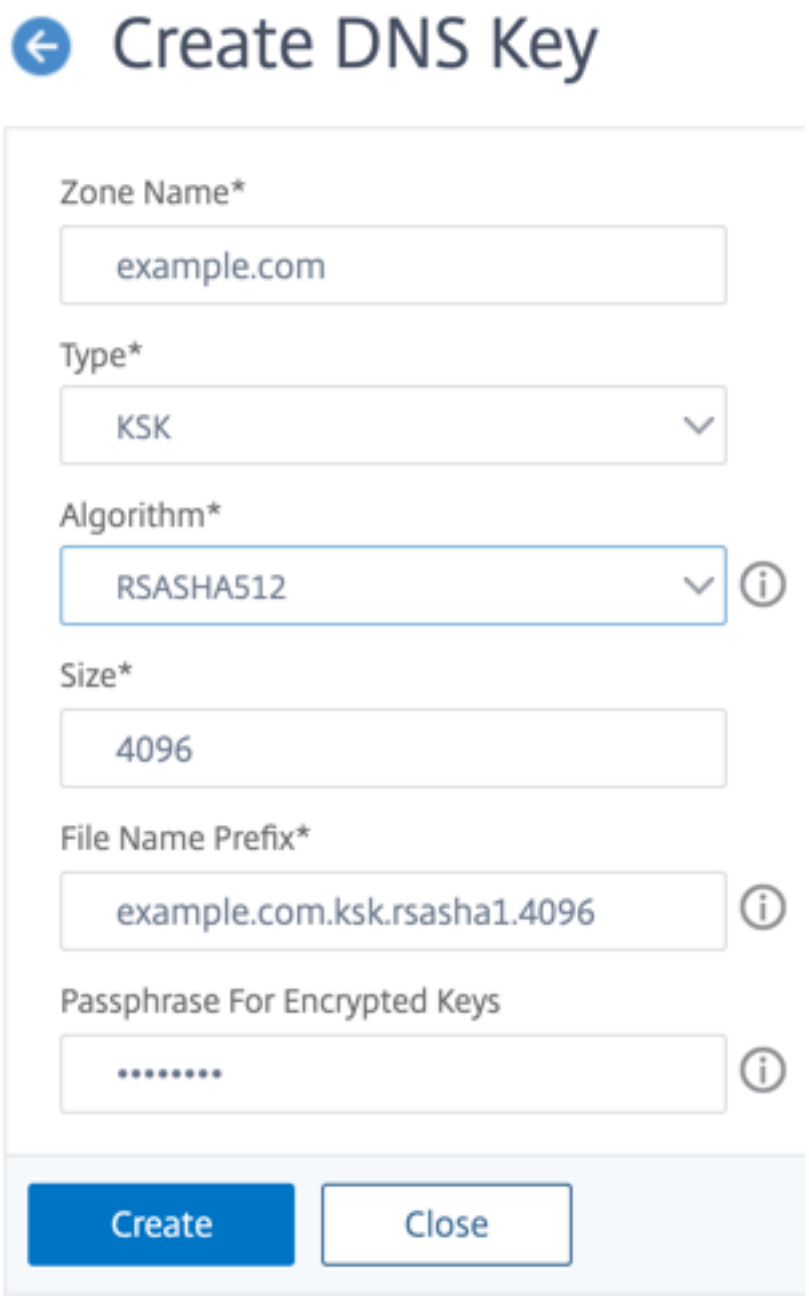
```
1 > create dns key -zoneName example.com -keyType zsk -algorithm
     RSASHA256 -keySize 1024 -fileNamePrefix example.com.zsk.rsasha1.1024
2 File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /
     nsconfig/dns/example.com.zsk.rsasha1.1024.private (private); /
     nsconfig/dns/example.com.zsk.rsasha1.1024.ds (ds)
3 This operation may take some time, Please wait...
4  Done
5 > create dns key -zoneName example.com -keyType ksk -algorithm
     RSASHA512 -keySize 4096 -fileNamePrefix example.com.ksk.rsasha1.4096
6 File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /
     nsconfig/dns/example.com.ksk.rsasha1.4096.private (private); /
     nsconfig/dns/example.com.ksk.rsasha1.4096.ds (ds)
7 This operation may take some time, Please wait...
8  Done
```



```
<!--NeedCopy-->
```

GUI を使用して **DNS** キーを作成する

1. [トラフィック管理**] > [DNS**] に移動します。
2. 詳細領域で、[**DNS** キーの作成] をクリックします。
3. さまざまなパラメータの値を入力し、[作成] をクリックします。



注：既存のキーのファイル名プレフィックスを変更するには、次のようにします。

- [参照] ボタンの横にある矢印をクリックします。
- [ローカル] または [アプライアンス] をクリックします（既存のキーがローカルコンピュータに保存されているか、アプライアンス上の/nsconfig/dns/ ディレクトリに保存されているかに応じて）。
- キーの場所を参照し、キーをダブルクリックします。
  [ファイル名の接頭辞] ボックスには、既存のキーの接頭語のみが入力されます。それに応じて接頭辞を変更します。

ゾーン内の **DNS** キーを発行する

キー（ゾーン署名キーまたはキー署名キー）は、ADC アプライアンスにキーを追加することによってゾーンで公開されます。ゾーンに署名する前に、キーをゾーンに発行する必要があります。

ゾーンでキーを公開する前に、そのキーが **/nsconfig/dns/** ディレクトリにある必要があります。別のコンピュータに DNS キーを作成した場合 (たとえば、bind-keygen プログラムを使用して)、そのキーが/nsconfig/dns/ ディレクトリに追加されていることを確認します。次に、ゾーンでキーを発行します。ADC GUI を使用して、キーを/nsconfig/dns/ ディレクトリに追加します。または、Secure FTP (SFTP) などの他のプログラムを使用して、キーをディレクトリにインポートします。

特定のゾーンで公開する公開鍵と秘密鍵のペアごとにadd dns key コマンドを使用します。ゾーンの ZSK ペアと KSK ペアを作成した場合は、add dns key コマンドを使用して、まずゾーン内のキーペアの 1 つを公開します。コマンドを繰り返して、もう一方のキーペアを公開します。ゾーンで発行するキーごとに、ゾーンに DNSKEY リソースレコードが作成されます。

コマンド構文に続く例では、まず、ゾーン内のゾーン署名キーペア (example.com ゾーン用に作成された) を公開します。次に、コマンドを使用して、ゾーン内のキー署名キーペアを公開します。

CLI を使用してゾーンにキーを発行する

コマンドプロンプトで次のコマンドを入力して、ゾーンにキーを発行し、構成を確認します。

```
1 -  add dns key <keyName> <publickey> <privatekey> [-expires <
     positive_integer> [<units>]] [-notificationPeriod <positive_integer>
      [<units>]] [-TTL <secs>]
2 -  show dns zone [<zoneName> | -type <type>]
3 <!--NeedCopy-->
```

例**:**


```
> add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.com.zsk.rsasha1.1024.private
 Done
> add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.com.ksk.rsasha1.4096.private
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY
         Domain Name : ns1.example.com
                 Record Types : A
         Domain Name : ns2.example.com
                 Record Types : A
 Done
<!--NeedCopy-->
```

GUI を使用して **DNS** ゾーンにキーを発行する

[トラフィック管理] **>** **[DNS] > [**キー**]** に移動します。

注**:** [公開キー] と [秘密キー] の場合、ローカルコンピュータに保存されているキーを追加するには、[参照] ボタンの横にある矢印をクリックし、[ローカル] をクリックしてキーの場所を参照し、キーをダブルクリックします。

DNS キーを構成する

ゾーンで公開されているキーのパラメータを設定できます。キーの有効期限、通知期間、および有効期限 (TTL) パラメータを変更できます。キーの有効期限を変更すると、アプライアンスはゾーン内のすべてのリソースレコードをキーで自動的に再署名します。再署名は、ゾーンが特定のキーで署名されている場合に発生します。

CLI を使用したキーの設定

コマンドプロンプトで次のコマンドを入力して、キーを構成し、構成を確認します。

```
-  set dns key <keyName> [-expires <positive_integer> [<units>]] [-notificationPeriod <positive_integer> [<units>]] [-TTL <secs>]
-  show dns key [<keyName>]
<!--NeedCopy-->
```

例**:**


```
> set dns key example.com.ksk -expires 30 DAYS -notificationPeriod 3
   DAYS -TTL 3600
 Done
> show dns key example.com.ksk
1)      Key Name: example.com.ksk
        Expires: 30 DAYS        Notification: 3 DAYS    TTL: 3600
        Public Key File: example.com.ksk.rsasha1.4096.key
        Private Key File: example.com.ksk.rsasha1.4096.private
 Done
<!--NeedCopy-->
```

GUI を使用したキーの設定

1. [トラフィック管理] **>** **[DNS] >** [キー] に移動します。
2. 詳細ウィンドウで、構成するキーをクリックし、[開く] をクリックします。
3. [DNS キーの構成] ダイアログボックスで、次のパラメータの値を次のように変更します。
   - 期限切れ-期限切れ
   - 通知期間-notificationPeriod
   - TTL：TTL
4. [OK] をクリックします。

DNS ゾーンの署名と署名の解除

DNS ゾーンをセキュリティで保護するには、ゾーンで発行されたキーを使用してゾーンに署名する必要があります。ゾーンに署名すると、Citrix ADC は所有者名ごとに NSEC（Next Secure）リソースレコードを作成します。次に、キー署名キーを使用して DNSKEY リソースレコードセットに署名します。最後に、ZSK を使用して、ゾーン内のすべてのリソースレコードセット (DNSKEY リソースレコードセットや NSEC リソースレコードセットを含む) に署名します。署名操作を行うたびに、ゾーン内のリソースレコードセットの署名が作成されます。署名は、RRSIG リソースレコードと呼ばれる新しいリソースレコードにキャプチャされます。

ゾーンに署名したら、構成を保存します。

CLI を使用してゾーンに署名する

コマンドプロンプトで次のコマンドを入力してゾーンに署名し、構成を確認します。


```
-  sign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
-  save config
<!--NeedCopy-->
```

例:

```
> sign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY RRSIG NSEC
         Domain Name : ns1.example.com
                 Record Types : A RRSIG NSEC
         Domain Name : ns2.example.com
                 Record Types : A RRSIG
         Domain Name : ns2.example.com
                 Record Types : RRSIG NSEC
Done
> save config
 Done
<!--NeedCopy-->
```

CLI を使用してゾーンの署名を解除する

コマンド・プロンプトで次のコマンドを入力して、ゾーンの署名を解除し、構成を確認します。

```
-  unsign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
<!--NeedCopy-->
```

例:

```
> unsign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
         Zone Name : example.com
```



```
5           Proxy Mode : NO
6           Domain Name : example.com
7                   Record Types : NS SOA DNSKEY
8           Domain Name : ns1.example.com
9                   Record Types : A
10          Domain Name : ns2.example.com
11                  Record Types : A
12  Done
13 <!--NeedCopy-->
```

GUI を使用してゾーンに署名または署名解除する

1. [トラフィック管理] **>** **[DNS]** **>** **[**ゾーン**]** に移動します。
2. 詳細ウィンドウで、署名するゾーンをクリックし、[署名/署名解除] をクリックします。
3. [DNS ゾーンの署名/署名解除] ダイアログボックスで、次のいずれかの操作を行います。
   - ゾーンに署名するには、ゾーンに署名するキー（ゾーン署名キーとキー署名キー）のチェックボックスをオンにします。
     ゾーンには、複数のゾーン署名キーまたはキー署名キーペアを使用して署名できます。
   - ゾーンの署名を解除するには、ゾーンの署名を解除するキー（ゾーン署名キーとキー署名キー）のチェックボックスをオフにします。
     複数のゾーン署名キーまたはキー署名キーペアを使用して、ゾーンの署名を解除できます。
4. [OK] をクリックします。

ゾーン内の特定のレコードの **NSEC** レコードを表示する

Citrix ADC がゾーン内の各所有者名に対して自動的に作成する NSEC レコードを表示できます。

CLI を使用してゾーン内の特定のレコードの **NSEC** レコードを表示する

コマンドプロンプトで次のコマンドを入力して、ゾーン内の特定のレコードの NSEC レコードを表示します。

```
show dns nsecRec [<hostName> | -type (ADNS | PROXY | ALL)]
```

例**:**

```
1 > show dns nsecRec example.com
2 1)      Domain Name : example.com
3         Next Nsec Name: ns1.example.com
4         Record Types : NS SOA DNSKEY RRSIG NSEC
5  Done
6 <!--NeedCopy-->
```


GUI を使用して、ゾーン内の特定のレコードの **NSEC** レコードを表示する

1. [トラフィック管理] **>** **[DNS]** **>** **[**レコード**]** **>** **[**次のセキュアレコード] に移動します。
2. 詳細ウィンドウで、NSEC レコードを表示するレコードの名前をクリックします。選択したレコードの NSEC レコードが [詳細] 領域に表示されます。

DNS キーを削除する

キーの有効期限が切れたとき、またはキーが侵害された場合に、公開されているゾーンからキーを削除します。ゾーンからキーを削除すると、ゾーンは自動的にキーで署名されません。このコマンドでキーを削除しても、/nsconfig/dns/ ディレクトリにあるキーファイルは削除されません。キーファイルが不要になった場合は、ディレクトリから明示的に削除する必要があります。

CLI を使用して **Citrix ADC** からキーを削除する

コマンドプロンプトで次のコマンドを入力してキーを削除し、構成を確認します。

```
-  rm dns key <keyName>
-  show dns key <keyName>
<!--NeedCopy-->
```

例**:**

```
> rm dns key example.com.zsk
 Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]

<!--NeedCopy-->
```

GUI を使用して **Citrix ADC** からキーを削除する

1. [トラフィック管理] **>** **[DNS]** **>** **[**キー**]** に移動します。
2. 詳細ウィンドウで、ADC から削除するキーの名前をクリックし、[削除] をクリックします。

Citrix ADC がゾーンに対して権限を持つ場合に **DNSSEC** を構成する

October 7, 2021

Citrix ADC が特定のゾーンに対して権限を持つ場合、ゾーン内のすべてのリソースレコードが ADC 上で構成されます。権限のあるゾーンに署名するには、ゾーンのゾーン署名キーとキー署名キーを作成し、キーを ADC に追加してから、ゾーンに署名する必要があります。詳しくは、次のトピックを参照してください:

- ゾーンの DNS キーの作成
- ゾーン内の DNS キーを発行する
- DNS ゾーンに署名し、署名解除します。

ADC で構成された GSLB ドメインが署名されているゾーンに属している場合、GSLB ドメイン名は、ゾーンに属する他のレコードとともに署名されます。

ゾーンに署名した後、DNSSEC 対応クライアントからの要求に対する応答には、RRSIG リソースレコードと、要求されたリソースレコードが含まれます。DNSSEC は ADC で有効にする必要があります。DNSSEC の有効化の詳細については、DNSSEC の有効化および無効化を参照してください。

最後に、DNSSEC を権限ゾーン用に構成したら、Citrix ADC 構成を保存する必要があります。

Citrix ADC が DNS プロキシサーバーであるゾーンに対して DNSSEC を構成する

October 7, 2021

Citrix ADC が DNS プロキシサーバーとして構成されているゾーンに署名する手順は、ADC がバックエンドネームサーバーが所有するゾーン情報のサブセットを所有しているかどうかによって異なります。含まれている場合、その構成は部分的なゾーン所有権構成と見なされます。ADC がゾーン情報のサブセットを所有していない場合、バックエンドサーバーを管理するための Citrix ADC 構成は、ゾーンのない DNS プロキシサーバー構成と見なされます。両方の Citrix ADC 構成に対する基本的な DNSSEC 構成作業は同じです。ただし、Citrix ADC で部分ゾーンに署名するには、追加の構成手順が必要です。

注: ゾーンレスプロキシサーバー構成および部分ゾーンという用語は、Citrix ADC アプライアンスのコンテキストでのみ使用されます。

重要: プロキシモードに設定されている場合、ADC はキャッシュを更新する前に DNSSEC 応答で署名検証を実行しません。

DNSSEC 認識リゾルバ (サーバー) を負荷分散するために、ADC を DNS プロキシとして構成する場合は、DNS 仮想サーバーを構成するときに [再帰可能] オプションを設定する必要があります。Checking Disabled（CD）ビットが設定された状態で DNSSEC クエリが到着した場合、クエリは CD ビットを保持したままサーバーに渡されます。サーバーからの応答はキャッシュされません。

ゾーンレス DNS プロキシサーバー構成用に DNSSEC を構成する

ゾーンのない DNS プロキシサーバー構成の場合、ゾーン署名はバックエンドネームサーバーで実行する必要があります。Citrix ADC で、ADC をゾーンの DNS プロキシサーバーとして構成します。プロトコルタイプ DNS の負荷分

散仮想サーバーを作成します。ネームサーバーを表すように ADC でサービスを構成します。次に、サービスを負荷分散仮想サーバーにバインドします。これらの構成タスクの詳細については、「NetScaler を DNS プロキシサーバーとして構成する」を参照してください。

クライアントが DNSSEC OK（DO）ビットが設定された DNS 要求を ADC に送信すると、ADC は要求された情報についてキャッシュをチェックします。リソースレコードがキャッシュで利用できない場合、ADC は要求を DNS ネームサーバーの 1 つに転送します。次に、ネームサーバーからクライアントに応答を中継します。また、ADC は、ネームサーバーからの応答とともに RRSIG リソースレコードをキャッシュします。DNSSEC 対応クライアントからの後続の要求は、存続可能時間 (TTL) パラメータに従って、キャッシュ (RRSIG リソースレコードを含む) から提供されます。クライアントが DO ビットを設定せずに DNS 要求を送信すると、ADC は要求されたリソースレコードのみで応答します。DNSSEC に固有の RRSIG リソースレコードは含まれていません。

部分ゾーン所有権設定用の **DNSSEC** の構成

一部の ADC 構成では、ゾーンの権限がバックエンドネームサーバーにある場合でも、ゾーンに属するリソースレコードのサブセットが ADC で構成されている場合があります。ADC は、このレコードのサブセットのみを所有している（または権限がある）。このようなレコードのサブセットは、ADC の 部分的なゾーンを構成すると考えることができます。ADC は部分ゾーンを所有します。他のすべてのレコードは、バックエンドネームサーバーによって所有されます。

Citrix ADC の一般的な部分ゾーン構成は、次の場合に見られます。

- グローバルサーバー負荷分散（GSLB）ドメインは ADC で構成されます
- GSLB ドメインは、バックエンドネームサーバーが権限を持つゾーンの一部です。

ADC の部分ゾーンのみを含むゾーンに署名するには、次のことが必要です。

- バックエンドネームサーバーゾーンファイルに部分的なゾーン情報を含める
- バックエンドネームサーバーでゾーンに署名する
- ADC の部分ゾーンに署名します。

ネームサーバーのゾーンと ADC の部分ゾーンに署名するには、同じキーセットを使用する必要があります。

バックエンドネームサーバーでゾーンに署名します

1. 部分ゾーンに含まれるリソースレコードを、ネームサーバーのゾーンファイルに含めます。
2. キーを作成し、そのキーを使用してバックエンドネームサーバーのゾーンに署名します。

Citrix ADC で部分ゾーンに署名する

1. バックエンドネームサーバーが所有するゾーンの名前でゾーンを作成します。部分ゾーンを構成する場合は、proxyMode パラメータを YES に設定します。このゾーンは、ADC が所有するリソースレコードを含む部分的なゾーンです。

たとえば、バックエンドネームサーバーで構成されているゾーンの名前が example.com の場合、ADC で example.com という名前のゾーンを作成する必要があります。proxyMode パラメーターを YES に設定します。ゾーンの追加の詳細については、「DNS ゾーンの構成」を参照してください。

> 注
>
> ゾーンに SOA レコードと NS レコードを追加しないでください。これらのレコードは、ADC が権限を持つゾーンの ADC に存在する必要があります。

2. キーを（バックエンドネームサーバーの 1 つから）ADC にインポートしてから、/nsconfig/dns/ ディレクトリ。キーをインポートして ADC に追加する方法の詳細については、「ゾーンでの DNS キーの公開」を参照してください。
3. インポートされたキーで部分ゾーンに署名します。キーを使用して部分ゾーンに署名すると、ADC はリソースレコードセットの RRSIG レコードと NSEC レコードをそれぞれ生成し、部分ゾーン内の個々のリソースレコードを生成します。ゾーンの署名の詳細については、「DNS ゾーンの署名と署名の解除」を参照してください。

グローバルサーバー負荷分散（**GSLB**）ドメイン名用に **DNSSEC** を構成する

October 7, 2021

GSLB が Citrix ADC で構成されており、ADC が GSLB ドメイン名が属するゾーンに対して権限を持っている場合、ゾーンが署名されるときにすべての GLSB ドメイン名が署名されます。ADC が権限のあるゾーンの署名の詳細については、「Citrix ADC アプライアンスがゾーンに対して権限を持つ場合に DNSSEC を構成する」を参照してください。

GSLB ドメインが、バックエンドネームサーバーが権限を持つゾーンに属している場合は、次のことを行う必要があります。

- まず、ネームサーバーでゾーンに署名します。
- 次に、ADC の部分ゾーンに署名して、ゾーンの DNSSEC 構成を完了します。

詳細については、「ゾーンの所有権の一部構成の DNSSEC の構成」を参照してください。

ゾーンのメンテナンス

October 7, 2021

DNSSEC の観点から見ると、ゾーンのメンテナンスには、キーの有効期限が近づいたときに、ゾーン署名キーとキー署名キーのロールオーバーが含まれます。これらのゾーン保守タスクは手動で実行する必要があります。ゾーンは自動的に再署名されるため、手動による介入は必要ありません。

更新されたゾーンの再署名

ゾーンが更新（レコードの追加または既存のレコードの変更）されると、アプライアンスは自動的に新しい（または変更された）レコードを再署名します。ゾーンに複数のゾーン署名キーが含まれている場合、アプライアンスはゾーンの署名に使用したキーを使用して新しい（または変更された）レコードに再署名します。

DNSSEC キーをロールオーバーする

注**:** 有効期限が切れる前に、DNSSEC キー (KSK、ZSK) を手動でロールオーバーします。

Citrix ADC では、事前公開方法と二重署名方法を使用して、ゾーン署名キーとキー署名キーのロールオーバーを実行できます。これらの 2 つのロールオーバー方法の詳細については、RFC 4641「DNSSEC 運用慣行」を参照してください。

次のトピックでは、ADC のコマンドを、RFC 4641 で説明したロールオーバー手順のステップにマッピングします。

キーの有効期限の通知は、dnskeyExpiry という SNMP トラップを介して送信されます。dnskey 有効期限 SNMP トラップと共に、dnskeyName、有効期限が切れるまでの時間、および期限切れの dnskey 単位の 3 つの MIB 変数が送信されます。詳細については、『*NetScaler 12.0 SNMP OID* リファレンス』の「*Citrix*NetScaler SNMP OID リファレンス」を参照してください。

公開前キーのロールオーバー

RFC 4641「DNSSEC 運用慣行」では、公開前キーのロールオーバー方法の 4 つの段階、つまり、初期、新しい DNSKEY、新しい RRSIG、および DNSKEY の削除が定義されています。各ステージは、ADC で実行する必要がある一連のタスクに関連付けられています。次に、各ステージの説明と実行する必要があるタスクを示します。ここで説明するロールオーバー手順は、キー署名キーとゾーン署名キーの両方に使用できます。

- ステージ **1:** イニシャル。ゾーンには、ゾーンが現在署名されているキーセットのみが含まれます。初期段階でのゾーンの状態は、キーのロールオーバープロセスを開始する直前のゾーンの状態です。

  例**:**

  ゾーン example.com が署名されているキー example.com.zsk1 を考えてみましょう。ゾーンには、example.com.zsk1 キーによって生成された RRSG だけが含まれます。この期限は期限切れになります。キー署名キーは example.com.ksk1 です。

- ステージ **2**: 新しい **DNSKEY**。新しいキーが作成され、ゾーンで公開されます。つまり、キーは ADC に追加されますが、プレロールフェーズが完了するまで、ゾーンは新しいキーで署名されません。この段階では、ゾーンには古いキー、新しいキー、および古いキーによって生成された RRSG が含まれます。プレロールフェーズの全期間にわたって新しいキーを公開すると、新しいキーに対応する DNSKEY リソースレコードがセカンダリネームサーバーに伝播されます。

  例**:**

新しいキー example.com.zsk2 が example.com ゾーンに追加されます。プリロールフェーズが完了するまで、ゾーンには example.com.zsk2 で署名されません。このゾーンには、example.com.zsk1 と example.com.zsk2 の両方の DNSKEY リソースレコードが含まれています。

Citrix ADC コマンドは以下のとおりです。

ADC で次のタスクを実行します。

- `create dns key`コマンドを使用して DNS キーを作成します。

  例を含む DNS キーの作成の詳細については、「ゾーンの DNS キーを作成する」を参照してください。

- `add dns key`コマンドを使用して、ゾーン内の新しい DNS キーを発行します。

  例など、ゾーンでのキーの公開の詳細については、「ゾーンでの DNS キーの公開」を参照してください。

- ステージ **3:** 新しい **RRSIGs**。ゾーンは新しい DNS キーで署名され、古い DNS キーで署名されていません。古い DNS キーはゾーンから削除されず、古いキーによって生成された RRSG の有効期限が切れるまで公開されたままになります。

  例**:**

  ゾーンは example.com.zsk2 で署名され、次に example.com.zsk1 で署名されていません。ゾーンは、example.com.zsk1 によって生成された RRSG の有効期限が切れるまで example.com.zsk1 を発行し続けます。

  Citrix ADC コマンドは以下のとおりです。

  ADC で次のタスクを実行します。

  - `sign dns zone`コマンドを使用して、新しい DNS キーを使用してゾーンに署名します。
  - `unsign dns zone`コマンドを使用して、古い DNS キーを使用してゾーンの署名を解除します。

  例を含むゾーンの署名と署名の解除の詳細については、「DNS ゾーンの署名と署名の解除」を参照してください。

- ステージ **4**: **DNSKEY** の取り外し。古い DNS キーによって生成された RRSG の有効期限が切れると、古い DNS キーがゾーンから削除されます。

  例**:**

  古い DNS キー example.com.zsk1 が example.com ゾーンから削除されます。

  Citrix ADC コマンド

  ADC で、`rm dns key` コマンドを使用して、古い DNS キーを削除します。例など、ゾーンからキーを削除する方法の詳細については、「DNS キーの削除」を参照してください。

二重署名キーのロールオーバー

RFC 4641、「DNSSEC 運用慣行」では、二重署名キーのロールオーバーについて、初期、新しい DNSKEY、および DNSKEY 削除の 3 つの段階が定義されています。各ステージは、ADC で実行する必要がある一連のタスクに関連付

けられています。次に、各ステージの説明と実行する必要があるタスクを示します。ここで説明するロールオーバー手順は、キー署名キーとゾーン署名キーの両方に使用できます。

- ステージ **1:** イニシャル。ゾーンには、ゾーンが現在署名されているキーセットのみが含まれます。初期段階でのゾーンの状態は、キーのロールオーバープロセスを開始する直前のゾーンの状態です。

  例**:**

  ゾーン example.com が署名されているキー example.com.zsk1 を考えてみましょう。ゾーンには、example.com.zsk1 キーによって生成された RRSG だけが含まれます。この期限は期限切れになります。キー署名キーは example.com.ksk1 です。

- ステージ **2**: 新しい **DNSKEY**。新しいキーがゾーンで発行され、ゾーンは新しいキーで署名されます。ゾーンには、古いキーと新しいキーによって生成される RRSG が含まれます。ゾーンに両方の RRSG セットを含める必要がある最小期間は、すべての RRSG が期限切れになるまでの時間です。

  例**:**

  新しいキー example.com.zsk2 が example.com ゾーンに追加されます。ゾーンは example.com.zsk2 で署名されています。example.com ゾーンに、両方のキーから生成された RRSG が含まれるようになりました。

  Citrix ADC コマンド

  ADC で次のタスクを実行します。

  - `create dns key`コマンドを使用して DNS キーを作成します。

    例を含む DNS キーの作成の詳細については、「ゾーンの DNS キーを作成する」を参照してください。

  - `add dns key`コマンドを使用して、ゾーンに新しいキーを発行します。

    例など、ゾーンでのキーの公開の詳細については、「ゾーンでの DNS キーの公開」を参照してください。

  - `sign dns zone`コマンドを使用して、新しいキーでゾーンに署名します。

    例を含むゾーンの署名の詳細については、「DNS ゾーンの署名と署名の解除」を参照してください。

- ステージ **3**: **DNSKEY** の取り外し。古い DNS キーによって生成された RRSG の有効期限が切れると、古い DNS キーがゾーンから削除されます。

  例**:**

  古い DNS キー example.com.zsk1 が example.com ゾーンから削除されます。

  Citrix ADC コマンドは以下のとおりです。

  ADC で、`rm dns key` コマンドを使用して、古い DNS キーを削除します。

  例など、ゾーンからキーを削除する方法の詳細については、「DNS キーの削除」を参照してください。

DNSSEC の動作を Citrix ADC にオフロードする

October 7, 2021

DNS サーバーが権限を持っている DNS ゾーンの場合、DNSSEC 操作を ADC アプライアンスにオフロードできます。DNSSEC オフロード配置では、DNS サーバーは署名なし応答を送信します。ADC は、応答をクライアントに中継する前に動的に署名します。また、ADC は符号付き応答をキャッシュします。DNS サーバの負荷を軽減する以外に、DNSSEC の動作を ADC にオフロードすると、次のような利点があります。

- DNS サーバーがプログラムによって生成するレコードに署名できます。このようなレコードは、DNS サーバーで実行されるルーチンゾーン署名操作では署名できません。
- サーバーに DNSSEC を実装していない場合でも、署名付き応答をクライアントに提供できます。

DNSSEC オフロードを設定するには、DNS 負荷分散仮想サーバーを構成し、DNS サーバーを表すサービスを構成してから、サービスを仮想サーバーにバインドする必要があります。DNS 負荷分散仮想サーバーの構成、サービスの構成、および仮想サーバーへのサービスのバインドの詳細については、「DNS ゾーンの構成」を参照してください。

DNSSEC 操作をオフロードする DNS ゾーンごとに、ADC にゾーンエンティティを作成します。DNS ゾーンごとに、プロキシモードおよび DNSSEC オフロードパラメータを有効にする必要があります。オプションで、オフロードゾーンの NSEC レコード生成を構成できます。DNSSEC オフロード用の DNS ゾーンエンティティを作成するには、このトピックの手順に従います。

構成を完了するには、ゾーンの DNS キーを生成し、ゾーンにキーを追加してから、キーを使用してゾーンに署名する必要があります。このプロセスは、通常の DNSSEC と同じです。キーの作成、ゾーンへのキーの追加、およびゾーンの署名については、「ドメインネームシステムのセキュリティ拡張」を参照してください。

DNS オフロードを構成したら、Citrix ADC 上の DNS キャッシュをフラッシュする必要があります。DNS キャッシュをフラッシュすると、キャッシュ内の署名されていないレコードがすべて削除され、署名されたレコードに置き換えられます。DNS キャッシュのフラッシュの詳細については、「DNS レコードのフラッシュ」を参照してください。

CLI を使用してゾーンの DNSSEC オフロードを有効にする

コマンドラインで次のコマンドを入力して、ゾーンの DNSSEC オフロードを有効にし、構成を確認します。

```
1 -  add dns zone <zoneName> -proxyMode YES -dnssecOffload ENABLED [-nsec
       ( ENABLED | DISABLED )
2 -  show dns zone
3 <!--NeedCopy-->
```

例:


```
> add dns zone example.com -proxyMode YES -dnssecOffload ENABLED nsec
   ENABLED
 Done
> show dns zone example.com
      Zone Name : example.com
      Proxy Mode : YES
      DNSSEC Offload: ENABLED    NSEC: ENABLED
 Done
<!--NeedCopy-->
```

GUI を使用してゾーンの **DNSSEC** オフロードを有効にする

1. [トラフィック管理] **>** **[DNS]** **>** **[**ゾーン**]** に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - Citrix ADC にゾーンを作成するには、［追加］をクリックします。
   - 既存のゾーンの DNSSEC オフロードを設定するには、ゾーンをダブルクリックします。
3. [DNS ゾーンの作成] または [DNS ゾーンの構成] ダイアログボックスで、[プロキシモード] および [DNSSEC オフロード] チェックボックスをオンにします。
4. 必要に応じて、Citrix ADC でゾーンの NSEC レコードを生成する場合は、［NSEC］チェックボックスをオンにします。

DNSSEC の管理パーティションのサポート

October 7, 2021

パーティション分割された Citrix ADC アプライアンスでは、生成された DNS キーは次の場所に格納されます。

- デフォルトのパーティション:/nsconfig/dns/
- デフォルト以外のパーティション:/nsconfig/パーティション/ <partitionname> /dns/

これで、DNS キーにパスワードを追加できます。DNS キーにパスワードを追加するには、最初に `create dns key` コマンドでパスワードを追加する必要があります。次に、ADC アプライアンスに DNS キーを追加するときに、`add dns key` コマンドで同じパスワードを入力します。次に例を示します:

```
create dns key -zoneName com -keytype ksK -algorithm rsASHA1 -keysize 4096
- fileNamePrefix com.ksk.rsasha1.4096 -password 1jsfd3Wa

add dns key com.zsk.4096 /nsconfig/dns/com.zsk.rsasha1.4096.private -
password 1jsfd3Wa
```

注:

- デフォルトのパーティション環境の場合、キーはデフォルトから読み取られます location/nsconfig/dns/. ただし、キーが別の場所に保存されている場合は、`add dns key -private` コマンドでパス名を指定する必要があります。例: `add dns key -private <path name>`。
- デフォルト以外のパーティション環境の場合、キーはデフォルトの場所 /nsconfig/partitions/<partitionname>/dns/から読み取られます。

ワイルドカード **DNS** ドメインのサポート

October 7, 2021

ワイルドカード DNS ドメインは、存在しないドメインおよびサブドメインの要求を処理するために使用されます。ゾーンでは、ドメインごとに個別のリソースレコード（RR）を作成する代わりに、ワイルドカードドメインを使用して、存在しないすべてのドメインまたはサブドメインのクエリを特定のサーバーにリダイレクトします。ワイルドカード DNS ドメインの最も一般的な使用方法は、インターネットから他のメールシステムにメールを転送するために使用できるゾーンを作成することです。

DNS 解決では、ワイルドカード RR はワイルドカードドメインをサポートします。ワイルドカード RR は、存在しないドメイン名のクエリーに対する応答を合成するために使用されます。たとえば、http://image.example.comを照会し、サブドメイン「image」が存在しなかった場合、example.com にリダイレクトされる可能性があります。

ワイルドカードレコードには、ドメイン名の左端のラベルとしてアスタリスク (*) 文字があります。たとえば、*.example.comなどです。ドメイン名の他の場所にあるアスタリスクは、ワイルドカード DNS レコードを示します。たとえば、**new**.*.example.com は有効なワイルドカード DNS レコードではありません。

> 注
>
> - ワイルドカードドメインは、Citrix ADC アプライアンスがゾーンに対して権限を持ち、ADNS または DNS プロキシサーバーとして構成されている場合にのみサポートされます。
> - ワイルドカードドメインは、NS レコードおよび SOA レコードではサポートされません。
> - クエリが別のゾーンにある場合は、ワイルドカードドメインを適用できません。
> - ワイルドカードドメインは、QNAME またはワイルドカードドメインと QNAME の間の名前が存在することがわかっている場合は適用できません。

設定例

```
add dns soaRec example.com -originServer n1.example.com -contact admin.
    example.com

add dns nsRec example.com n1.example.com

```



```
add dns nsRec example.com n2.example.com

add dns zone example.com  -proxyMode no

add dns addrec www.example.com 2.2.2.2

add dns addrec *.example.com 10.10.10.10

add dns addrec *.example.com 10.10.10.11

add dns aaaarec *.example.com 2001::1
<!--NeedCopy-->
```

この例では、ワイルドカードドメイン名が A および AAAA レコードに追加されています。

ゾーンに存在するドメイン名のクエリを受信すると、Citrix ADC アプライアンスは対応する応答で応答します。たとえば、www.example.comの場合、アプライアンスは例の 2.2.2.2 で応答します。

ワイルドカード型と一致する存在しないドメイン名の場合、合成された応答が配信されます。

この例では、Citrix ADC アプライアンスは、ドメイン名 nonexist.example.com または xyz.example.com に対して 10.10.10.10 および 10.10.10.11 で応答します。

ワイルドカード合成は、ゾーンに存在するドメイン名には適用されません。

たとえば、クエリ www.example.com およびタイプ AAAA の場合、www.example.com はタイプ A で存在するため、Citrix ADC アプライアンスはワイルドカードで合成されません。この例では、Citrix ADC アプライアンスは NODATA 応答で応答します。

abc.example.com と入力して AAAA と入力するクエリの場合、Citrix ADC アプライアンスは合成された応答で応答します。たとえば、www.example.comの場合、アプライアンスは次のように応答します。2001::1 例では。

DNS DDoS 攻撃の軽減

October 7, 2021

DNS サーバーは、ネットワークの最も重要なコンポーネントの 1 つであり、攻撃から保護する必要があります。DNS 攻撃の最も基本的な種類の 1 つは、DDoS 攻撃です。このタイプの攻撃は増加しており、破壊的である可能性があります。DDoS 攻撃を軽減するには、次の操作を実行できます。

- ネガティブレコードをフラッシュします。
- ネガティブレコードの存続時間（TTL）を制限します。
- DNS キャッシュによって消費されるメモリを制限することにより、Citrix ADC メモリを保持します。
- DNS レコードをキャッシュに保持します。
- DNS キャッシュバイパスを有効にします。

ネガティブ・レコードのフラッシュ

DNS 攻撃は、キャッシュに負のレコード (NXDOMAIN および NODATA) を埋め尽くします。その結果、正当な要求に対する応答はキャッシュされないため、新しい要求は DNS 解決のためにバックエンドサーバーに送信されます。したがって、応答が遅れます。

Citrix ADC アプライアンスの DNS キャッシュからネガティブ DNS レコードをフラッシュできるようになりました。

CLI を使用したネガティブ・キャッシュ・レコードのフラッシュ

コマンドプロンプトで入力します。

```
flush dns proxyrecords -type (dnsRecordType | negRecType)NXDOMAIN | NODATA
```

例**:**

```
flush dns proxyrecords -negRecType NODATA
```

GUI を使用したネガティブ・キャッシュ・レコードのフラッシュ

1. [設定] > [トラフィック管理] > [**DNS**] > [レコード] に移動します。
2. 詳細ペインで、[プロキシレコードのフラッシュ] をクリックします。
3. 「フラッシュ・タイプ」ボックスで、「ネガティブ・レコード」を選択します。
4. [ネガティブレコードタイプ] ボックスで、[**NXDOMAIN**] または [**NODATA**] のいずれかを選択します。

ランダムなサブドメインおよび **NXDOMAIN** 攻撃に対する保護

ランダムなサブドメインおよび NXDOMAIN 攻撃を防ぐために、DNS キャッシュメモリを制限し、負のレコードの TTL 値を調整できます。

DNS キャッシュによって消費されるメモリの量を制限するには、最大キャッシュサイズ（MB 単位）と、否定応答を格納するためのキャッシュサイズ（MB 単位）を指定します。いずれかの制限に達すると、それ以上のエントリはキャッシュに追加されません。また、syslog メッセージが記録され、SNMP トラップが設定されている場合は SNMP トラップが生成されます。これらの制限が設定されていない場合、システムメモリを使い果たすまでキャッシュが続行されます。

負のレコードの TTL 値が高いと、長期間価値のないレコードが保存される可能性があります。TTL 値が低いほど、バックエンドサーバーに要求が送信されます。

負のレコードの TTL は、TTL 値または SOA レコードの「有効期限」値のいずれか小さい方の値に設定されます。

注:

- この制限は、パケットエンジンごとに追加されます。たとえば、maxCacheSize が 5 MB に設定され、アプライアンスに 3 つのパケットエンジンがある場合、合計キャッシュサイズは 15 MB になります。
- 負のレコードのキャッシュサイズは、最大キャッシュサイズ以下である必要があります。

- DNS キャッシュメモリの制限を、すでにキャッシュされているデータの量よりも低い値に減らすと、データが期限切れになるまで、キャッシュサイズは制限を超えたままになります。つまり、TTL0 を超えるか、フラッシュされます（`flush dns proxyrecords` コマンド、または Citrix ADC GUI のプロキシレコードのフラッシュ）。
- SNMP トラップを構成するには、「SNMP トラップを生成するように NetScaler を構成する」を参照してください。

CLI を使用して **DNS** キャッシュによって消費されるメモリを制限する

コマンドプロンプトで入力します。

```
set dns parameter -maxCacheSize <MBytes> -maxNegativeCacheSize <MBytes>
```

例**:**

```
set dns parameter - maxCacheSize 100 -maxNegativeCacheSize 25
```

GUI を使用して **DNS** キャッシュによって消費されるメモリを制限する

[設定] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックして、次のパラメータを設定します。

- 最大キャッシュサイズ（MB）
- 負の最大キャッシュサイズ (MB)

CLI を使用して、ネガティブレコードの **TTL** を制限します

コマンドプロンプトで入力します。

```
set dns parameter -maxnegcacheTTL <secs>
```

例**:**

```
set dns parameter -maxnegcacheTTL 360
```

GUI を使用して、ネガティブレコードの **TTL** を制限します

1. [設定] > [トラフィック管理] > [**DNS**] に移動します。
2. [**DNS** 設定の変更] をクリックし、[ネガティブキャッシュの最大 **TTL (**秒**)**] パラメータを設定します。

DNS レコードをキャッシュに保持する

攻撃により、DNS キャッシュが重要でないエントリでいっぱいになる可能性がありますが、すでにキャッシュされている正当なレコードがフラッシュされて、新しいエントリ用のスペースが確保される可能性があります。攻撃が無効なデータをキャッシュに埋め込むのを防ぐために、正規のレコードが TTL 値を超えた後でも保持できます。

cacheNoExpire パラメータを有効にした場合、キャッシュ内の現在のレコードは、パラメータを無効にするまで保持されます。

注:

- このオプションは、最大キャッシュサイズが指定されている場合にのみ使用できます (maxCacheSize パラメータ)。
- maxnegcacheTTL が設定されていて、キャッシュ NoExpire が有効になっている場合、キャッシュ NoExpire が優先されます。

CLI を使用して DNS レコードをキャッシュに保存する

コマンドプロンプトで入力します。

```
set dns parameter -cacheNoExpire ( ENABLED | DISABLED)
```

例:

```
set dns parameter -cacheNoExpire ENABLED
```

GUI を使用して DNS レコードをキャッシュに保存する

1. [設定] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2. [キャッシュの有効期限なし] を選択します。

DNS キャッシュバイパスを有効にする

DNS 要求の可視性と制御を向上させるには、cacheHitBypass パラメーターを設定して、すべての要求をバックエンドサーバーに転送し、キャッシュを構築できるようにしますが、使用しないようにします。キャッシュが構築されたら、パラメータを無効にして、リクエストがキャッシュから送られるようにすることができます。

CLI を使用して DNS キャッシュバイパスを有効にする

コマンドプロンプトで入力します。

```
set dns parameter -cacheHitBypass ( ENABLED | DISABLED )
```

例:

```
set dns parameter -cacheHitBypass ENABLED
```

GUI を使用して DNS キャッシュバイパスを有効にする

1. [設定] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2. [キャッシュヒットバイパス] を選択します。

Slowloris 攻撃を防ぐ

複数のパケットにまたがる DNS クエリは、Slowloris 攻撃の潜在的な脅威を示します。Citrix ADC アプライアンスは、複数のパケットに分割された DNS クエリをサイレントにドロップできます。

クエリが複数のパケットに分割されている場合は、splitPktQueryProcessing パラメータを ALLOW または DROP に設定できます。

注意: この設定は、DNS TCP にのみ適用されます。

CLI を使用して DNS クエリを 1 つのパケットに制限する

コマンドプロンプトで入力します。

```
set dns parameter -splitPktQueryProcessing ( ALLOW | DROP )
```

例:

```
set dns parameter -splitPktQueryProcessing DROP
```

GUI を使用して DNS クエリを 1 つのパケットに制限する

1. [設定] > [トラフィック管理] > [**DNS**] に移動し、[**DNS** 設定の変更] をクリックします。
2. [分割パケットクエリ処理] ボックスで、[**ALLOW**] または [**DROP**] を選択します。

キャッシュから提供される DNS 応答の統計情報を収集する

キャッシュから提供される DNS 応答の統計を収集できます。次に、これらの統計を使用して、より多くの DNS トラフィックがドロップされるしきい値を作成し、帯域幅ベースのポリシーでこのしきい値を適用します。以前は、キャッシュから提供された要求の数が報告されなかったため、DNS 負荷分散仮想サーバーの帯域幅の計算は正確ではありませんでした。

プロキシモードでは、要求バイト、応答バイト、受信した合計パケット数、および送信した合計パケット数の統計情報が継続的に更新されます。以前は、特に DNS 負荷分散仮想サーバーでは、これらの統計情報が常に更新されるとは限りません。

プロキシモードでは、キャッシュから提供される DNS 応答の数も決定できるようになりました。これらの統計を収集するために、次のオプションが stat lb vserver <DNSvirtualServerName> コマンドに追加されました。

- リクエスト –DNS またはが受信したリクエストの総数 DNS_TCP 仮想サーバー。バックエンドに転送された要求と、キャッシュから応答された要求が含まれます。
- **Vserver** ヒット –バックエンドに転送されたリクエストの総数。キャッシュから提供されるリクエストの数は、リクエストの総数と仮想サーバーから提供されるリクエストの数の差です。

- 応答 –この仮想サーバーによって送信された応答の総数。たとえば、DNS LB 仮想サーバーが 5 つの DNS 要求を受信し、そのうちの 3 つをバックエンドに転送し、キャッシュから 2 つを提供した場合、これらの統計の対応する値は次のようになります。
  - **Vserver** ヒット: 3
  - リクエスト: 5
  - 反応: 5

ファイアウォール負荷分散

October 7, 2021

ファイアウォールロードバランシングは、複数のファイアウォールにトラフィックを分散し、フォールトトレランスとスループットを向上させます。ファイアウォールの負荷分散は、次の方法でネットワークを保護します。

- ファイアウォール間の負荷を分割することで、単一障害点を排除し、ネットワークの拡張が可能になります。
- 高可用性の向上。

ファイアウォール負荷分散のための Citrix ADC アプライアンスの構成は、負荷分散の構成と似ていますが、推奨されるサービスタイプは ANY、推奨されるモニタータイプは PING で、負荷分散仮想サーバーモードは MAC に設定されています。

サンドイッチ、エンタープライズ、または複数ファイアウォールの環境構成で、ファイアウォールの負荷分散を設定できます。サンドイッチ環境は、外部からネットワークに入るトラフィックとネットワークをインターネットに送るトラフィックの負荷分散に使用されます。この環境では、ファイアウォールの各側に 1 つずつ、2 つの Citrix ADC アプライアンスを構成します。ネットワークをインターネットに送信するトラフィックをロードバランシングするためのエンタープライズ環境を設定します。エンタープライズ環境では、内部ネットワークとインターネットへのアクセスを提供するファイアウォールとの間に単一の Citrix ADC アプライアンスを構成します。マルチファイアウォール環境は、別のファイアウォールからのトラフィックのロードバランシングに使用されます。Citrix ADC アプライアンスの両側でファイアウォールの負荷分散を有効にすると、出力方向と入力方向のトラフィックフローが改善され、トラフィックの処理速度が向上します。複数のファイアウォール環境では、2 つのファイアウォール間に挟まれた Citrix ADC アプライアンスを構成します。

重要：Citrix ADC アプライアンスで宛先 IP アドレスに静的ルートを設定し、L3 モードを有効にすると、Citrix ADC アプライアンスはトラフィックを負荷分散仮想サーバーに送信する代わりに、ルーティングテーブルを使用してトラフィックをルーティングします。

注：FTP が機能するためには、Citrix ADC アプライアンス上で追加の仮想サーバーまたはサービスを構成する必要があります。IP アドレスとポートは *、21 で、サービスタイプは FTP として指定されます。この場合、Citrix ADC アプライアンスは、FTP 制御接続を受け入れ、ペイロードを変更し、データ接続を管理することで、FTP プロトコルをすべて同じファイアウォール経由で管理します。

ファイアウォールの負荷分散は、Citrix ADC アプライアンスでサポートされている負荷分散方法の一部のみをサポートします。また、いくつかのタイプの永続性とモニターのみを構成できます。

ファイアウォールロードバランシング方式

ファイアウォールの負荷分散では、次の負荷分散方式がサポートされています。

- 最小接続数
- ラウンドロビン
- 最小パケット
- 最小帯域幅
- 送信元 IP ハッシュ
- 宛先 IP ハッシュ
- 送信元 IP 宛先 IP ハッシュ
- 送信元 IP 送信元ポートのハッシュ
- 最小応答時間方式（LRTM）
- カスタムロード

ファイアウォールの永続性

ファイアウォールロードバランシングでは、SOURCEIP、DESTIP、および SOURCEIPDESTIP ベースの永続性だけがサポートされます。

ファイアウォールサーバのモニタリング

ファイアウォールロードバランシングでは、PING モニタとトランスペアレントモニタだけがサポートされます。ファイアウォールを表すバックエンドサービスに PING モニタ（デフォルト）をバインドできます。ファイアウォールが ping パケットに応答しないように設定されている場合、透過モニタを設定して、信頼側のホストを個々のファイアウォール経由で監視できます。

サンドイッチ環境

October 7, 2021

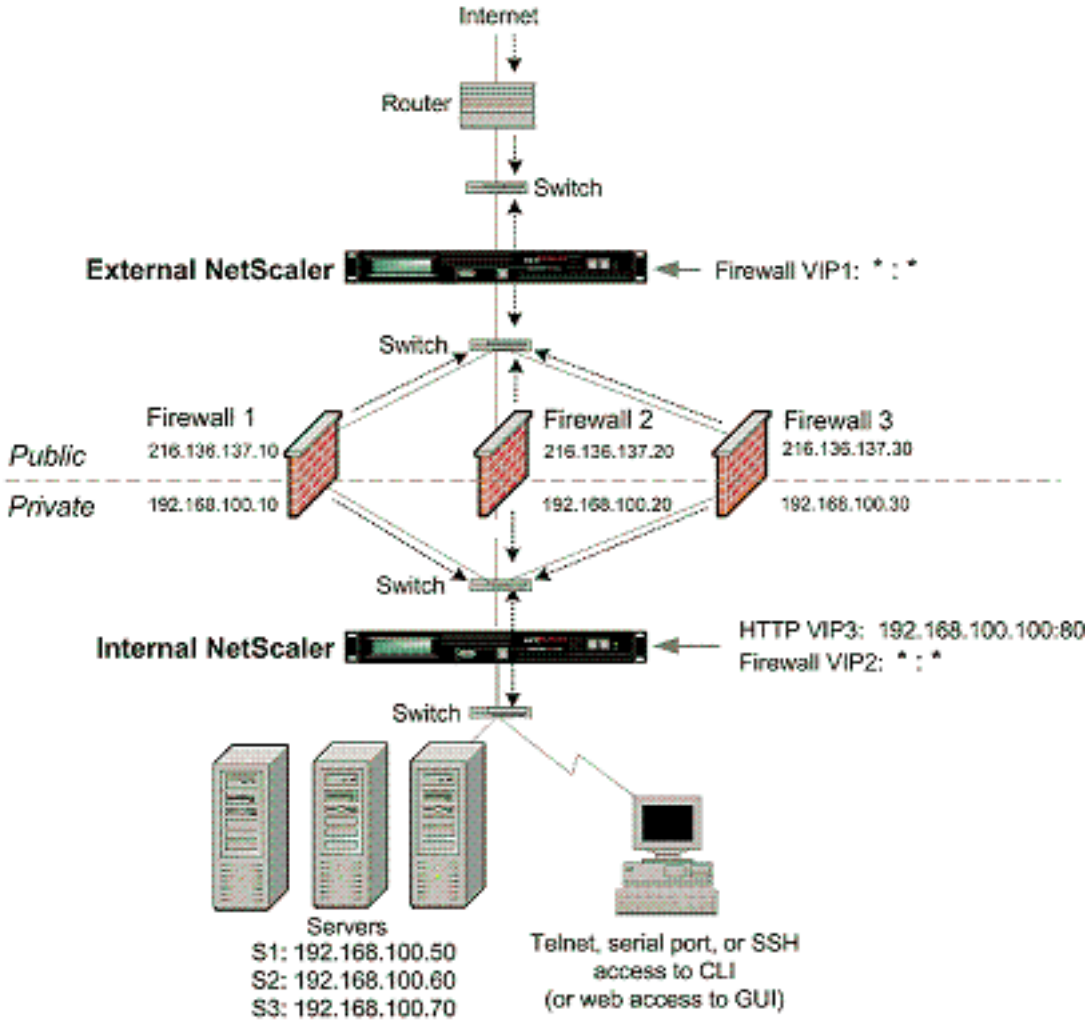
サンドイッチモードでの Citrix ADC の展開では、ファイアウォールを通過するネットワークトラフィックを、入力（インターネットなどの外部からネットワークに入るトラフィック）と出力（ネットワークからインターネットに出るトラフィック）の両方向で負荷分散できます。

この設定では、Citrix ADC が一連のファイアウォールの両側に配置されています。ファイアウォールとインターネットの間に配置された Citrix ADC は、入力トラフィックを処理する外部 Citrix ADC と呼ばれ、構成された方法に基づいて最適なファイアウォールを選択します。ファイアウォールとプライベートネットワーク間の Citrix ADC は、内部 Citrix ADC と呼ばれるセッションの最初のパケットを受信するファイアウォールを追跡します。次に、そのセッションの後続のすべてのパケットが同じファイアウォールに送信されることを確認します。

内部 Citrix ADC を通常のトラフィックマネージャーとして構成して、プライベートネットワークサーバー間でトラフィックを負荷分散できます。この設定では、プライベートネットワーク（出力）から発信されるトラフィックを、ファイアウォール全体でロードバランシングすることもできます。

次の図は、サンドイッチファイアウォールの負荷分散環境を示しています。

図 1: ファイアウォール負荷分散（サンドイッチ）



サービスタイプ ANY は、すべてのトラフィックを受け入れるように Citrix ADC を構成します。

HTTP および TCP に関連する利点を活用するには、サービスおよび仮想サーバを HTTP または TCP タイプで構成します。FTP が機能するには、FTP タイプを使用してサービスを構成します。

サンドイッチ環境での外部 **Citrix ADC** 構成

サンドイッチ環境で外部 Citrix ADC を構成するには、次のタスクを実行します。

- ロードバランシング機能を有効にします。
- ファイアウォールごとにワイルドカードサービスを構成します。

- ワイルドカードサービスごとにモニタを設定します。
- インターネットからのトラフィック用にワイルドカード仮想サーバーを構成します。
- MAC 書き換えモードで仮想サーバを設定します
- ワイルドカードの仮想サーバにサービスをバインドします。
- 設定を保存して確認します。

負荷分散機能の有効化

コマンドラインインターフェイスを使用して負荷分散を有効にするには

コマンドプロンプトで次のコマンドを入力して、負荷分散を有効にし、構成を確認します。

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

例**:**

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散を有効にするには

[システム**] > [**設定**]** に移動し、[基本機能の構成] で [負荷分散] を選択します。

ファイアウォールごとにワイルドカードサービスを構成する

コマンドラインインターフェイスを使用してファイアウォールごとにワイルドカードサービスを構成するには

コマンドプロンプトで入力します。

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

構成ユーティリティを使用して各ファイアウォールにワイルドカードサービスを構成するには

Traffic Management > Load Balancing > Services に移動してサービスを追加します。[プロトコル] フィールドに **ANY** を指定し、[ポート] フィールドで [*] を指定します。

ワイルドカードサービスごとにモニタを構成する

PING モニターは、デフォルトでサービスにバインドされます。個々のファイアウォールを介して信頼できる側のホストを監視するには、透過的なモニターを構成する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、Citrix ADC アプライアンスとアップストリームデバイス間の接続のみを監視します。トランスペアレントモニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスへのパスに存在するすべてのデバイスを監視します。トランスペアレントモニタが設定されておらず、ファイアウォールのステータスが UP であるにもかかわらず、そのファイアウォールからのネクストホップデバイスのいずれかがダウンしている場合、アプライアンスはロードバランシングの実行中にファイアウォールを含み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスのいずれかがダウンしているため、パケットは最終的な宛先に配信されません。トランスペアレントモニタをバインドすることにより、いずれかのデバイス（ファイアウォールを含む）がダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォールロードバランシングを実行するときにファイアウォールは含まれません。

透過モニターをバインドすると、PING モニターが上書きされます。トランスペアレントモニタに加えて PING モニタを構成するには、トランスペアレントモニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

コマンドラインインターフェイスを使用してトランスペアレントモニタを構成するには

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。


```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

例:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
To bind a PING monitor, type the following command:
bind monitor PING fw-svc1
<!--NeedCopy-->
```

構成ユーティリティを使用してトランスペアレントモニタを作成してバインドするには

[トラフィック管理] > [負荷分散] > [モニター] に移動し、トランスペアレントモニターを作成してバインドします。

インターネットからのトラフィック用にワイルドカード仮想サーバーを構成する

コマンドラインインターフェイスを使用して、インターネットからのトラフィック用にワイルドカード仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

構成ユーティリティを使用して、インターネットからのトラフィックに対してワイルドカード仮想サーバーを構成するには

Traffic Management > Load Balancing > Virtual Servers に移動してワイルドカード仮想サーバーを作成します。[プロトコル] フィールドに **ANY** を指定し、[ポート] フィールドで [*] を指定します。

MAC 書き換えモードで仮想サーバを設定します

コマンドラインインターフェイスを使用して **MAC** 書き換えモードで仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

構成ユーティリティを使用して **MAC** 書き換えモードで仮想サーバを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動してワイルドカード仮想サーバーを作成します。
2. **[**基本設定**]** セクションを編集し、[その他] をクリックします。
3. [リダイレクションモード] ドロップダウンリストから、[**MAC Based**] を選択します。

ワイルドカード仮想サーバーへのサービスのバインド

コマンドラインインターフェイスを使用してサービスをワイルドカード仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスをワイルドカード仮想サーバーにバインドするには

1. トラフィック管理 **>** 負荷分散 **>** 仮想サーバーに移動し、サービスをバインドする仮想サーバーを選択します。
2. 「サービス」セクションをクリックし、バインドするサービスを選択します。

設定の保存と確認

設定タスクが完了したら、必ず設定を保存してください。設定が正しいことを確認してください。

コマンドラインインターフェイスを使用して構成を保存および確認するには

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。

```
save ns config
show vserver
<!--NeedCopy-->
```

例:

```
save config
sh lb vserver FWLBVIP1
FWLBVIP1 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 06:40:14 2010
        Time since last state change: 0 days, 00:00:11.240
        Effective State: UP  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: SRCIPDESTIPHASH
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw_svc_1 (10.102.29.251: *) - ANY State: UP  Weight: 1
2) fw_svc_2 (10.102.29.18: \*) - ANY State: UP   Weight: 1
 Done
show service fw-svc1
        fw-svc1 (10.102.29.251:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 10:04:50 2010
        Time since last state change: 0 days, 00:00:38.120
        Server Name: 10.102.29.251
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
```



```
29          Client Keepalive(CKA): NO
30          Access Down Service: NO
31          TCP Buffering(TCPB): YES
32          HTTP Compression(CMP): NO
33          Idle timeout: Client: 120 sec   Server: 120 sec
34          Client IP: DISABLED
35          Cacheable: NO
36          SC: OFF
37          SP: OFF
38          Down state flush: ENABLED
39
40  1)      Monitor Name: monitor-HTTP-1
41                  State: DOWN     Weight: 1
42                  Probes: 5       Failed [Total: 5 Current: 5]
43                  Last response: Failure - Time out during TCP connection
                        establishment stage
44                  Response Time: 2000.0 millisec
45  2)      Monitor Name: ping
46                  State: UP       Weight: 1
47                  Probes: 3       Failed [Total: 0 Current: 0]
48                  Last response: Success - ICMP echo reply received.
49                  Response Time: 1.415 millisec
50   Done
51  <!--NeedCopy-->
```

サンドイッチ環境での内部 **Citrix ADC** 構成

サンドイッチ環境で内部 Citrix ADC を構成するには、次のタスクを実行します。

サーバからのトラフィック（出力）

- ロードバランシング機能を有効にします。
- ファイアウォールごとにワイルドカードサービスを構成します。
- ワイルドカードサービスごとにモニタを設定します。
- ワイルドカード仮想サーバを設定して、ファイアウォールに送信されるトラフィックの負荷を分散します。
- MAC 書き換えモードで仮想サーバを設定します
- ファイアウォールサービスをワイルドカード仮想サーバーにバインドします。

プライベートネットワークサーバー間のトラフィック用

- 各仮想サーバのサービスを構成します。
- サービスごとにモニタを設定します。
- サーバに送信されるトラフィックのバランスをとるように HTTP 仮想サーバを設定します。
- HTTP サービスを HTTP 仮想サーバにバインドします。
- 設定を保存して確認します。

負荷分散機能の有効化

負荷分散機能が無効になっている場合、サービスや仮想サーバーなどの負荷分散エンティティを構成できます。ただし、この機能を有効にするまで機能しません。

コマンドラインインターフェイスを使用して負荷分散を有効にするには

コマンドプロンプトで次のコマンドを入力して、負荷分散を有効にし、構成を確認します。

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散を有効にするには

[システム] > [設定] に移動し、[基本機能の構成] で [負荷分散] を選択します。

ファイアウォールごとにワイルドカードサービスを構成する

コマンドラインインターフェイスを使用してファイアウォールごとにワイルドカードサービスを構成するには

コマンドプロンプトで入力します。


```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

構成ユーティリティを使用して各ファイアウォールにワイルドカードサービスを構成するには

Traffic Management > Load Balancing > Services に移動してサービスを追加します。[プロトコル] フィールドに **ANY** を指定し、[ポート] フィールドで [*] を指定します。

ワイルドカードサービスごとにモニタを構成する

PING モニターは、デフォルトでサービスにバインドされます。個々のファイアウォールを介して信頼できる側のホストを監視するには、透過的なモニターを構成する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、Citrix ADC アプライアンスとアップストリームデバイス間の接続のみを監視します。トランスペアレントモニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスへのパスに存在するすべてのデバイスを監視します。トランスペアレントモニタが設定されておらず、ファイアウォールのステータスが UP であるにもかかわらず、そのファイアウォールからのネクストホップデバイスのいずれかがダウンしている場合、アプライアンスはロードバランシングの実行中にファイアウォールを含み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスのいずれかがダウンしているため、パケットは最終的な宛先に配信されません。トランスペアレントモニタをバインドすることにより、いずれかのデバイス（ファイアウォールを含む）がダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォールロードバランシングを実行するときにファイアウォールは含まれません。

透過モニターをバインドすると、PING モニターが上書きされます。トランスペアレントモニタに加えて PING モニタを構成するには、トランスペアレントモニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

コマンドラインインターフェイスを使用してトランスペアレントモニタを構成するには

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```


例:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

構成ユーティリティを使用してトランスペアレントモニタを作成してバインドするには

1. **[**トラフィック管理**] > [**負荷分散**] > [**モニター**]** に移動し、モニターを作成します。
2. [モニターの作成] ダイアログボックスで、必要なパラメーターを入力し、[透過**]** を選択します。

ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散する

コマンドラインインターフェイスを使用してファイアウォールに送信されるトラフィックを負荷分散するようにワイルドカード仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

構成ユーティリティを使用して、インターネットからのトラフィックに対してワイルドカード仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動してワイルドカード仮想サーバーを作成します。
2. [プロトコル] フィールドに **ANY** を指定し、* [ポート] フィールド。**

構成ユーティリティを使用してファイアウォールに送信されるトラフィックの負荷を分散するようにワイルドカード仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。

2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次に示すパラメータの値を指定します。
   - 名前—name
4. [プロトコル] で [ANY] を選択し、[IP アドレス] と [ポート] で [*] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ウィンドウに表示されます。

MAC 書き換えモードで仮想サーバを設定します

コマンドラインインターフェイスを使用して **MAC** 書き換えモードで仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

構成ユーティリティを使用して **MAC** 書き換えモードで仮想サーバを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動してワイルドカード仮想サーバーを作成します。
2. **[**基本設定**]** セクションを編集し、[その他] をクリックします。
3. [リダイレクションモード] ドロップダウンリストから、[**MAC Based**] を選択します。

ファイアウォールサービスをワイルドカード仮想サーバーにバインドする

コマンドラインインターフェイスを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```


例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには

1. **[**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動し、仮想サーバーを選択します。
2. [サービス] セクションをクリックして、バインドするサービスを選択します。

注: サービスを複数の仮想サーバーにバインドできます。

各仮想サーバのサービスを構成する

コマンドラインインターフェイスを使用して各仮想サーバーのサービスを構成するには

コマンドプロンプトで入力します。

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

構成ユーティリティを使用して各仮想サーバのサービスを構成するには

1. **[**トラフィック管理**] > [**負荷分散**] > [**サービス**]** に移動し、各仮想サーバーのサービスを構成します。
2. [プロトコル**]** フィールドで **HTTP** を指定し、[使用可能なモニター**]** で [**HTTP]** を選択します。

構成ユーティリティを使用して各仮想サーバのサービスを構成するには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [サービスの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - サービス名: 名前
   - サーバ: サーバ名

- ポートポート

4. 「プロトコル」で、HTTP を指定します。[使用可能なモニター] で、[HTTP] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [Services] ペインに表示されます。

サービスごとにモニターを構成する

コマンドラインインターフェイスを使用してモニタをサービスにバインドするには

コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

例:

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してモニタをサービスにバインドするには

[トラフィック管理**] > [**負荷分散**] > [**サービス**]** に移動し、サービスをダブルクリックしてモニタを追加します。

サーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成する

コマンドラインインターフェイスを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```


構成ユーティリティを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

1. **[**トラフィック管理**] > [**負荷分散**] > [**仮想サービス**]** に移動し、HTTP 仮想サーバーを設定します。
2. [プロトコル] フィールドに **HTTP** を指定します。

構成ユーティリティを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次に示すパラメータの値を指定します。
   - 名前—name
   - IP アドレス-IP アドレス
     注：仮想サーバーが IPv6 を使用している場合は、
     IPv6 チェックボックスを選択し、アドレスを IPv6 形式で入力します（たとえば、1000:0000:0000:0000:0005:0600:700a:888b ）。
     **
   - ポートポート
4. [プロトコル] で、[HTTP] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ウィンドウに表示されます。

設定の保存と確認

設定タスクが完了したら、必ず設定を保存してください。また、設定が正しいことを確認する必要があります。

コマンドラインインターフェイスを使用して構成を保存および確認するには

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。

- `save ns config`
- `show vserver`

例**:**

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
```



```
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
```



```
51   Done
52 <!--NeedCopy-->
```

構成ユーティリティを使用して構成を保存および確認するには

1. [詳細] ペインで、[保存] をクリックします。
2. [構成の保存] ダイアログボックスで、[はい] をクリックします。
3. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
4. [詳細] ペインで、手順 5 で作成した仮想サーバーを選択します。
5. 詳細ペインに表示される設定が正しいことを確認します。
6. **Traffic Management > Load Balancing > Services** に移動します。
7. [詳細] ペインで、手順 5 で作成したサービスを選択します。
8. 詳細ペインに表示される設定が正しいことを確認します。

サンドイッチ環境で設定されたファイアウォール負荷分散の監視

構成がアップして実行されたら、各サービスと仮想サーバーの統計情報を表示して、考えられる問題をチェックする必要があります。

仮想サーバの統計情報の表示

仮想サーバーのパフォーマンスを評価したり、問題のトラブルシューティングを行うために、Citrix ADC アプライアンスで構成された仮想サーバーの詳細を表示できます。すべての仮想サーバの統計情報のサマリーを表示することも、仮想サーバの名前を指定して、その仮想サーバの統計情報だけを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバの状態
- 受信したリクエストの割合
- ヒット率

コマンドラインインターフェイスを使用して仮想サーバーの統計情報を表示するには

Citrix ADC に現在構成されているすべての仮想サーバー、または単一の仮想サーバーの統計情報のサマリーを表示するには、コマンドプロンプトで次のように入力します。

```
1 stat lb vserver [-detail] [<name>]
2 <!--NeedCopy-->
```


例:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
         0/s
Two                          *     0          TCP         DOWN      0/s
         0/s
Three                        *  2598          TCP         DOWN      0/s
         0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
         0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
         0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
         0/s
 Done

<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバの統計情報を表示するには

1. **Traffic Management > Load Balancing > Virtual Servers > Statistics** に移動します。
2. 1 つの仮想サーバーのみの統計を表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計情報の表示

サービス統計情報を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバ接続などを表示できます。

コマンドラインインターフェイスを使用してサービスの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat service <name>
<!--NeedCopy-->
```


例:

```
1 stat service Service-HTTP-1
2 <!--NeedCopy-->
```

構成ユーティリティを使用してサービスの統計情報を表示するには

1. **Traffic Management > Load Balancing > Virtual Servers > Statistics** に移動します。
2. 1 つのサービスのみの統計を表示する場合は、サービスを選択して、[統計] をクリックします。

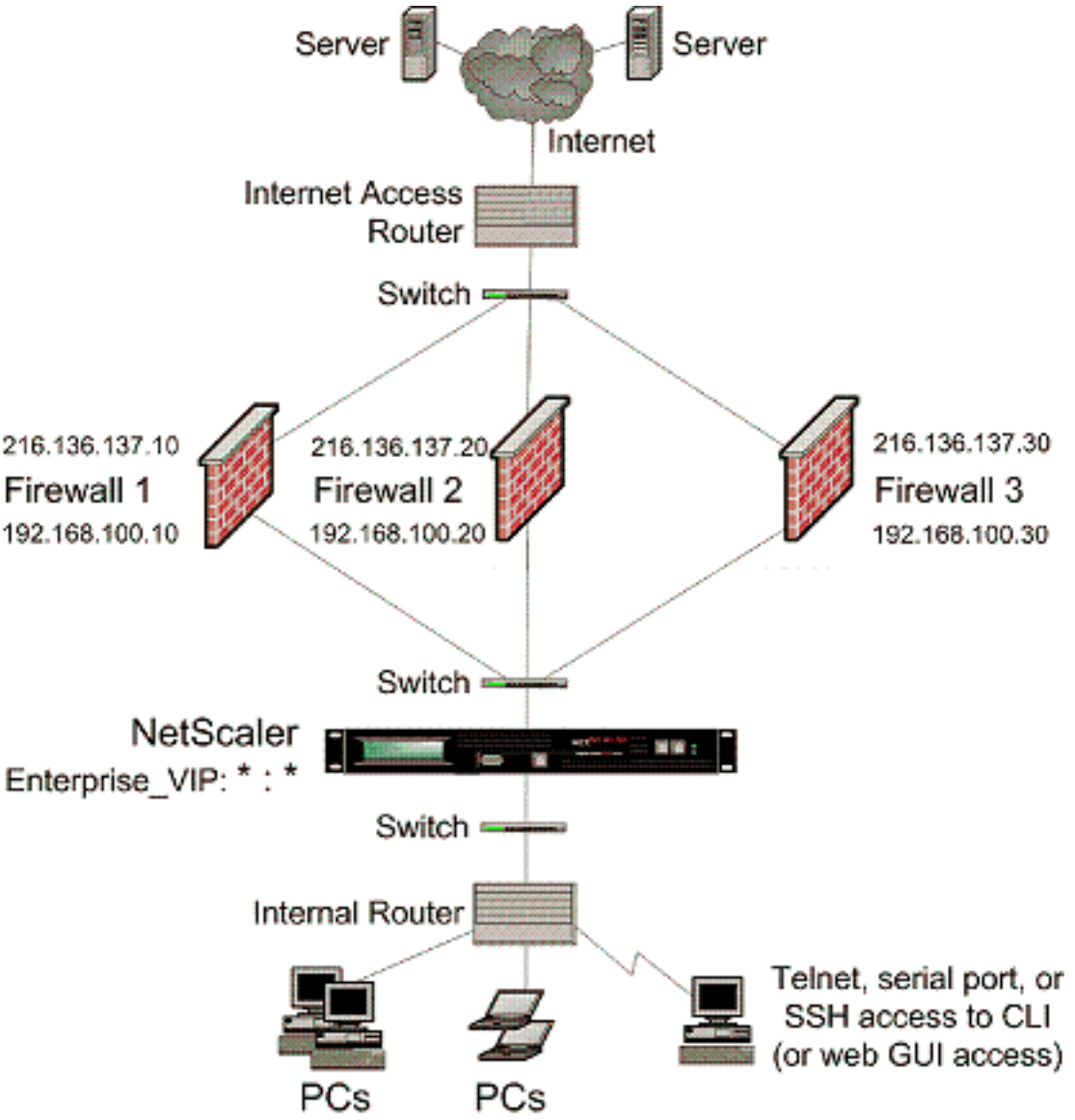
エンタープライズ環境

October 7, 2021

企業のセットアップでは、Citrix ADC は、パブリックインターネットに接続されたファイアウォールと内部プライベートネットワークの間に配置され、出力トラフィックを処理します。Citrix ADC は、設定された負荷分散ポリシーに基づいて、最適なファイアウォールを選択します。

次の図は、エンタープライズファイアウォールの負荷分散環境を示しています。

図 1: ファイアウォール負荷分散（エンタープライズ）



サービスタイプ ANY は、すべてのトラフィックを受け入れるように Citrix ADC を構成します。

HTTP および TCP に関連する利点を活用するには、サービスと vserver を HTTP または TCP タイプで設定します。FTP が機能するには、FTP タイプを使用してサービスを構成します。

エンタープライズ環境での **Citrix ADC** 構成

エンタープライズ環境で Citrix ADC を構成するには、次のタスクを実行します。

サーバからのトラフィック（出力）

- ロードバランシング機能を有効にします。
- ファイアウォールごとにワイルドカードサービスを構成します。
- ワイルドカードサービスごとにモニタを設定します。

- ワイルドカード仮想サーバを設定して、ファイアウォールに送信されるトラフィックの負荷を分散します。
- MAC 書き換えモードで仮想サーバを設定します
- ファイアウォールサービスをワイルドカード仮想サーバーにバインドします。

プライベートネットワークサーバー間のトラフィック用

- 各仮想サーバのサービスを構成します。
- サービスごとにモニタを設定します。
- サーバに送信されるトラフィックのバランスをとるように HTTP 仮想サーバを設定します。
- HTTP サービスを HTTP 仮想サーバにバインドします。
- 設定を保存して確認します。

負荷分散機能の有効化

負荷分散機能が無効になっている場合は、サービスや仮想サーバなどの負荷分散エンティティを設定できますが、機能を有効にするまで機能しません。

コマンドラインインターフェイスを使用して負荷分散を有効にするには

コマンドプロンプトで次のコマンドを入力して、負荷分散を有効にし、構成を確認します。

- enable ns feature LB
- show ns feature

例:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```


構成ユーティリティを使用して負荷分散を有効にするには

System > Settings に移動して Configure Basic Features で Load Balancing を選択します。

ファイアウォールごとにワイルドカードサービスを構成する

コマンドラインインターフェイスを使用してファイアウォールごとにワイルドカードサービスを構成するには

コマンドプロンプトで入力します。

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

構成ユーティリティを使用して各ファイアウォールにワイルドカードサービスを構成するには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [サービスの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - サービス名：名前
   - サーバ：サーバ名
4. [プロトコル] で [ANY] を選択し、[ポート] で [*] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [Services] ペインに表示されます。

ワイルドカードサービスごとにモニタを構成する

PING モニターは、デフォルトでサービスにバインドされます。個々のファイアウォールを介して信頼側のホストを監視するように、トランスペアレントモニタを構成する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、Citrix ADC アプライアンスとアップストリームデバイス間の接続のみを監視します。トランスペアレントモニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスへのパスに存在するすべてのデバイスを監視します。トランスペアレントモニタが設定されておらず、ファイアウォールのステータスが UP であるにもかかわらず、そのファイアウォールからのネクストホップデバイスのいずれかがダウンしている場合、アプライアンスはロードバランシングの実行中にファイアウォールを含み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスのいずれかがダウンしているため、パケットは最

終的な宛先に配信されません。トランスペアレントモニタをバインドすることにより、いずれかのデバイス（ファイアウォールを含む）がダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォールロードバランシングを実行するときにファイアウォールは含まれません。

トランスペアレントモニタをバインドすると、PING モニタが上書きされます。トランスペアレントモニタに加えて PING モニタを構成するには、トランスペアレントモニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

コマンドラインインターフェイスを使用してトランスペアレントモニタを構成するには

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

例:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

構成ユーティリティを使用してトランスペアレントモニタを作成してバインドするには

1. Traffic Management > Load Balancing > Monitors に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [モニタの作成] ダイアログボックスで、次のように値を指定します。
   - Name*
   - タイプ *—タイプ
   - 接続先 IP
   - 透明

   -* 必須パラメータ
4. [Create] をクリックしてから、[Close] をクリックします。[Monitors] ペインで、構成したモニタを選択し、画面の下部に表示される設定が正しいことを確認します。

ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散する

コマンドラインインターフェイスを使用してファイアウォールに送信されるトラフィックを負荷分散するようにワイルドカード仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

構成ユーティリティを使用してファイアウォールに送信されるトラフィックの負荷を分散するようにワイルドカード仮想サーバーを構成するには

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次に示すパラメータの値を指定します。
   - 名前—name
4. [プロトコル] で [ANY] を選択し、[IP アドレス] と [ポート] で [*] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ウィンドウに表示されます。

MAC 書き換えモードで仮想サーバを設定します

コマンドラインインターフェイスを使用して **MAC** 書き換えモードで仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```


構成ユーティリティを使用して **MAC** 書き換えモードで仮想サーバを構成するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ウィンドウで、リダイレクションモードを構成する仮想サーバー (Vserver-LB-1 など) を選択し、[開く] をクリックします。
3. [詳細設定] タブの [リダイレクションモード] で、[MAC ベース] をクリックします。
4. ［OK］をクリックします。

ファイアウォールサービスをワイルドカード仮想サーバーにバインドする

コマンドラインインターフェイスを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してファイアウォールサービスをワイルドカード仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. [サービス] セクションをクリックして、バインドするサービスを選択します。

注: サービスを複数の仮想サーバーにバインドできます。

各仮想サーバのサービスを構成する

コマンドラインインターフェイスを使用して各仮想サーバーのサービスを構成するには

コマンドプロンプトで入力します。

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```


例:

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

構成ユーティリティを使用して各仮想サーバのサービスを構成するには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [サービスの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - サービス名：名前
   - サーバ：サーバ名
   - ポートポート
4. 「プロトコル」で、HTTP を指定します。[使用可能なモニター] で、[HTTP] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [Services] ペインに表示されます。

サービスごとにモニターを構成する

コマンドラインインターフェイスを使用してモニタをサービスにバインドするには

コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

例:

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用してモニタをサービスにバインドするには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. サービスを開き、モニターを追加します。

サーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成する

コマンドラインインターフェイスを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

例**:**

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

構成ユーティリティを使用してサーバーに送信されるトラフィックのバランスをとるように **HTTP** 仮想サーバーを構成するには

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次に示すパラメータの値を指定します。
   - 名前—name
   - IP アドレス: IP アドレス
     注: 仮想サーバで IPv6 を使用している場合は、「
     IPv6」チェック・ボックスを選択し、IPv6 形式でアドレスを入力します（たとえば、
     1000:000000:0000:0005:0600:700 a: 888b）。
   - ポートポート
4. [プロトコル] で、[HTTP] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ウィンドウに表示されます。

HTTP サービスを **HTTP** 仮想サーバーにバインドする

コマンドラインインターフェイスを使用して **HTTP** サービスをワイルドカード仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```


例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **HTTP** サービスをワイルドカード仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. [サービス] セクションをクリックして、バインドするサービスを選択します。

注: サービスを複数の仮想サーバーにバインドできます。

設定の保存と確認

設定タスクが完了したら、必ず設定を保存してください。また、設定が正しいことを確認する必要があります。

コマンドラインインターフェイスを使用して構成を保存および確認するには

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。

- save ns config
- show vserver

例:

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

```



```
1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して構成を保存および確認するには

1. 詳細ウィンドウで、[保存] をクリックします。
2. [構成の保存] ダイアログボックスで、[はい] をクリックします。
3. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
4. 詳細ウィンドウで、手順 5 で作成した仮想サーバーを選択し、[詳細] ウィンドウに表示される設定が正しいこ

とを確認します。

5. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
6. 詳細ウィンドウで、手順 5 で作成したサービスを選択し、[詳細] ウィンドウに表示される設定が正しいことを確認します。

エンタープライズ環境でのファイアウォールロードバランシング設定の監視

構成がアップして実行されたら、各サービスと仮想サーバーの統計情報を表示して、考えられる問題をチェックする必要があります。

仮想サーバの統計情報の表示

仮想サーバーのパフォーマンスを評価したり、問題のトラブルシューティングを行うために、Citrix ADC アプライアンスで構成された仮想サーバーの詳細を表示できます。すべての仮想サーバの統計情報のサマリーを表示することも、仮想サーバの名前を指定して、その仮想サーバの統計情報だけを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバの状態
- 受信したリクエストの割合
- ヒット率

コマンドラインインターフェイスを使用して仮想サーバーの統計情報を表示するには

Citrix ADC アプライアンスで現在構成されているすべての仮想サーバー、または単一の仮想サーバーの統計情報のサマリーを表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

例:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
                           Hits/s
One                        *    80         HTTP           UP      5/s
          0/s
```



```
5  Two                          *      0          TCP          DOWN       0/s
           0/s
6  Three                        *   2598          TCP          DOWN       0/s
           0/s
7  dnsVirtualNS     10.102.29.90     53          DNS          DOWN       0/s
           0/s
8  BRVSERV             10.10.1.1     80         HTTP          DOWN       0/s
           0/s
9  LBVIP            10.102.29.66     80         HTTP            UP       0/s
           0/s
10  Done
11
12
13 <!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバの統計情報を表示するには

1. Traffic Management > Load Balancing > Virtual Servers > Statistics に移動します。
2. 1 つの仮想サーバーのみの統計を表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計情報の表示

更新：2013-08-28

サービス統計情報を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバ接続などを表示できます。

コマンドラインインターフェイスを使用してサービスの統計情報を表示するには

コマンドプロンプトで入力します。

```
1 stat service <name>
2 <!--NeedCopy-->
```

例:

```
1 stat service Service-HTTP-1
2 <!--NeedCopy-->
```


構成ユーティリティを使用してサービスの統計情報を表示するには

1. Traffic Management > Load Balancing > Virtual Servers > Statistics に移動します。
2. 1 つのサービスのみの統計を表示する場合は、サービスを選択して、[統計] をクリックします。

複数のファイアウォール環境

October 7, 2021

複数ファイアウォール環境では、Citrix ADC アプライアンスは、パブリックインターネットに接続する外部セット、および内部プライベートネットワークに接続する内部セットの 2 つのファイアウォールセットの間に配置されます。通常、外部セットは出力トラフィックを処理します。これらのファイアウォールは、主に外部リソースへのアクセスを許可または拒否するアクセス制御リストを実装します。通常、内部セットは入力トラフィックを処理します。これらのファイアウォールは、入力トラフィックの負荷分散とは別に、悪意のある攻撃からイントラネットを保護するセキュリティを実装します。マルチファイアウォール環境では、別のファイアウォールからのトラフィックをロードバランシングできます。デフォルトでは、ファイアウォールからのトラフィックは、Citrix ADC アプライアンスの他のファイアウォールでは負荷分散されません。Citrix ADC 両側でファイアウォールの負荷分散を有効にすると、出力方向と入力方向のトラフィックフローが改善され、トラフィックの処理速度が向上します。

次の図は、複数ファイアウォールの負荷分散環境を示しています。

図 1: ファイアウォールの負荷分散（複数ファイアウォール）

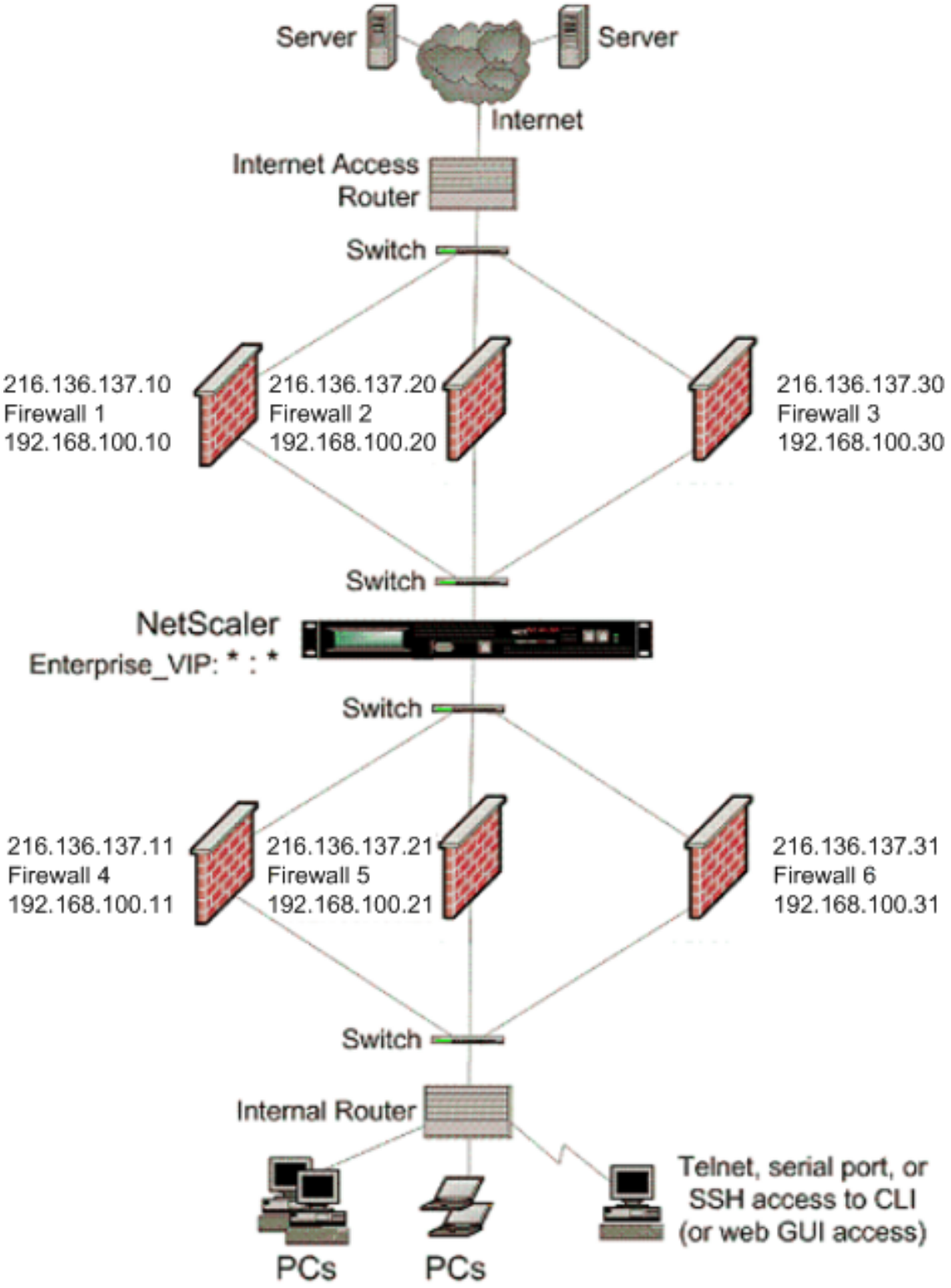


図 1 のような構成では、外部ファイアウォールによって負荷分散されている場合でも、内部ファイアウォールを通過するトラフィックの負荷分散を行うように Citrix ADC を構成できます。たとえば、この機能を設定すると、外部ファ

イアウォール（ファイアウォール 1、2、3）からのトラフィックは内部ファイアウォール（ファイアウォール 4、5、6）でロードバランシングされ、その逆も同様です。

ファイアウォールロードバランシングは、MAC モード LB 仮想サーバに対してのみサポートされます。

サービスタイプ ANY は、すべてのトラフィックを受け入れるように Citrix ADC を構成します。

HTTP および TCP に関連する利点を活用するには、サービスおよび仮想サーバを HTTP または TCP タイプで構成します。FTP が機能するには、FTP タイプを使用してサービスを構成します。

複数のファイアウォール環境での **Citrix ADC** 構成

複数のファイアウォール環境で Citrix ADC アプライアンスを構成するには、負荷分散機能を有効にし、外部ファイアウォールで送信トラフィックの負荷分散を行うように仮想サーバーを構成し、内部ファイアウォールを越えて入力トラフィックの負荷分散を行うように仮想サーバーを構成する必要があります。Citrix ADC アプライアンスでファイアウォールの負荷分散を有効にします。マルチファイアウォール環境で、ファイアウォール全体でトラフィックの負荷分散を行うように仮想サーバを構成するには、次の作業を行う必要があります。

1. ファイアウォールごとにワイルドカードサービスを構成する
2. ワイルドカードサービスごとにモニタを構成する
3. ワイルドカード仮想サーバーを構成して、ファイアウォールに送信されるトラフィックの負荷を分散する
4. MAC 書き換えモードで仮想サーバを設定します
5. ファイアウォールサービスをワイルドカード仮想サーバーにバインドする

ロードバランシング機能の有効化

サービスや仮想サーバーなどの負荷分散エンティティを構成して実装するには、Citrix ADC デバイスで負荷分散機能を有効にする必要があります。

CLI を使用してロードバランシングを有効にするには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、負荷分散を有効にし、構成を確認します。

```
enable ns feature <featureName>
show ns feature
<!--NeedCopy-->
```

例**:**

```
enable ns feature LoadBalancing
Done
show ns feature
Feature Acronym Status
```



```
------- ------- ------
1) Web Logging WL OFF
2) Surge Protection SP ON
3) Load Balancing LB ON
.
.
.
24) NetScaler Push push OFF
Done
<!--NeedCopy-->
```

GUI を使用してロードバランシングを有効にするには、次の手順を実行します。

1. ナビゲーションペインで、[System] を展開し、[Settings] をクリックします。
2. [設定] ウィンドウの [モードと機能] で、[基本機能の変更] をクリックします。
3. [基本機能の構成] ダイアログボックスで、[負荷分散] チェックボックスをオンにし、[OK] をクリックします。

ファイアウォールごとにワイルドカードサービスを構成する

すべてのプロトコルからのトラフィックを受け入れるには、すべてのプロトコルとポートのサポートを指定して、ファイアウォールごとにワイルドカードサービスを設定する必要があります。

CLI を使用して各ファイアウォールにワイルドカードサービスを設定するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、すべてのプロトコルとポートのサポートを構成します。

```
add service <name>@ <serverName> <serviceType> <port_number>
<!--NeedCopy-->
```

例**:**

```
add service fw-svc1 10.102.29.5 ANY *
<!--NeedCopy-->
```

GUI を使用して各ファイアウォールにワイルドカードサービスを構成するには、次の手順を実行します。

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [サービスの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - サービス名：名前
   - サーバ：サーバ名

-* 必須パラメータ

4. [プロトコル] で [任意] を選択し、[ポート] で [*] を選択します。
5. [Create] をクリックしてから、[Close] をクリックします。作成したサービスが [Services] ペインに表示されます。

各サービスのモニターの構成

PING モニターは、デフォルトでサービスにバインドされます。個々のファイアウォールを介して信頼側のホストを監視するように、トランスペアレントモニタを構成する必要があります。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、Citrix ADC アプライアンスとアップストリームデバイス間の接続のみを監視します。トランスペアレントモニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスへのパスに存在するすべてのデバイスを監視します。トランスペアレントモニタが設定されておらず、ファイアウォールのステータスが UP であるにもかかわらず、そのファイアウォールからのネクストホップデバイスのいずれかがダウンしている場合、アプライアンスはロードバランシングの実行中にファイアウォールを含み、パケットをファイアウォールに転送します。ただし、ネクストホップデバイスのいずれかがダウンしているため、パケットは最終的な宛先に配信されません。トランスペアレントモニタをバインドすることにより、いずれかのデバイス（ファイアウォールを含む）がダウンしている場合、サービスはダウンとしてマークされ、アプライアンスがファイアウォールロードバランシングを実行するときにファイアウォールは含まれません。

トランスペアレントモニタをバインドすると、PING モニタが上書きされます。トランスペアレントモニタに加えて PING モニタを構成するには、トランスペアレントモニタを作成してバインドした後、PING モニタをサービスにバインドする必要があります。

CLI を使用してトランスペアレントモニタを設定するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

例**:**

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

Citrix ADC アプライアンスは、サービスにバインドされているモニターからサーバー L2 パラメーターを学習します。UDP-ECV モニタの場合は、受信文字列を設定して、アプライアンスがサーバの L2 パラメータを学習できるように

します。受信ストリングが設定されておらず、サーバが応答しない場合、アプライアンスは L2 パラメータを学習しませんが、サービスは UP に設定されます。このサービスのトラフィックはブラックホールです。

CLI を使用して受信ストリングを設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

```
1 add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
     transparent (YES | NO )] [-send <string>] [-recv <string>]
2 <!--NeedCopy-->
```

例**:**

```
1 add lb monitor monitor-udp-1 udp-ecv -destip 10.10.10.11 -transparent
     YES  - send "test message"  - recv "site_is_up"
2 <!--NeedCopy-->
```

GUI を使用してトランスペアレントモニタを作成してバインドするには、次の手順を実行します。

1. Traffic Management > Load Balancing > Monitors に移動します。
2. 詳細ペインで、[Add] をクリックします。
3. [モニタの作成] ダイアログボックスで、次に示すパラメータの値を指定します。
   - Name*
   - タイプ *—タイプ
   - 接続先 IP
   - 透明

   -* 必須パラメータ
4. [Create] をクリックしてから、[Close] をクリックします。[Monitors] ペインで、構成したモニタを選択し、画面の下部に表示される設定が正しいことを確認します。

ファイアウォールに送信されるトラフィックの負荷分散のための仮想サーバーの構成

あらゆる種類のトラフィックをロードバランシングするには、プロトコルとポートを任意の値として指定する、ワイルドカード仮想サーバを設定する必要があります。

CLI を使用してファイアウォールに送信されるトラフィックをロードバランシングするように仮想サーバを設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。


```
add lb vserver <name>@ <serviceType> <IPAddress> <port_number>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

GUI を使用してファイアウォールに送信されるトラフィックをロードバランシングするように仮想サーバを設定するには、次の手順を実行します。

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインで、[Add] をクリックします。
3. [プロトコル] で [任意] を選択し、[IP アドレス] と [ポート] で [*] を選択します。
4. [Create] をクリックしてから、[Close] をクリックします。作成した仮想サーバーが [負荷分散仮想サーバー] ウィンドウに表示されます。

仮想サーバの **MAC** 書き換えモードへの設定

着信トラフィックの転送に MAC アドレスを使用するように仮想サーバを設定するには、MAC 書き換えモードを有効にする必要があります。

CLI を使用して **MAC** 書き換えモードで仮想サーバを設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

GUI を使用して **MAC** 書き換えモードで仮想サーバを設定するには、次の手順を実行します。

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ウィンドウで、リダイレクションモードを構成する仮想サーバー (Vserver-LB1 など) を選択し、[開く] をクリックします。
3. [詳細設定] タブの [リダイレクトモード] で、[開く] をクリックします。
4. [OK] をクリックします。

仮想サーバーへのファイアウォールサービスのバインド

Citrix ADC アプライアンス上のサービスにアクセスするには、そのサービスをワイルドカード仮想サーバーにバインドする必要があります。

CLI を使用してファイアウォールサービスを仮想サーバにバインドするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

```
bind lb vserver <name>@ <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してファイアウォールサービスを仮想サーバにバインドするには、次の手順を実行します。

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ウィンドウで、リダイレクションモードを構成する仮想サーバー (Vserver-LB1 など) を選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (負荷分散)] ダイアログボックスの [サービス] タブで、仮想サーバーにバインドするサービスの横にある [アクティブ] チェックボックスをオンにします (Service-HTTP-1 など)。
4. [OK] をクリックします。

Citrix ADC アプライアンスでの複数ファイアウォールの負荷分散の構成

ファイアウォールの負荷分散を使用して Citrix ADC 両側のトラフィックを負荷分散するには、vServerSpecificMac パラメータを使用してマルチファイアウォールの負荷分散を有効にする必要があります。

CLI を使用して複数ファイアウォールのロードバランシングを設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

```
set lb parameter -vServerSpecificMac <status>
<!--NeedCopy-->
```

例:


```
set lb parameter -vServerSpecificMac ENABLED
<!--NeedCopy-->
```

GUI を使用して複数ファイアウォールのロードバランシングを設定するには、次の手順を実行します。

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 詳細ウィンドウで、リダイレクションモードを構成する仮想サーバーを選択します (たとえば、[負荷分散パラメーターの構成])。
3. [負荷分散パラメータの設定] ダイアログボックスで、[仮想サーバー固有の MAC] チェックボックスをオンにします。
4. [OK] をクリックします。

設定の保存と確認

設定タスクが完了したら、必ず設定を保存してください。また、設定が正しいことを確認する必要があります。

CLI を使用して設定を保存および確認するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、トランスペアレントモニタを構成し、構成を確認します。

- save ns config
- show vserver

例**:**

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: \*) - ANY State: UP Weight: 1
```



```
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

GUI を使用して設定を保存および確認するには、次の手順を実行します。

1. 詳細ウィンドウで、[保存] をクリックします。
2. [構成の保存] ダイアログボックスで、[はい] をクリックします。
3. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
4. 詳細ウィンドウで、手順 5 で作成した仮想サーバーを選択し、[詳細] ウィンドウに表示される設定が正しいことを確認します。
5. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
6. 詳細ウィンドウで、手順 5 で作成したサービスを選択し、[詳細] ウィンドウに表示される設定が正しいことを

確認します。

複数ファイアウォール環境でのファイアウォールロードバランシング設定の監視

構成がアップして実行されたら、各サービスと仮想サーバーの統計情報を表示して、考えられる問題をチェックする必要があります。

仮想サーバの統計情報の表示

仮想サーバーのパフォーマンスを評価したり、問題のトラブルシューティングを行うために、Citrix ADC アプライアンスで構成された仮想サーバーの詳細を表示できます。すべての仮想サーバの統計情報のサマリーを表示することも、仮想サーバの名前を指定して、その仮想サーバの統計情報だけを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバの状態
- 受信したリクエストの割合
- ヒット率

コマンドラインインターフェイスを使用して仮想サーバーの統計情報を表示するには

Citrix ADC アプライアンスで現在構成されているすべての仮想サーバー、または単一の仮想サーバーの統計情報のサマリーを表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

例:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
                           Hits/s
One                        *    80         HTTP           UP      5/s
        0/s
Two                        *     0          TCP         DOWN      0/s
        0/s
```



```
6  Three                         *  2598           TCP          DOWN       0/s
          0/s
7  dnsVirtualNS     10.102.29.90    53           DNS          DOWN       0/s
          0/s
8  BRVSERV             10.10.1.1    80          HTTP          DOWN       0/s
          0/s
9  LBVIP            10.102.29.66    80          HTTP            UP       0/s
          0/s
10  Done
11
12
13 <!--NeedCopy-->
```

GUI を使用して仮想サーバの統計情報を表示するには、次の手順を実行します。

1. Traffic Management > Load Balancing > Virtual Servers > Statistics に移動します。
2. 1 つの仮想サーバーのみの統計を表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計情報の表示

サービス統計情報を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバ接続などを表示できます。

CLI を使用してサービスの統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 stat service <name>
2 <!--NeedCopy-->
```

例**:**

```
1 stat service Service-HTTP-1
2 <!--NeedCopy-->
```

GUI を使用してサービスの統計情報を表示するには、次の手順を実行します。

1. Traffic Management > Load Balancing > Virtual Servers > Statistics に移動します。
2. 1 つのサービスのみの統計を表示する場合は、サービスを選択して、[統計] をクリックします。

Global Server Load Balancing

October 7, 2021

> メモ:
>
> - リリース 13.0 ビルド 41.x 以降、Citrix ADC アプライアンスを使用したグローバルサーバー負荷分散（GSLB）の展開は、2019 年の DNS フラグに完全に準拠しています。
> - GSLB 機能は、Citrix ADCAdvance および Premium エディションのライセンスに含まれています。Citrix ADC オプションライセンスは、Standard エディションでサポートされています。

GSLB 用に構成された Citrix ADC アプライアンスは、災害復旧を提供し、WAN の障害点から保護することにより、アプリケーションの継続的な可用性を確保します。GSLB は、クライアント要求を最も近いデータセンターまたは最高のパフォーマンスを発揮するデータセンターに転送することによって、データセンター間で負荷を分散します。また、停止が発生した場合は存続しているデータセンターに分散します。

一般的な構成では、ローカル DNS サーバーは、GSLB サービスにバインドされている GSLB 仮想サーバーにクライアント要求を送信します。GSLB サービスは、ローカルサイトまたはリモートサイトにあることができる負荷分散またはコンテンツスイッチング仮想サーバーを識別します。GSLB 仮想サーバーがリモートサイトで負荷分散またはコンテンツスイッチング仮想サーバーを選択すると、仮想サーバーの IP アドレスを DNS サーバーに送信します。DNS サーバーは、クライアントに送信します。次に、クライアントは新しい IP で新しい仮想サーバーに要求を再送信します。

設定する必要がある GSLB エンティティは、GSLB サイト、GSLB サービス、GSLB 仮想サーバー、負荷分散またはコンテンツスイッチング仮想サーバー、および権威ある DNS (ADNS) サービスです。MEP も設定する必要があります。また、DNS ビューを構成して、ネットワークのさまざまな部分を異なる場所からネットワークにアクセスするクライアントに公開することもできます。

> 注:
>
> GSLB 機能を最大限に活用するには、ADC アプライアンスを各データセンターで負荷分散またはコンテンツスイッチングに使用して、GSLB 構成が独自の MEP を使用してサイトメトリックスを交換できるようにします。

GSLB のしくみ

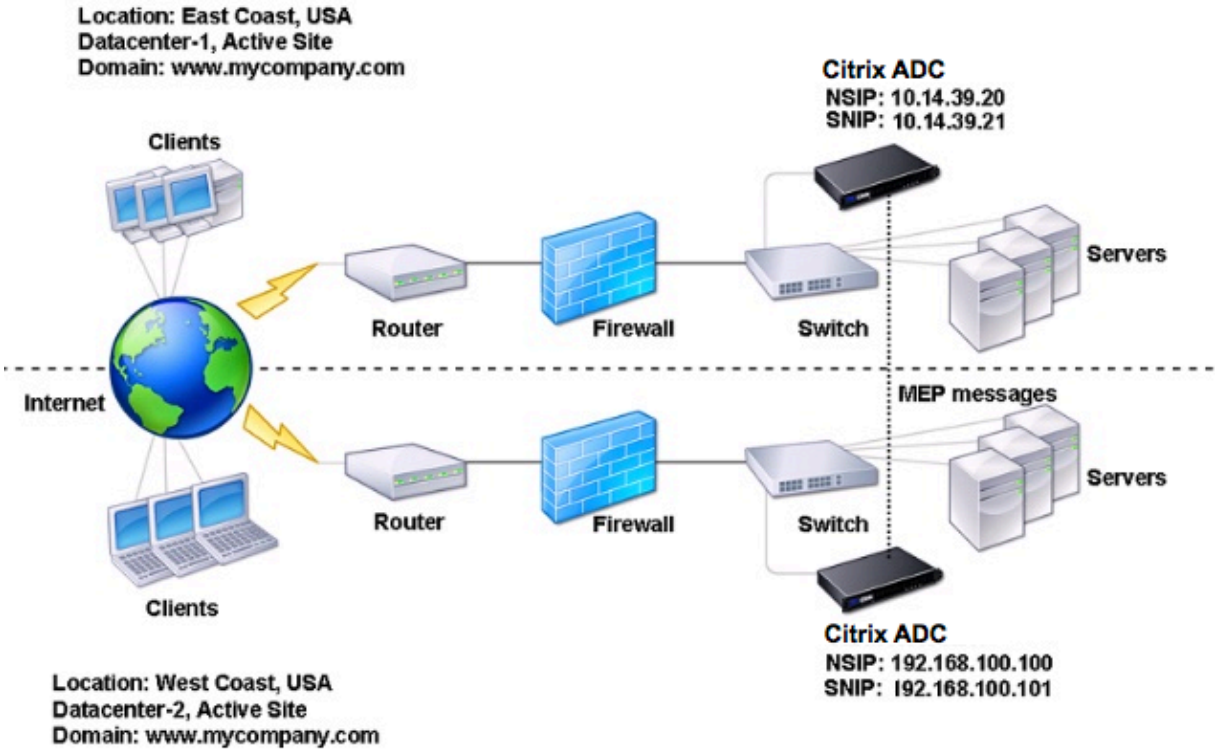
通常の DNS では、クライアントがドメインネームシステム (DNS) 要求を送信すると、ドメインまたはサービスの IP アドレスの一覧を受信します。通常、クライアントはリストの最初の IP アドレスを選択し、そのサーバーとの接続を開始します。DNS サーバーは、DNS ラウンドロビンと呼ばれる技術を使用して、リスト上の IP を循環させます。最初の IP アドレスをリストの最後に送信し、各 DNS 要求に応答した後に他のアドレスを昇格させます。この手法では、負荷の均等な分散が保証されますが、障害復旧、サーバーの負荷または近接性に基づく負荷分散、または永続性はサポートされません。

ADC アプライアンスで GSLB を設定し MEP を有効にすると、DNS インフラストラクチャを使用して、設定された基準に最も適合するデータセンターにクライアントを接続します。基準は以下を指定できます。

- 最も負荷の少ないデータセンター
- 最寄りのデータセンター
- クライアントの場所からの要求に最も迅速に応答するデータセンター
- これらのメトリックと SNMP メトリックの組み合わせ。

アプライアンスは、各データセンターの場所、パフォーマンス、負荷、可用性を追跡します。これらの要因を使用して、クライアント要求を送信するデータセンターを選択します。

次の図は、基本的な GSLB トポロジを示しています。



GSLB 構成は、構成内の各アプライアンス上の GSLB エンティティのグループで構成されます。これらのエンティティには、GSLB サイト、GSLB サービス、GSLB サービスグループ、GSLB 仮想サーバー、負荷分散サーバー、コンテンツスイッチングサーバー、および ADNS サービスが含まれます。

GSLB デプロイメントのタイプ

October 7, 2021

グローバルサーバー負荷分散（GSLB）用に構成された Citrix ADC アプライアンスは、ディザスタリカバリを提供し、WAN（ワイド・エリア・ネットワーク）における障害点からの保護によってアプリケーションの継続的な可用性を確保します。GSLB は、クライアント要求を最も近いデータセンターまたは最高のパフォーマンスを発揮するデー

タセンターに転送することによって、データセンター間で負荷を分散できます。また、停止が発生した場合は存続するデータセンターに分散させることができます。

以下に、一般的な GSLB デプロイメントの種類の一部を示します。

- アクティブ-アクティブサイトの展開
- アクティブ-パッシブサイトの展開
- 親子トポロジの配置

アクティブ-アクティブサイトの展開

October 7, 2021

アクティブ/アクティブ・サイトは、複数のアクティブなデータ・センターで構成されます。クライアント要求は、アクティブなデータセンター間で負荷分散されます。この展開の種類は、分散環境でトラフィックをグローバルに分散する必要がある場合に使用できます。

アクティブ/アクティブ展開内のすべてのサイトがアクティブであり、特定のアプリケーション/ドメインのすべてのサービスが同じ GSLB 仮想サーバーにバインドされます。サイトは、メトリック交換プロトコル（MEP）を介してメトリックを交換します。サイト間で交換されるサイトメトリックには、各ロードバランシングおよびコンテンツスイッチング仮想サーバーのステータス、現在の接続数、現在のパケットレート、および現在の帯域幅の使用状況が含まれます。Citrix ADC アプライアンスは、サイト間で負荷分散を実行するためにこの情報を必要とします。

MEP では 32 を超えるサイトを同期できないため、アクティブ/アクティブ配置には最大 32 の GSLB サイトを含めることができます。この展開の種類では、バックアップサイトは構成されていません。

Citrix ADC アプライアンスは、GSLB 構成で指定された GSLB 方式によって決定された適切な GSLB サイトにクライアント要求を送信します。

アクティブ-アクティブ展開の場合、次の GSLB メソッドを設定できます。

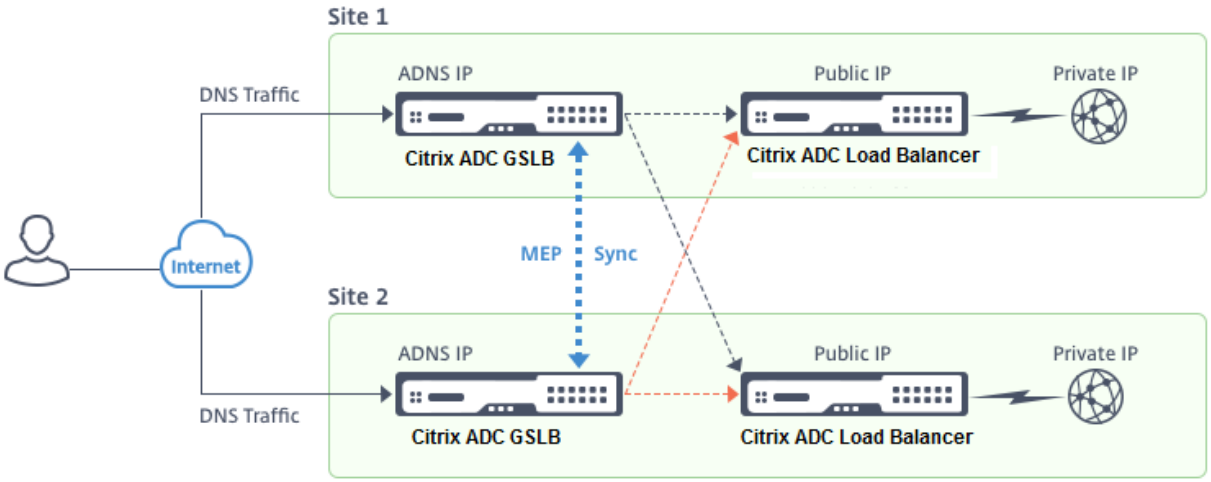
- ラウンドロビン
- 最小接続数
- 最短応答時間
- 最小帯域幅
- 最小パケット
- 送信元 IP ハッシュ
- カスタムロード
- ラウンドトリップ時間（RTT）
- 静的な近接

> 注:
>
> - MEP が無効の場合、次の GSLB メソッドはデフォルトでラウンドロビン方式に設定されます。

  - RTT
  - 最小接続
  - 最小帯域幅
  - 最小パケット
  - 最小応答時間
- 静的近接 GLSB 方式では、アプライアンスは近接基準に最もよく一致するサイトの IP アドレスに要求を送信します。
- ラウンドトリップ時間方式では、動的ラウンドトリップ時間（RTT）値は、最もパフォーマンスの高いサイトの IP アドレスを選択することです。RTT は、クライアントのローカル DNS サーバーとデータリソース間のネットワークの遅延の尺度です。

GSLB アクティブ/アクティブデータセントートポロジ

この図では、サイト 1 とサイト 2 はアクティブな GLSB サイトです。



クライアントは DNS 要求を送信すると、アクティブサイトの 1 つに着陸します。

サイト 1 がクライアント要求を受信すると、サイト 1 の GSLB 仮想サーバーは、負荷分散またはコンテンツスイッチング仮想サーバーを選択し、仮想サーバーの IP アドレスをクライアントに送信する DNS サーバーに送信します。次に、クライアントは新しい仮想サーバに新しい IP アドレスで要求を再送信します。

両方のサイトがアクティブであるため、GSLB アルゴリズムは、設定された GSLB 方式によって決定された選択を行うときに、両方のサイトのサービスを評価します。

アクティブ-パッシブサイトの展開

October 7, 2021

アクティブ/パッシブサイトは、アクティブデータセンターとパッシブデータセンターで構成されます。この展開タイプは、ディザスタリカバリに最適です。

このタイプの展開では、一部のサイト（リモート・サイト）は災害復旧専用に予約されています。これらのサイトは、すべてのアクティブなサイトが DOWN になるまで、いかなる意思決定にも参加しません。災害イベントがフェイルオーバーをトリガーしない限り、パッシブ・サイトは稼働状態になりません。

プライマリデータセンターを構成したら、バックアップデータセンターの設定を複製し、そのサイトの GSLB 仮想サーバーをバックアップ仮想サーバーとして指定することで、パッシブ GSLB サイトとして指定します。

MEP では 32 を超えるサイトを同期できないため、アクティブ/パッシブ展開には最大 32 の GSLB サイトを含めることができます。

アクティブ-パッシブ展開では、次の GSLB メソッドを構成できます。

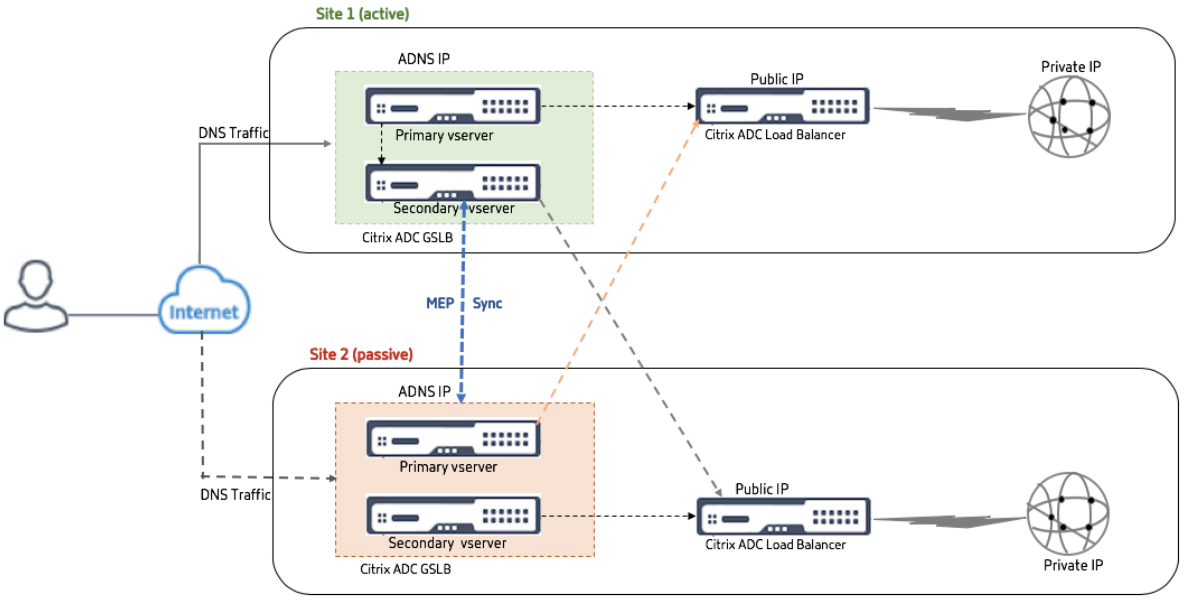
- ラウンドロビン
- 最小接続数
- 最短応答時間
- 最小帯域幅
- 最小パケット
- 送信元 IP ハッシュ
- カスタムロード
- ラウンドトリップ時間（RTT）
- 静的な近接

注:

- MEP が無効の場合、次のアルゴリズム方式はデフォルトでラウンドロビン（Round Robin）に設定されます。
  - RTT
  - 最小接続数
  - 最小帯域幅
  - 最小パケット
  - 最短応答時間
- 静的近接 GLSB 方式では、アプライアンスは近接基準に最もよく一致するサイトの IP アドレスに要求を送信します。
- ラウンドトリップ時間方式では、動的ラウンドトリップ時間（RTT）値は、最もパフォーマンスの高いサイトの IP アドレスを選択することです。RTT は、クライアントのローカル DNS サーバーとデータリソース間のネットワークの遅延の尺度です。

GSLB アクティブ**-**パッシブデータセントートポロジ

この図では、サイト 1 はアクティブサイト、サイト 2 はパッシブサイトであり、サイト 1 と同じ構成になっています。



サイト 1 が DOWN になると、サイト 2 が動作可能になります。

クライアントが DNS 要求を送信すると、要求はどのサイトにも着陸できます。ただし、サービスが UP の限り、アクティブなサイト（Site1）からのみ選択されます。

パッシブサイト（サイト 2）からのサービスは、アクティブサイト（サイト 1）が DOWN の場合にのみ選択されます。

MEP プロトコルを使用した親子トポロジ展開

March 8, 2022

Citrix ADC GSLB は、関係するすべてのサイト間にメッシュ接続を作成し、インテリジェントな負荷分散の決定を行うことにより、グローバルなサーバー負荷分散と障害回復を提供します。各サイトは他のサイトと通信し、メトリック交換プロトコル (MEP) を通じて定期的にサーバとネットワークのメトリックを交換します。ただし、ピアサイトの数が増えると、メッシュトポロジが原因で MEP トラフィックの量は指数関数的に増加します。これを解決するには、親子トポロジを使用します。親子トポロジでは、大規模な展開もサポートされます。32 の親サイトに加えて、1024 の子サイトを構成できます。

GSLB 親子トポロジは、次の特徴を持つ 2 レベルの階層設計です。

- 最上位レベルには親サイトがあり、親サイトには他の親とピア関係があります。
- 各親は複数の子サイトを持つことができます。
- 各親サイトは、子サイトや他の親サイトとヘルス情報を交換します。
- 子サイトは親サイトとのみ通信します。
- GSLB の親子関係では、親サイトだけが ADNS クエリに応答します。子サイトは通常の負荷分散サイトとして機能します。

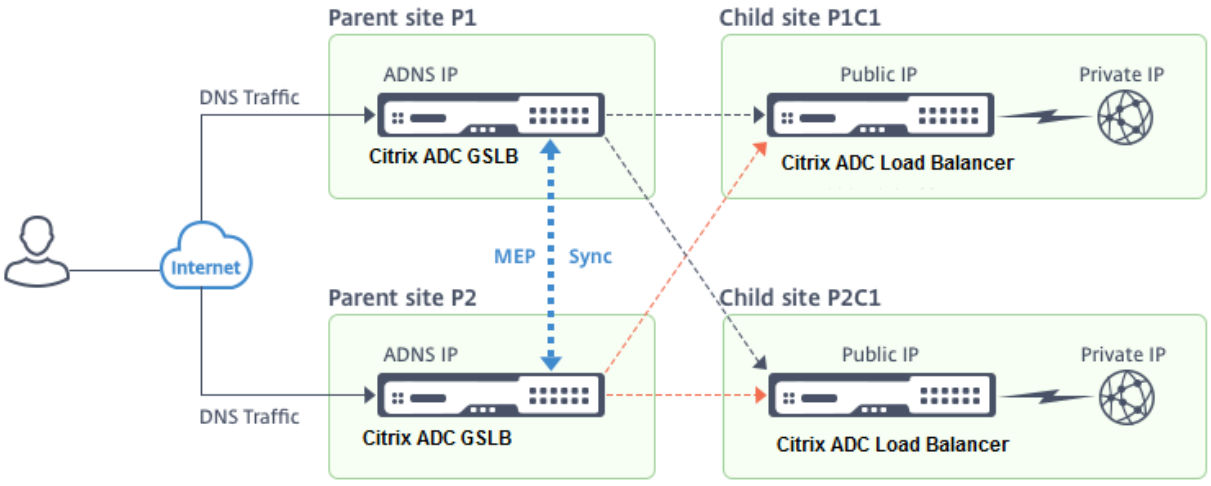
- ADNS サービスまたは DNS 負荷分散仮想サーバーは、親サイトでのみ構成します。
- 親サイトは通常の GSLB 構成、つまりローカルサイトとすべてのリモートサイトからのサービスを持つことができますが、子サイトはローカルサービスのみを持つことができます。また、GSLB 仮想サーバーが構成されているのは親サイトのみです。

> 注
>
> - 親子トポロジでは、2 つの IP アドレスのうち低い方からサイトメトリックの交換が開始されます。ただし、Citrix ADC リリース 11.1 ビルド 51.x 以降、親サイトは子サイトへの接続を開始し、その逆ではありません。親サイトには、GSLB セットアップのすべての子サイトに関する情報があるためです。

- 親-親接続では、2 つの IP アドレスの下位 IP からサイトメトリックの交換が開始されます。
- 親子トポロジでは、子サイトで GSLB サービスを構成する必要は必ずしもありません。ただし、クライアント認証、クライアント IP アドレスの挿入、その他の SSL 固有の要件など、さらに多くの構成がある場合は、子サイトに明示的な GSLB サービスを追加し、それに応じて構成する必要があります。
- 親子トポロジでは、親サイトと子サイトが異なる Citrix ADC ソフトウェアバージョン上に存在する可能性があります。ただし、GSLB AutomaticConfigSync オプションを使用して親サイト間で設定を同期するには、すべての親サイトが同じ Citrix ADC ソフトウェアバージョン上にある必要があります。AutomaticConfigSync オプションを使用していない場合、親サイトと子サイトは異なる Citrix ADC ソフトウェアバージョン上に存在する可能性がありますが、最新リリースの新機能を使用していないことを確認してください。これは一般に、GSLB に参加している 2 つの Citrix ADC ノードにも当てはまります。

基本的な親子トポロジ

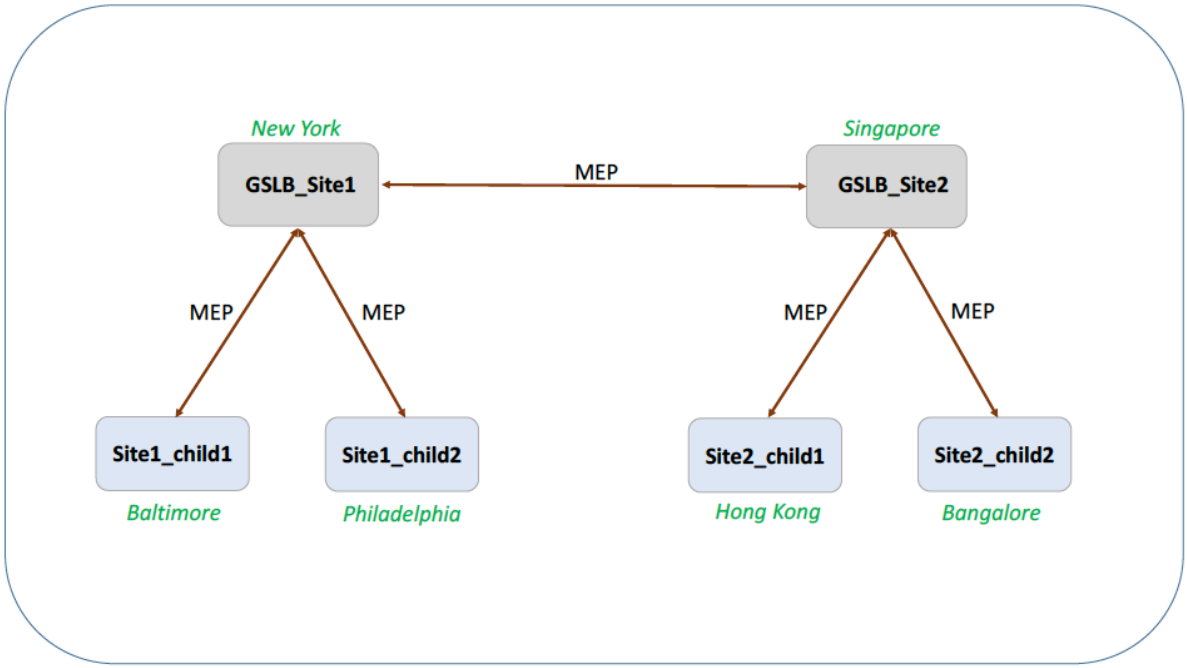
この図では、SiteP1 と SiteP2 はピア関係の親サイトです。サイト P1C1 と P2C1 はそれぞれ P1 と P2 の子サイトです。



GSLB の親子構成を設定する

GSLB サイトでファイアウォールが設定されている場合は、ポート 3011 が開いていることを確認します。

次の図に、親子構成の例を示します。



- 子サイトの構成には子サイトとその親サイトが含まれますが、他の親サイトや子サイトは含まれません。
- RTT やパーシステンスセッション情報などのネットワークメトリックは、親サイト間でのみ同期されます。したがって、NWMetricExchange や SessionExchange などのパラメーターは、すべての子サイトで既定で無効になっています。
- 正しい親子構成を確認するには、親サイトにバインドされているすべての GSLB サービスの状態を確認します。

CLI を使用して **GSLB** の親子構成を設定するには、次の手順を実行します。

1. 各親サイトで、すべての子サイト、ピアの親サイト、およびピアサイトに関連付けられた子サイトを構成します。

   親サイトを追加する際には、以下のコマンドを使用します。

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>]
<!--NeedCopy-->
```

   子サイトを追加する際には、以下のコマンドを使用します。

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```


2. 子サイトで、子サイトを構成し、子サイトを親サイトに関連付けます。

   > 注:
   >
   > 親サイトと子サイトの関連付けを正しく構成してください。たとえば、Site1_child1_child1 を GSLB_Site1 で設定する必要があります。Site1_child1 を GSLB_Site2 で設定することはできません。

   子サイトが関連付けられている親サイトを構成するには、以下のコマンドを使用します。

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>]
<!--NeedCopy-->
```

   子サイトを追加して親サイトに関連付けるには、以下のコマンドを使用します。

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用した親子設定の完全な例については、CLI を使用した完全な親子設定の例を参照してください。

> 注
>
> 負荷分散仮想サーバーの IP アドレスがプライベート IP アドレスで、パブリック IP アドレスがこの IP アドレスと異なる場合は、子サイトのローカル負荷分散仮想サーバーに GSLB サービスを構成する必要があります。これは、親サイトと子サイト間の統計収集に必要です。
>
> 子サイトのコマンドプロンプトで、次のように入力します。
>
> `add gslb service <name> <private IP/lb vserver IP> http 80 -sitename <childsite name> -publicip <public IP of LB vserver>`
>
> 例:
>
> `add gslb service Service-GSLB 192.168.1.3 http 80 -GSLB_Site11 site 11_lb1 172.16.1.1`
>
> 192.168.1.3 は負荷分散仮想サーバーのプライベート IP アドレス、172.16.1.1 は負荷分散仮想サーバーのパブリック IP アドレスです。

親サイトをバックアップする

注：この機能は、Citrix ADC リリース 11.1 ビルド 51.x で導入されました。バックアップ親サイトトポロジを使用するには、親サイトと子サイトが Citrix ADC 11.1 build 51.x 以降にあることを確認します。

親サイトのバックアップトポロジは、1 つの親サイトに多数の子サイトが関連付けられている場合に便利です。この親サイトが DOWN になると、その子サイトはすべて使用できなくなります。これを防ぐために、元の親サイトが DOWN の場合に子サイトが接続できるバックアップ親サイトを構成できるようになりました。親サイトは MEP メッセージを通じてバックアップ親リストを子サイトに送信します。

親サイトが DOWN になると、GSLB 内の他の親サイトは MEP を通じて特定の親サイトが DOWN であることを認識します。これは、その親サイトに対する MEP が DOWN であるためです。GSLB セットアップ内の他の親サイトは、ピアの親のバックアップチェーンを検索します。優先度が最も高い親サイトは、DOWN になった親の子サイトを採用します。その後、新しい親が子サイトとの接続を開始します。子サイトは、既存の接続とバックアップリストの情報を評価した後に、接続を許可または拒否できます。バックアップの親が子サイトを採用するまでに数秒かかります。元の親サイトがバックアップされると、別の親サイトに移行した子サイトとの接続を確立しようとします。接続が成功すると、子サイトは元の親サイトに再割り当てされます。

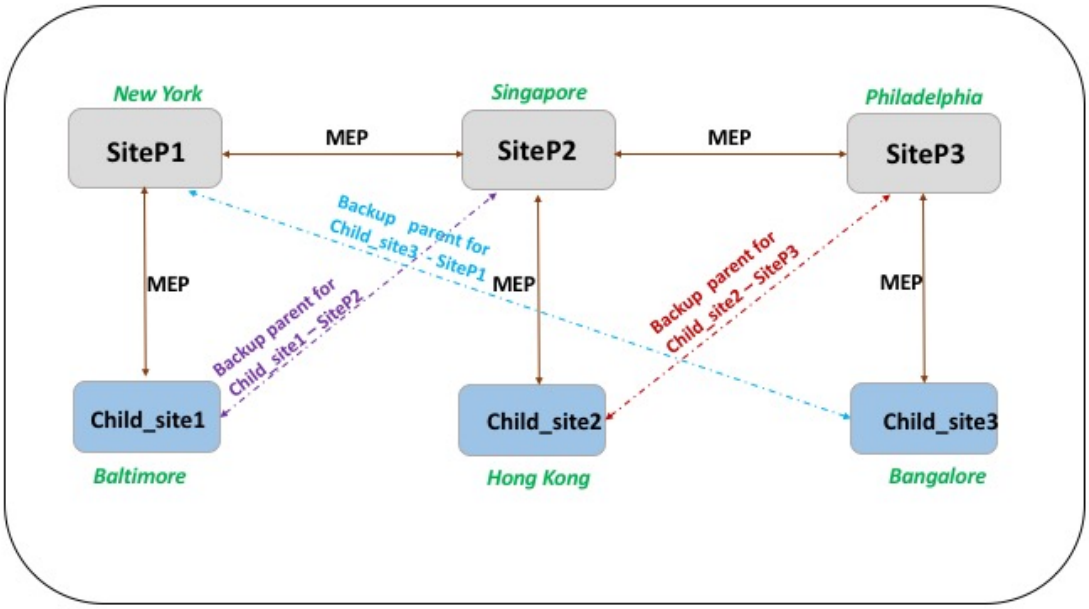
注:

- バックアップとして構成できるのは親サイトのみで、この構成は親サイトでのみ実行できます。
- すべての子サイトがバックアップの親セットを使用します。
- 同期は親サイトでのみ行われます。GSLB 子サイトの構成は同期の影響を受けません。これは、親サイトと子サイトの構成が同一ではないためです。子サイトの構成は、そのサイトとその親サイトの詳細のみで構成されます。また、GSLB サービスは必ずしも子サイトで設定する必要はありません。

次の図に示す構成について考えてみます。

- SiteP1、siteP2、および siteP3 は親サイトです。
- child_site1、child_site2、および child_site3 は、それぞれ siteP1、siteP2、および siteP3 の子サイトです。
- 親サイトをバックアップする。
  - siteP1 バックアップの親-siteP2 (より高い優先度) と siteP3
  - siteP2 バックアップの親 — siteP3 (優先度が高い) と siteP1
  - siteP3 バックアップの親 — siteP1 (優先度が高い) と siteP2

注: 説明のため、この図は親サイトごとに 1 つのバックアップ親のみを示しています。



次の一覧は、さまざまなシナリオにおける親サイトと子サイトの動作をまとめたものです。

- シナリオ 1: SiteP1 がダウンする。
  - siteP2 と siteP3 は、siteP1 の MEP 接続がダウンしていることを検出します。siteP2 は siteP1 のバックアップ親のプリファレンスリストの上位にあるため、child_site1 への接続を開始しようとします。siteP3 は、child_site1 が親 siteP2 の子サイトになったと想定しています。
  - siteP2 は、Child_Site1 に SiteP1 のバックアップ親 (siteP2 および siteP3) のリストを child_Site1 に送信します。Child_site1 はこのリストを使用して、SiteP2 からの接続を受け入れるか拒否するかを決定します。接続を受け入れ、siteP2 の子になります。
  - siteP1 がバックアップされると、child_site1 に接続要求が送信されます。新しいリクエストが優先され、child_site 1 は siteP1 に移行されます。
- シナリオ 2: SiteP1 と SiteP2 の間の MEP 接続だけがダウンしました。Child_site1 は siteP2 の接続要求を拒否します。これは、その親 siteP1 がまだ UP であるためです。
- シナリオ 3: siteP3 と Child_Site1 が siteP1 がダウンしていることを検出し、siteP3 と SiteP2 の間の MEP 接続もダウンしています。ただし、siteP2 は siteP1 が UP であることを検出し、siteP1 と SiteP2 の間の MEP 接続は UP です。
  - SiteP2 は何のアクションも実行しません。
  - SiteP3 は SiteP1 のバックアップリストをチェックし、SiteP2 が SiteP3 よりも優先度が高いことを検出します。しかし、siteP2 はダウンしているため、siteP3 は child_site1 との接続を確立しようとします。child_site1 は siteP1 がダウンしていることを検出したため、SiteP3 の接続要求を受け入れます。
  - これで、siteP1 と siteP2 の間の接続がダウンします。SiteP2 は SiteP1 のバックアップリストをチェックし、自身を最も優先されるバックアップとして検出し、child_site1 への接続を試みます。

child_site1 は siteP1 のリストに基づいて新しい接続要求を評価し、SiteP2 を最も優先されるバックアップとして検出し、siteP3 から siteP2 に移行します。

コマンドラインインターフェイスを使用してバックアップ親サイトを構成するには

コマンドプロンプトで次のように入力します。

```
set gslb site <sitename> -backupParentlist <bkp_site1> <bkp_site2> … <
   bkp_site5>
<!--NeedCopy-->
```

<sitename> は現在の親サイトです。

例**:**

親サイト (SiteP1) では、サイト (SiteP2 と SiteP3) がバックアップ親サイトとして構成されます。

```
set gslb site SiteP1 -backupParentlist SiteP2 SiteP3
<!--NeedCopy-->
```

注:

- 新しいサイトをバックアップの親として追加することはできません。最初にすべてのサイトを追加してから、そのサイトをバックアップの親として構成する必要があります。
- バックアップの親を削除するには、unset コマンドを使用する必要があります。このコマンドは、以前にバックアップ親サイトとして構成されていたすべてのサイトの設定を解除します。

GUI を使用してバックアップ親サイトを構成するには

1. 構成 > トラフィック管理 > **GSLB** > サイトに移動します。
2. 新しいサイトを追加するか、既存のサイトを選択します。
3. GSLB サイトを作成または構成するときに、**[**親サイトのバックアップ] オプションボックスを選択します。

GSLB 設定エンティティ

October 7, 2021

GSLB 構成は、構成内の各アプライアンス上の GSLB エンティティのグループで構成されます。これらのエンティティには、次のものがあります。

- GSLB サイト
- GSLB サービス
- GSLB 仮想サーバ
- 負荷分散またはコンテンツスイッチング仮想サーバー
- ADNS サービス
- DNS リバインダー

GSLB サイト

一般的な GSLB セットアップはデータセンターで構成され、各データセンターには Citrix ADC アプライアンスである場合とそうでない場合があります。データセンターは GSLB サイトと呼ばれています。各 GSLB サイトは、そのサイトのローカルである Citrix ADC アプライアンスによって管理されます。これらのアプライアンスは、それぞれ独自のサイトをローカル・サイトとして扱い、他のアプライアンスによって管理される他のすべてのサイトをリモート・サイトとして扱います。

サイトを管理するアプライアンスが、そのデータセンター内の唯一の Citrix ADC アプライアンスである場合、そのアプライアンスでホストされている GSLB サイトは、メトリックを収集できないため、監査の目的で簿記プレースホルダとして機能します。通常、これは、アプライアンスが GSLB に対してのみ使用され、データセンター内の他の製品が負荷分散やコンテンツスイッチングに使用される場合に発生します。

GSLB サイト間の関係

サイトの概念は、Citrix ADC GSLB の実装の中心です。特に指定がない限り、サイトは相互の相互関係を形成します。この関係は、最初に正常性情報を交換し、次に選択したアルゴリズムによって決定される負荷を分散するために使用されます。しかし、多くの状況では、すべての GSLB サイト間のピア関係は望ましくありません。オールピアの実装を持たない理由は、

- GSLB サイトを明確に分離する。たとえば、DNS クエリの解決に関与するサイトをトラフィック管理サイトから分離します。
- Metric Exchange Protocol（MEP; メトリック交換プロトコル）トラフィックの量を減らすため、ピアサイトの数が増えるにつれて指数関数的に増加します。

これらの目標は、親と子の GSLB サイトを使用することによって達成できます。

GSLB サービス

GSLB サービスは、通常、負荷分散またはコンテンツスイッチング仮想サーバーの表現ですが、任意のタイプの仮想サーバーを表すことができます。GSLB サービスは、仮想サーバーの IP アドレス、ポート番号、およびサービスタイプを識別します。GSLB サービスは、GSLB サイトを管理する Citrix ADC アプライアンス上の GSLB 仮想サーバーにバインドされます。同じデータセンター内の GSLB 仮想サーバーにバインドされた GSLB サービスは、GSLB 仮想サーバーに対してローカルです。別のデータセンター内の GSLB 仮想サーバーにバインドされた GSLB サービスは、その GSLB 仮想サーバーからリモートです。

> 注
>
> サイトとサービスは本質的にリンクされ、両者の間の近接を示します。つまり、すべてのサービスはサイトに属していなければならず、近接目的で GSLB サイトと同じ場所にあると仮定されます。同様に、サービスと仮想サーバーはリンクされているため、ロジックは利用可能なリソースにリンクされます。

GSLB 仮想サーバ

GSLB 仮想サーバーには、1 つ以上の GSLB サービスがバインドされており、それらのサービス間でトラフィックの負荷分散を行います。設定された GSLB メソッド（アルゴリズム）を評価して、クライアント要求を送信する適切なサービスを選択します。GSLB サービスはローカルサーバーまたはリモートサーバーを表すことができるため、要求に対して最適な GSLB サービスを選択すると、クライアント要求を処理するデータセンターを選択する効果があります。

グローバルサーバー負荷分散が設定されているドメインは、GSLB 仮想サーバーにバインドする必要があります。これは、仮想サーバーにバインドされた 1 つ以上のサービスがそのドメインに対して行われた要求を処理するためです。

Citrix ADC アプライアンス上で構成されている他の仮想サーバーとは異なり、GSLB 仮想サーバーには独自の仮想 IP アドレス（VIP）がありません。

負荷分散またはコンテンツスイッチング仮想サーバー

負荷分散またはコンテンツスイッチング仮想サーバーは、ローカルネットワーク上の 1 つまたは複数の物理サーバーを表します。クライアントは、負荷分散またはコンテンツスイッチング仮想サーバーの仮想 IP (VIP) アドレスに要求を送信し、仮想サーバーは物理サーバー間で負荷を分散します。GSLB 仮想サーバーがローカルまたはリモートの負荷分散またはコンテンツスイッチング仮想サーバーを表す GSLB サービスを選択した後、クライアントはその仮想サーバーの VIP アドレスに要求を送信します。

負荷分散またはコンテンツスイッチング仮想サーバーおよびサービスの詳細については、「[負荷分散またはコンテンツ切り替え](/ja-jp/citrix-adc/13/content-switching.html)」を参照してください。

ADNS サービス

ADNS サービスは、Citrix ADC アプライアンスが権限を持つドメインに対する DNS 要求にのみ応答する特別な種類のサービスです。ADNS サービスが構成されると、アプライアンスはその IP アドレスを所有し、アドバタイズします。ADNS サービスによって DNS 要求を受信すると、アプライアンスはそのドメインにバインドされている GSLB 仮想サーバーをチェックします。GSLB 仮想サーバーがドメインにバインドされている場合、DNS 応答の送信先として最適な IP アドレスが照会されます。

DNS リバインダー

DNS 仮想 IP は、Citrix ADC アプライアンス上の負荷分散 DNS 仮想サーバーを表す仮想 IP（VIP）アドレスです。Citrix ADC アプライアンスが権限を持つドメインの DNS 要求は、DNS VIP に送信できます。

GSLB メソッド

October 7, 2021

構成されたサーバーの IP アドレスを単純に応答する従来の DNS サーバーとは異なり、GSLB 用に構成された Citrix ADC アプライアンスは、構成された GSLB 方式によって決定されたサービスの IP アドレスで応答します。デフォルトでは、GSLB 仮想サーバーは最小接続方法に設定されています。すべての GSLB サービスがダウンしている場合、アプライアンスは設定されているすべての GSLB サービスの IP アドレスで応答します。

GSLB メソッドは、GSLB 仮想サーバーが最高のパフォーマンス GSLB サービスを選択するために使用するアルゴリズムです。Web アドレスのホスト名が解決されると、クライアントは、解決されたサービス IP アドレスにトラフィックを直接送信します。

Citrix ADC アプライアンスは、次の GSLB 方式を提供します。

- ラウンドロビン
- 最小接続数
- 最短応答時間
- 最小帯域幅
- 最小パケット
- 送信元 IP ハッシュ
- カスタムロード
- ラウンドトリップ時間（RTT）
- 静的な近接

GSLB メソッドをリモートサイトで使用するには、MEP を有効にするか、明示的なモニタをリモートサービスにバインドする必要があります。MEP が無効の場合、RTT、最小接続、最小帯域幅、最小パケット数、最小応答時間方式はデフォルトでラウンドロビンに設定されます。

静的近接および RTT ロードバランシング方式は、GSLB に固有です。

静的近接またはダイナミック **RTT** 以外の **GSLB** 方式の指定

ラウンドロビン、最小接続、最小応答時間、最小帯域幅、最小パケット、送信元 IP ハッシュ、またはカスタムロード方法の詳細については、ロードバランシングを参照してください。

CLI を使用して **GSLB** メソッドを変更するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -lbMethod GSLBMethod
<!--NeedCopy-->
```


例:

```
set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN
<!--NeedCopy-->
```

GUI を使用して **GSLB** メソッドを変更するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. 詳細ペインで、GSLB 仮想サーバーを選択し、[開く] をクリックします。
3. [GSLB 仮想サーバーの構成] ダイアログボックスの [メソッドと永続性] タブの [メソッド] で、[メソッドの選択] リストからメソッドを選択します。
4. [**OK**] をクリックし、選択した方法が、画面下部の [詳細] に表示されていることを確認します。

GSLB アルゴリズム

October 7, 2021

GSLB では、次のアルゴリズムがサポートされています。

- ラウンドロビン：GSLB 仮想サーバーがラウンドロビン方式を使用するように構成されている場合、それは継続的にそれにバインドされているサービスのリストをローテーションします。仮想サーバは、要求を受信すると、リスト内の最初のサービスに接続を割り当てて、そのサービスをリストの一番下に移動します。
- 最小応答時間：GSLB 仮想サーバーが最小応答時間方式を使用するように構成されている場合、最小値を持つサービスを選択します。ここで、最小値は、現在のアクティブな接続 X 平均応答時間。

  この方法は、HTTP サービスおよびセキュアソケットレイヤー (SSL) サービスに対してのみ構成できます。応答時間（TTFB とも呼ばれます）は、要求パケットをサービスに送信してからサービスから最初の応答パケットを受信するまでの時間間隔です。NetScaler アプライアンスは、応答コード 200 を使用して TTFB を計算します。
- 最小接続：GSLB 仮想サーバーが最小接続 GSLB アルゴリズム（または方法）を使用するように構成されている場合、アクティブな接続が最も少ないサービスを選択します。これは、ほとんどの状況で最高のパフォーマンスを提供するため、デフォルトの方法です。
- 最小帯域幅: 最小帯域幅方式を使用するように構成された GSLB 仮想サーバーは、現在最も少ないトラフィック量を処理しているサービスを選択します。メガビット/秒 (Mbps) 単位で測定されます。
- 最小パケット：最小パケット方式を使用するように設定された GSLB 仮想サーバは、過去 14 秒間に最も少ないパケットを受信したサービスを選択します。

- 送信元 **IP** ハッシュ: 送信元 IP ハッシュ方式を使用するように構成された GSLB 仮想サーバーは、クライアントの IPv4 アドレスまたは IPv6 アドレスのハッシュ値を使用してサービスを選択します。特定のネットワークに属する送信元 IP アドレスからのすべての要求を特定の宛先サーバーに転送するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、netMask パラメーターを使用します。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。
- カスタムロード: カスタムロードバランシングは、CPU 使用率、メモリ、応答時間などのサーバーパラメータに対して実行されます。カスタムロード方式を使用する場合、Citrix ADC アプライアンスは通常、アクティブなトランザクションを処理していないサービスを選択します。GSLB セットアップのすべてのサービスがアクティブなトランザクションを処理している場合、アプライアンスは負荷が最も小さいサービスを選択します。ロードモニタと呼ばれる特殊なタイプのモニタは、ネットワーク内の各サービスの負荷を計算します。負荷モニターはサービスの状態をマークしませんが、サービスが UP でない場合は GSLB の決定からサービスを取り出します。

詳細については、「負荷分散」を参照してください。

静的な近接

October 7, 2021

GSLB の静的近接方法は、IP アドレスベースの静的近接データベースを使用して、クライアントのローカル DNS サーバーと GSLB サイト間の近接性を判断します。Citrix ADC アプライアンスは、近接基準に最も一致するサイトの IP アドレスで応答します。

地理的に異なる場所にある 2 つ以上の GSLB サイトが同じコンテンツを提供する場合、Citrix ADC アプライアンスは IP アドレス範囲のデータベースを維持し、着信クライアント要求を送信する GSLB サイトについての決定にデータベースを使用します。

静的近接方式を機能させるには、Citrix ADC アプライアンスを設定して、ロケーションファイルを介して入力された既存の静的近接データベースを使用するか、静的近接データベースにカスタムエントリを追加する必要があります。カスタムエントリを追加した後、ロケーション修飾子を設定できます。データベースを設定したら、GSLB メソッドとして静的な近接を指定する準備が整いました。

静的近接の設定の詳細については、静的近接の設定を参照してください。

動的ラウンドトリップ時間方式

October 7, 2021

動的ラウンドトリップ時間 (RTT) は、クライアントのローカル DNS サーバーとデータリソース間のネットワークにおける時間または遅延の尺度です。動的 RTT を測定するために、Citrix ADC アプライアンスはクライアントのロー

カル DNS サーバーを検査し、RTT メトリック情報を収集します。次に、アプライアンスはこのメトリックを使用して、ロードバランシングの決定を行います。グローバルサーバロードバランシングは、ネットワークのリアルタイムのステータスを監視し、RTT 値が最も小さいデータセンターにクライアント要求を動的に送信します。

ドメインに対するクライアントの DNS 要求が、そのドメインの権限のある DNS として構成された Citrix ADC アプライアンスに届くと、アプライアンスは RTT 値を使用して最適なパフォーマンスのサイトの IP アドレスを選択し、DNS 要求に対する応答として送信します。

Citrix ADC アプライアンスは、ICMP エコー要求または応答（PING）、UDP、TCP などのさまざまなメカニズムを使用して、ローカル DNS サーバーと参加サイト間の接続の RTT メトリックを収集します。アプライアンスは、最初に ping プローブを送信して RTT を決定します。ping プローブが失敗した場合は、DNS UDP プローブが使用されます。このプローブも失敗した場合、アプライアンスは DNS TCP プローブを使用します。

これらのメカニズムは、Citrix ADC アプライアンスでは負荷分散モニターとして表され、「ldns」プレフィックスを使用しているため、簡単に識別できます。3 つのモニタは、デフォルトの順序で、次のとおりです。

- `ldns-ping`
- `ldns-dns`
- `ldns-tcp`

これらのモニターはアプライアンスに組み込まれており、安全なデフォルトに設定されています。ただし、アプライアンスの他のモニターと同様にカスタマイズできます。

デフォルトの順序は、GSLB パラメーターとして明示的に設定することで変更できます。たとえば、DNS UDP クエリに続いて PING、TCP の順に順序を設定するには、次のコマンドを入力します。

```
1 set gslb parameter -ldnsprobeOrder DNS PING TCP
2 <!--NeedCopy-->
```

カスタマイズされていない限り、Citrix ADC アプライアンスはポート 53 で UDP および TCP プローブを実行しますが、通常の負荷分散モニターとは異なり、プローブが有効な RTT 情報を提供するために成功する必要はありません。ICMP ポート使用不可メッセージ、TCP リセット、および DNS エラー応答（通常は障害を構成する）はすべて、RTT 値の計算に許容されます。

RTT データがコンパイルされると、アプライアンスは独自のメトリック交換プロトコル（MEP）を使用して、参加サイト間で RTT 値を交換します。RTT メトリックを計算した後、アプライアンスは RTT 値をソートして、最良の（最小の）RTT メトリックでデータセンターを特定します。

RTT 情報が利用できない場合（クライアントのローカル DNS サーバーが初めてサイトにアクセスする場合など）、Citrix ADC アプライアンスはラウンドロビン方式を使用してサイトを選択し、クライアントをサイトに誘導します。

動的方式を設定するには、サイトの GSLB 仮想サーバーを動的 RTT 用に構成します。また、ローカル DNS サーバーをプローブする間隔をデフォルト以外の値に設定することもできます。

ダイナミック **RTT** 用の **GSLB** 仮想サーバーの構成

GSLB 仮想サーバーをダイナミック RTT 用に設定するには、RTT ロードバランシング方式を指定します。

Citrix ADC アプライアンスは、特定のローカルサーバーのタイミング情報を定期的に検証します。遅延の変更が設定された許容係数を超えた場合、アプライアンスはデータベースを新しいタイミング情報で更新し、MEP 交換を実行して新しい値を他の GSLB サイトに送信します。デフォルトの許容係数は 5 ミリ秒 (ms) です。

RTT 許容係数は、GSLB ドメイン全体で同一である必要があります。サイトでこれを変更する場合は、GSLB ドメインに展開されているすべての Citrix ADC アプライアンスで同一の RTT 許容係数を設定する必要があります。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーを動的 **RTT** 用に構成するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -lbMethod RTT -tolerance <value>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーを動的 **RTT** 用に構成するには

トラフィック管理に移動します > **GSLB** > 仮想サーバーをクリックし、仮想サーバーをダブルクリックします。

ローカル **DNS** サーバーのプローブ間隔を設定する

Citrix ADC アプライアンスは、ICMP エコー要求または応答（PING）、TCP、UDP などのさまざまなメカニズムを使用して、ローカル DNS サーバーと参加している GSLB サイト間の接続の RTT メトリックを取得します。デフォルトでは、アプライアンスは ping モニターを使用し、5 秒ごとにローカル DNS サーバーをプローブします。次に、アプライアンスは応答を 2 秒間待ちます。その時間内に応答が受信されない場合は、TCP DNS モニターを使用してプローブします。

ただし、ローカル DNS サーバをプローブする時間間隔は、設定に合わせて変更できます。

コマンドラインインターフェイスを使用してプローブ間隔を変更するには

コマンドプロンプトで入力します。


```
set lb monitor <monitorName> <type> -interval <integer> <units> -
    resptimeout <integer> <units>
<!--NeedCopy-->
```

例:

```
set lb monitor ldns-tcp LDNS-TCP -interval 10 sec -resptimeout 5 sec
<!--NeedCopy-->
```

構成ユーティリティを使用してプローブ間隔を変更するには

[トラフィック管理**] > [**負荷分散**] > [**モニター**]** に移動し、変更するモニター（**ping** など）をダブルクリックします。

API メソッド

October 7, 2021

API メソッドを使用して、最適なパフォーマンスの GSLB サービスを判断できます。GSLB の API メソッドは、REST API を使用して、最適なパフォーマンスの GSLB サービスを決定します。

API メソッドでは、GSLB がクライアントから DNS 要求を受信すると、指定されたルールに対して要求を評価します。GSLB が HTTP コールアウト式 SYS.HTTP_CALLOUT (<name>) を検出すると、HTTP コールアウトエージェントに対して REST API リクエストを呼び出します。GSLB は、最高のパフォーマンスを決定するために、HTTP コールアウトエージェントからの応答を使用します。DNS 応答では、GSLB は、最もパフォーマンスの高いサービスの IP アドレスをクライアントに返します。

CLI を使用して GSLB API メソッドを設定するには

GSLB API メソッドを設定するには、次の手順を実行します。

1. HTTP コールアウトを設定します。

   詳細については、「HTTP コールアウトの設定」を参照してください。

   コマンドプロンプトで入力します。


```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-
    port <port>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod (GET | POST)] [-hostExpr <string>] [-urlStemExpr <
    string>] [-headers <name(value)> ...] [-parameters <name(value)
    > ...] [-bodyExpr <string>] [-fullReqExpr <string>] [-scheme (
    http | https)] [-resultExpr <string>] [-cacheForSecs <secs>] [-
    comment <string>]
<!--NeedCopy-->
```

例:

```
add policy httpCallout GSLB_Method_API -IPAddress 208.111.39.237 -
    port 443 -returnType TEXT -hostExpr  “\” hopx.gslb.com\ “”  -
    urlStemExpr  “\” /zones/1/customers/92395/apps/6/decision\ “”
    -headers Authorization( “Basic 19fbe6db-4332-4e3f-a8bc-
    ee47bdc726f8”) -parameters ip(DNS.REQ.OPT.ECS.IP.
    TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme
    https -resultExpr  “HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).
    XPATH_JSON(xp%/providers/Val[1]/provider%)”  -cacheForSecs 30
<!--NeedCopy-->
```

2. ロードバランシングの API メソッドを指定します。GSLB は、指定されたルールに対して DNS 要求を評価します。

   コマンドプロンプトで入力します。

```
add gslb vserver <name> <serviceType>  [-lbMethod <lbMethod>] [-
    backupLBMethod <backupLBMethod>] -rule <expression>
<!--NeedCopy-->
```

例:

```
add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN
     -rule  “sys.http_callout(GSLB_Method_API)”
<!--NeedCopy-->
```

LB メソッドとして **API** を使用して **GSLB** と **ITM** を統合するためのサンプル構成

この構成により、GSLB は Citrix インテリジェントトラフィック管理（ITM）のインターネット可視性の側面を使用して、最高のパフォーマンスを発揮する GSLB サービスを決定できます。


```
/* Enable ns features */

enable ns feature lb gslb cs

/* This is a named expression that is used in the HTTP callout, used
   for result expression. */

add policy expression exp1 "HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).
   XPATH_JSON(xp%/providers/Val[1]/provider%)"

/* This is a named expression that is used in HTTP callout, used for
   host expression. */

add policy expression exp2 ""hopx.cedexis.com""

/* This is the HTTP callout configured to request the ITM for the GSLB
   decision. */

add policy httpCallout ITM_OpenMix_API -IPAddress 208.111.39.237 -port
   80 -returnType TEXT -hostExpr exp2 -urlStemExpr ""/zones/1/customers
   /61770/apps/3/decision"" -headers Authorization("Basic a310697a-1d69
   -48bf-8f36-55742a8e894e") -parameters ip(DNS.REQ.OPT.ECS.IP.
   TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme http -
   resultExpr exp1 -cacheForSecs 30

/* Add service 1 */
add service sg1 98.136.103.24 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
   svrTimeout 360 -CKA NO -TCPB NO -CMP NO

/* Add service 2 */
add service sg2 172.217.194.113 HTTP 80 -gslb NONE -maxClient 0 -maxReq
    0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180
   -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

/* Add ADNS service */

add service adns1 10.102.217.106 ADNS 53 -gslb NONE -maxClient 0 -
   maxReq 0 -cip DISABLED -usip NO -useproxyport NO -sp OFF -cltTimeout
    120 -svrTimeout 120 -CKA NO -TCPB NO -CMP NO

/* Add lb vserver 1 for service 1 */
add lb vserver lbvs1 HTTP 10.102.217.116 80 -persistenceType NONE -
```



```
    cltTimeout 180
29
30 /* Add lb vserver 2 for service 2 */
31 add lb vserver lbvs2 HTTP 10.102.217.117 80 -persistenceType NONE -
    cltTimeout 180
32
33 /* Bind service 1 to lb vserver 1 */
34
35 bind lb vserver lbvs1 sg1
36
37 /* Bind service 2 to lb vserver 2 */
38
39 bind lb vserver lbvs2 sg2
40
41 /* Configure API GSLB method on GSLB virtual server to call the HTTP
    callout. This HTTP callout requests the ITM for the GSLB decision
    and returns GSLB service name, which should serve the request. */
42
43 add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN -
    rule "sys.http_callout(ITM_OpenMix_API)" -tolerance 0
44
45 /* Add GSLB site */
46
47 add gslb site site1 10.102.217.106 -publicIP 10.102.217.106
48
49 /* Add GSLB service 1 */
50
51 add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai_1
    10.102.217.116 HTTP 80 -publicIP 10.102.217.116 -publicPort 80 -
    maxClient 0 -siteName site1 -sitePersistence HTTPRedirect -
    sitePrefix gs2. -cltTimeout 180 -svrTimeout 360 -downStateFlush
    ENABLED
52
53 /* Add GSLB service 2 */
54
55 add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai 10.102.217.117
    HTTP 80 -publicIP 10.102.217.117 -publicPort 80 -maxClient 0 -
    siteName site1 -sitePersistence HTTPRedirect -sitePrefix gs1. -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
56
57 /* Bind the GSLB service 1 to GSLB server 1 */
58 bind gslb vserver vs1 -serviceName
    aws_ec2_ap_south_1_asia_pacific_mumbai_1
59
60 /* Bind the GSLB service 2 to GSLB server 2 */
```



```
bind gslb vserver vs1 -serviceName
    aws_ec2_ap_south_1_asia_pacific_mumbai

/* Bind a domain name to the GSLB virtual server */
bind gslb vserver vs1 -domainName testruchit104.com -TTL 5

<!--NeedCopy-->
```

静的な近接の設定

October 7, 2021

静的近接方式を機能させるには、Citrix ADC アプライアンスを設定して、ロケーションファイルを介して入力された既存の静的近接データベースを使用するか、静的近接データベースにカスタムエントリを追加する必要があります。カスタムエントリを追加した後、ロケーション修飾子を設定できます。データベースを設定したら、GSLB メソッドとして静的な近接を指定する準備が整いました。

このドキュメントでは、次の内容について説明します。

- 位置ファイルを追加して静的近接データベースを作成する
- 静的近接データベースへのカスタムエントリの追加
- ロケーション修飾子の設定
- 近接方法を指定する
- GSLB 静的近接データベースの同期

ロケーションファイルを追加して静的近接データベースを作成する

April 7, 2022

静的近接データベースは UNIX ベースの ASCII ファイルです。ロケーションファイルからこのデータベースに追加されたエントリは、静的エントリと呼ばれます。Citrix ADC アプライアンスにロードできるロケーションファイルは 1 つだけです。新しい位置情報ファイルを追加すると、既存のファイルが上書きされます。静的近接データベースのエントリ数は、Citrix ADC アプライアンスに構成されているメモリによって制限されます。

静的近接データベースは、既定の形式、または市販のサードパーティデータベース (www.maxmind.com や www.ip2location.com など) から派生した形式で作成できます。

Citrix ADC アプライアンスには、次の 2 つの IP ジオロケーションデータベースファイルが含まれています。これらは、MaxMind によって公開された GeoLite2 ファイルです。

- Citrix_netscaler_inbuilt_geoip_db_IPv4
- Citrix_netscaler_inbuilt_geoip_db_ipv6

これらのデータベースファイルは、Citrix ADC アプライアンスがサポートする形式の /var/netscaler/inbuilt_db ディレクトリにあります。

これらの IP ジオロケーションデータベースは、静的近接ベースの GSLB 方式のロケーションファイルとして、またはロケーションベースのポリシーで使用できます。

これらのデータベースは、提供する詳細が異なります。デフォルトのファイルには書式タグがある場合を除き、データベースファイル形式の厳密な適用はありません。データベースファイルは、フィールド区切り文字としてカンマを使用する ASCII ファイルです。フィールドの構造と、ロケーションの IP アドレスの表現には違いがあります。

format パラメータは、Citrix ADC アプライアンスに対するファイルの構造を記述します。format オプションに不正な値を指定すると、内部データが破損する可能性があります。

> 注
>
> - アップグレード後、/var/netscaler/inbuilt_db/ ディレクトリに以前の Citrix ADC ソフトウェアバージョンのデータベースファイル（Citrix_Netscaler_InBuilt_GeoIP_DB.csv）が含まれている場合、そのファイルは保持されます。
> - データベースファイルのデフォルトの場所は/var/netscaler/locdb です。高可用性（HA）セットアップでは、ファイルの同一のコピーが両方の Citrix ADC アプライアンスの同じ場所に存在する必要があります。
> - ロケーションファイルが既定の場所以外の場所に保存されている場合は、ロケーションファイルのパスを指定します。
> - 管理パーティションの場合、デフォルトのパスは /var/partitions/<partitionName>/netscaler/locdb です。
> - 一部のデータベースでは、ISO-3166 に準拠した短い国名や長い国名も提供されています。Citrix ADC は、修飾子を格納および照合するときに短縮名を使用します。
> - 静的近接データベースを作成するには、Citrix ADC アプライアンスの UNIX シェルにログオンし、エディターを使用して、Citrix ADC がサポートする形式のいずれかで場所の詳細を含むファイルを作成します。
> - Citrix ADC アプライアンスは GeoLite2 データベース（IPv4 および IPv6）に同梱されていますが、Citrix は MaxMind GeoLite2 データベースを定期的に保守または更新しません。必要に応じて、www.maxmind.com から GeoLite2 データベースを取得し、Citrix ADC データベース形式に変換できます。詳しくは、MaxMind GeoLite2 データベース形式を Citrix ADC データベース形式に変換するスクリプトを参照してください。

CLI を使用して静的ロケーションファイルを追加するには

コマンドプロンプトで入力します。


```
add locationFile <locationFile> [-format <format>]
-  show locationFile
<!--NeedCopy-->
```

例:

```
add locationFile /var/netscaler/locdb/nsgeo1.0 -format netscaler
Done

show locationFile
Location File: /var/netscaler/locdb/nsgeo1.0
Format: netscaler
 Done
>
<!--NeedCopy-->
```

例:

```
add locationFile /var/netscaler/inbuilt_db/
   Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4 -format netscaler

add locationFile6 /var/netscaler/inbuilt_db/
   Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6 -format netscaler
<!--NeedCopy-->
```

GUI を使用して静的ロケーションファイルを追加するには、次の手順を実行します。

1. [**AppExpert**] > [場所] に移動し、[静的データベース] タブをクリックします。
2. [追加] をクリックして、静的位置情報ファイルを追加します。

インポートしたロケーションファイルデータベースは、構成ユーティリティの [データベースの表示] ダイアログボックスを使用して表示できます。同等の CLI はありません。

GUI を使用して静的ロケーションファイルを表示するには、次の手順を実行します。

1. [**AppExpert**] > [場所] に移動し、[静的データベース] タブをクリックします。
2. 静的ロケーションファイルを選択し、[アクション] リストから [データベースの表示] をクリックします。

ロケーションファイルを **Citrix ADC** 形式に変換するには:

デフォルトでは、位置情報ファイルを追加すると、Citrix ADC 形式で保存されます。他の形式のロケーションファイルを Citrix ADC 形式に変換できます。

> 注: nsmapオプションには、コマンドラインインターフェイスからのみアクセスできます。この変換は、Citrix ADC 形式にのみ可能です。

静的データベース形式を変換するには、**CLI** プロンプトで次のコマンドを入力します。

```
nsmap -f <inputFileFormat> -o <outputFileName> <inputFileName>
<!--NeedCopy-->
```

例**:**

```
 nsmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.
    csv
<!--NeedCopy-->
```

MaxMind GeoLite2 データベース形式を **Citrix ADC** データベース形式に変換するスクリプト

MaxMind GeoIP データベースは、Citrix ADC で直接使用することはできません。MaxMind GeoIP データベースは、Citrix ADC 形式に変換し、GSLB 静的近接方式やポリシーなどの他の機能で IP ロケーションを検出するためにロードする必要があります。
スクリプトを使用して、GeoLite2 データベース形式を Citrix ADC データベース形式に変換できます。このスクリプトは、IPv4 ファイルと IPv6 ファイルの両方を変換するために使用できます。
スクリプトは次の場所にあります。https://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Format

GeoIP2 データベースを **Citrix ADC** 形式に変換する手順

1. GeoLite2 City または GeoLite2 国データベースを.csv 形式でhttps://dev.maxmind.com/geoip/geoip2/geolite2/からダウンロードします。
2. Citrix ADC ディレクトリ（/var など）にファイルをコピーします。次のシェルコマンドを使用してファイルを解凍すると、同じ名前のディレクトリが作成されます。

   ```
   tar -xf <filename>
   ```

3. スクリプト convert_geoipdb_to_netscaler_format.pl をhttps://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Formatからダウンロードし、ステップ #2 で作成したディレクトリにコピーします。
4. スクリプトの実行に使用できるオプションを確認するには、以下のコマンドを実行します。

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl -help
   ```

   以下のようなさまざまなオプションを利用できます。

- <filename> IPv4 出力ファイル。既定の出力ファイル名:Netscaler_Maxmind_GeoIP_DB_IPv4.csv
- -p <filename> IPv6 出力ファイル。既定の出力ファイル名:Netscaler_Maxmind_GeoIP_DB_IPv6.csv
- -logfile <filename> イベント/メッセージのリストを含むファイル
- -debug すべてのメッセージを STDOUT に出力します。

5. 次のコマンドを実行して、GeoLite2 データベース形式を Citrix ADC データベース形式に変換します。

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl
   ```

   > 注: 操作には最大 5 分かかる場合があります。

   スクリプトで使用される既定のファイル名は、MaxMind GeoLite2 City ベースのデータベースのファイル名です。GeoLite2 Country データベースをダウンロードした場合は、リストされているとおりに入力ファイル名を指定する必要があります。

   - -b <filename> 変換する IPv4 ブロックファイルの名前。既定のファイル名:GeoLite2-City-Blocks-IPv4.csv
   - -i <filename> 変換する IPv6 ブロックファイルの名前。既定のファイル名:GeoLite2-City-Blocks-IPv6.csv
   - -l <filename> 変換するロケーションファイルの名前。既定のファイル名:GeoLite2-City-Locations-en.csv

   例:

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl -b GeoLite2-Country-Blocks-IPv4.csv -i GeoLite2-Country-Blocks-IPv6.csv -l GeoLite2-Country-Locations-en.csv
   <!--NeedCopy-->
   ```

   スクリプトの実行後に生成される出力ファイルは次のとおりです。

   - Netscaler_Maxmind_GeoIP_DB_IPv4.csv
   - Netscaler_Maxmind_GeoIP_DB_IPv6.csv

6. データベースの Citrix ADC 形式への変換が完了したら、次のコマンドを使用して使用を開始します。

   ```
   add locationFile <locationFile>
   ```

Citrix ADC アプライアンスにサードパーティの静的データベースファイルを追加する

Citrix ADC アプライアンスにサードパーティの静的データベースファイルを追加するには、次の手順を実行します。

1. www.maxmind.com や www.ip2location.com などのサードパーティベンダーからロケーションデータベースファイルを入手します。

2. WinSCP ユーティリティを使用して、ロケーションデータベースファイルを Citrix ADC アプライアンスにコピーします。

   > 注
   >
   > アプライアンス上のデータベース・ファイルのデフォルトの場所は **/var/netscaler/locdb** です。

3. 以下のコマンドを実行して、静的な位置情報ファイルを追加します。

```
add location file <locationfile Name> -format LocationFormat
<!--NeedCopy-->
```

4. 次のコマンドを実行して、ロケーションデータベースがロードされていることを確認します。

```
show location parameter
<!--NeedCopy-->
```

   このコマンドは、スタティックエントリの数などのパラメータを表示します。データベースが正しく読み込まれていない場合は、エラーメッセージも表示されます。最大 3M-1 (300 万から 1 を引いた) エントリをロードできます。

5. 次のコマンドを実行して、GSLB サイトの場所を表示します。

```
show gslb service
<!--NeedCopy-->
```

   > 注
   >
   > - データベースが正しくロードされると、GSLB サイトの場所がデータベースに自動的に入力されます。
   > - アプライアンスの設定には、ロケーションファイルを 1 つだけ指定できます。
   > - アプライアンスが高可用性セットアップの場合、一方のアプライアンスはもう一方のアプライアンスからデータベースをコピーする必要があります。
   > - 着信 IP アドレスに一致するものが見つからない場合、要求はラウンドロビン方式を使用して処理されます。

6. 次のコマンドを実行して、アプライアンスで GSLB メソッドを設定します。

```
set gslb vserver GSLBVserverName -lbMethod MethodType
<!--NeedCopy-->
```


静的近接データベースへのカスタムエントリの追加

October 7, 2021

カスタムエントリは、近接データベースのスタティックエントリよりも優先されます。最大 500 個のカスタムエントリを追加できます。カスタム・エントリの場合は、省略されたすべての修飾子をアスタリスク（*）で指定します。修飾子の名前にピリオドまたはスペースが含まれている場合は、パラメータを二重引用符で囲みます。最初の 31 文字は、各修飾子に対して評価されます。静的近接 GSLB 方式でサービスを選択するために、IP アドレス範囲の地理的位置の経度と緯度を指定することもできます。

コマンドラインインターフェイスを使用してカスタムエントリを追加するには

コマンドプロンプトで次のコマンドを入力して、スタティック近接データベースにカスタムエントリを追加し、構成を確認します。

```
add location < IPfrom> < IPto> <preferredLocation> [-longitude <integer
    >[-latitude <integer>]]
show location
<!--NeedCopy-->
```

例:

```
>add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
<!--NeedCopy-->
```

```
>show location
<!--NeedCopy-->
```

カスタムエントリを追加するためのパラメータ

- IPfrom

  範囲内の最初の IP アドレス（ドット付き 10 進表記）。これは必須の引数です。

- IPto

  範囲内の最後の IP アドレス（ドット付き 10 進表記）。これは必須の引数です。

- preferredLocation

  IP アドレス範囲の地理的位置を表す修飾子の文字列(ドット付き)。各修飾子は、大陸.country.region.city.isp.組織のように、それに先行する修飾子よりも具体的です。たとえば、”NA.US.CA.San Jose.ATT.citrix” と表示されます。

  注: ドット (.) またはスペース () を含む修飾子は、二重引用符で囲む必要があります。

  これは必須の引数です。最大長: 197

- longitude

  IP アドレス範囲の地理的位置の経度を指定する数値値（度単位)。

  注：経度と緯度のパラメータは、静的近接 GSLB メソッドでサービスを選択するために使用されます。指定しない場合、その場所に対して指定された修飾子に基づいて選択されます。

  最大値:180

- latitude

  IP アドレス範囲の地理的位置の緯度を指定する数値値（度単位)。

  注：経度と緯度のパラメータは、静的近接 GSLB メソッドでサービスを選択するために使用されます。指定しない場合、その場所に対して指定された修飾子に基づいて選択されます。

  最大値:180

構成ユーティリティを使用してカスタムエントリを追加するには

[**AppExpert**] > [場所] に移動し、[カスタムエントリ] タブをクリックして、カスタムエントリを追加します。

ロケーション修飾子の設定

April 25, 2022

静的近接の実装に使用されるデータベースには、GSLB サイトの場所があります。各ロケーションには IP アドレス範囲と、その範囲に対して最大 6 つの修飾子があります。修飾子はリテラル文字列で、実行時に所定の順序で比較されます。すべての場所には、少なくとも 1 つの修飾子が必要です。クオリファイアラベルは、ユーザ定義であるクオリファイア（コンテキスト）の意味を定義します。Citrix ADC には 2 つの組み込みコンテキストがあります。

地理的コンテキスト。次の修飾子ラベルがあります。

- 予選 1 —「大陸」
- 予選 2 —「国」
- 修飾子 3 —「状態」
- 予選 4 —「都市」

- 予選 5 —「ISP」
- 予選 6 —「組織」

カスタムエントリ。次のクオリファイア・ラベルがあります。

- 予選 1 —「予選 1」
- クオリファイア 2 —「クオリファイア 2」
- 予選 3 —「予選 3」
- クオリファイア 4 —「クオリファイア 4」
- クオリファイア 5 —「クオリファイア 5」
- クオリファイア 6 —「クオリファイア 6」

地理コンテキストが [大陸] 修飾子なしで設定されている場合、[大陸] は [国] から派生します。組み込みの修飾子ラベルもコンテキストに基づいており、ラベルは変更できます。これらの修飾子ラベルは、静的近接性の決定に使用される IP アドレスでマッピングされる場所を指定します。

静的近接ベースの決定を実行するために、Citrix ADC アプライアンスは、ローカル DNS サーバーリゾルバーの IP アドレスから派生した場所属性（修飾子）を、参加サイトの場所属性と比較します。一致するサイトが 1 つだけの場合、アプライアンスはそのサイトの IP アドレスを返します。複数の一致がある場合、選択されたサイトは、一致する GSLB サイトでのラウンドロビンの結果です。一致するものがない場合、選択されたサイトは、すべての構成済みサイトでのラウンドロビンの結果です。修飾子がないサイトは一致と見なされます。

ロケーションベースのポリシー式の GEO 規則を使用すると、ワイルドカードの一致を確認できます。この機能は、ワイルドカード修飾子がワイルドカード以外の修飾子を含む他の修飾子と一致するかどうかをチェックします。ワイルドカード照合は、`set locationParameter`コマンドに追加された`matchWildcardtoany`属性を使用して実行されます。

`matchWildcardtoany`属性は、次の値に設定できます。

- はい**:** ワイルドカード修飾子は他の修飾子と一致します。
- いいえ**:** ワイルドカード修飾子は非ワイルドカード修飾子とは一致しませんが、他のワイルドカード修飾子と一致します。デフォルトのオプションは [いいえ] です。
- 式：式内のワイルドカード修飾子は、LDNS ロケーションの任意の修飾子と一致しますが、LDNS ロケーションのワイルドカード修飾子は式の非ワイルドカード修飾子と一致しません。

例:

```
1 add dns policy policy1 ”CLIENT.IP.SRC.MATCHES_LOCATION(”Continent.
     country \*.\*.\*.\* \ “)”  <action>
2 <!--NeedCopy-->
```

CLI を使用してロケーションパラメータを設定するには

コマンドプロンプトで入力します。


```
set locationparameter -context <context> -q1label <string> [-q2label <
    string>] [-q3label <string>] [-q4label <string>] [-q5label <string>]
     [-q6label <string>] -matchWildcardtoany [Yes | No | Expression]
<!--NeedCopy-->
```

例:

```
set locationparameter -context custom -q1label asia -matchWildcardtoany
     Yes
<!--NeedCopy-->
```

GUI を使用してロケーションパラメータを設定するには

1. [トラフィック管理] > [GSLB] > [データベースとエントリ] に移動します。
2. [設定] で、[位置パラメータの変更] をクリックします。
3. [ロケーションパラメータの設定] ページで、ロケーションパラメータを設定します。

設定例 (CLI を使用)

次のネットワーク設定について考えてみます。

- GSLB 仮想サーバー名:gv1
- GSLB 仮想サーバー IP アドレス:1.1.1.2
- GSLB サービス:gsvc1 が gv1 にバインドされました
- ロケーションデータベースファイル名:sample.csv
- 位置情報修飾子: 修飾子 1 と 2 が設定されます。残りはワイルドカードと一致するように設定されます。
  - 予選 1 — アジア
  - 修飾子 2 — IR
  - 予選 3-*
  - 予選 4-*
  - 予選 5-*
  - 予選 6-*
- DNS Policy: ポリシーの pol1 は、一致するパケットがあればパケットをドロップするように設定されています。

location パラメータを設定し、DNS ポリシーを次のように設定します。


```
set locationParameter -q2label Country_Code -q3label Subdivision_1_Name
    -q4label Subdivision_2_Name -q5label City

add locationFile "/var/netscaler/inbuilt_db/sample.csv"

add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0

add dns policy pol1 "CLIENT.IP.SRC.MATCHES_LOCATION("Asia.IR
   .\*.\*.\*.\*")||CLIENT.IP.SRC.MATCHES_LOCATION("Asia.SY.\*.\*.\*.\*"
   )||CLIENT.IP.SRC.MATCHES_LOCATION("Asia.SD.\*.\*.\*.\*")||CLIENT.IP.
   SRC.MATCHES_LOCATION("Asia.KP.\*.\*.\*.\*")||CLIENT.IP.SRC.
   MATCHES_LOCATION("North America.CU.\*.\*.\*.\*")||CLIENT.IP.SRC.
   MATCHES_LOCATION("Europe.UA.Crimea.\*.\*.\*.\*")"
   dns_default_act_Drop

bind dns global pol1 1 -gotoPriorityExpression 65535 -type REQ_DEFAULT

add gslb service gsvc1 1.1.1.2 HTTP 80 -publicIP 1.1.1.2 -publicPort 80
    -maxClient 0 -healthMonitor NO -siteName s1 -cltTimeout 180 -
   svrTimeout 360 -downStateFlush ENABLED

bind gslb vserver gv1 -serviceName gsvc1

bind gslb vserver gv1 -domainName www.gslbnew.com -TTL 5
<!--NeedCopy-->
```

ロケーション DB ファイルに次のクライアントエントリを追加します。この例では、ロケーション DB ファイル名は sample.csv です。

```
10.106.24.170,10.106.24.190,,,,,,,8.0000,47.0000

10.102.82.170,10.102.82.190,Asia,,,,,,-73.9924,40.7553

10.106.24.140,10.106.24.150,,IR,,,,,51.4231,35.6961
<!--NeedCopy-->
```

前述の設定によると、10.106.24.170 ～10.106.24.190 の間のクライアントには、ワイルドカード修飾子が定義されていません。10.106.24.140 と 10.106.24.150 の間のクライアントには、IR として修飾子 2 があります。

match ワイルドカード修飾子を NO に設定します。

```
set locationparameter -matchWildcardtoany no
```



```
<!--NeedCopy-->
```

match ワイルドカード修飾子が NO に設定されている場合、ワイルドカード修飾子は定義されたワイルドカード修飾子にのみ一致します。他のワイルドカード以外の修飾子には一致しません。

- 10.106.24.147 の DNS クエリは、定義されているワイルドカード修飾子（修飾子 2 = IR）に一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.147 クライアントでこの`dig @10.102.82.13 www.gslbnew.com`コマンドを実行すると、サーバに到達できなかったことが出力に表示されます。

```
root@ns#  dig @10.102.82.13 www.gslbnew.com

; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
<!--NeedCopy-->
```

- 10.106.24.180 から送信された DNS クエリが、定義された修飾子と一致しません。DNS ポリシーは有効にならず、クエリが処理されます。

  10.106.24.180 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、GSLB 仮想サーバーの IP アドレスが表示されます。

```
root@ns# dig @10.102.82.13 www.gslbnew.com

; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64265
;; flags: qr aa rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
   ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;www.gslbnew.com. IN    A

;; ANSWER SECTION:
www.gslbnew.com.    5   IN  A   1.1.1.2
```



```
18
19   ;; Query time: 12 msec
20   ;; SERVER: 10.102.82.13#53(10.102.82.13)
21   ;; WHEN: Tue Mar 29 22:46:40 UTC 2022
22   ;; MSG SIZE  rcvd: 60
23   <!--NeedCopy-->
```

match ワイルドカード修飾子を Yes に設定します。

```
1 set locationparameter -matchWildcardtoany yes
2 <!--NeedCopy-->
```

match ワイルドカード修飾子が yes に設定されている場合、ワイルドカード修飾子は任意のワイルドカード修飾子 (定義済みおよび非ワイルドカード修飾子) に一致します。

- 10.106.24.147 の DNS クエリは、定義された修飾子（修飾子 2 = IR）に一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.147 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、サーバが到達不能だったことが示されています。

```
1   root@ns#  dig @10.102.82.13 www.gslbnew.com
2
3   ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
4   ; (1 server found)
5   ;; global options: +cmd
6   ;; connection timed out; no servers could be reached
7   <!--NeedCopy-->
```

- 10.106.24.180 からのクエリは、ワイルドカード以外の修飾子と一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.180 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、サーバが到達不能だったことが示されています。

```
1   root@ns# dig @10.102.82.13 www.gslbnew.com
2
3   ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
4   ; (1 server found)
5   ;; global options: +cmd
6   ;; connection timed out; no servers could be reached
```



```
7   <!--NeedCopy-->
```

match ワイルドカード修飾子を [式:] に設定します。

```
1 set locationparameter -matchWildcardtoany expression
2 <!--NeedCopy-->
```

match ワイルドカード修飾子が expression に設定されている場合、ワイルドカード修飾子は DNS ポリシーで使用可能な修飾子、またはロケーション DB ファイルで使用可能な修飾子のいずれかに一致します。

- 10.106.24.147 の DNS クエリは、DNS ポリシーで定義されているワイルドカード修飾子と一致します。そのため、DNS ポリシーが有効になり、クエリが破棄されます。

  10.106.24.147 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、サーバが到達不能だったことが示されています。

```
1   root@ns#  dig @10.102.82.13 www.gslbnew.com
2
3   ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
4   ; (1 server found)
5   ;; global options: +cmd
6   ;; connection timed out; no servers could be reached
7   <!--NeedCopy-->
```

- 10.106.24.180 から送信されたクエリが、DNS ポリシーの修飾子と一致しません。そのため、DNS ポリシーは有効にならず、クエリが処理されます。

  10.106.24.180 クライアントで`dig @10.102.82.13 www.gslbnew.com`コマンドを実行します。出力には、GSLB 仮想サーバーの IP アドレスが表示されます。

```
1   root@ns# dig @10.102.82.13 www.gslbnew.com
2
3   ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
4   ; (1 server found)
5   ;; global options: +cmd
6   ;; Got answer:
7   ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64265
8   ;; flags: qr aa rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
       ADDITIONAL: 1
9   ;; WARNING: recursion requested but not available
10
```



```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;www.gslbnew.com.  IN   A

;; ANSWER SECTION:
www.gslbnew.com.    5   IN  A   1.1.1.2

;; Query time: 12 msec
;; SERVER: 10.102.82.13#53(10.102.82.13)
;; WHEN: Tue Mar 29 22:46:40 UTC 2022
;; MSG SIZE  rcvd: 60
<!--NeedCopy-->
```

近接方法を指定

October 7, 2021

スタティック近接データベースを設定したら、GLSB 方式としてスタティック近接を指定できます。

コマンドラインインターフェイスを使用して静的な近接を指定するには

コマンドプロンプトで次のコマンドを入力して、静的な近接性を構成し、構成を確認します。

```
set gslb vserver <name> -lbMethod STATICPROXIMITY
show gslb vserver <name>
<!--NeedCopy-->
```

例:

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
<!--NeedCopy-->
```

構成ユーティリティを使用して静的近接を指定するには

1. トラフィック管理に移動します > GSLB > 仮想サーバーをクリックし、仮想サーバーをダブルクリックします。

2. [メソッド] セクションをクリックし、[メソッドの選択] ドロップダウンリストから [**STATICPROXIMITY**] を選択します。

GSLB 静的近接データベースの同期

October 7, 2021

グローバルサーバー負荷分散（GSLB）静的近接データベースを同期するには、いずれかのサイトをマスター GSLB ノードとして識別する必要があります。トポロジ内の任意のサイトをマスターノードとして指定することができます。残りの GSLB ノードは自動的にスレーブノードとして指定されます。

GSLB 静的近接データベースを同期すると、スレーブノード間で /var/netscaler/locdb ディレクトリ内のファイルが同期されます。同期処理中、マスターノードは各スレーブノードから実行構成をフェッチし、マスターノード上の構成と比較します。マスター GSLB ノードは rsync プログラムを使用して、スレーブノード間で静的な近接データベースを同期します。同期プロセスを高速化するために、rsync プログラムは、2 つのファイルの違いを排除するのに十分な変更を加えます。同期プロセスをロールバックできません。

次の例では、スレーブサイトである Site2 をマスターサイト Site1 に同期します。管理者は、サイト 1 で **sync gslb config** コマンドを入力します。

```
sync gslb config -nowarn
Sync Time: Feb 24 2014 14:56:16
Retrieving local site info: ok
Retrieving all participating gslb sites info:
0 bytes in 0 blocks
ok
site1[Master]:
        Getting Config: ok
site2[Slave]:
        Syncing gslb static proximity database: ok
        Getting Config: ok
        Comparing config: ok
        Applying changes: ok
 Done
<!--NeedCopy-->
```

サイト間通信の構成

October 7, 2021

GSLB サイト間通信は、通信サイトに関連付けられているリモートプロシージャコール (RPC) ノード間で行われます。マスター GSLB サイトは、スレーブサイトとの接続を確立して、GSLB 構成情報を同期し、サイトメトリックを交換します。

RPC ノードは、GSLB サイトが作成されると自動的に作成され、内部で生成されたユーザー名とパスワードが割り当てられます。Citrix ADC アプライアンスは、このユーザー名とパスワードを使用して、接続の確立中にリモート GSLB サイトに対して自身を認証します。RPC ノードの設定手順は必要ありませんが、任意のパスワードを指定したり、GSLB サイトが交換する情報を暗号化してセキュリティを強化したり、RPC ノードのソース IP アドレスを指定することができます。

アプライアンスでは、他の GSLB サイトと通信するときにソース IP アドレスとして使用する Citrix ADC が所有する IP アドレスが必要です。既定では、RPC ノードはサブネット IP (SNIP) アドレスのいずれかを使用しますが、任意の IP アドレスを指定することもできます。

以下のトピックでは、Citrix ADC アプライアンスでの RPC ノードの動作と構成について説明します。

RPC ノードのパスワードの変更

各 RPC ノードのパスワードを変更して、GSLB セットアップ内のサイト間の通信を保護することをお勧めします。ローカルサイトの RPC ノードのパスワードを変更した後、各リモートサイトの RPC ノードに変更を手動で伝達する必要があります。

パスワードは暗号化された形式で保存されます。パスワードが変更されたことを確認するには、show rpcNode コマンドを使用して、変更前と変更後のパスワードの暗号化形式を比較します。

> 注：GSLB は、内部ユーザーアカウントを使用しています。セキュリティを強化するには、インターネットユーザーアカウントのパスワードも変更することをお勧めします。内部ユーザーアカウントのパスワードは、RPC ノードパスワードによって変更されます。

コマンドラインインターフェイスを使用して **RPC** ノードのパスワードを変更するには

コマンドラインで次のコマンドを入力して、RPC ノードのパスワードを変更します。

```
set ns rpcNode <IPAddress> {
 -password }

show ns rpcNode
<!--NeedCopy-->
```

例**:**


```
> set rpcNode 192.0.2.4 -password mypassword
 Done
> show rpcNode
.
.
.
2)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
    SrcIP:  *                Secure:  OFF
Done
>

<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **RPC** ノードのパスワードの設定を解除するには

CLI を使用して RPC ノードのパスワードの設定を解除するには、unset rpcNode コマンド、RPC ノードの IP アドレス、およびパスワードパラメータを値なしで入力します。

構成ユーティリティを使用して **RPC** ノードのパスワードを変更するには

[システム] > [ネットワーク] > [RPC] に移動し、RPC ノードを選択してパスワードを変更します。

サイト指標の交換を暗号化

GSLB セットアップで RPC ノードのセキュアオプションを設定することで、GSLB サイト間で交換される情報を保護できます。Secure オプションを設定すると、Citrix ADC アプライアンスは、ノードから他の RPC ノードに送信されるすべての通信を暗号化します。

コマンドラインインターフェイスを使用してサイトメトリックスの交換を暗号化するには

コマンドプロンプトで次のコマンドを入力して、サイトメトリックの交換を暗号化し、構成を確認します。

```
set ns rpcNode <IPAddress> [-secure ( YES | NO )]
show rpcNode
<!--NeedCopy-->
```

例**:**


```
> set rpcNode 192.0.2.4 -secure YES
 Done
>
> show rpcNode
.
.
.
3)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:
    192.0.2.3     Secure:  ON
 Done
>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してセキュアパラメーターを設定解除するには

CLI を使用してセキュアパラメータの設定を解除するには、unset rpcNode コマンド、RPC ノードの IP アドレス、および secure パラメータを値なしで入力します。

Citrix ADC 構成ユーティリティを使用してサイトメトリックの交換を暗号化するには

1. [システム] > [ネットワーク] > [RPC] に移動し、RPC ノードをダブルクリックします。
2. 「セキュア」オプションを選択し、「**OK**」をクリックします。

RPC ノードの送信元 **IP** アドレスを構成する

デフォルトでは、Citrix ADC アプライアンスは RPC ノードの送信元 IP アドレスとして Citrix ADC 所有サブネット IP（SNIP）アドレスを使用しますが、特定の SNIP アドレスを使用するようにアプライアンスを構成できます。SNIP アドレスが使用できない場合、GSLB サイトは他のサイトと通信できません。このようなシナリオでは、RPC ノードのソース IP アドレスとして NSIP アドレスまたは仮想 IP (VIP) アドレスのいずれかを構成する必要があります。RPC ノードがリモートノードである場合にのみ、VIP アドレスを RPC ノードの送信元 IP アドレスとして使用できます。VIP アドレスを送信元 IP アドレスとして設定し、その VIP アドレスを削除すると、アプライアンスでは SNIP アドレスが使用されます。

> 注
>
> NetScaler 11.0.64.x 以降では、RPC ノードのソース IP アドレスとして GSLB サイトの IP アドレスを使用するようにアプライアンスを構成できます。

コマンドラインインターフェイスを使用して **RPC** ノードの送信元 **IP** アドレスを指定するには

コマンドプロンプトで次のコマンドを入力して、RPC ノードの送信元 IP アドレスを変更し、構成を確認します。


```
set ns rpcNode <IPAddress> [-srcIP <ip_addr|ipv6_addr|*>]
show ns rpcNode
<!--NeedCopy-->
```

例:

```
set rpcNode 192.0.2.4 -srcIP 192.0.2.3
Done
show rpcNode
<!--NeedCopy-->
```

```
IPAddress: 192.0.2.4 Password: d336004164d4352ce39e SrcIP: 192.0.2.3
   Secure: OFF
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して送信元 **IP** アドレスパラメーターを設定解除するには

CLI を使用して送信元 IP アドレスパラメータの設定を解除するには、設定解除された rpcNodecand、RPC ノードの IP アドレス、および srcIP パラメータを値なしで入力します。

Citrix ADC 構成ユーティリティを使用して **RPC** ノードの送信元 **IP** アドレスを指定するには

1. [システム] > [ネットワーク] > [RPC] に移動し、RPC ノードをダブルクリックします。
2. [Source IP Address] フィールドに、RPC ノードが送信元 IP アドレスとして使用する IP アドレスを入力し、[OK] をクリックします。

> 重要
>
> RPC ノードの送信元 IP アドレスは各 Citrix ADC アプライアンスに固有であるため、GSLB に参加しているサイト間でソース IP アドレスを同期できません。したがって、同期を強制した後（sync gslb config —forceSync コマンドを使用するか、GUI で ForceSync オプションを選択）、他の Citrix ADC アプライアンスのソース IP アドレスを手動で変更する必要があります。

メトリック交換プロトコルの構成

October 7, 2021

GSLB セットアップのデータセンターは、Citrix ADC アプライアンスの独自のプロトコルであるメトリック交換プロトコル（MEP）を介して相互にメトリックを交換します。メトリック情報の交換は、GSLB サイトの作成時に開始されます。これらのメトリックは、ロード、ネットワーク、およびパーシステンス情報で構成されます。

MEP は、データセンターの可用性を確保するためにデータセンターの健全性チェックに必要です。ネットワークメトリックスを交換するための接続（ラウンドトリップ時間）は、交換に関係するいずれかのデータセンターから開始できますが、サイトメトリックスを交換するための接続は、常に下位の IP アドレスを持つデータセンターによって開始されます。デフォルトでは、データセンターはサブネット IP アドレス（SNIP）を使用して、別のデータセンターの IP アドレスへの接続を確立します。ただし、メトリック交換のソース IP アドレスとして、特定の SNIP、仮想 IP（VIP）アドレス、または NSIP アドレスを構成できます。GSLB サイト間の通信プロセスでは TCP ポート 3011 または 3009 が使用されるため、このポートは Citrix ADC アプライアンス間のファイアウォールで開く必要があります。

注:SNIP または GSLB サイトの IP アドレスは、メトリック交換のソース IP アドレスとして設定できます。詳細については、「RPC ノードの送信元 IP アドレスの構成」を参照してください。

送信元サイトとターゲットサイト（MEP 接続を開始するサイト、および接続要求を受信するサイト）にプライベート IP アドレスとパブリック IP アドレスの両方が設定されている場合、サイトはパブリック IP アドレスを使用して MEP 情報を交換します。

「GSLB サービスのモニタリング」の説明に従って、モニターをバインドしてリモートサービスの状態をチェックすることもできます。モニタがバインドされている場合、メトリック交換はリモートサービスの状態を制御しません。モニタがリモートサービスにバインドされ、メトリック交換が有効になっている場合、モニタは健全性ステータスを制御します。モニターをリモートサービスにバインドすると、Citrix ADC アプライアンスが非シトリックス ADC 負荷分散デバイスと対話できるようになります。Citrix ADC アプライアンスは、非シトリックス ADC デバイスを監視できますが、モニターがすべての GSLB サービスにバインドされ、静的負荷分散方法（ラウンドロビン、静的近接、ハッシュベースの方法など）のみが使用されない限り、それらのデバイスで負荷分散を実行できません。。

NetScaler リリース 11.1.51.x 以降では、不要なサービスの中断を回避するために、MEP 接続がダウンしたときに GSLB サービスを DOWN としてマークする時間遅延を設定できます。

高可用性セットアップでの **MEP** 状態

高可用性セットアップでは、プライマリノードがリモートサイトとの接続を確立し、MEP 状態がプライマリノードからセカンダリノードに同期されません。したがって、セカンダリノードの MEP 状態は DOWN のままです。セカンダリノードがプライマリになると、新しい GSLB サイトとの MEP 接続を確立し、それに応じて MEP の状態を更新します。

サイトメトリックス交換を有効にする

GSLB サイト間で交換されるサイトメトリックには、各負荷分散の状態、コンテンツスイッチング仮想サーバーの状態、現在の接続数、現在のパケットレート、および現在の帯域幅の使用状況に関する情報が含まれます。

Citrix ADC アプライアンスは、サイト間の負荷分散を実行するためにこの情報を必要とします。サイトメトリック交換間隔は 1 秒です。リモート GSLB サービスをローカル GSLB 仮想サーバーにバインドして、リモートサービスとの

サイトメトリックの交換を有効にする必要があります。

コマンドラインインターフェイスを使用してサイトメトリック交換を有効または無効にするには

コマンドプロンプトで次のコマンドを入力して、サイトメトリック交換を有効または無効にし、構成を確認します。

```
set gslb site <siteName> -metricExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

例:

```
set gslb site Site-GSLB-East-Coast -metricExchange ENABLED
set gslb site Site-GSLB-East-Coast -metricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

GUI を使用してサイトメトリック交換を有効または無効にするには

1. 「トラフィック管理」>「**GSLB** 」>「サイト」に移動し、サイトを選択します。
2. [**GSLB** サイトの構成] ダイアログボックスで、[メトリック交換] オプションを選択します。

ネットワークメトリック交換を有効にする

GSLB サイトでラウンドトリップ時間 (RTT) 負荷分散方式を使用している場合は、クライアントのローカル DNS サービスに関する RTT 情報の交換を有効または無効にできます。この情報は 5 秒ごとに交換されます。

GSLB メソッドを RTT に基づくメソッドに変更する方法の詳細については、GSLB メソッドを参照してください。

コマンドラインインターフェイスを使用してネットワークメトリック情報の交換を有効または無効にするには

コマンドプロンプトで次のコマンドを入力して、ネットワークメトリック情報の交換を有効または無効にし、構成を確認します。

```
set gslb site <siteName> -nwmetricExchange (ENABLED|DISABLED)
show gslb site <<siteName>
<!--NeedCopy-->
```


例:

```
set gslb site Site-GSLB-East-Coast -nwmetricExchange ENABLED
set gslb site Site-GSLB-East-Coast -nwmetricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

GUI を使用してネットワークメトリック情報交換を有効または無効にするには

1. [トラフィック管理] > [**GSLB**] > [サイト] に移動します。
2. [**GSLB** サイトの構成] ダイアログボックスで、[ネットワークメトリック交換] オプションを選択します。

MEP 接続が **DOWN** になったときに **GSLB** サービスが **DOWN** としてマークされるまでの遅延時間の設定

リモートサイトへの MEP 接続のステータスが DOWN に変わると、そのリモートサイト上のすべての GSLB サービスのステータスは DOWN としてマークされますが、サイトは実際に DOWN ではない可能性があります。

サイトが DOWN としてマークされる前に、MEP 接続を再確立するための遅延を設定できるようになりました。遅延が期限切れになる前に MEP 接続が UP 状態に戻った場合、サービスは影響を受けません。

たとえば、遅延 10 を設定すると、MEP 接続が 10 秒間ダウンされるまで、GSLB サービスは DOWN としてマークされます。MEP 接続が 10 秒以内に UP に戻った場合、GSLB サービスは UP 状態のままになります。

注：この遅延は、モニタにバインドされていないサービスにのみ適用されます。遅延はトリガーモニタには影響しません。

コマンドラインインターフェイスを使用して遅延時間を設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set gslb parameter** - GSLBSvcStateDelayTime <sec>
<!--NeedCopy-->
```

例:

gslb パラメータを設定する -GSLBSvc 状態遅延時間 10

> 注
>
> 階層展開（親子トポロジ）で、親サイトと子サイトの両方で GSLB サービスを構成する場合は、親サイトと子サイトの両方で GSLB パラメーターを設定します。子サイトで GSLB サービスを構成しない場合は、親サイト

> でのみ GSLB パラメーターを設定してください。

GUI を使用して遅延時間を設定するには

1. [設定] > [トラフィック管理] > [**GSLB**] > [**GSLB** 設定の変更] に移動します。
2. [**GSLB** サービス状態遅延時間 **(**秒**)**] ボックスに、遅延時間を秒単位で入力します。

GSLB サービスのフラップを回避するために、**MEP** 接続ステータスがアップしたときの **GSLB** サービスの学習時間を設定します

ノードが再起動するとき、または HA フェールオーバー中にシステムが初期化されます。次に、ノードは MEP を介してサービス状態をリモートノードに伝達するために、設定されたローカルおよび子サービスに関する最新情報を学習する必要があります。ノードは正しい情報を学習するのに少し時間がかかります。一方、ピアノードがこのノードに接続して更新を要求すると、ノードが誤ったサービス状態と統計を送信する可能性があります。この誤った情報は、リモートピアノードでサービスフラップやその他の機能に関連する問題を引き起こす可能性があります。このシナリオを回避するために、ローカルおよび子の GSLB サービスの学習時間を設定できるようになりました。

学習タイムアウトを設定すると、GSLB サイトはローカルサービスおよび子サービスに関する正しい統計情報を学習するためのバッファ時間（ラーニングタイムアウト）を取得します。サービスがラーニングフェーズにあるとき、リモート GSLB サイトは MEP アップデートでこの情報を取得し、そのサービスについて MEP を通じて受信したプライマリサイトの状態と統計情報を尊重しません。

GSLB サービスは、次のいずれかのシナリオで学習フェーズに入ります。

- Citrix ADC アプライアンスが再起動されました
- 高可用性フェイルオーバーが発生しました
- クラスタ GSLB セットアップの所有者ノードが変更されました
- MEP がローカルノードで有効になっている
- GSLB のサイトは島のシナリオから出てくる。GSLB サイトは、他のサイトに接続されていない場合、アイランドになります。

親子展開では、プライマリ親がダウンすると、バックアップの親（構成されている場合）は、採用された子サイトの GSLB サービスを学習フェーズに選択的に移動します。

CLI を使用してサービス状態の学習時間を設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set gslb parameter - SvcStateLearningTime <sec>
<!--NeedCopy-->
```


“SvcStateLearningTime” は秒単位で設定できます。デフォルト値は 0 で、最大値は 3600 です。このパラメーターは、モニターが GSLB サービスにバインドされていない場合にのみ適用されます。

例:

```
set gslb parameter  -  SvcStateLearningTime 10
<!--NeedCopy-->
```

GUI を使用してサービス状態の学習時間を設定するには

1. [構成] **>** [トラフィック管理] **>** [**GSLB**] **>** [ダッシュボード] **>** [**GSLB** 設定の変更] に移動します。

   [**GSLB** パラメータの設定] ページが表示されます。

2. [**GSLB** サービス状態の学習時間 **(**秒**)**] フィールドに、学習時間を秒単位で入力します。

パーシステンス情報交換の有効化

Citrix ADC アプライアンスを構成して、グループ内の任意の仮想サーバーへのクライアント転送を、同じクライアントから以前の転送を受信したサーバーに送信できるようにすることができます。

各サイトでパーシステンス情報の交換を有効または無効にできます。この情報は、GSLB に参加している Citrix ADC アプライアンス間で 5 秒に 1 回交換されます。

パーシステンスの設定の詳細については、永続接続の設定を参照してください。

コマンドラインインターフェイスを使用してパーシステンス情報の交換を有効または無効にするには

コマンドプロンプトで次のコマンドを入力して、パーシステンス情報交換を有効または無効にし、構成を確認します。

```
set gslb site <siteName> -sessionExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

例:

```
set gslb site Site-GSLB-East-Coast -sessionExchange  ENABLED
set gslb site Site-GSLB-East-Coast -sessionExchange  DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```


GUI を使用してパーシステンス情報の交換を有効または無効にするには

1. 「トラフィック管理」>「**GSLB** 」>「サイト」に移動し、サイトをダブルクリックします。
2. [**GSLB** サイトの構成] ダイアログボックスで、[永続セッションエントリ交換] チェックボックスをオンまたはオフにします。

ウィザードを使用した **GSLB** の構成

October 7, 2021

ウィザードを使用して、GSLB 展開の種類 (アクティブ-アクティブ、アクティブ-パッシブ、親子など) を構成できるようになりました。

このウィザードは GUI で使用できます。ウィザードにアクセスするには、[設定] > [トラフィック管理] > [**GSLB**] に移動し、[は じめに] をクリックします。

GSLB ダッシュボードからこのウィザードにアクセスすることもできます。[設定] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動し、[**GSLB** の設定] をクリックします。

注: GSLB エンティティを個別に設定することもできます。

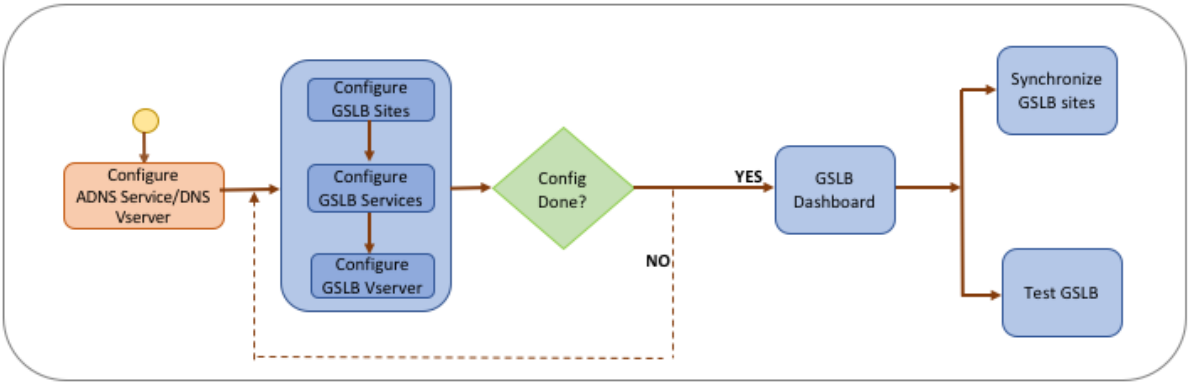
- アクティブ-アクティブサイトの構成
- アクティブ-パッシブサイトの構成
- 親子トポロジの設定

> 重要
>
> この機能は、高可用性展開でサポートされ、管理パーティションおよびクラスタ展開ではサポートされません。

アクティブ-アクティブサイトの構成

October 7, 2021

次の図は、GSLB アクティブ-アクティブサイト構成に関連するワークフローを示しています。



アクティブ-アクティブサイトの構成を開始する前に、各サーバーファームまたはデータセンターに対して標準の負荷分散設定を構成していることを確認してください。

また、デプロイメント内の GSLB サイト間で GSLB 設定を同期するには、次のことを確認してください。

- ローカル GLSB サイトは、GSLB 設定内のすべてのアプライアンスで設定されます。
- 構成内のすべての GSLB サイトで管理アクセスを有効にしました。
- 自動同期および MEP 接続を受け入れるようにファイアウォールを設定しました。
- マスターおよびスレーブの Citrix ADC アプライアンスでは、同じバージョンの Citrix ADC ソフトウェアが実行されています。
- サイトとして参加するすべての Citrix ADC アプライアンスは、同じ Citrix ADC ソフトウェアバージョンを持つ必要があります（これらのサイトはマスタースレーブ関係ではありません）。
- RPC ノードのパスワードは、GSLB 設定内のすべての GSLB サイトで同じです。

ウィザードを使用してアクティブ-アクティブサイトを構成するには

[構成] タブで、次の操作を行います。

1. [トラフィック管理] > [**GSLB**] に移動し、[は じめに] をクリックします。
2. サイトの ADNS サービスまたは DNS 仮想サーバーを構成していない場合は、ここで構成できます。
   a) [表示] をクリックし、[追加] をクリックします。
   b) サービス名、IP アドレスを入力し、サービスとデータを交換するプロトコル (ADNS/ADNS_TCP) を選択します。
3. [アクティブ-アクティブサイト] を選択します。
4. 完全修飾ドメイン名を入力し、DNS プロキシによってレコードがキャッシュされる期間を指定します。
5. GSLB サイトを設定します。各サイトはローカル GSLB サイトで構成する必要があり、各サイトの構成には他のすべてのサイトをリモート GSLB サイトとして含める必要があります。ローカルサイトは 1 つだけ存在でき、その他すべてのサイトはリモートサイトです。
   a) サイト名やサイトの IP アドレスなど、サイトの詳細を入力します。
   b) [リモート] または [ローカル] サイトタイプを選択します。
   c) 必要に応じて、RPC パスワードを変更し、必要に応じてセキュリティで保護します。

d) モニタを GSLB サービスにバインドする場合は、モニタがサービスを監視する条件を選択します。これは、モニタがサービスにバインドされた後にのみ有効になります。考えられる条件は次のとおりです。

- **ALWAYS**。常に GSLB サービスを監視します。
- **MEP** が失敗する。MEP によるメトリックの交換が失敗した場合のみ、GSLB サービスを監視します。
- **MEP** が失敗し、サービス **ID** がダウンします。MEP によるメトリック交換は有効ですが、メトリック交換によって更新されたサービスのステータスは DOWN です。

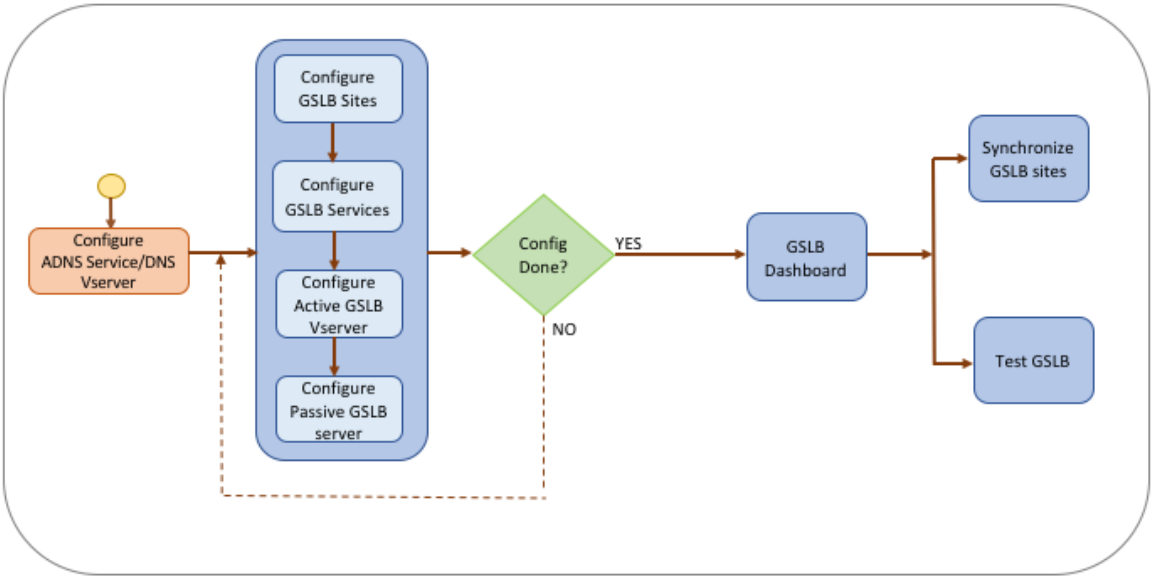
6. GSLB サービスを設定します。アクティブ-アクティブサイトを作成するには、少なくとも 2 つの GSLB サービスを追加する必要があります。
   a) サービス名、サービスタイプ、ポート番号など、サービスの詳細を入力します。
   b) GSLB サービスが属する GSLB サイトを選択して、サービスをサイト（ローカルまたはリモート）に関連付けます。
   c) 必要に応じて、MEP が失敗したときにサービスにバインドする必要があるモニタを選択します。サービスは、既存のサーバーか、新しいサーバーまたは仮想サーバーを作成することができます。
   d) 既存のサーバを関連付けるには、サーバ名を選択します。サービスの IP アドレスが自動的に入力されます。
      - パブリック IP アドレスが NAT 環境で発生する可能性があるサーバ IP と異なる場合は、パブリック IP アドレスとパブリックポートのポート番号を入力します。
      - 新しいサーバを関連付けるには、サーバの IP 詳細、そのパブリック IP アドレス、およびパブリックポート番号を入力して、サーバを作成します。
      - 仮想サーバを関連付けるには、既存の仮想サーバを選択するか、[+] をクリックして新しい仮想サーバを追加します。この vserver は、この GSLB サービスが関連付けられるロードバランシング vserver です。
7. GSLB 仮想サーバーを設定します。
   a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
   b)「サービスの選択」ボックスで「**>** 」をクリックし、GSLB 仮想サーバーにバインドする GSLB サービスを選択します。
   c)「ドメインバインディング」ボックスで「**>** 」をクリックして、この GSLB 仮想サーバーにバインドするドメインを選択します。
   d) 最もパフォーマンスに優れた GSLB サービスを選択するための GSLB 方式を選択します。デフォルトでは、GSLB 方式、バックアップ方式、および動的重み付けのデフォルト値が自動入力されます。必要に応じて変更できます。
      - [**Algorithm based**] 方式を選択した場合は、プライマリ方式とバックアップ方式を選択し、動的重み付けオプションも指定します。
      - [静的近接] 方法を選択した場合は、バックアップ方法と動的ウェイト方法を選択します。また、[>] アイコンをクリックしてデータベースファイルの場所を指定するか、[ロケーションデータベースの選択] ボックスで [**+**] をクリックして新しい場所を追加します。
      - 動的近接（**RTT**）方式を選択した場合は、バックアップ方式を選択し、動的重み付けオプションと最もパフォーマンスの高いサービスを選択するラウンドトリップ時間値を指定します。

8. 設定が完了したら、[完了] をクリックします。GSLB ダッシュボードが表示されます。
9. GSLB サイト構成を変更した場合は、ダッシュボードの [**GSLB** の自動同期] をクリックして、GSLB 設定内の他のサイトに設定を同期します。
    - 同期の前に、ローカルサイトの構成にリモートサイトに関する情報が含まれていることを確認してください。また、同期を成功させるには、他の Citrix ADC アプライアンスでローカルサイトを構成する必要があります。
    - リアルタイム同期が有効になっている場合は、[自動同期] **[GSLB]** をクリックする必要はありません。同期は自動的に行われます。リアルタイム同期を有効にするには、次の手順を実行します。
        a) **Traffic Management** > **GSLB** > **Dashboard** に移動して **Change GSLB Settings** をクリックします。
        b) [自動構成同期] チェックボックスをオンにします。
10. [**GSLB** セットアップのテスト] をクリックして、ADNS サービスまたは DNS サーバーが GSLB セットアップで構成されているドメイン名の正しい IP アドレスで応答していることを確認します。

アクティブ-パッシブサイトの構成

October 7, 2021

次の図は、アクティブ/パッシブサイト構成に関連するワークフローを示しています。



アクティブ/パッシブサイトの構成を開始する前に、各サーバーファームまたはデータセンターに対して標準の負荷分散設定を構成していることを確認してください。

また、デプロイメント内の GSLB サイト間で GSLB 設定を同期するには、次のことを確認してください。

- ローカル GLSB サイトは、GSLB 設定内のすべてのアプライアンスで設定されます。

- 構成内のすべての GSLB サイトで管理アクセスを有効にしました。
- 自動同期および MEP 接続を受け入れるようにファイアウォールを設定しました。
- マスターおよびスレーブの Citrix ADC アプライアンスでは、同じバージョンの Citrix ADC ソフトウェアが実行されています。
- サイトとして参加するすべての Citrix ADC アプライアンスは、同じ Citrix ADC ソフトウェアバージョンを持つ必要があります（これらのサイトはマスタースレーブ関係ではありません）。
- RPC ノードのパスワードは、GSLB 設定内のすべての GSLB サイトで同じです。

ウィザードを使用してアクティブ/パッシブサイトを構成するには

[構成] タブで、次の操作を行います。

1. [トラフィック管理] > [**GSLB**] に移動し、[は じめに] をクリックします。
2. サイトの ADNS サービスまたは DNS 仮想サーバーを構成していない場合は、ここで構成できます。
   a) [表示] をクリックし、[追加] をクリックします。
   b) サービス名、IP アドレスを入力し、サービスとデータを交換するプロトコル (ADNS/ADNS_TCP) を選択します。
3. [アクティブ**-**パッシブサイト] を選択します。
4. 完全修飾ドメイン名を入力し、DNS プロキシによってレコードがキャッシュされる期間を指定します。
5. GSLB サイトを設定します。各サイトはローカル GSLB サイトで構成する必要があり、各サイトの構成には他のすべてのサイトをリモート GSLB サイトとして含める必要があります。ローカルサイトは 1 つだけ存在でき、その他すべてのサイトはリモートサイトです。
   a) サイト名やサイトの IP アドレスなど、サイトの詳細を入力します。
   b) [リモート] または [ローカル] サイトタイプを選択します。
   c) 必要に応じて、RPC パスワードを変更し、必要に応じてセキュリティで保護します。
   d) モニタを GSLB サービスにバインドする場合は、モニタがサービスを監視する条件を選択します。これは、モニタがサービスにバインドされた後にのみ有効になります。考えられる条件は次のとおりです。
      - **ALWAYS**。常に GSLB サービスを監視します。
      - **MEP** が失敗する。MEP によるメトリックの交換が失敗した場合のみ、GSLB サービスを監視します。
      - **MEP** が失敗し、サービス **ID** がダウンします。MEP によるメトリック交換は有効ですが、メトリック交換によって更新されたサービスのステータスは DOWN です。
6. GSLB サービスを設定します。
   a) サービス名、サービスタイプ、ポート番号など、サービスの詳細を入力します。
   b) GSLB サービスが属する GSLB サイトを選択して、サービスをサイト（ローカルまたはリモート）に関連付けます。
   c) 必要に応じて、MEP が失敗したときにサービスにバインドする必要があるモニタを選択します。サービスは、既存のサーバーか、新しいサーバーまたは仮想サーバーを作成することができます。
   d) 既存のサーバを関連付けるには、サーバ名を選択します。サービスの IP アドレスが自動的に入力されます。

- パブリック IP アドレスが NAT 環境で発生する可能性があるサーバ IP と異なる場合は、パブリック IP アドレスとパブリックポートのポート番号を入力します。
- 新しいサーバを関連付けるには、サーバの IP 詳細、そのパブリック IP アドレス、およびパブリックポート番号を入力して、サーバを作成します。
- 仮想サーバを関連付けるには、既存の仮想サーバを選択するか、[**+**] をクリックして新しい仮想サーバを追加します。この vserver は、この GSLB サービスが関連付けられるロードバランシング vserver です。

7. GSLB バックアップ仮想サーバーを設定します。GSLB バックアップ仮想サーバは、プライマリ GSLB 仮想サーバにアクセスできないか、何らかの理由で DOWN とマークされている場合にのみ動作可能になります。
   a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
   b)「サービスバインディング」で「**>**」をクリックし、GSLB 仮想サーバーにバインドする必要がある GSLB サービスを選択します。
   c) 最もパフォーマンスに優れた GSLB サービスを選択するための GSLB 方式を選択します。デフォルトでは、GSLB 方式、バックアップ方式、および動的重み付けのデフォルト値が自動入力されます。必要に応じて変更できます。
      - [アルゴリズムベースの方法] を選択した場合は、プライマリ方法とバックアップ方法を選択します。
      - [静的近接] 方法を選択した場合は、バックアップ方法を選択し、データベースファイルの場所を指定します。
      - **Dynamic Proximity**（**RTT**）方式を選択した場合は、バックアップ方式を選択し、最もパフォーマンスの高いサービスを選択するためのサービス重みと RTT 値を指定します。
8. GSLB 仮想サーバーを設定します。
   a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
   b)「サービスの選択」ボックスで「**>**」をクリックし、GSLB 仮想サーバーにバインドする GSLB サービスを選択します。
   c)「ドメインバインディング」ボックスで「**>**」をクリックして、この GSLB 仮想サーバーにバインドするドメインを選択します。
   d) 最もパフォーマンスに優れた GSLB サービスを選択するための GSLB 方式を選択します。デフォルトでは、GSLB 方式、バックアップ方式、および動的重み付けのデフォルト値が自動入力されます。必要に応じて変更できます。
      - [**Algorithm based**] 方式を選択した場合は、プライマリ方式とバックアップ方式を選択し、動的重み付けオプションも指定します。
      - [静的近接] 方法を選択した場合は、バックアップ方法と動的ウェイト方法を選択します。また、**>** アイコンをクリックしてデータベースファイルの場所を指定するか、[場所データベースの選択] ボックスで **+** をクリックして新しい場所を追加します。
      - 動的近接（**RTT**）方式を選択した場合は、バックアップ方式を選択し、動的重み付けオプションと最もパフォーマンスの高いサービスを選択するラウンドトリップ時間値を指定します。
9. 設定が完了したら、[完了] をクリックします。GSLB ダッシュボードが表示されます。
10. GSLB サイト構成を変更した場合は、ダッシュボードの [**GSLB** の自動同期] をクリックして、GSLB 設定内

の他のサイトに設定を同期します。

- 同期の前に、ローカルサイトの構成にリモートサイトに関する情報が含まれていることを確認してください。また、同期を成功させるには、他の Citrix ADC アプライアンスでローカルサイトを構成する必要があります。
- リアルタイム同期が有効になっている場合は、[自動同期] **[GSLB]** をクリックする必要はありません。同期は自動的に行われます。リアルタイム同期を有効にするには、次の手順を実行します。
  a) **Traffic Management** > **GSLB** > **Dashboard** に移動して **Change GSLB Settings** をクリックします。
  b) [自動構成同期] チェックボックスをオンにします。

11. [**GSLB** セットアップのテスト] をクリックして、ADNS サービスまたは DNS サーバーが GSLB セットアップで構成されているドメイン名の正しい IP アドレスで応答していることを確認します。

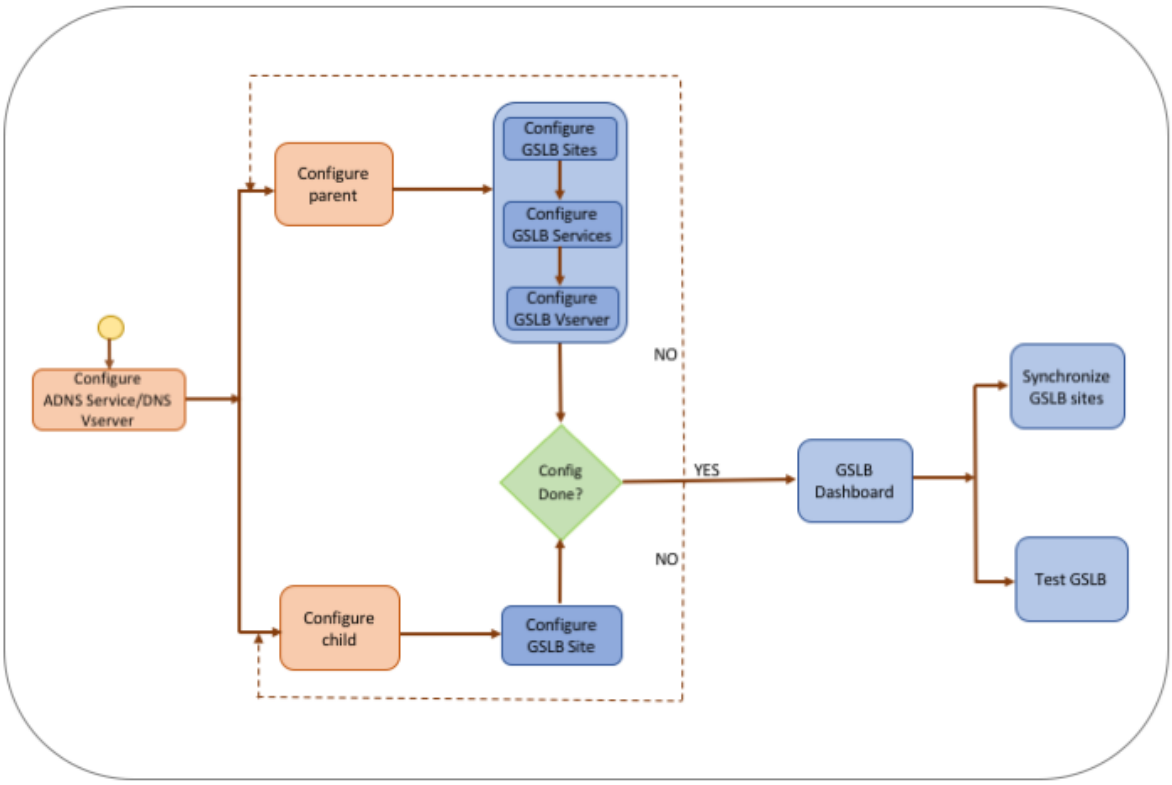
> 注
>
> ディザスタリカバリ用のアクティブ/パッシブ GSLB セットアップの GSLB エンティティの構成の詳細については、「ディザスタリカバリ用の GSLB の設定」を参照してください。

親子トポロジの設定

October 7, 2021

親子トポロジでは、最上位レベルには親サイトがあり、他の親とのピア関係があります。各親サイトは複数の子サイトを持つことができ、各親サイトはその子サイトや他の親サイトと正常性情報を交換します。ただし、子サイトは親サイトとのみ通信します。

次の図は、GSLB の親子トポロジ設定に関連するワークフローを示しています。



親子トポロジの展開の構成を開始する前に、各サーバーファームまたはデータセンターの標準的な負荷分散セットアップを構成していることを確認してください。

また、デプロイメント内の GSLB サイト間で GSLB 設定を同期するには、次のことを確認してください。

- ローカル GLSB サイトは、GSLB 設定内のすべてのアプライアンスで設定されます。
- 構成内のすべての GSLB サイトで管理アクセスを有効にしました。
- 自動同期および MEP 接続を受け入れるようにファイアウォールを設定しました。
- サイトとして参加するすべての Citrix ADC アプライアンスは、同じ Citrix ADC ソフトウェアバージョンを持つ必要があります（これらのサイトはマスタースレーブ関係ではありません）。
- RPC ノードのパスワードは、GSLB 設定内のすべての GSLB サイトで同じです。

ウィザードを使用して親子展開を構成するには

[構成] タブで、次の操作を行います。

1. [トラフィック管理] > [**GSLB**] に移動し、[は じめに] をクリックします。
2. サイトの ADNS サーバーまたは DNS 仮想サーバーを構成していない場合は、ここで構成できます。
   a) [表示] をクリックし、[追加] をクリックします。
   b) サービス名、IP アドレスを入力し、サービスとデータを交換するプロトコル (ADNS/ADNS_TCP) を選択します。
3. [親子トポロジ] を選択します。

4. [サイトタイプの選択] フィールドで、[] を選択します。
    - 親 — 親サイトを構成する場合、関連する子サイトを構成し、GSLB セットアップで他の親サイトも構成する必要があります。
    - 子 — サイトを構成する場合、子サイトとその親サイトのみを構成する必要があります。

親サイトを構成するには

1. 完全修飾ドメイン名を入力し、DNS プロキシによってレコードがキャッシュされる期間を指定します。
2. GSLB サイトを設定します。各サイトはローカル GSLB サイトで構成する必要があり、各サイトの構成には他のすべてのサイトをリモート GSLB サイトとして含める必要があります。ローカルサイトは 1 つだけ存在できます。その他のサイトはすべてリモートサイトです。指定したサイトの IP アドレスがアプライアンスによって所有されている場合（たとえば、MIP アドレスや SNIP アドレス）、サイトはローカルサイトです。それ以外の場合は、リモートサイトになります。
3. サイト名やサイトの IP アドレスなど、サイトの詳細を入力します。
    a) サイトタイプを選択します。
    b) 必要に応じて、RPC パスワードを変更し、必要に応じてセキュリティで保護します。
    c) モニタを GSLB サービスにバインドする場合は、モニタがサービスを監視する条件を選択します。これは、モニタがサービスにバインドされた後にのみ有効になります。考えられる条件は次のとおりです。
        - **Always**。常に GSLB サービスを監視します。
        - **MEP** が失敗する。MEP によるメトリックの交換が失敗した場合のみ、GSLB サービスを監視します。
        - **MEP** が失敗し、サービスが **DOWN** です。MEP によるメトリック交換は有効ですが、メトリック交換によって更新されたサービスのステータスは DOWN です。
4. GSLB サービスを設定します。
    a) サービス名、サービスタイプ、ポート番号などのサービスの詳細を入力します。
    b) GSLB サービスが属する GSLB サイトを選択して、サービスをサイト（ローカルまたはリモート）に関連付けます。
    c) 必要に応じて、MEP が失敗したときにサービスにバインドする必要があるモニタを選択します。サービスは、既存のサーバーか、新しいサーバーまたは仮想サーバーを作成することができます。
        - 既存のサーバを関連付けるには、サーバ名を選択します。サービス IP アドレスは自動入力されます。
        - 新しいサーバを関連付けるには、サーバの IP 詳細、そのパブリック IP アドレス、およびパブリックポート番号を入力して、サーバを作成します。
        - 仮想サーバを関連付けるには、既存の仮想サーバを選択するか、[**+**] をクリックして新しい仮想サーバを追加します。この vserver は、この GSLB サービスが関連付けられる負荷分散 vserver です。パブリック IP アドレスが NAT 環境で発生する可能性があるサーバ IP と異なる場合は、パブリック IP アドレスとパブリックポート番号を入力します。
5. GSLB 仮想サーバーを設定します。
    a) GSLB 仮想サーバー名の名前を入力し、DNS レコードタイプを選択します。
    b)「サービスの選択」ボックスで「**>** 」をクリックし、GSLB 仮想サーバーにバインドする GSLB サービス

を選択します。

c)「ドメインバインディング」ボックスで「**>** 」をクリックして、GSLB 仮想サーバーにバインドされているドメイン名を表示します。

d) 最もパフォーマンスに優れた GSLB サービスを選択するための GSLB 方式を選択します。GSLB 方式、バックアップ方式、および動的加重のデフォルト値は、デフォルトで自動的に設定されます。必要に応じて変更できます。

- [**Algorithm based**] 方式を選択した場合は、プライマリ方式とバックアップ方式を選択し、動的重み付けオプションも指定します。
- [静的近接] 方法を選択した場合は、バックアップ方法と動的ウェイト方法を選択します。また、**>** アイコンをクリックしてデータベースファイルの場所を指定するか、[場所データベースの選択] ボックスで **+** をクリックして新しい場所を追加します。
- **Dynamic Proximity**（**RTT**）方式を選択した場合は、バックアップ方式を選択し、最もパフォーマンスの高いサービスを選択するためのサービス重みと RTT 値を指定します。

6. 設定が完了したら、[完了] をクリックします。GSLB ダッシュボードが表示されます。
7. GSLB 親サイト構成を変更した場合は、[**GSLB** の自動同期] をクリックして、構成を GSLB セットアップ内の他の親サイトに同期します。親子トポロジでは、子サイトの同期はスキップされます。
   - 同期の前に、ローカルサイトの構成にリモートサイトに関する情報が含まれていることを確認してください。
   - リアルタイム同期が有効になっている場合は、[自動同期**] [GSLB**] をクリックする必要はありません。同期は自動的に行われます。リアルタイム同期を有効にするには、次の手順を実行します。
     a) **Traffic Management** > **GSLB** > **Dashboard** に移動して **Change GSLB Settings** をクリックします。
     b) [自動構成同期] チェックボックスをオンにします。
8. [**GSLB** セットアップのテスト] をクリックして、ADNS サービスまたは DNS サーバーが GSLB セットアップで構成されているドメイン名の正しい IP アドレスで応答していることを確認します。

子サイトを構成するには

1. GSLB サイトを設定します。
   a) サイト名やサイトの IP アドレスなど、サイトの詳細を入力します。
   b) サイトタイプを選択します。
   c) 必要に応じて、RPC パスワードを変更し、必要に応じてセキュリティで保護します。4. モニタが GSLB サービスにバインドされている場合は、モニタがサービスを監視する条件を選択します。考えられる条件は次のとおりです。
      - **Always**。常に GSLB サービスを監視します。
      - **MEP** が失敗する。MEP によるメトリックの交換が失敗した場合のみ、GSLB サービスを監視します。
      - **MEP** が失敗し、サービスが **DOWN** です。MEP によるメトリック交換は有効ですが、メトリック交換によって更新されたサービスのステータスは DOWN です。
2. 設定が完了したら、[完了] をクリックします。GSLB ダッシュボードが表示されます。

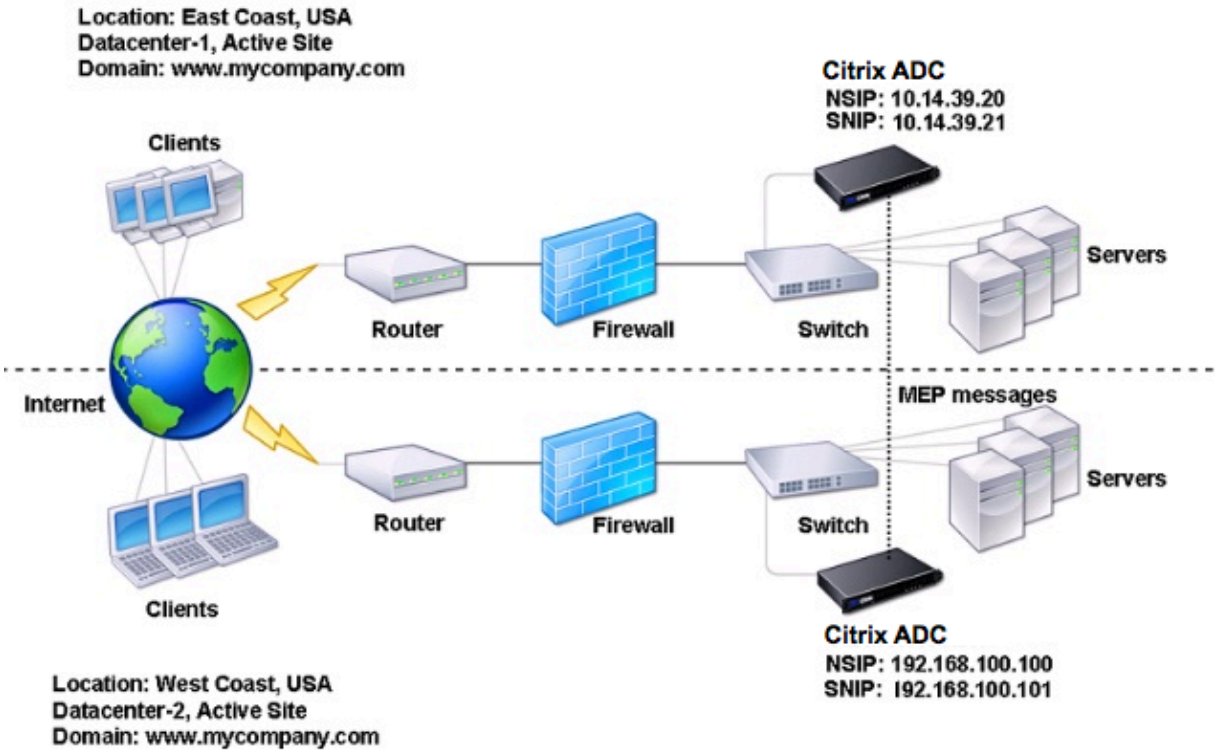
3. [**GSLB** セットアップのテスト] をクリックして、ADNS サービスまたは DNS サーバーが GSLB セットアップで構成されているドメイン名の正しい IP アドレスで応答していることを確認します。

GSLB エンティティを個別に設定する

October 7, 2021

グローバルサーバー負荷分散は、理想的には異なる地理的な場所にある 2 つの別々のサーバーファームでホストされている Web サイトへのトラフィックフローを管理するために使用されます。たとえば、地理的に分離された 2 つのサーバーファームまたはデータセンターでホストされている Web サイト www.mycompany.com を考えてみましょう。どちらのサーバーファームも Citrix ADC アプライアンスを使用します。これらのサーバーファーム内の Citrix ADC アプライアンスは、ワンアームモードでセットアップされ、www.mycompany.com ドメインの権限のある DNS サーバーとして機能します。次の図は、この構成を示しています。

図 1: 基本的な GSLB トポロジ



このような GSLB セットアップを設定するには、まず、各サーバーファームまたはデータセンターの標準ロードバランシング設定を構成する必要があります。これにより、各サーバーファーム内の異なるサーバー間で負荷を分散できます。次に、両方の Citrix ADC アプライアンスを権限のある DNS（ADNS）サーバーとして構成します。次に、各サーバーファームの GSLB サイトを作成し、各サイトの GSLB 仮想サーバーを構成し、GLSB サービスを作成し、GSLB サービスを GSLB 仮想サーバーにバインドします。最後に、ドメインを GSLB 仮想サーバーにバインドしま

す。2 つの異なるサイトにある 2 つのアプライアンスの GSLB 構成は同じですが、各サイトのロードバランシング構成はそのサイトに固有です。

注：Citrix ADC クラスタ設定で GSLB サイトを構成するには、クラスタでの GSLB の設定を参照してください。

標準ロードバランシング設定の設定

負荷分散仮想サーバは、データセンター内の異なる物理サーバ間で負荷を分散します。これらのサーバーは Citrix ADC アプライアンス上でサービスとして表され、サービスは負荷分散仮想サーバーにバインドされます。

基本的な負荷分散設定の設定の詳細については、負荷分散を参照してください。

権限のある **DNS** サービスを構成する

October 7, 2021

Citrix ADC アプライアンスを権限のある DNS サーバーとして構成すると、クライアントからの DNS 要求を受け入れ、クライアントが要求を送信するデータセンターの IP アドレスで応答します。

注：Citrix ADC アプライアンスの権限を持つためには、SOA レコードと NS レコードも作成する必要があります。SOA および NS レコードの詳細については、「ドメインネームシステム」を参照してください。

コマンドラインインターフェイスを使用して **ADNS** サービスを作成するには

コマンドプロンプトで、次のコマンドを入力して ADNS サービスを作成し、構成を確認します。

```
add service <name> <IP>@ ADNS <port>

show service <name>
<!--NeedCopy-->
```

例:

```
add service Service-ADNS-1 10.14.39.21 ADNS 53

show service Service-ADNS-1
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **ADNS** サービスを変更するには

コマンドプロンプトで、次のコマンドを入力します。

```
set service <name> <IPAddress> ADNS <port>
<!--NeedCopy-->
```

例:

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **ADNS** サービスを削除するには

コマンドプロンプトで、次のコマンドを入力します。

```
rm service <name>
<!--NeedCopy-->
```

例:

```
rm service Service-ADNS-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **ADNS** サービスを構成するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 新しい ADNS サービスを追加するか、既存のサービスを選択してその設定を編集します。

基本的な **GSLB** サイトの構成

October 7, 2021

GSLB サイトは、ネットワーク内のデータセンターを表し、GSLB 仮想サーバー、サービス、およびその他のネットワークエンティティの論理的なグループです。通常、GSLB セットアップでは、クライアントに同じコンテンツを提

供するために装備されている多くの GSLB サイトがあります。これらは、通常、1 つのサイトが完全にダウンした場合でも、ドメインがアクティブになるように、地理的に分離されています。GSLB 構成のすべてのサイトは、GSLB サイトをホストするすべての Citrix ADC アプライアンスで構成する必要があります。つまり、各サイトで、ローカル GSLB サイトと各リモート GSLB サイトを構成します。

ドメインに対して GSLB サイトが作成されると、Citrix ADC アプライアンスは、構成された GSLB アルゴリズムによって決定された適切な GSLB サイトにクライアント要求を送信します。

コマンドラインインターフェイスを使用して **GSLB** サイトを作成するには

コマンドプロンプトで次のコマンドを入力して GSLB サイトを作成し、構成を確認します。

```
add gslb site <siteName> <siteIPAddress>
show gslb site <siteName>
<!--NeedCopy-->
```

例**:**

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show glsb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サイトを変更または削除するには

- GSLB サイトを変更するには、set gslb サイトコマンドを使用します。これは、既存の GSLB サイトの名前を入力することを除いて、add gslb サイトコマンドを使用する場合と同じです。
- サイトパラメータの設定を解除するには、unset gslb site コマンドを使用し、続いて siteName 値と、デフォルト値にリセットするパラメータの名前を指定します。
- GSLB サイトを削除するには、rm gslb site コマンドを使用します。このコマンドは、<name> 引数だけを受け入れます。

構成ユーティリティを使用して基本的な **GSLB** サイトを構成するには

1. [トラフィック管理] > [**GSLB**] > [サイト] に移動します。
2. 新しい GSLB サイトを追加するか、既存の GSLB サイトを選択してその設定を編集します。

コマンドラインインターフェイスを使用して **GSLB** サイトの統計情報を表示するには

コマンドプロンプトで入力します。


```
stat gslb site <siteName>
<!--NeedCopy-->
```

例:

```
stat gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** サイトの統計情報を表示するには

1. [トラフィック管理] > [**GSLB**] > [サイト] に移動します。
2. GSLB サイトを選択し、[統計] をクリックします。

GSLB サービスの設定

October 7, 2021

GSLB サービスは、負荷分散またはコンテンツスイッチング仮想サーバーの表現です。ローカル GSLB サービスは、ローカル負荷分散またはコンテンツスイッチング仮想サーバーを表します。リモート GSLB サービスは、GSLB セットアップ内の他のサイトの 1 つで設定された負荷分散またはコンテンツスイッチング仮想サーバーを表します。GSLB セットアップの各サイトでは、1 つのローカル GSLB サービスと任意の数のリモート GSLB サービスを作成できます。

> 重要
>
> 負荷分散仮想サーバーが GSLB ノード自体にあるか、子ノード (親子展開) にあり、モニターが GSLB サービスにバインドされていない場合は、
>
> 次のことを確認します。GLSB サービスの IP アドレス、ポート番号、およびプロトコルが仮想サーバーです。それ以外の場合は、サービスの状態は DOWN としてマークされます。

コマンドラインインターフェイスを使用して **GSLB** サービスを作成するには

コマンドプロンプトで次のコマンドを入力して、GSLB サービスを作成し、構成を確認します。

```
add gslb service <serviceName> <serverName | IP> <serviceType> <port>-
    siteName <string>
```



```
show gslb service <serviceName>
<!--NeedCopy-->
```

例:

```
add gslb service Service-GSLB-1 10.14.39.14 HTTP 80  -siteName Site-
    GSLB-East-Coast
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サービスを変更または削除するには

- GSLB サービスを変更するには、set gslb service<serviceName> コマンドを使用します。このコマンドでは、設定を変更する GSLB サービスの名前を指定します。パラメーターの既存の値は、ユーザーが指定した値または既定で設定した値に変更できます。同じコマンドで複数のパラメータの値を変更できます。パラメータの詳細については、add gslb service コマンドを参照してください。例

```
> set gslb service SKP_GSLB_NOTCNAME_SVC2 -maxBandWidth 25 -
    maxClient 8
Done
> sh gslb service SKP_GSLB_NOTCNAME_SVC2
SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
...
Max Conn: 8  Max Bandwidth: 25 kbits
<!--NeedCopy-->
```

- パラメータをデフォルト値にリセットするには、unset gslb service<serviceName> コマンドと設定解除するパラメータを使用します。例

```
> unset gslb service SKP_GSLB_NOTCNAME_SVC2 maxBandWidth
Done
> sh gslb service SKP_GSLB_NOTCNAME_SVC2
SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
...
Max Conn: 8  Max Bandwidth: 0 kbits
<!--NeedCopy-->
```

- GSLB サービスを削除するには、rm gslb サービス<serviceName> コマンドを使用します。

構成ユーティリティを使用して **GSLB** サービスを作成するには

1. **Traffic Management** > **GSLB** > **Services** に移動します。
2. 新しい GSLB サービスを追加するか、既存のサービスを選択してその設定を編集します。

コマンドラインインターフェイスを使用して **GSLB** サービスの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat gslb service <serviceName>
<!--NeedCopy-->
```

例**:**

```
stat gslb service Service-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** サービスの統計情報を表示するには

1. **Traffic Management** > **GSLB** > **Services** に移動します。
2. GSLB サービスを選択し、[統計] をクリックします。

GSLB サービスグループを構成する

February 21, 2022

サービスグループを使用すると、サービスグループを 1 つのサービスと同じくらい簡単に管理できます。たとえば、サービスグループに対して圧縮、ヘルスモニタリング、グレースフルシャットダウンなどのオプションを有効または無効にすると、そのサービスグループのすべてのメンバーに対してオプションが有効または無効になります。

サービスグループを作成したら、仮想サーバにバインドし、そのグループにサービスを追加できます。また、モニタをサービスグループにバインドすることもできます。

> 重要
>
> :
>
> 負荷分散仮想サーバーが GSLB ノード自体または子ノード (親子展開) にあり、モニターが GSLB サービスにバインドされていない場合は、次のことを確認してください。GLSB サービスグループの IP アドレス、ポート番号、およびプロトコルが仮想サービスが表しているサーバ。それ以外の場合、サービス状態は DOWN とマーク

> されます。

Citrix ADC は、次の種類の GSLB サービスグループをサポートしています。

- IP アドレスベースのサービスグループ
- ドメイン名ベースのサービスグループ
- ドメイン名ベースの Autoscale サービスグループ

GSLB ドメイン名ベースの **Autoscale** サービスグループ

Citrix ADC ハイブリッドおよびマルチクラウドのグローバルサーバー負荷分散（GSLB）ソリューションを使用すると、ハイブリッドクラウド、複数のクラウド、オンプレミスの複数のデータセンターにアプリケーショントラフィックを分散できます。Citrix ADC GSLB ソリューションは、Citrix ADC ロードバランサー、AWS の弾性負荷分散(ELB)、その他のサードパーティ製ロードバランサーなど、さまざまな負荷分散ソリューションをサポートします。また、GSLB ソリューションでは、GSLB レイヤと負荷分散レイヤが個別に管理されている場合でも、グローバル負荷分散が実行されます。

クラウド展開では、管理目的で負荷分散ソリューションにアクセスする際に、参照としてドメイン名がユーザーに与えられます。外部エンティティは、これらのドメイン名が解決する IP アドレスを使用しないことをお勧めします。また、負荷分散層は負荷に応じてスケールアップまたはスケールダウンされるため、IP アドレスが静的であることは保証されません。したがって、IP アドレスではなくドメイン名を使用して負荷分散エンドポイントを参照することをお勧めします。これには、IP アドレスではなくドメイン名を使用して GSLB サービスを参照する必要があり、負荷分散レイヤードメイン名に対して返されたすべての IP アドレスを消費し、GSLB で同じ表現を持つ必要があります。

負荷分散エンドポイントを参照するときに IP アドレスの代わりにドメイン名を使用するには、GSLB のドメイン名ベースのサービスグループを使用できます。

GSLB ドメイン名ベースのサービスグループの監視

Citrix ADC アプライアンスには、TCP ベースのアプリケーションを監視する tcp-default と ping-default の 2 つの内蔵モニターがあります。tcp-default モニタはすべての TCP サービスにバインドされ、ping デフォルトモニタは TCP 以外のすべてのサービスにバインドされます。ビルトインモニターは、デフォルトで GSLB サービスグループにバインドされます。ただし、アプリケーション固有のモニターを GSLB サービスグループにバインドすることを推奨します。

トリガモニタオプションを **MEPDOWN** に設定するための推奨事項

[Trigger Monitor] オプションを使用すると、GSLB サイトで常にモニターを使用する必要があるか、メトリック交換プロトコル (MEP) が DOWN のときにモニターを使用する必要があるかを示すことができます。

[トリガーモニター] オプションは、デフォルトで ALWAYS に設定されています。

Trigger Monitor オプションが ALWAYS に設定されている場合、各 GSLB ノードはモニターを個別にトリガーします。各 GSLB ノードが個別にモニターをトリガーする場合、各 GSLB ノードは異なる GSLB サービスセットで動作

する可能性があります。これにより、これらの GSLB ノードに到達する DNS 要求に対する DNS 応答に不一致が生じる可能性があります。また、各 GSLB ノードが個別に監視している場合は、ロードバランサーエンティティに到達する監視プローブの数が増加します。永続性エントリも GSLB ノード間で互換性がなくなります。

したがって、GSLB サイトエンティティの [トリガーモニター] オプションを MEPDOWN に設定することをお勧めします。[Trigger Monitor] オプションが MEPDOWN に設定されている場合、負荷分散ドメインの解決と監視の所有権はローカル GSLB ノードにあります。[Trigger Monitor] オプションが MEPDOWN に設定されている場合、負荷分散ドメインの解決とそれ以降の監視は、GSLB サービスグループのローカル GSLB ノードによって実行されます。その結果は、メトリック交換プロトコル (MEP) を使用して GSLB に参加している他のすべてのノードに伝播されます。

また、負荷分散ドメインに関連付けられている IP アドレスのセットが更新されるたびに、MEP を通じて通知されます。

GSLB サービスグループの制限事項

- 負荷分散ドメインの場合、DNS 応答で返される IP アドレスは一般にパブリック IP アドレスです。負荷分散ドメインが解決されると、プライベート IP アドレスを動的に適用することはできません。したがって、GSLB ドメイン名ベースの Autoscale サービスグループの IP ポートバインディングのパブリック IP ポートとプライベート IP ポートは同じです。これらのパラメータは、ドメイン名ベースの Autoscale サービスグループに対して明示的に設定することはできません。
- GSLB サービスグループでは、サイト永続性、DNS ビュー、およびクラスタリングはサポートされていません。

CLI を使用して **GSLB** サービスグループを設定および管理する

GSLB サービスグループを追加するには、次の手順を実行します。

```
1 add gslb serviceGroup <serviceGroupName>@ <serviceType> [-autoScale (
     DISABLED | DNS )] -siteName <string>
2 <!--NeedCopy-->
```

例:

```
1 add gslb serviceGroup Service-Group-1 http -siteName Site1 -autoScale
     DNS
2 <!--NeedCopy-->
```

GSLB サービスグループを仮想サーバーにバインドするには


```
bind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>) | <serverName
    >@ | (-monitorName <string>@))
<!--NeedCopy-->
```

例:

```
bind gslb serviceGroup Service-Group-1 203.0.113.2
bind gslb serviceGroup Service-Group-1 S1 80
bind gslb serviceGroup Service-Group-1 -monitorName Mon1
<!--NeedCopy-->
```

GSLB サービスグループを仮想サーバにバインド解除するには

```
unbind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>) | <
    serverName>@ | -monitorName <string>@)
<!--NeedCopy-->
```

例:

```
unbind gslb serviceGroup Service-Group-1 -monitorName Mon1
<!--NeedCopy-->
```

GSLB サービスグループのパラメータを設定するには、次の手順を実行します。

```
set gslb serviceGroup <serviceGroupName>@ [(<serverName>@ <port> [-
    weight <positive_integer>] [-hashId <positive_integer>] [-publicIP <
    ip_addr|ipv6_addr|*>] [-publicPort <port>]) | -maxClient <
    positive_integer> | -cip ( ENABLED | DISABLED ) | <cipHeader> | -
    cltTimeout <secs> | -svrTimeout <secs> | -maxBandwidth <
    positive_integer> | -monThreshold <positive_integer> | -
    downStateFlush ( ENABLED | DISABLED )] [-monitorName <string> -
    weight <positive_integer>] [-healthMonitor ( YES | NO )] [-comment <
    string>] [-appflowLog ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

GSLB サービスグループからパラメータの設定を解除するには、次の手順を実行します。


```
unset gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port> [-
    weight] [-hashId] [-publicIP] [-publicPort]] [-maxClient] [-cip] [-
    cltTimeout] [-svrTimeout] [-maxBandwidth] [-monThreshold] [-
    appflowLog] [-monitorName] [-weight] [-healthMonitor] [-cipHeader]
    [-downStateFlush] [-comment]
<!--NeedCopy-->
```

GSLB サービスグループを有効にするには、次の手順を実行します。

```
enable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
<!--NeedCopy-->
```

例:

```
enable gslb serviceGroup SG1 S1 80
<!--NeedCopy-->
```

GSLB サービスグループを無効にするには、次の手順を実行します。

```
disable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>] [-
    delay <secs>] [-graceFul ( YES /| NO )]
<!--NeedCopy-->
```

例:

```
disable gslb serviceGroup SRG2 S1 80
<!--NeedCopy-->
```

> 注:
>
> 無効にする必要があるサービスグループは、Autoscale サービスグループではなく DBS サービスグループである必要があります。

GSLB サービスグループを削除するには、次の手順を実行します。

```
rm gslb serviceGroup <serviceGroupName>
<!--NeedCopy-->
```


例:

```
rm gslb serviceGroup Service-Group-1
<!--NeedCopy-->
```

GSLB サービスグループの統計情報を表示するには、次の手順を実行します。

```
stat gslb serviceGroup [<serviceGroupName>]
<!--NeedCopy-->
```

例:

```
stat gslb serviceGroup Service-Group-1
<!--NeedCopy-->
```

GSLB サービスグループのプロパティを表示するには、次の手順を実行します。

```
show gslb serviceGroup [<serviceGroupName>  -includeMembers]
<!--NeedCopy-->
```

例:

```
show gslb serviceGroup SG1
show gslb serviceGroup -includeMembers
<!--NeedCopy-->
```

既存の **GSLB CLI** コマンドへの変更

GSLB サービスグループの導入後、既存の GSLB コマンドに次の変更が加えられました。

- `bind gslb vserver` -bind コマンドにサービスグループ名が追加されます。

  例:


```
bind gslb vserver <name> ((-serviceName <string> [-weight <
    positive_integer>] ) | <serviceGroupName>@ | | (-domainName <
    string> [-TTL <secs>] [-backupIP<ip_addr|ipv6_addr|*>] [-
    cookieDomain <string>] [-cookieTimeout <mins>][-sitedomainTTL
    <secs>]) | (-policyName <string>@ [-priority<positive_integer
    >] [-gotoPriorityExpression <expression>] [-type REQUEST |
    RESPONSE )]))
<!--NeedCopy-->
```

- `unbind gslb vserver` -サービスグループが unbind コマンドに追加されます。

  例:

```
unbind gslb vserver <name> (-serviceName <string> <
    serviceGroupName> @ /(-domainName <string> [-backupIP] [-
    cookieDomain]) | -policyName <string>@)
<!--NeedCopy-->
```

- `show gslb site` -このコマンドを実行すると、GSLB サービスグループも表示されます。
- `show gslb vs` -このコマンドを実行すると、GSLB サービスグループが表示されます。
- `stat gslb vs` -このコマンドを実行すると、GSLB サービスグループの統計情報も表示されます。
- `show lb monitor bindings` -このコマンドを実行すると、GSLB サービスグループバインディングも表示されます。

GUI を使用して GSLB サービスグループを構成する

1. [トラフィック管理] > [**GSLB**] > [サービスグループ] に移動します。
2. サービスグループを作成し、AutoScale モードを DNS に設定します。

GSLB サービスグループのサイト永続性を構成する

IP アドレスベースおよびドメイン名ベースのサービスグループに対して、サイト永続性を構成できます。ドメイン名ベースの Autoscale サービスグループでは、サイトの永続性はサポートされていません。

CLI を使用して HTTP Cookie に基づいてサイト永続性を設定するには

- 接続プロキシの永続性については、サイトプレフィックスを設定する必要はありません。

コマンドプロンプトで入力します。


```
set gslb service group <serviceGroupName> [-sitePersistence <
    sitePersistence>]
<!--NeedCopy-->
```

- HTTP リダイレクトの永続性については、まずサービスグループのメンバのサイトプレフィックスを設定し、次にサービスグループの HttpRedirect パーシステンスパラメータを設定する必要があります。

コマンドプロンプトで入力します。

```
set gslb servicegroup <serviceGroupName>  <serviceGroup member name|Ip>
     <port>  [-sitePrefix <string>]

set gslb servicegroup <serviceGroupName> [-sitePersistence <
    sitePersistence>]
<!--NeedCopy-->
```

例:

- 接続プロキシの永続性

```
set gslbservicegroup sg1 -sitePersistence connectionProxy
<!--NeedCopy-->
```

- httpRedirect パーシスタンス

```
set gslb servicegroup sg2 test1 80 -sitePrefix vserver-GSLB-1
<!--NeedCopy-->
```

```
set gslb servicegroup sg2 -sitePersistence HTTPRedirect
<!--NeedCopy-->
```

GUI を使用して **Cookie** に基づいてサイトの永続性を設定するには

1. [トラフィック管理] > [**GSLB**] > [サービスグループ] に移動し、サイトの永続性を設定するサービスグループ (ServiceGroup-GSLB-1 など) を選択します。
2. [サイトの永続性] セクションをクリックし、要件を満たすパーシステンスを設定します。

> ヒント
>
> GSLB サービスグループの展開シナリオと設定例については、以下のトピックを参照してください。
>
> - ユースケース: ドメイン名ベースの Autoscale サービスグループの展開
> - ユースケース:IP アドレスベースの Autoscale サービスグループの展開

GSLB 仮想サーバーの構成

October 7, 2021

GSLB 仮想サーバーは、1 つ以上の GSLB サービスを表し、それらの間のトラフィックのバランスをとるエンティティです。設定された GSLB メソッドまたはアルゴリズムを評価して、クライアント要求を送信する GSLB サービスを選択します。

> 注
>
> GSLB 仮想サーバープロトコルの要件は、主に仮想サーバーと仮想サーバーにバインドされているサービスの間の関係を作成することです。これにより、CLI/API が他のタイプの仮想サーバに対して一貫性が保たれます。サービスまたは仮想サーバーの Service Type パラメーターは、DNS 要求の処理中には使用されません。代わりに、サイトの永続性、モニタリング、および MEP を介したルックアップの実行中に参照されます。

コマンドラインインターフェイスを使用して GSLB 仮想サーバーを作成するには

コマンドプロンプトで次のコマンドを入力して、GSLB 仮想サーバーを追加し、構成を確認します。

```
1 -  add gslb vserver <name> <serviceType> -ipType (IPv4 | IPv6)
2 -  show gslb vserver <name>
3 <!--NeedCopy-->
```

例:

```
1 add gslb vserver Vserver-GSLB-1 HTTP -ipType IPv4
2 add gslb vserver Vserver-GSLB-2 HTTP -ipType IPv6
3 show gslb vserver Vserver-GSLB-1
4 show gslb vserver Vserver-GSLB-2
5 <!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーを変更または削除するには

- GSLB 仮想サーバーを変更するには、`set gslb vserver` コマンドを使用します。このコマンドは`add gslb vserver`コマンドと似ていますが、既存の GSLB 仮想サーバーの名前を入力する点が異なります。
- パラメータをデフォルト値にリセットするには、`unset gslb vserver` コマンドの後に vserverName 値、および設定解除するパラメータの名前を指定します。
- GSLB 仮想サーバーを削除するには、name 引数だけを受け付ける`rm gslb vserver` コマンドを使用します。

構成ユーティリティを使用して **GSLB** 仮想サーバーを構成するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. 新しい GSLB 仮想サーバーを追加するか、既存の GSLB 仮想サーバーを選択してその設定を編集します。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
stat gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーの統計情報を表示するには

[トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、仮想サーバーを選択して [統計] をクリックします。

GSLB 仮想サーバの統計情報

Citrix ADC バージョン 12.1 ビルド 51.xx 以降から、GSLB 仮想サーバーの統計情報には、vserver ヒット、現在の永続性セッション、要求バイト、応答バイト、スピルオーバーしきい値、スピルオーバーヒット、現在のクライアント確立などの詳細情報も表示されます。接続、および vserver のバックアップヒットがダウンします。

- プライマリ **LB** メソッドの失敗: プライマリ GSLB メソッドが失敗した回数。
- バックアップ **LB** メソッドの失敗: バックアップ GSLB メソッドが失敗した回数。

- **Vserver** パーシステンスヒット：パーシスタンスセッションを通じてリクエストが処理された回数。

GSLB 仮想サーバーの統計情報には、仮想サーバーにバインドされたサービスグループメンバーの統計情報も表示されます。

> 注：
> プライマリ方式が静的な近接方式で、バックアップ方式が RTT の場合、プライマリ方式またはバックアップ方式が失敗する可能性があります。このシナリオでは、LDNS IP に対応するロケーションがない場合、スタティック近接は失敗し、バックアップ方法が試行されます。統計は、以下に基づいて更新されます。
>
> - バックアップ方式が成功した場合、プライマリ方式の障害統計情報だけが増分されます。
> - RTT 計算が成功しなかった場合、バックアップ方法も失敗します。この場合、プライマリ方式とバックアップ方式の障害統計の両方が増加します。
>
> バックアップ方法が失敗した場合は、最後の手段としてラウンドロビンが使用されます。

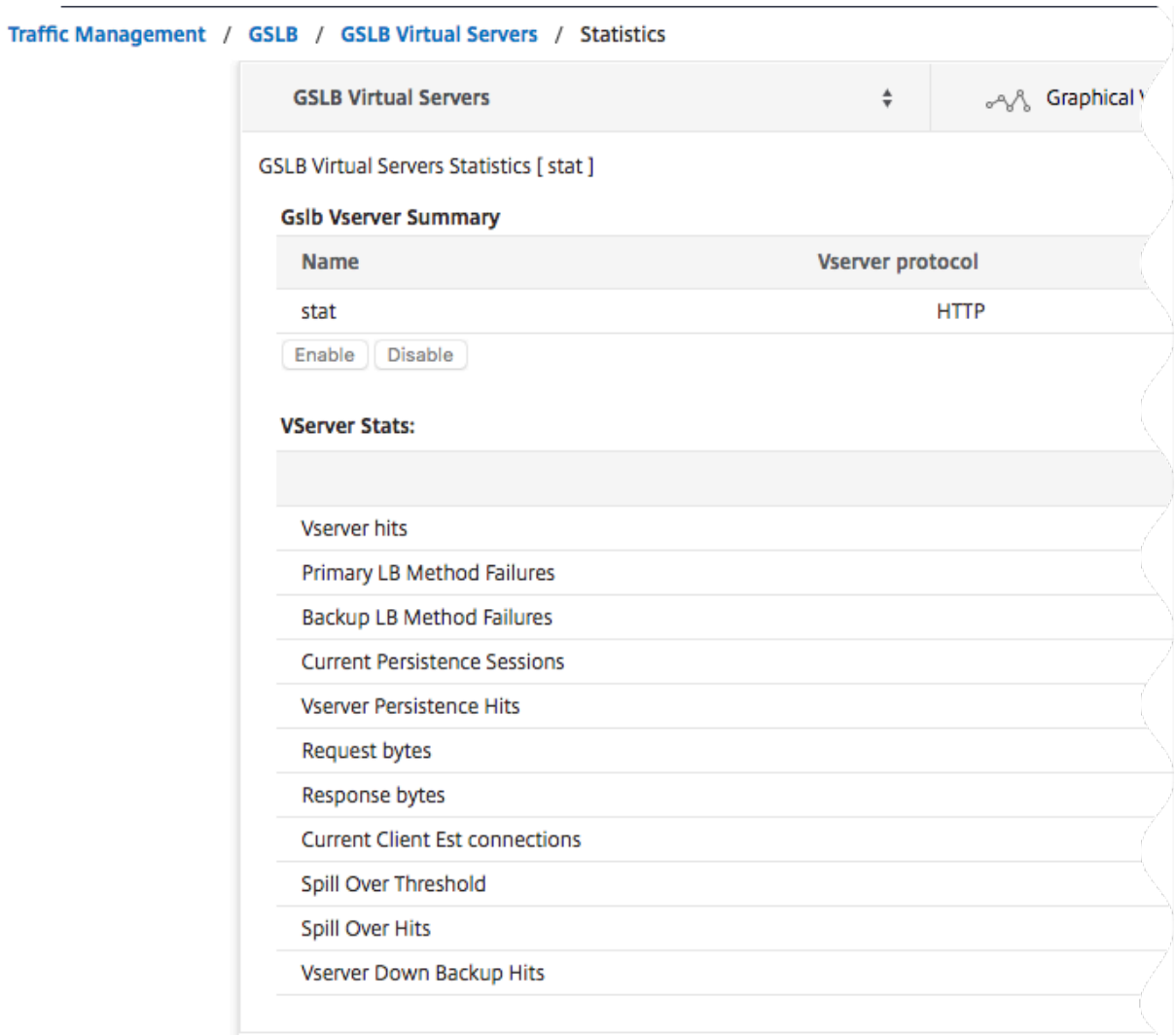
次の図は、CLI からの GSLB 仮想サーバーの統計情報のサンプルです。

```
Gslb Vserver Summary
                Protocol        State   Health  actSvcs inactSvc
gslbvip             HTTP         DOWN        0        0        0

VServer Stats:
                                        Rate (/s)                Total
Vserver hits                                    0                    0
Primary LB Method Failures                     --                    0
Backup LB Method Failures                      --                    0
Current Persistence Sessions                   --                    0
Vserver Persistence Hits                       --                    0
Request bytes                                   0                    0
Response bytes                                  0                    0
Current Client Est connections                 --                    0
Spill Over Threshold                           --                    0
Spill Over Hits                                --                    0
Vserver Down Backup Hits                       --                    0

Note: The above counters are the sum of all bound GSLB services
 Done
```

次の図は、GUI からの GSLB 仮想サーバー統計のサンプルです。



GSLB サービスの統計情報

コマンドラインから stat gslb service コマンドを実行するか、構成ユーティリティから [**Statistics**] リンクをクリックすると、サービスの次の詳細が表示されます。

- 要求バイト数。このサービスまたは仮想サーバで受信した要求バイトの合計数。
- 応答バイト数。このサービスまたは仮想サーバが受信した応答バイト数。
- 現在のクライアントが接続を確立しました。ESTABLISHED 状態のクライアント接続の数。
- サービスの現在の負荷。サービスの負荷（サービスにバインドされた負荷モニターから計算）。

要求と応答の数、および現在のクライアントとサーバーの接続数のデータが表示されないか、対応する負荷分散仮想サーバーのデータと同期されないことがあります。

GSLB 仮想サーバーまたはサービス統計情報のクリア

注：この機能は、NetScaler リリース 10.5.e で使用できます。

GSLB 仮想サーバーおよびサービスの統計情報をクリアできるようになりました。Citrix ADC には、統計情報をクリアするための次の 2 つのオプションがあります。

- 基本: 仮想サーバーに固有の統計をクリアしますが、バインドされた GLSB サービスによって提供された統計は保持します。
- [完全]: 仮想サーバーとバインドされた GSLB サービス統計の両方をクリアします。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーの統計情報をクリアするには

コマンドプロンプトで入力します。

```
stat gslb vserver <name> -clearstats <basic | full>
<!--NeedCopy-->
```

例**:**

```
stat gslb vserver Vserver-GSLB-1  - clearstats basic
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サービスの統計情報をクリアするには

コマンドプロンプトで入力します。

```
stat gslb service <name> -clearstats <basic | full>
<!--NeedCopy-->
```

例**:**

```
stat gslb service service-GSLB-1  - clearstats basic
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーの統計情報を消去するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. GSLB 仮想サーバーを選択し、[統計] をクリックし、[クリア] をクリックします。
3. [クリア] ドロップダウンリストから、[基本] または [完全] を選択し、[**OK**] をクリックします。

構成ユーティリティを使用して **GSLB** サービスの統計情報を消去するには

1. **Traffic Management** > **GSLB** > **Services** に移動します。
2. GSLB サービスを選択し、[統計] をクリックし、[クリア] をクリックします。
3. [クリア] ドロップダウンリストから、[基本] または [完全] を選択し、[**OK**] をクリックします。

GSLB 仮想サーバーの有効化と無効化

GSLB 仮想サーバーを作成すると、デフォルトで有効になります。GSLB 仮想サーバーを無効にした場合、DNS 要求を受信すると、Citrix ADC アプライアンスは構成されている GSLB 方式に基づいて GSLB 決定を行いません。代わりに、DNS クエリへの応答には、仮想サーバーにバインドされているすべてのサービスの IP アドレスが含まれます。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーを有効または無効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
enable gslb vserver <name>@

disable gslb vserver <name>@
<!--NeedCopy-->
```

例**:**

```
enable gslb vserver Vserver-GSLB-1
disable gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** 仮想サーバーを有効または無効にするには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. 仮想サーバを選択し、［アクション］リストから［有効］または［無効］を選択します。

GSLB サービスを **GSLB** 仮想サーバーにバインドする

October 7, 2021

GSLB サービスと仮想サーバーが設定されると、関連する GSLB サービスを GSLB 仮想サーバーにバインドして構成をアクティブにする必要があります。

コマンドラインインターフェイスを使用して **GSLB** サービスを **GSLB** 仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、GSLB サービスを GSLB 仮想サーバーにバインドし、構成を確認します。

```
bind gslb vserver <name> -serviceName <string>

show gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーから **GSLB** サービスをバインド解除するには

コマンドプロンプトで入力します。

```
unbind gslb vserver <name> -serviceName <string>
<!--NeedCopy-->
```

構成ユーティリティーを使用して **GSLB** サービスをバインドするには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、仮想サーバーをダブルクリックします。
2. [**Domains**] セクションをクリックし、ドメインを構成してドメインをバインドします。

ドメインを **GSLB** 仮想サーバーにバインドする

October 7, 2021

Citrix ADC アプライアンスをドメインの権限のある DNS サーバーにするには、ドメインを GSLB 仮想サーバーにバインドする必要があります。ドメインを GSLB 仮想サーバーにバインドすると、Citrix ADC アプライアンスは GSLB 仮想サーバーの名前を含むドメインのアドレスレコードを追加します。GSLB ドメインの権限 (SOA) レコードとネームサーバー (NS) レコードを手動で追加する必要があります。

SOA レコードと NS レコードの設定の詳細については、「ドメインネームシステム」を参照してください。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーにドメインをバインドするには

コマンドプロンプトで次のコマンドを入力して、ドメインを GSLB 仮想サーバーにバインドし、設定を確認します。

```
bind gslb vserver <name> -domainName <string>
show gslb vserver <name>
<!--NeedCopy-->
```

例:

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーから **GSLB** ドメインをバインド解除するには

コマンドプロンプトで入力します。

```
unbind gslb vserver <name> -domainName <string>
<!--NeedCopy-->
```

構成ユーティリティーを使用して **GSLB** 仮想サーバーにドメインをバインドするには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. [GSLB 仮想サーバー] ペインで、ドメインをバインドする GSLB 仮想サーバー (Vserver-GSLB-1 など) を選択し、[開く] をクリックします。
3. [GSLB 仮想サーバーの構成] ダイアログボックスの [ドメイン] タブで、次のいずれかの操作を行います。
   - 新しいドメインを作成するには、[追加] をクリックします。
   - 既存のドメインを変更するには、ドメインを選択し、[開く] をクリックします。
4. [GSLB ドメインの作成] または [GSLB ドメインの設定] ダイアログボックスで、次に示すパラメータの値を指定します。
   - ドメイン名 * — ドメイン名 (例:www.mycompany.com)

* 必須パラメータ

5. ［作成］をクリックします。
6. ［OK］をクリックします。

コマンドラインインターフェイスを使用してドメインの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat gslb domain <name>
<!--NeedCopy-->
```

例:

```
stat gslb domain www.mycompany.com
<!--NeedCopy-->
```

注：特定の GSLB ドメインの統計情報を表示するには、Citrix ADC アプライアンスに追加されたドメインの名前を正確に入力します。ドメイン名を指定しない場合、または不正なドメイン名を指定した場合は、設定されているすべての GSLB ドメインの統計情報が表示されます。

構成ユーティリティを使用してドメインの統計情報を表示するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動します。
2. [GSLB 仮想サーバー] ペインで、GSLB 仮想サーバー (Vserver-GSLB-1 など) を選択し、[開く] をクリックします。
3. [GSLB 仮想サーバーの構成] ダイアログボックスの [ドメイン] タブで、ドメインを選択し、[統計] をクリックします。

コマンドラインを使用して **GSLB** ドメインにバインドされたエンティティの設定の詳細を表示するには

注：この機能は、NetScaler リリース 10.5.e で使用できます。

コマンドプロンプトで入力します。

```
show gslb domain <name>
<!--NeedCopy-->
```


例:

```
show gslb domain gslb1.com
       gslb1.com
       gvs1 - HTTP    state: DOWN
       DNS Record Type: A
       Configured Method: LEASTCONNECTION
       Backup Method: ROUNDROBIN
       Persistence Type: NONE
       Empty Down Response: DISABLED
       Multi IP Response: DISABLED
       Dynamic Weights: DISABLED

       gsvc1 (10.102.239.165: 80)- HTTP State: DOWN    Weight: 1
              Dynamic Weight: 0    Cumulative Weight: 1
              Effective State: DOWN
              Threshold : BELOW

              Monitor Name : http
                     State: DOWN   Weight: 1
                     Probes: 144   Failed [Total: 144 Current: 144]
                     Last response: Failure - TCP syn sent, reset
                        received.
                     Response Time: 2000 millisec

       gsvc2 (10.102.239.179: 80)- HTTP State: DOWN    Weight: 1
              Dynamic Weight: 0    Cumulative Weight: 1
              Effective State: DOWN
              Threshold : BELOW

              Monitor Name : http-ecv
                     State: DOWN   Weight: 1
                     Probes: 141   Failed [Total: 141 Current: 141]
                     Last response: Failure - TCP syn sent, reset
                        received.
                     Response Time: 2000 millisec
Done
<!--NeedCopy-->
```


設定ユーティリティを使用して **GSLB** ドメインにバインドされたエンティティの設定の詳細を表示するには

注：この機能は、NetScaler リリース 10.5.e で使用できます。

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、仮想サーバーをダブルクリックします。
2. [ドメイン] ペインの下のフィールドをクリックします。
3. [**GSLB** 仮想サーバーのドメインバインド] ダイアログボックスで、ドメインを選択し、[バインドの表示] をクリックします。

GSLB のセットアップと設定の例

October 7, 2021

組織は地理的に分散したネットワークを持ち、米国、メキシコ、コロンビアに 3 つのデータセンターがあります。これらのロケーションに関連する設定では、これらはそれぞれ US、MX、CO と呼ばれます。各拠点には、同じコンテンツを提供するサーバーファームがあり、セットアップは期待どおりに機能しています。各ロケーションの Citrix ADC アプライアンスは、ポート 80 で HTTP プロトコルを使用する仮想サーバーを介して構成されます。

組織は、各サイトにサイト識別子を追加することにより、GSLB セットアップを実装しました。サイト識別子には、Citrix ADC アプライアンスが所有し、GSLB 通信に使用されるサイト名と IP アドレスが含まれます。
各サイトには、アプライアンスに対してローカルなサイトがあります。また、各サイトには、ローカルアプライアンスに対してリモートの 2 つのサイトがあります。各サイトで、同じ名前の GSLB 仮想サーバーが作成されます。この仮想サーバーは、組織の Web サイトをグローバルに識別し、IP アドレスが関連付けられていません。
セットアップには、各仮想サーバーの IP アドレス、プロトコル、ポート番号を指定して、各 GSLB サイトで構成された負荷分散仮想サーバーを指す GSLB サービスも設定されています。これらのサービスは GSLB 仮想サーバーにバインドされます。

注: 以下の手順では、コマンドは GSLB サイトのプライベート IP アドレスを使用します。パブリックサイトおよび GSLB サービスの場合は、これらのサイトにパブリック IP アドレスを使用してください。

次の表に、例で使用される IP アドレスと場所を示します。

IP アドレス	位置情報
10.3.1.101	ローカル Citrix ADC サイト IP。
172.16.1.101	リモートロケーションサイト MX のサイト IP。
192.168.1.101	リモート・ロケーションのサイト-CO のサイト IP。
172.16.1.100	リモートロケーションサイト MX のサービス IP。
10.3.1.100	ローカル Citrix ADC サービス IP。

IP アドレス	位置情報
192.168.1.100	リモートロケーションサイト CO のサービス IP。

GSLB サイトを追加するときに、サイトがインターネットを介してのみ通信する場合は、「パブリック IP」フィールドを使用します。たとえば、GSLB サイト間にサイト間の VPN 接続がない場合などです。

CLI コマンドを使用して Citrix ADC アプライアンスで GSLB セットアップを構成するには

1. まだ行われていない場合は、GSLB 機能を有効にします。

```
enable ns feature gslb
<!--NeedCopy-->
```

2. ローカル GSLB サイトを追加するための SNIP を特定します。
3. ローカル Citrix ADC アプライアンスの GSLB サイトを追加します。

```
add gslb site site-US 10.3.1.101
<!--NeedCopy-->
```

4. リモート Citrix ADC アプライアンス用の GSLB サイトを追加します。

```
add gslb site site-MX 172.16.1.101
add gslb site site-CO 192.168.1.101
<!--NeedCopy-->
```

5. GSLB セットアップで使用されているサービスを参照する GSLB 仮想サーバーを追加します。

```
add gslb vserver gslb-lb HTTP
<!--NeedCopy-->
```

6. GSLB セットアップに参加している各サイトの GSLB サービスを追加します。

```
add gslb service gslb_SVC30 172.16.1.100 HTTP 80 -siteName site-MX
add gslb service gslb_SVC10 10.3.1.100 HTTP 80 -siteName site-US
```



```
add gslb service gslb_SVC20 192.168.1.100 HTTP 80 -siteName site-
    CO
<!--NeedCopy-->
```

7. GSLB サービスを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb-lb -serviceName gslb_SVC10
bind gslb vserver gslb-lb -serviceName gslb_SVC20
bind gslb vserver gslb-lb -serviceName gslb_SVC30
<!--NeedCopy-->
```

8. ドメインを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb-lb -domainName www.mycompany.com -TTL 30
<!--NeedCopy-->
```

9. DNS クエリをリッスンする ADNS サービスを追加します。

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

GSLB セットアップの設定を同期する

October 7, 2021

通常、GSLB セットアップには、データセンターごとに GSLB サイトが設定されたいくつかのデータセンターがあります。GSLB に参加する各 Citrix ADC で、1 つの GSLB サイトをローカルサイトとして構成し、他のサイトをリモートサイトとして構成します。後で別の GSLB サイトを追加する場合は、すべての GSLB サイトにわたる構成が同一であることを確認する必要があります。Citrix ADC の GSLB 構成同期オプションを使用して、GSLB サイト間で構成を同期できます。

同期オプションを使用する Citrix ADC アプライアンスは、「メインサイト」と呼ばれ、構成がコピーされる GSLB サイトは「下位サイト」と呼ばれます。GSLB 設定を同期すると、GSLB セットアップに参加しているすべての GSLB サイトの構成は、メインサイトの構成と同様になります。

同期は親サイトでのみ行われます。同期は GSLB 子サイトの構成には影響しません。これは、親サイトと子サイトの構成が同一ではないためです。子サイトの構成は、そのサイトとその親サイトの詳細のみで構成されます。また、GSLB サービスは必ずしも子サイトで設定する必要はありません。

- メインノードは、メインノードと下位ノードの構成の違いを検出し、下位ノードの構成をメインノードに似せるように変更します。

  同期を強制すると（「強制同期」オプションを使用）、アプライアンスは下位ノードから GSLB 設定を削除し、メインノードと類似するように下位ノードを構成します。

- 同期中にコマンドが失敗した場合、同期は中断されず、エラー・メッセージは **/var/netscaler/gslb** ディレクトリ内の **.err** ファイルに記録されます。

- 同期は親サイトでのみ行われます。同期は GSLB 子サイトの構成には影響しません。これは、親サイトと子サイトの構成が同一ではないためです。子サイトの構成は、そのサイトとその親サイトの詳細のみで構成されます。また、GSLB サービスは必ずしも子サイトで設定する必要はありません。

- 内部ユーザーログインを無効にすると、GSLB 自動同期は SSH キーを使用して設定を同期します。ただし、パーティション環境で GSLB 自動同期を使用するには、内部ユーザーログインを有効にし、ローカルおよびリモート GSLB サイトのパーティションユーザー名が同じであることを確認する必要があります。

> 注
>
> - リモート GSLB サイトの RPC ノードで、リモートサイトの IP（クラスタセットアップ用のクラスタ IP アドレス）とポート（RPC の場合は 3010、セキュア RPC の場合は 3008）を指定して、自動同期接続を受け入れるようにファイアウォールを構成します。ほとんどの場合、リモートサイトに到達するデフォルトルートが管理サブネット内にある場合は、NSIP が送信元 IP アドレスとして使用されます。
>
> 異なる送信元 IP アドレスを設定するには、GSLB サイトの IP アドレスと SNIP が別のサブネットにある必要があります。また、GSLB サイト IP サブネットを介してリモートサイト IP アドレスへの明示的なルートを定義する必要があります。
>
> セキュリティを強化するには、内部ユーザーアカウントと RPC ノードのパスワードを変更することをお勧めします。内部ユーザーアカウントのパスワードは、RPC ノードパスワードによって変更されます。詳細については、「RPC ノードのパスワードを変更する」を参照してください。

saveconfig オプションを使用すると、同期プロセスに参加しているサイトは、次の方法で自動的に構成を保存します。

リモート GSLB サイトの RPC ノードで、リモートサイトの IP（クラスタセットアップ用のクラスタ IP アドレス）とポート（RPC の場合は 3010、セキュア RPC の場合は 3008）を指定して、自動同期接続を受け入れるようにファイアウォールを構成します。ほとんどの場合、リモートサイトに到達するデフォルトルートが管理サブネット内にある場合は、NSIP が送信元 IP アドレスとして使用されます。

異なる送信元 IP アドレスを設定するには、GSLB サイトの IP アドレスと SNIP が別のサブネットにある必要があります。また、GSLB サイト IP サブネットを介してリモートサイト IP アドレスへの明示的なルートを定義する必要があります。RPC ノードの送信元 IP アドレスは各 Citrix ADC アプライアンスに固有であるため、GSLB に参加しているサイト間でソース IP アドレスを同期できません。したがって、同期を強制した後（sync gslb config-forceSync コマンドを使用するか、GUI で ForceSync オプションを選択）、他の Citrix ADC アプライアンスのソース IP アドレスを手動で変更する必要があります。ポート 22 は、データベースファイルをリモートサイトに同期させるためにも必要です。

すべての **GSLB** サイトでの構成の同期にかかる時間を改善するには

コマンドプロンプトで TCP プロファイル設定を次のように構成します。

```
1 set tcpprofile nstcp_internal_apps -bufferSize 4194304 -sendBuffsize
     4194304 -tcpmode ENDPOINT
2 <!--NeedCopy-->
```

同期の制限

- メインサイトでは、リモート GSLB サイトの名前は、それらのサイトをホストしている Citrix ADC アプライアンスで構成されたサイトの名前と同じである必要があります。
- 同期中に、トラフィックの中断が発生することがあります。
- Citrix ADC は、構成の最大 20 万行までの同期がテストされています。
- 同期が失敗することがあります。
  - 流出方法が「接続」から「ダイナミック接続」に変更された場合。
  - メインノード上の GSLB 仮想サーバーにバインドされた GSLB サービスのサイトプレフィックスを交換し、同期を試みる場合。
  - RPC ノードのパスワードが NSIP アドレスとループバック IP アドレスで異なる場合。
  - 同じ Citrix ADC アプライアンスの異なるパーティションで構成されている GSLB サイトで同期を実行する場合。
- GSLB サイトをハイアベイラビリティ（HA）ペアとして設定した場合、プライマリノードとセカンダリノードの RPC ノードのパスワードは同じである必要があります。
- GSLB 設定の一部である GLSB エンティティの名前を変更する場合（「show gslb runningConfig」コマンドを使用して GSLB 設定を表示します）。設定を他の GSLB サイトに同期するには、強制同期オプションを使用する必要があります。

> 注：
>
> - 増分同期では、設定を他の GSLB サイトに同期するために強制同期オプションを使用する必要はありません。これは、Citrix ADC リリース 13.0 ビルド 79.x 以降の起動に適用されます。

注：GSLB 設定の一部の設定による制限を克服するには、強制同期オプションを使用できます。ただし、強制同期オプションを使用すると、GSLB エンティティが削除され、設定に再追加され、GSLB 統計はゼロにリセットされます。そのため、設定変更中にトラフィックが中断されます。

GSLB セットアップの同期を開始する前に注意すべきポイント

GSLB セットアップの同期を開始する前に、次のことを確認してください。

- メインサイトを含むすべての GSLB サイトでは、対応する GSLB サイトの IP アドレスに対して管理アクセスと SSH を有効にする必要があります。GSLB サイトの IP アドレスは、Citrix ADC アプライアンスが所有す

る IP アドレスである必要があります。GSLB サイトの IP アドレスの追加と管理アクセスの有効化の詳細については、「基本的な GSLB サイトの構成」を参照してください。

- メインサイトと見なされる Citrix ADC アプライアンスの GSLB 構成は完全であり、すべてのサイトでコピーするのが適切です。
- GSLB 設定を初めて同期する場合は、GSLB に参加するすべてのサイトに、それぞれのローカルサイトの GSLB サイトエンティティが必要です。
- 設計上、同じ構成を持たないサイトを同期していません。
- メインサイトと下位サイトは同じ Citrix ADC バージョンを実行します。リリース 12.1、ビルド 50.x 以降、アプライアンスは同期を開始する前にメインサイトと下位サイトのファームウェアバージョンをチェックします。メインサイトと下位サイトが異なるバージョンを実行している場合、バージョン間で互換性のない変更がプッシュされないように、そのリモートサイトの同期は中止されます。また、同期が中止されたサイトの詳細を示すエラーメッセージが表示されます。

  次の図に、CLI および GUI からのエラーメッセージの例を示します。

```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status        Reason
-------------------------------------------------------------------------------------------------------------
        s2                       Failure       Error: Different netscaler release on the remote site. Local Site: 13.0, Remote Site: 12.1
        s1                       Success       All Done
        s3                       Success       All Done
Done
>
```

```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status        Reason
-------------------------------------------------------------------------------------------------------------
        s2                       Failure       Error: Different netscaler release on the remote site. Local Site: 13.0, Remote Site: 12.1
        s1                       Success       All Done
        s3                       Success       All Done
Done
>
```

> 重要
>
> 次のディレクトリは、GSLB 設定の同期の一部として同期されます。
>
> - /var/netscaler/locdb/
> - /var/netscaler/ssl/
> - /var/netscaler/inbuilt_db/

GSLB に参加しているサイト間の手動同期

October 7, 2021

マスターサイトとスレーブサイト間での GSLB 構成の手動同期は、次の方法で実行されます。

- マスターサイトは、自身のサイトとスレーブサイトの構成の違いを検出します。
- マスターサイトは、構成の違いをスレーブサイトに適用します。
- マスターサイトは、GSLB セットアップ内のすべてのスレーブサイトとの構成同期を実行し、同期プロセスを完了します。

重要: After GSLB 構成が同期されているため、どの GSLB サイトでも構成をロールバックすることはできません。同期プロセスがリモートサイトの構成を上書きしないことが確実な場合にのみ、同期を実行します。ローカルサイトとリモートサイトの構成が設計上異なる場合、サイトの同期は望ましくなく、サイトの停止につながります。一部のコマンドが失敗し、一部のコマンドが成功した場合、成功したコマンドはロールバックされません。

注意事項

- 同期を強制すると（「強制同期」オプションを使用）、Citrix ADC アプライアンスはスレーブサイトから GSLB 構成を削除します。次に、マスターサイトはスレーブサイトを自分のサイトと同様になるように構成します。
- 同期中にコマンドが失敗しても、同期は中止されません。エラーメッセージは、の.err ファイルに記録されます。/var/netscaler/gslb ディレクトリ。
- saveconfig オプションを使用すると、同期プロセスに参加しているサイトは、次のように構成を自動的に保存します。
  - マスターサイトは、同期プロセスを開始する直前に構成を保存します。
  - スレーブサイトは、同期のプロセスが完了した後、構成を保存します。スレーブサイトは、構成の違いが正常に適用された場合にのみ、その構成を保存します。スレーブサイトで同期が失敗した場合は、失敗の原因を手動で調査し、修正措置を講じる必要があります。

CLI を使用して **GSLB** 設定を同期するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、GSLB サイトを同期し、構成を確認します。

```
sync gslb config [-preview | -forceSync <string> | -nowarn | -
   saveconfig] [-debug]
show gslb syncStatus
<!--NeedCopy-->
```

例**:**

```
sync gslb config
```



```
2
3 [WARNING]: Syncing config may cause configuration loss on other site.
4
5 Please confirm whether you want to sync-config (Y/N)? [N]:y
6
7 Sync Time: Dec 9 2011 10:56:9
8
9 Retrieving local site info: ok
10
11 Retrieving all participating gslb sites info: ok
12
13 Gslb_site1[Master]:
14
15 Getting Config: ok
16
17 Gslb_site2[Slave]:
18
19 Getting Config: ok
20
21 Comparing config: ok
22
23 Applying changes: ok
24
25 Done
26 <!--NeedCopy-->
```

GUI を使用して **GSLB** 設定を同期するには、次の手順を実行します。

1. 「トラフィック管理」**>**「**GSLB**」**>**「ダッシュボード」に移動します。
2. 「自動同期 **GSLB** 」をクリックし、「**ForceSyn** 」を選択します。
3. 「**GSLB** サイト名」で、マスターノード構成と同期する GSLB サイトを選択します。

GSLB 同期のプレビュー

GSLB 同期操作をプレビューすると、マスターノードと各スレーブノードの違いを確認できます。矛盾がある場合は、GSLB 設定を同期する前にトラブルシューティングできます。

CLI を使用して **GSLB** 同期出力をプレビューするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

```
1 sync gslb config -preview
2 <!--NeedCopy-->
```


GUI を使用して **GSLB** 同期出力をプレビューするには、次の手順を実行します。

1. [構成] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動します。
2. 「自動同期 **GSLB** 」をクリックし、「プレビュー」を選択します。
3. [実行] をクリックします。
   プログレスウィンドウに、構成の不一致が表示されます。

同期処理中にトリガーされたコマンドのデバッグ

同期処理中にトリガーされた各コマンドのステータス（成功または失敗）を表示し、それに応じてトラブルシューティングを行うことができます。

CLI を使用して **GSLB** 同期コマンドをデバッグするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力します。

```
sync gslb config -debug
<!--NeedCopy-->
```

GUI を使用して **GSLB** 同期コマンドをデバッグするには、次の手順を実行します。

1. 構成に移動 > トラフィック管理 > **GSLB** > ダッシュボード。
2. [自動同期 **GSLB**] をクリックし、[デバッグ] を選択します。
3. [実行] をクリックします。進行状況ウィンドウには、同期中にトリガーされた各コマンドのステータスが表示されます。

GSLB に参加しているサイト間のリアルタイム同期

October 7, 2021

「AutomaticConfigSync」オプションを使用すると、メインサイトのリアルタイム GSLB 構成をすべての下位サイトに自動的に同期できます。設定を同期するために AutoSync オプションを手動でトリガーする必要はありません。

差分同期または完全同期を使用して、メインサイトの GSLB 構成をすべての下位サイトに自動的に同期できます。「gslbSyncMode」パラメータを使用すると、同期モードを選択できます。

> 注:
>
> Citrix ADC リリース 13.0 ビルド 79.x 以降、GSLB 同期の増分同期がサポートされています。デフォルトでは、同期は差分同期を使用して実行されます。増分同期は、”IncrementalSync” パラメータを有効にすることによって実行できます。

リアルタイム同期機能の使用に関するベストプラクティス

- サイトとして参加するすべての Citrix ADC アプライアンスは、同じバージョンの Citrix ADC ソフトウェアを使用することをお勧めします。
- RPC ノードのパスワードを変更するには、まず下位サイトでパスワードを変更し、次にメインサイトでパスワードを変更します。
- GSLB に参加している各サイトでローカル GSLB サイトを構成します。
- 構成が実行されるサイトの 1 つで automaticConfigSync を有効にします。このサイトは、最終的に他の GSLB サイトに同期されます。
- 新しい設定がある場合、または既存の設定に変更が加えられた場合は、「show gslb syncStatus」コマンドを使用してステータスを確認し、変更がすべてのサイトで同期されているかどうか、またはエラーがあったかどうかを確認します。
- RSYNC ポートモニタリングを有効にする必要があります。

CLI を使用してリアルタイム同期を有効にするには

コマンドプロンプトで入力します。

```
set gslb parameter -automaticConfigSync (ENABLED | DISABLED)
<!--NeedCopy-->
```

例:

```
set gslb parameter -automaticConfigSync ENABLED
<!--NeedCopy-->
```

GUI を使用してリアルタイム同期を有効にするには

1. [設定] > [トラフィック管理] > [**GSLB**] > [**GSLB** 設定の変更] に移動します。
2. [自動設定] **[**同期**]** を選択します。

   注: このオプションは、構成が実行されるサイトでのみ有効にする必要があります。

次のトピックの詳細については、GSLB に参加しているサイト間の手動同期を参照してください。

- GSLB 同期のプレビュー
- 同期処理中にトリガーされたコマンドのデバッグ

注意事項

- リアルタイム同期に関連する統合ログファイルは /var/netscaler/gslb/periodic_sync.log ディレクトリに保存されます。
- デフォルトの設定ファイルはに保存されます /var/netscaler/gslb_sync/ ディレクトリ。
- メインサイトでは、次のディレクトリ構造を使用しています。
  - メインサイトは、すべてのファイルを /var/netscaler/gslb_sync/master ディレクトリに保存します。
  - メインサイトは、下位サイトと同期する必要がある構成ファイルを/var/netscaler/gslb_sync/master/gslbconf/ディレクトリに保存します。
  - すべての下位サイトから取得されたステータスファイルは、/var/netscaler/gslb_sync/master/slavestatus/ディレクトリに保存されます。
- 下位サイトは、次のディレクトリ構造を使用します。
  - 下位サイトは、/var/netscaler/gslb_sync/slave/gslbconf ディレクトリから適用する最新の構成ファイルを選択します。
  - 下位サイトは、ステータスファイルを /var/netscaler/gslb_sync/slave/gslbstatus ディレクトリに保存します。
- 管理パーティションのセットアップでは、/var/partitions/`partition name`/netscaler/gslb_sync という場所に同じディレクトリ構造が維持されます。
- すべてのサイトの時計は、協定世界時 (UTC) などのリアルタイム標準に正確に設定する必要があります。

GSLB 設定の増分同期

GSLB 設定の自動同期機能は、メインサイトの構成の変更を 10 秒間隔でチェックし、同期を実行します。この同期間隔値は設定可能です。

差分同期では、最後の同期と後続の同期間隔（10 秒）の間にメインサイトで変更された構成のみが、すべての下位サイト間で同期されます。増分同期がデフォルトの動作です。インクリメンタル構成のみをプッシュすると、構成ファイルのサイズが大幅に減少し、同期時間が短縮されます。差分同期が失敗すると、システムは完全な構成同期を自動的に実行します。

差分同期は、次の方法で実行されます。

- メインサイトは、最新の変更のみで構成された構成ファイルをすべての下位サイトにプッシュします。最新の変更とは、最後の同期とそれ以降の同期間隔（10 秒）の間に変更された設定を指します。
- 各下位サイトは、最新の変更を独自のサイトに適用します。
- ダウン状態の下位サイトでは、差分同期は試行されません。サイトが復帰すると、再び同期が実行されます。
- 下位サイトは、各ステップでステータスログを生成し、特定の場所のファイルにコピーします。
- メインサイトは、指定された場所からステータスログファイルをプルします。
- メインサイトは、すべての下位サイトからのログを組み合わせたログファイルを準備します。
- この結合されたログファイルはに保存されます /var/netscaler/gslb/periodic_sync.log ファイル。

設定ファイルが格納されているディレクトリの詳細については、「注意事項」セクションを参照してください。

CLI を使用して **GSLB** 設定の増分同期を有効にするには

```
set gslb parameter -AutomaticConfigSync (ENABLED | DISABLED) -
    GSLBSyncMode (IncrementalSync | FullSync) -GslbConfigSyncMonitor (
    ENABLED | DISABLED) -GSLBSyncInterval <secs> -GSLBSyncLocFiles (
    ENABLED | DISABLED)
<!--NeedCopy-->
```

例**:**

```
set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
    IncrementalSync
<!--NeedCopy-->
```

増分同期では、次の設定可能なパラメータが提供され、GSLB 設定の同期にかかる全体的な時間が短縮されます。

- **gslbConfigSyncMonitor**: GSLB 構成同期モニタパラメータを有効にして、下位サイトの RSYNC ポート（リモート GSLB サイト IP アドレス上の SSH ポート 22）の状態を監視します。モニタの下位サイトの状態が DOWN と表示されている場合、そのサイトに対する RSYNC 操作はスキップされます。これにより、ダウンしているリモートサイトに接続しようとしたことによる同期の遅延が軽減されます。

  CLI で **RSYNC** ポートモニタリングを有効にする例を示します。

  ```
  set gslb parameter -GSLBSyncMode IncrementalSync -
      GslbConfigSyncMonitor ENABLED
  <!--NeedCopy-->
  ```

- **gslbSyncInterval**: GSLB 設定の同期が行われる時間間隔（秒単位）を設定します。デフォルトでは、GSLB 設定の自動同期機能は、10 秒ごとに自動的に GSLB 設定を同期します。時間間隔は任意の値に変更できます。たとえば、5 秒未満など、この値を低い値に設定しないでください。頻繁に同期すると、管理 CPU 消費が増加する可能性があるためです。

  > 注:
  >
  > 管理パーティションのセットアップでは、時間間隔はグローバルパラメータであるため、デフォルトパーティションでのみ設定できます。

  同期間隔を設定する例**:**

  ```
  set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
      IncrementalSync -GSLBSyncInterval 7
  <!--NeedCopy-->
  ```


- **gslbSyncLocFiles**-GSLB 設定の同期中に、デフォルトでは、場所 DB ファイルの変更が検出され、自動的に同期されます。場所 DB ディレクトリは頻繁に変更されないため、管理者は場所 DB ファイルの自動同期を無効にできます。代わりに、管理者は場所 DB ファイルを GSLB 下位サイトに手動でコピーする必要があります。場所 DB ファイルの同期には長い時間がかかります。したがって、これを避けると、全体の同期時間が短縮されます。

  場所 **DB** ファイルの自動同期を無効にする例**:**

```
set gslb parameter -GSLBSyncMode IncrementalSync -
    GSLBSyncLocFiles DISABLED
<!--NeedCopy-->
```

GUI を使用して **GSLB** 増分同期を有効にするには

1. [トラフィック管理**] > [GSLB] > [**ダッシュボード**] > [GSLB** 設定の変更] に移動します。
2. **[GSLB** パラメータの設定] ページで、次の操作を実行できます。
   - 増分同期を有効にするには、[**GSLB **** 同期モード**]** ドロップダウンメニューから **[Incremental-Sync****] を選択します。
   - 自動 GSLB 設定の同期間隔を設定するには、[**GSLB** 同期間隔] フィールドに時間を秒単位で入力します。
   - RSYNC ポートの監視を有効にするには、[**GSLB** 構成同期モニタ] チェックボックスをオンにします。
   - 場所 DB ファイルの自動同期を無効にするには、[**GSLB** 同期ロックファイル] チェックボックスをオフにします。

GSLB 設定の完全同期

メインサイトで設定が変更されると、メインサイトの完全な GSLB 実行構成がすべての下位サイトにプッシュされます。
差分同期が構成されている場合でも、メインサイトが下位サイトの構成ステータスを認識しない場合、完全同期が実行されます。このようなシナリオのいくつかは次のとおりです。

- GSLB 設定の自動同期機能を初めて有効にします。
- Citrix ADC アプライアンスを再起動します。
- GSLB の展開には複数のメインサイトがあり、別のメインサイトがアクティブなメインサイトになります。
- GSLB デプロイメントに新しい下位サイトを追加します。

GSLB 設定の完全同期は、次の方法で実行されます。

- メインサイトは、最新の構成ファイルをすべての下位サイトにプッシュします。
- 各下位サイトは、メインサイトから送信された最新の構成ファイルと、独自の構成を比較します。下位サイトは、構成の違いを特定し、独自のサイトに対してデルタ構成を適用します。

- 下位サイトは、各ステップでステータスログを生成し、特定の場所のファイルにコピーします。
- メインサイトは、指定された場所からステータスログファイルをプルします。
- メインサイトは、すべての下位サイトからのログを組み合わせたログファイルを準備します。
- この結合されたログファイルはに保存されます /var/netscaler/gslb/periodic_sync.log ファイル。

自動同期中にサイトを (`sync gslb config`コマンドで) 手動で同期しようとすると、「同期が進行中です」というエラーメッセージが表示されます。自動同期は、手動で同期しているサイトでは起動できません。

> 注意**:**
>
> Citrix ADC 12.1 ビルド 49.37 以降、GSLB 構成を同期すると SNMP トラップが生成されます。リアルタイム同期では、最初の SNMP トラップの同期ステータスが失敗としてキャプチャされます。実際の同期ステータスを持つ最初のトラップの直後に、2 番目の SNMP トラップが自動的に生成されるので、このステータスを無視できます。ただし、2 回目の試行で同期が失敗した場合、同期ステータスが以前の同期ステータスから変更されていないため、SNMP トラップは生成されません。

トラップを生成するための Citrix ADC アプライアンスの構成の詳細については、「SNMP トラップを生成するように Citrix ADC を構成する」を参照してください。

CLI を使用して **GSLB** 完全同期を有効にするには

```
set gslb parameter -GSLBSyncMode (IncrementalSync | FullSync)
<!--NeedCopy-->
```

例**:**

```
set gslb parameter -GSLBSyncMode FullSync
<!--NeedCopy-->
```

GUI を使用して GSLB インクリメンタル同期を有効にするには、次の手順を実行します。

1. [トラフィック管理**] > [GSLB] > [**ダッシュボード**] > [GSLB** 設定の変更] に移動します。
2. [**GSLB** パラメータの設定] ページで、[**GSLB** 同期モード**]** ドロップダウンメニューから **[FullSync**] を選択します。

GSLB 展開内の複数のメインサイト

Citrix ADC アプライアンスは、アクティブ/パッシブ展開で複数のメインサイトをサポートします。GSLB メインサイトの障害に対処するために、GSLB 展開に 2 つのメインサイトを用意することをお勧めします。メインサイトが 2 つあると、GSLB 設定の同期の単一点障害を回避できます。いつでも、ユーザーから GSLB 設定をアクティブに処理できるメインサイトは 1 つだけです。構成の変更が複数のメインサイトで同時に実行されると、構成の不一致または構成が失われる可能性があります。したがって、一度に 1 つのメインサイトからの構成変更のみを実行し、アクティ

ブなメインサイトに障害が発生した場合は、もう一方のメインサイトをバックアップとして使用することをお勧めします。

> 注:
>
> GSLB 展開で複数のメインサイトを使用する場合は、RSYNC モニタリングを有効にする必要があります。

GSLB 構成同期のメインサイトの 1 つとして GSLB ノードを作成するには、次のコマンドを実行します。

```
set gslb parameter -automaticConfigSync Enabled
<!--NeedCopy-->
```

GSLB 同期のステータスと概要の表示

October 7, 2021

GSLB 構成が GSLB サイト間で同期された後、詳細なステータスと最後の GSLB 同期操作の概要を表示できます。これは、手動およびリアルタイムの GSLB 同期の両方に適用されます。

CLI を使用して GSLB 同期ステータスまたはサマリーを表示するには

コマンドプロンプトで入力します。

```
show gslb sync status
<!--NeedCopy-->
```

または

```
show gslb syncStatus -summary
<!--NeedCopy-->
```

GSLB 手動同期のサンプル構成出力

次の出力は、GSLB 構成の手動同期のステータスを示しています。


```
> sh gslb syncStatus
Displaying the status of the manual GSLB configuration synchronization:

gslb_site1[Master]:
        Getting Config:  ok
gslb_site2[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
gslb_natsite1[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
 Done
>
```

次の出力は、GSLB 構成の手動同期のステータスの概要を示しています。

```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status          Reason
-------------------------------------------------------------------------------------------------------------
        gslb_site1               Success         All Done
        gslb_site2               Failure         Error executing command on gslb site...ERROR: Connection failed
        gslb_natsite1            Success         All Done
Done
>
```

GSLB リアルタイム同期のサンプル構成出力

次の出力は、マスターサイトのリアルタイム GSLB 構成同期のステータスを示しています。

```
1 > sh gslb syncStatus
2 Displaying the status of the real time GSLB configuration
      synchronization as master node:
```



```
3
4  site2[Master]:
5                  New GSLB configuration detected at Fri Jan 23 20:54:24
                       2020
6                  Fetching current configuration: Done
7                  Updating default.conf file: Done
8  site1[Slave]:
9                  Syncing gslb static proximity database to node site1:
                       Done
10                 Syncing inbuilt GSLB static proximity database to node
                       site1: Done
11                 Syncing ssl certificates, keys and CRLS to node site1:
                       Done
12                 Syncing current configuration to site1: Done
13                 Pulling status files from site1: Status file not
                       available yet(Sync in progress)
14                 Pulling status files from site1: Done
15                 site1 received new configuration from 10.102.217.205 in
                        file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
                       conf
16                 Firing set gslb parameter -startConfigSync ENABLED
                       command: Done
17                 Fetching running GSLB Config: Done
18                 Comparing config: Done
19                 Applying changes: Done
20                 Firing set gslb parameter -startConfigSync DISABLED
                       command: Done
21                 Updating default.conf file: Done
22 Done
23 <!--NeedCopy-->
```

次の出力は、スレーブサイトのリアルタイム GSLB 構成同期のステータスを示しています。

```
1  > sh gslb syncStatus
2  Displaying the status of the real time GSLB configuration
       synchronization as slave node:
3
4                  site1 received new configuration from 10.102.217.205 in
                        file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
                       conf
5                  Firing set gslb parameter -startConfigSync ENABLED
                       command: Done
6                  Fetching running GSLB Config: Done
```



```
                Comparing config: Done
                Applying changes: Done
                Firing set gslb parameter -startConfigSync DISABLED
                    command: Done
                Updating default.conf file: Done
Done
<!--NeedCopy-->
```

次の出力は、マスターサイトのリアルタイム GSLB 構成同期のステータスの概要を示しています。

```
> sh gslb syncStatus -summary
Displaying the status summary of the real time GSLB configuration
    synchronization as master node:

------------------------------------------------------------------------------

                Site Name                                   Status
                                Reason
------------------------------------------------------------------------------

                site2                                       Success
                            All Done
                site1                                       Success
                            All Done

Done
<!--NeedCopy-->
```

次の出力は、スレーブサイトのリアルタイム GSLB 構成同期のステータスの概要を示しています。

```
> sh gslb syncStatus  - summary
Displaying the status summary of the real time GSLB configuration
    synchronization as slave node:

------------------------------------------------------------------------------

                Site Name                                   Status
                                Reason
------------------------------------------------------------------------------

                site1                                       Success
                            All Done
```



```
8
9  Done
10 <!--NeedCopy-->
```

GUI を使用して **GSLB** 同期のステータスまたはサマリーを表示するには

1. [構成] > [トラフィック管理] > [**GSLB**] > [ダッシュボード**]** に移動します。
2. 必要に応じて、[同期サマリーの表示**]** または **[**同期ステータスの表示] をクリックします。

GSLB 設定の同期化のための **SNMP** トラップ

October 7, 2021

Citrix ADC 12.1 ビルド 49.xx 以降、Citrix ADC アプライアンスは、GSLB 構成を同期するときに、ローカルサイトとリモートサイトの両方に対して SNMP トラップを生成します。SNMP トラップは、手動同期とリアルタイム同期の両方に対して生成されます。

GSLB 設定を初めて同期すると、SNMP トラップが生成されます。以降の同期試行では、以前の同期ステータスから同期ステータスが変更された場合にのみ、SNMP トラップが生成されます。また、SNMP トラップは、同期ステータスが以前の状態から変更されたサイトに対してのみ生成されます。

たとえば、最初の GSLB 設定の同期が成功したとします。構成を 2 回目に同期し、同期が再び成功した場合、ステータスが変更されないため、SNMP トラップは生成されません。ただし、3 回目の試行では、いずれかのサイトで同期が失敗すると、そのサイトだけに SNMP トラップが生成されます。

高可用性とクラスタのセットアップでは、以前の同期ステータスに関係なく、新しいノードから GSLB 設定を同期すると、アプライアンスは SNMP トラップを生成します。また、SNMP トラップオプションが以前に無効になってから有効になっている場合、以前の同期ステータスに関係なく、その時点から SNMP トラップが生成されます。

GSLB 設定同期の SNMP トラップは、次の詳細を提供します。

- SNMP トラップが送信される GSLB サイトの名前。
- GSLB 設定の同期ステータス: 成功または失敗。
- GSLB 構成同期モード： インクリメンタル同期または完全同期。
- （任意）SNMP トラップに関する詳細情報。

SNMP トラップは、次のシナリオで生成されます。

- GSLB サイトの GSLB 同期ステータスは、[成功] から [失敗]、逆に反転します。
- GSLB 同期モードは、増分同期から完全同期に変更され、逆に変化します。

> 注:
>
> 差分同期が有効になっている場合でも、何らかの理由で GSLB サイトで完全同期が実行された場合、完全同期

> の理由は、トラップメッセージの「詳細情報」セクションに記載されています。たとえば、新しい GSLB サイトが GSLB 構成に追加された場合などです。

SNMP トラップメッセージの例

次の図に、gslb_site2 の SNMP トラップの例を示します。このトラップでは、完全同期モードを使用して GSLB 構成の同期が成功します。

```
2021-03-18 18:18:58 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (667165) 1:51:11.65    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Success, Full Sync Mode, Switching to Inc Sync
 Mode"  iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

次の図に、gslb_site2 の SNMP トラップの例を示します。この場合、インクリメンタル同期モードを使用して GSLB 構成の同期が正常になります。

```
2021-03-18 18:24:18 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (699113) 1:56:31.13    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Success, Inc Sync Mode"   iso.3.6.1.4.1.5951.4
.1.1.2.0 = IpAddress: 10.102.146.2
```

次の図に、差分同期モードを使用した GSLB 設定の同期が失敗する gslb_site2 の SNMP トラップの例を示します。エラーメッセージは、同期を完了するために手動でエラーを修正する必要があることを示します。

```
2021-03-18 18:17:34 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (658753) 1:49:47.53    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure, Inc Sync Mode, Site is not in sync, I
ncremental config application has failed, Switching to Full Sync Mode." iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
2021-03-18 18:17:49 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (660256) 1:50:02.56    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure,Full Sync Mode, Site is not in sync, F
ull sync config application has failed, Please fix the errors." iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

次の図に、差分同期モードを使用した GSLB 設定の同期が失敗する gslb_site2 の SNMP トラップの例を示します。また、同期が失敗した理由、つまりサイトモニターがダウンしている理由も示します。

```
2021-03-18 18:21:39 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (683289) 1:53:52.89    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure, Inc Sync Mode, Syncing current config
uration to gslb_site2: Skipped, Site Monitor is down"   iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

GSLB ダッシュボード

October 7, 2021

GSLB ダッシュボードで GSLB に参加している GSLB サイトの全体的なステータスを表示できます。

ダッシュボードから GSLB 設定にアクセスできます。ダッシュボードから GSLB 設定ウィザードを起動することもできます。さらに、同期を実行し、ダッシュボードから GSLB 設定をテストできます。

GSLB ダッシュボードにアクセスするには、[設定] > [トラフィック管理] > [**GSLB**] > [ダッシュボード] に移動します。

GSLB サービスの監視

October 7, 2021

リモートサービスを GSLB 仮想サーバーにバインドすると、GSLB サイトは、ラウンドトリップ時間と永続性情報であるネットワークメトリック情報を含むメトリック情報を交換します。

いずれかの参加サイト間でメトリック交換接続が一時的に失われた場合、リモートサイトは DOWN としてマークされ、UP 中の残りのサイトでロードバランシングが実行されます。サイトのメトリック交換が DOWN の場合、サイトに属するリモートサービスも DOWN とマークされます。

Citrix ADC アプライアンスは、MEP またはリモートサービスに明示的にバインドされているモニターを使用して、リモート GSLB サービスの状態を定期的に評価します。ローカル GSLB サービスの状態は MEP を使用してデフォルトで更新されるため、明示的なモニタをローカルサービスにバインドする必要はありません。ただし、明示的なモニターをリモートサービスにバインドすることはできます。モニタが明示的にバインドされている場合、リモートサービスの状態はメトリック交換によって制御されません。

デフォルトでは、モニターをリモート GSLB サービスにバインドすると、Citrix ADC アプライアンスはモニターから報告されたサービスの状態を使用します。ただし、以下の状況では、モニターを使用してサービスを評価するように Citrix ADC アプライアンスを構成できます。

- 常にモニタを使用する (デフォルト設定)。
- MEP が DOWN のときは、モニタを使用します。
- リモートサービスと MEP が DOWN のときは、モニタを使用します。

上記の設定の 2 番目と 3 番目の設定では、MEP が UP しているときにアプライアンスがモニタリングを停止できます。たとえば、階層的な GSLB 設定では、GSLB サイトは子サイトに関する MEP 情報を親サイトへ提供します。このような中間サイトは、ネットワークの問題のために子サイトの状態を DOWN として評価する場合がありますが、サイトの実際の状態は UP です。この場合、モニタを親サイトのサービスにバインドし、MEP を無効にして、リモートサービスの実際の状態を判断できます。このオプションを使用すると、リモートサービスの状態を判別する方法を制御できます。

モニターを使用するには、まずモニターを作成してから、GSLB サービスにバインドします。

モニタトリガーの設定

GSLB サイトは、常にモニタを使用する（デフォルト）か、MEP がダウンしているときにモニタを使用する、またはリモートサービスと MEP の両方がダウンしているときにモニタを使用するように設定できます。後者の 2 つのケースでは、Citrix ADC アプライアンスは、MEP が UP 状態に戻ったときに監視を停止します。

コマンドラインインターフェイスを使用してモニタトリガーを構成するには

コマンドプロンプトで入力します。


```
set gslb site <siteName>  -triggerMonitor (ALWAYS | MEPDOWN |
   MEPDOWN_SVCDOWN)
<!--NeedCopy-->
```

例:

```
set gslb site Site-GSLB-North-America  -triggerMonitor Always
<!--NeedCopy-->
```

構成ユーティリティを使用してモニターのトリガーを構成するには

1. 「トラフィック管理」>「**GSLB** 」>「サイト」に移動し、サイトをダブルクリックします。
2. [トリガーモニター] ドロップダウンリストで、モニタリングをトリガーするタイミングのオプションを選択します。

モニタの追加または削除

モニタを追加するには、タイプとポートを指定します。サービスにバインドされているモニターは削除できません。まず、モニタをサービスからバインド解除する必要があります。

コマンドラインインターフェイスを使用してモニタを追加するには

コマンドプロンプトで次のコマンドを入力して、モニタを作成し、構成を確認します。

```
add lb monitor <monitorName> -type <monitorType> -destPort <portNumber>

show lb monitor <monitorName>
<!--NeedCopy-->
```

例:

```
add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用してモニタを削除するには

コマンドプロンプトで入力します。

```
rm lb monitor <monitorName>
<!--NeedCopy-->
```

構成ユーティリティを使用してモニタを追加するには

[
トラフィック管理] > [負荷分散] > [モニター] に移動し、モニタを追加または削除します。

モニタを **GSLB** サービスにバインドする

モニターを作成したら、それらを GSLB サービスにバインドする必要があります。モニタをサービスにバインドするときは、モニタの重みを指定できます。1 つ以上の加重モニターをバインドした後、サービスのモニターしきい値を構成できます。このしきい値は、バインドされたモニタの重みの合計がしきい値を下回ると、サービスを停止します。

注：構成ユーティリティでは、モニタをバインドすると同時に重みと監視しきい値の両方を設定できます。コマンドラインを使用する場合は、別のコマンドを発行して、サービスの監視しきい値を設定する必要があります。

コマンドラインインターフェイスを使用してモニターを **GSLB** サービスにバインドするには

コマンドプロンプトで入力します。

```
bind monitor <name> <serviceName> [ -state (Enabled | Disabled) ] -
    weight <positiveInteger>
<!--NeedCopy-->
```

例:

```
bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サービスの監視しきい値を設定するには

コマンドプロンプトで入力します。


```
set gslb service <ServiceName> -monThreshold <PositiveInteger>
<!--NeedCopy-->
```

例**:**

```
set gslb service service-GSLB-1 -monThreshold 9
<!--NeedCopy-->
```

構成ユーティリティを使用してモニターを **GSLB** サービスにバインドするには

1. Traffic Management > GSLB > Services に移動します。
2. [**Monitor**] セクションをクリックし、モニタを GSLB サービスにバインドします。

構成ユーティリティを使用して **GSLB** サービスの監視しきい値を設定するには

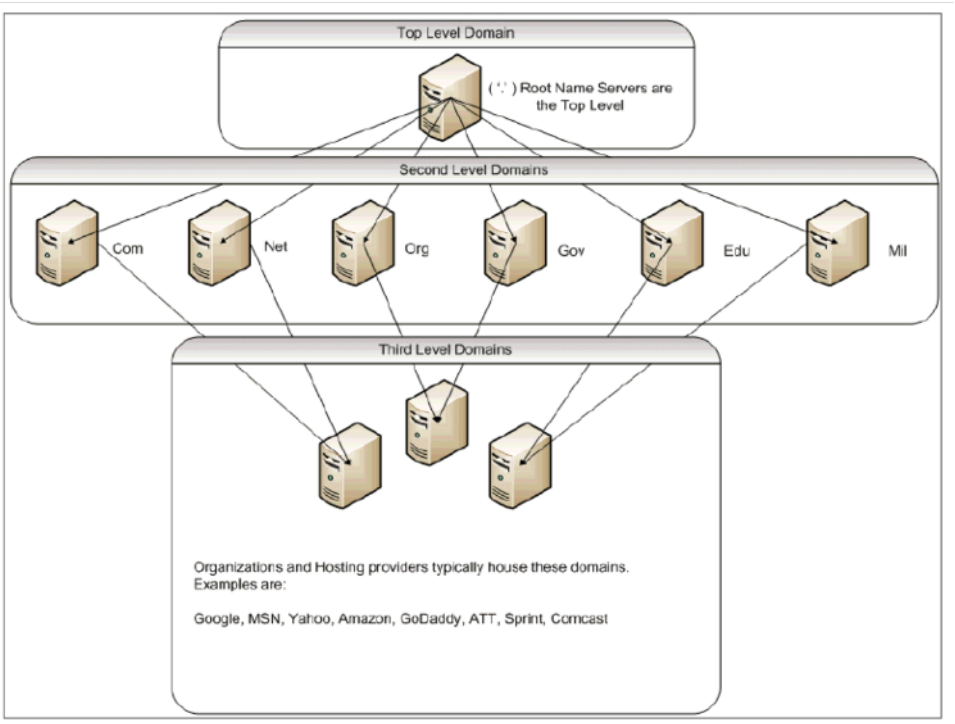
1. Traffic Management > GSLB > Services に移動します。
2. [モニタのしきい値] セクションをクリックし、しきい値を入力します。

ドメインネームシステムが **GSLB** をサポートする方法

October 7, 2021

ドメインネームシステム (DNS) は、クライアント/サーバーアーキテクチャを使用する分散データベースと見なされます。ネームサーバーはアーキテクチャ内のサーバーであり、リゾルバは、ネットワークを介してクエリを作成して送信するオペレーティングシステムにインストールされているライブラリルーチンであるクライアントです。

DNS の論理階層を次の図に示します。



> 注:
>
> 第 2 レベルのルートサーバーは、.com、.net、.org、.gov ドメインなどのネームサーバー委任のネームサーバーからアドレスへのマッピングを維持する責任があります。第 2 レベルドメイン内の各ドメインは、下位レベルの組織ドメインのネームサーバからアドレスへのマッピングを維持する責任があります。組織レベルでは、WWW、FTP、およびその他のサービスを提供するホストに対して、個々のホストアドレスが解決されます。

委任

現在の DNS トポロジの主な目的は、1 つの機関ですべてのアドレスレコードを維持する負担を軽減することです。これにより、組織の名前空間を特定の組織に委任できます。その後、組織はそのスペースを組織内のサブドメインにさらに委任できます。たとえば、citrix.com では、`sales.citrix.com`、`education.citrix.com`、`support.citrix.com`というサブドメインを作成できます。対応する部門は、サブドメインに対して権限のある独自のネームサーバーセットを保持し、ホスト名とアドレスマッピングの独自のセットを維持できます。すべての Citrix アドレスレコードを管理する責任は、単一の部門ではありません。各部門は、アドレスを変更したり、トポロジを変更したり、上位レベルのドメインや組織でより多くの作業を課すことはありません。

階層トポロジの利点

階層トポロジの利点には、次のようなものがあります。

- スケーラビリティ
- キャッシュ機能を各レベルでネームサーバーに追加する。DNS 要求は、特定のドメインに対して権限がないがクエリへの応答を提供できるホストによって処理され、輻輳と応答時間が短縮されます。

- キャッシングは、サーバー障害に対する冗長性と回復力ももたらします。1 つのネームサーバーに障害が発生しても、同じレコードの最近キャッシュされたコピーを持つ他のサーバーからレコードを提供できる可能性があります。

リゾルバ

リゾルバは DNS システム内のクライアントコンポーネントです。ドメイン・ネーム・スペースからの情報を必要とするホスト上で実行されているプログラムは、リゾルバを使用します。リゾルバは次の処理を行います。

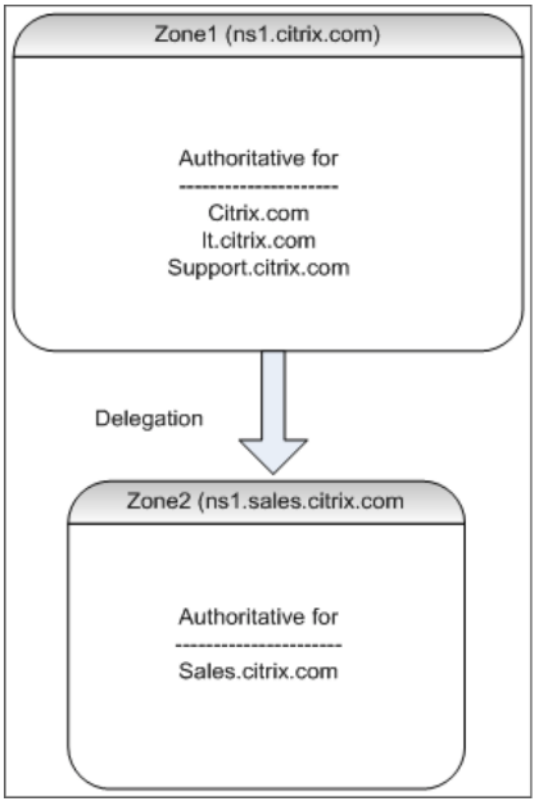
- ネームサーバを照会する。
- レスポンスの解釈（リソースレコードまたはエラーなど）。
- 情報を要求したプログラムに情報を返します。

リゾルバは、telnet、FTP、ping などのプログラムにコンパイルされるライブラリルーチンのセットです。それらは別々のプロセスではありません。リゾルバはクエリをまとめ、送信し、回答を待つことができます。また、特定の時間内に返答されない場合は、もう一度送信してください (おそらくセカンダリネームサーバに)。これらのタイプのリゾルバは、スタブ・リゾルバと呼ばれます。一部のリゾルバーには、レコードをキャッシュし、存続時間 (TTL) を尊重する機能が追加されています。Windows では、この機能は DNS クライアントサービスを通じて使用できます。「services.msc」コンソールから表示できます。

ネームサーバー

ネームサーバーは通常、ドメインネームスペース (ゾーンと呼ばれる) の特定の部分に関する完全な情報を格納します。ネームサーバーには、そのゾーンに対する権限があると言われます。また、複数のゾーンに対して権限を持つこともできます。

ドメインとゾーンの違いは微妙です。ドメインとは、サブドメインを含むエンティティの完全なセットであり、ゾーンはドメイン内の情報であり、別のネームサーバーに委任されません。ゾーンの例としては `citrix.com`、サブドメイン内の別のネームサーバーにゾーンが委任されている場合、`sales.citrix.com`は別のゾーンです。この場合、プライマリ Citrix ゾーンには `citrix.com`、`it.citrix.com`、および`support.citrix.com`を含めることができます。`sales.citrix.com`が委任されるため、`citrix.com`ネームサーバが権限を持つゾーンの一部ではありません。次の図は、2 つのゾーンを示しています。

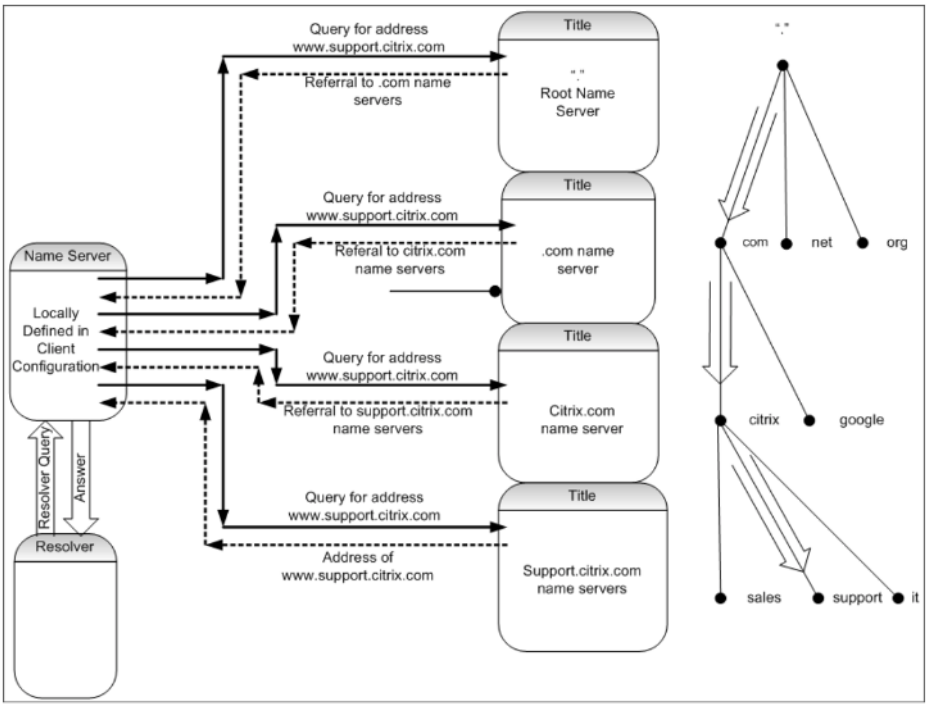


サブドメインを適切に委任するには、サブドメインの権限を別のネームサーバーに割り当てる必要があります。前の例では、ns1.citrix.comにはsales.citrix.com サブドメインに関する情報は含まれていません。代わりに、ns1.sales.citrix.com サブドメインに対して権限のあるネームサーバーへのポインタが含まれます。

ルートネームサーバーとクエリ解決

ルートネームサーバは、第 2 レベルドメインに対して権限を持つすべてのネームサーバの IP アドレスを認識します。ネームサーバーが独自のデータファイルに特定のドメインに関する情報を持っていない場合、最終的に特定のドメインに到達するために、**DNS** ツリー構造の適切なブランチを走査するためにルートサーバーに連絡するだけで済みます。これには、ツリートラバーサルで次の権限のあるネームサーバーを見つけるための複数のネームサーバーへの一連の要求が含まれます。このリクエストは、さらなる解決のために連絡する必要があります。

次の図は、トラバーサル中に要求された名前のキャッシュレコードがないと仮定した、典型的な DNS 要求を示しています。次の例では、Citrix ドメインのモックアップを使用します。



再帰クエリと非再帰クエリ

上記の例では、発生する 2 つのタイプのクエリを示します。

- 再帰クエリ: リゾルバとローカルに設定されたネームサーバー間のクエリは再帰的です。つまり、ネームサーバーはクエリを受信し、クエリが完全に応答されるか、エラーが返されるまでリゾルバに応答しません。ネームサーバがクエリへの参照を受信すると、ネームサーバが最終的に返された回答（IP アドレス）を受け取るまで、ネームサーバは参照に従います。
- 非再帰クエリ: ローカルで構成されたネームサーバーが、後続の権限のあるドメインレベルのネームサーバーに対して行うクエリは、非再帰 (または反復) です。各リクエストは、下位レベルの権限のあるサーバーへの参照、またはクエリに対する応答のいずれかで即座に応答されます（クエリされたネームサーバーがそのデータファイルまたはそのキャッシュに回答が含まれている場合）。

キャッシュ

解決プロセスには関与しており、複数のホストへの小さなリクエストが必要になる可能性がありますが、高速です。DNS 解決の速度を上げる要因の 1 つがキャッシュです。ネームサーバーは、再帰クエリを受信するたびに、最終的に特定の要求に対して適切な権限のあるサーバーに到達するために、他のサーバーと通信しなければならない場合があります。これは、将来の参照のために受け取ったすべての情報を格納します。次のクライアントが、別のホストで同じドメイン内で同様の要求を行う場合、そのドメインに対して権限のあるネームサーバーをすでに認識しており、ルートネームサーバーで起動するのではなく、そこで直接リクエストを送信できます。

キャッシュは、存在しないホストのクエリなど、否定的な応答に対しても発生する可能性があります。この場合、サーバは、ホストが存在しないことを把握するために、要求されたドメインを権限のあるネームサーバに照会してはなりません。時間を節約するために、ネームサーバーはキャッシュをチェックし、ネガティブレコードで応答します。

ネームサーバーはレコードを無期限にキャッシュしないか、IP アドレスを更新することはできません。同期の問題を回避するために、DNS 応答には存続時間 (TTL) が含まれています。このフィールドは、キャッシュがレコードを破棄し、更新されたレコードがあるかどうかを権限のあるネームサーバーで確認する前に、キャッシュがレコードを保存できる時間間隔を示します。レコードが変更されていない場合、TTL を使用すると、GSLB を実行するデバイスからの迅速な動的応答も可能になります。

リソースレコードタイプ

さまざまな RFC は、DNS リソースレコードタイプとその説明の包括的なリストを提供します。次の表に、一般的なリソースレコードタイプを示します。

リソースレコードタイプ	説明	RFC			
A	ホストアドレス	RFC 1035			
NS	権威あるネームサーバー	RFC 1035			
MD	メールの送信先 (廃止-MX を使用)	RFC 1035			
MF	メールフォワーダー (廃止-MX を使用)	RFC 1035			
cmame	エイリアスの正規名	RFC 1035	SOA	権限ゾーンの開始をマーク	RFC 1035
WKS	よく知られているサービス説明	RFC 1035			
PTR	ドメイン名ポインタ	RFC 1035			
hinfo	ホスト情報	RFC 1035			
minfo	メールボックスまたはメールリスト情報	RFC 1035			

リソースレコードタイプ	説明	RFC
mx	Mail Exchange	RFC 1035
txt	テキスト文字列	RFC 1035
AAA	IP6 アドレス	RFC 3596
SRV	サーバ選択	RFC 2782]

GSLB が DNS をサポートする方法

GSLB は、DNS クエリで送信する必要がある IP アドレスを決定するアルゴリズムとプロトコルを使用します。GSLB サイトは地理的に分散されており、Citrix ADC アプライアンスでサービスとして実行されている各サイトには DNS 権限のあるネームサーバーがあります。関係するさまざまなサイトのすべてのネームサーバーは、同じドメインに対して権限を持ちます。各 GSLB ドメインは、委任が設定されるサブドメインです。したがって、GSLB ネームサーバーは権限があり、さまざまな負荷分散アルゴリズムのいずれかを使用して、どの IP アドレスを返すかを決定できます。

委任は、親ドメインのデータベースファイルに GSLB ドメインのネームサーバーレコードを追加し、委任に使用されるネームサーバーのアドレスレコードを追加することによって作成されます。たとえば、www.citrix.comに GSLB を使用する場合は、次のバインド SOA ファイルを使用して、リクエストをwww.citrix.comからネームサーバーに委任できます。Netscaler1 と Netscaler2。

```
; ##############################################################################

@ IN SOA citrix.com. hostmaster.citrix.com. (
1 ; serial
3h ; refresh
1h ; retry
1w ; expire
1h ) ; negative caching TTL
IN NS ns1
IN NS ns2
IN MX 10 mail

ns1   IN A 10.10.10.10
ns2   IN A 10.10.10.20
mail  IN A 10.20.20.50

```



```
### Old Configuration if www was not delegated to a GSLB name server
www IN A 10.20.20.50

### Updated Configuration
Netscaler1 IN A xxx.xxx.xxx.xxx
Netscaler2 IN A yyy.yyy.yyy.yyy
www IN NS Netscaler1.citrix.com.
www IN NS Netscaler2.citrix.com.
###
IN MX 20 mail2
mail2 IN A 10.50.50.20
################################################################################


<!--NeedCopy-->
```

BIND を理解することは、DNS を設定するための要件ではありません。準拠する DNS サーバーの実装には、同等の委任を作成する方法があります。Microsoft DNS サーバーを委任用に構成するには、「[ゾーン委任を作成する](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc785881(v=ws.10)」の手順を使用します。redirectedfrom =MSDN)。

Citrix ADC アプライアンスの GSLB が標準の DNS サービスを使用してトラフィックを分散するのとは異なるのは、Citrix ADC GSLB サイトがメトリック交換プロトコル（MEP）と呼ばれる独自のプロトコルを使用してデータを交換することです。MEP を使用すると、GSLB サイトは他のすべてのサイトに関する情報を維持できます。DNS 要求を受信すると、MEP は GSLB メトリックスを考慮して、次のような情報を決定します。

- 現在の接続数が最も少ないサイト
- ラウンドトリップ時間 (RTT) に基づいてリクエストを送信した LDNS サーバーに最も近いサイト。

使用できる負荷分散アルゴリズムはいくつかありますが、GSLB は、参加サイトのメトリックに基づいてどのアドレスを送信する必要があるかをネームサーバー（Citrix ADC アプライアンス上でホストされている）に知らせる DNS です。

GSLB が提供するその他の利点は、永続性（またはサイトアフィニティ）を維持できることです。着信 DNS クエリに対する応答をソース IP アドレスと比較して、そのアドレスが最近特定のサイトに誘導されたかどうかを判断できます。その場合は、クライアントセッションが確実に維持されるように、同じアドレスが DNS 応答に送信されます。

別の形式の永続性は、HTTP リダイレクト、または HTTP プロキシを使用してサイトレベルで取得されます。これらの形式の永続性は、DNS 応答が発生した後に発生します。したがって、リクエストを別の参加サイトに誘導する Cookie を含むサイトで HTTP リクエストを受け取った場合、リダイレクトで応答するか、リクエストを適切なサイトにプロキシすることができます。

メトリック交換プロトコル

メトリック交換プロトコル (MEP) は、GSLB 計算で使用されるデータをサイト間で共有するために使用されます。MEP 接続を使用して、3 種類のデータを交換します。これらの接続は、TCP ポート 3011 を介してセキュアである必要はなく、TCP ポート 3009 経由の SSL を使用してセキュアにすることもできます。

以下の 3 種類のデータが交換され、独自の間隔と交換方法がある。

- サイトメトリック交換**:** これはポーリング交換モデルです。たとえば、site1 に site2 サービスの構成がある場合、第 2 サイト 1 は site2 に GSLB サービスのステータスを要求します。Site2 は、状態およびその他のロードの詳細で応答します。
- ネットワークメトリック交換：これは、動的近接負荷分散アルゴリズムで使用される LDNS RTT 情報交換です。これはプッシュ交換モデルです。5 秒ごとに、各サイトはそのデータを他の参加サイトにプッシュします。
- 永続性交換**:** これは SOURCEIP 永続性交換用です。これはプッシュ交換モデルでもあります。5 秒ごとに、各サイトはそのデータを他の参加サイトにプッシュします。

デフォルトでは、サイトサービスはポーリング情報のみに基づいて MEP を介して監視されます。モニタ間隔に基づいてモニタをバインドすると、状態は更新され、それに応じてモニタリング間隔を設定することで更新の頻度を制御できます。

GSLB 展開のアップグレードに関する推奨事項

October 7, 2021

このセクションでは、さまざまな GSLB セットアップで GSLB ノードをアップグレードする必要がある順序に関する推奨事項について説明します。また、いくつかの FAQ にも対処しています。

注： GSLB の同期が開始される Citrix ADC アプライアンスは、「メインサイト」と呼ばれ、構成がコピーされる GSLB サイトは「下位サイト」と呼ばれます。

アップグレードプロセスを開始する前に、次のトピックで説明されている前提条件をお読みください。

- はじめに
- ハイアベイラビリティペアをアップグレードします。
- クラスターをアップグレードします。

GSLB セットアップをアップグレードするときの注意点

- HA セットアップでは、まず下位サイトをアップグレードし、次にメインサイトをアップグレードします。
- HA セットアップでは、古いビルドプライマリノードから新しいビルドセカンダリノードにサービス状態が伝播しないことがあります。ただし、ビルドのバージョンが異なるが HA バージョンが同じ場合、サービスの状態は引き続き伝播する可能性があります。

- GSLB がクラスタ内で構成されている場合は、まず所有者以外のノードをアップグレードしてから、所有者ノードをアップグレードします。クラスタ内に 1 つのサイトまたは複数のサイトがある場合は、各サイトで同じアップグレード手順に従います。
- 新しい GSLB 機能は、すべてのノードを新しいビルドにアップグレードした後にのみ有効にします。
- すべての GSLB ノードを最新のビルドにアップグレードします。GSLB ノードの一部が古いバージョンを使用し、一部の GSLB ノードが新しいバージョンにアップグレードされている場合、使用可能な機能に機能的影響はありません。

よくある質問

- インスタンスが異なるソフトウェアバージョンを実行する場合、**GSLB** サービスの状態は伝播されますか。

  GSLB MEP は、インスタンスが異なるバージョンで実行され、GSLB サービスの状態が GSLB サイト間で伝播される場合に機能します。アップグレード後にインスタンスが異なるバージョンを実行する場合、MEP 通信には影響しません。

- アップグレード中に構成の変更を行うことを推奨しますか。

  In a GSLB setup, when a main site is being upgraded, it is not recommended to do configuration changes on any other GSLB nodes.

関連リソース

次のリソースは、Citrix ADM を使用した Citrix ADC インスタンスのアップグレードに関する情報を提供します。

- Citrix ADM サービスが Citrix ADC アップグレードをより簡単にサポートする 10 の方法
- Citrix ADM サービスを使用して Citrix ADC インスタンスをアップグレードする
- Citrix ADM ソフトウェアを使用して Citrix ADC インスタンスをアップグレードする

ユースケース: ドメイン名ベースの **AutoScale** サービスグループの展開

October 7, 2021

> ヒント
>
> GSLB サービスグループの詳細については、「GSLB サービスグループの設定」を参照してください。

導入シナリオ

2 つのデータセンターが、シドニーと北バージニアの 2 つの AWS リージョンにデプロイされています。別のデータセンターが Azure にデプロイされています。各 AWS リージョンの AWS ELB は、アプリケーションサーバーの負荷

分散に使用されます。ALB は、Azure がアプリケーションサーバーを負荷分散するために使用されます。Citrix ADC アプライアンスは、GSLB ドメイン名ベースの AutoScale サービスグループを使用して、ELB および ALB の GSLB 用に構成されます。

> 重要
>
> AWS で必要なセキュリティグループを設定し、GSLB インスタンスにアタッチする必要があります。ポート 53 は、セキュリティグループのインバウンドルールとアウトバウンドルールで許可する必要があります。また、MEP 通信用のポート（セキュア MEP 設定によっては 3009 または 3011）が開いている必要があります。アプリケーションモニタリングでは、対応するポートがセキュリティグループのアウトバウンドルールで許可されている必要があります。

上記のデプロイメントシナリオと対応する CLI コマンドの設定手順は次のとおりです。

1. データセンターを作成します (GSLB サイトで表されます)。

   ```
   add gslb site aws-sydney 192.0.2.2

   add gslb site aws-nvirginia 198.51.100.111

   add gslb site alb-southindia 203.0.113.6
   ```

2. GSLB ノードが追加される DNS Gateway IP アドレスを持つネームサーバーを追加します。これは、すべてのデータセンターで行う必要があります。

   ```
   add dns nameServer 8.8.8.8
   ```

3. ELB と ALB のサーバーを追加します。

   ```
   add server aws-sydney_server lb-sydney-1052691850.ap-southeast-2.elb.
   amazonaws.com

   add server aws-nvirginia_server LB-nvirginia-860559595.us-east-1.elb.
   amazonaws.com

   add server alb-southindia_server alb.southindia.cloudapp.azure.com
   ```

4. ELB と ALB ごとに GSLB AutoScale サービスグループを追加し、各サーバーをそれぞれのサービスグループにバインドします。

   ```
   add gslb serviceGroup aws-nvirginia_sg HTTP -autoScale DNS -siteName
   aws-nvirginia

   add gslb serviceGroup aws-sydney_sg HTTP -autoScale DNS -siteName aws-
   sydney

   add gslb serviceGroup alb-southindia_sg HTTP -autoScale DNS -siteName
   alb-southindia

   bind gslb serviceGroup aws-nvirginia_sg aws-nvirginia_server 80

   bind gslb serviceGroup aws-sydney_sg aws-sydney_server 80
   ```



```
bind gslb serviceGroup alb-southindia_sg alb-southindia_server 80
```

5. GSLB 仮想サーバーを追加し、アプリケーションドメインとサービスグループをこの仮想サーバーにバインドします。

```
add gslb vserver gv1 HTTP

bind gslb vserver gv1 -serviceGroupName aws-nvirginia_sg

bind gslb vserver gv1 -serviceGroupName aws-sydney_sg

bind gslb vserver gv1 -serviceGroupName alb-southindia_sg
```

ユースケース:IP アドレスベースの GSLB サービスグループの展開

October 7, 2021

> ヒント
>
> GSLB サービスグループの詳細については、GSLB サービスグループの設定を参照してください。

導入シナリオ

同じアプリケーションサーバーでホストされている複数のアプリケーションがある場合、GSLB はこれらのアプリケーションをプローブして、アプリケーションが応答しているかどうかを確認する必要があります。アプリケーションが応答しない場合は、アプリケーションが UP のサーバーにユーザーを誘導する必要があります。また、アプリケーションの 1 つが DOWN の場合、他のアプリケーションが UP であるため、サーバーに DOWN とマークしないでください。

次の例では、複数のアプリケーション（HTTPS）が各 GSLB サイト内の 1 つのサーバーでホストされているため、これらのアプリケーションはそれぞれのサイトの 1 つの IP アドレスに解決されます。

GSLB サービスグループを使用すると、IP アドレスとポートを複数のサービスグループにバインドし、各サービスグループが異なるアプリケーションを表している同じサーバーを持つことができます。

アプリケーション固有のモニターは、アプリケーションが DOWN の場合、サービスグループを DOWN としてマークするサービスグループにバインドされます。したがって、アプリケーションが DOWN のたびに、そのアプリケーションだけがセットアップから取り出され、サーバーからは取り出されません。

```
1 ```
2 add gslb serviceGroup app1_site1 HTTP -maxClient 0 -cip DISABLED -
      cltTimeout 180 -svrTimeout 360 -siteName s1
3
```



```
add gslb serviceGroup app2_site1 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s1

add gslb serviceGroup app1_site2 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s2

add gslb serviceGroup app2_site2 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s2

add lb monitor http_app2 HTTP -respCode 200 -httpRequest ”GET /testsite
    /app2.html”

add lb monitor http_app1 HTTP -respCode 200 -httpRequest ”GET /testsite
    /app1.html”

bind gslb serviceGroup app1_site1 192.0.2.140 80

bind gslb serviceGroup app1_site1 -monitorName http_app1

bind gslb serviceGroup app2_site1 192.0.2.140 80

bind gslb serviceGroup app2_site1 -monitorName http_app2

bind gslb serviceGroup app1_site2 192.0.2.142 80

bind gslb serviceGroup app1_site2 -monitorName http_app1

bind gslb serviceGroup app2_site2 192.0.2.142 80

bind gslb serviceGroup app2_site2 -monitorName http_app2
<!--NeedCopy--> ```
```

ハウツー記事

October 7, 2021

GSLB のハウツー記事には、GSLB 構成のカスタマイズ、永続的な接続の構成、災害復旧など、重要な GSLB 構成の一部に関する情報が含まれています。

GSLB 設定のカスタマイズ

持続的接続の構成

クライアント接続の管理

GSLB 設定のカスタマイズ

October 7, 2021

基本的な GSLB 設定が動作可能になったら、GSLB サービスの帯域幅の変更、CNAME ベースの GSLB サービス、スタティック近接、ダイナミック RTT、永続接続、またはサービスの動的重みの設定、または GSLB メソッドの変更によってカスタマイズできます。

GSLB サービスのモニタリングを構成して、その状態を判断することもできます。

これらの設定は、ネットワーク配置と、サーバーに接続する予定のクライアントの種類によって異なります。

GSLB サービスの最大接続または最大帯域幅の変更

仮想サーバーを表す GSLB サービスのクライアントの最大数または最大帯域幅を構成することで、負荷分散またはコンテンツスイッチング仮想サーバーに同時に接続できる新しいクライアントの数を制限できます。

コマンドラインインターフェイスを使用して **GSLB** サービスの最大クライアントまたは帯域幅を変更するには

コマンドプロンプトで次のコマンドを入力して、GSLB サービスのクライアント接続の最大数または最大帯域幅を変更し、構成を確認します。

```
set gslb service <serviceName> [-maxClients <positive_integer>] [-
    maxBandwidth <positive_integer>]
show gslb service <serviceName>
<!--NeedCopy-->
```


例:

```
set glsb service Service-GSLB-1 -maxBandwidth 100 -maxClients 100
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **GSLB** サービスの最大クライアントまたは帯域幅を変更するには

1. **Traffic Management** > **GSLB** > **Services** に移動してサービスをダブルクリックします。
2. [その他の設定] セクションをクリックし、次のパラメータを設定します。
   - 最大クライアント数：最大クライアント数
   - 最大帯域幅：最大帯域幅

CNAME ベースの **GSLB** サービスの作成

GSLB サービスを設定するには、サーバーの IP アドレスまたはサーバーの正規名を使用できます。1 つの IP アドレスから複数のサービス (FTP サーバーや Web サーバーなど、それぞれが異なるポートで実行される) を実行する場合や、同じ物理ホスト上の同じポート上で異なる名前を持つ複数の HTTP サービスを実行する場合は、サービスに正規名 (CNAMES) を使用できます。

たとえば、同じドメイン example.com にある FTP サービスおよび HTTP サービス用に、ftp.example.com と www.example.com という 2 つのエントリを DNS に持つことができます。CNAME ベースの GSLB サービスは、マルチレベルドメインリゾルバ設定またはマルチレベルドメイン負荷分散に役立ちます。CNAME ベースの GSLB サービスを構成すると、物理サーバーの IP アドレスが変更される可能性が高い場合にも役立ちます。

GSLB ドメインに対して CNAME ベースの GSLB サービスを構成する場合、GSLB ドメインに対してクエリが送信されると、Citrix ADC アプライアンスは IP アドレスではなく CNAME を提供します。この CNAME レコードの A レコードが構成されていない場合、クライアントは CNAME ドメインに IP アドレスを問い合わせる必要があります。この CNAME レコードの A レコードが構成されている場合、Citrix ADC アプライアンスは CNAME に対応する A レコード（IP アドレス）を提供します。Citrix ADC アプライアンスは、GSLB 方式で決定された DNS クエリの最終的な解決を処理します。CNAME レコードは、別の Citrix ADC アプライアンスまたはサードパーティ製システムで保持できます。

IP アドレスベースの GSLB サービスでは、サービスの状態は、それが表すサーバーの状態によって決まります。ただし、CNAME ベースの GSLB サービスの状態はデフォルトで UP に設定されています。仮想サーバーの IP（VIP）アドレスまたはメトリック交換プロトコル（MEP）は、その状態の決定には使用されません。デスクトップベースのモニターが CNAME ベースの GSLB サービスにバインドされている場合、サービスの状態はモニタープローブの結果に基づいて決定されます。

CNAME ベースの GSLB サービスは、DNS レコードの種類が CNAME である GSLB 仮想サーバーにのみバインドできます。また、Citrix ADC アプライアンスは、特定の CNAME エントリを持つ GSLB サービスを最大で 1 つ含めることができます。

CNAME ベースの GSLB サービスでサポートされる機能の一部を次に示します。

- GSLB ポリシーベースのサイトアフィニティがサポートされ、CNAME が優先される場所として使用されます。
- 送信元 IP パーシステンスがサポートされています。永続性エントリには、選択したサービスの IP アドレスとポートの代わりに CNAME 情報が含まれます。

CNAME ベースの GSLB サービスの制限事項を次に示します。

- CNAME によって参照されるサービスは、任意のサードパーティの場所に存在することができるため、サイトの永続性はサポートされていません。
- 1 つのドメインが複数の CNAME エントリを持つことができないため、複数の IP アドレス応答はサポートされません。
- ソース IP ハッシュとラウンドロビンは、サポートされている唯一の負荷分散方式です。CNAME が IP アドレスに関連付けられておらず、静的な近接は IP アドレスに従ってのみ維持できるため、静的近接方式はサポートされません。

注：Empty-Down-Response 機能は、CNAME ベースの GSLB サービスをバインドする GSLB 仮想サーバーで有効にする必要があります。Empty-Down-Response 機能を有効にすると、GSLB 仮想サーバーがダウンまたは無効になると、この仮想サーバーにバインドされたドメインの DNS クエリへの応答に、エラーではなく IP アドレスのない空のレコードが含まれますコード。

コマンドラインインターフェイスを使用して **CNAME** ベースの **GSLB** サービスを作成するには

コマンドプロンプトで入力します。

```
1 add gslb service <serviceName> -cnameEntry <string> -siteName <string>
2 <!--NeedCopy-->
```

例**:**

```
1 add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -
     siteName Site-GSLB-East-Coast
2 add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -
     siteName Site-GSLB-West-Coast
3 <!--NeedCopy-->
```

構成ユーティリティを使用して **CNAME** ベースの **GSLB** サービスを作成するには

1. **Traffic Management** > **GSLB** > **Services** に移動します。
2. サービスを作成し、[タイプ] を **[**正規名ベース**]** に設定します。

GSLB でのサービス外の状態の移行 **(TROFS)** の構成

サービスがバインドされている GSLB 仮想サーバー上でパーシステンスを設定すると、サービスは無効になった後もクライアントからのリクエストを処理し続け、永続性を維持するためだけに新しいリクエストまたは接続を受け入れます。構成された期間（正常なシャットダウン期間）が経過すると、新しい要求や接続はサービスに送信されず、既存の接続はすべて閉じられます。

サービスを無効にする場合は、delay 引数を使用して、正常なシャットダウン期間を秒単位で指定できます。正常なシャットダウン期間中に、サービスが仮想サーバにバインドされている場合、その状態は Out of Service と表示されます。

サービスのダイナミックウェイトの設定

一般的なネットワークでは、他のサーバよりもトラフィックの容量が大きいサーバがあります。ただし、通常の負荷分散構成では、異なるサービスが異なる容量のサーバーを表している場合でも、負荷はすべてのサービスに均等に分散されます。

GSLB リソースを最適化するには、GSLB 仮想サーバーで動的重みを設定できます。動的重み付けは、仮想サーバにバインドされたサービスの合計数、または仮想サーバにバインドされた個々のサービスの重みの合計に基づきます。トラフィック分散は、サービスに設定された重みに基づいて行われます。

GSLB 仮想サーバーで動的重みが設定されている場合、要求は負荷分散方式、GSLB サービスの加重、および動的重みに従って分散されます。GSLB サービスの重量と動的重量の積は、累積重量として知られています。したがって、GSLB 仮想サーバーで動的重みが設定されている場合、要求は負荷分散方式と累積重みに基づいて分散されます。

仮想サーバの動的重みが無効になっている場合、数値は 1 に設定されます。これにより、累積ウェイトは常にゼロ以外の整数になります。

動的重み付けは、負荷分散仮想サーバにバインドされているアクティブなサービスの合計数またはサービスに割り当てられた重みに基づいて設定できます。

ドメイン用に構成された 2 つの GSLB サイトで構成され、各サイトにはクライアントにサービスを提供できる 2 つのサービスがあるとします。いずれかのサイトのサービスがダウンした場合、そのサイトのもう一方のサーバは、もう一方のサイトのサービスの 2 倍のトラフィックを処理する必要があります。動的な重み付けがアクティブなサービスの数に基づいている場合、両方のサービスがアクティブなサイトの重みが 2 倍になり、1 つのサービスがダウンしているため、トラフィックが 2 倍になります。

または、最初のサイトのサービスが、2 番目のサイトのサーバーの 2 倍のパワーのサーバーを表す構成を検討します。動的重みが、サービスに割り当てられた重みに基づいている場合、最初のサイトに 2 倍の数のトラフィックを送信できます。

注: 負荷分散サービスへの重みの割り当ての詳細については、サービスへの重みの割り当てを参照してください。

動的加重の計算方法を示す図として、GSLB サービスがバインドされている GSLB 仮想サーバーについて考えます。GSLB サービスは、順番に 2 つのサービスがバインドされている負荷分散仮想サーバーを表します。GSLB サービス

に割り当てられる重みは 3 です。2 つのサービスに割り当てられる重みは、それぞれ 1 と 2 です。この例では、ダイナミックウェイトがに設定されている場合:

- **Disabled**:GSLB 仮想サーバーの累積加重は、動的加重 (無効 = 1) と GSLB サービスの加重 (3) の積であるため、累積加重は 3 になります。
- **SERVICECOUNT**：カウントは、GSLB サービス（2）に対応する負荷分散仮想サーバーにバインドされたサービス数の合計です。累積加重は、動的加重（2）と GSLB サービス（3）の加重積（6）の積です。
- **SERVICEWEIGHT**：動的加重は、GSLB サービス（3）に対応する負荷分散仮想サーバーにバインドされたサービスの加重の合計です。累積加重は、動的加重（3）と GSLB サービス（3）の加重の積です。

注：コンテンツスイッチング仮想サーバーが設定されている場合、動的重みは適用されません。

コマンドラインインターフェイスを使用して動的重みを使用するように **GSLB** 仮想サーバーを設定するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -dynamicWeight SERVICECOUNT | SERVICEWEIGHT
<!--NeedCopy-->
```

例**:**

```
set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT
<!--NeedCopy-->
```

構成ユーティリティを使用して動的重みを使用するように **GSLB** 仮想サーバーを設定するには

1. 「トラフィック管理」>「GSLB」>「仮想サーバー」に移動し、メソッドを変更する GSLB 仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. [方法] セクションをクリックし、[動的重量] ドロップダウンリストから [**SERVICECOUNT**] または [**SERVICEWEIGHT**] を選択します。

GSLB で永続性を構成する方法

October 7, 2021

永続性により、特定のドメイン名に対する一連のクライアント要求が、負荷分散されるのではなく、同じデータセンターに送信されます。特定のドメインに対してパーシステンスが設定されている場合は、設定された GSLB メソッドよりも優先されます。永続性は、クライアントトランザクションに関連する情報が、最初のリクエストを処理した

インスタンスにローカルに保存されるデプロイメントに使用できます。たとえば、ショッピングカートを使用する電子商取引の展開では、サーバーはトランザクションを追跡するために接続の状態を維持する必要があります。Citrix ADC アプライアンスは、クライアント要求を処理するデータセンターを選択します。永続性を有効にすると、後続のすべてのドメインネームシステム（DNS）要求に対して、選択したデータセンターの同じ IP アドレスが転送されます。永続セッションがダウンしているデータセンターを指している場合、Citrix ADC アプライアンスは構成済みの GSLB メソッドを使用して新しいデータセンターを選択します。その後、クライアントからの後続の要求に対して永続的になります。
GSLB で永続化するには、すべてのデータセンターの GSLB 仮想サーバーで同じ永続性識別子のセット（persistID）を構成する必要があります。GSLB モジュールは、永続性識別子を使用して GSLB 仮想サーバーを一意に識別します。GSLB 仮想サーバーでソース IP 永続性が有効になっている場合、永続性セッションもメトリック交換の一部として交換されます。Citrix ADC アプライアンスがサイト間の永続性をサポートするには、参加しているすべての GSLB サイトで永続性関連の構成を行う必要があります。ステートフルアプリケーションには GSLB での永続性を推奨します。これにより、クライアントは後続の要求のために同じアプリケーションインスタンスに再接続する必要があります。

次の方法で GSLB の永続性を実現できます。

- GSLB 仮想サーバーでの永続性
- GSLB サービスでのサイトの永続性

GSLB 仮想サーバーでの永続性

GSLB 仮想サーバーでの永続性は、DNS 要求中に使用されます。DNS 要求の送信元 IP アドレスは、クライアントとデータセンター間の永続セッションを作成するために使用されます。DNS クライアントは通常、ローカル DNS (LDNS）または DNS ゲートウェイであり、その背後にあるクライアントのセットをプロキシします（ISP 内)。GSLB 仮想サーバーでの永続性は、アプリケーションプロトコルに依存しません。
一般に、複数の DNS ゲートウェイまたはローカルドメインネームサーバー（LDNS）がクライアントネットワークに構成されます。後続の DNS 要求では、ADC アプライアンスへの接続に使用されるアップストリーム LDNS デバイスに関係なく、クライアントは以前の要求を処理した同じデータセンターに永続化できるため、適切な永続性マスクを構成することをお勧めします。LDNS IP アドレスの永続化セッションが作成された後、その LDNS を使用して接続するすべてのエンドクライアントには、同じデータセンター IP アドレスが割り当てられます。

GSLB サービスでのサイトの永続性

サイトの永続性は、アプリケーション要求の処理中に有効になります。永続性は HTTPCookie を使用して実現されるため、サイトの永続性は HTTP および HTTPS トラフィックに対してのみ機能します。Cookie は HTTP クライアント（ブラウザー）で維持されるため、DNS ゲートウェイの背後にあるクライアントを可視化できます。Cookie を使用してクライアントの永続性を実現する場合、着信クライアントごとに ADC アプライアンスでリソースが消費されることはありません。GSLB サービスを遅延時間で DOWN にすると、サービスはアウトオブサービス（TROFS）状態に移行します。サービスが UP または TROFS 状態である限り、永続性がサポートされます。つまり、サービスが TROFS とマークされた後、同じクライアントが同じサービスの要求を指定された遅延時間内に送信すると、同じ GSLB サイト (データセンター) が要求を処理します。

エイリアスを介してアプリケーションにアクセスする場合は、CNAME レコードが Citrix ADC アプライアンスでも構成されていることを確認してください。親子トポロジでは、エイリアスを介してアプリケーションにアクセスすると、サイトの永続性が機能しません。

> 注
>
> 接続プロキシがサイトの永続化方法として指定されていて、LB 仮想サーバーでも永続性を構成する場合は、ソース IP の永続性はお勧めしません。接続がプロキシされると、クライアントの実際の IP アドレスではなく、ADC アプライアンスが所有する IP アドレスが使用されます。
> HTTP (S) 要求のソース IP を使用してクライアントを識別することのない適切な永続性を構成します (Cookie の永続性やルールベースの永続性など)。

送信元 **IP** アドレスに基づく永続性の設定

GSLB 仮想サーバーで送信元 IP 永続性が構成されている場合、DNS 要求の送信元 IP アドレスに対して永続化セッションが作成されます。拡張クライアントサブネット（ECS）機能に応じて、DNS 要求の送信元 IP アドレスは次のいずれかから取得されます。

- 着信 DNS 要求パケットの IP ヘッダーの送信元 IP
- DNS 要求の ECS オプション ECS の詳細については、「グローバルサーバー負荷分散に EDNS0 クライアントサブネットオプションを使用する」を参照してください。

クライアントの永続セッションは、永続タイムアウトまで続きます。タイムアウト期間が終了すると、既存の永続セッションがクリアされます。その後の要求では、新しい GSLB 決定が行われ、別の GSLB サービス IP アドレスが選択される場合があります。
GSLB 仮想サーバーでのソース IP の永続性と、GSLB サービスでのサイトの永続性は互いに補完し合っています。GSLB 仮想サーバーでソース IP の永続性が無効になっている場合、GSLB 仮想サーバーは DNS が解決を試みるたびに異なる GSLB サービスを選択します。また、クライアントは別の GSLB サービスに接続し、アプリケーション要求プロキシを受信するデータセンターは、最初にクライアントにサービスを提供したデータセンターへの接続をプロキシします。これにより、待ち時間が増える可能性があります。したがって、GSLB 仮想サーバーでソース IP の永続性を有効にすることで、アプリケーション要求に対するこのような頻繁な複数のホップを回避できます。ソース IP 永続性セッションが期限切れになり、その後クライアントが再接続した場合、サイト永続性はクライアントを、最初にクライアントにサービスを提供していたデータセンターに接続し直します。また、クライアントが構成された永続性マスクの範囲内にない DNS ゲートウェイを介して接続し直す場合、サイトの永続性は、クライアントが最初の要求を処理したデータセンターに固執するのにも役立ちます。

CLI を使用して送信元 **IP** アドレスに基づいて永続性を構成するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -persistenceType (SOURCEIP|NONE) -persistenceId
    <positive_integer> [-persistMask <netmask>] - [timeout <mins>]
```



```
<!--NeedCopy-->
```

例:

```
set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -
    persistenceId 23 -persistMask 255.255.255.255  - timeout 2
<!--NeedCopy-->
```

GUI を使用して送信元 **IP** アドレスに基づいて永続性を構成するには

1. トラフィック管理 > **GSLB** > 仮想サーバーに移動して、メソッドを変更する GSLB 仮想サーバーをダブルクリックします（たとえば、vserver-GSLB-1）。
2. 「持続性」セクションをクリックし、「持続性」ドロップダウンリストから「**SOURCEIP** 」を選択し、次のパラメータを設定します。
   - 持続性 ID：持続性 ID
   - タイムアウト：タイムアウト
   - IPv4 ネットマスクまたは IPv6 マスクの長さ：永続マスク

HTTPCookie に基づいてサイトの永続性を構成する

サイトの永続性は、HTTP Cookie（「サイト Cookie」と呼ばれる）を使用してクライアントを同じサーバーに再接続することで実現されます。GSLB アプライアンスが選択した GSLB サイトの IP アドレスを送信してクライアント DNS 要求に応答すると、クライアントはその GSLB サイトに HTTP 要求を送信します。その GSLB サイトのアプリケーションエンドポイントは HTTP ヘッダーにサイト Cookie を追加し、サイトの永続性が有効になります。
クライアントキャッシュの有効期限が切れた後にクライアントが DNS クエリを送信すると、DNS 要求が別の GSLB サイトに送信される可能性があります。新しい GSLB サイトは、クライアント要求ヘッダーにあるサイト Cookie を使用して永続性を実装します。サイト永続化機能は、次の条件下でアクティブになります。

- ホストヘッダーのドメイン名が GSLB ドメインの 1 つと一致する場合
- アプリケーショントラフィックを受信する仮想サーバーを表す GSLB サービスでサイトの永続性が有効になっている場合。

サイトクッキーには、クライアントが永続的な接続を持つ選択された GSLB サービスに関する情報が含まれています。Cookie が指す GSLB サービスがダウンしているか、GLSB 構成から削除されている場合、トラフィックを受信する仮想サーバーは引き続きトラフィックを処理します。クッキーの有効期限は、Citrix ADC アプライアンスで設定されたクッキーのタイムアウトに基づいています。仮想サーバー名がすべてのサイトで同一でない場合は、永続識別子を使用する必要があります。挿入されたクッキーは、RFC 2109 に準拠しています。

Citrix ADC は、次の 2 種類のサイト永続性をサポートしています。

- 接続プロキシ
- HTTP リダイレクト

接続プロキシ

サイト永続性の接続プロキシモードでは、後続のアプリケーション要求を受信するデータセンターは、接続を確立するために次のタスクを実行します。

1. サイト Cookie を挿入した GSLB サイトへの接続を作成します。
2. クライアント要求を元のサイトにプロキシします。

   注:

   プロキシサーバーは、次の詳細を使用して元のサイトとの接続を確立します。

   ```
   1 – 新しいサイトのSNIPは送信元IPアドレスです。
   ```

   - 元のサイトの GSLB サービスのパブリック IP アドレスが宛先 IP アドレスです。
   - エフェメラルポートは送信元ポートであり、GSLB サービスポートは宛先ポートです。
   - GSLB サービスタイプに応じて、HTTP または HTTPS プロトコルのいずれかを使用します。
3. 元の GSLB サイトから応答を受け取ります。
4. その応答をクライアントに中継します。
5. 接続を閉じます。

HTTP リダイレクト

GSLB 設定が HTTP リダイレクト永続性を使用する場合、新しいサイトは Cookie を最初に挿入したサイトに要求をリダイレクトします。リダイレクト URL のドメイン名はサイトドメインです。Cookie と SSL 証明書の両方が GSLB ドメインとサイトドメインの両方に適用可能であることを確認してください。GSLB とサイトドメインの両方に Cookie を適用するには、Cookie ドメインがサイトから GSLB ドメインである必要があります。SSL 証明書を GSLB とサイトドメインの両方に適用するには、SSL 仮想サーバーにバインドされた証明書がワイルドカード証明書である必要があります。

接続プロキシは、次の条件が満たされた場合に発生します。

- GSLB に参加しているドメインに対してリクエストが送信されます。ドメインは URL/Host ヘッダーから取得されます。
- ローカル GSLB サービスで接続プロキシが有効になっています。
- 要求には、アクティブなリモート GSLB サービスの IP アドレスを含む有効な Cookie が含まれています。

> 注
>
> GSLB 親子構成では、子サイトで GSLB サービスが構成されていない場合でも、接続プロキシは意図したとおりに機能します。ただし、クライアント認証、クライアント IP アドレスの挿入、またはその他の SSL 固有の要件などの追加の構成がある場合は、サイトに明示的な GSLB サービスを追加し、それに応じて構成する必要が

> あります。
>
> 親子トポロジの詳細については、「MEP プロトコルを使用した親子トポロジの展開」を参照してください。

CLI を使用して **HTTPCookie** に基づいて永続性を設定するには

コマンドプロンプトで入力します。

```
set gslb service <serviceName> -sitePersistence (ConnectionProxy [-
    sitePrefix <prefix>] | HTTPredirect -sitePrefix <prefix>)
<!--NeedCopy-->
```

例:

```
set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -
    sitePrefix vserver-GSLB-1
<!--NeedCopy-->
```

GUI を使用して **Cookie** に基づいて永続性を設定するには

1. **Traffic Management** > **GSLB** > **Services** に移動し、サイト永続性を設定するサービス（service-GSLB-1 など）を選択します。
2. [サイトの永続性] セクションをクリックし、Cookie に基づいて永続性を設定します。

クライアント接続の管理

October 7, 2021

クライアント接続の管理を容易にするために、仮想サーバーへの接続の遅延クリーンアップを有効にできます。その後、DNS ポリシーを設定して、ローカル DNS トラフィックを管理できます。

仮想サーバー接続の遅延クリーンアップを有効にする

仮想サーバーの状態は、バインドされているサービスの状態によって異なり、各サービスの状態は、バインドされているモニターによって異なります。サーバが低速またはダウンしている場合は、モニタリングプローブがタイムアウトし、サーバを表すサービスが DOWN としてマークされます。仮想サーバは、バインドされているすべてのサービスが DOWN としてマークされている場合にのみ、DOWN としてマークされます。サービスおよび仮想サーバは、ダ

ウン時にすべての接続を終了するか、または接続の通過を許可するように設定できます。後者の設定は、サーバーが低速であるためにサービスが DOWN としてマークされる状況のためのものです。

ダウン状態のフラッシュオプションを構成すると、Citrix ADC アプライアンスは、ダウンしている GSLB サービスへの接続の遅延クリーンアップを実行します。

コマンドラインインターフェイスを使用して仮想サーバー接続の遅延クリーンアップを有効にするには

コマンドプロンプトで次のコマンドを入力して、遅延接続クリーンアップを構成し、構成を確認します。

```
set gslb service <name> -downStateFlush (ENABLED | DISABLED)
show gslb service <name>
<!--NeedCopy-->
```

例**:**

```
set gslb service Service-GSLB-1 -downStateFlush ENABLED
Done

show gslb service Service-GSLB-1
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバー接続の遅延クリーンアップを有効にするには

1. **Traffic Management** > **GSLB** > **Services** に移動してサービスをダブルクリックします。
2. [その他の設定] セクションをクリックし、[ダウン状態のフラッシュ] オプションを選択します。

DNS ポリシーを使用してローカル **DNS** トラフィックを管理する

DNS ポリシーを使用して、ローカル DNS リゾルバーまたはネットワークの IP アドレスから事前定義されたターゲット GSLB サイトにトラフィックを送信することで、サイトアフィニティを実装できます。これは、DNS 式を使用して DNS ポリシーを作成し、Citrix ADC アプライアンス上でグローバルにポリシーをバインドすることによって構成されます。

ドメインネームシステム

Citrix ADC アプライアンスには、ドメイン固有のアクションを構成するために使用できる事前定義された DNS 式が用意されています。このようなアクションは、たとえば、特定のリクエストの削除、特定のドメインの特定のビューの選択、特定のリクエストの特定のロケーションへのリダイレクトなどが可能です。

これらの DNS 式（ルールとも呼ばれる）を組み合わせて DNS ポリシーを作成し、Citrix ADC アプライアンスでグローバルにバインドします。

Citrix ADC アプライアンスで使用可能な事前定義された DNS 修飾子の一覧を次に示します。

- CLIENT.UDP.DNS.DOMAIN.EQ(“domainname”)
- CLIENT.UDP.DNS.IS_AREC
- CLIENT.UDP.DNS.IS_AAAAREC
- CLIENT.UDP.DNS.IS_SRVREC
- CLIENT.UDP.DNS.IS_MXREC
- CLIENT.UDP.DNS.IS_SOAREC
- CLIENT.UDP.DNS.IS_PTRREC
- CLIENT.UDP.DNS.IS_CNAME
- CLIENT.UDP.DNS.IS_NSREC
- CLIENT.UDP.DNS.IS_ANYREC

CLIENT.UDP.DNS.DOMAIN DNS 式は、文字列式で使用できます。式の一部としてドメイン名を使用する場合は、ピリオド (.) で終わる必要があります。たとえば、CLIENT.UDP.DNS.DOMAIN.ENDSWITH(“abc.com.”)。

構成ユーティリティを使用して式を作成するには

1. 「式」(Expression) テキストボックスの横にあるアイコンをクリックします。[追加] をクリックします。([Flow Type] ドロップダウンリストボックスと [Protocol] ドロップダウンリストボックスは空のままにします)。ルールを作成する手順は、次のとおりです。
2. 「Qualifier」ボックスで、修飾子（「LOCATION」など）を選択します。
3. 「演算子」ボックスで、演算子 (== など) を選択します。
4. [値] ボックスに、値 (たとえば、アジア、日本...) を入力します。
5. ［OK］をクリックします。[作成] をクリックし、[閉じる] をクリックします。ルールが作成されます。
6. ［OK］をクリックします。

DNS アクションを構成する

DNS ポリシーには、ポリシールールが TRUE と評価されたときに実行される DNS アクションの名前が含まれます。DNS アクションでは、次のいずれかを実行できます。

- DNS ビューを構成した IP アドレスをクライアントに送信します。DNS ビューの詳細については、「DNS ビューの追加」を参照してください。
- 静的近接動作をオーバーライドする優先ロケーションのリストを参照した後、クライアントに GSLB サービスの IP アドレスを送信します。優先ロケーションの詳細については、「優先ロケーションの設定による静的近接動作のオーバーライド」を参照してください。
- DNS クエリーまたは応答 (DNS 応答書き換え) の評価によって決定された特定の IP アドレスをクライアントに送信します。
- アプライアンスの DNS キャッシュ内で検索を実行せずに、ネーム・サーバーに要求を転送します。

- リクエストを削除します。

DNS 要求を削除したり、アプライアンスの DNS キャッシュをバイパスしたりするための DNS アクションを作成することはできません。DNS 要求を削除する場合は、組み込みアクション dns_default_act_Drop を使用します。DNS キャッシュをバイパスする場合は、組み込みアクション dns_default_act_Cachebypass を使用します。どちらの操作も、[DNS ポリシーの作成] ダイアログボックスと [DNS ポリシーの構成] ダイアログボックスでカスタム操作とともに使用できます。これらの組み込みアクションは変更または削除できません。

コマンドラインインターフェイスを使用して **DNS** アクションを構成するには

コマンドプロンプトで次のコマンドを入力して、DNS アクションを構成し、構成を確認します。

```
add dns action <actionName> <actionType> (-IPAddress <ip_addr |
    ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
    ...) [-TTL <secs>]

show dns action [<actionName>]
<!--NeedCopy-->
```

例

例 **1: DNS** 応答の書き換えの設定次の DNS アクションは、アクションがバインドされているポリシーが true と評価されると、事前設定された IP アドレスをクライアントに送信します。

```
add dns action dns_act_response_rewrite Rewrite_Response -IPAddress
    192.0.2.20 192.0.2.56 198.51.100.10
 Done

show dns action dns_act_response_rewrite
1)  ActionName:  dns_act_response_rewrite ActionType:  Rewrite_Response
     TTL:  3600               IPAddress:  192.0.2.20 192.0.2.56
    198.51.100.10
 Done
<!--NeedCopy-->
```

例 **2: DNS** ビューベースの応答の設定次の DNS アクションは、DNS ビューを構成した IP アドレスをクライアントに送信します。

```
add dns action send_ip_from_view_internal_ip ViewName -viewName
    view_internal_ip
```



```
Done

show dns action send_ip_from_view_internal_ip
1)  ActionName:  send_ip_from_view_internal_ip ActionType:  ViewName
      ViewName:  view_internal_ip
 Done
<!--NeedCopy-->
```

例 **3:** 優先ロケーションリストに基づく応答の設定次の DNS アクションは、指定されたロケーションリストから選択した優先ロケーションに対応する IP アドレスをクライアントに送信します。

```
add dns action send_preferred_location GslbPrefLoc -preferredLocList NA
   .tx.ns1.\*.\*.* NA.tx.ns2.\*.\*.* NA.tx.ns3.\*.\*.*
Done

show dns action  send_preferred_location
1)  ActionName:  send_preferred_location ActionType:  GslbPrefLoc
   PreferredLocList:  ”NA.tx.ns1.\*.\*.\*” ”NA.tx.ns2.\*.\*.\*” ”NA.tx.
   ns3.\*.\*.*”
 Done
<!--NeedCopy-->
```

Citrix ADC 構成ユーティリティを使用して **DNS** アクションを構成するには

1. [トラフィック管理] > [DNS] > [アクション] に移動し、DNS アクションを作成または編集します。
2. [DNS アクションの作成] または [DNS アクションの構成] ダイアログボックスで、次のパラメータを設定します。
   - アクション名 (既存の DNS アクションでは変更できません)
   - 種類 (既存の DNS 操作では変更できません)

     Type パラメータ

     を設定するには、次のいずれかの操作を行います。
     - DNS ビューに関連付けられた DNS アクションを作成するには、[View Name] を選択します。次に、[View Name] ボックスの一覧から、アクションで使用する DNS ビューを選択します。
     - 優先ロケーションリストを使用して DNS アクションを作成するには、[優先ロケーションリスト] を選択します。[優先する場所] に場所を入力し、[追加] をクリックします。DNS の場所を必要な数だけ追加します。
     - ポリシー評価に基づいて DNS 応答を書き換える DNS アクションを構成するには、[Rewrite Response] を選択します。[IP アドレス] に IP アドレスを入力し、[追加] をクリックします。必要な数の IP アドレスを追加します。
   - TTL（「応答の書き換え」アクションタイプにのみ適用）

DNS ポリシーを構成する

DNS ポリシーは、静的 IP アドレスとカスタム IP アドレスを使用するロケーションデータベース上で動作します。着信ローカル DNS 要求の属性は式の一部として定義され、ターゲットサイトは DNS ポリシーの一部として定義されます。アクションと式を定義するときに、一重引用符 (‘) のペアをワイルドカード修飾子として使用して、複数の場所を指定できます。DNS ポリシーが設定され、GSLB 要求が受信されると、最初にカスタム IP アドレスデータベースに、ソースのロケーション属性を定義するエントリが照会されます。

- DNS クエリーが LDNS から来た場合、LDNS の特性は、設定されたポリシーに対して評価されます。一致すると、適切なアクション (サイトアフィニティ) が実行されます。LDNS 特性が複数のサイトと一致する場合、要求は LDNS 特性と一致するサイト間で負荷分散されます。
- エントリがカスタムデータベースで見つからない場合、スタティック IP アドレスデータベースでエントリが照会され、一致する場合は上記のポリシー評価が繰り返されます。
- カスタムデータベースまたは静的データベースでエントリが見つからない場合、最適なサイトが選択され、構成された負荷分散方法に基づいて DNS 応答で送信されます。

Citrix ADC アプライアンスで作成される DNS ポリシーには、次の制限が適用されます。

- 最大 64 のポリシーがサポートされます。
- DNS ポリシーは Citrix ADC アプライアンスに対してグローバルであり、特定の仮想サーバーやドメインには適用できません。
- ドメインまたは仮想サーバ固有のポリシーのバインディングはサポートされていません。

DNS ポリシーを使用して、特定の IP アドレス範囲に一致するクライアントを特定のサイトに誘導できます。たとえば、地理的に分離された複数の GSLB サイトを持つ GSLB セットアップがある場合、IP アドレスが特定の範囲内にあるすべてのクライアントを特定のデータセンターに誘導できます。

TCP ベースと UDP ベースの DNS トラフィックの両方を評価できます。ポリシー式は、サーバー上の UDP ベースの DNS トラフィックと、クライアント側の UDP ベースの DNS トラフィックと TCP ベースの DNS トラフィックの両方に使用できます。さらに、次の DNS 質問タイプ (または QTYPE 値) のみを含むクエリと応答を評価する式を構成できます。

- A
- AAAA
- NS
- SRV
- PTR
- CNAME
- SOA
- MX
- ANY

次の応答コード (RCODE 値) もサポートされています。

- NOERROR-エラーなし

- FORMERR-フォーマットエラー
- SERVFAIL-サーバの障害
- NXDOMAIN-存在しないドメイン
- NOTIMP-クエリタイプが実装されていません
- REFUSED-クエリが拒否されました

DNS トラフィックを評価する式を設定できます。DNS 式は、DNS.REQ または DNS.RES プレフィックスで始まります。関数は、クエリーされたドメイン、クエリータイプ、およびキャリアプロトコルを評価するために利用できます。DNS 式の詳細については、「ポリシーの設定およびリファレンス」の「DNS メッセージの評価とそのキャリアプロトコルを識別するための式」を参照してください。

コマンドラインインターフェイスを使用して **DNS** ポリシーを追加するには

コマンドプロンプトで次のコマンドを入力して、DNS ポリシーを作成し、構成を確認します。

```
add dns policy <name> <rule> <actionName>
show dns policy <name>
<!--NeedCopy-->
```

例:

```
> add dns policy policy-GSLB-1 'CLIENT.UDP.DNS.DOMAIN.EQ(”domainname”)'
    my_dns_action
 Done
> show dns policy policy-GSLB-1
    Name: policy-GSLB-1
    Rule: CLIENT.UDP.DNS.DOMAIN.EQ(”domainname”)
    Action Name: my_dns_action
    Hits: 0
    Undef Hits: 0

 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、構成された **DNS** ポリシーを削除するには

コマンドプロンプトで入力します。

```
rm dns policy <name>
```



```
<!--NeedCopy-->
```

Citrix ADC 構成ユーティリティを使用して **DNS** ポリシーを構成するには

1. [トラフィック管理] > [DNS] > [ポリシー] に移動し、DNS ポリシーを作成します。
2. [DNS ポリシーの作成] または [DNS ポリシーの構成] ダイアログボックスで、次のパラメータを設定します。
   - ポリシー名（既存のポリシーでは変更できません）
   - 操作（アクション）
   - [式] 式

     を指定するには、次の操作を行います。

     a) [追加] をクリックし、表示されるドロップダウンボックスで、式の開始に使用する式要素を選択します。2 番目のリストが表示されます。リストには、firs 式要素の直後に使用できる式要素のセットが含まれています。

     b) 2 番目のリストで、使用する式要素を選択し、ピリオドを入力します。

     c) 各選択の後、ピリオドを入力すると、有効な式要素の次のセットがリストに表示されます。式の要素を選択し、必要な式が得られるまで、関数の引数を入力します。
3. [作成] または [OK] をクリックし、[閉じる] をクリックします。

DNS ポリシーのバインド

DNS ポリシーは、Citrix ADC アプライアンス上でグローバルにバインドされ、構成されたすべての GSLB 仮想サーバーで使用できます。DNS ポリシーはグローバルにバインドされていますが、式でドメインを指定することで、ポリシーの実行を特定の GSLB 仮想サーバーに制限できます。

注：binddns グローバルコマンドは受け入れますが REQ_OVERRIDE そして RES_OVERRIDE DNS ポリシーはグローバルにのみバインドできるため、有効なバインドポイントとして、これらのバインドポイントは冗長です。DNS ポリシーを REQ_DEFAULT と RES_DEFAULT バインドポイントにのみバインドします。

コマンドラインインターフェイスを使用して **DNS** ポリシーをグローバルにバインドするには

コマンドプロンプトで次のコマンドを入力して、DNS ポリシーをグローバルにバインドし、構成を確認します。

```
bind dns global <policyName> <priority> [-gotoPriorityExpression <
    string>] [-type <type>]
show dns global -type <type>
<!--NeedCopy-->
```

例**:**


```
bind dns global policy-GSLB-1 10 -gotoPriorityExpression END
 Done
show dns global -type REQ_DEFAULT
1)  Policy Name: policy-GSLB-1
    Priority: 10
    GotoPriorityExpression: END
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **DNS** ポリシーをグローバルにバインドするには

1. Traffic Management > DNS > Policies に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックします。
3. [グローバルへの DNS ポリシーのバインド/バインド解除] ダイアログボックスで、[ポリシーの挿入] をクリックします。
4. [Policy Name] 列で、バインドするポリシーをリストから選択します。または、一覧の [新しいポリシー] をクリックし、[DNS ポリシーの作成] ダイアログボックスでパラメーターを設定して DNS ポリシーを作成します。
5. 既にグローバルにバインドされているポリシーを変更するには、ポリシーの名前をクリックし、[Modify Policy] をクリックします。次に、[DNS ポリシーの構成] ダイアログボックスで、ポリシーを変更し、[OK] をクリックします。
6. ポリシーのバインドを解除するには、ポリシーの名前をクリックし、[ポリシーのバインド解除] をクリックします。
7. ポリシーに割り当てられた優先度を変更するには、優先度の値をダブルクリックし、新しい値を入力します。
8. 割り当てられた優先順位を再生成するには、再生成の優先順位をクリックします。優先度の値は、評価の順序に影響を与えずに、10 の増分で 100 から始まるように変更されます。
9. [OK] をクリックします。

コマンドラインインターフェイスを使用して **DNS** ポリシーのグローバルバインディングを表示するには

コマンドプロンプトで入力します。

```
show dns global
```

構成ユーティリティを使用して **DNS** ポリシーのグローバルバインディングを表示するには

1. **Traffic Management** > **DNS** > **Policies** に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックします。このダイアログボックスには、すべての DNS ポリシーのグローバルバインディングが表示されます。

DNS ビューの追加

DNS ビューを設定して、さまざまな種類のクライアントを識別し、同じ GSLB ドメインを照会するクライアントのグループに適切な IP アドレスを提供できます。DNS ビューは、クライアントに送信される IP アドレスを選択する DNS ポリシーを使用して構成されます。

たとえば、会社のドメインに GSLB を設定し、会社のネットワークでサーバーをホストしている場合、会社の内部ネットワーク内からドメインを照会するクライアントには、パブリック IP アドレスではなくサーバーの内部 IP アドレスを使用できます。一方、インターネットからドメインの DNS を照会するクライアントには、ドメインのパブリック IP アドレスを指定できます。

DNS ビューを追加するには、最大 31 文字の名前を割り当てます。先頭の文字は、数字または文字でなければなりません。次の文字も使用できます:@ _-. (ピリオド): (コロン) # とスペース ()。ビューを追加したら、ポリシーをクライアントおよびネットワークの一部に関連付けるように設定し、ポリシーをグローバルにバインドします。DNS ポリシーを構成およびバインドするには、「DNS ポリシー を使用してローカル **DNS** トラフィックを管理する」を参照してください。

コマンドラインインターフェイスを使用して **DNS** ビューを追加するには

コマンドプロンプトで次のコマンドを入力して、DNS ビューを作成し、構成を確認します。

```
add dns view <viewName>
show dns view <viewName>
<!--NeedCopy-->
```

例:

```
add dns view PrivateSubnet
show dns view PrivateSubnet
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **DNS** ビューを削除するには

コマンドプロンプトで入力します。

```
rm dns view <viewName>
<!--NeedCopy-->
```


構成ユーティリティを使用して **DNS** ビューを追加するには

[トラフィック管理] > [DNS] > [ビュー] に移動し、DNS ビューを追加します。

DNS ポリシーを作成する方法と DNS ポリシーをグローバルにバインドする方法の詳細については、「DNS ポリシーを使用してローカル **DNS** トラフィックを管理する」を参照してください。

近接性のための **GSLB** の設定

October 7, 2021

GSLB を近接するように設定すると、クライアント要求は最も近いデータセンターに転送されます。近接ベースの GSLB 方式の主な利点は、利用可能な最も近いデータセンターを選択することで応答時間が短縮されることです。このような展開は、大量のデータへの高速アクセスを必要とするアプリケーションにとって重要です。

ラウンドトリップ時間（RTT）、スタティック近接、または 2 つの組み合わせに基づいて、近接性の GSLB を設定できます。

動的ラウンドトリップ時間（**RTT**）方式の設定

動的ラウンドトリップ時間 (RTT) は、クライアントのローカル DNS サーバーとデータリソース間のネットワークにおける時間または遅延の尺度です。動的 RTT を測定するために、Citrix ADC アプライアンスはクライアントのローカル DNS サーバーを検査し、RTT メトリック情報を収集します。次に、アプライアンスはこのメトリックを使用して、ロードバランシングの決定を行います。グローバル・サーバ・ロード・バランシングは、ネットワークのリアルタイム・ステータスを監視し、最も低い RTT 値を持つデータ・センターにクライアント要求を動的に送信します。

動的方式との近接性のために GSLB を設定するには、まず基本的な GSLB 設定を設定し、次にダイナミック RTT を設定する必要があります。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイトの GSLB 仮想サーバーと GSLB サービスを作成し、そのサービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、ダイナミック RTT 方式を設定します。

ロードバランシングにダイナミック RTT 方式を使用するように GSLB 仮想サーバーを構成する方法の詳細については、ダイナミック RTT の設定を参照してください。

静的な近接の設定

GSLB の静的近接方式では、IP アドレスベースの静的近接データベースを使用して、クライアントのローカル DNS サーバーと GSLB サイト間の近接性を判断します。Citrix ADC アプライアンスは、近接基準に最も一致するサイトの IP アドレスで応答します。

地理的に異なる場所にある 2 つ以上の GSLB サイトが同じコンテンツを提供する場合、Citrix ADC アプライアンスは IP アドレス範囲のデータベースを維持し、着信クライアント要求を送信する GSLB サイトについての決定にデータベースを使用します。

GSLB をスタティック近接に設定するには、まず基本的な GSLB 設定を設定し、次にスタティック近接を設定する必要があります。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイトの GSLB 仮想サーバーと GSLB サービスを作成し、そのサービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、静的な近接を設定します。

負荷分散に静的近接を使用するように GSLB 仮想サーバーを構成する方法の詳細については、静的近接の設定を参照してください。

スタティック近接およびダイナミック **RTT** の設定

ブランチオフィスのような内部ネットワークから来ているクライアントがある場合、静的近接と動的 RTT の組み合わせを使用するように GSLB 仮想サーバーを設定できます。ブランチオフィスまたは他の内部ネットワークから来るクライアントが、クライアントネットワークに地理的に近い特定の GSLB サイトに誘導されるように、GSLB を設定できます。他のすべての要求では、ダイナミック RTT を使用できます。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイトの GSLB 仮想サーバーと GSLB サービスを作成し、そのサービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、内部ネットワークから発信されるすべてのトラフィックに静的近接性を使用し、その他すべてのトラフィックにダイナミック RTT を使用するように GSLB 仮想サーバを設定します。

スタティック近接の設定方法の詳細については、スタティック近接の設定を参照してください 。ダイナミック RTT の設定方法の詳細については、ダイナミック RTT の設定を参照してください。

GSLB セットアップを障害から保護する

October 7, 2021

GSLB サイトまたは GSLB 仮想サーバーの障害から GSLB セットアップを保護するには、以下を構成します。

- バックアップ GSLB 仮想サーバー
- 複数の IP アドレスで応答する Citrix ADC アプライアンス
- GSLB ドメインのバックアップ IP アドレス

また、スピルオーバーを使用して、余分なトラフィックをバックアップ仮想サーバーに転送することもできます。

バックアップ GSLB 仮想サーバーの構成

GSLB 仮想サーバーのバックアップエンティティを構成すると、GSLB 仮想サーバーがダウンしても、サイトへの DNS トラフィックが中断されることがなくなります。バックアップエンティティは、別の GSLB 仮想サーバーでも、バックアップ IP アドレスでもかまいません。バックアップエンティティが設定されている場合、プライマリ GSLB 仮想サーバーがダウンすると、バックアップエンティティは DNS 要求を処理します。プライマリ GSLB 仮想サーバーが再び復帰したときに何が起こるかを指定するには、プライマリ仮想サーバーを手動で引き継ぐまで(DisablePrimaryonDown オプションを使用して)、トラフィックを処理し続けるようにバックアップエンティティを構成できます。

注:1 つのバックアップエンティティを複数の GSLB 仮想サーバーのバックアップとして構成できます。

コマンドラインインターフェイスを使用してバックアップ GSLB 仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、GSLB 仮想サーバーをバックアップ仮想サーバーとして設定し、構成を確認します。

```
set gslb vserver <name> -backupVServer <name> [-disablePrimaryOnDown (
    ENABLED | DISABLED)]

show gslb vserver <name>
<!--NeedCopy-->
```

例:

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -
    disablePrimaryOnDown ENABLED
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```


構成ユーティリティを使用して **GSLB** 仮想サーバーをバックアップ仮想サーバーとして設定するには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、GSLB 仮想サーバーをダブルクリックします。
2. [バックアップ仮想サーバ] セクションを選択し、バックアップ仮想サーバを選択します。

複数の **IP** アドレスで応答するように **GSLB** セットアップを構成する

一般的な DNS 応答には、最もパフォーマンスに優れる GSLB サービスの IP アドレスが含まれます。ただし、複数の IP 応答（MIR）を有効にすると、Citrix ADC アプライアンスはレスポンスの最初のレコードとして最適な GSLB サービスを送信し、残りのアクティブなサービスを追加レコードとして追加します。MIR が無効になっている場合（デフォルト）、Citrix ADC アプライアンスはレスポンスの唯一のレコードとして最適なサービスを送信します。

コマンドラインインターフェイスを使用して複数の **IP** 応答用に **GSLB** 仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、GSLB 仮想サーバーを複数の IP 応答用に構成し、構成を確認します。

```
set gslb vserver<name> -MIR (ENABLED | DISABLED)
-  show gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED
 show gslb vserver <vserverName>
<!--NeedCopy-->
```

構成ユーティリティを使用して複数の **IP** 応答用に **GSLB** 仮想サーバーを設定するには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、バックアップ仮想サーバーを設定する GSLB 仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. ［詳細設定］タブの［この仮想サーバーが「稼働中」の場合］で、［すべての「アクティブな」サービス IP をレスポンス（MIR）で送信する］チェックボックスをオンにして、［**OK**］を選択します。

DOWN 時に空のアドレスレコードで応答するように **GSLB** 仮想サーバを設定する

DNS 応答には、要求されたドメインの IP アドレス、またはドメインの IP アドレスが DNS サーバーによって認識されていないことを示す応答のいずれかを含めることができます。この場合、クエリは別のネームサーバーに転送されます。これらは、DNS クエリに対する唯一の可能な応答です。

GSLB 仮想サーバーが無効または DOWN 状態の場合、その仮想サーバーにバインドされている GSLB ドメインに対する DNS クエリへの応答には、仮想サーバーにバインドされているすべてのサービスの IP アドレスが含まれます。ただし、この場合、空のダウン応答（EDR）を送信するように GSLB 仮想サーバーを構成できます。このオプションを設定すると、DOWN 状態にある GSLB 仮想サーバーからの DNS 応答に IP アドレスレコードは含まれませんが、応答コードは成功します。これにより、クライアントがダウンしている GSLB サイトに接続しようとするのを防ぐことができます。

注: この設定は、適用する仮想サーバーごとに構成する必要があります。

コマンドラインインターフェイスを使用して **GSLB** 仮想サーバーを空ダウン応答用に構成するには

コマンドプロンプトで入力します。

```
set gslb vserver<name> -EDR (ENABLED | DISABLED)
<!--NeedCopy-->
```

例**:**

```
> set gslb vserver vserver-GSLB-1 -EDR ENABLED
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して空のダウン応答用に **GSLB** 仮想サーバーを設定するには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、バックアップ仮想サーバーを設定する GSLB 仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. [詳細設定] タブの [この仮想サーバーが [ダウンしているとき] で、[応答でサービスの IP アドレスを送信しない (EDR)] チェックボックスをオンにします。
3. [**OK**] をクリックします。

GSLB ドメインのバックアップ **IP** アドレスの設定

GSLB 設定のバックアップサイトを設定できます。この構成を設定すると、すべてのプライマリサイトがダウンすると、バックアップサイトの IP アドレスが DNS 応答で提供されます。

通常、GSLB 仮想サーバーがアクティブな場合、その仮想サーバーは、構成された GSLB 方式によって選択されたアクティブサイト IP アドレスのいずれかを使用して DNS 応答を送信します。GSLB 仮想サーバーで設定されているすべてのプライマリサイトが非アクティブ（ダウン状態）の場合、オーソリテイティブドメインネームシステム (ADNS) サーバーまたは DNS サーバーは、バックアップサイトの IP アドレスとともに DNS 応答を送信します。

注：バックアップ IP アドレスが送信されると、永続性は考慮されません。

コマンドラインインターフェイスを使用してドメインのバックアップ **IP** アドレスを設定するには

コマンドプロンプトで次のコマンドを入力して、バックアップ IP アドレスを設定し、構成を確認します。

```
set gslb vserver <name> -domainName <string> -backupIP <IPAddress>
show gslb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP
    10.102.29.66
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用してドメインのバックアップ **IP** アドレスを設定するには

1. 「トラフィック管理」>「**GSLB** 」>「仮想サーバー」に移動し、バックアップドメインをバインドする GSLB 仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. [ドメイン] セクションをクリックし、GSLB ドメインを設定し、[バックアップ IP] フィールドにバックアップドメイン の **IP** アドレスを指定します。

過剰なトラフィックをバックアップ仮想サーバに転送する

プライマリ GSLB 仮想サーバーへの接続数が設定されたしきい値を超えたら、spillover オプションを使用して、新しい接続をバックアップ GSLB 仮想サーバーに転送できます。このしきい値は、動的に計算することも、手動で設定することもできます。プライマリ仮想サーバーへの接続数がしきい値を下回ると、プライマリ GSLB 仮想サーバーはクライアント要求の処理を再開します。

パーシステンスはスピルオーバーで設定できます。パーシステンスを構成すると、そのクライアントがプライマリ仮想サーバに接続されていない場合、新しいクライアントはバックアップ仮想サーバに転送されます。パーシステンスが設定されている場合、バックアップ仮想サーバに誘導された接続は、プライマリ仮想サーバへの接続数がしきい値を下回った後にプライマリ仮想サーバに戻されません。代わりに、バックアップ仮想サーバは、ユーザーによって終了されるまでこれらの接続を処理し続けます。一方、プライマリ仮想サーバは新しいクライアントを受け入れます。

しきい値は、接続の数、帯域幅、およびサービスの状態によって測定できます。

バックアップ仮想サーバが設定されたしきい値に達し、追加の負荷をかけることができない場合、プライマリ仮想サーバはすべての要求を指定されたリダイレクト URL に転送します。リダイレクト URL がプライマリ仮想サーバで設定されていない場合、後続の要求はドロップされます。

スピルオーバー機能により、プライマリ GSLB 仮想サーバーに障害が発生したときに、リモートバックアップ GSLB サービス (バックアップ GSLB サイト) がクライアント要求でフラッディングされるのを防ぎます。これは、モニターがリモート GSLB サービスにバインドされ、サービスで障害が発生し、状態が DOWN になる場合に発生します。ただし、スピルオーバー機能により、モニタはリモート GSLB サービスの UP 状態を維持し続けます。

この問題の解決の一環として、GSLB サービスでは、プライマリ状態と有効状態の 2 つの状態が維持されます。プライマリ状態はプライマリ仮想サーバーの状態であり、有効状態は仮想サーバー（プライマリおよびバックアップチェーン）の累積状態です。仮想サーバのチェーン内のいずれかの仮想サーバが UP の場合、実効状態は UP に設定されます。プライマリ VIP がしきい値に達したことを示すフラグも提供されます。しきい値は、接続数または帯域幅のいずれかによって測定できます。

サービスは、そのプライマリ状態が UP である場合にのみ、GSLB に対して考慮されます。トラフィックがバックアップ GSLB サービスに送信されるのは、すべてのプライマリ仮想サーバが DOWN の場合だけです。通常、このような展開には、バックアップ GSLB サービスが 1 つしかありません。

GSLB サービスにプライマリ状態と有効状態を追加すると、次の効果があります。

- ソース IP パーシステンスが設定されている場合、ローカル DNS は、選択したサイトのプライマリ仮想サーバが UP でしきい値を下回っている場合に限り、以前に選択したサイトに転送されます。持続性は、ラウンドロビンモードでは無視できます。
- Cookie ベースの永続性が構成されている場合、クライアント要求は、選択したサイトのプライマリ仮想サーバーが UP している場合にのみリダイレクトされます。
- プライマリ仮想サーバが飽和状態に達し、バックアップ VIP が存在しないかダウンしている場合、有効状態は DOWN に設定されます。
- 外部モニターが HTTP-HTTPS 仮想サーバーにバインドされている場合、モニターはプライマリ状態を決定します。
- プライマリ仮想サーバへのバックアップ仮想サーバが存在せず、プライマリ仮想サーバがそのしきい値に達した場合、有効状態は DOWN に設定されます。

コマンドラインインターフェイスを使用してバックアップ **GSLB** 仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、バックアップ GSLB 仮想サーバーを構成し、構成を確認します。

```
set gslb vserver <name> -soMethod <method> -soThreshold <threshold> -
    soPersistence ( \*\*ENABLED\*\* | \*\*DISABLED\*\* ) -
    soPersistenceTimeout <timeout>
show gslb vserver <name>
<!--NeedCopy-->
```

例**:**


```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000
    -soPersistence ENABLED -soPersistenceTimeout 2
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用してバックアップ **GSLB** 仮想サーバーを構成するには

1. [トラフィック管理] > [**GSLB**] > [仮想サーバー] に移動し、バックアップとして構成する仮想サーバー (vserver-LB-1 など) をダブルクリックします。
2. [**SpillOver**] セクションをクリックし、次のパラメータを設定します。
   - メソッド — soMethod
   - しきい値: soThreshold
   - 持続性タイムアウト (分) — soPersistenceTimeout
3. 「持続性」オプションを選択し、「**OK**」をクリックします。

災害復旧のための **GSLB** の構成

October 7, 2021

ダウンタイムにはコストがかかりますので、ディザスタリカバリ機能は重要です。GSLB 用に構成された Citrix ADC アプライアンスは、負荷が最も低いデータセンターまたはパフォーマンスの高いデータセンターにトラフィックを転送します。この構成は、アクティブ-アクティブ設定と呼ばれ、パフォーマンスが向上するだけでなく、セットアップの一部であるデータセンターがダウンした場合に、トラフィックを他のデータセンターにルーティングすることで、ディザスタリカバリを即座に実行できます。または、ディザスタリカバリ専用のアクティブ/スタンバイ GSLB セットアップを構成することもできます。

アクティブ/スタンバイのデータセンターのセットアップで **GSLB** をディザスタリカバリ用に構成する

従来のディザスタリカバリ設定には、アクティブなデータセンターとスタンバイデータセンターが含まれます。スタンバイデータセンターはリモートサイトです。障害イベントの結果としてフェイルオーバーが発生すると、プライマリアクティブデータセンターが非アクティブになります。スタンバイデータセンターは稼動状態になります。

アクティブ/スタンバイデータセンターセットアップでのディザスタリカバリの構成は、次のタスクで構成されます。

- アクティブなデータセンターを作成します。
  - ローカル GSLB サイトを追加します。
  - アクティブなデータセンターを表す GSLB 仮想サーバーを追加します。
  - ドメインを GSLB 仮想サーバーにバインドします。
  - gslb サービスを追加し、アクティブな GSLB 仮想サーバーにサービスをバインドします。

- スタンバイデータセンターを作成します。
  - リモート gslb サイトを追加します。
  - スタンバイデータセンターを表す gslb vserver を追加します。
  - スタンバイデータセンターを表す gslb サービスを追加し、スタンバイ gslb vserver にサービスをバインドします。
  - スタンバイ GSLB 仮想サーバーをアクティブ GSLB 仮想サーバーのバックアップ仮想サーバーとして構成して、スタンバイデータセンターを指定します。

プライマリデータセンターを構成したら、バックアップデータセンターの設定を複製し、そのサイトの GSLB 仮想サーバーをバックアップ仮想サーバーとして指定することで、スタンバイ GSLB サイトとして指定します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

コマンドラインインターフェイスを使用してスタンバイ **GSLB** サイトを指定するには

アクティブサイトとリモートサイトの両方で、コマンドプロンプトで次のように入力します。

```
1 set gslb vserver <name> -backupVserver <string>
2 <!--NeedCopy-->
```

例**:**

```
1 set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
2 <!--NeedCopy-->
```

構成ユーティリティを使用してスタンバイサイトを構成するには

1. 「トラフィック管理」>「GSLB」>「仮想サーバー」に移動し、プライマリサイトの GSLB 仮想サーバーをダブルクリックします。
2. [仮想サーバーのバックアップ] セクションをクリックし、バックアップ仮想サーバーを選択します。

デフォルトでは、プライマリ仮想サーバがアクティブになると、トラフィックの受信が開始されます。ただし、プライマリ仮想サーバーがアクティブになった後でも、トラフィックをバックアップ仮想サーバーに転送する場合は、「プライマリの停止を無効にする」オプションを使用します。

アクティブ-アクティブ・データ・センター・セットアップでの災害復旧の構成

両方の GSLB サイトがアクティブであるアクティブ-アクティブ GSLB デプロイメントは、スタンバイデータセンターを持つ際に発生する可能性のあるリスクを排除します。このような設定により、Web コンテンツやアプリケーショ

ンのコンテンツを地理的に離れた場所にミラーリングできます。これにより、分散した各データセンターでデータを一貫して利用できるようになります。

アクティブ-アクティブデータセンター設定で GSLB をディザスタリカバリ用に構成するには、まず最初のデータセンターで基本 GSLB セットアップを構成してから、他のすべてのデータセンターを構成する必要があります。

まず、少なくとも 2 つの GSLB サイトを作成します。次に、ローカルサイトの GSLB 仮想サーバーと GSLB サービスを作成し、そのサービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。最後に、ローカルサイトで、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

最初のデータセンターを構成したら、セットアップの他のデータセンターの構成を複製します。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

加重ラウンドロビンを使用したディザスタリカバリの構成

重み付きラウンドロビン方式を使用するように GSLB を設定すると、重みが GSLB サービスに追加され、設定された着信トラフィックの割合が各 GSLB サイトに送信されます。たとえば、トラフィックの 80% をあるサイトに転送し、トラフィックの 20% を別のサイトに転送するように GSLB 設定を構成できます。これを実行すると、Citrix ADC アプライアンスは、2 番目のサイトに送信される各リクエストについて、最初のサイトに 4 つのリクエストを送信します。

加重ラウンドロビン方式を設定するには、まずローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイトの GSLB 仮想サーバーと GSLB サービスを作成し、サービスを仮想サーバーにバインドします。GSLB 方式をラウンドロビンとして設定します。次に、ADNS サービスを作成し、GSLB を構成するドメインを GSLB 仮想サーバーにバインドします。最後に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。

ネットワーク内の物理サーバを表す各サービスには、重み付けが関連付けられています。したがって、GSLB サービスには、バインドされているすべてのサービスの重みの合計である動的重みが割り当てられます。トラフィックは、特定のサービスの動的重みと合計重みとの比率に基づいて、GSLB サービス間で分割されます。また、動的重みではなく、GSLB サービスごとに個別の重みを設定することもできます。

サービスにウェイトが関連付けられていない場合、GSLB 仮想サーバーを構成して、バインドされたサービス数を使用してウェイトを動的に計算できます。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB 設定を設定したら、個々のサービスに設定された重みに従って、設定された GSLB サイト間でトラフィックが分割されるように、重み付けラウンドロビン方式を設定する必要があります。

コマンドラインインターフェイスを使用してサービスに重みを割り当てるように仮想サーバを構成するには

コマンドプロンプトで、新しい負荷分散仮想サーバーを作成するか、既存のサーバーを構成するかに応じて、次のコマンドのいずれかを入力します。


```
add lb vserver <name>@ -weight <WeightValue> <ServiceName>
set lb vserver <name>@ -weight <WeightValue> <ServiceName>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用してダイナミックウェイトを設定するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -dynamicWeight DynamicWeightType
<!--NeedCopy-->
```

例:

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **GSLB** サービスに重みを追加するには

コマンドプロンプトで入力します。

```
set gslb vserver <name> -serviceName GSLBServiceName -weight
   WeightValue
<!--NeedCopy-->
```

例:

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
<!--NeedCopy-->
```


構成ユーティリティを使用してサービスに重みを割り当てるように仮想サーバを構成するには

1. 「トラフィック管理」>「負荷分散」>「仮想サーバー」に移動し、仮想サーバー (Vserver-LB-1 など) をダブルクリックします。
2. [サービス] セクションをクリックし、サービスの重みを設定します。

構成ユーティリティを使用して **GSLB** サービスに重みを追加するには

1. 「トラフィック管理」>「GSLB」>「仮想サーバー」に移動し、仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. [Services] セクションをクリックし、[Weight] フィールドでサービスの重みを設定します。

構成ユーティリティを使用して動的重みを設定するには

1. 「トラフィック管理」>「GSLB」>「仮想サーバー」に移動し、仮想サーバー（vserver-GSLB-1 など）をダブルクリックします。
2. [方法] セクションをクリックし、[動的重量] ドロップダウンリストから [**SERVICEWEIGHT**] を選択します。

データセンターの永続性を伴うディザスタリカバリの構成

データセンターの永続性は、要求をロードバランシングするのではなく、同じサーバーとの接続を維持する必要がある Web アプリケーションに対して必要です。たとえば、e コマースポータルでは、クライアントと同じサーバー間の接続を維持することが重要です。このようなアプリケーションでは、HTTP リダイレクトの永続性をアクティブ-アクティブ設定で設定できます。

GSLB をデータセンターの永続性を使用して障害復旧用に設定するには、まず基本的な GSLB 設定を設定し、次に HTTP リダイレクト永続性を設定する必要があります。

まず、ローカルとリモートの 2 つの GSLB サイトを作成します。次に、ローカルサイトの GSLB 仮想サーバーと GSLB サービスを作成し、そのサービスを仮想サーバーにバインドします。次に、ADNS サービスを作成し、GSLB を構成するドメインをローカルサイトの GSLB 仮想サーバーにバインドします。次に、GSLB サービスと同じ仮想サーバー IP アドレスを持つ負荷分散仮想サーバーを作成します。最後に、リモート構成の前の手順を複製するか、Citrix ADC アプライアンスを構成して GSLB 構成を自動同期させます。

基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

基本的な GSLB セットアップを設定したら、HTTP リダイレクトの優先順位を設定して、データセンターの永続性を有効にします。

コマンドラインインターフェイスを使用して **HTTP** リダイレクトを構成するには

コマンドプロンプトで、次のコマンドを入力して HTTP リダイレクトを構成し、構成を確認します。


```
set gslb service <serviceName> -sitePersistence <sitePersistence> -
    sitePrefix <string>
show gslb service <serviceName>
<!--NeedCopy-->
```

例:

```
set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -
    sitePrefix vserver-GSLB-1
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

構成ユーティリティを使用して **HTTP** リダイレクトを構成するには

1. 「トラフィック管理」>「GSLB」>「サービス」に移動し、設定する GSLB サービスをダブルクリックします。
2. 「サイトの永続性」セクションをクリックし、「**HTTPRedirect** 」オプションを選択し、「サイトプレフィックス」テキストボックスにサイトプレフィックス（vserver-GSLB-1 など）を入力します。

> 注
>
> サイトの永続性が構成されておらず、ローカル GSLB サービスとして構成されている負荷分散仮想サーバーが DOWN の場合、HTTP 要求は 302 リダイレクトを使用して他の正常な GSLB サイトにリダイレクトされます。

優先ロケーションを設定することにより、静的な近接動作を無効にする

October 7, 2021

ローカル DNS (LDNS) サーバーまたはネットワークからのトラフィックを、静的近接方式がそのトラフィックに対して選択する GSLB サービス以外の GSLB サービスに誘導する場合があります。つまり、そのトラフィックに優先するロケーションがあるということです。静的近接メソッドを優先位置でオーバーライドするには、次の操作を行います。

1. 優先ロケーションのリストで構成される DNS アクションを設定します。DNS アクションの構成の詳細については、「DNS アクションの設定」を参照してください。
2. 静的近接を上書きする LDNS サーバまたはネットワークから着信するトラフィックを識別するように DNS ポリシーを設定し、ポリシーでアクションを適用します。
3. ポリシーをグローバル要求バインドポイントにバインドします。

DNS アクションでは、最大 8 つの優先ロケーションのリストを設定できます。ロケーションはドット付き修飾子表記で指定する必要があります。ドット付き修飾子表記は、スタティック近接データベースにカスタムのロケーション

を追加するときの表記です。場所には、省略する修飾子のワイルドカードを含めることができます。ロケーションのドット付き修飾子の表記法については、静的近接データベースへのカスタムエントリの追加を参照してください。優先ロケーションを入力するときは、優先順位の降順で入力する必要があります。

ポリシーが
TRUE と評価されると、Citrix ADC アプライアンスは、優先される場所を GSLB サービスの場所と優先順に照合します。一致には、次の 2 つのタイプがあります。

- 優先ロケーションにあるワイルドカード以外の修飾子がすべて GSLB サービスのロケーションにある対応する修飾子と一致する場合、一致は完全一致とみなされます。たとえば、GSLB サービスの場所は *.UK.*.* または Europe.UK.*.* 好みの場所にぴったりです *.UK.*.*.
- ワイルドカード以外の修飾子のサブセットだけが一致する場合、その一致は部分一致と見なされます。たとえば、GSLB サービスの場所が Europe.EG の場合、優先する場所 Europe.UK の部分一致です。

DNS ポリシーが
TRUE と評価されると、次のアルゴリズムを使用して GSLB サービスを選択します。

1. アプライアンスは、優先順位が最も高い優先ロケーションを評価し、優先ロケーションと GSLB サービスのロケーション間で完全に一致が見つかるまで優先順を下に移動します。

   完全一致が見つかった場合、アプライアンスは対応する GSLB サービスが稼働しているかどうかをチェックします。起動している場合は、GSLB サービスの IP アドレスを DNS 応答で返します。複数の完全一致が見つかった場合（優先ロケーションで 1 つ以上のワイルドカードが使用されている場合に発生する可能性があります）、アプライアンスは対応する GSLB サービスの各状態をチェックし、稼動している GSLB サービスの負荷分散を行います。

2. いずれかの優先ロケーションで完全一致が見つからない場合、アプライアンスは優先度の高い優先ロケーションに戻り、優先ロケーションと GSLB サービスのロケーションの間で部分一致が見つかるまで、優先順を下に移動します。

   部分一致が見つかった場合、アプライアンスは対応する GSLB サービスが稼働しているかどうかをチェックします。起動している場合は、GSLB サービスの IP アドレスを DNS 応答で返します。複数の部分一致が見つかった場合、アプライアンスは対応する GSLB サービスの状態をチェックし、稼働している GSLB サービスの負荷分散を行います。

3. 完全一致と部分一致のいずれもアップしていない場合、アプライアンスは他のすべての利用可能な GSLB サービスを負荷分散します。

   このようにして、アプライアンスは DNS ポリシーに一致するトラフィックに対してサイトアフィニティのタイプを実装します。

例

次の 8 つの GSLB サービスで構成される GSLB 設定を考えてみましょう。

- Asia.IN

- Asia.JPN
- Asia.HK
- Europe.UK
- Europe.RU
- Europe.EG
- Africa.SD
- アフリカ.ZMB

さらに、次の DNS アクションとポリシー設定について検討してください。

```
1 > add dns action prefLoc11 GslbPrefLoc -preferredLocList ”Asia.HK” ”
    Europe.UK”
2  Done
3 > add dns policy dnsPolPrefLoc ”CLIENT.IP.SRC.MATCHES_LOCATION(”\*.ZMB
    .\*.*”)” prefLoc11
4  Done
5 <!--NeedCopy-->
```

アプライアンスがロケーション
.ZMB からリクエストを受信したとき。*.** の場合、優先される場所は次のように評価されます。

1. アプライアンスは、最も優先順位の高い優先ロケーションである Asia.HK と完全に一致する GSLB サービスを検索しようとします。Asia.HK の GSLB サービスが完璧にマッチしていることがわかりました。GSLB サービスが稼働している場合、GSLB サービスの IP アドレスをクライアントに送信します。
2. Asia.HK の GSLB サービスがダウンしている場合、アプライアンスは 2 番目に優先される場所 Europe.UK に対して完全一致を検出しようとします。Europe.UK の GSLB サービスが完璧にマッチしていることがわかりました。GSLB サービスが稼働している場合、サービスの IP アドレスをクライアントに送信します。
3. Europe.UK の GSLB サービスがダウンしている場合、優先度が最も高い優先度 Asia.HK を持つ優先ロケーションに戻り、部分一致を検索します。Asia.HK では、アジア.IN とアジア.JPN が部分一致であることが検出されます。対応する GSLB サービスの 1 つだけが稼働している場合は、クライアントにサービスの IP アドレスを送信します。両方のロケーションが稼働している場合は、2 つのサービスの負荷分散を行います。
4. Asia.HK の部分一致がすべてダウンしている場合、アプライアンスは Europe.UK の部分一致を検索します。これにより、Europe.RU と Europe.EG が優先ロケーションの部分一致であることが検出されます。対応する GSLB サービスの 1 つだけが稼働している場合は、クライアントにサービスの IP アドレスを送信します。両方のロケーションが稼働している場合は、2 つのサービスの負荷分散を行います。
5. Europe.UK のすべての部分一致がダウンした場合、アプライアンスは他のすべての利用可能な GSLB サービスを負荷分散します。現在の例では、残りの 6 つの GSLB サービスがダウンしていることが判明したため、アプライアンスでは Africa.SD と Africa.ZMB の負荷を分散します。

コンテンツスイッチを使用した **GSLB** サービス選択の設定

October 7, 2021

一般的な GSLB デプロイメントでは、GSLB 仮想サーバーにバインドされた GSLB サービスの選択に優先順位を付けることができますが、次の操作を行うことはできません。

- 特定のドメインの GSLB 仮想サーバーにバインドされた GSLB サービスのサブセットから GSLB サービスの選択を制限します。
- デプロイメント内の GSLB サービスの異なるサブセットに、異なる負荷分散メソッドを適用します。
- GSLB サービスのサブセットにスピルオーバーポリシーを適用し、GSLB サービスのサブセットのバックアップを作成することはできません。
- 異なるコンテンツを提供するために GSLB サービスのサブセットを設定します。つまり、異なる GSLB サイト内のサーバー間でコンテンツを切り替えることはできません。GSLB 設定では、サーバーに同じコンテンツが含まれていることを前提としています。
- 異なる優先順位を持つサブセット GSLB サービスを定義し、サブセット内のサービスを要求に適用する順序を指定します。

コンテンツスイッチング (CS) ポリシーを設定して GSLB 展開をカスタマイズできるようになりました。まず、GSLB サービスのセットを設定し、GSLB 仮想サーバーにバインドします。次に、ターゲットタイプ GSLB の CS 仮想サーバーを構成し、ターゲット仮想サーバーとして GSLB 仮想サーバーを使用して CS ポリシーとアクションを定義し、CS ポリシーを CS 仮想サーバーにバインドします。

> 重要
>
> - DNS ベースの式を持つ CS ポリシーのみを、ターゲットタイプ GSLB の CS 仮想サーバーにバインドできます。
> - GLSB サービスが、GSLB 仮想サーバーを介して CS 仮想サーバーにバインドされている場合、同じ GSLB サービスでバインドされた別の GSLB 仮想サーバーを、CS 仮想サーバーにバインドすることはできません。

例

2 つの GSLB サイトを含む GLSB デプロイを考えてみましょう。各サイトでは、4 つの GSLB サービス（S-1、S-2、S-3、および S-4）が GSLB 仮想サーバー VS-1 にバインドされています。ターゲットタイプ GSLB のコンテンツスイッチング（CS）仮想サーバーを構成し、VS-1 をターゲット仮想サーバーとして CS ポリシーとアクションを定義することができます。これにより、英語でのコンテンツのリクエストは S-1 と S-2 でのみ処理され、ローカル言語のコンテンツのリクエストは S-3 と S-4 でのみ処理されます。

S-1 プライオリティを設定するには、バックアップ仮想サーバを VS-1 に設定し、S-2 をバックアップ仮想サーバにバインドします。S-1 はクライアント要求を処理します。サーバ S-1 がダウンした場合、S-2 は要求を処理します。S-1 と S-2 の両方がダウンしている場合、クライアントは空の応答を受け取ります。

コンテンツスイッチングを使用して **GSLB** サービス選択を設定するには:

1. GSLB を設定します。手順については、「グローバルサーバー負荷分散の設定」を参照してください。

2. ターゲットタイプ GSLB のコンテンツスイッチング (CS) 仮想サーバーを構成します。詳細については、「コンテンツスイッチング仮想サーバーの作成」を参照してください。
3. コンテンツスイッチング（CS）ポリシーを設定します。詳細については、「コンテンツスイッチングポリシーの設定」を参照してください。
4. GSLB 仮想サーバーをターゲット仮想サーバーとして指定する CS アクションを設定します。詳細については、「コンテンツ切り替えアクションの設定」を参照してください。
5. CS ポリシーを CS 仮想サーバにバインドします。詳細については、コンテンツスイッチング仮想サーバへのポリシーのバインドを参照してください。
6. ドメインを GSLB 仮想サーバーではなく CS 仮想サーバーにバインドします。

構成例

次の設定例では、IP アドレス 5.5.5.5 を持つクライアントからの要求を SERVICE_GSLB1 および SERVICE_GSLB2 に送信します。SERVICE_GSLB1 は SERVICE_GSLB2 よりも高い優先度を持ち、SERVICE_GSLB2 は SERVICE_GSLB1 がダウンしている場合にのみクライアント要求を処理します。SERVICE_GSLB1 と SERVICE_GSLB2 の両方がダウンしている場合、SERVICE_GSLB3 と service-GSLB4 は考慮されず、ブランクの応答がクライアントに送信されます。

```
add cs vs CSVSERVER_GSLB http  -targettype GSLB
Done
add gslb vs VSERVER_GSLB1 http
Done
add gslb vs VSERVER_GSLB2 http
Done
add gslb vs VSERVER_GSLB_BACKUP1 http
Done
set gslb vs VSERVER_GSLB1 -backupvserver VSERVER_GSLB_BACKUP1
Done
add gslb service SERVICE_GSLB1 1.1.1.1 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB2 1.1.1.2 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB3 1.1.1.3 HTTP 80 -sitename site2
Done
add gslb service SERVICE_GSLB4 1.1.1.4 HTTP 80 -sitename site2
Done
bind gslb vs VSERVER_GSLB1 -servicename SERVICE_GSLB1
Done
bind gslb vs VSERVER_GSLB_BACKUP1 -servicename SERVICE_GSLB2
Done
bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB3
Done
```



```
25 bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB4
26 Done
27 add cs action a1 -targetvserver VSERVER_GSLB1
28 Done
29 add cs policy p1 -rule ”CLIENT.IP.SRC.EQ(5.5.5.5)” -action a1
30 Done
31 bind cs vs CSVSERVER_GSLB -domainName www.abc.com
32 Done
33 bind cs vs CSVSERVER_GSLB -policyname p1 -priority 1
34 Done
35 add cs action a2 -targetvserver VSERVER_GSLB2
36 Done
37 add cs policy p2 -rule ”CLIENT.IP.SRC.EQ(6.6.6.6)” -action a2
38 Done
39 bind cs vs CSVSERVER_GSLB -policyname p2 -priority 2
40 Done
41 <!--NeedCopy-->
```

ターゲット仮想サーバー式を **GSLB** コンテンツスイッチングアクションに関連付ける

ターゲット仮想サーバー式を GSLB コンテンツスイッチングアクションに関連付けることができるようになりました。これにより、GSLB コンテンツスイッチング仮想サーバーは、DNS 要求を処理しながら、ターゲット GSLB 仮想サーバー名を構成するためにポリシー式を使用することができます。

CLI を使用して式を指定するコンテンツスイッチングアクションを設定するには

コマンドプロンプトで次のコマンドを入力して、HTTP コールアウト応答を取得するようにコンテンツスイッチングアクションを構成します。

```
1 add cs action <name> -targetVserverExpr <expression>
2 <!--NeedCopy-->
```

例:

```
1 add cs action csact_GSLB_VServer -targetVserverExpr ”SYS.HTTP_CALLOUT(
     GSLB_Method_API)”
2 <!--NeedCopy-->
```


GUI を使用して式を指定するコンテンツスイッチングアクションを構成するには

1. [トラフィック管理] **>** [コンテンツの切り替え] **>** [アクション] に移動します。
2. コンテンツスイッチングアクションを構成し、ターゲットの負荷分散仮想サーバーの名前を動的に計算する 式を指定します。

NAPTR レコードを使用した **DNS** クエリの **GSLB** を構成する

October 7, 2021

一般的なグローバルサーバー負荷分散（GSLB）環境では、Citrix ADC アプライアンスは A/AAAA レコードの DNS クエリを受信し、構成された負荷分散方法に従って最適な GSLB サービスを選択し、DNS クエリへの応答としてサービスの IP アドレスを返します。これで、NAPTR レコードの DNS クエリーを受信し、ドメインに設定されたサービスのリストに応答するようにアプライアンスを設定できるようになりました。アプライアンスはサービスの健全性を監視し、応答として稼働中のサービスのみのリストを提供します。

例**:**

Telco 展開では、モバイル管理エンティティ（MME）などのクライアントから NAPTR レコードを含む DNS クエリーを受信するように Citrix ADC アプライアンスを構成できます。MME は、ドメイン名によって提供されるすべてのサービスを検出するための DNS リゾルバの役割を果たします。アプライアンスは、アップしているすべてのサービスの NAPTR レコードを含むクエリーに応答します。MME は、この NAPTR 応答を使用して S-NAPTR プロシージャを実行し、提供されるサービス、コロケーション、トポロジの近さなどに基づいてノードを選択できます。

複数のノードが選択の対象となる場合、MME は Citrix ADC アプライアンスからの NAPTR レコードのプリファレンスフィールドを使用してノードを決定できます。

NAPTR レコード形式

NAPTR レコードを含む DNS クエリに応答する際、Citrix ADC アプライアンスは各 GSLB サービスに対する応答 NAPTR レコードを作成します。

次の表に、NAPTR レコード内のファイルを示します。

フィールド	
ドメイン	GSLB ドメイン
TTL	NAPTR レコードをキャッシュできる時間。
クラス	レコードのクラス。デフォルトでは、この値は IN に設定されています。
種類	DNS レコードタイプ。

フィールド	
Order	NAPTR レコードを処理する順序を指定します。GSLB サービスで順序を指定できます。それ以外の場合は、1 に設定されます。
環境設定	等しい「order」値を持つ NAPTR レコードが処理される順序を指定します。小さい数値は、高い数値の前に処理されます。順序が GSLB サービスで指定されていない場合は、1 に設定されます。
フラグ	レコード内のフィールドの書き換えと解釈の側面を制御します。Citrix ADC アプライアンスは、この値を A に設定します。
サービス	使用可能なサービスを指定します。
正規表現	正規表現はサポートされていないため、この値は NULL に設定されます。
置換	サービスをホストするノードのドメイン名。

設定手順

GSLB の設定手順の詳細については、「グローバルサーバー負荷分散 (GSLB) の設定」を参照してください。次の作業を行ってください。

- GSLB 仮想サーバーを追加するときに、次のパラメータを設定します。
  - serviceType: ANY
  - dnsRecordType: NAPTR
  - lbMethod: CUSTOMLOAD

例:

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD
<!--NeedCopy-->
```

- GSLB サイトを追加するときに、NAPTR レコードに埋め込むドメイン名に *naptrReplacementSuffix* パラメータを設定します。

例:

```
add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com
```



```
<!--NeedCopy-->
```

- GSLB サービスを追加するときに、次のパラメータを設定します。
  - naptrreplacement
  - naptrOrder
  - naptrServices
  - naptrDomainTTL
  - naptrPreference

構成例

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD

Done

add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com

Done

add gslb service sgw1 3.3.3.13 ANY *  -siteName site1 -naptrreplacement
     sgw1.site1. -naptrOrder 2 -naptrServices x-3gpp-sgw:x-s5-gtp -
    naptrDomainTTL 20 -naptrPreference   200

Done

add gslb service sgw2 3.3.3.11 ANY *  -siteName site1 -naptrreplacement
     sgw2.site1. -naptrOrder 5 -naptrServices x-3gpp-sgw:x-s5-gtp  -
    naptrDomainTTL 20  naptrPreference  100

Done

add gslb service sgw3 3.3.3.12 ANY *  -siteName site2 -naptrreplacement
     sgw3.site1. -naptrOrder 10 -naptrServices x-3gpp-sgw:x-s5-gtp  -
    naptrDomainTTL 20 naptrPreference   300

bind gslb vserver gslb_vs -serviceName sgw1

Done

bind gslb vserver gslb_vs -serviceName sgw2

Done

```



```
bind gslb vserver gslb_vs -serviceName sgw3

Done

bind gslb service sgw1 -monitorName ping

Done

bind gslb service sgw2 -monitorName ping

Done

bind gslb service sgw3 -monitorName ping

Done

bind gslb vserver gslb_vs -domainName gslb.com -TTL 5

Done
<!--NeedCopy-->
```

> 注
>
> NAPTR レコードを持つ DNS クエリーは、親子構成ではサポートされません。

ワイルドカードドメインの **GSLB** の設定

October 7, 2021

ワイルドカード DNS ドメインを GSLB 仮想サーバーにバインドできます。ワイルドカードドメインの背後にあるアプリケーションにアクセスするユーザーは、それらのアプリケーションをホストする最適なデータセンターにルーティングされます。ワイルドカードドメインは、存在しないドメインおよびサブドメインに対する要求を処理します。ワイルドカードドメインの詳細については、「ワイルドカード DNS ドメインのサポート」を参照してください。

ワイルドカードドメインの GSLB を設定するには、まず基本的な GSLB 設定を設定する必要があります。基本的な GSLB の設定方法の詳細については、GSLB エンティティの個別設定を参照してください。

CLI を使用してワイルドカードドメインの **GSLB** 設定を設定するには

ワイルドカードドメインの GSLB 設定を設定するには、次の手順を実行します。

1. GSLB サイトを作成します。


```
add gslb site site1 10.0.1.10
add gslb site site2 20.0.1.10
<!--NeedCopy-->
```

2. GSLB セットアップに参加している各サイトの GSLB サービスを追加します。

```
add gslb service svc1 -sitename site1 10.0.1.10  http 80
add gslb service svc2 -sitename site1 10.0.1.10  http 80
add gslb service svc3 -sitename site2 20.0.1.10  http 80
add gslb service svc4 -sitename site2 20.0.1.10  http 80
<!--NeedCopy-->
```

3. GSLB セットアップで使用されているサービスを参照する GSLB 仮想サーバーを追加します。

```
add gslb vserver gslb_vs http
<!--NeedCopy-->
```

4. DNS クエリをリッスンする ADNS サービスを追加します。

```
add service adns_udp 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

5. GSLB サービスを GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb_vs -service svc1
bind gslb vserver gslb_vs -service svc2
bind gslb vserver gslb_vs -service svc3
bind gslb vserver gslb_vs -service svc4
<!--NeedCopy-->
```

6. ゾーンを作成します。

```
add dns soaRec test.com -originServer n1.test.com -contact n1.test
    .com
add dns nsrec test.com n1.test.com
add dns nsrec test.com n2.test.com
add dns zone test.com -proxymode no
```



```
<!--NeedCopy-->
```

7. ドメイン名を GSLB 仮想サーバーにバインドします。

```
bind gslb vserver gslb_vs -domainName *.test.com
<!--NeedCopy-->
```

グローバルサーバー負荷分散に **EDNS0** クライアントサブネットオプションを使用する

October 7, 2021

EDNS クライアントサブネット（ECS）は、クライアントサブネットの詳細を提供するドメインネームサーバー（DNS）ヘッダー拡張機能です。これらの詳細を使用して、Citrix ADC グローバルサーバー負荷分散 (GSLB) の精度を向上させることができます。DNS リゾルバの場所ではなく、クライアントのネットワークの場所を使用して、クライアントのトポロジ的な近さを判断します。

> 注
>
> Citrix ADC は、EDNS0 のみをサポートしています。

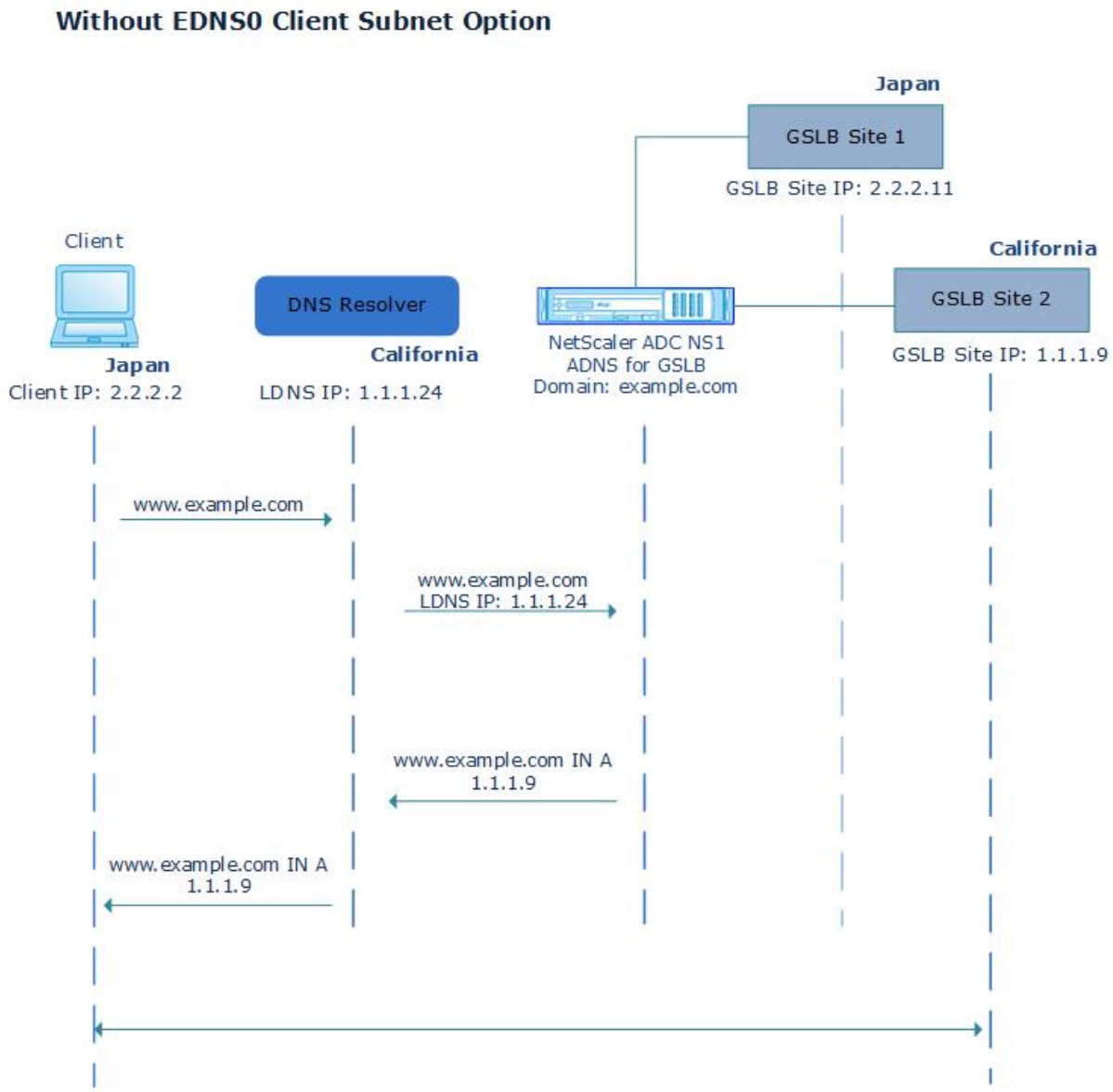
重要:

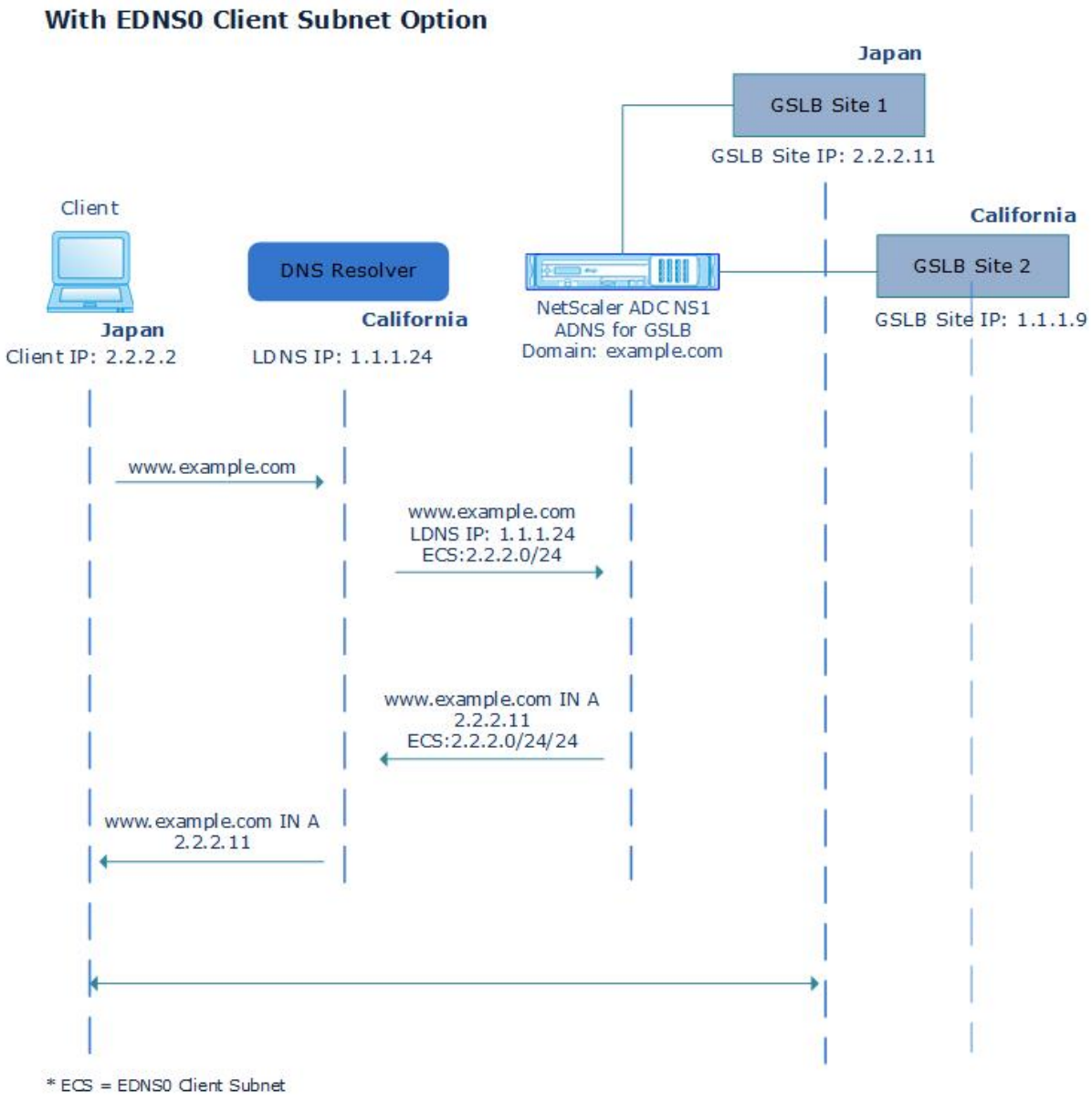
> 展開内のローカルドメインネームサーバー（LDNS）が EDNS0 クライアントサブネットをサポートしていることを確認して、着信 DNS クエリに EDNS0 クライアントサブネットオプションが含まれ、Citrix ADC アプライアンスが DNS クエリの処理中に ECS アドレスを使用するようにします。

Citrix ADC アプライアンスは、クライアントのトポロジの近さを判断するために LDNS IP アドレスを使用し、GSLB を実行します。そのため、静的近接や動的ラウンドトリップ時間（RTT）などの近接ベースの負荷分散方法を使用します。これは、一般的な GSLB 展開で発生します。ただし、Google DNS や OpenDNS などの集中型 DNS リゾルバーが展開に関与している場合、Citrix ADC アプライアンスは DNS 要求を、クライアントに近くない可能性がある集中型 DNS リゾルバーに近いデータセンターに送信します。たとえば、静的近接負荷分散方式を使用する一般的な Citrix ADC GSLB 展開では、日本からのエンドユーザー要求は日本のデータセンターに送信され、カリフォルニアからのエンドユーザー要求はカリフォルニアのデータセンターに送信されます。ただし、集中型 DNS リゾルバーが関係している場合、Citrix ADC アプライアンスは日本からカリフォルニアのデータセンターに要求を送信する可能性があります。

ECS オプションは、GSLB ドメインの権威 DNS（ADNS）サーバーとして構成された Citrix ADC アプライアンスを含む展開で使用できます。ロードバランシング方式として静的近接を使用する場合は、LDNS IP アドレスの代わりに EDNS ヘッダーで IP サブネットを使用できます。これは、クライアントの地理的な近さを判断するのに役立ちます。プロキシモードの展開では、Citrix ADC アプライアンスは ECS 対応の DNS クエリをそのままバックエンドサーバーに転送します。アプライアンスは、ECS 対応の DNS 応答をキャッシュしません。

> 注
>
> ECS オプションは、GSLB 以外のドメインの ADNS モード、リゾルバモード、フォワーダモードなど、他のすべての展開モードには適用されません。ECS オプションは、前述のモードの Citrix ADC アプライアンスでは無視されます。また、デフォルトでは、GSLB デプロイメントでは ECS が無効になっています。





コマンドラインインターフェイスを使用して **EDNS0** クライアントサブネットオプションを有効にするには:

コマンドプロンプトで入力します。

```
set gslb vserver <vserver_name> **-ECS ENABLED

set gslb vserver vserver-GSLB-1 -ECS ENABLED
<!--NeedCopy-->
```


アドレスの検証

DNS クエリの EDNS0 Client Subnet (ECS) オプションによって返されたアドレスが、プライベート IP アドレスまたはルーティング不能な IP アドレスではないことを確認するように GSLB 仮想サーバーを構成できます。アドレス検証が有効な場合、Citrix ADC アプライアンスは DNS クエリ内の ECS アドレスを次の表にリストしている場合は無視し、代わりに LDNS IP アドレスを使用してグローバルサーバーの負荷分散を行います。

> 注
>
> デフォルトでは、アドレス検証は無効になっています。

アドレスの種類	アドレス	説明
IPV4	10.0.0.0/8	私的使用の場合
	172.16.0.0/12	私的使用の場合
	192.168.0.0/16	私的使用の場合
	0.0.0.0/8	ネットワーク上のホストを参照します。
	100.64.0.0/10	共有アドレス空間
	127.0.0.0/8	ループバックアドレス
	169.254.0.0/16	RFC 3927 で定義されているリンクローカル IPv4 アドレス
	192.0.0.0/24	IETF プロトコルの割り当てに使用され、プライベートスペース 192.168.0.0/16 が含まれます。
	192.0.2.0/24	ドキュメント作成の目的で使用します。
	192.88.99.0/24	6to4 Relay Anycast に使用
	198.18.0.0/15	デバイスのベンチマークテストに使用
	198.51.100.0/24	ドキュメント作成の目的で使用します。
	203.0.113.0/24	ドキュメント作成の目的で使用します。
	240.0.0.0/4	予約済みとして使用
	255.255.255.255/32	ブロードキャストに使用

アドレスの種類	アドレス	説明
IPv6	::1/128	ループバックアドレス
	::/128	未指定のアドレス
	::ffff:0:0/96	IPv4 マッピングアドレス
	100::/64	廃棄のみのアドレスブロック
	2001::/23	IETF プロトコルの割り当てに使用
	2001::/32	TEREDO
	2001:2::/48	ベンチマークに使用
	2001:db8::/32	ドキュメント作成の目的で使用します。
	2001:10::/28	ORCHID
	2002::/16	6to4 Relay Anycast に使用
	fc00::/7	一意ローカル
	fe80::/10	リンクローカルユニキャストアドレス

コマンドラインインターフェイスを使用してアドレスの検証を有効にするには

コマンドプロンプトで入力します。

```
set gslb vserver <vserver_name> -ecsAddrValidation ENABLED

set gslb vserver vserver-GSLB-1 -ecsAddrValidation ENABLED
<!--NeedCopy-->
```

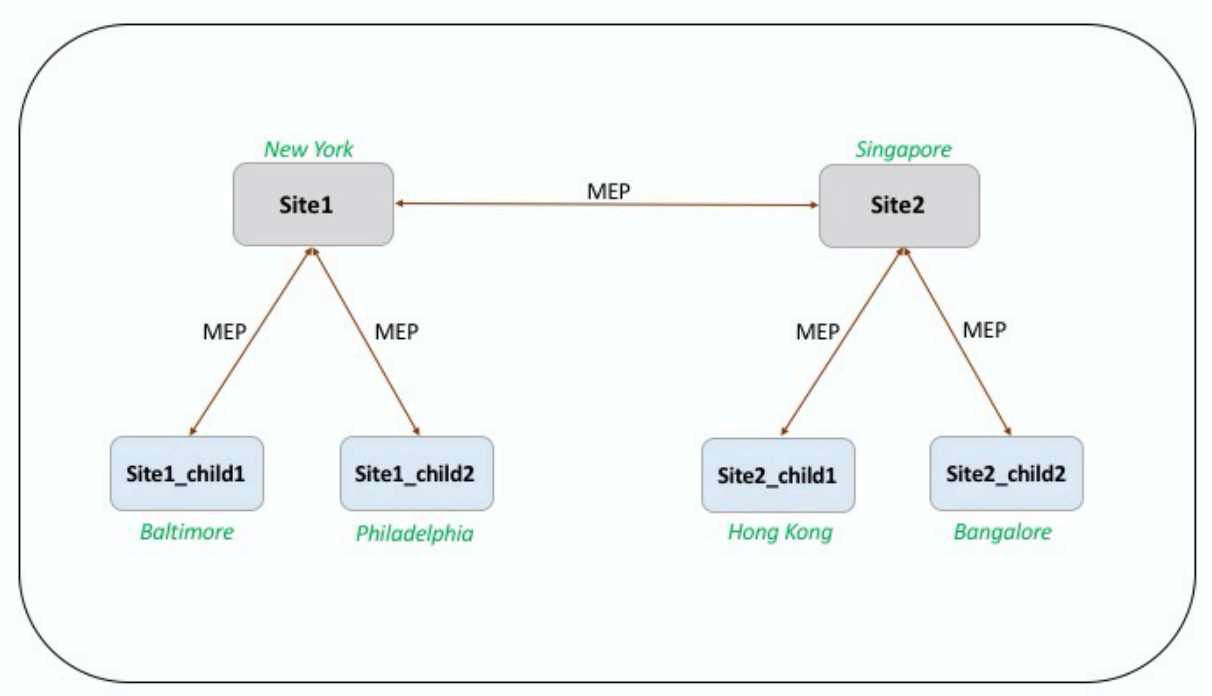
メトリック交換プロトコルを使用した完全な親子設定の例

October 7, 2021

GSLB サイトがグローバルに分散されている次の親子トポロジを考えてみましょう。

- Site1 と Site2 は親サイトです。
- Site1_child1 と Site1_child2 は、Site1 の子サイトです。

- Site2_child1 および Site2_child2 は、Site2 の子サイトです。



次のコマンドは、親子トポロジの完全な設定を示しています。

site1

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
```



```
12
13 add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
      publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
      site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
14
15 add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
       10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
      cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
16
17 add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
      publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
      site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
18
19 add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
       10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
      cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
20
21 add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
      appflowLog DISABLED
22
23 bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1
24
25 bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1
26
27 bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1
28
29 bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1
30
31 bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
32 <!--NeedCopy-->
```

site1_child1

```
1 add gslb site site1 10.102.82.164 -publicIP 10.102.82.164
2
3 add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
      nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
      site1
4 <!--NeedCopy-->
```

ロードバランシング設定には、次のコマンドを追加できます。


```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.102.82.132 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site1_child2

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

You can add the following commands for load balancing configuration:

add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.102.82.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

```



```
7  add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
       nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
       site1
8
9  add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
       nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
       site2
10
11 add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
       nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
       site2
12
13 add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
       publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
       site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
14
15 add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
        10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
       cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
16
17 add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
       publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
       site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
18
19 add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
        10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
       cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
20
21 add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
       appflowLog DISABLED
22
23 bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1
24
25 bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1
26
27 bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1
28
29 bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1
30
31 bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
32 <!--NeedCopy-->
```

site2_child1


```
add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.134 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2_child2

```
add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
\`\`\`
<!--NeedCopy-->
```


リンク負荷分散

October 7, 2021

Link Load Balancing（LLB; リンク負荷分散）は、異なるサービスプロバイダが提供する複数のインターネット接続間で発信トラフィックを分散します。LLB を使用すると、Citrix ADC アプライアンスでトラフィックを監視および制御できるため、パケットは可能な限り最良のリンクを介してシームレスに送信されます。LLB では、サービスがサーバを表すサーバ負荷分散とは異なり、サービスはルータまたはネクストホップを表します。リンクは、Citrix ADC アプライアンスとルーター間の接続です。

リンク負荷分散を構成するには、多くのユーザが基本設定をデフォルト設定で構成することから始めます。基本的なセットアップには、サービス、仮想サーバー、モニター、ルート、LLB メソッド、および永続性（オプション）が含まれます。基本セットアップが運用可能になったら、環境に合わせてカスタマイズできます。

LLB に適用可能な負荷分散方式は、ラウンドロビン、宛先 IP ハッシュ、最小帯域幅、および最小パケットです。オプションで、特定のリンクで維持される接続の永続性を構成できます。使用可能な永続性タイプは、送信元 IP アドレスベース、宛先 IP アドレスベース、送信元 IP アドレスベースと宛先 IP アドレスベースです。PING がデフォルトのモニタですが、トランスペアレントモニタを設定することをお勧めします。

リバース NAT（RNAT）とバックアップリンクを設定することで、設定をカスタマイズできます。

基本的な **LLB** セットアップの設定

October 7, 2021

LLB を設定するには、まず、インターネットサービスプロバイダ (ISP) に対する各ルータを表すサービスを作成します。デフォルトでは、PING モニターは各サービスにバインドされます。トランスペアレントモニタのバインドはオプションですが、推奨されます。次に、仮想サーバーを作成し、サービスを仮想サーバーにバインドし、仮想サーバーのルートを構成します。ルートは、仮想サーバをサービスによって表される物理ルータへの Gateway として識別します。仮想サーバは、指定したロードバランシング方式を使用してルータを選択します。任意で、特定のセッションのすべてのトラフィックが特定のリンクで送信されるように永続性を設定できます。

基本的な LLB 設定を設定するには、次の手順を実行します。

- サービスの構成
- LLB 仮想サーバーの構成とサービスのバインド
- LLB 方式と永続性の設定
- LLB ルートの設定
- トランスペアレントモニタの作成とバインド

サービスの構成

デフォルトモニタ（PING）は、サービスの作成時にサービスタイプの ANY に自動的にバインドされますが、トランスペアレントモニタの作成とバインドの説明に従って、デフォルトモニタをトランスペアレントモニタに置き換えることができます。

コマンドラインインターフェイスを使用してサービスを作成するには

コマンドプロンプトで入力します。

```
add service <name> <IP> <serviceType> <port>

show service <name>
<!--NeedCopy-->
```

例:

```
add service ISP1R_svc_any 10.10.10.254 any *
show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: DOWN
        Last state change was at Tue Aug 31 04:31:13 2010
        Time since last state change: 2 days, 05:34:18.600
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 244705  Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
```



```
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスを作成するには

Traffic Management > Load Balancing > Services に移動してサービスを作成します。

構成ユーティリティを使用してサービスを作成するには

1. ［Traffic Management］ > ［Load Balancing］ > ［Services］の順に移動します。
2. 詳細ペインで、［Add］をクリックします。
3. [サービスの作成] ダイアログボックスで、次のパラメータの値を指定します。
   - サービス名 *—name
   - サーバ：IP
   - プロトコル *：serviceType（ドロップダウンリストから [ANY] を選択します）。
   - ポート *：port

必須パラメータ

1. ［作成］をクリックします。
2. ステップ 2 ～4 を繰り返して、別のサービスを作成します。
3. ［閉じる］をクリックします。
4. [Services] ペインで、構成したサービスを選択し、画面の下部に表示される設定が正しいことを確認します。

LLB 仮想サーバーの構成とサービスのバインド

サービスを作成したら、仮想サーバーを作成し、サービスを仮想サーバーにバインドします。LLB では、最小接続のデフォルトの LB 方式はサポートされていません。LB 方式の変更については、LLB 方式と永続性の設定を参照してください。

リンク負荷分散仮想サーバーを作成し、コマンドラインインターフェイスを使用してサービスをバインドするには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType>

bind lb vserver < name> <serviceName>
```



```
4
5 show lb vserver < name>
6 <!--NeedCopy-->
```

例:

```
1 add lb vserver LLB-vip any
2 bind lb vserver LLB-vip ISP1R_svc_any
3 sh lb vserver LLB-vip
4         LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
5         State: DOWN
6         Last state change was at Thu Sep  2 10:51:32 2010
7         Time since last state change: 0 days, 17:51:46.770
8         Effective State: DOWN
9         Client Idle Timeout: 120 sec
10        Down state flush: ENABLED
11        Disable Primary Vserver On Down : DISABLED
12        No. of Bound Services :  1 (Total)       0 (Active)
13        Configured Method: ROUNDROBIN
14        Mode: IP
15        Persistence: NONE
16        Connection Failover: DISABLED
17
18 1) ISP1R_svc_any (10.10.10.254: *) - ANY State: DOWN    Weight: 1
19  Done
20 <!--NeedCopy-->
```

リンク負荷分散仮想サーバーを作成し、構成ユーティリティを使用してサービスをバインドするには

1. トラフィック管理に移動します > 負荷分散 > 仮想サーバー、およびリンク負荷分散用の仮想サーバーを作成します。[プロトコル] フィールドに **ANY** を指定します。
2. [**IP** アドレスの種類] ドロップダウンリストで、目的のオプションを選択します。直接アクセスできない仮想サーバを作成するには、[非アドレス可能] を選択します。
3. [サービス] タブの [アクティブ] 列で、仮想サーバーにバインドするサービスのチェックボックスをオンにします。

LLB 方式と永続性の設定

デフォルトでは、Citrix ADC アプライアンスは最小接続方式を使用して各クライアント要求をリダイレクトするサービスを選択しますが、LLB 方式はサポートされている方式のいずれかに設定する必要があります。また、同じクライアントからの異なる転送が同じサーバーに送信されるように、永続性を設定することもできます。

コマンドラインインターフェイスを使用して **LLB** メソッドまたはパーシステンスを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
set lb vserver <name> -lbMethod <lbMethod> -persistenceType <
    persistenceType>

show lb vserver <name>
<!--NeedCopy-->
```

例:

```
set lb vserver LLB-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 00:52:21.200
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: SOURCEIP
     Persistence Mask: 255.255.255.255       Persistence v6MaskLength:
         128   Persistence Timeout: 2 min
        Connection Failover: DISABLED
<!--NeedCopy-->
```

構成ユーティリティを使用してリンクロードバランシング方式またはパーシステンス（あるいはその両方）を構成するには

1. 「トラフィック管理」>「負荷分散」>「仮想サーバー」に移動し、負荷分散方式または持続性設定を構成する仮想サーバーを選択します。
2. [詳細設定] セクションで、[方法] を選択し、負荷分散方法を構成します。
3. [詳細設定] セクションで、[持続性] を選択し、持続性パラメータを設定します。

LLB ルートの設定

IPv4 または IPv6 サービス、仮想サーバ、LLB メソッド、および永続性を設定した後、LLB 仮想サーバを Gateway として指定する、ネットワークの IPv4 または IPv6 LLB ルートを設定します。ルートは、ロードバランシングされたリンクの集合です。要求は、すべての発信トラフィックの Gateway として機能する LLB 仮想サーバの IP アドレスに送信され、設定された LLB 方式に基づいてルータが選択されます。

コマンドラインインターフェイスを使用して **IPv4 LLB** ルートを構成するには

コマンドプロンプトで入力します。

```
add lb route <network> <netmask> <gatewayName>

show lb route [<network> <netmask>]
<!--NeedCopy-->
```

例:

```
add lb route 0.0.0.0 0.0.0.0 LLB-vip
show lb route 0.0.0.0 0.0.0.0
        Network          Netmask        Gateway/VIP              Flags
      -----------      -------------   ---------------          --------
1)      0.0.0.0          0.0.0.0        LLB-vip                UP
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **IPv6 LLB** ルートを構成するには

コマンドプロンプトで入力します。

```
add lb route6 <network> <gatewayName>

show lb route6
<!--NeedCopy-->
```

例**:**

```
add lb route6 ::/0 llb6_vs show lb route6 Network VIP Flags -----------
     --------- -------- 1) ::/0 llb6_vs UP
```



```
<!--NeedCopy-->
```

設定ユーティリティを使用して **LLB** ルートを設定するには

[システム] > [ネットワーク] > [ルート] に移動し、[**LLB**] を選択し、LLB ルートを設定します。

注： IPV6 ルートを設定するには、LLBV6 を選択します。

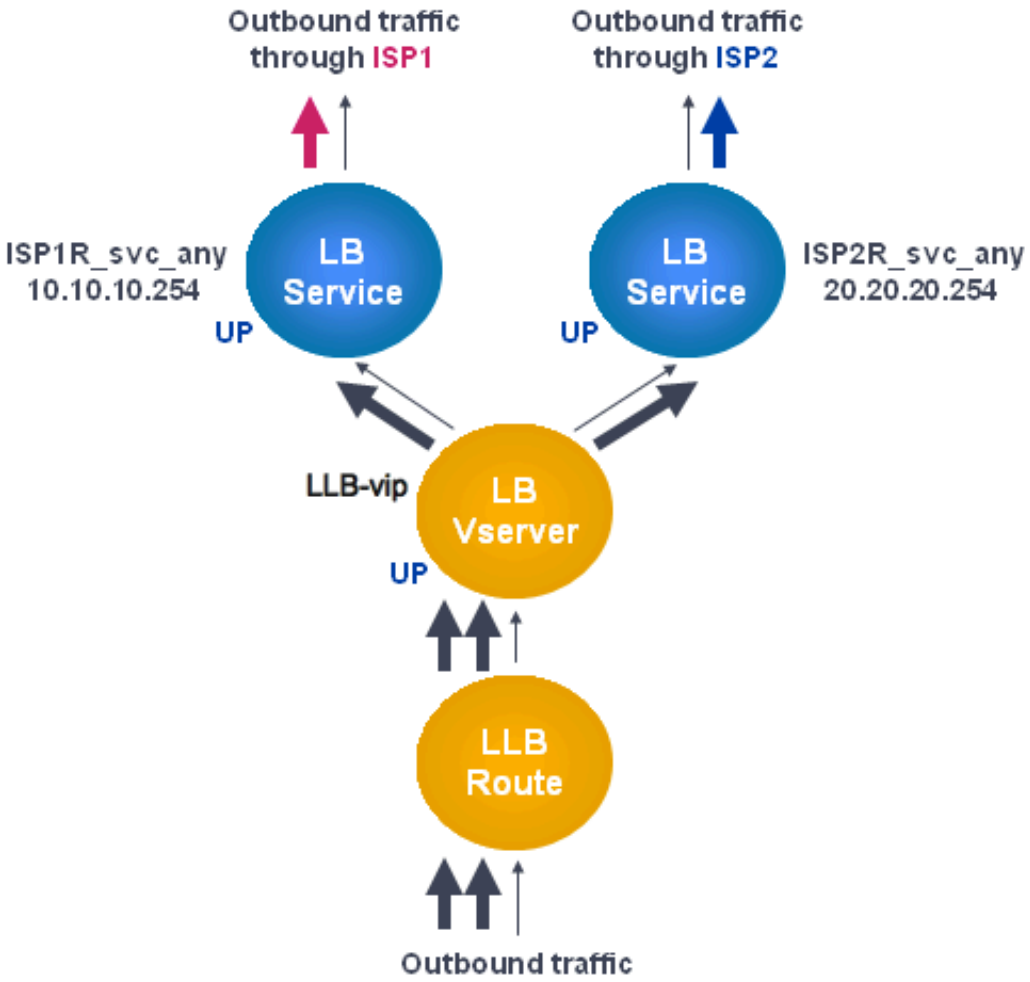
設定ユーティリティを使用して **LLB** ルートを設定するには

1. [システム] > [ネットワーク] > [ルート] に移動します。
2. 詳細ペインで、次のいずれかを選択します。
   - IPv4 ルートを設定するには、[LLB] をクリックします。
   - IPv4 ルートを設定するには、[LLBV6] をクリックします。
3. [LB ルートの作成] または [LB IPV6 ルートの作成] ダイアログボックスで、次のパラメータを設定します。
   - ネットワーク *
   - ネットマスク *： IPV4 ルートに必要です。
   - ゲートウェイ名 *： gatewayName

     * 必須パラメータ
4. [Create] をクリックしてから、[Close] をクリックします。作成したルートが [Routes] ペインの [LLB] タブまたは [LLB6] タブに表示されます。

次の図は、基本的な LLB 設定を示しています。サービスが 2 つのリンク (ISP) ごとに構成され、PING モニターはデフォルトでこれらのサービスにバインドされます。リンクは、設定された LLB 方式に基づいて選択されます。

図 1： 基本的な LLB セットアップ



> 注
>
> インターネットサービスプロバイダが IPv6 アドレスを提供している場合は、上の図の IPv4 サービスを IPv6 サービスと置き換えます。

トランスペアレントモニタの作成とバインド

トランスペアレントモニタを作成して、ルータなどのアップストリームデバイスの状態を監視します。その後、透過モニターをサービスにバインドできます。デフォルトの PING モニターは、Citrix ADC アプライアンスとアップストリームデバイス間の接続のみを監視します。トランスペアレントモニタは、アプライアンスからモニタで指定された宛先 IP アドレスを所有するデバイスへのパスに存在するすべてのデバイスを監視します。トランスペアレントモニタが設定されておらず、ルータのステータスが UP であるにもかかわらず、そのルータからのネクストホップデバイスのいずれかがダウンしている場合、アプライアンスはロードバランシングの実行中にルータを含み、パケットをルータに転送します。ただし、ネクストホップデバイスのいずれかがダウンしているため、パケットは最終的な宛先に配信されません。トランスペアレントモニタをバインドすることにより、デバイス（ルータを含む）のいずれかがダ

ウンしている場合、サービスは DOWN としてマークされ、アプライアンスがリンクロードバランシングを実行するときにルータは含まれません。

コマンドラインインターフェイスを使用してトランスペアレントモニタを作成するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> -destIP <ip_addr|*> -transparent
   YES

show lb monitor [<monitorName>]
<!--NeedCopy-->
```

例:

```
add lb monitor monitor-1 PING -destIP 10.10.10.11 -transparent YES
> show lb monitor monitor-1
1)   Name.......: monitor-1  Type......:      PING   State....:
   ENABLED
Standard parameters:
  Interval.........:            5 sec   Retries...........:
                    3
  Response timeout.:            2 sec   Down time.........:
     30 sec
  Reverse..........:              NO   Transparent.......:
                  YES
  Secure...........:              NO   LRTM..............:
     ENABLED
  Action...........:   Not applicable   Deviation.........:
     0 sec
  Destination IP...:      10.10.10.11
  Destination port.:    Bound service
  Iptunnel.........:              NO
  TOS..............:              NO   TOS ID............:
                    0
  SNMP Alert Retries:               0     Success Retries..:
                    1
  Failure Retries..:                0
<!--NeedCopy-->
```


構成ユーティリティを使用してトランスペアレントモニタを作成するには

[トラフィック管理] > [負荷分散] > [モニター] に移動し、トランスペアレントモニターを設定します。

構成ユーティリティを使用してトランスペアレントモニタを作成するには

1. Traffic Management > Load Balancing > Monitors に移動します。
2. [モニター] ウィンドウで、[追加] をクリックします。
3. [モニタの作成] ダイアログボックスで、次のパラメータを設定します。
   - Name*
   - タイプ *
   - 接続先 IP
   - 透明

   * 必須パラメータ
4. [Create] をクリックしてから、[Close] をクリックします。
5. [モニター] ペインで、構成したモニタを選択し、[詳細] ペインに表示される設定が正しいことを確認します。

構成ユーティリティを使用してモニタをサービスにバインドするには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. [モニター] タブの [使用可能] で、サービスにバインドするモニターを選択し、[追加] をクリックします。

コマンドラインインターフェイスを使用してモニタをサービスにバインドするには

コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <serviceName>

show service <name>
<!--NeedCopy-->
```

例:

```
bind lb monitor monitor-HTTP-1 ISP1R_svc_any
 Done
> show service ISP1R_svc_any
```



```
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: UP
        Last state change was at Thu Sep  2 10:51:07 2010
        Time since last state change: 0 days, 18:41:55.130
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP  Weight: 1
                Probes: 1256       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用してモニタをサービスにバインドするには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. 詳細ウィンドウで、モニタをバインドするサービスを選択し、[開く] をクリックします。
3. [サービスの構成] ダイアログボックスの [モニター] タブの [使用可能] で、サービスにバインドするモニターを選択し、[追加] をクリックします。
4. [OK] をクリックします。
5. [サービス] ペインで、構成したサービスを選択し、[詳細] ペインに表示される設定が正しいことを確認します。

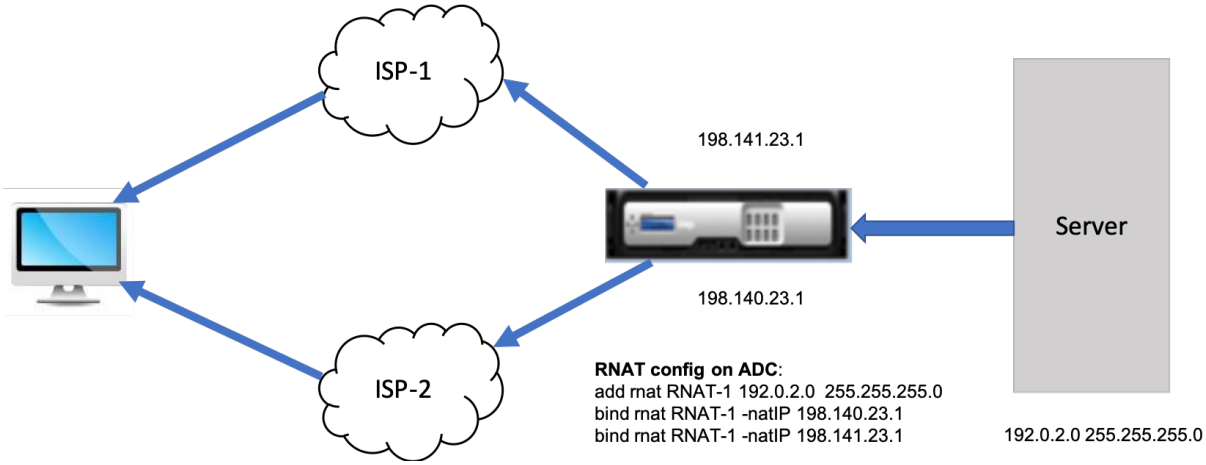
LLB で RNAT を構成する

October 7, 2021

発信トラフィックのリバースネットワークアドレス変換 (RNAT) の LLB 設定を構成できます。これにより、特定のフ

ローのリターンネットワークトラフィックが同じパスを経由するようになります。基本的な LLB セットアップの設定の説明に従って基本的な LLBを設定し、次に RNAT の設定の説明に従って RNAT を設定します。次に、「サブネット IP（USNIP）を使用する」モードを有効にします。

次の図では、Citrix ADC アプライアンスが LLB を使用してアウトバウンドトラフィックをさまざまなリンクにルーティングしています。RNAT の動作中に、ADC アプライアンスはアウトバウンドトラフィックの送信元 IP アドレスをパブリック NAT IP アドレス（198.141.23.1）に置き換えて、トラフィックを ISP-1 経由でルーティングします。同様に、ADC アプライアンスは送信元 IP アドレスを 198.140.23.1 に置き換えて、トラフィックを ISP-2 経由でルーティングします。



CLI を使用して **ISP** ルーターの **SNIP** を追加するには

コマンドプロンプトで入力します。

```
add NS IP <subnet of first ISP in the IP router> <subnet mask> -type
    SNIP

add NS IP <subnet of second ISP in the IP router> <subnet mask> -type
    SNIP
<!--NeedCopy-->
```

例:

```
add ns ip 198.140.23.1 255.255.255.0 -type snip

add ns ip 198.141.23.1 255.255.255.0 -type snip
<!--NeedCopy-->
```

CLI を使用して RNAT を構成するには

コマンドプロンプトで入力します。

```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ …

show rnat <name>
<!--NeedCopy-->
```

例:

```
add rnat RNAT-1 192.0.2.0 255.255.255.0
bind rnat RNAT-1 -natIP 198.140.23.1
bind rnat RNAT-1 -natIP 198.141.23.1

> show rnat RNAT-1
    1) RNAT Name: RNAT-1     Network: 192.0.2.0      Netmask:
       255.255.255.0     Traffic Domain: 0
        UseProxyPort: ENABLED

        NatIP: 198.140.23.1
        NatIP: 198.141.23.1
<!--NeedCopy-->
```

GUI を使用して RNAT を構成するには

1. [システム] > [ネットワーク] > [**NAT**] に移動します。
2. [**RNAT**] タブで、[**RNAT** の構成] をクリックします。
3. RNAT を実行するネットワークを指定します。

> 注
>
> アクセスコントロールリスト（ACL）を使用して RNAT を設定することもできます。詳細については、「RNAT の設定」を参照してください。

CLI を使用してサブネット IP モードを使用を有効にするには

コマンドプロンプトで入力します。


```
enable ns mode USNIP

show ns mode
<!--NeedCopy-->
```

例:

```
enable ns mode USNIP

show ns mode
        Mode                            Acronym             Status
        -------                         -------             ------
 1)     Fast Ramp                       FR                  ON
 2)     … .
 8)     Use Subnet IP                   USNIP               ON
 9)     …
<!--NeedCopy-->
```

GUI を使用してサブネット **IP** モードを使用を有効にするには

1. [システム] **>** [設定] に移動し、[モードと機能] で [モードの構成] をクリックします。
2. [モードの構成] ダイアログボックスで、[サブネット **IP** を使用] を選択し、[**OK**] をクリックします。

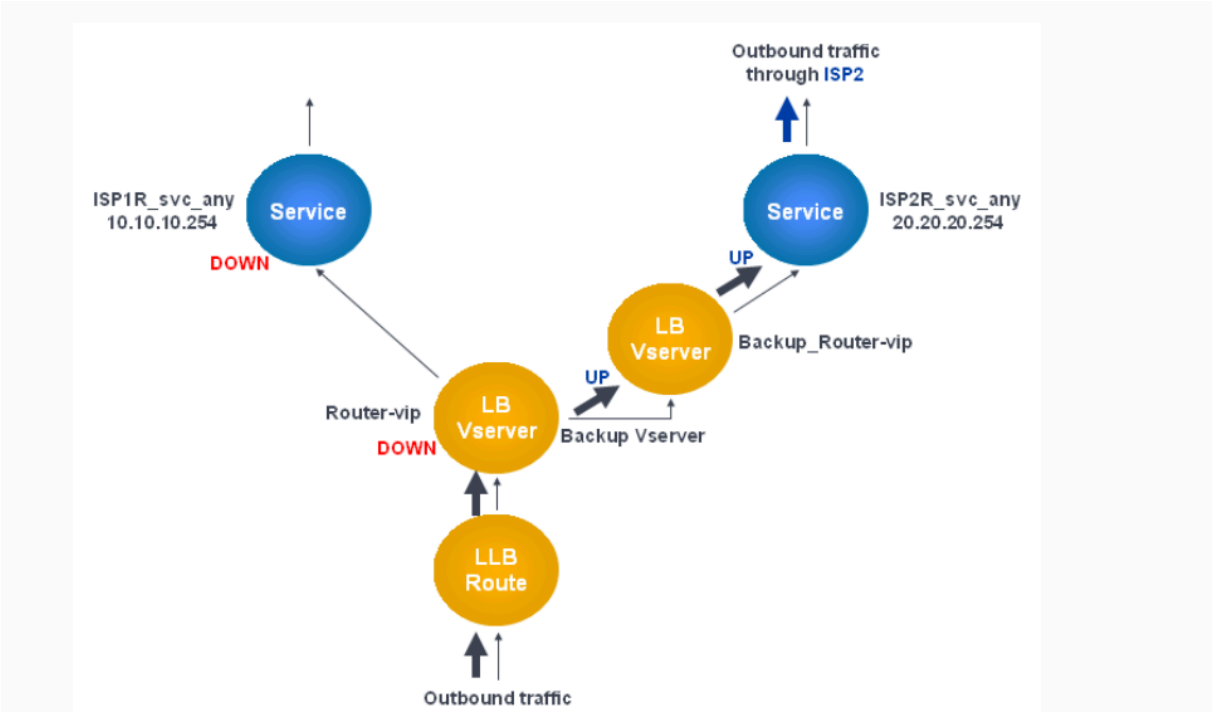
バックアップルートを構成する

October 7, 2021

プライマリルートがダウンしたときにサービスが中断されないように、バックアップルートを設定できます。バックアップルートが構成されると、Citrix ADC アプライアンスはプライマリルートに障害が発生したときにそのルートを自動的に使用します。最初に、LLB 仮想サーバの設定およびサービスのバインドの説明に従って、プライマリ仮想サーバを作成します。バックアップルートを構成するには、プライマリ仮想サーバと同様のセカンダリ仮想サーバを作成し、この仮想サーバをバックアップ仮想サーバ（ルート）として指定します。

次の図では、**Router-vip** がプライマリ仮想サーバであり、**Backup_Router-vip** がバックアップ仮想サーバとして指定されているセカンダリ仮想サーバです。

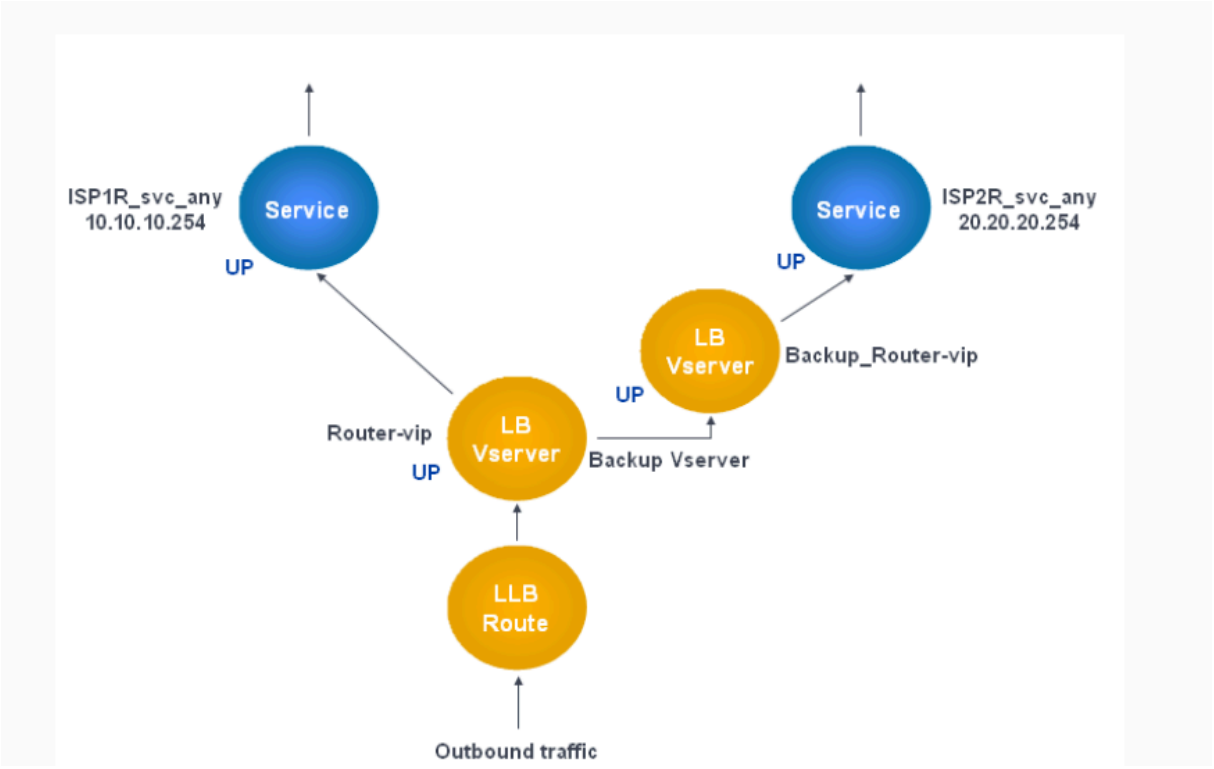
図 1: バックアップルート設定



注： ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えます。

デフォルトでは、すべてのトラフィックがプライマリルート経由で送信されます。ただし、プライマリルートに障害が発生すると、次の図に示すように、すべてのトラフィックがバックアップルートに迂回されます。

図 2：運用中のルーティングのバックアップ



注: ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えます。

コマンドラインインターフェイスを使用してセカンダリ仮想サーバをバックアップ仮想サーバとして設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

例:

```
set lb vserver Router-vip -backupVServer Backup_Router-vip
> show lb vserver Router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 03:09:45.600
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: DESTIP     Persistence Mask: 255.255.255.255
                  Persistence v6MaskLength: 128   Persistence Timeout: 2
            min
        Backup: Router2-vip
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用してセカンダリ仮想サーバをバックアップ仮想サーバとして設定するには

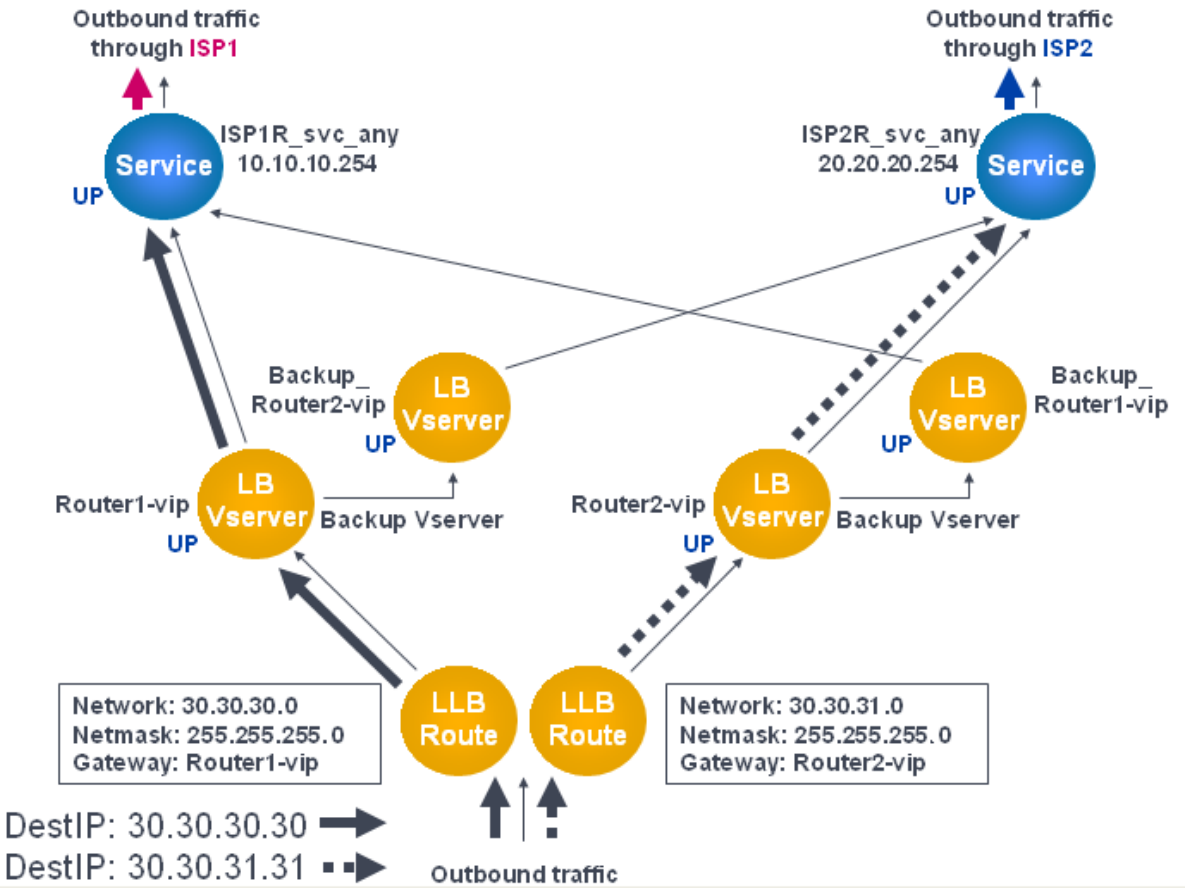
1. **[**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動し、バックアップ仮想サーバーを設定するセカンダリ仮想サーバーを選択します。
2. [仮想サーバーの負荷分散] ダイアログボックスの [詳細] で、[保護] を選択します。
3. [仮想サーバーのバックアップ] ドロップダウンリストで、セカンダリバックアップ仮想サーバーを選択し、[**OK**] をクリックします。

回復力のある **LLB** 展開シナリオ

October 7, 2021

次の図には、30.30.30.0 と 30.30.31.0 という 2 つのネットワークがあります。リンク負荷分散は、宛先 IP アドレスに基づいて設定されます。2 つのルートは、それぞれゲートウェイ **Router1-vip** および **Router2-vip** を使用して設定されます。**Router1-vip** は、**Router2-vip** のバックアップとして構成されています。宛先 IP が 30.30.30.30 として指定されたすべてのトラフィックは、**Router1-vip** を介して送信され、宛先 IP が 30.30.31.31 として指定されたトラフィックは、**Router2-vip** を介して送信されます。

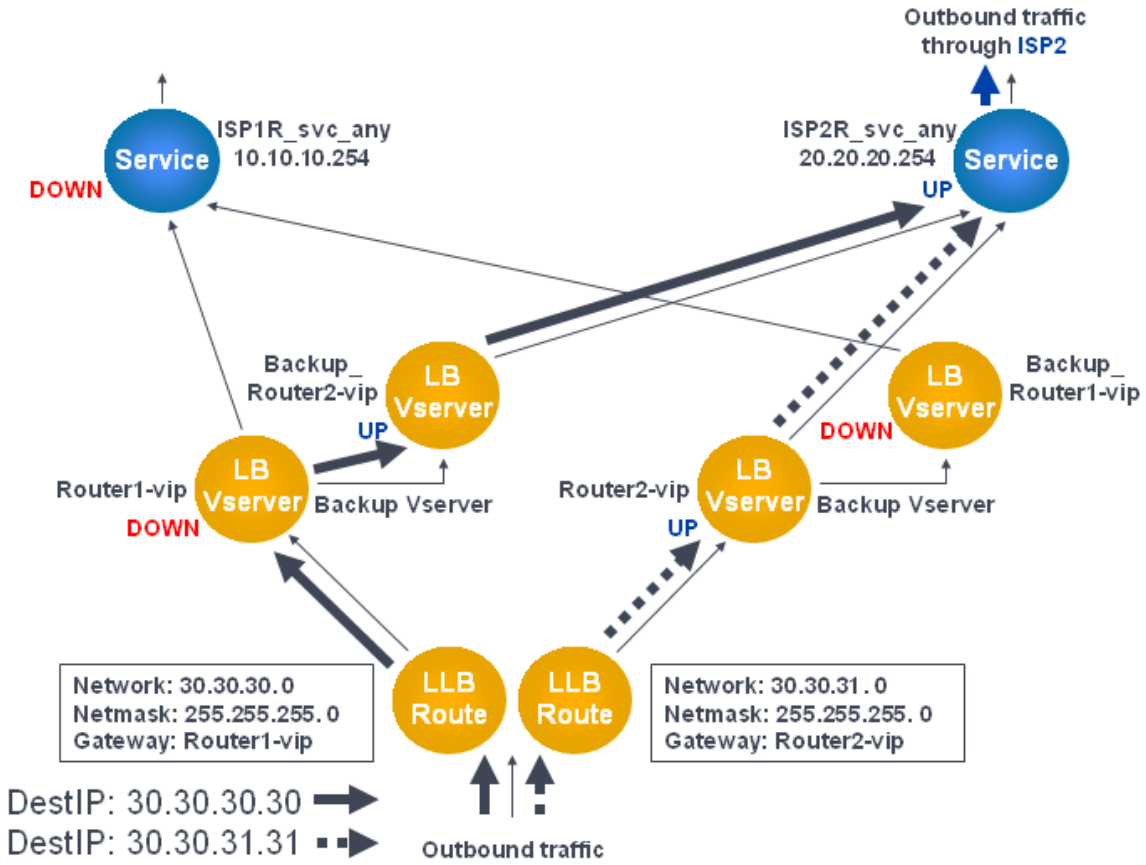
図 1: 耐障害性 LLB 展開セットアップ



注: ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えます。

ただし、いずれかのゲートウェイ（**Router1-vip** または **Router2-vip**）が DOWN の場合、トラフィックはバックアップルータを経由してルーティングされます。次の図では、ISP1 の **Router1-vip** が DOWN であるため、宛先 IP が 30.30.30.30 として指定されたすべてのトラフィックも ISP2 経由で送信されます。

図 2: 耐障害性 LLB 展開シナリオ



注: ISP が IPv6 アドレスを提供している場合は、IPv4 サービスを前の図の IPv6 サービスに置き換えます。

LLB セットアップを監視する

October 7, 2021

構成が稼働したら、各サービスと仮想サーバーの統計を表示して、考えられる問題を確認できます。

仮想サーバーの統計を表示する

仮想サーバーのパフォーマンスを評価したり、問題のトラブルシューティングを行うために、Citrix ADC アプライアンスで構成された仮想サーバーの詳細を表示できます。すべての仮想サーバーの統計の要約を表示できます。仮想サーバーの名前を指定して、その仮想サーバーの統計のみを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル

- 仮想サーバの状態
- 受信したリクエストの割合
- Rate of hits

CLI を使用して仮想サーバーの統計を表示する

Citrix ADC に現在構成されているすべての仮想サーバー、または単一の仮想サーバーの統計情報のサマリーを表示するには、コマンドプロンプトで次のように入力します。

```
stat lb vserver -detail] [<name>]
<!--NeedCopy-->
```

例:

```
stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
        0/s
Two                          *     0          TCP         DOWN      0/s
        0/s
Three                        *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done
<!--NeedCopy-->
```

GUI を使用して仮想サーバーの統計を表示する

1. **Traffic Management > Load Balancing > Virtual Servers > Statistics** に移動します。
2. 1 つの仮想サーバーのみの統計を表示する場合は、詳細ペインで仮想サーバーを選択し、[統計] をクリックします。

サービスの統計を表示する

サービス統計を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバー接続などのレートを表示できます。

CLI を使用してサービスの統計を表示する

コマンドプロンプトで入力します。

```
stat service <name>
<!--NeedCopy-->
```

例:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスの統計を表示する

1. **Traffic Management > Load Balancing > Virtual Servers > Statistics** に移動します。
2. 1 つのサービスのみの統計を表示する場合は、サービスを選択して、[統計] をクリックします。

負荷分散

October 7, 2021

負荷分散機能は、Web ページおよびその他の保護されたアプリケーションに対するユーザー要求を、すべて同じコンテンツをホスト（またはミラー）する複数のサーバーに分散します。負荷分散は、主に使用頻度の高いアプリケーションへのユーザー要求を管理するために使用します。これにより、パフォーマンスの低下や停止を防ぎ、保護されたアプリケーションにユーザーがアクセスできるようになります。ロードバランシングは、フォールトトレランスも提供します。保護されたアプリケーションをホストする 1 つのサーバが使用できなくなると、この機能により、同じアプリケーションをホストする他のサーバにユーザ要求が分散されます。

ロードバランシング機能は、次のように設定できます。

- 特定の保護された Web サイト、アプリケーション、またはリソースに対するすべての要求を、同じ構成の 2 つ以上のサーバー間で分散します。

- いくつかの異なるアルゴリズムのいずれかを使用して、現在のユーザー接続数が最も少ないサーバーや負荷が最も軽いサーバーなど、さまざまな要因に基づいて各受信ユーザー要求を受信する必要があるサーバーを特定します。

負荷分散機能は、Citrix ADC アプライアンスのコア機能です。ほとんどのユーザーは、まず動作中の基本構成をセットアップしてから、接続の永続性などのさまざまな設定をカスタマイズします。さらに、障害からの構成の保護、クライアントトラフィックの管理、サーバーの管理と監視、および大規模な展開の管理のための機能を構成できます。

負荷分散の仕組み

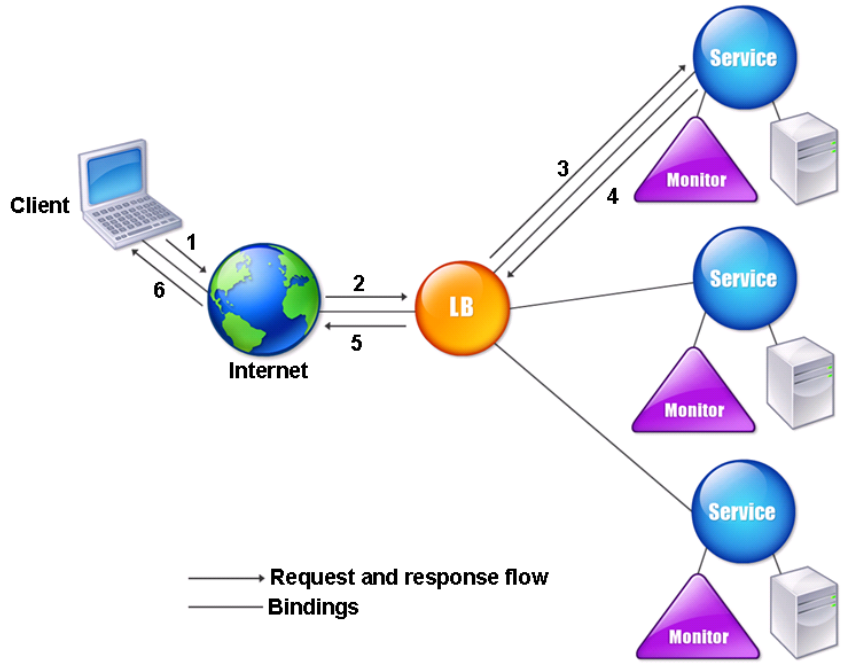
October 7, 2021

基本的な負荷分散設定では、クライアントは Citrix ADC アプライアンスで構成された仮想サーバーの IP アドレスに要求を送信します。仮想サーバーは、負荷分散アルゴリズムと呼ばれる事前設定されたパターンに従って、負荷分散アプリケーションサーバーに分散します。場合によっては、特定の IP アドレスではなく、負荷分散仮想サーバーにワイルドカードアドレスを割り当てたい場合があります。アプライアンスのグローバル HTTP ポートを指定する手順については、「グローバル **HTTP** ポート」を参照してください。

負荷分散の基本

負荷分散の設定には、負荷分散仮想サーバーと複数の負荷分散アプリケーションサーバーが含まれます。仮想サーバーは、着信クライアント要求を受信し、負荷分散アルゴリズムを使用してアプリケーションサーバーを選択し、選択したアプリケーションサーバーに要求を転送します。次の概念図は、一般的な負荷分散配置を示しています。もう 1 つのバリエーションとして、グローバル HTTP ポートを割り当てます。

図 1: 負荷分散アーキテクチャ



負荷分散仮想サーバーは、複数のアルゴリズム (またはメソッド) を使用して、管理する負荷分散サーバー間で負荷を分散する方法を決定できます。デフォルトの負荷分散方式は、最小接続方式です。この方式では、Citrix ADC アプライアンスは、現在アクティブなユーザー接続が最も少ない負荷分散アプリケーションサーバーに各着信クライアント接続を転送します。

一般的な Citrix ADC 負荷分散セットアップで構成するエンティティは次のとおりです。

- 仮想サーバの負荷分散。クライアントが特定の負荷分散された Web サイトまたはアプリケーションに対する接続要求を送信する IP アドレス、ポート、およびプロトコルの組み合わせ。アプリケーションがインターネットからアクセス可能な場合、仮想サーバーの IP (VIP) アドレスはパブリック IP アドレスです。アプリケーションが LAN または WAN からのみアクセス可能である場合、VIP は通常、プライベート（ICANN ルーティング不可）IP アドレスです。
- **Service.** 特定の負荷分散アプリケーションサーバーに要求をルーティングするために使用される IP アドレス、ポート、およびプロトコルの組み合わせ。サービスは、アプリケーションサーバー自体、または複数のアプリケーションをホストするサーバー上で実行されているアプリケーションを論理的に表現できます。サービスを作成したら、負荷分散仮想サーバーにバインドします。
- サーバーオブジェクト。IP アドレスでサーバーを識別する代わりに、物理サーバーに名前を割り当てることができる仮想エンティティ。サーバーオブジェクトを作成する場合、サービスの作成時にサーバーの IP アドレスの代わりにその名前を指定できます。それ以外の場合は、サービスの作成時にサーバーの IP アドレスを指定する必要があり、その IP アドレスがサーバーの名前になります。

- モニタ。Citrix ADC アプライアンス上のエンティティで、サービスを追跡し、サービスが正しく動作していることを確認します。モニタは、割り当てた各サービスを定期的にプローブ（またはヘルスチェックを実行）します。タイムアウトで指定された時間内にサービスが応答せず、指定した数のヘルスチェックが失敗した場合、そのサービスには DOWN とマークされます。Citrix ADC アプライアンスは、負荷分散を実行するときに、サービスが応答を終了する原因となった問題が修正されるまで、そのサービスをスキップします。

負荷分散セットアップの仮想サーバー、サービス、および負荷分散アプリケーションサーバーは、インターネットプロトコルバージョン 4 (IPv4) またはインターネットプロトコルバージョン 6 (IPv6) の IP アドレスのいずれかを使用できます。IPv4 アドレスと IPv6 アドレスを 1 つのロードバランシングセットアップで混在させることができます。

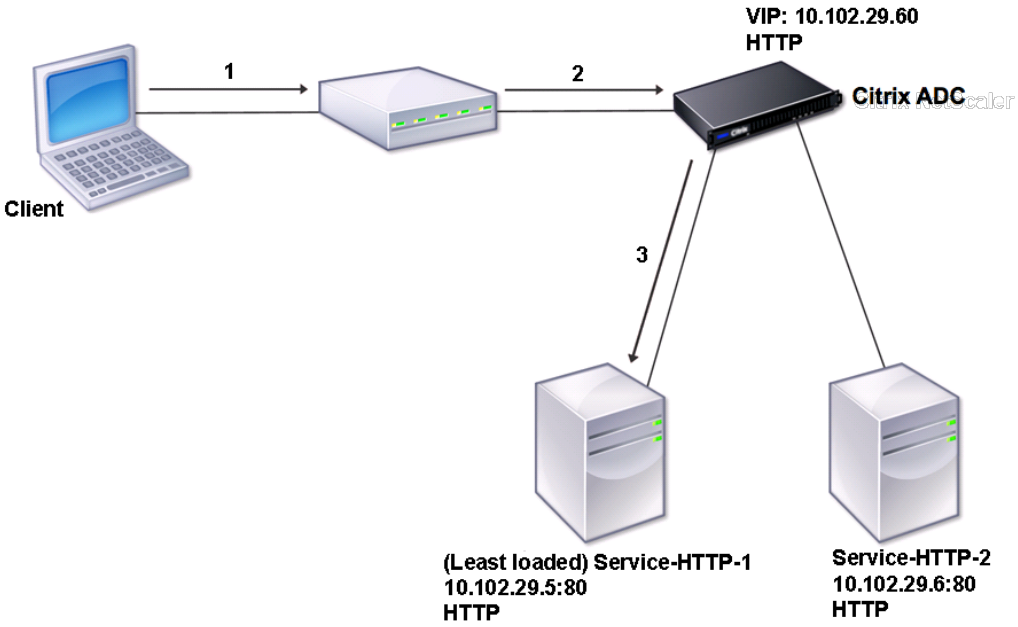
負荷分散設定のバリエーションについては、次のユースケースを参照してください。

- ダイレクトサーバリターンモードでのロードバランシングの設定
- DSR モードでの LINUX サーバーの構成
- TOS 使用時の DSR モードの設定
- IP Over IP を使用した DSR モードでのロードバランシングの設定
- ワンアームモードでのロードバランシングの設定
- インラインモードでのロードバランシングの設定
- 侵入検知システムサーバの負荷分散
- リモートデスクトッププロトコルサーバーの負荷分散

トポロジについて

負荷分散設定では、負荷分散サーバーは、クライアントとサーバーファームの間に論理的に配置され、サーバーファーム内のサーバーへのトラフィックフローを管理します。Citrix ADC アプライアンスでは、アプリケーションサーバーはサービスと呼ばれる仮想エンティティで表されます。次の図は、基本的な負荷分散構成のトポロジを示しています。

図 2：基本的な負荷分散トポロジ

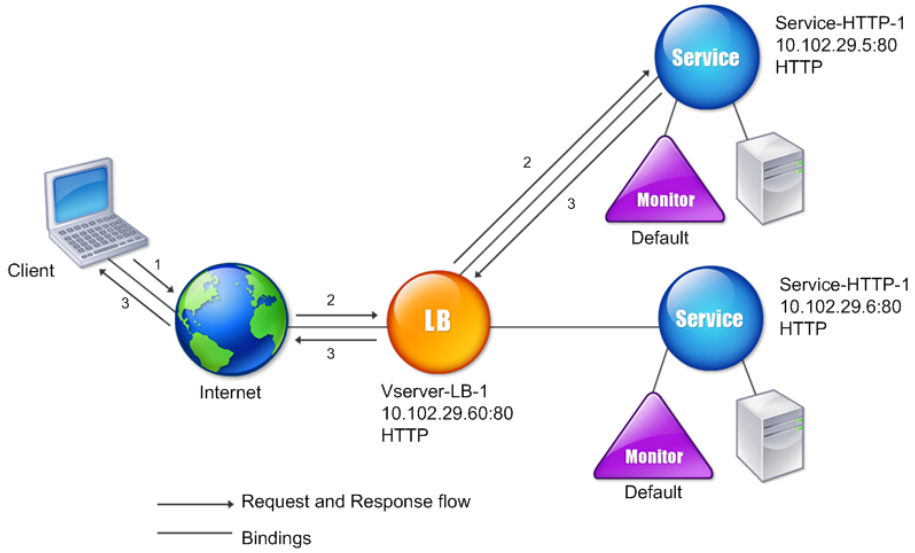


この図では、ロードバランシングを使用してサーバへのトラフィックフローを管理します。仮想サーバーは、クライアントからの要求に対してサービスを選択して割り当てます。サービス Service-HTTP-1 と Service-HTTP-2 が作成され、Vserver-LB-1 という名前の仮想サーバーにバインドされるシナリオを考えてみましょう。Vserver-LB-1 は、クライアント要求を Service-HTTP-1 または Service-HTTP-2 に転送します。Citrix ADC アプライアンスは、最小接続負荷分散方式を使用して、各要求のサービスを選択します。次の表に、アプライアンス上で設定する必要がある基本エンティティの名前と値を示します。

エンティティ	名前	IP アドレス	ポート	プロトコル
仮想サーバ	Vserver-LB-1	10.102.29.60	80	HTTP
サービス	Service-HTTP-1	10.102.29.5	80	HTTP
	Service-HTTP-2	10.102.29.6	80	HTTP
モニター	デフォルト	なし	なし	なし

次の図は、前述の表で説明したロードバランシングのサンプル値と必須パラメータを示しています。

図 3：負荷分散エンティティモデル



IP アドレスとポートの代わりにワイルドカードを使用する

場合によっては、仮想サーバーの IP アドレス、ポート、またはサービスのポートにワイルドカードを使用する必要がある場合があります。次の場合は、ワイルドカードの使用が必要になる場合があります。

- Citrix ADC アプライアンスが透過パススルーとして構成されている場合、Citrix ADC アプライアンスは、送信先の IP またはポートに関係なく、送信されるすべてのトラフィックを受け入れる必要があります。
- 1 つ以上のサービスがよく知られていないポートをリッスンする場合。
- 1 つ以上のサービスの場合は、時間の経過とともに、リッスンするポートを変更します。
- 単一の Citrix ADC アプライアンスで構成できる IP アドレスとポート数の上限に達した場合。
- 特定の仮想 LAN 上のすべてのトラフィックをリッスンする仮想サーバーを作成する場合。

ワイルドカードで構成された仮想サーバーまたはサービスがトラフィックを受信すると、Citrix ADC アプライアンスは実際の IP アドレスまたはポートを特定し、サービスおよび関連する負荷分散アプリケーションサーバーのレコードを作成します。動的に作成されたこれらのレコードは、動的に学習されたサーバおよびサービスレコードと呼ばれます。

たとえば、ファイアウォールの負荷分散構成では、IP アドレスとポートの両方にワイルドカードを使用できます。ワイルドカード TCP サービスをこの種類の負荷分散仮想サーバーにバインドすると、仮想サーバーは他のサービスまたは仮想サーバーと一致しないすべての TCP トラフィックを受信して処理します。

次の表に、ワイルドカード構成の種類とその使用時期について説明します。

IP	ポート	プロトコル	説明
*	*	TCP	Citrix ADC アプライアンス上の任意の IP アドレスおよびポートに送信されるトラフィックを受け入れる一般的なワイルドカード仮想サーバー。ワイルドカード仮想サーバを使用する場合、アプライアンスは各サービスの IP とポートを動的に学習し、トラフィックを処理するときに必要なレコードを作成します。
*	*	TCP	ファイアウォールの負荷分散仮想サーバー。ファイアウォールサービスをこの仮想サーバーにバインドできます。Citrix ADC アプライアンスは、トラフィックをファイアウォール経由で宛先に渡します。
IP アドレス	*	TCP、UDP、および任意	ポートに関係なく、指定された IP アドレスに送信されたすべてのトラフィックを受け入れる仮想サーバ。このタイプの仮想サーバには、トラフィックをリダイレクトするサービスを明示的にバインドする必要があります。ダイナミックに学習するわけではありません。

IP	ポート	プロトコル	説明
			注意: グローバル HTTP ポートのサービスまたは仮想サーバーは構成しません。この場合、特定のポートをグローバル HTTP ポートとして設定します（たとえば、ns param-httpPort 80 を設定します）。アプライアンスは、ポート番号と一致するすべてのトラフィックを受け入れ、HTTP トラフィックとして処理します。アプライアンスは、このトラフィックのサービスを動的に学習して作成します。

IP	ポート	プロトコル	説明
*	port	SSL, SSL_TCP	特定のポート上の任意の IP アドレスに送信されたすべてのトラフィックを受け入れる仮想サーバ。グローバルな透過的な SSL オフロードに使用されます。同じプロトコルタイプのサービスに対して通常実行されるすべての SSL、HTTP、および TCP 処理が、この特定のポートに送信されるトラフィックに適用されます。アプライアンスは、ポートを使用して、使用する必要のあるサービスの IP を動的に学習します。—cleartext が指定されていない場合、Citrix ADC アプライアンスはエンドツーエンド SSL を使用します。
*	port	該当なし	ポートへのトラフィックを受け入れることができる他のすべての仮想サーバ。これらの仮想サーバーにはサービスをバインドしません。Citrix ADC アプライアンスはそれらを動的に学習します。

注：Citrix ADC アプライアンスをグローバル（ワイルドカード）ポートを使用するトランスペアレントパススルーとして構成している場合は、エッジモードをオンにすることができます。
詳細については、「エッジモードの設定」を参照してください。

Citrix ADC アプライアンスは、最初に完全一致を試行して、仮想サーバーとサービスの検索を試みます。見つからない場合は、ワイルドカードに基づいて次の順序で一致を検索します。

1. 特定の IP アドレスと特定のポート番号
2. 特定の IP アドレスと * (ワイルドカード) ポート
3. • (ワイルドカード) IP アドレスと特定のポート
4. • (ワイルドカード) IP アドレスと * (ワイルドカード) ポート

アプライアンスが IP アドレスまたはポート番号で仮想サーバを選択できない場合、要求で使用されるプロトコルに基づいて、次の順序で仮想サーバを検索します。

1. HTTP
2. TCP
3. ANY

グローバル **HTTP** ポートの設定

グローバル HTTP ポートのサービスまたは仮想サーバは構成しません。代わりに、set ns param コマンドを使用して特定のポートを設定します。このポートを構成すると、Citrix ADC アプライアンスはポート番号に一致するすべてのトラフィックを受け入れ、HTTP トラフィックとして処理し、そのトラフィックのサービスを動的に学習および作成します。

グローバル HTTP ポートとして複数のポート番号を設定できます。単一の set ns param コマンドで複数のポート番号を指定する場合は、ポート番号を 1 つの空白で区切ります。1 つ以上のポートがグローバル HTTP ポートとしてすでに指定されており、現在設定されているポートを削除せずに 1 つ以上のポートを追加する場合は、コマンドで現在のポート番号と新しいポート番号を指定する必要があります。ポート番号を追加する前に、show ns param コマンドを使用して、現在設定されているポートを表示します。

コマンドラインインターフェイスを使用してグローバル **HTTP** ポートを構成するには

コマンドプロンプトで次のコマンドを入力して、グローバル HTTP ポートを構成し、構成を確認します。

```
set ns param  -httpPort <port>

show ns param
<!--NeedCopy-->
```

例 **1:** ポートをグローバル **HTTP** ポートとして設定する

この例では、ポート 80 がグローバル HTTP ポートとして設定されています。


```
set ns param -httpPort 80
Done
show ns param
         Global configuration settings:
                           HTTP port(s): 80
                        Max connections: 0
            Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
         Persistence Cookie Secure Flag: ENABLED
         ...
         ...
<!--NeedCopy-->
```

例 **2:1** つ以上のグローバル **HTTP** ポートがすでに設定されている場合のポートの追加 **

この例では、ポート 8888 がグローバル HTTP ポートリストに追加されます。ポート 80 は、すでにグローバル HTTP ポートとして設定されています。

```
> show ns param
        Global configuration settings:
                           HTTP port(s): 80
                        Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                           Min Path MTU: 576
        ...
        ...
 Done
> set ns param -httpPort 80 8888
 Done
>  show ns param

        Global configuration settings:
                           HTTP port(s): 80,8888
                        Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
```



```
                              Min Path MTU: 576

          ...
          ...
 Done
>
<!--NeedCopy-->
```

構成ユーティリティを使用してグローバル **HTTP** ポートを構成するには

1. [システム] **>** [設定] **>** [**HTTP** パラメータの変更] に移動し、HTTP ポート番号を追加します。

基本的な負荷分散の設定

October 7, 2021

初期負荷分散設定を設定する前に、負荷分散機能を有効にします。次に、負荷分散グループのサーバーごとに少なくとも 1 つのサービスを作成します。サービスを構成すると、負荷分散仮想サーバーを作成し、各サービスを仮想サーバーにバインドできます。これで初期セットアップは完了です。さらに構成を進める前に、構成を確認し、各要素が適切に設定され、期待どおりに動作していることを確認します。

負荷分散の有効化

負荷分散機能が無効になっている場合は、サービスや仮想サーバなどの負荷分散エンティティを設定できますが、機能を有効にするまで機能しません。

CLI を使用して負荷分散を有効にするには

コマンドプロンプトで次のコマンドを入力して、負荷分散を有効にし、構成を確認します。

- enable ns feature LB
- show ns feature

例

```
1      > enable ns feature LoadBalancing
2
3       Done
4
5      > show ns feature
```



```
6
7
8
9               Feature                         Acronym              Status
10
11              -------                         -------              ------
12
13      1)      Web Logging                     WL                   OFF
14
15      2)      Surge Protection                SP                   ON
16
17      3) Load Balancing LB ON
18
19      .
20
21      .
22
23      .
24
25      24)     NetScaler Push                  push                 OFF
26
27      Done
28  <!--NeedCopy-->
```

GUI を使用して負荷分散を有効にするには

System > **Settings** に移動して **Configure Basic Features** で **Load Balancing** を選択します。

サーバオブジェクトの設定

Citrix ADC アプライアンスでサーバーのエントリを作成します。Citrix ADC アプライアンスは、IP アドレスベースのサーバーとドメインベースのサーバーをサポートします。IP アドレスベースのサーバーを作成する場合、サービスの作成時に IP アドレスの代わりにサーバーの名前を指定できます。ドメインベースのサーバーの DNS の設定については、「ドメインネームシステム」を参照してください。

CLI を使用してサーバオブジェクトを作成するには

コマンドプロンプトで入力します。

```
1  add server `<name>`@ `<IPAddress>`@ | `<domain>`
2  <!--NeedCopy-->
```


IP アドレスベースのネームサーバーを追加する例:

```
add server web_serv 10.102.27.150
<!--NeedCopy-->
```

ドメインベースのサーバーを追加する例:

```
add server web_serv test.com
<!--NeedCopy-->
```

GUI を使用してサーバーオブジェクトを作成するには

「トラフィック管理」>「負荷分散」>「サーバー」に移動し、サーバーオブジェクトを追加します。

サービスの構成

負荷分散機能を有効にした後、負荷分散設定に含めるアプリケーションサーバーごとに少なくとも 1 つのサービスを作成する必要があります。構成するサービスは、Citrix ADC アプライアンスと負荷分散サーバー間の接続を提供します。各サービスには名前があり、IP アドレス、ポート、および提供されるデータの種類を指定します。

最初にサーバオブジェクトを作成せずにサービスを作成する場合、サービスの IP アドレスもサービスをホストするサーバの名前になります。サーバーを IP アドレスではなく名前で識別する場合は、サーバーオブジェクトを作成し、サービスの作成時に IP アドレスの代わりにサーバー名を指定できます。

トランスポート層プロトコルとして UDP を使用するサービスを作成すると、ping モニターが自動的にサービスにバインドされます。ping モニターは、組み込みモニターの中で最も基本的なものです。トランスポート層プロトコルとして TCP を使用するサービスを作成すると、TCP_default モニターが自動的にサービスにバインドされます。負荷分散設定を管理するための戦略を開発する場合、異なるタイプのモニター、または複数のモニターをサービスにバインドすることもできます。

サービスの作成

サービスを作成する前に、さまざまなサービスタイプとその使用方法を理解する必要があります。以下の一覧では、Citrix ADC アプライアンスでサポートされるサービスの種類について説明します。

HTTP

標準の Web サイトや Web アプリケーションなどの HTTP トラフィックを受け付ける負荷分散サーバーに使用されます。HTTP サービスタイプを使用すると、Citrix ADC アプライアンスは、レイヤー 7 Web サーバーに対して圧縮、

コンテンツフィルタリング、キャッシュ、クライアントキープアライブをサポートできます。このサービスタイプは、仮想サーバの IP ポートの挿入、リダイレクトポートの書き換え、Web 2.0 プッシュ、および URL リダイレクトのサポートもサポートします。

HTTP は TCP ベースのアプリケーションプロトコルであるため、Web サーバーに TCP サービスタイプを使用することもできます。ただし、この操作を行うと、Citrix ADC アプライアンスはレイヤー 4 の負荷分散のみを実行できます。前述のレイヤ 7 サポートは提供できません。

SSL

e コマースのウェブサイトやショッピングカートのアプリケーションなど、HTTPS トラフィックを受け入れるサーバーに使用されます。SSL サービスタイプを使用すると、Citrix ADC アプライアンスはセキュアな Web アプリケーションの SSL トラフィックの暗号化と復号化（SSL オフロードの実行）が可能になります。また、HTTP パーシステンス、コンテンツの切り替え、書き換え、仮想サーバーの IP ポートの挿入、Web 2.0 プッシュ、および URL リダイレクトもサポートしています。

SSL_BRIDGE、SSL_TCP、または TCP サービスタイプを使用することもできます。ただし、この操作を行うと、アプライアンスはレイヤ 4 負荷分散のみを実行します。SSL オフロードまたは説明されているレイヤー 7 サポートのいずれかを提供することはできません。

FTP

FTP トラフィックを受け入れるサーバーに使用されます。FTP サービスタイプを使用すると、Citrix ADC アプライアンスは FTP プロトコルの特定の詳細をサポートできます。

FTP サーバーには TCP または ANY サービスタイプを使用することもできます。

TCP

さまざまな種類の TCP トラフィックを受け入れるサーバーや、より具体的な種類のサービスを使用できない種類の TCP トラフィックを受け入れるサーバーに使用されます。

これらのサーバーには、ANY サービスタイプを使用することもできます。

SSL_TCP

SSL オフロードをサポートするために、非 HTTP ベースの SSL トラフィックを受け入れるサーバーに使用されます。

これらのサービスには、TCP サービスタイプを使用することもできます。その場合、Citrix ADC アプライアンスはレイヤー 4 の負荷分散と SSL オフロードの両方を実行します。

UDP

UDP トラフィックを受け入れるサーバーに使用されます。ANY サービスタイプを使用することもできます。

SSL_BRIDGE

Citrix ADC アプライアンスで SSL オフロードを実行しない場合に、SSL トラフィックを受け入れるサーバーに使用します。または、SSL_TCP サービスタイプを使用することもできます。

NNTP

ネットワークニュース転送プロトコル (NNTP) トラフィックを受け入れるサーバー (通常は Usenet サイト) に使用されます。

DNS

DNS トラフィックを受け入れるサーバー (通常はネームサーバー) に使用されます。DNS サービスタイプでは、Citrix ADC アプライアンスは各 DNS 要求と応答のパケットフォーマットを検証します。また、DNS 応答をキャッシュすることもできます。DNS ポリシーを DNS サービスに適用できます。

これらのサービスには、UDP サービスタイプを使用することもできます。ただし、Citrix ADC アプライアンスはレイヤー 4 の負荷分散のみを実行できます。DNS 固有の機能のサポートを提供することはできません。

ANY

任意のタイプの TCP、UDP、または ICMP トラフィックを受け入れるサーバーに使用されます。ANY パラメータは、主にファイアウォール負荷分散およびリンク負荷分散で使用されます。

SIP-UDP

UDP ベースのセッション開始プロトコル (SIP) トラフィックを受け入れるサーバーに使用されます。SIP は、マルチメディア通信セッションを開始、管理、および終了し、インターネットテレフォニー（VoIP）の標準として登場しました。

これらのサービスには、UDP サービスタイプを使用することもできます。ただし、Citrix ADC アプライアンスは、レイヤー 4 の負荷分散のみを実行します。SIP 固有の機能のサポートを提供できません。

DNS-TCP

DNS トラフィックを受け入れるサーバーに使用します。Citrix ADC アプライアンスは、DNS サーバーに送信される TCP トラフィックのプロキシとして機能します。DNS-TCP サービスタイプのサービスでは、Citrix ADC アプライアンスは各 DNS 要求と応答のパケット形式を検証し、DNS サービスタイプと同様に DNS 応答をキャッシュできます。

これらのサービスには、TCP サービスタイプを使用することもできます。ただし、Citrix ADC アプライアンスは、外部 DNS ネームサーバーのレイヤー 4 負荷分散のみを実行します。DNS 固有の機能のサポートを提供することはできません。

RTSP

リアルタイムストリーミングプロトコル (RTSP) トラフィックを受け入れるサーバーに使用されます。RTSP は、マルチメディアおよびその他のストリーミングデータの配信を提供します。オーディオ、ビデオ、その他の種類のストリーミングメディアをサポートするには、このタイプを選択します。

これらのサービスには、TCP サービスタイプを使用することもできます。ただし、Citrix ADC アプライアンスは、レイヤー 4 の負荷分散のみを実行します。RTSP ストリームを解析したり、RTSPID 永続性または RTSP NAT をサポートしたりすることはできません。

DHCPRA

DHCP トラフィックを受け入れるサーバーに使用されます。DHCPRA サービスタイプは、VLAN 間で DHCP 要求と応答をリレーするために使用できます。

DIAMETER

複数の Diameter サーバー間の Diameter トラフィックの負荷分散に使用されます。Diameter は、メッセージベースの負荷分散を使用します。

SSL_DIAMETER

SSL を介した Diameter トラフィックの負荷分散に使用されます。

Citrix ADC アプライアンスが関連する負荷分散サーバーに接続し、動作していることを確認するまで、サービスは DISABLED と指定されます。この時点で、サービスは ENABLED と指定されます。その後、Citrix ADC アプライアンスはサーバーのステータスを定期的に監視し、監視プローブ（ヘルスチェックと呼ばれます）に応答しないものは、応答するまで DISABLED 状態に戻します。

注：一連のサービスは、単一の CLI コマンドまたは同じダイアログボックスから作成できます。範囲内の名前は、接尾辞/接頭辞として使用される番号によって異なります。たとえば、サービス 1、サービス 2 などです。構成ユーティリティから、IP アドレスの最後のオクテットでのみ範囲を指定できます。これは IPv4 アドレスの場合は 4 番目、IPv6 アドレスの場合は 8 番目です。コマンドラインから、IP アドレスの任意のオクテットで範囲を指定できます。

QUIC

UDP ベースの QUIC ビデオトラフィックを受け入れる負荷分散サーバによって使用されます。このサービスにより、Citrix ADC アプライアンスは、UDP プロトコル経由で暗号化された ABR ビデオトラフィックを最適化できます。

CLI を使用してサービスを作成するには

コマンドプロンプトで入力します。


```
add service <name> <serverName> <serviceType> <port>

add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

GUI を使用してサービスを作成するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 詳細ペインで、[追加] をクリックします。
3. [サービスの作成] ダイアログボックスで、次のパラメータの値を指定します。
   - サービス名：名前
   - サーバ：サーバ名
   - プロトコル：serviceType
   - ポートポート
4. [**Create**] をクリックしてから、[**Close**] をクリックします。作成したサービスが [Services] ペインに表示されます。

仮想サーバーの作成

サービスを作成したら、負荷分散された Web サイト、アプリケーション、またはサーバーのトラフィックを受け入れる仮想サーバーを作成する必要があります。負荷分散が構成されると、ユーザーは仮想サーバーの IP アドレスまたは FQDN を使用して、負荷分散された Web サイト、アプリケーション、またはサーバーに接続します。

注:

- 「app_」というプレフィックスが付いた仮想サーバ名は、ns.conf ファイルに存在し、show コマンドを実行すると表示されます。ただし、GUI には「app」という接頭辞が付いた仮想サーバー名が表示されます。
- 仮想サーバーは、作成したサービスをバインドし、Citrix ADC アプライアンスがこれらのサービスに接続して動作していることを確認するまで、DOWN として指定されます。それ以外の場合、仮想サーバーが UP として指定されます。

CLI を使用して仮想サーバーを作成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
```



```
<!--NeedCopy-->
```

GUI を使用して仮想サーバーを作成するには

[トラフィック管理**] > [**負荷分散**] >[**仮想サーバー**]** に移動し、仮想サーバーを作成します。

仮想サーバーへのサービスのバインド

注: サービスは、最大 500 台の仮想サーバーにバインドできます。

サービスと仮想サーバーを作成したら、サービスを仮想サーバーにバインドする必要があります。通常、サービスは同じタイプの仮想サーバーにバインドされますが、次に示すように、特定のタイプのサービスを特定のタイプの仮想サーバーにバインドできます。

仮想サーバーの種類	サービスの種類	コメント
HTTP	SSL	通常、暗号化を行うには、SSL サービスを HTTP 仮想サーバーにバインドします。
SSL	HTTP	通常、SSL オフロードを行うには、HTTP サービスを SSL 仮想サーバーにバインドします。
SSL_TCP	TCP	通常、TCP サービスを SSL_TCP 仮想サーバーにバインドして、他の TCP の SSL オフロード (コンテンツ認識なしの SSL 復号化) を実行します。

仮想サーバにバインドされたサービスの状態によって、仮想サーバの状態が決まります。バインドされたすべてのサービスが DOWN の場合、仮想サーバには DOWN とマークされ、バインドされたサービスの 1 つが UP または OUT OF SERVICE の場合、仮想サーバの状態は UP になります。

CLI を使用してサービスを負荷分散仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <serviceName>

bind lb vserver Vserver-LB-1 Service-HTTP-1
```



```
<!--NeedCopy-->
```

GUI を使用してサービスを負荷分散仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. [サービス] セクションをクリックして、バインドするサービスを選択します。

注: サービスを複数の仮想サーバーにバインドできます。

設定の確認

基本構成が完了したら、負荷分散セットアップで各サービスと負荷分散仮想サーバーのプロパティを表示し、それぞれが正しく構成されていることを確認できます。構成が起動して実行されると、各サービスおよび負荷分散仮想サーバーの統計情報を表示して、考えられる問題をチェックできます。

サーバー・オブジェクトのプロパティの表示

Citrix ADC アプライアンス構成で、サーバーオブジェクトの名前、状態、IP アドレスなどのプロパティを表示できます。

コマンドラインインターフェイスを使用してサーバーオブジェクトのプロパティを表示するには

コマンドプロンプトで入力します。

```
show server <serverName>

show server server-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサーバーオブジェクトのプロパティを表示するには

Traffic Management > **Load Balancing** > **Servers** に移動します。使用可能なサーバーのパラメーター値が詳細ウィンドウに表示されます。

仮想サーバーのプロパティの表示

仮想サーバーの名前、状態、有効状態、IP アドレス、ポート、プロトコル、メソッド、バインドされたサービスの数などのプロパティを表示できます。基本的な負荷分散設定以上の設定を構成している場合は、仮想サーバーの永続性設定、それらにバインドされているポリシー、および仮想サーバーにバインドされているキャッシュリダイレクトとコンテンツスイッチング仮想サーバーを表示できます。

CLI を使用して負荷分散仮想サーバーのプロパティを表示するには

コマンドプロンプトで入力します。

```
show lb vserver <name>

show lb vserver Vserver-LB-1
<!--NeedCopy-->
```

GUI を使用して負荷分散仮想サーバーのプロパティを表示するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 詳細ウィンドウで、仮想サーバーをクリックして、詳細ウィンドウの下部にそのプロパティを表示します。
3. この仮想サーバーにバインドされているキャッシュリダイレクトおよびコンテンツスイッチング仮想サーバーを表示するには、[**CS/CR** バインディングの表示] をクリックします。

サービスのプロパティの表示

設定されたサービスの名前、状態、IP アドレス、ポート、プロトコル、最大クライアント接続数、接続あたりの最大要求、およびサーバタイプを表示し、この情報を使用してサービス構成の誤りをトラブルシューティングできます。

CLI を使用してサービスのプロパティを表示するには

コマンドプロンプトで入力します。

```
show service <name>

show service Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスのプロパティを表示するには

[トラフィック管理] > [負荷分散] > [サービス] に移動します。使用可能なサービスの詳細が [Services] ペインに表示されます。

サービスのバインディングの表示

サービスがバインドされている仮想サーバーのリストを表示できます。バインディング情報には、サービスがバインドされている仮想サーバの名前、IP アドレス、ポート、および状態も表示されます。バインド情報を使用して、サービスを仮想サーバーにバインドする場合の問題をトラブルシューティングできます。

CLI を使用してサービスのバインディングを表示するには

コマンドプロンプトで入力します。

```
show service bindings <name>

show service bindings Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスのバインディングを表示するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 詳細ウィンドウで、バインド情報を表示するサービスを選択します。
3. [操作] タブで、[バインドの表示] をクリックします。

仮想サーバの統計情報の表示

仮想サーバーのパフォーマンスを評価したり、問題のトラブルシューティングを行うために、Citrix ADC アプライアンスで構成された仮想サーバーの詳細を表示できます。すべての仮想サーバの統計情報のサマリーを表示することも、仮想サーバの名前を指定して、その仮想サーバの統計情報だけを表示することもできます。次の詳細を表示できます。

- 名前
- IP アドレス
- ポート
- プロトコル
- 仮想サーバの状態
- 受信したリクエストの割合
- ヒット率

CLI を使用して仮想サーバの統計情報を表示するには

アプライアンスで現在構成されているすべての仮想サーバ、または単一の仮想サーバの統計情報のサマリーを表示するには、コマンド・プロンプトで次のように入力します。


```
stat lb vserver [`<name>`]
<!--NeedCopy-->
```

例:

```
stat lb vserver server-1
<!--NeedCopy-->
```

次の図は、サンプル統計を示しています。


```
> stat lbvserver
[
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
vserver1        10.102.20.200    80          SSL         DOWN      0/s

lb1               203.1.113.5   443         DTLS         DOWN      0/s

vicap                       *     0          TCP         DOWN      0/s

lbicap                2.2.3.4  1344          TCP         DOWN      0/s

app_...stest          0.0.0.0     0         HTTP         DOWN      0/s

app_...ttest          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...test1          0.0.0.0     0         HTTP         DOWN      0/s

app_...1test          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...est12          0.0.0.0     0         HTTP         DOWN      0/s

app_...sting          0.0.0.0     0         HTTP         DOWN      0/s

test                  2.2.2.2    80         HTTP         DOWN      0/s

shar...lt-lb          0.0.0.0     0         HTTP         DOWN      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ts-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ns-lb          0.0.0.0     0         HTTP           UP      0/s

shar...as-lb          0.0.0.0     0         HTTP           UP      0/s

forward-vs            0.0.0.0     0          TCP         DOWN      0/s

tcpcs                 0.0.0.0     0          TCP         DOWN      0/s

test124               0.0.0.0     0          SSL         DOWN      0/s

testssl               0.0.0.0     0          SSL         DOWN      0/s
```


GUI を使用して仮想サーバの統計情報を表示するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 1 つの仮想サーバーの統計情報のみを表示する場合は、詳細ウィンドウで、統計情報を表示する仮想サーバーを選択します。
3. 詳細ペインで、[統計] をクリックします。

サービスの統計情報の表示

サービス統計を使用して、要求、応答、要求バイト、応答バイト、現在のクライアント接続、サージキュー内の要求、現在のサーバー接続などのレートを表示できます。

CLI を使用してサービスの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat service <name>
<!--NeedCopy-->
```

例**:**

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスの統計情報を表示するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. 詳細ペインで、統計情報を表示するサービス（たとえば、Service-HTTP-1）を選択します。
3. [統計] をクリックします。統計情報が新しいウィンドウに表示されます。

仮想サーバとサービスの状態の負荷分散

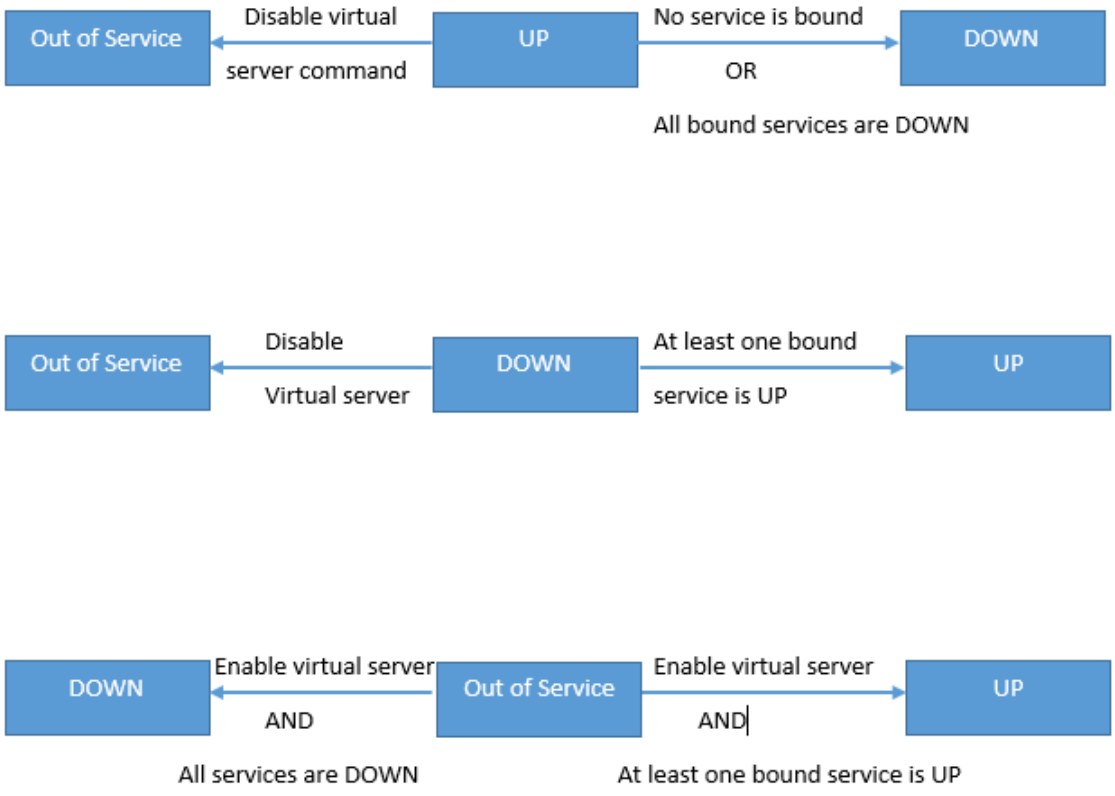
October 7, 2021

バックアップ仮想サーバを持たない負荷分散仮想サーバは、バインドされたサービスの状態および管理上の無効化かどうかに応じて、次の状態になります。

- **UP**: 仮想サーバーにバインドされているサービスの少なくとも 1 つが UP 状態です。
- **DOWN**: 仮想サーバーにバインドされているすべてのサービスがダウンしているか、負荷分散機能が有効になっていません。
- アウトオブサービス **(OFS)**: 仮想サーバーを管理上無効にすると、OFS 状態になりますが、有効状態は DOWN です。管理者は、OFS 状態への移行を DOWN または UP 状態から、または OFS 状態から DOWN または UP 状態への移行を制御できます。

バックアップ仮想サーバが構成されていない場合、仮想サーバの状態と有効状態は同じです。ただし、バックアップ仮想サーバまたはバックアップ仮想サーバのチェーンが設定されている場合、有効状態は、プライマリ仮想サーバとバックアップ仮想サーバにバインドされているサービスの状態から得られます。チェーン内のいずれかのバックアップ仮想サーバが UP の場合、プライマリ仮想サーバにバインドされているすべてのサービスが DOWN であっても、プライマリ仮想サーバの有効な状態は UP になります。

次の図は、仮想サーバーが 1 つの状態から別の状態に移行する条件を示しています。



サービスは、次の状態を取ることができます。

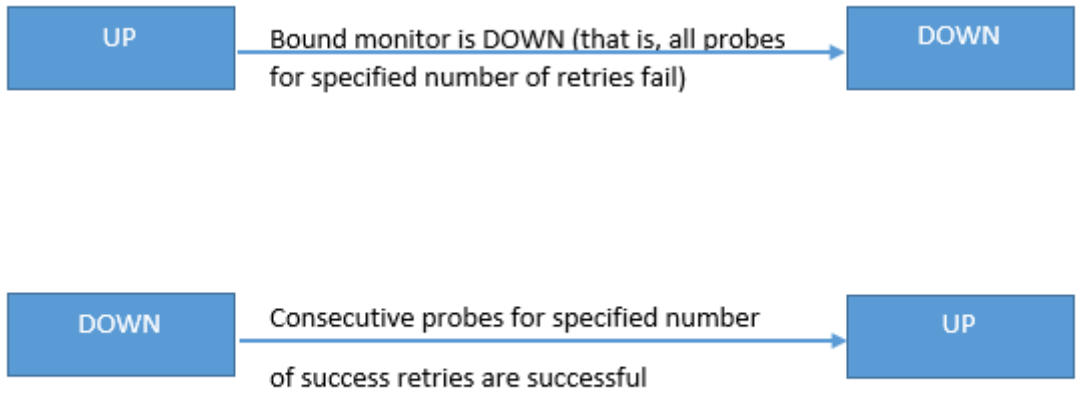
- **UP**: サービスにバインドされているすべてのモニタからのプローブが成功した場合。
- **DOWN**: サービスに対する監視プローブが、設定された制限時間内に応答されない場合。
- **OUT OF SERVICE**: サービスを管理上無効にした場合、またはサービスを正常にシャットダウンし、サービスに対するアクティブなトランザクションがない場合

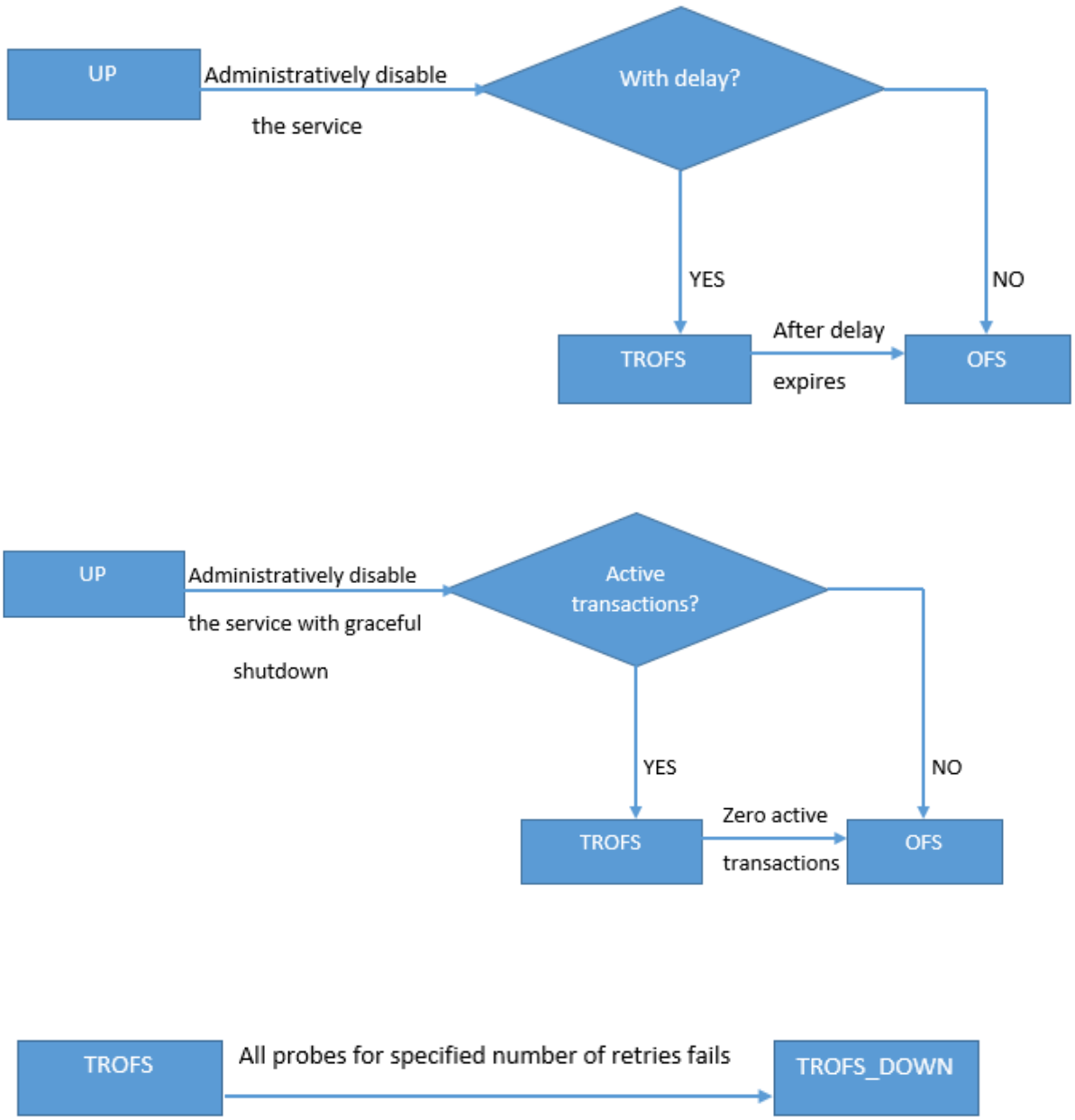
- **GOING OUT OF SERVICE**（**TROFS**）：遅れてサービスを管理上無効にするか、サービスを正常にシャットダウンし、サービスへのアクティブなトランザクションがある場合。詳細については、「サービスのグレースフルシャットダウン」を参照してください。
- サービス外に出るときに停止 **(TROFS_DOWN)[]** サービスがアウトオブサービス状態にある間、モニタリングプローブが失敗します。

UP から OFS への移行プロセス中のサービスは、サービス終了状態です。ダウンから OFS に移行するサービスは、DOWN WHEN GOING OUT OF SERVICE 状態になります。たとえば、サービスが DOWN で、遅延を伴って無効にした場合、サービスは DOWN WHEN GOING OUT OF SERVICE 状態に移行し、その後 OUT OF SERVICE 状態に移行します。サービスが UP で、遅延を伴って無効化すると、サービスは GOING OUT OF SERVICE に移行します。この間、サーバに対するモニタリングプローブが失敗すると、サービスは DOWN WHEN GOING OUT OF SERVICE に移行し、遅延時間が経過すると OFS 状態になります。

> 注
>
> バックアップ仮想サーバーへのスピルオーバーを構成するには、”healthThreshold” パラメーターを 0 以外の正の値に設定します。次に、プライマリ仮想サーバにバインドされている単一のサービスが DOWN WHEN GOING OUT OF SERVICE 状態に移行し、ヘルスのしきい値に達していない場合、プライマリ仮想サーバには DOWN とマークされ、新しい接続がバックアップ仮想サーバに送信されます。

次の図は、サービスが 1 つの状態から別の状態に移行する条件を示しています。





負荷分散プロファイルのサポート

October 7, 2021

負荷分散構成には多数のパラメータがあるため、複数の仮想サーバーで同じパラメータを設定すると面倒になることがあります。リリース 11.1 から、ロードバランシング (LB) プロファイルにより、この作業が簡単になりました。各仮

想サーバーでこれらのパラメータを設定する代わりに、プロファイルで負荷分散パラメータを設定し、このプロファイルを仮想サーバーに関連付けることができるようになりました。

現在、LB プロファイルでは、次のパラメータがサポートされています。

- HTTPonlyflag—永続クッキーに HttpOnly 属性を含めます。HttpOnly 属性は、クッキーのスコープを HTTP 要求に制限し、クロスサイトスクリプティング攻撃のリスクを軽減するのに役立ちます。
- usesecuredPersistenceCookie：SHA2 ハッシュアルゴリズムを使用してパーシステンスクッキー値を暗号化します。
- Cookiepassphrase：セキュアパーシステンス Cookie 値の生成に使用するパスフレーズを指定します。
- DBS_LB-MySQL および MSSQL サービスタイプのデータベース固有のロードバランシングを有効にします。
- Cl_process_local：クラスタ内の仮想サーバ宛てのパケットは制御されません。単一パケット要求応答モードの場合、またはアップストリームデバイスが接続ベースの配信に適切な RSS を実行している場合に、オプションを有効にします。
- lbHashAlgorithm：次のハッシュベースのロードバランシング方式に使用するハッシュアルゴリズムを指定します。
  - URL ハッシュ方式
  - ハッシュ関数
  - 宛先 IP ハッシュ方式
  - 送信元 IP ハッシュ方式
  - 送信元 IP 宛先 IP ハッシュ方式
  - 送信元 IP 送信元ポートのハッシュ方式
  - コール ID ハッシュ方式
  - トークン方式

  指定可能な値:DEFAULT、PRAC、JARH
  デフォルト値:DEFAULT
- lbHashFingers-ハッシュベースの LB メソッドの PRAC および JARH アルゴリズムで使用する指の数を指定します。フィンガーの数を増やすと、追加のメモリを犠牲にしてトラフィックの分散が向上します。

デフォルト値:256
最小値:1
最大値:1024

> 注
>
> DBS_LB パラメータと Cl_process_local パラメータは、仮想サーバーおよびプロファイルで設定できます。仮想サーバでこれらのパラメータを有効にし、この仮想サーバにプロファイルを設定すると、その仮想サーバの”show lb vserver”コマンドの出力にパラメータが無効として表示されます。プロファイルをチェック

> して、これらのパラメータの実際のステータスを確認します。さらに、プロファイルを仮想サーバーに設定してから設定解除すると、パラメータはその仮想サーバーのデフォルト値で設定されます。

CLI を使用して LB プロファイルを作成するには

コマンドプロンプトで入力します。

```
add lb profile <lbprofilename> -dbsLb ( ENABLED | DISABLED ) -
    processLocal ( ENABLED | DISABLED ) -httpOnlyCookieFlag ( ENABLED |
    DISABLED ) -cookiePassphrase -useSecuredPersistenceCookie ( ENABLED
    | DISABLED ) -lbHashAlgorithm <lbHashAlgorithm> -lbHashFingers <
    positive_integer>
<!--NeedCopy-->
```

例:

```
> sh lb profile p1
LB Profile name:  p1
DBS LB : DISABLED   Process Local: DISABLED
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
No of vservers bound: 0
Store MQTT clientid and username in transactional logs: NO
Hash LB algorithm used in LB decision: DEFAULT
Number of fingers for Hash LB algorithm: 256
Done

<!--NeedCopy-->
```

GUI を使用して LB プロファイルを作成するには

[システム] > [プロファイル] > [LB プロファイル] に移動し、プロファイルを追加します。

CLI を使用して LB プロファイルを LB 仮想サーバーに関連付けるには

コマンドプロンプトで入力します。

```
set lb vserver <name> -lbprofilename <string>
<!--NeedCopy-->
```


例

```
set lbvserver lbvip1 -lbprofile p1

Done

sh lb vserver lbvip1

lbvip1 (203.0.113.1:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Wed May 25 12:36:20 2016
Time since last state change: 0 days, 00:01:26.140
Effective State: UP  ARP:DISABLED
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: ENABLED
Port Rewrite : DISABLED
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION      BackupMethod: ROUNDROBIN
Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule: none
L2Conn: OFF
Skip Persistency: None
Listen Policy: NONE
IcmpResponse: PASSIVE
RHIstate: PASSIVE
New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
Mac mode Retain Vlan: DISABLED
DBS_LB: DISABLED
Process Local: DISABLED
Traffic Domain: 0
LB Profile: p1
Done
<!--NeedCopy-->
```

GUI を使用して **LB** プロファイルを **LB** 仮想サーバーに関連付けるには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。

2. 仮想サーバを選択し、[**Edit**] をクリックします。
3. [詳細設定**]** で、[プロファイル] をクリックします。
4. [**LB** プロファイル] リストで、この仮想サーバーに関連付けるプロファイルを選択します。

負荷分散アルゴリズム

October 7, 2021

負荷分散アルゴリズムは、Citrix ADC アプライアンスが各クライアント要求をリダイレクトするサービスを選択するために使用する基準を定義します。ロードバランシングアルゴリズムによって、異なる基準が使用されます。たとえば、最小接続アルゴリズムは、アクティブな接続が最も少ないサービスを選択し、ラウンドロビンアルゴリズムはアクティブなサービスの実行中のキューを維持し、各接続をキュー内の次のサービスに分散し、そのサービスをキューの最後に送信します。

一部の負荷分散アルゴリズムは、Web サイト上のトラフィックを処理し、DNS サーバーへのトラフィックを管理し、E コマースや会社の LAN や WAN で使用される複雑な Web アプリケーションの処理に最も適しています。次の表に、Citrix ADC アプライアンスがサポートする各負荷分散アルゴリズムと、それぞれの動作方法について簡単に説明します。

名前	サーバ選択基準
LEASTCONNECTION	現在クライアント接続数が最も少ないサービス。これがデフォルトのロードバランシングアルゴリズムです。
ROUNDROBIN	サービスのリストの一番上にあるサービス。そのサービスが接続用に選択されると、そのサービスはリストの一番下に移動します。
LEASTRESPONSETIME	現在最も速い応答時間を持つ負荷分散されたサーバー。
URLHASH	宛先 URL のハッシュ。
DOMAINHASH	宛先ドメインのハッシュ。
DESTINATIONIPHASH	宛先 IP アドレスのハッシュ。
SOURCEIPHASH	送信元 IP アドレスのハッシュ。
SRCIPDESTIPHASH	送信元と宛先 IP アドレスのハッシュ。
CALLIDHASH	SIP ヘッダー内のコール ID のハッシュ。
SRCIPSRCPORTHASH	クライアントの IP アドレスとポートのハッシュ。
LEASTBANDWIDTH	現在、帯域幅の制約が最も少ないサービス。

名前	サーバ選択基準
LEASTPACKETS	現在最も少ないパケットを受信しているサービス。
CUSTOMLOAD	負荷モニターからのデータ。
TOKEN	設定されたトークン。
LRTM	アクティブな接続数が最小で、平均応答時間が最短です。

負荷分散を行うサービスのプロトコルに応じて、Citrix ADC アプライアンスは、クライアントとサーバー間の各接続を、異なる時間間隔で持続するように設定します。これは、ロードバランシングの粒度と呼ばれ、要求ベース、接続ベース、時間ベースの粒度の 3 種類があります。次の表では、粒度の種類と、それぞれの粒度を使用するタイミングについて説明します。

細分性	負荷分散サービスのタイプ	Specifies
要求ベース	HTTP または HTTPS	TCP 接続とは関係なく、各 HTTP 要求に対して新しいサービスが選択されます。すべての HTTP 要求と同様に、Web サーバーが要求を満たすと、接続は閉じられます。
コネクションベース	HTTP 以外の TCP および TCP ベースのプロトコル	サービスは新しい TCP 接続ごとに選択されます。接続は、サービスまたはクライアントによって終了されるまで保持されます。
時間ベース	UDP およびその他の IP プロトコル	UDP パケットごとに新しいサービスが選択されます。サービスを選択すると、指定した期間、サービスとクライアントの間にセッションが作成されます。この時間が経過すると、セッションは削除され、追加のパケットが同じクライアントから送信された場合でも、新しいサービスが選択されます。

仮想サーバの起動時、または仮想サーバの状態が変化するたびに、仮想サーバは最初にラウンドロビン方式を使用してクライアント要求を物理サーバ間で分散できます。このタイプの配布は、スタートアップラウンドロビンと呼ばれ、

最初の要求が処理される際、単一サーバーでの不要な負荷を防ぐのに役立ちます。起動時にラウンドロビン方式を使用すると、仮想サーバは仮想サーバ上で指定されたロードバランシング方式に切り替わります。

スタートアップ RR 係数は、次のように動作します。

- スタートアップ RR 係数がゼロに設定されている場合、アプライアンスは要求レートに応じて指定されたロードバランシング方式に切り替わります。
- Startup RR Factor が 0 以外の数値である場合、アプライアンスは、指定された数の要求に対してラウンドロビン方式を使用し、指定されたロードバランシング方式に切り替えます。
- 既定では、[スタートアップ RR 係数] はゼロに設定されています。

注: 個々の仮想サーバーに対してスタートアップ RR 係数を設定することはできません。指定する値は、Citrix ADC アプライアンス上のすべての仮想サーバーに適用されます。

CLI を使用して起動ラウンドロビン係数を設定するには

コマンドプロンプトで入力します。

```
set lb parameter -startupRRFactor <positive_integer>
```

例

```
set lb parameter -startupRRFactor 25000
```

GUI を使用して起動ラウンドロビン係数を設定するには

1. [トラフィック管理**] > [**ロードバランシング**] > [**ロードバランシングパラメータの設定] に移動し、[スタートアップ RR 係数] を設定します。

最小接続方式

October 7, 2021

仮想サーバーが最小接続負荷分散アルゴリズム（または方法）を使用するように構成されている場合、アクティブな接続が最も少ないサービスが選択されます。これは、ほとんどの状況で最高のパフォーマンスを提供するため、デフォルトの方法です。

TCP、HTTP、HTTPS、SSL_TCP サービスの場合、Citrix ADC アプライアンスの既存の接続リストに次の接続タイプが含まれます。

- サービスへのアクティブな接続。クライアントが仮想サーバーに送信し、仮想サーバーがサービスに転送した要求を表す接続。HTTP および HTTPS サービスの場合、アクティブな接続は、まだ応答を受信していない HTTP または HTTPS 要求のみを表します。

- サージキュー内の接続を待機しています。サージキューで待機中で、まだサービスに転送されていない仮想サーバーへの接続。接続は、次のいずれかの理由により、いつでもサージキューに構築できます。
  - サービスには接続制限があり、負荷分散設定のすべてのサービスはその制限にあります。
  - サージ保護機能が構成され、仮想サーバへの要求のサージによってアクティブ化されています。
  - 負荷分散されたサーバーが内部制限に達したため、新しい接続を開きません。(たとえば、Apache サーバーの接続制限に達したとします)。

仮想サーバーが最小の接続方法を使用する場合、待機中の接続は特定のサービスに属するものと見なされます。したがって、これらのサービスへの新しい接続は開かれません。

UDP サービスの場合、最小接続アルゴリズムが考慮する接続には、クライアントとサービスの間のすべてのセッションが含まれます。これらのセッションは、論理的な時間ベースのエンティティです。セッション内の最初の UDP パケットが到着すると、Citrix ADC アプライアンスは送信元 IP アドレスとポートと宛先 IP アドレスとポートの間にセッションを作成します。

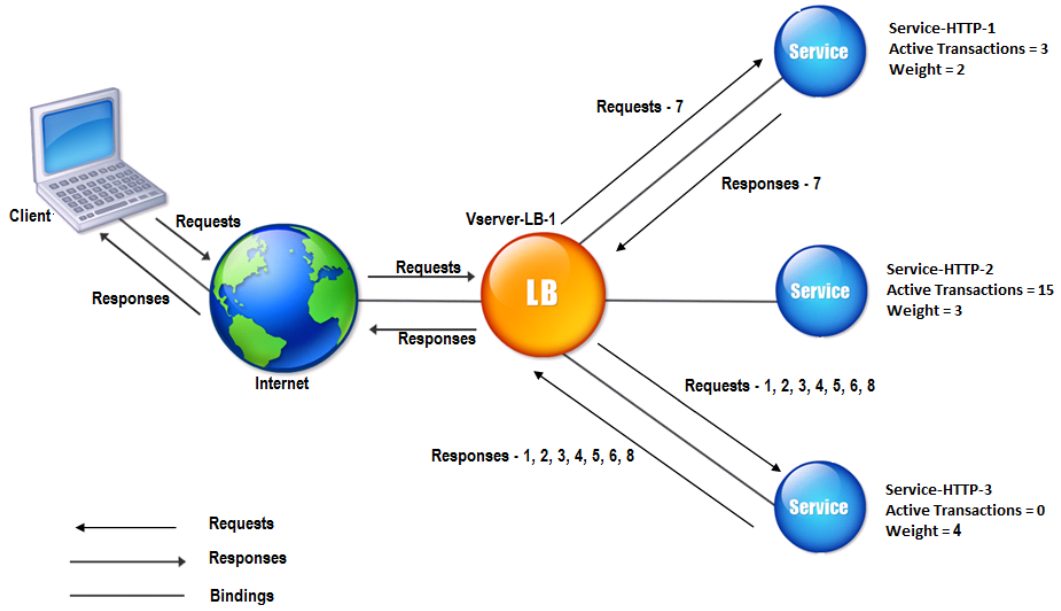
リアルタイムストリーミングプロトコル（RTSP）接続の場合、Citrix ADC アプライアンスはアクティブな制御接続の数を使用して、RTSP サービスへの接続の最小数を決定します。

次の例は、仮想サーバが、最小接続方式を使用してロードバランシングするサービスを選択する方法を示しています。次の 3 つのサービスを検討してください。

- Service-HTTP-1 は 3 つのアクティブなトランザクションを処理しています。
- Service-HTTP-2 は 15 個のアクティブなトランザクションを処理しています。
- Service-HTTP-3 はアクティブなトランザクションを処理していません。

次の図は、Citrix ADC アプライアンスが最小の接続方法を使用する場合に着信要求を転送する方法を示しています。

図 1: 最小接続負荷分散方式の仕組み



この図では、仮想サーバーは、アクティブなトランザクションが最も少ないサーバーを選択して、各着信接続のサービスを選択します。

接続は次のように転送されます。

- Service-HTTP-3 は、アクティブなトランザクションを処理していないため、最初の要求を受信します。

  注意: アクティブなトランザクションがないサービスが最初に選択されます。

- Service-HTTP-3 は 2 番目と 3 番目の要求を受信します。これは、サービスのアクティブなトランザクション数が次に最小であるためです。
- Service-HTTP-1 は 4 番目の要求を受信します。Service-HTTP-1 と Service-HTTP-3 のアクティブトランザクション数が同じであるため、仮想サーバはラウンドロビン方式を使用してそれらのいずれかを選択します。
- Service-HTTP-3 は 5 番目の要求を受信します。
- Service-HTTP-1 は、Service-HTTP-1 と Service-HTTP-3 の両方が Service-HTTP-2 と同じ数の要求を処理するまで、6 番目の要求を受信します。次に、Citrix ADC アプライアンスは、負荷が最も低いサービスであるか、またはラウンドロビンキューでターンが起動したときに、Service-HTTP-2 への要求の転送を開始します。

  メモ:Service-HTTP-2 への接続が終了すると，他の 2 つのサービスに 15 個のアクティブなトランザクションがある前に，新しい接続を取得することがあります。

次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

着信接続	選択されたサービス	現在のアクティブな接続数	注釈
Request-1	Service-HTTP-3;（N = 0）	1	Service-HTTP-3 のアクティブな接続数が最も少なくなります。
Request-2	Service-HTTP-3;（N = 1）	2	Service-HTTP-3 のアクティブな接続数が最も少なくなります。
Request-3	Service-HTTP-3; (N = 2)	3	-
Request-4	Service-HTTP-1; (N = 3)	4	Service-HTTP-1 と Service-HTTP-3 は、同じ数のアクティブな接続を持ちます。
Request-5	Service-HTTP-3; (N = 3)	4	Service-HTTP-1 と Service-HTTP-3 は、同じ数のアクティブな接続を持ちます。
Request-6	Service-HTTP-1;（N = 4）	5	-
Request-7	Service-HTTP-3; (N = 4)	5	-
Request-8	Service-HTTP-1; (N = 5)	6	-

Service-HTTP-2 は、アクティブなトランザクションを完了して現在の接続が閉じるとき、または他のサービス (Service-HTTP-1 および Service-HTTP-3）がそれぞれ 15 以上の接続を持つときに、負荷分散のために選択されます。

Citrix ADC アプライアンスは、ウェイトをサービスに割り当てるときに最小の接続方法を使用することもできます。次の式の値 (Nw) を使用してサービスを選択します。

Nw =（アクティブなトランザクションの数）*（10000/重量）

次の例は、Citrix ADC アプライアンスがサービスに割り当てられるときに、最小接続方式を使用して負荷分散するサービスを選択する方法を示しています。前の例では、Service-HTTP-1 にウェイト 2 が割り当てられ、Service-HTTP-2 にウェイト 3 が割り当てられ、Service-HTTP-3 にウェイト 4 が割り当てられているとします。接続は次のように転送されます。

- Service-HTTP-3 は、サービスがアクティブなトランザクションを処理していないため、最初のを受信します。

  注：サービスがアクティブなトランザクションを処理していない場合、Citrix ADC アプライアンスは、各サービスに割り当てられた重みに関係なく、ラウンドロビン方式を使用します。

- Service-HTTP-3 は、サービスの Nw 値が最小であるため、2 番目、3 番目、4 番目、5 番目、6 番めの要求を受信します。
- Service-HTTP-1 は 8 番目の要求を受信します。Service-HTTP-1 と Service-HTTP-3 は同じ Nw 値を持つため、アプライアンスはラウンドロビン方式でロードバランシングを実行します。したがって、Service-HTTP-3 は 9 番目の要求を受信します。

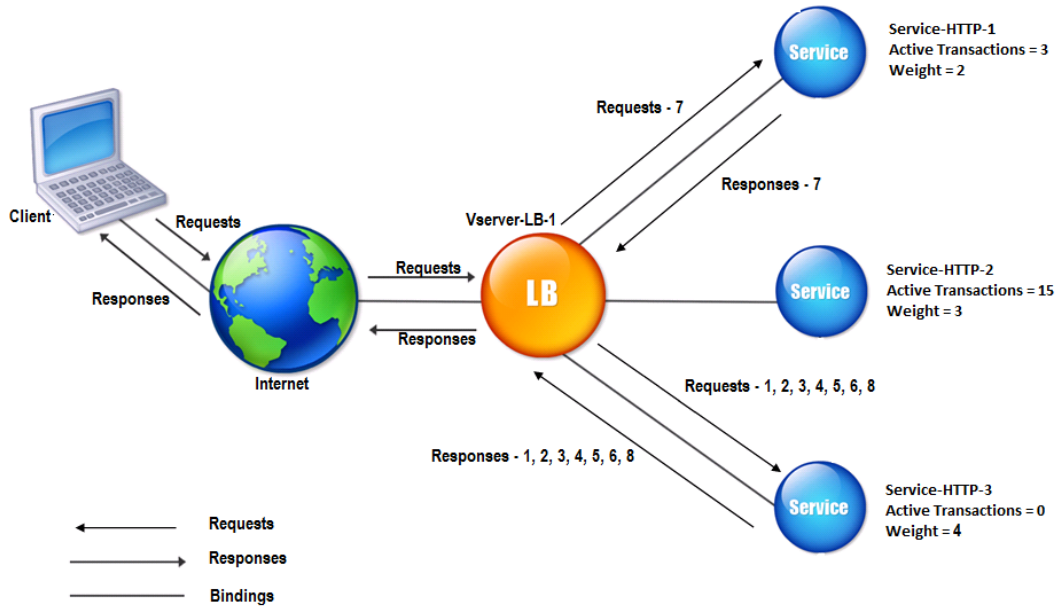
次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

リクエストを受け取りました	選択されたサービス	現在の Nw（アクティブなトランザクション数）*（10000/重量）値	注釈
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 2500	Service-HTTP-3 の Nw 値は最小です。
Request-2	Service-HTTP-3; (Nw = 2500)	Nw = 5000	
Request-3	Service-HTTP-3; (Nw = 5000)	Nw = 7500	
Request-4	Service-HTTP-3; (Nw = 7500)	Nw = 10000	
Request-5	Service-HTTP-3; (Nw = 10000)	Nw = 12500	
Request-6	Service-HTTP-3; (Nw = 12500)	Nw = 15000	
Request-7	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 と Service-HTTP-3 は Nw の値が同じです。
Request-8	Service-HTTP-3; (Nw = 15000)	Nw = 17500	

Service-HTTP-2 は、アクティブなトランザクションを完了するか、他のサービス（Service-HTTP-1 および Service-HTTP-3）の Nw 値が 50000 に等しい場合に、ロードバランシングの対象として選択されます。

次の図は、Citrix ADC アプライアンスがサービスに重みを割り当てるときに最小接続方法を使用する方法を示しています。

図 2：重み付け時における最小接続負荷分散手法の仕組み



最小接続方法を構成するには、ポリシーを含まない負荷分散方式の構成を参照してください。

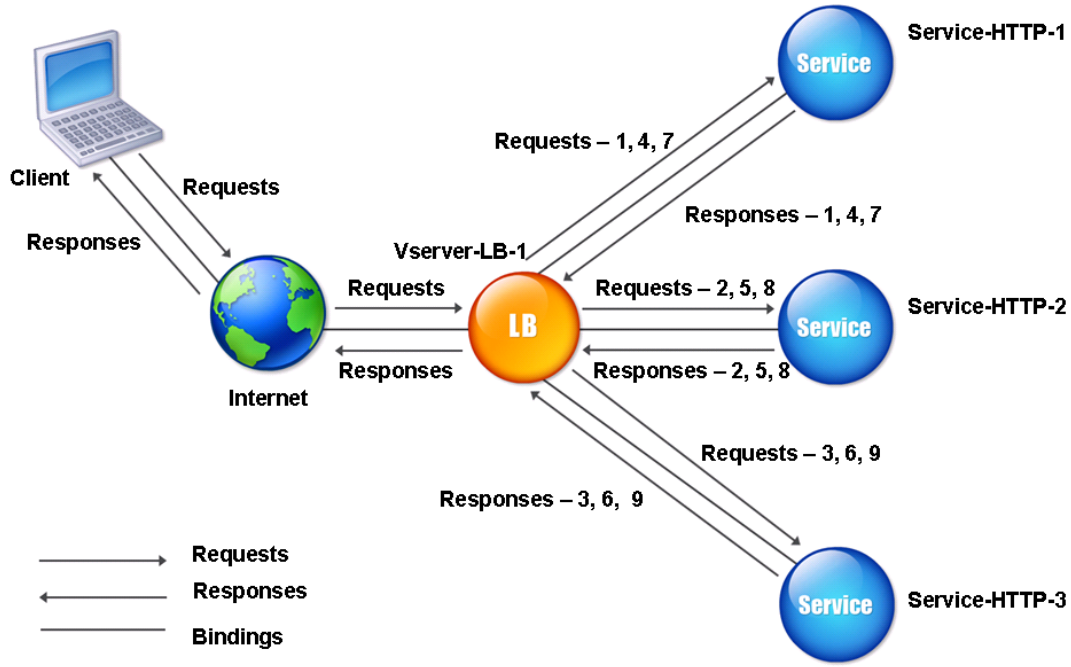
ラウンドロビン方式

October 7, 2021

ロードバランシング仮想サーバーがラウンドロビン方式を使用するように構成されている場合、バインドされているサービスの一覧を継続的にローテーションします。仮想サーバは、要求を受信すると、リスト内の最初のサービスに接続を割り当てて、そのサービスをリストの一番下に移動します。

次の図は、Citrix ADC アプライアンスが、負荷分散された 3 つのサーバーとそれに関連するサービスを含む負荷分散セットアップでラウンドロビン方式を使用する方法を示しています。

図 1：ラウンドロビンロードバランシング方式の動作



各サービスに異なる重みを割り当てると、Citrix ADC アプライアンスは着信接続の加重ラウンドロビン分散を実行します。これは、より低い重み付けされたサービスを適切な間隔でスキップすることによって行われます。

たとえば、3 つのサービスを使用した負荷分散設定があるとします。Service-HTTP-1 をウェイト 2 に、Service-HTTP-2 をウェイト 3 に、Service-HTTP-3 をウェイト 4 に設定します。サービスは Vserver-LB-1 にバインドされます。Vserver-LB-1 は、ラウンドロビン方式を使用するように設定されています。この設定では、受信要求は次のように配信されます。

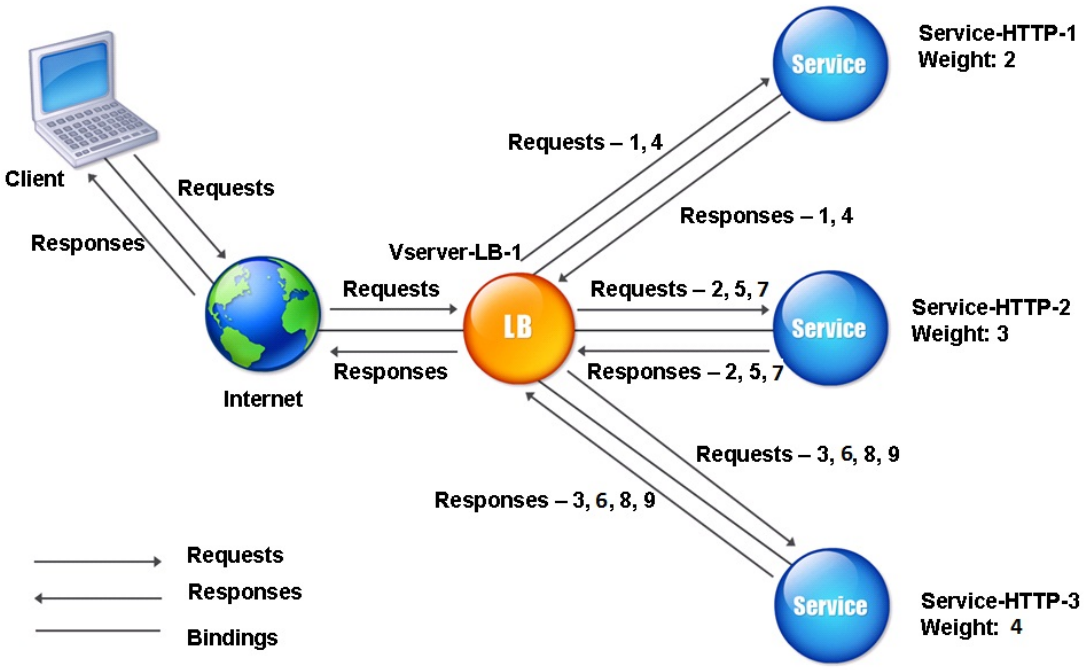
- Service-HTTP-1 は、最初の要求を受信します。
- Service-HTTP-2 は 2 番目の要求を受信します。
- Service-HTTP-3 は 3 番目の要求を受信します。
- Service-HTTP-1 は 4 番目の要求を受信します。
- Service-HTTP-2 は 5 番目の要求を受信します。
- Service-HTTP-3 は 6 番目の要求を受信します。
- Service-HTTP-2 は 7 番目の要求を受信します。
- Service-HTTP-3 は、8 番目と 9 番目の要求の両方を受信します。

注：複数のサービスが同じサーバーを使用したり、サーバーが過負荷にならないように、サービスの重みを設定することもできます。

その後、同じパターンを使用して新しいサイクルが開始されます。

次の図は、加重ラウンドロビン方式を示しています。

図 2：ラウンドロビンロードバランシング方式による加重サービスのサポート方法



ラウンドロビン方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

最短応答時間法

October 7, 2021

負荷分散仮想サーバーが最小の応答時間方式を使用するように構成されている場合、アクティブな接続が最も少なく、平均応答時間が最も小さいサービスを選択します。この方法は、HTTP および SSL (セキュアソケットレイヤー) 負荷分散仮想サーバーに対してのみ構成できます。応答時間（TTFB とも呼ばれます）は、要求パケットをサービスに送信してからサービスから最初の応答パケットを受信するまでの時間間隔です。Citrix ADC アプライアンスは、応答コード 200 を使用して TTFB を計算します。

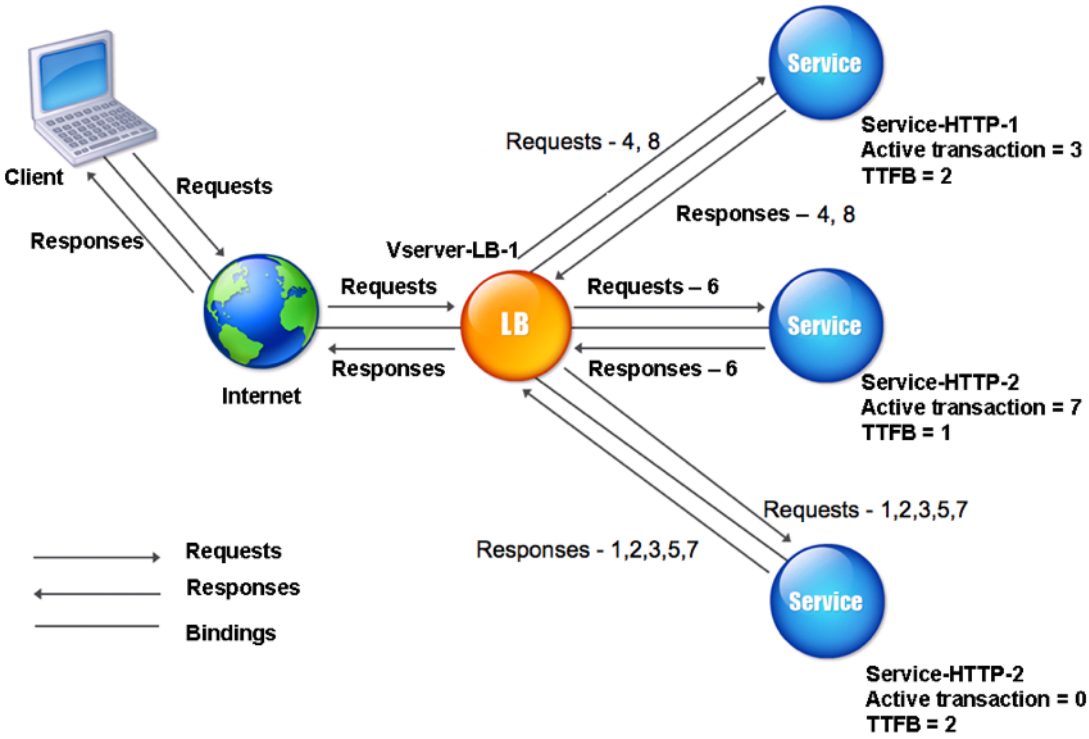
次の例は、仮想サーバが、最小応答時間方式を使用してロードバランシングするサービスを選択する方法を示しています。次の 3 つのサービスを検討してください。

- Service-HTTP-1 は 3 つのアクティブなトランザクションを処理しており、TTFB は 2 秒です。
- Service-HTTP-2 は 7 つのアクティブなトランザクションを処理し、TTFB は 1 秒です。

- Service-HTTP-3 はアクティブなトランザクションを処理せず、TTFB は 2 秒です。

次の図は、Citrix ADC アプライアンスが最小応答時間方式を使用して接続を転送する方法を示しています。

図 1: 最小応答時間のロードバランシング方式の動作



仮想サーバは、各サービスのアクティブなトランザクション数に TTFB を掛けてから、最も低い結果を持つサービスを選択することによって、サービスを選択します。上記の例では、仮想サーバは次のように要求を転送します。

- Service-HTTP-3 は、サービスがアクティブなトランザクションを処理していないため、最初の要求を受信します。
- Service-HTTP-3 は、2 番目と 3 番目の要求も受信します。これは、3 つのサービスの中で最も低い結果になるためです。
- Service-HTTP-1 は 4 番目の要求を受信します。Service-HTTP-1 と Service-HTTP-3 は同じ結果になるため、Citrix ADC アプライアンスはラウンドロビン方式を適用して選択します。
- Service-HTTP-3 は 5 番目の要求を受信します。
- Service-HTTP-2 は 6 番目の要求を受信します。これは、この時点で結果が最も低いためです。
- この時点では、Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ結果になるため、アプライアンスはラウンドロビン方式に切り替わり、その方式を使用して接続を配信し続けます。

次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

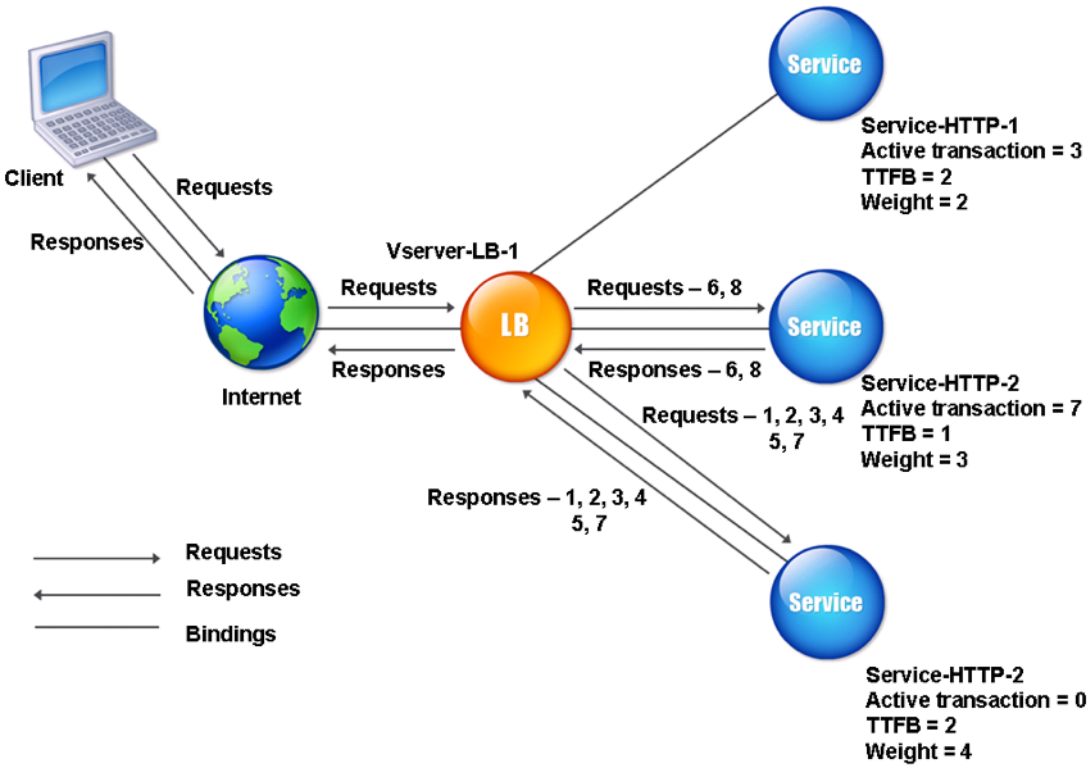
リクエストを受け取りました	選択されたサービス	現在の N 値（アクティブ・トランザクション数 * TTFB）	注釈
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 の最小値は N です。
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 の最小値は N です。
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 の最小値は N です。
Request-4	Service-HTTP-1;（N = 6）	N = 8	Service-HTTP-1 と Service-HTTP-3 は、同じ N 値を持ちます。アプライアンスは、ラウンドロビン方式を使用して要求を配信します。
Request-5	Service-HTTP-3;（N = 6）	N = 8	Service-HTTP-1 と Service-HTTP-3 は、同じ N 値を持ちます。
Request-6	Service-HTTP-2;（N = 7）	N = 8	Service-HTTP-2 は最小 N 値を持ちます。
Request-7	Service-HTTP-3;（N = 8）	N = 10	Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、同じ N 値を持ちます。Citrix ADC アプライアンスは、ラウンドロビン方式を使用してリクエストを配信します。
Request-8	Service-HTTP-1;（N = 8）	N = 10	Service-HTTP-1 と Service-HTTP-2 は同じ値を持ちます。アプライアンスは、ラウンドロビン方式を使用して要求を分散します。

Service-HTTP-1 は、アクティブなトランザクションが完了したとき、またはその N 値が他のサービス（Service-HTTP-2 および Service-HTTP-3）よりも小さい場合に、ロードバランシングのために再び選択されます。

ウェイト割り当て時のサービスの選択

次の図は、重みの割り当て時に Citrix ADC アプライアンスが最小応答時間方式を使用する方法を示しています。

図 2：重みが割り当てられている場合の最小応答時間負荷分散方法の仕組み



仮想サーバーは、次の式の値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000/weight)。ここで N = (アクティブなトランザクション数 * TTFB)

Service-HTTP-1 に重みが 2、Service-HTTP-2 に重みが 3、Service-HTTP-3 に重みが 4 が割り当てられているとします。

Citrix ADC アプライアンスは、次のように要求を配信します。

- Service-HTTP-3 は、アクティブなトランザクションを処理していないため、最初の要求を受信します。

  サービスがアクティブなトランザクションを処理していない場合、アプライアンスは割り当てられている重みに関係なくトランザクションを選択します。
- Service-HTTP-3 は、Nw の値が最小であるため、2 番目、3 番目、4 番目、5 番めの要求を受信します。
- Service-HTTP-2 は 6 番目の要求を受信します。これは、このサービスが Nw の値が最も小さいためです。
- Service-HTTP-3 は、このサービスの Nw 値が最小であるため、7 番目の要求を受信します。
- Service-HTTP-2 は 8 番目の要求を受信します。これは、このサービスが Nw の値が最も小さいためです。

Service-HTTP-1 は重みが最も低く、したがって Nw 値が最大であるため、仮想サーバはロードバランシング用に選択しません。

次の表では、前述の 3 サービス負荷分散設定での接続の分散方法について説明します。

リクエストを受け取りました	選択されたサービス	現在のニュート値 = (N) * (10000/重量)	注釈
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 5000	Service-HTTP-3 の Nw 値は最小です。
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 の Nw 値は最小です。
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 の Nw 値は最小です。
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 の Nw 値は最小です。
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 の Nw 値は最小です。
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 は、最小 Nw 値を持っています。
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 の Nw 値は最小です。
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 は、最小 Nw 値を持っています。

Service-HTTP-1 は、アクティブなトランザクションが完了したとき、またはその Nw 値が他のサービス (Service-HTTP-2 および Service-HTTP-3) よりも小さい場合に、ロードバランシングの対象として選択されます。

CLI を使用して最小応答時間のロードバランシング方式を設定するには

コマンドプロンプトで、次のように入力します。

```
set lb vserver <name> -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

例**:**


```
set lb vserver Vserver-LB-1 -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

GUI を使用して最小応答時間のロードバランシング方式を構成するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[**LEASTRESPONSETIME**] を選択します。

モニタの構成の詳細については、「負荷分散セットアップでのモニタの設定」を参照してください。

LRTM (計算機)

October 7, 2021

注：LRTM は、モニタ（LRTM）を使用した最小応答時間メソッドの略です。

負荷分散仮想サーバーが LRTM 方式を使用するように構成されている場合、既存の監視インフラストラクチャを使用して、応答時間を最速にします。次に、負荷分散仮想サーバは、アクティブなトランザクションの数が最小で、応答時間が最小であるサービスを選択します。LRTM 方式を使用する前に、アプリケーション固有のモニタを各サービスにバインドし、これらのモニタで LRTM モードを有効にする必要があります。Citrix ADC アプライアンスは、監視プローブから計算した応答時間に基づいて、負荷分散を決定します。

LRTM 方式を使用して、非 HTTP サービスと HTTPS 以外のサービスをロードバランシングすることもできます。この方法は、複数のモニタがサービスにバインドされている場合にも使用できます。各モニターは、バインドされているサービスに対して測定するプロトコルを使用して、応答時間を決定します。次に、仮想サーバは結果を平均化して、そのサービスの平均応答時間を計算します。

次の表に、さまざまなモニタの応答時間の計算方法をまとめます。

モニター	応答時間の計算
PING	ICMP ECHO 要求と ICMP ECHO 応答の時間差。
TCP	SYN 要求と SYN+ACK 応答の時間差。
HTTP	HTTP 要求（TCP 接続が確立された後）と HTTP 応答の時間差。
TCP-ECV	データ送信文字列が送信されてから、データ受信文字列が返される時間の差。送受信文字列を使用しない TCP-ECV モニタの設定が正しくないと見なされます。
HTTP-ECV	HTTP 要求と HTTP 応答の時間差。

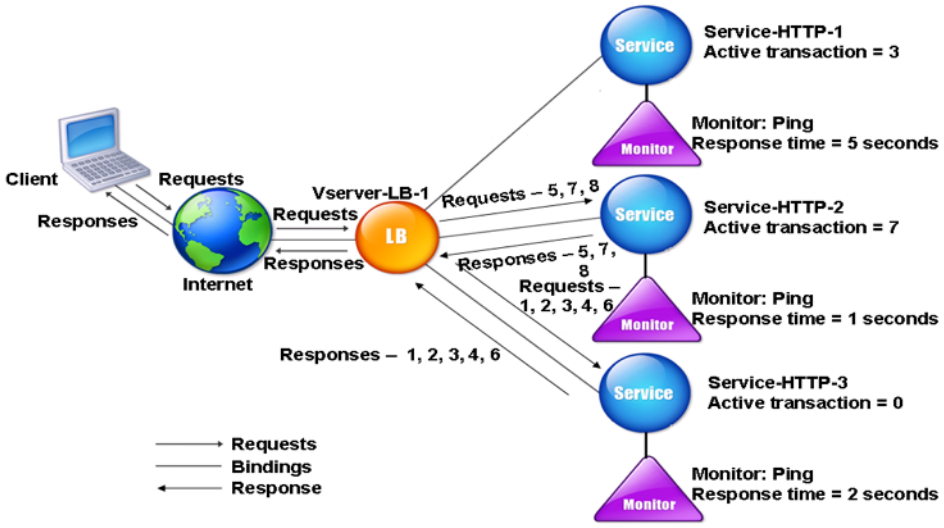
モニター	応答時間の計算
UDP-ECV	UDP の送信文字列と受信文字列の間の時間差。受信文字列がない UDP-ECV モニタは、正しくない構成と見なされます。
DNS	DNS クエリと DNS 応答の時間差。
TCPS	SYN 要求と SSL ハンドシェイク完了の時間差。
FTP	ユーザー名の送信とユーザー認証の完了の時間差。
HTTPS (HTTPS 要求を監視する)	時差は HTTP モニタの場合と同じです。
HTTPS-ECV（HTTPS 要求の監視）	時差は HTTP-ECV モニタの場合と同じです
USER	リクエストがディスパッチャに送信された時刻と、ディスパッチャレスポンスが受信された時刻の差。

次の例では、Citrix ADC アプライアンスが LRTM 方式を使用して負荷分散するサービスを選択する方法を示します。次の 3 つのサービスを検討してください。

- Service-HTTP-1 は 3 つのアクティブなトランザクションを処理しており、応答時間は 5 秒です。
- Service-HTTP-2 は 7 つのアクティブなトランザクションを処理しており、応答時間は 1 秒です。
- Service-HTTP-3 はアクティブなトランザクションを処理せず、応答時間は 2 秒です。

次の図は、Citrix ADC アプライアンスが要求を転送するときに従うプロセスを示しています。

図 1: LRTM メソッドの仕組み



仮想サーバーは、次の式で値 (N) を使用してサービスを選択します。

N =（アクティブなトランザクション数 * モニタによって決定される応答時間）

仮想サーバは、次のように要求を配信します。

- Service-HTTP-3 は、このサービスはアクティブなトランザクションを処理していないため、最初の要求を受信します。
- Service-HTTP-3 は、2 番目、3 番目、4 番目の要求を受信します。これは、このサービスの N 値が最も小さいためです。
- Service-HTTP-2 は、5 番目の要求を受信します。これは、このサービスの N 値が最小であるためです。
- 現在、Service-HTTP-2 と Service-HTTP-3 の両方が同じ N 値を持つため、Citrix ADC アプライアンスはラウンドロビン方式に切り替わります。したがって、Service-HTTP-3 は 6 番目の要求を受信します。
- Service-HTTP-2 は、7 番目と 8 番目の要求を受信します。これは、このサービスの N 値が最も小さいためです。

Service-HTTP-1 は、他の 2 つのサービスと比べて負荷が高い（N 値が最も高い）ため、ロードバランシングには考慮されません。ただし、Service-HTTP-1 がアクティブなトランザクションを完了すると、Citrix ADC アプライアンスは再びそのサービスを負荷分散と見なします。

次の表は、サービスの N の計算方法をまとめたものです。

リクエストを受け取りました	選択されたサービス	現在の N 値（アクティブ・トランザクション数 * TTFB）	注釈
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 の最小値は N です。
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 の最小値は N です。
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 の最小値は N です。
Request-4	Service-HTTP-3;（N = 6）	N = 8	Service-HTTP-3 の最小値は N です。
Request-5	Service-HTTP-2;（N = 7）	N = 8	Service-HTTP-2 は最小 N 値を持ちます。
Request-6	Service-HTTP-3;（N = 8）	N = 10	Service-HTTP-2 と Service-HTTP-3 は、同じ N 値を持ちます。Citrix ADC アプライアンスがラウンドロビン方式に切り替わり、Service-HTTP-3 を選択

リクエストを受け取りました	選択されたサービス	現在の N 値（アクティブ・トランザクション数 * TTFB）	注釈
Request-7	Service-HTTP-2;（N = 8）	N = 9	Service-HTTP-2 は最小 N 値を持ちます。
Request-8	Service-HTTP-2;（N = 9）	N = 10	Service-HTTP-2 は最小 N 値を持ちます。

Service-HTTP-1 は、アクティブなトランザクションが完了したとき、またはその N 値が他のサービス（Service-HTTP-2 および Service-HTTP-3）よりも小さい場合に、ロードバランシングのために再び選択されます。

ウェイト割り当て時のサービスの選択

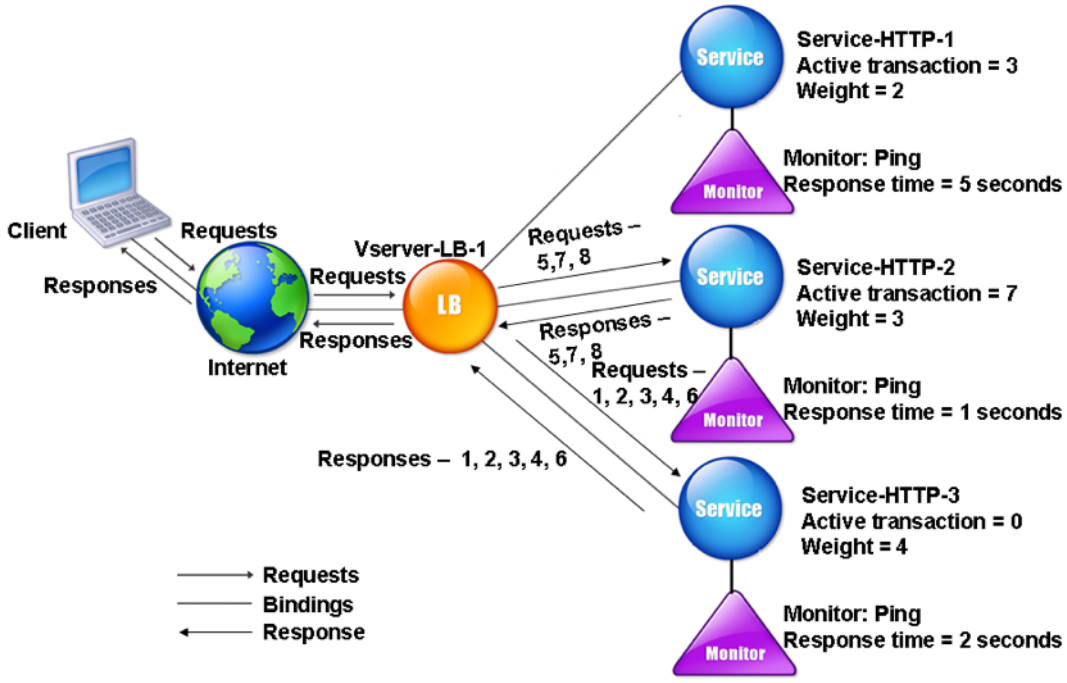
また、Citrix ADC アプライアンスは、アクティブなトランザクションの数、応答時間、および異なる重みが割り当てられている場合の重みを使用して負荷分散を実行します。Citrix ADC アプライアンスは、次の式の値（Nw）を使用してサービスを選択します。

Nw = (N) * (10000/重量)

N =（アクティブなトランザクションの数 × モニタによって決定される応答時間）

次の図は、重みが割り当てられたときに、仮想サーバが LRTM 方式を使用する方法を示しています。

図 2：重みが割り当てられている場合の最小応答時間負荷分散方法の仕組み



この例では、Service-HTTP-1 に重みが 2、Service-HTTP-2 に重みが 3、Service-HTTP-3 に重みが 4 が割り当てられているとします。

Citrix ADC アプライアンスは、次のように要求を配信します。

- Service-HTTP-3 は、アクティブなトランザクションを処理していないため、最初の要求を受信します。
- Service-HTTP-3 は、Nw の値が最小であるため、2 番目、3 番目、4 番目、5 番めの要求を受信します。
- Service-HTTP-2 は 6 番目の要求を受信します。これは、このサービスが Nw の値が最も小さいためです。
- Service-HTTP-3 は、このサービスの Nw 値が最小であるため、7 番目の要求を受信します。
- Service-HTTP-2 は 8 番目の要求を受信します。これは、このサービスが Nw の値が最も小さいためです。

Service-HTTP-1 は最小重みと最大 Nw 値であるため、Citrix ADC アプライアンスは負荷分散のために選択しません。

次の表は、さまざまなモニタに対する Nw の計算方法をまとめたものです。

リクエストを受け取りました	選択されたサービス	現在の Nw 値 (N) * (10000/重量)	注釈
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 5000	Service-HTTP-3 の Nw 値は最小です。

リクエストを受け取りました	選択されたサービス	現在の Nw 値 (N) * (10000/重量)	注釈
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 の Nw 値は最小です。
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 の Nw 値は最小です。
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 の Nw 値は最小です。
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 の Nw 値は最小です。
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 は、最小 Nw 値を持っています。
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 の Nw 値は最小です。
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 は、最小 Nw 値を持っています。

Service-HTTP-1 は、アクティブなトランザクションが完了したとき、またはその Nw 値が他のサービス (Service-HTTP-2 および Service-HTTP-3) よりも小さい場合に、ロードバランシングの対象として選択されます。

CLI を使用して **LRTM** ロードバランシング方式を設定するには

コマンドプロンプトで、次のように入力します。

```
set lb vserver <name> [-lbMethod <lbMethod>]
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -lbMethod LRTM
<!--NeedCopy-->
```


GUI を使用して **LRTM** ロードバランシング方式を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [**LRTM**] を選択します。

CLI を使用してモニタで **LRTM** オプションを有効にするには

コマンドプロンプトで、次のように入力します。

```
set lb monitor <monitorName> <type> [-LRTM ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

例**:**

```
set lb monitor monitor-HTTP-1 HTTP -LRTM ENABLED
<!--NeedCopy-->
```

GUI を使用してモニタで **LRTM** オプションを有効にするには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動し、モニターを開きます。
2. 「拡張パラメータ」で、「**LRTM (**監視を使用した最小応答時間**)**」を選択します。

モニタの構成の詳細については、「負荷分散セットアップでのモニタの設定」を参照してください。

ハッシュ方式

October 7, 2021

特定の接続情報またはヘッダー情報のハッシュに基づく負荷分散方法は、Citrix ADC アプライアンスの負荷分散方式のほとんどを構成します。ハッシュは、基づいている情報よりも短く使いやすく、2 つの異なる情報が同じハッシュを生成しないため、互いに混乱しないように十分な情報を保持しています。

ハッシュ負荷分散方法は、キャッシュがインターネットまたは指定したオリジンサーバーからの幅広いコンテンツを提供する環境で使用できます。リクエストをキャッシュすると、リクエストとレスポンスのレイテンシーが削減され、リソース（CPU）の使用率が向上し、頻繁に使用される Web サイトやアプリケーションサーバーでキャッシュが普及します。これらのサイトは負荷分散の恩恵を受けるため、ハッシュ負荷分散方式は広く有用です。

Citrix ADC アプライアンスには、次のハッシュ方式が用意されています。

- URL ハッシュ方式

- ハッシュ関数
- 宛先 IP ハッシュ方式
- 送信元 IP ハッシュ方式
- 送信元 IP 宛先 IP ハッシュ方式
- 送信元 IP 送信元ポートのハッシュ方式
- コール ID ハッシュ方式
- トークン方式

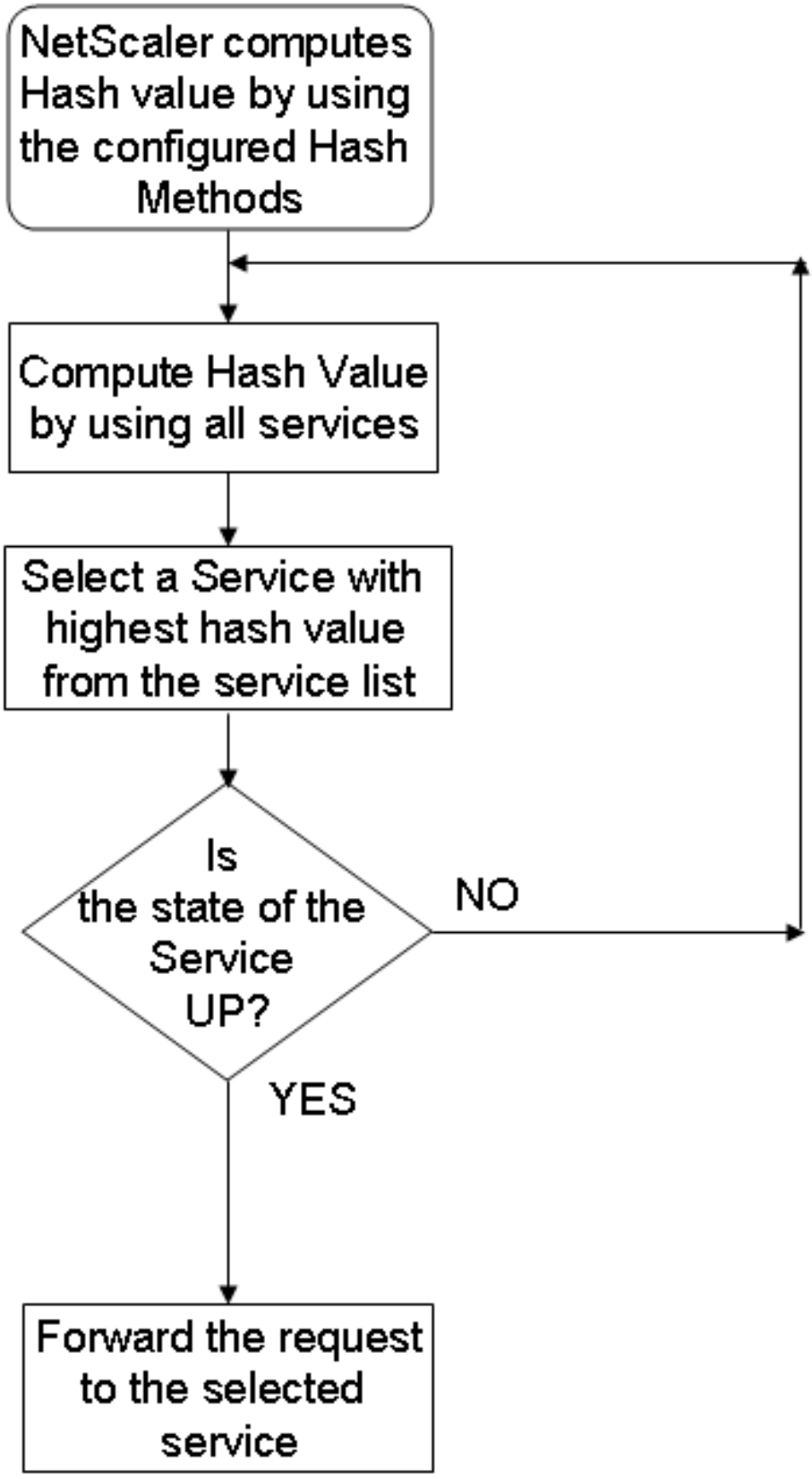
ほとんどのハッシュアルゴリズムは、次の 2 つのハッシュ値を計算します。

- サービスの IP アドレスとポートのハッシュ。
- 着信 URL、ドメイン名、送信元 IP アドレス、宛先 IP アドレス、または送信元と宛先 IP アドレスのハッシュ。設定されたハッシュ方式によって異なります。

次に、Citrix ADC アプライアンスは、これらのハッシュ値の両方を使用して新しいハッシュ値を生成します。最後に、ハッシュ値が最も高いサービスにリクエストを転送します。アプライアンスは各リクエストのハッシュ値を計算し、要求を処理するサービスを選択すると、キャッシュが生成されます。同じハッシュ値を持つ後続の要求は、同じサービスに送信されます。次のフローチャートは、このプロセスを示しています。

> 注
>
> Citrix ADC リリース 13.0 ビルド 79.x から、プライム再シャッフルアシスト CARP（PRAC）とジャンプテーブルアシストリングハッシュ（JAR）の一貫性のあるハッシュアルゴリズムがサポートされています。一貫性のあるハッシュアルゴリズムにより、ロードバランシングセットアップにサービスが追加または削除されたとき、またはロードバランシングセットアップでサービスフラップイベント中にサービスが中断される最小限に抑えられます。詳細については、「一貫性のあるハッシュアルゴリズム」を参照してください。

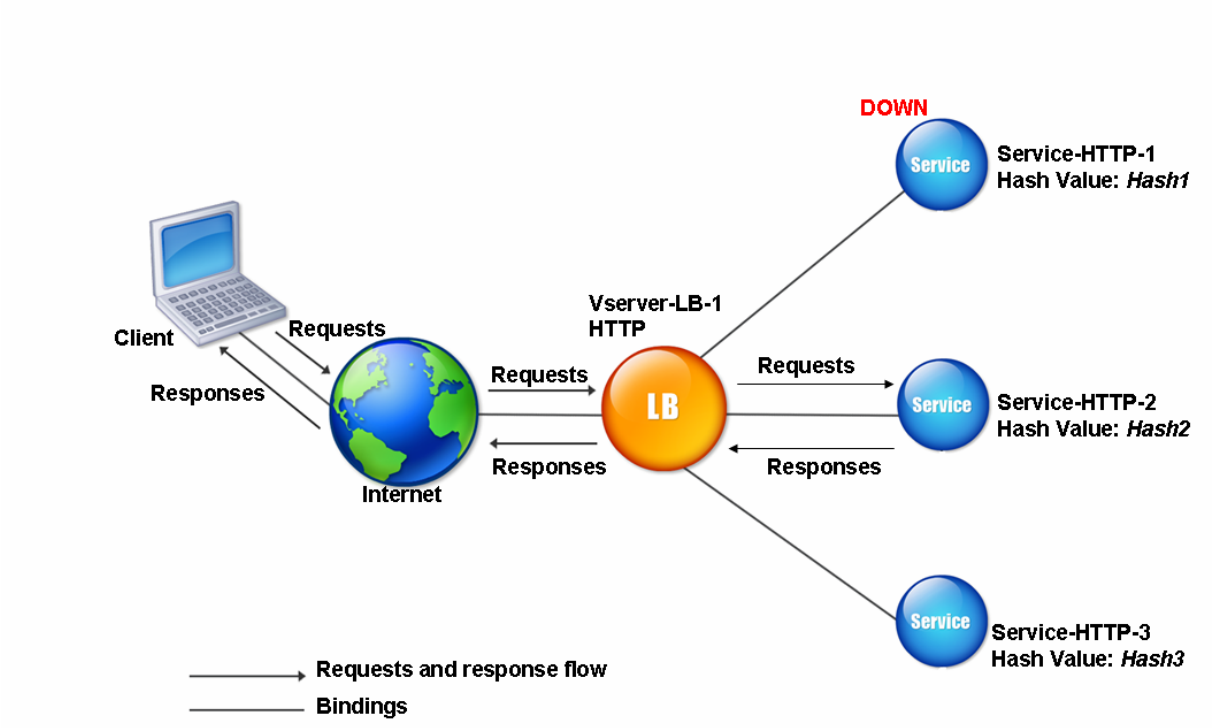
図 1: ハッシュメソッドによるリクエストの配布



ハッシュ方法は、IPv4 アドレスと IPv6 アドレスに適用できます。

3 つのサービス (Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3) が仮想サーバーにバインドされ、任意のハッシュ方式が構成され、ハッシュ値が Hash1 であるシナリオを考えてみましょう。設定されたサービスが UP になると、要求は Service-HTTP-1 に送信されます。Service-HTTP-1 がダウンしている場合、Citrix ADC アプライアンスはサービス数の最後のログのハッシュ値を計算します。アプライアンスは、Service-HTTP-2 など、ハッシュ値が最も大きいサービスを選択します。次の図は、このプロセスを示しています。

図 2：ハッシュメソッドのエンティティモデル



> 注
>
> Citrix ADC アプライアンスがハッシュ方式を使用してサービスを選択できない場合、デフォルトでは、着信要求のサービスを選択するための最小接続方法が使用されます。負荷分散設定のパフォーマンスに影響を与えずにキャッシュを再入力できるように、トラフィックが少ない時間帯にサービスを削除してサーバープールを調整します。

一貫性のあるハッシュアルゴリズム

一貫性のあるハッシュアルゴリズムは、ステートレスな永続性を実現するために使用されます。ハッシュベースの LB メソッドは、次の 3 つの一貫性のあるハッシュアルゴリズムのいずれかを使用します。

- キャッシュアレイルーティングプロトコル **(CAP)**

  CARP アルゴリズムは、複数のプロキシキャッシュサーバ間の HTTP 要求のロードバランシングに使用されます。このアルゴリズムはデフォルトで有効になっています。

- プライム再シャッフルアシストカープ **(PRAC)**

  Citrix ADC アプライアンスは、独自の PRAC アルゴリズムを使用して、均一なトラフィック分散を提供します。

- ジャンプテーブルアシストリングハッシュ **(JARH)**

  Citrix ADC アプライアンスは、独自の JARH アルゴリズムを使用して、トラフィックの一貫性と均一な分散を提供します。このアルゴリズムはハッシュフィンガーを使用します。指の数が多いほど、トラフィックの分布が向上します。ただし、指の数を増やすと、メモリ使用量も増加します。

CLI を使用して一貫性のあるハッシュアルゴリズムを選択するには

```
set lb parameter [-lbHashAlgorithm [DEFAULT|JARH|PRAC] [-lbHashFingers
    <positive_integer>]
<!--NeedCopy-->
```

例**:**

```
set lb parameter -lbHashAlgorithm JARH -lbHashFingers 10
<!--NeedCopy-->
```

引数**:**

- **lbHashAlgorithm**-次のハッシュベースのロードバランシング方法に使用するハッシュアルゴリズムを指定します。
  - URL ハッシュ方式
  - ハッシュ関数
  - 宛先 IP ハッシュ方式
  - 送信元 IP ハッシュ方式
  - 送信元 IP 宛先 IP ハッシュ方式
  - 送信元 IP 送信元ポートのハッシュ方式
  - コール ID ハッシュ方式
  - トークン方式

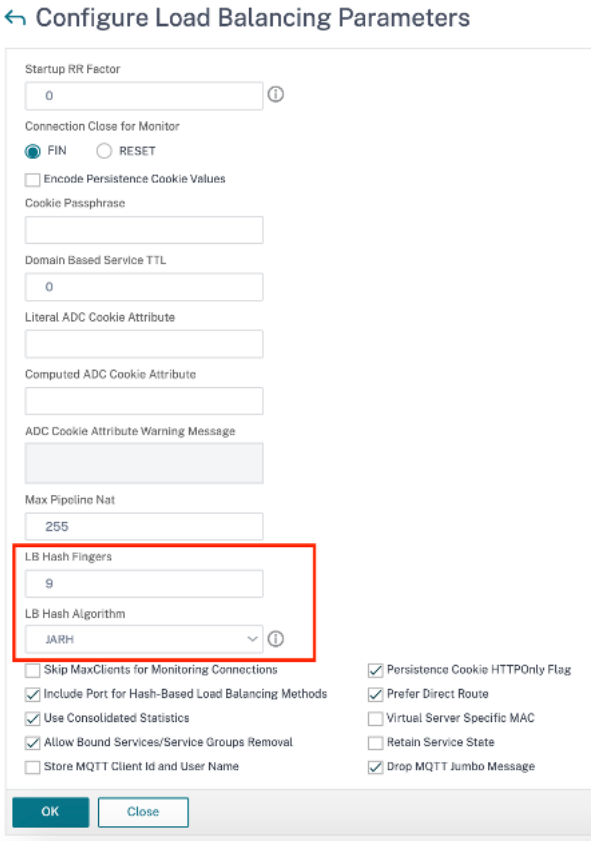
  指定可能な値:DEFAULT、PRAC、JARH
  デフォルト値:DEFAULT

- **lbHashFingers-**ハッシュベースの LB メソッドの PRAC および JARH アルゴリズムで使用する指の数を指定します。指の数を増やすと、余分なメモリを犠牲にしてトラフィックの分散が向上します。

  デフォルト値:256
  最小値:1
  最大値:1024

GUI を使用して一貫性のあるハッシュアルゴリズムを選択するには

1. **[**トラフィック管理**] > [**負荷分散**] > [**負荷分散パラメータの変更**]** に移動します。
2. [負荷分散パラメータの構成] ペインで、要件に基づいて次のフィールドに適切な値を入力します。
   - LB ハッシュフィンガー
   - [**LB Hash Algorithm**] フィールドで、ドロップダウンメニューから一貫性のあるハッシュアルゴリズムを選択します。



URL ハッシュメソッド

サービスの負荷分散に URL ハッシュ方式を使用するように Citrix ADC アプライアンスを構成すると、アプライアンスは受信要求に存在する HTTP URL のハッシュ値を生成します。ハッシュ値によって選択されたサービスが DOWN の場合、アルゴリズムには、アクティブなサービスのリストから別のサービスを選択する方法があります。アプライアンスは URL のハッシュ値をキャッシュし、同じ URL を使用する後続のリクエストを受信すると、同じサービスに転送します。アプライアンスが着信要求を解析できない場合、URL ハッシュ方式ではなく、ロードバランシングにラウンドロビン方式を使用します。

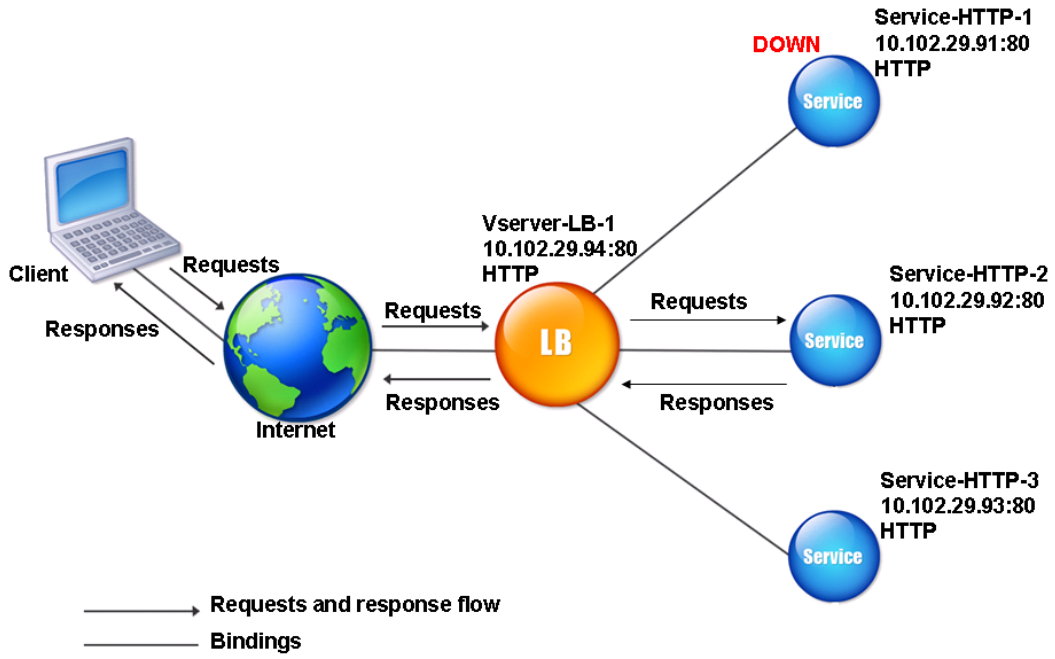
ハッシュ値を生成するために、アプライアンスは特定のアルゴリズムを使用し、URL の一部を考慮します。デフォルトでは、アプライアンスは URL の最初の 80 バイトを考慮します。URL が 80 バイト未満の場合は、完全な URL が使用されます。別の長さを指定できます。ハッシュの長さは 1 バイトから 4096 バイトまでです。一般に、数文字し

か異なる長い URL を使用する場合は、より均一な負荷分散を確保するために、ハッシュ長をできるだけ長くすることをお勧めします。

Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 という 3 つのサービスが仮想サーバーにバインドされ、仮想サーバーで構成される負荷分散方式が URL ハッシュ方式であるシナリオを考えてみます。仮想サーバーが要求を受信し、URL のハッシュ値は U1 です。アプライアンスはサービス HTTP-1 を選択します。Service-HTTP-1 が DOWN の場合、アプライアンスは Service-HTTP-2 を選択します。

次の図は、このプロセスを示しています。

図 3：URL ハッシュの動作



Service-HTTP-1 と Service-HTTP-2 の両方がダウンしている場合、アプライアンスはハッシュ値 U1 の要求をサービス HTTP-3 に送信します。

Service-HTTP-1 および Service-HTTP-2 がダウンしている場合、ハッシュ URL1 を生成する要求は Service-HTTP-3 に送信されます。これらのサービスが UP の場合、ハッシュ URL1 を生成する要求は次の方法で配布されます。

- Service-HTTP-2 が起動している場合、要求は Service-HTTP-2 に送信されます。
- Service-HTTP-1 が起動している場合、要求は Service-HTTP-1 に送信されます。
- Service-HTTP-1 と Service-HTTP-2 が同時にアップしている場合、要求は Service-HTTP-1 に送信されます。

URL ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。[URL Hash] としてロードバランシング方式を選択し、ハッシュ長をハッシュ値の生成に使用するバイト数に設定します。

ドメインハッシュメソッド

ドメインハッシュ方式を使用するように設定されたロードバランシング仮想サーバは、HTTP 要求内のドメイン名のハッシュ値を使用してサービスを選択します。ドメイン名は、HTTP 要求の着信 URL または Host ヘッダーのいずれかから取得されます。ドメイン名が URL と Host ヘッダーの両方に表示される場合、アプライアンスはその URL を優先します。

ドメイン名のハッシュを構成し、着信 HTTP リクエストにドメイン名が含まれていない場合、Citrix ADC アプライアンスはデフォルトでそのリクエストのラウンドロビン方式になります。

ハッシュ値の計算では、名前の長さまたはハッシュ長の値のいずれか小さい方が使用されます。デフォルトでは、Citrix ADC アプライアンスはドメイン名の最初の 80 バイトからハッシュ値を計算します。ハッシュ値を計算するときにドメイン名に異なるバイト数を指定するには、hashLength パラメータ（設定ユーティリティの Hash Length）を 1～4096（バイト）の値に設定します。

ドメインハッシュ方式を設定するには、ポリシーを含まない負荷分散方式の構成を参照してください。

宛先 **IP** ハッシュ方式

宛先 IP ハッシュ方式を使用するように設定されたロードバランシング仮想サーバは、宛先 IP アドレスのハッシュ値を使用してサーバを選択します。宛先 IP アドレスをマスクして、ハッシュ値の計算で使用する部分を指定できます。これにより、異なるネットワークからの要求が同じサブネット宛ての要求はすべて同じサーバーに送信されます。この方法は、IPv4 および IPv6 ベースの宛先サーバーをサポートします。

このロードバランシング方法は、キャッシュリダイレクト機能での使用に適しています。

IPv4 宛先サーバの宛先 IP ハッシュ方式を設定するには、netMask パラメータを設定します。この方法を IPv6 宛先サーバー用に構成するには、v6NetMaskLen パラメーターを使用します。構成ユーティリティで、[宛先 **IP** ハッシュ方式] を選択すると、これらのパラメータを設定するためのテキストボックスが表示されます。

宛先 IP ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の構成を参照してください。

送信元 **IP** ハッシュ方式

送信元 IP ハッシュ方式を使用するように構成された負荷分散仮想サーバーは、クライアントの IPv4 アドレスまたは IPv6 アドレスのハッシュ値を使用してサービスを選択します。特定のネットワークに属する送信元 IP アドレスからのすべての要求を特定の宛先サーバーに転送するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、netMask パラメーターを使用します。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。

送信元 IP ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

送信元 **IP** 宛先 **IP** ハッシュ方式

送信元 IP 宛先 IP ハッシュ方式を使用するように設定されたロードバランシング仮想サーバは、送信元と宛先 IP アドレス（IPv4 または IPv6）のハッシュ値を使用してサービスを選択します。ハッシュは対称です。ハッシュ値は、送信元と宛先 IP の順序に関係なく、同じです。これにより、特定のクライアントから同じ宛先に流れるすべてのパケットが同じサーバに送信されます。

特定のネットワークに属するすべての要求を特定の宛先サーバーに送信するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、netMask パラメーターを使用します。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。

送信元 IP 宛先 IP ハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

送信元 **IP** 送信元ポートのハッシュ方式

ソース IP ソースポートハッシュ方式を使用するように構成された負荷分散仮想サーバーは、送信元 IP（IPv4 または IPv6）のハッシュ値および送信元ポートのハッシュ値を使用してサービスを選択します。これにより、特定の接続上のすべてのパケットが同じサービスに送信されます。

この方法は、接続ミラーリングとファイアウォールの負荷分散で使用されます。接続ミラーリングの詳細については、「接続のフェイルオーバー」を参照してください。

特定のネットワークに属するすべての要求を特定の宛先サーバーに送信するには、送信元 IP アドレスをマスクする必要があります。IPv4 アドレスの場合は、netMask パラメーターを使用します。IPv6 アドレスの場合は、v6NetMaskLength パラメーターを使用します。

送信元 IP 送信元ポートのハッシュ方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

コール **ID** ハッシュメソッド

コール ID ハッシュ方式を使用するように設定されたロードバランシング仮想サーバは、SIP ヘッダーのコール ID のハッシュ値を使用してサービスを選択します。したがって、特定の SIP セッションのパケットは、常に同じプロキシサーバに送信されます。

この方法は、SIP ロードバランシングに適用できます。SIP ロードバランシングの詳細については、「SIP サービスのモニタリング」を参照してください。

コール ID ハッシュメソッドを設定するには、ポリシーを含まないロードバランシングメソッドの設定を参照してください。

最小帯域幅方式

October 7, 2021

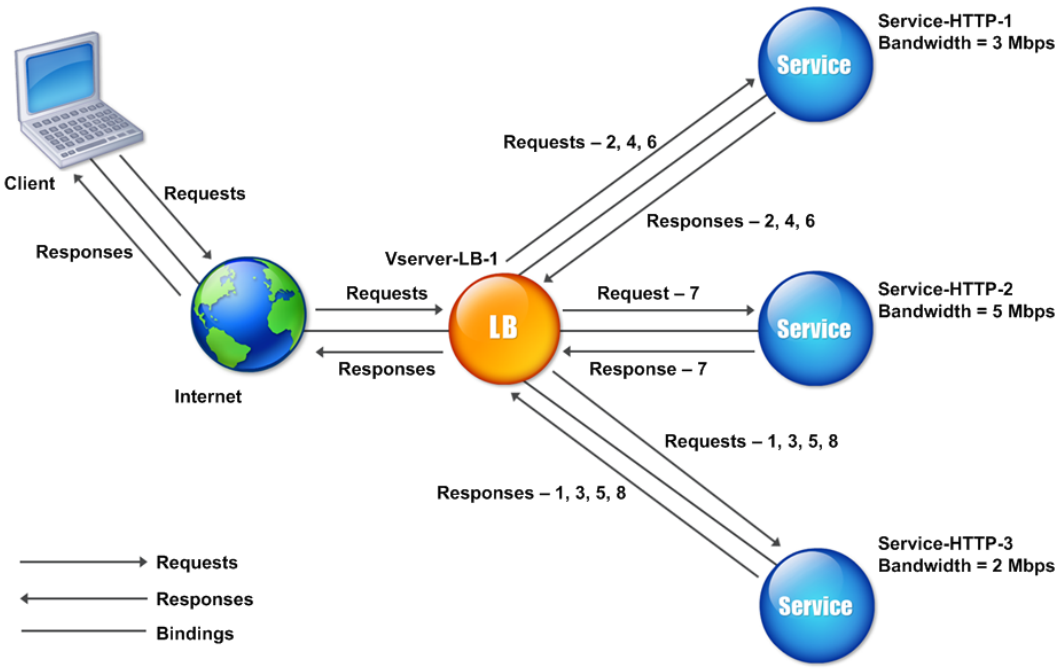
最小帯域幅方式を使用するように構成された負荷分散仮想サーバーは、現在最も少ないトラフィック量を処理しているサービスを選択します。メガビット/秒 (Mbps) 単位で測定されます。次に、仮想サーバが、最小帯域幅方式を使用してロードバランシングするサービスを選択する例を示します。

Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 の 3 つのサービスを検討してください。

- Service-HTTP-1 の帯域幅は 3 Mbps です。
- Service-HTTP-2 の帯域幅は 5 Mbps です。
- Service-HTTP-3 の帯域幅は 2 Mbps です。

次の図は、仮想サーバが最小帯域幅方式を使用して 3 つのサービスに要求を転送する方法を示しています。

図 1: 最小帯域幅ロードバランシング方式の動作



仮想サーバは、過去 14 秒間に送受信されたバイト数の合計である帯域幅値（N）を使用してサービスを選択します。要求ごとに 1 Mbps の帯域幅が必要な場合は、Citrix ADC アプライアンスは次のように要求を配信します。

- Service-HTTP-3 は最初の要求を受信します。これは、このサービスの N 値が最小であるためです。

- Service-HTTP-1 と Service-HTTP-3 は同じ N 値を持つため、仮想サーバはこれらのサーバのラウンドロビン方式に切り替わり、それらのサーバ間で交互に切り替えられます。Service-HTTP-1 は 2 番目の要求を受信し、Service-HTTP-3 は 3 番目の要求を受信し、Service-HTTP-1 は 4 番目の要求を受信し、Service-HTTP-3 は 5 番目の要求を受信し、Service-HTTP-1 は 6 番目の要求を受信します。
- Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ N 値を持つため、仮想サーバは Service-HTTP-2 をラウンドロビンリストに含めます。したがって、Service-HTTP-2 は 7 番目の要求を受信し、Service-HTTP-3 は 8 番目の要求を受信します。

次の表は、N の計算方法をまとめたものです。

リクエストを受け取りました	選択されたサービス	現在の N 値	注釈
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 の最小値は N です。
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 と Service-HTTP-3 は、同じ N 値を持ちます。
Request-3	Service-HTTP-3;（N = 3）	N = 4	Service-HTTP-1 と Service-HTTP-3 は、同じ N 値を持ちます。
Request-4	Service-HTTP-1;（N = 4）	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、同じ N 値を持ちます。
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、同じ N 値を持ちます。
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

注：仮想サーバーで RTSP NAT オプションを有効にすると、Citrix ADC アプライアンスは交換されたデータ数と制御バイト数を使用して、RTSP サービスの帯域幅使用率を決定します。RTSP NAT オプションの詳細については、「RTSP 接続の管理」を参照してください。

また、Citrix ADC アプライアンスは、異なるウェイトがサービスに割り当てられている場合、帯域幅と重みを使用して負荷分散を実行します。次の式で値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000/重量)

前の例のように、Service-HTTP-1 に重みが 2、Service-HTTP-2 に重みが 3、Service-HTTP-3 に重みが 4 が割り当てられているとします。Citrix ADC アプライアンスは、次のように要求を配信します。

- Service-HTTP-3 は、最初の 2 番目、3 番目、4 番目、5 番めの要求を受信します。これは、このサービスの Nw 値が最小であるためです。
- Service-HTTP-1 は 6 番目の要求を受信します。これは、このサービスの Nw 値が最小であるためです。
- Service-HTTP-3 は、このサービスの Nw 値が最小であるため、7 番目の要求を受信します。
- Service-HTTP-2 は 8 番目の要求を受信します。これは、このサービスが Nw の値が最も小さいためです。

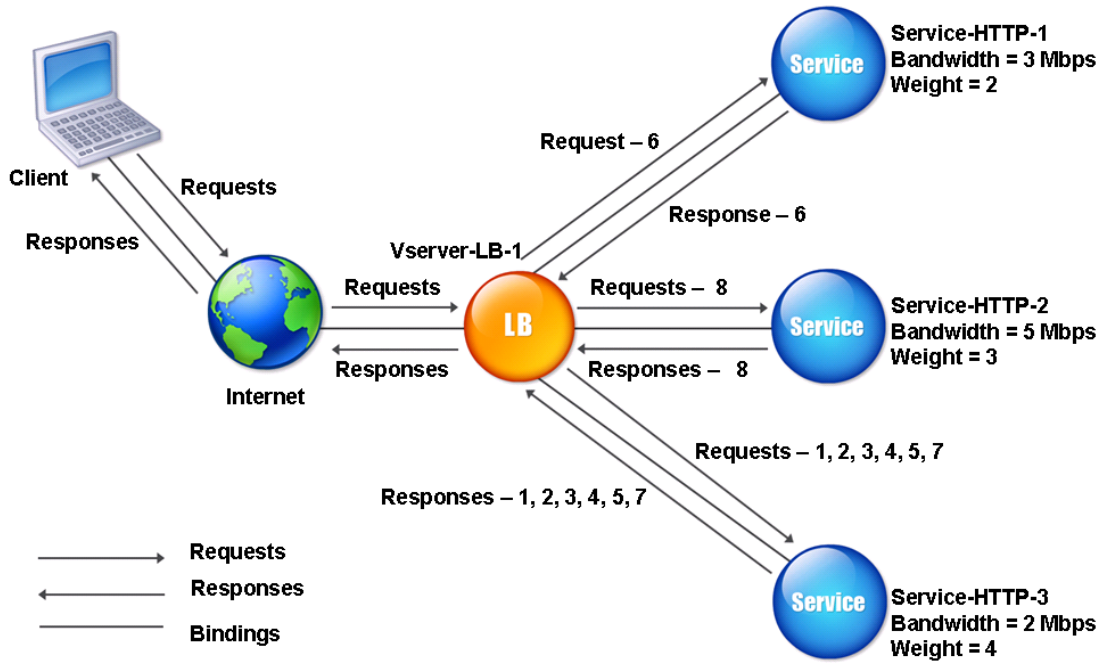
次の表は、Nw の計算方法をまとめたものです。

リクエストを受け取りました	選択されたサービス	現在の Nw 値（有効なトランザクション数）* (10000/重量)	注釈
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 の Nw 値は最小です。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3; (Nw = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 と Service-HTTP-3 は、同じ Nw 値を持ちます。
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 と Service-HTTP-3 は、同じ Nw 値を持ちます。
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 は、最小 Nw 値を持っています。

次の図は、重み付けがサービスに割り当てられるときに、仮想サーバが最小帯域幅方式を使用する方法を示していま

す。

図 2：重みが割り当てられている場合の最小帯域幅ロードバランシング方式の動作



最小帯域幅方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

最小パケット方式

October 7, 2021

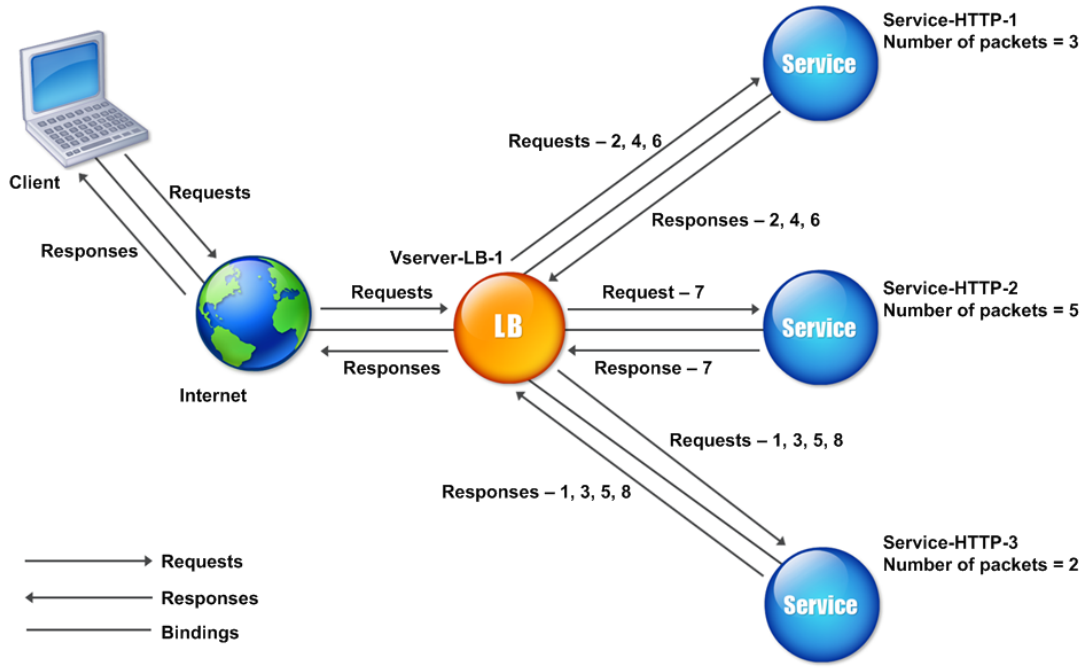
最小パケット方式を使用するように構成された負荷分散仮想サーバは、過去 14 秒間に最も少ないパケットを受信したサービスを選択します。

たとえば、Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 の 3 つのサービスを考えてみます。

- Service-HTTP-1 は過去 14 秒で 3 つのパケットを処理しました。
- Service-HTTP-2 は過去 14 秒で 5 つのパケットを処理しました。
- Service-HTTP-3 は過去 14 秒で 2 つのパケットを処理しました。

次の図は、Citrix ADC アプライアンスが最小パケット方式を使用して、受信する要求ごとにサービスを選択する方法を示しています。

図 1: 最小パケットロードバランシング方式の動作



Citrix ADC アプライアンスは、直近 14 秒間に各サービスが送受信したパケット数（N）を使用してサービスを選択します。このメソッドを使用すると、次のようにリクエストを配信します。

- Service-HTTP-3 は最初の要求を受信します。これは、このサービスの N 値が最小であるためです。
- Service-HTTP-1 と Service-HTTP-3 は同じ N 値を持つようになったため、仮想サーバはラウンドロビン方式に切り替わります。したがって、Service-HTTP-1 は 2 番目の要求を受信し、Service-HTTP-3 は 3 番目の要求を受信し、Service-HTTP-1 は 4 番目の要求を受信し、Service-HTTP-3 は 5 番目の要求を受信し、Service-HTTP-1 は 6 番目の要求を受信します。
- Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ N 値を持つため、仮想サーバは Service-HTTP-2 のラウンドロビン方式に切り替え、ラウンドロビンリストに含めます。したがって、Service-HTTP-2 は 7 番目の要求を受信し、Service-HTTP-3 は 8 番目の要求を受信します。

次の表は、N の計算方法をまとめたものです。

リクエストを受け取りました	選択されたサービス	現在の N 値	注釈
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 の最小値は N です。

リクエストを受け取りました	選択されたサービス	現在の N 値	注釈
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 と Service-HTTP-3 は、同じ N 値を持ちます。
Request-3	Service-HTTP-3; (N = 3)	N = 4	Service-HTTP-1 と Service-HTTP-3 は、同じ N 値を持ちます。
Request-4	Service-HTTP-1;（N = 4）	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、同じ N 値を持ちます。
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、同じ N 値を持ちます。
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

注：仮想サーバで RTSP NAT オプションを有効にすると、アプライアンスはデータおよび制御パケットの数を使用して、RTSP サービスのパケット数を計算します。RTSP NAT オプションの詳細については、「RTSP 接続の管理」を参照してください。

また、Citrix ADC アプライアンスは、各サービスに異なる重みが割り当てられている場合に、パケット数と重みを使用して負荷分散を実行します。次の式で値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000/重量)

前の例のように、Service-HTTP-1 に重みが 2、Service-HTTP-2 に重みが 3、Service-HTTP-3 に重みが 4 が割り当てられているとします。Citrix ADC アプライアンスは、次のように要求を配信します。

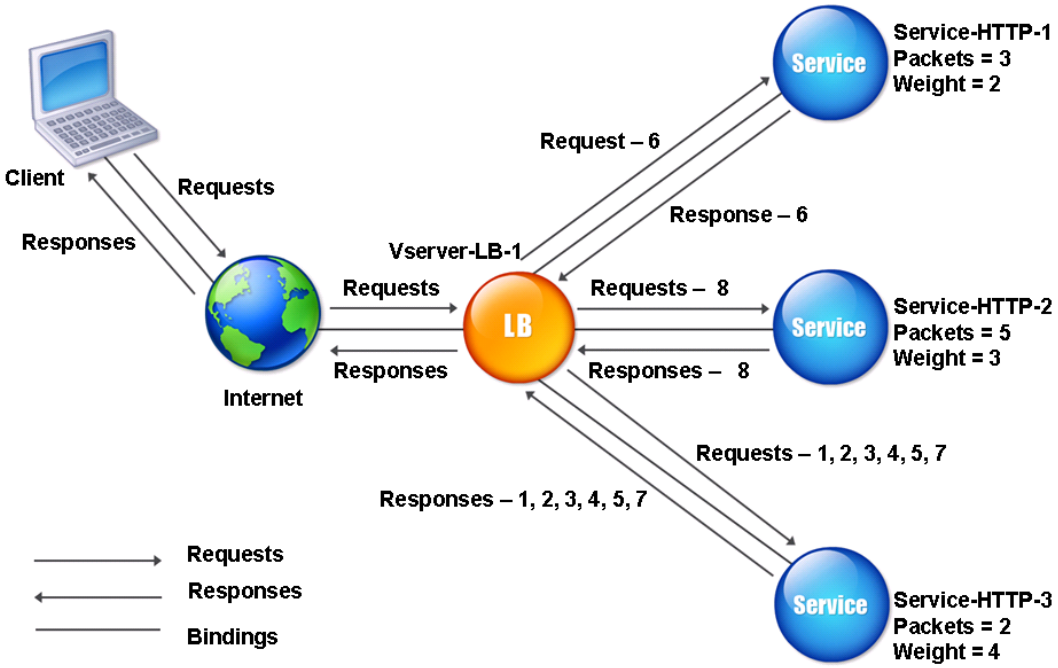
- Service-HTTP-3 は、最初の 2 番目、3 番目、4 番目、5 番めの要求を受信します。これは、このサービスの Nw 値が最小であるためです。
- Service-HTTP-1 は 6 番目の要求を受信します。これは、このサービスの Nw 値が最小であるためです。
- Service-HTTP-3 は、このサービスの Nw 値が最小であるため、7 番目の要求を受信します。
- Service-HTTP-2 は 8 番目の要求を受信します。これは、このサービスが Nw の値が最も小さいためです。

次の表は、Nw の計算方法をまとめたものです。

リクエストを受け取りました	選択されたサービス	現在の Nw 値（有効なトランザクション数）* (10000/重量)	注釈
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 の Nw 値は最小です。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3;（北方 =7500）	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 と Service-HTTP-3 は、同じ Nw 値を持ちます。
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 と Service-HTTP-3 は、同じ Nw 値を持ちます。
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 は、最小 Nw 値を持っています。

次の図は、重みが割り当てられたときに、仮想サーバが最小パケット方式を使用する方法を示しています。

図 2：重みが割り当てられている場合の最小パケット方式の動作



最小パケット方式を設定するには、ポリシーを含まないロードバランシング方式の設定を参照してください。

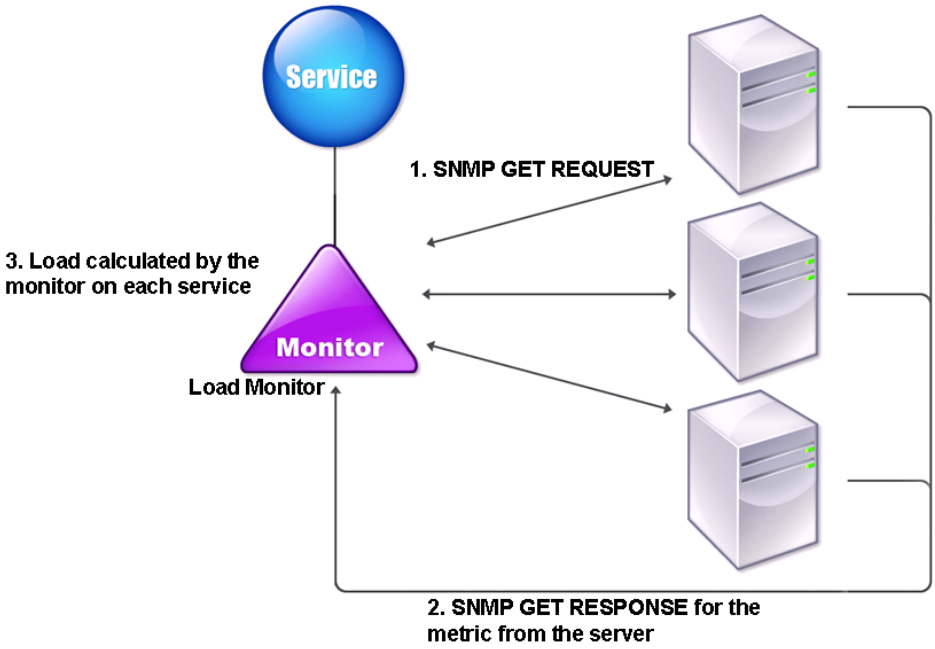
カスタムロード方法

October 7, 2021

カスタム負荷分散は、CPU 使用率、メモリ、応答時間などのサーバパラメータに対して実行されます。カスタムロード方式を使用する場合、Citrix ADC アプライアンスは通常、アクティブなトランザクションを処理していないサービスを選択します。ロードバランシング設定のすべてのサービスがアクティブなトランザクションを処理している場合、アプライアンスは負荷が最小のサービスを選択します。ロードモニタと呼ばれる特殊なタイプのモニタは、ネットワーク内の各サービスの負荷を計算します。ロードモニタはサービスの状態をマークしませんが、サービスが UP でない場合、ロードバランシングの決定からサービスを取り出します。

負荷モニタの詳細については、「負荷モニタについて」を参照してください。次の図は、ロードモニタの動作を示しています。

図 1: 負荷モニタの動作方法



負荷モニタは SNMP プローブを使用して、SNMP GET 要求をサービスに送信して、各サービスの負荷を計算します。このリクエストには、1 つ以上のオブジェクト ID (OID) が含まれます。サービスは、SNMP OID に対応するメトリックとともに、SNMP GET 応答で応答します。負荷モニターは、レスポンスメトリックを使用してサービスの負荷を計算します。

負荷モニターは、次のパラメーターを使用してサービスの負荷を計算します。

- Citrix ADC アプライアンスのテーブルとして存在する、SNMP プローブを介して取得されるメトリック値。
- 各メトリックに設定されたしきい値。
- 各指標に割り当てられた重み。

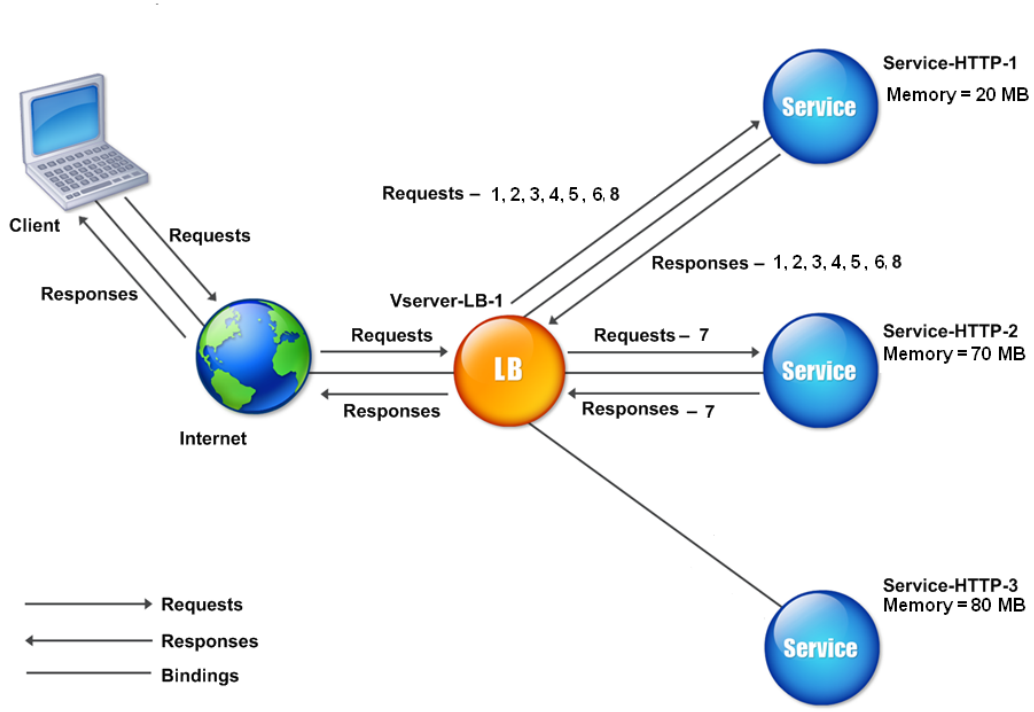
たとえば、Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 の 3 つのサービスを考えてみます。

- Service-HTTP-1 は 20 MB のメモリを使用しています。
- Service-HTTP-2 では、70 MB のメモリを使用しています。
- Service-HTTP-3 は 80 MB のメモリを使用しています。

負荷分散されたサーバーは、CPU やメモリ使用量などのメトリックをサービスにエクスポートし、それらを負荷モニターに提供することができます。ロードモニタは、OID が含まれた SNMP GET 要求をサービスに送信します。STRING タイプの SNMP OID はサポートされません。これは、STRING OID を使用して負荷を計算できないためです。負荷は、INT や gauge32 などの他のデータ型を使用して計算できます。3 つのサービスが要求に応答します。Citrix ADC アプライアンスは、エクスポートされたメトリックを比較し、使用可能なメモリが多くなるため、

Service-HTTP-1 を選択します。次の図は、このプロセスを示しています。

図 2：カスタムロードメソッドの仕組み



各要求が 10 MB のメモリを使用する場合、Citrix ADC アプライアンスは次のように要求を配信します。

- Service-HTTP-1 は、N 値が最小であるため、1 番目、2 番目、3 番目、4 番目、5 番めの要求を受信します。
- Service-HTTP-1 と Service-HTTP-2 は同じ負荷を持つようになったため、仮想サーバーはこれらのサーバーのラウンドロビン方式に戻ります。したがって、Service-HTTP-2 は 6 番目の要求を受信し、Service-HTTP-1 は 7 番目の要求を受信します。
- Service-HTTP-1、Service-HTTP-2、Service-HTTP-3 はすべて同じ負荷を持つため、仮想サーバも Service-HTTP-3 のラウンドロビン方式に戻ります。したがって、Service-HTTP-3 は 8 番目の要求を受信します。

次の表は、N の計算方法をまとめたものです。

リクエストを受け取りました	選択されたサービス	現在の N 値（有効取引数）	注釈
Request-1	Service-HTTP-1; (N = 20)	N = 30	Service-HTTP-3 の最小値は N です。
Request-2	Service-HTTP-1; (N = 30)	N = 40	-

リクエストを受け取りました	選択されたサービス	現在の N 値（有効取引数）	注釈
Request-3	Service-HTTP-1; (N = 40)	N = 50	-
Request-4	Service-HTTP-1; (N = 50)	N = 60	-
Request-5	Service-HTTP-1; (N = 60)	N = 70	-
Request-6	Service-HTTP-1; (N = 70)	N = 80	Service-HTTP-2 と Service-HTTP-3 は、同じ N 値を持ちます。
Request-7	Service-HTTP-2; (N = 70)	N = 80	Service-HTTP-3 は、同じ N 値を持ちます。
Request-8	Service-HTTP-1; (N = 80)	N = 90	Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、同じ N 値を持ちます。

サービスに異なる重みが割り当てられている場合、カスタムロードアルゴリズムは、各サービスの負荷と各サービスに割り当てられた重みの両方を考慮します。次の式で値 (Nw) を使用してサービスを選択します。

Nw = (N) * (10000/重量)

前の例のように、Service-HTTP-1 に重みが 4、Service-HTTP-2 に重みが 3、Service-HTTP-3 に重みが 2 が割り当てられているとします。各要求が 10 MB のメモリを使用する場合、Citrix ADC アプライアンスは次のように要求を配信します。

- Service-HTTP-1 は、Nw の値が最小であるため、1 番目、2 番目、3 番目、4 番目、5 番目、6 番目、7 番目、8 番めの要求を受信します。
- Service-HTTP-2 は、このサービスの Nw 値が最も小さいので、9 番目の要求を受信します。

Service-HTTP-3 は最大 Nw 値であるため、ロードバランシングには考慮されません。

次の表は、Nw の計算方法をまとめたものです。

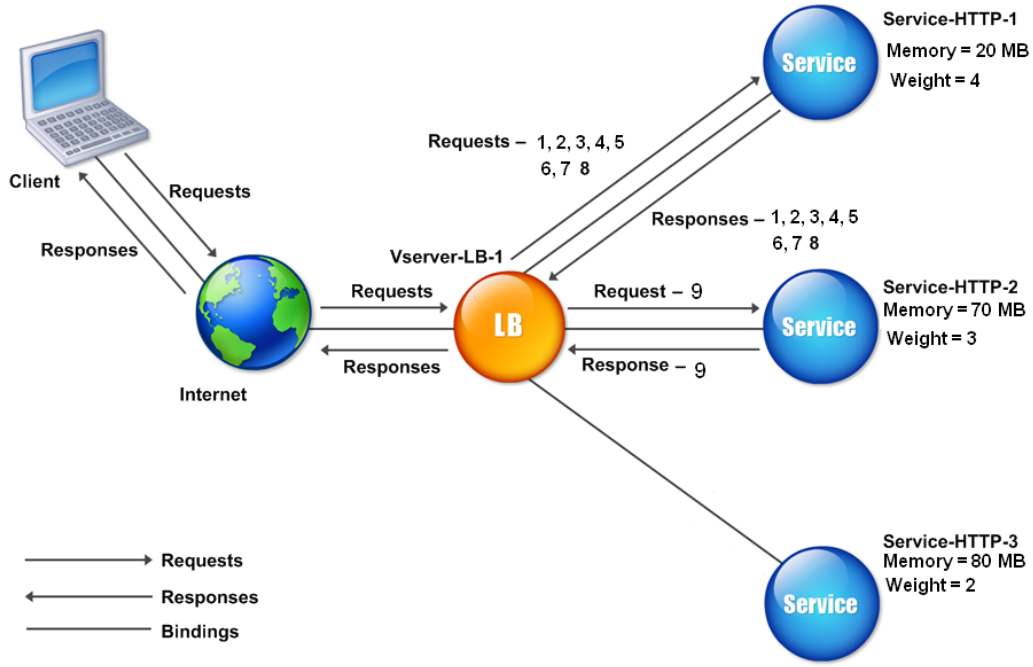
リクエストを受け取りました	選択されたサービス	現在の Nw 値（有効なトランザクション数）* (10000/重量)	注釈
Request-1	Service-HTTP-1; (Nw = 50000)	Nw = 75000	Service-HTTP-1 は Nw 値の最小値を持ちます。

リクエストを受け取りました	選択されたサービス	現在の Nw 値（有効なトランザクション数）* (10000/重量)	注釈
Request-2	Service-HTTP-1; (Nw = 5000)	Nw = 100000	-
Request-3	Service-HTTP-1; (Nw = 15000)	Nw = 125000	-
Request-4	Service-HTTP-1; (Nw = 20000)	Nw = 150000	-
Request-5	Service-HTTP-1; (Nw = 23333.34)	Nw = 175000	-
Request-6	Service-HTTP-1; (Nw = 25000)	Nw = 200000	-
Request-7	Service-HTTP-1; (Nw = 23333.34)	Nw = 225000	-
Request-8	Service-HTTP-1; (Nw = 25000)	Nw = 250000	-
Request-9	Service-HTTP-2; (Nw = 233333.34)	Nw = 266666.67	Service-HTTP-2 has the lowest Nw value.

Service-HTTP-1 は、アクティブなトランザクションを完了するか、他のサービス (Service-HTTP-2 および Service-HTTP-3) の Nw 値が 400,000 に等しい場合に、ロードバランシングの対象として選択されます。

次の図は、重みの割り当て時に Citrix ADC アプライアンスがカスタムロード方法を使用する方法を示しています。

図 3：ウェイトが割り当てられている場合のカスタムロード方法の動作



カスタムロード方式を構成するには、ポリシーを含まない負荷分散方式の構成を参照してください。

静的近接法

October 7, 2021

仮想サーバが静的近接方式を使用するように構成されている場合、近接基準に最も一致するサービスが選択されます。

静的近接方式を機能させるには、Citrix ADC アプライアンスを設定して、ロケーションファイルを介して入力された既存の静的近接データベースを使用するか、静的近接データベースにカスタムエントリを追加する必要があります。カスタムエントリを追加した後、ロケーション修飾子を設定できます。データベースを設定したら、静的な近接をロードバランシング方式として指定できます。

詳細については、次のトピックを参照してください。

- 位置ファイルを追加して静的近接データベースを作成する
- 静的近接データベースへのカスタムエントリの追加
- ロケーション修飾子の設定
- 静的近接法の指定

近接方法を指定する

スタティック近接データベースを設定したら、GLSB 方式としてスタティック近接を指定できます。

コマンドラインインターフェイスを使用して静的な近接を指定するには

コマンドプロンプトで次のコマンドを入力して、静的な近接性を構成し、構成を確認します。

```
set lb vserver <name> -lbMethod STATICPROXIMITY

show lb vserver <name>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -lbMethod STATICPROXIMITY

show lb vserver
<!--NeedCopy-->
```

GUI を使用して静的近接を指定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを選択します。
2. [編集**]** をクリックし、[メソッド] セクションを展開します。
3. [負荷分散方法] ボックスの一覧で、**[STATICPROXIMITY]** を選択します。

トークン方式

October 7, 2021

トークンメソッドを使用するように構成されたロードバランシング仮想サーバは、クライアント要求から抽出されたデータセグメントの値に基づいてサービスの選択を行います。データセグメントは、トークンと呼ばれます。トークンの場所とサイズを設定します。同じトークンを持つ後続の要求では、仮想サーバーは最初の要求を処理したのと同じサービスを選択します。

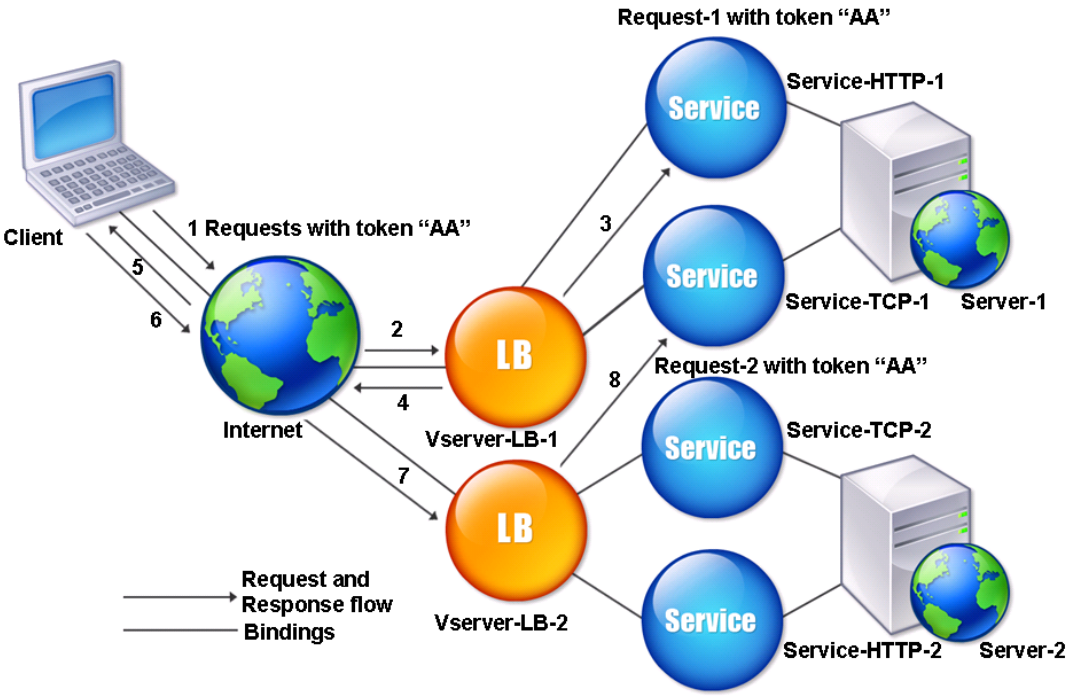
このメソッドはコンテンツ認識です。TCP、HTTP、および HTTPS 接続では、動作が異なります。HTTP サービスまたは HTTPS サービスの場合、トークンは HTTP ヘッダー、URL、または BODY にあります。トークンを検索するには、クラシックエクスプレッションまたはアドバンスエクスプレッションを指定または作成します。クラシック式または高度な式の詳細については、ポリシーの設定とリファレンスを参照してください。

HTTP サービスの場合、仮想サーバは TCP ペイロードの最初の 24 キロバイト（KB）で設定されたトークンを検索します。非 HTTP（TCP、SSL、および SSL_TCP）サービスの場合、仮想サーバは、16 パケットの合計サイズが 24 KB 未満の場合、最初の 16 パケットで設定されたトークンを検索します。ただし、16 パケットの合計サイズが 24 KB を超える場合、アプライアンスは最初の 24 KB のペイロードでトークンを検索します。異なるタイプの仮想サーバー間でこの負荷分散方法を使用すると、使用されるプロトコルに関係なく、同じトークンを提示する要求が適切なサービスに送信されるようにできます。

たとえば、Web コンテンツを含むサーバーで構成される負荷分散の設定を考えてみます。リクエストの URL クエリ部分内で特定の文字列（トークン）を検索するように Citrix ADC アプライアンスを構成する場合。Server-1 には、Service-HTTP-1 と Service-TCP-1 の 2 つのサービスがあり、Server-2 には、Service-HTTP-2 と Service-TCP-2 の 2 つのサービスがあります。TCP サービスは Vserver-LB-2 にバインドされ、HTTP サービスは Vserver-LB-1 にバインドされます。

Vserver-LB-1 がトークン AA で要求を受信すると、サービス Service-HTTP-1（server-1 にバインド）を選択して要求を処理します。Vserver-LB-2 が同じトークン (AA) を持つ別の要求を受信した場合、この要求は Service-TCP-1 サービスに送信されます。次の図は、このプロセスを示しています。

図 1: トークンメソッドの仕組み



コマンドラインインターフェイスを使用してトークンロードバランシング方式を構成するには

コマンドプロンプトで次のコマンドを入力して、トークンロードバランシング方式を構成し、構成を確認します。

```
set lb vserver <name> -lbMethod TOKEN -rule <rule> -datalength <length>
     -dataoffset <offset>

show lb vserver <name>
<!--NeedCopy-->
```

例:

```
set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -
    dataoffset 25

show lb vserver LB-VServer-1
<!--NeedCopy-->
```

構成ユーティリティを使用してトークンの負荷分散方法を構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[方法] をクリックします。
3. [ロードバランシング方法] リストで、[トークン] を選択し、式を指定します。

ポリシーを含まない負荷分散方式の構成

October 7, 2021

負荷分散セットアップの負荷分散アルゴリズムを選択したら、そのアルゴリズムを使用するように Citrix ADC アプライアンスを構成する必要があります。CLI または設定ユーティリティを使用して設定できます。

注:

トークン方式はポリシーベースであり、ここで説明するよりも多くの設定が必要です。トークンメソッドを設定するには、「トークンメソッド」を参照してください。

一部のハッシュベースの方法では、IP アドレスをマスクして、同じサブネットに属する要求を同じサーバーに送信できます。詳細については、「ハッシュメソッド」を参照してください。

コマンドラインインターフェイスを使用して負荷分散方法を設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -lbMethod <method>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -lbMethod LeastConnection
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散方法を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [方法] をクリックし、[負荷分散方法] ボックスの一覧で方法を選択します。

永続性と永続的な接続

September 27, 2022

HTTP などの負荷分散ステートレスプロトコルは、永続性が構成されていない場合、クライアント接続に関する状態情報の保守を中断します。すべての送信が同じセッションの一部であっても、同じクライアントからの異なる送信が異なるサーバーに送信されることがあります。ショッピングカートアプリケーションなど、特定の種類の Web アプリケーションを処理する負荷分散仮想サーバーで永続性を構成できます。

パーシスタンスを設定する前に、さまざまなタイプのパーシスタンス、その使用方法、およびそれらの意味を理解する必要があります。次に、Citrix ADC アプライアンスを構成して、それらを必要とする Web サイトおよび Web アプリケーションに永続的な接続を提供する必要があります。

また、バックアップパーシステンスを構成することもできます。これは、負荷分散仮想サーバーに対して構成されたプライマリタイプの永続性に障害が発生した場合に有効になります。グループの任意の仮想サーバへのクライアント転送を、同じクライアントから以前の転送を受信したサーバに転送できるように、永続性グループを設定できます。

RADIUS 負荷分散によるパーシステンスについては、永続性を使用した RADIUS 負荷分散の設定を参照してください。

持続性について

October 7, 2021

特定の負荷分散仮想サーバーに対して、いくつかのタイプの永続性から選択して、同じユーザーからショッピングカートアプリケーション、Web ベースの電子メール、またはその他のネットワークアプリケーションへのすべての接続を同じサービスにルーティングします。パーシステンスセッションは、指定した時間だけ有効です。

パーシステンスセッションに参加しているサーバが DOWN になると、ロードバランシング仮想サーバは設定されたロードバランシング方式を使用して新しいサービスを選択し、そのサービスによって表されるサーバとの新しいパーシステンスセッションを確立します。サーバが OUT OF SERVICE になった場合、既存の永続性セッションは引き続き処理されますが、仮想サーバは新しいトラフィックを送信しません。シャットダウン期間が経過すると、仮想サーバは既存のクライアントからサービスへの接続の転送を停止し、既存の接続を閉じ、必要に応じてこれらのクライアントを新しいサービスにリダイレクトします。

構成するパーシステンスのタイプによっては、Citrix ADC アプライアンスがソース IP、宛先 IP、SSL セッション ID、ホストヘッダーまたは URL ヘッダー、またはこれらのヘッダーの組み合わせを調べて、各接続を適切なパーシステンスセッションに配置する場合があります。また、Web サーバーによって発行された Cookie、任意に割り当てられたトークン、または論理ルールに基づいてパーシステンスをベースにすることもできます。アプライアンスが適切な永続セッションと接続を一致させることができ、永続性の基礎として使用されるほとんどすべてのもの。

次の表は、Citrix ADC アプライアンスで使用可能なパーシステンスタイプをまとめたものです。

永続性タイプ	説明
接続元 IP	SOURCEIP. 同じクライアント IP アドレスからの接続は、同じ永続性セッションの一部です。
HTTP Cookie	COOKIEINSERT. 同じ HTTP Cookie ヘッダーを持つ接続は、同じ永続性セッションの一部です。
SSL Session ID	SSLSESSION. 同じ SSL セッション ID を持つ接続は、同じ永続性セッションの一部です。
URL Passive	URLPASSIVE。同じ URL への接続は、同じ永続性セッションの一部として扱われます。
Custom Server ID	CUSTOMSERVERID。同じ HTTP HOST ヘッダーを持つ接続は、同じ永続性セッションの一部として扱われます。
接続先 IP	DESTIP。同じ宛先 IP への接続は、同じ永続性セッションの一部として扱われます。
送信元と宛先 IP	SRCIPDESTIP。同じ送信元 IP と宛先 IP の両方からの接続は、同じ永続性セッションの一部として扱われます。

永続性タイプ	説明
SIP コール ID	CALLID。SIP ヘッダーに同じコール ID を持つ接続は、同じ永続性セッションの一部として扱われます。
RTSP セッション ID	RTSPSID。同じ RTSP セッション ID を持つ接続は、同じ永続性セッションの一部として扱われます。
ユーザー定義ルール	RULE。ユーザー定義の規則に一致する接続は、同じ永続性セッションの一部として扱われます。

表 1. 持続性のタイプ

構成したパーシステンスのタイプに応じて、仮想サーバーでは、Citrix ADC アプライアンス上の RAM 容量によって課される制限まで、250,000 の同時永続接続または任意の数の永続接続をサポートできます。次の表は、各カテゴリに分類される永続性のタイプを示しています。

永続性タイプ	サポートされる同時持続接続の数
送信元 IP、SSL セッション ID、ルール、宛先 IP、送信元 IP/宛先 IP、SIP コール ID、RTSP セッション ID	250 K
クッキー、URL サーバー ID、カスタムサーバー ID	メモリの上限。CookieInsert では、タイムアウトが 0 以外の場合、接続数はメモリによって制限されます。

表 2. サポートされる同時接続のパーシステンスタイプと数

一部のタイプの永続性は、特定のタイプの仮想サーバに固有のものです。次の表は、各タイプの永続性と、どのタイプの仮想サーバーでサポートされる永続性のタイプを示しています。

永続性タイプ	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEI	はい	はい	はい	はい	はい	はい	いいえ	いいえ
COOKIEINSERT	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
SSLSESS	いいえ	はい	いいえ	いいえ	はい	はい	いいえ	いいえ
URLPASSIVE	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
CUSTOMS	はい	はい	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
RULE	はい	はい	はい	いいえ	いいえ		いいえ	いいえ
SRCIPDE	はい	はい	はい	はい	はい	はい	いいえ	いいえ
DESTIP	はい	はい	はい	はい	はい	はい	いいえ	いいえ

永続性タイプ	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
CALLID	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい
RTSPID	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい	いいえ

表 3. パーシステンスタイプと仮想サーバーのタイプの関係

送信元 IP アドレスの永続性

October 7, 2021

送信元 IP パーシステンスが設定されている場合、ロードバランシング仮想サーバは、設定されたロードバランシング方式を使用して最初の要求のサービスを選択し、送信元 IP アドレス（クライアント IP アドレス）を使用して、そのクライアントからの後続の要求を識別し、同じサービスに送信します。タイムアウト値を設定して、セッションの最大非アクティブ期間を指定できます。タイムアウト値が期限切れになると、セッションは破棄され、設定された負荷分散アルゴリズムを使用して新しいサーバーが選択されます。

注意: 状況によっては、ソース IP アドレスに基づくパーシステンスを使用すると、サーバーが過負荷になることがあります。単一の Web サイトまたはアプリケーションに対するすべての要求は、複数の場所にリダイレクトされる場合でも、単一のゲートウェイを介して Citrix ADC アプライアンスにルーティングされます。複数のプロキシ環境では、クライアント要求が同じクライアントから送信される場合でも、クライアント要求が異なる送信元 IP アドレスを持つことがよくあります。その結果、単一のセッションを作成する必要がある永続性セッションが急速に増加します。この問題は、「メガプロキシの問題」と呼ばれます。これを防ぐには、ソース IP ベースの永続性の代わりに HTTP cookie ベースの永続性を使用できます。

送信元 IP アドレスに基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの設定を参照してください。

注：すべての着信トラフィックがネットワークアドレス変換（NAT）デバイスまたはプロキシの背後から送られる場合、Citrix ADC アプライアンスでは単一の送信元 IP アドレスから送られるように見えます。これにより、送信元 IP パーシステンスが正常に機能しなくなります。この場合は、別のパーシステンスタイプを選択する必要があります。

HTTP クッキーの永続性

October 7, 2021

HTTP Cookie の永続性が構成されている場合、Citrix ADC アプライアンスは最初のクライアント要求の HTTP ヘッダーに Cookie を設定します。Cookie には、負荷分散アルゴリズムによって選択されたサービスの IP アドレスとポートが含まれます。任意の HTTP 接続と同様に、クライアントはそのクッキーを後続のリクエストに含めます。

Citrix ADC アプライアンスはクッキーを検出すると、クッキーのサービス IP とポートに要求を転送し、接続の永続性を維持します。このタイプの永続性は、HTTP または HTTPS タイプの仮想サーバーで使用できます。このパーシステンスタイプはアプライアンスリソースを消費しないため、永続的なクライアントの数に制限はありません。

注: クライアントの Web ブラウザが Cookie を拒否するように構成されている場合、HTTP Cookie ベースの永続性は機能しません。ウェブサイトでクッキーチェックを設定し、クッキーを適切に保存していないように見えるクライアントに、ウェブサイトで使用する場合は Cookie を有効にする必要があることを警告することをお勧めします。

Citrix ADC アプライアンスが挿入するクッキーの形式は次のとおりです。

```
NSC_XXXX=<ServiceIP ><ServicePort>
```

各項目の意味は次のとおりです:

- NSC_XXXX は、仮想サーバー名から派生した仮想サーバー ID です。
- サービス IP とサービスポートは、それぞれサービス IP アドレスとサービスポートのエンコードされた表現です。IP アドレスとポートは別々にエンコードされます。

このタイプの永続性のタイムアウト値を設定して、セッションの非アクティブ期間を指定できます。指定された期間接続がアクティブでない場合、Citrix ADC アプライアンスは永続セッションを破棄します。同じクライアントからの後続の接続では、設定された負荷分散方式に基づいて新しいサーバが選択され、新しい永続セッションが確立されます。

注: タイムアウト値を 0 に設定すると、Citrix ADC アプライアンスは有効期限を指定しませんが、クライアントのブラウザーのシャットダウン時に保存されないセッション Cookie を設定します。

デフォルトでは、クライアントブラウザとの互換性を最大限に高めるために、Citrix ADC アプライアンスは HTTP バージョン 0 の Cookie を設定します。(特定の HTTP プロキシのみがバージョン 1 の Cookie を理解します。最も一般的に使用されるブラウザでは理解しません)。RFC2109 に準拠するために、HTTP バージョン 1 のクッキーを設定するようにアプライアンスを設定できます。HTTP バージョン 0 のクッキーの場合、アプライアンスはクッキーの有効期限日時を絶対世界協定時刻（GMT）として挿入します。この値は、アプライアンスの現在の GMT 時間とタイムアウト値の合計として計算されます。HTTP バージョン 1 のクッキーの場合、アプライアンスは HTTP クッキーの「最大経過時間」属性を設定することにより、相対的な有効期限を挿入します。この場合、クライアントのブラウザは実際の有効期限を計算します。

アプライアンスによって挿入された Cookie に基づいてパーシステンスを設定するには、ルールを必要としない永続性タイプの設定を参照してください。

HTTP Cookie では、アプライアンスはデフォルトで Cookie がスクリプト不能であり、クライアントアプリケーションに公開してはならないことを示すHTTPOnlyフラグを設定します。したがって、クライアント側のスクリプトは Cookie にアクセスできず、クライアントはクロスサイトスクリプティングの影響を受けません。

ただし、一部のブラウザはHTTPOnlyフラグをサポートしていないため、Cookie を返さない場合があります。その結果、永続性が壊れます。このフラグをサポートしていないブラウザでは、永続性 Cookie のHTTPOnlyフラグを省略できます。

CLI を使用して HTTPOnly フラグ設定を変更するには

コマンドプロンプトで入力します。

```
set lb parameter -httpOnlyCookieFlag (ENABLED|DISABLED)
<!--NeedCopy-->
```

例:

```
>  set lb parameter -httpOnlyCookieFlag disabled
   Done
> show lb parameter
  Global LB parameters:
       Persistence Cookie HttpOnly Flag: DISABLED
       Use port for hash LB: YES
 Done
<!--NeedCopy-->
```

GUI を使用して HTTPOnly フラグの設定を変更するには

1. [トラフィック管理] > [ロードバランシング] > [ロードバランシングパラメータの設定] に移動し、[パーシステンス **Cookie HttpOnly**] フラグを選択またはクリアします。

クッキーの暗号化

リリース 10.5 ビルド 55.8 から、任意の SSL 暗号化に加えてクッキーを暗号化できます。

コマンドラインインターフェイスを使用して **cookie** を暗号化するには、コマンドプロンプトで

```
set lb parameter -UseEncryptedPersistenceCookie ENABLED -
    cookiePassphrase test
<!--NeedCopy-->
```

構成ユーティリティを使用して **Cookie** を暗号化するには

1. [トラフィック管理] > [ロードバランシングパラメータの変更] に移動し、[永続性 **Cookie** 値のエンコード] を選択し、[**Cookie** パスフレーズ] にパスフレーズを入力します。

SSL セッション ID の永続性

October 7, 2021

SSL セッション ID の永続性が構成されている場合、Citrix ADC アプライアンスは、SSL ハンドシェイクプロセスの一部である SSL セッション ID を使用して、最初の要求がサービスに送信される前に永続セッションを作成します。負荷分散仮想サーバは、同じ SSL セッション ID を持つ後続の要求を同じサービスに送信します。このタイプの永続性は、SSL ブリッジサービスに使用されます。

注:

このタイプの永続性を選択する前に、ユーザーが考慮する必要のある問題が 2 つあります。まず、このタイプのパーシステンスは、Citrix ADC アプライアンス上のリソースを消費します。これにより、サポートできる同時パーシスタンスセッションの数が制限されます。複数のパーシステンス・セッションをサポートする場合は、別のタイプの永続性を選択できます。

次に、クライアントと負荷分散サーバーがトランザクション中にセッション ID を再ネゴシエートする必要がある場合、永続性は維持されず、クライアントの次の要求を受信したときに新しい永続セッションが作成されます。これにより、Web サイトでのクライアントのアクティビティが中断され、クライアントからセッションの再認証または再起動を求められることがあります。また、タイムアウトの値が大きすぎると、多数の放棄セッションが発生する可能性があります。

SSL セッション ID に基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの構成を参照してください。

> 注
>
> セッション・チケットでは、SSL セッション ID の永続化はサポートされていません。

SSL セッション ID のパーシステンスのサポートをバックアップする

NetScaler リリース 12.0 ビルド 56.20 から、ソース IP の永続性は、SSL セッション ID の永続性のバックアップパーシステンスタイプとしてサポートされています。クライアントと負荷分散サーバがセッションを再ネゴシエートし、ソース IP パーシステンスがバックアップパーシステンスとして設定されている場合、クライアント要求は同じサーバに転送されます。

SSL セッション ID のバックアップパーシステンスをサポートするために、Citrix ADC アプライアンスは、クライアント要求が初めて受信されたときにソース IP と SSL セッション ID の両方のセッションエントリを作成します。同じセッション ID を含む後続の要求では、SSL セッション ID が使用されます。ただし、クライアントと負荷分散されたサーバーがセッションを再ネゴシエートすると、クライアント要求は、ソース IP 永続性を使用して同じサーバーに転送され、新しい SSL セッション ID 永続性エントリが作成されます。

バックアップパーシステンスの構成の詳細については、バックアップ永続性の構成を参照してください。

Diameter AVP 番号の永続性

October 7, 2021

Diameter メッセージの属性値ペア (AVP) 番号に基づくパーシステンスを使用して、Diameter セッションを作成できます。Citrix ADC アプライアンスは、Diameter メッセージで AVP を検出すると、AVP の値に基づいてパーシステンスセッションを作成します。AVP の値と一致する後続のすべてのメッセージは、以前に選択したサーバに送信されます。AVP の値が永続性セッションと一致しない場合、新しい値に対して新しいセッションが作成されます。

注：AVP 番号が Diameter ベースプロトコル RFC 6733 で定義されておらず、その番号がグループ化された AVP 内にネストされている場合は、AVP 番号のシーケンス（最大 3）を親から子の順に定義する必要があります。たとえば、持続する AVP 番号 X が Z にネストされている AVP Y 内にネストされている場合、リストを Z Y X として定義します。

コマンドラインインターフェイスを使用して仮想サーバーで Diameter ベースの永続性を構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
set lb vserver <name> -PersistenceType <type-> persistAVPno <
    positive_integer>
<!--NeedCopy-->
```

例:

```
set lb vserver diameter_vs -persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

カスタムサーバー ID の永続性

October 7, 2021

カスタムサーバー ID の永続性メソッドでは、クライアント要求で指定されたサーバー ID を使用して永続性を維持します。このタイプの永続性を機能させるには、まずサービスでサーバー ID を設定する必要があります。Citrix ADC アプライアンスは、クライアント要求の URL をチェックし、指定されたサーバー ID に関連付けられたサーバーに接続します。サービスプロバイダーは、特定のサービスに対するリクエストで提供されるサーバー ID をユーザーが認識していることを確認する必要があります。

たとえば、サイトがイメージ、テキスト、マルチメディアなどの異なるタイプのデータを異なるサーバから提供している場合は、各サーバにサーバ ID を割り当てることができます。Citrix ADC アプライアンスでは、対応するサービ

スに対してこれらのサーバー ID を指定し、対応する負荷分散仮想サーバーでカスタムサーバー ID パーシステンスを構成します。要求を送信するとき、クライアントは、必要なデータ型を示すサーバー ID を URL に挿入します。

カスタム・サーバ ID の永続性を構成するには、次の手順を実行します。

- 負荷分散の設定で、永続性を維持するためにユーザー定義のサーバー ID を使用する各サービスにサーバー ID を割り当てます。英数字のサーバ ID を使用できます。
- デフォルトの構文式言語で、サーバ ID の URL クエリを調べて、トラフィックを対応するサーバに転送する規則を指定します。
- カスタムサーバ ID の永続性を設定します。

注: 永続性タイムアウト値は、カスタムサーバー ID のパーシステンスタイプには影響しません。このパーシステンスタイプはクライアント情報を格納しないため、永続クライアントの最大数に制限はありません。

例:

2 つのサービスを使用するロードバランシング設定では、サーバ ID 2345-photo-56789 を Service-1 に、サーバ ID 2345-drawing-abb123 を Service-2 に割り当てます。これらのサービスを Web11 という名前の仮想サーバにバインドします。

```
1 set service Service-1 10.102.29.5 -CustomServerID 2345-photo-56789
2
3 set service Service-2 10.102.29.6 -CustomServerID 2345-drawing-abb123
4 <!--NeedCopy-->
```

仮想サーバー Web11 で、カスタムサーバー ID の永続性を有効にします。

文字列「sid=」を含むすべての URL クエリが検査されるように、次の式を作成します。

HTTP.REQ.URL.AFTER_STR(“sid=”)

例:

```
1 set lb vserver Web11 -persistenceType customserverID -rule ”HTTP.REQ.
     URL.AFTER_STR(”sid=”)”
2
3 bind lb vserver Web11 Service-[1-2]
4 <!--NeedCopy-->
```

クライアントが次の URL を持つ要求を Web11 の IP アドレスに送信すると、アプライアンスは要求を Service-2 に送信し、永続性を尊重します。

例:

```
http://www.example.com/index.asp?&sid=2345-drawing-abb123
```

デフォルト構文ポリシー式の詳細については、「ポリシーの設定およびリファレンス」を参照してください。

構成ユーティリティを使用してカスタムサーバー **ID** の永続性を構成するには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. サービスを開き、サーバー ID を設定します。
3. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
4. [詳細設定] で、[持続性] を選択します。
5.「カスタム」(CUSTOMESERVERID) を選択し、式を指定します。

IP アドレスの永続性

October 7, 2021

永続性は、宛先 IP アドレス、または送信元 IP アドレスと宛先 IP アドレスの両方に基づいて設定できます。

宛先 **IP** アドレスに基づくパーシステンス

宛先 IP アドレスベースの永続性では、Citrix ADC アプライアンスが新しいクライアントから要求を受信すると、仮想サーバーによって選択されたサービスの IP アドレス（宛先 IP アドレス）に基づいて永続性セッションが作成されます。後で、同じ宛先 IP に要求を同じサービスに送信します。このタイプの永続性は、リンクロードバランシングで使用されます。リンク負荷分散の詳細については、リンク負荷分散を参照してください。

宛先 IP パーシステンスのタイムアウト値は、送信元 IP アドレスに基づく永続性で説明されている送信元 IPパーシステンスのタイムアウト値と同じです。

宛先 IP アドレスに基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの設定を参照してください。

送信元および宛先 **IP** アドレスに基づくパーシステンス

送信元と宛先の IP アドレスベースのパーシステンスでは、Citrix ADC アプライアンスが要求を受信すると、クライアントの IP アドレス（送信元 IP アドレス）と仮想サーバーによって選択されたサービスの IP アドレス（宛先 IP アドレス）の両方に基づいてパーシステンスセッションが作成されます。後で、同じ送信元 IP および同じ宛先 IP からの要求を同じサービスに送信します。

宛先 IP パーシステンスのタイムアウト値は、送信元 IP アドレスに基づく永続性で説明されている送信元 IPパーシステンスのタイムアウト値と同じです。

送信元 IP アドレスと宛先 IP アドレスの両方に基づいてパーシステンスを設定するには、ルールを必要としない永続性のタイプの設定を参照してください。

SIP コール ID の永続性

October 7, 2021

SIP コール ID の永続性により、Citrix ADC アプライアンスは SIP ヘッダーのコール ID に基づいてサービスを選択します。これにより、特定の SIP セッションのパケットを同じサービス、つまり同じロードバランシングされたサーバに送信できます。このパーシステンスタイプは、特に SIP ロードバランシングに適用されます。SIP ロードバランシングの詳細については、「SIP サービスのモニタリング」を参照してください。

SIP コール ID に基づいてパーシステンスを設定するには、ルールを必要としない永続性タイプの設定を参照してください。

RTSP セッション ID パーシステンス

October 7, 2021

RTSP セッション ID 永続性では、Citrix ADC アプライアンスが新しいクライアントから要求を受信すると、RTSP パケットヘッダーにリアルタイムストリーミングプロトコル（RTSP）セッション ID に基づいて永続セッションが作成され、構成された負荷分散によって選択された RTSP サービスに要求が送信されます。メソッド。これは、同じセッション ID を含む後続の要求を同じサービスに送信します。このパーシステンスタイプは、特に SIP ロードバランシングに適用されます。SIP ロードバランシングの詳細については、「SIP サービスのモニタリング」を参照してください。

注：RTSP セッション ID パーシステンスは、RTSP 仮想サーバ上でデフォルトで設定されており、その設定を変更することはできません。

異なる RTSP サーバが同じセッション ID を発行することがあります。この場合、RTSP セッション ID のみを使用して、クライアントと RTSP サーバ間で一意のセッションを作成できません。同じセッション ID を発行する可能性のある複数の RTSP サーバがある場合は、セッション ID にサーバの IP アドレスとポートを追加するようにアプライアンスを設定して、永続性を確立するために使用できる一意のトークンを作成できます。これをセッション ID マッピングと呼びます。

RTSP セッション ID に基づいてパーシステンスを設定するには、ルールを必要としない永続性タイプの設定を参照してください。

重要: セッション ID マッピングを使用する必要がある場合は、負荷分散セットアップ内で各サービスを構成するときに次のパラメータを設定する必要があります。また、非永続的な接続が RTSP 仮想サーバを経由してルーティングされていないことを確認します。

URL パッシブ永続性の設定

October 7, 2021

URL パッシブパーシブパーシブでは、Citrix ADC アプライアンスがクライアントから要求を受信すると、クライアント要求からサーバーの IP アドレスポート情報（単一の 16 進数で表される）が抽出されます。

URL パッシブ永続性では、サーバの IP アドレスポート情報を含むクエリー要素を指定する高度な式を設定する必要があります。クラシックおよび高度なポリシー式の詳細については、「ポリシーと式」を参照してください。

次の式は、文字列「urlp=」を含む URL クエリーの要求を検査し、サーバーの IP アドレスポート情報を抽出し、16 進数の文字列から IP およびポート番号に変換し、要求をこの IP アドレスで構成されたサービスに転送するようにアプライアンスを構成します。ポート番号。

HTTP.REQ.URL.AFTER_STR(“urlp=”)

URL パッシブパーシスタンスが有効で、前の式が設定されている場合、次の URL とサーバの IP アドレスポート文字列を含む要求は 10.102.29. 10:80 に送信されます。

```
http://www.example.com/index.asp?&urlp=0A661D0A0050
```

永続性のタイムアウト値は、この永続タイプには影響しません。サーバーの IP アドレス/ポート情報をクライアント要求から抽出できる限り、永続性は維持されます。このパーシステンスタイプはアプライアンスリソースを消費しないため、永続的なクライアントの数に制限はありません。

URL パッシブパーシスタンスを設定するには、ルールを必要としない永続性のタイプの設定で説明されているように、まずパーシステンスを設定します。パーシステンスタイプを URLPASSIVE に設定します。次に、次の手順を実行します。

CLI を使用して URL パッシブパーシステンスを設定するには

コマンドプロンプトで入力します。

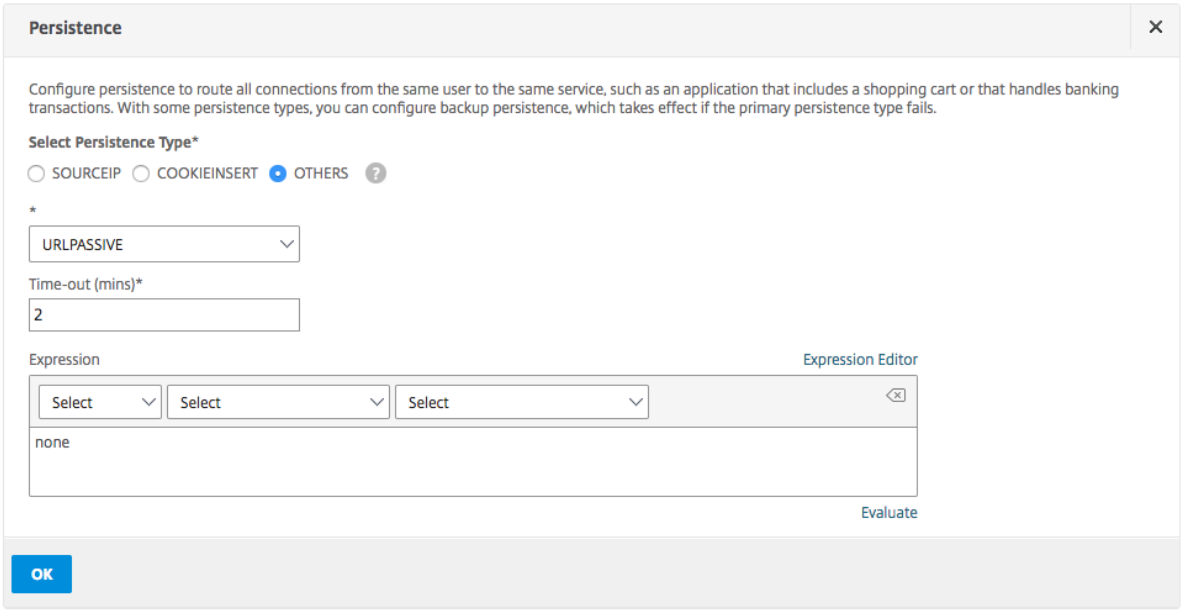
```
set lb vserver <vserverName>  [-persistenceType <persistenceType>]  [-
    rule <expression>]
<!--NeedCopy-->
```

例:

```
set lb vserver LB-VServer-1 -persistenceType URLPASSIVE  -rule HTTP.REQ
    .URL.AFTER_STR( “urlp=” )
<!--NeedCopy-->
```


GUI を使用して仮想サーバーで永続性を構成するには

1. [** トラフィック管理 **] > [** 負荷分散 **] > [** 仮想サーバー **] に移動し、仮想サーバーを開きます。
2. 「持続性」セクションで、要件を満たすパーシステンスタイプを選択します。仮想サーバーに最も適したパーシステンスタイプは、オプションボタンとして使用できます。特定の仮想サーバータイプに適用可能なその他の永続性タイプは、[その他] リストから選択できます。



> 注:
>
> NetScaler リリース 12.0 ビルド 56.20 より前のバージョンでは、オプションボタンがない単一の永続性ドロップダウンリストですべてのパーシステンスタイプを使用できます。

ユーザー定義のルールに基づくパーシステンスの設定

October 7, 2021

ルールベースのパーシステンスを構成すると、Citrix ADC アプライアンスは、一致するルールの内容に基づいてパーシステンスセッションを作成してから、設定された負荷分散方式で選択されたサービスにリクエストを送信します。後で、ルールに一致するすべてのリクエストを同じサービスに送信します。HTTP、SSL、RADIUS、ANY、TCP、および SSL_TCP タイプのサービスに対して、規則ベースの永続性を設定できます。

ルールベースの永続性には、クラシックまたはデフォルトの構文式が必要です。従来の式を使用してリクエストヘッダーを評価したり、デフォルトの構文式を使用してリクエストヘッダー、リクエスト内の Web フォームデータ、レスポンスヘッダー、またはレスポンス本文を評価したりできます。たとえば、従来の式を使用して、HTTP Host ヘッダーの内容に基づいて永続性を設定できます。デフォルトの構文式を使用して、レスポンス Cookie またはカスタムヘッダー内のアプリケーションセッション情報に基づいて永続性を設定することもできます。クラシック構文式とデフォルトの構文式の作成と使用の詳細については、「ポリシーと式」を参照してください。

設定できる式は、ルールベースの永続性を構成するサービスのタイプによって異なります。たとえば、RADIUS 以外のプロトコルでは RADIUS 固有の式を使用できず、ANY タイプ以外のサービスタイプでは TCP オプションベースの式を使用できません。TCP および SSL_TCP サービスタイプでは、TCP/IP プロトコルデータ、レイヤ 2 データ、TCP オプション、および TCP ペイロードを評価する式を使用できます。

注意:TCP 経由で送信される財務情報交換（「FIX」）プロトコルデータに基づくルールベースの永続性を構成するユースケースについては、TCP バイトストリーム内の名前と値のペアに基づくルールベースの永続性の構成を参照してください。

ルールベースの永続性は、Citrix SD-WAN アプライアンス、Citrix SD-WCitrix SD-WAN プラグイン、キャッシュサーバー、アプリケーションサーバーなどのエンティティで永続性を維持するために使用できます。

注意: ANY 仮想サーバーでは、応答に対してルールベースの永続性を設定できません。

ユーザー定義のルールに基づいてパーシステンスを設定するには、「ルールを必要としない永続性のタイプの設定」の説明に従ってパーシステンスを設定し、永続性タイプを RULE に設定します。その後、次の手順を実行できます。設定ユーティリティまたは CLI を使用して、ルールベースの永続性を設定できます。

CLI を使用してユーザー定義ルールに基づいて永続性を構成するには

コマンドプロンプトで入力します。

```
set lb vserver <vserverName> [-rule <expression>][-resRule <expression
    >]
<!--NeedCopy-->
```

例:

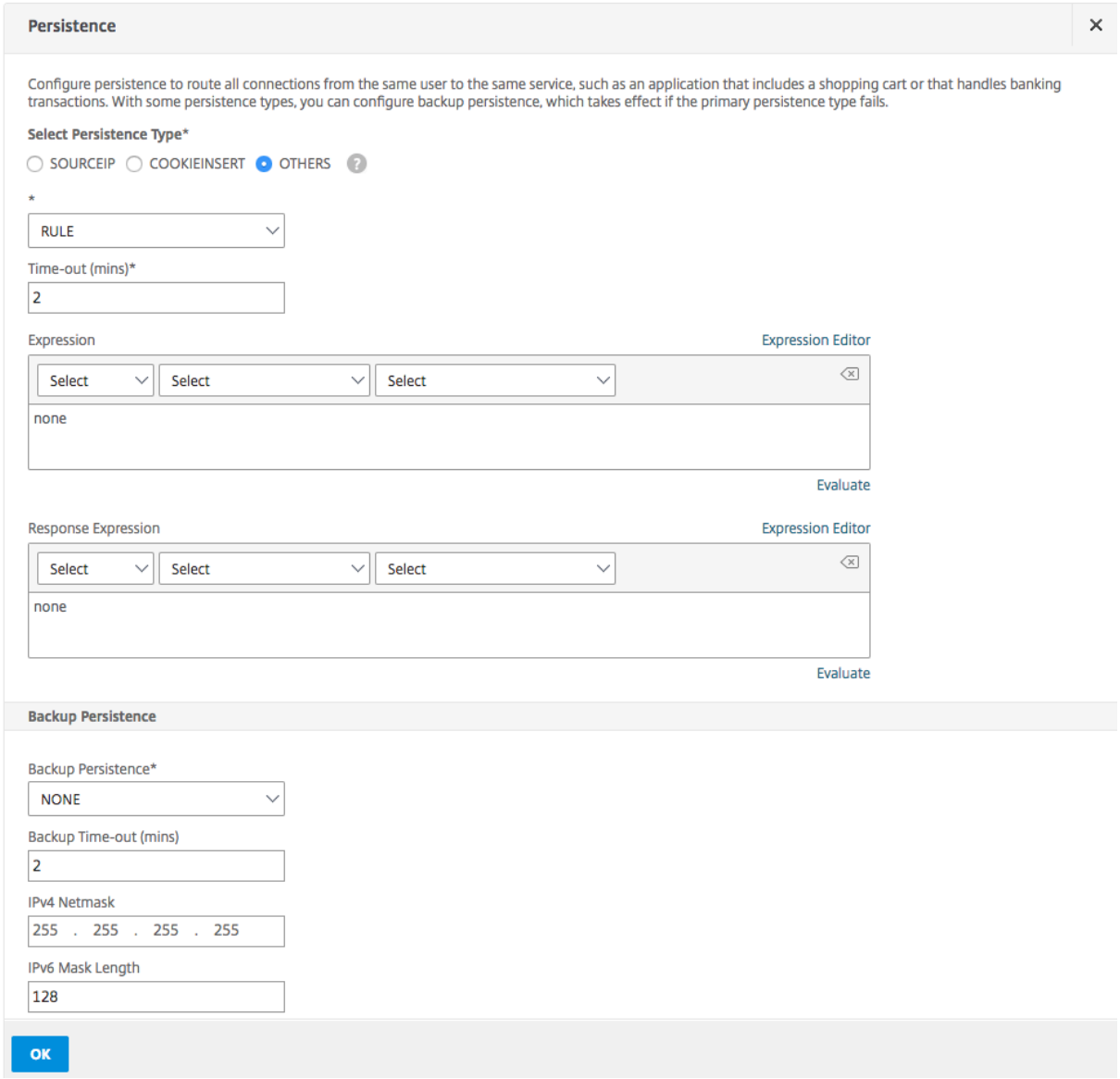
```
set lb vserver vsvr_name  -rule http.req.header(”cookie”).value(0).
    typecast_nvlist_t('=',';').value(”server”)

set lb vserver vsvr_name  -resrule http.res.header(”set-cookie”).value
    (0).typecast_nvlist_t('=',';').value(”server”)

<!--NeedCopy-->
```

GUI を使用してユーザー定義ルールに基づいてパーシステンスを構成するには

1. **Traffic Management** > **Load Balancing** > **Virtual Servers** に移動して仮想サーバーを開きます。
2. 「持続性」セクションで、要件を満たすパーシステンスタイプを選択します。仮想サーバーに最も適したパーシステンスタイプは、オプションボタンとして使用できます。特定の仮想サーバータイプに適用可能なその他の永続性タイプは、[その他] リストから選択できます。



> 注
>
> NetScaler リリース 12.0 ビルド 56.20 より前のバージョンでは、オプションボタンがない単一の永続性ドロップダウンリストですべてのパーシステンスタイプを使用できます。

例: 要求ペイロードのクラシック式

次の古典的な式は、文字列「MyBrowser」を含む User-Agent HTTP ヘッダーの存在に基づいて永続性セッションを作成し、このヘッダーと文字列を含む後続のクライアント要求を、最初の要求用に選択されたのと同じサーバーに送信します。

```
1 http header User-Agent contains MyBrowser
```



```
<!--NeedCopy-->
```

例**:** 要求ヘッダーのデフォルトの構文式

次のデフォルトの構文式は、前のクラシック式と同じ動作を行います。

HTTP.REQ.HEADER(“User-Agent”).CONTAINS (“MyBrowser”)

例**:** レスポンス **Cookie** のデフォルトの構文式

次の式は、「サーバー」クッキーの応答を調べ、そのクッキーを含むすべての要求を、最初の要求で選択されたのと同じサーバーに送信します。

HTTP.RES.HEADER(“SET-COOKIE”).VALUE(0).TYPECAST_NVLIST_T(‘=’,’;’).VALUE(“server”)

規則を必要としないパーシステンスタイプの構成

October 7, 2021

パーシステンスを構成するには、基本的な負荷分散の設定の説明に従って、まず負荷分散仮想サーバーをセットアップする必要があります。次に、仮想サーバーで永続性を構成します。

CLI を使用して仮想サーバーで永続性を構成するには

コマンドプロンプトで次のコマンドを入力して、永続性を構成し、構成を確認します。

```
set lb vserver <name> -PersistenceType <type> [-timeout <integer>]

show lb vserver
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -persistenceType SOURCEIP -timeout 60

show lb vserver
<!--NeedCopy-->
```


「タイムアウト」は、持続性セッションが有効になる期間です。タイムアウトのデフォルト値と最小値（分単位）は、次の表に示すパーシステンスタイプによって異なります。

永続性のタイプ	デフォルト値	最小値	最大値
クッキーの挿入/グループクッキーの挿入	2	0	1440
その他のパーシステンスタイプ	2	2	1440

注

- グループ cookie 挿入パーシステンスタイプは、ロードバランシンググループに設定できます。
- IP ベースの永続性については、persistMask パラメータを設定することもできます。
- デフォルトでは、永続性タイプは NONE に設定されています。

GUI を使用して仮想サーバーで永続性を構成するには

1. **Traffic Management** > **Load Balancing** > **Virtual Servers** に移動して仮想サーバーを開きます。
2.「持続性」セクションで、要件を満たすパーシステンスタイプを選択します。仮想サーバーに最も適したパーシステンスタイプは、オプションボタンとして使用できます。特定の仮想サーバータイプに適用可能なその他の永続性タイプは、[その他] リストから選択できます。

注 Citrix ADC リリース 12.0 ビルド 56.20 より前は、すべての永続性タイプは、オプションボタンなしで単一の永続性ドロップダウンリストで使用できます。

バックアップパーシステンスを構成する

October 7, 2021

プライマリ永続タイプが失敗したときに、ソース IP パーシステンスタイプを使用するように仮想サーバを設定できます。

次の表に、プライマリおよびセカンダリのバックアップパーシステンスタイプの組み合わせ、およびバックアップパーシステンスを使用する場合の条件を示します。

プライマリパーシステンス	バックアップパーシステンス	プライマリパーシステンスルックアップが失敗すると...
クッキーの挿入	接続元 IP	アプライアンスは、クライアントブラウザが要求に cookie を返さない場合にのみ、source-IP ベースの永続性に戻ります。ただし、ブラウザがクッキー（必ずしも永続性クッキーではない）を返す場合、ブラウザはクッキーをサポートしているとみなされるため、バックアップパーシステンスはトリガーされません。
規則	接続元 IP	アプライアンスは、ルールで指定されたパラメータが着信要求に欠落している場合、source-IP ベースの永続性を使用します。

注

- プライマリ永続タイプが HTTP-Cookie ベースの永続性で、バックアップ永続性タイプが Source IP ベースの場合は、バックアップパーシステンスのタイムアウト値を設定できます。手順については、アイドル状態のクライアント接続のタイムアウト値の設定を参照してください。
- プライマリパーシステンスがルールベースの場合はバックアップパーシステンスのタイムアウト値を設定できません。この場合、セカンダリパーシステンスのタイムアウト値はプライマリパーシステンスのタイムアウト値と同じである必要があります。したがって、プライマリとセカンダリは同時に期限切れになります。

コマンドラインインターフェイスを使用して仮想サーバーのバックアップパーシステンスを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -persistenceType <PersistenceType> -
    persistenceBackup <BackupPersistenceType>
<!--NeedCopy-->
```

例:


```
set lb vserver Vserver-LB-1 -persistenceType CookieInsert -
    persistenceBackup SourceIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -
    persistenceBackup SourceIP

set lb vserver Vserver-LB-1  - persistenceType RULE  - rule http.req.
    header("User-Agent").value(0).contains("MyBrowser")  -
    persistenceBackup SOURCEIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -
    persistenceBackup SourceIP
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバのバックアップパーシステンスを設定するには

1. **Traffic Management** > **Load Balancing** > **Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[永続性] を選択し、バックアップパーシステンスのタイプを指定します。

注：プライマリパーシステンスは、「COOKIEINSERT」、「RULE」、または「SSLSESSION」に設定する必要があります。

持続性グループの設定

October 7, 2021

複数の異なる種類の接続 (マルチメディアをホストする Web サーバーなど) を処理する負荷分散サーバーがある場合、これらの接続を処理するように仮想サーバーグループを構成できます。仮想サーバーグループを作成するには、負荷分散サーバーが受け入れる接続の種類ごとに 1 つずつ、異なる種類の仮想サーバーを 1 つのグループにバインドします。次に、グループ全体にパーシステンスタイプを設定します。

永続性グループには、ソース IP ベースの永続性または HTTP cookie ベースの永続性を設定できます。グループ全体に対してパーシステンスを設定した後は、グループ内の個々の仮想サーバーに対してパーシステンスを変更することはできません。グループでパーシステンスを構成し、新しい仮想サーバーをグループに追加すると、新しい仮想サーバーのパーシステンスは、グループのパーシステンス設定と一致するように変更されます。

永続性が仮想サーバのグループに設定されている場合、各クライアント要求を受信するグループ内の仮想サーバに関係なく、初期要求に対して永続性セッションが作成され、後続の要求は初期要求と同じサービスに送信されます。

永続セッションを持つ仮想サーバーを、異なる永続タイプの負荷分散グループに追加すると、古い永続タイプに固有の既存の永続セッションが削除されます。永続セッションは、トラフィックが同じ仮想サーバーに送信される必要が

あるか、別のサーバーに送信される必要があるかを決定します。したがって、既存の確立された接続は影響を受けません。

負荷分散グループの永続性タイプは、仮想サーバーのプロトコルタイプに関係なく、そのグループにバインドされているすべての仮想サーバーに適用されます。負荷分散グループは、次の永続性タイプをサポートします。

- SourceIP
- CookieInsert
- 規則

一部の仮想サーバーは、特定の永続性タイプのみをサポートします。たとえば、タイプの仮想サーバー SSL_BRIDGE LB グループには SourceIP 永続性タイプのみを使用できます。

HTTP クッキーベースの永続性を設定する場合、HTTP クッキーのドメイン属性が設定されます。この設定により、異なる仮想サーバーが異なるパブリックホスト名を持つ場合、クライアントソフトウェアはクライアント要求に HTTP cookie を追加します。CookieInsert パーシステンスタイプの詳細については、「HTTP クッキーに基づく永続性」を参照してください。

コマンドラインインターフェイスを使用して仮想サーバー永続性グループを作成するには

コマンドプロンプトで入力します。

```
bind lb group <vServerGroupName> <vServerName> -persistenceType <
    PersistenceType>
<!--NeedCopy-->
```

例:

```
bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType
    CookieInsert
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバグループを変更するには

1. **Traffic Management > Load Balancing > Persistency Groups** に移動して永続性グループを作成し、このグループの一部となる仮想サーバーを指定します。

コマンドラインインターフェイスを使用して仮想サーバーグループを変更するには

コマンドプロンプトで入力します。


```
set lb group <vServerGroupName> -PersistenceBackup <
    BackupPersistenceType> -persistMask <SubnetMaskAddress>
<!--NeedCopy-->
```

例:

```
set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask
    255.255.255.255
<!--NeedCopy-->
```

仮想サーバ間で永続的なセッションを共有する

October 7, 2021

一部のお客様の環境（テレコムおよび ISP）では、1 台のサーバで制御トラフィックとデータトラフィックの両方を処理します。特定のクライアント IP アドレスに対して、制御トラフィックとデータトラフィックの両方を同じバックエンドサーバーに送信する必要があります。このためには、クライアント認証トラフィックを処理するために 1 つの仮想サーバーが必要で、通常はルールベースの永続性が設定されます。例: Radius.req.avp(8).value.typecast_text_t'。データトラフィックを処理するための 2 番目の仮想サーバー。通常、SourceIP パーシステンスが設定されています。

以前は、永続性エントリは仮想サーバーに対してローカルでした。複数の仮想サーバーに永続性を適用する必要がある場合は、仮想サーバーを負荷分散グループに追加し、グループに共通の永続性タイプを適用する必要がありました。負荷分散グループにバインドされているすべての仮想サーバが、グループに設定された永続性を継承しているため、この要件は達成できません。

仮想サーバー間の永続性共有機能を使用すると、グループ設定から継承するのではなく、グループ内の仮想サーバーが独自の永続性パラメータを使用できるように、負荷分散グループの新しい`useVserverPersistency`パラメータを設定できます。各仮想サーバーで個別のルールベースの永続性を構成できます。

必要に応じて、グループ内の仮想サーバーの 1 つをメイン仮想サーバーとして指定することもできます。仮想サーバーがメイン仮想サーバーとして指定されている場合、その仮想サーバーだけが永続性エントリを作成します。永続エントリは、グループ内のすべての仮想サーバーによって使用されます。メイン仮想サーバーがダウンしている場合、Citrix ADC アプライアンスは永続性エントリを作成しません。

注: 仮想サーバー間でのパーシステンス共有は、ルールベースの永続性メソッドでのみサポートされます。メンバー仮想サーバーで互換性のあるルールベースの永続性パラメータを設定します。

例:

v1 と v2 がロードバランシンググループにバインドされ、v1 が RADIUS タイプの仮想サーバ、v2 が HTTP タイプの仮想サーバであると仮定します。「Radius.req.avp (8) .value.typecast_text_t」永続性は v1 上で構成され、「client.ip.src」は v2 上で構成されています。

トラフィックが RADIUS 仮想サーバ v1 を通過すると、評価されたルール文字列に基づいて永続的なエントリが作成されます。その後、トラフィックが HTTP タイプの仮想サーバー v2 に到達すると、v2 はロードバランシンググループの永続性エントリをチェックし、同じ永続セッションを使用してトラフィックを同じバックエンドサーバーに送信します。

永続セッションの共有の設定

負荷分散グループ内の仮想サーバー間で永続性パラメータを共有するには、まず usevServerPersistency パラメーターを有効にし、グループ内の仮想サーバーの 1 つをメインサーバーとして指定する必要があります。

コマンドラインインターフェイスを使用して **useVserverPersistency** パラメーターを有効にするには

コマンドプロンプトで入力します。

```
set lb group <name> -useVserverPersistency ( ENABLED)
<!--NeedCopy-->
```

例**:**

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

GUI を使用してサーバー永続性を使用可能にするには、次の手順に従います

1. [設定] > [トラフィック管理] > [ロードバランシング] > [持続性グループ**]** に移動します。
2. [**Add**] をクリックして新しいグループを追加するか、既存のグループを選択して [**Edit**] をクリックします。
3. 「仮想サーバーの持続性を使用」を選択します。

コマンドラインインターフェイスを使用して仮想サーバーをメイン仮想サーバーとして指定するには

コマンドプロンプトで入力します。

```
set lb group <name> -useVserverPersistency ( ENABLED ) -masterVserver <
    string>
<!--NeedCopy-->
```

例:


```
set lb group lb_grp1 -useVserverPersistency ENABLED  - masterVserver vs1
<!--NeedCopy-->
```

GUI を使用して仮想サーバをメイン仮想サーバとして指定するには

1. [設定] > [トラフィック管理] > [ロードバランシング] > [持続性グループ] に移動します。
2. [**Add**] をクリックして新しいグループを追加するか、既存のグループを選択して [**Edit**] をクリックします。
3.「仮想サーバーの持続性を使用」を選択します。
4. [仮想サーバー名] ボックスで、[**+**] をクリックして、仮想サーバーをグループに追加します。使用可能な仮想サーバーを選択するか、仮想サーバーを作成できます。
5. 新しいグループを追加する場合は [作成] をクリックし、既存のグループを変更する場合は [閉じる] をクリックします。
6. usevServerPersistency パラメータを有効にしたグループを選択し、[編集] をクリックして、永続性エントリを作成するメインとして仮想サーバーを設定します。
7. [マスター **vServer**] リストから、メイン仮想サーバーとして指定する必要がある仮想サーバーを選択します。

引数

useVserverPersistency

グループ設定から永続性設定を継承するのではなく、グループ内の仮想サーバーが独自の永続性パラメータを使用して永続セッションを作成できるようにします。このパラメータを有効にすると、負荷分散グループに永続性を設定できません。

このパラメーターを無効にすると、グループの仮想サーバーはグループ設定から永続性パラメーターを継承します。

負荷分散グループでこのパラメーターを切り替えると、Citrix ADC アプライアンスは、グループとメンバー仮想サーバーの対応する永続性エントリをすべてフラッシュします。

設定可能な値：ENABLED, DISABLED

デフォルト：DISABLED

例**:**

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

masterVserver

負荷分散グループ内のメイン仮想サーバーとして仮想サーバーを指定します。指定すると、メイン仮想サーバーのみが、グループが使用する永続エントリを作成できます。

メモ: このパラメータは，useVserverPersistency パラメータが有効になっている場合にのみ設定できます。

例**:**

```
1 set lb group lb_grp1 -masterVserver vs1
2 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用した持続セッション共有の設定例

仮想サーバーが作成されます。

```
1 add lb vs vs1 http 10.1.10.11 80 -persistence rule -rule 'client.ip.
    src'
2
3 add lb vs vs2 radius 10.2.2.2 1812 -persistenceType rule -rule '
    Radius.req.avp(8).value.typecast_text_t'
4 <!--NeedCopy-->
```

グループが作成されます。

```
1 add lb group lb_grp1 -persistenceType NONE -useVserverPersistency
    ENABLED
2 <!--NeedCopy-->
```

グループ内の仮想サーバは、メイン仮想サーバとして指定されます。

```
1 set lb group lb_grp1 -masterVserver vs1
2 <!--NeedCopy-->
```

仮想サーバはグループにバインドされます。

```
1 bind lb group lb_grp1 vs1
2 bind lb group lb_grp1 vs2
3 <!--NeedCopy-->
```

詳細については、[基本的な負荷分散の設定および永続性グループの構成を参照してください](/ja-jp/citrix-adc/13/load-balancing/load-balancing-persistence/persistence-groups.html)。

永続性を使用した **RADIUS** 負荷分散の設定

October 7, 2021

今日の複雑なネットワーク環境では、大容量の負荷分散構成と堅牢な認証と承認の調整が必要になることがよくあります。アプリケーションユーザーは、コンシューマグレードの DSL またはケーブル接続、WiFi、さらにはダイヤルアップノードなどのモバイルアクセスポイントを介して VPN に接続できます。これらの接続は、通常、接続中に変更される可能性がある動的 IP を使用します。

Citrix ADC アプライアンスで RADIUS 負荷分散を構成して、RADIUS 認証サーバーへの永続的なクライアント接続をサポートする場合、アプライアンスはセッション ID としてクライアント IP の代わりにユーザーのログオンまたは指定された RADIUS 属性を使用し、ユーザーセッションを同じ RADIUS サーバに送信します。したがって、ユーザーは、クライアント IP または WiFi アクセスポイントが変更されたときに切断されることなく、モバイルアクセスロケーションから VPN にログオンできます。

永続性を持つ RADIUS ロードバランシングを設定するには、まず VPN の RADIUS 認証を設定する必要があります。詳細および手順については、AAA アプリケーショントラフィックの「認証、認可、監査（AAA)」の章を参照してください。また、設定のベースとしてロードバランシング機能またはコンテンツスイッチング機能を選択し、選択した機能が有効になっていることを確認します。どちらの機能でも設定プロセスはほぼ同じです。

次に、2 つのロードバランシングまたは 2 つのコンテンツスイッチ仮想サーバを設定します。1 つは RADIUS 認証トラフィックを処理し、もう 1 つは RADIUS アカウンティングトラフィックを処理します。次に、負荷分散仮想サーバーごとに 1 つずつ、2 つのサービスを構成し、各負荷分散仮想サーバーをそのサービスにバインドします。最後に、負荷分散永続性グループを作成し、永続性タイプを RULE に設定します。

ロードバランシングまたはコンテンツスイッチング機能の有効化

負荷分散機能またはコンテンツスイッチング機能を使用するには、まずその機能が有効になっていることを確認する必要があります。以前に構成されていない新しい Citrix ADC アプライアンスを構成する場合は、これらの機能の両方がすでに有効になっているため、次のセクションに進んでください。Citrix ADC アプライアンスを以前の構成で構成していて、使用する機能が有効になっていることを確認できない場合は、今すぐ実行する必要があります。

- ロードバランシング機能を有効にする手順については、ロードバランシングの有効化を参照してください。
- コンテンツスイッチング機能を有効にする手順については、「コンテンツスイッチングの有効化」を参照してください。

仮想サーバの構成

ロードバランシングまたはコンテンツスイッチング機能を有効にした後、RADIUS 認証をサポートするように 2 つの仮想サーバを設定する必要があります。

- **RADIUS** 認証仮想サーバ。この仮想サーバとその関連サービスは、RADIUS サーバへの認証トラフィックを処理します。認証トラフィックは、保護されたアプリケーションまたは仮想プライベートネットワーク（VPN）

にログオンするユーザーに関連付けられた接続で構成されます。

- **RADIUS** アカウンティング仮想サーバ。この仮想サーバとその関連サービスは、RADIUS サーバへのアカウンティング接続を処理します。アカウンティングトラフィックは、保護されたアプリケーションまたは VPN での認証済みユーザーのアクティビティを追跡する接続で構成されます。

重要：RADIUS パーシステンス構成で使用するには、負荷分散仮想サーバーのペアまたはコンテンツスイッチング仮想サーバーのペアを作成する必要があります。仮想サーバタイプを混在させることはできません。

コマンドラインインターフェイスを使用して負荷分散仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、負荷分散仮想サーバーを作成し、構成を確認します。

```
add lb vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
    <rule>

show lb vserver <name>
<!--NeedCopy-->
```

既存の負荷分散仮想サーバーを構成するには、前述の`add lb virtual server`コマンドを同じ引数を取る`set lb vserver`コマンドに置き換えます。

コマンドラインインターフェイスを使用してコンテンツスイッチ仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、コンテンツスイッチング仮想サーバーを作成し、構成を確認します。

```
add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
    <rule>

show cs vserver <name>
<!--NeedCopy-->
```

既存のコンテンツスイッチング仮想サーバを設定するには、前述の`add cs vserver`コマンドを同じ引数を取る`set cs vserver`コマンドに置き換えます。

例**:**

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

```



```
add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
<!--NeedCopy-->
```

構成ユーティリティを使用して負荷分散またはコンテンツスイッチング仮想サーバーを構成するには

[トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動するか、[トラフィック管理] **>** [コンテンツスイッチング] **>** [仮想サーバー] に移動し、仮想サーバーを設定します。

サービスの構成

仮想サーバを設定したら、作成した仮想サーバごとに 1 つずつ、2 つのサービスを構成する必要があります。

注：これらのサービスは、いったん構成されると、Citrix ADC アプライアンスが RADIUS サーバーの認証およびアカウンティング IP に接続してステータスを監視できるまで、DISABLED 状態になります。手順については、「サービスの設定」を参照してください。

サービスへの仮想サーバーのバインド

サービスを構成したら、作成した各仮想サーバーを適切なサービスにバインドする必要があります。手順については、仮想サーバーへのサービスのバインドを参照してください。

Radius の持続性グループの設定

ロードバランシング仮想サーバを対応するサービスにバインドしたら、永続性をサポートするように RADIUS ロードバランシング構成を設定する必要があります。これを行うには、RADIUS ロードバランシング仮想サーバおよびサービスを含むロードバランシング永続性グループを設定し、ルールベースの永続性を使用するようにロードバランシング永続性グループを構成します。認証とアカウンティングの仮想サーバが異なるため、永続性グループが必要です。また、1 人のユーザの認証とアカウンティングメッセージの両方が同じ RADIUS サーバに到達する必要があるためです。永続性グループを使用すると、両方の仮想サーバーで同じセッションを使用できます。手順については、永続性グループの構成を参照してください。

RADIUS 共有シークレットの設定

リリース 12.0 以降、Citrix ADC アプライアンスは RADIUS 共有シークレットをサポートしています。RADIUS クライアントとサーバは、クライアントとサーバ上で構成された共有シークレットを使用して相互に通信します。RADIUS クライアントとサーバ間のトランザクションは、共有シークレットを使用して認証されます。このシークレットは、RADIUS パケット内の情報の一部を暗号化するためにも使用されます。

RADIUS 共有秘密キー検証シナリオ

RADIUS 共有秘密キーの検証は、次のシナリオで行われます。

- **RADIUS** 共有秘密鍵は、**RADIUS** クライアントと **RADIUS** サーバーの両方に対して構成されます。Citrix ADC アプライアンスは、クライアント側とサーバー側の両方で RADIUS 秘密鍵を使用します。検証が成功すると、アプライアンスは RADIUS メッセージの通過を許可します。それ以外の場合は、RADIUS メッセージがドロップされます。
- **RADIUS** 共有秘密鍵が **RADIUS** クライアントまたは **RADIUS** サーバーに対して構成されていない： Citrix ADC アプライアンスは、RADIUS メッセージをドロップします。これは、radkey が構成されていないノードでは共有秘密鍵の検証を実行できないためです。
- **RADIUS** 共有秘密キーが、**RADIUS** クライアントと **RADIUS** サーバーの両方に対して構成されていません。Citrix ADC アプライアンスは、RADIUS シークレットキーの検証をバイパスし、RADIUS メッセージの通過を許可します。

デフォルトの RADIUS 共有秘密を設定することも、クライアント単位またはサブネット単位で設定することもできます。RADIUS ポリシーが設定されているすべての配置に、RADIUS 共有秘密キーを追加することを推奨します。アプライアンスは、RADIUS パケットの送信元 IP アドレスを使用して、使用する共有シークレットを決定します。RADIUS クライアントとサーバ、および対応する共有シークレットを次のように構成できます。

CLI プロンプトで、次のように入力します。

```
add radiusNode <clientPrefix/Subnet> -radKey <Shared_secret_key>
<!--NeedCopy-->
```

引数

IPaddress

RADIUS クライアントの IP アドレスまたはサブネット（CIDR 形式）。アプライアンスは、着信要求パケットの送信元 IP アドレスを使用して、クライアントの IP アドレスを照合します。クライアント IP アドレスを構成する代わりに、クライアントネットワークアドレスを構成できます。最長のプレフィクスが照合され、着信クライアント要求の共有秘密が識別されます。

Radkey

クライアント、Citrix ADC アプライアンス、およびサーバー間の共有シークレット。最大長:31.

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add service radius_auth_service1 192.168.41.68 RADIUS 1812

add service radius_acct_service1 192.168.41.70 RADIUS 1813

bind lb vserver radius_auth_vs1 radius_auth_service1

bind lb vserver radius_acct_vs1 radius_acct_service[1-3]

add radiusNode 192.168.41.0/24 -radKey  serverkey123

add radiusNode 203.0.113.0/24 -radkey clientkey123
<!--NeedCopy-->
```

共有シークレットは、RADIUS クライアントとサーバの両方に設定する必要があります。コマンドは同じです。サブネットは、共有シークレットがクライアント用かサーバー用かを決定します。

たとえば、指定したサブネットがクライアントサブネットの場合、共有シークレットはクライアント用です。指定されたサブネットがサーバー・サブネット（前の例では 192.168.41.0/24）である場合、共有秘密はサーバー用です。

0.0.0.0/0 のサブネットは、すべてのクライアントとサーバーのデフォルトの共有シークレットであることを意味します。

> 注:
>
> RADIUS 共有シークレットでは、PAP および CHAP 認証方式だけがサポートされています。

持続性セッションの表示

October 7, 2021

グローバルに、または特定の仮想サーバーに対して有効なさまざまな持続性セッションを表示できます。

> 注：Citrix ADC nCore アプライアンスは、パケット処理に複数の CPU コアを使用します。CPU コアは、アプライアンス上のすべてのセッションを所有しています。アプライアンスがセッションが存在しない要求を受信すると、セッションが作成され、コアの 1 つがそのセッションの所有者として指定されます。
>
> そのセッションに属する後続の要求は、常にオーナーコアに到着して処理されるとは限りません。その場合、コア間メッセージングにより、オーナーコアのセッション情報が常に最新のものになります。

ただし、コアが別のコアが所有する永続性セッションに属する要求を受信すると、コア間メッセージングは永続性セッションのタイムアウト値を更新しません。

したがって、オーナコアからのタイムアウト値のみを表示する show lb persistentSessions コマンドの出力では、永続セッションがアクティブのままであっても、永続性セッションのタイムアウト値が 0（ゼロ）に減少することがあります。

コマンドラインインターフェイスを使用して永続セッションを表示するには

コマンドプロンプトで、すべての仮想サーバーに関連する永続セッションを表示するには、次のように入力します。

```
show lb persistentSessions [<vServer>]
<!--NeedCopy-->
```

コマンドプロンプトで、仮想サーバーに関連する永続性セッションを表示するには、次のように入力します。

```
show lb persistentSessions <vServername>
<!--NeedCopy-->
```

例**:**

```
show lb persistentSessions myVserver
<!--NeedCopy-->
```

GUI を使用してパーシステンスセッションを表示するには

[トラフィック管理] > [仮想サーバ永続セッション] に移動します。

持続性セッションのクリア

October 7, 2021

セッションがタイムアウトに失敗した場合は、Citrix ADC アプライアンスからパーシステンスセッションをクリアする必要があります。次のいずれかの操作を実行できます。

- すべての仮想サーバのすべてのセッションを一度にクリアします。
- 特定の仮想サーバのすべてのセッションを一度にクリアします。
- 特定の仮想サーバに関連付けられている特定のセッションをクリアします。

コマンドラインインターフェイスを使用して永続性セッションをクリアするには

コマンド・プロンプトで次のコマンドを入力して、永続セッションをクリアし、構成を確認します。

```
clear lb persistentSessions [<vServer> [-persistenceParam <string>]]

show persistentSessions <vServer>
<!--NeedCopy-->
```

例:

例 1 は、ロードバランシング仮想サーバー lbvip1 のすべての永続性セッションをクリアします。

例 2 は、まず負荷分散仮想サーバー lbvip1 の永続性セッションを表示し、persistence パラメーター xls を使用してセッションをクリアし、セッションがクリアされたことを検証する永続セッションを表示します。

例 **1**:

```
> clear persistentSessions lbvip1
 Done
> show persistentSessions
 Done
>
<!--NeedCopy-->
```

例 **2**:

```
> show persistentSessions lbvip1
Type        SRC-IP    ...    PERSISTENCE-PARAMETER
RULE        0.0.0.0   ...    xls
RULE        0.0.0.0   ...    txt
RULE        0.0.0.0   ...    html
 Done
> clear persistentSessions lbvip1 -persistenceParam xls
 Done
```



```
> show persistentSessions lbvip1
Type         SRC-IP     ...     PERSISTENCE-PARAMETER
RULE         0.0.0.0    ...     txt
RULE         0.0.0.0    ...     html
 Done
>
<!--NeedCopy-->
```

構成ユーティリティを使用して永続性セッションをクリアするには

1. [トラフィック管理] **>** [永続セッションのクリア] に移動します。

オーバーロードされたサービスの永続設定の上書き

October 7, 2021

サービスがロードされたり、使用できない場合には、クライアントへのサービスが低下します。この場合、オーバーロードされたサービスに関連付けられた永続性セッションに含まれる要求を他のサービスに一時的に転送するように、Citrix ADC アプライアンスを構成する必要がある場合があります。つまり、負荷分散仮想サーバー用に構成されている永続性設定を上書きする必要があります。この機能を実現するには、skippersistency パラメータを設定します。この skippersistence パラメータが設定され、仮想サーバーがオーバーロードされたサービスの新しい接続を受信すると、次のことが起こります。

- 仮想サーバーは、サービスがリクエストを受け入れることができる状態に戻るまで、そのサービスに関連付けられている既存の永続性セッションを無視します。
- 他のサービスに関連付けられた持続性セッションは影響を受けません。

この機能は、タイプが ANY または UDP の仮想サーバーでのみ使用できます。

Branch Repeater の負荷分散構成では、負荷モニターを構成し、サービスにバインドする必要もあります。モニタは、サービスの負荷が設定されたしきい値を下回るまで、後続の負荷分散決定からサービスを引き出します。仮想サーバーの負荷モニタの構成の詳細については、「負荷モニタについて」を参照してください。

永続性セッションの一部を形成する要求に対して、次のいずれかのアクションを実行するように仮想サーバーを設定できます。

- 各リクエストを他のサービスのいずれかに送信します。仮想サーバーは負荷分散の決定を行い、負荷分散方式に基づいて各要求を他のサービスに送信します。すべてのサービスが過負荷になると、サービスが使用可能になるまで要求がドロップされます。

  ワイルドカードと IP アドレスベースの仮想サーバの両方で、このオプションがサポートされます。このアクションは、仮想サーバが Branch Repeater アプライアンスまたはファイアウォールの負荷分散を行う展開を含む、すべての展開に適しています。

- 仮想サーバサービス構成をバイパスします。仮想サーバは、負荷分散の決定を行いません。代わりに、リクエスト内の宛先 IP アドレスに基づいて、各リクエストを物理サーバにブリッジするだけです。

  bypass オプションがサポートされるのは、ANY および UDP タイプのワイルドカード仮想サーバだけです。ワイルドカード仮想サーバーには、*IP* とポートの組み合わせがあります。この操作は、仮想サーバーを使用して Branch Repeater アプライアンスまたはファイアウォールのロードバランシングを行う展開に適しています。これらの展開では、Citrix ADC アプライアンスはまずブランチリピーターアプライアンスまたはファイアウォールに要求を転送し、次に処理された応答を物理サーバーに転送します。仮想サーバーは、以下の条件で、宛先 IP アドレスにリクエストを直接送信します。

  - 仮想サーバ（オーバーロードされたサービスのサービス構成）をバイパスするように仮想サーバを構成します。
  - ブランチリピータアプライアンスまたはファイアウォールが過負荷になります。

  仮想サーバは、Branch Repeater アプライアンスまたはファイアウォールが要求を受け入れることができるまで、宛先の IP アドレスに要求を直接送信します。

CLI を使用してオーバーロードされたサービスの永続設定を上書きするには

コマンドプロンプトで次のコマンドを入力して、オーバーロードされたサービスの永続性設定を上書きし、構成を確認します。

```
set lb vserver <name> -skippersistency <skippersistency>

show lb vserver <name>
<!--NeedCopy-->
```

例

```
> set lb vserver mylbvserver -skippersistency ReLb
 Done
> show lb vserver mylbvserver
    mylbvserver (\*:\*) - ANY Type: ADDRESS
       . . .
       . . .
 Skip Persistency: ReLb
       . . .
 Done
>
<!--NeedCopy-->
```


GUI を使用してオーバーロードされたサービスの永続設定を上書きするには

1. トラフィック管理 **>** 負荷分散 **>** 仮想サーバーに移動し、UDP または ANY タイプの仮想サーバーを選択します。
2. [詳細設定] ペインで、[トラフィック設定] を選択し、[スキップパーシステンシー] のタイプを指定します。

トラブルシューティング

October 7, 2021

- **Citrix ADC VPX** アプライアンスからの統計は、アプライアンスがセッション持続性制限に達したことを示しています。その結果、永続性セッションは失敗します。セッション永続性の制限を増やすことは可能ですか**?**

  原因**:** Citrix ADC アプライアンスでは、コアに対する持続性セッションのシステム制限が 250,000 になっています。

  解決策: この問題を解決するには、次のいずれかのタスクを実行できます。

  - 永続性のタイムアウト値を減らす
  - アプライアンスのコア数を増やす

- **Citrix ADC** アプライアンスで **Cookie** 挿入パーシステンスを構成した後、ユーザーはしばらくの間接続が正常に動作すると報告されますが、切断が開始されます。永続性を設定する際にはどのようなベストプラクティスに従うべきですか**?**

  原因: デフォルトでは、Cookie 挿入の永続性のタイムアウト値は 120 秒です。

  解決策: アイドル時間を決定できないアプリケーションの永続性を構成する場合は、Cookie 挿入の永続性タイムアウト値を 0 に設定します。この設定では、接続がタイムアウトしません。

- **Citrix ADC** アプライアンスで **HTTP** 仮想サーバーを構成した後、ユーザーが要求されたコンテンツに対して常に同じサーバーに接続することを確認する必要があるため、**SourceIP** の永続性を構成しました。現在、永続性のタイムアウト値を増やすと、レイテンシーが導入されています。パフォーマンスに影響を与えずにタイムアウト値を増やすにはどうすればよいですか**?**

  解決策: タイムアウト値を 0 に設定して Cookie 挿入永続性を使用することを検討してください。この設定では、アプライアンスがクッキーを期限切れにする時間を指定しないため、長期間持続性設定が有効になります。

- **Citrix ADC** アプライアンスで **Cookie** 挿入パーシステンスを構成した後、同じタイムゾーンのクライアントがコンテンツにアクセスすると、正常に動作します。ただし、別のタイムゾーンのクライアントが接続を試みると、接続がすぐにタイムアウトします。

  原因: 同じタイムゾーンのクライアントが接続を行うと、時間ベースの Cookie 挿入永続性が期待どおりに機能します。ただし、クライアントマシンと Citrix ADC アプライアンスが異なるタイムゾーンにある場合、Cookie は無効です。たとえば、EST タイムゾーンのクライアントが午前 11:00 にクッキーを PST タイムゾ

ーンの Citrix ADC アプライアンスに送信すると、アプライアンスは太平洋標準時午後 2 時に Cookie を受信します。時間の差異の結果、クッキーは有効ではなく、接続がすぐにタイムアウトします。

解決策: Cookie 挿入の永続性のタイムアウト値を 0 に設定します。

- **Citrix ADC** アプライアンスは、**Oracle Weblogic** サーバーなどのアプリケーションサーバーの負荷分散に使用されます。クライアントがこれらのサーバーへの永続的な接続を確実に取得できるように、**SourceIP** 永続性が構成されています。これは、コンピュータから接続が確立されたときに期待どおりに動作します。ただし、シンクライアントがターミナルサーバーを介して接続を試みると、アプライアンスは同じ **IP** アドレス（ターミナルサーバーの **IP** アドレス）から複数のクライアントからの要求を受信します。したがって、すべてのシンクライアントからの接続は同じアプリケーションサーバーに送信されます。クライアントの **IP** アドレスに基づいて、個々のシンクライアントからの要求の永続性を構成することは可能ですか**?**

  原因: Citrix ADC アプライアンスがターミナルサーバーから要求を受信し、要求の送信元 IP アドレスは同じままです。その結果、アプライアンスはシンクライアントから受信した要求を区別できず、シンクライアントからの要求に従ってパーシステンスを提供できません。

  解決策: この問題を回避するには、各シンクライアントの一意のパラメータ値に基づいてルールの永続性を構成します。

- **Citrix ADC** アプライアンスは、**Web** インターフェイスサーバーの負荷分散に使用されます。サーバーにアクセスすると、ユーザーは「状態エラー」エラーメッセージが表示されます。さらに、**Web** インターフェイスサーバーのいずれかがシャットダウンまたは使用できない場合、一部のユーザーには、エラーメッセージが表示されます。

  原因: Web Interface サーバーに永続性がないと、ユーザーがサーバーに接続しようとしたときにエラーメッセージが表示されることがあります。

  解決策： Web Interface サーバーの負荷を分散するときは、Citrix ADC アプライアンスで Cookie 挿入パーシステンスメソッドを指定することをお勧めします。

ADC で生成された Cookie に Cookie 属性を挿入します

October 7, 2021

Web 管理者は、Citrix ADC アプライアンスによって生成された Cookie に他の Cookie 属性を挿入できます。これらの追加の Cookie 属性は、アプリケーションのアクセスパターンに基づいて ADC が生成した Cookie に必要なポリシーを適用するのに役立ちます。

次の機能は、ADC で生成された Cookie を使用して永続性を実現します。

- Cookie の永続性の負荷分散
- 負荷分散グループの Cookie の永続性
- GSLB サイトの永続性

- コンテンツスイッチングクッキーの永続性

次のパラメータを使用して、ADC が生成した Cookie に他の Cookie 属性を挿入できます。

- **literaladcCookieAttribute:** ADC が生成したクッキーに、他のクッキー属性を文字列として追加します。
- **ComputedADCCookieAttribute**： ADC ns 変数を使用して、クライアントまたはサーバーの属性（ユーザーエージェントのバージョンなど）に基づいて、ADC で生成された Cookie に Cookie 属性を条件付きで追加します。

> 注
>
> リテラル ADC クッキー属性と計算された ADC クッキー属性の両方を、負荷分散パラメータまたは単一の負荷分散プロファイルで同時に設定することはできません。

ユースケース：**SameSiteCookie** 属性を構成する

すべての Cookie にはドメインが関連付けられています。Cookie のドメインがユーザーのアドレスバーの Web サイトのドメインと一致する場合、これは同じサイト（またはファーストパーティ）のコンテキストと見なされます。Cookie に関連付けられたドメインが、ユーザーのアドレスバーの Web サイトではなく外部サービスと一致する場合、これはクロスサイト（またはサードパーティ）コンテキストと見なされます。

SameSite 属性は、Cookie をクロスサイトコンテキストで使用するか、同じサイトコンテキストに対してのみ使用できるかをブラウザに示します。また、アプリケーションにクロスサイトコンテキストでアクセスする場合は、HTTPS 接続を介してのみアクセスすることができます。詳細については、RFC6265を参照してください。

2020 年 2 月まで、**SameSite** プロパティは **Citrix** ADC で明示的に設定されていませんでした。ブラウザはデフォルト値として None を採用し、Citrix ADC の展開に影響を与えませんでした。

ただし、Google Chrome 80 などの特定のブラウザのアップグレードでは、Cookie のデフォルトのクロスドメイン動作に変更があります。**SameSite** 属性は、次のいずれかの値に設定できます。Google Chrome のデフォルト値は Lax に設定されています。

- なし**:** ブラウザがセキュアな接続でのみクロスサイトコンテキストで Cookie を使用するように指示します。
- **Lax:** ブラウザが同じサイトコンテキストでのリクエストに Cookie を使用するように指示します。クロスサイトコンテキストでは、GET リクエストなどの安全な HTTP メソッドのみが Cookie を使用できます。
- **Strict**：同じサイトのコンテキストでのみ Cookie を使用します。

Cookie に SameSite 属性がない場合、GoogleChrome は次の機能を引き継ぎます。SameSite=Lax.

> 注
>
> 他のブラウザーの特定のバージョンでは、SameSite 属性のデフォルト値が **None** に設定されることがあります。一部のブラウザバージョンでは、「SameSite = 「なし」は別の方法で扱うことができます。たとえば、次のブラウザは「SameSite」を含む Cookie を拒否します = なし”:

> - Chrome51 から Chrome66 までの Chrome のバージョン（両端を含む）
> - Android の UC ブラウザのバージョン 12.13.2 より前のバージョン

ADC で生成された Cookie を構成する

ADC で生成された cookie 属性を設定するには、次の手順を実行します。

1. 負荷分散仮想サーバーの作成
2. LB パラメーターまたは LB プロファイルを使用して、負荷分散仮想サーバーの ADCCookie 属性を設定します。
3. LB プロファイルを使用する場合は、LB プロファイルを負荷分散仮想サーバーに設定します。
4. 計算された ADCCookie 属性を使用することを選択した場合は、関連する書き換えポリシーを構成します。

> 注
>
> LB プロファイルが LB 仮想サーバーにバインドされている場合、グローバル LB パラメータ設定の代わりにプロファイルパラメータ設定が考慮されます。

ADC が生成する Cookie 属性は、次の方法で設定できます。

- ロードバランシングパラメータの ADC Cookie 属性の設定
- ロードバランシングプロファイルの ADC Cookie 属性の設定

CLI を使用してロードバランシングパラメータで ADC Cookie 属性を設定する

Citrix ADC アプライアンスで構成されたすべてのアプリケーションの ADC 生成 Cookie にポリシーを均一に適用するには、グローバル LB パラメータで ADC Cookie 属性を設定します。

リテラル **ADCCookie** 属性設定を使用すると、ADC で生成された Cookie に Cookie 属性を無条件に挿入できます。

コマンドプロンプトで入力します。

```
set lb parameter -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

例:

```
set lb parameter -LiteralADCCookieAttribute SameSite=None
<!--NeedCopy-->
```

Computed ADC Cookie Attribute 設定を使用すると、クライアントまたはサーバーの属性に基づいて、ADC で生成された Cookie に Cookie 属性を条件付きで挿入できます。

コマンドプロンプトで入力します。


```
set lb parameter -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

例:

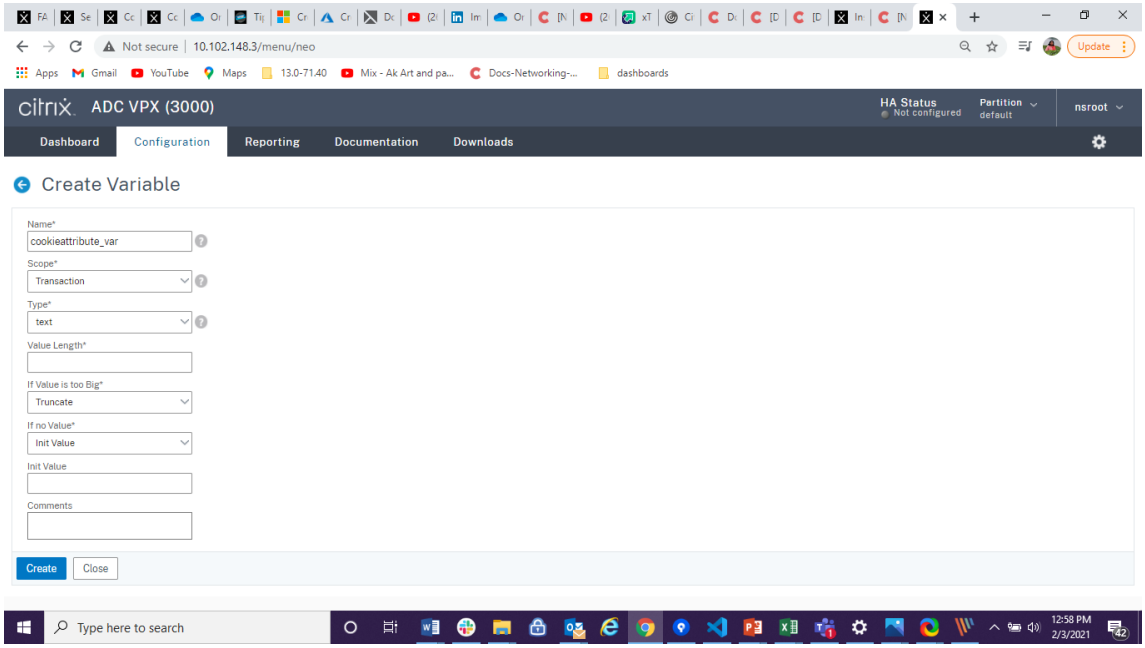
```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

bind rewrite global exception_samesite_attribute 90 110 -type
    RES_OVERRIDE
bind rewrite global append_samesite_attribute 100 110 -type
    RES_OVERRIDE
<!--NeedCopy-->
```

GUI を使用して変数を構成する

1. **AppExpert > Variables** に移動して **Add** をクリックします。
2. [変数の作成] ページで、ドロップダウンメニューから [トランザクションとして スコープ] および [テキストとして タイプ] を選択します。

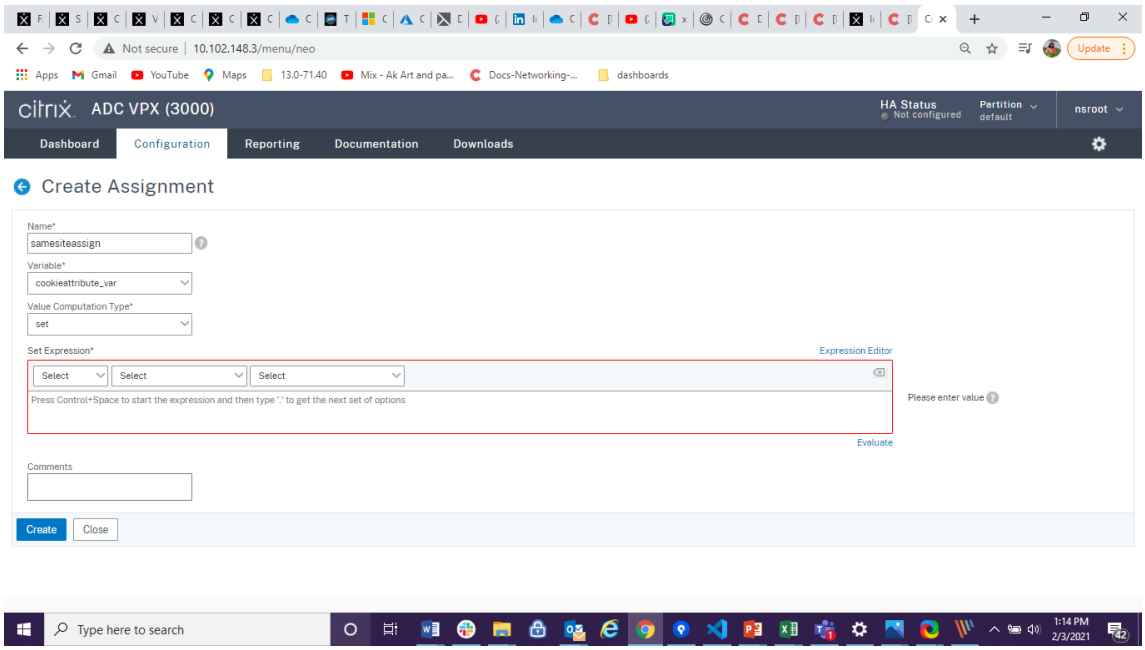


3. その他の詳細を入力し、[作成] をクリックします。

GUI を使用して割り当てを作成する

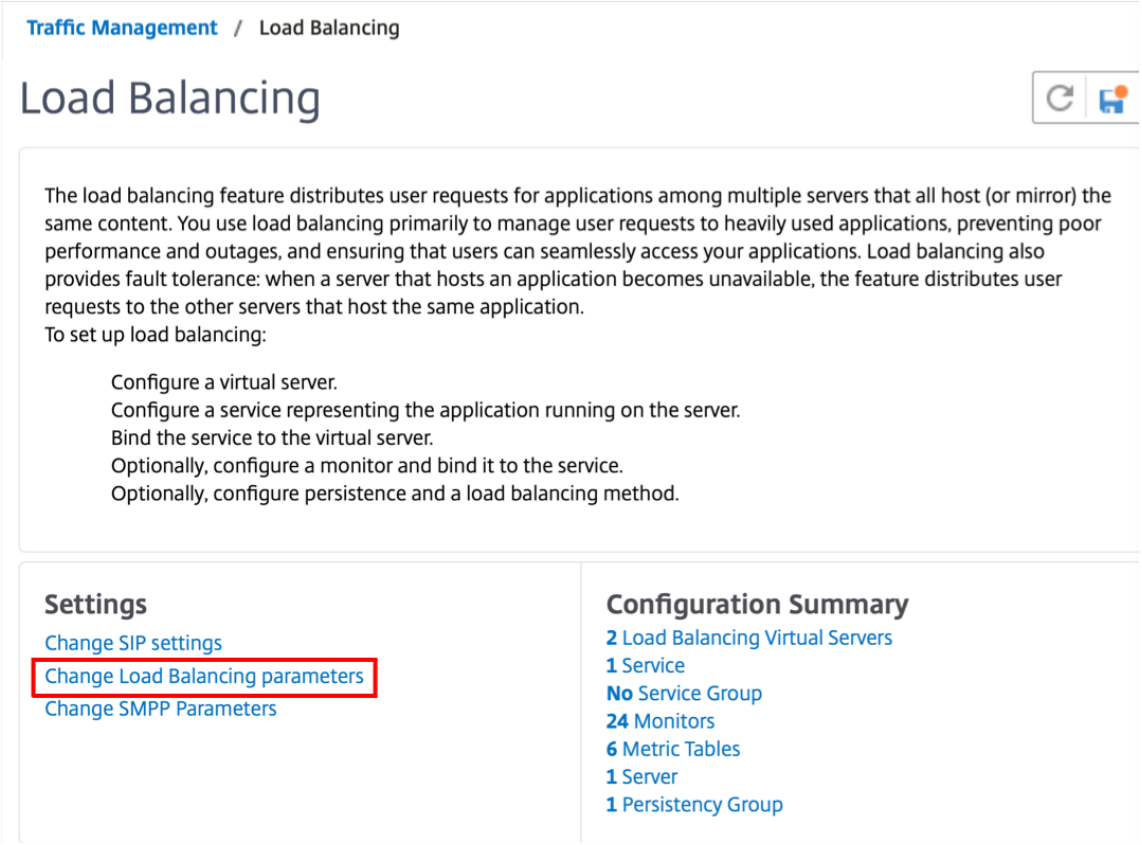
変数を構成した後、割り当てを作成することにより、値を割り当てるか、変数に対して実行する操作を指定できます。

1. **AppExpert > Assignments** に移動して **Add** をクリックします。
2. [割り当ての作成] ページで、詳細を入力し、[作成] をクリックします。



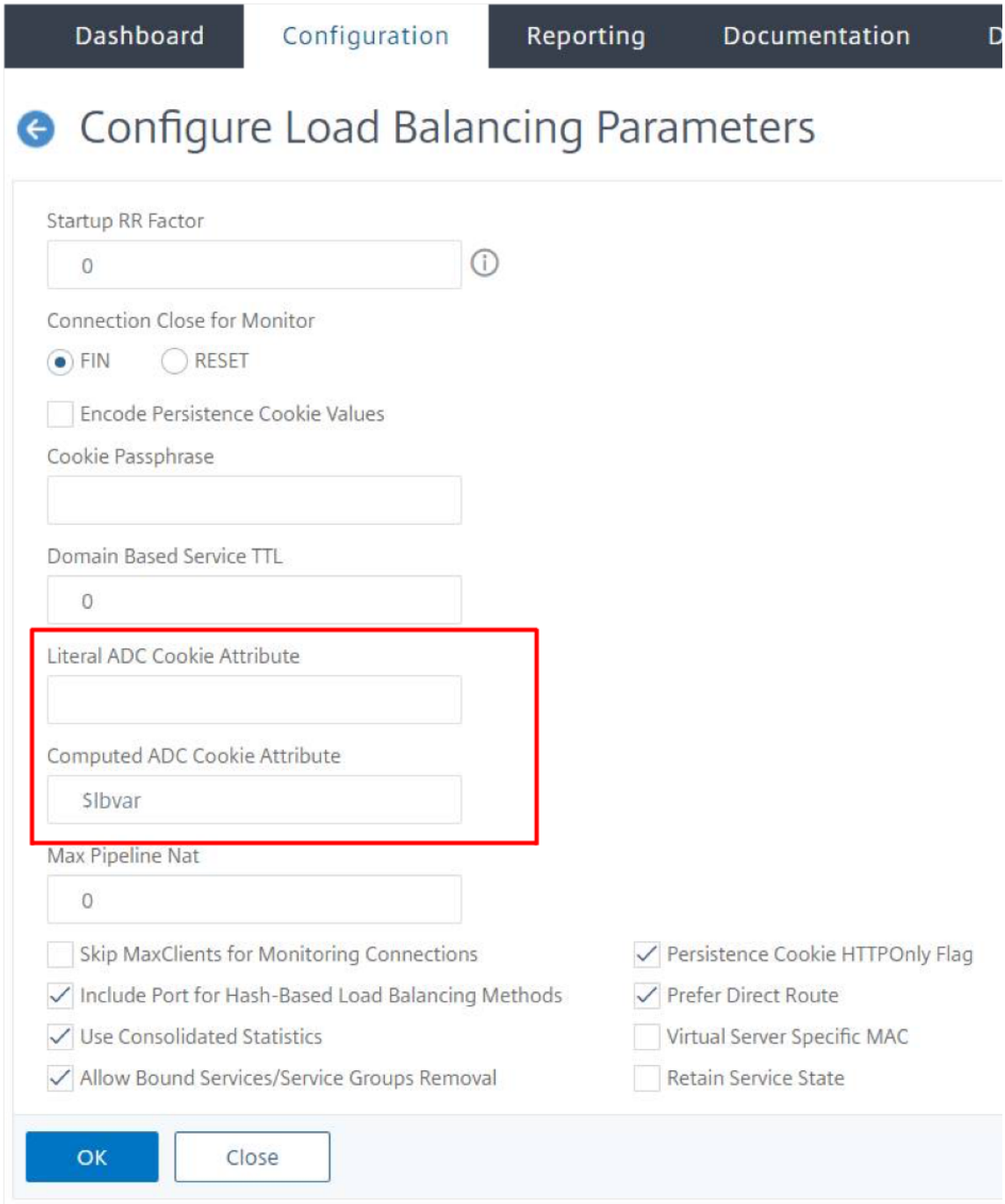
GUI を使用したロードバランシングパラメータでの **ADC Cookie** 属性の設定

1. **[**トラフィック管理**] > [**負荷分散**] > [**負荷分散パラメータの変更**]** に移動します。



2. [負荷分散パラメーターの構成] ペインで、要件に基づいていずれかのフィールドに適切な値を入力します。

   - リテラル **ADCCookie** 属性
   - 計算された **ADCCookie** 属性



3. [**OK**] をクリックします。

CLI を使用したロードバランシングプロファイルでの **ADC Cookie** 属性の設定

Citrix ADC アプライアンスで構成された特定のアプリケーションにポリシーを適用するには、アプリケーション固有の LB 仮想サーバーにバインドされた LB プロファイルの Cookie 属性パラメータを設定できます。

LB プロファイルの リテラル **ADCCookie** 属性設定を使用すると、仮想サーバーに固有の ADC 生成 Cookie に

Cookie 属性を無条件に挿入できます。

コマンドプロンプトで入力します。

```
add lb profile <profile name> -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

例:

```
add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
    =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
<!--NeedCopy-->
```

LB プロファイルの **ComputedADC Cookie Attribute** 設定を使用すると、クライアントまたはサーバーの属性に基づいて、ADC で生成された Cookie に Cookie 属性を条件付きで挿入できます。次に、この LB プロファイルを LB 仮想サーバーに設定します。

コマンドプロンプトで入力します。

```
add lb profile <profile name> -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

例:

```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""
add lb profile LB-Vserver-Profile-1 -ComputedADCCookieAttributE "
    $cookieattribute_var"

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
```

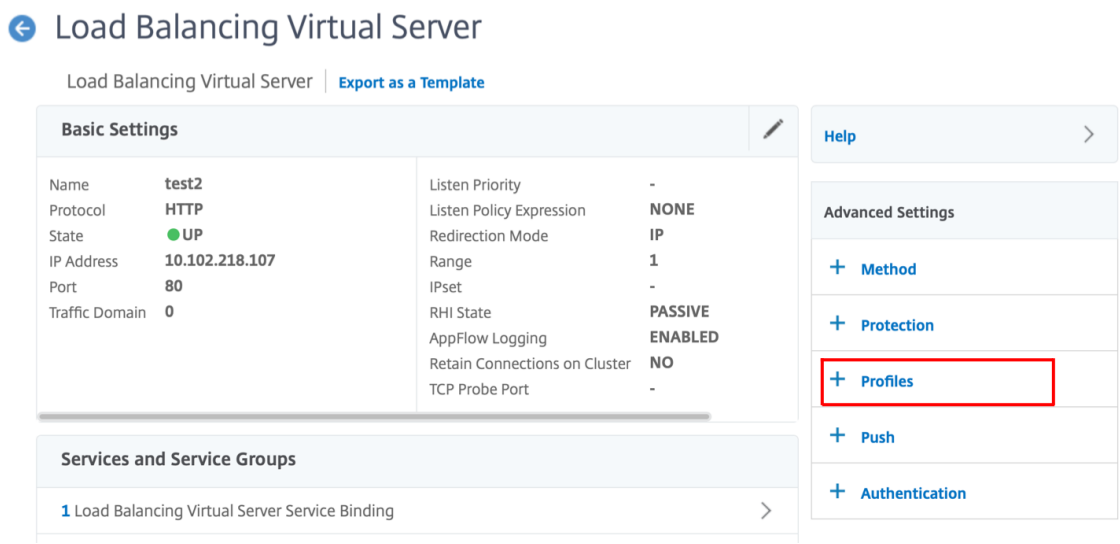


```
add rewrite policy exception_samesite_attribute ”pol_iphone || pol_chrome ” NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
bind lb vserver LB-VServer-1 -policyName exception_samesite_attribute -priority 90 -gotoPriorityExpression 110 -type RESPONSE
bind lb vserver LB-VServer-1 -policyName append_samesite_attribute -priority 100 -gotoPriorityExpression 110 -type RESPONSE
<!--NeedCopy-->
```

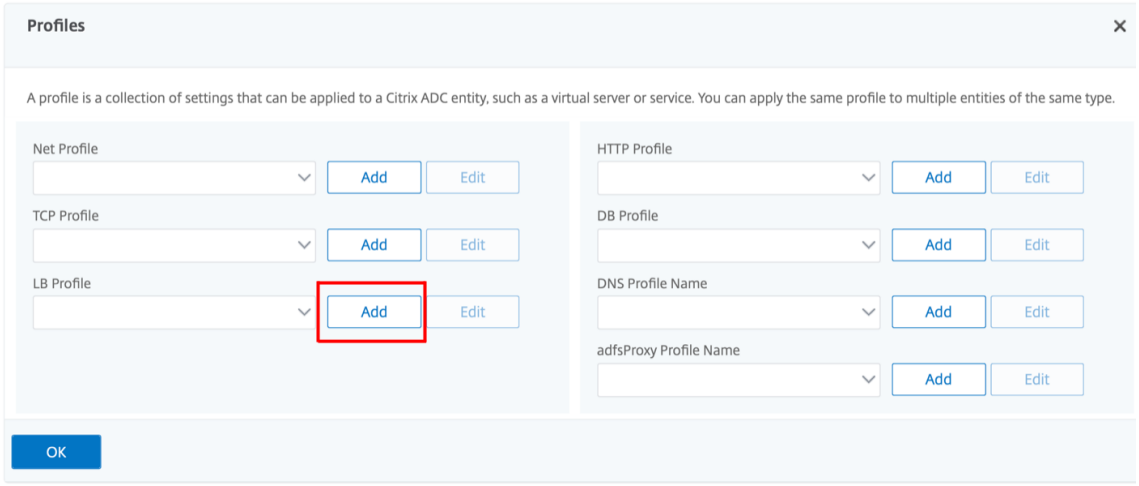
GUI を使用したロードバランシングプロファイルでの **ADC Cookie** 属性の設定

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを選択し、[**Edit**] をクリックします。
3. **[**詳細設定**]** セクションで、[プロファイルの追加] をクリックします。



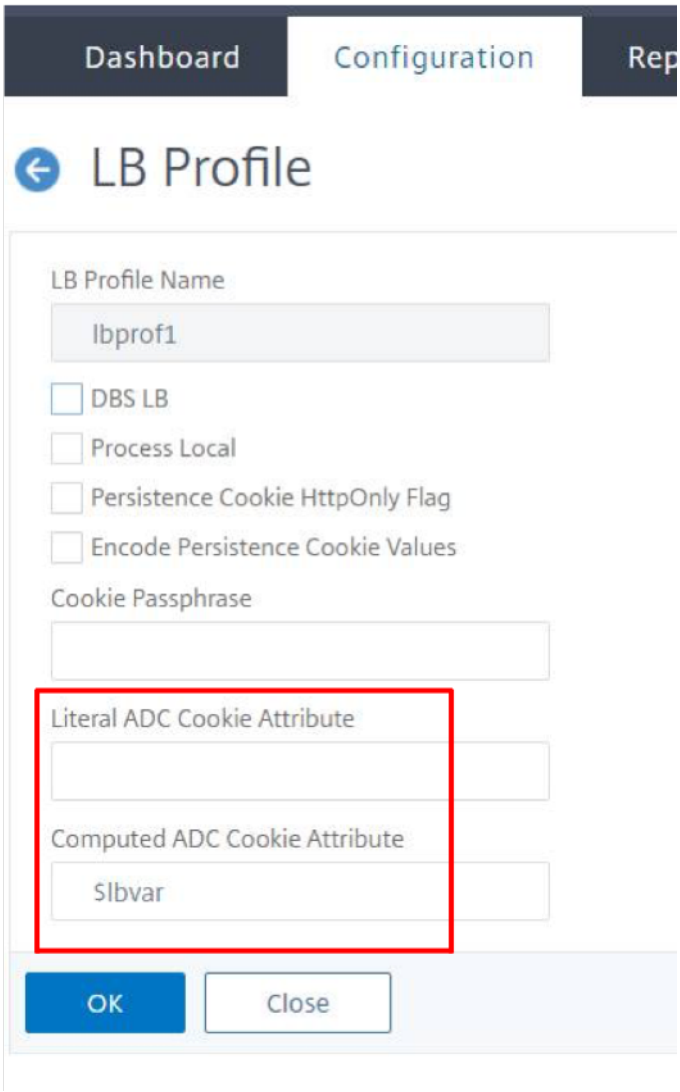
4. [プロファイル] セクションで、[追加] をクリックして LB プロファイルを作成します。

   プロファイルをすでに作成している場合は、[**LB Profile]** ドロップダウンメニューからプロファイルを選択します。



5. [**LB** プロファイル] ペインで、要件に基づいて、いずれかのフィールドに適切な値を入力します。

- リテラル **ADCCookie** 属性
- 計算された **ADCCookie** 属性



1. [**OK**] をクリックします。
2. 作成した LB プロファイルを 手順 **1** で作成した LB 仮想サーバーに設定します。

ns 変数の構成を確認します

ADC ns 変数が LB パラメータまたは LB プロファイルで適切に設定されていることを確認するには、show lbparameter または showlbprofile コマンドを使用します。

次の表に、ns 変数が正しく構成されていない場合のさまざまな警告メッセージとその原因を示します。

警告メッセージ	理由
NS 変数が構成されていません。タイプ text () と変数のスコープトランザクションで構成します	NS 変数はまだ構成されていません。

警告メッセージ	理由
構成された NS 変数のスコープはトランザクションではありません。	変数は構成されていますが、スコープが「トランザクション」に設定されていません。
変数のタイプは Text（）ではありません。	変数は構成されていますが、タイプが「テキスト」に設定されていません。
NS 変数に設定された値の最大サイズが 255 を超えています。	NS 変数に設定された値は 255 文字を超えています。注： ADC で生成された Cookie には、最大 255 文字の長さを追加できます。最大長を超える文字は切り捨てられます。

サンプル出力

次の例では、ns 変数が構成されていない場合に警告メッセージが表示されます。

```
set lb parameter -ComputedADCCookieAttribute ”$lbvar”

Warning: NS Variable is not configured. Please configure it with type
    text() and scope transaction
Done
<!--NeedCopy-->
```

警告メッセージは、次の `show lb parameter` コマンドの出力に表示されます。

```
show lb parameter

Global LB parameters:
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
Use Port For Hash LB: YES
Prefer direct route: YES
Retain Service State: OFF
Start RR Factor: 0
Skip Maxclient for Monitoring: DISABLED
Monitor Connection Close: FIN
Use consolidated stats for LeastConnection: YES
Allow mac mode based vserver to pick thereturn traffic from services:
    DISABLED
Allow bound service removal: ENABLED
```



```
TTL for Domain Based Server: 0 secs

Citrix ADC Cookie Variable Name: $lbvar(NS Variable is not configured.
    Please configure it with type text() and scope transaction)

Done
<!--NeedCopy-->
```

GSLB デプロイメントに Cookie 属性を挿入するためのサンプル構成

次の設定例は、LB 仮想サーバーに対応する GSLB サービスで設定されたサイトの永続性に適用されます。GSLB Cookie にいくつかの追加の Cookie 属性を追加するには、次の設定を実行します。

- LB プロファイル（LB-Vserver-Profile-1）で ADCCookie 属性を設定します。
- たとえば、リテラル ADCCookie 属性値を設定します “SameSite=None”, LB プロファイルで。
- LB プロファイルを、GSLB サービスを表す負荷分散仮想サーバー（LB-VServer-1）に設定します。

```
add gslb vserver GSLB-VServer-1 SSL -backupLBMethod ROUNDROBIN -
    tolerance 0 -appflowLog DISABLED
add gslb site site1 10.102.148.4 -publicIP 10.102.148.4
add gslb service site1_gsvc1 10.102.148.35 SSL 443 -publicIP
    10.102.148.35 -publicPort 443 -maxClient 0 -siteName site1 -
    sitePersistence HTTPRedirect -sitePrefix ss1 -cltTimeout 180 -
    svrTimeout 360 -downStateFlush ENABLED

bind gslb vserver GSLB-VServer-1 -serviceName site1_gsvc1
bind gslb vserver GSLB-VServer-1 -domainName www.gslb.com -TTL 5

add service service-1 10.102.84.140 SSL 443

add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
    =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1

bind lb vserver LB-VServer-1 service-1
<!--NeedCopy-->
```

> 注
>
> 計算された ADCCookie 属性を使用して、条件付きで Cookie 属性を挿入することもできます。

コンテンツスイッチングデプロイメントに **cookie** 属性を挿入するための設定例

次のサンプル構成は、複数のアプリケーションがコンテンツスイッチング仮想サーバーの背後でホストされている場合に適用されます。すべてのアプリケーションに同じポリシーを適用するには、次のように書き換えポリシーを LB 仮想サーバーではなくコンテンツスイッチング仮想サーバーにバインドします。

- LB パラメータで ADCcookie 属性を設定します。

  > 注:
  >
  > LB プロファイルで ADCCookie 属性を設定することもできます。

- ns 変数を構成します (cookieattribute_var) タイプをテキストに設定し、スコープをトランザクションに設定します。
- ns 変数を使用して、グローバル LB パラメーターに計算された ADCCookie 属性を設定します。
- 書き換えポリシーを設定する (exception_samesite_attribute そして append_samesite_attribute) Cookie 属性を挿入するためのコンテンツスイッチング仮想サーバーに。

```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.35 443
add lb vserver LB-VServer-2 SSL 10.102.148.36 443

add cs vserver CS-VServer-1 SSL 10.102.148.42 443 -persistenceType
    COOKIEINSERT

add cs action act1 -targetLBVserver v1
add cs action act2 -targetLBVserver v2
```

```
17 add cs policy CS-policy-1 -rule "HTTP.REQ.URL.CONTAINS("file1.html")" -
      action act1
18 add cs policy CS-policy-2 -rule "HTTP.REQ.URL.CONTAINS("file2.html")" -
      action act2
19
20 bind cs vserver CS-VServer-1 -policyName CS-policy-1 -priority 1
21 bind cs vserver CS-VServer-1 -policyName CS-policy-2 -priority 2
22
23 bind cs vserver -policyname exception_samesite_attribute 90 110 -type
      RES_OVERRIDE
24 bind cs vserver -policyname append_samesite_attribute 100 110 -type
      RES_OVERRIDE
25 <!--NeedCopy-->
```

負荷分散構成のカスタマイズ

October 7, 2021

基本的なロードバランシング設定を構成した後、必要に応じて負荷を分散するように、いくつかの変更を加えることができます。ロードバランシング機能は複雑です。基本要素を変更するには、次のいずれか 1 つまたは複数を実行します。

- ロードバランシングアルゴリズムの変更
- ロードバランシンググループを構成し、それらを使用してロードバランシング構成を作成する
- 永続的なクライアント/サーバー接続の設定
- リダイレクションモードの設定
- キャパシティが異なる異なるサービスに、異なる重みを割り当てる。

Citrix ADC アプライアンスのデフォルトの負荷分散アルゴリズムは、最小の接続方法です。最小接続方式では、アプライアンスは現在最も少ない接続を処理しているサービスに、各着信接続を送信します。異なる負荷分散アルゴリズムを指定できます。各アルゴリズムは、異なる条件に適しています。

ショッピングカートなどのアプリケーションに対応するために、同じユーザーからのすべての要求を同じサーバーに送る必要がある場合は、クライアントとサーバー間の永続的な接続を維持するようにアプライアンスを構成できます。仮想サーバーのグループに対して、パーシステンスを指定することもできます。永続性により、グループ内のどの仮想サーバがクライアント要求を受信するかに関係なく、アプライアンスは個々のクライアント要求を同じサービスに送信できます。

ユーザー要求のリダイレクト時にアプライアンスが使用するリダイレクトモードを有効にして構成し、IP ベースと MAC ベースの転送を選択できます。また、各サービスへの着信負荷の割合を指定して、異なるサービスに重みを割り当てることもできます。重みを割り当てると、同じ負荷分散設定に異なる容量のサーバーを含めることができます。

- 低容量サーバの過負荷や

- 大容量のサーバがアイドル状態になるようにします。

仮想サーバー間での永続性のためのハッシュアルゴリズムのカスタマイズ

October 7, 2021

Citrix ADC アプライアンスは、仮想サーバー間の永続性を維持するために、ハッシュベースのアルゴリズムを使用します。デフォルトでは、ハッシュベースの負荷分散方式では、サービスの IP アドレスとポート番号のハッシュ値が使用されます。同じサーバー上の異なるポートでサービスが使用可能になった場合、アルゴリズムは異なるハッシュ値を生成します。したがって、異なる負荷分散仮想サーバーは、同じアプリケーションに対する要求を異なるサービスに送信し、疑似永続性を破る可能性があります。

ポート番号を使用してハッシュ値を生成する代わりに、サービスごとに一意のハッシュ識別子を指定できます。サービスの場合、すべての仮想サーバで同じハッシュ識別子値を指定する必要があります。物理サーバーが複数のタイプのアプリケーションを提供する場合、各アプリケーションタイプには一意のハッシュ識別子が必要です。

サービスのハッシュ値を計算するアルゴリズムは、次のように動作します。

- デフォルトでは、グローバル設定では、ハッシュ計算でポート番号の使用が指定されます。
- サービスのハッシュ識別子を構成すると、そのハッシュ識別子が使用され、ポート番号はグローバル設定に関係なく使用されません。
- ハッシュ識別子を設定せず、グローバル設定のデフォルト値を変更してポート番号の使用を指定しない場合、ハッシュ値はサービスの IP アドレスだけに基づきます。
- ハッシュ識別子を構成しない場合、またはポート番号を使用するようにグローバル設定のデフォルト値を変更しない場合、ハッシュ値はサービスの IP アドレスとポート番号に基づきます。

CLI を使用してサービスをサービスグループにバインドするときに、ハッシュ ID を指定することもできます。設定ユーティリティでは、サービスグループを開き、[Members] タブでハッシュ識別子を追加できます。

CLI を使用して **use-port-number** グローバル設定を変更するには

コマンドプロンプトで入力します。

lb パラメータの設定-使用ポートハッシュ Lb (はい)　　いいえ)

例:

```
> set lb parameter -usePortForHashLb NO
 Done
>show lb parameter
```



```
Global LB parameters:
        Persistence Cookie HttpOnly Flag: DISABLED
        Use port for hash LB: NO
 Done
<!--NeedCopy-->
```

GUI を使用して use-port-number グローバル設定を変更するには

1. 「トラフィック管理」>「ロードバランシング」>「ロードバランシングパラメータの設定」に移動します。
2. [ハッシュベースの LB メソッドにポートを使用] を選択または選択解除します。

CLI を使用して新しいサービスを作成し、サービスのハッシュ識別子を指定するには

コマンドプロンプトで次のコマンドを入力して、ハッシュ ID を設定し、設定を確認します。

add service < name > (< ip >	< serverName >) < serviceType > < port > -hashId < positive_integer >

```
show service <name>
<!--NeedCopy-->
```

例:

```
> add service flbkng 10.101.10.1 http 80 -hashId 12345
 Done
>show service flbkng
        flbkng (10.101.10.1:80) - HTTP
        State: DOWN
        Last state change was at Thu Nov  4 10:14:52 2010
        Time since last state change: 0 days, 00:00:15.990
        Server Name: 10.101.10.1
        Server ID : 0   Monitor Threshold : 0

        Down state flush: ENABLED
        Hash Id: 12345

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1

```



```
 Done
<!--NeedCopy-->
```

CLI を使用して既存のサービスのハッシュ識別子を指定するには

set service コマンド、サービスの名前、**-hashID** の後に **ID** 値を入力します。

サービスグループメンバーの追加時にハッシュ **ID** を指定するには

グループに追加する各メンバーのハッシュ識別子を指定し、設定を確認するには、コマンドプロンプトで次のコマンドを入力します (各メンバーに一意の hashID を指定してください。):

```
bind servicegroup <serviceGroupName> <memberName> <port> -hashId <
    positive_integer>

show servicegroup <serviceGroupName>
<!--NeedCopy-->
```

例**:**

```
 bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
        SRV - HTTP
        State: ENABLED  Monitor Threshold : 0
        ...

        1)          1.1.1.1:80   State: DOWN     Server Name: 1.1.1.1
              Server ID: 123  Weight: 1
        Hash Id: 32211

                Monitor Name: tcp-default       State: DOWN
        ...

        2)          2.2.2.2:80   State: DOWN     Server Name: 2.2.2.2
              Server ID: 123  Weight: 1
        Hash Id: 12345

                Monitor Name: tcp-default       State: DOWN
        ...
```



```
Done

<!--NeedCopy-->
```

GUI を使用してサービスのハッシュ識別子を指定するには

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. 新しいサービスを作成するか、既存のサービスを開いてハッシュ ID を指定します。

GUI を使用して、すでに設定されているサービスグループメンバーのハッシュ **ID** を指定するには

1. Traffic Management > Load Balancing > Service Groups に移動します。
2. メンバーを開き、一意のハッシュ ID を入力します。

リダイレクションモードを構成する

October 7, 2021

リダイレクションモードは、着信トラフィックの転送先を決定するために仮想サーバが使用する方式を設定します。Citrix ADC アプライアンスでは、次のリダイレクトモードがサポートされています。

- IP ベースの転送（デフォルト）
- MAC ベースの転送

Direct Server Return（DSR; ダイレクトサーバリターン）トポロジ、リンクロードバランシング、またはファイアウォールロードバランシングを使用するネットワークで MAC ベースの転送を設定できます。MAC ベースの転送の詳細については、「MAC ベースの転送の設定」を参照してください。

CLI を使用してリダイレクションモードを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -m <RedirectionMode>
<!--NeedCopy-->
```

例**:**


```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

> 注
>
> -m MAC オプションが有効になっている仮想サーバにバインドされたサービスの場合は、非ユーザーモニタをバインドする必要があります。

GUI を使用してリダイレクションモードを構成するには

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. 仮想サーバーを開き、リダイレクトモードを選択します。

VLAN 単位のワイルドカード仮想サーバの設定

October 7, 2021

特定の Virtual Local Area Network（VLAN; 仮想ローカルエリアネットワーク）上のトラフィックに対してロードバランシングを設定する場合は、指定された VLAN 上のトラフィックだけを処理するように制限するリッスンポリシーを持つワイルドカード仮想サーバを作成できます。

CLI を使用して特定の **VLAN** をリッスンするワイルドカード仮想サーバを設定するには

コマンドプロンプトで次のコマンドを入力して、特定の VLAN をリッスンするワイルドカード仮想サーバーを構成し、構成を確認します。

```
add lb vserver <name> <serviceType> IPAddress * Port * -listenpolicy <
    expression> [-listenpriority <positive_integer>]

show vserver
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-vlan1 ANY -listenpolicy ”CLIENT.VLAN.ID.EQ(2)
    ” -listenpriority 10

show vserver Vserver-LB-vlan1
<!--NeedCopy-->
```


GUI を使用して特定の **VLAN** をリッスンするワイルドカード仮想サーバを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 新しい仮想サーバーを作成するか、既存の仮想サーバーを開きます。
3. リッスンポリシーの優先順位と式を指定します。

この仮想サーバーを作成したら、「基本負荷分散の設定」の説明に従って、仮想サーバーを 1 つ以上のサービスにバインドします。

サービスへの重みの割り当て

October 7, 2021

ロードバランシング設定では、各サービスに送信するトラフィックの割合を示すためにサービスに重みを割り当てます。重み付けが大きいサービスはより多くの要求を処理できます。重み付けが小さいサービスでは、処理する要求が少なくなります。サービスに重みを割り当てると、Citrix ADC アプライアンスは、各負荷分散サーバーが処理できるトラフィックの量を判断できるため、負荷をより効果的に分散できます。

注：サービスの重み付けをサポートする負荷分散方式（ラウンドロビン方式など）を使用する場合は、サービスに重みを割り当てることができます。

次の表では、重み付けをサポートする負荷分散方式について説明し、各サービスの選択方法に重み付けがどのように影響するかを簡単に説明します。

負荷分散方法	重み付きサービスの選択
ラウンドロビン	仮想サーバは、使用可能なサービスのキューに優先順位を付けます。重みが最も高いサービスが、重みが最も小さいサービスよりも頻繁にキューの前面に配置され、比例してトラフィックを受信します。完全な説明については、ラウンドロビン方式を参照してください。
最小接続	仮想サーバは、アクティブなトランザクションが最も少なくて重みが最も高い組み合わせでサービスを選択します。完全な説明については、「最小接続方式」を参照してください。
モニタを用いた最小応答時間・最小応答時間手法	仮想サーバは、最も少ないアクティブなトランザクションと最速の平均応答時間の最適な組み合わせでサービスを選択します。詳細な説明については、「最小応答時間の方法」を参照してください。

負荷分散方法	重み付きサービスの選択
最小帯域幅	仮想サーバは、最小トラフィックと最大帯域幅の最適な組み合わせでサービスを選択します。詳細な説明については、「最小帯域幅方式」を参照してください。
最小パケット	仮想サーバは、最も少ないパケットと最も高い重みの最適な組み合わせでサービスを選択します。詳細な説明については、「最小パケット方式」を参照してください。
カスタムロード	仮想サーバは、負荷の最小化と重みの最適な組み合わせでサービスを選択します。詳細な説明については、「カスタムロードメソッド」を参照してください。
ハッシュメソッドとトークンメソッド	これらのロードバランシング方式では、重み付けはサポートされません。

CLI を使用してサービスに重みを割り当てるように仮想サーバを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -weight <Value> <ServiceName>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスに重みを割り当てるように仮想サーバを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを開き、[サービス] セクションをクリックします。
3. サービスの重み列で、サービスに重みを割り当てます。

MySQL および Microsoft の SQL サーバーのバージョン設定を構成します

October 7, 2021

MSSQL および MySQL タイプの負荷分散仮想サーバーに対して、Microsoft® SQL Server® のバージョンと MySQL サーバーのバージョンを指定できます。一部のクライアントが MySQL または Microsoft SQL Server 製品と同じバージョンを実行していないことが予想される場合は、バージョン設定をお勧めします。バージョン設定では、すべての通信がサーバーのバージョンに準拠していることを確認することで、クライアント側接続とサーバー側の接続間の互換性を提供します。

CLI を使用して Microsoft SQL サーバーのバージョンパラメーターを設定するには

コマンドプロンプトで次のコマンドを入力して、負荷分散仮想サーバーの Microsoft SQL Server バージョンパラメーターを設定し、構成を確認します。

```
set lb vserver <name> -mssqlServerVersion <mssqlServerVersion>

show lb vserver <name>
<!--NeedCopy-->
```

例

```
> set lb vserver myMSSQLvip -mssqlServerVersion 2008R2
 Done
> show lb vserver myMSSQLvip
    myMSSQLvip (190.0.2.12:1433) - MSSQL    Type: ADDRESS
 . . .
 . . .
    MSsql Server Version: 2008R2
    . . .
    . . .
 Done
>
<!--NeedCopy-->
```

CLI を使用して MySQL サーバーのバージョンパラメーターを設定するには

コマンドプロンプトで次のコマンドを入力して、負荷分散仮想サーバーの MySQL Server バージョンパラメータを設定し、設定を確認します。


```
set lb vserver <name> -mysqlServerVersion <string>

show lb vserver <name>
<!--NeedCopy-->
```

例

```
> set lb vserver mysqlsvr -mysqlserverversion 5.5.30
 Done
> sh lb vserver mysqlsvr
      mysqlsvr (2.22.2.222:3306) - MYSQL      Type: ADDRESS
  . . .
  . . .
      Mysql Server Version: 5.5.30
     . . .
     . . .
 Done
>
<!--NeedCopy-->
```

GUI を使用して **MySQL** サーバーまたは **SQL** サーバーのバージョンパラメーターを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. MySQL または MSSQL タイプの仮想サーバーを開き、サーバーのバージョンを設定します。

マルチ **IP** 仮想サーバー

June 1, 2022

Citrix ADC は、VIP タイプの複数の非連続/連続 IPv4 および IPv6 アドレスを持つ単一の負荷分散仮想サーバーの作成をサポートしています。仮想サーバーにバインドされた各 VIP アドレスは、個別の仮想サーバーとして扱われます。これらの仮想サーバーには、同じプロトコルとその他の仮想サーバーレベルの設定があります。複数の VIP アドレスを持つ仮想サーバーは、マルチ IP 仮想サーバーとも呼ばれます。

マルチ IP 仮想サーバーを使用する利点は次のとおりです。

- マルチ IP 仮想サーバーは、同じ設定とサービスバインディングを持つ多数の仮想サーバーを作成する作業を軽減します。
- マルチ IP 仮想サーバーは、仮想サーバーエンティティの上限に達する可能性を効果的に減らします。

- 1 つのマルチ IP 仮想サーバーを異なるサブネットのクライアントに使用して、同じサーバーセットに接続できます。
- IPv6 クライアントと IPv4 クライアントが同じサーバーセットに接続するために使用できるマルチ IP 仮想サーバーは 1 つだけです。

マルチ **IP** 仮想サーバーを構成する

マルチ IP 仮想サーバーの構成は、次のタスクで構成されます。

- IPSet を作成し、複数の IP アドレスをバインドします。
- IPSet を負荷分散仮想サーバーにバインドします。

IPSet の設定に関連する次の点に注意してください。

- IPSet には次のものが含まれます。
  - 非連続/連続 IPv4 アドレスと IPv6 アドレス
  - IPv4 アドレスと IPv6 アドレスの組み合わせ
- IPset を使用して仮想サーバーに関連付けるすべての IPv4/IPv6 アドレスは、VIP タイプである必要があります。
- 1 つの IP セットを複数の仮想サーバーにバインドできます。
- IPv4/IPv6 アドレスは、仮想サーバーへの既存の IPset バインディングに関係なく、IPset にバインド/バインド解除できます。
- 新しい IPSet をバインドする前に、仮想サーバーへの IPSet バインディングの設定を解除する必要があります。

CLI を使用して **IP** セットを追加し、複数の **VIP** アドレスをバインドするには

コマンドプロンプトで入力します。

```
add ipset <name>

bind ipset <name> <IPaddress1 … >

bind ipset <name> <IPaddress2… >

show ipset <name>
<!--NeedCopy-->
```

CLI を使用して **IPset** を仮想サーバーにバインドするには

コマンドプロンプトで入力します。


```
set lb vserver <name> -ipset <ipset name>

show lb vserver <name>
<!--NeedCopy-->
```

GUI を使用して **IPset** を追加し、複数の **VIP** アドレスをバインドするには

[システム] > [ネットワーク] > [**IP** セット**]** に移動し、複数の VIP アドレスを持つ IP セットを作成します。

GUI を使用して **IPset** を仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、作成した IPset をバインドする仮想サーバーを開きます。
2.「基本設定」で、「**IPset**」パラメータを作成した IPset の名前に設定します。

```
> add ipset IPSET-1


Done

> bind ipset IPSET-1 9.9.9.10


Done

> bind ipset IPSET-1 1000::20


Done

> add lb vserver LBVS-1 HTTP 8.8.8.10 80  - ipset IPSET-1


Done

> add service SVC-1 3.3.3.10 HTTP 80


Done

```



```
> add service SVC-2 3.3.3.100 HTTP 80


Done

> bind lb vserver LBVS-1 SVC-1


Done

> bind lb vserver LBVS-1 SVC-2


Done
```

マルチ **IP** 仮想サーバーの **GSLB** サポート

フローティング IP アドレスは、高可用性展開に必要です。クラウド展開は Floating IP アドレスをサポートしていません。そのため、IP セット機能は、クラウド展開での高可用性をサポートするのに役立ちます。IP セット機能を使用すると、プライベート IP アドレスをプライマリインスタンスとセカンダリインスタンスのそれぞれに関連付けることができます。仮想サーバーの作成時に、プライベート IP アドレスの 1 つが追加されます。もう 1 つの IP アドレスは IP セットにバインドされます。次に、IP セットが仮想サーバーに関連付けられます。通常、パブリック IP アドレスは、どのアプライアンスがトラフィックを受信しているかに基づいて、プライベート IP アドレスの 1 つにマッピングされます。フェールオーバー中、このマッピングは動的に変化し、トラフィックを新しいプライマリにルーティングします。

GSLB 展開では、GSLB サービスは仮想サーバーを表し、仮想サーバーのプライベート IP アドレスとパブリック IP アドレスの両方が必要です。クラウド展開では、IP セットとして表される複数のプライベート IP アドレスがありますが、GSLB サービスは 1 つのプライベート IP アドレスのみを受け入れることができます。そのため、GSLB サービスを構成する際には、仮想サーバーを追加するときに構成された IP アドレス、または IP セット内の IP アドレスの 1 つを指定することをお勧めします。GSLB サービスで IP セット機能を設定する必要はありません。GSLB サービスに関連付けられている負荷分散仮想サーバーで構成されている IP セットで十分です。

GSLB 親子トポロジでは、子サイトの負荷分散仮想サーバーに IP セットを関連付けることができます。このトポロジに対応する GSLB サービスは、パブリック IP アドレスと 1 つのプライベート IP アドレスを保持します。プライベート IP アドレスは、IP セットの IP アドレス、または子サイトに仮想サーバーを追加するときに構成された IP アドレスです。親サイトと子サイト間の通信は、常にパブリック IP アドレスと GSLB サービスのパブリックポートを使用します。

また、IP セットのサポートにより、IPv4 トラフィックと IPv6 トラフィックの両方に対して単一の仮想サーバーエンドポイントを持つことができます。以前は、IPv4 トラフィックと IPv6 トラフィック用に異なる仮想サーバーを構成する必要がありました。IP セットのサポートにより、IPv4 と IPv6 の IP アドレスを同じ IP セットに関連付けることができます。IPv4 エンドポイントと IPv6 エンドポイントを表すさまざまな GSLB サービスを追加できます。

クライアント接続での同時要求の数を制限する

October 7, 2021

1 つのクライアント接続での同時要求の数を制限できます。同時要求の数を制限することで、セキュリティ上の脆弱性からサーバーを保護できます。クライアント接続が指定された上限値に達すると、Citrix ADC アプライアンスは、未処理の要求数が制限値を下回るまで、その接続に対する後続の要求をドロップします。

maxPipelineNat パラメーターを構成して、1 つのクライアント接続での同時要求の数を制限できます。このパラメータは、次のサービスタイプにのみ適用できます。また、「svrTimeout」がゼロに設定されている場合にも適用されます。

- ANY
- DNS を除くすべての UDP サービスタイプ

maxPipelineNat パラメータのデフォルト値は 255 です。値をゼロ（0）にすると、同時要求の数に制限はありません。制限が設定されていない場合、Citrix ADC アプライアンスはすべての要求を実行します。

> 注
>
> MaxpipelineNAT を高い値に設定すると、スプーフィング攻撃の可能性が高くなります。したがって、MaxpipelineNAT を小さい値に設定することをお勧めします。

CLI を使用してクライアントの同時接続数を制限するには

コマンドプロンプトで入力します。

```
set lb parameter -maxPipelineNat <positive_integer>
<!--NeedCopy-->
```

例:

```
set lb parameter -maxPipelineNat 199
<!--NeedCopy-->
```

GUI を使用してクライアントの同時接続数を制限するには

[トラフィック管理] > [ロードバランシング] > [ロードバランシングパラメータの設定] に移動し、[最大パイプライン NAT 要求] の値を指定します。

Diameter の負荷分散を設定する

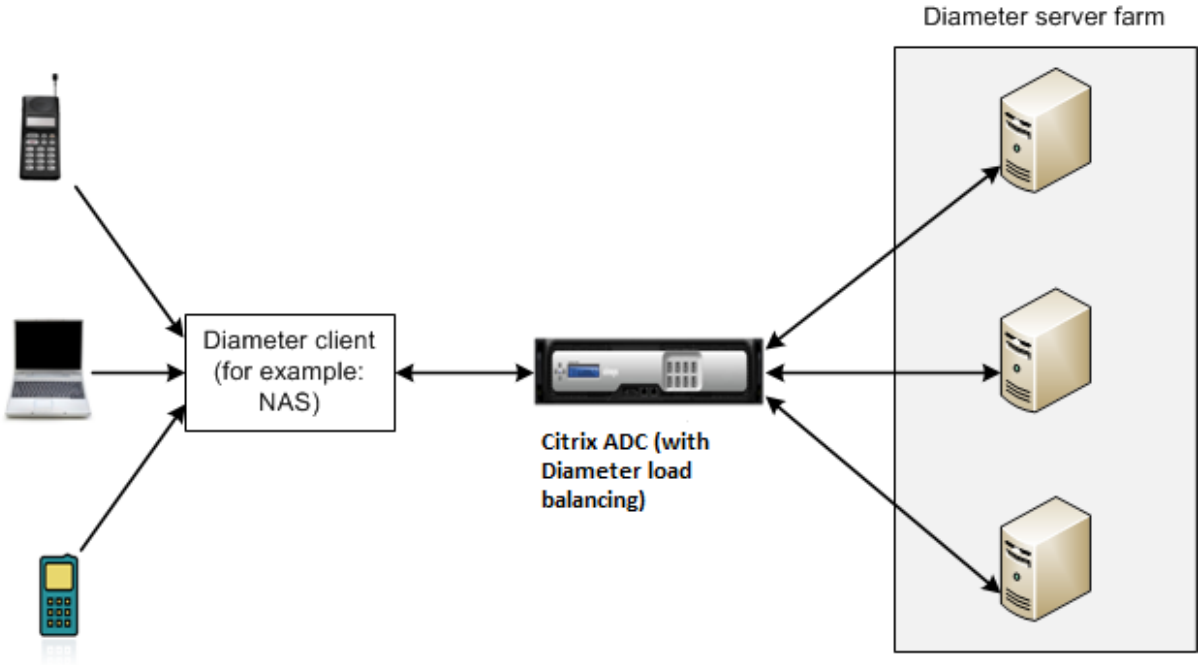
October 7, 2021

Diameter プロトコルは、主にラップトップや携帯電話などのモバイルデバイスで使用される次世代の認証、認可、アカウンティング（AAA）シグナリングプロトコルです。これは、他のほとんどのプロトコルで使用される従来のクライアント/サーバーモデルとは対照的に、ピアツーピアプロトコルです。ただし、ほとんどの Diameter 展開では、クライアントは要求を発信し、サーバーは要求に応答します。

Diameter メッセージが交換されると、Diameter サーバーは通常 Diameter クライアントよりも多くの処理を行います。コントロールプレーンシグナリングボリュームが増加すると、Diameter サーバがボトルネックになります。したがって、Diameter メッセージは複数のサーバーに負荷分散する必要があります。Diameter メッセージのロードバランシングを実行する仮想サーバには、次の利点があります。

- Diameter サーバーの負荷が軽くなり、エンドユーザーへの応答時間が短縮されます。
- サーバの健全性監視とフェイルオーバー機能の向上。
- クライアント構成を変更することなく、サーバーの追加に関するスケーラビリティが向上します。
- 高可用性。
- SSL Diameter オフロード。

次の図は、Citrix ADC 展開環境での Diameter システムを示しています。



Diameter システムには、次のコンポーネントがあります。

- **Diameter** クライアント。基本プロトコルに加えて Diameter クライアントアプリケーションをサポートします。Diameter クライアントは、多くの場合、ネットワークのエッジにあるデバイスに実装され、そのネッ

トワークにアクセスコントロールサービスを提供します。Diameter クライアントの典型的な例は、ネットワークアクセスサーバ（NAS）とモバイル IP 外部エージェント（FA）です。

- **Diameter** エージェント。リレー、プロキシ、リダイレクト、または翻訳サービスを提供します。Citrix ADC アプライアンス（Diameter 負荷分散仮想サーバーで構成されている）は、Diameter エージェントの役割を果たします。
- **Diameter** サーバー。特定のレルムの認証、認可、アカウンティング要求を処理します。Diameter サーバーは、基本プロトコルに加えて Diameter サーバーアプリケーションをサポートする必要があります。

典型的な Diameter トポロジでは、エンドユーザデバイス（携帯電話など）がサービスを必要とする場合、Diameter クライアントに要求を送信します。各 Diameter クライアントは、Diameter ベースプロトコル RFC 6733 で指定された Diameter サーバーとの単一の接続 (TCP 接続 — SCTP はまだサポートされていません) を確立します。接続は長期間有効であり、2 つの Diameter ノード (クライアントとサーバー) 間のすべてのメッセージがこの接続を介して交換されます。Citrix ADC は、メッセージベースの負荷分散を使用します。

例**:**

モバイルサービスプロバイダは、その課金システムのための Diameter を使用しています。ユーザがプリペイド番号を使用する場合、Diameter クライアントは、使用可能な残高を確認するために、サーバに要求を繰り返し送信します。Diameter プロトコルは、クライアントとサーバー間の接続を確立し、すべての要求はその接続を介して交換されます。接続は 1 つしかないため、接続ベースのロードバランシングは無意味です。ただし、接続上に多数のメッセージがあると、メッセージベースのロードバランシングにより、プリペイドモバイルサブスクライバへの課金プロセスが迅速になります。

Diameter 負荷分散の仕組み

Disconnect Peer Request（DPR; 切断ピア要求）は、接続を閉じる理由とともに、ピアが接続を閉じる意図を示します。ピアは DPA で応答します（TCP は常に正常な DPA を提供します）。

- アプライアンスは、クライアントから DPR を受信すると、DPR をすべてのサーバーにブロードキャストし、ただちに DPA でクライアントに応答します。サーバは DPA で応答しますが、アプライアンスは DPA を無視します。クライアントは FIN を送信し、アプライアンスがすべてのサーバーにブロードキャストします。
- アプライアンスは、サーバから DPR を受信すると、そのサーバだけに DPA を返信し、再使用プールからサーバを削除しません。サーバーが FIN を送信すると、アプライアンスは FIN/ACK で応答し、再利用プールから接続を削除します。
- アプライアンスがクライアントから FIN を受信すると、クライアントに FIN/ACK を送信し、FIN をブロードキャストして、再使用プールからサーバー接続をただちに削除します。
- アプライアンスがサーバーから FIN を受信すると、FIN/ACK が送信され、再使用プールから削除されます。このサーバーに対する新しいメッセージは、新しい接続で送信されます。

Diameter トラフィックの負荷分散

クライアントが Citrix ADC アプライアンスに要求を送信すると、アプライアンスは要求を解析し、永続 AVP に基づいてコンテキスト的に Diameter サーバーに負荷分散します。アプライアンスはクライアント ID をサーバにアドバ

タイズしているため、サーバはクライアントからのメッセージを直接受信するため、ルートエントリを追加しません。

サーバが開始する要求は、クライアント要求ほど頻繁ではありません。サーバが開始する要求は、クライアントが開始する要求に似ていますが、次の点を除きます。

- メッセージは複数のサーバーから受信されるため、アプライアンスは転送された各要求メッセージに一意の Hop by Hop（HbyH）番号を追加して、トランザクションの状態を維持します。メッセージ応答が（同じ HbyH 番号で）到着すると、アプライアンスはこの HbyH 番号を、要求が到着したときにサーバで受信された HbyH 番号に変換します。
- Citrix ADC アプライアンスは、クライアントはアプライアンスをリレーエージェントとして認識するため、アイデンティティを入力してルートエントリを追加します。

注：Diameter メッセージが複数のパケットにまたがる場合、アプライアンスはパケットを不完全なヘッダーキューに蓄積し、メッセージ全体が蓄積されるとサーバーに転送します。同様に、1 つのパケットに複数の Diameter メッセージが含まれている場合、アプライアンスはパケットを分割し、ロードバランシング仮想サーバによって決定されたとおりにメッセージをサーバに転送します。

セッションの切断

Disconnect Peer Request（DPR; 切断ピア要求）は、接続を閉じる理由とともに、ピアが接続を閉じる意図を示します。ピアは DPA で応答します（TCP は常に正常な DPA を提供します）。

- Citrix ADC アプライアンスは、クライアントから DPR を受信すると、DPR をすべてのサーバーにブロードキャストし、ただちに DPA でクライアントに応答します。サーバは DPA で応答しますが、アプライアンスは DPA を無視します。クライアントは FIN を送信し、アプライアンスがすべてのサーバーにブロードキャストします。
- アプライアンスは、サーバから DPR を受信すると、そのサーバだけに DPA を返信し、再使用プールからサーバを削除しません。サーバーが FIN を送信すると、アプライアンスは FIN/ACK で応答し、再利用プールから接続を削除します。
- アプライアンスがクライアントから FIN を受信すると、クライアントに FIN/ACK を送信し、FIN をブロードキャストして、再使用プールからサーバー接続をただちに削除します。
- アプライアンスがサーバーから FIN を受信すると、FIN/ACK が送信され、再使用プールから削除されます。このサーバーに対する新しいメッセージは、新しい接続で送信されます。

Diameter トラフィックのロードバランシングの構成

Diameter トラフィックを負荷分散するように Citrix ADC アプライアンスを構成するには、まずアプライアンスで Diameter パラメータを設定し、Diameter モニターを追加し、Diameter サービスを追加し、サービスをモニターにバインドし、Diameter 負荷分散仮想サーバーを追加し、サービスを仮想サーバーにインストールします。

コマンドラインインターフェイスを使用して **diameter** トラフィックのロードバランシングを構成するには

Diameter パラメータを設定します。


```
set ns diameter -identity <string> -realm <string> -
   serverClosePropagation <YES|NO>
<!--NeedCopy-->
```

例:

```
set ns diameter -identity mydomain.org -realm org -
   serverClosePropagation YES
<!--NeedCopy-->
```

Diameter モニタを追加します。

```
add lb monitor <monitorName> DIAMETER -originHost <string> -originRealm
    <string>
<!--NeedCopy-->
```

例:

```
add lb monitor diameter_mon DIAMETER -originHost mydomain.org -
   originRealm org
<!--NeedCopy-->
```

Diameter サービスを作成します。

```
add service <name> <IP> DIAMETER <port>
<!--NeedCopy-->
```

例:

```
add service diameter_svc0 10.102.82.86 DIAMETER 3868

add service diameter_svc1 10.102.82.87 DIAMETER 3868

add service diameter_svc2 10.102.82.88 DIAMETER 3868

add service diameter_svc3 10.102.82.89 DIAMETER 3868
```



```
<!--NeedCopy-->
```

Diameter サービスを Diameter モニタにバインドします。

```
bind service <name>@ monitorName <monitorName>
<!--NeedCopy-->
```

例:

```
bind service diameter_svc0 -monitorName diameter_mon

bind service diameter_svc1 -monitorName diameter_mon

bind service diameter_svc2 -monitorName diameter_mon

bind service diameter_svc3 -monitorName diameter_mon
<!--NeedCopy-->
```

Diameter 永続性を持つ Diameter 負荷分散仮想サーバーを追加します。

```
add lb vserver <name> DIAMETER <IPAddress> <port> -persistenceType
   DIAMETER -persistAVPno <positive_integer>
<!--NeedCopy-->
```

例:

```
add lb vserver diameter_vs DIAMETER 10.102.112.152 3868 -
   persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

Diameter サービスを Diameter ロードバランシング仮想サーバにバインドします。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:


```
bind lb vserver diameter_vs diameter_svc0

bind lb vserver diameter_vs diameter_svc1

bind lb vserver diameter_vs diameter_svc2

bind lb vserver diameter_vs diameter_svc3
<!--NeedCopy-->
```

構成を保存します。

```
save ns config
<!--NeedCopy-->
```

注: **SSL_DIAMETER** サービスタイプを使用して、SSL を介した Diameter トラフィックの負荷分散を構成することもできます。

構成ユーティリティを使用して **Diameter** トラフィックの負荷分散を構成するには

1. [システム] **>** [設定] **>** [**Diameter** パラメータを変更] に移動し、Diameter パラメータを設定します。
2. トラフィック管理 > 負荷分散 > 仮想サーバーに移動し、Diameter タイプの負荷分散仮想サーバーを作成します。
3. Diameter タイプのサービスを作成します。
4. Diameter タイプのモニタを作成します。「特殊パラメータ」で、オリジナル・ホストとオリジナル・レルムを設定します。
5. モニタをサービスにバインドし、Diameter 仮想サーバにサービスをバインドします。
6. [詳細設定] で、[持続性] をクリックし、直径を指定し、持続性 AVP 番号を入力します。
7. [保存] をクリックし、[完了] をクリックします。

FIX 負荷分散を構成する

October 7, 2021

金融情報交換（FIX）プロトコルは、取引相手国間の証券取引に関連する情報の電子交換のために、金融業界で使用されるオープンメッセージ標準です。FIX/SSL_FIX プロトコルは、買い側と売り側の企業、取引プラットフォーム、および貿易情報を伝達するための規制当局によって広く使用されています。

この機能を使用すると、着信 FIX メッセージを配布し、FIX メッセージングでセキュリティを提供するために、FIX または SSL_FIX ロードバランシング仮想サーバーを構成できます。Citrix ADC は、FIX 4.1、FIX 4.2、FIX 4.3、お

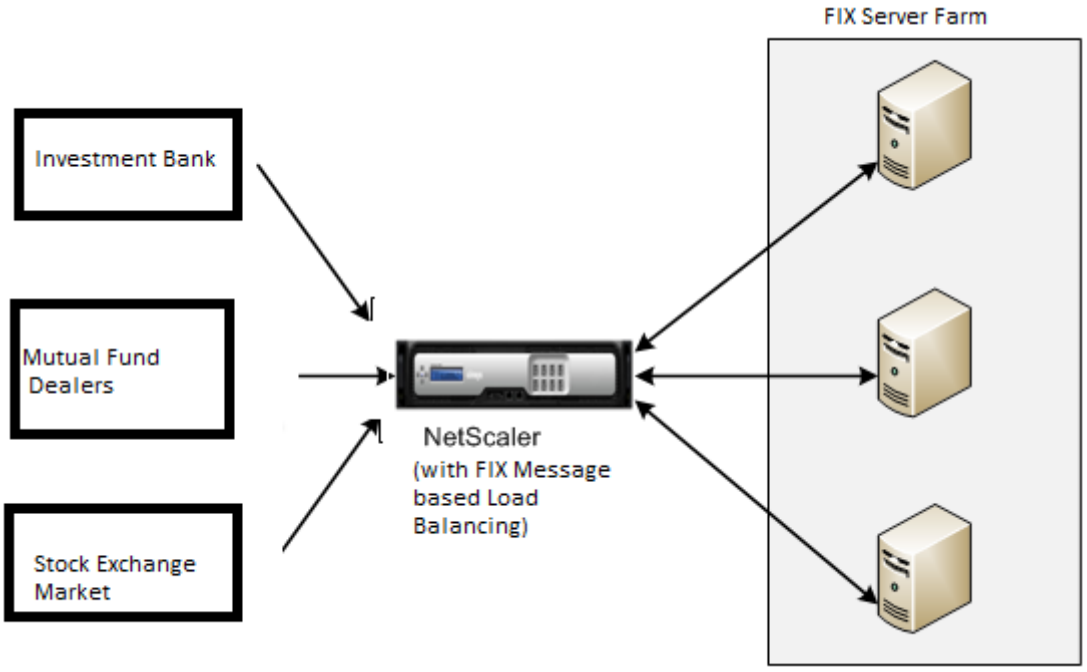
よび FIX 4.4 バージョンの FIX メッセージベースの負荷分散（MBLB）をサポートしています。

Citrix ADC アプライアンスの FIX MBLB には、次のような利点があります。

1. 高可用性と健全性の監視により、FIX または SSL_FIX サーバーの効率的な管理。
2. すべての FIX サーバーまたは SSL_FIX サーバーに SYN 保護を適用します。
3. FIX セッションの永続性。

FIX ロードバランシングのしくみ

FIX MBLB セットアップには、FIX ロードバランシング仮想サーバーと複数のロードバランシングされた FIX サーバーが含まれます。FIX 仮想サーバーは、着信クライアントトラフィックを受信し、着信トラフィックを FIX メッセージに解析し、各 FIX メッセージに対して FIX サーバーを選択し、選択した FIX サーバーにメッセージを転送します。次の概念図は、標準的な FIX 負荷分散の設定を示しています。



基本的な FIX MBLB セットアップでは、FIX 仮想サーバーは、ラウンドロビン負荷分散方式を使用して、クライアントからの FIX メッセージを負荷分散された FIX サーバーに配信します。タイプ FIXSESSION の永続性が有効になっていると、FIX 仮想サーバーは、同じ FIX セッションに属する異なる FIX メッセージに対して同じサーバーを選択します。FIX セッションは、**FIX** フィールド senderCompid (タグ 49) と targetCompid (タグ 56) の値に基づいて決定されます。

FIX トラフィックの負荷分散を構成および監視する

FIX メッセージトラフィックのロードバランシングを行うために必要な設定は次のとおりです。

1. FIX 負荷分散仮想サーバーの構成

2. SSL_FIX ロードバランシング仮想サーバーの構成
3. FIX 負荷分散サービスの構成
4. SSL_FIX ロード・バランシング・サービスの構成
5. FIXSESSION 永続性の設定
6. 持続性タイムアウトの設定
7. FIX/SSL_FIX 統計情報の表示
8. FIX/SSL_FIX 永続セッションの監視

コマンドラインインターフェイスを使用して **FIX** 負荷分散サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> FIX <IP> <PORT>
<!--NeedCopy-->
```

例

```
add lb vserver vs1 FIX 10.102.82.86 3868
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SSL_FIX** ロードバランシング仮想サーバーを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name> SSL_FIX <IP> <PORT>
<!--NeedCopy-->
```

例

```
add lb vserver vs1 SSL_FIX 10.102.82.86 3868
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **FIX** サービスを構成するには

コマンドプロンプトで入力します。


```
add service <name> <ip-addr> FIX <port>
<!--NeedCopy-->
```

例

```
add service_svc1 10.102.82.86 FIX 3868
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **SSL_FIX** サービスを構成するには

コマンドプロンプトで入力します。

```
add service <name> <ip-addr> SSL_FIX <port>
<!--NeedCopy-->
```

例

```
add service svc1 10.102.82.86 SSL_FIX 3868
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **FIXSESSION** 永続性を構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -persistenceType FIXSESSION
<!--NeedCopy-->
```

例

```
set lb vserver vs1 -persistenceType FIXSESSION
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して永続性タイムアウトを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -timeout <value>
<!--NeedCopy-->
```

例

```
set lb vserver vs1  - timeout 2
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **FIX** 統計を表示するには

コマンドプロンプトで入力します。

```
stat lb vserver <name>
<!--NeedCopy-->
```

例

```
stat lb vserver_svc1
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **FIX** サービスを **FIX** 仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name> <service name>
<!--NeedCopy-->
```

例

```
bind lb vserver vs1 svc1
<!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **FIX** 永続セッションを表示するには

コマンドプロンプトで入力します。

```
show lb persistentSessions <name>
<!--NeedCopy-->
```

例

```
show lb persistentSessions vs1
<!--NeedCopy-->
```

> 注
>
> 注:SSL_FIX サービスタイプを使用して、SSL を介した FIX トラフィックの負荷分散を構成できるようになりました。このサービスは、FIX メッセージのセキュリティで保護された通信を提供します。

GUI を使用して FIX 負荷分散仮想サーバーを構成するには

1. [構成] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、[追加] をクリックして FIX 負荷分散仮想サーバーを作成します。
2. 「負荷分散仮想サーバー」ページで、サーバー・パラメータを設定します。
   a) 仮想サーバー名
   b) 「FIX」としてのプロトコルの種類
   c) サーバーの IP アドレスの種類
   d) サーバー IP アドレス
   e) サーバのポート番号
3. 「**OK**」および「続行」をクリックして、他のパラメータを設定します。
4. [サービス] セクションで、新しい FIX 負荷分散仮想サービスを選択または追加し、FIX サーバーにバインドします。
5. 「持続性」セクションで、次のパラメータを設定します。
   a) パーシステンスタイプ 'FIXSESSION'
   b) タイムアウト間隔
6. [**OK]**、[完了] の順にクリックします。

GUI を使用して FIX 負荷分散仮想サーバーを編集するには

[構成] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、FIX サーバーを選択して [編集] をクリックします。

GUI を使用して **FIX** 負荷分散仮想サーバーを削除するには

[構成] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、FIX サーバーを選択して [削除] をクリックします。

GUI を使用して **FIX** 負荷分散仮想サービスを構成するには

1. [構成] > [トラフィック管理] > [負荷分散] > [サービス] ページに移動し、[追加] をクリックして FIX 負荷分散仮想サービスを作成します。
2. [サービス] ページで、次のパラメータを設定します。[詳細] 矢印をクリックして、トラフィックドメイン、ハッシュ ID、サーバ ID、キャッシュタイプ、アクティブ接続数などの他のパラメータを設定できます。
   a) サービス名 — FIX 仮想サービス名
   b) 仮想サーバーの種類を（新規または既存）として選択
   c) プロトコル：プロトコルタイプ「FIX」
   d) サーバ：仮想サーバの IP アドレス
   e) ポート：サーバのポート番号
3. [**OK]** をクリックし、**[**続行**]** をクリックして、[モニター]、[しきい値とタイムアウト]、[プロファイル]、[ポリシー] などのその他のパラメータを設定します。
4. [**OK]**、[完了**]** の順にクリックします。

GUI を使用して **FIX** 負荷分散仮想サービスを編集するには

[構成] > [トラフィック管理] > [負荷分散] > [サービス] ページに移動し、**FIX** サービスを選択し、[編集] をクリックします。

GUI を使用して **FIX** 負荷分散仮想サービスを削除するには

[構成] > [トラフィック管理] > [負荷分散] > [サービス] ページに移動し、FIX サービスを選択し、[削除] をクリックします。

FIX 負荷分散サーバーの統計情報を表示するには

[構成] > [トラフィック管理] > [負荷分散] > [仮想サーバー] ページに移動し、[統計] をクリックして FIX サーバーの統計情報を表示します。

GUI を使用して **FIX** サーバーの永続セッションを表示するには

[設定] > [トラフィック管理] ページに移動し、[セッションの監視**]** で **[**仮想サーバの永続セッション **]** をクリックします。

GUI を使用して **FIX** サーバーの永続的なセッションをクリアするには

1. [設定] > [トラフィック管理] ページに移動し、[セッションの監視] で [永続セッションのクリア] をクリックします。
2. [永続セッションのクリア] ページで、次のパラメータを設定します。
   a) 仮想サーバー — FIX 仮想サーバーの選択
   b) 永続性パラメータ — FIX 永続性パラメータの選択
3. [**OK**] をクリックします。

MQTT 負荷分散

October 7, 2021

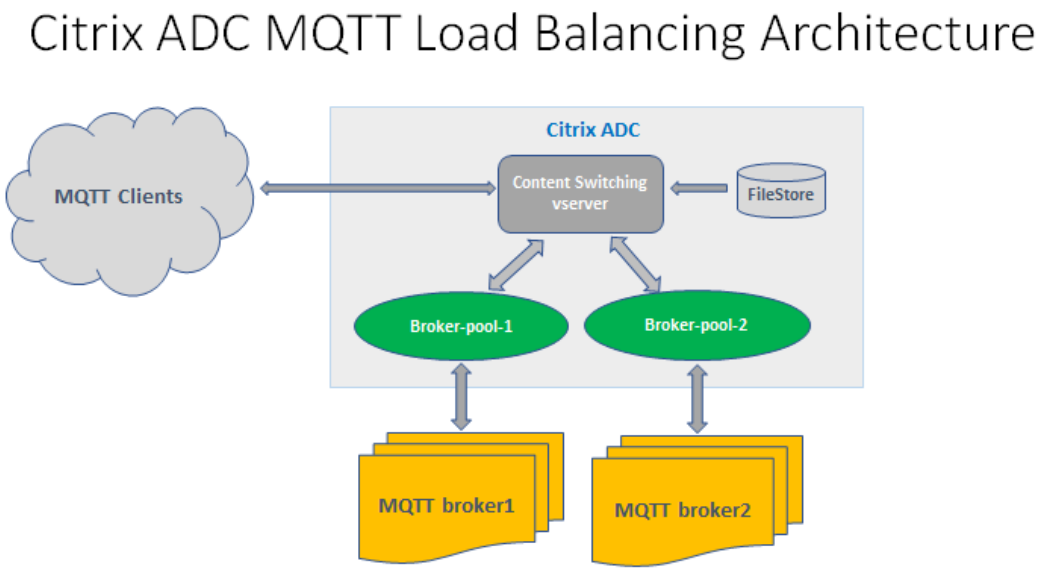
メッセージキューテレメトリトランスポート（MQTT）は、モノのインターネット（IoT）用の OASIS 標準メッセージングプロトコルです。MQTT は、IoT システム内で効果的な通信を提供する柔軟で使いやすいテクノロジーです。MQTT はブローカーベースのプロトコルであり、クライアントとブローカー間のメッセージ交換を容易にするために広く使用されています。

MQTT の次の主な利点により、MQTT は IoT デバイスに最適なオプションになります。

- 信頼性
- 速い応答時間
- 無制限のデバイスをサポートする機能
- Publish/subscribe 多対多のコミュニケーションに最適なメッセージング

IoT は、センサー、ソフトウェア、ネットワーク接続、および必要な電子機器が組み込まれた相互接続されたデバイスのネットワークです。組み込みコンポーネントにより、IoT デバイスはデータを収集および交換できます。IoT デバイスの使用の増加は、ネットワークインフラストラクチャに複数の課題をもたらし、スケールが目立ったものです。IoT デバイスの大規模な展開では、各 IoT デバイスによって生成されたデータを迅速に分析する必要があります。スケール要件とリソースの効率的な使用を実現するには、ブローカープールの負荷を均等に分散する必要があります。MQTT プロトコルのサポートにより、IoT 展開で Citrix ADC アプライアンスを使用して、MQTT トラフィックの負荷分散を行うことができます。

次の図は、Citrix ADC アプライアンスを使用して MQTT トラフィックの負荷を分散する MQTT アーキテクチャを示しています。



MQTT プロトコルを使用した IoT デプロイメントには、次のコンポーネントがあります。

- **MQTT** ブローカー。クライアントからすべてのメッセージを受信し、メッセージを適切な宛先クライアントにルーティングするサーバー。ブローカーは、すべてのメッセージの受信、メッセージのフィルタリング、各メッセージのサブスクライブ者の決定、およびこれらのサブスクライブされたクライアントへのメッセージの送信を担当します。ブローカーは、すべてのメッセージが通過する必要のある中央ハブです。
- **MQTT** クライアント。マイクロコントローラーから本格的なサーバーまで、MQTT ライブラリを実行し、ネットワーク経由で MQTT ブローカーに接続するすべてのデバイス。パブリッシャーとサブスクライバーはどちらも MQTT クライアントです。パブリッシャーとサブスクライバーのラベルは、クライアントがメッセージを公開しているか、メッセージを受信するようにサブスクライブしているかを示します。
- **MQTT** ロードバランサー。Citrix ADC アプライアンスは、MQTT トラフィックを負荷分散するために MQTT 負荷分散仮想サーバーで構成されています。

一般的な IoT 展開では、ブローカー（サーバーのクラスタ）が IoT デバイスのグループ（IoT クライアント）を管理します。Citrix ADC アプライアンスは、クライアント ID、トピック、ユーザー名などのさまざまなパラメーターに基づいて、ブローカーへの MQTT トラフィックの負荷を分散します。

MQTT トラフィックの負荷分散を構成する

Citrix ADC アプライアンスで MQTT トラフィックの負荷を分散するには、次の構成タスクを実行します。

1. 構成、設定 MQTT/MQTT_TLS サービスまたはサービスグループ。
2. 構成、設定 MQTT/MQTT_TLS 仮想サーバーの負荷分散。
3. バインドする MQTT/MQTT_TLS へのサービス MQTT/MQTT_TLS 仮想サーバーの負荷分散。
4. 構成、設定 MQTT/MQTT_TLS コンテンツスイッチング仮想サーバー。

5. ターゲットの負荷分散仮想サーバーを指定するコンテンツスイッチングアクションを構成します
6. コンテンツスイッチングポリシーを構成します。
7. コンテンツスイッチングポリシーを、特定の負荷分散仮想サーバーにリダイレクトするように既に構成されているコンテンツスイッチング仮想サーバーにバインドします。
8. 構成を保存します。

CLI を使用して **MQTT** トラフィックのロードバランシングを設定するには

構成、設定 MQTT/MQTT_TLS サービスまたはサービスグループ。

```
add service <name> <IP> <protocol> <port>
add servicegroup <ServiceGroupName> <Protocol>
bind servicegroup <serviceGroupName> <IP> <port>
<!--NeedCopy-->
```

例:

```
add service srvc1 10.106.163.3 MQTT 1883
add servicegroup srvcg1 MQTT
bind servicegroup srvcg1 10.106.163.3 1883
<!--NeedCopy-->
```

構成、設定 MQTT/MQTT_TLS 仮想サーバーの負荷分散。

```
add lb vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver lb1 MQTT 10.106.163.9 1883
<!--NeedCopy-->
```

バインドする MQTT/MQTT_TLS MQTT 負荷分散仮想サーバーへのサービスまたはサービスグループ。

```
bind lb vserver <name> <serviceName>
bind lb vserver <name> <servicegroupName>
<!--NeedCopy-->
```


例:

```
bind lb vserver lb1 srvc1
bind lb vserver lb1 srvcg1
<!--NeedCopy-->
```

構成、設定 MQTT/MQTT_TLS コンテンツスイッチング仮想サーバー。

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

例:

```
add cs vserver cs1 MQTT 10.106.163.13 1883
<!--NeedCopy-->
```

ターゲットの負荷分散仮想サーバーを指定するコンテンツスイッチングアクションを構成します。

```
add cs action <name> -targetLBVserver <string> [-comment <string>]
<!--NeedCopy-->
```

例:

```
add cs action act1 -targetlbvserver lbv1
<!--NeedCopy-->
```

コンテンツスイッチングポリシーを構成します。

```
add cs policy <policyName> [-url <string> | -rule <expression>] -
    action <actName>
<!--NeedCopy-->
```

例:


```
add cs policy cspol1  -rule   “MQTT.COMMAND.EQ(CONNECT) && MQTT.CONNECT
    .FLAGS.QOS.eq(2)”  -action act1
<!--NeedCopy-->
```

コンテンツスイッチングポリシーを、特定の負荷分散仮想サーバーにリダイレクトするように既に構成されているコンテンツスイッチング仮想サーバーにバインドします。

```
bind cs vserver <virtualServerName> -policyName <policyName> -priority
    <positiveInteger>
<!--NeedCopy-->
```

例:

```
bind cs vserver cs1  - policyName cspol1 -priority 20
<!--NeedCopy-->
```

構成を保存します。

```
save ns config
<!--NeedCopy-->
```

GUI を使用して **MQTT** トラフィックの負荷分散を構成するには

1. トラフィック管理に移動します > 負荷分散 > 仮想サーバー、およびタイプ **MQTT** またはの負荷分散仮想サーバーを作成します MQTT_TLS 。**
2. タイプ MQTT のサービスまたはサービスグループを作成します。
3. サービスを MQTT 仮想サーバーにバインドします。
4. [保存] をクリックします。

MQTT メッセージの長さの制限

Citrix ADC アプライアンスは、メッセージ長が 65536 バイトを超えるメッセージをジャンボパケットとして扱い、デフォルトで破棄します。dropmqttjumbomessage lb パラメーターは、ジャンボパケットを処理するかどうかを決定します。このパラメーターはデフォルトで **YES** に設定されています。これは、ジャンボ MQTT パケットがデフォルトでドロップされることを意味します。このパラメータが **NO** に設定されている場合、ADC アプライアンスはメッセージ長が 65536 バイトを超えるパケットも処理します。

CLI を使用してジャンボパケットを処理するように ADC アプライアンスを構成するには:


```
Set lb parameter  -dropMqttJumboMessage [YES | NO]
<!--NeedCopy-->
```

例:

```
set lb parameter  -dropMqttJumboMessage no
<!--NeedCopy-->
```

負荷分散構成を障害から保護する

October 7, 2021

負荷分散仮想サーバーに障害が発生した場合、または仮想サーバーが過剰なトラフィックを処理できない場合、負荷分散の設定が失敗することがあります。以下の設定により、ロードバランシングの設定を障害から保護できます。

- Citrix ADC アプライアンスは、余分なトラフィックを代替 URL にリダイレクトし、
- バックアップ負荷分散仮想サーバ、および
- ステートフル接続のフェイルオーバー。

クライアント要求を代替 **URL** にリダイレクトする

October 7, 2021

HTTP または HTTPS タイプの負荷分散仮想サーバが DOWN になったり、無効になったりした場合は、HTTP 302 リダイレクトを使用して代替 URL に要求をリダイレクトできます。代替 URL は、サーバーのステータスに関する情報を提供できます。構成されたリダイレクト URL は、HTTP 応答のロケーションヘッダーで指定されます。応答で指定される正確な URL は、次の構成オプションによって異なります。

- 設定済みのリダイレクト URL にhttp://www.sample1.example.comなどのドメイン名のみが含まれる場合、HTTP 応答で指定されたリダイレクト URL に Uniform Resource Identifier(URI; ユニフォームリソース ID)が追加されます。設定されたドメイン名への HTTP リクエストで指定されます。たとえば、リクエストに GET http://www.sample2.example.com/images/site_nav.png ヘッダーが含まれている場合、リダイレクト応答の location ヘッダーは location: http://www.sample1.example.com/images/site_nav.png ヘッダーを指定します。

> 注
>
> 要求と応答のドメイン名は異なる場合があります。このトピックでは、概念を説明するために、2 つのドメインを sample1.example.com および sample2.example.com と呼びます。

- 構成されたリダイレクト URL に完全なパスが含まれている場合、リダイレクト応答は、要求の URI に関係なく、完全な構成済み URL を指定します。たとえば、次のような URL は次のとおりです。
  - リクエストされた URL- http://www.redirect.com/en/index.html
  - リダイレクト URL- http://www.redirect.com/en/site_down.html

次の表に、前述の構成オプションを示します。

構成されたリダイレクト URL	HTTP リクエストの URL	HTTP 応答のヘッダー
http://www.sample1.example.com	http://www.sample2.example.com/en/index.html	http://www.sample1.example.com/en/index.html
http://www.sample1.example.com/en/error.html	http://www.sample2.example.com/en/index.html	http://www.sample1.example.com/en/error.html

> 注
>
> - リダイレクト URL を設定する場合、http://example.com URL は http://example.com/ URL と同じではありません。後者には Webroot パス/への完全なパスが含まれているためです。
> - 負荷分散仮想サーバーで、バックアップ仮想サーバーとリダイレクト URL の両方を構成した場合、バックアップ仮想サーバーがリダイレクト URL よりも優先されます。リダイレクトは、プライマリ仮想サーバとバックアップ仮想サーバの両方が DOWN している場合にのみ使用されます。

CLI を使用してクライアント要求を URL にリダイレクトするように仮想サーバーを構成するには

1. 負荷分散仮想サーバーを作成します。

   ```
   set lb vserver -redirect url
   ```

2. リダイレクト URL オプションが期待どおりに機能していることを確認します。仮想サーバーを無効にします。

   ```
   disable vserver <vserver_name>
   ```

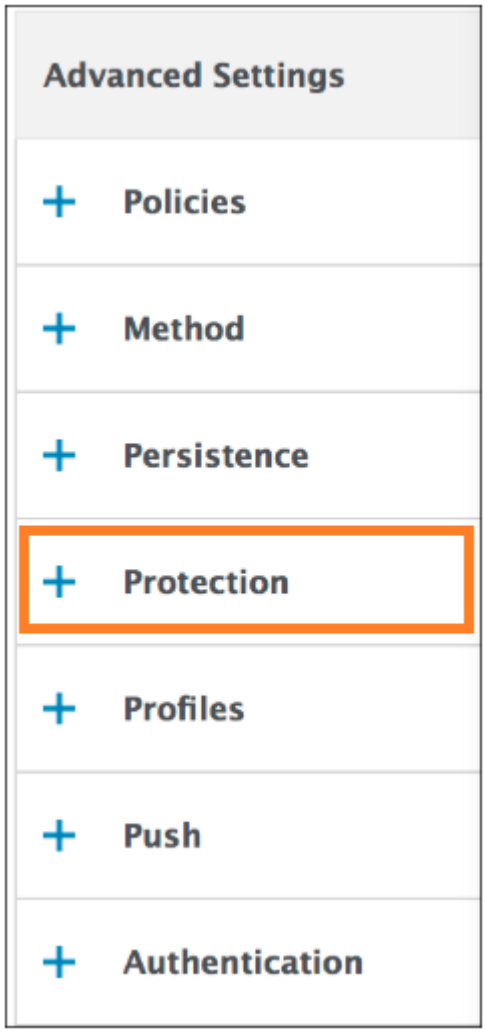
3. Web ブラウザから Web サイトの URL にアクセスして、要求が期待どおりにリダイレクトされることを確認します。Web サイトにアクセスする前に、Web ブラウザのキャッシュをクリアして新しい接続を確立する必要がある場合があります。

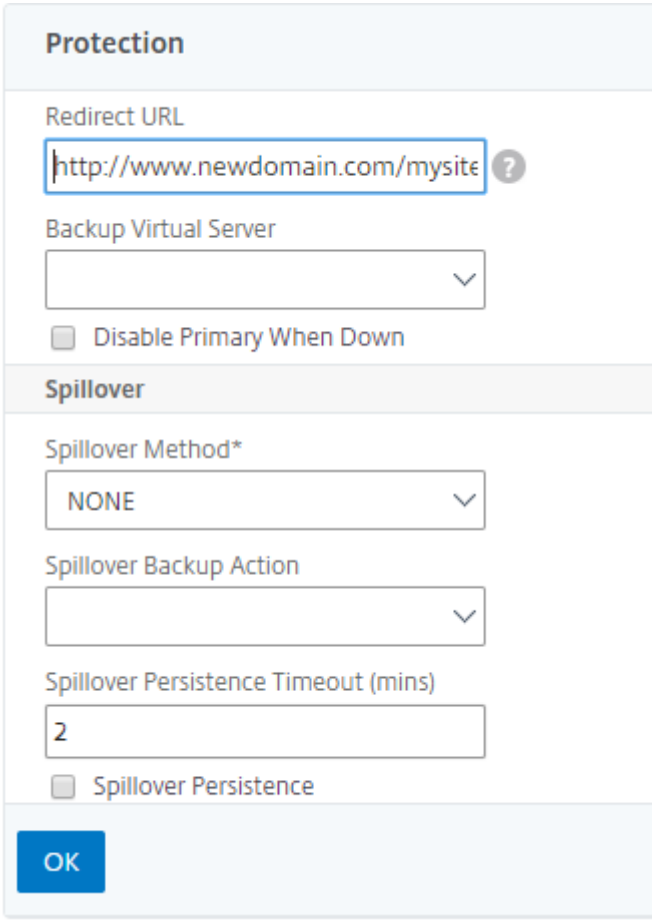
4. 仮想サーバーを有効にします。

   ```
   enable vserver <vserver_name>
   ```


GUI を使用してクライアント要求を **URL** にリダイレクトするように仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、新しい仮想サーバーを追加するには、[追加] をクリックします。
3. 既存の仮想サーバーを編集するには、リストから仮想サーバーを選択し、[編集] をクリックします。
4. [詳細設定] タブで、[保護] をクリックします。[リダイレクト **URL**] フィールドに、リダイレクト URL（http://www.newdomain.com/mysite/maintenanceなど）を入力します。





5. [**OK**] をクリックします。

バックアップ負荷分散仮想サーバーの構成

October 7, 2021

プライマリ負荷分散仮想サーバーがダウンまたは使用できないときに、バックアップ仮想サーバーに要求を送信するように Citrix ADC アプライアンスを構成できます。バックアップ仮想サーバーはプロキシであり、クライアントに対して透過的です。アプライアンスは、サイトの停止に関する通知メッセージをクライアントに送信することもできます。

> 注:
>
> プライマリ仮想サーバが削除または無効化された後でも、バックアップ仮想サーバは、既存の接続を処理し続けます。

バックアップロードバランシング仮想サーバは、作成時に構成することも、既存の仮想サーバのオプションパラメータを変更することもできます。既存のバックアップ仮想サーバーのバックアップ仮想サーバーを構成して、カスケー

ドバックアップ仮想サーバーを作成することもできます。バックアップ仮想サーバーをカスケードする最大の深さは、10 です。

2 つのサーバに接続する仮想サーバが複数ある場合は、プライマリ仮想サーバが DOWN してから再びアップした場合の動作を選択できます。デフォルトの動作では、プライマリ仮想サーバの役割をプライマリとして再開します。ただし、バックアップ仮想サーバが引き継ぐときに制御を維持するように構成することはできます。たとえば、バックアップ仮想サーバ上の更新をプライマリ仮想サーバに同期し、元のプライマリサーバにそのロールを手動で強制的に再開させることができます。この場合、プライマリ仮想サーバがダウンしてから復帰したときに制御を維持するようにバックアップ仮想サーバを指定できます。

プライマリロードバランシング仮想サーバのリダイレクト URL は、プライマリ仮想サーバとバックアップ仮想サーバの両方が DOWN しているか、要求を処理するためのしきい値に達した場合のフォールバックとして設定できます。仮想サーバーにバインドされたサービスが OUT OF SERVICE の場合、アプライアンスはリダイレクト URL を使用します。

注：負荷分散仮想サーバーがバックアップ仮想サーバーとリダイレクト URL の両方で構成されている場合、バックアップ仮想サーバーはリダイレクト URL よりも優先されます。リダイレクトは、プライマリ仮想サーバとバックアップ仮想サーバがダウンしている場合にのみ使用されます。

CLI を使用してバックアップ仮想サーバを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <vServerName> -backupVserver <BackupVServerName> [-
    disablePrimaryOnDown]
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -
    disablePrimaryOnDown
<!--NeedCopy-->
```

GUI を使用してバックアップ仮想サーバを設定するには

1. **Traffic Management** > **Load Balancing** > **Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[保護] をクリックし、バックアップ仮想サーバーを選択します。
3. プライマリ仮想サーバーが復旧した場合でも、手動でプライマリ仮想サーバーを有効にするまで、バックアップ仮想サーバーを制御したままにするには、[ダウン時にプライマリを無効にする] を選択します。

> 注： Citrix ADC バージョン 12.1 ビルド 51.xx 以降、GUI にはそのサーバーの有効状態が表示され、バックアップがアクティブかどうかが示されます。
>
> 現在のサーバーの有効状態は、次のいずれかになります。
>
> - **UP** — サーバーが UP 中であることを示します。
> - **DOWN** — サーバーがダウンしていることを示します。
> - **UP**（**Backup Active**）：プライマリ仮想サーバまたはセカンダリ仮想サーバのいずれかが UP で、トラフィックがバックアップ仮想サーバに送信されることを示します。
> - **DOWN**（**Backup Active**）：プライマリ仮想サーバとバックアップ仮想サーバの両方がダウンし、トラフィックがバックアップ仮想サーバにルーティングされることを示します。
>
> プライマリ仮想サーバで **[Disable Primary When Down**] オプションが有効になっていて、プライマリサーバが DOWN して再び UP 状態になった場合、プライマリ仮想サーバが明示的に再び有効になるまで、トラフィックはバックアップ仮想サーバによって処理されます。`enable lb vserver <vserver_name>`コマンドを使用して、プライマリ仮想サーバを再度有効にすることができます。

スピルオーバーの構成

October 7, 2021

アプライアンスのスピルオーバー構成は、スピルオーバー方式、スピルオーバーしきい値、およびバックアップ仮想サーバで構成されるプライマリ仮想サーバで構成されます。バックアップ仮想サーバーをスピルオーバー用に構成して、バックアップ仮想サーバーのチェーンを作成することもできます。

スピルオーバー方式では、スピルオーバー構成のベースとなる動作条件（確立された接続の数、帯域幅、またはサーバファームの健全性の組み合わせなど）を指定します。新しい接続が到着すると、アプライアンスはプライマリ仮想サーバがアップしていることを確認し、動作状態と設定されたスピルオーバーしきい値を比較します。しきい値に達すると、スピルオーバー機能は、新しい接続をバックアップチェーン内の最初に使用可能な仮想サーバに誘導します。バックアップ仮想サーバは、プライマリの負荷がしきい値を下回るまで、受信する接続を管理します。

スピルオーバー永続性を設定すると、プライマリの負荷がしきい値を下回った後でも、バックアップ仮想サーバは受信した接続を処理し続けます。スピルオーバー持続性とスピルオーバー持続性タイムアウトを構成すると、バックアップ仮想サーバーは、プライマリの負荷がしきい値を下回った後、指定された期間だけ接続を処理します。

注意: 通常、波及効果は、波及効果メソッドに関連付けられた値がしきい値（接続数など）を超えた場合にトリガーされます。ただし、サーバーヘルススピルオーバー方式では、サーバーファームのヘルスがしきい値を下回ると、スピルオーバーがトリガーされます。

次のいずれかの方法でスピルオーバーを設定できます。

- 定義済みのスピルオーバー方法を指定します。4 つの定義済みメソッドが利用可能で、一般的なスピルオーバー要件を満たします。

- ポリシーベースのスピルオーバーを構成します。ポリシーベースの波及効果では、Citrix ADC ルールを使用して、波及効果が発生する条件を指定します。Citrix ADC ルールを使用すると、さまざまな動作条件に合わせてスピルオーバーを柔軟に設定できます。

定義済みのメソッドが要件を満たしていない場合は、ポリシーベースのスピルオーバーを使用します。プライマリ仮想サーバの両方を構成する場合、ポリシーベースのスピルオーバー構成は、事前定義された方法よりも優先されます。

まず、バックアップチェーンに必要なプライマリ仮想サーバーと仮想サーバーを作成します。バックアップチェーンを設定するには、1 つの仮想サーバーをプライマリのバックアップとして指定し (つまり、セカンダリ仮想サーバーを作成する)、セカンダリのバックアップとして仮想サーバーを指定する (第 3 の仮想サーバーを作成する) などを指定します。次に、定義済みのスピルオーバー方法を指定するか、スピルオーバーポリシーを作成してバインドすることによって、スピルオーバーを構成します。

仮想サーバーを別の仮想サーバーのバックアップとして割り当てる手順については、「バックアップ負荷分散仮想サーバーの構成」を参照してください。

定義済みのスピルオーバーメソッドの構成

定義済みのスピルオーバーメソッドは、より一般的なスピルオーバー要件を満たすものです。定義済みのスピルオーバー方法の 1 つを使用するには、プライマリ仮想サーバーでスピルオーバーパラメーターを構成します。バックアップ仮想サーバのチェーンを作成するには、バックアップ仮想サーバのスピルオーバーパラメータも構成します。

バックアップ仮想サーバーが独自のしきい値に達し、サービスタイプが TCP の場合、Citrix ADC アプライアンスはクライアントに TCP リセットを送信します。サービスタイプ HTTP、SSL、および RTSP の場合、新しい要求をプライマリ仮想サーバー用に構成されたリダイレクト URL に転送します。リダイレクト URL は、HTTP、SSL、および RTSP 仮想サーバに対してのみ指定できます。リダイレクト URL が構成されていない場合、Citrix ADC アプライアンスはクライアントに TCP リセット（仮想サーバーのタイプが TCP の場合）または HTTP 503 応答（仮想サーバーのタイプが HTTP または SSL の場合）を送信します。

注：RTSP 仮想サーバーでは、Citrix ADC アプライアンスはスピルオーバーにデータ接続のみを使用します。バックアップ RTSP 仮想サーバが使用できない場合、要求は RTSP URL にリダイレクトされ、RTSP リダイレクトメッセージがクライアントに送信されます。

コマンドラインインターフェイスを使用して仮想サーバーの定義済みのスピルオーバー方法を構成するには

コマンドプロンプトで入力します。

```
1 set lb vserver <vServerName> -soMethod <spillOverType> -soThreshold <
    positiveInteger> -soPersistence ENABLED -soPersistenceTimeout <
    positiveInteger>
2 <!--NeedCopy-->
```


例

```
set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーの定義済みのスピルオーバー方法を構成するには

1. **Traffic Management** > **Load Balancing** > **Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[保護] をクリックし、スピルオーバーパラメータを設定します。

ポリシーベースのスピルオーバーの構成

ルール（式）に基づくスピルオーバーポリシーを使用すると、より広範なスピルオーバーシナリオに合わせてアプライアンスを設定できます。たとえば、仮想サーバーの応答時間に基づいて、または仮想サーバーのサージキュー内の接続数に基づいてスピルオーバーを構成できます。

ポリシーベースのスピルオーバーを設定するには、まずスピルオーバーアクションを作成します。次に、波及効果ポリシーで使用する式を選択し、ポリシーを構成して、アクションをそれに関連付けます。最後に、スピルオーバーポリシーを負荷分散、コンテンツスイッチング、またはグローバルサーバーの負荷分散仮想サーバーにバインドします。複数のスピルオーバーポリシーを 1 つの仮想サーバに、プライオリティ番号でバインドできます。アプライアンスは、スピルオーバーポリシーをプライオリティ番号の昇順に評価し、TRUE と評価される最後のポリシーに関連付けられたアクションを実行します。

仮想サーバには、バックアップアクションを設定することもできます。バックアップアクションは、仮想サーバーに 1 つ以上のバックアップ仮想サーバーがない場合、またはすべてのバックアップ仮想サーバーがダウンしているか、無効になっているか、独自の波及効果の制限に達した場合に実行されます。

スピルオーバーポリシーによって UNDEF 条件 (ポリシー評価の結果が未定義の場合にスローされる例外) が発生すると、UNDEF アクションが実行されます。UNDEF アクションは、常に ACCEPT アクションです。選択した UNDEF アクションを指定することはできません。

スピルオーバーアクションの設定

スピルオーバーアクションは、関連付けられているスピルオーバーポリシーが TRUE と評価されたときに実行されます。現在、スピルオーバーアクションは SPILLOVER のみサポートされています。

コマンドラインインターフェイスを使用してポリシーベースのスピルオーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、スピルオーバーポリシーを構成し、構成を確認します。


```
add spillover action <name> -action SPILLOVER

show spillover action <name>
<!--NeedCopy-->
```

例

```
add spillover action mySoAction -action SPILLOVER
Done
<!--NeedCopy-->
```

```
show spillover action mySoAction
1) Name: mySoAction Action: SPILLOVER
Done
<!--NeedCopy-->
```

スピルオーバーポリシーの式の選択

ポリシー式では、ブール値を返す任意の仮想サーバベースの式を使用できます。たとえば、次のいずれかの式を使用できます。

```
SYS.VSERVER("vserver").RESPTIME.GT(<int>)
SYS.VSERVER("vserver").STATE.EQ( " <string> " ), and
SYS.VSERVER("vserver").THROUGHPUT.LT    (<int>)
<!--NeedCopy-->
```

RESPTIME、STATE、THROUGHPUT などの既存の機能に加えて、この機能で導入された次の仮想サーバベースの機能を使用できます。

Averagesurgecount

アクティブなサービスのサージキューにあるリクエストの平均数を返します。アクティブなサービスがない場合は 0 (ゼロ) を返します。コンテンツスイッチングまたはグローバルサーバーの負荷分散仮想サーバーで使用すると、UNDEF 条件が発生します。

Activeservices

アクティブなサービスの数を返します。コンテンツスイッチングまたはグローバルサーバーの負荷分散仮想サーバーで使用すると、UNDEF 条件が発生します。

Activetransactions

現在のアクティブなトランザクションの仮想サーバーレベルのカウンタの値を返します。

is_dynamic_limit_reached

仮想サーバーが管理する接続の数が動的に計算されたしきい値と等しい場合、ブール値の TRUE を返します。動的しきい値は、UP しているバインドされたサービスの最大クライアント（Max Clients）設定の合計です。

ポリシー式を使用して、事前定義された任意のスピルオーバーメソッドを実装できます。次の表は、定義済みのスピルオーバーメソッドを、それらを実装するために使用できる式にマップします。

表 1. 定義済みのスピルオーバーメソッドをポリシー式に変換する

定義済みのスピルオーバー方法	対応する式
CONNECTION	SYS.VSERVER(“<vserver-name>”).CONNECTIONS, used with the GT(int) arithmetic function.
帯域幅	SYS.VSERVER(“<vserver-name>”).THROUGHPUT, used with the GT(int) arithmetic function.
HEALTH	SYS.VSERVER(“<vserver-name>”).HEALTH, used with the LT(int) arithmetic function.
DYNAMICCONNECTION	SYS.VSERVER(“<vserver-name>”).IS_DYNAMIC_LIMIT_REACHED 注:IS_DYNAMIC_LIMIT_REACHED 関数を使用してポリシーベースのスプルオーバーを実装する場合は、スピルオーバーに必要な統計が機能するように、仮想サーバーに対して事前定義された DYNAMICCONNECTION メソッドも構成する必要があります。が収集されます。

スピルオーバーポリシーの設定

スピルオーバーポリシーは、ブール式をルールとして使用して、スピルオーバーが発生するために満たす必要がある条件を指定します。

コマンドラインインターフェイスを使用してスピルオーバーポリシーを構成するには

コマンドプロンプトで次のコマンドを入力して、スピルオーバーポリシーを構成し、構成を確認します。

```
add spillover policy <name> -rule <expression> -action <string> [-
    comment <string>]

show spillover policy <name>
<!--NeedCopy-->
```

例

```
> add spillover policy mySoPolicy -rule SYS.VSERVER(”v1”).RESPTIME.GT
    (50) -action mySoAction -comment ”Triggers spillover when the
    vserver's response time is greater than 50 ms.”
Done

> show spillover policy mySoPolicy

1)  Name: mySoPolicy Rule: ”SYS.VSERVER(”v1”).RESPTIME.GT(50)” Action:
    mySoAction Hits: 0 ActivePolicy: 0
Comment: ”Triggers spillover when the vserver's response time is
    greater than 50 ms.”
Done
>
<!--NeedCopy-->
```

仮想サーバーへのスピルオーバーポリシーのバインド

スピルオーバーポリシーは、負荷分散、コンテンツスイッチング、またはグローバルサーバーの負荷分散仮想サーバーにバインドできます)。評価のフローを制御する Goto 式を使用して、複数のポリシーを仮想サーバにバインドできます。

コマンドラインインターフェイスを使用してスピルオーバーポリシーを仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力して、スピルオーバーポリシーを負荷分散、コンテンツスイッチング、またはグローバルサーバーの負荷分散仮想サーバーにバインドし、構成を確認します。

```
bind (lb | cs | gslb) vserver <name> -policyName <string> -priority <
    positive_integer> [-gotoPriorityExpression <expression>]

show (lb | cs | gslb) vserver <name>
<!--NeedCopy-->
```

例

```
> bind lb vserver vserver1 -policyName mySoPolicy -priority 5
Done
> show lb vserver vserver1
vserver1 (2.2.2.12:80) - HTTP Type: ADDRESS
. . .

1)  Spillover Policy Name: mySoPolicy Priority: 5
GotoPriority Expression: END
Flowtype: REQUEST
Done
>
<!--NeedCopy-->
```

スピルオーバーイベントのバックアップアクションの設定

バックアップアクションでは、スピルオーバーしきい値に達したが、1 つ以上のバックアップ仮想サービスが設定されていないか、ダウン、無効、または独自のしきい値に達した場合の処理を指定します。

注: 仮想サーバー上で直接構成される定義済みのスピルオーバーメソッドの場合 (「スピルオーバーメソッド」パラメーターの値として)、バックアップアクションは構成できません。デフォルトでは、アプライアンスはクライアントに TCP リセット（仮想サーバのタイプが TCP の場合）または HTTP 503 応答（仮想サーバのタイプが HTTP または SSL の場合）を送信します。

バックアップアクションは、仮想サーバ上で構成されます。仮想サーバーを構成して、要求を受け入れる（ポリシーで指定されたしきい値に達した後)、クライアントを URL にリダイレクトする、または要求の数がしきい値を下回るまで TCP または SSL 接続を確立する前でも要求を単にドロップすることができます。したがって、データ構造を割り当てる前でも接続がリセットされるため、使用されるメモリリソースは少なくなります。

CLI を使用してスピルオーバーのバックアップアクションを設定するには

コマンドプロンプトで次のコマンドを入力して、バックアップアクションを構成し、構成を確認します。

```
set lb vserver <name> -soBackupAction <soBackupAction>

show lb vserver <name>
<!--NeedCopy-->
```

例:

```
set lb vserver vs1 -soBackupAction REDIRECT -redirectURL `http://www.
    mysite.com/maintenance`
Done
> show lb vserver vs1
vs1 (10.102.29.76:80) - HTTP Type: ADDRESS
State: UP
. . .
Redirect URL: `http://www.mysite.com/maintenance`
. . .
Done
<!--NeedCopy-->
```

GUI を使用してスピルオーバーのバックアップアクションを構成するには

1. **Traffic Management** > **Load Balancing** > **Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[保護] をクリックし、スピルオーバーバックアップ操作を指定します。

接続フェイルオーバー

July 15, 2022

接続フェイルオーバーは、分散環境に展開されているアプリケーションへのアクセスの中断を防ぐのに役立ちます。Citrix ADC 高可用性（HA）セットアップでは、接続フェイルオーバー（または接続ミラーリング-CM）とは、フェールオーバーが発生したときに、確立された TCP または UDP 接続をアクティブに保つことを指します。新しいプライマリ Citrix ADC アプライアンスには、フェイルオーバー前に確立された接続に関する情報があり、それらの接続を引き続き提供します。フェイルオーバー後も、クライアントは同じ物理サーバに接続されたままになります。新しいプライマリアプライアンスは、情報を新しいセカンダリアプライアンスと同期します。L2Conn パラメータが設定されている場合、レイヤ 2 接続パラメータもセカンダリと同期されます。

> 注:
>
> HA セットアップについて考えてみます。クライアントはプライマリノードとのセッションを確立し、プライマリノードがバックエンドサーバーとのセッションを確立します。この状態でフェールオーバーがトリガーされると、既存のクライアントおよびサーバノードから新しいプライマリで受信されたパケットは古いパケットとして扱われ、クライアントとサーバの接続がリセットされます。一方、ステートレス接続フェールオーバーが有効になっている場合（USIP がオン）、フェールオーバー後、クライアントまたはサーバーノードからパケットを受信しても、接続がリセットされません。代わりに、クライアント接続とサーバー接続は動的に作成されます。

接続フェールオーバーは、ステートレスモードまたはステートフルモードのいずれかで設定できます。ステートレス接続フェイルオーバーモードでは、HA ノードはフェールオーバーされた接続に関する情報を交換しません。このメソッドには実行時のオーバーヘッドはありません。

ステートフル接続フェールオーバーモードでは、プライマリアプライアンスはフェールオーバー接続のデータを新しいセカンダリアプライアンスと同期します。

接続フェールオーバーは、展開に長期間接続がある場合に役立ちます。たとえば、FTP 経由で大きなファイルをダウンロードしていて、ダウンロード中にフェールオーバーが発生した場合、接続が切断され、ダウンロードが中止されます。ただし、ステートフルモードで接続フェールオーバーを設定すると、フェールオーバー後もダウンロードが継続されます。

Citrix ADC アプライアンスでの接続フェイルオーバーの仕組み

ステートレス接続フェールオーバーでは、新しいプライマリアプライアンスは、受信したパケットに含まれる情報に基づいてパケットフローを再作成しようとします。

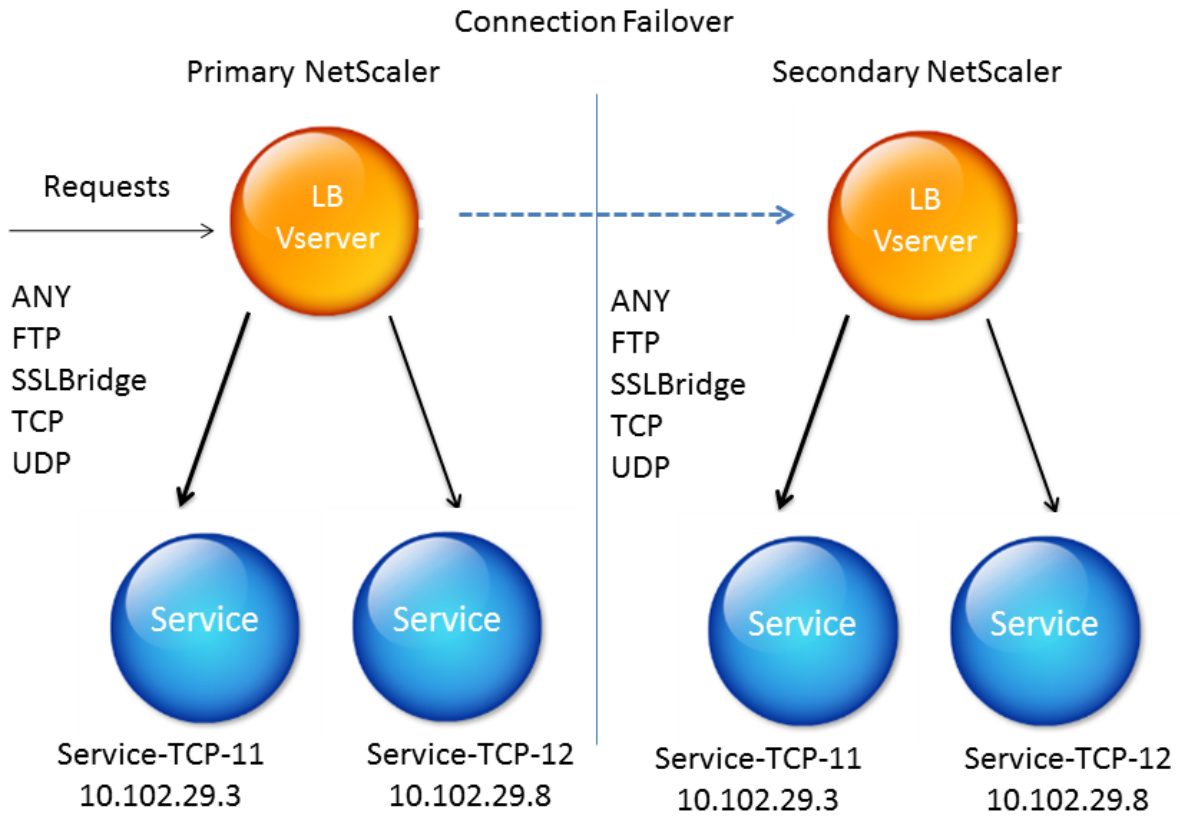
ステートフルフェールオーバーでは、ミラーリングされた接続に関する最新の情報を維持するために、プライマリアプライアンスはセカンダリアプライアンスにメッセージを送信します。セカンダリアプライアンスは、パケットに関連するデータを保持しますが、フェールオーバーの場合にのみ使用します。フェイルオーバーが発生すると、新しいプライマリ（古いセカンダリ）アプライアンスは、ミラーリングされた接続に関する保存されたデータを使用してトラフィックを受け入れます。移行期間中、クライアントとサーバーで短時間の中断と再送信が発生する可能性があります。

注:

プライマリアプライアンスがセカンダリアプライアンスで自分自身を認証できることを確認します。パスワードの正しい設定を確認するには、コマンドラインから show `rpcnode`コマンドを使用するか、GUI の [ネットワーク] メニューの [RPC] オプションを使用します。

接続フェイルオーバーを使用する基本的な HA 設定には、次の図に示すエンティティが含まれます。

図 1: 接続フェールオーバーエンティティの図



> 注
>
> 次のいずれかのイベントが発生すると、接続フェールオーバーはサポートされません。

```
1 - An upgrade to a later release.
2 - An upgrade to a later build within the same release, if the new
      build uses a different HA version.
```

サポートされているセットアップ

接続フェールオーバーは、負荷分散仮想サーバーでのみ構成できます。コンテンツスイッチング仮想サーバーでは構成できません。コンテンツスイッチング仮想サーバーに接続されている負荷分散仮想サーバーで接続のフェールオーバーを有効にすると、負荷分散仮想サーバーは最初にトラフィックを受け入れないため、接続のフェールオーバーは機能しません。

次の表に、接続フェイルオーバーでサポートされているセットアップを示します。

表 1. 接続フェールオーバー-サポートされている設定

設定	ステートレス	ステートフル
サービスタイプ	どれでも。	任意、UDP、TCP、FTP、SSL_BRIDGE。
負荷分散方法	サービスタイプ ANY でサポートされるすべてのメソッド。ただし、ソース IP 永続性が設定されていない場合は、SRCIPSRCPORTHASH メソッドを使用する必要があります。	サポートされているサービスタイプに適用されるすべての方法。
持続性タイプ	SOURCEIP パーシステンス。	サポートされているサービスタイプに適用されるすべてのタイプがサポートされています。
USIP	オンにする必要があります。	制限なし。オンまたはオフにすることができます。
サービスバインディング	サービスは 1 つの仮想サーバーにのみバインドできます。	サービスは 1 つ以上の仮想サーバーにバインドできます。
インターネットプロトコル (IP) バージョン	IPv4 と IPv6	IPv4 と IPv6
冗長性サポート	クラスタリングと高可用性	高可用性

> 注:
>
> ステートフル接続フェールオーバーは、TCP などの接続ベースのスイッチングサービスでのみサポートされます。HTTP は要求ベースのスイッチングを使用するため、接続のフェールオーバーはサポートされません。SSL では、フェールオーバー後に既存の接続がリセットされます。

接続フェイルオーバーの影響を受ける機能

次の表に、接続フェールオーバーが構成されている場合に影響を受ける機能を示します。

表 2. 接続フェールオーバーが Citrix ADC 機能に与える

機能	接続フェールオーバーの影響
SYN プロテクション	どの接続でも、アプライアンスが SYN-ACK を発行した後、最終的な ACK を受信する前にフェールオーバーが発生した場合、接続フェールオーバーによって接続はサポートされません。クライアントは、接続を確立するために要求を再発行する必要があります。

機能	接続フェールオーバーの影響
サージ保護	サーバとの接続が確立される前にフェイルオーバーが発生した場合、新しいプライマリアプライアンスはサーバとの接続を確立しようとします。また、サージ保護中に保持されているすべてのパケットを再送信します。
アクセスダウン	有効にすると、アクセスダウン機能は接続のフェールオーバーよりも優先されます。
Application firewall	アプリケーションファイアウォール機能はサポートされていません。
INC	独立したネットワーク設定は、ハイアベイラビリティモードではサポートされません。
TCP バッファリング	TCP バッファリングは接続ミラーリングと互換性がありません。
応答時に閉じる	フェールオーバー後、応答時に NATPCB が閉じられないことがあります。

GUI を使用して接続のフェールオーバーを構成するには

トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。仮想サーバーを開き、[詳細設定] で [保護] をクリックし、[ステートフル] として [接続フェイルオーバー] を選択します。

CLI を使用して接続フェールオーバーを構成するには

コマンドプロンプトで、次の操作を行います。

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -connFailover stateful
Done
<!--NeedCopy-->
```


仮想サーバで接続フェイルオーバーを無効にすると、仮想サーバに割り当てられたリソースが解放されます。

CLI を使用して接続フェールオーバーを無効にするには

コマンドプロンプトで、次の操作を行います。

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -connFailover disable
Done
<!--NeedCopy-->
```

GUI を使用して接続のフェールオーバーを無効にするには

トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。仮想サーバーを開き、[保護] で [接続フェイルオーバー] で [無効] を選択します。

サージキューをフラッシュする

October 7, 2021

物理サーバが要求の急増を受信すると、現在接続しているクライアントへの応答が遅くなり、ユーザが不満になり、不満や不満が残ります。多くの場合、オーバーロードにより、クライアントはエラーページを受信します。Citrix ADC アプライアンスは、サービスへの新しい接続を確立できる速度を制御し、過負荷を回避するサージ保護などの機能を提供します。

アプライアンスは、クライアントと物理サーバ間の接続の多重化を行います。サーバ上のサービスにアクセスするためのクライアント要求を受信すると、アプライアンスはすでに確立されているサーバへの接続を空いているか探します。空き接続が見つかった場合、その接続を使用してクライアントとサーバー間の仮想リンクを確立します。既存の空き接続が見つからない場合、アプライアンスはサーバーとの新しい接続を確立し、クライアントとサーバーの間に仮想リンクを確立します。ただし、アプライアンスがサーバーとの新しい接続を確立できない場合、クライアント要求をサージキューに送信します。負荷分散またはコンテンツスイッチング仮想サーバーにバインドされているすべての物理サーバーがクライアント接続の上限（最大クライアント値、サージ保護しきい値、またはサービスの最大容量）に達した場合、アプライアンスはどのサーバーとも接続を確立できません。サージ保護機能は、サージキューを使用

して、物理サーバとの接続が開かれる速度を調整します。アプライアンスは、仮想サーバにバインドされたサービスごとに異なるサージキューを維持します。

アプライアンスが接続を確立できない要求が来るたびに、サージキューの長さが長くなります。サージキューの長さは、次の条件のいずれかで減少します。

- キュー内のリクエストがサーバーに送信されます。
- リクエストがタイムアウトし、キューから削除されます。

サービスまたはサービスグループのサージキューが長くなりすぎる場合は、それをフラッシュすることをお勧めします。特定のサービスまたはサービスグループ、または負荷分散仮想サーバーにバインドされているすべてのサービスとサービスグループのサージキューをフラッシュできます。サージキューをフラッシュしても、既存の接続には影響しません。サージキューに存在する要求だけが削除されます。これらの要求の場合、クライアントは新しい要求を行う必要があります。

コンテンツスイッチング仮想サーバーのサージキューをフラッシュすることもできます。コンテンツスイッチング仮想サーバーが特定の負荷分散仮想サーバーにいくつかの要求を転送し、負荷分散仮想サーバーが他のいくつかの要求も受信する場合、コンテンツスイッチング仮想サーバーのサージキューをフラッシュすると、このコンテンツスイッチングから受信した要求のみが仮想サーバーがフラッシュされます。負荷分散仮想サーバーのサージキュー内の他の要求はフラッシュされません。

注: キャッシュリダイレクト、認証、VPN、または GSLB 仮想サーバーまたは GSLB サービスのサージキューをフラッシュすることはできません。

注:[ソース IP (USIP) を使用] が有効になっている場合は、サージ保護機能を使用しないでください。

CLI を使用してサージキューをフラッシュするには

flush ns surgeQ コマンドは、次のように動作します。

- サージキューをフラッシュする必要があるサービス、サービスグループ、または仮想サーバの名前を指定できます。
- コマンドの実行中に名前を指定すると、指定したエンティティのサージキューがフラッシュされます。複数のエンティティが同じ名前を持っている場合、アプライアンスはそれらすべてのエンティティのサージキューをフラッシュします。
- コマンドの実行中にサービスグループの名前、サーバー名、およびポートを指定すると、アプライアンスは指定されたサービスグループメンバーのみのサージキューをフラッシュします。
- サービスグループの名前（<name>）を指定せずにサービスグループメンバー（<serverName>と<port>）を直接指定することはできません。また、<serverName>なしで<port>を指定することはできません。特定のサービスグループメンバーのサージキューをフラッシュする場合は、<serverName>および<port>を指定します。
- 名前を指定せずにコマンドを実行すると、アプライアンスはアプライアンスに存在するすべてのエンティティのサージキューをフラッシュします。

- サービスグループメンバーがサーバ名で識別される場合は、このコマンドでサーバ名を指定する必要があります。その IP アドレスを指定することはできません。

コマンドプロンプトで入力します。

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
<!--NeedCopy-->
```

例

```
flush ns surgeQ  -name SVC1ANZGB  -serverName 10.10.10.1 80
<!--NeedCopy-->
```

前のコマンドは、SVC1ANZGB という名前で IP アドレスが 10.10.10 のサービスまたは仮想サーバのサージキューをフラッシュします。

```
flush ns surgeQ
<!--NeedCopy-->
```

前のコマンドは、アプライアンスのすべてのサージキューをフラッシュします。

GUI を使用してサージキューをフラッシュするには

[トラフィック管理] > [コンテンツスイッチング] > [仮想サーバ] に移動し、仮想サーバを選択し、[アクション] リストで [サージキューのフラッシュ] を選択します。

負荷分散設定の管理

October 7, 2021

既存の負荷分散設定では、変更されていない限り維持するために多大な作業を必要としませんが、ほとんどは長い間変更されません。負荷を増やすには、新しい負荷分散サーバーと、最終的に新しい Citrix ADC アプライアンスを構成し、既存のセットアップに追加する必要があります。古いサーバーは消耗し、交換する必要があります。一部のサーバーを取り外し、他のサーバーを追加する必要があります。ネットワーク機器のアップグレードやトポロジの変更には、ロードバランシングの設定の変更が必要になる場合があります。したがって、サーバーオブジェクト、サービス、および仮想サーバーに対して操作を実行する必要があります。ビジュアライザーは、構成をグラフィカルに表示し、表示内のエンティティに対して操作を実行することができます。また、ロードバランシング設定を通じてトラフィックの管理を容易にする他の機能も利用できます。

サーバオブジェクトの管理

October 7, 2021

基本的な負荷分散セットアップ時に、サービスを作成するときに、サービスの IP アドレスを持つサーバーオブジェクトが存在しない場合は作成されます。IP アドレスではなくドメイン名で名前が付けられたサービスオブジェクトを使用する場合は、1 つ以上のサーバーオブジェクトを手動で作成することもできます。任意のサーバーオブジェクトを有効化、無効化、または削除できます。

サーバーオブジェクトを有効または無効にすると、そのサーバーオブジェクトに関連付けられたすべてのサービスを有効または無効にします。サーバーオブジェクトを無効にした後に Citrix ADC アプライアンスを更新すると、そのサービスの状態は OUT OF SERVICE と表示されます。サーバーオブジェクトを無効にするときに待機時間を指定すると、サーバーオブジェクトは確立された接続を指定した時間だけ処理し続け、新しい接続は拒否します。サーバーオブジェクトを削除すると、そのオブジェクトがバインドされているサービスも削除されます。

CLI を使用してサーバーを有効にするには

コマンドプロンプトで入力します。

```
enable server <name>
<!--NeedCopy-->
```

例:

```
enable server 10.102.29.5
<!--NeedCopy-->
```

GUI を使用してサーバーオブジェクトを有効または無効にするには

1. [トラフィック管理] > [負荷分散] > [サーバー] に移動します。
2. サーバーを選択し、[アクション] リストで [有効] または [無効] を選択します。

CLI を使用してサーバオブジェクトを無効にするには

コマンドプロンプトで入力します。

```
disable server <name> <delay>
<!--NeedCopy-->
```


例:

```
disable server 10.102.29.5 30
<!--NeedCopy-->
```

CLI を使用してサーバオブジェクトを削除するには

コマンドプロンプトで入力します。

```
rm server <name>
<!--NeedCopy-->
```

例:

```
rm server 10.102.29.5
<!--NeedCopy-->
```

GUI を使用してサーバーオブジェクトを削除するには

1. [トラフィック管理] **>** [負荷分散] **>** [サーバー] に移動します。
2. サーバーを選択し、[削除] をクリックします。

サービスの管理

October 7, 2021

サービスは、作成時にデフォルトで有効になっています。各サービスを個別に無効または有効にできます。サービスを無効にする場合は、通常、サービスが確立された接続を処理し続け、新しい接続を拒否してからシャットダウンするまでの待機時間を指定します。待機時間を指定しない場合、サービスはただちにシャットダウンします。待機時間の間、サービスの状態は OUT OF SERVICE です。

使用されなくなったサービスは、削除できます。サービスを削除すると、そのサービスは仮想サーバーからバインド解除され、Citrix ADC 構成から削除されます。

CLI を使用してサービスを有効または無効にするには

コマンドプロンプトで入力します。

```
enable service <name>

disable service <name> <DelayInSeconds>
<!--NeedCopy-->
```

例:

```
enable service Service-HTTP-1
disable service Service-HTTP-1 30
<!--NeedCopy-->
```

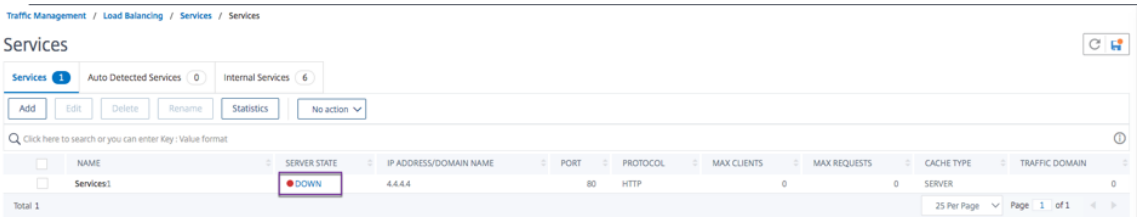
GUI を使用してサービスを有効または無効にするには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. サービスを開き、[操作] リストで [有効] または [無効] を選択します。

GUI を使用して、DOWN とマークされたサービス状態の原因を特定する

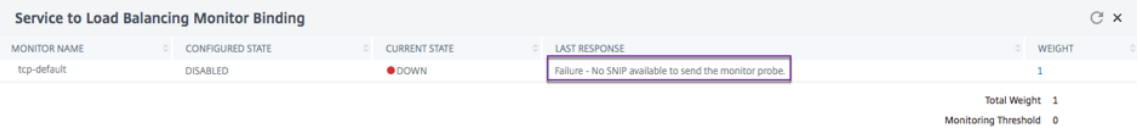
Citrix ADC バージョン 13.0 ビルド 41.20 以降、監視バインディングインターフェイスに移動することなく、ダウンしているサービスの GUI で監視プローブ情報を表示できます。[サービス] ページの [サーバーの状態] 列の値はクリック可能です。**DOWN** をクリックすると、サービスが DOWN とマークされている根本原因を特定できます。

1. **Traffic Management > Load Balancing > Services** に移動します。
2. **DOWN** であるサービスに対応する [サーバーの状態] 列で [DOWN] をクリックします。



「サービスと負荷分散モニターのバインド」ページが表示されます。

[**Last Response**] 列には、サービスが [DOWN] とマークされている理由が表示されます。



負荷分散仮想サーバーの管理

October 7, 2021

仮想サーバは、作成時にデフォルトで有効になります。仮想サーバーは手動で無効化および有効化できます。仮想サーバーを無効にすると、仮想サービスの状態は OUT OF SERVICE と表示されます。この場合、仮想サーバーは、downStateFLush パラメーターの設定に応じて、直ちにまたは既存の接続の完了を許可した後で、すべての接続を終了します。downStateFLush が ENABLED (デフォルト) の場合、すべての接続がフラッシュされます。DISABLED の場合、仮想サーバーは既存の接続で要求を処理し続けます。

仮想サーバーが不要になった場合にのみ、仮想サーバーを削除します。削除する前に、すべてのサービスをバインド解除する必要があります。

CLI を使用して仮想サーバーを有効または無効にするには

コマンドプロンプトで入力します。

```
enable lb vserver <name>
<!--NeedCopy-->
```

```
disable lb vserver <name>
<!--NeedCopy-->
```

例:

```
enable lb vserver Vserver-LB-1
disable lb vserver Vserver-LB-1
<!--NeedCopy-->
```

GUI を使用して仮想サーバーを有効または無効にするには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを選択し、［操作］リストで［有効］または［無効］を選択します。

CLI を使用して仮想サーバからサービスをバインド解除するには

コマンドプロンプトで入力します。


```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
unbind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

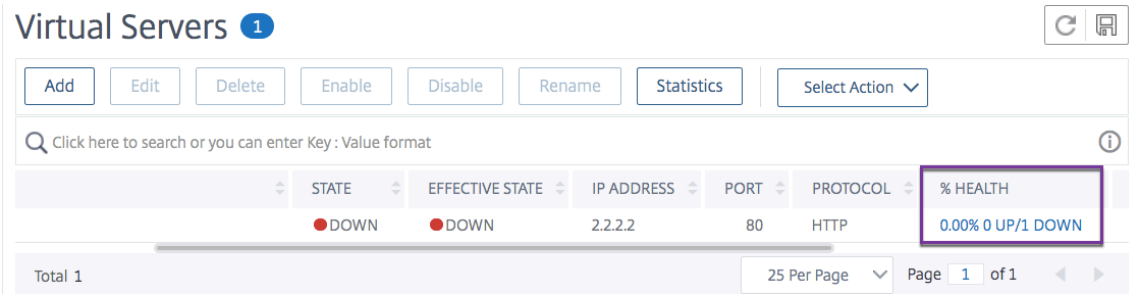
GUI を使用して仮想サーバーからサービスをバインド解除するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを開き、[サービス] セクションのをクリックします。
3. サービスを選択し、[バインド解除] をクリックします。

GUI を使用して、DOWN とマークされた仮想サーバ状態の原因を特定する

Citrix ADC バージョン 13.0 ビルド 41.20 以降、モニタバインディングインターフェイスに移動することなく、ダウンしている仮想サーバーの GUI でモニタープローブ情報を表示できます。[仮想サーバー] ページの [**% HEALTH**] 列の値は、クリック可能です。[**% HEALTH**] 列の値をクリックすると、仮想サーバーが DOWN とマークされている根本原因を特定できます。

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 停止している仮想サーバーに対応する [**% HEALTH**] 列の値をクリックします。

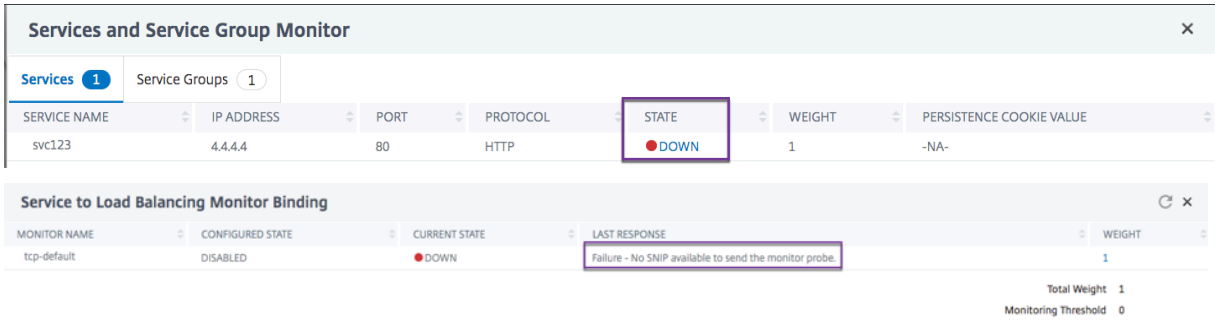


[サービスおよびサービスグループモニタ] ページが表示されます。この仮想サーバにバインドされたサービスとサービスグループがそれぞれのタブに表示されます。

仮想ロードバランシングにバインドされたサービスを使用している場合は、次の手順を実行します。

[サービス] タブで、ダウンしているサービスに対応する **DOWN** をクリックします。

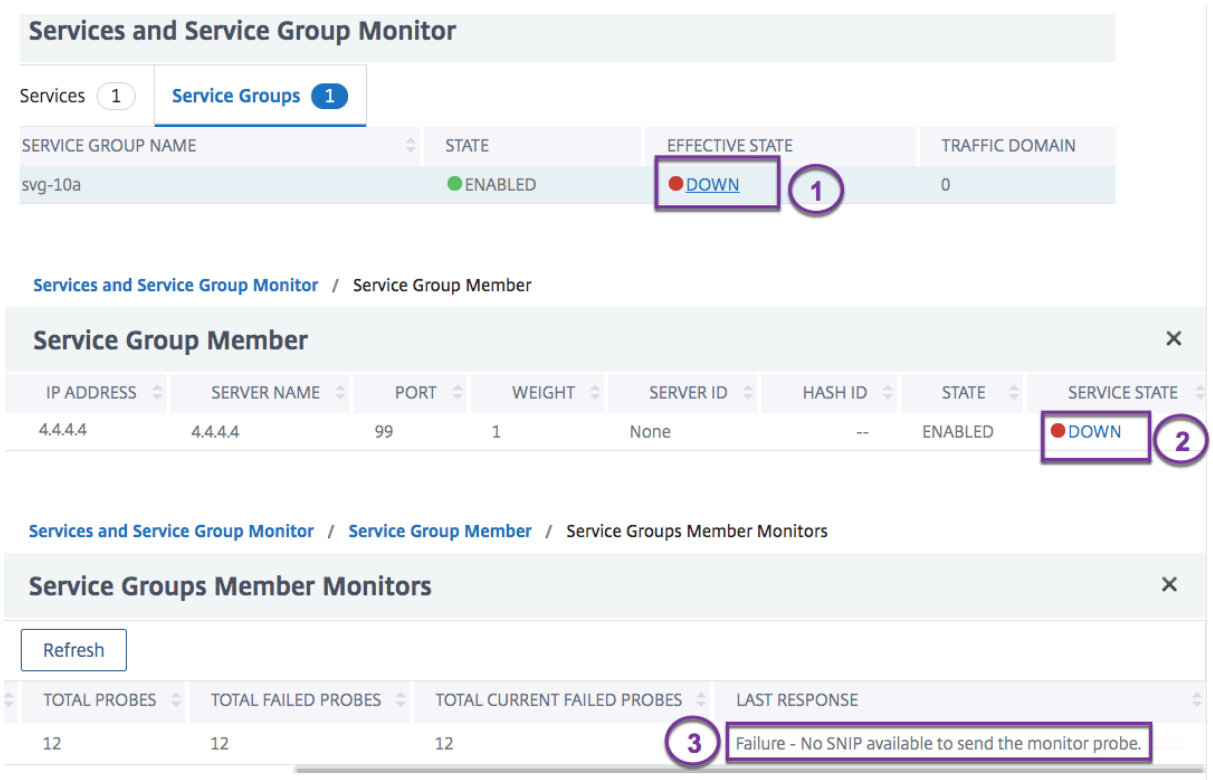
[負荷分散モニターのバインドへのサービス] ページの [最後の応答] 列に、仮想サーバーがマークダウンされた理由が表示されます。



仮想ロードバランシングにバインドされたサービスグループを使用している場合は、次の手順を実行します。

[サービスグループ] タブで、[サービスとサービスグループモニター] ページの **DOWN** をクリックし、[サービスグループメンバー] ページで **DOWN** をクリックします。

[サービスグループメンバーモニター] ページの [最後の応答] 列には、仮想サーバがマークダウンされた理由が表示されます。



負荷分散のビジュアライザ

October 7, 2021

負荷分散ビジュアライザは、負荷分散構成をグラフィカル形式で表示および変更するために使用できるツールです。以下は、ビジュアライザー表示の例です。

図 1: ロード・バランシング・ビジュアライザ表示

ビジュアライザーを使用すると、次の内容を表示できます。

- 仮想サーバにバインドされているサービスおよびサービスグループ。
- 各サービスにバインドされているモニタ。
- 仮想サーバにバインドされるポリシー。
- ポリシーラベル（設定されている場合）。
- 表示された要素の設定の詳細。

また、ビジュアライザーを使用して、新しいオブジェクトの追加とバインド、既存のオブジェクトの変更、オブジェクトの有効化または無効化を行うこともできます。ビジュアライザーに表示されるほとんどの構成要素は、構成ユーティリティの他の部分と同じ名前で表示されます。ただし、他の構成ユーティリティとは異なり、Visualizer は、同じ構成の詳細と監視バインディングを持つサービスをサービスコンテナと呼ばれるエンティティにグループ化します。

サービスコンテナーは、単一の負荷分散仮想サーバーにバインドされた同様のサービスとサービスグループのセットです。コンテナ内のサービスは、名前、IP アドレス、ポートを除いて同じプロパティを持ち、それらのモニタバインディングのウェイトとバインディング状態が同じである必要があります。新しいサービスを仮想サーバーにバインドすると、その構成とモニターのバインディングが他のサービスのものと一致する場合、そのサービスは既存のコンテナに配置されます。それ以外の場合は、独自のコンテナに配置されます。

次の手順では、ビジュアライザーを使用する基本的な手順のみを示します。ビジュアライザーは、負荷分散機能の他の領域の機能を複製するため、ビジュアライザーで構成できるすべての設定を表示または構成するその他の方法は、負荷分散のドキュメント全体で提供されています。

注記: ビジュアライザーにはグラフィックインターフェースが必要なので、設定ユーティリティでのみ使用できます。

ビジュアライザーを使用して負荷分散仮想サーバーのプロパティを表示するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。

ビジュアライザーを使用してサービス、サービスグループ、およびモニターの構成の詳細を表示するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。
3. [Load Balancing Visualizer] ダイアログボックスで、エンティティをダブルクリックして、この仮想サーバーにバインドされているエンティティの構成の詳細を表示します。次の操作を実行できます。

構成ユーティリティーのビジュアライザーを使用してポリシーとポリシーラベルの構成の詳細を表示するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、表示する仮想サーバーを選択し、[ビジュアライザー] をクリックします。

3. [負荷分散ビジュアライザ] ダイアログボックスで、ポリシーエンティティをダブルクリックして、この仮想サーバーにバインドされているポリシーを表示します。

ビジュアライザーを使用して負荷分散構成内のリソースを変更するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、構成する仮想サーバーを選択し、[ビジュアライザー] をクリックします。
3. [負荷分散ビジュアライザー] ダイアログボックスの [ビジュアライザー] イメージで、変更するリソースをダブルクリックします。

ビジュアライザーを使用して負荷分散構成を追加するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ウィンドウで、構成する仮想サーバーを選択し、[ビジュアライザー] をクリックします。
3. [負荷分散ビジュアライザー] ダイアログボックスで、[**+**] をクリックしてリソースを追加します。

クライアントトラフィックを管理する

October 7, 2021

クライアント接続を適切に管理することで、Citrix ADC アプライアンスの負荷が高い場合でも、ユーザーがアプリケーションを使用できるようにすることができます。アプライアンスで使用可能なさまざまなロードバランシング機能やその他の機能をロードバランシング設定に統合して、負荷をより効率的に処理し、必要に応じて負荷を迂回し、アプライアンスが実行する必要があるタスクの優先順位付けを行うことができます。

- セッションレスロードバランシング。DSR または IDS（侵入検知システム）を使用する設定でセッションを作成することなく、セッションレスロードバランシング仮想サーバを設定し、ロードバランシングを実行できます。
- 統合キャッシュ。HTTP リクエストをキャッシュにリダイレクトできます。
- プライオリティキューイング。設定をプライオリティキューイング機能と統合することで、プライオリティに基づいて要求を送信できます。
- **SureConnect**。Sure Connect 機能でロードバランシングを使用すると、重要な要求をカスタム Web ページにリダイレクトし、ネットワークの輻輳による遅延からそれらを隔離できます。
- クリーンアップの遅延。仮想サーバー接続の遅延クリーンアップを構成して、Citrix ADC アプライアンスの負荷が高い期間にクリーンアッププロセスで CPU サイクルが使用されないようにすることができます。
- 書き換え。書き換え機能を使用して、HTTP リダイレクトを実行するときにポートとプロトコルを変更したり、仮想サーバーの IP アドレスとポートをカスタムの Request ヘッダーに挿入したりできます。
- **RTSP NAT**。
- レートベースの監視。レートベースのモニタリングを有効にして、過剰なトラフィックを転送できます。

- レイヤ **2** パラメータ。L2 パラメーターを使用して接続を識別するように仮想サーバーを構成できます。
- **ICMP** 応答。設定に従って、PING 要求に ICMP 応答を送信するようにアプライアンスを設定できます。仮想サーバに対応する IP アドレスで、ICMP 応答を VSVR_CNTRLD に設定し、仮想サーバで`ICMP VSERVER RESPONSE`を設定します。
  仮想サーバーでは、次の設定を行うことができます。
  - すべての仮想サーバで`ICMP VSERVER RESPONSE`を PASSIVE に設定すると、アプライアンスは常に応答します。
  - すべての仮想サーバで `ICMP VSERVER RESPONSE`を ACTIVE に設定すると、1 つの仮想サーバが稼働していてもアプライアンスは応答します。
  - 一部では`ICMP VSERVER RESPONSE`を ACTIVE、他では PASSIVE に設定すると、ACTIVE に設定された 1 つの仮想サーバが稼働していてもアプライアンスは応答します。

セッションレス負荷分散仮想サーバーの構成

October 7, 2021

Citrix ADC アプライアンスは、負荷分散を実行するときに、クライアントとサーバー間のセッションを作成および管理します。セッション情報のメンテナンスにより、アプライアンスのリソースに大きな負荷がかかり、ダイレクトサーバリターン（DSR）セットアップや侵入検知システム（IDS）のロードバランシングなどのシナリオでは、セッションは必要ない場合があります。セッションが不要なときにセッションを作成しないようにするには、セッションレスロードバランシング用にアプライアンスの仮想サーバーを構成します。セッションレスロードバランシングでは、アプライアンスはパケット単位でロードバランシングを実行します。

セッションレスロードバランシングは、MAC ベースの転送モードまたは IP ベースの転送モードで動作します。

MAC ベースの転送では、トラフィックの転送先となるすべての物理サーバ上で、セッションレス仮想サーバの IP アドレスを指定する必要があります。

セッションレスロードバランシングの IP ベースの転送では、仮想サーバの IP アドレスとポートを物理サーバ上で指定する必要はありません。これは、この情報が転送されるパケットに含まれているためです。クライアントから物理サーバにパケットを転送する場合、アプライアンスは IP アドレスやポートなどのクライアントの詳細を変更せずに、宛先の IP アドレスとポートを追加します。

サポートされているセットアップ

Citrix ADC セッションレス負荷分散では、次のサービスタイプと負荷分散方式がサポートされています。

サービスタイプ

- ANY（MAC ベースリダイレクション）
- IP ベースのリダイレクション用の ANY、DNS、UDP

負荷分散方法

- ラウンドロビン
- 最小帯域幅
- LRTM（最小応答時間方式）
- 送信元 IP ハッシュ
- 宛先 IP ハッシュ
- 送信元 IP 宛先 IP ハッシュ
- 送信元 IP 送信元ポートハッシュ
- カスタムロード

制限事項

セッションレスロードバランシングには、次の制限があります。

- アプライアンスは 2 アームモードで展開する必要があります。
- サービスは、1 つの仮想サーバにのみバインドする必要があります。
- セッションレスロードバランシングは、サービスグループではサポートされません。
- セッションレスロードバランシングは、ドメインベースのサービス (DBS サービス) ではサポートされません。
- IP モードでのセッションレスロードバランシングは、プライマリ仮想サーバへのバックアップとして構成されている仮想サーバではサポートされません。
- スピルオーバーモードを有効にすることはできません。
- セッションレス負荷分散仮想サーバーにバインドされているすべてのサービスについて、[Use Source IP (USIP)] オプションを有効にする必要があります。
- ワイルドカード仮想サーバーまたはサービスの場合、宛先 IP アドレスは変更されません。

注:

- セッションレスロードバランシング用に仮想サーバを設定する場合は、サポートされているロードバランシング方式を明示的に指定します。デフォルトの方法である [最小接続] は、セッションレスロードバランシングには使用できません。
- 仮想サーバーで MAC ベースリダイレクトモードでセッションレス負荷分散を構成するには、Citrix ADC アプライアンスで MAC ベースの転送オプションを有効にする必要があります。

CLI を使用してセッションレス仮想サーバーを追加するには

コマンドプロンプトで次のコマンドを入力して、セッションレス仮想サーバーを追加し、構成を確認します。

```
add lb vserver <name>@ <serviceType> <IPAddress>@ <port> -m <
    redirectionMode> -sessionless <(ENABLED|DISABLED)> -lbMethod <
    load_balancing_method>
```



```
2
3 show lb vserver <name>
4 <!--NeedCopy-->
```

例:

```
1  add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -
      lbMethod roundrobin -m ip
2       Done
3  show lb vserver sesslessv1
4       sesslessv1 (11.11.12.123:54) - ANY Type: ADDRESS
5       State: DOWN
6       ...
7       Effective State: DOWN
8       Client Idle Timeout: 120 sec
9       Down state flush: ENABLED
10      ...
11      Persistence: NONE
12      Sessionless LB: ENABLED
13      Connection Failover: DISABLED
14      L2Conn: OFF
15      1) Policy : cmp_text Priority:8680 Inherited
16      2) Policy : cmp_nocmp_ie60 Priority:8690 Inherited
17 <!--NeedCopy-->
```

既存の仮想サーバーでセッションレス負荷分散を構成するには

コマンドプロンプトで入力します。

```
1 set lb vserver <name>@ -m <redirectionMode> -sessionless <(ENABLED|
     DISABLED)> -lbMethod <load_balancing_method>
2 <!--NeedCopy-->
```

例

```
1 set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
2   Done
3 <!--NeedCopy-->
```


> 注
>
> `-m MAC`オプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

GUI を使用してセッションレス仮想サーバーを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[詳細設定] で [トラフィックの設定] をクリックし、[セッションレス負荷分散] を選択します。

HTTP 要求をキャッシュにリダイレクトする

October 7, 2021

Citrix ADC キャッシュリダイレクト機能は、HTTP 要求をキャッシュにリダイレクトします。キャッシュリダイレクト機能を適切に実装することで、HTTP リクエストへの応答の影響を大幅に軽減し、Web サイトのパフォーマンスを向上させることができます。

キャッシュは、頻繁に要求された HTTP コンテンツを格納します。仮想サーバーでキャッシュリダイレクトを構成すると、Citrix ADC アプライアンスはキャッシュ可能な HTTP リクエストをキャッシュに送信し、キャッシュ不可能な HTTP リクエストを元の Web サーバーに送信します。

CLI を使用して仮想サーバーでキャッシュリダイレクトを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -cacheable <Value>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -cacheable yes
<!--NeedCopy-->
```

GUI を使用して仮想サーバーでキャッシュリダイレクトを構成するには

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] をクリックし、[キャッシュ可能] を選択します。

優先度に応じた直接要求

October 7, 2021

Citrix ADC アプライアンスは、優先度キューイング機能によるクライアント要求の優先順位付けをサポートします。この機能を使用すると、重要なクライアントからの要求など、特定の要求を優先度要求として指定し、アプライアンスが最初に応答するように「ラインの最前面」に送信できます。これにより、Web サイトでのデマンドサージや DDoS 攻撃を通じて、これらのクライアントに中断のないサービスを提供できます。

CLI を使用して仮想サーバでプライオリティキューイングを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -pq <Value>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -pq yes
<!--NeedCopy-->
```

GUI を使用して仮想サーバでプライオリティキューイングを構成するには

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] をクリックし、[優先キューイング] を選択します。

注: プライオリティキューイングが正常に機能するように、グローバルにプライオリティキューイングを設定します。

カスタム **Web** ページへのリクエストの直接作成

December 7, 2021

Citrix ADC アプライアンスには SureConnect オプションがあり、サーバー容量や処理速度に制限があることが原因で発生する遅延にもかかわらず Web アプリケーションが応答できるようにします。SureConnect は、プライマリ Web ページをホストするサーバーが使用できないか、応答が遅くなっているときに、選択した別のウェブページを表示することによってこれを行います。

仮想サーバーで SureConnect を構成するには、まず代替コンテンツを構成する必要があります。SureConnect ウェブサイトの設定の詳細については、SureConnectを参照してください。Web サイトを構成したら、負荷分散仮想サーバーで SureConnect を有効にして、SureConnect カスタム Web ページを使用します。

注:SureConnect を正しく機能させるには、グローバルに設定する必要があります。

CLI を使用して仮想サーバーで SureConnect を有効にするには

コマンドプロンプトで入力します。

```
set lb vserver <name> -sc <Value>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -sc yes
<!--NeedCopy-->
```

GUI を使用して仮想サーバーで SureConnect を有効にするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] をクリックし、[SureConnect] を選択します。

仮想サーバー接続のクリーンアップを有効にする

October 7, 2021

特定の条件下では、サービスまたは仮想サーバーが DOWN とマークされたときに、既存の接続をただちに終了するように downStateFlush 設定を構成できます。既存の接続を終了すると、リソースが解放され、場合によっては負荷分散セットアップの回復速度が速くなります。

仮想サーバの状態は、バインドされているサービスの状態によって異なります。各サービスの状態は、プローブに対する負荷分散されたサーバーの応答と、そのサービスにバインドされているモニターによって送信されるヘルスチェックによって異なります。負荷分散されたサーバーが応答しないことがあります。サーバが低速またはビジー状態の場合、モニタリングプローブがタイムアウトすることがあります。設定されたタイムアウト期間内に繰り返しモニタリングプローブに応答しない場合、サービスには DOWN とマークされます。

仮想サーバは、バインドされているすべてのサービスが「ダウン」とマークされている場合にのみ、DOWN とマークされます。仮想サーバーが DOWN になると、その直ちに、または既存の接続の完了を許可した後で、すべての接続が終了します。

トランザクションを完了する必要があるアプリケーションサーバーでは、DownStateFlush 設定を有効にしないでください。この設定は、DOWN とマークされたときに接続を安全に終了できる Web サーバー上で有効にできます。

次の表は、仮想サーバー Vserver-LB-1 で構成され、1 つのサービスが Service-TCP-1 にバインドされた構成例に対するこの設定の影響をまとめたものです。この表では、E と D は downStateFlush 設定の状態を示します。E は Enabled、D は Disabled です。

Vserver-LB-1	Service-TCP-1	接続の状態
E	E	クライアント接続とサーバー接続の両方が終了します。
E	D	TCP などの一部のサービスタイプでは、Citrix ADC アプライアンスが接続の再利用をサポートしていない場合、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプの場合、クライアントとサーバーの両方の接続は、それらの接続でトランザクションがアクティブである場合にのみ終了します。トランザクションがアクティブでない場合、クライアント接続のみが終了します。

Vserver-LB-1	Service-TCP-1	接続の状態
D	E	TCP などの一部のサービスタイプでは、Citrix ADC アプライアンスが接続の再利用をサポートしていない場合、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプの場合、クライアントとサーバーの両方の接続は、それらの接続でトランザクションがアクティブである場合にのみ終了します。トランザクションがアクティブでない場合、サーバー接続のみが終了します。
D	D	クライアント接続もサーバー接続も終了しません。

サーバーまたはクライアントによって確立されたすべての接続が閉じられた場合にのみサービスを無効にする場合は、正常なシャットダウンオプションを使用できます。サービスのグレースフルシャットダウンの詳細については、「サービスのグレースフルシャットダウン」を参照してください。

CLI を使用して仮想サーバーでダウン状態のフラッシュ設定を構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -downStateFlush <Value>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -downStateFlush enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーでダウン状態のフラッシュ設定を構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] をクリックし、[ダウン状態のフラッシュ] を選択します。

HTTP リダイレクション用のポートとプロトコルの書き換え

October 7, 2021

仮想サーバーとそれにバインドされているサービスは、異なるポートを使用する場合があります。サービスがリダイレクトで HTTP 接続に応答する場合、リダイレクトが正常に行われるようにポートとプロトコルを変更するように Citrix ADC アプライアンスを構成する必要があります。これを行うには、redirectPortRewrite 設定を有効にして構成します。

この設定は、HTTP トラフィックと HTTPS トラフィックにのみ影響します。この設定が仮想サーバーで有効になっている場合、仮想サーバーはリダイレクト時にポートを書き換え、サービスが使用するポートを、仮想サーバーが使用するポートに置き換えます。

仮想サーバまたはサービスのタイプが SSL の場合は、仮想サーバまたはサービスで SSL リダイレクトを有効にする必要があります。仮想サーバとサービスの両方が SSL タイプである場合は、仮想サーバで SSL リダイレクトを有効にします。

RedirectPortRewrite 設定は、次のシナリオで使用できます。

- 仮想サーバのタイプは HTTP で、サービスのタイプは SSL です。
- 仮想サーバのタイプは SSL で、サービスのタイプは HTTP です。
- 仮想サーバのタイプは HTTP で、サービスのタイプは HTTP です。
- 仮想サーバのタイプは SSL で、サービスのタイプは SSL です。

シナリオ 1: 仮想サーバーのタイプが HTTP で、サービスのタイプが SSL です。SSL リダイレクト、およびオプションでポートリライトがサービスで有効になっています。ポートリライトが有効の場合、HTTPS URL のポートは書き換えられます。サーバーからの HTTP URL はそのままクライアントに送信されます。

SSL リダイレクトだけが有効になります。仮想サーバは、任意のポートに設定できます。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL リダイレクトとポートリライトは有効です。仮想サーバはポート *80* で設定されます。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

SSL リダイレクトとポートリライトは有効です。仮想サーバは、ポート *8080* に設定されています。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	http://domain.com:8080/
https://domain.com:444/	http://domain.com:8080/

シナリオ 2: 仮想サーバーのタイプが SSL で、サービスのタイプが HTTP です。ポートリライトが有効の場合、HTTP URL のポートだけが書き換えられます。サーバからの HTTPS URL はそのままクライアントに送信されます。

仮想サーバーで *SSL* リダイレクトが有効になっています。仮想サーバは、任意のポートに設定できます。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

仮想サーバーで *SSL* リダイレクトとポートリライトが有効になります。仮想サーバは、ポート *443* に設定されています。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL リダイレクトとポートリライトは有効です。仮想サーバは、ポート *444* で設定されます。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	https://domain.com:444/
http://domain.com:8080/	https://domain.com:444/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

シナリオ 3: 仮想サーバーとサービスの種類が HTTP です。仮想サーバでポート書き換えを有効にする必要があります。HTTP URL のポートだけが書き換えられます。サーバからの HTTPS URL はそのままクライアントに送信されます。

仮想サーバはポート *80* で設定されます。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

仮想サーバは、ポート *8080* 上に構成されています。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com:8080/
http://domain.com:8080/	http://domain.com:8080/

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

シナリオ 4: 仮想サーバーとサービスのタイプが SSL です。ポートリライトが有効の場合、HTTPS URL のポートだけが書き換えられます。サーバーからの HTTP URL はそのままクライアントに送信されます。

仮想サーバーで *SSL* リダイレクトが有効になっています。仮想サーバは、任意のポートに設定できます。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

仮想サーバーで *SSL* リダイレクトとポートリライトが有効になります。仮想サーバは、ポート *443* に設定されています。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

仮想サーバーで *SSL* リダイレクトとポートリライトが有効になります。仮想サーバは、ポート *444* で設定されます。次の表を参照してください。

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com:444/

サーバーからのリダイレクト URL	クライアントに送信されたリダイレクト URL
https://domain.com:445/	https://domain.com:444/

CLI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -redirectPortRewrite (ENABLED | DISABLED)
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバーで **HTTP** リダイレクトを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[詳細設定] ウィンドウで [トラフィックの設定] をクリックし、[書き換え] を選択します。

CLI を使用して **SSL** 仮想サーバーまたはサービスで **SSL** リダイレクトを構成するには

コマンドプロンプトで入力します。

```
set ssl vserver <vServerName> - sslRedirect (ENABLED | DISABLED)

set ssl service <serviceName> - sslRedirect (ENABLED | DISABLED)
<!--NeedCopy-->
```

例:

```
set ssl vserver Vserver-SSL-1 -sslRedirect enabled

set ssl service service-SSL-1 -sslRedirect enabled
<!--NeedCopy-->
```


GUI を使用して **SSL** 仮想サーバーまたはサービスで **SSL** リダイレクトおよび **SSL** ポートの書き換えを構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [SSL パラメータ] をクリックし、[SSL リダイレクト] を選択します。

要求ヘッダーに仮想サーバーの **IP** アドレスとポートを挿入する

October 7, 2021

同じサービス上の異なるアプリケーションと通信する複数の仮想サーバーがある場合は、次の操作を実行する必要があります。

Citrix ADC アプライアンスを構成して、そのサービスに送信される HTTP リクエストに適切な仮想サーバーの IP アドレスとポート番号を追加します。この設定により、サービスで実行されているアプリケーションが、要求を送信した仮想サーバーを識別できるようになります。

プライマリ仮想サーバがダウンし、バックアップ仮想サーバが稼働している場合、バックアップ仮想サーバの構成設定がクライアント要求に追加されます。要求がプライマリ仮想サーバからのものかバックアップ仮想サーバからのものかに関係なく、同じヘッダータグを追加する場合は、両方の仮想サーバで必要なヘッダータグを設定する必要があります。

注: このオプションは、ワイルドカード仮想サーバーまたはダミー仮想サーバーではサポートされていません。

CLI を使用して仮想サーバの **IP** アドレスとポートをクライアント要求に挿入するには

コマンドプロンプトで入力します。

```
set lb vserver <name> -insertVserverIPPort <insertVserverIPPort> [<
    vipHeader>]
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr
<!--NeedCopy-->
```

GUI を使用して、クライアント要求に仮想サーバの **IP** アドレスとポートを挿入するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。

2. 仮想サーバーを開き、[詳細設定] ウィンドウで、[トラフィックの設定] をクリックし、[仮想サーバー IP ポートの挿入] を選択し、仮想サーバーの IP ポートヘッダーを指定します。

バックエンド通信に指定したソース **IP** を使用する

December 7, 2021

物理サーバーまたは他のピアデバイスとの通信では、Citrix ADC アプライアンスはソース IP アドレスとして所有する IP アドレスを使用します。Citrix ADC アプライアンスは IP アドレスのプールを維持し、サーバーとの接続中に IP アドレスを動的に選択します。物理サーバが配置されているサブネットに応じて、アプライアンスは使用する IP アドレスを決定します。このアドレスプールは、トラフィックおよびモニタプローブの送信に使用されます。

多くの場合、アプライアンスで、バックエンド通信に特定の IP アドレスまたは特定の IP アドレスのセットからの任意の IP アドレスを使用したい場合があります。次に、いくつかの例を示します。

- モニタプローブに使用される送信元 IP アドレスが特定のセットに属している場合、サーバはモニタプローブとトラフィックを区別できます。
- サーバーのセキュリティを向上させるために、特定の一連の IP アドレスからの要求や、場合によっては単一の特定の IP アドレスからの要求に応答するようにサーバーを構成できます。この場合、アプライアンスは、サーバによって受け入れられた IP アドレスのみを送信元 IP アドレスとして使用できます。
- アプライアンスは、IP アドレスを IP セットに分散し、セットのアドレスを特定のサービスへの接続にのみ使用できれば、内部接続を効率的に管理できます。

指定した送信元 IP アドレスを使用するようにアプライアンスを設定するには、ネットプロファイル（ネットワークプロファイル）を作成し、そのプロファイルを使用するようにアプライアンスエンティティを構成します。ネットプロファイルは、負荷分散またはコンテンツスイッチング仮想サーバー、Citrix Gateway VPN 仮想サーバー、サービス、サービスグループ、またはモニターにバインドできます。ネットプロファイルには、Citrix ADC が所有する IP アドレス（SNIP および VIP）があり、送信元 IP アドレスとして使用できます。単一の IP アドレスでも、IP セットと呼ばれる IP アドレスのセットでもかまいません。ネットプロファイルに IP セットがある場合、アプライアンスは接続時に IP セットから IP アドレスを動的に選択します。プロファイルに 1 つの IP アドレスがある場合、同じ IP アドレスが送信元 IP として使用されます。

ネットプロファイルが負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドされている場合、そのプロファイルは、バインドされているすべてのサービスにトラフィックを送信するために使用されます。ネットプロファイルがサービスグループにバインドされている場合、アプライアンスはサービスグループのすべてのメンバーに対してそのプロファイルを使用します。ネットプロファイルがモニターにバインドされている場合、アプライアンスはモニターから送信されたすべてのプローブに対してプロファイルを使用します。

> 注:
>
> - Citrix ADC アプライアンスが VIP アドレスを使用してサーバーと通信する場合、セッションエントリを使用して、VIP アドレス宛てのトラフィックがサーバーからの応答であるか、クライアントからの要求で

あるかを識別します。

- ネットプロファイルを Citrix Gateway VPN 仮想サーバーにバインドできます。ただし、ネットプロファイルをバインドするときは、いくつかの点に注意する必要があります。詳細については、「ネットプロファイルを VPN 仮想サーバーにバインドするときの注意点」を参照してください。
- サービスまたはサービスグループにバインドされたネットプロファイル IP は、対応するバックエンドサーバーへのトラフィックの送信だけでなく、未解決のバックエンド FQDN によってトリガーされる DNS 要求にも使用されます。

トラフィック送信のためのネットプロファイルの使用

[Use Source IP Address (USIP)] オプションが有効になっている場合、アプライアンスはクライアントの IP アドレスを使用し、すべてのネットプロファイルを無視します。USIP オプションが有効になっていない場合、アプライアンスは次の方法でソース IP を選択します。

- 仮想サーバまたはサービス/サービスグループにネットプロファイルがない場合、アプライアンスはデフォルトの方式を使用します。
- サービス/サービスグループにのみネットプロファイルがある場合、アプライアンスはそのネットプロファイルを使用します。
- 仮想サーバ上にのみネットプロファイルがある場合、アプライアンスはそのネットプロファイルを使用します。
- 仮想サーバとサービス/サービスグループの両方にネットプロファイルがある場合、アプライアンスはサービス/サービスグループにバインドされたネットプロファイルを使用します。

モニタプローブの送信にネットプロファイルを使用します。

モニタプローブの場合、アプライアンスは次の方法でソース IP を選択します。

- モニターにバインドされたネットプロファイルがある場合、アプライアンスはモニターのネットプロファイルを使用します。仮想サーバまたはサービス/サービスグループにバインドされたネットプロファイルは無視されます。
- モニターにバインドされたネットプロファイルがない場合は、
  - サービス/サービスグループにネットプロファイルがある場合、アプライアンスはサービス/サービスグループのネットプロファイルを使用します。
  - サービス/サービスグループにもネットプロファイルがない場合、アプライアンスはデフォルトの方法でソース IP を選択します。

注: サービスにバインドされたネットプロファイルがない場合、サービスがサービスグループにバインドされている場合、アプライアンスはサービスグループ上のネットプロファイルを検索します。

指定した送信元 IP アドレスを通信に使用するには、次の手順に従います。

1. Citrix ADC アプライアンスが所有する SNiP および VIP のプールから IP セットを作成します。IP セットは、SNIP アドレスと VIP アドレスの両方で構成できます。手順については、IP セットの作成を参照してください。
2. ネットプロファイルを作成します。手順については、ネットプロファイルの作成を参照してください。

3. ネットプロファイルをアプライアンスのエンティティにバインドします。手順については、Citrix ADC エンティティへのネットプロファイルのバインドを参照してください。

> 注:
>
> - ネットプロファイルは、Citrix ADC アプライアンスで SNIP および VIP として指定された IP アドレスのみを持つことができます。
> - Citrix ADC が開始したパケットでは、送信元 IP パーシステンスは適用されません。

ネットプロファイルの管理

ネットプロファイル（ネットワークプロファイル）には、1 つの IP アドレスまたは IP セットが含まれます。物理サーバーまたはピアとの通信中、Citrix ADC アプライアンスは、プロファイルで指定されたアドレスを送信元 IP アドレスとして使用します。

- ネットワークプロファイルの作成手順については、ネットワークプロファイルの作成を参照してください。
- ネットワークプロファイルを Citrix ADC エンティティにバインドする手順については、Citrix ADC エンティティへのネットプロファイルのバインドを参照してください。

IP セットを作成する

IP セットは、Citrix ADC アプライアンスでサブネット IP アドレス（SNIP）または仮想 IP アドレス（VIP）として構成される IP アドレスのセットです。IP セットには、そのセットに含まれる IP アドレスの用途を識別するためのわかりやすい名前を付けます。IP セットを作成するには、IP アドレスセットを追加し、Citrix ADC 所有の IP アドレスをバインドします。SNIP アドレスと VIP アドレスは、同じ IP セット内に存在できます。

CLI を使用して IP セットを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add ipset <name>

bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

または

```
bind ipset <name> <IPAddress>

show ipset [<name>]
<!--NeedCopy-->
```


上記のコマンドは、名前を渡さない場合に、アプライアンス上のすべての IP セットの名前を表示します。名前を渡すと、指定した IP セットにバインドされた IP アドレスが表示されます。

例

```
1.
> add ipset skpnwipset
 Done
> bind ipset skpnwipset 21.21.20.1
 Done

2.
 > add ipset testnwipset
 Done
> bind ipset testnwipset 21.21.21.[21-25]
 IPAddress ”21.21.21.21” bound
 IPAddress ”21.21.21.22” bound
 IPAddress ”21.21.21.23” bound
 IPAddress ”21.21.21.24” bound
 IPAddress ”21.21.21.25” bound
 Done

3.
 > bind ipset skpipset 11.11.11.101
 ERROR: Invalid IP address
[This IP address could not be added because this is not an IP address
   owned by the Citrix ADC appliance]
 > add ns ip 11.11.11.101 255.255.255.0 -type SNIP
 ip ”11.11.11.101” added
 Done
 > bind ipset skpipset 11.11.11.101
 IPAddress ”11.11.11.101” bound
 Done
4.
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
3) Name: ipset-3
4) Name: skpnewipset
 Done

5.
> sh ipset skpnewipset
```



```
IP:21.21.21.21
IP:21.21.21.22
IP:21.21.21.23
IP:21.21.21.24
IP:21.21.21.25
 Done
<!--NeedCopy-->
```

GUI を使用して **IP** セットを作成するには

[システム] **>** [ネットワーク] **>** [**IP** セット] に移動し、IP セットを作成します。

ネットプロファイルの作成

ネットプロファイル（ネットワークプロファイル）は、Citrix
ADC アプライアンスの 1 つ以上の SNIP アドレスまたは VIP アドレスで構成されます。

CLI を使用してネットプロファイルを作成するには

コマンドプロンプトで入力します。

```
add netprofile <name> [-srcIp <srcIpVal>]
<!--NeedCopy-->
```

このコマンドで srcipVal が指定されていない場合は、後で`set netprofile`コマンドを使用して指定することができます。

例

```
add netprofile skpnetprofile1 -srcIp 21.21.20.1
Done

add netprofile baksnp -srcIp bakipset
Done

set netprofile yahnp -srcIp 12.12.23.1
Done

set netprofile citkbnp -srcIp citkbipset
```



```
Done
<!--NeedCopy-->
```

ネットプロファイルを **Citrix ADC** エンティティにバインドする

ネットプロファイルは、負荷分散仮想サーバー、サービス、サービスグループ、またはモニターにバインドできます。

注：Citrix ADC エンティティの作成時にネットプロファイルをバインドすることも、既存のエンティティにバインドすることもできます。

コマンドラインインターフェイスを使用してネットプロファイルをサーバーにバインドするには

ネットプロファイルは、負荷分散仮想サーバーとコンテンツスイッチング仮想サーバーにバインドできます。適切な仮想サーバを指定します。

コマンドプロンプトで入力します。

```
set lb vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

または

```
set cs vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

例

```
set lb vserver skpnwvs1 -netProfile gntnp
 Done
set cs vserver mmdcsv -netProfile mmdnp
 Done
<!--NeedCopy-->
```

GUI を使用してネットプロファイルを仮想サーバーにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で [プロファイル] をクリックし、ネットプロファイルを設定します。

CLI を使用してネットプロファイルをサービスにバインドするには

コマンドプロンプトで入力します。

```
set service <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

例

```
set service brnssvc1 -netProfile brnsnp
 Done
<!--NeedCopy-->
```

GUI を使用してネットプロファイルをサービスにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを開きます。
2. [詳細設定] で [プロファイル] をクリックし、ネットプロファイルを設定します。

CLI を使用してネットプロファイルをサービスグループにバインドするには

コマンドプロンプトで入力します。

```
set servicegroup <serviceGroupName> -netProfile <net_profile_name>
<!--NeedCopy-->
```

例

```
set servicegroup ndhsvcgrp -netProfile ndhnp
 Done
<!--NeedCopy-->
```

GUI を使用してネットプロファイルをサービスグループにバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [サービスグループ] に移動し、サービスグループを開きます。
2. [詳細設定] で [プロファイル] をクリックし、ネットプロファイルを設定します。

CLI を使用してネットプロファイルをモニターにバインドするには

コマンドプロンプトで入力します。

```
set monitor <monitor_name> -netProfile <net_profile_name>
```

例

```
1 set monitor brnsecvmon1 -netProfile brnsmonnp
2  Done
3 <!--NeedCopy-->
```

GUI を使用してネットプロファイルをモニターにバインドするには

1. トラフィック管理 **>** 負荷分散 **>** モニターに移動します。
2. モニターを開き、ネットプロファイルを設定します。

アイドル状態のクライアント接続のタイムアウト値を設定する

October 7, 2021

構成されたタイムアウト期間 (秒単位) が経過した後、アイドル状態のクライアント接続を終了するように仮想サーバーを構成できます。この設定を構成すると、Citrix ADC アプライアンスは指定した時間待機し、その時間以降にクライアントがアイドル状態になると、クライアント接続を閉じます。デフォルトでは、クライアントのアイドルタイムアウト値は 180 秒に設定されています。

CLI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
1 set lb vserver <name> -cltTimeout <Value>
2 <!--NeedCopy-->
```

例**:**

```
1 set lb vserver Vserver-LB-1 -cltTimeout 100
2 <!--NeedCopy-->
```


GUI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[トラフィックの設定] をクリックし、クライアントのアイドルタイムアウト値を秒単位で設定します。

RTSP 接続の管理

October 7, 2021

Citrix ADC アプライアンスは、2 つのトポロジ（NAT オンモードまたは NAT オフモード）のいずれかを使用して、RTSP サーバーの負荷を分散できます。NAT オンモードでは、ネットワークアドレス変換 (NAT) がアプライアンスで有効化され、構成されます。RTSP 要求と応答は両方ともアプライアンスを通過します。したがって、ネットワークアドレス変換（NAT）を実行してデータ接続を識別するようにアプライアンスを設定する必要があります。

NAT の有効化と設定の詳細については、IP アドレッシングを参照してください。

NAT オフモードでは、NAT は有効になっておらず、設定されていません。アプライアンスは、クライアントから RTSP 要求を受信し、設定されたロードバランシング方式を使用して選択したサービスにルーティングします。負荷分散された RTSP サーバは、アプライアンスをバイパスして、クライアントに応答を直接送信します。したがって、ダイレクトサーバーリターン (DSR) モードを使用するようにアプライアンスを設定し、DNS でパブリックにアクセス可能な FQDN をロードバランシングされた RTSP サーバーに割り当てる必要があります。

DSR モードの有効化と構成の詳細については、「ダイレクトサーバーリターンモードでの負荷分散の構成」を参照してください。DNS の構成の詳細については、「ドメインネームシステム」を参照してください。いずれの場合も、RTSP ロードバランシングを設定する場合は、ロードバランシング設定のトポロジと一致するように rtspNat も設定する必要があります。

CLI を使用して RTSP NAT を設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name>  - RTSPNAT <ValueOfRTSPNAT>
<!--NeedCopy-->
```

例:

```
set lb vserver vserver-LB-1  - RTSPNAT ON
<!--NeedCopy-->
```


GUI を使用して **RTSP NAT** を設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、RTSP タイプの仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定**]** をクリックし、[**RTSP** ナット] を選択します。

トラフィックレートに基づいてクライアントトラフィックを管理する

October 7, 2021

負荷分散仮想サーバーを通過するトラフィックのレートを監視し、トラフィックレートに基づいて Citrix ADC アプライアンスの動作を制御できます。次に例を示します:

- トラフィックフローが高すぎる場合は、トラフィックフローをスロットリングします。
- トラフィックレートに基づいて情報をキャッシュします。
- トラフィックレートが高すぎる場合は、余分なトラフィックを別の負荷分散仮想サーバーにリダイレクトします。
- HTTP およびドメインネームシステム (DNS) 要求にレートベースのモニタリングを適用します。

レートベースのポリシーの詳細については、レート制限を参照してください。

レイヤ **2** パラメータによる接続の識別

October 7, 2021

一般に、Citrix ADC アプライアンスは、接続を識別するために、クライアント IP アドレス、クライアントポート、宛先 IP アドレス、宛先ポートの 4 タプルを使用します。[L2 接続] オプションを有効にすると、通常の 4 タプルに加えて、接続のレイヤ 2 パラメータ（チャネル番号、MAC アドレス、VLAN ID）が使用されます。

ロードバランシング仮想サーバの L2Conn パラメータを有効にすると、同じ 4 タプル（<source IP>:<source port>::<destination IP>:<destination port>）で複数の TCP 接続と非 TCP 接続が可能になります。) を Citrix ADC アプライアンスに共存させることができます。アプライアンスは、4 タプルとレイヤ 2 の両方のパラメータを使用して、TCP 接続と非 TCP 接続を識別します。

L2Conn オプションは、次のシナリオで有効にできます。

- Citrix ADC アプライアンスには複数の VLAN が設定され、各 VLAN にファイアウォールが設定されます。
- 1 つの VLAN 内のサーバから発信され、別の VLAN 内の仮想サーバ宛てのトラフィックが、両方の VLAN に設定されたファイアウォールを通過するようにします。

したがって、1 つ以上の負荷分散仮想サーバーに l2Conn パラメーターが設定されている nCore Citrix ADC アプライアンスが、l2Conn パラメーターをサポートしない Classic ビルドまたは nCore ビルドにダウングレードされると、l2Conn パラメーターを使用する負荷分散構成は無効になります。

CLI を使用して **L2** 接続オプションを設定するには

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <IPAddress>@ <port> -l2Conn ON
<!--NeedCopy-->
```

例

```
 add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON
<!--NeedCopy-->
```

GUI を使用して **L2** 接続オプションを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[レイヤ 2 パラメータ] を選択します。

[ダイレクトルートの優先] オプションを構成する

October 7, 2021

ワイルドカード負荷分散仮想サーバーでは、宛先へのルートを明示的に構成した場合、デフォルトでは、Citrix ADC アプライアンスは設定されたルートに従ってトラフィックを転送します。アプライアンスが構成済みのルートを検索しないようにするには、[直接ルートを優先] オプションを [いいえ] に設定します。

デバイスが Citrix ADC アプライアンスに直接接続されている場合、アプライアンスはトラフィックをデバイスに直接転送します。たとえば、パケットの宛先がファイアウォールの場合、パケットを別のファイアウォール経由でルーティングする必要はありません。ただし、デバイスが直接接続されていても、トラフィックがファイアウォールを通過したい場合があります。このような場合は、[直接ルートを優先] オプションを [いいえ] に設定できます。

注：preferDirectRoute 設定は、Citrix ADC アプライアンス上のすべてのワイルドカード仮想サーバーに適用されます。

CLI を使用して **[ダイレクトルートの優先]** オプションを設定するには

コマンドプロンプトで入力します。


```
set lb parameter -preferDirectRoute (YES | NO)
<!--NeedCopy-->
```

例:

```
set lb parameter -preferDirectRoute YES
<!--NeedCopy-->
```

GUI を使用して **[**ダイレクトルートを優先**]** オプションを設定するには

1. [トラフィック管理**] > [**負荷分散**] > [**ロードバランシングパラメータの設定] に移動します。
2. [直接ルートを優先] を選択します。

バックエンド通信に指定したポート範囲の送信元ポートを使用する

October 7, 2021

デフォルトでは、USIP オプションが無効になっているか、USIP でプロキシポートオプションを使用する構成の場合、Citrix ADC アプライアンスはランダムなソースポート（1024 以上）からサーバーと通信します。

アプライアンスは、指定されたポート範囲の送信元ポートを使用して、サーバーとの通信をサポートします。この機能の使用例の 1 つは、ロギングおよびモニタリングを目的として、送信元ポートに基づいて特定のセットに属する受信トラフィックを識別するように設定されたサーバを対象としています。たとえば、ロギング目的で内部および外部トラフィックを識別します。

サーバーとの通信にポート範囲の送信元ポートを使用するように Citrix ADC アプライアンスを構成するには、次のタスクを実行します。

- ネットプロファイルを作成し、送信元ポート範囲パラメータを設定します。送信元ポート範囲パラメータは、1 つ以上のポート範囲を指定します。アプライアンスは、指定されたポート範囲から空きポートの 1 つをランダムに選択し、サーバーへの接続ごとに送信元ポートとして使用します。
- ネットプロファイルをロードバランシング仮想サーバ、サービス、またはサービスグループにバインドする: ソースポート範囲が設定されたネットプロファイルを、ロードバランシング構成の仮想サーバ、サービス、またはサービスグループにバインドできます。仮想サーバーへの接続の場合、アプライアンスはネットプロファイルの指定されたポート範囲から空きポートの 1 つをランダムに選択し、このポートをバインドされたサーバーの 1 つに接続するための送信元ポートとして使用します。

CLI を使用して送信元ポート範囲を指定するには

コマンドプロンプトで入力します。

```
bind netProfile <name> (-srcPortRange <int[-int]> ...)

show netprofile <name>
<!--NeedCopy-->
```

GUI を使用して送信元ポート範囲を指定するには

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. NetProfiles の追加または変更時に、送信元ポート範囲パラメータを設定します。

構成例

次の設定例では、ネットプロファイル PARTI-NAT-1 には部分的な NAT 設定があり、タイプ ANY のロードバランシング仮想サーバー LBVS-1 にバインドされています。192.0.0.0/8 から LBVS-1 で受信したパケットの場合、Citrix ADC アプライアンスはパケットの送信元 IP アドレスの最後のオクテットを 100 に変換します。たとえば、送信元 IP アドレスが 192.0.2.30 のパケットが LBVS-1 で受信された場合、Citrix ADC アプライアンスは、送信元 IP アドレスを 100.0.2.30 に変換してから、バインドされたサーバーの 1 つを送信します。

```
```
> add netprofile CUSTOM-SRCPORT-NP-1
Done
> bind netprofile CUSTOM-SRCPRT-NP-1 - srcportrange 2000-3000

Done
> bind netprofile CUSTOM-SRCPRT-NP-1 - srcportrange 5000-6000

Done
> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1

Done
<!--NeedCopy--> ```
```

 
```

バックエンド通信の送信元 **IP** パーシステンシーを構成する

October 7, 2021

デフォルトでは、USIP オプションを無効にし、ネットプロファイルを仮想サーバーまたはサービスグループにバインドした負荷分散構成の場合、Citrix ADC アプライアンスはラウンドロビンアルゴリズムを使用して、サーバーと通信するためのネットプロファイルから IP アドレスを選択します。この選択方法により、選択される IP アドレスは、特定のクライアントのセッションによって異なる場合があります。

場合によっては、Citrix ADC アプライアンスがトラフィックをサーバーに送信するときに、特定のクライアントのトラフィックをすべて同じ IP アドレスからルーティングする必要があります。たとえば、サーバは、ロギングおよびモニタリングを目的として、特定のセットに属するトラフィックを識別できます。

ネットプロファイルのソース IP 永続性オプションを使用すると、Citrix ADC アプライアンスは、ネットプロファイルに指定された同じアドレスを使用して、特定のクライアントから仮想サーバーへ開始されたすべてのセッションについてサーバーと通信できます。

CLI を使用してネットプロファイルで送信元 **IP** 永続性を有効にするには

ネット・プロファイルの追加中にソース IP の永続性を有効にするには、コマンド・プロンプトで次のように入力します。

```
add netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

既存のネットプロファイルでソース IP の永続性を有効にするには、コマンド・プロンプトで次のように入力します。

```
set netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

GUI を使用してネットプロファイルで送信元 **IP** 永続性を有効にするには

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. ネット・プロファイルの追加または変更時に、「ソース **IP** の持続性」を選択します。

例

次の設定例では、ネットプロファイル NETPROFILE-IPPRSTNCY-1 が送信元 IP 永続性オプションが有効になっており、ロードバランシング仮想サーバー LBVS-1 にバインドされています。

Citrix ADC アプライアンスは、特定のクライアントから仮想サーバーへのすべてのセッションについて、LBVS-1 にバインドされたサーバーと通信するために、常に同じ IP アドレス（この例では 192.0.2.11）を使用します。

```
```
> add ipset IPSET-1

Done
> bind ipset IPSET-1 192.0.2.[11-15]
 IPAddress ”192.0.2.11” bound
 IPAddress ”192.0.2.12” bound
 IPAddress ”192.0.2.13” bound
 IPAddress ”192.0.2.14” bound
 IPAddress ”192.0.2.15” bound
 Done
> add netprofile NETPROFILE-IPPRSTNCY-1 -srcIp IPSET-1 -
    srcippersistency ENABLED

Done
> set lb vserver LBVS-1 -netprofile NETPROFILE-IPPRSTNCY-1

 Done
<!--NeedCopy--> ```
```

## ロードバランシングセットアップのサーバー側で **IPv6** リンクローカルアドレスを使用する

October 7, 2021

IPv6 リンクローカルアドレスは、ロードバランシング設定のサービス、サービスグループ、およびサーバでサポートされます。サービス、サービスグループ、およびサーバ設定で、関連する VLAN ID とともにリンクローカル IPv6 アドレスを指定できます。Citrix ADC アプライアンスは、サービス、サービスグループ、およびサーバー構成で指定された同じ VLAN のリンクローカル SNIP6 アドレスを使用して、それらと通信します。

リンクローカル IPv6 アドレスおよび関連する VLAN ID は、サービス、サービスグループ、およびサーバの設定で次の形式で指定されます。<IPv6_Addrs>%<vlan_id>

 
```

たとえば、fe80:123:4567::a%2048:, fe80:123:4567::a はリンクローカルアドレスで、2048 は VLAN ID です。

```
> add service SERVICE-1 fe80:123:4567::a%2048 HTTP 80

Done
> bind servicegroup SERVICE-GROUP-1 fe80::1%24 80

Done
> add server SERVER-1 fe80:b:c:d::e:f:a/64%1028

Done
```

高度な負荷分散設定

October 7, 2021

仮想サーバーの構成に加えて、サービスの詳細設定を構成できます。

詳細なロードバランシング設定を構成するには、次の項を参照してください。

- 仮想サーバレベルの低速スタートにより、新しいサービスの負荷を段階的にステップアップ
- サービスの監視なしオプション
- 保護されたサーバ上のアプリケーションをトラフィックの急増から保護する
- 仮想サーバーおよびサービス接続のクリーンアップを有効にする
- サービスの正常なシャットダウン
- TROFS サービスでの持続性セッションの有効化または無効化
- カスタム Web ページへのリクエストの直接作成
- ダウン時にサービスへのアクセスを有効にする
- 応答の TCP バッファリングを有効にします
- 圧縮を有効化
- 複数のクライアント要求に対してクライアント接続を維持する
- リクエストヘッダーにクライアントの IP アドレスを挿入する
- ジオロケーションデータベースを使用してユーザーの IP アドレスから場所の詳細を取得する
- サーバーへの接続時にクライアントの送信元 IP アドレスを使用する
- サーバー側接続用の送信元ポートの構成
- クライアント接続数の制限を設定する
- サーバーへの接続あたりの要求数の制限を設定する
- サービスにバインドされたモニターのしきい値を設定する
- アイドル状態のクライアント接続のタイムアウト値を設定する

- アイドル状態のサーバー接続のタイムアウト値を設定する
- クライアントによる帯域幅使用量の制限を設定する
- クライアント要求をキャッシュにリダイレクトする
- VLAN 透過性のために VLAN ID を保持する
- バインドされたサービスの正常性の割合に基づいて自動状態移行を構成する

仮想サーバレベルの低速スタートにより、新しいサービスの負荷を段階的にステップアップ

October 7, 2021

Citrix ADC アプライアンスは、サービスが負荷分散構成に追加された直後、または DOWN から UP への状態が変化した直後に、サービスの負荷（サービスが毎秒受信する要求の数）を徐々に増加するように構成できます（このドキュメントでは、「新しいサービス」という用語は、どちらの状況にも使用されます）。選択した負荷値と間隔（手動スロースタート）で手動で負荷を増やしたり、サービスが設定内の他のサービスと同じ数の要求を受信するまで、指定した間隔（自動スロースタート）で負荷を増やしたりするようにアプライアンスを設定できます。新しいサービスのランプアップ期間中、アプライアンスは設定された負荷分散方式を使用します。

この機能はグローバルには使用できません。仮想サーバごとに構成する必要があります。この機能は、次の負荷分散方法のいずれかを使用する仮想サーバーでのみ使用できます。

- ラウンドロビン
- 最小接続
- 応答時間の最短短短縮
- 最小帯域幅
- 最小パケット
- LRTM（最小応答時間方式）
- カスタムロード

この機能を使用するには、次のパラメータを設定する必要があります。

- 新しいサービスリクエストレート。レートが増加するたびに、新しいサービスに送信されるリクエストの数または割合を増加させる量です。つまり、増分サイズを、1 秒あたりのリクエスト数または既存のサービスによってその時点で負担される負荷の割合で指定します。この値を 0 (ゼロ) に設定すると、新しいサービスではスロースタートは実行されません。

  注: 自動スロースタートモードでは、指定した値が他のサービスよりも新しいサービスに負荷がかかる場合、最終的な増分は指定された値より小さくなります。

- 増分間隔（秒単位）。この値を 0 (ゼロ) に設定すると、ロードは自動的に増加しません。手動でインクリメントする必要があります。

自動スロースタートでは、次のいずれかの条件が適用されると、サービスはスロースタートフェーズから取り出されます。

- 実際の要求レートが、新しいサービス要求レートよりも低くなっています。
- サービスは、連続する 3 つの増分間隔でトラフィックを受信しません。
- 要求レートが 200 回増加しました。
- 新しいサービスが受信する必要があるトラフィックの割合が 100 以上です。

手動のスロースタートでは、サービスはそのフェーズから外れるまで、スロースタートフェーズのままです。

手動スロースタート

新しいサービスの負荷を手動で増やす場合は、負荷分散仮想サーバーの増分間隔を指定しないでください。新しいサービスリクエストレートと単位だけを指定します。間隔を指定しないと、アプライアンスは定期的に負荷を増やしません。いずれかのパラメータを手動で変更するまで、新しいサービスリクエストレートとユニットの組み合わせで指定された値で新しいサービスの負荷が維持されます。たとえば、新しいサービスリクエストレートとユニットパラメータをそれぞれ 25、「per second」に設定した場合、アプライアンスは、いずれかのパラメータを変更するまで、新しいサービスの負荷を 1 秒あたり 25 リクエストで維持します。新しいサービスがスロースタートモードを終了し、既存のサービスと同じ数の要求を受信する場合は、新しいサービスリクエストレートパラメータを 0 に設定します。

たとえば、仮想サーバを使用して、ラウンドロビンモードで Service1 と Service2 の 2 つのサービスをロードバランシングしているとします。さらに、仮想サーバが 1 秒あたり 240 要求を受信し、負荷をサービス間で均等に分散していると仮定します。新しいサービス Service3 が構成に追加されると、負荷の全シェアを送信する前に、1 秒あたり 10、20、および 40 のリクエストの値を使用して手動で負荷を増やすことができます。次の表に、3 つのパラメータを設定する値を示します。

表 1. パラメータ値

パラメーター	値
間隔 (秒)	0
新しいサービス・リクエスト・レート	選択した間隔で 10、20、40、0
新しいサービス要求レートの単位	1 秒あたりのリクエスト数

新しいサービスリクエストのレートパラメータを 0 に設定すると、Service3 は新しいサービスとは見なされなくなり、負荷の完全なシェアを受け取ります。

Service3 のランプアップ期間中に、別のサービス Service4 を追加することを想定しています。この例では、新しいサービスリクエストレートパラメータが 40 に設定されている場合、Service4 が追加されます。したがって、Service4 は 1 秒あたり 40 要求を受信し始めます。

次の表に、この例で説明した期間中のサービスの負荷分散を示します。

表 2. 負荷を手動でステップアップするときのサービスの負荷分散

	新しいサービスリクエストのレート = 10 要求/秒（サービス 3 追加）	新しいサービスリクエストのレート = 20 要求/秒	新しいサービスリクエストのレート = 40 要求/秒（サービス 4 追加）	新しいサービスリクエストレート = 0 要求/秒（新しいサービスはスロースタートモードを終了します）
Service1	115	110	80	60
Service2	115	110	80	60
Service3	10	20	40	60
Service4	-	-	40	60
合計要求/秒（仮想サーバの負荷）	240	240	240	240

自動スロースタート

アプライアンスが新しいサービスの負荷を一定の間隔で自動的に増加させたい場合は、新しいサービスリクエストレートパラメータ、units パラメータ、および増分間隔を設定します。すべてのパラメータが 0 以外の値に設定されている場合、アプライアンスは、指定された間隔で、サービスが負荷の全シェアを受け取るまで、新しいサービスへの負荷を新しいサービスリクエストレートの値だけ増やします。

たとえば、サービス 1、サービス 2、サービス 3、サービス 4 の 4 つのサービスがロードバランシング仮想サーバー vserver1 にバインドされているとします。さらに、vserver1 は 1 秒あたり 100 件の要求を受信し、負荷をサービス全体に均等に分散します (1 サービスあたり 1 秒あたり 25 件の要求)。構成に 5 番目のサービス Service5 を追加すると、アプライアンスは 1 秒あたり 20 件のリクエストを受信するまで、最初の 10 秒間、1 秒あたり 4 件のリクエストを、次の 10 秒間は 1 秒あたり 8 件というように新しいサービスを送信します。この要件については、次の表に、3 つのパラメータを設定する値を示します。

表 3. パラメータ値

パラメーター	値
間隔 (秒)	10
増分値	4
新しいサービス要求レートの単位	1 秒あたりのリクエスト数

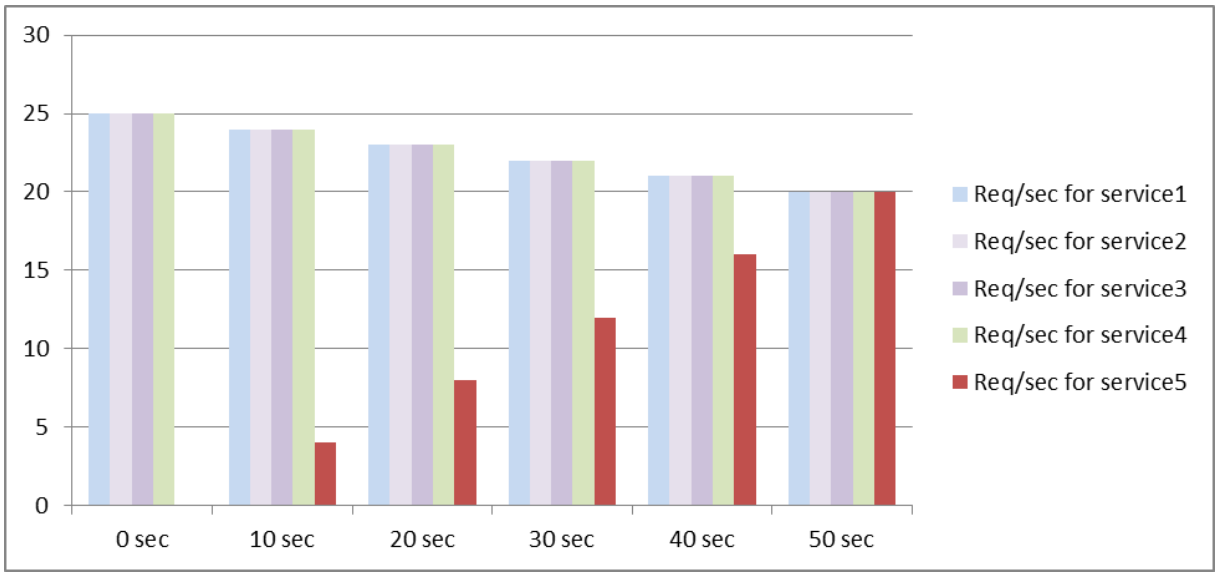
この設定では、新しいサービスは、追加または状態が DOWN から UP に変更されてから 50 秒後に既存のサービス

と同じ数の要求を受信し始めます。この期間の各間隔で、アプライアンスは、段階的な増分がない場合に、新しいサービスに送信された過剰な要求を既存のサーバーに分配します。たとえば、段階的な増分がない場合、Service5 を含む各サービスは 1 秒あたり 20 のリクエストを受信することになります。Service5 が 1 秒あたり 4 リクエストしか受信しない最初の 10 秒間に段階的に増分すると、アプライアンスは 1 秒あたり 16 件を超えるリクエストを既存のサービスに分配します。その結果、次の表と図に示す分散パターンが 50 秒間に渡って生成されます。50 秒が経過すると、Service5 は新しいサービスとは見なされなくなり、トラフィックの通常のシェアを受信します。

表 4. Service5 の追加直後の 50 秒間の全サービスの負荷分散パターン

	0 秒	10 秒	20 秒	30 秒	40 秒	50 秒
サービス **1** の所要時間/秒	25	24	23	22	21	20
サービス **2** の所要時間/秒	25	24	23	22	21	20
サービス **3** の所要時間/秒	25	24	23	22	21	20
サービス **4** の所要時間/秒	25	24	23	22	21	20
サービス **5** の所要時間/秒	0	4	8	12	16	20
合計要求/秒（仮想サーバの負荷）	100	100	100	100	100	100

図 1: Service5 追加直後の 50 秒間の全サービス負荷分散パターンのグラフ



別の要件として、アプライアンスが既存のサービスに対する負荷の Service5 25% を、最初の 5 秒間に 50% を、次の 5 秒間に 50% を、1 秒あたり 20 のリクエストを受信するまで送信するということが挙げられます。この要件については、次の表に、3 つのパラメータを設定する値を示します。

表 5. パラメータ値

パラメーター	値
間隔 (秒)	5
増分値	25
新しいサービス要求レートの単位	パーセント

この設定では、サービスは、追加または状態が DOWN から UP に変更されてから 20 秒後に、既存のサービスと同じ数の要求を受信し始めます。新しいサービスのランプアップ期間中のトラフィック分散は、前述のトラフィック分散と同じです。ステップ単位の単位は「1 秒あたりの要求数」です。

スロースタートパラメーターの設定

スロースタートパラメータは、`set lb vserver`または`add lb vserver`コマンドを使用して設定します。次のコマンドは、仮想サーバーを追加するときにスロースタートパラメータを設定するためのものです。

コマンドラインインターフェイスを使用して新しいサービスの段階的な負荷増分を構成するには

コマンドプロンプトで次のコマンドを入力して、サービスの負荷の段階的な増分を構成し、構成を確認します。

```
add lb vserver <name> <serviceType> <IPAddress> <port> [-
    newServiceRequest <positive_integer>] [<newServiceRequestUnit>] [-
    newServiceRequestIncrementInterval <positive_integer>]

show lb vserver <name>
<!--NeedCopy-->
```

例

```
set lb vserver BR_LB -newServiceRequest 5 PER_SECOND -
    newServiceRequestIncrementInterval 10
Done

```



```
show lb vserver BR_LB
BR_LB (192.0.2.33:80) - HTTP Type: ADDRESS
State: UP
...
...
New Service Startup Request Rate: 5 PER_SECOND, Increment Interval: 10
...
...
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して新しいサービスの段階的な負荷増分を構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で [方法] を選択し、次のスロースタートパラメータを設定します。
   - 新しいサービス起動要求レート。
   - 新しいサービス・リクエスト・ユニット。
   - 増分間隔。

サービスの監視なしオプション

October 7, 2021

外部システムを使用してサービスのヘルスチェックを実行し、Citrix ADC アプライアンスがサービスの正常性を監視しない場合は、サービスに対して監視なしオプションを設定できます。その場合、アプライアンスはサービスの健全性をチェックするためのプローブを送信しませんが、サービスが UP と表示されます。サービスが DOWN になった場合でも、アプライアンスは、ロードバランシング方式で指定されたとおりにクライアントからサービスにトラフィックを送信し続けます。

モニターなしオプションを設定すると、モニターは ENABLED または DISABLED 状態になることがあり、モニターなしオプションを削除すると、モニターの以前の状態が再開されます。

サービスの作成時に、サービスの監視なしオプションを設定できます。また、既存のサービスに no-monitor オプションを設定することもできます。

no-monitor オプションを設定すると、次のような結果になります。

- no-monitor オプションを有効にしたサービスがダウンしても、アプライアンスはサービスを UP として表示し続け、トラフィックをサービスに転送し続けます。サービスへの永続的な接続は、状況を悪化させる可能性があります。その場合、または UP と表示されている多くのサービスが実際に DOWN の場合、システムに障害が発生する可能性があります。このような状況を回避するには、サービスを監視する外部メカニズムがサービスを DOWN と報告するときに、Citrix ADC 構成からサービスを削除します。

- サービスで no-monitor オプションを構成する場合は、ダイレクトサーバーリターン (DSR) モードでロードバランシングを構成できません。既存のサービスに対して、no-monitor オプションを設定した場合、サービスの DSR モードを構成できません。

CLI を使用して新しいサービスの no-monitor オプションを設定するには

コマンドプロンプトで次のコマンドを入力して、ヘルスモニターオプションを使用してサービスを作成し、構成を確認します。

```
add service <serviceName> <IP | serverName> <serviceType> <port> -
   healthMonitor (YES|NO)
<!--NeedCopy-->
```

例:

```
add service nomonsrvc 10.102.21.21 http 80 -healthMonitor no
Done

show service nomonsrvc
nomonsrvc (10.102.21.21:80) - HTTP
State: UP
Last state change was at Mon Nov 15 22:41:29 2010
Time since last state change: 0 days, 00:00:00.970
Server Name: 10.102.21.21
 Server ID : 0 Monitor Threshold : 0
...
Access Down Service: NO
...
Down state flush: ENABLED
Health monitoring: OFF

1 bound monitor:
1) Monitor Name: tcp-default
State: UNKNOWN Weight: 1
Probes: 3 Failed [Total: 3 Current: 3]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: N/A
Done
<!--NeedCopy-->
```


CLI を使用して既存のサービスに **no-monitor** オプションを設定するには

コマンドプロンプトで次のコマンドを入力して、ヘルスモニターオプションを設定します。

```
set service <name> -healthMonitor (YES|NO)
<!--NeedCopy-->
```

例:

```
By default, the state of a service and the state of the corresponding
   monitor are UP.
 >show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP


1) Monitor Name: http-ecv
 State: UP Weight: 1
Probes: 99992 Failed [Total: 0 Current: 0]
Last response: Success - Pattern found in response.
Response Time: 3.76 millisec
Done

When the no-monitor option is set on a service, the state of the
   monitor changes to UNKNOWN.
set service LB-SVC1 -healthMonitor NO
Done

show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010.
Time since last state change: 5 days, 18:55:48.710
Health monitoring: OFF

1) Monitor Name: http-ecv
   State: UNKNOWN Weight: 1
         Probes: 100028 Failed [Total: 0 Current: 0]
   Last response: Probe skipped - Health monitoring is turned off.
     Response Time: 0.0 millisec
  Done
When the no-monitor option is removed, the earlier state of the monitor
    is resumed.
```



```
> set service LB-SVC1 -healthMonitor YES
Done
>show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010
Time since last state change: 5 days, 18:57:47.880
1) Monitor Name: http-ecv
   State: UP Weight: 1
   Probes: 100029 Failed [Total: 0 Current: 0]
   Last response: Success - Pattern found in response.
   Response Time: 5.690 millisec
   Done
<!--NeedCopy-->
```

GUI を使用してサービスの監視なしオプションを設定するには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. サービスを開き、[ヘルスモニタリング] をクリアします。

保護されたサーバ上のアプリケーションをトラフィックの急増から保護する

October 7, 2021

Citrix ADC アプライアンスは、サーバーまたはキャッシュの容量を維持するためのサージ保護オプションを提供します。アプライアンスは、サーバーへのクライアント要求のフローを制御し、サーバーに同時にアクセスできるクライアントの数を制御します。アプライアンスはサーバーに渡されたサージをブロックし、サーバーの過負荷を防ぎます。

サージ保護を正しく機能させるには、サージ保護をグローバルに有効にする必要があります。サージ保護の詳細については、サージ保護を参照してください。

CLI を使用してサービスにサージ保護を設定するには

コマンドプロンプトで入力します。

```
set service <name> -sp <Value>
<!--NeedCopy-->
```

例:


```
set service Service-HTTP-1 -sp ON
<!--NeedCopy-->
```

GUI を使用してサービスのサージ保護を設定するには

1. [トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、ソースを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[サージ保護] を選択します。

仮想サーバーおよびサービス接続のクリーンアップを有効にする

October 7, 2021

仮想サーバの状態は、バインドされているサービスの状態によって異なります。各サービスの状態は、そのサービスにバインドされているモニターによって送信されるプローブまたはヘルスチェックに対する負荷分散されたサーバーの応答によって異なります。負荷分散されたサーバーが応答しないことがあります。サーバが低速またはビジー状態の場合、モニタリングプローブがタイムアウトすることがあります。設定されたタイムアウト期間内に繰り返しモニタリングプローブに応答しない場合、サービスには DOWN とマークされます。サービスまたは仮想サーバーが DOWN とマークされている場合は、サーバーとクライアント側の接続をフラッシュする必要があります。既存の接続を終了すると、リソースが解放され、場合によっては負荷分散セットアップの回復速度が速くなります。

特定の条件下では、サービスまたは仮想サーバーが DOWN とマークされたときに、既存の接続をただちに終了するように **downStateFlush** 設定を構成できます。トランザクションを完了する必要があるアプリケーションサーバーでは、DownStateFlush 設定を有効にしないでください。この設定は、DOWN とマークされたときに接続を安全に終了できる Web サーバー上で有効にできます。

次の表は、仮想サーバー Vserver-LB-1 で構成され、1 つのサービスが Service-1 にバインドされた構成例に対するこの設定の影響をまとめたものです。この表では、E と D は downStateFlush 設定の状態を示します。E は Enabled、D は Disabled です。

Vserver-LB-1	Service-1	接続の状態
E	E	クライアント接続とサーバー接続の両方が終了します。

Vserver-LB-1	Service-1	接続の状態
E	D	TCP などの一部のサービスタイプでは、Citrix ADC アプライアンスが接続の再利用をサポートしていない場合、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプの場合、クライアントとサーバーの両方の接続は、それらの接続でトランザクションがアクティブである場合にのみ終了します。トランザクションがアクティブでない場合、クライアント接続のみが終了します。
D	E	TCP などの一部のサービスタイプでは、Citrix ADC アプライアンスが接続の再利用をサポートしていない場合、クライアント接続とサーバー接続の両方が終了します。アプライアンスが接続の再利用をサポートする HTTP などのサービスタイプの場合、クライアントとサーバーの両方の接続は、それらの接続でトランザクションがアクティブである場合にのみ終了します。トランザクションがアクティブでない場合、サーバー接続のみが終了します。
D	D	クライアント接続もサーバー接続も終了しません。

サーバーまたはクライアントによって確立されたすべての接続が閉じられた場合にのみサービスを無効にする場合は、正常なシャットダウンオプションを使用できます。サービスのグレースフルシャットダウンの詳細については、「サービスのグレースフルシャットダウン」を参照してください。

CLI を使用してサービスでダウン状態のフラッシュを設定するには

コマンドプロンプトで入力します。

```
set service <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -downStateFlush enabled
<!--NeedCopy-->
```

GUI を使用してサービスのダウン状態のフラッシュを設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[ダウン状態フラッシュ] を選択します。

CLI を使用して仮想サーバでダウン状態のフラッシュを設定するには

コマンドプロンプトで入力します。

```
set lb vserver <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

例**:**

```
set lb vserver vsvr1 -downStateFlush enabled
<!--NeedCopy-->
```

GUI を使用して仮想サーバでダウン状態のフラッシュを設定するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[ダウン状態フラッシュ] を選択します。

サービスの正常なシャットダウン

October 7, 2021

システムのアップグレードやハードウェアのメンテナンスなどのスケジュールされたネットワーク停止中に、一部のサービスを終了または無効にする必要があります。後で”enable service<name> コマンドを使用して、サービスを有効にすることができます。

確立されたセッションが中断されないようにするには、次のいずれかの操作を行って、サービスを Transition Out of Service (TROFS) 状態にします。

- モニタへの TROFS コードまたは文字列の追加：モニタプローブへの応答として特定のコードまたは文字列を送信するようにサーバを設定します。
- サービスを明示的に無効にし、次の操作を行います。
  - 遅延を秒単位で設定します。
  - グレースフルシャットダウンを有効にします。

TROFS コードまたは文字列の追加

1 つのサービスにモニタを 1 つだけバインドし、そのモニタが TROFS 対応の場合、モニタプローブに対するサーバの応答に基づいて、サービスを TROFS 状態にすることができます。この応答は、HTTP モニターの場合は trofsCode パラメーターまたは HTTP-ECV モニターまたは TCP-ECV モニターの場合は trofsString パラメーター内の値と比較されます。コードが一致すると、サービスは TROFS 状態になります。この状態では、永続的な接続を維持し続けます。

複数のモニターがサービスにバインドされている場合、サービスの有効状態は、サービスにバインドされているすべてのモニターの状態に基づいて計算されます。TROFS 応答を受信すると、この計算のために TROFS 対応モニターの状態が UP と見なされます。Citrix ADC アプライアンスがサービスを UP として指定する方法の詳細については、「サービスにバインドされたモニターのしきい値を設定する」を参照してください。

重要:

- 1 つのサービスに複数のモニタをバインドできますが、複数のモニタを TROFS で有効にすることはできません。
- TROFS 対応モニターを TROFS 対応でないモニターに変換することはできますが、その逆はできません。

コマンドラインインターフェイスを使用してモニタで **TROFS** コードまたは文字列を構成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add lb monitor <monitor-name> HTTP -trofsCode <respcode>
```



```
add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>

add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **TROFS** コードまたは文字列を変更するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
set lb monitor <trofs monitorname> HTTP -trofscode <newcode>

set lb monitor <trofs monitorname> HTTP-ECV -trofsstring <new string>

set lb monitor <trofs monitorname> TCP-ECV -trofsstring <new string>
<!--NeedCopy-->
```

注：set コマンドは、TROFS 対応モニターが以前に追加された場合にのみ使用できます。このコマンドを使用して、TROFS が有効でないモニタの TROFS コードまたは文字列を設定することはできません。

構成ユーティリティを使用してモニターで **TROFS** コードまたは文字列を構成するには

1. Traffic Management > Load Balancing > Monitors に移動します。
2. [モニター] ウィンドウで、[追加] をクリックし、次のいずれかの操作を行います。
   - 「HTTP」としてタイプを選択し、「TROFS コード」を指定します。
   - 「タイプ」として「HTTP-ECV」または「TCP-ECV」を選択し、「TROFS 文字列」を指定します。

サービスの無効化

ただし、多くの場合、サービスへのすべての接続が既存のトランザクションを完了するために必要な時間を推定することはできません。待機時間が経過したときにトランザクションが完了していない場合、サービスをシャットダウンすると、データが失われる可能性があります。この場合、サービスに対して正常なシャットダウンを指定できます。これにより、現在アクティブなすべてのクライアント接続がサーバーまたはクライアントによって閉じられた場合にのみサービスが無効になります。正常なシャットダウンに加えて待機時間を指定した場合の動作については、次の表を参照してください。

正常なシャットダウンを有効にしても、指定された方法に従って永続性が維持されます。モニタによるチェックの結果として正常なシャットダウン状態の間にサービスが DOWN とマークされない限り、システムは、クライアントからの新しい接続を含む、すべての永続的なクライアントに対して引き続きサービスを提供します。

次の表では、グレースフルシャットダウンオプションについて説明します。

State	結果
正常なシャットダウンが有効になり、待機時間が指定されます。	待機時間が経過していない場合でも、現在アクティブなクライアントの最後の接続が処理された後、サービスはシャットダウンされます。アプライアンスは毎秒 1 回、接続のステータスをチェックします。待機時間が経過すると、開いているセッションはすべて閉じられます。
正常なシャットダウンは無効になり、待機時間が指定されます。	サービスがシャットダウンされるのは、待機時間が経過した後で、確立されたすべての接続が期限切れになる前に提供されている場合のみです。
正常なシャットダウンは有効で、待機時間は指定されません。	最後の接続の処理にかかった時間に関係なく、以前に確立された最後の接続が処理された後にのみ、サービスがシャットダウンされます。
正常なシャットダウンは無効で、待機時間は指定されません。	正常なシャットダウンはありません。サービスは、disable オプションが選択されるか、disable コマンドが発行された直後にシャットダウンされます。(デフォルトの待機時間は 0 秒です)。

サービスまたは仮想サーバーが DOWN とマークされている場合に既存の接続を終了するには、[Down State Flush] オプションを使用します。詳細については、「仮想サーバー接続のクリーンアップの有効化」を参照してください。

コマンドラインインターフェイスを使用してサービスの正常なシャットダウンを構成するには

コマンドプロンプトで次のコマンドを入力して、サービスを正常にシャットダウンし、構成を確認します。

```
disable service <name> [<delay>] [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

例:

```
> disable service svc1 6000 -graceFul YES
Done
>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: GOING OUT OF SERVICE (Graceful, Out Of Service in 5998 seconds)
```



```
Last state change was at Mon Nov 15 22:44:15 2010
Time since last state change: 0 days, 00:00:01.160
...
Down state flush: ENABLED

1 bound monitor:
1) Monitor Name: tcp-default
State: UP              Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - live traffic to service.
Response Time: N/A
Done

>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: OUT OF SERVICE
Last state change was at Mon Nov 15 22:44:19 2010
Time since last state change: 0 days, 00:00:03.250
Down state flush: ENABLED

1 bound monitor:
1)  Monitor Name: tcp-default
State: UNKNOWN           Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - service state OFS.
Response Time: N/A
Done
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスの正常なシャットダウンを構成するには

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。
2. サービスを開き、[アクション] リストから [無効にする] をクリックします。待機時間を入力し、[Graceful] を選択します。

TROFS サービスでの持続性セッションの有効化または無効化

October 7, 2021

TrofsPersistence フラグを設定して、トランジションアウトオブサービス (TROFS) 状態のサービスが永続セッションを維持する必要があるかどうかを指定できます。モニタが TROFS が有効な場合、モニタプローブに対する

サーバの応答に基づいて、サービスを TROFS 状態にすることができます。この応答は、HTTP モニターの場合は trofsCode パラメーターまたは HTTP-ECV モニターまたは TCP-ECV モニターの場合は trofsString パラメーター内の値と比較されます。コードが一致すると、サービスは TROFS 状態になります。この状態では、アクティブなクライアント接続を優先し続けます。場合によっては、有効なアクティブなセッションに永続的なセッションを含める必要があります。ただし、その他の場合、特に長寿命の永続セッションやカスタムサーバー ID などの永続性メソッドを含む場合、永続セッションを尊重すると、サービスがアウトオブサービス状態に移行するのを防ぐことができます。

trofsPersistence フラグを ENABLED に設定すると、永続セッションが優先されます。[DISABLED] に設定すると、無効になります。

コマンドラインインターフェイスを使用して **trofsPersistence** フラグを設定するには

コマンドプロンプトで、次のコマンドのいずれかを入力して、新しい仮想サーバーまたは既存の仮想サーバーの`trofsPersistence` フラグを設定するか、設定を既定値に戻します。

```
add lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

set lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

unset lb vserver <name> [-trofsPersistence]
<!--NeedCopy-->
```

引数

trofsPersistence。サービスが TROFS 状態の場合、永続性セッションで現在アクティブなクライアント接続と新しい要求を尊重します。

設定可能な値：ENABLED, DISABLED。デフォルト：ENABLED。

例:

```
add lb vserver v1 http 10.102.217.42 80 -persistencetype SOURCEIP -
    trofsPersistence ENABLED

set lb vserver v1 -trofsPersistence DISABLED

unset lb vserver v1 -trofsPersistence
<!--NeedCopy-->
```


カスタム **Web** ページへのリクエストの直接作成

December 7, 2021

> 警告
>
> SureConnect（SC）は NetScaler 12.0 ビルド 56.20 以降から廃止され、別の方法として、AppQoE 機能を使用することをお勧めします。詳細については、AppQoEを参照してください。

SureConnect が正しく機能するには、グローバルに設定する必要があります。Citrix ADC には、アプリケーションからの応答を確実にする SureConnect オプションが用意されています。
SureConnect オプションの詳細については、「確実接続」を参照してください。

CLI を使用してサービスに **SureConnect** を設定するには

コマンドプロンプトで入力します。

```
set service <name> -sc <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -sc ON
<!--NeedCopy-->
```

GUI を使用してサービスに **SureConnect** を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[確実接続] を選択します。

ダウン時にサービスへのアクセスを有効にする

October 7, 2021

サービスが無効または DOWN 状態のときにサービスへのアクセスを有効にするには、レイヤー 2 モードを使用してサービスに送信されたパケットをブリッジするように Citrix ADC アプライアンスを構成します。通常、DOWN のサービスに要求が転送されると、要求パケットはドロップされます。ただし、[**Access Down**] 設定を有効にすると、これらの要求パケットは負荷分散されたサーバに直接送信されます。

レイヤ 2 モードおよびレイヤ 3 モードの詳細については、IP アドレッシングを参照してください。

アプライアンスが DOWN サービスに送信されたパケットをブリッジするには、accessDown パラメータを使用してレイヤ 2 モードを有効にします。

CLI を使用してサービスのアクセスを有効にするには

コマンドプロンプトで入力します。

```
set service <name> -accessDown <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -accessDown YES
<!--NeedCopy-->
```

GUI を使用してサービスのアクセスを有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[アクセスダウン] を選択します。

応答の TCP バッファリングを有効にします

October 7, 2021

Citrix ADC アプライアンスは、負荷分散されたサーバーからの応答のみをバッファリングする TCP バッファリングオプションを提供します。これにより、アプライアンスは、クライアントが受け入れることができる最大速度でクライアントへのサーバー応答を配信できます。アプライアンスは、TCP バッファリングに 0 ～4095 MB（MB）のメモリを割り当てて、接続ごとに 4 ～20480 キロバイト（KB）のメモリを割り当てます。

注：サービスレベルで設定された TCP バッファリングは、グローバル設定よりも優先されます。
TCP バッファリングのグローバル設定の詳細については、TCP バッファリングを参照してください。

CLI を使用してサービスで TCP バッファリングを有効にするには

コマンドプロンプトで入力します。


```
set service <name> -TCPB <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -TCPB YES
<!--NeedCopy-->
```

GUI を使用してサービスで TCP バッファリングを有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[**TCP** バッファリング] を選択します。

圧縮を有効化

October 7, 2021

Citrix ADC アプライアンスには、組み込みの圧縮ポリシーのセットを使用して、HTML ファイルとテキストファイルを透過的に圧縮するための圧縮オプションがあります。圧縮により、帯域幅の要件が軽減され、帯域幅に制約があるセットアップでのサーバの応答性が大幅に向上します。圧縮ポリシーは、仮想サーバにバインドされたサービスに関連付けられます。ポリシーは、応答を圧縮できるかどうかを決定し、圧縮可能なコンテンツをアプライアンスに送信します。アプライアンスは圧縮してクライアントに送信します。

注: 圧縮を正しく機能させるには、グローバルに有効にする必要があります。圧縮をグローバルに設定する方法の詳細については、「圧縮」を参照してください。

CLI を使用して特定のサービスの圧縮を有効にするには

コマンドプロンプトで入力します。

```
set service <name> -CMP <YES | NO>
<!--NeedCopy-->
```

例:


```
set service Service-HTTP-1 -CMP YES
<!--NeedCopy-->
```

GUI を使用して特定のサービスの圧縮を有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[圧縮] を選択します。

UDP 仮想サーバーの外部 **TCP** ヘルスチェックを有効にする

October 7, 2021

パブリッククラウドでは、ネイティブロードバランサーを第 1 層として使用する場合、Citrix ADC アプライアンスを第 2 層ロードバランサーとして使用できます。ネイティブロードバランサーは、アプリケーションロードバランサー (ALB) またはネットワークロードバランサー (NLB) です。パブリッククラウドのほとんどは、ネイティブのロードバランサで UDP 正常性プローブをサポートしていません。UDP アプリケーションの状態を監視するために、パブリッククラウドでは、サービスに TCP ベースのエンドポイントを追加することをお勧めします。エンドポイントは UDP アプリケーションの正常性を反映します。

Citrix ADC アプライアンスは、UDP 仮想サーバーの外部 TCP ベースのヘルスチェックをサポートします。この機能により、仮想サーバーの VIP と構成済みポートに TCP リスナーが導入されます。TCP リスナーは、仮想サーバーのステータスを反映します。

CLI で **UDP** 仮想サーバの外部 **TCP** ヘルスチェックを有効にするには

コマンドプロンプトで次のコマンドを入力して、TcpProbeport オプションを指定して外部 TCP ヘルスチェックを有効にします。

```
add lb vserver <name> <serviceType> <IPAddress> <port> -tcpProbePort <
    tcpProbePort>
<!--NeedCopy-->
```

例**:**

```
add lb vserver Vserver-UDP-1 UDP 10.102.29.60 80 tcpProbePort 5000
<!--NeedCopy-->
```


GUI で UDP 仮想サーバの外部 TCP ヘルスチェックを有効にするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを作成します。
2. [**Add**] をクリックして、仮想サーバーを作成します。
3. [基本設定] ペインで、[**TCP** プローブポート] フィールドにポート番号を追加します。
4. [**OK**] をクリックします。

複数のクライアント要求に対してクライアント接続を維持する

October 7, 2021

クライアントキープアライブパラメータを設定して、HTTP または SSL サービスを構成して、Web サイトへのクライアント接続を複数のクライアント要求にわたって開いたままにできます。クライアントのキープアライブが有効になっている場合、負荷分散された Web サーバーが接続を閉じた場合でも、Citrix ADC アプライアンスはクライアントとそれ自体の間の接続を開いたままにします。この設定では、サービスは 1 つのクライアント接続で複数のクライアント要求を処理できます。

この設定を有効にしない場合、クライアントは Web サイトに送信するリクエストごとに新しい接続を開きます。クライアントのキープアライブ設定により、接続の確立と終了に必要なパケットの往復時間が節約されます。この設定により、各トランザクションが完了するまでの時間も短縮されます。クライアントのキープアライブは、HTTP または SSL サービスタイプでのみ有効にできます。

サービスレベルで設定されたクライアントキープアライブは、グローバルクライアントキープアライブ設定よりも優先されます。クライアントキープアライブの詳細については、「クライアントキープアライブ」を参照してください。

CLI を使用してサービスでクライアントキープアライブを有効にするには

コマンドプロンプトで入力します。

```
set service <name> -CKA <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -CKA YES
<!--NeedCopy-->
```


GUI を使用してサービスでクライアントキープアライブを有効にするには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[クライアントキープアライブ] を選択します。

リクエストヘッダーにクライアントの **IP** アドレスを挿入します

October 7, 2021

Citrix ADC は、サブネット IP（SNIP）アドレスを使用してサーバーに接続します。サーバーはクライアントを認識する必要はありません。

ただし、状況によっては、サーバーがサービスを提供する必要があるクライアントを認識する必要があります。クライアントの IP 設定を有効にすると、アプライアンスはクライアントの IPv4 または IPv6 アドレスを挿入し、要求をサーバーに転送します。サーバは、このクライアント IP を応答のヘッダーに挿入します。したがって、サーバーはクライアントを認識します。

> 注: 複数のヘッダーを挿入するには、次のいずれかを実行する必要があります。
>
> - CLIENT.IS_SSL をチェックし、適切なヘッダーを挿入する書き換えポリシーを追加します。
> - タイプに基づいて、各仮想サーバに適切な書き換えポリシーをバインドします。

CLI を使用してクライアント要求にクライアント **IP** アドレスを挿入するには

コマンドプロンプトで入力します。

```
set service <name> -CIP <Value> <cipHeader>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -CIP enabled X-Forwarded-For
<!--NeedCopy-->
```

GUI を使用してクライアント要求にクライアント **IP** アドレスを挿入するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを編集します。
2. **[**サービス設定**]** ペインで、編集アイコンをクリックします。
3. [負荷分散サービス**]** ペインで、[クライアント **IP** アドレスの挿入] チェックボックスをオンにします。

ジオロケーションデータベースを使用してユーザーの IP アドレスから場所の詳細を取得する

October 7, 2021

> 注この機能は、Citrix ADC リリース 12.1 ビルド 50.x 以降で使用できます。

Citrix ADC アプライアンスは、大陸、郡、市などのユーザーの場所の詳細を取得できます。地理的位置データベースからの任意のパブリック IP アドレス。これは、高度なポリシーインフラストラクチャを使用して実行されます。取り出された場所の詳細は、次のユースケースを実行するための書き換えアクションまたは応答側アクションで使用されます。

- クライアント要求をバックエンドサーバーに送信するときに、ユーザーの場所の詳細（国、都市情報など）を含む HTTP ヘッダーを挿入します。
- 無効なユーザーの HTML ページ応答に国名を追加します。

アプライアンスは、監査ロギングメカニズムを使用してロケーションの詳細を記録することもできます。

ジオロケーション関数を使用したユーザーの位置情報の取得

コンポーネントは、次のように相互作用します。

1. ユーザーは、特定の地理的な場所からクライアント要求を送信します。
2. Citrix ADC アプライアンスは、クライアント要求からユーザー IP アドレスを探し、地理的位置の詳細を取得します。詳細には、大陸、国、地域、都市、ISP、組織、またはジオロケーションデータベースのカスタム詳細が含まれます。
3. 場所の詳細を取得すると、アプライアンスは応答側ポリシーまたは書き換えポリシーのいずれかを使用して要求を評価します。
4. 書き換えポリシーでは、アプライアンスは地理的位置の詳細を含むヘッダーを追加し、それをバックエンドサーバーに送信します。たとえば、国情報を含むカスタム HTTP ヘッダーを挿入します。
5. レスポンダーポリシーでは、アプライアンスは HTTP 要求を評価し、ポリシー評価に基づいて、ユーザーへのアクセスを許可するか、ユーザーをエラーページにリダイレクトします。これは、アプリケーションにアクセスしている地域がアクセスできないことを示しています。

ジオロケーションデータベースの設定

前提条件として、Citrix ADC アプライアンスで実行するジオロケーションデータベースが必要です。ジオロケーションデータベースファイルは、Citrix ADC ファームウェアで使用できます。ベンダーからデータベースファイルをダウンロードするには、Citrix ADC 形式に変換してアプライアンスにインポートします。
位置情報データベースの詳細については、「静的近接データベースを作成するためのロケーションファイルの追加」トピックを参照してください。

ジオロケーション関数

次の表は、任意のパブリック IP アドレスの場所の詳細を取得するジオロケーション関数のリストです。これらの関数は、書き換えポリシーまたはレスポンダーポリシーで使用できます。

位置情報関数	例
CLIENT.IP.SRC.LOCATION	Asia.In.Karnataka.Bangalore
CLIENT.IP.SRC.LOCATION.GET (1).LOCATION_LONG	India
CLIENT.IP.SRC.LOCATION(3)	Asia.In.Karnataka
CLIENT.IP.SRC.LAT_LONG	12,77
CLIENT.IPV6.SRC.LOCATION	North America.US.California.Santa Clara.Verizon.Citrix
CLIENT.IPV6.SRC.LOCATION(3)	North America.US.California
CLIENT.IPV6.SRC.LOCATION.GET(1).LOCATION_L(	米国
CLIENT.IPV6.SRC.LOCATION.GET(3)	California
CLIENT.IPV6.SRC.LAT_LONG	36, -119

位置情報関数の設定

高度なポリシーインフラストラクチャを使用してジオロケーション機能を設定するには、ロードバランシング、リライト、およびレスポンダー機能を有効にしてから、次のユースケースを完了する必要があります。

負荷分散、レスポンダー、書き換え機能の有効化

Citrix ADC アプライアンスで特定の地理的位置からのユーザーアクセスを許可する場合は、負荷分散、書き換え、およびレスポンダー機能を有効にする必要があります。

```
1 enable ns feature loadbalancing rewrite responder
2 <!--NeedCopy-->
```

ユースケース **1**: 地理的位置以外の無効なユーザーをリダイレクトするためのジオロケーション機能の構成

インドのユーザーがウェブページへのアクセスを要求している場合、リクエストをブロックし、国名の HTML ページで応答します。

次の手順は、このユースケースの構成を完了するのに役立ちます。

- 応答側アクションの追加
- レスポンダーポリシーの追加
- レスポンダーポリシーをロードバランシングサーバーにバインドする

書き換えアクションと書き換えポリシー構成の GUI 手順の詳細については、Responder のトピックを参照してください。

応答側アクションの追加

国名の HTML ページで応答するレスポンダーアクションを追加します。
コマンドプロンプトで入力します。

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>][-reasonPhrase <
    string>]
<!--NeedCopy-->
```

例:

```
add responder action responder_act respondwith "HTTP.REQ.VERSION + "
    304 Requested Page not allowed in your country - " + CLIENT.IP.SRC.
    LOCATION.GET (1).LOCATION_LONG + "\r\n""
<!--NeedCopy-->
```

監査ログメッセージアクションの追加

syslog 形式のみ、または syslog とnewnslog形式の両方で、さまざまなログレベルでメッセージをログに記録するように監査メッセージアクションを設定できます。監査メッセージアクションでは、式を使用して監査メッセージの形式を指定します。
コマンドラインインターフェイスを使用して監査メッセージアクションを作成するには

コマンドプロンプトで入力します。

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-logtoNewnslog
 (YES|NO)] [-bypassSafetyCheck (YES|NO)]
```

例:

```
add audit messageaction msg1 DEBUG ""Request Location: "+CLIENT.IP.SRC.
    LOCATION"
<!--NeedCopy-->
```


レスポンダーポリシーの追加

インドからのリクエストを識別するレスポンダーポリシーを追加し、レスポンダーアクションをこのポリシーに関連付けます。
コマンドプロンプトで入力します。

```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```

例:

```
1 add responder policy responder_pol CLIENT.IP.SRC.MATCHES_LOCATION("Asia
     .India.\*.\*.\*.\*") responder_act -logaction msg1
2 <!--NeedCopy-->
```

レスポンダーポリシーをロードバランシングサーバーにバインドする

レスポンダーポリシーを HTTP/SSL タイプの負荷分散仮想サーバーにバインドします。
コマンドプロンプトで入力します。

```
1 bind lb vserver <vserver name> -policyName < policy_name >  -priority
     <> -type <L7InlineREQUEST | L4Inline-REQUEST>
2 <!--NeedCopy-->
```

例:

```
1 bind lb vserver http_vserver -policyName responder_pol -priority 100 -
     type REQUEST
2 <!--NeedCopy-->
```

ユースケース **2**: バックエンドが応答するための場所の詳細を持つ新しい **HTTP** ヘッダーを挿入するためのジオロケーション関数の設定

サーバーがビジネスロジックの情報を使用できるように、Citrix ADC アプライアンスがアプリケーションサーバーに送信される要求の HTTP ヘッダーにユーザーの場所を挿入する必要があるシナリオを考えてみます。
次の手順は、このユースケースの構成を完了するのに役立ちます。

- 書き換えアクションを追加
- 書き換えポリシーの追加
- リライト・ポリシーをロード・バランシングにバインドする

書き換えアクションおよび書き換えポリシー構成の GUI 手順の詳細については、Responder のトピックを参照してください。

書き換えアクションを追加

リライトアクションを追加して、リクエストにユーザーの地理位置情報の詳細を含むカスタム HTTP ヘッダーを挿入し、バックエンドサーバーを送信します。
コマンドプロンプトで入力します。

```
1 add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
     pattern <expression> | -search <expression>] [-refineSearch <string
     >][-comment <string>]
2 <!--NeedCopy-->
```

例:

```
1 add rewrite action rewrite_act insert_http_header ”User_location”
     CLIENT.IP.SRC.LOCATION
2 <!--NeedCopy-->
```

書き換えポリシーの追加

書き換えポリシーを追加して、書き換えアクションを実行する必要があるかどうかを評価します。この場合、アプリケーションサーバーに送信されるすべての要求にはカスタム HTTP ヘッダーが必要です。したがって、ルールは「true」になります。
コマンドプロンプトで入力します。

```
1 add rewrite policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>]
2 <!--NeedCopy-->
```

例:


```
add rewrite policy rewrite_pol true rewrite_act -logaction log_act
<!--NeedCopy-->
```

リライト・ポリシーをロード・バランシングにバインドする

書き換えポリシーを、HTTP/SSL タイプの必要な負荷分散仮想サーバーにバインドします。
コマンドプロンプトで入力します。

```
bind lb vserver <vserver name> -policyName < policy_name >  -priority
    <> -type <L7InlineREQUEST | L4Inline-REQUEST>
<!--NeedCopy-->
```

例:

```
bind lb vserver http_vserver -policyName rewrite_pol -priority 100 -
    type REQUEST
<!--NeedCopy-->
```

ジオロケーションの詳細をロギングするための **Syslog** サポート（オプション）

ユーザーの地理位置情報の詳細をログに記録する場合は、要求がポリシーに一致したときに実行される SYSLOG アクションを指定する必要があります。アプライアンスは、詳細をログ・メッセージとして ns.log ファイルに保存します。
SYSLOG および NSLOG 監査の詳細については、監査ログのトピックを参照してください。

ユーザーの地理位置情報の詳細の出力

次の出力は、バンガロールの場所からアプリケーションにアクセスしようとして、アプライアンスがジオロケーション機能「CLIENT.IP.SRC.LOCATION」を使用する場合、SYSLOG またはnewnslogアクションを使用してアプライアンスに記録されます。

```
Asia.India.Karnataka.Banglore
<!--NeedCopy-->
```

出力ログの例:


```
07/23/2018:19:03:54 GMT  Debug 0-PPE-0 : default REWRITE Message 22 0 :
     ”Request Location: asia.in.karnataka.bangalore.\*.\*”
07/23/2018:19:23:55 GMT  Debug 0-PPE-0 : default RESPONDER Message 32 0
Done
<!--NeedCopy-->
```

サーバーへの接続時にクライアントの送信元 IP アドレスを使用する

October 7, 2021

送信元 IP アドレスを変更せずに、クライアントからサーバーにパケットを転送するように Citrix ADC アプライアンスを構成できます。これは、HTTP 以外のサービスを使用する場合など、クライアントの IP アドレスをヘッダーに挿入できない場合に便利です。

USIP をグローバルに設定する方法の詳細については、「送信元 IP モードの使用の有効化」を参照してください。

CLI を使用してサービスの USIP モードを有効にするには

コマンドプロンプトで入力します。

```
set service <name> -usip (YES | NO)
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -usip YES
<!--NeedCopy-->
```

GUI を使用してサービスの USIP モードを有効にするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを開きます。
2. [詳細設定] の [サービス設定] セクションで、[送信元 IP アドレスの使用] を選択します。

v4-v6 ロードバランシング構成でのバックエンド通信にクライアント送信元 IP アドレスを使用する

October 7, 2021

v4 から v6 への負荷分散構成では、USIP が無効になっているサービスの場合、Citrix ADC アプライアンスは、構成済みの IPv6 SNIP（SNIP6）アドレスのいずれかから関連サーバーと通信します。

USIP が有効なサービスでは、グローバルな USIP NAT プレフィクスパラメータを設定して、関連するサーバが要求パケットのクライアントの IP アドレスを認識できるようにする必要があります。USIP NAT プレフィックスは、Citrix ADC アプライアンス上で構成された長さ 32/40/48/56/64/96 ビットのグローバル IPv6 プレフィックスです。

USIP が有効な負荷分散サービスの場合、アプライアンスは IPv4 要求パケットを IPv6 パケットに変換し、変換された IPv6 パケットの送信元 IP アドレスを次の連結に設定します。

- 32/40/48/56/64/96 ビットの長さの USIP NAT プレフィクス。
- USIP NAT プレフィクス長が 96 ビット未満の場合、ゼロが埋められます。ゼロで埋められたビット数 = 96 USIP NAT プレフィクス長。たとえば、USIP NAT プレフィクス長が 64 の場合、ゼロで埋められるビット数は 96-64 = 32 です。
- 要求パケットで受信された IPv4 送信元アドレス [32 ビット] 。つまり、送信元 IPv6 アドレスの最後の 32 ビットは、クライアントの IPv4 アドレスに設定されます。

サーバーから IPv6 応答パケットを受信すると、Citrix ADC アプライアンスは IPv6 パケットを IPv4 パケットに変換し、変換された IPv4 パケットの宛先 IP アドレスを IPv6 パケットの宛先 IP アドレスの最後の 32 ビットに設定します。

> 注：この機能は、Citrix Gateway 構成およびコンテンツスイッチングおよびキャッシュリダイレクトの負荷分散構成ではサポートされません。

構成の手順

v4-v6 へのロードバランシング設定の USIP の設定は、次の作業で構成されます。

- グローバル **USIP NAT** プレフィクスを追加します。これは、アプライアンスで設定される長さ 32/40/48/56/64/96 ビットのグローバル IPv6 プレフィックスです。
- グローバル **USIP** モードを有効にします。詳細については、「送信元 IP モードの使用を有効にする」を参照してください。
- 負荷分散サービスの **USIP** モードを有効にします。詳細については、「サーバーに接続するときにクライアントの送信元 IP アドレスを使用する」を参照してください。

CLI を使用してグローバル **USIP NAT** プレフィクスを追加するには、次の手順を実行します。

- `set ipv6 -usipnatprefix <prefix/prefix_length>`

- show ipv6

GUI を使用してグローバル **USIP NAT** プレフィクスを追加するには、次の手順を実行します。

1. [システム] > [ネットワーク] に移動し、[**IPv6** 設定の変更] をクリックします。
2. [**IPV6** の構成を構成] 画面で、**USIP NAT** プレフィクスパラメータを設定します。

構成例

```
> set ipv6 -usipnatprefix 2001:DB8:90::/64
Done

> enable ns mode USIP
Done

> add lb vserver LBVS-1 HTTP 203.0.113.90 80
Done

> add service SVC-1 2001:DB8:5001::30 HTTP 80 -usip yes
Done

> add service SVC-2 2001:DB8:5001::60 HTTP 80 -usip yes
Done

> bind lb vserver LBVS-1 SVC-1
Done

> bind lb vserver LBVS-1 SVC-2
Done

<!--NeedCopy-->
```

サーバー側接続用の送信元ポートの構成

October 7, 2021

Citrix ADC アプライアンスが物理サーバーに接続するときに、クライアントのリクエストからのソースポートを使用するか、接続のソースポートとしてプロキシポートを使用できます。Use Proxy Port パラメーターを YES に設定すると、次のような状況を処理できます。

- Citrix ADC アプライアンスは、LBVS1 と LBVS2 の 2 つの負荷分散仮想サーバーで構成されます。
- 両方の仮想サーバは、同じサービス S-ANY にバインドされます。

- サービスで (クライアントの) 送信元 IP アドレス (USIP) を使用する。
- クライアント C1 は、同じサービスに対して、Req1 と Req2 の 2 つの要求を送信します。
- LBVS1 は Req1 を受信し、LBVS2 は Req2 を受信します。
- LBVS1 と LBVS2 は要求を S-ANY に転送し、S-ANY が応答を送信すると、LBVS1 と LBVS2 は応答をクライアントに転送します。
- 次の 2 つのケースを検討します。
  - クライアントポートを使用します。アプライアンスがクライアント・ポートを使用する場合、仮想サーバーは両方ともクライアントの IP アドレス（USIP が ON のため）とサーバーへの接続時にクライアントのポートを使用します。したがって、サービスが応答を送信すると、アプライアンスは応答を受信する必要がある仮想サーバを特定できません。
  - プロキシポートを使用します。アプライアンスがプロキシ・ポートを使用する場合、仮想サーバーはクライアントの IP アドレスを使用します（USIP が ON であるため）。ただし、サーバーへの接続時にはポートは異なります。したがって、サービスが応答を送信すると、ポート番号は応答を受信する必要がある仮想サーバーを識別します。

ただし、完全に透過的なキャッシュのリダイレクト構成など、完全に透過的な構成が必要な場合は、Citrix ADC アプライアンスがクライアントの要求からソースポートを使用できるように、プロキシポート設定の使用を無効にする必要があります。

[Use Proxy Port] オプションは、[Use source IP (USIP)] オプションが有効になっている場合に関連します。TCP、HTTP、SSL などの TCP ベースのサービスタイプの場合、このオプションはデフォルトで有効になっています。UDP や DNS など ANY を含む UDP ベースのサービスタイプの場合、このオプションはデフォルトで無効になっています。USIP オプションの詳細については、「ソース IP モードの使用を有効にする」を参照してください。

プロキシポートの使用（**Use Proxy Port** ）設定は、グローバルに、または特定のサービスで設定できます。

サービスのプロキシポートの使用設定を構成する

グローバル設定を上書きする場合は、サービスの [プロキシポートを使用] 設定を構成します。

CLI を使用してサービスの **[**プロキシポートを使用**]** 設定を構成するには

コマンドプロンプトで入力します。

```
set service <name> -useProxyPort (YES | NO)
<!--NeedCopy-->
```

例:

```
set service svc1 -useproxyport YES
```



```
Done

show service svc1
svc1 (10.102.29.30:80) - HTTP
State: UP
. . .
Use Source IP: YES Use Proxy Port: YES
. . .
Done
<!--NeedCopy-->
```

GUI を使用してサービスの **[**プロキシポートを使用**]** 設定を構成するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[プロキシポートを使用] を選択します。

プロキシポートの使用設定をグローバルに構成する

Citrix ADC アプライアンス上のすべてのサービスに設定を適用する場合は、［プロキシポートを使用する］設定をグローバルに構成します。サービス固有の [プロキシポートを使用] 設定は、グローバル設定よりも優先されます。

CLI を使用して **[**プロキシポートを使用**]** 設定をグローバルに設定するには

コマンドプロンプトで次のコマンドを入力して、[プロキシポートを使用] 設定をグローバルに設定し、構成を確認します。

```
set ns param -useproxyport ( ENABLED | DISABLED )`
show ns param`
<!--NeedCopy-->
```

例**:**

```
set ns param -useproxyport ENABLED

Done

show ns param
Global configuration settings:
. . .
Use Proxy Port: ENABLED
```



```
Done
<!--NeedCopy-->
```

GUI を使用して **[**プロキシポートを使用**]** 設定をグローバルに設定するには

[システム**] > [**設定**] > [**グローバルシステム設定の変更] に移動し、[プロキシポートを使用] を選択または選択解除します。

クライアント接続数の制限を設定する

October 7, 2021

各負荷分散サーバーが処理できるクライアント接続の最大数を指定できます。Citrix ADC アプライアンスは、この制限に達するまでサーバーへのクライアント接続を開きます。負荷分散されたサーバーが限界に達すると、監視プローブはスキップされ、既存の接続の処理が完了して容量が解放されるまで、サーバーは負荷分散に使用されません。

最大クライアント設定の詳細については、「ドメイン名ベースのサービスの負荷分散」を参照してください。

注: クローズ処理中の接続は、この制限では考慮されません。

CLI を使用してクライアント接続数の制限を設定するには

コマンドプロンプトで入力します。

```
set service <name> -maxclient <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -maxClient 1000
<!--NeedCopy-->
```

GUI を使用してクライアント接続数の制限を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[最大クライアント数] を選択します。

サーバーへの接続あたりの要求数の制限を設定する

October 7, 2021

Citrix ADC アプライアンスは、接続を再利用してパフォーマンスを向上させるように構成できます。ただし、一部のシナリオでは、負荷分散された Web サーバーで多数の要求に対して接続が再利用されるときに問題が発生することがあります。HTTP サービスまたは SSL サービスの場合は、max request オプションを使用して、負荷分散された Web サーバーへの 1 つの接続を介して送信される要求の数を制限します。

注：最大要求オプションは、HTTP サービスまたは SSL サービスに対してのみ設定できます。

CLI を使用して接続あたりのクライアント要求数を制限するには

コマンドプロンプトで入力します。

```
set service <ServiceName> -maxReq <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -maxReq 100
<!--NeedCopy-->
```

GUI を使用して接続あたりのクライアント要求数を制限するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[最大要求数] を選択します。

サービスにバインドされたモニターのしきい値を設定する

October 7, 2021

Citrix ADC アプライアンスは、そのサービスにバインドされ、UP しているすべてのモニターの重みの合計が、サービスに設定されているしきい値以上である場合にのみ、サービスを UP として指定します。モニタの重みは、モニタが UP としてバインドされるサービスの指定にどの程度貢献しているかを指定します。

デフォルトでは、モニタのしきい値は 0 に設定され、モニタの重みは 1 に設定されます。すべてのモニターは、その後、等しい重量を持っており、モニターのいずれかが DOWN になると、サービスは DOWN に行くことができます。

たとえば、モニター-HTTP-1、モニター-HTTP-2、モニター-HTTP-3 という 3 つのモニターがそれぞれ Service-HTTP-1 にバインドされ、サービスで構成されているしきい値が 3 であるとします。各モニタに次の重みが割り当てられているとします。

- モニター-HTTP-1 の重みは 1 です。
- モニター HTTP-2 の重みは 3 です。
- モニタ-HTTP-3 の重みは 1 です。

サービスは、次のいずれかに該当する場合にのみ UP とマークされます。

- モニター-HTTP-2 は UP しています。
- モニター-HTTP-2 およびモニター-HTTP-1 またはモニター-HTTP-3 が UP である
- 3 台のモニタはすべて UP です。

CLI を使用してサービスの監視しきい値を設定するには

コマンドプロンプトで入力します。

```
set service <name> -monThreshold <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -monThreshold 100
<!--NeedCopy-->
```

GUI を使用してサービスのモニタのしきい値を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[モニタのしきい値] を選択します。

アイドル状態のクライアント接続のタイムアウト値を設定する

October 7, 2021

タイムアウト値を使用してサービスを構成し、設定した時間が経過したときにアイドル状態のクライアント接続を終了できます。構成された時間内にクライアントがアイドル状態の場合、Citrix ADC アプライアンスはクライアント接続を閉じます。

CLI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
set service <name> -cltTimeout <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -cltTimeout 100
<!--NeedCopy-->
```

GUI を使用してアイドル状態のクライアント接続のタイムアウト値を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[クライアントアイドルタイムアウト] を選択します。

アイドル状態のサーバー接続のタイムアウト値を設定する

October 7, 2021

タイムアウト値を使用してサービスを構成し、設定された時間（秒単位）が経過したときにアイドル状態のサーバー接続を終了できます。サーバーが構成された時間アイドル状態の場合、Citrix ADC アプライアンスはサーバー接続を切断します。

CLI を使用してアイドル状態のサーバー接続のタイムアウト値を設定するには

コマンドプロンプトで入力します。

```
set service <name> -svrTimeout <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -svrTimeout 100
<!--NeedCopy-->
```


GUI を使用してアイドル状態のサーバー接続のタイムアウト値を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[サーバーアイドルタイムアウト] を選択します。

クライアントによる帯域幅使用量の制限を設定する

October 7, 2021

場合によっては、サーバーの帯域幅が制限されてクライアント要求を処理し、過負荷になることがあります。サーバの過負荷を防ぐために、サーバによって処理される帯域幅の上限を Kbps 単位で指定できます。Citrix ADC アプライアンスは、この制限に達するまで、負荷分散されたサーバーに要求を転送します。

CLI を使用してサービスの最大帯域幅制限を設定するには

コマンドプロンプトで入力します。

```
set service <name> -maxBandwidth <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-HTTP-1 -maxBandwidth 100
<!--NeedCopy-->
```

GUI を使用してサービスの最大帯域幅制限を設定するには

1. **Traffic Management > Load Balancing > Services** に移動してサービスを開きます。
2. [詳細設定] で、[しきい値とタイムアウト] を選択し、[最大帯域幅] を選択します。

クライアント要求をキャッシュにリダイレクトする

October 7, 2021

クライアント要求をキャッシュにリダイレクトするようにサービスを設定し、キャッシュ不可能な要求を、設定された負荷分散方式で選択されたサービスに転送できます。

CLI を使用してサービスにキャッシュリダイレクトを設定するには

コマンドプロンプトで入力します。

```
set service <name> -cacheable <Value>
<!--NeedCopy-->
```

例:

```
set service Service-HTTP-1 -cacheable YES
<!--NeedCopy-->
```

GUI を使用してサービスのキャッシュリダイレクトを設定するには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. サービスを開き、キャッシュタイプを設定します。

VLAN 透過性のために VLAN ID を保持する

October 7, 2021

サーバに転送されるパケットにクライアントの VLAN ID を保持するように、ロードバランシング仮想サーバを設定できます。仮想サーバは、タイプ ANY のワイルドカード仮想サーバであり、MAC モードで機能している必要があります。

CLI を使用してクライアント VLAN ID を保持するようにロードバランシング仮想サーバを設定するには

コマンドプロンプトで次のコマンドを入力して、クライアント VLAN ID を保持し、構成を確認するようにロードバランシング仮想サーバーを構成します。

```
set lb vserver <name> -m MAC -macmodeRetainvlan ENABLED

show lb vserver <name>
<!--NeedCopy-->
```

> 注
>
> `-m MAC`オプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

GUI を使用してクライアント **VLAN ID** を保持するようにロードバランシング仮想サーバを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[**VLAN ID** を保持] を選択します。

バインドされたサービスの正常性の割合に基づいて自動状態移行を構成する

October 7, 2021

アクティブなサービスの割合が設定されたしきい値を下回ると、自動的に UP 状態から DOWN 状態に移行するように、負荷分散仮想サーバを設定できます。たとえば、10 個のサービスをロードバランシング仮想サーバにバインドし、その仮想サーバのしきい値を 50% に設定した場合、6 つ以上のサービスが DOWN の場合、UP から DOWN に移行します。稼働率がしきい値を超えると、仮想サーバは UP 状態に戻ります。

また、バインドされたサービスの正常性の割合によって仮想サーバーの状態が変化したときに Citrix ADC アプライアンスから通知されるようにする場合は、ENTITY-STATE という SNMP アラームを有効にすることもできます。

CLI を使用してパーセンテージベースの自動状態移行を設定するには

コマンドプロンプトで次のコマンドを入力して、仮想サーバーの自動状態遷移を構成し、構成を確認します。

```
set lb vserver <name> -healthThreshold <positive_integer>

show lb vserver <name>
<!--NeedCopy-->
```

GUI を使用してパーセンテージに基づく自動状態遷移を構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。
2. [詳細設定] で、[トラフィック設定] を選択し、[ヘルスしきい値] を設定します。

CLI を使用して **ENTITY-STATE** アラームを有効にするには

コマンドプロンプトで次のコマンドを入力して、ENTITY-STATE SNMP アラームを有効にし、設定を確認します。


```
enable snmp alarm ENTITY-STATE

show snmp alarm
<!--NeedCopy-->
```

GUI を使用して **ENTITY-STATE** アラームを有効にするには

1. [システム] **>** **[SNMP]** **>** [アラーム] に移動します。
2. [**ENTITY-STATE**] を選択し、[アクション] リストで [有効] を選択します。

内蔵モニタ

October 7, 2021

Citrix ADC アプライアンスには、サービスを監視するために使用できるさまざまな組み込みモニターが含まれています。これらの組み込みモニタは、一般的なプロトコルのほとんどを処理します。これらは、要件に合わせて間隔、応答タイムアウトなどの一部のパラメータを変更するオプションを提供します。ただし、モニタ名とプロトコルは変更できません。詳細については、「モニターの変更」を参照してください。組み込みモニターをサービスにバインドし、サービスからバインド解除することもできます。

> 注:
>
> 組み込みモニターに基づいてカスタムモニターを作成できます。カスタムモニターの作成方法については、「負荷分散セットアップでのモニターの構成」を参照してください。

TCP ベースのアプリケーション監視

October 7, 2021

Citrix ADC アプライアンスには、TCP ベースのアプリケーションを監視する 2 つの組み込みモニターがあります。`tcp-default`および`ping-default`。サービスを作成すると、適切なデフォルトモニターが自動的にバインドされるため、サービスが UP であればすぐに使用できます。tcp-default モニタは、すべての TCP サービスにバインドされます。ping-default モニタは、TCP 以外のすべてのサービスにバインドされます。

既定のモニターを削除または変更することはできません。他のモニタを TCP サービスにバインドすると、デフォルトモニタはサービスからバインド解除されます。次の表に、モニタタイプ、および各タイプに関連付けられたパラメータとモニタリングプロセスを示します。

モニタタイプ	特定のパラメータ	プロセス
tcp	該当なし	Citrix ADC アプライアンスは、モニターの宛先との 3 ウェイハンドシェイクを確立し、接続を閉じます。アプライアンスが宛先への TCP トラフィックを監視する場合、TCP モニタリング要求は送信されません。これは、LRTM が無効になっている場合に発生します。デフォルトでは、このモニタでは LRTM は無効になっています。
http	httprequest [「HEAD/」] -サービスに送信される HTTP リクエスト。respcode [200] -サービスから一連の HTTP 応答コードが期待されます。	Citrix ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは HTTP 要求を送信し、応答コードと構成済みの応答コードのセットを比較します。
tcp-ecv	send [「」] -サービスに送信されるデータです。文字列の最大許容長さは 512 バイトです。recv [「」] -サービスからの期待される応答。文字列の最大許容長は 128 バイトです。最後の文字は NULL 終端です。	Citrix ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは送信パラメータを使用して特定のデータをサービスに送信し、受信パラメータを介して特定の応答を期待します。サーバーによって異なるサイズのセグメントが送信されます。ただし、パターンは 16 個の TCP セグメント内にある必要があります。

モニタタイプ	特定のパラメータ	プロセス
http-ecv	send [「」] -サービスに送信される HTTP データ。recv [「」] -サービスからの期待される HTTP 応答データ	Citrix ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは send パラメータを使用して HTTP データをサービスに送信し、receive パラメータで指定された HTTP 応答を期待します。(HTTP ヘッダーを含まない HTTP ボディパーツ)。空の応答データは、任意の応答と一致します。期待されるデータは、レスポンスの HTTP 本体の最初の 24 K バイトの任意の場所にある可能性があります。
ping	該当なし	Citrix ADC アプライアンスは、モニタの宛先に ICMP エコー要求を送信し、ICMP エコー応答を期待します。

TCP ベースのアプリケーションの組み込みモニターを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

CLI を使用して **TCP** ベースのモニタを構成するには

次のコマンドを入力します。

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

TCP モニタタイプの例**:**

```
add lb monitor Exch2010-RPC-AddressBook TCP -LRTM ENABLED -interval 10
    -resptimeout 5 -destPort 59601
<!--NeedCopy-->
```

HTTP モニタタイプの例**:**

```
add lb monitor Mon_S4B_FE_2 HTTP -respCode 200 -httpRequest ”GET /
    Autodiscover/XFrame/XFrame.html” -LRTM ENABLED -retries 10 -secure
    YES
<!--NeedCopy-->
```

HTTP-ECV モニタタイプの例**:**

```
add lb monitor STM_EXC2016_SSLBridge_MON HTTP-ECV -send ”GET /owa/
    healthcheck.htm” -recv ”200 OK” -LRTM ENABLED -destPort 443 -secure
    YES
<!--NeedCopy-->
```

PING モニタタイプの例**:**

```
add lb monitor lbmon-localhost-ping PING -LRTM DISABLED -destIP
    127.0.0.1
<!--NeedCopy-->
```

SSL サービスの監視

October 7, 2021

Citrix ADC アプライアンスには、セキュリティで保護されたモニター、TCPS、HTTPS が組み込まれています。セキュアモニタを使用して、HTTP トラフィックと HTTP 以外のトラフィックを監視できます。セキュア HTTP モニターを構成するには、モニターの種類を HTTP として選択し、セキュアフラグを設定します。セキュア TCP モニタを構成するには、モニタの種類を TCP として選択し、セキュアフラグを設定します。セキュアモニタは、次のように動作します。

- セキュア **TCP** モニタリング。Citrix ADC アプライアンスは、TCP 接続を確立します。接続が確立されると、アプライアンスはサーバーと SSL ハンドシェイクを実行します。ハンドシェイクが終了すると、アプライアンスは接続を閉じます。
- セキュア **HTTP** モニタリング。Citrix ADC アプライアンスは、TCP 接続を確立します。接続が確立されると、アプライアンスはサーバーと SSL ハンドシェイクを実行します。SSL 接続が確立されると、アプライアンスは暗号化されたチャネルを介して HTTP 要求を送信し、応答コードをチェックします。

次の表に、SSL サービスを監視するための組み込みモニターを示します。

モニタの種類	プローブ	成功基準（直接条件）
TCP	TCP 接続、SSL ハンドシェイク	成功した TCP 接続が確立され、SSL ハンドシェイクが成功しました。
HTTP	TCP 接続、SSL ハンドシェイク、暗号化された HTTP 要求	成功した TCP 接続が確立され、成功した SSL ハンドシェイクが実行され、サーバの HTTP 応答で予期される HTTP 応答コードが暗号化されます。
TCP-ECV	TCP 接続。SSL ハンドシェイク (サーバーに送信されるデータは暗号化されます。)	成功した TCP 接続が確立され、正常な SSL ハンドシェイクが実行され、予期された TCP データがサーバーから受信されます。
HTTP-ECV	TCP 接続、SSL ハンドシェイク (暗号化された HTTP 要求)	成功した TCP 接続が確立され、正常な SSL ハンドシェイクが実行され、期待される HTTP データがサーバーから受信されます。

HTTPS-ECV ヘルスチェックモニタの設定例

HTTP サービスには、拡張コンテンツ検証 (ECV) が可能なモニターが事前に定義されています。
これらのモニターは、正常な TCP 接続を超えて検証が必要な場合に使用されます。これらのモニタは、次の条件がすべて満たされた場合に、サービスを UP として検証します。

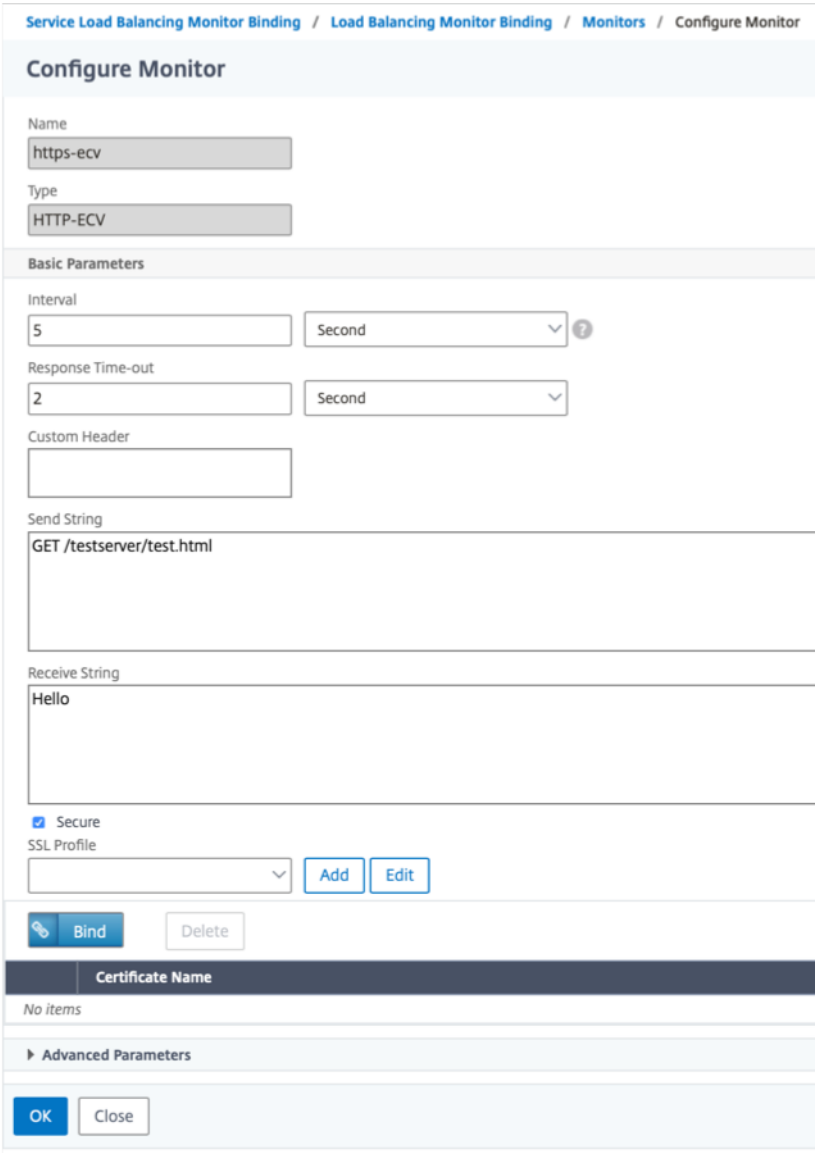
- TCP 接続に成功しました。
- 特定のタイプの要求を生成する必要があります。
- 受信文字列からの応答には、特定のメッセージが期待されます。

これらのモニタでは、要求文字列と応答文字列が設定されます。Citrix ADC モニターによって受信された応答文字列が設定された文字列と一致する場合、サービスは UP とマークされます。

GUI を使用してモニタをサービスにバインドする

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを作成し、プロトコルを **SSL** として指定します。[**OK**] をクリックします。
2. [負荷分散モニタのバインド] ペインで、[バインドの追加] をクリックします。
3. モニタの種類として **HTTPS-ECV** を選択し、[**Edit**] をクリックします。
4. [モニタの構成] ペインの [基本パラメータ] タブで、次のパラメータの値を入力します。

- 送信文字列 — モニタがサービスに送信する必要がある文字列。
- 受信文字列 — サービスを UP としてマークするためにモニタが受信する必要がある文字列。



5. [**OK]** をクリックして、モニタの構成を完了します。
6. [**Select**] をクリックします。
7. [バインド] をクリックして、**HTTPS-ECV** モニターをサービスにバインドします。
8. [閉じる] をクリックします。

CLI を使用してモニタをサービスにバインドする

コマンドプロンプトで入力します。


```
bind service <servicename> -monitorName https-ecv
<!--NeedCopy-->
```

例:

```
bind services1 -monitorName https-ecv
<!--NeedCopy-->
```

HTTP/2 サービスモニタリング

October 7, 2021

Citrix ADC アプライアンスは、HTTP/2 サービスのヘルスステータスを監視するための HTTP/2 モニターをサポートしています。

HTTP/2 モニタは、2 つの異なる方法で設定できます。トラフィックタイプに応じて、HTTP/2 モニタを設定できます。

- **HTTP/2** ダイレクト。HTTP/2 Direct は、非セキュアな HTTP/2 サービスを監視するように設定できます。
- **HTTP/2 SSL**。SSL を介したセキュアなトラフィックをモニタリングするために HTTP/2 SSL を設定できます。HTTP/2 で secure flag パラメータを有効にして、SSL トラフィックを監視します。

http2direct と http2ssl は、HTTP/2 プロトコルでサポートされている 2 つの異なる組み込みモニターです。

次の表に、各タイプに関連付けられた構成タイプ、およびモニタリングプロセスを示します。

構成タイプ	プローブ	成功基準
HTTP/2 ダイレクト	TCP 接続; HTTP2 接続の序文 & 設定ネゴシエーション; HTTP2 要求	HTTP/2 応答ステータスコードは、設定されたレスポンスコードと一致する必要があります。
HTTP/2 SSL	TCP 接続; SSL ハンドシェイク; HTTP2 接続の序文と設定ネゴシエーション; HTTP2 要求	サーバーは常に HTTP/2 プロトコルでALPNを選択する必要があり、HTTP/2 応答ステータスコードは、設定された応答コードと一致する必要があります。

CLI を使用して HTTP/2 モニタをサービスにバインドする

コマンドプロンプトで入力します。

- `bind service <servicename> -monitorName <name>`
- `bind service <servicename> -monitorName <name>`

例:

- `bind service s1 -monitorName http2direct`
- `bind service s2 -monitorName http2ssl`

プロキシプロトコルサービスの監視

October 7, 2021

プロキシプロトコルを備えた Citrix ADC アプライアンスは、モニターチェックをサポートしています。モニタチェックでは、バックエンドサーバーがプロキシプロトコルもサポートしていることを確認します。Citrix ADC アプライアンスには、HTTP または TCP 関連サービス用の 4 つのビルトインモニタタイプ（HTTP、HTTPS、HTTP-ECV、TCP-ECV）があります。

次の表に、モニタタイプ、および各タイプに関連付けられたパラメータとモニタリングプロセスを示します。

構成タイプ	プローブ	成功基準
HTTP	`httprequest` [「HEAD/」] -サービスに送信される HTTP リクエスト。`respcode` [200] -サービスから一連の HTTP レスポンスコードが必要です。	Citrix ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは HTTP 要求を送信し、応答コードと構成済みの応答コードのセットを比較します。
HTTPS	`httprequest` [「HEAD/」] -サービスに送信される HTTPS リクエスト。`respcode` [200] -サービスから一連の HTTPS 応答コードが期待されます。	Citrix ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは HTTPS 要求を送信し、応答コードを設定済みの応答コードセットと比較します。

構成タイプ	プローブ	成功基準
HTTP-ECV	send [””] -サービスに送信される HTTP データ。Received [””] -サービスから期待される HTTP 応答データ	Citrix ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは送信パラメータを使用して HTTP データをサービスに送信し、受信パラメータで指定される HTTP 応答を期待します。(HTTP ヘッダを含まない HTTP ボディ部分)。空の応答データは、任意の応答と一致します。期待されるデータは、レスポンスの HTTP 本体の最初の 24 K バイトの任意の場所にある可能性があります。
TCP-ECV	send [””] - is the data that is sent to the service. 文字列の最大許容長さは 512 K バイトです。受信 [””] -サービスからの期待される応答。文字列の最大許容長は 128 K バイトです。	Citrix ADC アプライアンスは、モニターの送信先と 3 ウェイハンドシェイクを確立します。接続が確立されると、アプライアンスは送信パラメータを使用して特定のデータをサービスに送信し、受信パラメータによる特定の応答を期待します。異なるサーバによって、異なるサイズのセグメントが送信されます。ただし、パターンは 16 個の TCP セグメント内にある必要があります。

プロキシプロトコルモニタは、`netprofile`を使用して設定できます。

CLI を使用したプロキシプロトコルモニタの設定

コマンドプロンプトで入力します。

1. プロキシプロトコルを有効にしてネットプロファイルを追加する

```
add netprofile <name> -proxyProtocol ( ENABLED | DISABLED )
```

例:


```
add netprofile profile1 -proxyProtocol ENABLED
```

1. ネットプロファイルをサービスにバインドします。

   ```
   set service <name> -netprofile <netprofile-name>
   ```

例:

```
set service S1 -netprofile profile1
```

> 注
>
> ネットプロファイルをサービスにバインドする場合は、前述のコマンドを実行できます。

1. ネットプロファイルをモニターにバインドします。

   ```
   set lb monitor <monitor-name> <type> -netprofile <netprofile-name>
   ```

例:

```
set lb monitor http1 HTTPS -netprofile profile1
```

> 注
>
> - ネットプロファイルをモニタにバインドする場合は、前述のコマンドを実行できます。
> - お好みのモニタタイプを選択できます。HTTP、HTTPS、TCP-ECV、または HTTP-ECV のいずれかになります。
>
> 重要
>
> - 一般に、サービスにバインドされたネットプロファイル（プロキシプロトコルが有効）が考慮されます。
> - ネットプロファイルがモニターとサービスの両方にバインドされている場合、モニターにバインドされたネットプロファイルが考慮されます。サービスにバインドされたネットプロファイルは無視されます。

FTP サービスの監視

October 7, 2021

FTP サービスを監視するために、Citrix ADC アプライアンスは FTP サーバーへの 2 つの接続を開きます。まず、制御ポートに接続します。制御ポートは、クライアントと FTP サーバ間でコマンドを転送するために使用されます。予想される応答を受信すると、データポートに接続します。データポートは、クライアントと FTP サーバー間でファイルを転送するために使用されます。FTP サーバーが期待どおりに応答する場合のみ、両方の接続で UP とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

Citrix ADC アプライアンスには、FTP サービス用の 2 つのモニターが内蔵されています。FTP モニターと FTP-EXTENDED です。FTP-EXTENDED モニタは、スクリプト可能なモニタです。これは、nsftp.pl スクリプトを使用します。FTP-EXTENDED モニタスクリプトは、セキュアなプローブを FTP サービスに送信するように拡張されました。FTP-EXTENDED タイプのモニタを作成できます。nsftp.pl スクリプトは、デフォルトのディレクトリから自動的に取得されます。

CLI を使用してセキュア **FTP** プローブを **FTP** サービスに送信するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> -username <string> -password <
    string> -filename <filename>
<!--NeedCopy-->
```

例

```
add monitor mon1 FTP-EXTENDED -username root -password freebsd -
    filename fsdf
<!--NeedCopy-->
```

GUI を使用してセキュアな **FTP** プローブを **FTP** サービスに送信するには

1. [トラフィック管理] > [負荷分散] > [モニタ] に移動します。
2. モニタの種類を **FTP-EXTENDED** として指定し、パラメータを設定します。
3.「特殊パラメータ」で、ファイル名、ユーザー名、および パスワードを指定します。

FTP サービスの状態をチェックするように組み込みモニターを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

SFTP によるサーバーの安全な監視

October 7, 2021

SSH ファイル転送プロトコル (SFTP) 監視をサポートするために、ユーザースクリプト「nssftp.pl」が追加されました。これは、組み込みの Citrix ADC ユーザーモニターの現在のリストにあり、/netscaler/モニターディレクトリ

にあります。SFTP モニタは、指定したユーザ名とパスワードを使用して、ファイルがサーバに存在するかどうかを確認します。

CLI を使用して **SFTP** を使用してセキュアモニタリングを設定するには

コマンドプロンプトで入力します。

```
1 add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
     string> -secure ( YES | NO )
2 <!--NeedCopy-->
```

例**:**

```
1 add monitor SFTP_MON USER -scriptname nssftp.pl -scriptargs "file=
     example.txt;user=sam;password=sam_passwd"
2 <!--NeedCopy-->
```

GUI を使用して **SFTP** を使用してセキュアモニタリングを設定するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動し、**[**タイプ**]** で **USER** を指定します。
2. [特殊パラメータ**]** の [スクリプト名] で、[nssftp.pl] を選択します。
3. スクリプト引数を指定します。

セキュリティで保護されたモニターでの **SSL** パラメーターの設定

October 7, 2021

> 重要
>
> この機能は、新しいデフォルトプロファイルでのみサポートされます。これらのプロファイルの詳細については、「拡張 SSL プロファイルインフラストラクチャの概要」を参照してください。

モニタは、グローバル設定またはバインドされているサービスの設定のいずれかを継承します。モニターが SSL_BRIDGE などの非 SSL_TCP サービスまたは非 SSL_TCP サービスにバインドされている場合、プロトコルのバージョンや使用する暗号などの SSL 設定では構成できません。したがって、展開でバックエンドサーバーの SSL ベースの監視が必要な場合、監視は効果がありません。

SSL プロファイルをモニターにバインドすることで、バックエンドサーバーの SSL ベースの監視をより詳細に制御できます。SSL プロファイルには、SSL パラメータ、暗号バインディング、および ECC バインディングが含まれ

ます。たとえば、SSL プロファイルでサーバー認証、暗号、およびプロトコルのバージョンを設定し、そのプロファイルをモニターにバインドできます。サーバ認証を実行するには、CA 証明書をモニタにバインドする必要もあります。クライアント認証を実行するには、クライアント証明書をモニタにバインドする必要があります。「bind lb monitor」コマンドの新しいパラメータを使用すると、新しいパラメータを使用できます。

> 注
>
> SSL 設定は、セキュリティで保護されたモニターを追加した場合にのみ有効になります。また、SSL プロファイルタイプは **BackEnd** である必要があります。

SSL プロファイルをサポートするモニタタイプ

SSL プロファイルは、次のモニタタイプにバインドできます。

- HTTP
- HTTP-ECV
- TCP
- TCP-ECV
- HTTP-INLINE

コマンドラインを使用してモニターを追加するときに **SSL** プロファイルを指定するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> -secure  YES   -sslprofile <string>

set lb monitor <monitorName> <type> -secure YES -sslprofile <string>
<!--NeedCopy-->
```

例:

```
add ssl profile prof1 -sslProfileType  BackEnd

add lb monitor mon1 HTTP -secure  YES -sslprofile prof1
<!--NeedCopy-->
```

コマンドラインを使用して証明書とキーのペアをモニタにバインドするには

コマンドプロンプトで入力します。


```
bind monitor  <monitor name> -certkeyName <string>  [(-CA  [-crlCheck (
    Mandatory | Optional ) | -ocspCheck ( Mandatory | Optional )]
<!--NeedCopy-->
```

SIP サービスのモニタリング

October 7, 2021

Citrix ADC には、**SIP-UDP** モニタと **SIP-TCP** モニタの 2 つの組み込みモニタがあります。SIP モニタは、SIP 要求方式を SIP サービスに送信することによって、SIP モニタがバインドされている SIP サービスを定期的にチェックします。SIP サービスが応答コードで応答した場合、モニタはサービスを UP としてマークします。SIP サービスが応答しない場合、または正しく応答しない場合は、DOWN としてマークされます。

パラメーター	Specifies
sipURI	SIP サーバの SIP アドレッシングスキーマ。
sipmethod	SIP サービスのプローブに使用される SIP 要求のタイプ。次の方法のいずれかを指定します。INVITE、OPTION（デフォルト）、登録
respcode	SIP サービスがプローブ要求に応答する SIP 応答コード。デフォルトは 200 です。

RADIUS サービスのモニタリング

October 7, 2021

Citrix ADC アプライアンスの RADIUS モニターは、認証要求をサービスに送信することで、バインドされている RADIUS サービスの状態を定期的にチェックします。RADIUS サーバは RADIUS モニタを認証し、応答を送信します。デフォルトでは、モニタは RADIUS サーバからの応答コード 2（デフォルトの Access-Accept 応答）を受信すると想定しています。モニタが適切な応答を受信している限り、サービスが UP とマークされます。

注：RADIUS モニタでは、PAP タイプ認証だけがサポートされます。

- クライアントの認証に成功すると、RADIUS サーバは Access-Accept 応答を送信します。デフォルトのアクセス受け入れ応答コードは 2 で、これはアプライアンスが使用するコードです。
- クライアントが認証に失敗した場合（ユーザー名、パスワード、秘密キーに不一致がある場合など）、RADIUS サーバは Access-Reject 応答を送信します。デフォルトのアクセス拒否応答コードは 3 で、これはアプライ

アンスが使用するコードです。

パラメーター	Specifies
userName	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 サーバーのユーザー名。このユーザー名はプローブで使用されます。
password	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP サーバの監視に使用されるパスワード。
radKey	RADIUS サーバがクライアント認証時に使用する共有秘密キーの値。
radNASid	アクセス要求が行われたときにペイロードにカプセル化される NAS-ID。
radNASip	アクセス要求が行われたときにペイロードにカプセル化される IP アドレス。radNASip が構成されていない場合、Citrix ADC アプライアンスはマッピングされた IP アドレス（MIP）を NAS の IP アドレスとして RADIUS サーバーに送信します。

RADIUS サービスを監視するには、バインド先の RADIUS サーバを次のように設定する必要があります。

1. モニタが認証に使用するクライアントのユーザ名とパスワードを RADIUS 認証データベースに追加します。
2. クライアントの IP アドレスと秘密キーを適切な RADIUS データベースに追加します。
3. アプライアンスが RADIUS パケットを RADIUS データベースに送信するために使用する IP アドレスを追加します。Citrix ADC アプライアンスに複数のマッピング IP アドレスがある場合、またはサブネット IP アドレス（SNIP）を使用する場合は、すべての IP アドレスに同じ秘密キーを追加する必要があります。

   注意：アプライアンスで使用される IP アドレスが RADIUS データベースに追加されない場合、RADIUS サーバはすべてのパケットを廃棄します。

RADIUS サーバの状態を確認するように組み込みモニタを構成するには、ロードバランシングセットアップでのモニタの設定を参照してください。

RADIUS サーバからのアカウンティング情報の配信を監視する

October 7, 2021

RADIUS アカウンティングモニタと呼ばれるモニタを構成して、認証、認可、アカウンティング（Citrix ADC AAA）に使用される RADIUS サーバーが想定どおりにアカウンティング情報を配信しているかどうかを判断できます。モニタのタイプは RADIUS_ACCOUNTING です。プローブは、/nsconfig/monitors/ ディレクトリにある nsbmradius.pl という名前の Perl スクリプトによって生成されます。スクリプトは、連続するアカウンティング要求プローブを RADIUS サーバに送信します。プローブが成功したと見なされるのは、RADIUS アカウンティングサーバが Code フィールドが 5 に設定されているパケットで応答した場合（RFC 2866 によれば、アカウンティング応答パケットを示す）だけです。

RADIUS アカウンティングモニタを設定する場合は、秘密キーを指定する必要があります。オプションのパラメータを指定できます。各パラメータは、Acct-Status-Type や Framed-IP-Address などの RADIUS 属性を表します。これらの属性の詳細については、RFC 2865、「リモート認証ダイヤルインユーザーサービス (RADIUS)」、および RFC 2866「RADIUS アカウンティング」を参照してください。

コマンドラインインターフェイスを使用して **RADIUS** アカウンティングモニタを構成するには

コマンドプロンプトで次のコマンドを入力して、RADIUS アカウンティングモニタを設定し、設定を確認します。

```
add lb monitor <monitorName> RADIUS_ACCOUNTING [-userName <string>] {
 -password  }
  {
 -radKey  }
  [-radNASip <ip_addr>] [-radAccountType <positive_integer>] [-
     radFramedIP <ip_addr>] [-radAPN <string>] [-radMSISDN <string>] [-
     radAccountSession <string>]

show lb monitor <monitorName>
<!--NeedCopy-->
```

例

```
add lb monitor radAccntMon RADIUS_ACCOUNTING -radKey ”8d#>9jr4rV)L7%a2-
   zW13sM”
<!--NeedCopy-->
```

DNS および **DNS-TCP** サービスの監視

October 7, 2021

Citrix ADC アプライアンスには、DNS サービスと DNS-TCP という、DNS サービスを監視するために使用できる 2 つのモニターが内蔵されています。サービスにバインドされると、いずれかのモニターは、DNS クエリを送信することによって、その DNS サービスの状態を定期的にチェックします。クエリーは IPv4 アドレスまたは IPv6 アドレスに解決されます。その IP アドレスは、設定したテスト IP アドレスのリストと照合されます。リストには、最大 5 つの IP アドレスを含めることができます。解決された IP アドレスがリストの 1 つ以上の IP アドレスと一致する場合、DNS サービスはアップとしてマークされます。解決済み IP が一覧の IP アドレスと一致しない場合、DNS サービスはダウンとしてマークされます。

パラメーター	説明
クエリを実行します	監視対象の DNS サービスに送信された DNS クエリ (ドメイン名)。デフォルト値:「\ 007」DNS クエリが成功すると、サービスは UP としてマークされます。それ以外の場合は、DOWN とマークされます。リバースモニタの場合、DNS クエリが成功すると、サービスは DOWN としてマークされます。それ以外の場合は、UP とマークされます。応答が受信されない場合、サービスは DOWN としてマークされます。
クエリータイプ	送信される DNS クエリの種類。可能な値:Possible values: Address, Zone.、ゾーン。
IPAddress	DNS モニタリングプローブに対する応答に対してチェックされる IP アドレスのリスト。
IPv6	IP アドレスが IPv6 形式を使用する場合は、このチェックボックスをオンにします。

組み込み DNS または DNS-TCP モニタを構成するには、ロードバランシングセットアップでのモニタの構成を参照してください。

LDAP サービスの監視

October 7, 2021

Citrix ADC アプライアンスには、LDAP サービスの監視に使用できる 1 つのモニターが内蔵されています。LDAP モニターです。認証と検索クエリの送信によって、バインドされている LDAP サービスを定期的にチェックします。検索が成功すると、サービスは UP とマークされます。LDAP サーバがエントリを見つけられない場合、LDAP モニタに障害メッセージが送信され、サービスに DOWN とマークされます。

LDAP モニタを設定して、クエリの送信時に実行する必要のある検索を定義します。Base DN パラメータを使用して、LDAP サーバーがテストクエリを開始する必要があるディレクトリ階層内の場所を指定できます。Attribute パ

ラメーターを使用して、ターゲットエンティティの属性を指定できます。

注：監視プローブは、NSIP アドレスから発信されます。

パラメーター	Specifies
baseDN	LDAP 検索の開始元である LDAP モニターのベース名。LDAP サーバーがローカルで実行されている場合、base のデフォルト値は dc =netscaler、dc=com.
bindDN	LDAP モニタの BDN 名。
filter	LDAP モニタのフィルタ。クエリで filter パラメーターを使用して、結果の数を制限します。クエリでこのパラメーターを指定しない場合、フィルタはオブジェクトクラス全体に適用されます。これは、CPU 使用率が高くなるなど、コストのかかる操作になる可能性があります。
password	LDAP サーバの監視に使用するパスワード。
attribute	LDAP モニタの属性。

組み込み LDAP モニタを設定するには、ロードバランシングセットアップでのモニタの設定を参照してください。

MySQL サービスのモニタリング

October 7, 2021

Citrix ADC アプライアンスには、MySQL サービスの監視に使用できる 1 つのモニタが内蔵されています。これは、定期的に検索クエリを送信することによって、それがバインドされている MySQL サービスをチェックします。検索が成功すると、サービスは UP とマークされます。MySQL サーバーが応答しないか、検索が失敗した場合は、MySQL モニターにエラーメッセージが送信され、サービスに DOWN とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

パラメーター	Specifies
データベース	MySQL モニターに使用されるデータベース。
sqlQuery	MySQL モニターに使用される SQL クエリ。

組み込み MySQL モニターを構成するには、負荷分散セットアップでのモニタの設定を参照してください。

CLI を使用して MySQL モニターを設定するには

次のコマンドを入力します。

```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
    string>
<!--NeedCopy-->
```

例:

```
add lb monitor mysql1 USER -scriptName nsmysql.pl -scriptArgs ”database
    =cloud;user=cloud;password=password;query=show tables from cloud”
<!--NeedCopy-->
```

SNMP サービスのモニタリング

October 7, 2021

Citrix ADC アプライアンスには、SMNP サービスの監視に使用できる 1 つのモニターが内蔵されています。SNMP モニターです。監視用に構成したエンタープライズ ID (OID) のクエリーを送信することで、バインド先のサービス上の SNMP エージェントを定期的にチェックします。クエリが成功すると、サービスは UP とマークされます。指定した OID が SNMP サービスによって検出されると、クエリーは成功し、SNMP モニタはサービスを UP にマークします。OID が見つからない場合、クエリーは失敗し、SNMP モニタはサービスを DOWN とマークします。

注: 監視プローブは、NSIP アドレスから発信されます。

パラメーター	Specifies
SNMPOID	SNMP モニタに使用される OID。
snmpCommunity	SNMP モニタに使用されるコミュニティ。
snmpThreshold	SNMP モニタに使用されるしきい値。
snmpVersion	ロードモニタリングに使用される SNMP バージョン。 可能な値:V1、V2。

組み込み SNMP モニタを設定するには、ロードバランシングセットアップでのモニタの設定を参照してください。

NNTP サービスの監視

October 7, 2021

Citrix ADC アプライアンスには、NNTP サービスの監視に使用できるモニターが 1 つ組み込まれています。NNTP モニターです。サービスに接続し、指定したニュースグループの存在を確認することによって、バインドされている NNTP サービスを定期的にチェックします。ニュースグループが存在する場合、検索は成功し、サービスには UP とマークされます。NNTP サービスが応答しないか、検索に失敗すると、サービスに DOWN とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

NNTP モニタは、オプションで、ニュースグループにテストメッセージを投稿するように設定できます。

パラメーター	Specifies
userName	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 サーバーのユーザー名。このユーザー名はプローブで使用されます。
password	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP サーバの監視に使用されるパスワード。
グループ	Group name to be queried for NNTP monitor.

組み込み NNTP モニタを設定するには、ロードバランシングセットアップでのモニタの設定を参照してください。

POP3 サービスの監視

October 7, 2021

Citrix ADC アプライアンスには、POP3 サービスの監視に使用できるモニターが 1 つ組み込まれています。POP3 サーバーとの接続を開いて、バインドされている POP3 サービスを定期的にチェックします。POP3 サーバーは、設定された期間内に正しい応答コードで応答すると、サービスが UP とマークされます。POP3 サービスが応答しない、または正しく応答しない場合は、サービスが DOWN とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

パラメーター	Specifies
userName	ユーザー名 POP3 サーバー。このユーザー名はプローブで使用されます。

パラメーター	Specifies
password	POP3 サーバーの監視に使用するパスワード。
scriptName	実行するスクリプトのパスと名前。
dispatcherIP	プローブの送信先となるディスパッチャの IP アドレス。
dispatcherPort	プローブの送信先であるディスパッチャのポート。

組み込み POP3 モニターを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

CLI を使用して POP3 モニターを構成するには

次のコマンドを入力します。

```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
    string>
<!--NeedCopy-->
```

例:

```
add lb monitor pop31 USER -scriptName nspop3.pl -scriptArgs ”user=
    test@lbmon1.net;password=Freebsd123”

<!--NeedCopy-->
```

SMTP サービスの監視

October 7, 2021

Citrix ADC アプライアンスには、SMTP サービス（SMTP モニター）を監視するために使用できるビルトインモニターがあります。モニタは、バインドされている SMTP サービスをチェックし、サーバが正常に動作していることを確認するための一連のハンドシェイクを実行します。SMTP サービスがハンドシェイクを正しく完了すると、モニタはサービスを UP とマークします。それ以外の場合は、SMTP サービスが応答しない、または正しく応答しない場合、サービスが DOWN とマークされます。

注：監視プローブは、NSIP アドレスから発信されます。

パラメーター	Specifies
scriptName	実行するスクリプトのパスと名前。
dispatcherIP	プローブの送信先であるディスパッチャの IP アドレス。
dispatcherPort	プローブの送信先であるディスパッチャのポート。

組み込み SMTP モニタを構成するには、ロードバランシングセットアップでのモニタの構成を参照してください。

RTSP サービスモニタリング

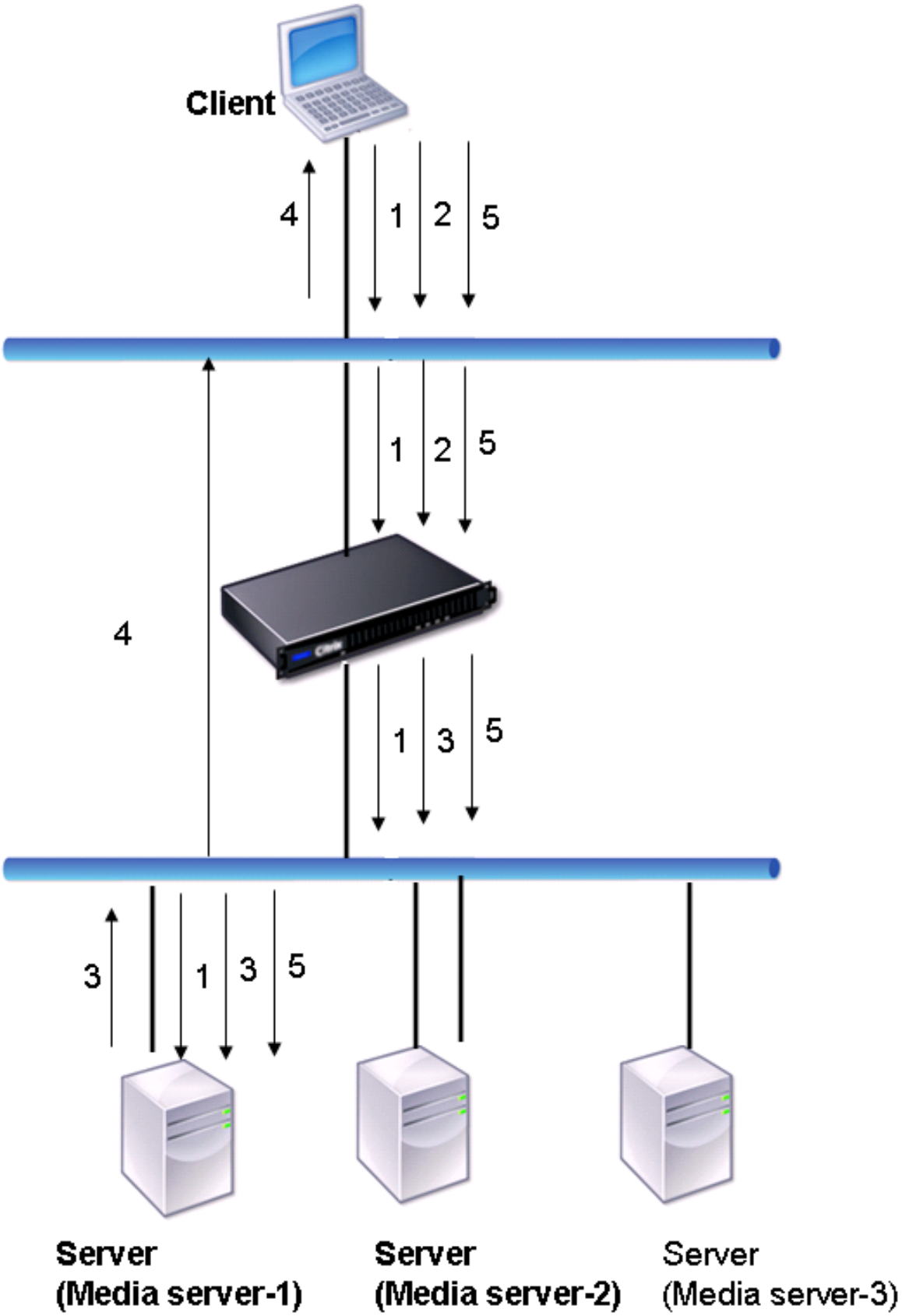
October 7, 2021

Citrix ADC アプライアンスには、RTSP サービスの監視に使用できる 1 つのモニタが内蔵されています。RTSP モニタです。ロードバランシングされた RTSP サーバとの接続を開いて、バインドされている RTSP サービスを定期的にチェックします。開く接続のタイプ、および予期する応答は、ネットワーク構成によって異なります。RTSP サービスが設定された期間内に期待どおりに応答すると、サービスが UP とマークされます。サービスが応答しない場合、または正しく応答しない場合は、サービスが DOWN とマークされます。

Citrix ADC アプライアンスは、NAT オフと NAT オンの 2 つのトポロジを使用して RTSP サーバーの負荷分散を構成できます。RTSP サーバは、アプライアンスをバイパスして、クライアントに応答を直接送信します。アプライアンスは、ネットワークが使用するトポロジに応じて、RTSP サービスを異なる方法で監視するように設定する必要があります。アプライアンスは、NAT オフモードと NAT オンモードの両方で、インラインモードまたは非インラインモードのいずれかに展開できます。

NAT オフモードでは、アプライアンスはルータとして動作します。クライアントから RTSP 要求を受信し、設定されたロードバランシング方式を使用して選択したサービスにルーティングします。負荷分散された RTSP サーバーに DNS でパブリックにアクセス可能な FQDN が割り当てられている場合、負荷分散されたサーバーはアプライアンスをバイパスして、クライアントに応答を直接送信します。次の図は、この構成を示しています。

図 1: RTSP in NAT-off Mode



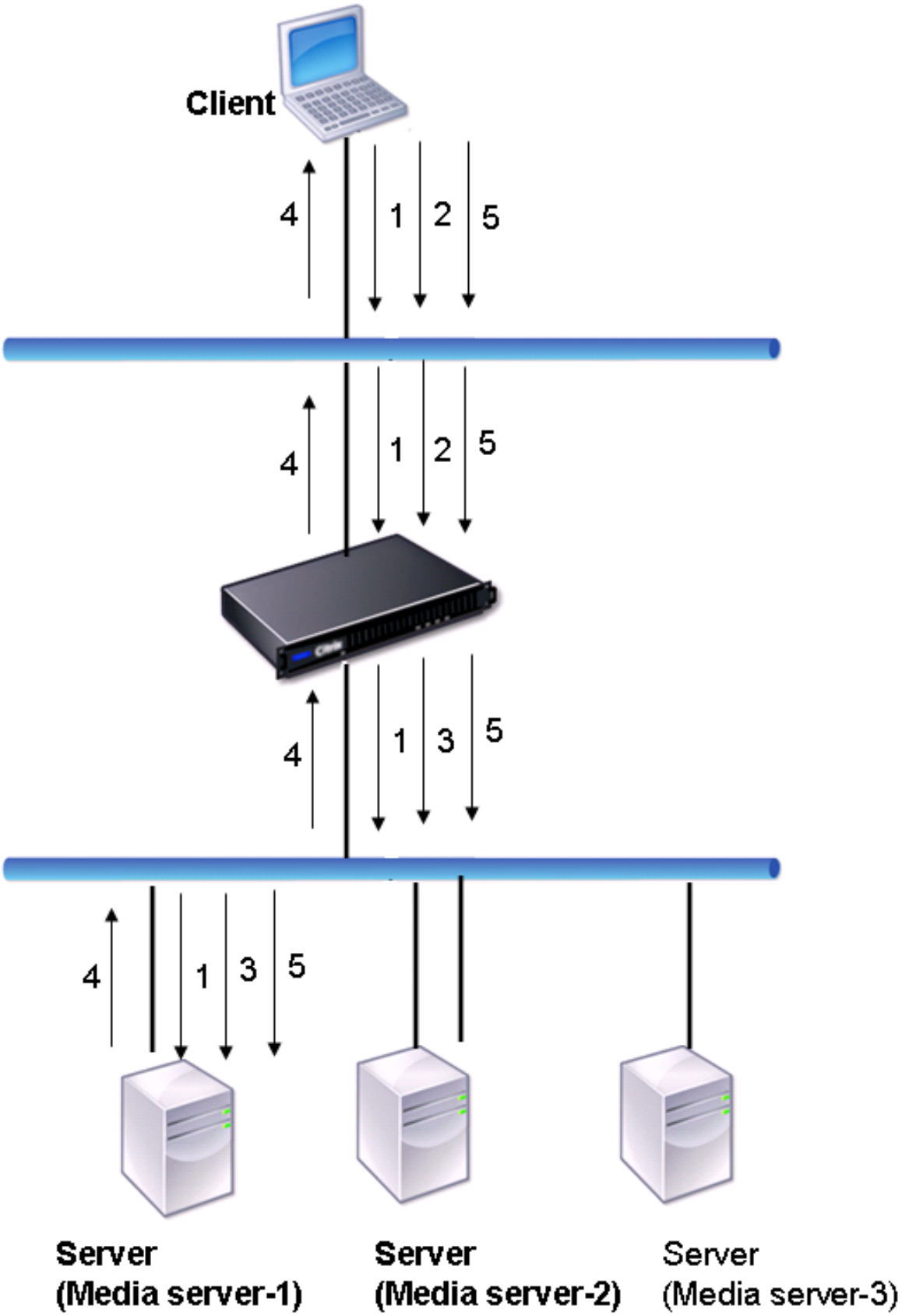
このシナリオでの要求と応答の流れは次のとおりです。

1. クライアントは、DESCRIBE 要求をアプライアンスに送信します。アプライアンスは、設定されたロードバランシング方式を使用してサービスを選択し、要求を Media Server-1 にルーティングします。
2. クライアントは、アプライアンスに SETUP 要求を送信します。RTSP セッション ID が DESCRIBE 要求で交換された場合、アプライアンスは RTSPSID パーシステンスを使用して、要求を Media Server-1 にルーティングします。RTSP セッション ID が SETUP 要求で交換された場合、アプライアンスは次のいずれかを実行します。
   - RTSP 要求が同じ TCP 接続で送信された場合、要求は Media Server-1 にルーティングされ、永続性が維持されます。
   - 要求が別の TCP 接続に着信した場合、設定されたロードバランシング方式を使用してサービスを選択し、永続性を維持せず、そのサービスに要求を送信します。これは、リクエストが別のサービスに送信される可能性があることを意味します。
3. Media Server-1 は、アプライアンスから SETUP 要求を受信し、RTSP 要求を処理するためのリソースを割り当てて、適切なセッション ID をクライアントに送信します。

   注：アプライアンスは、RTSP 接続によってバイパスされるため、RTSP 接続を識別するために NAT を実行しません。
4. 後続の要求では、クライアントはセッション ID を使用してセッションを識別し、制御メッセージをメディアサーバに送信します。Media Server-1 は、要求されたアクション（再生、転送、巻き戻しなど）を実行します。

NAT-on モードでは、アプライアンスはクライアントから RTSP 要求を受信し、設定されたロード・バランシング方式を使用してこれらの要求を適切なメディア・サーバーにルーティングします。次に、次の図に示すように、メディアサーバはアプライアンスを介してクライアントに応答を送信します。

図 2：RTSP in NAT-on Mode



このシナリオでの要求と応答の流れは次のとおりです。

1. クライアントは、DESCRIBE 要求をアプライアンスに送信します。アプライアンスは、設定されたロードバランシング方式を使用してサービスを選択し、要求を Media Server-1 にルーティングします。
2. クライアントは、アプライアンスに SETUP 要求を送信します。RTSP セッション ID が DESCRIBE 要求で交換された場合、アプライアンスは RTSPSID パーシステンスを使用して、要求を Media Server-1 にルーティングします。RTSP セッション ID が SETUP 要求で交換された場合、アプライアンスは次のいずれかを実行します。
   - RTSP 要求が同じ TCP 接続で送信された場合、要求は Media Server-1 にルーティングされ、永続性が維持されます。
   - 要求が別の TCP 接続に着信した場合、設定されたロードバランシング方式を使用してサービスを選択し、永続性を維持せず、そのサービスに要求を送信します。これは、リクエストが別のサービスに送信される可能性があることを意味します。
3. Media Server-1 は、アプライアンスから SETUP 要求を受信し、RTSP 要求を処理するためのリソースを割り当てて、適切なセッション ID をクライアントに送信します。
4. アプライアンスは NAT を実行して RTSP データ接続のクライアントを識別し、RTSP 接続はアプライアンスを通過し、正しいクライアントにルーティングされます。
5. その後の要求では、クライアントはセッション ID を使用してセッションを識別し、制御メッセージをアプライアンスに送信します。アプライアンスは RTSPSID パーシステンスを使用して適切なサービスを識別し、要求を Media Server-1 にルーティングします。Media Server-1 は、要求されたアクション（再生、転送、巻き戻しなど）を実行します。

RTSP モニタは、RTSP プロトコルを使用して RTSP サービスの状態を評価します。RTSP モニタは RTSP サーバに接続し、一連のハンドシェイクを実行して、サーバが正常に動作していることを確認します。

パラメーター	Specifies
rtspRequest	RTSP サーバに送信される RTSP 要求ストリング（OPTIONS * など）。デフォルト値は 07 です。リクエストの長さは 163 文字を超えないようにしてください。
respCode	サービスから期待される応答コードのセット。

RTSP モニタの構成手順については、ロードバランシングセットアップでのモニタの構成を参照してください。

XML ブローカーサービスの監視

October 7, 2021

Citrix ADC アプライアンスには、モニタータイプ CITRIX-XML-SERVICE が組み込まれており、これを使用して XML ブローカーサービスを監視するためのモニターを作成できます。XML ブローカーサービスは、Citrix XenApp によって使用されます。モニタは、サービスへの接続を開き、バインドされている XML サービスを定期的に調査します。設定した期間内にサーバが期待どおりに応答した場合、モニタはサービスを UP とマークします。サービスが応答しない場合、または正しく応答しない場合、モニターは、サービスを DOWN とマークします。

CITRIX-XML-SERVICE モニタを設定するには、標準パラメータの設定に加えてアプリケーション名を指定する必要があります。アプリケーション名は、XML Broker サービスの状態を監視するために実行する必要があるアプリケーションの名前です。既定のアプリケーションはメモ帳です。

XML Broker Services のモニターを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

> 注
>
> Citrix XML サービスモニタの「アプリケーション名」パラメータは、XenApp および Citrix Virtual Desktops バージョン 7 以降では無効です。XA/XD 7 では、このパラメータを使用しないことをお勧めします。このパラメータを設定する場合、このパラメータは内部的には使用されません。XA/XD 7 では、プローブ基準は異なります。ただし、XA/XD 7 以前のバージョンでは、「アプリケーション名」パラメーターを使用できます。

ARP 要求のモニタリング

October 7, 2021

Citrix ADC アプライアンスには、ARP 要求の監視に使用できる 1 つのモニターが内蔵されています。ARP モニターです。このモニタは、バインドされているサービスに ARP 要求を定期的に送信し、予想される応答を待機します。予想される応答を受信すると、サービスが UP にマークされます。応答がなかったり、間違った応答を受け取ったりすると、サービスが DOWN とマークされます。

ネットワーク層アドレスだけがわかっている場合、ARP は負荷分散サーバーのハードウェアアドレスを検索します。ARP は IPv4 と互換性があり、IP アドレスをイーサネット MAC アドレスに変換します。ARP モニタリングは IPv6 ネットワークには関係しないため、これらのネットワークではサポートされません。

ARP モニタには特別なパラメータはありません。

ARP モニタの設定手順については、ロードバランシングセットアップでのモニタの設定を参照してください。

XenDesktop Delivery Controller サービスの監視

October 7, 2021

デスクトップ仮想化では、Citrix ADC アプライアンスを使用して、Citrix XenDesktop 環境によって展開された Web インターフェイス（WI）サーバーと XenDesktop XenDesktop opDelivery Controller サーバーの負荷分散を行うことができます。Citrix ADC アプライアンスには、XenDesktop Delivery Controller サーバーを監視するビルトインモニター、CITRIX-XD-DDCモニターが用意されています。ヘルスチェックに加えて、XenDesktop Delivery Controller サーバーの有効なユーザーによってプローブが送信されたかどうかを確認することもできます。

モニターは、XML メッセージの形式で XenDesktop Delivery Controller サーバーにプローブを送信します。サーバがサーバファームの ID でプローブに応答すると、プローブは成功と見なされ、サーバのステータスは UP としてマークされます。HTTP 応答に成功コードがない場合、またはサーバーファームの ID が応答に存在しない場合、プローブは失敗と見なされ、サーバーのステータスは DOWN としてマークされます。

[認証情報の検証] オプションでは、モニターから XenDesktop Delivery Controller サーバーに送信されるプローブを指定します。つまり、サーバー名のみを要求するか、ログイン資格情報も検証するかを指定します。

注:
CITRIX-XD-DDCモニターでユーザー資格情報（ユーザー名、パスワード、ドメイン）が指定されているかどうかにかかわらず、XenDesktop Delivery Controller er サーバーは資格情報を検証するオプションがモニターで有効になっている場合にのみユーザーの資格情報を検証します。

ウィザードを使用して XenDesktop サーバーの負荷分散を構成すると、CITRIX-XD-DDCモニターが自動的に作成され、XenDesktop Delivery Controller er サービスにバインドされます。

コマンドラインインターフェイスを使用して、資格情報の検証オプションを指定して **XD-DDC** モニターを追加するには

コマンドプロンプトで次のコマンドを入力して、XD-DDC モニタを追加し、構成を確認します。

```
add lb monitor <monitorName> <monitorType> -userName <userName> -
    password <password> -domain <domain_name> -validateCred YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

例:


```
 > add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -
    password E12Dc35450a1 -domain dhop -validateCred YES
 Done
> show lb monitor xdddcmon
1) Name......:xdddcmon Type......:CITRIX-XD-DDC State......: ENABLED

Standard parameters:
Interval......:..5 sec...Retries......:..3
Response timeout......:..2 sec...Down time......:..30 sec
Reverse......:..NO...Transparent......:..NO
Secure......:..NO...LRTM......:..ENABLED
Action......:..Not applicable...Deviation......:..0 sec
Destination IP......:..Bound service
Destination port......:..Bound service
Iptunnel......:..NO
TOS......:..NO...TOS ID......:..0
SNMP Alert Retries......:..0...Success Retries......:..1
Failure Retries......:..0

Special parameters:
User Name..........:”Administrator”
Password..........:*****
DDC Domain.........: ”dhop”
Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **XD-DDC** モニターで資格情報の検証オプションを指定するには

コマンドプロンプトで入力します。

```
set lb monitor <monitorName> <monitorType> -userName -password -domain
   <domain_name> -validateCred YES
<!--NeedCopy-->
```

例:

```
 set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName
    Administrator -password D123S1R2A123 -domain dhop -validateCred YES
Done
<!--NeedCopy-->
```


構成ユーティリティを使用して **XD-DDC** モニターの資格情報の検証オプションを構成するには

[トラフィック管理] **>** [負荷分散] **>** [モニタ] に移動し、タイプ`Citrix-XD-DDC`のモニタを作成します。

Citrix StoreFront ストアの監視

October 7, 2021

Citrix StoreFront ストアのユーザーモニターを構成できます。モニターは、アカウントサービス、ディスカバリサービス、および認証エンドポイント（Citrix StoreFront Store が認証ストアの場合）を連続して調べることによって、StoreFront ストアの状態を決定します。これらのサービスのいずれかがプローブに応答しない場合、モニタープローブは失敗し、StoreFront ストアは DOWN としてマークされます。モニタは、バインドされたサービスの IP アドレスとポートにプローブを送信します。詳細については、「Citrix StoreFront ストアサービス API」を参照してください。

注：監視プローブは、NSIP アドレスから発信されます。ただし、StoreFront サーバーのサブネットがアプライアンスのサブネットと異なる場合は、サブネット IP（SNIP）アドレスが使用されます。

リリース 10.1 ビルド 120.13 以降、StoreFront モニターをサービスグループにバインドすることもできます。モニタはサービスグループの各メンバにバインドされ、プローブはバインドされたメンバ（サービス）の IP アドレスとポートに送信されます。また、サービスグループの各メンバーはメンバーの IP アドレスを使用して監視されるため、StoreFront モニターを使用して、サービスグループのメンバーとして追加された StoreFront クラスターノードを監視できるようになりました。

以前のリリースでは、StoreFront モニターが匿名ストアの認証を試みました。その結果、サービスを DOWN としてマークすることができ、負荷分散仮想サーバーの URL を使用して XenApp または XenDesktop を起動することはできません。

ビルド 64.x から、プローブの順序が変更されました。モニターは、アカウントサービス、検出ドキュメント、認証サービスを連続的に調査して StoreFront ストアの状態を判断し、匿名ストアの認証をスキップします。

StoreFront モニターのホスト名パラメータは非推奨です。HTTP（デフォルト）と HTTPS のどちらを使用してモニタプローブを送信するかを決定するために、secure パラメータが使用されるようになりました。

HTTPS を使用するには、セキュリティで保護されたオプションを [はい] に設定します。

コマンドラインインターフェイスを使用して **StoreFront** モニターを作成するには

コマンドプロンプトで次のコマンドを入力して、StoreFront モニターを構成し、構成を確認します。

```
add lb monitor <monitorName> STOREFRONT <string> -storeName <string> [-
    storefrontacctservice ( YES | NO )] -secure ( YES | NO )
```



```
2
3 show lb monitor <monitorName>
4 <!--NeedCopy-->
```

例

```
1 add lb monitor storefront_ssl STOREFRONT -storename myStore -
     storefrontacctservice YES -secure YES
2 <!--NeedCopy-->
```

構成ユーティリティを使用して **StoreFront** モニターを作成するには

トラフィック管理 **>** 負荷分散 **>** モニターに移動し、**STOREFRONT** タイプのモニターを作成します。

> 注
>
> StoreFront モニターの詳細については、StoreFront StoreFront のドキュメントを参照してください。

カスタムモニター

October 7, 2021

組み込みのモニタに加えて、カスタムモニタを使用してサービスの状態を確認することもできます。Citrix ADC アプライアンスは、Citrix ADC オペレーティングシステムに含まれるスクリプトに基づいて、いくつかのタイプのカスタムモニターを提供します。スクリプトを使用して、サービスに送信されるサービスまたはネットワークトラフィックの負荷に基づいて、サービスの状態を判断できます。カスタムモニタは、インラインモニタ、ユーザモニタ、および負荷モニタです。

これらのタイプのモニターでは、提供された機能を使用するか、独自のスクリプトを作成してこれらのスクリプトを使用して、モニタがバインドされているサービスの状態を判断できます。

HTTP インラインモニターを構成する

October 7, 2021

インラインモニタは、これらのサービスがクライアント要求を受信した場合にのみ、バインドされているサービスからの応答を分析し、プローブします。インラインモニタのタイプは HTTP-INLINE で、HTTP および HTTPS サービスでのみ設定できます。インラインモニタは、バインドされているサービスが UP であると判断し、送信される要求

に対する応答をチェックします。クライアント要求がサービスに送信されない場合、インラインモニタは、設定された URL を使用してサービスをプローブします。

注：インラインモニターは、HTTP または HTTPS グローバルサーバー負荷分散（GSLB）リモートまたはローカルサービスにバインドできません。これらのサービスは、実際の負荷分散 Web サーバーではなく仮想サーバーを表すためです。

インラインモニタには、プローブが失敗したときのタイムアウト値と再試行回数があります。Citrix ADC アプライアンスが障害発生時に実行するアクションの種類を次の中から選択できます。

- なし。明示的なアクションは実行されません。サービスとモニターを表示できます。モニターには、現在連続しているエラー応答の数とチェックされた累積応答の数が表示されます。
- **LOG**。イベントを ns/syslog に記録し、カウンタを表示します。
- **DOWN**。サービスをマークダウンし、トラフィックをサービスに送信しません。この設定により、サービスへの永続的な接続が切断されます。このアクションはまた、イベントをログに記録し、カウンタを表示します。

サービスがダウンした後、設定されたダウンタイムはサービスが DOWN のままになります。ダウンタイムが経過すると、インラインモニタは設定された URL を使用してサービスをプローブし、サービスが再び利用可能かどうかを調べます。プローブが成功すると、サービスの状態が UP に変わります。トラフィックはサービスに送信され、監視は以前と同じように再開されます。

インラインモニタを構成するには、負荷分散セットアップでのモニタの構成を参照してください。

CLI を使用して HTTP インラインモニタを設定するには

次のコマンドを入力します。

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

例:

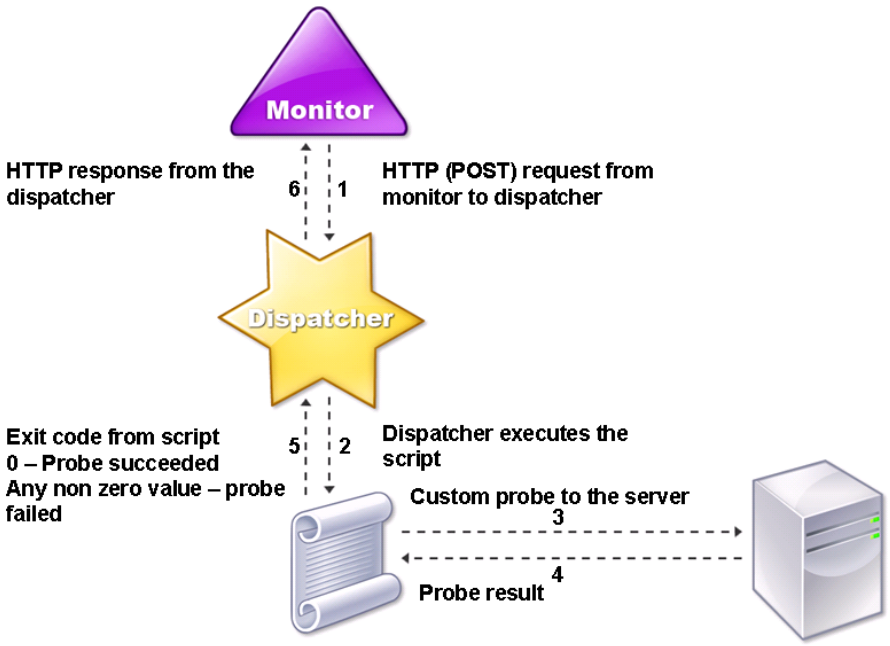
```
add lb monitor http_inline HTTP-INLINE -respCode 200 304 -httpRequest "
    HEAD /var/static/empty.htm" -resptimeout 4 -retries 1 -downTime 2 -
    action NONE
<!--NeedCopy-->
```


ユーザーモニターを理解する

August 12, 2022

ユーザーモニターは、カスタムモニターの範囲を拡張します。ユーザーモニターを作成して、Citrix ADC アプライアンスがサポートしていないカスタマイズされたアプリケーションとプロトコルの健全性を追跡できます。次の図は、ユーザーモニターの動作を示しています。

図 1: ユーザーモニター



ユーザーモニターには、次のコンポーネントが必要です。

- ディスパッチャー。モニタリング要求をリッスンするアプライアンス上のプロセス。ディスパッチャは、ループバック IP アドレス（127.0.0.1）およびポート 3013 上に置くことができます。ディスパッチャは、内部ディスパッチャとしても知られています。ディスパッチャは、Common Gateway Interface (CGI) をサポートする Web サーバーでも使用できます。このようなディスパッチャは、外部ディスパッチャとも呼ばれます。これらは、.NET スクリプトなど、FreeBSD 環境では動作しないカスタムスクリプトに使用されます。

  注: モニターで「セキュア」オプションを有効にして、HTTP ではなく HTTPS を使用するようにモニターとディスパッチャーを構成し、外部ディスパッチャーとして構成できます。ただし、内部ディスパッチャは HTTP のみを認識し、HTTPS を使用することはできません。

高可用性セットアップでは、ディスパッチャはプライマリとセカンダリの両方の Citrix ADC アプライアンスで実行されます。ディスパッチャは、セカンダリアプライアンスで非アクティブのままです。

[スクリプト]。スクリプトは、負荷分散サーバーにカスタムプローブを送信し、ディスパッチャに応答コードを返すプログラムです。スクリプトはディスパッチャに任意の値を返すことができますが、プローブが成功した場合、スクリプトは値 0 を返さなければなりません。ディスパッチャは、その他の値をプローブ障害と見なします。

Citrix ADC アプライアンスには、一般的に使用されるプロトコルのサンプルスクリプトがバンドルされています。これらのスクリプトは、/nsconfig/monititors ディレクトリに存在します。スクリプトを追加する場合は、そこにスクリプトを追加します。既存のスクリプトをカスタマイズするには、新しい名前でコピーを作成し、それを修正します。

重要:

- Citrix ADC リリース 13.0 ビルド 41.20 以降、`nsntlm-lwp.pl`スクリプトを使用して安全な NTLM サーバーを監視するためのモニターを作成できます。
- リリース 10.1 ビルド 122.17 以降、ユーザモニタ用のスクリプトファイルが新しい場所にあります。

  MPX または VPX 仮想アプライアンスをリリース 10.1 ビルド 122.17 以降にアップグレードする場合、変更点は次のとおりです。

  - /nsconfig/monitors/ に conflicts という名前の新しいディレクトリが作成され、以前のビルドのすべての組み込みスクリプトがこのディレクトリに移動されます。
  - すべての新しい組み込みスクリプトは、/netscaler/monitors/ ディレクトリで利用できます。すべてのカスタムスクリプトは、/nsconfig/monitors/ ディレクトリにあります。
  - 新しいカスタムスクリプトを /nsconfig/monitors/ ディレクトリに保存します。
  - アップグレードの完了後、カスタムスクリプトが作成され、組み込みスクリプトと同じ名前で /nsconfig/monitors/ ディレクトリに保存されると、/netscaler/monitors/ ディレクトリ内のスクリプトが優先されます。カスタムスクリプトは実行されません。

  リリース 10.1 ビルド 122.17 以降で仮想アプライアンスをプロビジョニングする場合、変更内容は次のとおりです。

  - すべての組み込みスクリプトは、/netscaler/monitors/ ディレクトリにあります。
  - /nsconfig/monitors/ ディレクトリが空です。
  - カスタムスクリプトを作成する場合は、/nsconfig/monitors/ ディレクトリに保存する必要があります。

スクリプトが正しく機能するには、次の手順を実行します。

- スクリプト名の最大文字数は 63 文字以下でなければなりません。
- スクリプトに指定できるスクリプト引数の最大数は 512 を超えないようにしてください。
- パラメータースクリプト引数に指定できる最大文字数は 639 文字以下でなければなりません。

スクリプトをデバッグするには、CLI から nsumon-debug.pl スクリプトを使用してスクリプトを実行する必要があります。nsumon-debug.pl スクリプトの引数として、スクリプト名 (引数付き)、IP アドレス、およびポートを

使用します。ユーザーは、nsumon-debug.pl スクリプトのスクリプト名、IP アドレス、ポート、タイムアウト、およびスクリプト引数を使用する必要があります。

CLI で、次のように入力します。

```
nsumon-debug.pl <scriptname> <IP> <port> <timeout> <partitionID> [
    scriptarguments][is_secure]
<!--NeedCopy-->
```

重要: リリース 10.5 ビルド 57.x 以降、ユーザーモニター用の 11.0 スクリプトファイルは IPv6 アドレスをサポートし、次の変更を含めました。

- 次のプロトコルでは、IPv6 サポート用の新しい`pm files`が含まれています。
  - RADIUS
  - NNTP
  - POP3
  - SMTP
- /netscaler/monitors/ の次のサンプルスクリプトが IPv6 サポート用に更新されました。
  - nsbmradius.pl
  - nsldap.pl
  - nsnntp.pl
  - nspop3 nssf.pl
  - nssnmp.pl
  - nswi.pl
  - nstftp.pl
  - nssmtp.pl
  - nsrdp.pl
  - nsntlm-lwp.pl
  - nsftp.pl
  - nsappc.pl

  リリース 10.5 ビルド 57.x または 11.0 にアップグレードした後、IPv6 サービスで既存のカスタムスクリプトを使用する場合は、の更新されたサンプルスクリプトで提供された変更で既存のカスタムスクリプトを更新してください。/netscaler/monitors/.

  注: サンプルスクリプト nsmysql.pl は IPv6 アドレスをサポートしていません。IPv6 サービスが nsmysql.pl を使用するユーザーモニターにバインドされている場合、プローブは失敗します。

- IPv6 アドレスをサポートするために、次の LB モニタタイプが更新されました。
  - USER
  - SMTP
  - NNTP
  - LDAP
  - SNMP
  - POP3
  - FTP_EXTENDED
  - StoreFront
  - APPC
  - CITRIX_WI_EXTENDED

  これらの LB モニタタイプのいずれかを使用するカスタムスクリプトを作成する場合は、カスタムスクリプトに IPv6 サポートを含めるようにしてください。IPv6 サポート用のカスタムスクリプトで行う必要がある変更については、/netscaler/monitors/ の関連するサンプルスクリプトを参照してください。

サーバーのステータスを追跡するために、モニターは設定されたディスパッチャに HTTP POST 要求を送信します。この POST リクエストには、サーバーの IP アドレスとポート、および実行する必要があるスクリプトが含まれます。ディスパッチャは、ユーザー定義パラメータ (存在する場合) を使用して、スクリプトを子プロセスとして実行します。次に、スクリプトはプローブをサーバーに送信します。スクリプトは、プローブのステータス (応答コード) をディスパッチャに送信します。ディスパッチャは、レスポンスコードを HTTP レスポンスに変換し、モニタに送信します。HTTP 応答に基づいて、モニタはサービスをアップまたはダウンとしてマークします。

Citrix ADC アプライアンスは、エラーメッセージを /var/nslog/nsumond.log ユーザーモニタープローブが失敗した場合のファイル。これらの詳細なエラーメッセージは、GUI および`show service`/`service group`コマンドの CLI に表示されます。

次の表は、ユーザーモニターと、考えられる失敗の原因の一覧です。

ユーザーモニターの種類	プローブ失敗の理由
SMTP	モニタがサーバへの接続を確立できない。
NNTP	モニタがサーバへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。
	モニタが NNTP グループを見つけられない。
LDAP	モニタがサーバへの接続を確立できない。

ユーザーモニターの種類	プローブ失敗の理由
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。
	モニタが LDAP サーバにバインドできない。
	モニタが LDAP サーバでターゲットエンティティのエントリを見つけない。
FTP	サーバーへの接続がタイムアウトします。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。
	ログオンが失敗する。
	モニタがサーバ上のファイルを検出できない。
POP3	モニタがデータベースへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。
	ログオンが失敗する。
POP3	モニタがデータベースへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。
	ログオンが失敗する。
	SQL クエリの準備が失敗します。
	SQL クエリの実行が失敗します。
SNMP	モニタがデータベースへの接続を確立できない。
	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。
	ログオンが失敗する。
	モニタが SNMP セッションを作成できない。
	モニタがオブジェクト識別子を見つけられない。
	モニタのしきい値の設定は、モニタの実際のしきい値以上です。
RDP (Windows ターミナルサーバー)	スクリプト引数がないか無効です。無効な数の引数または引数の形式が含まれる可能性があります。
	モニタがソケットを作成できない。

ユーザーモニターの種類	プローブ失敗の理由
	バージョンの不一致。
	モニタが接続を確認できない。

CLI からログファイルを表示するには、次のコマンドを使用します。このコマンドは BSD シェルを開き、ログファイルを画面に表示し、BSD シェルを閉じて CLI に戻ります。

```
> shell
root@ns# cat /var/nslog/nsumond.log
root@ns# exit
>
<!--NeedCopy-->
```

Citrix ADC リリース 13.0 ビルド 52.X より前の`show service/service group`コマンドでは、ユーザーモニターのプローブの失敗の原因として「プローブに失敗しました」という一般的なエラーメッセージが表示されました。

例:

```
show service ftp

Monitor Name: mon2
State: UNKNOWN  Weight: 1   Passive: 0
Probes: 3    Failed [Total: 0 Current: 0]
Last response: Failure - Probe failed.
Response Time: 1071.838 millisec
<!--NeedCopy-->
```

Citrix ADC リリース 13.0 ビルド 52.X 以降では、`show service/service group`コマンドがユーザーユーザーモニタープローブの失敗の実際の原因を表示します。

例:

```
show service ftp

Monitor Name: mon2
State: DOWN   Weight: 1   Passive: 0
Probes: 729   Failed [Total: 726 Current: 726]
Last response: Failure - Login failed.
```



```
7 Response Time: 8000.0 millisec
8 <!--NeedCopy-->
```

ユーザーモニターには、タイムアウト値とプローブ障害の再試行回数もあります。ユーザーモニターは、ユーザーモニター以外のモニターでも使用できます。CPU 使用率が高い場合、非ユーザモニタを使用すると、サーバ障害をより迅速に検出できます。

CPU 使用率が高いときにユーザモニタプローブがタイムアウトしても、サービスの状態は変更されません。

Example1:

```
1 add lb monitor <name> USER -scriptname <script-name> -resptimeout 5
     seconds
2 <!--NeedCopy-->
```

> 注:
>
> スクリプト可能なモニタの場合、応答タイムアウトは、予想されるタイムアウト + 1 秒に等しい値に設定する必要があります。たとえば、タイムアウトが 4 秒になると予想される場合、応答タイムアウトを 5 秒に設定します。

Example2:

```
1 add lb monitor <name> USER -scriptname <script-name> -scriptargs <
     Arguments> -secureargs <Arguments>
2 <!--NeedCopy-->
```

> 注:
>
> スクリプトに関連する機密データの場合は、パラメーターscriptargsの代わりにパラメーターsecureargsを使用することをお勧めします。

ユーザーモニターを使用して Web サイトを確認する方法

October 7, 2021

特定の HTTP コードを使用して HTTP サーバーによって報告される特定の Web サイトの問題をチェックするように、ユーザーモニターを構成できます。次の表は、このユーザーモニターが期待する HTTP 応答コードの一覧です。

HTTP レスポンス・コード	意味
200-成功	成功のプローブ。
503-サービスは利用できません	プローブの失敗。
404-見つかりません	スクリプトが見つからないか、実行できません。
500-内部サーバエラー	ディスパッチャの内部エラー/リソース制約 (メモリ不足、接続が多すぎる、予期しないシステムエラー、またはプロセスが多すぎる)。サービスに DOWN のマークが付けられていません。
400-不正な要求	HTTP 要求の解析中にエラーが発生しました。
502-不正なゲートウェイ	スクリプトの応答のデコード中にエラーが発生しました。

HTTP のユーザーモニターを構成するには、次のパラメーターを使用します。

パラメーター	Specifies
scriptName	実行するスクリプトのパスと名前。
scriptArgs	POST データに追加される文字列。それらはそのままリクエストにコピーされます。
dispatcherIP	プローブの送信先となるディスパッチャの IP アドレス。
dispatcherPort	プローブの送信先であるディスパッチャのポート。
localfileName	ローカルシステム上のモニタスクリプトファイルの名前。
destPath	アップロードされたローカルファイルが保存されている Citrix ADC アプライアンス上の特定の場所。

HTTP を監視するユーザーモニターを作成するには、負荷分散セットアップでのモニタの構成を参照してください。

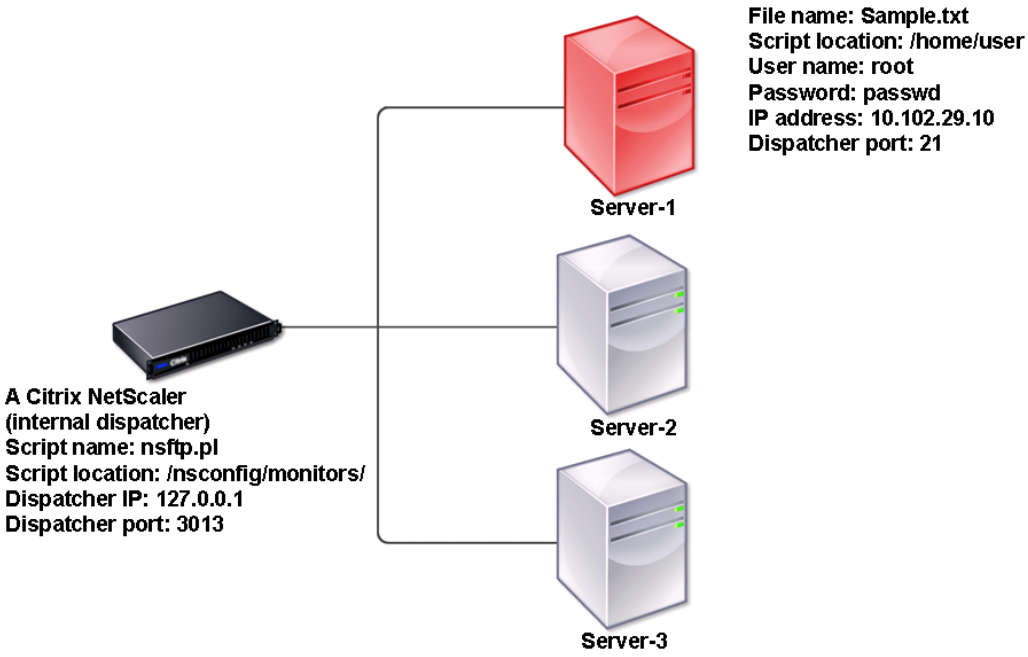
内部ディスパッチャの理解

October 7, 2021

内部ディスパッチャでカスタムユーザーモニターを使用できます。サーバー上のファイルの存在に基づいて、サーバ

ーの健全性を追跡する必要がある場合を考えてみましょう。次の図は、このシナリオを示しています。

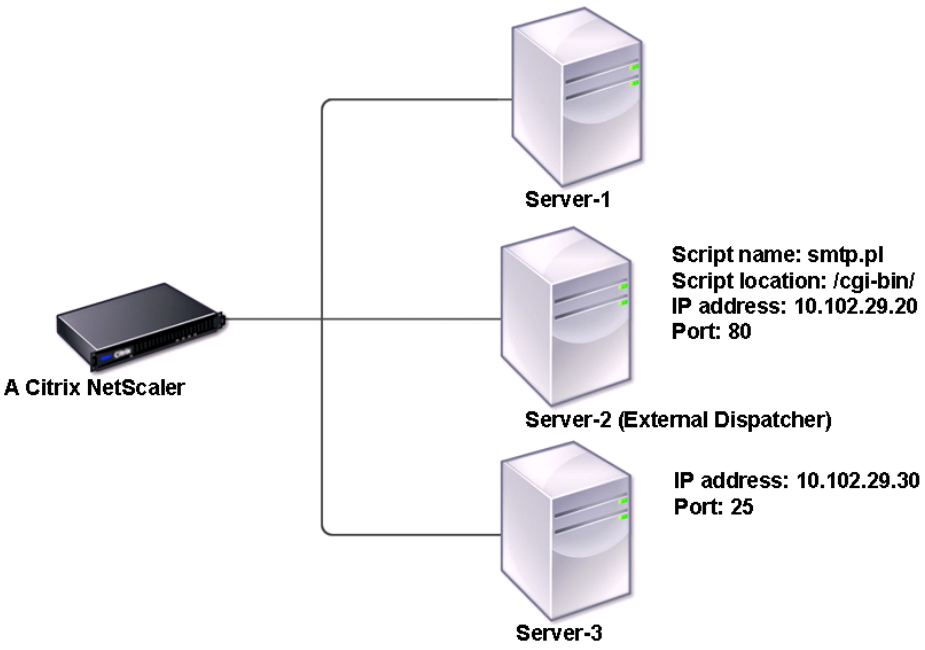
図 1：内部ディスパッチャを使用したユーザーモニタの使用



考えられる解決策は、サーバーとの FTP セッションを開始し、ファイルの存在をチェックする Perl スクリプトを使用することです。その後、Perl スクリプトを使用するユーザーモニターを作成できます。Citrix ADC アプライアンスでは、/nsconfig/monitors/ ディレクトリに、このような Perl スクリプト（nsftp.pl）が含まれています。

外部ディスパッチャでユーザーモニターを使用できます。別のサーバー上の SMTP サービスの状態に基づいて、サーバーの正常性を追跡する必要がある場合を考えてみましょう。このシナリオを次の図に示します。

図 2：外部ディスパッチャを使用したユーザーモニタの使用



可能な解決策は、サーバー上の SMTP サービスの状態をチェックする Perl スクリプトを作成することです。その後、Perl スクリプトを使用するユーザーモニターを作成できます。

ユーザーモニターを構成する

August 12, 2022

ユーザーモニターは、Citrix ADC アプライアンスがサポートしていないカスタムアプリケーションとプロトコルの健全性を追跡します。これは、カスタムモニターの拡張スコープです。ユーザモニタを設定するには、次の手順を実行する必要があります。

- バインドされたサービスを監視できるスクリプトを記述します。
- スクリプトを Citrix ADC アプライアンスの/nsconfig/monitorsディレクトリにアップロードします。
- スクリプトに実行権限を与えます。

モニタタイプがアプライアンスでサポートされていないプロトコルの場合は、タイプ **USER** のモニタを使用する必要があります。ユーザーモニターは Perl および Bash タイプのスクリプトのみをサポートします。Python スクリプトはサポートしていません。

> 注
>
> モニタープローブは NSIP アドレスから発信されます。モニタタイプ **USER** に設定されたscriptargsは、実行構成ファイルと ns.conf ファイルに表示されます。

モニターの詳細については、「モニターの構成」を参照してください。

CLI を使用してユーザモニタを設定するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> USER -scriptname <NameOfScript> -
    scriptargs <Arguments> -secureargs <Arguments>
<!--NeedCopy-->
```

Example1:

```
add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs ”file
    =/home/user/
sample.txt;user=root;password=passwd”
<!--NeedCopy-->
```

Example2:

```
add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs ”file
    =/home/user/
sample.txt -secureargs ”user=root;password=passwd”
<!--NeedCopy-->
```

> 注
>
> secureargsパラメータは、スクリプト引数をプレーンテキスト形式ではなく暗号化された形式で格納します。スクリプトに関連する機密データ（ユーザー名やパスワードなど）には、scriptargs パラメーターの代わりにパラメーターsecureargsを使用することをお勧めします。両方のパラメータを一緒に使用する場合、-scriptnameで指定されたスクリプトは、<scriptargs> <secureargs>の順序で引数を受け入れる必要があります。最初のいくつかの引数を<scriptargs>パラメータに指定し、残りの引数を<secureargs>パラメータに指定します。つまり、引数に定義された順序を維持します。Secure 引数は内部ディスパッチャにのみ適用されます。外部ディスパッチャーを使用する場合は、スクリプト内の脆弱なデータを保護することをお勧めします。

例 **3:**

scriptargsパラメータに「a=b; c=d; e=f」という引数を設定しているとします。

```
add monitor mon1 USER -scriptargs ”a=b;c=d;e=f”
<!--NeedCopy-->
```

scriptargsパラメータの代わりにsecureargsパラメータを使用する場合は、次の操作を行います。

- scriptargsパラメータを無効にします。
- secureargsパラメータの下にすべての引数を指定します。

```
set monitor mon1 USER -scriptargs ”” -secureargs ”a=b;c=d;e=f”
<!--NeedCopy-->
```

GUI を使用してユーザモニタを設定するには

1. [トラフィック管理] **>** [負荷分散] **>** [モニタ] に移動し、[追加] をクリックします。
2. [モニターの作成] ページで、次の操作を行います。
   - モニタータイプとして **USER** を選択します。
   - ドロップダウンメニューからスクリプトを選択するか、独自のスクリプトをアップロードします。
   - [スクリプト引数] フィールドと [** セキュア引数 **] フィールドに適切な値を入力します。
   - [**Create**] をクリックします。

   ユーザーモニターが作成されます。

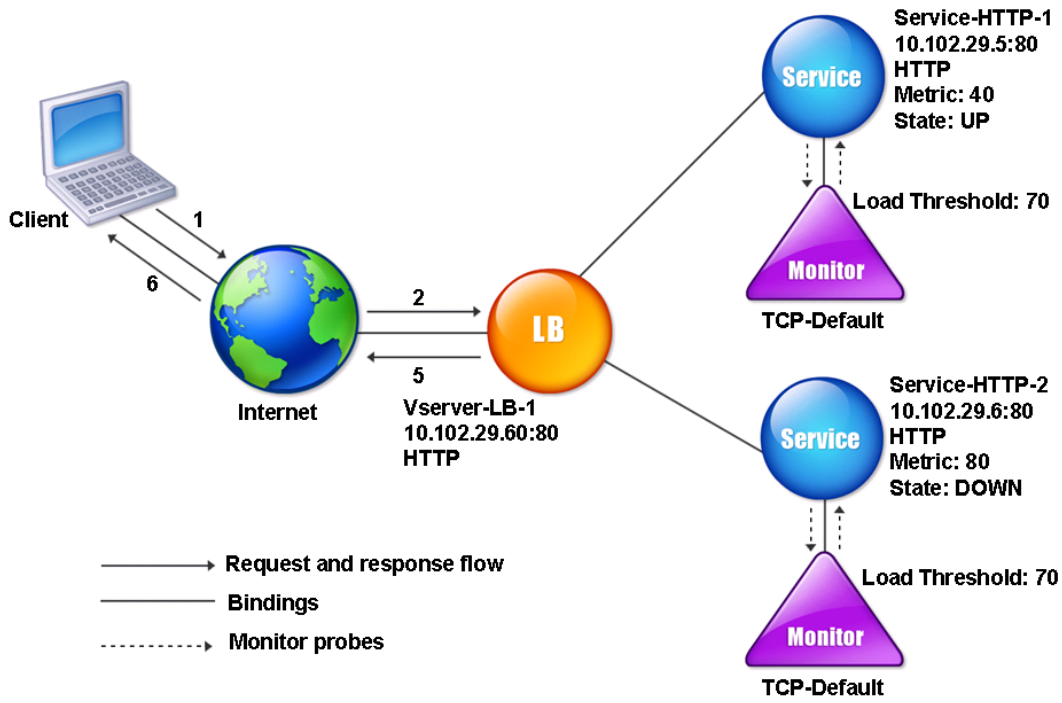
負荷モニタの理解

October 7, 2021

ロードモニタは、SNMP ポーリングされた OID を使用して負荷を計算します。ロードモニタは、バインドされているサービスの IP アドレス（宛先 IP アドレス）をポーリングに使用します。SNMP クエリーをサービスに送信し、メトリックの OID を指定します。メトリックには、CPU、メモリ、またはサーバー接続数を指定できます。サーバーは、メトリック値でクエリに応答します。レスポンスのメトリックス値がしきい値と比較されます。Citrix ADC アプライアンスは、メトリックがしきい値より小さい場合にのみ、負荷分散のためのサービスを考慮します。負荷値が最も小さいサービスが最初に考慮されます。

次の図は、基本負荷分散の設定で説明されている基本的な負荷分散設定で説明されているサービスに対して構成された負荷モニタを示しています。

図 1: 負荷モニタの動作



注: 負荷モニターは、サービスの状態を決定しません。アプライアンスは、負荷分散のためのサービスを考慮するだけを可能にします。

ロードモニタを設定したら、モニタが使用するメトリックを設定する必要があります。負荷評価では、負荷モニターはメトリックスと呼ばれるサーバーパラメーターを考慮します。このパラメーターは、アプライアンス構成のメトリックステーブル内で定義されます。メトリックテーブルには、次の 2 つのタイプがあります。

- ローカル。デフォルトでは、このテーブルはアプライアンス内に存在します。このメトリックは、接続、パケット、応答時間、帯域幅の 4 つのメトリックで構成されます。アプライアンスはサービスに対してこれらのメトリックを指定し、これらのサービスに対して SNMP クエリーは発信されません。これらのメトリックは変更できません。
- カスタム。ユーザー定義テーブル。各メトリックは OID に関連付けられています。

デフォルトでは、アプライアンスは次のテーブルを生成します。

- NetScaler
- RADWARE
- CISCO-CSS
- LOCAL
- FOUNDRY

- ALTEON

アプライアンスで生成されたメトリック・テーブルを追加することも、次の表に示すように、自分で選択したテーブルを追加することもできます。指標テーブルの値は、例としてのみ提供されています。実際のシナリオでは、メトリックの実際の値について検討します。

メトリック名	OID	重要度	しきい値
CPU	1.2.3.4	2	70
メモリ	4.5.6.7	3	80
接続	5.6.7.8	4	90

1 つ以上のメトリックスの負荷を計算するには、各メトリックスに重みを割り当てます。デフォルトのウェイトは 1 です。重みは、各メトリックに与えられた優先順位を表します。重みが大きい場合、優先順位は高いです。アプライアンスは、SOURCEIPDESTIP ハッシュアルゴリズムに基づいてサービスを選択します。

また、各メトリックスのしきい値を設定することもできます。しきい値により、アプライアンスは、サービスのメトリック値がしきい値よりも小さい場合に、ロードバランシングするサービスを選択できます。しきい値によって、各サービスの負荷も決まります。

負荷監視の構成

October 7, 2021

ロードモニタを設定するには、最初にロードモニタを作成します。モニターの作成手順については、モニターの作成を参照してください。次に、メトリックテーブルを選択するか、作成して、サーバーの状態を決定するメトリックのセットを定義し、（メトリックテーブルを作成した場合）各メトリックをメトリックテーブルにバインドします。

コマンドラインインターフェイスを使用してメトリックテーブルを作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add lb metricTable <metricTableName>

bind lb metricTable <metricTableName> <metric> <SNMPOID>
<!--NeedCopy-->
```

例:


```
add metricTable Table-Custom-1

bind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5 11
<!--NeedCopy-->
```

構成ユーティリティを使用してメトリック表を作成し、その表にメトリックをバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [メトリックテーブル] に移動し、メトリックテーブルを作成します。
2. メトリックをバインドするには、[バインド] をクリックし、メトリックと SNMP OID を指定します。

メトリックス表からメトリックスをバインド解除する

October 7, 2021

メトリックスを変更する必要がある場合、またはメトリックステーブルを完全に削除する場合は、メトリックステーブルからメトリックスをバインド解除できます。

コマンドラインインターフェイスを使用してメトリックテーブルからメトリックをバインド解除するには

コマンドプロンプトで入力します。

```
unbind lb metricTable <metricTable> <metric>
<!--NeedCopy-->
```

例**:**

```
unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5
<!--NeedCopy-->
```

構成ユーティリティを使用してメトリック表からメトリックをバインド解除するには

1. [トラフィック管理] > [ロードバランシング] > [メトリックテーブル] に移動します。
2. メトリックテーブルを開き、メトリックを選択して [**Delete**] をクリックします。

名前やタイプなど、構成されたすべてのメトリックテーブルの詳細を表示して、メトリックテーブルが内部であるか、作成されて構成されているかを判断できます。

サービスのリバース監視を構成する

October 7, 2021

リバースモニタは、プローブ基準が満たされた場合はサービスを DOWN としてマークし、満たされなかった場合は UP としてマークします。たとえば、プライマリサービスが DOWN のときだけバックアップサービスがトラフィックを受信するようにする場合は、リバースモニタをセカンダリサービスにバインドし、プライマリサービスをプローブするように設定します。

Citrix ADC アプライアンスは、次のリバースモニターをサポートしています。

- HTTP
- ICMP
- TCP (リリース 11.1 ビルド 49.x から)

サービスの **HTTP** リバースモニタリングの設定

次の表に、サービスの HTTP ダイレクト監視およびリバース監視の条件を示します。

条件	直接	[反転]
接続が確立されていません。	失敗します	失敗します
HTTP 応答コードはプローブの仕様と一致します。	成功	失敗します
HTTP 応答コードがプローブの仕様と一致しません。	失敗します	成功
プローブがタイムアウトしました。	失敗します	失敗します

CLI を使用してサービスの **HTTP** リバースモニタリングを設定するには

コマンドプロンプトで入力します。

```
1 add lb monitor <Monitor_Name> HTTP -respCode 200 -httpRequest ”HEAD /”
     -destIP <Primary_Service_IP_Address> -destPort 80 -reverse YES
2
3 bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
4 <!--NeedCopy-->
```


サービスの **ICMP** リバースモニタリングの設定

次の表に、サービスの ICMP 直接監視および逆監視の条件を示します。

条件	直接	[反転]
ICMP エコー応答を受信します。	成功	失敗します
プローブがタイムアウトしました。	失敗します	成功

CLI を使用してサービスの **ICMP** リバース監視を構成するには

コマンドプロンプトで入力します。

```
add lb monitor <Monitor_Name> PING -destIP <Primary_Service_IP_Address>
    -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

サービスの **TCP** リバースモニタリングの設定

ダイレクト TCP モニタがモニタプローブに応答して RESET を受信すると、サービスは DOWN とマークされます。ただし、リバース TCP モニタが RESET 応答を受信すると、プローブは成功したと見なされ、サービスが UP とマークされます。

次の表に、サービスの TCP リバース監視の条件を示します。

条件	直接	[反転]
TCP 接続が確立されます。	成功	失敗します
プローブがタイムアウトしました。	失敗します	失敗します
プローブに対する応答はリセットされます。	失敗します	成功

CLI を使用してサービスの **TCP** リバース監視を構成するには

コマンドプロンプトで入力します。


```
add lb monitor <Monitor_Name> TCP  -destip <Primary_Service_IP_Address>
     -destport <primary_service_port>  -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

GUI を使用してリバース監視を構成するには

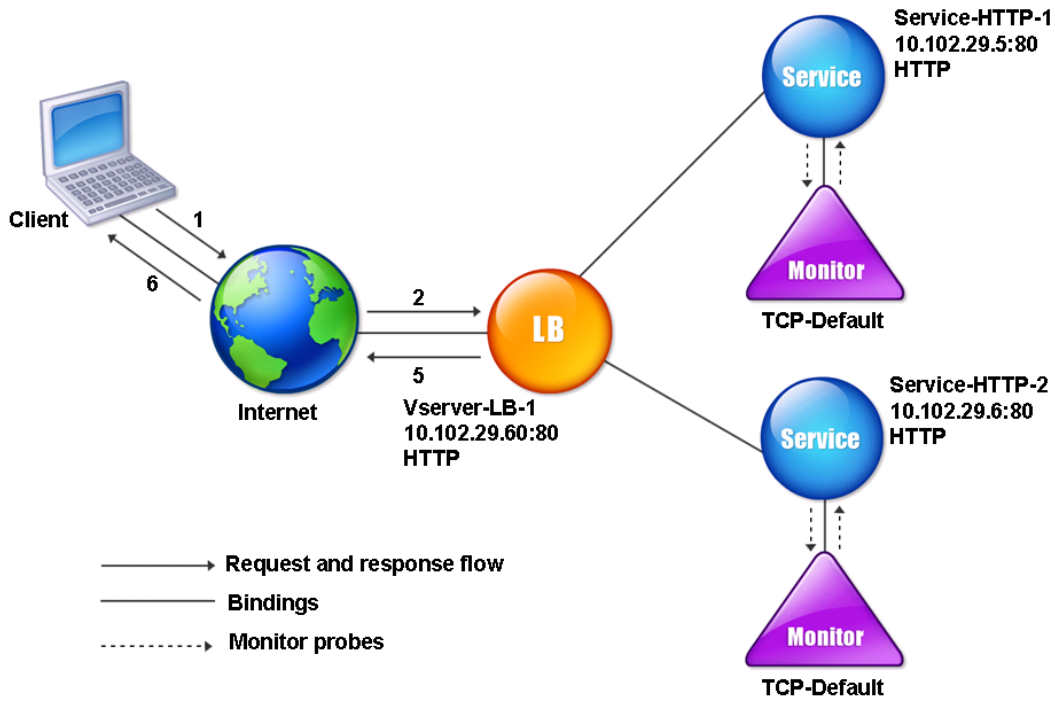
1. [トラフィック管理] > [負荷分散] > [モニタ] に移動します。
2. HTTP、ICMP、または TCP モニタを作成し、[リバース] を選択します。

負荷分散セットアップでのモニタの構成

October 7, 2021

Web サイトでモニターを構成するには、まずビルトインモニターを使用するか、独自のモニターを作成するかを決定します。モニターを作成する場合は、組み込みモニターに基づいてモニターを作成するか、作成したスクリプトを使用してサービスをモニターするカスタムモニターを作成するかを選択できます。カスタムモニターの作成の詳細については、「カスタムモニター」を参照してください。モニターを選択または作成したら、それを適切なサービスにバインドします。モニター名の長さは最大 255 文字です。次の概念図は、モニターを使用した基本的な負荷分散のセットアップを示しています。

図 1: モニターの動作方法



図のように、各サービスにはモニタがバインドされています。モニターは、サービスを介して負荷分散サーバーをプローブします。負荷分散されたサーバーがプローブに応答する限り、モニターはサーバーにマークを付けます。負荷分散されたサーバが、指定された時間内に指定された数のプローブに応答しなかった場合、モニタはそのプローブを DOWN とマークします。

ここでは、次の詳細について説明します。

- モニターの作成
- サービスの状態を判断するための監視パラメータの設定
- モニタとサービスのバインド
- モニターの変更
- モニターの有効化と無効化
- モニターのバインド解除
- モニターの取り外し
- モニターの表示
- モニター接続を閉じる
- モニタプローブのクライアント接続の上限を無視する

モニターの作成

October 7, 2021

Citrix ADC アプライアンスは、一連の組み込みモニターを提供します。また、組み込みモニターに基づいて、または最初からカスタムモニターを作成することもできます。

CLI を使用してモニターを作成するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <monitorType> [<interval>]

add lb mon monitor-HTTP-1 HTTP

add lb mon monitor-HTTP-2 TCP 2
<!--NeedCopy-->
```

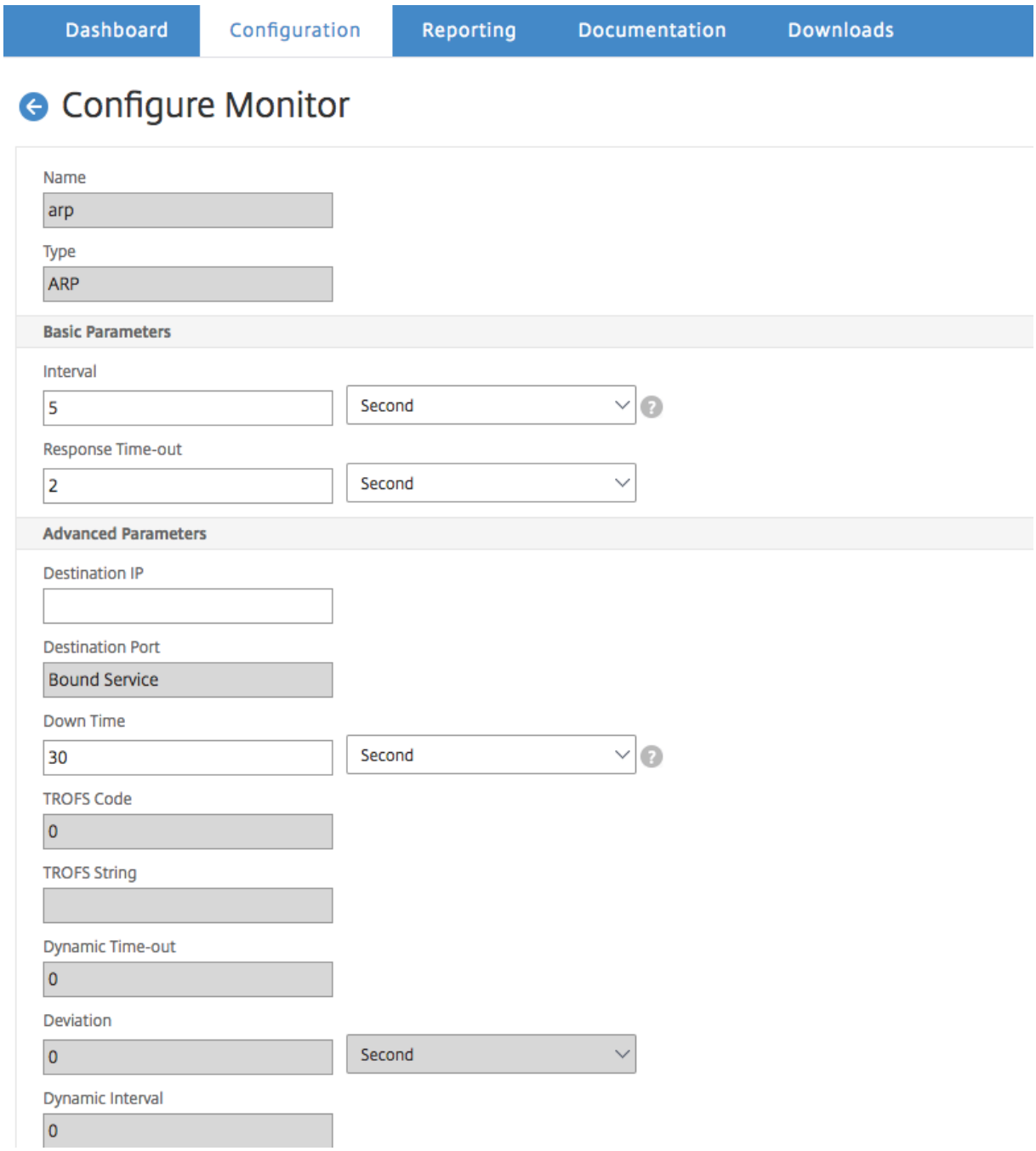
GUI を使用してモニターを作成するには

1. [トラフィック管理] > [負荷分散] > [モニター] に移動します。
2. [追加] をクリックし、要件を満たすモニタタイプを作成します。

[モニタの作成] 画面には、**[**基本パラメータ**]** と **[**拡張パラメータ**]** の 2 つのセクションがあります。

モニタの種類に応じて、[基本パラメータ**]** セクションには、モニタごとに設定する必要のあるパラメータが表示されます。**[**拡張パラメータ**]** セクションには、高度な使用例で使用できるパラメータが含まれています。

次の図は、ARP モニタータイプの [モニターの作成] ページのサンプルです。



> 注
>
> NetScaler リリース 12.0 ビルド 56.20 より前では、基本パラメーターと詳細パラメーターは、それぞれ標準パラメーターと特殊パラメーターと呼ばれていました。

サービスの状態を決定するためのモニタパラメータの設定

October 7, 2021

次のモニタリングパラメータを設定して、モニタリングプローブに基づいてサービスを DOWN としてマークできます。

再試行

監視プローブが失敗したサービスの状態を確立するために送信するプローブの最大数。

failureRetries

サービスが DOWN としてマークされるために、Retries パラメーターに指定された数のうち、失敗する必要がある再試行の数。たとえば、Retries パラメータが 10 に設定され、Failure Retries パラメータが 6 に設定されている場合、送信された 10 個のプローブのうち、サービスが DOWN としてマークされる場合は、少なくとも 6 つのプローブが失敗する必要があります。

alertRetries

アプライアンスが monProbeFailed と呼ばれる SNMP トラップを生成した後の連続したプローブ障害の数。

alertRetries を再試行値よりも大きい値に設定する

Citrix ADC アプライアンスが monProbeFailed と呼ばれる SNMP トラップを生成した後の連続する監視プローブ障害の最大数を指定する alertRetries パラメーターを、Retries 値（送信するプローブの最大数を指定する）よりも高い値に設定できるようになりました。監視プローブが失敗したサービスの状態を確立するため)。alertRetries の値が Retries の値よりも大きい場合、SNMP トラップは、サービスが DOWN するまで送信されません。

たとえば、[再試行] を 3、[alertRetries] を 12、[時間間隔] を 5 秒に設定した場合、サービスは 15 秒 (3*5)* 後に *DOWN* とマークされますが、アラートは生成されません。モニタープローブが *60* 秒後も失敗する場合 (*12*5)、Citrix ADC アプライアンスは monProbeFailed トラップを生成します。プローブが 15～60 秒の間に成功した場合、サービスは UP とマークされ、アラートは生成されません。

alertRetries 値を Retries 値よりも高い値に設定すると、正規のアラートのみを生成し、スケジュールされた再起動中に誤検知を回避するのに役立ちます。

コマンドラインインターフェイスを使用して、**alertRetries** パラメーター値を **Retries** 値よりも高い値に設定するには

コマンドプロンプトで入力します。


```
add lb monitor <monitorName> [-retries <integer>] [-alertRetries <
    integer>]
<!--NeedCopy-->
```

例**:**

add lb monitor monitor-HTTP-1 HTTP -retries 3 -alertRetries 12

GUI を使用して、**alertRetries** パラメーター値を **Retries** 値よりも高い値に設定するには

1. [設定] > [トラフィック管理] > [ロードバランシング] > [モニター] に移動します。
2. [**Add**] をクリックして新しいモニタを追加するか、既存のモニタを選択して [**Edit**] をクリックします。
3. [再試行] ボックスに、[再試行] パラメータの値を入力します。
4. [**SNMP** アラートの再試行] ボックスに、`alertRetries` パラメータの値を入力します。

モニタをサービスにバインドする

October 7, 2021

モニターを作成したら、それをサービスにバインドします。1 つまたは複数のモニターをサービスにバインドできます。1 つのモニターをサービスにバインドすると、そのモニターは、サービスが UP または DOWN のどちらとしてマークされているかを判別します。

複数のモニターをサービスにバインドする場合、Citrix ADC アプライアンスはすべてのモニターの状態をチェックしてから、サービスの状態を決定します。モニターにさまざまな重みを構成できます。モニターの重みは、そのモニターがサービスを UP または DOWN として指定するのにどの程度寄与するかを指定します。重量が大きいモニターほど、サービスを UP または DOWN としてマークする際の優先度が高くなります。デフォルトの重みは 1 です。したがって、モニターの 1 つに障害が発生した場合でも、サービスは DOWN としてマークされます。詳細については、「サービスにバインドされたモニターのしきい値を設定する」を参照してください。

注意: モニタープローブの宛先 IP アドレスは、サーバーの IP アドレスおよびポートとは異なる場合があります。

CLI を使用してモニタをサービスにバインドするには

コマンドプロンプトで入力します。

```
bind service <name> (-monitorName <string>)
<!--NeedCopy-->
```


例:

```
bind service s1 -monitorName tcp
<!--NeedCopy-->
```

GUI を使用してモニタをサービスにバインドするには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. サービスを開き、モニターを追加します。

モニタの変更

October 7, 2021

作成したモニターの設定を変更できます。

注: モニターには、タイプに関係なくすべてのモニターに適用されるパラメーターと、モニターのタイプに固有のパラメーターの 2 つのセットが適用されます。特定のモニター・タイプのパラメーターについては、そのタイプのモニターの説明を参照してください。

CLI を使用して既存のモニタを変更するには

コマンドプロンプトで入力します。

```
set lb monitor <monitorName> <type> -interval <interval> -resptimeout <
    resptimeout>
<!--NeedCopy-->
```

例:

```
set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli
<!--NeedCopy-->
```

GUI を使用して既存のモニタを変更するには

[トラフィック管理] **>** [負荷分散] **>** [モニタ] に移動し、変更するモニタを開きます。

モニタの有効化と無効化

October 7, 2021

デフォルトでは、サービスおよびサービスグループにバインドされたモニターが有効になっています。モニターを有効にすると、モニターはバインドされているサービスのプローブを開始します。サービスにバインドされたモニターを無効にすると、サービスにバインドされた他のモニターを使用してサービスが決定される状態になります。サービスが 1 つのモニターのみにバインドされている場合、およびモニターを無効にすると、サービスの状態はデフォルトのモニターを使用して決定されます。

CLI を使用してモニタを有効にするには

コマンドプロンプトで入力します。

```
enable lb monitor <monitorName>
<!--NeedCopy-->
```

例:

```
enable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

GUI を使用してモニタを有効にするには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. モニターを選択し、[アクション] リストから [有効] または [無効] を選択します。

CLI を使用してモニタを無効にするには

コマンドプロンプトで入力します。

```
disable lb monitor <monitorName>
<!--NeedCopy-->
```

例:


```
disable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

モニタのバインド解除

October 7, 2021

サービスおよびサービスグループからモニターのバインドを解除できます。モニターをサービスグループからバインド解除すると、モニターはサービスグループを構成する個々のサービスからバインド解除されます。モニターをサービスまたはサービスグループからバインド解除すると、モニターはサービスまたはサービスグループをプローブしません。

注：ユーザーが構成したすべてのモニターをサービスまたはサービス・グループからバインド解除すると、デフォルトのモニターはサービスおよびサービス・グループにバインドされます。次に、デフォルトのモニターがサービスまたはサービスグループをプローブします。

CLI を使用してサービスからモニタをバインド解除するには

コマンドプロンプトで入力します。

```
unbind lb monitor <monitorName>
<!--NeedCopy-->
```

例:

```
unbind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

GUI を使用してサービスからモニタをバインド解除するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、変更するサービスを開きます。
2. [モニター] セクションをクリックし、モニタを選択し、[バインド解除] をクリックします。

モニタの削除

October 7, 2021

サービスから作成したモニターのバインドを解除した後、Citrix ADC 構成からそのモニターを削除できます。（モニターがサービスにバインドされている場合、それを削除することはできません。）

注：サービスにバインドされているモニターを削除すると、デフォルトのモニターがサービスにバインドされます。デフォルトのモニターを削除することはできません。

CLI を使用してモニタを削除するには

コマンドプロンプトで入力します。

```
rm lb monitor <monitorName> <type>
<!--NeedCopy-->
```

例:

```
rm lb monitor monitor-HTTP-1 HTTP
<!--NeedCopy-->
```

GUI を使用してモニターを削除するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. モニタを選択し、[削除] をクリックします。

モニタの表示

October 7, 2021

モニターにバインドされているサービスとサービスグループを表示できます。モニターの設定を確認して、Citrix ADC 構成のトラブルシューティングを行うことができます。次の手順では、モニターのサービスおよびサービスグループへのバインディングを表示する手順について説明します。

CLI を使用してモニタバインディングを表示するには

コマンドプロンプトで入力します。

```
show lb monbindings <MonitorName>
<!--NeedCopy-->
```


例:

```
show lb monbindings monitor-HTTP-1
<!--NeedCopy-->
```

GUI を使用してモニタバインディングを表示するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. モニターを選択し、[アクション] リストで、[バインドの表示] をクリックします。

CLI を使用してモニターを表示するには

コマンドプロンプトで入力します。

```
show lb monitor <monitorName>
<!--NeedCopy-->
```

例:

```
show lb mon monitor-HTTP-1
<!--NeedCopy-->
```

GUI を使用してモニターを表示するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。使用可能なモニターの詳細が [モニター] ペインに表示されます。

モニタ接続を閉じる

October 7, 2021

Citrix ADC アプライアンスは、サービスにバインドされたモニターを介してプローブをサービスに送信します。デフォルトでは、アプライアンスと物理サーバーのモニターは、モニタープローブの場合でも完全なハンドシェイク手順に従います。ただし、この手順では、モニタリングプロセスにオーバーヘッドが加わり、必ずしも必要になるとは限りません。

TCP タイプのモニターの場合、サービスから SYN-ACK を受信した後、モニターとプローブの接続を閉じるようにアプライアンスを構成できます。これを行うには、monitorConnectionClose パラメーターの値を RESET に設定します。モニターとプローブの接続で完全な手順を実行する場合は、値を FIN に設定します。

注意: monitorConnectionClose 設定は、タイプ TCP および TCP-Default モニターにのみ適用できます。

コマンドラインインターフェイスを使用してモニター接続の閉鎖を構成するには、次の手順に従います。

コマンドプロンプトで入力します。

```
set lb parameter -monitorConnectionClose <monitor_conn_close_option>
<!--NeedCopy-->
```

例

```
set lb parameter -monitorConnectionClose RESET
<!--NeedCopy-->
```

構成ユーティリティを使用してモニター接続のクローズを構成するには、次のようにします。

1. [トラフィック管理] **>** [負荷分散] **>** [ロードバランシングパラメータの設定] に移動します。
2. [**FIN**] または [リセット] を選択します。

サービスまたはサービスグループレベルでモニター接続を閉じる

monConnectionClose パラメーターを設定することにより、サービスおよびサービスグループレベルでモニタープローブ接続を閉じるようにアプライアンスを構成することもできます。このパラメーターが設定されていない場合、モニター接続は、グローバル負荷分散パラメーターで設定された値を使用して閉じられます。このパラメーターがサービスまたはサービスグループレベルで設定されている場合、モニター接続は、FIN または RESET ビットが設定された接続終了メッセージをサービスまたはサービスグループに送信することによって閉じられます。

CLI を使用して、サービスレベルでモニター接続のクローズを構成するには

コマンドプロンプトで入力します。

```
set service <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```


CLI を使用して、サービスグループレベルでモニター接続の閉鎖を構成するには

コマンドプロンプトで入力します。

```
1  set serviceGroup <service_name> -monConnectionClose ( RESET | FIN )
2 <!--NeedCopy-->
```

GUI を使用して、サービスレベルでモニター接続の閉鎖を構成するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. サービスを追加または編集し、**[基本設定]** で [監視接続クローズビット] を設定します。

GUI を使用して、サービスグループレベルでモニター接続の閉鎖を構成するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを追加または編集し、**[基本設定]** で [モニタリングコネクションクローズビット] を設定します。

> 注：グローバル負荷分散パラメーターを使用してモニターとプローブの接続を閉じるには、monitorConnectionClose を FIN または RESET に構成できます。monitorConnectionClose パラメーターを次に設定する場合。
>
> - FIN：アプライアンスは完全な TCP ハンドシェイクを実行します。
> - RESET：アプライアンスは、サービスから SYN-ACK を受信した後、接続を閉じます。
>
> Citrix ADC CPX のより軽いバージョンでは、MonitorConnectionClose パラメータ値はデフォルトでリセットに設定され、グローバルレベルで FIN に変更することはできません。ただし、サービスレベルで monitorConnectionClose パラメーターを FIN に変更できます。

モニタプローブのクライアント接続の上限を無視する

October 7, 2021

物理サーバーの容量などの考慮事項に応じて、任意のサービスに対して行われるクライアント接続の最大数の制限を指定できます。サービスにこのような制限を設定した場合、Citrix ADC アプライアンスは、しきい値に達するとサービスへの要求の送信を停止し、既存の接続の数が制限内に収まった後、サービスへの接続の送信を再開します。モニタープローブ接続をサービスに送信するときにこのチェックをスキップするようにアプライアンスを構成できます。

注：個々のサービスの maximum-client-connections チェックをスキップすることはできません。このオプションを指定すると、Citrix ADC アプライアンスで構成されているすべてのサービスにバインドされているすべてのモニターに適用されます。

CLI を使用して **[**モニタ接続の **MaxClients** をスキップ**]** オプションを設定するには

コマンドプロンプトで入力します。

```
set lb parameter -monitorSkipMaxClient (ENABLED|DISABLED)
<!--NeedCopy-->
```

例**:**

```
set lb parameter -monitorSkipMaxClient enabled
<!--NeedCopy-->
```

GUI を使用して **[**モニタ接続の **MaxClients** をスキップ**]** オプションを設定するには

1. [トラフィック管理**] > [**負荷分散**] > [**ロードバランシングパラメータの設定] に移動します。
2. [接続を監視する **MaxClients** をスキップ] を選択します

大規模な展開の管理

October 7, 2021

Citrix ADC アプライアンスには、大規模な負荷分散展開を構成するときに役立ついくつかの機能が含まれています。仮想サーバとサービスを個別に設定する代わりに、仮想サーバとサービスのグループを作成できます。また、仮想サーバおよびサービスの範囲を作成したり、仮想サーバおよびサービス IP アドレスを変換またはマスクしたりできます。

仮想サーバーのグループに対して永続性を設定できます。モニタをサービスのグループにバインドできます。同じタイプの仮想サーバーとサービスの範囲を作成すると、それらのサーバーを 1 つの手順で設定および構成できます。これにより、これらの仮想サーバーとサービスの構成にかかる時間が大幅に短縮されます。

IP アドレスを変換またはマスキングすることで、仮想サーバーとサービスを停止できます。これにより、サービスおよび仮想サーバー定義を広範囲に再構成することなく、インフラストラクチャに変更を加えることができます。

仮想サーバとサービスの範囲

October 7, 2021

負荷分散を構成すると、仮想サーバとサービスの範囲を作成できるため、仮想サーバとサービスを個別に設定する必要がなくなります。たとえば、1 つの手順を使用して、対応する 3 つの IP アドレスを持つ 3 つの仮想サーバーを作成できます。複数の引数が 1 つの範囲を使用する場合、範囲は同じサイズでなければなりません。

次に、サービスおよび仮想サーバーを構成に追加するときに指定できる範囲のタイプを示します。

- 数値の範囲。1 つの数値を入力する代わりに、連続する数値の範囲を指定できます。

  たとえば、開始 IP アドレス (10.102.29.30 など) を指定し、その範囲を示す最後のバイト値 (34 など) を入力することで、仮想サーバーの範囲を作成できます。この例では、10.102.29.30 から 10.102.29.34 の範囲の IP アドレスを使用して 5 つの仮想サーバーが作成されます。

  注: 仮想サーバーとサービスの IP アドレスは連続している必要があります。

- アルファベットの範囲。リテラル文字を入力する代わりに、[C-G] などの任意の 1 文字の範囲を代入できます。これにより、範囲内のすべての文字（この場合は C、D、E、F、G）が含まれます。

  たとえば、Vserver-x、Vserver-y、Vserver-zという名前の仮想サーバーが 3 つある場合は、別々に構成する代わりに、「vserver [x-z]」と入力してすべてを構成できます。

さまざまな仮想サーバの作成

CLI を使用して仮想サーバーの範囲を作成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@[<rangeValue>]> <protocol> <IPAddress[<rangeValue
    >]> [<port>]
<!--NeedCopy-->
```

例**:**

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

または

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80

```



```
3 vserver ”vserverP” added
4
5 vserver ”vserverQ” added
6
7 vserver ”vserverR” added
8
9 Done
10 <!--NeedCopy-->
```

CLI を使用して仮想サーバーの範囲を作成するには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
1 add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]
2
3 add lb vserver <name>@\*\*[\*\*<rangeValue>\*\*]\*\*> <protocol> <
    IPAddress[<rangeValue>]> [<port>]
4 <!--NeedCopy-->
```

例**:**

```
1 add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
2 <!--NeedCopy-->
```

または

```
1 add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
2 vserver ”vserverP” added
3 vserver ”vserverQ” added
4 vserver ”vserverR” added
5 Done
6 <!--NeedCopy-->
```

GUI を使用して仮想サーバーの範囲を作成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを追加し、範囲を指定します。

さまざまなサービスの作成

サービス名の範囲を指定する場合は、IP アドレスの範囲も指定します。

CLI を使用してサービスの範囲を作成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add service <name>@ <IP>@ <protocol> <port>
<!--NeedCopy-->
```

例**:**

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service ”serv1” added
service ”serv2” added
service ”serv3” added
 Done
<!--NeedCopy-->
```

サービスグループの設定

October 7, 2021

サービスグループを設定すると、サービスグループを 1 つのサービスと同じくらい簡単に管理できます。たとえば、サービスグループに対して圧縮、ヘルスモニタリング、グレースフルシャットダウンなどのオプションを有効または無効にすると、サービスグループのすべてのメンバーに対してオプションが有効になります。

サービスグループを作成したら、仮想サーバにバインドし、そのグループにサービスを追加できます。モニタをサービスグループにバインドすることもできます。

サービスグループのメンバーは、IP アドレスまたはサーバ名で識別されます。

ドメインネームベースのサービス（DBS）グループメンバーを使用すると、メンバーの IP アドレスが変更された場合に Citrix ADC アプライアンスのメンバーを再構成する必要がないため、便利です。アプライアンスは、設定されたネームサーバを介してこのような変更を自動的に検出します。この機能は、サービスプロバイダーが物理サーバーを変更したり、サービスの IP アドレスを変更したりできるクラウドシナリオで役立ちます。DBS グループメンバーを指定すると、アプライアンスはその IP アドレスを動的に学習します。

IP ベースメンバーと DBS メンバーの両方を同じサービスグループにバインドできます。

注：DBS サービスグループメンバーを使用する場合は、ネームサーバーが指定されているか、Citrix ADC アプライアンスで DNS サーバーが設定されていることを確認してください。ドメイン名は、対応するアドレスレコードがアプライアンスまたはネームサーバに存在する場合のみ、IP アドレスに解決されます。

サービスグループの作成

Citrix ADC アプライアンスでは、最大 8192 のサービスグループを構成できます。

コマンドラインを使用してサービスグループを作成するには

コマンドプロンプトで入力します。

```
add servicegroup <ServiceGroupName> <Protocol>
<!--NeedCopy-->
```

例:

```
add servicegroup Service-Group-1 HTTP
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを作成するには

「トラフィック管理」>「ロードバランシング」>「サービスグループ」に移動し、サービスグループを追加します。

仮想サーバへのサービスグループのバインド

サービスグループを仮想サーバにバインドすると、メンバーサービスは仮想サーバにバインドされます。

コマンドラインインターフェイスを使用してサービスグループを仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind lb vserver <name>@ <serviceGroupName>
<!--NeedCopy-->
```

例:


```
bind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

GUI を使用してサービスグループを仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[サービスグループ] を選択します。

メンバーをサービスグループにバインドする

サービスグループにサービスを追加すると、サービスグループがサーバを管理できるようになります。IP アドレスまたはサーバ名を指定して、サーバをサービスグループに追加できます。

GUI で、ドメイン名ベースのサービスグループメンバーを追加する場合は、[**Server Based**] を選択します。

このオプションを使用すると、名前が IP アドレスであるか、ユーザーによって割り当てられた名前であるかに関係なく、名前が割り当てられた任意のサーバーを追加できます。

コマンドラインインターフェイスを使用してサービスグループにメンバーを追加するには

サービスグループを構成するには、コマンドプロンプトで次のように入力します。

```
bind servicegroup <serviceGroupName> (<IP>@ | <serverName>) <port>
<!--NeedCopy-->
```

例:

```
bind servicegroup Service-Group-1 10.102.29.30 80

bind servicegroup Service-Group-2 1000:0000:0000:0000:0005:0600:700a
    :888b 80

bind servicegroup CitrixEdu s1.citrite.net
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループにメンバーを追加するには

1. [トラフィック管理] **>** [負荷分散] **>** [サービスグループ] に移動し、サービスグループを開きます。

2. [Service Group] セクションをクリックし、次のいずれかの操作を行います。

   - IP ベースのサービスグループメンバーを追加するには、[IP ベース] を選択します。
   - サーバ名ベースのサービスグループメンバーを追加するには、[Server Based] を選択します。

   ドメイン名ベースのサービスグループメンバーを追加する場合は、[**Server Based**] を選択します。このオプションを使用すると、名前が IP アドレスであるか、ユーザーによって割り当てられた名前であるかに関係なく、名前が割り当てられた任意のサーバーを追加できます。

3. 新しい IP ベースのメンバーを追加する場合は、[IP アドレス] テキストボックスに IP アドレスを入力します。IP アドレスが IPv6 形式を使用する場合は、[IPv6] チェックボックスをオンにして、[IP アドレス] テキストボックスにアドレスを入力します。

   注:IP アドレスの範囲を追加できます。範囲内の IP アドレスは連続している必要があります。[IP アドレス] テキストボックスに開始 IP アドレスを入力して、範囲を指定します (10.102.29.30 など)。[Range] の下のテキストボックスに、IP アドレス範囲の終了バイトを指定します (35 など)。[ポート] テキストボックスにポート (80 など) を入力し、[追加] をクリックします。

4. ［作成］をクリックします。

モニタをサービスグループにバインドする

サービスグループを作成すると、そのグループに適切なタイプのデフォルトモニタが自動的にバインドされます。モニタは、バインドされているサービスグループ内のサーバを定期的にプローブし、サービスグループの状態を更新します。

自分で選択した別のモニタをサービスグループにバインドできます。

コマンドラインインターフェイスを使用してモニタをサービスグループにバインドするには

コマンドプロンプトで入力します。

```
bind serviceGroup <serviceGroupName> -monitorName <string> -monState (
    ENABLED | DISABLED)
<!--NeedCopy-->
```

例:

```
bind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```


構成ユーティリティーを使用してサービスグループにモニターをバインドする

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを開き、[詳細設定] で [モニター] をクリックします。

仮想サーバを無効にして有効にした後も、サービスグループメンバの元の状態を保持する

新しいグローバルオプション —retainDisableServer であるビルド 64.x から、サーバが無効になってから再び有効になったときに、サービスグループメンバーの状態を保持できます。

以前は、次の条件セットでメンバーの状態が DISABLED から ENABLED に変更されていました。

- 仮想サーバー上の同じポートに 2 つのアプリケーションがデプロイされます。
- 共通メンバーを持つ 2 つのサービスグループがこの仮想サーバにバインドされ、共通メンバーは一方のグループで有効になり、もう一方のグループでは無効になります。
- サーバが無効になり、再び有効になります。

これらの条件下では、サーバを無効にすると、すべてのサービスグループメンバーが無効になり、サーバを再度有効にすると、以前の状態に関係なく、デフォルトですべてのメンバーが有効になります。メンバーを元の状態に戻すには、サービスグループ内のメンバーを手動で無効にする必要があります。これは面倒な作業であり、エラーが発生しやすい。

サービスグループの管理

October 7, 2021

サービスグループのサービスの設定を変更したり、サービスグループの有効化、無効化、削除などのタスクを実行できます。サービスグループからメンバーをバインド解除することもできます。サービスグループの詳細については、「サービスグループの構成」を参照してください。

サービスグループの変更

サービスグループメンバーの属性を変更できます。最大クライアント、Sure Connect、圧縮など、サービスグループのいくつかの属性を設定できます。属性は、サービスグループの個々のサーバに設定されます。トランスポート情報（IP アドレスとポート）、重み、サーバ ID などのパラメータをサービスグループに設定することはできません。

注: サービスグループに設定したパラメーターは、個々のサービスではなく、グループ内のメンバーサーバーに適用されます。

コマンドラインインターフェイスを使用してサービスグループを変更するには

コマンドプロンプトで、次のコマンドを 1 つ以上の省略可能なパラメーターで入力します。


```
set servicegroup <serviceGroupName> [-type <type>] [-maxClient <
    maxClient>] [-maxReq <maxReq>] [-cacheable (YES|NO)] [-cip (ENABLED|
    DISABLED)] [-cipHeader <cipHeader>] [-usip (YES|NO)] [-sc (ON|OFF)]
    [-sp (ON|OFF)] [-cltTimeout <cltTimeout>] [-svrTimeout <svrTimeout>]
     [-cka (YES|NO)] [-TCPB (YES|NO)] [-CMP (\*\*YES\*\*|\*\*NO\*\*)] [-
    maxBandwidth <maxBandwidth>] [-maxThreshold <maxThreshold>] [-state
    (ENABLED|DISABLED)] [-downStateFlush (ENABLED|DISABLED)
<!--NeedCopy-->
```

例:

```
set servicegroup Service-Group-1 -type TRANSPARENT

set servicegroup Service-Group-1 -maxClient 4096

set servicegroup Service-Group-1 -maxReq 16384

set servicegroup Service-Group-1 -cacheable YES
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを変更するには

「トラフィック管理」>「ロードバランシング」>「サービスグループ」に移動し、変更するサービスグループを開きます。

サービスグループの削除

サービスグループを削除しても、グループにバインドされたサーバーは個々の設定を保持し、Citrix ADC アプライアンス上に引き続き存在します。

コマンドラインインターフェイスを使用してサービスグループを削除するには

コマンドプロンプトで入力します。

```
rm servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

例:


```
rm servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを削除するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[**Delete**] をクリックします。

サービスグループからのメンバーのバインド解除

サービスグループからメンバーのバインドを解除すると、サービスグループに設定された属性は、バインドを解除したメンバーには適用されなくなります。ただし、メンバーサービスは個々の設定を保持し、Citrix ADC アプライアンスに引き続き存在します。

コマンドラインインターフェイスを使用してサービスグループからメンバーのバインドを解除するには

コマンドプロンプトで入力します。

```
unbind servicegroup <serviceGroupName> <IP>@ [<port>]
<!--NeedCopy-->
```

例:

```
unbind servicegroup Service-Group-1 10.102.29.30 80
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループからメンバーのバインドを解除するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを開き、[サービスグループメンバー] セクションのをクリックします。
3. サービスグループメンバーを選択し、[**Unbind**] をクリックします。

仮想サーバからのサービスグループのバインド解除

仮想サーバーからサービスグループをバインド解除すると、メンバーサービスは仮想サーバーからバインド解除され、Citrix ADC アプライアンスに引き続き存在します。

コマンドラインインターフェイスを使用して仮想サーバからサービスグループをバインド解除するには

コマンドプロンプトで入力します。

```
unbind lb vserver <name>@ <ServiceGroupName>
<!--NeedCopy-->
```

例:

```
unbind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバからサービスグループをバインド解除するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバを開き、[Service Group] セクションのをクリックします。
3. サービスグループを選択し、[**Unbind**] をクリックします。

サービスグループからのモニタのバインド解除

モニタをサービスグループからバインド解除すると、バインド解除したモニタは、グループを構成する個々のサービスを監視しなくなります。

コマンドラインインターフェイスを使用してサービスグループからモニタをバインド解除するには

コマンドプロンプトで入力します。

```
unbind serviceGroup <serviceGroupName> -monitorName <string>
<!--NeedCopy-->
```

例:

```
unbind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```


構成ユーティリティを使用してサービスグループからモニタをバインド解除するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを開き、[Monitors] セクションのをクリックします。
3. モニタを選択し、[バインド解除] をクリックします。

サービスグループの有効化または無効化

サービスグループとサーバを有効にすると、そのサービスグループに属するサービスが有効になります。同様に、サービスグループに属するサービスを有効にすると、サービスグループとサービスが有効になります。デフォルトでは、サービスグループは有効になっています。

有効なサービスを無効にした後、構成ユーティリティまたはコマンドラインを使用してサービスを表示し、サービスが DOWN になるまでの残り時間を確認できます。

コマンドラインインターフェイスを使用してサービスグループを無効にするには

コマンドプロンプトで入力します。

```
disable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

例:

```
disable servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを無効にするには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[アクション] リストで [無効] をクリックします。

コマンドラインインターフェイスを使用してサービスグループを有効にするには

コマンドプロンプトで入力します。

```
enable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```


例:

```
enable servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを有効にするには

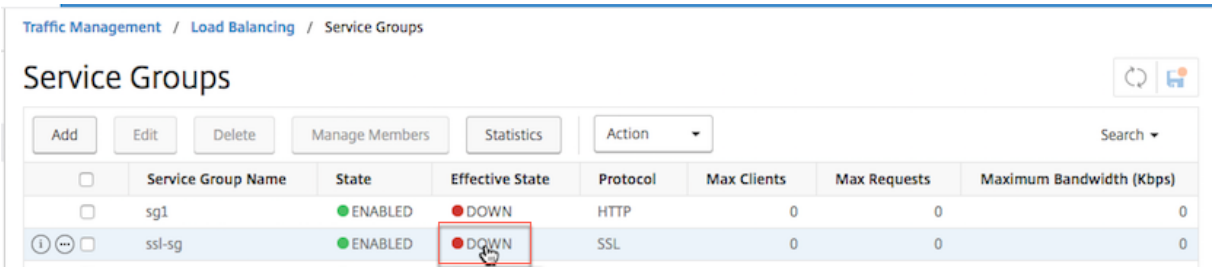
1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[アクション] リストで [**Enable**] をクリックします。

サービスグループメンバーのステータスの表示

Traffic Management > **Load Balancing** > **Service Groups** に移動します。

[サービスグループ] ページの [有効状態] 列には、サービスグループのステータスが表示されます。状態 UP/DOWN [有効状態] 列のクリック可能です。ステータスをクリックすると、同じビューでメンバーのリストとそのステータスを取得できます。メンバーを選択し、[モニターの詳細] ボタンをクリックして、ステータスが DOWN である理由を表示します。

注意: NetScaler リリース 12.0 ビルド 56.20 より前は、[有効状態] 列のステータスはクリックできませんでした。



サービスグループのプロパティの表示

構成されたサービスグループの次の設定を表示できます。

- 名前
- IP アドレス
- State
- プロトコル
- 最大クライアント接続
- 接続あたりの最大リクエスト数
- 最大帯域幅
- しきい値を監視する

設定の詳細を表示すると、設定のトラブルシューティングに役立ちます。

コマンドラインインターフェイスを使用してサービスグループのプロパティを表示するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、グループのプロパティ、またはプロパティとグループメンバを表示します。

```
show servicegroup <ServiceGroupName>

show servicegroup <ServiceGroupName> -includemembers
<!--NeedCopy-->
```

例:

```
show servicegroup Service-Group-1
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループのプロパティを表示するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループの横にある矢印をクリックします。

サービスグループ統計情報の表示

要求レート、応答、要求バイト数、応答バイト数などのサービスグループの統計データを表示できます。Citrix ADC アプライアンスは、サービスグループの統計を使用してサービスの負荷を分散します。

コマンドラインインターフェイスを使用してサービスグループの統計情報を表示するには

コマンドプロンプトで入力します。

```
stat servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

例:

```
stat servicegroup Service-Group-1
<!--NeedCopy-->
```


構成ユーティリティを使用してサービスグループの統計情報を表示するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[**Statistics**] をクリックします。

サービスグループにバインドされた仮想サーバのロードバランシング

大規模な展開では、同じサービスグループを複数の負荷分散仮想サーバーにバインドできます。この場合、各仮想サーバを表示してバインドされているサービスグループを表示する代わりに、サービスグループにバインドされているすべてのロードバランシング仮想サーバのリストを表示できます。各仮想サーバーの次の詳細を表示できます。

- 名前
- State
- IP アドレス
- ポート

コマンドラインインターフェイスを使用してサービスグループにバインドされた仮想サーバーを表示するには

コマンドプロンプトで次のコマンドを入力して、サービスグループにバインドされている仮想サーバーを表示します。

```
show servicegroupbindings <serviceGroupName>
<!--NeedCopy-->
```

例:

```
> show servicegroupbindings SVCGRPDTLS
SVCGRPDTLS - State :ENABLED
1) Test-pers (10.10.10.3:80) - State : DOWN
2) BRVSERV (10.10.1.1:80) - State : DOWN
3) OneMore (10.102.29.136:80) - State : DOWN
4) LBVIP1 (10.102.29.66:80) - State : UP
Done
>
<!--NeedCopy-->
```

設定ユーティリティを使用してサービスグループにバインドされた仮想サーバを表示するには

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. サービスグループを選択し、[アクション] リストで [バインドの表示] をクリックします。

1 つの NITRO API 呼び出しで、サービスグループに必要なサービスグループメンバーのセットを設定する

October 7, 2021

1 つの NITRO API コールで、サービスグループに必要なサービスグループメンバーセットを設定するためのサポートが追加されました。この構成をサポートするために、新しい API である [望ましい状態 API] が追加されました。必要な状態 API を使用すると、次の操作を実行できます。

- 「サービスグループメンバーリストバインディング」リソースに対する単一の PUT リクエストで、サービスグループメンバーのリストを提供します。
- その PUT リクエストに重量と状態（オプション）を指定します。
- アプライアンス構成を、アプリケーションサーバー周辺のデプロイメントの変更と効果的に同期します。

Citrix ADC アプライアンスは、要求された必要なメンバセットと構成済みのメンバセットを比較します。その後、自動的に新しいメンバーをバインドし、要求に存在しないメンバーのバインドを解除します。

注:

- この機能はタイプ`API.`のサービスのみでサポートされています。
- Desired State API を使用してのみ、IP アドレスベースのサービスをバインドできます。ドメイン名ベースのサービスは許可されません。
- 以前は、NITRO コールにバインドできるサービスグループメンバーは 1 つだけです。

> 重要
>
> ServiceGroup メンバーシップに必要な状態 API は、Citrix ADC クラスター展開でサポートされています。

ユースケース：Kubernetes などの大規模な展開環境で、展開の変更を Citrix ADC アプライアンスに同期する

大規模かつ高度に動的な展開（Kubernetes など）では、アプリケーショントラフィックを正確に処理するために、デプロイメントの変更率に応じてアプライアンスの構成を最新の状態に保つことが課題です。このような展開では、コントローラ（入力コントローラまたは E-W コントローラ）が ADC 構成の更新を担当します。デプロイメントに変更があるときはいつでも、`kube-api server`は「Endpoints イベント」を介して有効なエンドポイントセットをコントローラに送信します。Controller は、読み取り-デルタ-変更アプローチを使用して、次の処理を実行します。

- サービスに対して現在設定されているエンドポイントセット（サービスグループのサービスグループメンバーセット）を ADC アプライアンスからフェッチします。
- 設定されたエンドポイントセットを受信したイベント内のセットと比較します。
- 新しいエンドポイント（サービスグループメンバー）をバインドするか、削除されたエンドポイントのバインドを解除します。

この環境では変更率とサービスのサイズが高いため、この設定方法は効率的ではなく、設定の更新が遅れる可能性があります。

望ましい状態 API は、単一の API でサービスグループの意図されたメンバセットを受け入れることで問題を解決し、構成を効果的に更新します。

CLI を使用してタイプ API のサービスグループを作成する

コマンドプロンプトで次を入力します。

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>]
```

例:

```
add serviceGroup svg1 HTTP -autoScale API
```

autoDisablegracefulパラメーター、autoDisabledelayパラメーター、autoScaleパラメーターを設定するには、add serviceGroup コマンドまたは set serviceGroup コマンドを使用します。

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>] [-autoDisablegraceful ( YES | NO)] [-autoDisabledelay <
    secs>]

add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale (API |
    CLOUD | DISABLED| DNS |POLICY)]

set serviceGroup <serviceGroupName [-autoDisablegraceful ( YES | NO)]
    [-autoDisabledelay <secs>]

set serviceGroup <serviceGroupName [-autoScale (API |CLOUD | DISABLED|
    DNS |POLICY)]
```

例:

```
add serviceGroup svg1 HTTP autoDisablegraceful YES -autoDisabledelay
    100

add serviceGroup svg1 HTTP -autoScale API

set serviceGroup svg1 -autoDisablegraceful  YES -autoDisabledelay 100

```



```
7  set serviceGroup svg1 -autoScale API
```

引数

autoDisablegraceful

サービスの正常なシャットダウンを示します。このオプションを有効にすると、アプライアンスは、このサービスへの未処理の接続がすべて閉じられるのを待ってから、サービスを削除します。すでにシステム上に永続的なセッションを持つクライアントの場合、新しい接続または要求はこのサービスに送信され続けます。サービスメンバーは、未処理の接続がない場合にのみ削除されます。デフォルト値:NO

autoDisabledelay

正常なシャットダウンの許容時間（秒単位）を示します。この期間中、システム上に永続的なセッションがすでに存在するクライアントに対して、新しい接続または要求がこのサービスに送信され続けます。システム上に永続性セッションを持たない新しいクライアントからの接続または要求は、サービスに送信されません。代わりに、利用可能な他のサービス間で負荷分散されます。遅延時間が経過すると、サービスメンバーは削除されます。

Autoscale API

必要な状態 API を使用して、メンバセットを目的のサービスグループにバインドできるようにします。提供されたすべての条件が一致する場合は、サービスグループを非自動スケールから自動スケールタイプの Desired StateAPI に設定できます。

既存のメンバーバインディングが次のいずれかの条件を満たす場合、set serviceGroupAutoscale コマンドが失敗する可能性があります。

- サービスグループにバインドされたサーバがネームサーバまたはドメインベースのサーバである場合。
- サービスグループにバインドされているサーバの名前が IP アドレスの場合は、実際のサーバの IP アドレスと一致する必要があります。次の例では、サーバ名とサーバの IP アドレスが一致しません。
  - **CLI:** サーバの *IP* アドレスサーバ名を追加する
  - 例： add server 1.2.3.4 4.3.2.1
- ループバックサーバ名が 127.0.0.1 または 00:00:0000:00:0000:00:00:0000:00:0001 以外の場合。
- set serviceGroup コマンドでさまざまなタイプのオートスケール（クラウド、API、DNS、およびポリシー）を選択し、serviceGroup コマンドを追加する場合。

重要:

- autoDisablegraceful および autoDisabledelay パラメーターは、オートスケールタイプ「API」および「CLOUD」のサービスグループにのみ適用できます。
- autoDisablegraceful パラメータまたは autoDisabledelay パラメータが設定されていない場合、サービスメンバはただちに削除されます。

サービスグループメンバーを正常にバインド解除する

サービスグループメンバーのいずれかが目的のステートリストにない場合、autoDisablegraceful または autoDisabledelay パラメータ設定に基づいて、これらのメンバーは正常にバインド解除されます。

- これらのパラメータの 1 つが設定されている場合、サービスグループメンバーは正常にバインド解除されます。
- これらのパラメータのいずれも設定されていない場合、サービスグループメンバーはただちにバインド解除されます。

注:

- グレースフルバインド解除のために識別されたサービスグループメンバーは、show service group コマンドの実行時にだけ表示されます。
- 正常なアンバインドとして識別されたサービスグループメンバーに対して、操作（set、unset など）を実行できません。

次の図は、show servicegroup コマンドの例を示しています。

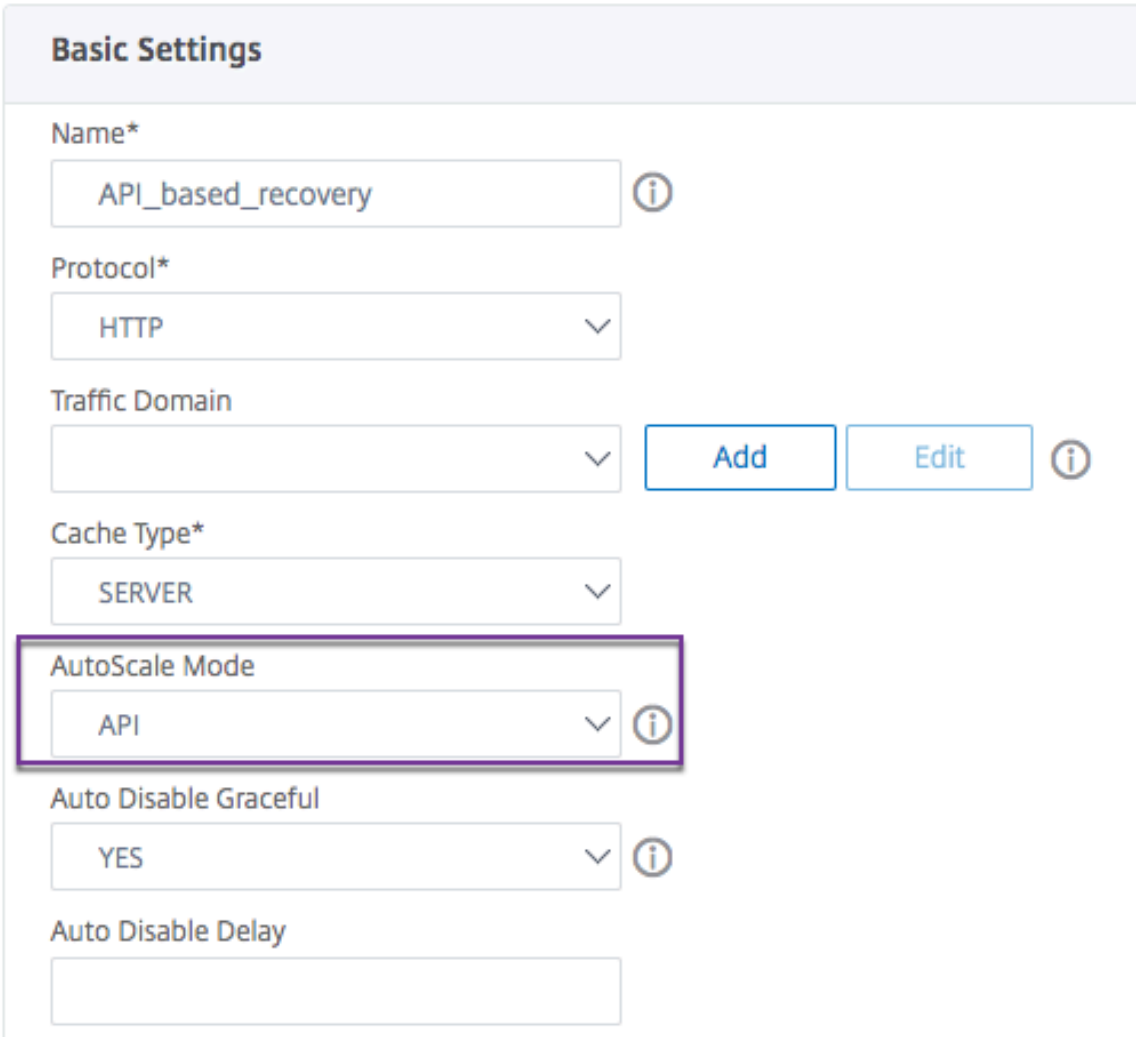
```
sh servicegroup sg1
        sg1 - HTTP
        State: ENABLED  Effective State: OUT OF SERVICE Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
        Monitor Connection Close : NONE
        Appflow logging: ENABLED
        Autoscale mode: API
        ContentInspection profile name: ???
        Process Local: DISABLED
        Traffic Domain: 0
        Unbind Graceful: NO
        Unbind Delay: 1000
```

GUI を使用してタイプ **API** のサービスグループを作成する

1. [トラフィック管理] **>** [負荷分散] **>** [サービスグループ] に移動し、[追加] をクリックします。
2. 「**AutoScale**」モードで、「**API**」を選択します。

GUI を使用して **API** タイプのサービスグループの正常なシャットダウンまたは遅延時間を設定します

1. [トラフィック管理] **>** [負荷分散] **>** [サービスグループ] に移動します。



2. 「**AutoScale**」モードで、「**API**」を選択します。
3. 「自動無効化」で、「はい」を選択します。
4. 「自動無効化遅延」に、正常なシャットダウンの待機時間を入力します。

注意**:** 「自動無効化」または「自動表示遅延」フィールドは、「**AutoScale**」モードで「**API** 」または「クラウド」を選択した場合にのみ有効になります。

ドメインベースのサービスグループの自動スケーリングの構成

February 21, 2022

ドメインベースのサービスグループは、サービスグループにバインドされたサーバーのドメイン名を解決することによって取得される IP アドレスを持つメンバーで構成されます。ドメイン名は、アプライアンス上で詳細を設定するネ

ームサーバによって解決されます。ドメインベースのサービスグループには、IP アドレスベースのメンバーを含めることもできます。

ドメインベースのサーバーの名前解決のプロセスでは、複数の IP アドレスが返されることがあります。DNS 応答に含まれる IP アドレスの数は、ネームサーバ上のドメイン名に設定されたアドレス (A) レコードの数によって決まります。名前解決プロセスが複数の IP アドレスを返す場合でも、1 つの IP アドレスだけがサービスグループにバインドされます。サービスグループをスケールアップまたはスケールダウンするには、サービスグループに対して他のドメインベースのサーバーを手動でバインドおよびバインド解除する必要があります。

ただし、ドメインベースのサーバーの DNS ネームサーバーから返される IP アドレスの完全なセットに基づいて自動的にスケーリングするように、ドメインベースのサービスグループを構成できます。自動スケーリングを設定するには、ドメインベースのサーバをサービスグループにバインドするときに、[Automatic Scaling] オプションを有効にします。自動的にスケーリングするドメインベースのサービスグループを設定する手順は、次のとおりです。

- ドメイン名を解決するためのネームサーバーを追加します。アプライアンスでのネームサーバの設定の詳細については、ネームサーバの追加を参照してください。
- ドメインベースのサーバーを追加します。ドメインベースのサーバーを追加する方法については、「サーバーオブジェクトの構成」を参照してください。
- サービスグループを追加し、ドメインベースのサーバーをサービスグループに関連付けます。[自動スケール] オプションは [DNS] に設定されています。サービスグループの追加については、サービスグループの構成を参照してください。

ドメインベースのサーバーがサービスグループにバインドされ、自動スケーリングオプションがバインディングに設定されている場合、UDP モニターと TCP モニターが自動的に作成され、ドメインベースのサーバーにバインドされます。2 台のモニタはリゾルバとして機能します。TCP モニターはデフォルトで無効になっており、アプライアンスは UDP モニターを使用して DNS クエリをネームサーバーに送信し、ドメイン名を解決します。DNS 応答が切り捨てられた（TC フラグが 1 に設定されている）場合、アプライアンスは TCP にフォールバックし、TCP モニタを使用して TCP 経由で DNS クエリを送信します。その後、アプライアンスは引き続き TCP モニターのみを使用します。

ネームサーバからの DNS 応答には、ドメイン名に対して複数の IP アドレスが含まれる場合があります。自動スケーリングオプションが設定されている場合、アプライアンスはデフォルトのモニターを使用して各 IP アドレスをポーリングし、稼働していて使用可能な IP アドレスのみをサービスグループに含めます。Time To Live（TTL; 存続可能時間）値で定義された IP アドレスレコードの有効期限が切れると、UDP モニタ（または、アプライアンスが TCP モニタを使用するようにフォールバックした場合は TCP モニタ）はネームサーバにドメイン解決を照会し、新しい IP アドレスをサービスグループに含めます。サービスグループの一部である IP アドレスが DNS 応答に存在しない場合、アプライアンスはグループメンバーへの既存の接続を正常に閉じた後、そのアドレスをサービスグループから削除します。このプロセスでは、メンバーとの新しい接続を確立できません。過去に正常に解決されたドメイン名が NXDOMAIN 応答になった場合、そのドメインに関連付けられているすべてのサービスグループメンバーが削除されます。

スタティック（IP アドレスベース）メンバとダイナミックスケーリングドメインベースメンバは、1 つのサービスグループに共存できます。また、Automatic Scaling オプションを設定して、異なるドメイン名を持つメンバーを 1 つのサービスグループにバインドすることもできます。ただし、サービスグループに関連付けられている各ドメイン名は、サービスグループ内で一意である必要があります。サービスグループの自動スケーリングに使用するドメインベ

ースのサーバごとに、[Automatic Scaling] オプションを有効にする必要があります。IP アドレスが 1 つ以上のドメインに共通する場合、その IP アドレスはサービスグループに 1 回だけ追加されます。

> 重要
>
> - DNS Autoscale はクラスタ展開でサポートされています。
> - Autoscale サービスグループのパス監視は、クラスタ展開ではサポートされていません。

コマンドラインインターフェイスを使用してサービスグループを自動的にスケーリングするように構成するには

コマンドプロンプトで次のコマンドを入力して、サービスグループを構成し、構成を確認します。

```
add serviceGroup <serviceGroupName> -autoScale (YES | NO)

show serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

例

次の例では、server1 はドメインベースのサーバーです。DNS 応答には複数の IP アドレスが含まれています。5 つのアドレスが使用可能で、サービスグループに追加されます。

```
> add serviceGroup servGroup -autoScale YES
 Done
> sh servicegroup servGroup
       servGroup - HTTP
        State: ENABLED  Monitor Threshold : 0
              . . .
              . . .
        1)   192.0.2.31:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        2)   192.0.2.32:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
```



```
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        3)   192.0.2.36:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        4)   192.0.2.55:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        5)   192.0.2.80:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用してサービスグループを自動的にスケーリングするように構成するには

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. サービスグループを作成し、Autoscale モードを DNS に設定します。

TTL 値の上書き

> 注:
>
> このオプションは、Citrix ADC 12.1 ビルド 51.xx 以降でサポートされています。

Citrix ADC アプライアンスは、アプリケーションの起動時に、アプリケーションに関連付けられた SRV レコードの更新について DNS サーバーに定期的にクエリを実行するように構成されています。デフォルトでは、このクエリの周期性は SRV レコードにパブリッシュされた TTL によって異なります。マイクロサービスまたはクラウドワールドアプリケーションでは、デプロイメントはより動的に変化します。そのため、プロキシはアプリケーションのデプロイメントに対する変更をより迅速に吸収する必要があります。したがって、ドメインベースのサービス TTL パラメータは、SRV レコード TTL よりも小さく、展開に最適な値に明示的に設定することをお勧めします。TTL 値は、次の 2 つの方法で上書きできます。

- メンバーをサービスグループにバインドしているとき
- set lb パラメータコマンドを使用して、TTL 値をグローバルに設定します。

TTL 値がサービスグループメンバーのバインド時とグローバルの両方で設定されている場合、サービスグループメンバーのバインド時に指定された TTL 値が優先されます。
サービスグループメンバーのバインド中またはグローバルレベルで TTL 値が指定されていない場合、DBS モニタ間隔は DNS 応答の TTL 値から導出されます。

CLI を使用した **TTL** 値の上書き

- バインド中に TTL 値を上書きするには、コマンドプロンプトで次のように入力します。

```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
    >])
<!--NeedCopy-->
```

例**:**

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```

- TTL 値をグローバルに上書きするには、コマンドプロンプトで次のように入力します。

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

例**:**

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```

GUI を使用した **TTL** 値の上書き

バインド中に **TTL** 値を上書きするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。

3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4.「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

TTL 値をグローバルレベルで上書きするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [負荷分散パラメータの変更] に移動します。
2. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

> 注**:**
>
> ドメインベースのサーバの TTL 値が 0 に設定されている場合は、データパケットの TTL 値が使用されます。

サービスグループとドメイン名のバインディングに異なるネームサーバを指定する

> 注:
>
> このオプションは、Citrix ADC 12.1 ビルド 51.xx 以降でサポートされています。

特定のグループ内の異なるドメイン名に対して、異なるネームサーバを設定できます。DBS サーバをサービスグループにバインドする場合、NameServer パラメータの設定は任意です。メンバーをサービスグループにバインドする際にネームサーバーが指定されない場合、グローバルに設定されたネームサーバーが考慮されます。

CLI を使用してサーバをサービスグループにバインドする際にネームサーバを指定する

コマンドプロンプトで入力します。

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
    ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
    -dbsTTL 10
<!--NeedCopy-->
```

GUI を使用してサーバをサービスグループにバインドする際にネームサーバを指定する

1. [トラフィック管理] > [負荷分散] > [サービスグループ] に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。

3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4.「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ネームサーバー] で、バインドされたドメインのクエリの送信先となるネームサーバー名を指定します。

DNS SRV レコードを使用したサービス検出

October 7, 2021

SRV レコード（サービスレコード）は、ドメインネームシステム内のデータの仕様で、場所、つまり指定されたサービスのサーバのホスト名およびポート番号を定義します。レコードは、各サーバーの重みと優先順位も定義します。

SRV レコードの例**:**

_http._tcp.example.com. 100 IN SRV 10 60 5060 a.example.com.

次の表に、SRV レコードの各項目を示します。

Service	Protocol	Name	TTL	Class	SRV	Priority	Weight	Port	Target
HTTP	TCP	example.com	100	IN	SRV	10	60	5060	a.example.com

DNS SRV レコードを使用して、サービスエンドポイントを検出できます。Citrix ADC アプライアンスは、サービスに関連付けられた SRV レコードを使用して DNS サーバーに定期的にクエリを実行するように構成されています。SRV レコードを受信すると、SRV レコードで公開された各ターゲットホストは、サービスに関連付けられたサービスグループにバインドされます。各バインディングは、SRV レコードからポート、プライオリティ、およびウェイトを継承します。サービス展開ごとに、Citrix ADC アプライアンスを起動中に一度設定する必要があるため、アプリケーションをワンタッチで展開できます。

重要：動的に学習されたサービスグループメンバーの重みは、CLI または GUI を使用して変更できません。

使用例**:** 負荷分散マイクロサービス

アプリケーションは、モノリシックアーキテクチャからマイクロサービスアーキテクチャに移行しています。マイクロサービス・アーキテクチャとバックエンド・サーバの Autoscale ソリューションへの移行により、アプリケーションの展開がよりダイナミックになります。このような動的な展開をサポートするには、プロキシまたは ADC がバックエンドアプリケーションまたはサービスインスタンスを動的に検出し、それらをプロキシ構成に吸収できる必要があります。
DNS SRV レコード機能を使用したサービス検出は、このような動的な展開シナリオでの Citrix ADC アプライアンスの構成を支援します。アプリケーション開発者は、オーケストレーションプラットフォームの一部を使用してアプリケーションをデプロイできます。オーケストレーションプラットフォームは、アプリケーションのデプロイ時にコンテナをインスタンス化するときに、これらのコンテナごとにプロトコル固有の標準ポートを割り当てない場合があります。このようなシナリオでは、ポート情報を検出することが Citrix ADC アプライアンスを構成するための鍵とな

ります。このようなシナリオでは、SRV レコードが役立ちます。優先順位や重みなどの SRV レコードパラメータを使用すると、アプリケーションの負荷分散を改善できます。

- Priority パラメータは、サーバプールのプライオリティを指定するために使用できます。
- Weight パラメータは、バックエンドサービスインスタンスの容量を決定するために使用できるため、加重負荷分散に使用できます。
- バックエンドサーバープールに変更がある場合（たとえば、バックエンドインスタンスがプールから削除されるなど）、インスタンスは既存のすべてのクライアント接続が受け入れられた後にのみ優雅に削除されます。

> 注:
>
> - A/AAAA レコードベースのサービスディスカバリでは、解決されるドメインに重みを割り当てるため、解決されるすべての IP アドレスは同じ重みを持ちます。
> - SRV 応答の重みが 100 より大きい場合、サービスは作成されません。

SRV レコードを使用した優先度ベースのロード・バランシング

SRV レコードを使用して、優先順位ベースの負荷分散を実行できます。優先度ベースのサーバプールは、バックアップ仮想サーバの代替として使用できます。ns.conf ファイルの構成は、バックアップ仮想サーバと比較して最小限です。

SRV レコードを使用した優先順位ベースの負荷分散では、各サーバプールに優先順位番号が割り当てられます。最小の数値が最も高い優先度を持ちます。優先順位が最も高いプール内のサーバの 1 つが、サーバの正常性と可用性に基づいて負荷分散対象として選択されます。優先順位が最も高いサーバプール内のすべてのサーバがダウンしている場合、次に高い優先順位を持つサーバが負荷分散対象として選択されます。ただし、最も優先順位の高いサーバプール内のサーバが再び起動している場合は、優先順位の高いプールからサーバが再び選択されます。

ある優先サーバープールから別のサーバープールへの切り替えは、既存のクライアントトランザクションをブリーディングすることによって優雅に行われます。したがって、現在のクライアントでは、アプリケーションアクセスにブレークは表示されません。

CLI を使用して SRV レコードのクエリを有効にするには

SRV レコードのクエリを有効にするには、次のタスクを実行します。

1. クエリタイプパラメータを SRV として指定して、サーバーを作成します。

   コマンドプロンプトで入力します。

```
add server <name> <domain> [-queryType <queryType>])
<!--NeedCopy-->
```

   例:


```
add server web_serv example.com -queryType SRV
<!--NeedCopy-->
```

注:

- デフォルトでは、IPv4 クエリーが送信されます。IPv6 クエリーを送信するには、IPv6 ドメインを有効にする必要があります。
- SRV ターゲットドメイン名は 127 文字以下にする必要があります。

2. Autoscale モードを DNS として持つサービスグループを作成します。

   コマンドプロンプトで入力します。

```
add serviceGroup <serviceGroupName> <serviceType> [-autoScale <
   autoScale>]
<!--NeedCopy-->
```

例:

```
add servicegroup svc_grp_1 http -autoscale dns
<!--NeedCopy-->
```

3. 手順 1 で作成したサーバをメンバーとしてサービスグループにバインドします。

   コマンドプロンプトで入力します。

```
bind serviceGroup <serviceGroupName> <serverName>
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv
<!--NeedCopy-->
```

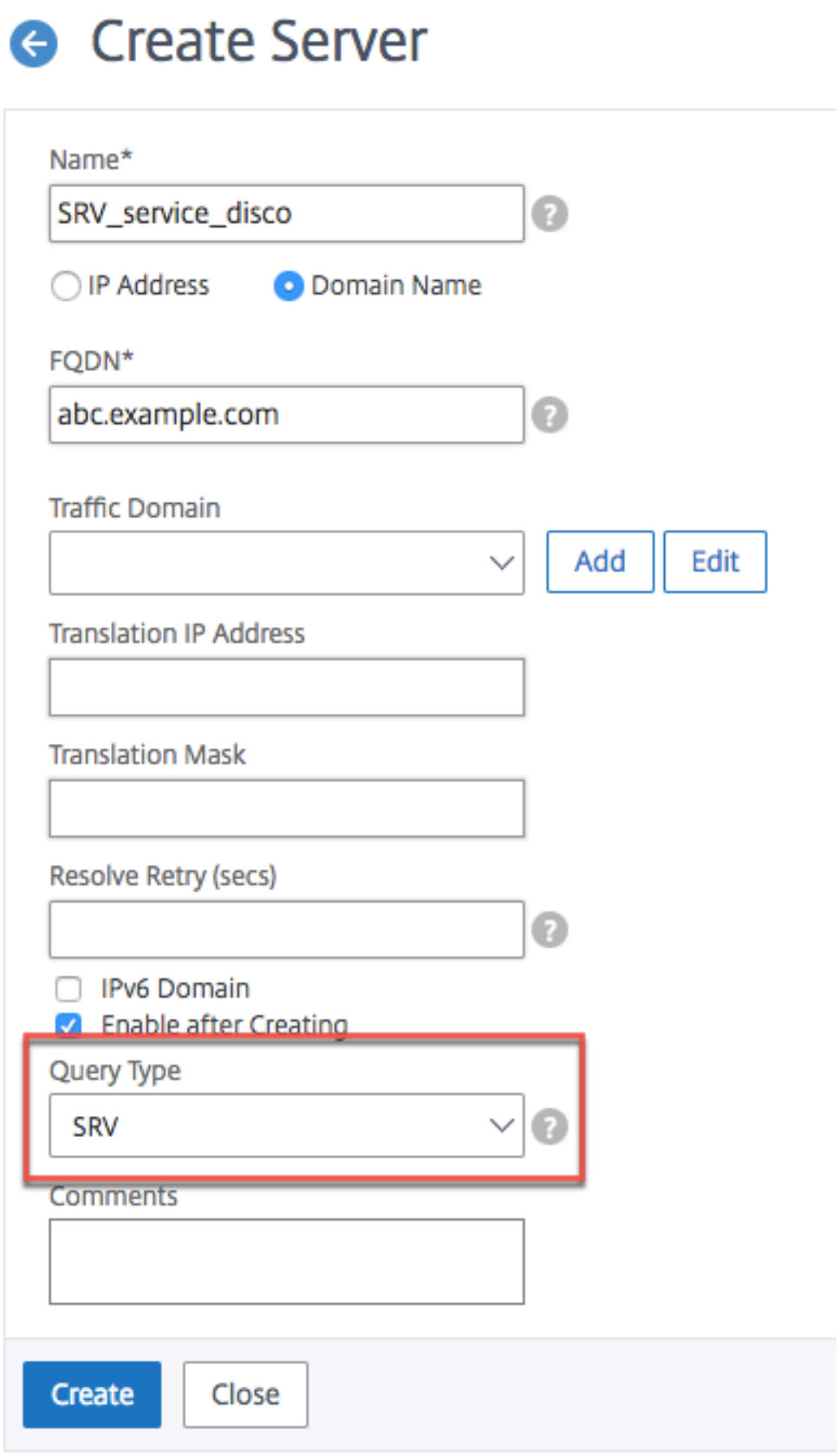
> 注:
>
> - サーバをサービスグループメンバーにバインドする場合、SRV サーバタイプのポート番号を入力する必

> 要はありません。SRV サーバタイプにポート番号を指定すると、エラーメッセージが表示されます。
>
> - サーバとサービスグループにバインドするときに、オプションでネームサーバと TTL 値を指定できます。

GUI を使用して **SRV** レコードのクエリーを有効にするには

サーバーを作成する

1. [トラフィック管理] > [負荷分散] > [サーバー] に移動し、[追加] をクリックします。



2.「サーバーの作成」ページで、ドメイン名を選択します。

3. 必要なすべてのパラメータの詳細を入力します。

4.「クエリー・タイプ」で「**SRV**」を選択します。

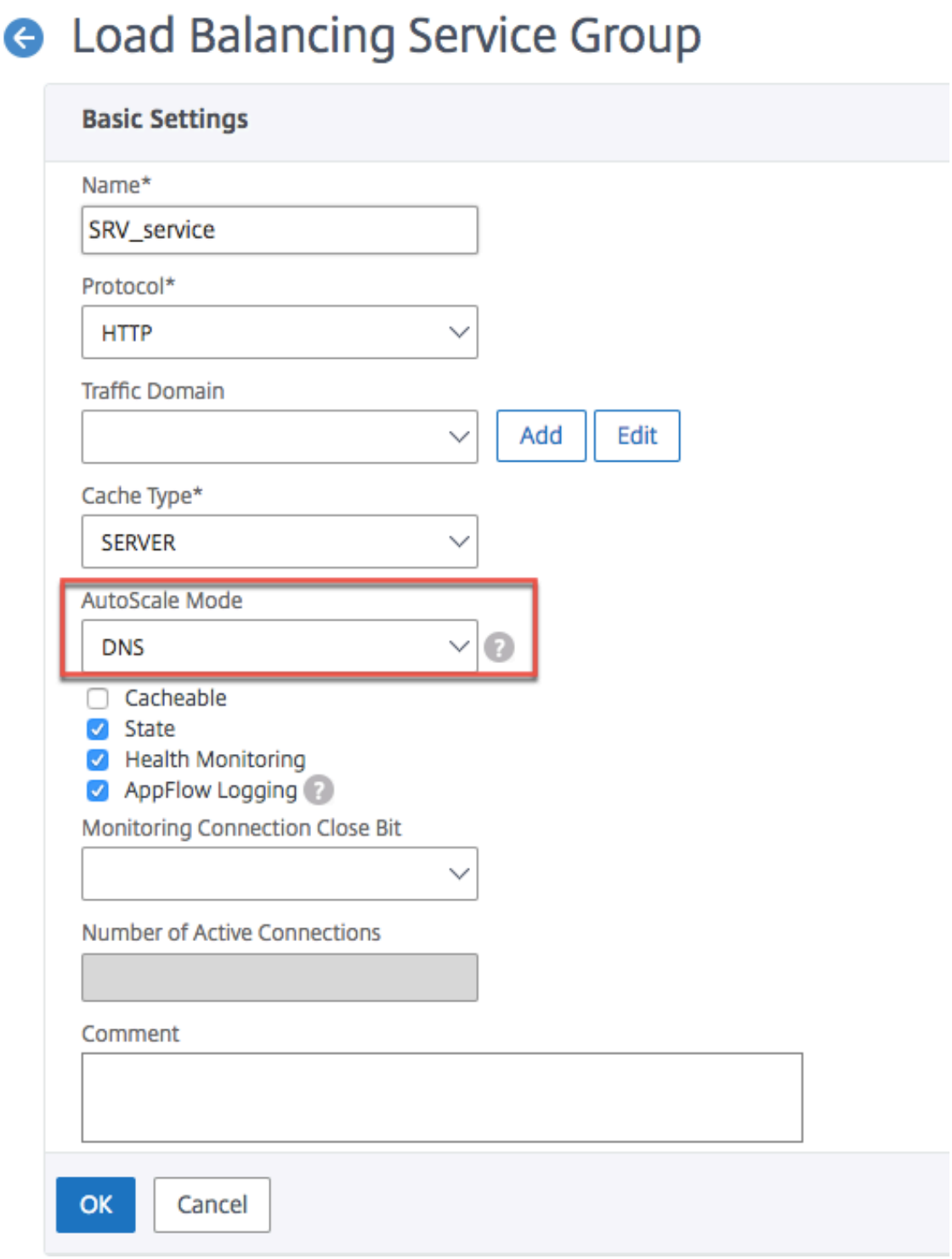
5.［作成］をクリックします。

Autoscale モードを **DNS** として持つサービスグループを作成する

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。

2.「負荷分散サービスグループ」ページで、必要なすべてのパラメータの詳細を入力します。

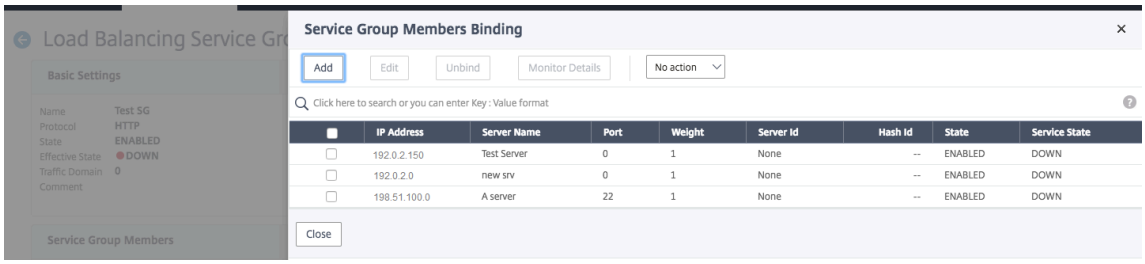
3.［**AutoScale Mode]** で、［**DNS**］を選択します。



4. [**OK**] をクリックします。

サーバをサービスグループメンバーにバインドする

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4. 「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「閉じる」をクリックします。



注:

- バインド中は、SRV サーバタイプのポート番号を入力する必要はありません。SRV サーバタイプのポート番号を入力すると、エラーメッセージが表示されます。
- サーバとサービスグループにバインドするときに、オプションでネームサーバと TTL 値を指定できます。

TTL 値の上書き

Citrix ADC アプライアンスは、アプリケーションの起動時に、アプリケーションに関連付けられた SRV レコードの更新について DNS サーバーに定期的にクエリを実行するように構成されています。デフォルトでは、このクエリの周期性は SRV レコードで発行された TTL に依存します。マイクロサービスまたはクラウド世界のアプリケーションでは、デプロイメントがより動的に変化します。その結果、プロキシは、アプリケーションの展開に対する変更を迅速に吸収する必要があります。したがって、ユーザーは、ドメインベースのサービス TTL パラメータを、SRV レコード TTL よりも低く、展開に最適な値に明示的に設定することをお勧めします。TTL 値は、次の 2 つの方法で上書きできます。

- メンバーをサービスグループにバインドするとき
- set lb パラメータコマンドを使用して、TTL 値をグローバルに設定します。

TTL 値がサービスグループメンバーのバインド時とグローバルの両方で設定されている場合、サービスグループメンバーのバインド時に指定された TTL 値が優先されます。
サービスグループメンバーのバインド中またはグローバルレベルで TTL 値が指定されていない場合、DBS モニタ間隔は DNS 応答の TTL 値から導出されます。

CLI を使用した TTL 値の上書き

- バインド中に TTL 値を上書きするには、コマンドプロンプトで次のように入力します。


```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
    >])
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```

- TTL 値をグローバルに上書きするには、コマンドプロンプトで次のように入力します。

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

例:

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```

GUI を使用した **TTL** 値の上書き

バインド中に **TTL** 値を上書きするには、次の手順を実行します。

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4. 「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

グローバルレベルで **TTL** 値を上書きするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [負荷分散パラメータの変更] に移動します。
2. [ドメインベースサービス **TTL**] に、TTL 値を入力します。

> 注: ドメインベースのサーバーの TTL 値が 0 に設定されている場合、データパケットの TTL 値が使用されます。

サービスグループおよびドメイン名のバインディングに異なるネームサーバを指定する

特定のグループ内の異なるドメイン名に対して異なるネームサーバを設定できます。DBS サーバをサービスグループにバインドする場合は、nameServer パラメータの設定はオプションです。メンバーをサービスグループにバインドするときにネームサーバーが指定されていない場合、グローバルに設定されたネームサーバーが考慮されます。

CLI を使用したサービスグループへのサーバのバインド時のネームサーバの指定

コマンドプロンプトで入力します。

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
    ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

例:

```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
    -dbsTTL 10
<!--NeedCopy-->
```

GUI を使用したサービスグループへのサーバのバインド時のネームサーバの指定

1. **Traffic Management** > **Load Balancing** > **Service Groups** に移動します。
2. [**Service Groups**] ページで、作成したサービスグループを選択し、[**Edit**] をクリックします。
3. [負荷分散サービスグループ] ページで、[サービスグループメンバー] をクリックします。
4. 「サービスグループメンバーのバインド」ページで、作成したサーバーを選択し、「編集」をクリックします。
5. [ネームサーバー] で、バインドされたドメインのクエリの送信先となるネームサーバー名を指定します。

ドメインベースのサーバーの **IP** アドレスを変換する

October 7, 2021

Citrix ADC アプライアンスおよびそれに接続されているドメインベースのサーバーでのメンテナンスを簡素化するために、IP アドレスマスクと変換 IP アドレスを構成できます。これらの関数は連携して着信 DNS パケットを解析し、DNS で解決された IP アドレスを新しい IP アドレスに置き換えます。

ドメインベースのサーバ用に設定されている場合、IP アドレス変換により、メンテナンスのためにサーバを停止したり、サーバに影響するその他のインフラストラクチャの変更を行った場合に、アプライアンスは代替サーバの IP アドレスを特定できます。

マスクを設定するときは、標準の IP マスク値（2 の累乗、1 を引いた値）とゼロ（255.255.0.0 など）を使用する必要があります。0 以外の値は、開始オクテットでのみ使用できます。

サーバの変換 IP を設定する場合、サーバ IP アドレスと、その IP アドレスの先頭または末尾のオクテットを共有する代替サーバとの間に 1:1 の対応を作成します。マスクは、元のサーバーの IP アドレスの特定のオクテットをブロックします。DNS で解決された IP アドレスは、変換 IP アドレスと変換マスクを適用することによって、新しい IP アドレスに変換されます。

たとえば、変換 IP アドレスを 10.20.0.0 に設定し、変換マスクを 255.255.0.0 に設定できます。サーバの DNS 解決済み IP アドレスが 40.50.27.3 の場合、このアドレスは 10.20.27.3 に変換されます。この場合、変換 IP アドレスは新しいアドレスの最初の 2 つのオクテットを提供し、マスクは元の IP アドレスから最後の 2 つのオクテットを通過します。DNS によって解決された元の IP アドレスへの参照は失われます。サーバがバインドされているすべてのサービスのモニタでも、変換された IP アドレスがレポートされます。

ドメインベースサーバの変換 IP アドレスを設定する場合は、DNS で解決された IP アドレスの変換先となるマスクと IP アドレスを指定します。

注:IP アドレスの変換は、ドメインベースのサーバーでのみ可能です。この機能は、IP ベースのサーバーには使用できません。アドレスパターンは、IPv4 アドレスのみに基づくことができます。

コマンドラインインターフェイスを使用してサーバーの変換 **IP** アドレスを構成するには

コマンドプロンプトで入力します。

```
add server <name>@ <serverDomainName> -translationIp <
    translationIPAddress> -translationMask <netMask> -state <ENABLED|
    DISABLED>
<!--NeedCopy-->
```

例:

```
add server myMaskedServer www.example.com -translationIp 10.10.10.10 -
    translationMask
255.255.0.0 -state ENABLED
<!--NeedCopy-->
```


構成ユーティリティを使用してサーバの変換 **IP** アドレスを構成するには

[トラフィック管理] **>** [負荷分散] **>** [サーバー] に移動し、ドメインベースのサーバーを作成し、変換 IP アドレスを指定します。

仮想サーバの **IP** アドレスをマスクする

October 7, 2021

仮想サーバの固定 IP アドレスの代わりにマスクとパターンを設定できます。これにより、マスクとパターンに一致する任意の IP アドレス宛てのトラフィックが、特定の仮想サーバーに再ルーティングされます。たとえば、IP アドレスの最初の 3 つのオクテットを可変にできるマスクを設定すると、111.11.198、22.22.198、および 33.33.198 へのトラフィックがすべて同じ仮想サーバに送信されます。

仮想サーバの IP アドレスのマスクを設定することで、ルーティングの変更や他のインフラストラクチャの変更による仮想サーバの再構成を回避できます。このマスクにより、仮想サーバを大規模に再構成することなく、トラフィックのフローを継続できます。

仮想サーバの IP アドレスのマスクは、ドメインベースサーバの IP アドレスの変換で説明されているサーバの IPパターン定義とは異なります。仮想サーバの IP アドレスマスクの場合、ゼロ以外のマスクは考慮されるオクテットとして解釈されます。サービスの場合、ゼロ以外の値はブロックされます。

また、仮想サーバーの IP アドレスマスクの場合は、先頭または末尾の値を考慮できます。仮想サーバの IP アドレスマスクが IP アドレスの左側の値を考慮する場合、これをフォワードマスクと呼びます。マスクがアドレスの右側にある値を考慮する場合、これは逆マスクと呼ばれます。

注：Citrix ADC アプライアンスは、リバースマスク仮想サーバーを評価する前に、すべてのフォワードマスク仮想サーバーを評価します。

仮想サーバの IP アドレスをマスキングする場合は、着信トラフィックを正しい仮想サーバと照合するための IP アドレスパターンも作成する必要があります。アプライアンスは、着信 IP パケットを受信すると、パケット内の宛先 IP アドレスと IP アドレスパターンで考慮されるビットが照合され、一致が見つかると IP アドレスマスクが適用され、最終的な宛先 IP アドレスが作成されます。

次の例を考えてみましょう。

- 着信パケットの宛先 IP アドレス：10.102.27.189
- IP アドレスパターン:10.102.0.0
- IP マスク：255.255.0.0
- 作成された（最終）宛先 IP アドレス：10.102.27.189。

この場合、元の宛先 IP アドレスの最初の 16 ビットがこの仮想サーバーの IP アドレスパターンと一致するため、この着信パケットはこの仮想サーバーにルーティングされます。

宛先 IP アドレスが複数の仮想サーバの IP パターンと一致する場合は、最も長い一致が優先されます。次の例を考えてみましょう。

- 仮想サーバ 1: IP パターン 10.10.0.0、IP マスク 255.255.0.0
- 仮想サーバ 2: IP パターン 10.10.10.0、IP マスク 255.255.255.0
- パケットの宛先 IP アドレス: 10.10.10.45。
- 選択した仮想サーバー: 仮想サーバー 2。

仮想サーバ 2 に関連付けられたパターンは、仮想サーバ 1 に関連付けられたビットよりも多くのビットと一致するため、一致する IP は仮想サーバ 2 に送信されます。

注: タイブレーカーが必要な場合は、ポートも考慮されます。

コマンドラインインターフェイスを使用して仮想サーバーの **IP** アドレスマスクを構成するには

コマンドプロンプトで入力します。

```
add lb vserver <name>@ http -ipPattern <ipAddressPattern> -ipMask <
   ipMask> <listenPort>
<!--NeedCopy-->
```

例:

プレフィクスオクテットに基づくパターンマッチング:

```
add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask
   255.255.0.0 80
<!--NeedCopy-->
```

末尾のオクテットに基づくパターンマッチング:

```
add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask
   0.0.255.255 80
<!--NeedCopy-->
```

パターンベースの仮想サーバーを変更します。

```
set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255
<!--NeedCopy-->
```


構成ユーティリティを使用して仮想サーバの **IP** アドレスマスクを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [アドレスの種類] リストで [IP パターン] を選択し、IP パターンと IP マスクを指定します。

一般的に使用されるプロトコルの負荷分散を構成する

October 7, 2021

ウェブサイトや Web ベースのアプリケーションに加えて、他の一般的なプロトコルを使用する他のタイプのネットワーク展開アプリケーションは、多くの場合、大量のトラフィックを受信するため、ロードバランシングの恩恵を受けます。これらのプロトコルのいくつかは、ロードバランシングを適切に動作させるために特定の設定を必要とします。その中には、FTP、DNS、SIP、および RTSP があります。

Citrix ADC アプライアンスを IP ではなくサーバーのドメイン名を使用するように構成する場合は、それらのサーバーの IP 変換とマスキングも設定する必要があります。

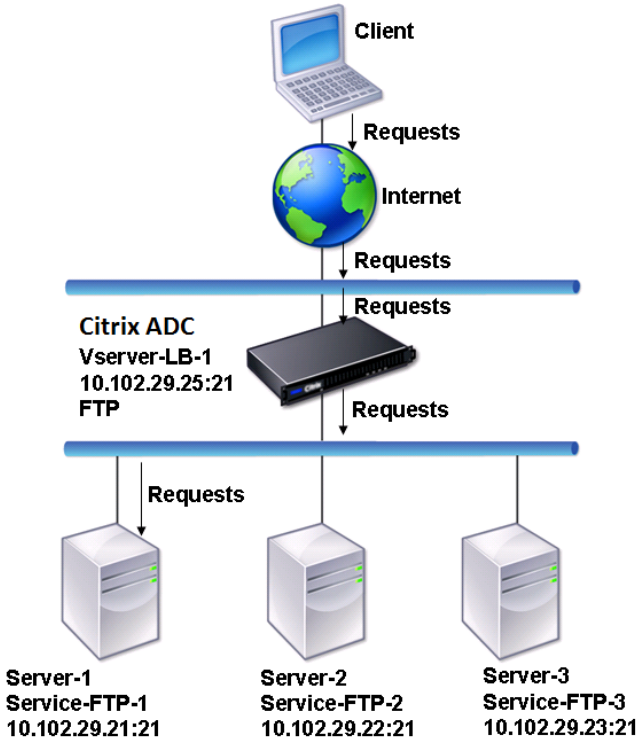
FTP サーバーのグループの負荷分散

October 7, 2021

Citrix ADC アプライアンスは、FTP サーバーの負荷分散に使用できます。FTP では、ユーザーが同じサーバーに対する 2 つの異なるポートで 2 つの接続を開始する必要があります。制御接続とは、クライアントがコマンドをサーバーに送信する制御接続と、サーバーがクライアントにデータを送信するデータ接続です。クライアントが FTP サーバーへの制御接続を開いて FTP セッションを開始すると、アプライアンスは設定された負荷分散方式を使用して FTP サービスを選択し、制御接続をそのサービスに転送します。負荷分散された FTP サーバーは、情報交換のためにクライアントへのデータ接続を開きます。

次の図は、FTP サーバーのグループの負荷分散構成のトポロジーを示しています。

図 1: FTP サーバの基本的なロードバランシングトポロジ



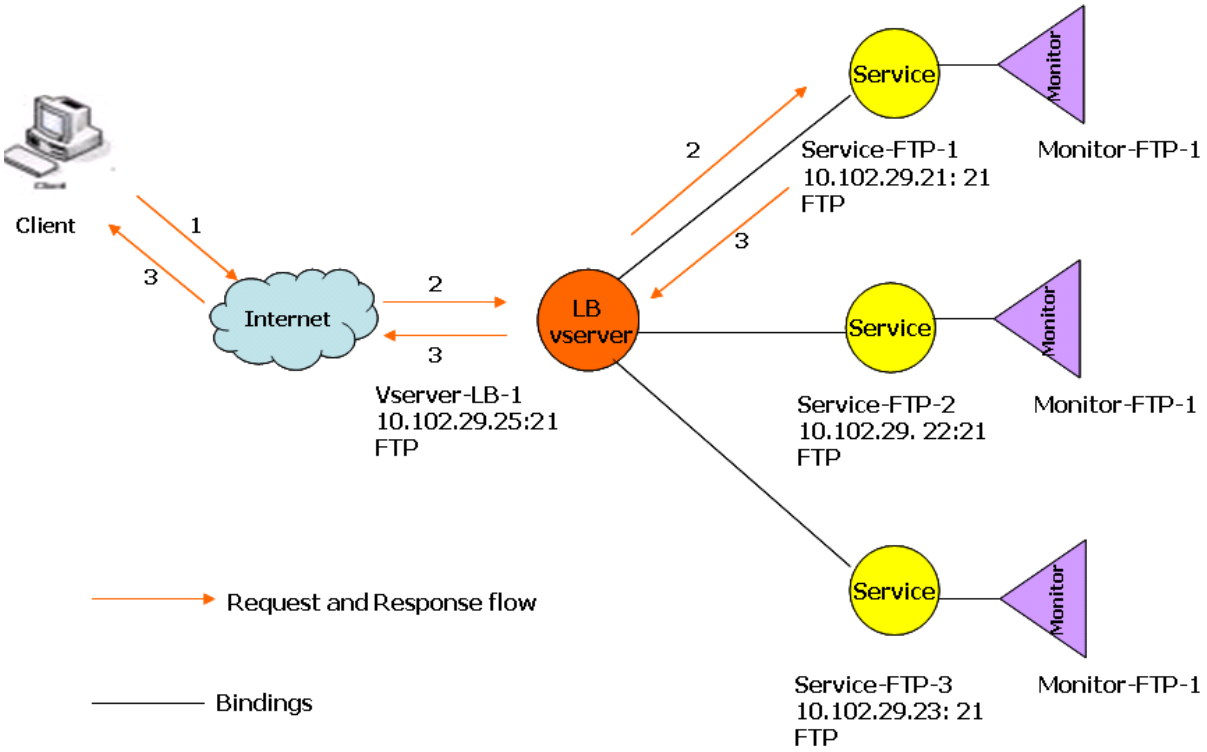
この図では、Service-FTP-1、Service-FTP-2、および Service-FTP-3 が仮想サーバ Vserver-LB-1 にバインドされています。Vserver-LB-1 は、最小接続負荷分散方式を使用して、クライアントの接続要求をサービスの 1 つに転送します。それ以降の要求は、アプライアンスが最初にロードバランシング用に選択したサービスに転送されます。

次の表に、アプライアンスで設定されている基本エンティティの名前と値を示します。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
Vserver	Vserver-LB-1	10.102.29.25	21	FTP
サービス	Service-FTP-1	10.102.29.21	21	FTP
	Service-FTP-2	10.102.29.22	21	FTP
	Service-FTP-3	10.102.29.23	21	FTP
モニター	FTP	なし	なし	なし

次の図は、負荷分散エンティティと、アプライアンス上で設定する必要があるパラメータの値を示しています。

図 2: 負荷分散 FTP サーバーエンティティモデル



アプライアンスは、ファイアウォールの外側から FTP サーバーにアクセスするためのパッシブ FTP オプションを提供することもできます。クライアントがパッシブ FTP オプションを使用し、FTP サーバへの制御接続を開始すると、FTP サーバはクライアントへの制御接続も開始します。その後、ファイアウォール経由でファイルを転送するためのデータ接続を開始します。

FTP タイプのサービスおよび仮想サーバーを作成するには、基本的な負荷分散の設定を参照してください。エンティティに名前を付け、パラメータを前の表の列で説明した値に設定します。基本的な負荷分散セットアップを構成すると、デフォルトのモニターがサービスにバインドされます。

次に、「モニタをサービスへのバインド」の項で説明されている手順に従って、FTP モニタをサービスにバインドします。

CLI を使用して FTP モニタを作成するには

コマンドプロンプトで入力します。

```
1 add lb monitor <MonitorName> FTP -interval <Interval> -userName <
     UserName> -password <Password>
2 <!--NeedCopy-->
```

例:


```
add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password
    User
<!--NeedCopy-->
```

GUI を使用して **FTP** モニタを作成するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. FTP タイプのモニターを作成し、「特殊パラメータ」でユーザー名とパスワードを指定します。

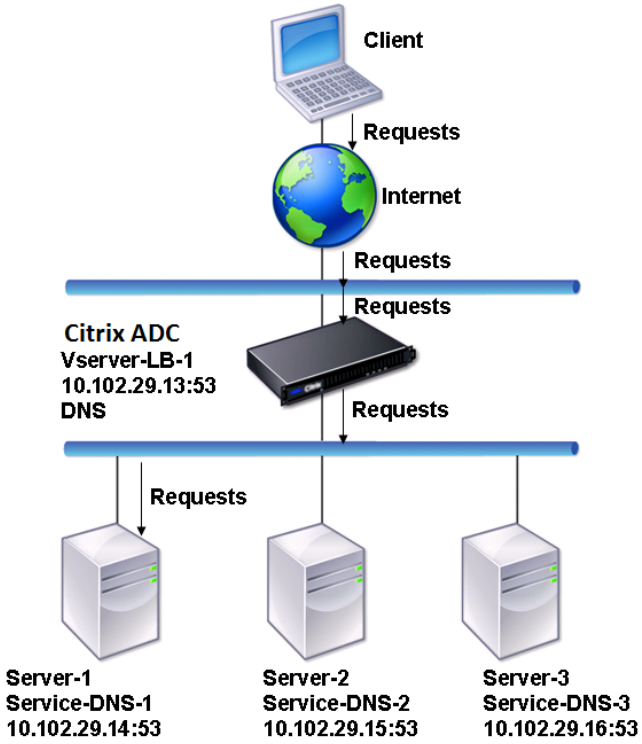
DNS サーバーの負荷分散

October 7, 2021

ドメイン名の DNS 解決を要求すると、Citrix ADC アプライアンスは構成された負荷分散方式を使用して DNS サービスを選択します。サービスがバインドされている DNS サーバーは、ドメイン名を解決し、応答として IP アドレスを返します。アプライアンスは DNS 応答をキャッシュし、キャッシュされた情報を使用して、同じドメイン名の解決を求める将来の要求に応答することもできます。DNS サーバーの負荷分散 DNS 応答時間が短縮されます。

次の図は、DNS サービスのグループを負荷分散する負荷分散構成のトポロジを示しています。

図 1: DNS サーバーの基本的な負荷分散トポロジ

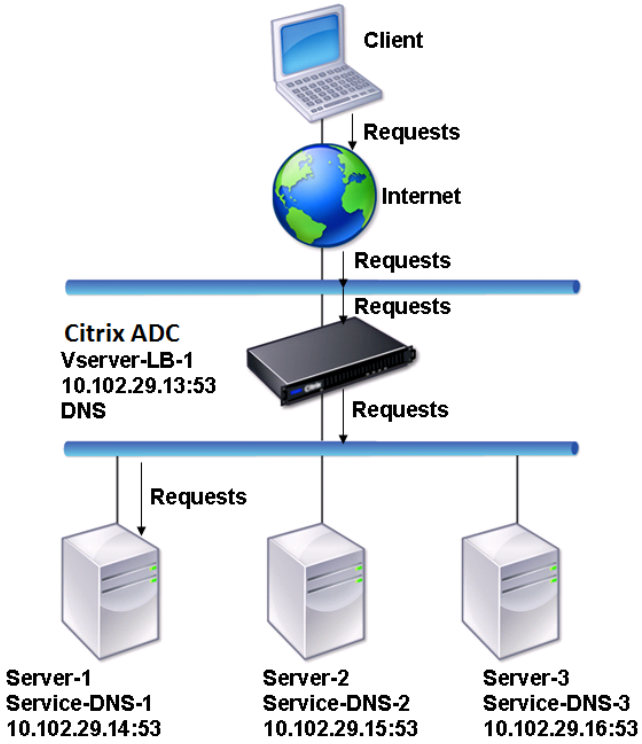


この図では、サービス DNS-1、サービス DNS-2、およびサービス DNS-3 が仮想サーバ Vserver-LB-1 にバインドされています。仮想サーバー Vserver-LB-1 は、最小接続ロードバランシング方式を使用して、クライアント要求をサービスに転送します。次の表に、アプライアンスで設定されている基本エンティティの名前と値を示します。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.13	53	DNS
サービス	Service-DNS-1	10.102.29.14	53	DNS
	Service-DNS-2	10.102.29.15	53	DNS
	Service-DNS-3	10.102.29.16	53	DNS
モニター	monitor-DNS-1	なし	なし	なし

次の図は、負荷分散エンティティと、アプライアンス上で設定する必要があるパラメータの値を示しています。

図 2：負荷分散 DNS サーバーエンティティモデル



基本的な DNS 負荷分散設定を構成するには、基本的な負荷分散の設定を参照してください。手順に従って、DNS タイプのサービスと仮想サーバーを作成し、エンティティに名前を付けて、前の表で説明した値を使用してパラメータを設定します。基本的な負荷分散設定を構成すると、デフォルトの ping モニターがサービスにバインドされます。DNS モニタを DNS サービスにバインドする手順については、「モニタをサービスへのバインド」を参照してください。

次の手順では、クエリに基づいてドメイン名を IP アドレスにマッピングするモニターを作成する手順について説明します。

CLI を使用して DNS モニタを設定するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> DNS -query <domainName> -queryType <
    Address|ZONE> -IPAddress <ipAddress>
<!--NeedCopy-->
```

例:


```
add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType
    Address -IPAddress 10.102.29.66

add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType
    Address -IPAddress
1000:0000:0000:0000:0005:0600:700a::888b-888d
<!--NeedCopy-->
```

GUI を使用して **DNS** モニタを構成するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. 種類が DNS のモニタを作成し、[特殊パラメータ] でクエリとクエリの種類を指定します。

ドメイン名ベースのサービスの負荷分散

October 7, 2021

負荷分散用のサービスを作成するときに、IP アドレスを指定できます。または、ドメイン名を使用してサーバーを作成することもできます。サーバー名（ドメイン名）を解決するには、IPv4 または IPv6 ネームサーバーを使用するか、権限のある DNS レコード（IPv4 の場合はレコード、IPv6 の場合は AAAA レコード）を Citrix ADC 構成に追加します。

IP アドレスではなくドメイン名を使用してサービスを構成し、ネームサーバーがドメイン名を新しい IP アドレスに解決する場合、サービスにバインドされたモニターは新しい IP アドレスに対してヘルスチェックを実行し、IP アドレスが正常であることが検出された場合にのみサービス IP アドレスを更新します。モニタは、サービスにバインドされたデフォルトのモニタにすることも、サポートされているその他のモニタをバインドすることもできます。監視パラメータで定義された一定の間隔でサービスをプローブします。ドメイン名が新しい IP アドレスに解決されると、モニタは新しいプローブを送信してサービスの状態をチェックします。後続のすべてのプローブは、事前に定義された間隔で行われます。

注: サーバーの IP アドレスを変更すると、対応するサービスが最初のクライアント要求に対してマークダウンされます。ネームサーバは、次の要求のためにサービス IP アドレスを変更された IP アドレスに解決し、サービスが UP とマークされます。

ドメイン名ベースのサービスには、次の制限があります。

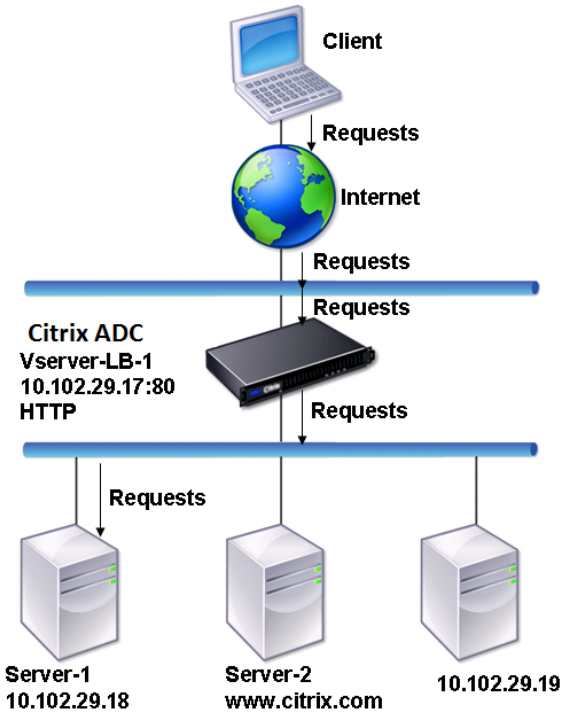
- ドメイン名の最大長は 255 文字です。
- Maximum Client パラメーターは、ドメイン名ベースのサーバーを表すサービスを構成するために使用されます。たとえば、仮想サーバにバインドされたサービスに 1000 の maxClient が設定されているとします。仮想サーバの接続数が 2000 に達すると、DNS リゾルバはサービスの IP アドレスを変更します。ただし、サ

ービスの接続カウンターがリセットされないため、古い接続がすべて閉じられるまで、仮想サーバーは新しい接続を取ることができません。

- サービスの IP アドレスが変更されると、永続性を維持することは困難です。
- タイムアウトによりドメイン名の解決が失敗した場合、アプライアンスは古い情報（IP アドレス）を使用します。
- モニタリングによってサービスがダウンしていることが検出されると、アプライアンスはサービス（ドメイン名ベースのサーバを表す）で DNS 解決を実行し、新しい IP アドレスを取得します。
- 統計はサービスで収集され、IP アドレスが変更されてもリセットされません。
- DNS 解決で「name error」(3) のコードが返された場合、アプライアンスはサービスをマークダウンし、IP アドレスをゼロに変更します。

アプライアンスは、サービスの要求を受信すると、ターゲットサービスを選択します。このようにして、アプライアンスはサービスの負荷のバランスをとります。次の図は、ドメインネームベースサーバー (DBS) のグループをロードバランシングするロードバランシング構成のトポロジを示しています。

図 1: DBS サーバの基本的なロードバランシングトポロジ



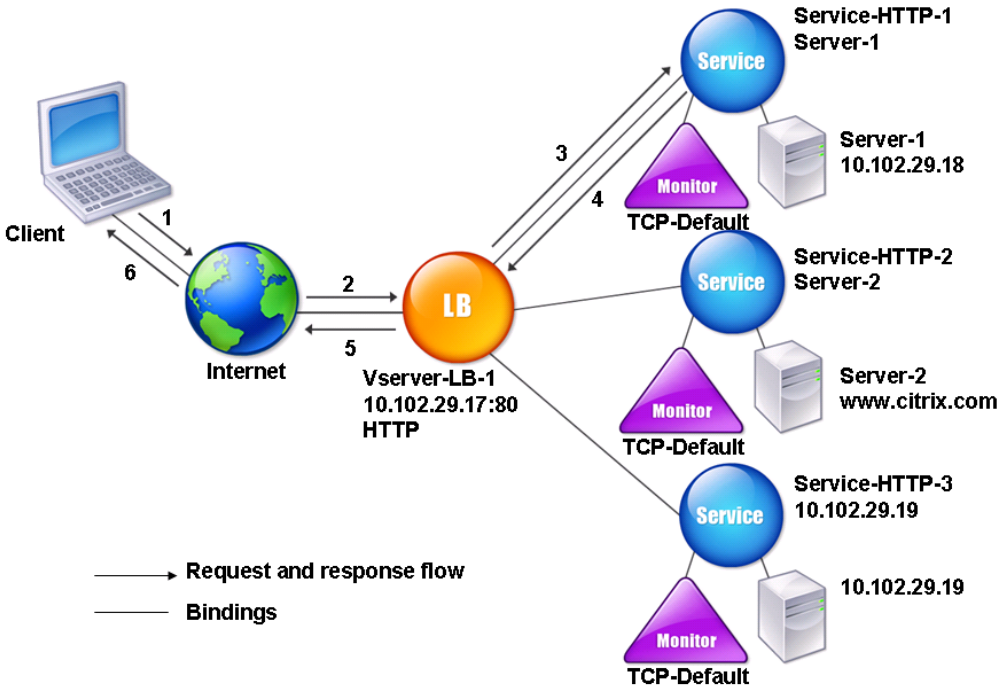
Service-HTTP-1、Service-HTTP-2、および Service-HTTP-3 は、仮想サーバ Vserver-LB-1 にバインドされます。仮想サーバー vserver-LB-1 は、サービスの選択に最小接続負荷分散方式を使用します。サービスの IP アドレスは、ネームサーバー Vserver-LB-2 を使用して解決されます。

次の表に、アプライアンスで設定されている基本エンティティの名前と値を示します。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.17	80	HTTP
	Vserver-LB-2	10.102.29.20	53	DNS
サーバー	server-1	10.102.29.18	80	HTTP
	server-2	www.citrix.co.jp	80	HTTP
サービス	Service-HTTP-1	server-1	80	HTTP
	Service-HTTP-2	server-2	80	HTTP
	Service-HTTP-2	10.102.29.19	80	HTTP
モニター	デフォルト	なし	なし	なし
ネーム・サーバ	なし	10.102.29.19	なし	なし

次の図は、負荷分散エンティティと、アプライアンス上で設定する必要があるパラメータの値を示しています。

図 2：負荷分散 DBS サーバーエンティティモデル



基本的な負荷分散設定を構成するには、基本負荷分散の設定を参照してください。HTTP タイプのサービスと仮想サ

ーバーを作成し、エンティティに名前を付け、前の表で説明した値を使用してパラメータを設定します。

外部ネームサーバーを追加、削除、有効化、および無効化することができます。IP アドレスを指定してネームサーバーを作成するか、既存の仮想サーバーをネームサーバーとして設定できます。

コマンドラインインターフェイスを使用してネームサーバーを追加するには

コマンドプロンプトで入力します。

```
add dns nameServer <dnsVserverName>
<!--NeedCopy-->
```

例**:**

```
add dns nameServer Vserver-LB-2
<!--NeedCopy-->
```

構成ユーティリティを使用してネームサーバーを追加するには

1. [****** トラフィック管理 **] > [**DNS**] > [****** ネームサーバ]** に移動します。
2. DNS 仮想サーバータイプの DNS ネームサーバーを作成し、[DNS 仮想サーバー] リストからサーバーを選択します。

ドメイン名を IP アドレスに解決する権限のあるネームサーバーを追加することもできます。

> 注
>
> TCP、UDP、または UDP_TCP タイプのネームサーバーをリゾルバ DBS プローブに追加できます。ただし、TCP と UDP ネームサーバーが共存し、UDP ネームサーバーが切り捨てられたビットの応答を受信した場合、この応答は TCP ネームサーバー上で再試行されません。

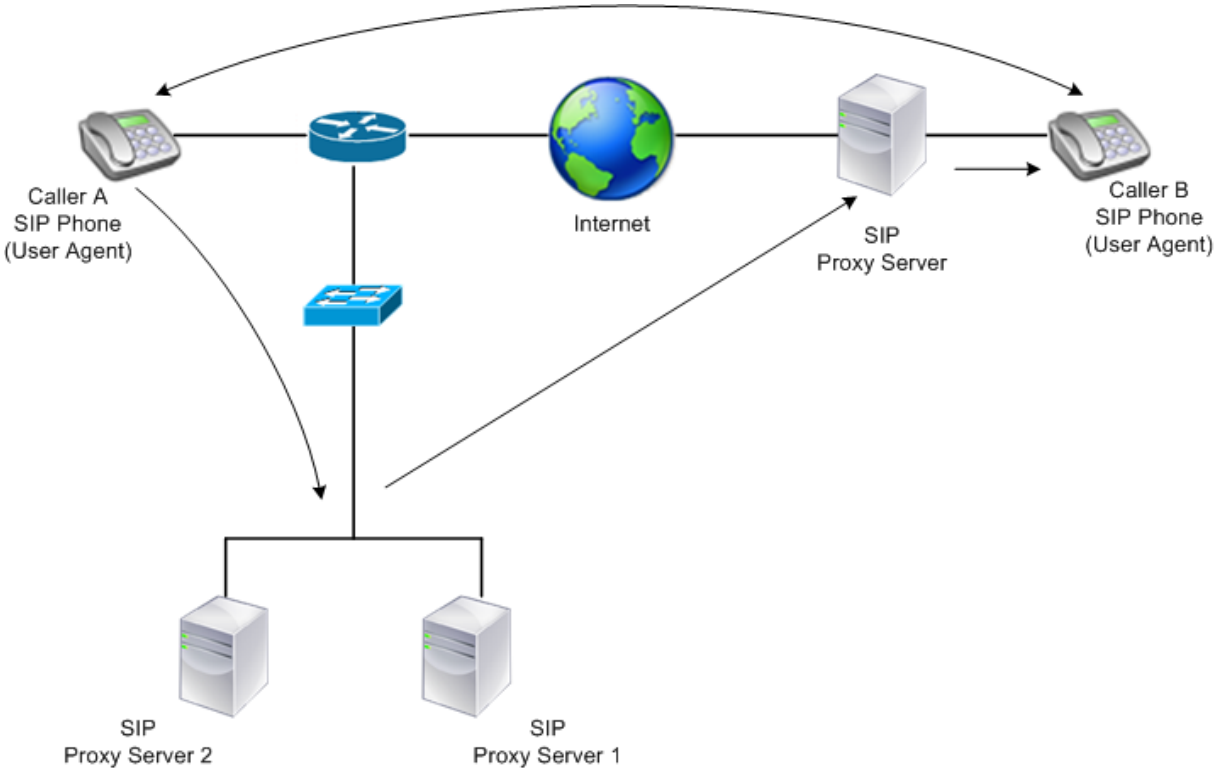
SIP サーバのグループの負荷分散

October 7, 2021

セッション開始プロトコル (SIP) は、マルチメディア通信セッションを開始、管理、および終了するように設計されています。これは、インターネット電話（VoIP）の標準として浮上しています。SIP メッセージは、TCP または UDP を介して送信できます。SIP メッセージには、要求メッセージと応答メッセージの 2 つのタイプがあります。

SIP ベースの通信システムのトラフィックは、専用のデバイスおよびアプリケーション（エンティティ）を介してルーティングされます。マルチメディア通信セッションでは、これらのエンティティはメッセージを交換します。次の図は、基本的な SIP ベースの通信システムを示しています。

図 1: SIP ベースの通信システム



Citrix ADC を使用すると、UDP または TCP（TLS を含む）経由で SIP メッセージを負荷分散できます。SIP プロキシサーバーのグループに対して SIP 要求を負荷分散するように Citrix ADC を構成できます。これを行うには、負荷分散方式と永続性のタイプを次のいずれかの組み合わせに設定して、負荷分散仮想サーバーを作成します。

- 持続性設定なしの Call-ID ハッシュロードバランシング方式
- 最小接続またはラウンドロビンロードバランシング方式による Call-ID ベースのパーシステンス
- 最小接続またはラウンドロビン負荷分散方式によるルールベースのパーシステンス

また、デフォルトでは、Citrix ADC は SIP 要求のヘッダーを介して RPORT を追加し、サーバーがリクエストの送信元 IP アドレスとポートに応答を返信します。

注: ロードバランシングが機能するには、プライベート IP アドレスまたはプライベートドメインが SIP ヘッダー/ペイロードに追加されないように SIP プロキシを設定する必要があります。SIP プロキシは、SIP 仮想サーバの IP アドレスに解決されるドメイン名を SIP ヘッダーに追加する必要があります。また、SIP プロキシは、登録情報を共有するために共通のデータベースと通信する必要があります。

サーバ開始トラフィック

SIP サーバーが開始するアウトバウンドトラフィックの場合は、クライアントが使用するプライベート IP アドレスがパブリック IP アドレスに変換されるように、Citrix ADC で RNAT を構成します。

RNAT 送信元ポートまたは宛先ポートを含む SIP パラメータを設定した場合、アプライアンスは要求パケットの送信元ポートと宛先ポートの値を、RNAT 送信元ポートおよび RNAT 宛先ポートと比較します。いずれかの値が一致すると、アプライアンスは VIA ヘッダーを RPORT で更新します。クライアントからの SIP 応答は、要求と同じパスを通過します。

サーバーが開始する SSL トラフィックの場合、Citrix ADC は組み込みの証明書とキーのペアを使用します。カスタム証明書とキーのペアを使用する場合は、カスタム証明書とキーのペアを **nsrnatsip-127.0.0.1-5061** という名前の Citrix ADC 内部サービスにバインドします。

ポリシーと式のサポート

Citrix ADC デフォルト式言語には、セッション開始プロトコル（SIP）接続で動作するいくつかの式が含まれています。これらの式は、SIP ベース (sip_udp、sip_tcp、sip_ssl) 仮想サーバーおよびグローバルバインドポイントにのみバインドできます。これらの式は、コンテンツスイッチング、レート制限、レスポンダ、および書き換えポリシーで使用できます。

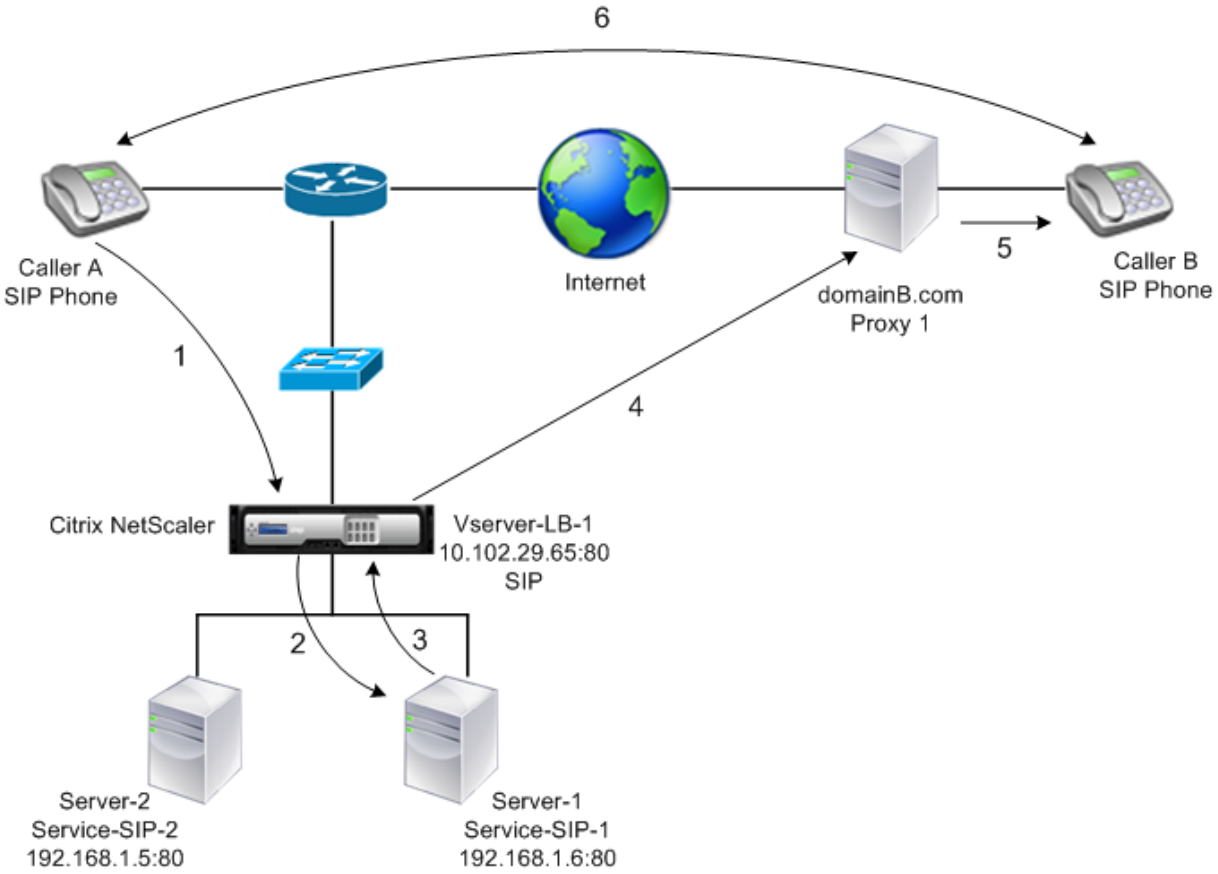
TCP または **UDP** を介した **SIP** シグナリングトラフィックのロードバランシングの設定

Citrix ADC は、TLS で保護された TCP トラフィックを含め、UDP または TCP を介して要求を送信する SIP サーバーの負荷を分散できます。ADC は、SIP サーバの負荷分散のために次のサービスタイプを提供します。

- SIP_UDP：SIP サーバが UDP を介して SIP メッセージを送信する場合に使用します。
- SIP_TCP：SIP サーバが TCP 経由で SIP メッセージを送信する場合に使用します。
- SIP_SSL：SSL または TLS を使用して TCP 経由の SIP シグナリングトラフィックを保護するために使用されます。Citrix ADC では、次のモードがサポートされています。
  - クライアント、ADC、および SIP サーバ間のエンドツーエンドの TLS 接続。
  - クライアントと ADC 間の TLS 接続、および ADC と SIP サーバ間の TCP 接続。
  - クライアントと ADC 間の TCP 接続、および ADC と SIP サーバ間の TLS 接続。

次の図は、TCP または UDP 経由で SIP メッセージを送信する SIP サーバのグループをロードバランシングするように設定されたセットアップのトポロジを示しています。

図 2：SIP ロードバランシングトポロジ



エンティティタイプ	名前	IP アドレス	ポート	サービスタイプ/プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.65	80	SIP_UDP / SIP_TCP / SIP_SSL
サービス	Service-SIP-1	192.168.1.6	80	SIP_UDP / SIP_TCP / SIP_SSL
	Service-SIP-2	192.168.1.5	80	SIP_UDP / SIP_TCP / SIP_SSL
モニター	デフォルト	なし	80	SIP_UDP / SIP_TCP / SIP_SSL

次に、SIP トラフィックの基本的なロードバランシングの設定の概要を示します。

1. サービスを設定し、ロードバランシングする SIP トラフィックのタイプごとに仮想サーバを設定します。
   - **SIP_UDP**：**UDP** 経由で SIP トラフィックをロードバランシングする場合。
   - **SIP_TCP**：**TCP** 経由で SIP トラフィックをロードバランシングする場合。
   - **SIP_SSL**：TCP を介した SIP トラフィックのロードバランシングおよびセキュリティ保護を行う場合。

   注:SIP_SSL を使用する場合は、必ず SSL 証明書とキーのペアを作成してください。詳細については、「証明書キーペアの追加」を参照してください。
2. サービスを仮想サーバーにバインドします。
3. デフォルト（**tcp-default**）以外のモニタでサービスの状態を監視する場合は、カスタムモニタを作成してサービスにバインドします。Citrix ADC には、**SIP-UDP** と **SIP-TCP** の **2** つのカスタムモニタータイプがあり、**SIP** サービスを監視できます。
4. SIP_SSL 仮想サーバーを使用している場合は、SSL 証明書とキーのペアを仮想サーバーにバインドします。
5. 展開環境で SIP サーバーの Gateway として Citrix ADC を使用している場合は、RNAT を構成します。
6. SIP サーバから開始された SIP メッセージに RPORT を追加する場合は、SIP パラメータを設定します。

コマンドラインインターフェイスを使用して **SIP** トラフィックの基本的なロードバランシング設定を構成するには

1 つ以上のサービスを作成します。コマンドプロンプトで入力します。

```
add service <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

例:

```
add service Service-SIP-UDP-1 192.0.2.5 SIP_UDP 80
<!--NeedCopy-->
```

作成したサービスを処理するために必要な数の仮想サーバーを作成します。仮想サーバのタイプは、バインドするサービスのタイプと一致する必要があります。コマンドプロンプトで入力します。

```
add lb vserver <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

例:


```
add lb vserver Vserver-LB-1 SIP_UDP 10.102.29.60 80
<!--NeedCopy-->
```

各サービスを仮想サーバにバインドします。コマンドプロンプトで入力します。

```
bind lb vserver <name> <serverName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-SIP-UDP-1
<!--NeedCopy-->
```

(オプション) SIP-UDP または SIP-TCP タイプのカスタムモニタを作成し、モニタをサービスにバインドします。コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <monitorType> [<interval>]

bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

例:

```
add lb monitor mon1 sip-UDP -sipMethod REGISTER -sipuRI sip:mon@test.
    com -sipregURI sip:mon@test.com -respcode 200

bind monitor mon1 Service-SIP-UDP-1
<!--NeedCopy-->
```

SIP_SSL 仮想サーバーを作成した場合は、SSL 証明書のキーペアを仮想サーバーにバインドします。コマンドプロンプトで、次のように入力します。コマンドプロンプトで、次のように入力します。

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -
    CA  - skipCAName
<!--NeedCopy-->
```


例:

```
bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1
<!--NeedCopy-->
```

ネットワークトポロジの必要に応じて RNAT を設定します。コマンドプロンプトで、次のコマンドのいずれかを入力して、ネットワークアドレスを条件として使用し、SNIP を NAT IP アドレスとして使用する RNAT エントリ、ネットワークアドレスを条件として使用する RNAT エントリ、および一意の IP をそれぞれ作成します。アドレスを NAT IP アドレス、ACL を条件として SNIP を NAT IP アドレスとして使用する RNAT エントリ、または ACL を条件として一意の IP アドレスを NAT IP アドレスとして使用する RNAT エントリ:

```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ …

show rnat
<!--NeedCopy-->
```

例:

```
add rnat RNAT-1 192.168.1.0 255.255.255.0

bind rnat RNAT-1 -natip 10.102.29.50
<!--NeedCopy-->
```

カスタム証明書とキーのペアを使用する場合は、カスタム証明書とキーのペアを nsrnatsip-127.0.0.1-5061 という名前の Citrix ADC 内部サービスにバインドします。

```
add ssl certKey <certkeyName> -cert <string> [-key <string>]

bind ssl service <serviceName> -certkeyName <string>
<!--NeedCopy-->
```

例:

```
add ssl certKey c1 -cert cert.epm -key key.ky

```



```
bind ssl service nsrnatsip-127.0.0.1-5061 -certkeyName c1
<!--NeedCopy-->
```

SIP サーバが開始する SIP メッセージに RPORT を追加する場合は、コマンドプロンプトで次のコマンドを入力します。

```
set lb sipParameters -rnatSrcPort <rnatSrcPort> -rnatDstPort<
    rnatDstPort> -retryDur <integer> -addRportVip <addRportVip> -
    sip503RateThreshold <sip503_rate_threshold_value>
<!--NeedCopy-->
```

UDP を介した **SIP** トラフィックのロードバランシングの設定例

```
add service service-UDP-1 10.102.29.5 SIP_UDP 80

Done

add lb vserver vserver-LB-1 SIP_UDP 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-UDP-1

Done

add lb mon mon1 sip-udp -sipMethod REGISTER -sipuRI sip:mon@test.com -
    sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-UDP-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
     -addRportVip ENABLED -sip503RateThreshold 1000

Done
```

```
<!--NeedCopy-->
```

TCP を介した **SIP** トラフィックのロードバランシングの設定例

```
add service service-TCP-1 10.102.29.5 SIP_TCP 80

Done

add lb vserver vserver-LB-1 SIP_TCP 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-TCP-1

Done

add lb mon mon1 sip-tcp -sipMethod REGISTER -sipuRI sip:mon@test.com -
    sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-TCP-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
     -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

TCP を介した **SIP** トラフィックのロードバランシングおよびセキュリティ保護の設定例

```
add service service-SIP-SSL-1 10.102.29.5 SIP_SSL 80

Done

```



```
add lb vserver vserver-LB-1 SIP_SSL 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-SIP-SSL

Done

add lb mon mon1 sip-tCP -sipMethod REGISTER -sipuRI sip:mon@test.com -
    sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-SIP-SSL

Done

bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
     -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

GUI を使用して **SIP** トラフィックの基本的なロードバランシング設定を構成するには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、タイプ SIP_UDP、SIP_TCP、または SIP_SSL の仮想サーバーを追加します。
2. [サービス] セクションをクリックし、タイプ SIP_UDP、SIP_TCP、または SIP_SSL のサービスを追加します。
3. （オプション）[**Monitor**] セクションをクリックし、タイプ SIP-UDP または SIP-TCP のモニタを追加します。
4. モニターをサービスにバインドし、サービスを仮想サーバーにバインドします。
5. SIP_SSL 仮想サーバーを作成した場合は、SSL 証明書のキーペアを仮想サーバーにバインドします。[証明書]

セクションをクリックし、証明書キーペアを仮想サーバーにバインドします。

6. ネットワークトポロジの必要に応じて RNAT を設定します。RNAT を設定するには、次の手順を実行します。

   a) [システム**] > [**ネットワーク**] > [**ルート] に移動します。
   b) [ルート] ページで、[**RNAT**] タブをクリックします。
   c) 詳細ウィンドウで、[**RNAT** の構成] をクリックします。
   d) [RNAT の構成] ダイアログボックスで、次のいずれかの操作を行います。
      - RNAT エントリを作成するための条件としてネットワークアドレスを使用する場合は、[ネットワーク] をクリックし、次のパラメータを設定します。
        - ネットワーク
        - ネットマスク
      - 拡張 ACL を RNAT エントリを作成するための条件として使用する場合は、[**ACL**] をクリックし、次のパラメータを設定します。
        - ACL 名
        - リダイレクトポート
   e) SNIP アドレスを NAT IP アドレスとして設定するには、手順 7 にスキップします。
   f) 一意の IP アドレスを NAT IP として設定するには、[使用可能な NAT IP] の一覧で、NAT IP として設定する IP アドレスを選択し、[追加] をクリックします。選択した NAT IP が [構成された NAT IP] リストに表示されます。
   g) ［Create］をクリックしてから、［Close］をクリックします。

   カスタム証明書とキーのペアを使用する場合は、カスタム証明書とキーのペアを **nsrnatsip-127.0.0.1-5061** という名前の Citrix ADC 内部サービスにバインドします。ペアをバインドするには、次の手順に従います。

   a) [トラフィック管理**] > [**負荷分散**] > [**サービス] に移動し、[内部サービス] タブをクリックします。
   b) nsrnatsip-127.0.0.1-5061 を選択し、［編集］をクリックします。
   c) [証明書] セクションをクリックし、証明書キーペアを内部サービスにバインドします。

7. SIP サーバが開始する SIP メッセージに RPORT を追加する場合は、SIP パラメータを設定します。[トラフィック管理**] > [**ロードバランシング] に移動し、[SIP 設定の変更] をクリックし、さまざまな SIP パラメータを設定します。

SIP 式およびポリシーの例：クライアント要求で圧縮が有効になっている

Citrix ADC は圧縮されたクライアント SIP 要求を処理できないため、クライアント SIP 要求は失敗します。

クライアントからの SIP NEGOTIATE メッセージを代行受信し、圧縮ヘッダーを検索する応答側ポリシーを設定できます。メッセージに圧縮ヘッダーが含まれている場合、ポリシーは「400 Bad Request」と応答し、クライアントは要求を圧縮せずに再送信します。

コマンドプロンプトで、次のコマンドを入力してレスポンダーポリシーを作成します。


```
add responder action sipaction1 respondwith q{
 ”SIP/2.0 400 Bad Requestrnrn” }


Done

add responder policy sippol1

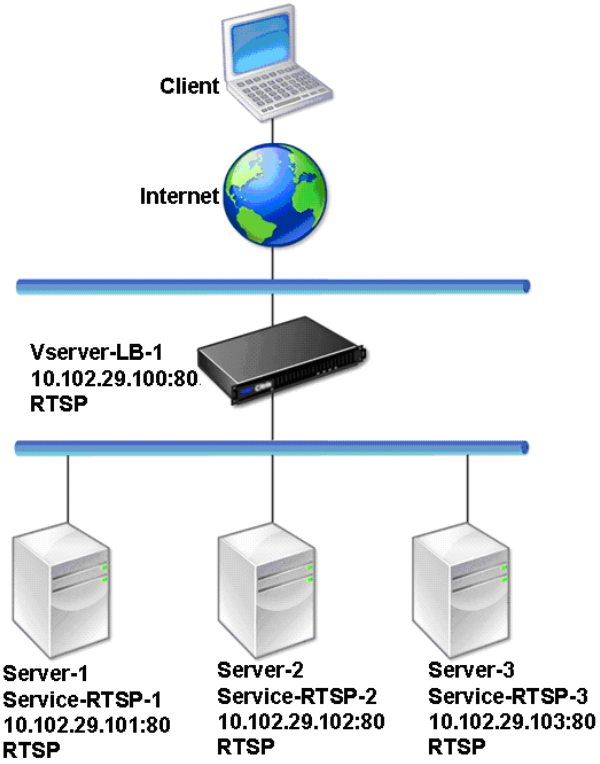
add responder policy sippol1 ”SIP.REQ.METHOD.EQ(”NEGOTIATE”)&&SIP.REQ.
   HEADER(”Compression”).EXISTS” sipaction1
<!--NeedCopy-->
```

RTSP サーバの負荷分散

October 7, 2021

Citrix ADC アプライアンスは、RTSP サーバーの負荷を分散して、ネットワークを介したオーディオおよびビデオストリームのパフォーマンスを向上させることができます。次の図は、RTSP サーバのグループを負荷分散するように構成されたロードバランシングセットアップのトポロジを示しています。

図 1: RTSP のロードバランシングトポロジ

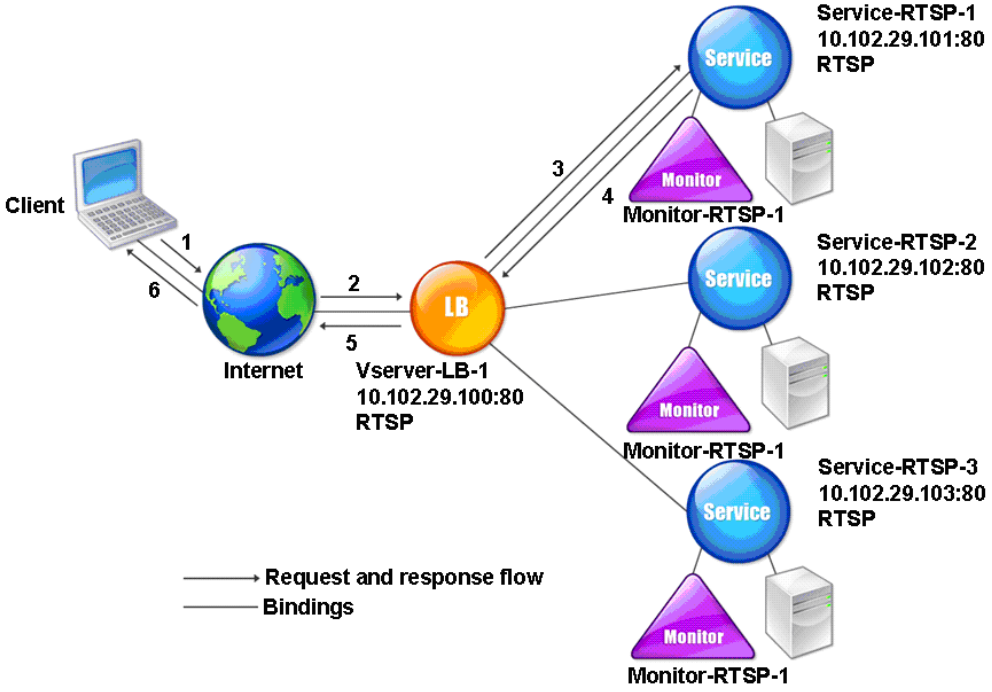


この例では、Service-RTSP-1、Service-RTSP-2、および Service-RTSP-3 が仮想サーバ Vserver-LB-1 にバインドされています。次の表に、例エンティティの名前と値を示します。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバー	Vserver-LB-1	10.102.29.100	554	RTSP
サービス	Service-RTSP-1	10.102.29.101	554	RTSP
	Service-RTSP-2	10.102.29.102	554	RTSP
	Service-RTSP-3	10.102.29.103	554	RTSP
モニター	Monitor-RTSP-1	なし	554	RTSP

次の図は、RTSP 設定で使用されるロードバランシングエンティティを示しています。

図 2: ロードバランシング RTSP サーバエンティティモデル



RTSP サーバーの基本的な負荷分散設定を構成するには、基本負荷分散の設定を参照してください。RTSP タイプのサービスおよび仮想サーバを作成します。基本的な負荷分散セットアップを構成すると、デフォルトの TCP デフォルトモニターがサービスにバインドされます。RTSP モニタをこれらのサービスにバインドするには、モニタをサービスへのバインドを参照してください。次の手順では、RTSP サーバをチェックするモニタの作成方法について説明します。

CLI を使用して RTSP モニタを設定するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type>
<!--NeedCopy-->
```

例:

```
add lb monitor Monitor-RTSP-1 RTSP
<!--NeedCopy-->
```


GUI を使用して **RTSP** モニタを設定するには

[トラフィック管理] > [ロードバランシング] > [モニター] に移動し、RTSP タイプのモニタを作成します。

リモートデスクトッププロトコルサーバーの負荷分散

October 7, 2021

リモートデスクトッププロトコル (RDP) はマルチチャネル対応のプロトコルで、プレゼンテーションデータ、シリアルデバイス通信、ライセンス情報、高度に暗号化されたデータ (キーボードとマウスの操作) などを転送するための個別の仮想チャネルを可能にします。

RDP は、ネットワーク上の別のコンピューターに GUI を提供するために使用されます。RDP は Windows ターミナルサーバーで使用され、低帯域幅の接続でも、マウスの動きとキー押下をほぼリアルタイムに伝送して高速アクセスを提供します。

リモートデスクトップサービスを提供するために複数のターミナルサーバーを展開すると、Citrix ADC アプライアンスはターミナルサーバー（Windows 2003 および 2008 Server エンタープライズエディション）の負荷分散を提供します。アプリケーションにリモートでアクセスしているユーザーは、アプリケーションをリモートマシンで実行したままにして、ローカルマシンをシャットダウンしたい場合があります。したがって、ユーザはリモートアプリケーションからログアウトせずにローカルアプリケーションを閉じます。リモートマシンに再接続した後、ユーザーはリモートアプリケーションを続行できる必要があります。この機能を提供するために、Citrix ADC RDP 実装では、ターミナルサービスセッションディレクトリまたはブローカーによって設定されたルーティングトークン（Cookie）が優先されるため、クライアントは以前に接続されていたターミナルサーバーに再接続できます。セッションディレクトリは、Windows 2003 ターミナルサーバーに実装され、Windows 2008 ターミナルサーバーではブローカーと呼ばれます。

クライアントと負荷分散仮想サーバー間で TCP 接続が確立されると、Citrix ADC は指定された負荷分散方法を適用し、いずれかのターミナルサーバーに要求を転送します。ターミナルサーバーはセッションディレクトリをチェックして、クライアントがドメイン内の他のターミナルサーバーで実行されているセッションがあるかどうかを判断します。

他のターミナルサーバーにアクティブなセッションがない場合、ターミナルサーバーはクライアント要求を処理して応答し、Citrix ADC アプライアンスはクライアントに応答を転送します。

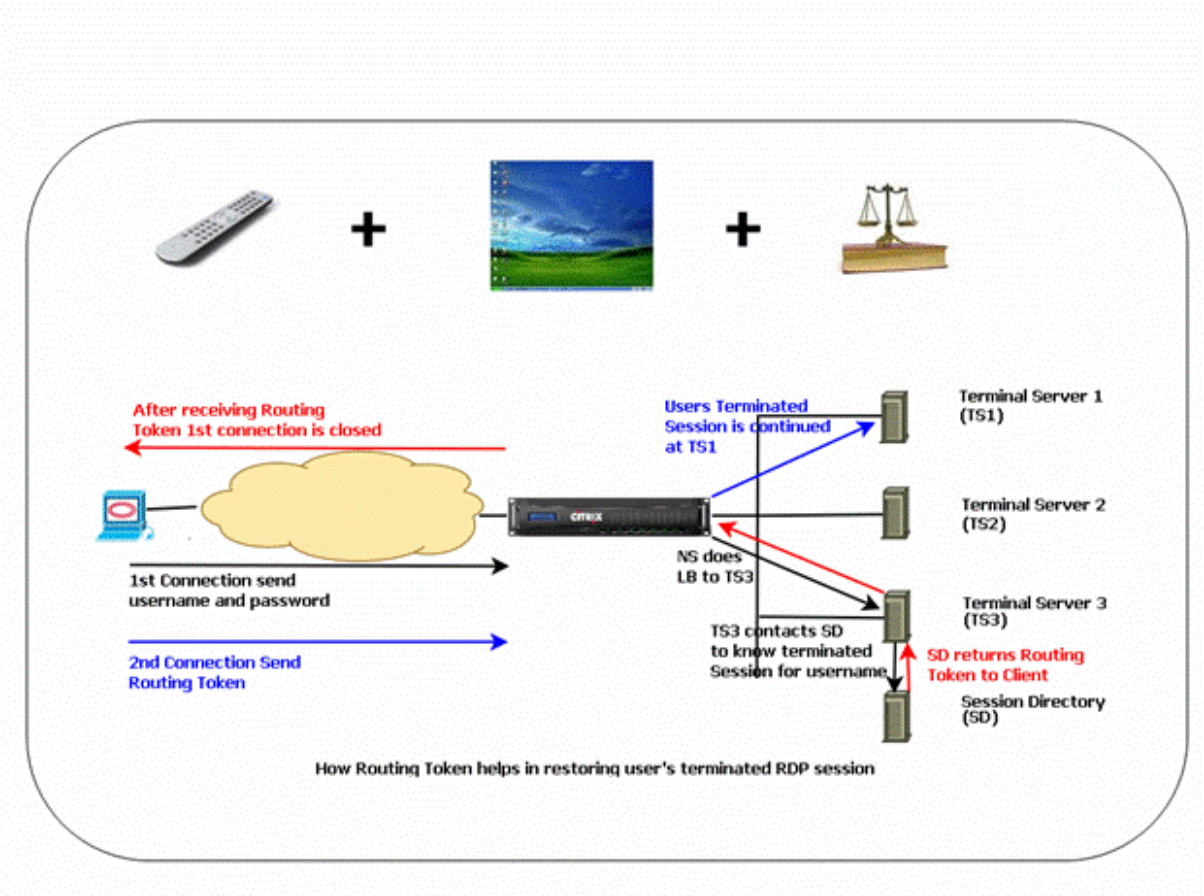
他のターミナルサーバーにアクティブなセッションがある場合、要求を受信するターミナルサーバーはアクティブなセッションの詳細を含むクッキー（ルーティングトークンと呼ばれる）を挿入し、パケットを Citrix ADC アプライアンスに返し、パケットをクライアントに返します。サーバーはクライアントとの接続を閉じます。クライアントが接続を再試行すると、Citrix ADC は Cookie 情報を読み取り、クライアントがアクティブなセッションを持つターミナルサーバーにパケットを転送します。

クライアントマシン上のユーザーがサービスの継続を経験し、特定のアクションを実行する必要はありません。

注:Windows セッションディレクトリ機能を使用するには、Windows XP で最初にリリースされたリモートデスクトップクライアントが必要です。Windows 2000 または Windows NT 4.0 ターミナルサーバークライアントとのセッションが切断され、クライアントが再接続すると、接続が確立されたサーバーが負荷分散アルゴリズムによって選択されます。

次の図は、RDP ロードバランシングを示しています。

図 1: RDP のロードバランシングトポロジ



> 注
>
> - RDP サービスが構成されると、ルーティングトークンを使用して永続性が自動的に維持されます。永続性を明示的に有効にする必要はありません。
> - Citrix ADC アプライアンスは、IP ベースのクッキーのみをサポートします。
> - nsrdp.pl スクリプトは、現在のバージョンの Windows サーバーではサポートされていません。

ログアウトせずに RDP セッションが切断されたときに、2 つのターミナルサーバー間でフラッピングが発生しないように、切断された RDP セッションがバックエンドのターミナルサーバーでクリアされていることを確認します。詳しくは、「https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758177(v=ws.10)##BKMK_2」を参照してください。

RDP サービスを追加すると、デフォルトで Citrix ADC は TCP タイプのモニターを追加し、サービスにバインドし

ます。デフォルトのモニターは、RDP サービス用に指定されたサーバーの 3389 ポートにリスニングプロセスが存在するかどうかを確認する単純な TCP モニターです。3389 でリスニングプロセスがある場合、Citrix ADC はこのサービスを UP としてマークし、リスニングプロセスがない場合は、サービスをダウンとしてマークします。

RDP サービスをより効率的に監視するには、既定のモニターに加えて、RDP プロトコル用のスクリプトモニターを構成できます。スクリプトモニターを構成すると、Citrix ADC は指定されたサーバーへの TCP 接続を開き、RDP パケットを送信します。モニタは、物理サーバから接続の確認を受信した場合のみ、サービスを UP としてマークします。したがって、スクリプトモニターから、Citrix ADC は、RDP サービスが要求を処理する準備ができているかどうかを知ることができます。

モニタはユーザータイプのモニタで、スクリプトは Citrix ADC /nsconfig/monitors/nsrdp.pl にあります。ユーザーモニターを構成すると、Citrix ADC によってスクリプトが自動的に実行されます。スクリプトモニターを構成するには、モニターを追加し、RDP サービスにバインドします。

RDP 負荷分散を構成するには、RDP タイプのサービスを作成し、RDP 仮想サーバーにバインドします。

コマンドラインインターフェイスを使用して **RDP** 負荷分散サービスを構成するには

コマンドプロンプトで次のコマンドを入力して、RDP 負荷分散セットアップを構成し、構成を確認します。

```
add service <name>@ <serverName> <serviceType> <port>
<!--NeedCopy-->
```

注：サービスを追加するには、前のコマンドを繰り返します。

例

```
> add service ser1 10.102.27.182 RDP 3389
Done
> add service ser2 10.102.27.183 RDP 3389
Done
>show service ser1
ser1 (10.102. 27.182:3389) - RDP
        State: UP
…
            Server Name: 10.102.27.182
            Server ID : 0           Monitor Threshold : 0
        Down state flush: ENABLED
…
1)      Monitor Name: tcp-default
                State: UP       Weight: 1
…
```



```
                Response Time: 4.152 millisec
 Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **RDP** 負荷分散サービスを構成するには

[トラフィック管理**] > [**負荷分散**] > [**サービス**]** に移動し、RDP タイプのサービスを作成します。

コマンドラインインターフェイスを使用して **RDP** 負荷分散仮想サーバーを構成するには

コマンドプロンプトで次のコマンドを入力して、RDP 負荷分散仮想サーバーを構成し、構成を確認します。

```
add lb vserver <name>@ <serviceType> <ipAddress> <port>

bind lb vserver <name>@ <serviceName>

Bind all the RDP services to be load balanced to the virtual server.
<!--NeedCopy-->
```

例:

この例では、RDP 仮想サーバーにバインドされた 2 つの RDP サービスがあります。

```
add lb vs v1 rdP 10.102.27.186 3389
Done

bind lb vs v1 ser1
service ”ser1” bound

bind lb vs v1 ser2
service ”ser2” bound
Done

sh lb vs v1
v1 (10.102.27.186:3389) - RDP   Type: ADDRESS
State: UP
…
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION
  Current Method: Round Robin, Reason: A new service is bound
Mode: IP
```



```
Persistence: NONE
  L2Conn: OFF

1) ser1 (10.102.27.182: 3389) - RDPState: UP   Weight: 1
2) ser2 (10.102.27.183: 3389) - RDPState: UP   Weight: 1
Done
<!--NeedCopy-->
```

構成ユーティリティを使用して **RDP** 負荷分散仮想サーバーを構成するには

[トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動し、RDP タイプの仮想サーバーを作成し、この仮想サーバーに RDP サービスをバインドします。

コマンドラインインターフェイスを使用して **RDP** サービスのスクリプトモニターを構成するには

コマンドプロンプトで、次のコマンドを入力します。

```
add lb monitor <monitorName> USER -scriptName nsrdp.pl

bind lb monitor <monitorName> <rdpServiceName>
<!--NeedCopy-->
```

例**:**

```
add service ser1 10.102.27.182 RDP 3389

add lb monitor RDP_MON USER -scriptName nsrdp.pl

bind lb monitor RDP_MON ser1

<!--NeedCopy-->
```

構成ユーティリティを使用して **RDP** サービスのスクリプトモニターを構成するには

1. **Traffic Management > Load Balancing > Monitors** に移動してタイプ USER のモニターを作成します。
2. [特殊パラメータ] の [スクリプト名] リストで [nsrdp.pl] を選択し、このモニタを RDP サービスにバインドします。

Microsoft Exchange サーバーの負荷分散

May 9, 2022

このドキュメントでは、Citrix ADC アプライアンスを使用した Microsoft Exchange サーバーの負荷分散に関する推奨される構成例について説明します。

Citrix ADM StyleBooks は、Exchange の Citrix ADC 負荷分散構成を簡素化します。詳細については、「Microsoft Exchange StyleBook」を参照してください。

> 注:
>
> Microsoft Exchange の負荷分散は、単一の負荷分散仮想サーバーを使用して実行できません。代わりに、このドキュメントに記載されている推奨設定に従ってください。

Microsoft Exchange 2016 とそれ以降のバージョンの違い

- RPC ポートは使用されないため、Exchange 2016 で静的リモートプロシージャコール (RPC) ポートを構成する必要はありません。
- 「2016 より下の Exchange のバージョン用」という名前のセクションはすべて、Exchange 2016 では必要ありません。
- 2016 以外のバージョンのいずれかをすでに設定し、2016 に移行した場合は、それらを削除する必要はありません。存在しても問題はないからです。

注意事項

- 2016 より下の Exchange サーバーを使用したリモートプロシージャコール (RPC) の場合、Exchange CAS サーバーを静的ポート割り当て用に構成する必要があります。詳細については、Microsoft ドキュメントの「Exchange 2010 クライアントアクセスサーバー: 静的 RPC ポートを構成する 」を参照してください。
- この構成では、SSL オフロードに Citrix ADC アプライアンスを使用することを前提としています。詳細については、「Exchange 2010 で SSL オフロードを構成する方法」または「Exchange2013 で SSL オフロードを構成する方法」を参照してください。
- Citrix ADC アプライアンスの SSL オフロード機能を使用しない場合は、サービスグループ`CAS_servicegroup_http`とモニターをタイプ`SSL`に変更し、バインドをポート443に変更します。
- サージ保護は Microsoft Exchange と互換性がありません。Microsoft Exchange に関連するサービスまたはサービスグループでは有効にしないでください。サージ保護を有効にすると、接続性と信頼性の問題が発生します。
- 次の変数を適切な情報に置き換えます。

- {HTTP パブリック IP}: パブリックエクスチェンジ HTTP エンドポイントの IP アドレス
- {RPC パブリック IP}: パブリック Exchange RPC エンドポイントの IP アドレス（HTTP パブリック IP と同じでもよい）
- {Timeout}: 必要なタイムアウト（秒）。標準的な作業シフト時間（つまり 8 時間）の長さにすることを推奨
- {perstimeOut}: 必要なタイムアウト（分単位）。前述の [タイムアウト] 設定に対応する必要があります。
- {AB ポート}: RPC アドレス帳 TCP ポート（通常 59601）
- {CA ポート}: RPC クライアントアクセス TCP ポート（通常は 59600）
- {certKey} — SSL 証明書キー
- {CAS-1 サーバ}: CAS サーバの IP アドレス
- {CAS-2 サーバ}: CAS サーバの IP アドレス

Microsoft Exchange サーバーのすべてのバージョンで推奨される構成例

サービスグループ**:**

```
add serviceGroup CAS_servicegroup_http HTTP -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout { Timeout }
 -svrTimeout { Timeout }
 -CKA NO -TCPB NO -CMP YES
add serviceGroup CAS_servicegroup_rpc_epm TCP -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout { Timeout }
 -svrTimeout { Timeout }
 -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_http {
 CAS-1 Server }
  80 -CustomServerID ””None””
bind serviceGroup CAS_servicegroup_http {
 CAS-2 Server }
  80 -CustomServerID ””None””
bind serviceGroup CAS_servicegroup_rpc_epm {
 CAS-1 Server }
  135 -CustomServerID ””None””
bind serviceGroup CAS_servicegroup_rpc_epm {
 CAS-2 Server }
  135 -CustomServerID ””None””
<!--NeedCopy-->
```


モニタ:

```
add lb monitor CAS_monitor_rpc_epm TCP -LRTM ENABLED -destPort 135
add lb monitor mon_http_ecv HTTP-ECV -recv 403 -LRTM DISABLED
bind serviceGroup CAS_servicegroup_http -monitorName mon_http_ecv
bind serviceGroup CAS_servicegroup_rpc_epm -monitorName
   CAS_monitor_rpc_epm
<!--NeedCopy-->
```

仮想サーバーの負荷分散:

```
add lb vserver CAS_vserver_owa SSL 0.0.0.0 0 -persistenceType
   COOKIEINSERT -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_as SSL 0.0.0.0 0 -persistenceType RULE -
   timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -rule "HTTP.REQ.HEADER("Authorization")" -
     cltTimeout {
 Timeout }

add lb vserver CAS_vserver_oa SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_ews SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_ad SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_oab SSL 0.0.0.0 0 -timeout {
```

```
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

set ssl vserver CAS_vserver_owa -sslRedirect ENABLED
bind ssl vserver CAS_vserver_owa -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_oab -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_as -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_oa -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_ews -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_ad -certkeyName {
 CertKey }

bind lb vserver CAS_vserver_owa CAS_servicegroup_http
bind lb vserver CAS_vserver_oab CAS_servicegroup_http
bind lb vserver CAS_vserver_as CAS_servicegroup_http
bind lb vserver CAS_vserver_oa CAS_servicegroup_http
bind lb vserver CAS_vserver_ews CAS_servicegroup_http
bind lb vserver CAS_vserver_ad CAS_servicegroup_http
add lb vserver CAS_vserver_rpc_epm TCP {
 RPC Public IP }
  135 -timeout {
 PersTimeout }
  -cltTimeout {
 Timeout }
  -comment ”vserver for RPC End Point Mapper”
bind lb vserver CAS_vserver_rpc_epm CAS_servicegroup_rpc_epm
<!--NeedCopy-->
```

永続性グループ**:**

```
add lb group CAS_persistency_group_sourceip
bind lb group CAS_persistency_group_sourceip CAS_vserver_oa
```



```
bind lb group CAS_persistency_group_sourceip CAS_vserver_oab
bind lb group CAS_persistency_group_sourceip CAS_vserver_ews
bind lb group CAS_persistency_group_sourceip CAS_vserver_ad
bind lb group CAS_persistency_group_sourceip CAS_vserver_rpc_epm
set lb group CAS_persistency_group_sourceip -persistenceType SOURCEIP -
    timeout {
 PersTimeout }

<!--NeedCopy-->
```

HTTP サービスのコンテンツスイッチング**:**

```
add cs vserver CAS_vserver_cs SSL {
 Public IP }
  443 -cltTimeout {
 Timeout }
  -caseSensitive OFF -comment "Exchange CS VServer"
bind ssl vserver CAS_vserver_cs -certkeyName {
 CertKey }

add cs action CAS_action_cs_owa -targetLBVserver CAS_vserver_owa
add cs action CAS_action_cs_oab -targetLBVserver CAS_vserver_oab
add cs action CAS_action_cs_as -targetLBVserver CAS_vserver_as
add cs action CAS_action_cs_oa -targetLBVserver CAS_vserver_oa
add cs action CAS_action_cs_ews -targetLBVserver CAS_vserver_ews
add cs action CAS_action_cs_autodiscover -targetLBVserver
    CAS_vserver_ad
add cs policy CAS_policy_cs_owa -rule "HTTP.REQ.URL.SET_TEXT_MODE(
    IGNORECASE).STARTSWITH("/owa")" -action CAS_action_cs_owa
add cs policy CAS_vserver_oab -rule "HTTP.REQ.URL.SET_TEXT_MODE (
    IGNORECASE).STARTSWITH("/OAB")"
add cs policy CAS_policy_cs_as -rule "HTTP.REQ.URL.SET_TEXT_MODE(
    IGNORECASE).STARTSWITH("/Microsoft-Server-ActiveSync")" -action
    CAS_action_cs_as
add cs policy CAS_policy_cs_autodiscover -rule "HTTP.REQ.URL.
    SET_TEXT_MODE(IGNORECASE).STARTSWITH("/Autodiscover")" -action
    CAS_action_cs_autodiscover
add cs policy CAS_policy_cs_oa -rule "HTTP.REQ.URL.SET_TEXT_MODE(
    IGNORECASE).STARTSWITH("/rpc")" -action CAS_action_cs_oa
add cs policy CAS_policy_cs_ews -rule "HTTP.REQ.URL.SET_TEXT_MODE(
    IGNORECASE).STARTSWITH("/EWS")" -action CAS_action_cs_ews

bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_oa -priority
```

```
    90
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_owa -priority
    100
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_oab -priority
    100
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_as -priority
    110
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_autodiscover -
    priority 120
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_ews -priority
    130
bind cs vserver CAS_vserver_cs -lbvserver CAS_vserver_owa
<!--NeedCopy-->
```

2016 年より前のバージョンの Microsoft Exchange サーバーの推奨構成例

追加のサービスグループ**:**

```
add serviceGroup CAS_servicegroup_rpc_ca TCP -maxClient 0 -maxReq 0 -
    cip DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
add serviceGroup CAS_servicegroup_rpc_ab TCP -maxClient 0 -maxReq 0 -
    cip DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_rpc_ca {
 CAS-1 Server }
  {
 CA Port }
  -CustomServerID ""None""
bind serviceGroup CAS_servicegroup_rpc_ca {
 CAS-2 Server }
  {
 CA Port }
  -CustomServerID ""None""
bind serviceGroup CAS_servicegroup_rpc_ab {
 CAS-1 Server }
```



```
23    {
24   AB Port }
25    -CustomServerID ””None””
26  bind serviceGroup CAS_servicegroup_rpc_ab {
27   CAS-2 Server }
28    {
29   AB Port }
30    -CustomServerID ””None””
31  <!--NeedCopy-->
```

追加のモニタ:

```
1  add lb monitor CAS_monitor_rpc_ca TCP -LRTM ENABLED -destPort {
2   CA Port }
3
4  add lb monitor CAS_monitor_rpc_ab TCP -LRTM ENABLED -destPort {
5   AB Port }
6
7  bind serviceGroup CAS_servicegroup_rpc_ca -monitorName
       CAS_monitor_rpc_ca
8  bind serviceGroup CAS_servicegroup_rpc_ab -monitorName
       CAS_monitor_rpc_ab
9  <!--NeedCopy-->
```

追加の負荷分散仮想サーバー:

```
1  add lb vserver CAS_vserver_rpc_ab TCP {
2   RPC Public IP }
3    {
4   AB Port }
5    -timeout {
6   PersTimeout }
7    -cltTimeout {
8   Timeout }
9    -comment ”vserver for  RPC Address Book”
10 add lb vserver CAS_vserver_rpc_ca TCP {
11  RPC Public IP }
12   {
13  CA Port }
14   -timeout {
15  PersTimeout }
16   -cltTimeout {
```

```
 Timeout }
  -comment ”vserver for  RPC Client Access”
bind lb vserver CAS_vserver_rpc_ab CAS_servicegroup_rpc_ab
bind lb vserver CAS_vserver_rpc_ca CAS_servicegroup_rpc_ca
<!--NeedCopy-->
```

追加の永続性グループ**:**

```
bind lb group CAS_persistency_group_sourceip CAS_vserver_rpc_ab
bind lb group CAS_persistency_group_sourceip CAS_vserver_rpc_ca
<!--NeedCopy-->
```

Microsoft Exchange サーバー **2016** 以降のバージョンの推奨構成例

追加の負荷分散仮想サーバー**:**

```
add lb vserver CAS_vserver_mapi SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

bind ssl vserver CAS_vserver_mapi -certkeyName {
 CertKey }

bind lb vserver CAS_vserver_mapi CAS_servicegroup_http
<!--NeedCopy-->
```

追加の永続性グループ**:**

```
bind lb group CAS_persistency_group_sourceip CAS_vserver_mapi
<!--NeedCopy-->
```

HTTP サービスのコンテンツスイッチング**:**

```
add cs action CAS_action_cs_mapi -targetLBVserver CAS_vserver_mapi
add cs policy CAS_policy_cs_mapi -rule ”HTTP.REQ.URL.SET_TEXT_MODE(
    IGNORECASE).STARTSWITH(”/mapi”)” -action CAS_action_cs_mapi
```



```
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_mapi -priority
    140
<!--NeedCopy-->
```

オプションの構成

Outlook ウェブアプリ **(OWA)** の **HTTPS** リダイレクト**:**

```
add lb vserver CAS_vserver_owa_http_redirect HTTP {
 HTTP Public IP }
  80 -persistenceType COOKIEINSERT -timeout {
 PersTimeout }
  -lbMethod ROUNDROBIN -redirectURL "https://mail.example.com/owa" -
     cltTimeout {
 Timeout }

<!--NeedCopy-->
```

注: 適切な HTTPS リダイレクト URL に置き換えてください。

/owa 書き換えのポリシー**:**

```
add rewrite action owa_rewrite replace http.REQ.URL ""/owa""
add rewrite policy owa_rewrite_policy "http.req.url.eq("/")"
   owa_rewrite
bind lb vserver CAS_vserver_owa -policyName owa_rewrite_policy -
   priority 100 -gotoPriorityExpression END -type REQUEST
add responder action action_responder_owa redirect ""https://www.
   example.com/owa""
add responder policy policy_responder_owa HTTP.REQ.IS_VALID
   action_responder_owa
set responder param -undefAction NOOP
bind lb vserver CAS_vserver_owa -policyName policy_responder_owa -
   priority 100 -gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

注: 適切な HTTPS リダイレクト URL に置き換えてください。

SMTP のサポート**:**

次の構成では、CAS サーバが検証のために送信側の SMTP サーバの IP アドレスを確認できるように、USIP を有効にする必要があります。また、この設定では、CAS サーバーのデフォルトゲートウェイが ADC アプライアンスの SNIP アドレスを指すように構成されている必要があります。


```
add lb vserver CAS_vserver_smtp TCP {
 HTTP Public IP }
  25 -persistenceType SOURCEIP -timeout 60 -lbMethod LEASTCONNECTION -
     cltTimeout 30
add serviceGroup CAS_servicegroup_smtp TCP -maxClient 0 -maxReq 0 -cip
    DISABLED -usip YES -SP OFF -useproxyport YES -cltTimeout 30 -
    svrTimeout 30 -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_smtp {
 CAS-1 Server }
  25 -CustomServerID ""None"" bind serviceGroup CAS_servicegroup_smtp {
 CAS-2 Server }
  25 -CustomServerID ""None""
bind lb vserver CAS_vserver_smtp CAS_servicegroup_smtp
<!--NeedCopy-->
```

郵便局プロトコルバージョン **3 (POP3)** のサポート**:**

```
add lb vserver CAS_vserver_pop3 TCP {
 HTTP Public IP }
  110 -persistenceType SOURCEIP -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add serviceGroup CAS_servicegroup_pop3 TCP -maxClient 0 -maxReq 0 -cip
    DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_pop3 {
 CAS-1 Server }
  110 -CustomServerID ""None"" bind serviceGroup CAS_servicegroup_pop3
     {
 CAS-2 Server }
  110 -CustomServerID ""None""
bind lb vserver CAS_vserver_pop3 CAS_servicegroup_pop3
<!--NeedCopy-->
```


> 注:
>
> SSL 暗号化 POP3 の前の構成を実行するには、ポートを 995 に、仮想サーバー/サービスタイプを SSL に変更します。適切な SSL 証明書もバインドします。

IMAP のサポート**:**

```
add lb vserver CAS_vserver_imap TCP {
 HTTP Public IP }
  143 -persistenceType SOURCEIP -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add serviceGroup CAS_servicegroup_imap TCP -maxClient 0 -maxReq 0 -cip
    DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_imap {
 CAS-1 Server }
  143 -CustomServerID ""None"" bind serviceGroup CAS_servicegroup_imap
      {
 CAS-2 Server }
  143 -CustomServerID ""None""
bind lb vserver CAS_vserver_imap CAS_servicegroup_imap
<!--NeedCopy-->
```

> 注:
>
> SSL 暗号化 IMAP の前の設定を実行するには、ポートを 993 に、仮想サーバ/サービスタイプを SSL に変更します。適切な SSL 証明書もバインドします。

そのほかの参照先

- 電子メールセキュリティフィルタリングを使用した Microsoft Exchange の負荷分散サーバの構成
- Microsoft Exchange 2016 で NetScaler を展開する

使用例 **1**：**SMPP** 負荷分散

October 7, 2021

個人と銀行、広告主、ディレクトリサービスなどの付加価値サービスプロバイダーの間で、Short Message peer to peer（SMPP; ショートメッセージピアツーピア）プロトコルを使用して、毎日数百万のショートメッセージが交換されます。多くの場合、サーバーが過負荷になり、トラフィックがサーバー間で最適に分散されないため、メッセージの配信が遅れます。Citrix ADC は SMPP 負荷分散をサポートし、サーバー間でメッセージを最適に分散し、パフォーマンスの低下や停止を防ぎます。

Citrix ADC は、クライアントからメッセージを受信する場合はサーバー側で、サーバーからメッセージを受信する場合はクライアント側で負荷分散を実行します。

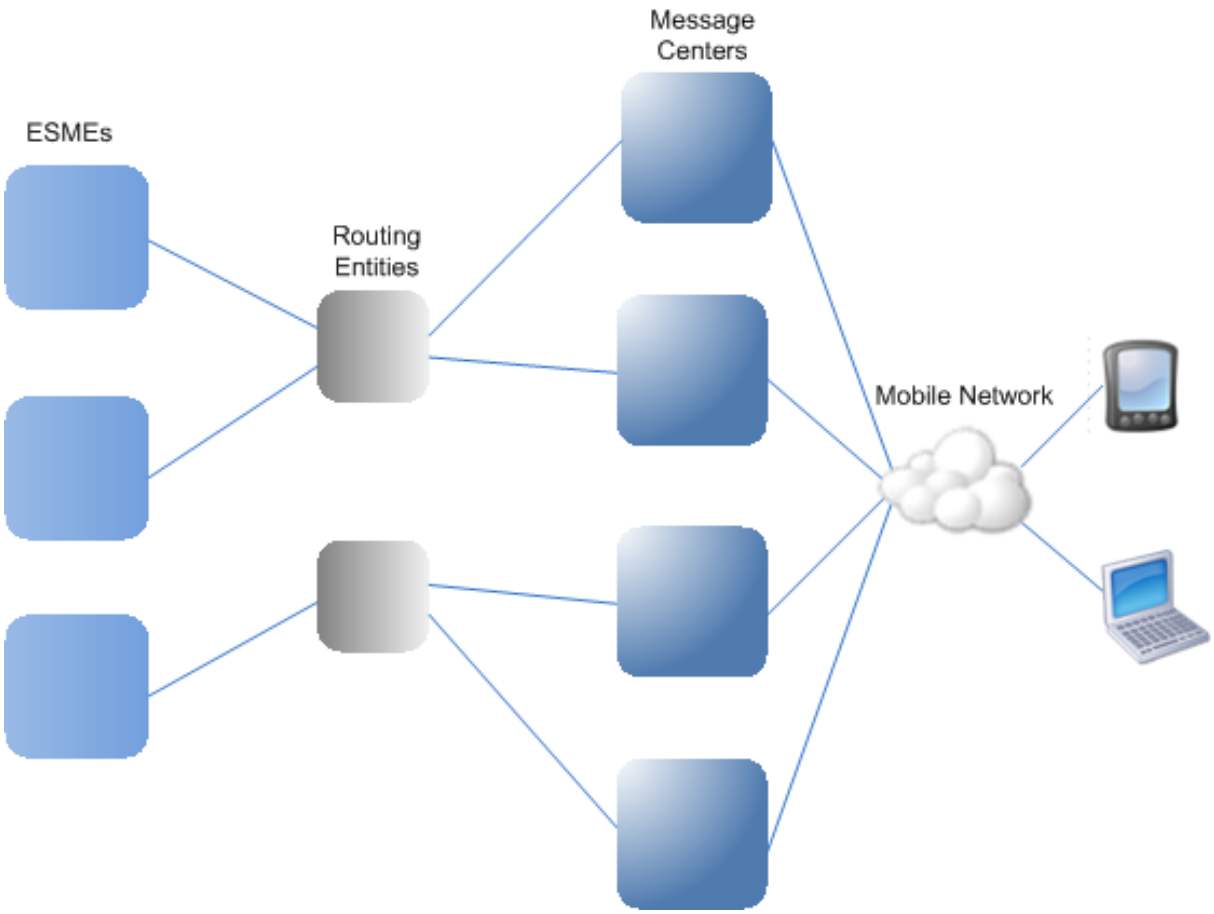
Citrix ADC による SMPP メッセージの負荷分散には、次のような利点があります。

- サーバへの負荷分散が改善され、エンドユーザーへの応答時間が短縮されます。
- サーバの健全性監視とフェイルオーバー機能の向上
- クライアント構成を変更することなく、新しいサーバ（メッセージ・センター）を迅速かつ容易に追加
- 高可用性

SMPP の概要

SMPP は、長寿命の TCP 接続を介した外部ショートメッセージエンティティ（ESME）、ルーティングエンティティ（RE）、およびメッセージセンター（MC）間でショートメッセージを転送するためのアプリケーション層プロトコルです。友人、連絡先、銀行（モバイルバンキング）、広告主（モバイルコマース）、ディレクトリサービスなどの第三者との間でショートメッセージサービス（SMS）メッセージを送信するために使用されます。ESME（非モバイルエンティティ）からのメッセージは MC に到着し、MC によって携帯電話などのショートメッセージエンティティ（SME）に配信されます。SMPP は、SME が第三者に短いメッセージを送信するためにも使用されます（たとえば、製品の購入、請求書の支払い、資金振替など）。これらのメッセージは MC に到着し、宛先 MC または ESME に転送されます。

次の図は、モバイルネットワーク内の ESME、RES、および MC のさまざまな SMPP エンティティを示しています。



モバイルネットワークにおけるさまざまな **SMPP** エンティティのアーキテクチャの概要

注: クライアントと ESME という用語は、ドキュメント全体で同じ意味で使用されます。

ESME（クライアント）は、送信機、受信機、またはトランシーバの 3 つのモードのいずれかで MC への接続を開きます。送信機として、配信のためのメッセージのみを送信できます。受信者は、メッセージのみを受信できます。トランシーバとして、ESME はメッセージの送信と受信の両方を行うことができます。ESME は、バインドトランスミッタ、バインドレシーバ、バインドトランシーバの 3 つのメッセージ（PDU とも呼ばれます）の 1 つを MC に送信します。MC は、要求に応じて、バインドトランスミッタレスプ、バインドレシーバレスプ、またはバインドトランスミッタレスプで応答します。

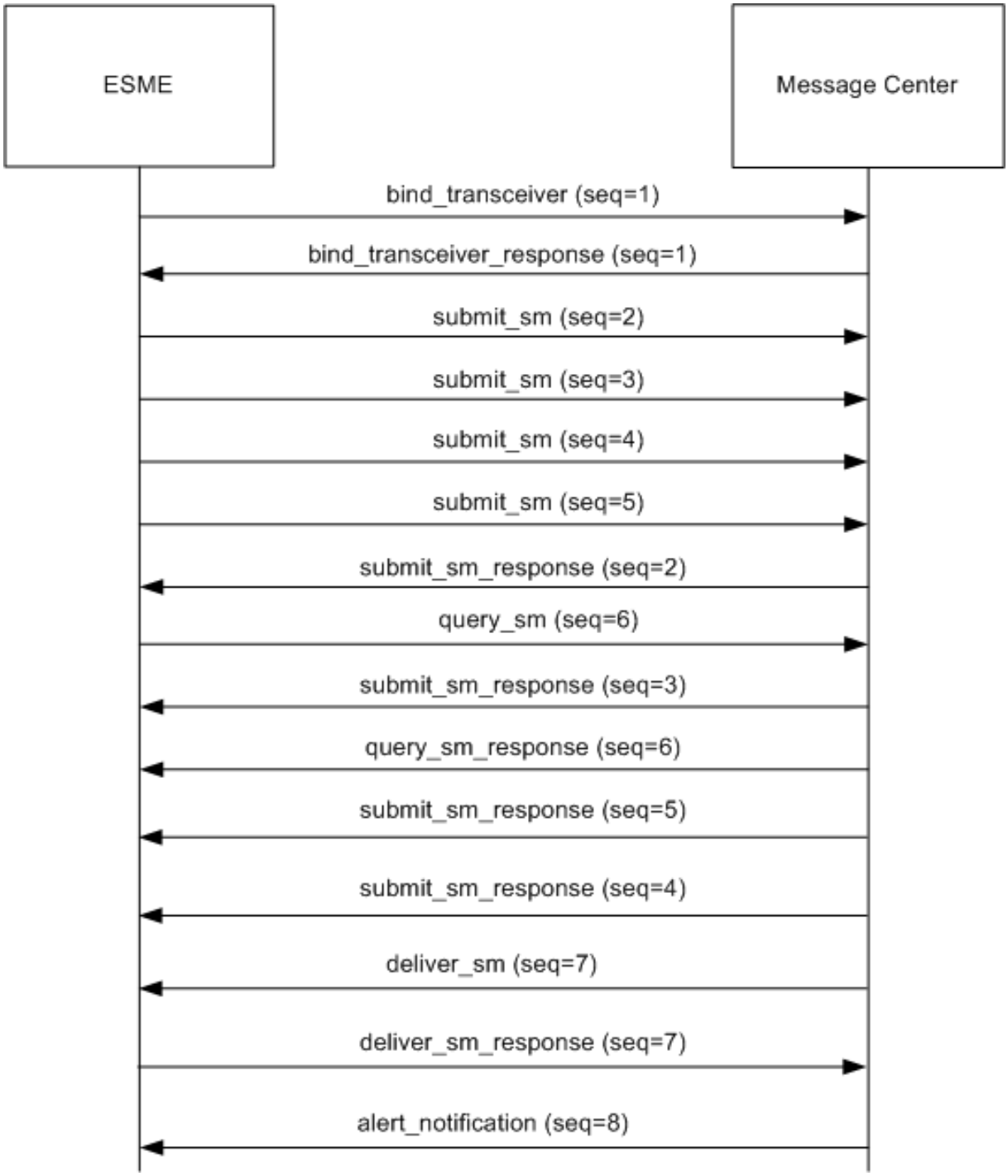
接続が確立されると、ESME は MC にバインドされているモードに応じて、submit_sm メッセージまたは data_sm メッセージの送信、deliver_sm メッセージまたは data_sm メッセージの受信、またはこれらのタイプのメッセージの送受信を行うことができます。ESME は、query_sm、replace_sm、cancel_sm などの補助メッセージを送信して、以前のメッセージ配信の状態を照会したり、以前のメッセージを新しいメッセージに置き換えたり、未配信メッセージをキャンセルしたりすることもできます。

ESME が使用できないか、モバイルユーザがオンラインでないためにメッセージが配信されない場合、メッセージはキューに入れられます。その後、MC はモバイルサブスクライバが到達可能になったことを検出すると、レシーバま

たはトランシーバセッションを介して alert_notification PDU を ESME に送信し、キューに入れられたメッセージの配信を要求します。

各要求 PDU には一意のシーケンス番号があります。応答 PDU のシーケンス番号は、元の要求と同じになります。SMPP を介したメッセージ交換は非同期モードになる可能性があるため、ESME または MC は一度に複数の要求を送信できます。シーケンス番号は、同じ SMPP セッションで応答を返す上で重要な役割を果たします。つまり、シーケンス番号によって、要求と応答のマッチングが可能になります。

次の図は、ESME がトランシーバとしてバインドされるときに、トラフィックフローがさまざまな PDU を使用する方法を示しています。



制限事項:

Citrix ADC アプライアンスはアウトバウンド操作をサポートしていません。つまり、メッセージセンターは、Citrix ADC アプライアンスを介して ESME との SMPP セッションを開始できません。

Citrix ADC での **SMPP** ロードバランシングのしくみ

ESME（クライアント）は、Citrix ADC への接続を開くためにバインドメッセージを送信します。ADC は各 ESME を認証し、成功すると適切なメッセージで応答します。Citrix ADC は、各メッセージセンターとの接続を確立し、これらのメッセージセンター間ですべてのメッセージを負荷分散します。ADC はクライアントからメッセージを受信すると、メッセージ・センターへのオープン・コネクションを再利用するか、オープン・コネクションが使用できない場合は、メッセージ・センターにバインド要求を送信します。

ADC は、クライアントとサーバから発信されたメッセージをロード・バランシングできます。メッセージセンターの健全性を監視し、連結されたメッセージを処理できます。また、メッセージセンターのコンテンツスイッチングサポートも提供します。

ESME から発信されるメッセージ

各 ESME は、認証のために Citrix ADC にユーザーとして追加する必要があります。クライアントは、バインド要求を送信することによって、ADC で設定された SMPP 仮想サーバと TCP 接続を確立します。ADC はクライアントを認証し、成功するとバインドメッセージを解析します。その後、ADC は、設定されたロード・バランシング方式で選択されたメッセージ・センターに要求を送信します。メッセージ・センターへの接続を再利用できない場合、ADC はメッセージ・センターに新しいバインド要求を送信することで、メッセージ・センターとの TCP 接続を開きます。

応答（submit_sm_resp または data_sm_resp）をメッセージ・センターからクライアントに転送する前に、ADC はカスタム・サーバー ID をメッセージ ID に追加して、クライアントによるメッセージの照会、置換、キャンセル要求などの補助操作のためのメッセージ・センターを識別します。他のクライアントからの要求も同様に負荷分散されます。

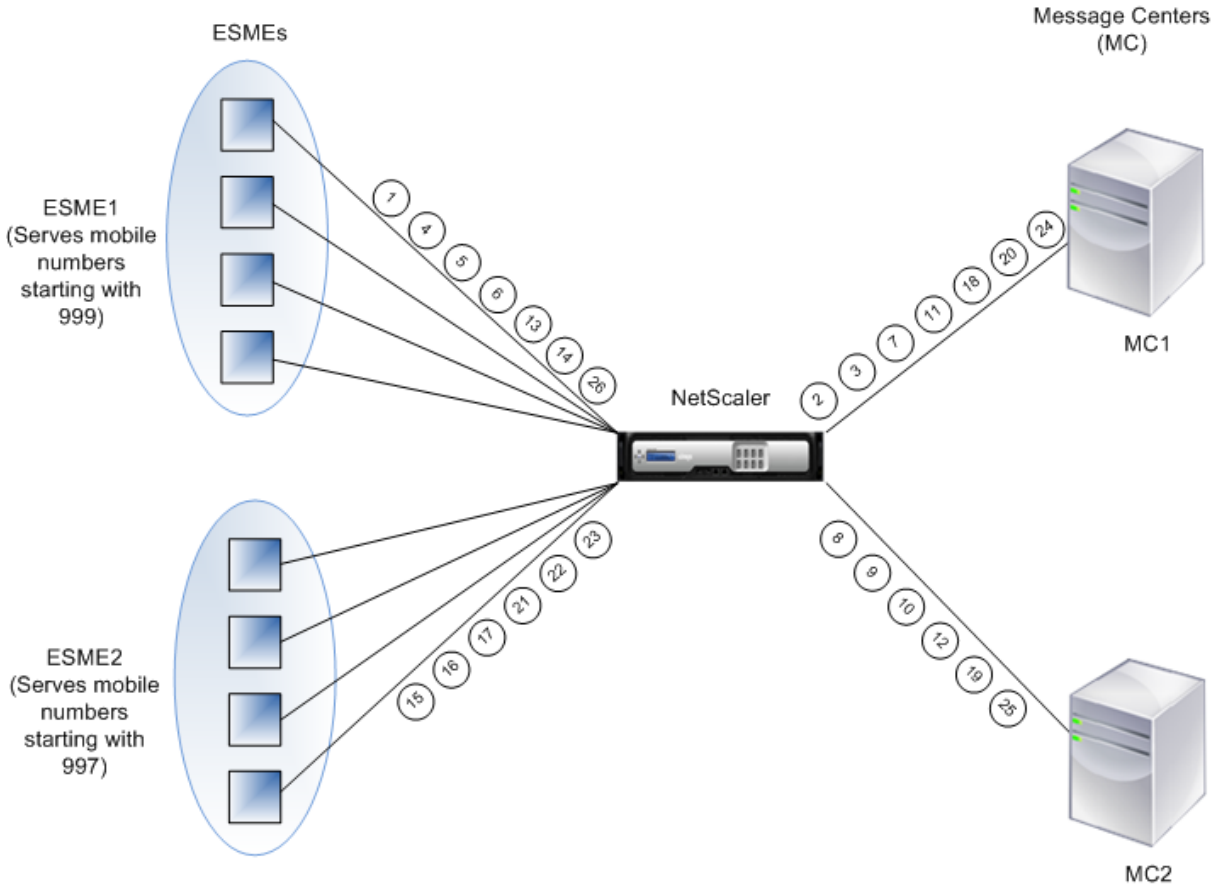
元のバインド要求では、クライアントが処理できるアドレス範囲を指定します。この範囲は、メッセージセンターからクライアントに deliver_sm または data_sm メッセージを転送するために使用されます。

メッセージセンターから発信されるメッセージ

特定のアドレス範囲を処理できる ESME は、クラスタにグループ化されます。クラスター内のすべてのノードが同じ認証情報を提供します。クラスタ内では、ロードバランシングにはラウンドロビン方式のみが使用されます。モバイル発信（MO）メッセージを配信するために、メッセージセンターは Citrix ADC に配信 _sm メッセージを送信します。宛先アドレス範囲（998 で始まる番号など）に対応できるクラスタが ADC にバインドされている場合、そのクラスタを選択し、そのクラスタ内の ESME ノード間でメッセージをロードバランシングします。

アドレス範囲の deliver_sm メッセージを提供できる ESME が ADC にバインドされておらず、メッセージキューイングが有効の場合、そのクライアントがレシーバモードまたはトランシーバモードで ADC にバインドするまで、メッセージはキューイングされます。キューのサイズを指定できます。

次の図は、ESME、Citrix ADC、およびメッセージセンター間の PDU の内部フローを示しています。簡単にするために、2 つの ESME と 2 つのメッセージセンターのみが表示されます。



メッセージ（PDU）のフロー：

1. ESME1 がバインド要求を NetScaler に送信する
2. NetScaler がバインド要求を MC1 に送信する
3. MC1 がバインド応答を NetScaler に送信する
4. NetScaler がバインド応答を ESME1 に送信する
5. ESME1 が submit_sm(1) を NetScaler に送信する
6. ESME1 が submit_sm(2) を NetScaler に送信する
7. NetScaler が submit_sm(1) を MC1 に転送する
8. NetScaler がバインド要求を MC2 に送信する
9. MC2 がバインド応答を NetScaler に送信する
10. NetScaler が submit_sm(2) を MC2 に転送する
11. MC1 が submit_sm(1) を NetScaler に送信する
12. MC2 が submit_sm(2) を NetScaler に送信する
13. NetScaler が submit_sm_resp(1) を ESME1 に転送する
14. NetScaler が submit_sm_resp(2) を ESME1 に転送する
15. ESME2 がバインド要求を NetScaler に送信する
16. NetScaler がバインド応答を ESME2 に送信する
17. ESME2 が submit_sm(3) を NetScaler に送信する

18. NetScaler が submit_sm(3) を MC1 に転送する
19. MC2 が deliver_sm を NetScaler に送信する（ESME2 はメッセージで指定されたアドレス範囲を提供します）
20. MC1 が submit_sm_resp(3) を NetScaler に送信する
21. NetScaler が submit_sm_resp(3) を ESME2 に転送する
22. NetScaler が deliver_sm を ESME2 に転送する
23. ESME2 が deliver_sm_resp を NetScaler に送信する
24. MC1 が alert_notification を NetScaler er に送信する（ESME1 はメッセージで指定されたアドレス範囲に対応します）
25. NetScaler が deliver_sm_resp を MC2 に転送する
26. NetScaler が alert_notification を ESME1 に転送する

メッセージ・センターのヘルス・モニタリング

デフォルトでは、TCP_default モニタは SMPP サービスにバインドされますが、SMPP タイプのカスタムモニタをバインドできます。カスタムモニタは、メッセージセンターへの TCP 接続を開き、enquire_link パケットを送信します。プローブの成功または失敗に応じて、サービスは UP または DOWN とマークされます。

メッセージセンターでのコンテンツの切り替え

メッセージセンターは、ESME からの複数の接続（またはバインド要求）を受け付けることができます。SMPP バインドパラメータに基づいて、これらの要求をコンテンツスイッチするように Citrix ADC を構成できます。メッセージセンターを選択するためのメソッドを設定するための一般的な式を次に示します。

- アドレス範囲に基づく： 次のサンプル式では、アドレス範囲が 988 から始まる場合、ADC は特定のメッセージ・センターを選択します。

例:

SMPP.BINDINFO.ADDRESS_RANGE.CONTAINS(“^988”)

- ESME ID に基づく: 次のサンプル式では、ESME ID が ESME1 の場合、ADC は特定のメッセージセンターを選択します。

例:

SMPP.BINDINFO.SYSTEM_ID.EQ(“ESME1”)

- ESME タイプに基づく： 次のサンプル式では、ESME タイプが VMS の場合、ADC は特定のメッセージ・センターを選択します。VMS は、ボイスメールシステムの略です。

例:

SMPP.BINDINFO.SYSTEM_TYPE.EQ(“VMS”)

- ESME の番号タイプ（TON）に基づく： 次のサンプル式では、TON が 1 の場合（1 は国際番号を表します）、ADC は特定のメッセージ・センターを選択します。

例**:**

SMPP.BINDINFO.ADDR_TON.EQ (1)

- ESME のナンバープランインジケータ（NPI）に基づく: 次のサンプル式では、NPI が 0 (0 は不明な接続を表します) の場合、ADC は特定のメッセージセンターを選択します。

例**:**

SMPP.BINDINFO.ADDR_NPI.EQ(0)

- バインドタイプに基づく: 次のサンプル式では、バインドタイプが TRANSCEIVER の場合、ADC は特定のメッセージセンターを選択します。（トランシーバはメッセージを送受信できます）。

例**:**

SMPP.BINDINFO.TYPE.EQ(TRANSCEIVER)

連結メッセージ処理

SMS は最大 140 バイトを保持できます。長いメッセージは、小さな部分に分割する必要があります。宛先モバイルが対応している場合、メッセージは結合され、1 つの長い SMS として配信されます。Citrix ADC は、メッセージのフラグメントを同じメッセージセンターに転送します。各メッセージには、参照番号、シーケンス番号、およびフラグメントの総数が含まれます。参照番号は、長いメッセージの各フラグメントで同じです。シーケンス番号は、メッセージ全体における特定のフラグメントの位置を指定します。すべてのフラグメントが受信された後、ESME は、1 つの長いメッセージにフラグメントを結合し、モバイル加入者にメッセージを配信します。

クライアントがアクティブな接続から切断しても、メッセージセンターへの接続は閉じられません。これは、他のクライアントからの要求のために再利用されます。

制限事項

メッセージセンターからの 59 バイトを超えるメッセージ ID はサポートされていません。メッセージセンターから返されたメッセージ ID の長さが 59 バイトを超える場合、補助操作は失敗し、Citrix ADC はエラーメッセージを返して応答します。

Citrix ADC での **SMPP** ロードバランシングの設定

ADC で SMPP ロードバランシングを設定するには、次の作業を実行します。

1. SMPP ユーザを追加します。ADC は、ユーザーからのバインド要求を受け入れる前に、ユーザーを認証します。通常、ユーザーは ESME です。
2. プロトコルを SMPP として指定して、ロードバランシング仮想サーバを追加します。
3. プロトコルを SMPP として指定し、サーバごとに一意のカスタムサーバ ID を指定して、サービスを追加します。以前に作成した負荷分散仮想サーバーにサービスをバインドします。
4. 必要に応じて、サービスグループを作成し、サービスグループにサービスを追加します。

5. オプションで、SMPP-ECV タイプのモニタを追加し、サービスにバインドします。TCP デフォルトモニタは、デフォルトでバインドされています。
6. クライアントモードやメッセージキューなどの SMPP パラメータを設定します。

コマンドラインを使用して **SMPP** ロードバランシングを設定するには

コマンドプロンプトで入力します。

```
add smpp user <username> -password <password>
add service <name> <IP> SMPP <port>  - customserverID <customserverID>
add lb vserver <name> <IP> SMPP <port>
bind lb vserver <name> <service name>
set smpp param
<!--NeedCopy-->
```

例

```
add smpp user smppclient1 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add smpp user smppclient2 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add service smmpsvc 10.102.84.140 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID ab -CKA NO -TCPB NO -CMP NO
add service smmpsvc2 10.102.81.175 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID xy -CKA NO -TCPB NO -CMP NO
add lb vserver smppvs SMPP 10.102.239.179 2775 -persistenceType NONE -
    cltTimeout 180
bind lb vserver smppvs smmpsvc2
bind lb vserver smppvs smmpsvc
set smpp param -addrrange ”d*”
<!--NeedCopy-->
```

構成ユーティリティを使用して **SMPP** ロードバランシングを設定するには

1. [システム] **>** [ユーザ管理] **>** [**SMPP** ユーザ] に移動し、SMPP ユーザを追加します。
2. [トラフィック管理] **>** [負荷分散] **>** [**SMPP** パラメータの設定] に移動し、展開で必要に応じてパラメータを設定します。

3. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、タイプ SMPP の仮想サーバーを追加します。
4. [Service] セクションをクリックし、SMPP タイプのサービスを追加して、サーバ ID を指定します。

ユースケース **2**：**TCP** バイトストリームの名前と値のペアに基づいてルールベースの永続性を構成する

October 7, 2021

プロトコルによっては、名前/値のペアを TCP バイトストリームで送信するものもあります。この例の TCP バイトストリームのプロトコルは、財務情報交換 (FIX) プロトコルです。XML 以外の実装では、FIX プロトコルにより、ネットワークを介して通信する 2 つのホストが、名前と値のペア（「FIX フィールド」と呼ばれる）のリストとしてビジネスまたは取引関連の情報を交換できます。フィールドの形式は<tag>=<value><delimiter>です。この従来のタグ値形式により、FIX プロトコルはユースケースに最適です。

FIX フィールドのタグは、フィールドの意味を示す数値識別子です。この例では、

- タグ 35 はメッセージタイプを示します。
- 等号の後の値は、指定されたタグの特定の意味を保持し、データ型に関連付けられています。タグ 35 の値が A の場合は、メッセージがログオンメッセージであることを示します。
- 区切り文字は、印刷されない「ヘッダーの開始」(SOH) ASCII 文字 (0x01) で、キャレット記号 (^) です。
- 各フィールドには名前も割り当てられます。タグ 35 のフィールドは MsgType フィールドです。

ログオンメッセージの例を次に示します。

8=FIX.4.1 9=61 35=A 49=INVMGR 56=BRKR 34=1 52=20000426-12:05:06 98=0 108=30 10=157

上記のようなタグ値リストのパーシステンスタイプの選択は、リストから特定の文字列を抽出するためのオプションによって決まります。トークンベースの永続性メソッドでは、ペイロードから抽出するトークンのオフセットと長さを指定する必要があります。FIX プロトコルでは、特定のフィールドのオフセットとその値の長さがメッセージによって異なる可能性があるため、これを行うことはできません。この変化は、メッセージタイプ、前のフィールド、および前の値の長さによって異なります。また、カスタムフィールドが定義されているかどうかによって、実装によっても異なります。このようなバリエーションにより、特定のフィールドの正確なオフセットを予測したり、トークンとして抽出される値の長さを指定したりすることは不可能になります。この場合、ルールベースの永続性が優先されるパーシステンスタイプです。

仮想サーバー fixlb1 が、Fix 対応アプリケーションのインスタンスをホストしているサーバーのファームへの TCP 接続の負荷分散を行うと仮定します。メッセージを送信する会社を識別する SenderCompid フィールドの値に基づいて、接続の永続性を構成します。この FIX フィールドのタグは 49 です (前述のログオンメッセージの例に示されています)。

ロードバランシング仮想サーバーのルールベースの永続性を構成するには、ロードバランシング仮想サーバーのパーシステンスタイプを RULE に設定し、ルールパラメータに式を設定します。式は、SenderCompID フィールドが見

つかると予想される TCP ペイロードの一部を抽出し、結果の文字列を区切り文字に基づいて名前/値リストに型キャストしてから、次のように SenderCompID フィールド（タグ 49）の値を抽出する式である必要があります。

```
set lb vserver fixlb1 -persistenceType RULE -rule ”CLIENT.TCP.PAYLOAD(300).
TYPECAST_NVLIST_T('=','^').VALUE(”\49\”)”
```

注：これは CLI コマンドであるため、式ではバックスラッシュ文字が使用されています。構成ユーティリティを使用している場合は、バックスラッシュ文字を入力しないでください。

クライアントが、前述のログオンメッセージの例の名前と値のリストを含む FIX メッセージを送信する場合、式は値 INVMGR を抽出し、Citrix ADC アプライアンスはこの値に基づいて永続性セッションを作成します。

PAYLOAD () 関数の引数は、関数によって抽出された文字列に SenderCompID フィールドを含めるために必要な大きさにすることができます。オプションで、フィールドの値を抽出するときにアプライアンスで大文字と小文字を区別する場合は、SET_TEXT_MODE（IGNORECASE）関数を使用し、抽出された値のハッシュに基づいて永続セッションを作成するために HASH 関数を使用できます。次の式では、SET_TEXT_MODE (IGNORECASE) 関数と HASH 関数を使用します。

```
CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T('=','^').SET_TEXT_MODE(IGNORECASE
).VALUE(”49”).HASH
```

次に、FIX 接続の永続性を構成するために使用できる規則の例を示します (<tag>は、値を抽出するフィールドのタグに置き換えます)。

- TCP ペイロードの最初の 300 バイトの FIX フィールドの値を抽出するには、CLIENT.TCP.PAYLOAD(300).BEFORE_STR(“<tag>=”) 式を使用できます。
- オフセット 80 で 20 バイトの長さの文字列を抽出するには、文字列の名前値リストにキャストし、目的のフィールドの値を抽出するには、式 CLIENT.TCP.PAYLOAD(100).SUBSTR(80,20).TYPECAST_NVLIST_T(‘=’,’^’).VALUE(“<tag>“) を使用します。
- TCP ペイロードの最初の 100 バイトを抽出するには、文字列を名前/値リストにキャストし、目的のフィールドの 3 番目の出現の値を抽出するには、式 CLIENT.TCP.PAYLOAD(100).TYPECAST_NVLIST_T(‘=’,’^’).VALUE(“<tag>“,2) を使用します。
  注：
  VALUE () 関数に渡される 2 番目の引数が
  n の場合、アプライアンスはフィールドの
  (n+1)
  th インスタンスの値を抽出します。これは、カウントがゼロ（
  0)。

次に、永続性の設定に使用できる規則の例を示します。FIX プロトコルを介して送信されるデータを評価できるのは、ペイロードベースの式だけです。その他の式は、より低いネットワーキングプロトコルに基づいてパーシステンスを設定するためのより一般的な式です。

- CLIENT.TCP.PAYLOAD(100)
- CLIENT.TCP.PAYLOAD(100).HASH

- CLIENT.TCP.PAYLOAD(100).SUBSTR(5,10)
- CLIENT.TCP.SRCPORT
- CLIENT.TCP.DSTPORT
- CLIENT.IP.SRC
- CLIENT.IP.DST
- CLIENT.IP.SRC.GET4
- CLIENT.IP.DST.GET4
- CLIENT.ETHER.SRCMAC.GET6
- CLIENT.ETHER.DSTMAC.GET5
- CLIENT.VLAN.ID

使用例 **3:** ダイレクトサーバーリターンモードで負荷分散を構成する

October 7, 2021

ダイレクトサーバーリターン（DSR）モードでの負荷分散により、サーバーは Citrix ADC アプライアンスを経由しないリターンパスを使用して、クライアントに直接応答できます。ただし、DSR モードでは、アプライアンスはサービスのヘルスチェックを継続して実行できます。大量のデータ・ボリューム環境では、DSR モードでクライアントに直接サーバ・トラフィックを送信すると、パケットがアプライアンスを通過しないため、アプライアンスの全体的なパケット処理能力が増加します。

DSR モードには、次の機能と制限があります。

- これは、ワンアームモードとインラインモードをサポートしています。
- アプライアンスは、アイドルタイムアウトに基づいてセッションを期限切れにします。
- アプライアンスは TCP 接続をプロキシしない（つまり、クライアントに SYN-ACK を送信しない）ため、SYN 攻撃をシャットダウンしません。SYN パケットレートフィルタを使用すると、サーバに対する SYN のレートを制御できます。SYN のレートを制御するには、SYN のレートのしきい値を設定します。SYN 攻撃から保護するには、TCP 接続をプロキシするようにアプライアンスを設定する必要があります。ただし、逆方向のトラフィックがアプライアンスを通過する必要があります。
- DSR 構成では、Citrix ADC アプライアンスは、負荷分散仮想サーバーの IP アドレスを宛先サーバーの IP アドレスに置き換えることはありません。代わりに、サーバーの MAC アドレスを使用してパケットをサービスに転送します。VIP をサーバに設定し、サーバに設定されている VIP に対して ARP を無効にする必要があります。これにより、クライアント要求がワンアームモードに設定されている場合、アプライアンスがバイパスされなくなります。たとえば、ユーザはループバックインターフェイスで VIP を設定し、同じ VIP の ARP を無効にする必要があります。
- アプライアンスは、サービスにバインドされたモニターからサーバーの MAC アドレスを取得します。ただし、Citrix ADC アプライアンスに保存されたスクリプトを使用するカスタムユーザーモニター（「USER」タイプのモニター）では、サーバーの MAC アドレスは認識されません。DSR 構成でカスタム・モニターのみを使用する場合、仮想サーバーが受信する要求ごとに、アプライアンスは（ARP 要求を送信して）宛先 IP アドレス

を MAC アドレスに解決しようとします。宛先 IP アドレスは Citrix ADC アプライアンスが所有する仮想 IP アドレスであるため、ARP 要求は常に Citrix ADC インターフェイスの MAC アドレスに解決されます。したがって、仮想サーバが受信したすべてのトラフィックは、アプライアンスにループバックされます。DSR 構成でユーザモニタを使用する場合は、サービス用に別のタイプのモニタ（PING モニタなど）も設定する必要があります。理想的には、プローブの間隔を長くして、サーバの MAC アドレスを学習できるようにします。

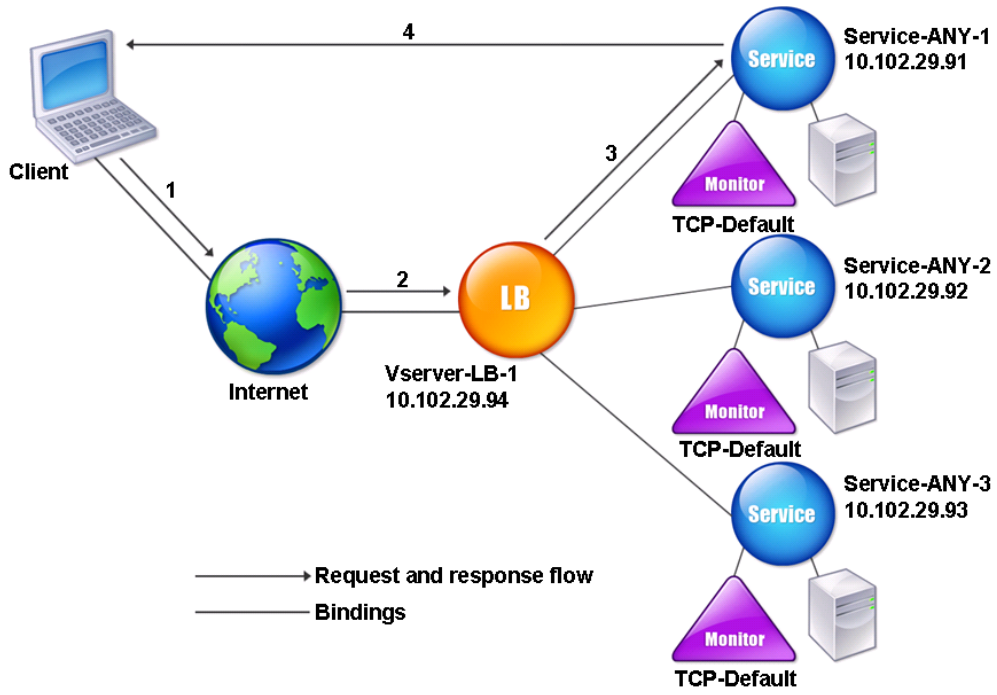
- Citrix ADC アプライアンスは、サービスにバインドされているモニターからサーバー L2 パラメーターを学習します。UDP-ECV モニタの場合は、受信文字列を設定して、アプライアンスがサーバの L2 パラメータを学習できるようにします。受信ストリングが設定されておらず、サーバが応答しない場合、アプライアンスは L2 パラメータを学習しませんが、サービスは UP に設定されます。このサービスのトラフィックはブラックホールです。

この例のシナリオでは、Service-ANY-1、Service-ANY-2、Service-ANY-3 が作成され、仮想サーバ Vserver-LB-1 にバインドされます。仮想サーバーはクライアント要求をサービスに対して負荷分散し、サービスは Citrix ADC アプライアンスをバイパスしてクライアントに直接応答します。次の表に、Citrix ADC アプライアンス上で DSR モードで構成されたエンティティの名前と値を示します。

エンティティタイプ	名前	IP アドレス	プロトコル
仮想サーバ	Vserver-LB-1	10.102.29.94	ANY
サービス	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
モニター	TCP	なし	なし

次の図は、アプライアンスで設定するパラメータの負荷分散エンティティと値を示しています。

図 1: DSR モデルにおける負荷分散のエンティティモデル



アプライアンスが DSR モードで正常に機能するには、クライアント要求の宛先 IP を変更する必要があります。代わりに、アプライアンスは宛先 MAC を選択したサーバーの MAC に変更します。この設定では、サーバーはサーバーをバイパスしながら、クライアントへの要求を転送するためのクライアント MAC アドレスを決定できます。

次に、基本負荷分散の設定の説明に従って、基本的な負荷分散設定を構成し、前の表で説明した値を使用してエンティティに名前を付けて、パラメータを設定します。

基本的なロード・バランシング設定を構成したら、DSR モード用にカスタマイズする必要があります。これを行うには、セッションレス仮想サーバーで Source IP ハッシュメソッドなど、サポートされている負荷分散メソッドを構成します。また、応答を転送するためのクライアント MAC アドレスをサーバが決定し、アプライアンスをバイパスするように、リダイレクションモードを設定する必要があります。

負荷分散方式とリダイレクションモードを構成したら、各サービスで USIP モードを有効にする必要があります。その後、サービスは応答を転送するときに送信元 IP アドレスを使用します。

コマンドラインインターフェイスを使用して、セッションレス仮想サーバーの負荷分散方法とリダイレクションモードを構成するには

コマンドプロンプトで入力します。


```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

例

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless
    enabled
<!--NeedCopy-->
```

> 注
>
> -m MAC オプションが有効になっている仮想サーバにバインドされたサービスの場合は、非ユーザーモニタをバインドする必要があります。

構成ユーティリティを使用して、セッションレス仮想サーバーの負荷分散方法とリダイレクションモードを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、MAC ベースとしてリダイレクトモードを選択し、SOURCEIPHASH としてメソッドを選択します。
3. 「トラフィックの設定」で、「セッションレス負荷分散」を選択します。

コマンドラインインターフェイスを使用して送信元 **IP** アドレスを使用するようにサービスを構成するには

コマンドプロンプトで入力します。

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```


構成ユーティリティを使用して送信元 **IP** アドレスを使用するようにサービスを構成するには

1. **Traffic Management > Load Balancing > Services** に移動します。
2. サービスを開き、[トラフィック設定] で、[送信元 **IP** アドレスの使用] を選択します。

特定の状況では、以降のセクションで説明する追加の手順が必要です。

ユースケース **4: DSR** モードで **LINUX** サーバーを構成する

October 7, 2021

LINUX オペレーティングシステムでは、DSR クラスタ内の各負荷分散サーバーで、Citrix ADC アプライアンスの仮想 IP アドレス（VIP）を使用してループバックインターフェイスを設定する必要があります。

DSR モードで **LINUX** サーバーを構成するには

負荷分散された各サーバー上の Citrix ADC アプライアンスの VIP とのループバックインターフェイスを作成するには、Linux OS プロンプトで次のコマンドを入力します。

```
ifconfig dummy0 up

ifconfig dummy0:0 inet <netscaler vip> netmask 255.255.255.255 up

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce
<!--NeedCopy-->
```

次に、TOS ID を VIP に再マップするソフトウェアを実行します。

注: ソフトウェアを実行する前に、正しいマッピングをソフトウェアに追加してください。上記のコマンドでは、LINUX サーバーは dummy0 を使用してネットワークに接続します。このコマンドを使用する場合は、LINUX サーバーがネットワークに接続するために使用するインターフェイスの名前を入力します。

使用例 **5: TOS** 使用時に **DSR** モードを設定する

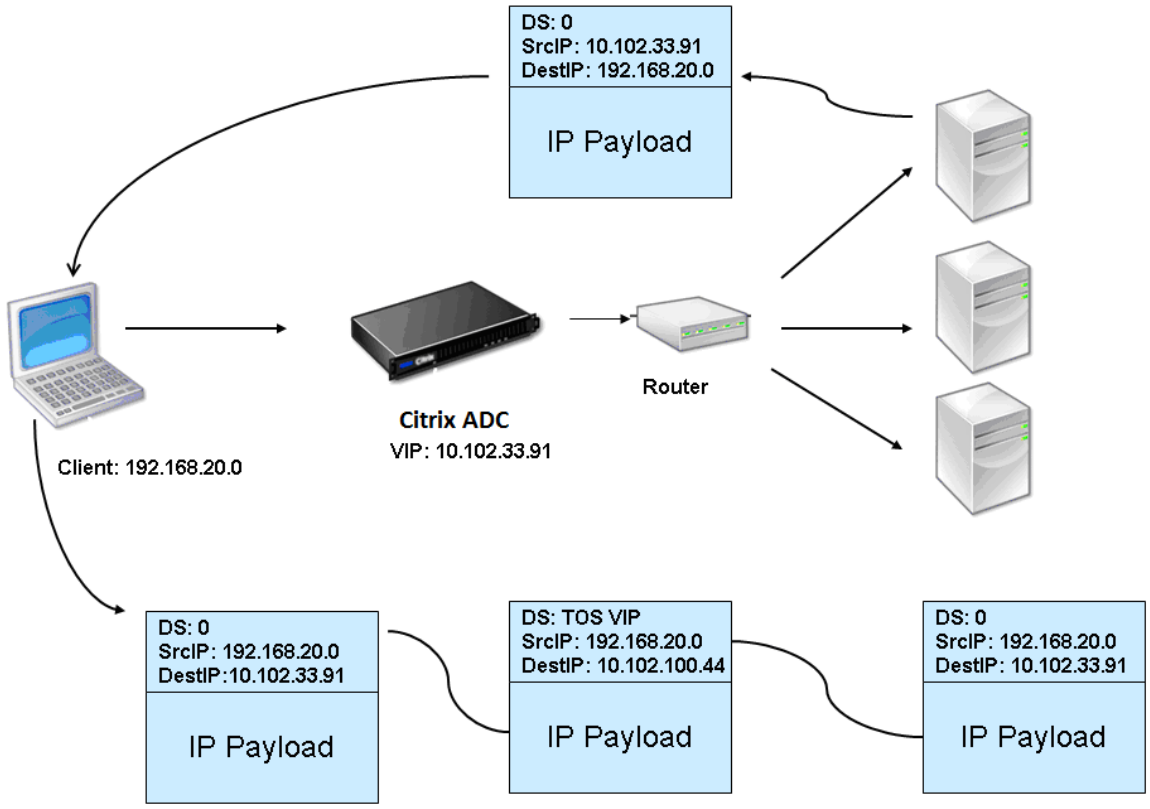
October 7, 2021

TOS（Type of Service）とも呼ばれる Differentiated Services（DS）は、IPv4 パケットヘッダーの一部であるフィールドです。IPv6 ヘッダーの同等のフィールドはトラフィッククラスです。TOS は、パケットのパスを最適化

するために上位層プロトコルによって使用されます。TOS 情報では、Citrix ADC アプライアンスの仮想 IP アドレス (VIP) がエンコードされ、負荷分散されたサーバーによって VIP が抽出されます。

次のシナリオでは、アプライアンスはパケットの **TOS** フィールドに VIP を追加してから、パケットを負荷分散サーバーに転送します。次に、負荷分散されたサーバーは、次の図に示すように、アプライアンスをバイパスして、クライアントに直接応答します。

図 1: DSR モードの Citrix ADC アプライアンス（TOS 付き）



TOS 機能は、制御された環境向けに次のようにカスタマイズされています。

- この環境には、ステートフルファイアウォールや TCP ゲートウェイなどのステートフルデバイスが、アプライアンスとロードバランシングされたサーバ間のパスに存在しないようにする必要があります。
- ネットワークへのすべてのエントリポイントにあるルータは、ロードバランシングされたサーバが別の TOS フィールドとアプライアンスによって追加された TOS フィールドを混同しないようにするために、すべての着信パケットから TOS フィールドを削除する必要があります。
- 各サーバーには 63 個の VIP しか設定できません。
- 中間ルータは、フラグメンテーションに関する ICMP エラーメッセージを送信しないでください。送信元 IP アドレスは負荷分散サーバーの IP アドレスであり、Citrix ADC VIP ではないため、クライアントはメッセージを理解しません。
- TOS は、IP ベースのサービスに対してのみ有効です。TOS ではドメイン名ベースのサービスを使用できませ

ん。

この例では、Service-ANY-1 が作成され、仮想サーバー Vserver-LB-1 にバインドされます。仮想サーバは、サービスに対するクライアント要求の負荷を分散し、アプライアンスをバイパスしてクライアントが直接応答します。次の表に、DSR モードでアプライアンス上で構成されたエンティティの名前と値を示します。

エンティティの種類	名前	IP アドレス	プロトコル
仮想サーバ	Vserver-LB-1	10.102.33.91	ANY
サービス	Service-ANY-1	10.102.100.44	ANY
モニター	PING	なし	なし

TOS を使用した DSR では、レイヤー 3 で負荷分散を設定する必要があります。レイヤ 3 の基本的なロードバランシング設定を構成するには、基本ロードバランシングの設定を参照してください。エンティティに名前を付け、前の表で説明した値を使用してパラメータを設定します。

負荷分散の設定を構成したら、DSR モードの負荷分散設定をカスタマイズする必要があります。このためには、リダイレクションモードを設定して、サーバーがデータパケットをカプセル化解除し、クライアントに直接応答してアプライアンスをバイパスできるようにします。

リダイレクト・モードを指定した後で、オプションでアプライアンスを有効にしてサーバーを透過的に監視できます。これにより、アプライアンスは負荷分散されたサーバーを透過的に監視できます。

コマンドラインインターフェイスを使用して仮想サーバーのリダイレクションモードを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <vServerName> -m <Value> -tosId <Value>
<!--NeedCopy-->
```

例:

```
set lb vserver Vserver-LB-1 -m TOS -tosId 3
<!--NeedCopy-->
```

構成ユーティリティを使用して仮想サーバーのリダイレクションモードを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを開き、リダイレクトモードで TOS ID を選択します。

コマンドラインインターフェイスを使用して **TOS** のトランスペアレントモニタを設定するには

コマンドプロンプトで入力します。

```
add monitor <MonitorName> <Type> -destip <DestinationIP> -tos <Value> -
    tosId <Value>
<!--NeedCopy-->
```

例**:**

```
add monitor mon1 PING -destip 10.102.33.91 -tos Yes  -tosId 3
<!--NeedCopy-->
```

設定ユーティリティを使用して **TOS** 用のトランスペアレントモニタを作成するには

1. **Traffic Management > Load Balancing > Monitors** に移動します。
2. モニタを作成し、[TOS] を選択し、仮想サーバに対して指定した TOS ID を入力します。

ワイルドカード **TOS** モニタ

TOS フィールドを使用した DSR モードのロードバランシング設定では、サービスをモニタリングするには、TOS モニタを作成し、これらのサービスにバインドする必要があります。TOS モニタでは、VIP アドレスのエンコード値を作成するために VIP アドレスと TOS ID が必要になるため、[TOS] フィールドを使用して DSR モードのロードバランシング設定ごとに個別の TOS モニタが必要です。モニターは、**TOS** フィールドが VIP アドレスのエンコードされた値に設定されているプローブパケットを作成します。次に、ロードバランシング設定のサービスによって表されるサーバにプローブパケットを送信します。

多くの負荷分散構成では、構成ごとに個別のカスタム TOS モニターを作成することは、重要で面倒な作業です。これらの TOS モニターの管理も重要なタスクです。これで、ワイルドカード TOS モニタを作成できます。同じプロトコル（TCP や UDP など）を使用するすべての負荷分散構成に対して、ワイルドカード TOS モニターを 1 つだけ作成します。

ワイルドカード TOS モニタには、次の必須設定があります。

- タイプ =<protocol>
- TOS = はい

次のパラメータは、値に設定することも、空白のままにすることもできます。

- 接続先 IP
- 宛先ポート
- TOS ID

DSR サービスにバインドされたワイルドカード TOS モニタ（宛先 IP、宛先ポート、および TOS ID が設定されていない）は、ロードバランシング仮想サーバの TOS ID および VIP アドレスを自動的に学習します。モニタは、エンコードされた VIP アドレスに設定された TOS フィールドを使用してプローブパケットを作成し、DSR サービスによって表されるサーバにプローブパケットを送信します。

CLI を使用してワイルドカード **TOS** モニタを作成するには

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <Type> -tos YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

CLI を使用してワイルドカード **TOS** モニタをサービスにバインドするには

コマンドプロンプトで入力します。

```
bind lb monitor <monitorName> <serviceName>

show lb monitor <monitorName>
<!--NeedCopy-->
```

GUI を使用してワイルドカード **TOS** モニタを作成するには

1. [トラフィック管理] > [負荷分散] > [モニタ] に移動します。
2. 次のパラメータ設定でモニタを追加します。
   - タイプ =<protocol>
   - TOS = はい

GUI を使用してワイルドカード **TOS** モニタをサービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. サービスをオープンし、ワイルドカード TOS モニタをそれにバインドします。

次の設定例では、V1、V2、V3 はタイプ ANY の仮想サーバのロードバランシングを行い、TOS ID はそれぞれ 1、2、3 に設定されています。S1、S2、S3、S4、および S5 は、タイプ ANY のサービスです。S1 と S2 は、V1 と V2 の両方にバインドされています。S3、S4、および S5 であり、V1 と V3 の両方にバインドされています。WLCD-TOS-MON は、タイプ TCP のワイルドカード TOS モニターであり、S1、S2、S3、S4、および S5 にバインドされています。

WLCD-TOS-MON は、S1、S2、S3、S4、および S5 にバインドされている仮想サーバの TOD ID および VIP アドレスを自動的に学習します。

S1 は V1 と V2 にバインドされているため、WLCD-TOS-MON は S1 の 2 種類のプローブパケットを作成します。1 つは、**TOS** フィールドが V1 のエンコードされた VIP アドレス（203.0.113.1）に設定され、もう 1 つは V2 の VIP アドレス（203.0.113.2）です。その後、Citrix ADC はこれらのプローブパケットを S1 で表されるサーバーに送信します。同様に、WLCD-TOS-MON は、S2、S3、S4、および S5 のプローブパケットを作成します。

```
add lb monitor WLCD-TOS-MON  TCP -tos YES

Done

add lb vserver V1 ANY 203.0.113.1 * -m TOS  - tosID 1

Done

add lb vserver V2 ANY 203.0.113.2 * -m TOS  - tosID 2

Done

add lb vserver V3 ANY 203.0.113.3 * -m TOS  - tosID 3

Done

add service S1 198.51.100.1 ANY *

Done

add service S2 198.51.100.2 ANY *

Done

add service S3 198.51.100.3 ANY *

Done

add service S4 198.51.100.4 ANY *

Done

add service S5 198.51.100.5 ANY *

Done
```



```
36
37 bind lb monitor WLCD-TOS-MON S1
38
39 Done
40
41 bind lb monitor WLCD-TOS-MON S2
42
43 Done
44
45 bind lb monitor WLCD-TOS-MON S3
46
47 Done
48
49 bind lb monitor WLCD-TOS-MON S4
50
51 Done
52
53 bind lb monitor WLCD-TOS-MON S5
54
55 Done
56
57 bind lb vserver V1 S1, S2, S3, S4, S5
58
59 Done
60
61 bind lb vserver V2, S1, S2
62
63 Done
64
65 bind lb vserver V3 S3, S4, S5
66
67 Done
68 <!--NeedCopy-->
```

使用例 **6: TOS** フィールドを使用して **IPv6** ネットワークの **DSR** モードで負荷分散を構成する

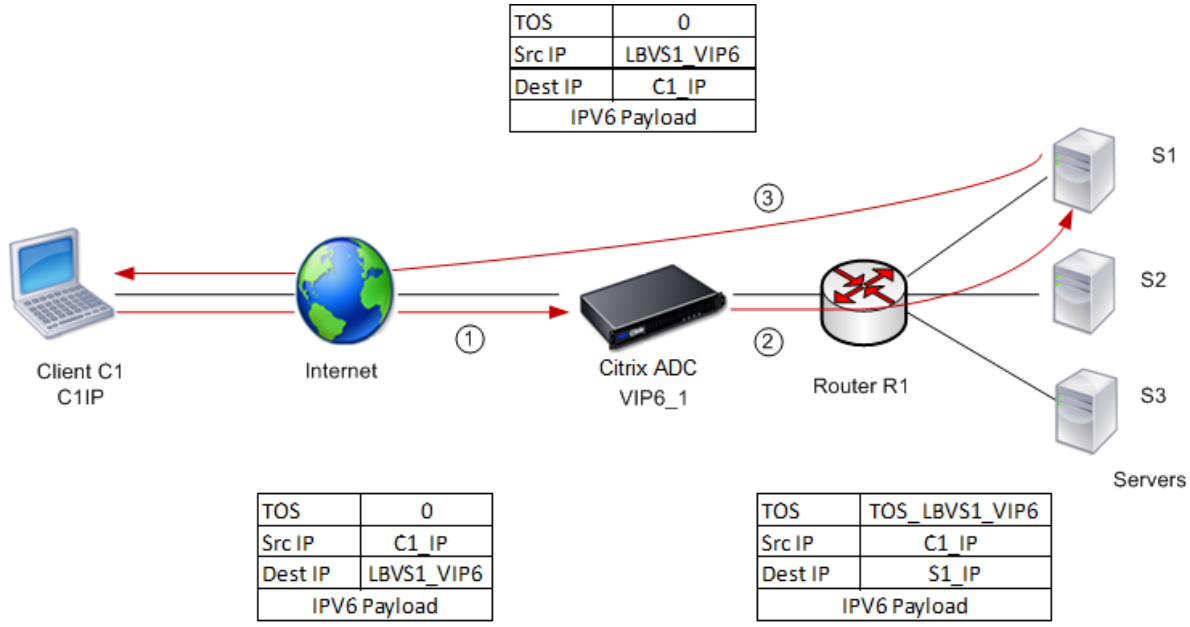
October 7, 2021

Citrix ADC アプライアンスとサーバーが異なるネットワークにある場合は、サービスの種類（TOS）フィールドを使用して、IPv6 ネットワークのダイレクトサーバー復帰（DSR）モードで負荷分散を構成できます。

注**:** TOS フィールドは、トラフィッククラスフィールドとも呼ばれます。

DSR モードでは、クライアントが Citrix ADC アプライアンス上の VIP6 アドレスに要求を送信すると、アプライアンスはパケットの宛先 IPv6 アドレスをサーバーの IPv6 アドレスに変更することでこの要求をサーバーに転送し、TOS（トラフィッククラスとも呼ばれます）に VIP6 アドレスのエンコード値を設定します。フィールドに IPv6 ヘッダーの値を指定します。TOS フィールドの情報を使用して、符号化された値から VIP6 アドレスを取得するようにサーバを設定できます。このアドレスは、応答パケットで送信元 IP アドレスとして使用されます。応答トラフィックは、アプライアンスをバイパスして、クライアントに直接送信されます。

Citrix ADC アプライアンス NS1 で構成された負荷分散仮想サーバー LBVS1 を使用して、サーバー S1、S2、S3 間でトラフィックの負荷分散を行う例を考えてみましょう。Citrix ADC アプライアンス NS1 とサーバー S1、S2、S3 は異なるネットワークにあるため、ルーター R1 は NS1 とサーバーの間に展開されます。



次の表に、この例で使用される設定を示します。

エンティティ	名前
クライアント C1 の IPv6 アドレス	C1_IP (参照目的のみ)
NS1 上の仮想サーバのロード・バランシング	LBVS1
LBVS1 の IPv6 アドレス	LBVS1_VIP6 (参照目的のみ)
TOS の値	TOS_LBVS1_VIP6 （参照のみを目的としています）
NS1 上のサーバ S1 のサービス	SVC_S1
サーバ S1 の IPv6 アドレス	S1_IP (参照目的のみ)
NS1 上のサーバ S2 のサービス	SVC_S2
サーバ S1 の IPv6 アドレス	S2_IP (参照目的のみ)

エンティティ	名前
NS1 上のサーバー S3 のサービス	SVC_S3
サーバ S1 の IPv6 アドレス	S3_IP (参照目的のみ)

シナリオ例のトラフィックフローを次に示します。

1. クライアント C1 は、仮想サーバ LBVS1 に要求を送信します。
2. LBVS1 の負荷分散アルゴリズムはサーバー S1 を選択し、アプライアンスは S1 への接続を開きます。NS1 は、次の情報を使用して S1 に要求を送信します。
   - TOS フィールドをに設定 TOS_LBVS1_VIP6.
   - 送信元 IP アドレスを C1_IP として指定します。
3. サーバ S1 は、要求を受信すると、TOS フィールドの情報を使用して LBVS1_VIP6 アドレスを取得します。これは NS1 上の仮想サーバ LBVS1 の IP アドレスです。サーバは、アプライアンスをバイパスして C1 に直接応答を送信します。
   - 送信元 IP アドレスは、派生した LBVS1_VIP6 アドレスに設定され、クライアントがサーバ S1 ではなく NS1 上の仮想サーバ LBVS1 と通信します。

TOS を使用して **DSR** モードでロードバランシングを設定するには、アプライアンスで次の手順を実行します

1. USIP モードをグローバルに有効にします。
2. サーバをサービスとして追加します。
3. ロードバランシング仮想サーバに TOS 値を設定します。
4. サービスを仮想サーバにバインドします。

コマンドラインインターフェイスを使用して **TOS** を使用して **DSR** モードでロードバランシングを設定するには

コマンドプロンプトで入力します。

```
enable ns mode USIP

add service <serviceName> <IP> <serviceType> <port>
<!--NeedCopy-->
```

Citrix ADC アプライアンスに各サーバーをサービスとして追加するには、前述のコマンドを必要な回数だけ繰り返します。


```
add lb vserver <name> <serviceType> <ip> <port> -m <redirectionMode> -
    tosId <positive_integer>

bind lb vserver <vserverName> <serviceName>
<!--NeedCopy-->
```

構成ユーティリティを使用して **USIP** モードを有効にするには

[システム] **>** [設定] **>** [モードの設定] に移動し、[送信元 **IP** アドレスの使用] を選択します。

構成ユーティリティを使用してサービスを作成するには

[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを作成します。

構成ユーティリティを使用して負荷分散仮想サーバーを作成し、サービスをバインドするには

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを作成します。
2. サービスをこの仮想サーバーにバインドするには、［サービス］セクションをクリックします。

使用例 **7: IP** オーバー **IP** を使用して **DSR** モードで負荷分散を構成する

December 7, 2021

IP トンネリング（*IP over* IP 構成とも呼ばれる）を使用して、レイヤー 3 ネットワーク全体でダイレクトサーバーリターン（DSR）モードを使用するように Citrix ADC アプライアンスを構成できます。DSR モードの標準の負荷分散構成と同様に、サーバーは Citrix ADC アプライアンスを経由するリターンパスを使用する代わりに、クライアントに直接応答できます。これにより、応答時間とスループットが向上します。標準の DSR モードと同様に、Citrix ADC アプライアンスはサーバーを監視し、アプリケーションポートでヘルスチェックを実行します。

IP over IP 構成では、Citrix ADC アプライアンスとサーバーが同じレイヤー 2 サブネット上にある必要はありません。代わりに、Citrix ADC アプライアンスはパケットをカプセル化してから宛先サーバーに送信します。宛先サーバはパケットを受信した後、パケットのカプセル化を解除し、応答をクライアントに直接送信します。これはしばしば L3DSR と呼ばれます。

Citrix ADC アプライアンスで L3-DSR モードを構成するには:

- 負荷分散仮想サーバーを作成します。モードを IPTUNNEL に設定し、セッションレストラッキングを有効にします。
- サービスを作成します。バックエンドアプリケーションごとにサービスを作成し、サービスを仮想サーバーにバインドします。

- カプセル化解除を設定します。Citrix ADC アプライアンスまたはバックエンドサーバーのいずれかをカプセル化解除機能として構成します。

  > 注:
  >
  > Citrix ADC アプライアンスを使用する場合、カプセル化解除の設定は、バックエンドが L2DSR を実サーバーに対して行う ADC アプライアンス間の IP トンネルです。

負荷分散仮想サーバーの構成

アプリケーションへの要求を処理するように仮想サーバーを設定します。サービスと一致するサービスタイプを割り当てるか、複数のサービスに対して ANY のタイプを使用します。
転送方式を IPTUNNEL に設定し、仮想サーバがセッションレスモードで動作できるようにします。使用する負荷分散方式を設定します。

コマンドラインインターフェイスを使用して **IP over IP DSR** 用の負荷分散仮想サーバーを作成および構成するには

コマンドプロンプトで次のコマンドを入力して、IP over IP DSR の負荷分散仮想サーバーを構成し、構成を確認します。

```
add lb vserver <name> serviceType <serviceType> IPAddress <ip> Port <port> -lbMethod <method> -m <ipTunnelTag> -sessionless [ENABLED | DISABLED]

show lb vserver <name>
<!--NeedCopy-->
```

例:

次の例では、負荷分散方式を sourceiPhash として選択し、セッションレス負荷分散を設定しています。

```
add lb vserver Vserver-LB-1 ANY 1.1.1.80 * -lbMethod SourceIPHash -m IPTUNNEL -sessionless ENABLED
<!--NeedCopy-->
```

GUI を使用して **IP over IP DSR** の負荷分散仮想サーバーを作成および構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを作成し、[リダイレクションモード] を [**IP** トンネルベース] に指定します。

IP 経由の IP DSR のサービスを構成する

負荷分散サーバーを作成したら、アプリケーションごとに 1 つのサービスを構成します。このサービスは、Citrix ADC アプライアンスからこれらのアプリケーションへのトラフィックを処理し、Citrix ADC アプライアンスが各アプリケーションの正常性を監視できるようにします。

USIP モードを使用するようにサービスを割り当て、トンネルベースのモニタリングのために IPTUNNEL タイプのモニタをサービスにバインドします。

コマンドラインインターフェイスを使用して **IP over IP DSR** 用のサービスを作成および構成するには

コマンドプロンプトで次のコマンドを入力してサービスを作成し、オプションでモニターを作成してサービスにバインドします。

```
add service <serviceName> <serverName> <serviceType> <port> -usip <usip
   >

add monitor <monitorName> <monitorType> -destip <ip> -iptunnel <
   iptunnel>

bind service <serviceName> -monitorName <monitorName>
<!--NeedCopy-->
```

例:

次の例では、タイプ IPTUNNEL のモニタが作成されます。

```
add monitor mon_DSR PING -destip 1.1.1.80 -iptunnel yes
add service svc_DSR01 2.2.2.100 ANY * -usip yes
bind service svc_DSR01 -monitorName mon_DSR
<!--NeedCopy-->
```

サーバと ADC アプライアンスの両方でルーティングを簡素化する別の方法として、ADC とサーバの両方が同じサブネットからの IP を使用するように設定する方法があります。そうすることで、トンネルエンドポイントの宛先を持つすべてのトラフィックがトンネルを介して送信されます。この例では、10.0.1.0/30 が使用されています。

> 注:
>
> モニタの目的は、IP トンネルを介して各サーバのループバックに到達することにより、トンネルがアクティブであることを確認することです。サービスが稼働していない場合は、ADC とサーバ間の外部 IP ルーティングが良好かどうかを確認します。また、内部 IP アドレスが IP トンネルを介して到達可能かどうかも確認します。サーバでルートが必要になる場合や、選択した実装に応じて PBR が ADC に追加されます。

例:

```
add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
add netProfile netProfile_DSR -srcIP 10.0.1.2
add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
   YES -netProfile netProfile_DSR
<!--NeedCopy-->
```

GUI を使用してモニターを構成するには

1. トラフィック管理 **>** 負荷分散 **>** モニターに移動します。
2. モニターを作成し、[**IP** トンネル] を選択します。

GUI を使用して **IP over IP DSR** のサービスを作成および構成するには

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. サービスを作成し、[設定] タブで [送信元 **IP** アドレスを使用] を選択します。

コマンドラインインターフェイスを使用してサービスを負荷分散仮想サーバーにバインドするには

コマンドプロンプトで次のコマンドを入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-DSR-1
<!--NeedCopy-->
```

GUI を使用してサービスを負荷分散仮想サーバーにバインドするには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[サービス] セクションをクリックして、サービスを仮想サーバーにバインドします。

トンネルパケットの **Outer** ヘッダーでのクライアント **IP** アドレスの使用

Citrix ADC は、IP トンネリングを使用したダイレクトサーバーリターンモードに関連するトンネルパケットの外側ヘッダーでクライアントソース IP アドレスを、送信元 IP アドレスとして使用することをサポートしています。この機能は、IPv4 を使用した DSR と IPv6 トンネリングモードを使用した DSR でサポートされます。この機能を有効にするには、IPv4 または IPv6 の **use client** 送信元 **IP** アドレスパラメータを有効にします。この設定は、IP トンネリングを使用するすべての DSR 設定にグローバルに適用されます。

CLI を使用してクライアント**-**送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

コマンドプロンプトで入力します。

- `set iptunnelparam -useclientsourceip [YES | NO]`
- `show iptunnelparam`

GUI を使用してクライアントの送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

1. [システム**] > [**ネットワーク] に移動します。
2. [設定] タブで、[**IPv4** トンネルのグローバル設定**]** をクリックします。
3. [**IPv4** トンネルグローバルパラメータの設定] ページで、[クライアントソース **IP** を使用] チェックボックスをオンにします。
4. [**OK**] をクリックします。

CLI を使用してクライアントの送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

コマンドプロンプトで入力します。

- `set ip6tunnelparam -useclientsourceip [YES | NO]`
- `show ip6tunnelparam`

GUI を使用してクライアントの送信元 **IP** アドレスを送信元 **IP** アドレスとして使用するには

1. [システム**] > [**ネットワーク] に移動します。
2. [設定] タブで、[**IPv6** トンネルのグローバル設定**]** をクリックします。
3. [**IPv6** トンネルグローバルパラメータの設定] ページで、[クライアントソース **IP** を使用] チェックボックスをオンにします。
4. [**OK**] をクリックします。

カプセル化解除設定

Citrix ADC アプライアンスまたはバックエンドサーバーのいずれかをカプセル化解除として構成できます。

Citrix ADC カプセル化解除

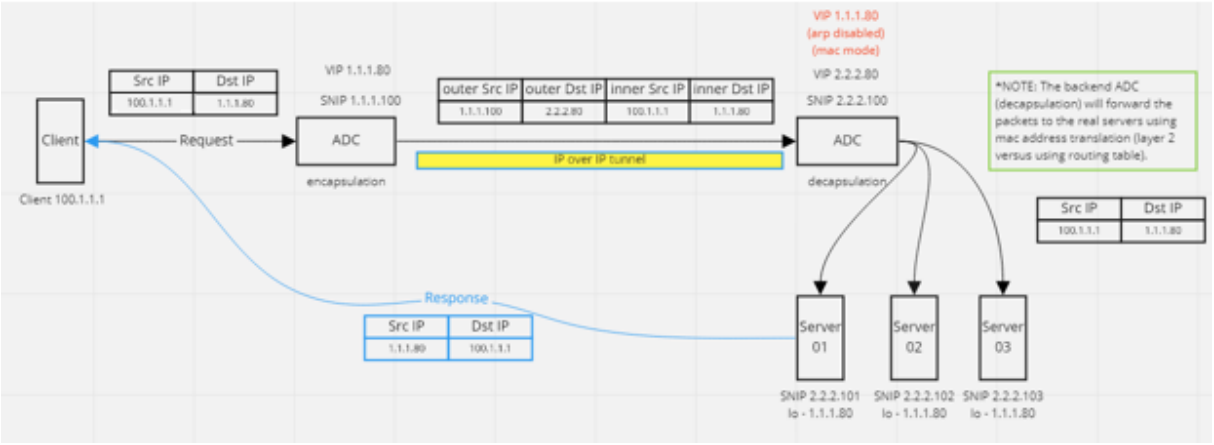
Citrix ADC アプライアンスをカプセル化解除として使用する場合は、Citrix ADC アプライアンスに IP トンネルを作成する必要があります。詳細については、IP トンネルの設定を参照してください。

Citrix ADC カプセル化解除セットアップは、次の 2 つの仮想サーバーで構成されます。

- 最初の仮想サーバがカプセル化されたパケットを受信し、外部 IP カプセル化を削除します。
- 2 番目の仮想サーバは、フロントエンド ADC 上の元のサービスの IP を持ち、MAC 変換を使用して、バインドされたサービスの MAC アドレスを使用してパケットをバックエンドに転送します。このセットアップは、通常 L2DSR と呼ばれます。この仮想サーバで ARP を無効にします。

設定例**:**

次の図は、ADC アプライアンスを使用したカプセル化解除の設定を示しています。



セットアップに必要な完全な構成は次のとおりです。

フロントエンド **ADC** 構成**:**

```
1 add service svc_DSR01 2.2.2.80 ANY * -usip YES -useproxyport NO
2 add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
     IPTUNNEL -sessionless ENABLED
3 bind lb vserver vip_DSR_ENCAP svc_DSR01
4 <!--NeedCopy-->
```

バックエンド **ADC** 構成**:**

```
1 add ipTunnel DSR-IPIP 1.1.1.100 255.255.255.255 *
2
3 add service svc_DSR01_01 2.2.2.101 ANY * -usip YES -useproxyport NO
4 add service svc_DSR01_02 2.2.2.102 ANY * -usip YES -useproxyport NO
5 add service svc_DSR01_03 2.2.2.103 ANY * -usip YES -useproxyport NO
```



```
6
7  add lb vserver vs_DSR_DECAP ANY 2.2.2.80 * -lbMethod SOURCEIPHASH -m
       IPTUNNEL -sessionless ENABLED -netProfile netProf_DSR_MBF_noIP
8
9  add ns ip 1.1.1.80 255.255.255.255 -type VIP -arp DISABLED -snmp
       DISABLED
10 add lb vserver vs_DSR_Relay ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
       MAC -sessionless ENABLED
11
12 bind lb vserver vs_DSR_DECAP svc_DSR01_01
13 bind lb vserver vs_DSR_DECAP svc_DSR01_02
14 bind lb vserver vs_DSR_DECAP svc_DSR01_03
15
16 bind lb vserver vip_DSR_Relay svc_DSR01_01
17 bind lb vserver vip_DSR_Relay svc_DSR01_02
18 bind lb vserver vip_DSR_Relay svc_DSR01_03
19
20 add netProfile netProf_DSR_MBF_noIP -MBF ENABLED
21 add lb monitor mon_DSR_MAC PING -netProfile netProf_DSR_MBF_noIP
22 bind service svc_DSR01_01 -monitorName mon_DSR_MAC
23 bind service svc_DSR01_02 -monitorName mon_DSR_MAC
24 bind service svc_DSR01_03 -monitorName mon_DSR_MAC
25 <!--NeedCopy-->
```

次の例は、apache2 を実行している Ubuntu サーバーと Red Hat サーバーを使用したテストセットアップを示しています。これらのコマンドは、各バックエンドサーバーで設定されます。

```
1 sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
2 sudo sysctl net.ipv4.conf.all.arp_ignore=1
3 sudo sysctl net.ipv4.conf.all.arp_announce=2
4 sudo sysctl net.ipv4.conf.eth4.rp_filter=2 (The interface has the
      external IP with route towards the ADC)
5 sudo sysctl net.ipv4.conf.all.forwarding=1
6 sudo ip link set dev lo arp on
7 <!--NeedCopy-->
```

バックエンドサーバーのカプセル化解除

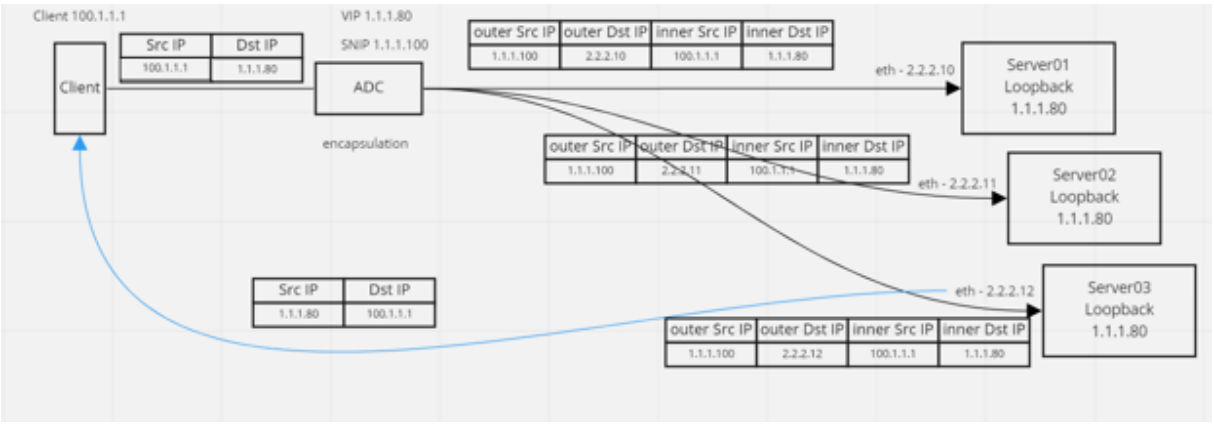
バックエンドサーバーをカプセル化解除として使用する場合、バックエンドの構成はサーバーの OS タイプによって異なります。次の手順に従って、バックエンドサーバーをカプセル化解除として構成できます。

1. サービス IP の IP を使用してループバックインターフェイスを設定します。

2. トンネルインターフェイスを作成します。
3. トンネルインターフェイスを介したルートを追加します。
4. トラフィックに必要なインターフェイス設定を構成します。

> 注:
>
> Windows OS サーバでは IP トンネリングをネイティブに実行できないため、Linux ベースのシステムの例としてコマンドが提供されています。Windows OS サーバーではサードパーティプラグインを使用できますが、この例の範囲外です。

次の図は、バックエンドサーバーを使用したカプセル化解除の設定を示しています。



設定例:

この例では、1.1.1.80 は Citrix ADC 仮想 IP（VIP）アドレスで、2.2.2.10-2.2.2.12 はバックエンドサーバーの IP アドレスです。VIP アドレスはループバックインターフェイスで設定され、ルートはトンネルインターフェイスを介して追加されます。モニタはサーバ IP を使用し、トンネルエンドポイントを使用して IP トンネル経由でモニタパケットをトンネリングします。

セットアップに必要な完全な構成は次のとおりです。

フロントエンド **ADC** 構成**:**

次の設定では、トンネルエンドポイントをソースとして使用するモニタが作成されます。次に、トンネル経由でサービス IP アドレスに ping を送信します。

```
1 add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
2 add netProfile netProfile_DSR -srcIP 10.0.1.2
3 add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
     YES -netProfile netProfile_DSR
4 <!--NeedCopy-->
```

次の設定では、元の送信元 IP アドレスを使用するサービスの VIP を作成します。次に、IP トンネル経由でバックエンドサーバーにトラフィックを転送します。


```
add service svc_DSR01 2.2.2.10 ANY * -usip YES -useproxyport NO
bind service svc_DSR01 -monitorName mon_DSR

add service svc_DSR02 2.2.2.11 ANY * -usip YES -useproxyport NO
bind service svc_DSR02 -monitorName mon_DSR

add service svc_DSR03 2.2.2.12 ANY * -usip YES -useproxyport NO
bind service svc_DSR03 -monitorName mon_DSR

add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED
bind lb vserver vip_DSR_ENCAP svc_DSR01
bind lb vserver vip_DSR_ENCAP svc_DSR02
bind lb vserver vip_DSR_ENCAP svc_DSR03
<!--NeedCopy-->
```

各サーバーのバックエンドサーバー構成**:**

次のコマンドは、バックエンドサーバーが IPIP パケットを受信し、外部カプセル化を削除し、ループバックから元のクライアント IP に応答するために必要です。そうすることで、クライアントが受信したパケット内の IP アドレスが元の要求の IP アドレスと一致するようにします。

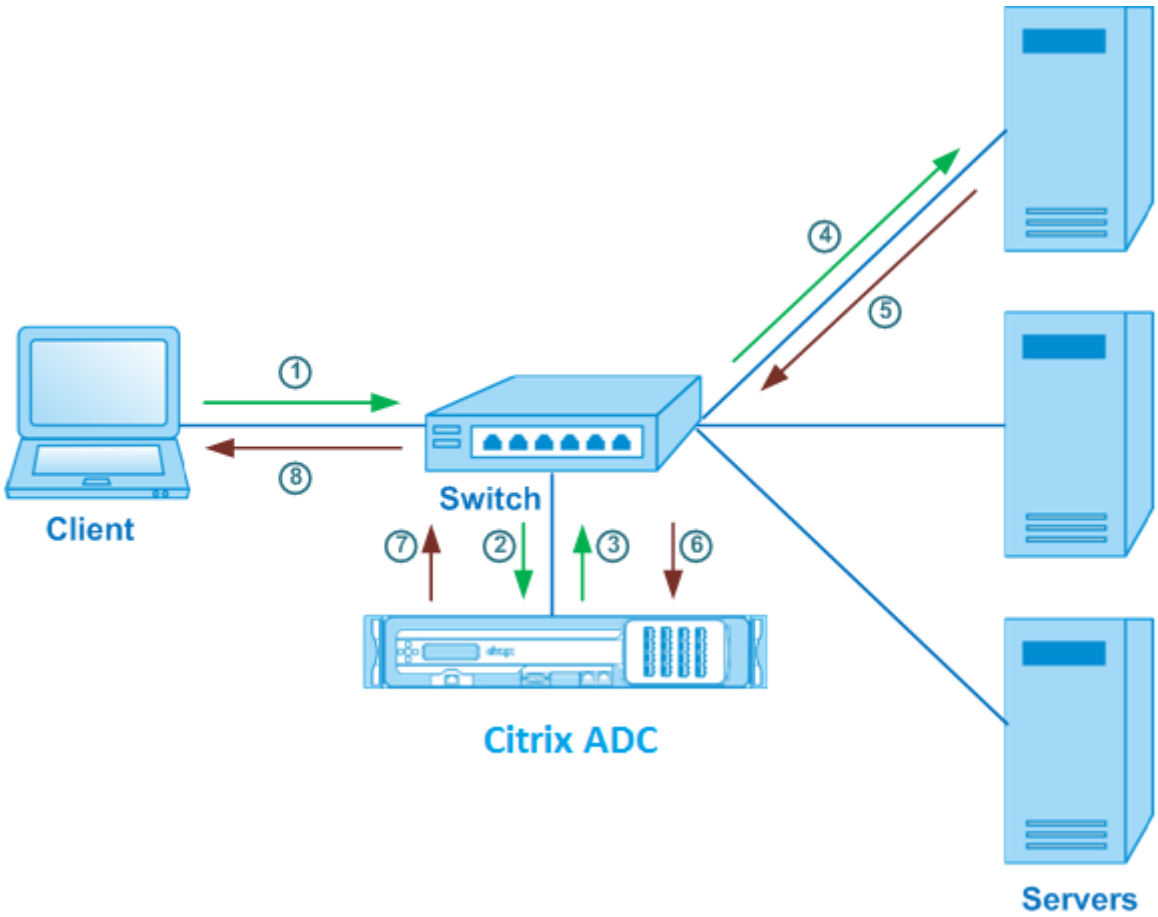
```
modprobe ipip
sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0
ifname tun0 remote 198.51.100.5 local 203.0.113.10
nmcli connection modify tun0 ipv4.addresses '10.0.1.1/30'
nmcli connection up tun0
sudo sysctl net.ipv4.conf.all.arp_ignore=1
sudo sysctl net.ipv4.conf.all.arp_announce=2
sudo sysctl net.ipv4.conf.tun0.rp_filter=2
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo ip link set dev lo arp off
<!--NeedCopy-->
```

使用例 **8:** ワンアームモードで負荷分散を設定する

October 7, 2021

ワンアーム設定では、Citrix ADC アプライアンスを単一の VLAN 経由でネットワークに接続します。アプライアンスは、単一の VLAN 上のクライアントから要求を受信し、同じ VLAN 上のサーバに要求を送信します。これは、ルータ、サーバ、アプライアンスをすべて同じスイッチに接続する最も簡単な導入シナリオの 1 つです。スイッチでのクライアント要求はアプライアンスに転送され、アプライアンスは設定されたロードバランシング方式を使用してサービスを選択します。

図 1: ワンアームモードでのロードバランシング



この例のシナリオでは、Service-ANY-1、Service-ANY-2、Service-ANY-3 が作成され、仮想サーバ Vserver-LB-1 にバインドされます。仮想サーバは、クライアント要求をサービスに対して負荷分散します。次の表に、ワンアームモードでアプライアンスに設定されているエンティティの名前と値を示します。

エンティティタイプ	名前	IP アドレス	プロトコル
仮想サーバ	Vserver-LB-1	10.102.29.94	ANY
サービス	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY

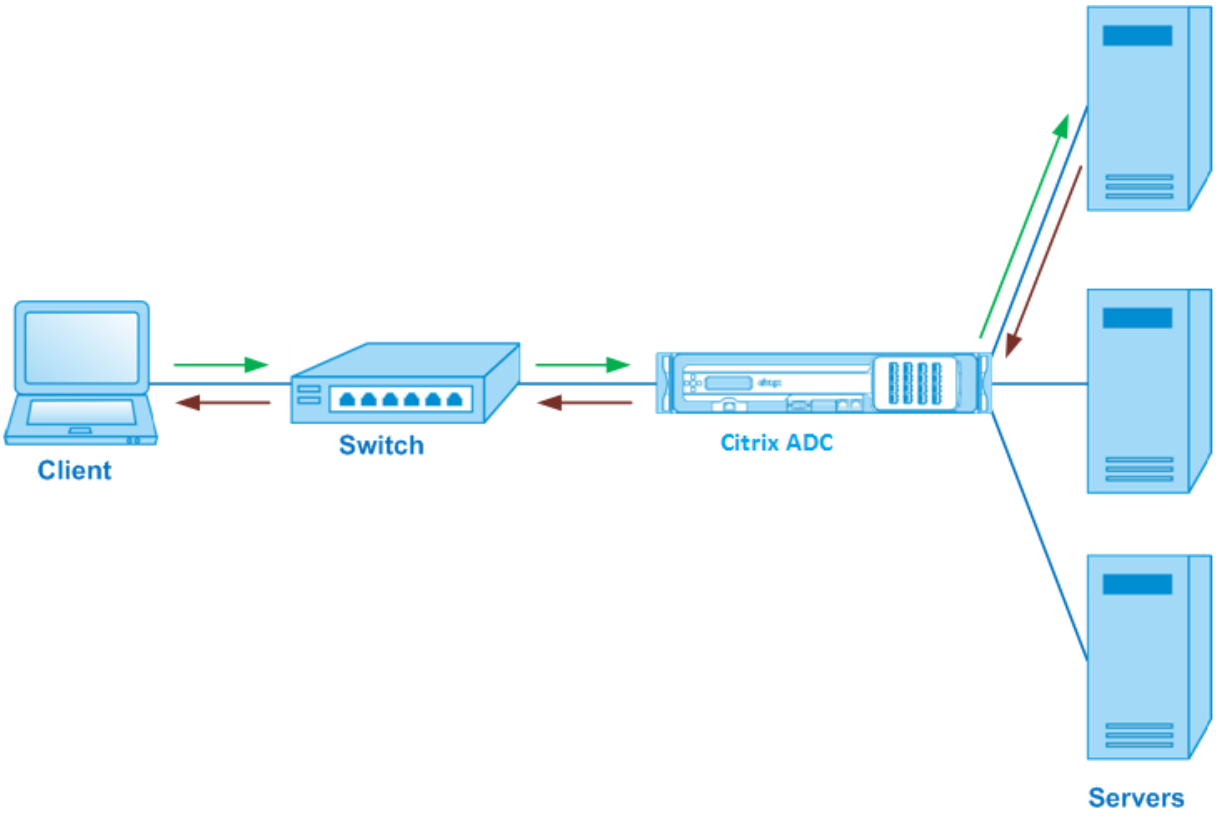
エンティティタイプ	名前	IP アドレス	プロトコル
モニター	TCP	なし	なし

ワンアームモードでロードバランシング設定を構成するには、基本ロードバランシングの設定を参照してください。

使用例 **9:** インラインモードで負荷分散を構成する

October 7, 2021

インラインモード（2 アームモードとも呼ばれます）では、Citrix ADC アプライアンスを複数の VLAN 経由でネットワークに接続します。アプライアンスは、1 つの VLAN 上のクライアントから要求を受信し、別の VLAN 上のサーバに要求を送信します。2 アームのセットアップでは、アプライアンスはサーバーとクライアントの間で接続されます。スイッチでのクライアント要求はアプライアンスに転送され、アプライアンスは設定されたロードバランシング方式を使用してサービスを選択します。



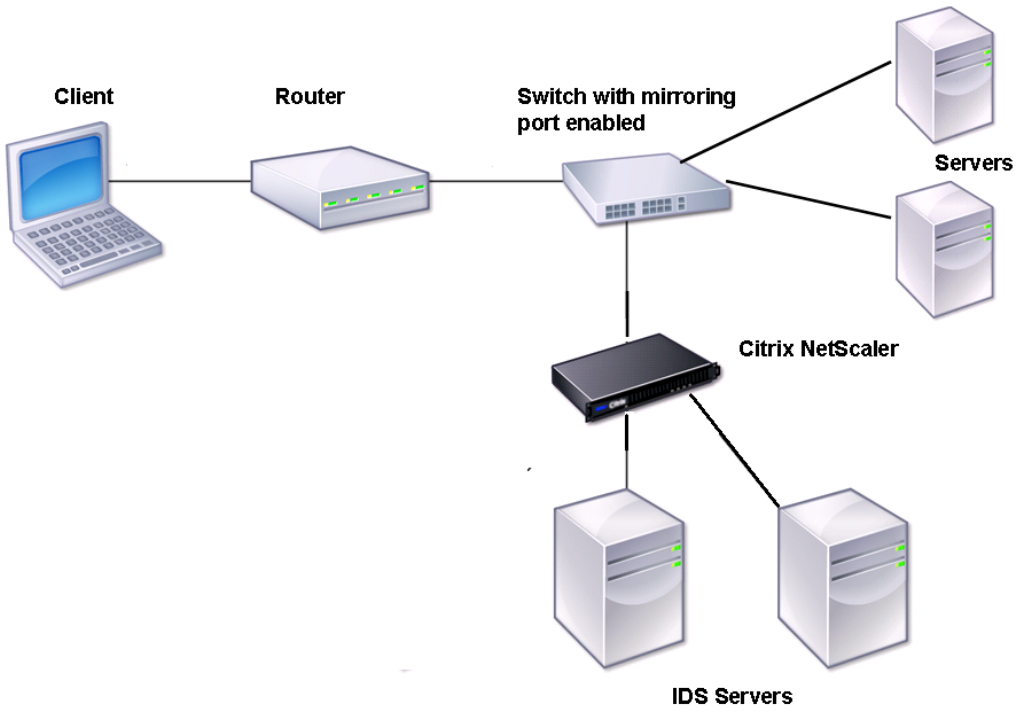
インラインモードの設定とエンティティ図は、ワンアームモードでのロードバランシングの設定で説明されているものと同じです。

使用例 **10:** 侵入検知システムサーバーの負荷分散

October 7, 2021

Citrix ADC アプライアンスが侵入検知システム（IDS）サーバの負荷分散をサポートできるようにするには、ポートミラーリングが有効になっているスイッチを介して IDS サーバとクライアントを接続する必要があります。クライアントは、サーバーに要求を送信します。スイッチ上でポートミラーリングが有効になっているため、要求パケットは Citrix ADC アプライアンスの仮想サーバーポートにコピーまたは送信されます。アプライアンスは、次の図に示すように、設定されたロードバランシング方式を使用して IDS サーバを選択します。

図 1: ロードバランシングされた IDS サーバのトポロジ



注: 現在、アプライアンスはパッシブ IDS デバイスのロードバランシングのみをサポートしています。

前の図に示すように、IDS ロードバランシングの設定は次のように機能します。

1. クライアント要求は IDS サーバに送信され、ミラーリングポートが有効になっているスイッチはこれらのパケットを IDS サーバに転送します。送信元 IP アドレスはクライアントの IP アドレスで、宛先 IP アドレスはサーバの IP アドレスです。送信元 MAC アドレスはルータの MAC アドレスで、宛先 MAC アドレスはサーバの MAC アドレスです。
2. スイッチを通過するトラフィックは、アプライアンスにミラーリングされます。アプライアンスは、レイヤ 3

情報（送信元 IP アドレスと宛先 IP アドレス）を使用して、送信元 IP アドレスまたは宛先 IP アドレスを変更せずに、選択した IDS サーバにパケットを転送します。送信元 MAC アドレスと宛先 MAC アドレスが、選択した IDS サーバの MAC アドレスに変更されます。

注：IDS サーバをロードバランシングする場合、SRCIPHASH、DESTIPHASH、または SRCIPDESTIPHASH ロードバランシング方式を設定できます。クライアントからアプライアンス上のサービスに流れるパケットは、単一の IDS サーバに送信する必要があるため、SRCIPDESTIPHASH 方式が推奨されます。

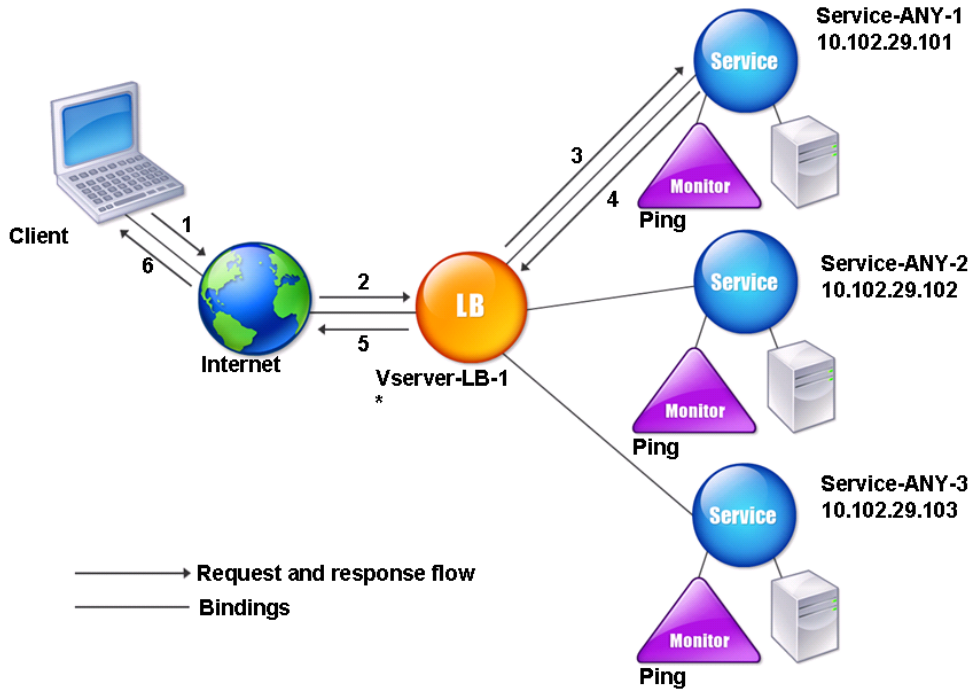
Service-ANY-1、Service-ANY-2、Service-ANY-3 が作成され、Vserver-LB-1 にバインドされているとします。仮想サーバは、サービスの負荷を分散します。次の表に、アプライアンス上で構成されているエンティティの名前と値を示します。

エンティティタイプ	名前	IP アドレス	ポート	プロトコル
仮想サーバ	Vserver-LB-1	*	*	ANY
サービス	Service-ANY-1	10.102.29.101	*	ANY
	Service-ANY-2	10.102.29.102	*	ANY
	Service-ANY-3	10.102.29.103	*	ANY
モニター	Ping	なし	なし	なし

注：IDS ロードバランシングの設定には、インラインモードまたはワンアームモードを使用できます。

次の図は、アプライアンスで設定するパラメータの負荷分散エンティティと値を示しています。

図 2：ロードバランシング IDS サーバのエンティティモデル



IDS ロードバランシング設定を設定するには、最初に MAC ベースの転送を有効にする必要があります。アプライアンスのレイヤ 2 モードとレイヤ 3 モードも無効にします。

コマンドラインインターフェイスを使用して **MAC** ベースの転送を有効にするには

コマンドプロンプトで入力します。

```
enable ns mode <ConfigureMode>
<!--NeedCopy-->
```

例:

```
enable ns mode MAC
<!--NeedCopy-->
```


構成ユーティリティを使用して **MAC** ベースの転送を有効にするには

[システム] **>** [設定] **>** [モードの設定] に移動し、[**MAC** ベースの転送] を選択します。

次に、基本的な負荷分散設定を構成するには、「基本負荷分散の設定」を参照してください。

基本的なロードバランシング設定を設定したら、サポートされているロードバランシング方式（セッションレス仮想サーバ上の SRCIPDESTIP Hash 方式など）を設定し、MAC モードを有効にして IDS 用にカスタマイズする必要があります。アプライアンスは接続の状態を維持せず、パケットを処理せずに IDS サーバにだけ転送します。仮想サーバが MAC モードであるため、宛先 IP アドレスとポートは変更されません。

コマンドラインインターフェイスを使用してセッションレス仮想サーバーの負荷分散方式とリダイレクトモードを構成するには

コマンドプロンプトで入力します。

```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

例**:**

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -
    sessionless enabled
<!--NeedCopy-->
```

> 注
>
> -m MAC オプションが有効になっている仮想サーバーにバインドされているサービスの場合は、非ユーザーモニターをバインドする必要があります。

構成ユーティリティを使用してセッションレス仮想サーバーの負荷分散方式とリダイレクトモードを構成するには

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを開き、[リダイレクションモード] で [MAC ベース] を選択します。
3. [詳細設定] で、[メソッド] をクリックし、[SRCIPDESTIPHASH] を選択します。[トラフィックの設定] をクリックし、[セッションレスロードバランシング] を選択します。

コマンドラインインターフェイスを使用して送信元 **IP** アドレスを使用するようにサービスを設定するには

コマンドプロンプトで入力します。

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

例**:**

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

構成ユーティリティを使用して送信元 **IP** アドレスを使用するようにサービスを設定するには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動します。
2. サービスを開き、[設定] で [送信元 **IP** アドレスを使用] を選択します。

USIP を正しく機能させるには、グローバルに設定する必要があります。USIP をグローバルに設定する方法の詳細については、「IP アドレッシング」を参照してください。

ユースケース **11:** リッスンポリシーを使用したネットワークトラフィックの分離

October 7, 2021

> 注
>
> シャドウ仮想サーバーを使用してマルチテナント分離をシミュレートするトラフィック分離ソリューションは、推奨されなくなりました。または、このような展開には Citrix ADC AdminPartitioning 機能を使用することをお勧めします。詳細については、「管理者のパーティショニング」を参照してください。

データセンターの一般的なセキュリティ要件は、さまざまなアプリケーションまたはテナントのトラフィック間のネットワークパスの分離を維持することです。1 つのアプリケーションまたはテナントのトラフィックは、他のアプリケーションまたはテナントのトラフィックから分離する必要があります。たとえば、ある金融サービス会社は、保険部門のアプリケーションのトラフィックを、金融サービスアプリケーションのトラフィックとは別に維持したいと考えています。これまで、これは、ファイアウォール、ロードバランサー、IdP などのネットワークサービスデバイスを物理的に分離し、スイッチングファブリックでネットワークを監視して論理的に分離することで簡単に実現できました。

データセンターアーキテクチャがマルチテナント仮想化データセンターに進化するにつれて、データセンターのアグリゲーションレイヤーのネットワークサービスが統合されています。この開発により、ネットワーク・パス・アイソレーションがネットワーク・サービス・デバイスにとって重要なコンポーネントとなり、ADC が L4〜L7 レベルでト

ラフィックを分離できるようにする必要性が高まっています。さらに、特定のテナントのすべてのトラフィックは、サービスレイヤに到達する前にファイアウォールを通過する必要があります。

ネットワークパスを分離する要件に対処するために、Citrix ADC アプライアンスはネットワークドメインを識別し、ドメイン間のトラフィックを制御します。Citrix ADC ソリューションには、リッスンポリシーとシャドウ仮想サーバーの 2 つの主要コンポーネントがあります。

分離される各ネットワークパスには、リッスンポリシーが定義されている仮想サーバーが割り当てられ、仮想サーバーは指定されたネットワークドメインからのトラフィックのみをリッスンします。

トラフィックを分離するために、リッスンポリシーは、いくつかのクライアントパラメータまたはそれらの組み合わせに基づくことができ、ポリシーに優先順位を割り当てることができます。次の表に、トラフィックを識別するためにリッスンポリシーで使用できるパラメータを示します。

カテゴリ	パラメーター
イーサネットプロトコル	送信元 MAC アドレス、宛先 MAC アドレス
ネットワークインターフェイス	ネットワーク ID、受信スループット、送信スループット、送信スループット
IP プロトコル	送信元 IP アドレス、宛先 IP アドレス
IPv6 プロトコル	送信元 IPv6 アドレス、宛先 IPv6 アドレス
TCP プロトコル	送信元ポート、宛先ポート、最大セグメントサイズ、ペイロード、およびその他のオプション
UDP プロトコル	送信元ポート、宛先ポート
VLAN	ID

表 1. リッスンポリシーの定義に使用されるクライアントパラメータ

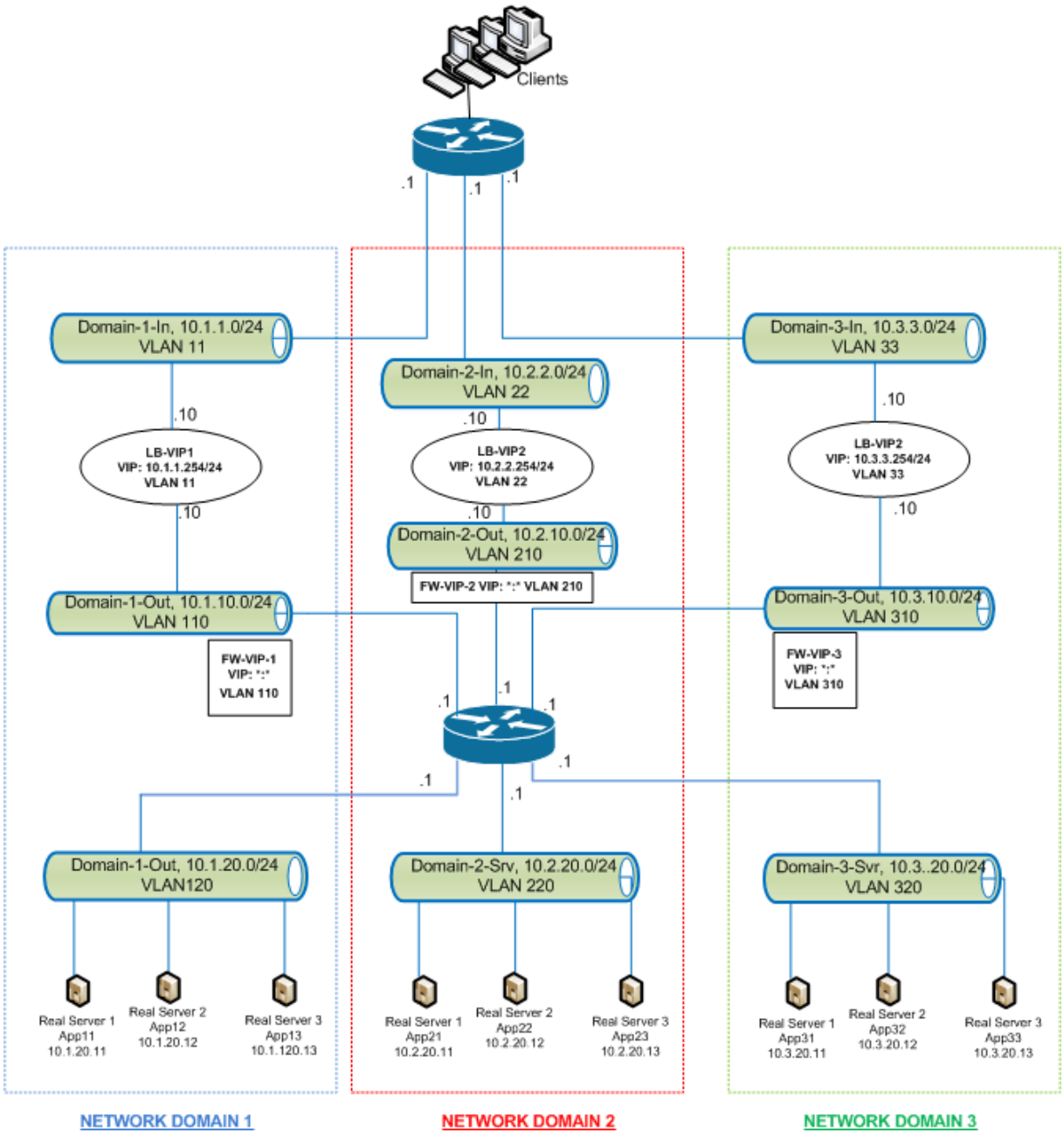
Citrix ADC アプライアンスでは、各ドメインに仮想サーバーが構成され、そのドメインに対するトラフィックのみをリッスンするように指定するリッスンポリシーが適用されます。また、各ドメインに対して構成されるシャドウ負荷分散仮想サーバーは、任意のドメイン宛てのトラフィックをリッスンします。シャドウロードバランシングの各仮想サーバーには、ワイルドカード (*) の IP アドレスとポートがあり、そのサービスタイプは ANY に設定されます。

各ドメインでは、ドメインのファイアウォールがサービスとしてシャドウ負荷分散仮想サーバーにバインドされ、すべてのトラフィックがファイアウォールを経由して転送されます。ローカルトラフィックは宛先に転送され、別のドメイン宛てのトラフィックはそのドメインのファイアウォールに転送されます。シャドウロードバランシング仮想サーバは、MAC モードリダイレクト用に構成されます。

ネットワークパスの分離方法

次の図は、ドメイン間の一般的なトラフィックフローを示しています。ネットワークドメイン 1 内、およびネットワークドメイン 1 とネットワークドメイン 2 の間のトラフィックフローを考慮します。

図 1: ネットワークパスの分離



ネットワークドメイン **1** 内のトラフィック

ネットワークドメイン 1 には、VLAN 11、VLAN 110、および VLAN 120 という 3 つの VLAN があります。次の手順では、トラフィックフローについて説明します。

- VLAN 11 からのクライアントは、VLAN 120 のサービスプールから利用可能なサービスの要求を送信します。
- ロードバランシング仮想サーバ LB-VIP1 は、VLAN 11 からのトラフィックをリッスンするように設定され、要求を受信して VLAN 110 に転送します。VLAN 110 の仮想サーバは、要求をシャドウロードバランシング仮想サーバ FW-VIP-1 に転送します。
- VLAN 110 からのトラフィックをリッスンするように設定された FW-VIP-1 は、要求を受信して VLAN 120 に転送します。
- VLAN 120 のロードバランシング仮想サーバは、物理サーバ App11、App12、または App13 のいずれかに要求をロードバランシングします。
- 物理サーバから送信された応答は、VLAN 11 内のクライアントへの同じパスで返されます。

この構成により、クライアントから発信されるすべてのトラフィックについて、常に Citrix ADC 内でトラフィックが分離されます。

ネットワークドメイン **1** とネットワークドメイン **2** の間のトラフィック

ネットワークドメイン 1 には、VLAN 11、VLAN 110、および VLAN 120 という 3 つの VLAN があります。また、ネットワークドメイン 2 には、VLAN 22、VLAN 210、および VLAN 220 という 3 つの VLAN があります。次の手順では、VALN 11 から VLAN 22 へのトラフィックフローについて説明します。

- ネットワークドメイン 1 に属する VLAN 11 のクライアントは、ネットワークドメイン 2 に属する VLAN 220 のサービスプールから利用可能なサービスの要求を送信します。
- ネットワークドメイン 1 では、VLAN 11 からのトラフィックをリッスンするように設定されたロードバランシング仮想サーバ LB-VIP1 が要求を受信し、その要求を VLAN 110 に転送します。
- シャドウロードバランシング仮想サーバー FW-VIP-1 は、他のドメインを宛先とする VLAN 110 トラフィックをリッスンするように構成されており、要求がネットワークドメイン 2 の物理サーバーに送信されるため、要求を受信し、ファイアウォール仮想サーバー FW-VIP-2 に転送します。
- ネットワークドメイン 2 では、FW-VIP-2 は要求を VLAN 220 に転送します。
- VLAN 220 のロードバランシング仮想サーバは、物理サーバ App21、App22、または App23 のいずれかに要求をロードバランシングします。
- 物理サーバから送信された応答は、ネットワークドメイン 2 のファイアウォールを経由した同じパスで返され、次にネットワークドメイン 1 に VLAN 11 のクライアントに到達します。

構成の手順

リッスンポリシーを使用してネットワークパス分離を構成するには、次の手順を実行します。

- リッスンポリシー式を追加します。各式は、トラフィックの宛先となるドメインを指定します。VLAN ID またはその他のパラメータを使用して、トラフィックを識別できます。
- 各ネットワークドメインについて、次のように 2 つの仮想サーバを構成します。
  - このドメイン宛てのトラフィックを識別するリッスンポリシーを指定する、負荷分散仮想サーバーを作成します。前に作成した式の名前を指定することも、仮想サーバーの作成中に式を作成することもできます。

  - シャドウ仮想サーバーと呼ばれる別の負荷分散仮想サーバーを作成します。このサーバーには、任意のドメイン宛てのトラフィックに適用されるリッスンポリシー式を指定します。この仮想サーバでは、サービスタイプを ANY に設定し、IP アドレスとポートをアスタリスク（*）に設定します。この仮想サーバで MAC ベースの転送を有効にします。
  - 両方の仮想サーバで [L2 接続] オプションを有効にします。
    一般に、Citrix ADC アプライアンスは、接続を識別するために、クライアント IP アドレス、クライアントポート、宛先 IP アドレス、宛先ポートの 4 タプルを使用します。[L2 接続] オプションを有効にすると、通常の 4 タプルに加えて、接続のレイヤ 2 パラメータ（チャネル番号、MAC アドレス、VLAN ID）が使用されます。
- ドメイン内のサーバプールを表すサービスを追加し、仮想サーバにバインドします。
- 各ドメインのファイアウォールをサービスとして構成し、すべてのファイアウォールサービスをシャドウ仮想サーバーにバインドします。

コマンドラインインターフェイスを使用してネットワークトラフィックを分離するには

コマンドプロンプトで、次のコマンドを入力します。

```
add policy expression <expressionName> <listenPolicyExpression>

add lb vserver <name> <serviceType> <ip> <port> -l2conn ON -
    listenPolicy <expressionName>
<!--NeedCopy-->
```

各ドメインに負荷分散仮想サーバーを追加します。この仮想サーバは、同じドメインのトラフィック用です。

```
add lb vserver <name> ANY * * -l2conn ON -m MAC -listenPolicy <
    expressionName>
<!--NeedCopy-->
```

ドメインごとにシャドウ負荷分散仮想サーバーを追加します。この仮想サーバは、他のドメインのトラフィック用です。

例:

```
add policy expression e110 client.vlan.id==110
add policy expression e210 client.vlan.id==210
add policy expression e310 client.vlan.id==310
add policy expression e11 client.vlan.id==11
add policy expression e22 client.vlan.id==22
```



```
6  add policy expression e33 client.vlan.id==33
7
8  add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -persistenceType NONE -
       listenPolicy e11
9  -cltTimeout 180 -l2Conn ON
10
11 add lb vserver LB-VIP2 HTTP 10.2.2.254 80 -persistenceType NONE -
       listenPolicy e22
12 -cltTimeout 180 -l2Conn ON
13
14 add lb vserver LB-VIP3 HTTP 10.3.3.254 80 -persistenceType NONE -
       listenPolicy e33
15 -cltTimeout 180 -l2Conn ON
16
17
18 add lb vserver FW-VIP-1 ANY * * -persistenceType NONE -lbMethod
       ROUNDROBIN - listenPolicy e110 -Listenpriority 1 -m MAC -cltTimeout
       120
19
20 add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod
       ROUNDROBIN - listenPolicy e210 -Listenpriority 2 -m MAC -cltTimeout
       120
21
22 add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod
       ROUNDROBIN - listenPolicy e310 -Listenpriority 3 -m MAC -cltTimeout
       120
23
24
25 add service RD-1 10.1.1.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
       DISABLED
26 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
       NO -TCPB NO -CMP NO
27
28 add service RD-2 10.2.2.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
       DISABLED
29 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
       NO -TCPB NO -CMP NO
30
31 add service RD-3 10.3.3.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
       DISABLED
32 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
       NO -TCPB NO -CMP NO
33
34
35 bind lb vserver FW-VIP-1 RD-1
```



```
36
37 bind lb vserver FW-VIP-2 RD-2
38
39 bind lb vserver FW-VIP-3 RD-3
40 <!--NeedCopy-->
```

構成ユーティリティを使用してネットワークトラフィックを分離するには

1. サービスの作成の説明に従って、サーバを表すサービスを追加します。
2. 各ファイアウォールをサービスとして追加します。
   a) **Traffic Management > Load Balancing > Services** に移動します。
   b) プロトコルを ANY、サーバーをファイアウォールの IP アドレス、ポートを 80 に指定して、サービスを作成します。
3. 負荷分散仮想サーバーを構成します。
4. シャドウ負荷分散仮想サーバーを構成します。
5. ネットワークドメインごとに、手順 3 と 4 を繰り返します。
6. [負荷分散仮想サーバー] ウィンドウで、作成した仮想サーバーを開き、設定を確認します。

ユースケース 12：負荷分散のための XenDesktop の構成

October 7, 2021

仮想デスクトップアプリケーションの配信のパフォーマンスを向上させるには、Citrix ADC アプライアンスを Citrix XenDesktop と統合し、Citrix ADC の負荷分散機能を使用して、Web インターフェイスサーバーとデスクトップ Delivery Controller（DDC）サーバーに負荷を分散します。

一般に、XenDesktop は、アプリケーションがターミナルサーバーまたは XenApp で実行すると互換性がない場合、または各仮想デスクトップに固有の要件がある場合に使用します。このような場合、接続するユーザーごとに 1 つのデスクトップホストが必要です。ただし、現在接続しているユーザーごとに 1 つのホストしか必要ないように、ホストをプールできます。

XenDesktop 用に展開されるコアアプリケーションサービスは、デスクトップ Delivery Controller（DDC）です。DDC はサーバーにインストールされ、その主な機能はデスクトップホストを登録し、クライアント接続をブローカリングすることです。

また、DDC はユーザーを認証し、デスクトップの状態を制御し、デスクトップを起動および停止することによって、ユーザーの仮想デスクトップ環境のアセンブリを管理します。

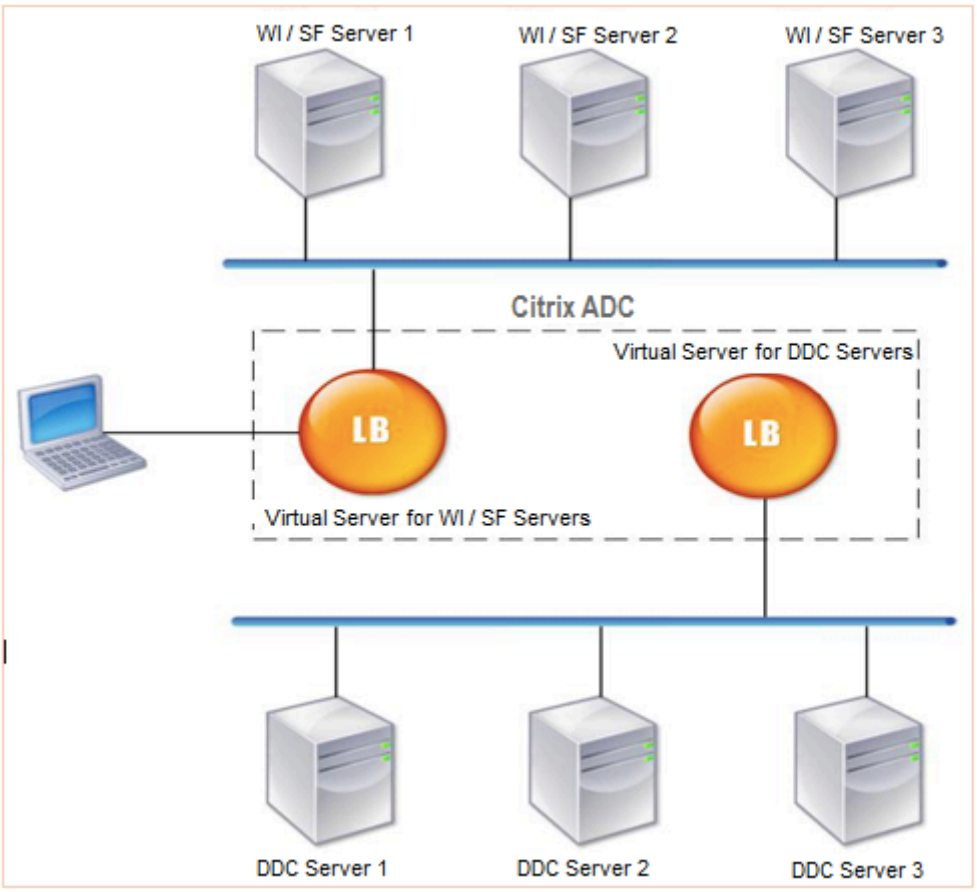
一般に、可用性を高めるために、複数の DDC がインストールされています。

Web Interface サーバーは、仮想デスクトップへの安全なアクセスを提供します。Web インターフェイスは、デスクトップ Delivery Controller（DDC）への初期接続ポータルです。ユーザーのデバイス上の Web ブラウザは、

Web サーバーに情報を送信します。Web サーバーはサーバーファームと通信し、仮想デスクトップへのアクセスをユーザーに提供します。

次の図は、XenDesktop で動作する Citrix ADC アプライアンスのトポロジーを示しています。

図 1: XenDesktop の負荷分散



> 注
>
> HTTP プロトコルを使用することもできますが、クライアントと Citrix ADC アプライアンス間の通信には SSL を使用することをお勧めします。クライアントとの通信に SSL プロトコルを使用していても、Citrix ADC と DDC サーバー間の通信には HTTP プロトコルを使用できます。

GUI を使用して XenDesktop の負荷分散を構成するには

1. サービスを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックします。
   b) 名前、IP アドレス、ポート、プロトコルの種類を指定してサービスを作成し、[**OK**] をクリックします。
2. 負荷分散仮想サーバーを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックします。

b) 名前、IP アドレス、ポート、プロトコルの種類を指定して仮想サーバーを作成し、[**OK**] をクリックします。

3. サービスを負荷分散仮想サーバーにバインドします。
4. [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択します。
   a) ［編集］をクリックします。
   b) [サービスとサービスグループ] で、[**>**] をクリックし、[バインドの追加] をクリックします。
   c) バインドするサービスを選択し、ウェイト値を入力します。
   d) ［バインド］をクリックします。

コマンドラインインターフェイスを使用して **XenDesktop** の負荷分散を構成するには

- サービスを作成するには、コマンドプロンプトで次のように入力します。

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

例**:**

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- 仮想サーバーを作成するには、コマンド・プロンプトで次のように入力します。

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

例**:**

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

- サービスを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例**:**

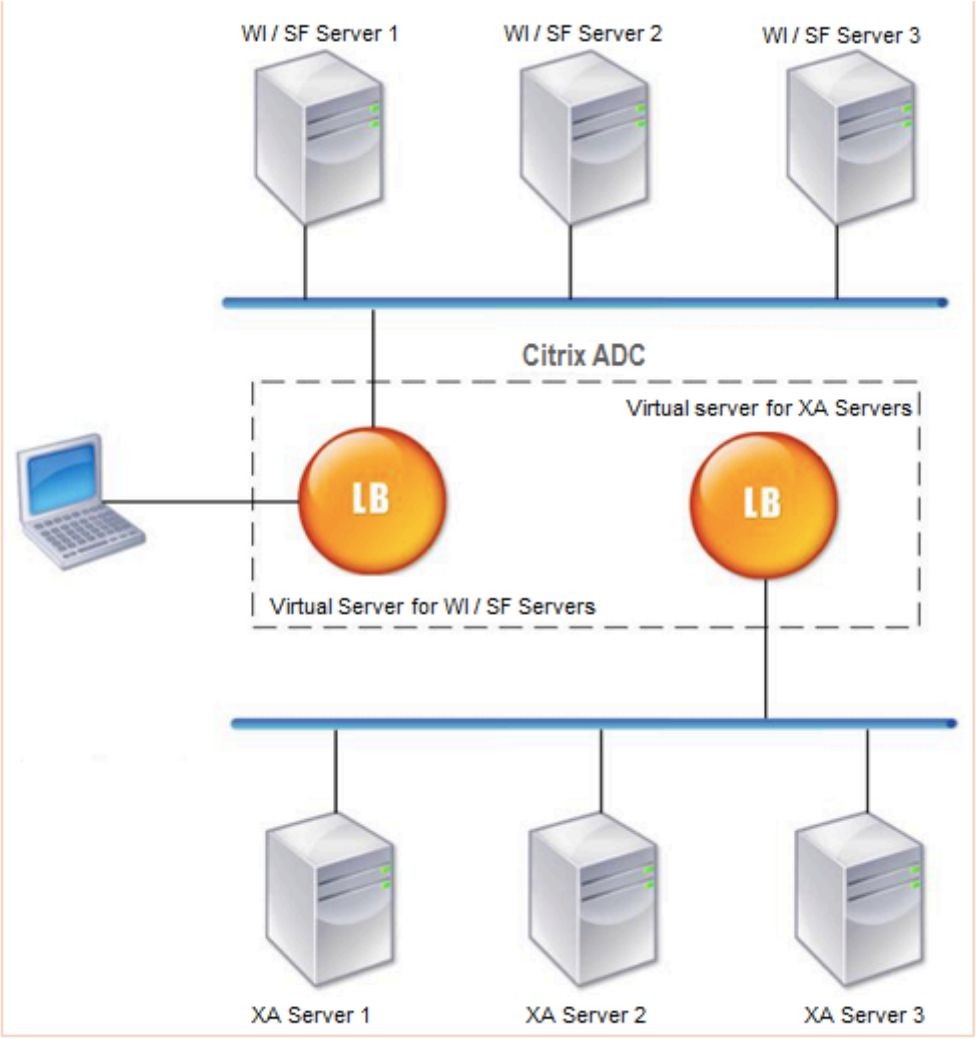

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

ユースケース 13：負荷分散のための XenApp の構成

October 7, 2021

アプリケーションを効率的に配信するには、Citrix ADC アプライアンスを Citrix XenApp に統合し、Citrix ADC の負荷分散機能を使用して XenApp サーバーファーム全体に負荷を分散します。次の図は、このようなセットアップのトポロジ図です。

図 1：XenApp の負荷分散



Web Interface サーバーは、ユーザーの Web ブラウザーを介して XenApp アプリケーションリソースに安全にアクセスできます。Web Interface クライアントは、XenApp サーバーファームで利用可能なアプリケーション、コンテンツ、デスクトップなどのすべてのリソースをユーザーに提供します。ユーザーは、標準の Web ブラウザまたは Citrix オンラインプラグインを使用して、公開されたリソースにアクセスできます。

ユーザーのデバイスの Web ブラウザが Web サーバーに情報を送信します。Web サーバーはサーバーファーム上のサーバーと通信し、リソースへのアクセスをユーザーに提供します。

Web インターフェイスと XML ブローカは、補完的なサービスです。Web Interface は、ユーザーにアプリケーションへのアクセスを提供し、XML Broker はユーザーの権限を評価して、Web Interface に表示されるアプリケーションを決定します。

XML サービスは、サーバーファーム内のすべてのサーバーにインストールされます。Web インターフェイスで指定された XML サービスは、XML ブローカとして機能します。Web Interface サーバーによって渡されたユーザー資格

情報に基づいて、XML Broker サーバーはユーザーがアクセスできるアプリケーションのリストを送信します。

複数の Web Interface サーバーと XML Broker サーバーを展開する大企業では、Citrix ADC アプライアンスを使用してこれらのサーバーの負荷を分散することをお勧めします。Web Interface サーバーを負荷分散するように 1 つの仮想サーバーを構成し、XML Broker サーバー用に別の仮想サーバーを構成します。負荷分散方式およびその他の機能は、必要に応じて仮想サーバ上で設定できます。

> 注
>
> HTTP プロトコルを使用することもできますが、クライアントと Citrix ADC 間の通信には SSL を使用することをお勧めします。クライアントとの通信に SSL プロトコルを使用していても、Citrix ADC と WI サーバー間の通信には HTTP プロトコルを使用できます。

GUI を使用して XenApp の負荷分散を構成するには

1. サービスを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックします。
   b) 名前、IP アドレス、ポート、プロトコルの種類を指定してサービスを作成し、[**OK**] をクリックします。
2. 負荷分散仮想サーバーを作成します。
   a) [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックします。
   b) 名前、IP アドレス、ポート、プロトコルの種類を指定して仮想サーバーを作成し、[**OK**] をクリックします。
3. サービスを負荷分散仮想サーバーにバインドします。
4. [設定] > [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択します。
   a) ［編集］をクリックします。
   b) [サービスとサービスグループ] で、[**>**] をクリックし、[バインドの追加] をクリックします。
   c) バインドするサービスを選択し、ウェイト値を入力します。
   d) ［バインド］をクリックします。

コマンドラインインターフェイスを使用して XenApp の負荷分散を構成するには

- サービスを作成するには、コマンドプロンプトで次のように入力します。

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

例:

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```


- 仮想サーバーを作成するには、コマンド・プロンプトで次のように入力します。

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

- サービスを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

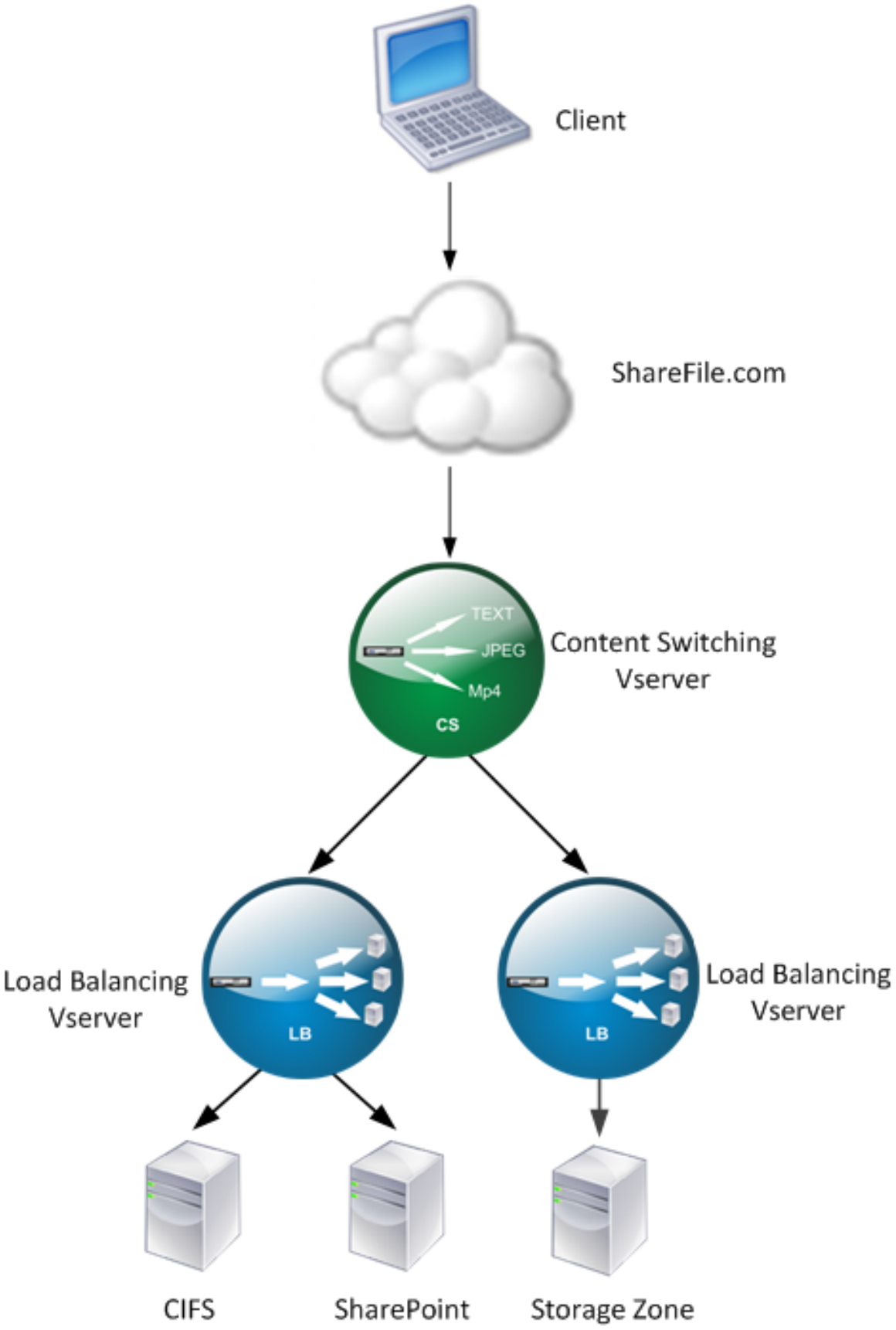
ユースケース 14: Citrix 共有ファイルの負荷分散のための ShareFile ウィザード

October 7, 2021

ウィザードを使用して、Citrix ShareFile 用の負荷分散を構成できます。Citrix ShareFile ウィザードは、要求されたコンテンツのタイプに基づいて、ShareFile サイトの負荷分散構成をセットアップするのに役立ちます。コンテンツスイッチングサーバーは、StorageZone、CIFS、または SharePoint 要求のいずれであるかに基づいて要求を送信します。コンテンツスイッチングは、ポリシーに基づいています。ウィザードは、要求が StorageZone、CIFS、または SharePoint のいずれであるかを識別するためのポリシーを自動生成します。コンテンツスイッチ仮想サーバーは、これらのポリシーを使用して、要求を正しい負荷分散サーバーに転送します。

次の図に示すように、一般的なデータフローを表すことができます。

図 1: ShareFile データ・ロード・バランシング

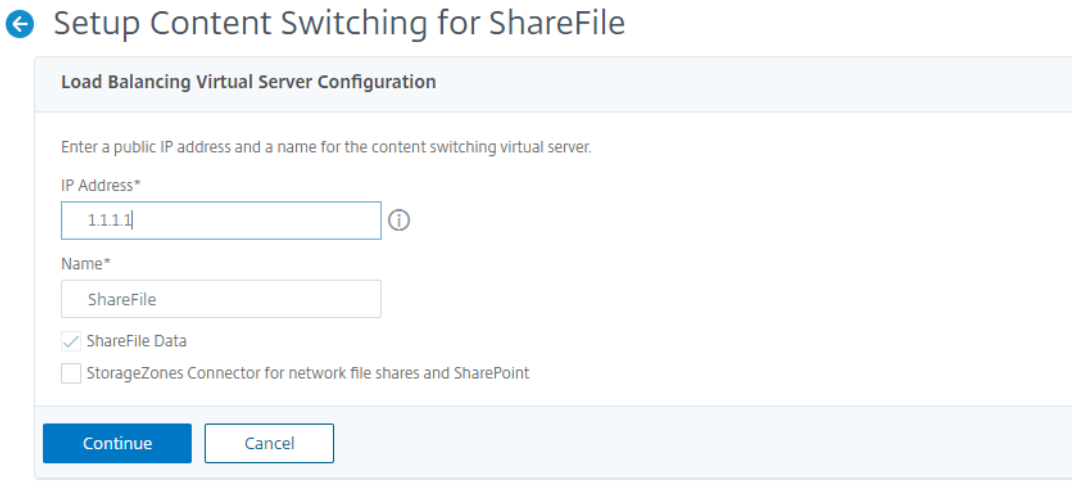


Traffic Management > Virtual Servers および **Services > Virtual Servers** に移動して ShareFile ウィザードが作成する負荷分散仮想サーバーを表示できます。ShareFile ウィザードを使用して作成した仮想サーバーを手動で削除することはできません。ウィザードを使用して、仮想サーバを削除します。

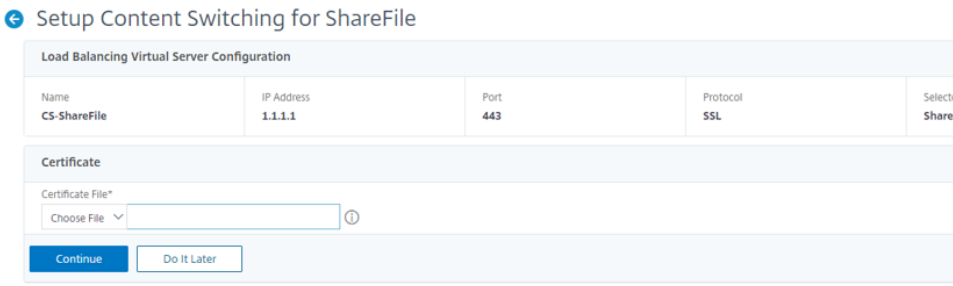
Citrix ADC は、SharePoint または CIFS 要求に対して LDAP 認証を使用します。ハッシュ認証は、StorageZone の要求を認証するために使用されます。

Citrix ADC アプライアンスを構成して **CitrixShareFile** 負荷分散を構成するには

1. ナビゲーションペインで、[トラフィック管理] をクリックします。
2. [**CitrixShareFile**] セクションで、[**Citrix ADC** を **ShareFile** 用に設定] をクリックします。
3. 「**ShareFile** 用コンテンツスイッチングの設定」ページで、次の情報を入力します。
   - IP アドレス: コンテンツスイッチ仮想サーバーの IP アドレス。
   - [名前]: コンテンツスイッチ仮想サーバーの名前。
   - CIFS または SharePoint の負荷分散を設定する場合は 、ネットワークファイル用の StorageZone コネクタをクリックします Shares/SharePoint チェックボックスをオンにして、[続行] をクリックします。既定では、[**ShareFile** データ] チェックボックスはオンになっています。

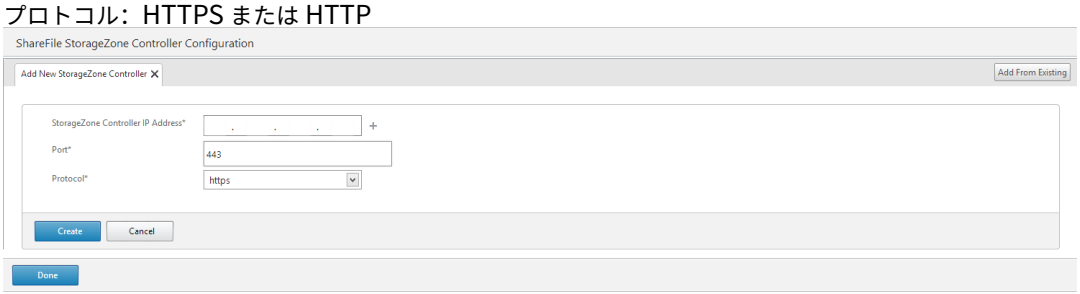


4. 有効な証明書を指定します。証明書がある場合は、[証明書の 選択] をクリックし、ドロップダウンリストから証明書を選択します。証明書をインストールする必要がある場合は、[証明書の インストール] をクリックして、証明書とキーのペアを指定します。

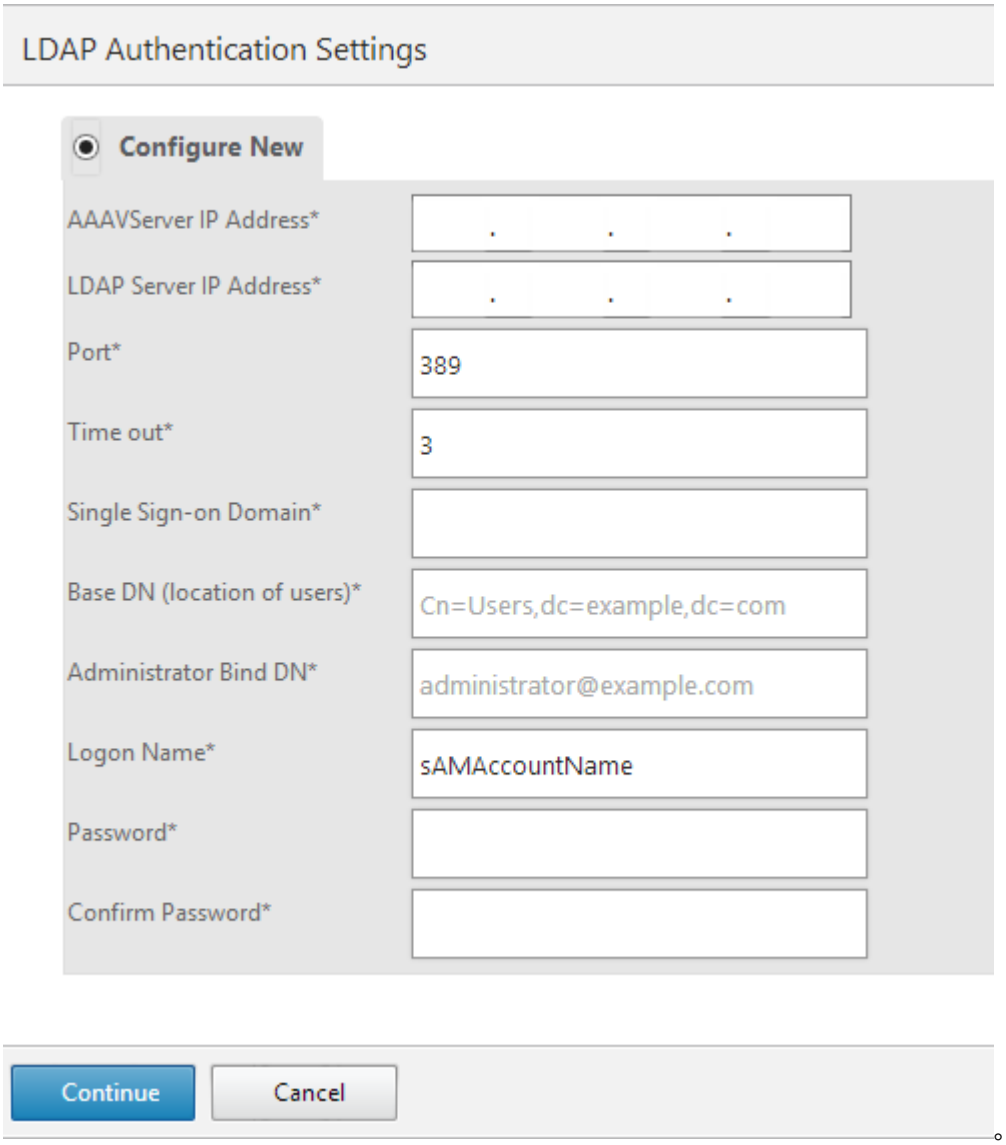


5. [続行] をクリックします。
6. [新しい **StorageZone** コントローラーの追加] ダイアログボックスで、次のパラメーターの値を指定します。
   - StorageZone・コントローラの IP アドレス： IP アドレス

- ポート：ポート番号。デフォルト値は 443 です。
- プロトコル：HTTPS または HTTP



7. [作成] をクリック してから、[完了] をクリックします。ウィザードは自動的にサービスを作成し、サービスの名前を自動生成します。
8. 手順 4.c で CIFS または SharePoint のロード・バランシングを選択した場合は、LDAP 認証設定の値を指定します。
   - Citrix ADC AAA 仮想サーバーの IP アドレス：Citrix ADC AAA 仮想サーバーの IP アドレス
   - LDAP サーバの IP アドレス：LDAP サーバの IP アドレス
   - ポート：ポート番号。デフォルト値は 389 です。
   - Timeout：タイムアウト値（分）
   - シングルサインオンドメイン-シングルサインオンドメイン名
   - ベース DN：ベースドメイン名
   - 管理者バインド DN：ドメイン名を持つ LDAP アカウント名（adminstrator@domainname.com など）
   - ログオン名-ログオン名は sAMAccountName です
   - パスワードとパスワードの確認：パスワードを入力し、パスワード



9. [続行] をクリック してから、[完了] をクリックします。

ShareFile 用の負荷分散構成を削除するには

1. ナビゲーションペインで、[トラフィック管理] をクリックします。
2. [**Citrix ShareFile**] セクションで、[**ShareFile** 構成の削除] をクリックします。

Use case 15: Configure layer 4 load balancing on the Citrix ADC appliance

December 7, 2021

The layer 4 load balancer (TCP and UDP ports) uses information provided in the networking transport

layer for routing client requests across the server groups.

When a layer 4 connection is established between a client and a server, it has a packet view of traffic exchanged between them. The layer 4 load balancer makes its routing decisions based on the address information extracted from the first few packets in the TCP stream, and doesn't inspect the packet content. Therefore, the layer 4 load balancing is also called as connection-based load balancing.

The layer 4 load balancer monitors the health of a server. Traffic is not routed to the server if it is DOWN.

The layer 4 load balancing is useful for various applications that uses TCP or UDP payloads. Such protocols exchange data as TCP payload and don't have a specific structure to follow.

To configure layer 4 load balancing using the command line interface

At the command prompt, type:

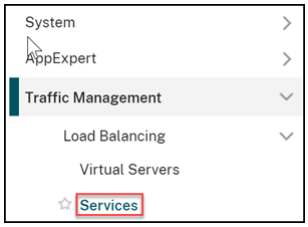
```
add service <name> <serverName> <serviceType> <port>
add lb vserver <name> <serviceType> <ip> <port>
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
add service TCPservice 192.0.2.3 TCP 1
add lb vserver TCPserver TCP 192.0.2.4 1
bind lb vserver TCPserver TCPservice
<!--NeedCopy-->
```

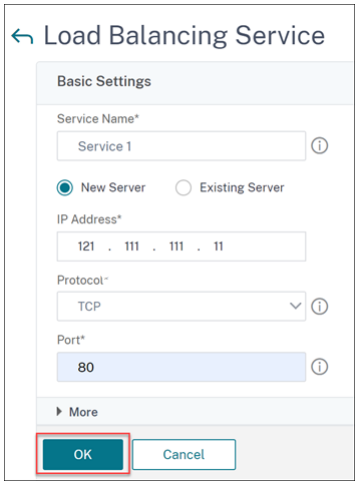
To configure layer 4 load balancing using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.



2. Click **Add** to a create a service.
3. Specify the required details in **Service Name** and **IP Address**.

4. Select either **TCP** or **UDP** in **Protocol**.
5. Click **OK**.



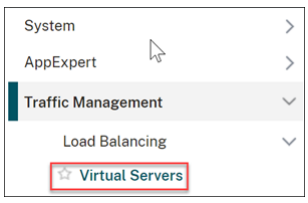
6. Click **Done**.

   A service is created.

   When you create a service using UDP as the transport layer protocol, a ping monitor (built-in monitor) is automatically bound to the service. When you create a service using TCP as the transport layer protocol, a **tcp_default** monitor is automatically bound to the service.

   For the load balancing setup, you can bind your service to a different type of monitor or multiple monitors. For advance monitoring requirements you can use the **tcp-ecv** monitor and configure the request and response messages.

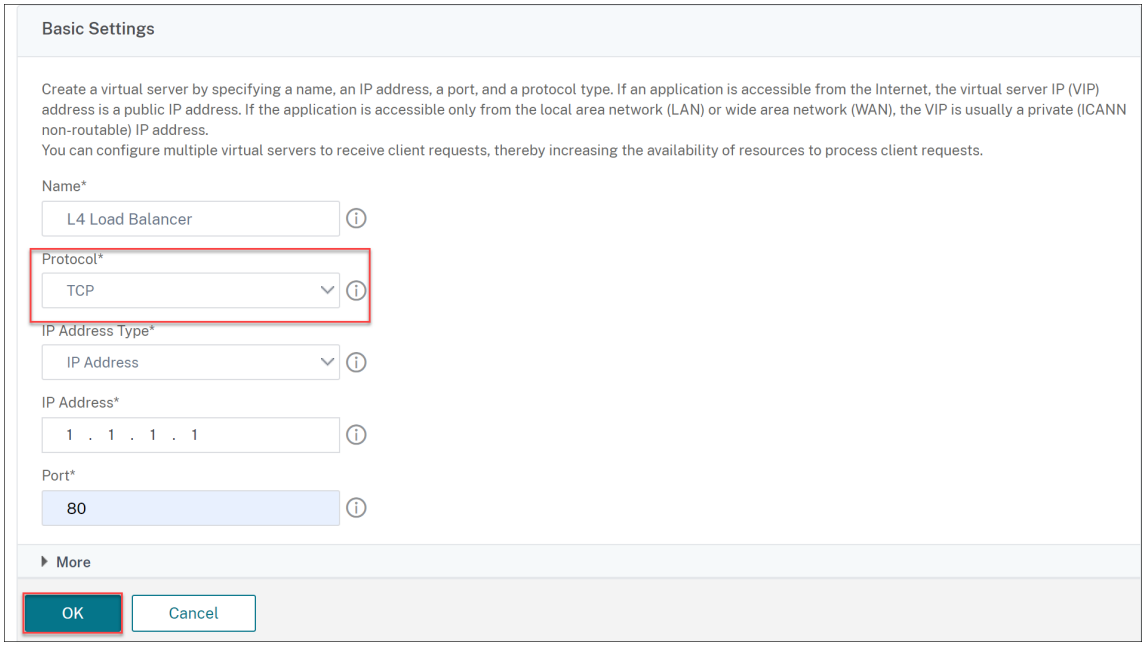
7. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.



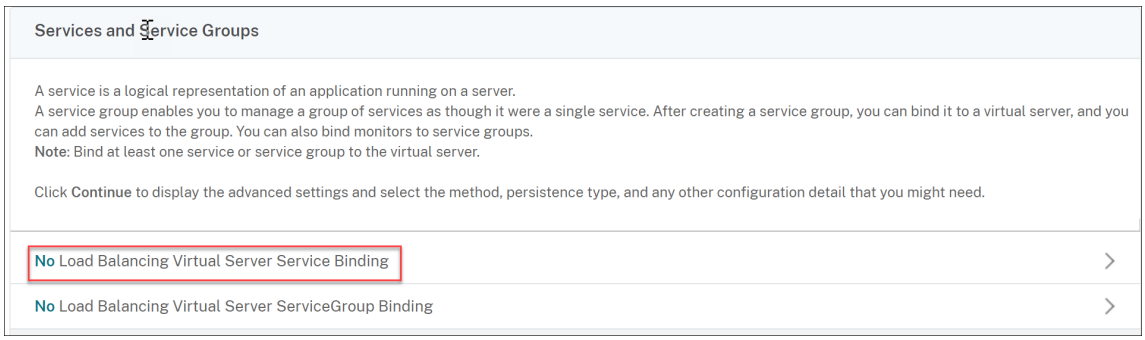
8. Click **Add** to create a new virtual server.

   When the load balancing is configured, you can connect to the load-balanced website, application, or server through the virtual server's IP address or FQDN.

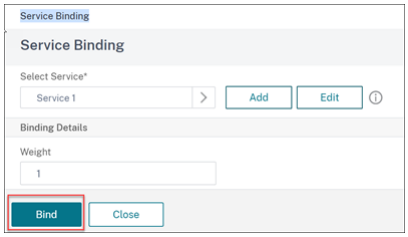
9. Specify the required details in **Name**, **IP Address Type**, and **IP Address**.
10. Select either **TCP** or **UDP** in **Protocol**.
11. Type a port number (0–1023 based on the type of service) in **Port**.
12. Click **OK**.



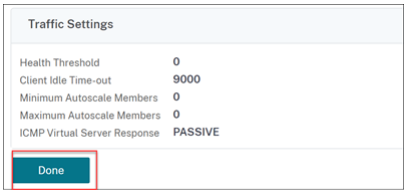
13. Click **No Load Balancing Virtual Server Service Binding** in **Services and Service Groups**.



14. In the **Service Binding** page, select **Click to Select** in **Select Service**.
15. Select the service to be bound and click **Select**.
16. Click **Bind** to bind the service to the virtual server.



17. Click **Continue**.
18. Click **Done**.



The layer 4 load balancing virtual server configuration is completed.

トラブルシューティング

October 7, 2021

構成後に負荷分散が期待どおりに機能しない場合は、一般的なツールを使用して Citrix ADC リソースにアクセスし、問題を診断できます。

ロードバランシングのトラブルシューティングに関するリソース

最適な結果を得るには、次のリソースを使用して、Citrix ADC アプライアンスのコンテンツスイッチングに関する問題のトラブルシューティングを行います。

- 最新の ns.conf ファイル
- 関連`newnslog`ファイル
- 可能であれば、アプライアンスおよび関連するクライアントに記録される Ethereal パケットトレース
- ns.log ファイル

上記のリソースに加えて、次のツールもトラブルシューティングを迅速に行えます。

- HTTP ヘッダーを表示できるブラウザアドオンツール。これは、永続性に関連する問題のトラブルシューティングに使用できます。
- Citrix ADC トレースファイル用にカスタマイズされた Wireshark アプリケーション。

ロードバランシングの問題のトラブルシューティング

- 問題

CPU 使用率が、-m MAC オプションが有効になっている仮想サーバーにバインドされているサービスにユーザーモニターがバインドされている場合、100% に達します。

- 解像度

ユーザー以外のモニターをサービスにバインドします。

- 問題

  監視用のユーザースクリプトを作成しましたが、動作していません。

解像度

スクリプト内の引数の数を確認します。制限は 512 です。512 を超える引数を持つスクリプトは、正しく動作しない可能性があります。CLI から nsumon-debug.pl スクリプトを使用して、スクリプトをデバッグします。

- 問題

多くのモニタプローブがあり、ネットワークトラフィックが不必要に増加しているようです。モニタープローブを外す方法はありますか?

解像度

モニタを無効にするか、set service コマンドの HealthMonitor パラメータの値を NO に設定することで、モニタプローブ接続をオフに設定できます。NO オプションを使用すると、アプライアンスは常に UP としてサービスを表示します。

- 問題

私はサービスのモニターを設定しましたが、接続はまだ DOWN しているサーバーに向けられています。

解像度

おそらく、モニタのプローブ間隔を短くする必要があります。Citrix ADC アプライアンスは、モニターがプローブを送信するまで、DOWN 状態を検出しません。

- 問題

モニターにバインドされたメトリックは、ローカルおよびカスタムメトリックステーブルに存在します。

解像度

メトリックがローカルメトリックテーブルから選択されている場合は、メトリック名にローカルプレフィックスを追加します。ただし、カスタムテーブルからメトリックスを選択した場合は、プレフィックスを追加する必要はありません。

- 問題

サービスに対するモニタプローブがサービスに到達していません。

解像度

サービスの接続数に制限が設定されているかどうかを確認します。[はい] の場合、monitorSkipMaxClient パラメータを ENABLED に設定して、モニタとプローブの接続をこの制限から除外します。

- 問題

私はサーバーに ping することができますが、サービスの状態は常に DOWN として表示されます。

解像度

構成されているモニタのタイプを確認します。たとえば、サーバーが SSL 用に構成されておらず、HTTPS モニターを使用する場合、サービスの状態は DOWN としてマークされます。この場合、TCP モニタを使用すると、サービスの状態を UP に変更する必要があります。

- 問題

  負荷モニターの重みを設定しても、サービスの状態の決定には役立ちません。

  解像度

  ロードモニタは、サービスの状態を判断できません。したがって、負荷モニタに重みを設定することは不適切です。

- 問題

  サービスが安定していません。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - 正しいサーバーがサービスにバインドされていることを確認します。
  - サービスにバインドされているモニタのタイプを確認します。
  - モニタの障害の原因を確認します。[サービス] ページからサービスを開き、[サービスの構成] ダイアログボックスの [Monitors] タブで、モニタのプローブの数、障害、および最後の応答ステータスの詳細を確認できます。詳細を表示するには、構成されているモニタをクリックします。
  - カスタムモニタの場合は、TCP または ping モニタをサービスにバインドし、モニタの状態を確認します。これで問題が解決した場合は、カスタムモニタに問題があり、モニタをさらに調査する必要があります。
  - Citrix ADC アプライアンスでパケットトレースを記録し、監視プローブとサーバーの応答を確認して詳細な調査を行うことができます。

- 問題

  仮想 IP（VIP）アドレスが安定していないか、ステータスが DOWN と表示されます。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - ロードバランシング機能がライセンスされていることを確認します。
  - 機能が有効になっていることを確認します。
  - 適切なサービスが仮想サーバにバインドされていることを確認します。
  - VIP アドレスのステータスが DOWN と表示される場合は、管理者がサービスを有効にしていることを確認します。そうでない場合、サービスのステータスは Out-of-Service である必要があります。このような場合は、サービスを有効にし、問題が解決されたかどうかを確認する必要があります。
  - 仮想サーバーにバインドされているサービスを確認し、サービスが安定していない問題に関するトラブルシューティング手順を完了します。
  - VIP アドレスが安定していない場合、仮想サーバーにバインドされているすべてのサービスが失敗する必要があります。したがって、すべてのサービスが同時に失敗しているかどうかを確認します。その場合は、Citrix ADC アプライアンスとサーバーの間にネットワークの問題があります。

- 問題

  サイトの負荷分散が不均一です。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - アプライアンスに構成されている負荷分散方式を確認します。
  - サービスに関連付けられている重みが期待どおりであることを確認します。
  - ロードバランシング方式がラウンドロビン以外の場合は、newnslogファイルにログインしているサーバへの接続数を確認します。次のコマンドを実行して、newnslogファイル内の番号を確認できます。

    ```
    ## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
    ```

    特定の仮想サーバのサービスを確認し、応答時間、オープン確立接続（OE）、要求数、永続要求、および永続レート（P）をチェックして、問題をさらにトラブルシューティングします。
  - ロードバランシング方式がラウンドロビンの場合は、前の手順で説明した永続リクエストを確認します。さらに、サービスが安定していないかどうかを確認します。そうでない場合は、サービスが不安定な問題について記載されているトラブルシューティング手順を完了します。
  - アプライアンスで永続性が構成されているかどうかを確認します。
  - 安定していないサービスがあるかどうかを確認します。「はい」の場合は、「サービスが安定していない」という問題について記載されているトラブルシューティング手順を完了します。

- 問題

  サービスのステータスは DOWN と表示されます。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - SNIP アドレスが構成されているかどうかを確認します。
  - 適切なモニタがサービスにバインドされていることを確認します。
  - カスタムモニタがサービスにバインドされている場合は、TCP モニタまたは ping モニタをサービスにバインドし、モニタの状態を確認します。これで問題が解決した場合は、カスタムモニタに問題があり、モニタをさらに調査する必要があります。
  - 別のサブネットにあるサーバーのサービスのステータスが DOWN と表示されているかどうかを確認します。[はい] の場合は、[サブネット IP] (USNIP) で問題が解決するかどうかを確認します。これは、MIP アドレスがサーバと通信できないことが原因であるためです。

- 問題

  応答時間に問題があります。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

- 次のコマンドを実行して、サービスの統計情報からサーバの応答時間を確認します。

  ```
  ## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
  ```

- サービスが安定していないか、サービスのステータスが DOWN の問題として表示されるかを確認します。

- 問題

  サーバーの 1 つが、他の負荷分散サーバーよりも多くの要求を処理しています。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - ロードバランシング方式を確認します。ラウンドロビン方式を使用して、サーバの負荷に関係なく、クライアント要求を均等に分散します。
  - ロードバランシング設定に対してパーシステンスが有効になっているかどうかを確認します。永続性が有効になっている場合、特に永続セッションが長い場合、特定のサーバーがそのセッションを維持するためにより重い負荷がかかっている可能性があります。
  - 各サービスに重みが割り当てられているかどうかを確認します。適切なウェイトを割り当てると、適切な負荷分散に役立ちます。

- 問題

  特定の負荷分散サーバーへの接続が停止します。たとえば、1 つの Outlook サーバーへのすべての接続が停止することがあります。

  解像度

  次のコンポーネントのトラブルシューティングを検討してください。

  - ロードバランシング方式を確認します。ラウンドロビンである場合は、メソッドを最小接続に変更することを検討してください。
  - モニターのタイムアウト期間を短縮することを検討してください。タイムアウト期間を短くすると、サービスをより早く DOWN（ダウン）としてマークできます。これにより、機能しているサーバにトラフィックを誘導するのに役立ちます。
  - 接続が長時間停止すると、サージキューが構築される可能性があります。サーバーの負荷が急増するのを避けるために、サージキューをフラッシュすることを検討してください。
  - サーバが最大レベルで動作している場合は、パフォーマンスを向上させるために新しいサーバを追加することを検討してください。

- 問題

  負荷分散の最小接続方法が構成されている場合でも、ほとんどの接続は特定のサーバーに送信されます。

  解像度

  パーシステンスが設定されていて、タイプがソース IP であるかどうかを判断します。最小接続方式でも送信元 IP パーシステンスが設定されている場合、要求は特定のサーバに送信されます。サーバーの IP アドレスは、

セッション情報を維持するために必要です。HTTP Cookie ベースの永続性を使用することを検討してください。

- トラブルシューティングのヒントその他の問題については
  、上記に記載されていない問題のトラブルシューティングを行うために、次のヒントを検討してください。
  - 複数の負荷モニターが 1 つのサービスにバインドされている場合、サービスの負荷は、そのサービスにバインドされている負荷モニターのすべての値の合計になります。負荷分散を正しく動作させるには、同じ一連のモニターをすべてのサービスにバインドする必要があります。
  - サービスにバインドされた負荷監視を無効にし、サービスが仮想サーバーにバインドされている場合、仮想サーバーは負荷分散にラウンドロビン方式を使用します。
  - 負荷分散方式が CUSTOMLOAD で、サービスのステータスが UP である仮想サーバーにサービスをバインドすると、仮想サーバーは負荷分散に初期ラウンドロビン方式を使用します。サービスにカスタムロードモニターがない場合、またはカスタムロードモニターの少なくとも 1 つのステータスが UP でない場合は、引き続きラウンドロビン状態になります。
  - 負荷分散方式が CUSTOMLOAD である仮想サーバーにバインドされているすべてのサービスでは、サービスには負荷監視がバインドされている必要があります。
  - CUSTOMLOAD ロードバランシング方式は、スタートアップラウンドロビンにも従います。
  - メトリックベースのバインディングを無効にし、これが最後のアクティブなメトリックである場合、特定の仮想サーバーはロードバランシングにラウンドロビン方式を使用します。メトリックしきい値をゼロに設定すると、メトリックが無効になります。
  - モニターにバインドされたメトリックがしきい値を超えると、その特定のサービスは負荷分散の対象にはなりません。すべてのサービスがしきい値に達した場合、仮想サーバーはロードバランシングにラウンドロビン方式を使用し、エラーメッセージ「5xx-サーバービジーエラー」が表示されます。
  - カスタムテーブルから最大 10 個のメトリックをモニターにバインドできます。
  - OID はスカラー変数でなければなりません。
  - ロードバランシングを成功させるには、間隔をできるだけ低くする必要があります。間隔が大きい場合は、負荷値を取得する期間が長くなります。その結果、不適切な値を使用して負荷分散が行われます。
  - ユーザーはローカルテーブルを変更できません。

負荷分散に関する **FAQ**

October 7, 2021

Citrix ADC アプライアンスで作成できるさまざまな負荷分散ポリシーについて

Citrix ADC アプライアンスでは、次の種類の負荷分散ポリシーを作成できます。

- 最小接続数
- ラウンドロビン

- 応答時間の最短短縮
- 最小帯域幅
- 最小パケット
- URL ハッシュ
- ドメイン名のハッシュ化
- 送信元 IP アドレスのハッシュ
- 宛先 IP アドレスのハッシュ
- 送信元 IP：宛先 IP ハッシュ
- トークン
- LRTM

Citrix ADC アプライアンスを使用して負荷分散を実装することで、**Web** ファームのセキュリティを実現できますか**?**

はい。Citrix ADC アプライアンスを使用して負荷分散を実装することで、Web ファームのセキュリティを実現できます。Citrix ADC アプライアンスでは、負荷分散機能の次のオプションを実装できます。

- IP アドレスの非表示: セキュリティ上の理由と IP アドレスの保存のために、実際のサーバーをプライベート IP アドレス空間にインストールできます。Citrix ADC アプライアンスはサーバーに代わって要求を受け入れるため、このプロセスはエンドユーザーに対して透過的です。アドレス隠蔽モードでは、アプライアンスによって 2 つのネットワークが完全に隔離されます。したがって、クライアントは、そのサービス用のアプライアンス上の異なる VIP を介して、FTP や Telnet サーバーなどのプライベートサブネットで実行されているサービスにアクセスできます。
- ポートマッピング: セキュリティ上の理由から、実際の TCP サービスを非標準ポートでホストできるようにします。このプロセスは、Citrix ADC アプライアンスが標準のアドバタイズ IP アドレスとポート番号でサーバーに代わって要求を受け入れるため、エンドユーザーには透過的です。

Citrix ADC アプライアンスの負荷分散に使用できるさまざまなデバイスは何ですか

Citrix ADC アプライアンスを使用して、次のデバイスを負荷分散できます。

- サーバーファーム
- キャッシュまたはリバースプロキシ
- ファイアウォールデバイス
- 侵入検知システム
- SSL オフロードデバイス
- 圧縮装置
- コンテンツ検査サーバー

Web サイトの負荷分散機能を実装するのはなぜですか

Web サイトの負荷分散機能を実装すると、次の利点があります。

- 応答時間の短縮: Web サイトの負荷分散機能を実装する場合、大きな利点の 1 つは、ロード時に期待できるブーストです。2 つ以上のサーバーが Web トラフィックの負荷を共有している場合、各サーバーのトラフィック負荷は 1 台のサーバーだけよりも少なくなります。これは、クライアント要求を満たすために利用可能なリソースがさらに存在することを意味します。これにより、ウェブサイトがより高速になります。
- 冗長性: 負荷分散機能を実装すると、少しの冗長性が導入されます。たとえば、ウェブサイトが 3 つのサーバー間でバランスが取れていて、そのうちの 1 つがまったく応答しない場合、他の 2 つは実行し続けることができ、ウェブサイトの訪問者もダウンタイムに気付かない。負荷分散ソリューションは、利用できないバックエンドサーバーへのトラフィックの送信を直ちに停止します。

リンクロードバランシング **(LLB)** の **Mac** ベースフォワーディング **(MBF)** オプションを無効にする必要があるのはなぜですか**?**

- MBF オプションを有効にすると、Citrix ADC アプライアンスは、クライアントからの着信トラフィックと同じクライアントへの発信トラフィックが同じアップストリームルーターを通過すると見なします。ただし、LLB 機能では、リターントラフィックに最適なパスを選択する必要があります。
- MBF オプションを有効にすると、着信クライアントトラフィックを転送したルータを介して発信トラフィックを送信することによって、このトポロジ設計が中断されます。

ネットワーク

October 7, 2021

以下のトピックでは、Citrix ADC アプライアンスのさまざまなネットワークコンポーネントを構成するための概念的なリファレンスおよび手順について説明します。

IP アドレッシング	Citrix ADC が所有するさまざまな IP アドレスの種類と、それらの作成、カスタマイズ、削除の方法について学習します。
インターフェイス	開始するために実行する必要がある基本的なネットワーク構成の一部を構成します。
アクセスコントロールリスト（ACL）	さまざまなタイプのアクセスコントロールリストと、それらの作成、カスタマイズ、および削除方法を設定します。

IP ルーティング	Citrix ADC アプライアンスのルーティング機能（静的および動的の両方）を学習および構成します。
インターネットプロトコルバージョン 6 (IPv6)	Citrix ADC アプライアンスが IPv6 をどのようにサポートするかについて説明します。
トラフィックドメイン	さまざまなアプリケーションのネットワークトラフィックをセグメント化するために、トラフィックドメインを学習および構成します。
VXLAN	データセンターにおけるスケーラビリティのニーズを満たすように、VXLAN を学習および構成します。

IP アドレッシング

October 7, 2021

Citrix ADC アプライアンスを構成する前に、NSIP アドレス（管理 IP アドレスとも呼ばれる）を割り当てる必要があります。また、サーバーを抽象化し、サーバーとの接続を確立するために、Citrix ADC が所有する他の IP アドレスを作成することもできます。このタイプの構成では、アプライアンスは抽象化されたサーバーのプロキシとして機能します。ネットワークアドレス変換 (INAT および RNAT) を使用してプロキシ接続を行うこともできます。接続をプロキシする場合、アプライアンスはブリッジング（レイヤ 2）デバイスまたはパケット転送（レイヤ 3）デバイスとして動作します。パケット転送を効率化するために、スタティック ARP エントリを設定できます。IPv6 の場合は、ネイバー検出（ND）を設定できます。

Citrix ADC 所有の IP アドレスの構成

October 7, 2021

Citrix ADC が所有する IP アドレス（NSIP アドレス、仮想 IP アドレス（VIP）、サブネット IP アドレス（SNIP）、、およびグローバルサーバー負荷分散サイト IP アドレス（GSLBIP））は、Citrix ADC アプライアンスにのみ存在します。NSIP は、ネットワーク上の Citrix ADC を一意に識別し、アプライアンスへのアクセスを提供します。VIP は、クライアントが要求を送信するパブリック IP アドレスです。Citrix ADC は VIP でクライアント接続を終了し、サーバーとの接続を開始します。この新しい接続では、サーバに転送されるパケットの送信元 IP アドレスとして SNIP または MIP が使用されます。地理的に分散している複数のデータセンターがある場合、各データセンターは固有の GSLBIP によって識別できます。管理アプリケーションへのアクセスを提供するために、Citrix ADC が所有する IP アドレスを構成できます。

NSIP アドレスの構成

October 7, 2021

NSIP アドレスは、管理目的で Citrix ADC アプライアンスにアクセスする IP アドレスです。アプライアンスは、管理 IP アドレスとも呼ばれる NSIP を 1 つだけ持つことができます。Citrix ADC を初めて構成するときは、この IP アドレスを追加する必要があります。NSIP アドレスは削除できません。セキュリティ上の理由から、NSIP は組織の LAN 上のルーティング不可能な IP アドレスである必要があります。

このアドレスを変更する場合は、Citrix ADC アプライアンスを再起動する必要があります。新しい NSIP アドレスのサブネットアドレスが前のサブネットアドレスと異なる場合は、新しい NSIP アドレスが LAN 上の他のネットワークから到達可能になるように、このサブネットのデフォルトルートを追加する必要があります。

> 重要
>
> NSIP アドレスの構成は必須です。

Citrix ADC アプライアンスの NSIP アドレスの変更は、次のタスクで構成されます。

- NSIP アドレスを変更します。
- NSIP アドレスのサブネット・アドレスにデフォルト・ルートが存在しない場合は、追加します。
- 構成を保存します。
- アプライアンスを再起動します。

コマンドラインプロシージャ

CLI を使用して NSIP アドレスを変更するには、次の手順で行います。

コマンドプロンプトで入力します。

- **set ns config -IPAddress** <ip_addr> **-netmask** <netmask>
- **show ns config**

CLI を使用してデフォルトルートを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add route 0 0** <gateway IP address>
- **show route**

CLI を使用して設定を保存するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **save config**

CLI を使用して Citrix ADC アプライアンスを再起動するには:

コマンドプロンプトで入力します。

- リブート

GUI の手順

GUI を使用して NSIP アドレスを構成するには、次の手順で行います。

1. [設定] ページの右上隅にある歯車アイコンをクリックします。
2. [**NSIP アドレス **] ペインをクリックします。
3. [**NSIP** アドレス] ページで、次のパラメータを設定し、[完了] をクリックします。
   - NSIP アドレス
   - ネットマスク

GUI を使用してデフォルトルートを追加するには、次の手順を実行します。

[システム] > [** ネットワーク]** > [ルート] に移動し、[基本] タブで、次のパラメータ設定でデフォルトルートを追加し、[作成] をクリックします。

- ネットワーク (ゼロに設定)
- ネットマスク（ゼロに設定）
- Gateway（ゲートウェイの IP アドレス）

GUI を使用して Citrix ADC を再起動するには:

1. **[**システム] ノードの [システム情報**]** タブページで、**[**再起動**]** をクリックします。
2. 再起動を求めるメッセージが表示されたら、[**Save Configuration**] を選択して、設定が失われないようにします。

構成例

次の例では、Citrix ADC アプライアンスの NSIP アドレスが 192.0.2.90 に変更され、以前の NSIP アドレスとは異なるサブネット・アドレス（192.0.2.0/24）が割り当てられています。したがって、新しい NSIP アドレスが他のネットワークから到達可能になるように、このサブネットにデフォルトルートが追加されます。

```
1  > set nsconfig -ipAddress 192.0.2.90 -netmask 255.255.255.0
2
3   Warning: The configuration must be saved and the system rebooted for
       these settings to take effect
4  > add route 0 0 192.0.2.1
5
6   Warning: The configuration must be saved and the system rebooted for
       these settings to take effect
7  > save config
8
9   Done
10 > reboot
```


仮想 IP（VIP）アドレスの設定と管理

October 7, 2021

Citrix ADC 初期構成では、仮想サーバー IP（VIP）アドレスの構成は必須ではありません。ロードバランシングを構成するときは、VIP アドレスを仮想サーバーに割り当てます。

ロードバランシングの設定の詳細については、「負荷分散」を参照してください。

場合によっては、VIP 属性をカスタマイズするか、VIP アドレスを有効または無効にする必要があります。通常、VIP アドレスは仮想サーバーに関連付けられ、一部の VIP 属性は仮想サーバーの要件を満たすようにカスタマイズされます。ARP 属性と ICMP 属性を使用して、同じブロードキャストドメインに存在する複数の Citrix ADC アプライアンス上で同じ仮想サーバーをホストできます。VIP（または任意の IP アドレス）を追加すると、アプライアンスは ARP 要求を送信し、応答します。無効にできる唯一の Citrix ADC が所有する IP アドレスは、VIP です。VIP アドレスを無効にすると、そのアドレスを使用する仮想サーバがダウンし、ARP、ICMP、または L4 サービスリクエストに応答しません。VIP アドレスを一度に 1 つずつ作成する代わりに、連続した VIP アドレスの範囲を指定できます。

CLI を使用して VIP アドレスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

例:

```
1 > add ns ip 10.102.29.59 255.255.255.0 -type VIP
2  Done
3 <!--NeedCopy-->
```

CLI を使用して VIP アドレスの範囲を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

例:

```
1 > add ns ip 10.102.29.[60-64] 255.255.255.0 -type VIP
2 ip ”10.102.29.60” added
3 ip ”10.102.29.61” added
4 ip ”10.102.29.62” added
5 ip ”10.102.29.63” added
```



```
ip ”10.102.29.64” added
 Done
<!--NeedCopy-->
```

CLI を使用して IPv4 VIP アドレスを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力して、VIP を有効または無効にし、構成を確認します。

- enable ns ip <IPAddress>
- show ns ip <IPAddress>
- disable ns ip <IPAddress>
- show ns ip <IPAddress>

例:

```
> enable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Enabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled
 Done
> disable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Disabled
```



```
30          arp: Enabled
31          icmp: Enabled
32          vserver: Enabled
33          management access: Disabled
34            telnet: Disabled
35            ftp: Disabled
36            ssh: Disabled
37            gui: Disabled
38            snmp: Disabled
39          Restrict access: Disabled
40          dynamic routing: Disabled
41          hostroute: Disabled
42
43   Done
44  <!--NeedCopy-->
```

GUI を使用して VIP アドレスを設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [IP] > [**IPV4**] に移動し、新しい IP アドレスを追加するか、既存のアドレスを編集します。

GUI を使用して VIP アドレスの範囲を作成するには、次の手順を実行します。

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. [アクション] リストで、[範囲の追加] を選択します。

GUI を使用して VIP アドレスを有効または無効にするには、次の手順を実行します。

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. 次のいずれかを行います:
   - VIP アドレスを選択します。
   - **Ctrl** キーを押しながら、複数のサーバアドレスエントリを選択します。
   - **Shift** キーを押しながら、サーバアドレスエントリの範囲を選択します。
   - ヘッダー行の左側にあるチェックボックスを選択して、すべてのアドレスを選択します。
3. ［操作］リストから［無効］または［有効］を選択します。

TTL 更新による **UDP** 負荷分散セットアップでの **Citrix ADC** アプライアンスの検出

次の表は、Citrix ADC アプライアンスが受信したパケットの TTL 値をさまざまな機能で処理する方法を示しています。

機能	TTL 値
仮想サーバー	要求をバックエンドサーバーに転送する場合、TTL は 255 に設定されます。TTL は、応答をクライアントに転送するときに 1 ずつデクリメントされます。
L2 モード	TTL は変更されません。
L3 モード	TTL は 255 に設定されています。
INAT	要求をバックエンドサーバーに転送する場合、TTL は 255 に設定されます。TTL は、応答をクライアントに転送するときに 1 ずつデクリメントされます。

監視アプリケーションを実行する企業やシナリオによっては、負荷分散設定の Citrix ADC アプライアンスが、traceroute のホップの 1 つとして検出される必要があります。負荷分散設定の Citrix ADC アプライアンスは、traceroute では検出されません。これは、デフォルトでは、要求をバックエンドサーバーに転送するときに TTL 値をデクリメントするのではなく 255 に設定するためです。

この要件を満たすために、VIP アドレスの デクリメント **TTL** パラメータを使用することができます。このパラメータは、この VIP を使用するすべての UDP 仮想サーバに適用されます。

VIP の デクリメント **TTL** パラメーターを有効にすると、Citrix ADC アプライアンスは、この VIP を使用する UDP 仮想サーバーで受信されるリクエストを転送するときに、TTL 値を 255 に設定するのではなく、TTL 値を 1 ずつ減らします。

traceroute データを使用する監視アプリケーションは、UDP 負荷分散セットアップの Citrix ADC アプライアンスの存在を検出できるようになりました。

はじめに

負荷分散セットアップのトレースルートで Citrix ADC アプライアンスを検出するように構成する前に、次の点に注意してください。

- デクリメント TTL パラメータは、UDP ロードバランシング仮想サーバに対してのみサポートされます。
- リクリメント TTL パラメータは、IPv4 VIP および IPv6 VIP（VIP6）アドレスでサポートされます。
- デクリメント TTL パラメータは、スタンドアロンの Citrix ADC アプライアンス、高可用性（HA）およびクラスタセットアップでサポートされています。

構成の手順

UDP 負荷分散セットアップのトレースルートで Citrix ADC アプライアンスを検出するように構成するには、次の作業を行います。

- UDP ロードバランシング設定を作成する

- VIP アドレスの減少 TTL パラメータを有効にします。

CLI の手順

CLI を使用して VIP アドレスの減少 TTL オプションを有効にするには、次の手順を実行します。

- VIP アドレスの追加中に VIP アドレスの減少 TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。
  - **add ns ip** <ip> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>
- 既存の VIP アドレスの減少 TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。
  - **set ns ip** <ip> <mask> -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>

CLI を使用して VIP6 アドレスの減少 TTL オプションを有効にするには、次の手順を実行します。

- VIP6 アドレスの追加中に VIP6 アドレスの減少 TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。
  - **add ns ip6** <IP6/prefix> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6/prefix>
- 既存の VIP6 アドレスの減少 TTL オプションを有効にするには、コマンドプロンプトで次のように入力します。
  - **set ns ip6** <ip6/prefix> <mask> -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6 address>

```
1      > add ns ip 203.0.113.30 -type VIP -decrementTTL ENABLED
2       Done
3
4      > add ns ip6  2001:DB8:5001::30 -type VIP -decrementTTL ENABLED
5       Done
6  <!--NeedCopy-->
```

GUI の手順

GUI を使用して VIP アドレスの減少 TTL オプションを有効にするには、次の手順を実行します。

[システム] > [ネットワーク] > [**IP**] > [**IPv4**] に移動し、新しい VIP アドレスを追加したり、既存のアドレスを編集したりするときに [デクリメント **TTL**] パラメータを有効にします。

GUI を使用して VIP6 アドレスの TTL デクリメントオプションを有効にするには、次の手順を実行します。

[システム] > [ネットワーク] > [**IP**] > [**IPv6**] に移動し、新しい VIP6 アドレスを追加するか、既存のアドレスを編集するときに、[減少 **TTL**] パラメータを有効にします。

仮想 **IP** アドレス（**VIP**）に対する **ARP** 応答抑制の設定

October 7, 2021

Citrix ADC アプライアンスは、その VIP に関連付けられた仮想サーバーの状態に基づいて、仮想 IP（VIP）アドレスに対する ARP 要求に応答するか応答しないかを構成できます。

たとえば、タイプ HTTP の仮想サーバー V1 およびタイプ HTTP の V2 が、Citrix ADC アプライアンス上で VIP アドレス 10.102.29.45 を共有している場合、V1 と V2 の両方が DOWN 状態の場合、VIP 10.102.29.45 に対する ARP 要求に応答しないようにアプライアンスを構成できます。

仮想 IP アドレスの ARP 応答抑制を設定するには、次の 3 つのオプションを使用できます。

- なし。Citrix ADC アプライアンスは、アドレスに関連付けられた仮想サーバーの状態に関係なく、VIP アドレスの ARP 要求に応答します。
- **ONE VSERVER**。Citrix ADC アプライアンスは、関連する仮想サーバーの少なくとも 1 つが UP 状態の場合、VIP アドレスの ARP 要求に応答します。
- **ALL VSERVER**。Citrix ADC アプライアンスは、関連するすべての仮想サーバーが UP 状態の場合、VIP アドレスの ARP 要求に応答します。

次の表に、2 つの仮想サーバーで構成された VIP の Citrix ADC アプライアンスの動作例を示します。

VIP の関連付けられた仮想サーバー	状態 1	状態 2	状態 3	状態 4
NONE				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP に対する ARP 要求に応答しますか。	はい	はい	はい	はい
ONE VSERVER				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP に対する ARP 要求に応答しますか。	はい	はい	はい	いいえ
ALL VSERVER				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN

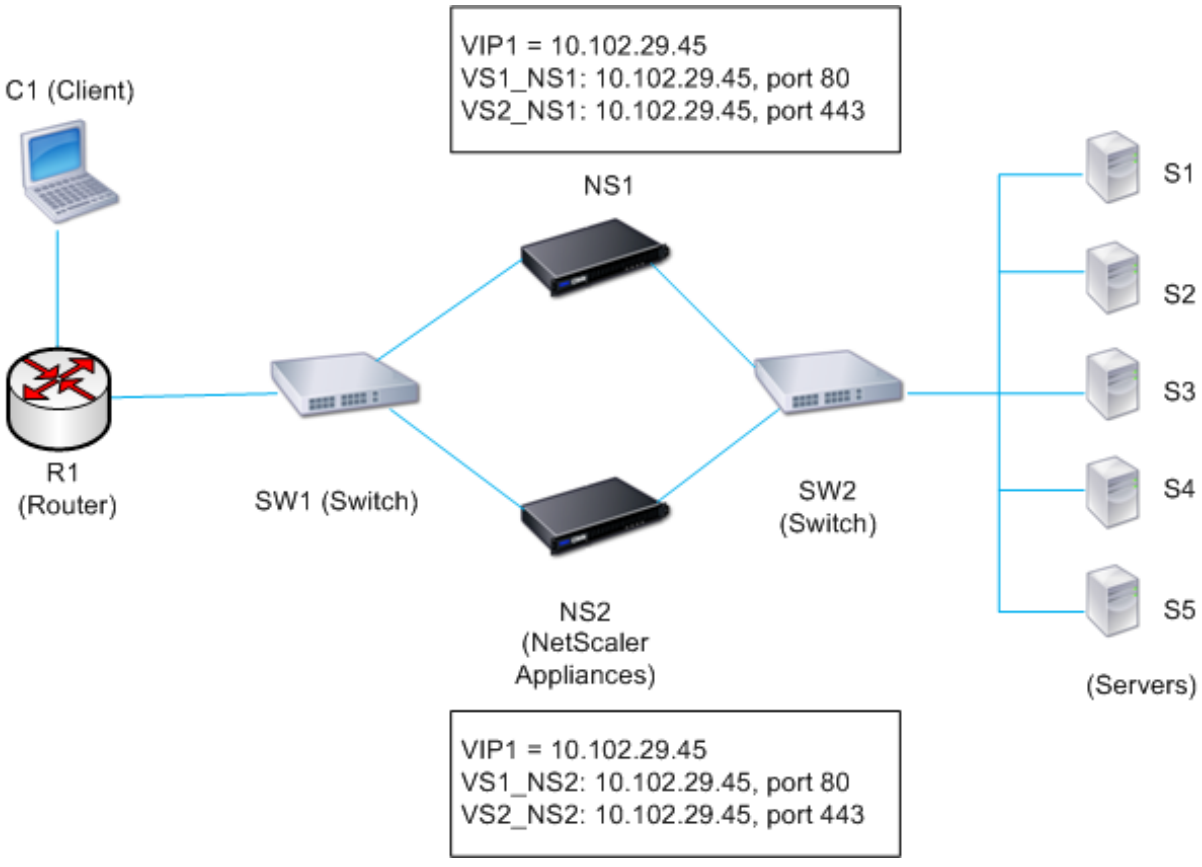
VIP の関連付けられた仮想サーバー	状態 1	状態 2	状態 3	状態 4
この VIP に対する ARP 要求に応答しますか。	はい	いいえ	いいえ	いいえ

たとえば、V1 と V2 の 2 つの仮想サーバーのパフォーマンスをテストしたいとします。V1 と V2 は同じ VIP アドレスを持っていてタイプが異なり、それぞれが Citrix ADC アプライアンス NS1 と NS2 上で構成されています。共有 VIP アドレス *VIP1* を呼び出してみましょう。

V1 は、サーバー S1、S2、および S3 の負荷を分散します。V2 は、サーバ S4 と S5 のロードバランシングを行います。

NS1 と NS2 の両方で、VIP1 の場合、ARP 抑制パラメータは ALL_VSERVER に設定されます。NS1 で V1 と V2 のパフォーマンスをテストする場合は、NS2 で V1 と V2 を手動で無効にして、NS2 が VIP1 の ARP 要求に応答しないようにする必要があります。

図 1:



実行フローは次のとおりです。

1. クライアント C1 は V1 に要求を送信します。要求は R1 に到達します。

2. R1 には V1 の IP アドレス（VIP1）に対する APR エントリがないため、R1 は VIP1 の ARP 要求をブロードキャストします。
3. NS1 は、送信元 MAC アドレス MAC1 および送信元 IP アドレス VIP1 で応答します。NS2 は ARP 要求に応答しません。
4. SW1 は ARP 応答から VIP1 のポートを学習し、ブリッジテーブルを更新し、R1 は MAC1 および VIP1 を使用して ARP エントリを更新します。
5. R1 は、NS1 上のアドレス VIP1 にパケットを転送します。
6. NS1 の負荷分散アルゴリズムはサーバー S2 を選択し、NS1 はその SNIP アドレスの 1 つと S2 の間の接続を開きます。S2 がクライアントに応答を送信すると、応答は同じパスで返されます。
7. ここで、NS2 で V1 と V2 のパフォーマンスをテストし、NS2 で V1 と V2 を有効にし、NS1 で V1 と V2 のパフォーマンスを無効にします。NS2 が VIP1 の ARP メッセージをブロードキャストするようになりました。メッセージでは、MAC2 が送信元 MAC アドレス、VIP1 が送信元 IP アドレスです。
8. SW1 は ARP ブロードキャストから MAC2 に到達するためのポート番号を知り、ブリッジテーブルを更新して VIP1 に対する後続のクライアント要求を NS2 に送信します。R1 は ARP テーブルを更新します。
9. ここで、R1 の ARP テーブルで VIP1 の ARP エントリがタイムアウトし、クライアント C1 が V1 の要求を送信したとします。R1 には VIP1 の APR エントリがないため、VIP1 の ARP 要求がブロードキャストされます。
10. NS2 は、送信元 MAC アドレスと VIP1 を送信元 IP アドレスとして応答します。NS1 は ARP 要求に応答しません。

CLI を使用して ARP 応答抑制を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set ns ip** -**arpResponse** <arpResponse>]
- **sh ns ip** <IPAddress>

例**:**

```
1 > set ns ip 10.102.29.96 -arpResponse ALL_VSERVERS
2  Done
3 <!--NeedCopy-->
```

GUI を使用して ARP 応答抑制を設定するには、次の手順を実行します。

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. IP アドレスのエントリを開き、ARP レスポンスのタイプを選択します。

サブネット **IP** アドレス（**SNIP**）の設定

October 7, 2021

サブネット IP アドレス（SNIP）は、Citrix ADC が所有する IP アドレスです。この IP アドレスは、Citrix ADC がサーバーと通信するために使用されます。

Citrix ADC は、サブネット IP アドレスを、サーバーへのクライアント接続をプロキシするための送信元 IP アドレスとして使用します。また、ダイナミックルーティングプロトコルに関連するパケットなどの独自のパケットを生成する場合や、モニタプローブを送信してサーバの健全性をチェックする場合にもサブネット IP アドレスを使用します。ネットワークトポロジによっては、シナリオごとに 1 つ以上の SNIP を構成しなければならない場合があります。

Citrix ADC で SNIP アドレスを構成するには、SNIP アドレスを追加し、グローバル USNIP（サブネット IP を使用）モードを有効にします。SNIP を 1 つずつ作成する代わりに、連続した SNIP の範囲を指定することもできます。

CLI を使用して SNIP アドレスを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

例:

```
> add ns ip 10.102.29.203 255.255.255.0 -type SNIP
 Done
<!--NeedCopy-->
```

CLI を使用して SNIP アドレスの範囲を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

例:

```
> add ns ip 10.102.29.[205-209] 255.255.255.0 -type SNIP
ip "10.102.29.205" added
ip "10.102.29.206" added
ip "10.102.29.207" added
ip "10.102.29.208" added
ip "10.102.29.209" added
 Done
<!--NeedCopy-->
```

CLI を使用して USNIP モードを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- enable ns modeUSNIP
- disable ns modeUSNIP

GUI を使用して SNIP アドレスを設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [IP] > [IPV4] に移動し、新しい SNIP アドレスを追加するか、既存のアドレスを編集します。

GUI を使用して SNIP アドレスの範囲を作成するには、次の手順を実行します。

1. System > Network > IPs > IPV4s に移動します。
2. [アクション] リストで、[範囲の追加] を選択します。

CLI を使用して USNIP モードを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- enable ns mode USNIP
- disable ns mode USNIP

GUI を使用して USNIP モードを有効または無効にするには、次の手順を実行します。

1. [システム] > [設定] に移動し、[モードと機能] で [モードの変更] をクリックします。
2. [サブネット IP を使用] オプションをオンまたはオフにします。

直接接続されたサーバーサブネットでの **SNIP** の使用

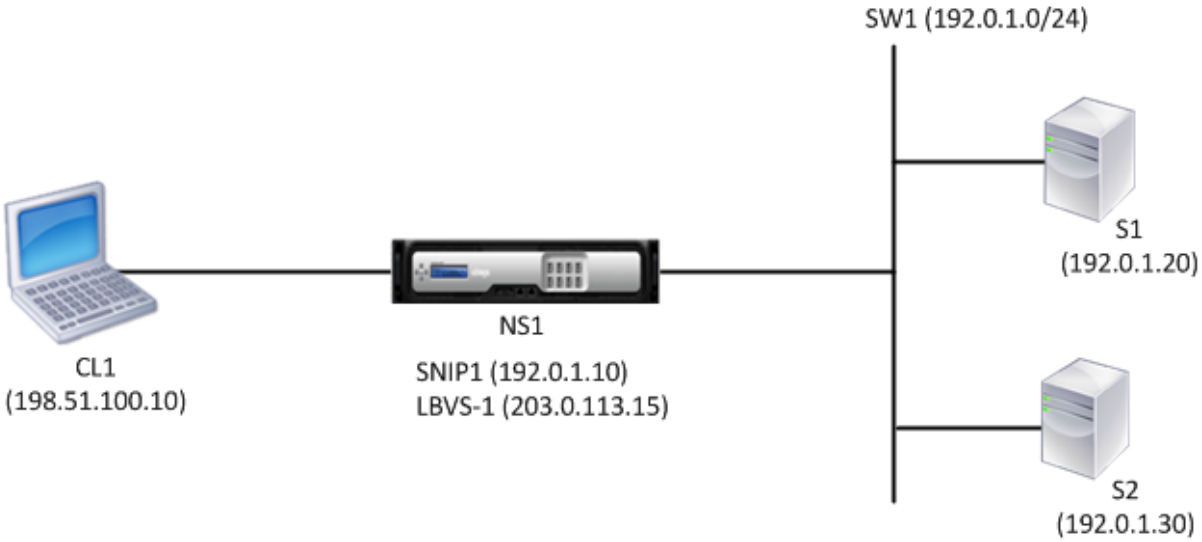
Citrix ADC と Citrix ADC に直接接続されるサーバーまたは L2 スイッチのみを介して接続されるサーバー間の通信を有効にするには、サーバーのサブネットに属するサブネット IP アドレスを構成する必要があります。直接接続された各サブネットに対して少なくとも 1 つのサブネット IP アドレスを構成する必要があります。ただし、NSIP を介して接続されている直接接続された管理サブネットは除きます。

たとえば、Citrix ADC NS1 上の負荷分散仮想サーバー LBVS1 を使用して、L2 スイッチ SW1 を介して NS1 に接続されているサーバー S1 と S2 の負荷分散を行います。S1 と S2 は同じサブネットに属します。

SNIP アドレス SNIP1 は、S1 および S2 と同じサブネットに属しており、NS1 上で構成されています。SNIP1 が設定されると、NS1 は SNIP1 の ARP パケットをブロードキャストします。

NS1 上のサービス SVC-S1 および SVC-S2 は、S1 および S2 を表します。これらのサービスが構成されるとすぐに、NS1 は S1 と S2 の ARP 要求をブロードキャストして IP-to-MAC マッピングを解決します。S1 と S2 が応答した後、NS1 はアドレス SNIP1 から定期的に監視プローブを送信し、正常性をチェックします。

Citrix ADC での負荷分散の構成の詳細については、「負荷分散」を参照してください。



次に、この例のトラフィックフローを示します。

1. クライアント C1 は要求パケットを LBVS-1 に送信します。要求パケットには次のものがあります。
   - 送信元 IP = クライアントの IP アドレス (198.51.100.10)
   - 宛先 IP = LBVS-1 の IP アドレス (203.0.113.15)
2. NS1 の LBVS1 が要求パケットを受信します。
3. LBVS1 の負荷分散アルゴリズムは、サーバー S2 を選択します。
4. S2 は NS1 に直接接続されており、SNIP1（192.0.1.10）は S2 と同じサブネットに属する唯一の IP アドレスであるため、NS1 は SNIP1 と S2 の間の接続を開きます。
5. NS1 は、SNIP1 から S2 に要求パケットを送信します。要求パケットには次のものがあります。
   - 送信元 IP = SNIP1 (192.0.1.10)
   - 宛先 IP = S2 の IP アドレス（192.0.1.30）
6. S2 の応答は同じパスで返されます。

ルータ経由で接続されたサーバサブネットでの **SNIP** の使用

ルーター経由で接続されたサブネット内の Citrix ADC とサーバー間の通信を有効にするには、ルーターに直接接続されたインターフェイスのサブネットに属するサブネット IP アドレスを少なくとも 1 つ構成する必要があります。ADC は、このサブネットの IP アドレスを使用して、ルータ経由で到達可能なサブネット内のサーバと通信します。

たとえば、Citrix ADC NS1 上の負荷分散仮想サーバー LBVS1 を使用して、ルーター R1 を介して NS1 に接続されているサーバー S1、S2、S3、および S4 を負荷分散します。

S1 と S2 は同じサブネット 192.0.2.0/24 に属し、L2 スイッチ SW1 を介して R1 に接続されています。S3 と S4 は別のサブネット 192.0.3.0/24 に属し、L2 スイッチ SW2 を介して R1 に接続されています。

Citrix ADC NS1 は、サブネット 192.0.1.0/24 を介してルーター R1 に接続されています。SNIP アドレス SNIP1 は、ルータに直接接続されたインターフェイスと同じサブネット（192.0.1.0/24）に属し、NS1 上で設定されます。NS1 は、このアドレスを使用して、サーバ S1 および S2 と通信し、サーバ S3 および S4 と通信します。

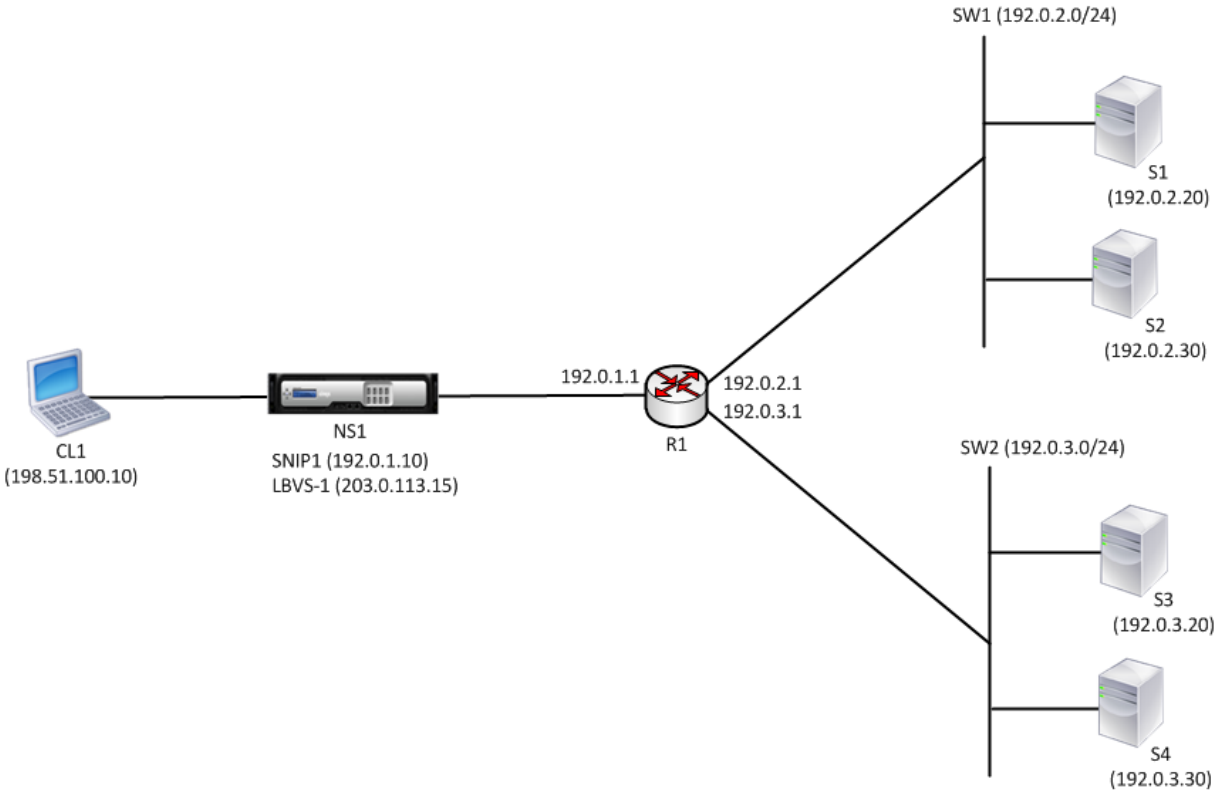
Citrix ADC での負荷分散の構成の詳細については、「負荷分散」を参照してください。

アドレス SNIP1 が設定されると、NS1 は SNIP1 の ARP アナウンスパケットをブロードキャストします。

NS1 のルーティングテーブルは、S1、S2、S3、および S4 から R1 までのルートエントリで構成されます。これらのルートエントリは、スタティックルートエントリであるか、ダイナミックルーティングプロトコルを使用して R1 から NS1 にアドバタイズされます。

NS1 上のサービス SVC-S1、SVC-S2、SVC-S3、および SVC-S4 は、サーバ S1、S2、S3、および S4 を表します。NS1 は、ルーティング・テーブル内で、これらのサーバが R1 を介して到達可能であることを検出します。NS1 は、アドレス SNIP1 から定期的に監視プローブを送信し、正常性をチェックします。

Citrix ADC での IP ルーティングの詳細については、「IP ルーティング」を参照してください。



次に、この例のトラフィックフローを示します。

1. クライアント C1 は要求パケットを LBVS-1 に送信します。要求パケットには次のものがあります。
   - 送信元 IP = クライアントの IP アドレス (198.51.100.10)
   - 宛先 IP = LBVS-1 の IP アドレス (203.0.113.15)
2. NS1 の LBVS1 が要求パケットを受信します。
3. LBVS1 の負荷分散アルゴリズムは、サーバー S3 を選択します。
4. NS1 はルーティングテーブルをチェックし、S3 が R1 を介して到達可能であることを検出します。SNIP1（192.0.1.10）は、ルータ R1 と同じサブネットに属する NS1 上の唯一の IP アドレスであり、NS1 は R1 を介して SNIP1 と S3 間の接続を開きます。
5. NS1 は、SNIP1 から R1 に要求パケットを送信します。要求パケットには次のものがあります。

- 送信元 IP アドレス = SNIP1 (192.0.1.10)
- 宛先 IP アドレス = S3 の IP アドレス (192.0.3.20)

6. リクエストは R1 に到達し、R1 はルーティングテーブルをチェックし、リクエストパケットを S3 に転送します。
7. S3 の応答は同じパスで返されます。

L2 スイッチ上の複数のサーバサブネット（VLAN）に対する SNIP の使用

Citrix ADC に接続された L2 スイッチに複数のサーバーサブネット（VLAN）がある場合、Citrix ADC がこれらのサーバーサブネットと通信できるように、サーバーサブネットごとに少なくとも 1 つの SNIP アドレスを構成する必要があります。

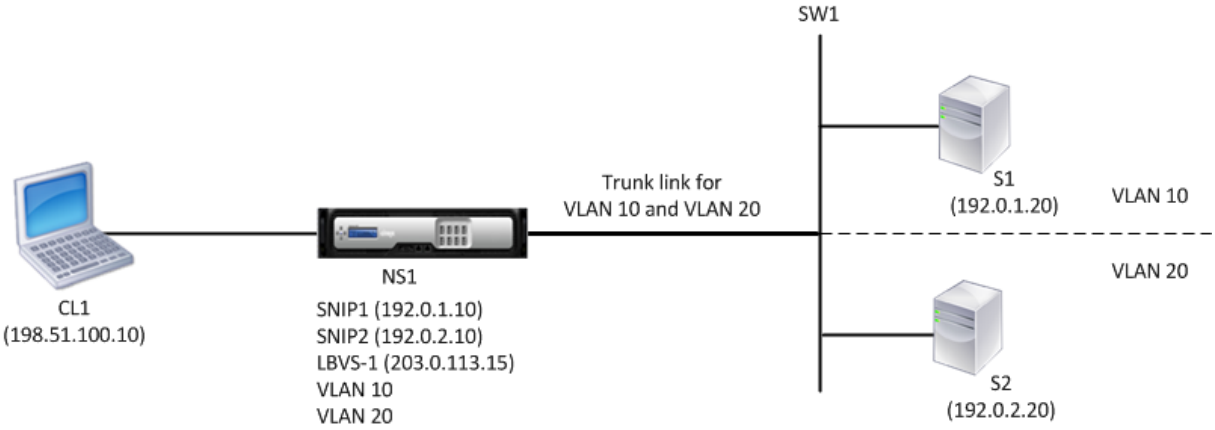
たとえば、Citrix ADC NS1 上の負荷分散仮想サーバー LBVS1 を使用して、L2 スイッチ SW1 を介して NS1 に接続されているサーバー S1 と S2 の負荷分散を行います。S1 と S2 は異なるサブネットに属し、それぞれ VLAN 10 と VLAN20 の一部です。NS1 と SW1 の間のリンクはトランクリンクであり、VLAN10 と VLAN20 によって共有されます。

Citrix ADC での負荷分散の構成の詳細については、「負荷分散」を参照してください。

サブネット IP アドレス SNIP1（参照目的のみ）および SNIP2（参照目的のみ）は、NS1 上で構成されます。NS1 は SNIP1（VLAN 10 上）を使用してサーバ S1 と通信し、SNIP2（VLAN 20 上）を使用して S2 と通信します。SNIP1 と SNIP2 が構成されると、NS1 は SNIP1 と SNIP2 の ARP アナウンスメント・パケットをブロードキャストします。

Citrix ADC での VLAN の構成の詳細については、「VLAN の構成」を参照してください。

NS1 上のサービス SVC-S1 および SVC-S2 は、サーバ S1 および S2 を表します。これらのサービスが構成されるとすぐに、NS1 はそれらの ARP 要求をブロードキャストします。S1 と S2 が応答した後、NS1 は定期的に監視プローブを送信し、稼働状態をチェックします。NS1 は監視プローブをアドレス SNIP1 から S1 に送信し、アドレス SNIP2 から S2 に送信します。



次に、この例のトラフィックフローを示します。

1. クライアント C1 は要求パケットを LBVS-1 に送信します。要求パケットには次のものがあります。
   - 送信元 IP = クライアントの IP アドレス (198.51.100.10)

- 宛先 IP = LBVS-1 の IP アドレス (203.0.113.15)

2. NS1 の LBVS1 が要求パケットを受信します。
3. LBVS1 の負荷分散アルゴリズムは、サーバー S2 を選択します。
4. S2 は NS1 に直接接続され、SNIP2（192.0.2.10）は NS1 上で S2 と同じサブネットに属する唯一の IP アドレスであるため、NS1 は SNIP2 と S2 の間の接続を開きます。
   注：S1 が選択されている場合、NS1 は SNIP1 と S1 間の接続を開きます。
5. NS1 は、SNIP2 から S2 に要求パケットを送信します。要求パケットには次のものがあります。
   - 送信元 IP = SNIP1（192.0.2.10）
   - 宛先 IP = S2 の IP アドレス（192.0.2.20）
6. S2 の応答は同じパスで返されます。

GSLB サイトの IP アドレスの設定 (GSLBIP)

October 7, 2021

GSLB サイトの IP（GSLBIP）アドレスは、GSLB サイトに関連付けられた IP アドレスです。Citrix ADC アプライアンスを最初に構成するときに、GSLBIP アドレスの指定は必須ではありません。GSLBIP アドレスは、GSLB サイトを作成する場合にのみ使用されます。

GSLB サイトの IP アドレスの作成の詳細については、「グローバルサーバーの負荷分散」を参照してください。

Citrix ADC が所有する IP アドレスの削除

October 7, 2021

NSIP 以外のすべての IP アドレスを削除できます。次の表に、さまざまな種類の IP アドレスを削除するために従う必要があるプロセスに関する情報を示します。VIP を削除する前に、関連付けられている仮想サーバを削除してください。

IP アドレスの種類	意味合いです
サブネット IP アドレス (SNIP)	削除する IP アドレスがサブネット内の最後の IP アドレスである場合、関連付けられたルートはルートテーブルから削除されます。削除する IP アドレスが対応するルートエントリの Gateway である場合、そのサブネットルートの Gateway は別の Citrix ADC が所有する IP アドレスに変更されます。

IP アドレスの種類	意味合いです
仮想サーバ IP アドレス（VIP）	VIP を削除する前に、まず VIP に関連付けられた仮想サーバを削除する必要があります。仮想サーバーの削除の詳細については、「負荷分散」を参照してください。
GSLB サイトの IP アドレス	GSLB サイトの IP アドレスを削除する前に、それに関連付けられているサイトを削除する必要があります。サイトの削除の詳細については、「グローバルサーバーの負荷分散」を参照してください。

CLI を使用して IP アドレスを削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

rm ns ip <IPaddress>

例**:**

```
1 > rm ns ip 10.102.29.54
2 Done
3 <!--NeedCopy-->
```

GUI を使用して IP アドレスを削除するには、次の手順を実行します。

[システム] > [ネットワーク] > [IP] > [**IPV4**] に移動し、IP アドレスを削除します。

アプリケーション・アクセス制御の設定

October 7, 2021

アプリケーション・アクセス制御は、管理アクセス制御とも呼ばれ、ユーザー認証を管理し、アプリケーションおよびデータへのユーザー・アクセスを決定するルールを実装するための統一されたメカニズムを形成します。SNIP を構成して、管理アプリケーションへのアクセスを提供できます。NSIP の管理アクセスはデフォルトで有効になっており、無効にすることはできません。ただし、ACL を使用して制御することはできます。

ACL の使用の詳細については、アクセス制御リスト (ACL)を参照してください。

Citrix ADC アプライアンスは、VIP への管理アクセスをサポートしていません。

次の表は、管理アクセスと Telnet の特定のサービス設定の間の相互作用をまとめたものです。

管理アクセス	Telnet（Citrix ADC で設定された状態）	Telnet（IP レベルでの有効状態）
有効化	有効化	有効化
有効化	無効化	無効化
無効化	有効化	無効化
無効化	無効化	無効化

次の表に、アウトバウンドトラフィックで送信元 IP アドレスとして使用される IP アドレスの概要を示します。

アプリケーション/ IP	NSIP	SNIP	VIP
ARP	はい	はい	いいえ
サーバー側のトラフィック	いいえ	はい	いいえ
RNAT	いいえ	はい	はい
ICMP PING	はい	はい	いいえ
動的ルーティング	はい	はい	はい

次の表に、これらの IP アドレスで使用できるアプリケーションの概要を示します。

アプリケーション/ IP	NSIP	SNIP	VIP
SNMP	はい	はい	はい
システムアクセス	はい	はい	いいえ

Telnet、SSH、GUI、FTP などのアプリケーションを使用して、Citrix ADC にアクセスして管理できます。

注：セキュリティ上の理由から、Citrix ADC では Telnet と FTP が無効になっています。有効にするには、カスタマーサポートに連絡してください。アプリケーションを有効にしたら、IP レベルでコントロールを適用できます。

これらのアプリケーションに応答するように Citrix ADC を構成するには、特定の管理アプリケーションを有効にする必要があります。IP アドレスの管理アクセスを無効にすると、その IP アドレスを使用する既存の接続は終了しませんが、新しい接続を開始することはできません。

また、基盤となる FreeBSD オペレーティングシステムで実行されている非管理アプリケーションはプロトコル攻撃を受けやすく、これらのアプリケーションは Citrix ADC アプライアンスの攻撃防止機能を活用しません。

SNIP または NSIP 上のこれらの非管理アプリケーションへのアクセスをブロックできます。アクセスがブロックさ

れると、SNIP または NSIP を使用して Citrix ADC に接続しているユーザーは、基盤となるオペレーティングシステムで実行されている非管理アプリケーションにアクセスできなくなります。

CLI を使用して IP アドレスの管理アクセスを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

set ns ip <IPAddress> -mgmtAccess <value> -**telnet** <value> -**ftp** <value> -**gui** <value> -**ssh** <value> -**snmp** <value> -**restrictAccess** (**ENABLED** | **DISABLED**)

例**:**

```
1  > set ns ip 10.102.29.54 -mgmtAccess enabled -restrictAccess ENABLED
2   Done
3 <!--NeedCopy-->
```

GUI を使用して IP アドレスの管理アクセスを有効にするには、次の手順を実行します。

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. IP アドレスエントリを開き、[一覧表示されたアプリケーションをサポートするために 管理アクセス制御を有効にする] オプションを選択します。

サブネット **IP** アドレス（**SNIP**）を使用して **Citrix ADC GUI** への安全なアクセスを有効にする

Citrix ADC IP（NSIP）では、Citrix ADC GUI へのセキュアアクセスがデフォルトで有効になっています。アプライアンスのサブネット IP アドレスを使用して、Citrix ADC アプライアンスへの安全なアクセスを有効にすることもできます。

ハイアベイラビリティペアへのセキュアアクセス用の SNIP アドレスを設定した後、SNIP アドレスにアクセスすると、プライマリアプライアンスがセキュアアクセスできるようになります。

Citrix ADC LI 手順

CLI を使用してサブネット IP アドレス（SNIP）を使用して Citrix ADC GUI への安全なアクセスを有効にするには:

コマンドプロンプトで入力します。

ns IP タイプに設定 <SNIP_Address>**SNIP-gui** セキュアオンリー**-mgmtAccess** 有効

例**:**

```
1  > set ns ip 203.0.113.99 -mgmtAccess enabled -restrictAccess ENABLED
2
3   Done
4 <!--NeedCopy-->
```


Citrix ADC プロキシの接続方法

October 7, 2021

クライアントが接続を開始すると、Citrix ADC アプライアンスはクライアント接続を終了し、適切なサーバーへの接続を開始し、パケットをサーバーに送信します。アプライアンスは、サービスタイプ UDP または ANY に対してこのアクションを実行しません。

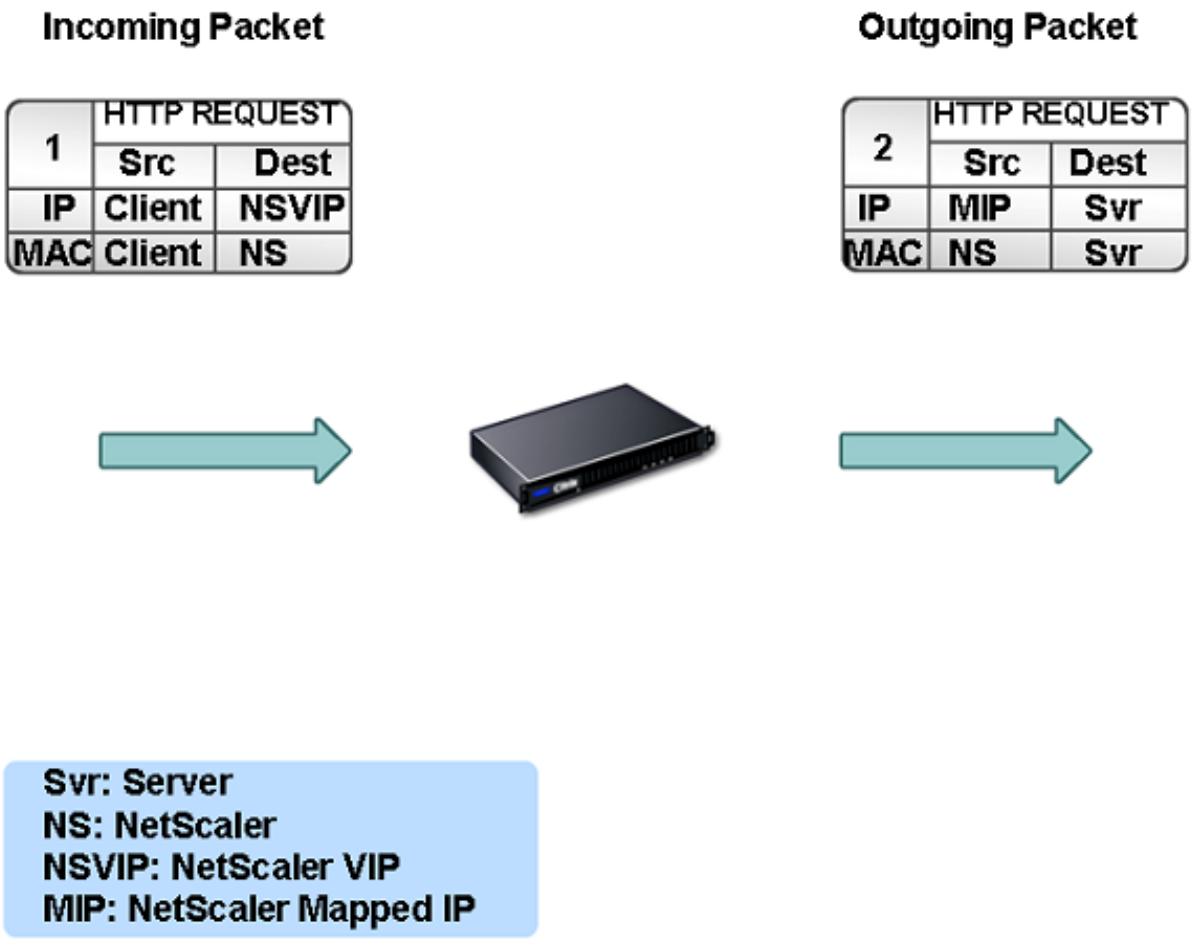
サービスタイプの詳細については、「負荷分散」を参照してください。

サーバーとの接続を開始する前に、パケットを処理するように Citrix ADC を構成できます。デフォルトの動作では、パケットをサーバに送信する前に、パケットの送信元と宛先 IP アドレスを変更します。［Use Source IP］モードを有効にすると、パケットの送信元 IP アドレスを保持するように Citrix ADC を構成できます。

宛先 IP アドレスの選択方法

Citrix ADC アプライアンスに送信されるトラフィックは、仮想サーバーまたはサービスに送信できます。アプライアンスは、仮想サーバとサービスへのトラフィックを別々に処理します。Citrix ADC は、次の図に示すように、仮想サーバー IP（VIP）アドレスで受信したトラフィックを終了し、宛先 IP アドレスをサーバーの IP アドレスに変更してから、トラフィックをサーバーに転送します。

図 1: VIP への接続のプロキシ



サービス宛のパケットは適切なサーバーに直接送信され、Citrix ADC は宛先 IP アドレスを変更しません。この場合、Citrix ADC はプロキシとして機能します。

送信元 **IP** アドレスの選択方法

Citrix ADC アプライアンスが物理サーバーまたはピアデバイスと通信する場合、デフォルトでは、クライアントの IP アドレスは使用されません。Citrix ADC は、サブネット IP アドレス（SNIP）のプールを維持し、物理サーバーへの接続の送信元 IP アドレスとして使用する IP アドレスをこのプールから選択します。物理サーバーが配置されているサブネットに応じて、Citrix ADC は特定の SNIP アドレスを選択します。

注:「ソース IP（USIP）の使用」オプションが有効になっている場合、アプライアンスはクライアントの IP アドレスを使用します。

送信元 **IP** モードの使用を有効にする

October 7, 2021

Citrix ADC アプライアンスが物理サーバーまたはピアデバイスと通信する場合、デフォルトでは、独自の IP アドレスがソース IP として使用されます。アプライアンスは、サブネット IP アドレス（SNIP）のプールを維持し、物理サーバへの接続のための送信元 IP アドレスとして使用する IP アドレスをこのプールから選択します。SNIP アドレスの選択は、物理サーバが存在するサブネットによって異なります。

必要に応じて、クライアントの IP アドレスをソース IP として使用するように Citrix ADC アプライアンスを構成できます。アプリケーションによっては、クライアントの実際の IP アドレスが必要です。次のユースケースは、いくつかの例です。

- Web アクセスログ内のクライアントの IP アドレスは、課金目的や使用状況分析に使用されます。
- クライアントの IP アドレスは、クライアントの出身国またはクライアントの発信元 ISP を決定するために使用されます。たとえば、Google などの多くの検索エンジンは、ユーザーが属する場所に関連するコンテンツを提供します。
- アプリケーションは、要求が信頼できる送信元からのものであることを確認するために、クライアントの IP アドレスを知っている必要があります。
- アプリケーションサーバーでクライアントの IP アドレスが必要ない場合でも、アプリケーションサーバーと Citrix ADC の間に配置されたファイアウォールで、トラフィックをフィルタリングするためにクライアントの IP アドレスが必要になることがあります。

Citrix ADC でサーバーとの通信にクライアントの IP アドレスを使用する場合は、「ソース IP モード（USIP）を使用」モードを有効にします。

次の図は、アプライアンスが USIP モードでの IP アドレスを使用する方法を示しています。

はじめに

USIP モードを有効にする前に、次の点に注意してください。

- 次の状況で USIP を有効にします。
  - IDS（侵入検知システム）サーバの負荷分散
  - SMTP 負荷分散
  - ステートレス接続のフェイルオーバー
  - セッションレス負荷分散
  - ダイレクトサーバーリターン (DSR) モードを使用する場合
- USIP グローバル設定は、USIP グローバル設定が行われた後に作成されるサービスにのみ適用されます。つまり、USIP グローバル設定が行われると、USIP グローバル設定は既存のサービスに適用されません。たとえば、USIP をグローバルに無効にしても、既存のサービスの USIP は無効になりません。しかし、その後、作成されたサービスが自動的に USIP を有効にするのを停止します。

  既存のサービスのセットで USIP を有効または無効にするには、これらの各サービスで USIP を有効または無効にする必要があります。
- USIP が有効な場合、サーバーの応答が常に Citrix ADC アプライアンスを通過するように、サーバーの Gateway を Citrix ADC 所有の IP アドレス（SNIP）のいずれかに設定する必要があります。
- USIP を有効にする場合は、サーバー接続のアイドルタイムアウトをデフォルト値よりも低い値に設定し、アイドル接続がサーバー側で迅速にクリアされるようにします。
- トランスペアレントキャッシュリダイレクションの場合、USIP を有効にする場合は、L2CONN も有効にします。
- USIP が有効な場合、HTTP 接続は再利用されないため、多数のサーバー側接続が蓄積する可能性があります。アイドル状態のサーバー接続は、他のクライアントの接続をブロックできます。したがって、サービスへの接続の最大数に制限を設定します。また、USIP が有効なサービスの HTTP サーバーのタイムアウト値をデフォルトよりも低い値に設定して、アイドル状態の接続がサーバー側で迅速にクリアされるようにすることをお勧めします。
- USIP モードの代わりに、クライアントの IP アドレスを必要とするアプリケーションサーバーのサーバー側接続の要求ヘッダーにクライアントの IP アドレス (CIP) を挿入することもできます。
- 以前の Citrix ADC リリースでは、USIP モードにはサーバー側接続用の次のソースポートオプションがありました。
  - クライアントのポートを使用します。このオプションでは、接続を再利用できません。クライアントからの要求ごとに、物理サーバとの新しい接続が確立されます。
  - プロキシポートを使用します。このオプションを使用すると、同じクライアントからのすべての要求で接続を再利用できます。

  新しい Citrix ADC リリースでは、USIP が有効になっている場合、デフォルトでは、サーバー側の接続にはプロキシポートが使用され、接続は再利用されません。接続を再利用しないと、接続の確立速度には影響しません。

  USIP モードが有効になっている場合、[プロキシポートを使用] オプションはデフォルトで有効になります。

注**:** USIP モードを有効にする場合は、[プロキシポートを使用] オプションを有効にすることをお勧めします。

[プロキシポートを使用] オプションの詳細については、「サーバー側接続の送信元ポートを構成する」を参照してください。

構成の手順

Citrix ADC でサーバーとの通信にクライアントの IP アドレスを使用する場合は、「ソース IP モード（USIP）を使用」モードを有効にします。デフォルトでは、USIP モードは無効です。USIP モードは、Citrix ADC または特定のサービスでグローバルに有効にできます。グローバルに有効にすると、以降に作成されるすべてのサービスに対して USIP がデフォルトで有効になります。特定のサービスに対して USIP を有効にすると、クライアントの IP アドレスはそのサービス宛てのトラフィックに対してのみ使用されます。

CLI のプロシージャ

CLI を使用して USIP モードをグローバルに有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **enable ns mode USIP**
- **disable ns mode USIP**

CLI を使用してサービスの USIP モードを有効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

set service <name>@ -**usip** (**YES** | いいえ)

例**:**

```
1 > set service Service-HTTP-1 -usip YES
2 Done
3 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して USIP モードをグローバルに有効または無効にするには、次の手順を実行します。

1. [システム**] > [**設定**]** に移動し、[モードと機能] で [モードの変更] をクリックします。
2. [ソース **IP** を使用] オプションをオンまたはオフにします。

GUI を使用してサービスの USIP モードを有効にするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを編集します。
2. [詳細設定**]** で、[サービス設定**]** を選択し、[送信元 **IP** アドレスの使用] を選択します。

ネットワークアドレス変換の構成

October 7, 2021

ネットワークアドレス変換（NAT）では、Citrix ADC アプライアンスを通過する IP パケットの送信元または宛先 IP アドレスおよび/または TCP/UDP ポート番号の変更が含まれます。アプライアンスで NAT を有効にすると、データが Citrix ADC を通過するときにネットワークの送信元 IP アドレスを変更することで、プライベートネットワークのセキュリティが強化され、インターネットなどのパブリックネットワークから保護されます。また、NAT エントリを使用すると、プライベートネットワーク全体をいくつかの共有パブリック IP アドレスで表すことができます。Citrix ADC では、次の種類のネットワークアドレス変換がサポートされています。

- インバウンド **NAT**（**INAT**）。Citrix ADC は、クライアントによって生成されたパケットの宛先 IP アドレスをサーバーのプライベート IP アドレスに置き換えます。
- リバース **NAT**（**RNAT**）。Citrix ADC は、サーバーによって生成されたパケットの送信元 IP アドレスをパブリック NAT IP アドレスに置き換えます。

インバウンドネットワークアドレス変換

October 7, 2021

クライアントが受信ネットワークアドレス変換（INAT）用に構成された Citrix ADC アプライアンスにパケットを送信すると、アプライアンスはパケットのパブリック宛先 IP アドレスをプライベート宛先 IP アドレスに変換し、そのアドレスのサーバーにパケットを転送します。

次の構成がサポートされています。

- **IPv4-IPv4** マッピング：Citrix ADC アプライアンス上のパブリック IPv4 アドレスは、プライベート IPv4 サーバーの代わりに接続要求をリッスンします。Citrix ADC アプライアンスは、パケットのパブリック宛先 IP アドレスをサーバーの宛先 IP アドレスに変換します。次に、アプライアンスはパケットをそのアドレスのサーバーに転送します。
- **IPv4-IPv6** マッピング：Citrix ADC アプライアンス上のパブリック IPv4 アドレスは、プライベート IPv6 サーバーの代わりに接続要求をリッスンします。Citrix ADC アプライアンスは、IPv6 サーバーの IP アドレスを宛先 IP アドレスとして持つ IPv6 要求パケットを作成します。
- **IPv6-IPv4** マッピング：Citrix ADC アプライアンス上のパブリック IPv6 アドレスは、プライベート IPv4 サーバーの代わりに接続要求をリッスンします。Citrix ADC アプライアンスは、宛先 IP アドレスとして IPv4 サーバーの IP アドレスを持つ IPv4 要求パケットを作成します。
- **IPv6-IPv6** マッピング：Citrix ADC アプライアンス上のパブリック IPv6 アドレスは、プライベート IPv6 サーバーの代わりに接続要求をリッスンします。Citrix ADC アプライアンスは、パケットのパブリック宛先 IP アドレスをサーバーの宛先 IP アドレスに変換します。次に、アプライアンスはパケットをそのアドレスのサーバーに転送します。

アプライアンスがパケットをサーバに転送すると、パケットに割り当てられた送信元 IP アドレスは次のように決定されます。

- サブネット IP（USNIP）の使用モードが有効で、送信元 IP（USIP）の使用モードが無効になっている場合、アプライアンスは送信元 IP アドレスとしてサブネット IP アドレス（SNIP）を使用します。
- USIP モードが有効で、USNIP モードが無効になっている場合、アプライアンスはクライアント IP（CIP）アドレスを送信元 IP アドレスとして使用します。
- USIP モードと USNIP モードの両方が有効になっている場合は、USIP モードが優先されます。
- また、proxyIP パラメータを設定することで、一意の IP アドレスが送信元 IP アドレスとして使用されるように Citrix ADC を構成することもできます。
- 上記のモードのいずれも有効にならず、一意の IP アドレスが指定されていない場合、Citrix ADC は MIP を送信元 IP アドレスとして使用しようとします。
- USIP モードと USNIP モードの両方が有効で、一意の IP アドレスが指定されている場合、優先順位は USIP 一意の IP-USNIP-MIP エラーです。

Citrix ADC を DoS 攻撃から保護するために、TCP プロキシを有効にすることができます。ただし、ネットワークで他の保護メカニズムが使用されている場合は、それらを無効にすることができます。

INAT ルールを構成する

INAT エントリを作成、変更、または削除できます。

CLI のプロシージャ

CLI を使用して INAT エントリを作成するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを入力して INAT エントリを作成し、その構成を確認します。

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** <**ip_addr > ipv6_addr>**]
- **show inat** [<name>]

例**:**

```
1 > add inat ip4-ip4 172.16.1.2 192.168.1.1 -proxyip 10.102.29.171
2  Done
3 <!--NeedCopy-->
```

CLI を使用して INAT エントリを変更するには、次の手順を実行します。

INAT エントリを変更するには、`set inat` コマンド、エントリの名前、および変更するパラメータを新しい値とともに入力します。

CLI を使用して INAT 設定を削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **rm inat** <name>

例**:**

```
> rm inat ip4-ip4
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して INAT エントリを設定するには、次の手順を実行します。

システムに移動 > 通信網 > ルート > **INAT**、および INAT エントリを追加するか、既存の INAT エントリを編集します。

GUI を使用して INAT 設定を削除するには、次の手順を実行します。

[システム] > [ネットワーク] > [ルート] > [**INAT**] に移動し、INAT 設定を削除します。

INAT ルールの接続フェイルオーバー

接続フェイルオーバーまたは接続ミラーリングにより、プライマリノードは接続と永続性の情報をセカンダリノードに高可用性で複製できます。接続ミラーリングが有効になっている場合、接続の状態情報は定期的にセカンダリノードと共有されます。

接続フェイルオーバーを有効にすると信頼性が向上しますが、状態情報の共有にシステム時間が消費されるという犠牲が伴います。接続データは、パケットまたはフロー状態が更新されるたびにスタンバイユニットに同期されます。したがって、接続レベルの信頼性が最も重要な場所でのみ使用する必要があります。

Citrix ADC アプライアンスの高可用性セットアップは、INAT 接続の接続フェイルオーバーをサポートします。プライマリノードは、INAT マッピングおよびその他の INAT 関連の接続情報を定期的にセカンダリノードに送信します。セカンダリアプライアンスは、フェイルオーバーが発生した場合にのみマッピングおよび接続情報を使用します。

フェイルオーバーが発生すると、新しいプライマリノードには、フェイルオーバーの前に確立された INAT 接続に関する情報があります。したがって、フェイルオーバー後も引き続きこれらの接続を提供します。

クライアントの観点からは、フェイルオーバーは透過的です。移行期間中、クライアントとサーバーで短時間の中断と再送信が発生する可能性があります。接続フェイルオーバーは、INAT ルールに従って有効にできます。

INAT ルールで接続フェイルオーバーを有効にするには、CLI を使用してその特定の RNAT ルールの `connFailover` パラメーターを有効にします。

CLI 手順

CLI を使用して INAT ルールの接続フェイルオーバーを有効にするには:

INAT ルールの追加中に接続フェイルオーバーを有効にするには、コマンドプロンプトで次のように入力します。

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** <ip_addr|ipv6_addr>] -**connfailover** (**ENABLED** | **DISABLED**)
- **show inat** <name>

既存の INAT ルールの変更中に接続フェイルオーバーを有効にするには、コマンドプロンプトで次のように入力します。

- **set inat -connfailover** （**ENABLED** | 無効）
- **show inat** <name>

INAT と仮想サーバの共存

October 7, 2021

INAT と RNAT の両方が設定されている場合、INAT ルールは RNAT ルールよりも優先されます。RNAT にネットワークアドレス変換 IP (NAT IP) アドレスが設定されている場合、その RNAT クライアントの送信元 IP アドレスとして NAT IP アドレスが選択されます。

INAT 構成のデフォルトのパブリック宛先 IP は、Citrix ADC デバイスの仮想 IP（VIP）アドレスです。仮想サーバーも VIP を使用します。INAT と仮想サーバの両方が同じ IP アドレスを使用する場合、仮想サーバの設定は INAT 設定を上書きします。

次に、構成セットアップのシナリオとその効果の例を示します。

サポート案件	結果
特定の Citrix ADC ポートで受信したすべてのデータパケットをサーバーに直接送信するように、仮想サーバーとサービスを構成しました。また、INAT を設定し、TCP を有効にしました。この方式で INAT を設定すると、TCP エンジンを介して受信したすべてのデータパケットがサーバに送信される前に送信されます。	Citrix ADC で受信されたすべてのパケットは、指定されたポートで受信されたパケットを除き、TCP エンジンを通過します。

サポート案件	結果
Citrix ADC 上の特定のポートで受信したサービスタイプ TCP のすべてのデータパケットを、TCP エンジンを通過した後にサーバーに送信するように仮想サーバーとサービスを構成しました。また、INAT を設定し、TCP を無効にしました。この方式で INAT を設定すると、直接受信したデータパケットがサーバに送信されます。	指定されたポートで受信されたパケットだけが TCP エンジンを通過します。
これで、受信したすべてのデータパケットを 2 つのサーバのいずれかに送信するように、仮想サーバとサービスを構成しました。受信したすべてのデータパケットを別のサーバに送信するように INAT を設定しようとしています。	INAT 設定は許可されていません。
受信したすべてのデータパケットをサーバに直接送信するように INAT を設定しました。仮想サーバーとサービスを構成して、受信したすべてのデータパケットを 2 つの異なるサーバーに送信しようとしています。	vserver の設定は許可されていません。

Stateless NAT46

October 7, 2021

ステートレスな NAT46 機能を使用すると、Citrix ADC アプライアンスでセッション情報を維持することなく、IPv4 から IPv6 へのパケット変換（またはその逆）を介して IPv4 と IPv6 ネットワーク間の通信が可能になります。

ステートレスな NAT46 構成の場合、アプライアンスは、RFC6145 および 2765 で定義されているように、IPv4 パケットを IPv4 に変換するか、IPv6 パケットを IPv4 に変換します。

Citrix ADC アプライアンスのステートレスな NAT46 構成には、次のコンポーネントがあります。

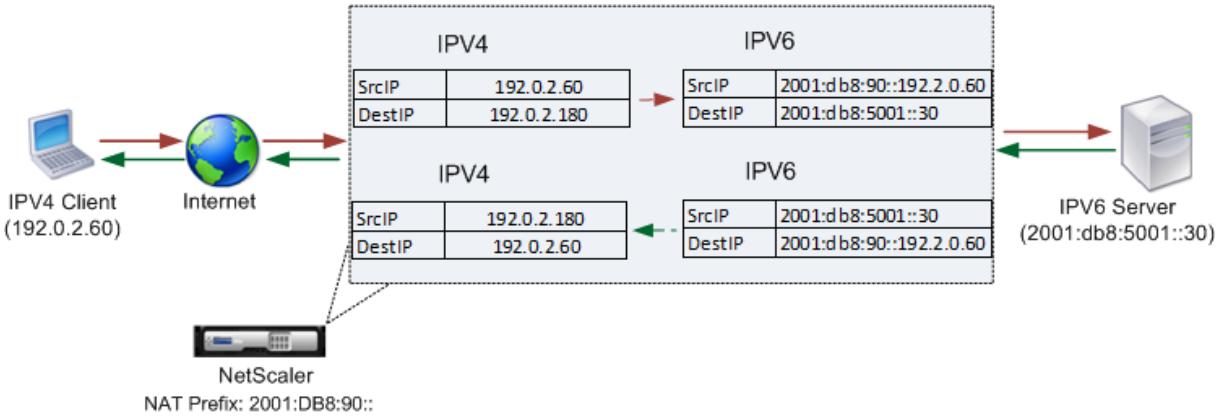
- **IPv4/IPv6** の **INAT** エントリです。IPv4 アドレスと IPv6 アドレスの 1:1 の関係を定義する INAT エントリ。つまり、アプライアンス上の IPv4 アドレスは、IPv6 サーバーに代わって接続要求をリッスンします。この IPv4 アドレスの IPv4 要求パケットが IPv6 パケットに変換され、IPv6 パケットが IPv6 サーバに送信されます。

  アプライアンスは、IPv6 応答パケットを IPv4 応答パケットに変換します。送信元 IP アドレスフィールドは INAT エントリで指定された IPv4 アドレスとして設定されています。次に、変換されたパケットがクライアントに送信されます。

- **NAT46 IPv6** プレフィクスです。アプライアンスで設定されている長さ 96 ビット（128-32=96）のグローバル IPv6 プレフィクス。IPv4 パケットから IPv6 パケットへの変換中に、アプライアンスは、変換された IPv6 パケットの送信元 IP アドレスを、要求パケットで受信した NAT46 IPv6 プレフィクス [96 ビットと][IPv4 送信元アドレス 32 ビットの連結に設定します] 。

  IPv6 パケットから IPv4 パケットへの変換中、アプライアンスは、変換された IPv4 パケットの宛先 IP アドレスを、IPv6 パケットの宛先 IP アドレスの最後の 32 ビットに設定します。

たとえば、企業が IPv6 アドレスを持つサーバ S1 上でサイト www.example.com をホストしているとします。IPv4 クライアントと IPv6 サーバー S1 間の通信を可能にするために、Citrix ADC アプライアンス NS1 は、サーバー S1 の IPv4-IPv6 INAT エントリと NAT46 プレフィックスを含むステートレス NAT46 構成で展開されます。INAT エントリには、アプライアンスが IPv6 サーバー S1 に代わって IPv4 クライアントからの接続要求をリッスンする IPv4 アドレスが含まれています。



次の表に、この例で使用される設定を示します。

エンティティ	名前	値
クライアントの IP アドレス	Client_IPv4 (参照用のみ)	192.0.2.60
サーバーの IPv6 アドレス	Sevr_IPv6 (参照用のみ)	2001:DB8:5001::30
IPv6 サーバー S1 の INAT エントリで定義された IPv4 アドレス	Map-Sevr-IPv4 (参照用のみ)	192.0.2.180
NAT 46 変換用の IPv6 プレフィクス	NAT46_Prefix (参照用のみ)	2001:DB8:90::

次に、この例のトラフィックフローを示します。

1. IPv4 クライアント CL1 は、Citrix ADC アプライアンス上のマップサーバ-IPv4（192.0.2.180）アドレスに要求パケットを送信します。
2. アプライアンスは要求パケットを受信し、NAT46 INAT エントリを検索して、マップ-sevr-IPv4 (192.0.2.180) アドレスにマッピングされた IPv6 アドレスを検索します。これは、Sevr-IPv6（2001：DB 8:5001：30）ア

ドレスを検索します。

3. アプライアンスは、次のように変換された IPv6 要求パケットを作成します。
   - 宛先 IP アドレスフィールド = Sevr-IPv6 = 2001: DB 8:5001:30
   - 送信元 IP アドレスフィールド = NAT プレフィクス（最初の 96 ビット）とクライアント _IPv4（最後の 32 ビット）の連結 = 2001：DB 8:90:192.0.2.60
4. アプライアンスは、変換された IPv6 要求を Sevr-IPv6 に送信します。
5. IPv6 サーバー S1 は、IPv6 パケットを Citrix ADC アプライアンスに送信して応答します。
   - 宛先 IP アドレスフィールド = NAT プレフィクス（最初の 96 ビット）とクライアント _IPv4（最後の 32 ビット）の連結 = 2001：DB 8:90:192.0.2.60
   - 送信元 IP アドレスフィールド = サーバ-IPv6 = 2001: DB 8:5001:30
6. アプライアンスは IPv6 応答パケットを受信し、宛先 IP アドレスがアプライアンスで設定された NAT46 プレフィクスと一致することを確認します。宛先アドレスは NAT46 プレフィクスと一致するため、アプライアンスは NAT46 INAT エントリで、Sevr-IPv6 アドレスに関連付けられた IPv4 アドレスを検索します（2001: DB 8:5001:30）。これは、Map-Sevr-IPv4 アドレス（192.0.2.180）を検索します。
7. アプライアンスは、次の項目を使用して IPv4 応答パケットを作成します。
   - 宛先 IP アドレスフィールド = IPv6 応答の宛先アドレスから削除された NAT46 プレフィクス = クライアント _IPv4 (192.0.2.60)
   - 送信元 IP アドレスフィールド = Map-Sevr-IPv4 アドレス（192.0.2.180）
8. アプライアンスは、変換された IPv4 応答をクライアント CL1 に送信します。

ステートレス **NAT46** の制限事項

ステートレス NAT46 には、次の制限事項が適用されます。

- IPv4 オプションの変換はサポートされていません。
- IPv6 ルーティングヘッダーの変換はサポートされていません。
- IPv6 パケットのホップバイホップ拡張ヘッダーの変換はサポートされていません。
- IPv4 パケットの ESP および EH ヘッダーの変換はサポートされていません。
- マルチキャストパケットの変換はサポートされていません。
- 宛先オプションヘッダーおよび送信元ルーティングヘッダーの変換はサポートされていません。
- UDP チェックサムを含まないフラグメント化された IPv4 UDP パケットの変換はサポートされません。

ステートレス **NAT46** の設定

Citrix ADC アプライアンスでステートレス NAT46 構成に必要なエンティティを作成するには、次の手順に従います。

1. ステートレスモードを有効にして、IPv4-IPv6 マッピング INAT エントリを作成します。
2. NAT46 IPv6 プレフィクスを作成します。

CLI のプロシージャ

CLI を使用して INAT マッピングエントリを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add inat <name> <publicIPv4> <privateIPv6> -mode STATELESS
- show inat <name>

CLI を使用して NAT46 プレフィクスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- set inatparam -nat46v6Prefix <ipv6_addr|*>
- show inatparam

例**:**

```
 > add inat exmpl-com-stls-nat46 192.0.2.180
2001:DB8:5001::30 -mode stateless
Done

> set inatparam -nat46v6Prefix 2001:DB8:90::/96
Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して INAT マッピングエントリを作成するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [ルート] > [INAT] に移動します。
2. 新しい INAT エントリを追加するか、既存の INAT エントリを編集します。
3. 次のパラメーターを設定します。
    - Name*
    - パブリック IP アドレス *
    - プライベート IP アドレス * ([IPv6] チェックボックスをオンにして、IPv6 形式でアドレスを入力します。)
    - モード (ドロップダウンリストから [ステートレス] を選択します。)

    * 必須パラメータ

GUI を使用して NAT46 プレフィクスを作成するには、次の手順を実行します。

システムに移動 > **[**ネットワーク**]** の **[**設定**]** グループで、[**INAT** パラメーターの構成] をクリックし、[プレフィックス**]** パラメーターを設定します。

ステートレス **NAT46** のグローバルパラメータの設定

アプライアンスには、ステートレス NAT46 設定用のオプションのグローバルパラメータがいくつか用意されています。

CLI を使用してステートレス NAT46 のグローバルパラメータを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

set inatparam [-**nat46IgnoreTOS** (**YES**	**NO**)] [-**nat46ZeroCheckSum** (**ENABLED**	**DISABLED**)] [-**nat46v6Mtu** <positive_integer>] [-**nat46FragHeader** (**ENABLED**	**DISABLED**)]

-
- **show inatparam**

例:

```
> set inatparam -nat46IgnoreTOS YES -nat46ZeroCheckSum DISABLED -
    nat46v6Mtu 1400 -nat46FragHeader DISABLED
 Done
<!--NeedCopy-->
```

GUI を使用してステートレス NAT46 のグローバルパラメータを設定するには、次の手順を実行します。

[システム] > [ネットワーク] に移動し、[設定] グループで [**INAT** パラメータの設定] をクリックします。

DNS64

October 7, 2021

Citrix ADC DNS64 機能は、IPv4 のみのドメインに対して AAAA 要求を送信する IPv6 クライアントに対して、合成された DNS AAAA レコードで応答します。DNS64 機能は NAT64 機能と共に使用され、IPv6 のみのクライアントと IPv4 のみのサーバー間のシームレスな通信が可能になります。DNS64 では、IPV6 のみのクライアントによる IPv4 ドメインの検出が可能になり、NAT64 ではクライアントとサーバー間の通信が可能になります。

AAAA レコードを合成するために、Citrix ADC アプライアンスは DNS サーバーから DNS A レコードをフェッチします。DNS64 プレフィックスは、Citrix ADC アプライアンス上で構成された 96 ビットの IPv6 プレフィックスで

す。Citrix ADC アプライアンスは、DNS64 プレフィックス（96 ビット）と IPv4 アドレス（32 ビット）を連結して AAAA レコードを合成します。

IPv6 クライアントと IPv4 サーバー間の通信を可能にするために、DNS64 および NAT64 構成の Citrix ADC アプライアンスは、IPv6 クライアント側または IPv4 サーバー側のいずれかに展開できます。どちらの場合も、Citrix ADC アプライアンスの DNS64 構成は似ており、DNS サーバーのプロキシサーバーとして機能する負荷分散仮想サーバーが含まれます。Citrix ADC アプライアンスをクライアント側に展開する場合は、負荷分散仮想サーバーを IPv6 クライアント上でドメインのネームサーバーとして指定する必要があります。

DNS64 および NAT64 構成を持つ Citrix ADC アプライアンスが IPv4 側で構成されている例を考えてみましょう。この例では、企業が IPv4 アドレスを持つサーバ S1 上のサイト www.example.com をホストします。IPv6 クライアントと IPv4 サーバー S1 間の通信を可能にするために、Citrix ADC アプライアンス NS1 は DNS64 およびステートフル NAT64 構成で展開されます。

DNS64 構成には、DNS ロードバランシング仮想サーバー LBVS-DNS64-1 が含まれています。このサーバーでは、DNS64 オプションが有効になっています。DNS64 ポリシー 1 という名前の DNS64 ポリシーと、関連する DNS64 アクション 1 というアクションも NS1 上で構成され、DNS64 ポリシー 1 は LBVS-DNS64-1 にバインドされます。LBVS-DNS64-1 は、DNS サーバー DNS-1 および DNS-2 の DNS プロキシサーバーとして機能します。

LBVS-DNS64-1 に到着するトラフィックが DNS64-Policy 1 で指定された条件と一致する場合、トラフィックは DNS64-アクション 1 の設定に従って処理されます。DNS64-Action-1 は、AAAA レコードを合成するために DNS サーバから受信した A レコードと共に使用される DNS64 プレフィクスを指定します。

Citrix ADC アプライアンスでグローバル DNS パラメータキャッシュコードが有効になっているため、アプライアンスは DNS レコードをキャッシュします。この設定は、DNS64 が正常に動作するために必要です。

次の表に、上記の例で使用した設定を示します。DNS64 の設定例。

次に、この例のトラフィックフローを示します。

1. IPv6 クライアント CL1 は、サイト www.example.com の IPv6 アドレスに対する DNS AAAA 要求を送信します。
2. この要求は、Citrix ADC アプライアンス NS1 上の DNS 負荷分散仮想サーバー LBVS-DNS64-1 によって受信されます。
3. NS1 は、要求された AAAA レコードについて DNS キャッシュレコードをチェックし、サイト www.example.com の AAAA レコードが DNS キャッシュに存在しないことを検出します。
4. LBVS-DNS64-1 の負荷分散アルゴリズムは、DNS サーバー DNS-1 を選択し、AAAA 要求をそれに転送します。
5. サイト www.example.com は IPv4 サーバーでホストされているため、DNS サーバー DNS-1 には、サイト www.example.com の AAAA レコードがありません。
6. DNS-1 は、空の DNS AAAA 応答またはエラーメッセージを LBVS-DNS64-1 に送信します。
7. LBVS-DNS64-1 で DNS64 オプションが有効になっており、CL1 からの AAAA 要求は DNS64 ポリシー 1 で指定された条件と一致するため、NS1 は www.example.com の IPv4 アドレスに対する DNS A 要求を DNS-1 に送信します。

8. DNS-1 は、www.example.com の DNS A レコードを LBVS-DNS64-1 に送信することで応答します。A レコードには、www.example.com の IPv4 アドレスが含まれます。
9. NS1 は、サイト www.example.com の AAAA レコードを次のように合成します。
   - サイト www.example.com の IPv6 アドレス = 関連付けられた DNS64 アクションで指定された DNS64 プレフィックス (96 ビット) の連結、DNS A レコードの IPv4 アドレス (32 ビット) = 2001: DB 8:300:: 192.0.2.60
10. NS1 は、統合された AAAA レコードを IPv6 クライアント CL1 に送信します。NS1 は、A レコードをメモリにキャッシュします。NS1 は、キャッシュされた A レコードを使用して、後続の AAAA 要求に対して AAAA レコードを合成します。

DNS64 構成で考慮すべきポイント

Citrix ADC アプライアンスで DNS64 を構成する前に、次の点を考慮してください。

- Citrix ADC アプライアンスの DNS64 機能は、RFC 6174 に準拠しています。
- Citrix ADC アプライアンスの DNS64 機能は、DNSSEC をサポートしていません。Citrix ADC アプライアンスは、DNS サーバーから受信した DNSSEC 応答からの AAAA レコードを合成しません。応答は、RRSIG レコードが含まれている場合のみ、DNSSEC 応答として分類されます。
- Citrix ADC アプライアンスは、96 ビットの長さの DNS64 プレフィックスをサポートしています。
- DNS64 機能は NAT64 機能とともに使用されていますが、DNS64 および NAT64 構成は Citrix ADC アプライアンスに依存しません。特定のフローでは、クライアントが受信した合成 IPv6 アドレスが特定の NAT64 設定にルーティングされるように、DNS64 プレフィクスと NAT64 プレフィクスパラメータに同じ IPv6 プレフィクス値を指定する必要があります。Citrix ADC アプライアンスでの NAT64 の構成の詳細については、ステートフル NAT64 を参照してください。
- Citrix ADC アプライアンスによる DN64 処理の異なるケースを次に示します。
  - DNS サーバーからの AAAA 応答に AAAA レコードが含まれている場合、応答の各レコードに、Citrix ADC アプライアンス上で特定の DNS64 構成に対して構成された除外ルールのセットがチェックされます。Citrix ADC は、除外ルールと一致する接頭辞を持つ IPv6 アドレスを応答から削除します。結果の応答に少なくとも 1 つの IPv6 レコードが含まれている場合、Citrix ADC アプライアンスはこの応答をクライアントに転送します。それ以外の場合は、アプライアンスはドメインの A レコードからの AAAA 応答を合成し、IPv6 クライアントに送信します。
  - DNS サーバからの AAAA 応答が空の応答応答である場合、アプライアンスは同じドメイン名の A リソースレコードを要求するか、アプライアンスがドメインの正規のドメインネームサーバである場合は、自身のレコードを検索します。要求の結果が空の応答またはエラーになった場合、同じ要求がクライアントに転送されます。
  - DNS サーバーからの応答に RCODE=1（フォーマットエラー）が含まれている場合、Citrix ADC アプライアンスは同じものをクライアントに転送します。タイムアウト前に応答がない場合、Citrix ADC アプライアンスは RCODE=2（サーバー障害）の応答をクライアントに送信します。

  - DNS サーバからの応答に CNAME が含まれる場合、終端の A レコードまたは AAAA レコードに到達するまでチェーンが続きます。CNAME に AAAA リソースレコードがない場合、Citrix ADC アプライアンスは、AAAA レコードの合成に使用する DNS A レコードを取得します。CNAME チェーンは、合成された AAAA レコードとともに応答セクションに追加され、クライアントに送信されます。

- Citrix ADC アプライアンスの DNS64 機能は、PTR 要求への応答もサポートしています。IPv6 アドレスのドメインに対する PTR 要求がアプライアンスで受信され、IPv6 アドレスが設定された DNS64 プレフィクスのいずれかと一致すると、アプライアンスは IP6-ARPA ドメインを対応する IN-ADDR にマッピングする CNAME レコードを作成します。ARPA ドメインと、新たに形成された IN-ADDR.ARPA ドメインが解決に使用されます。アプライアンスはローカルの PTR レコードを検索し、そのレコードが存在しない場合、アプライアンスは IN-ADDR.ARPA ドメインに対する PTR 要求を DNS サーバに送信します。Citrix ADC アプライアンスは、DNS サーバーからの応答を使用して、最初の PTR 要求に対する応答を合成します。

構成の手順

Citrix ADC アプライアンスでステートフル NAT64 構成に必要なエンティティを作成するには、次の手順に従います。

- **DNS** サービスを追加します。DNS サービスは、Citrix ADC アプライアンスが DNS プロキシサーバーとして機能する DNS サーバーを論理的に表現したものです。サービスのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。
- **DNS64** アクションと **DNS64** ポリシーを追加し、**DNS64** アクションを **DNS64** ポリシーにバインドします。DNS64 ポリシーは、関連する DNS64 アクションの設定に従って、DNS64 処理のトラフィックと照合する条件を指定します。DNS64 アクションは、必須の DNS64 プレフィクスと、オプションの除外ルールとマッピングされたルールの設定を指定します。
- **DNS** 負荷分散仮想サーバーを作成し、**DNS** サービスと **DNS64** ポリシーをバインドします。DNS 負荷分散仮想サーバーは、バインドされた DNS サービスによって表される DNS サーバーの DNS プロキシサーバーとして機能します。仮想サーバに着信するトラフィックは、DNS64 処理用のバインドされた DNS64 ポリシーと照合されます。負荷分散仮想サーバーのオプションパラメータの設定の詳細については、「負荷分散」を参照してください。

  注：CLI には、これら 2 つのタスクに対して個別のコマンドがありますが、GUI では 1 つのダイアログボックスにまとめられています。

  DNS レコードのキャッシュを有効にします。Citrix ADC アプライアンスのグローバルパラメータを有効にして、DNS プロキシ操作によって取得される DNS レコードをキャッシュします。DNS レコードのキャッシュの有効化の詳細については、「ドメインネームシステム」を参照してください。

CLI のプロシージャ

CLI を使用して DNS タイプのサービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add service <name> <IP> <serviceType> <port> …

CLI を使用して DNS64 アクションを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <expression>] [-excludeRule <expression>]

CLI を使用して DNS64 ポリシーを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add dns policy64 <name> -rule <expression> -action <string>

CLI を使用して DNS ロードバランシング仮想サーバを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED) [-bypassAAAA (YES | NO)] …

CLI を使用して DNS サービスと DNS64 ポリシーを DNS ロードバランシング仮想サーバにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- bind lb vserver <name> <serviceName> …
- bind lb vserver <name> -policyName <string> -priority <positive_integer> …

GUI のプロシージャ

GUI を使用して DNS タイプのサービスを作成するには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、新しいサービスを追加します。
2. 次のパラメーターを設定します。
    - サービス名 *
    - サーバ *
    - プロトコル *（ドロップダウンリストから DNS を選択します）
    - ポート *

GUI を使用して DNS64 アクションを作成するには、次の手順を実行します。

[トラフィック管理] > [DNS] > [アクション] に移動し、[DNS アクション 64] タブで、新しい DNS64 アクションを追加します。

GUI を使用して DNS64 ポリシーを作成するには、次の手順を実行します。

[トラフィック管理] > [DNS] > [ポリシー] に移動し、[DNS ポリシー 64] タブで、新しい DNS64 ポリシーを追加します。

GUI を使用して DNS ロードバランシング仮想サーバーを作成し、DNS サービスと DNS64 ポリシーをバインドするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、新しい仮想サーバーを追加します。
2. 次のパラメーターを設定します。
   - Name*
   - IP アドレス *
   - プロトコル *（ドロップダウンリストから DNS を選択します）
   - ポート *
3. [DNS64 を有効にする] オプションを選択します。
4. [サービス] ペインで、サービスを仮想サーバにバインドします。
5. [Policies] ペインで、ポリシーを仮想サーバにバインドします。

構成例

```
 > add service SVC-DNS-1 203.0.113.50 DNS 53
 Done

> add service SVC-DNS-2 203.0.113.60 DNS 53
 Done

> add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
 Done

> add dns Policy64 DNS64-Policy-1 -rule ”CLIENT.IPv6.SRC.IN_SUBNET
   (2001:DB8:5001::/64)”
-action DNS64-Action-1
 Done

> add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-1
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-2
 Done

> bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
 Done

<!--NeedCopy-->
```


ステートフル **NAT64** 変換

October 7, 2021

ステートフルな NAT64 機能を使用すると、Citrix ADC アプライアンスでセッション情報を維持しながら、IPv6 から IPv4 へのパケット変換（またはその逆）を介して IPv6 クライアントと IPv4 サーバー間の通信が可能になります。

Citrix ADC アプライアンスのステートフルな NAT64 構成には、次のコンポーネントがあります。

- **NAT64** ルール— ACL6 ルールとネットプロファイルで構成されるエントリで、Citrix ADC が所有する SNIP アドレスのプールで構成されます。
- **NAT64 IPv6** プレフィクス — アプライアンス上で構成されている長さ 96 ビット（128 〜32 = 96）のグローバル IPv6 プレフィクス。
  注：現在、Citrix ADC アプライアンスは、すべての NAT 64 ルールで一般的に使用される接頭辞を 1 つだけサポートしています。

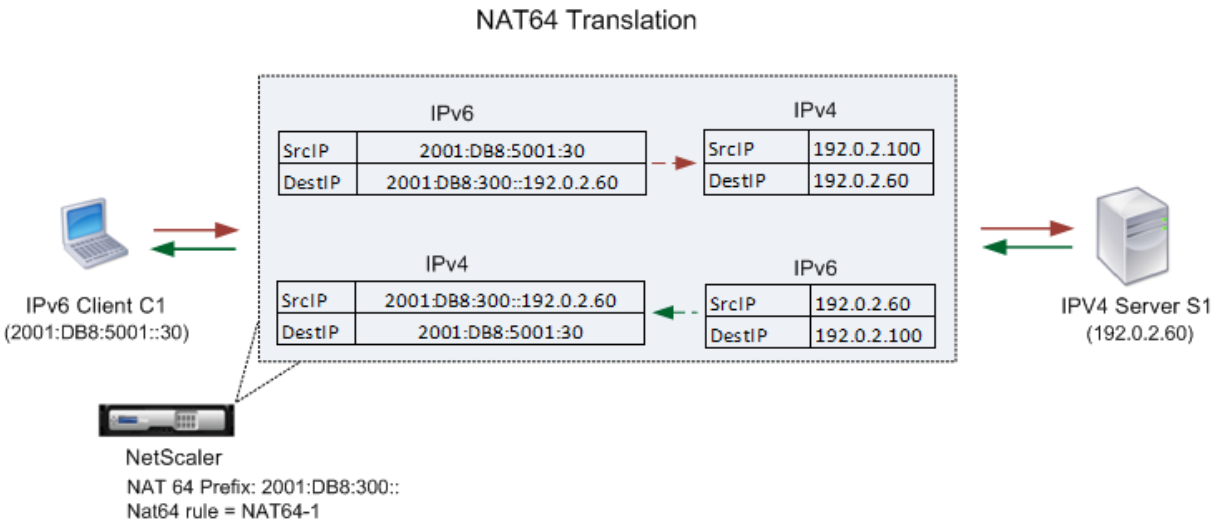
Citrix ADC アプライアンスは、次の条件がすべて満たされた場合に、NAT64 変換用の受信 IPv6 パケットを考慮します。

- 着信 IPv6 パケットは、NAT64 ルールにバインドされた ACL6 ルールと一致します。
- IPv6 パケットの宛先 IP アドレスは、NAT64 IPv6 プレフィクスと一致します。

Citrix ADC アプライアンスによって受信された IPv6 要求パケットが NAT64 ルールで定義された ACL6 と一致し、パケットの宛先 IP が NAT64 IPv6 プレフィックスと一致する場合、Citrix ADC アプライアンスは IPv6 パケットを変換対象とみなします。

アプライアンスは、この IPv6 パケットを、NAT64 ルールで定義されたネットプロファイルにバインドされた IP アドレスと一致する送信元 IP アドレスと、IPv6 要求パケットの宛先 IPv6 アドレスの最後の 32 ビットで構成される宛先 IP アドレスを持つ IPv4 パケットに変換します。Citrix ADC アプライアンスは、この特定のフローの NAT64 セッションを作成し、IPv4 サーバーにパケットを転送します。IPv4 サーバからの後続の応答と IPv6 クライアントからの要求は、特定の NAT64 セッションの情報に基づいてアプライアンスによって変換されます。

たとえば、企業が IPv4 アドレスを持つサーバ S1 上でサイト www.example.com をホストしているとします。IPv6 クライアントと IPv4 サーバー S1 間の通信を可能にするために、Citrix ADC アプライアンス NS1 は、NAT64 ルールと NAT64 プレフィックスを含むステートフル NAT64 構成で展開されます。サーバ S1 のマッピングされた IPv6 アドレスは、NAT64 IPv6 プレフィックス [96 ビットと IPv4 送信元アドレス [32 ビットを連結することによって形成されます]]。このマッピングされた IPv6 アドレスは、DNS サーバで手動で構成されます。IPv6 クライアントは、IPv4 サーバ S1 と通信するために DNS サーバからマッピングされた IPv6 アドレスを取得します。



次の表に、この例で使用される設定を示します。ステートフル NAT64 変換の例の設定。

次に、この例のトラフィックフローを示します。

1. IPv6 クライアント CL1 は、マップサーバ-IPv6（2001: DB 8:300:: 192.0.2.60）アドレスに要求パケットを送信します。
2. Citrix ADC アプライアンスは要求パケットを受信します。要求パケットが NAT64 ルールで定義された ACL6 と一致し、パケットの宛先 IP アドレスが NAT64 IPv6 プレフィックスと一致する場合、Citrix ADC は IPv6 パケットを変換対象とみなします。
3. アプライアンスは、次のように変換された IPv4 要求パケットを作成します。
   - IPv6 要求の宛先アドレスから削除された NAT64 プレフィクスを含む宛先 IP アドレスフィールド（Sevr_IPv4 = 192.0.2.60）
   - Netprofile-1 にバインドされた IPv4 アドレスのいずれかを含む送信元 IP アドレスフィールド（この例では 192.0.2.100）
4. Citrix ADC アプライアンスは、このフローの NAT64 セッションを作成し、変換された IPv4 要求をサーバー S1 に送信します。
5. IPv64 サーバー S1 は、以下の方法で IPv4 パケットを Citrix ADC アプライアンスに送信することで応答します。
   - 192.0.2.100 を含む宛先 IP アドレスフィールド
   - Sevr_IPv4 のアドレスを含む送信元 IP アドレスフィールド（192.0.2.60）
6. アプライアンスは IPv4 応答パケットを受信し、すべてのセッションエントリを検索し、IPv6 応答パケットがステップ 4 で作成した NAT64 セッションエントリと一致することを検出します。アプライアンスは IPv4 パケットを変換対象と見なします。
7. アプライアンスは、次のように変換された IPv6 応答パケットを作成します。
   - 宛先 IP アドレスフィールド = クライアント IPv6 = 2001: DB 8:5001:30

- 送信元 IP アドレスフィールド = NAT64 プレフィクス（最初の 96 ビット）とサーバ _IPv4（最後の 32 ビット）の連結 = 2001: DB 8:300:: 192.0.2.60

8. アプライアンスは、変換された IPv6 応答をクライアント CL1 に送信します。

ステートフル **NAT64** の制限事項

ステートフル NAT64 には、次の制限が適用されます。

- IPv4 オプションの変換はサポートされていません。
- IPv6 ルーティングヘッダーの変換はサポートされていません。
- IPv6 パケットのホップバイホップ拡張ヘッダーの変換はサポートされていません。
- IPv6 パケットの ESP および EH ヘッダーの変換はサポートされていません。
- マルチキャストパケットの変換はサポートされていません。
- Packets of Stream Control Transmission Protocol (SCTP)、Datagram Congestion Control Protocol (DCCP)、および IPSec のパケットは変換されません。

ステートフル **NAT64** の設定

Citrix ADC アプライアンスでステートフル NAT64 構成に必要なエンティティを作成するには、次の手順に従います。

1. アクションが許可された ACL6 ルールを追加します。
2. 複数の IP アドレスをバインドする ipset を追加します。
3. ネットプロファイルを追加し、ipset をバインドします。1 つの IP アドレスだけをバインドする場合は、ipset エンティティを作成する必要はありません。その場合は、IP アドレスを netprofile に直接バインドします。
4. NAT64 ルールを追加します。これには、ACl6 ルールとネットプロファイルを NAT 64 ルールにバインドすることが含まれます。
5. NAT64 IPv6 プレフィクスを追加します。

CLI のプロシージャ

CLI を使用して ACL6 ルールを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns acl6 <acl6name> <acl6action> …

CLI を使用して IP セットを追加し、複数の IP をバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ipset <name>
- bind ipset <name> <IPaddress …>

CLI を使用してネットプロファイルを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add netprofile <name> -srcIP <IPaddress or IPset>

CLI を使用して NAT64 ルールを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add nat64 <name> <acl6name> -netProfile <string>

CLI を使用して NAT64 プレフィクスを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- set ipv6 -natprefix <ipv6_addr|*>

例:

```
 > add acl6 ACL6-1 ALLOW -srcIPv6 2001:DB8:5001::30
 Done

> apply acls6
 Done

> add ip 192.0.2.100 255.255.255.0  - type SNIP
 Done

> add ip 192.0.2.102 255.255.255.0   - type SNIP
 Done

> add ipset IPset-1
 Done

> bind ipset IPset-1 192.0.2.100 192.0.2.102
IPAddress ”192.0.2.100” bound
IPAddress ”192.0.2.102” bound
 Done

> add netprofile Netprofile-1 -srcIP IPset-1
 Done

> add nat64 NAT64-1 ACL6-1 -netprofile Netprofile-1
 Done

> set ipv6 -natprefix 2001:DB8:300::/96
 Done
```



```
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して NAT64 ルールを追加するには、次の手順を実行します。

[システム] > [ネットワーク] > [ルート] > [NAT64] に移動して、新しい NAT64 ルールを作成するか、既存のルールを編集します。

GUI を使用して NAT64 プレフィクスを追加するには、次の手順を実行します。

システムに移動 > [ネットワーク] の [設定] グループで、[INAT パラメーターの構成] をクリックし、[プレフィックス] パラメーターを設定します。

RNAT

October 7, 2021

リバースネットワークアドレス変換（RNAT）では、Citrix ADC アプライアンスは、サーバーによって生成されたパケットの送信元 IP アドレスをパブリック NAT IP アドレスに置き換えます。デフォルトでは、アプライアンスは SNIP アドレスを NAT IP アドレスとして使用します。サブネットごとに一意の NAT IP アドレスを使用するようにアプライアンスを設定することもできます。アクセスコントロールリスト（ACL）を使用して RNAT を設定することもできます。[送信元 IP (USIP) を使用]、[サブネット IP (USNIP) を使用]、および [リンクロードバランシング (LLB)] の各モードは、RNAT の動作に影響します。統計情報を表示して、RNAT を監視できます。

注： Citrix ADC アプライアンスでの RNAT のエフェメラルポート範囲は、1024～65535 です。

RNAT エントリの条件として、ネットワークアドレスまたは拡張 ACL のいずれかを使用できます。

- ネットワークアドレスを使用する。ネットワークアドレスを使用すると、指定されたネットワークから着信するすべてのパケットに対して RNAT 処理が実行されます。
- 拡張 **ACL** の使用。ACL を使用すると、ACL に一致するすべてのパケットに対して RNAT 処理が実行されます。ACL に一致するトラフィックに一意の IP アドレスを使用するように Citrix ADC アプライアンスを設定するには、次の 3 つのタスクを実行する必要があります。

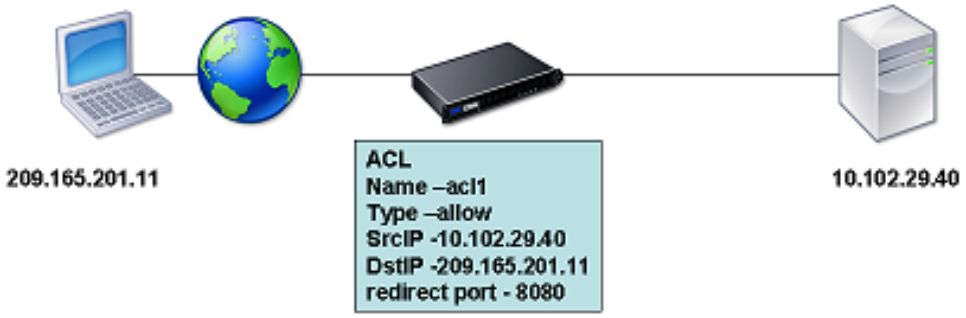
  1. ACL を設定します。
  2. 送信元 IP アドレスと宛先ポートを変更するように RNAT を設定します。
  3. ACL を適用します。

  次の図は、ACL が設定された RNAT を示しています。

  図 1: ACL を使用した RNAT

Source IP Address	Destination IP Address	Destination Port
209.165.202.129	209.165.201.11	8080

Source IP Address	Destination IP Address
10.102.29.40	209.165.201.11

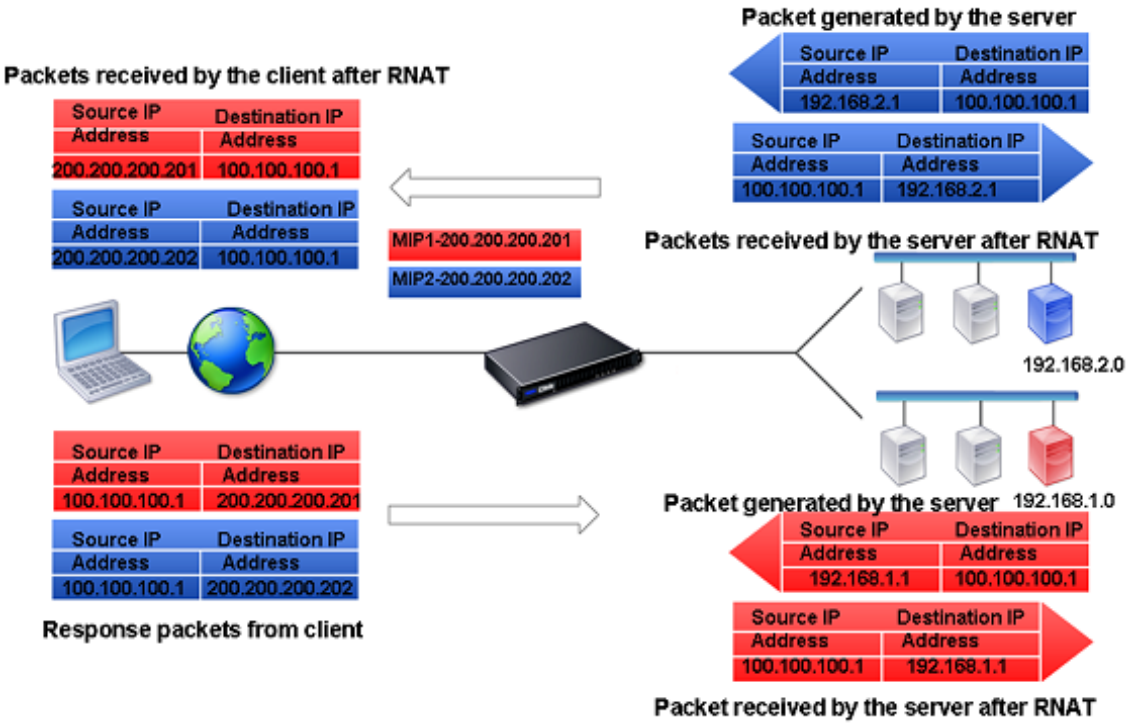


NAT IP アドレスのタイプには、次の基本的な選択肢があります。

- **SNIP** を **NAT IP** アドレスとして使用する。SNIP を NAT IP アドレスとして使用する場合、Citrix ADC アプライアンスはサーバー生成パケットの送信元 IP アドレスを SNIP に置き換えます。したがって、SNIP アドレスはパブリック IP アドレスである必要があります。サブネット IP（USNIP）モードが有効になっている場合、Citrix ADC はサブネット IP アドレス（SNIP）を NAT IP アドレスとして使用できます。
- 一意の **IP** アドレスを **NAT IP** アドレスとして使用する。一意の IP アドレスを NAT IP アドレスとして使用する場合、Citrix ADC アプライアンスは、サーバーによって生成されたパケットの送信元 IP アドレスを、指定された一意の IP アドレスに置き換えます。一意の IP アドレスは、Citrix ADC が所有するパブリック IP アドレスである必要があります。サブネットに対して複数の NAT IP アドレスが設定されている場合、NAT IP 選択ではラウンドロビンアルゴリズムが使用されます。

  この構成を次の図に示します。

  図 2：一意の IP アドレスを NAT IP アドレスとして使用する



はじめに

RNAT ルールを設定する前に、次の点を考慮してください。

- Citrix ADC アプライアンスで RNAT と USIP（ソース IP を使用）の両方が構成されている場合、RNAT が優先されます。つまり、RNAT ルールに一致するパケットの送信元 IP アドレスは、RNAT ルールの設定に従って置き換えられます。
- Citrix ADC アプライアンスがサーバーからのトラフィックに対してリンク負荷分散（LLB）と RNAT の両方を実行するトポロジでは、アプライアンスはルーターに基づいて送信元 IP アドレスを選択します。LLB 設定によって、ルータの選択が決まります。LLB の詳細については、リンクロードバランシングを参照してください。

RNAT の設定

次の手順では、異なる条件および異なるタイプの NAT IP アドレスを使用する RNAT エントリを作成するための個別のコマンドライン手順を示します。GUI では、すべてのバリエーションを同じダイアログ・ボックスで構成できるため、GUI ユーザーには 1 つの手順しかありません。

CLI のプロシージャ

CLI を使用して RNAT ルールを作成するには、次の手順を実行します。

コマンドプロンプトで、ルールを作成して構成を確認するには、次のように入力します。

- add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))
- bind rnat <name> <natIP>@ …
- show rnat

CLI を使用して RNAT ルールを変更または削除するには、次の手順を実行します。

- RNAT ルールを変更するには、次の手順に従います。

  set rnat <name> (<aclname> [-redirectPort <port>])

- RNAT ルールを削除するには、コマンドを入力します。

  rm rnat <name>

次のコマンドを使用して、設定を確認します。

- show rnat

例:

```
A network address as the condition and a SNIP address as the NAT IP address:

> add rnat RNAT-1 192.168.1.0 255.255.255.0
Done

A network address as the condition and a unique IP address as the NAT IP address:

> add rnat RNAT-2 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-2 -natip 10.102.29.50
Done

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the Citrix ADC-owned IP addresses, except the NSIP, that fall within the range specified:

> add rnat RNAT-3 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-3 -natip 10.102.29.[50-110]
Done


An ACL as the condition and a SNIP address as the NAT IP address:
```



```
24
25 > add rnat RNAT-4 acl1
26 Done
27
28 An ACL as a condition and a unique IP address as the NAT IP address:
29
30 > add rnat RNAT-4 acl1
31 Done
32
33 > bind rnat RNAT-4 -natip 10.102.29.50
34 Done
35
36 If instead of a single NAT IP address you specify a range, RNAT entries
       are created with all the Citrix ADC-owned IP addresses, except the
      NSIP, that fall within the range specified:
37
38 > add rnat RNAT-5 acl1
39 Done
40
41 > bind rnat RNAT-5 -natip 10.102.29.[50-70]
42 Done
43
44 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して RNAT エントリを作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT**] タブをクリックして新しい RNAT ルールを追加するか、既存のルールを編集します。

RNAT の監視

RNAT 統計情報を表示して、IP アドレス変換に関連する問題をトラブルシューティングできます。

次の表に、RNAT および RNAT IP に関連する統計情報を示します。

統計情報	説明
受信バイト数	RNAT セッション中に受信したバイト数
送信されたバイト数	RNAT セッション中に送信されたバイト数
受信したパケット	RNAT セッション中に受信されたパケット
送信されたパケット	RNAT セッション中に送信されるパケット

統計情報	説明
送信された Syn	RNAT セッション中に送信される接続の要求
現在のセッション	現在アクティブな RNAT セッション

CLI を使用して RNAT 統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat rnat**

例**:**

```
> stat rnat

RNAT summary
                                Rate (/s)            Total
Bytes Received                    0                    0
Bytes Sent                        0                    0
Packets Received                  0                    0
Packets Sent                      0                    0
Syn Sent                          0                    0
Current RNAT sessions            --                    0
 Done
>
<!--NeedCopy-->
```

GUI を使用して RNAT を監視するには、次の手順を実行します。

[システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT**] タブをクリックし、[統計] をクリックします。

RNAT6 の構成

IPv6 パケットのリバースネットワークアドレス変換 (RNAT) ルールは、RNAT6 と呼ばれます。サーバーによって生成された IPv6 パケットが RNAT6 ルールで指定された条件に一致すると、アプライアンスは IPv6 パケットの送信元 IPv6 アドレスを構成済みの NAT IPv6 アドレスに置き換えてから、宛先に転送します。NAT IPv6 アドレスは、Citrix ADC が所有する SNIP6 または VIP6 アドレスのいずれかです。

RNAT6 ルールを設定する場合、条件として IPv6 プレフィクスまたは ACL6 を指定できます。

- **IPv6** ネットワークアドレスを使用する。IPv6 プレフィックスを使用する場合、アプライアンスは IPv6 アドレスがプレフィックスと一致する IPv6 パケットに対して RNAT 処理を実行します。

- **ACL6** を使用します。ACL6 を使用する場合、アプライアンスは ACL6 で指定された条件に一致する IPv6 パケットに対して RNAT 処理を実行します。

NAT IP アドレスを設定するには、次のいずれかのオプションがあります。

- RNAT6 ルールに対して、Citrix ADC が所有する SNIP6 アドレスと VIP6 アドレスのセットを指定します。Citrix ADC アプライアンスは、このセットの IPv6 アドレスのいずれかを、各セッションの NAT IP アドレスとして使用します。選択はラウンドロビンアルゴリズムに基づいて行われ、セッションごとに行われます。
- RNAT6 ルールには、Citrix ADC が所有する SNIP6 アドレスまたは VIP6 アドレスを指定しないでください。Citrix ADC アプライアンスは、Citrix ADC が所有する SNIP6 アドレスまたは VIP6 アドレスのいずれかを NAT IP アドレスとして使用します。選択は、RNAT ルールに一致する IPv6 パケットの宛先となるネクストホップネットワークに基づいて行われます。

CLI のプロシージャ

CLI を使用して RNAT6 ルールを作成するには、次の手順を実行します。

コマンドプロンプトで、ルールを作成して構成を確認するには、次のように入力します。

- **add rnat6** <name> (<network> | (<acl6name> [-**redirectPort** <port>]))
- **bind rnat6** <name> <natIP6>@ …
- **rnat6**

CLI を使用して RNAT6 ルールを変更または削除するには、次の手順を実行します。

- 条件が ACL6 である RNAT6 ルールを変更するには、**set rnat6** <name> コマンドを入力し、続いて **redirectPort** パラメータの新しい値を入力します。
- RNAT6 ルールを削除するには、**clear rnat6** <name> コマンドを入力します。

GUI のプロシージャ

GUI を使用して RNAT6 ルールを設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT6**] タブをクリックして新しい RNAT6 ルールを追加するか、既存のルールを編集します。

モニター **RNAT6**

RNAT6 機能に関連する統計情報を表示して、パフォーマンスを監視したり、RNAT6 機能に関連する問題をトラブルシューティングしたりできます。RNAT6 ルールまたは特定の RNAT6 ルールの統計情報のサマリーを表示できます。統計カウンタには、Citrix ADC アプライアンスが最後に再起動されてからのイベントが反映されます。Citrix ADC アプライアンスが再起動されると、これらのカウンタはすべて 0 にリセットされます。

次に、RNAT6 機能に関連する統計カウンタの一部を示します。

- 受信バイト数-RNAT6 セッション中に受信した合計バイト数。
- 送信バイト数-RNAT6 セッション中に送信された合計バイト数。
- 受信パケット -RNAT6 セッション中に受信されたパケットの総数。
- 送信されたパケット -RNAT6 セッション中に送信されたパケットの総数。
- 送信された **Syn** -RNAT6 セッション中に送信された接続要求の合計数
- 現在のセッション -現在アクティブな RNAT6 セッション

CLI を使用してすべての RNAT6 ルールの要約統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat rnat6**

CLI を使用して、指定された RNAT6 ルールの統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat rnat6** [<rnat6 rule name>]

GUI を使用して RNAT6 統計情報を表示するには、次の手順を実行します。

[システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT6**] タブをクリックし、[統計] をクリックします。

```
> stat rnat6

RNAT6 summary

                           Rate (/s)                Total

Bytes Received              178                 20644

Bytes Sent                  178                 20644

Packets Received             5                   401

Packets Sent                 5                   401

Syn Sent                     0                     2

Current RNAT6 sessions      --                     1

Done

<!--NeedCopy-->
```


RNAT ログエントリのログ開始時刻と接続終了の理由をログに記録する

RNAT に関連する問題の診断またはトラブルシューティングのために、Citrix ADC アプライアンスは RNAT セッションを閉じるたびにログを記録します。

RNAT セッションのログメッセージは、次の情報で構成されます。

- ログメッセージの送信元である Citrix ADC が所有する IP アドレス（NSIP アドレスまたは SNIP アドレス）
- ログ作成のタイムスタンプ
- RNAT セッションのプロトコル
- 送信元 IP アドレス
- RNAT IP アドレス
- 宛先 IP アドレス
- RNAT セッションの開始時刻
- RNAT セッションの終了時刻
- この RNAT セッションで Citrix ADC アプライアンスによって送信された合計バイト数
- この RNAT セッションで Citrix ADC アプライアンスが受信した合計バイト数
- RNAT セッションが終了する理由。Citrix ADC アプライアンスは、アプライアンスの TCP プロキシ（TCP プロキシ無効）を使用しない TCP RNAT セッションの終了理由を記録します。TCP RNAT セッションで記録される終了理由のタイプを次に示します。
  - **TCP FIN**。送信元デバイスまたは宛先デバイスから TCP FIN が送信されたため、RNAT セッションが閉じられました。
  - **TCP RST**。RNAT セッションは、送信元デバイスまたは宛先デバイスによって送信された TCP リセットが原因で閉じられました。
  - **TIMEOUT**。RNAT セッションがタイムアウトしました。

次の表に、RNAT セッションのログエントリの例を示します。

エントリのタイプ	サンプルログエントリ
UDP RNAT セッションのサンプルログエントリ	Dec 1 15:28:12 10.102.53.114 12/01/2015:15:28:12 GMT 0-PPE-0 : default UDP NAT_OTHERCONN_DELINK 154 0 : Source 1.2.2.5:23431 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:4045 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:26:58 GMT - Delink Time 12/01/2015:15:28:12 GMT - Total_bytes_send 2511 - Total_bytes_recv 3725

エントリのタイプ	サンプルログエントリ
TCP RNAT セッションのサンプルログエントリ。ログエントリは、TCP リセットのためにセッションが閉じていることを示している	Dec 1 15:29:59 10.102.53.114 12/01/2015:15:27:59 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 152 0 : Source 1.2.2.5:33826 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:2384 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:27:40 GMT - Delink Time 12/01/2015:15:27:59 GMT - Total_bytes_send 2147 - Total_bytes_recv 3257 - Closure Reason TCP RST
TCP RNAT セッションのサンプルログエントリ。ログエントリは、セッションがタイムアウトしたことを示しています。	Dec 1 15:30:12 10.102.53.114 12/01/2015:15:30:12 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 155 0 : Source 1.2.2.5:64976 - Destination 192.168.123.115:22 - NatIP 192.168.123.1:19636 - Destination 192.168.123.115:22 - Start Time 12/01/2015:15:27:25 GMT - Delink Time 12/01/2015:15:30:12 GMT - Total_bytes_send 0 - Total_bytes_recv 0 - Closure Reason TIMEOUT

RNAT のステートフル接続フェールオーバー

接続フェイルオーバーは、分散環境にデプロイされたアプリケーションへのアクセスが中断されるのを防ぐのに役立ちます。Citrix ADC アプライアンスは、Citrix ADC 高可用性（HA）セットアップの RNAT ルールに関連する接続のステートフル接続フェイルオーバーをサポートするようになりました。HA セットアップでは、接続フェールオーバー（または接続ミラーリング）とは、フェールオーバーが発生したときに、確立された TCP または UDP 接続をアクティブに保つプロセスを指します。

プライマリアプライアンスは、セカンダリアプライアンスにメッセージを送信して、RNAT 接続に関する現在の情報を同期します。セカンダリアプライアンスは、フェールオーバーの場合にのみこの接続情報を使用します。フェイルオーバーが発生すると、新しいプライマリ Citrix ADC アプライアンスは、フェイルオーバー前に確立された接続に関する情報を保持するため、フェイルオーバー後もそれらの接続を処理し続けます。クライアントの観点からは、このフェイルオーバーは透過的です。移行期間中、クライアントとサーバーは短時間の中断や再送信が発生することがあります。

接続フェイルオーバーは、RNAT ルールごとに有効にできます。RNAT ルールで接続フェイルオーバーを有効にするには、CLI または GUI を使用して、特定の RNAT ルールの connFailover（接続フェイルオーバー）パラメータを有効にします。

CLI を使用して RNAT ルールの接続フェイルオーバーを有効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

- `set rnat <name> -connfailover (ENABLED | DISABLED)`
- `show rnat`

GUI を使用して RNAT ルールの接続フェイルオーバーを有効にするには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**NAT**] に移動し、[**RNAT**] タブをクリックします。
2. 新しい RNAT 規則の追加中、または既存の規則の編集中に [接続フェールオーバー] を選択します。

サーバーへの **RNAT** 接続用の送信元ポートを予約する

1 つ以上の RNAT IP アドレスとプロキシポートの使用パラメータが無効になっている RNAT 構成にヒットするリクエストの場合、Citrix ADC アプライアンスは RNAT IP アドレスと RNAT 要求のソースポートのいずれかを使用してサーバーに接続します。13.0 47.x ビルドより前のバージョンでは、同じソースポートが他の接続で既に使用されている場合、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は失敗します。

- 送信元ポートが **1024** 未満です。デフォルトでは、Citrix ADC アプライアンスは Citrix ADC 所有の IP アドレス（RNAT IP アドレスを含む）の最初の 1024 ポートを予約します。13.0 47.x ビルドより前のバージョンでは、RNAT 要求の送信元ポートが 1024 以下の場合、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は失敗します。13.0 47.x ビルドでは、RNAT 要求の送信元ポートが 1024 以下であっても、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は成功します。
- **1024** より大きい送信元ポート。13.0 47.x ビルドより前のバージョンでは、同じソースポートが他の接続で既に使用されている場合、サーバーへの RNAT 接続 (RNAT クライアントのソースポートを使用) は失敗します。13.0 47.x ビルドでは、RNAT 設定の一部として、`Retain Source Port range` (`retainsourceportrange`) パラメータで RNAT クライアントソースポートの範囲を指定できます。Citrix ADC アプライアンスは、サーバーへの RNAT 接続にのみ使用されるように、RNAT IP アドレス上のこれらの RNAT クライアントソースポートを予約します。

RNAT セッションの削除

不要な RNAT セッションや非効率的な RNAT セッションを Citrix ADC アプライアンスから削除できます。アプライアンスは、これらのセッションに割り当てられたリソース（NAT IP アドレスのポートやメモリなど）をただちに解放し、リソースを新しいセッションで使用できるようにします。アプライアンスは、これらの削除されたセッションに関連する後続のパケットもすべてドロップします。Citrix ADC アプライアンスからすべての RNAT セッションまたは選択した RNAT セッションを削除できます。

CLI を使用してすべての RNAT セッションをクリアするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **flush rnatsession**

CLI を使用して選択的 RNAT セッションをクリアするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **flush rnatsession** ((-**network** <ip_addr> -**netmask** <netmask>) | -**natIP** <ip_addr> | -**aclname** <string>)

GUI を使用してすべての RNAT セッションまたは選択的 RNAT セッションをクリアするには、次の手順を実行します。

1. システムに移動 > 通信網 > **NAT**、をクリックし、[**RNAT**] タブをクリックします。
2. [**Actions**] メニューで [**Flush RNAT Sessions**] をクリックして、すべての RNAT セッションまたは選択的な RNAT セッションを削除します（たとえば、特定の RNAT IP を持つ RNAT セッションの削除、または特定のネットワークまたは ACL ベースの RNAT ルールに属する RNAT セッションの削除）。

構成例:

```
    Clear all RNAT sessions existing on a Citrix ADC appliance

    > flush rnatsession

    Done

    Clear all RNAT sessions belonging to network based RNAT rules that
       has 203.0.113.0/24 network as the matching condition.

    > flush rnatsession -network 203.0.113.0 -netmask 255.255.255.0

    Done

    Clear all RNAT sessions with RNAT IP 192.0.2.90.

    > flush rnatsession -natIP 192.0.2.90

    Done

    Clear all RNAT sessions belonging to ACL based RNAT rules that has
       ACL-RNAT-1 as the matching condition.

    > flush rnatsession -aclname ACL-RNAT-1

    Done
<!--NeedCopy-->
```


プレフィクスベースの **IPv6-IPv4** 変換の設定

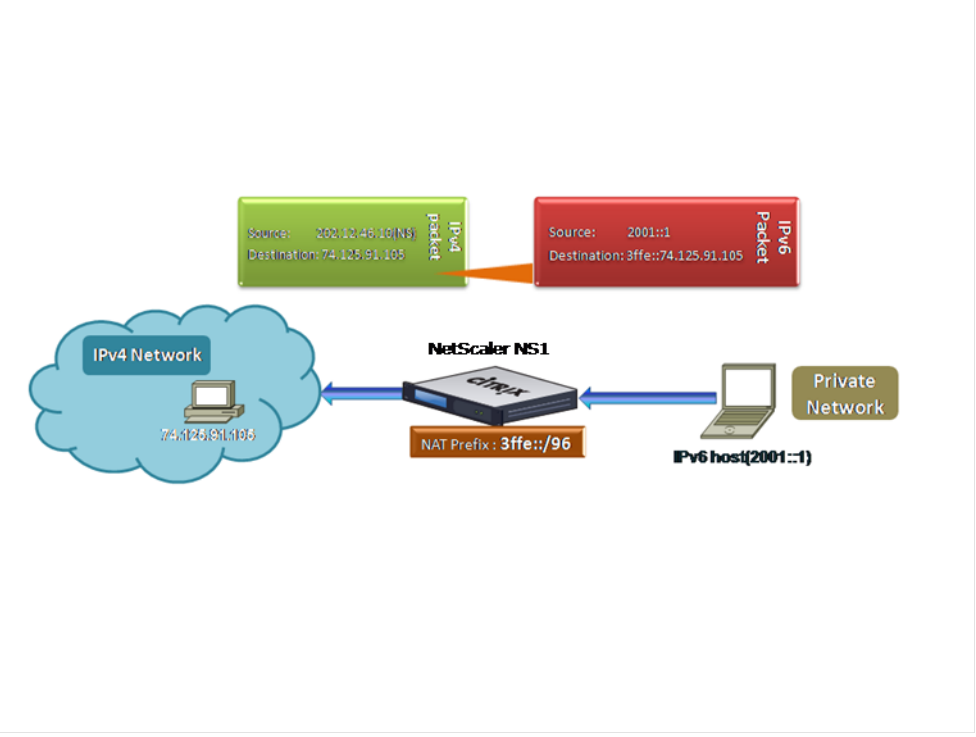
October 7, 2021

プレフィックスベースの変換は、Citrix ADC アプライアンスで構成された IPv6 プレフィックスを使用して、プライベート IPv6 サーバーから送信されたパケットを IPv4 パケットに変換するプロセスです。このプレフィクスの長さは 96 ビット（128 ～32 = 96）です。IPv6 サーバは、IPv6 パケットの宛先 IP アドレスフィールドの最後の 32 ビットに IPv4 サーバまたはホストの宛先 IP アドレスを埋め込みます。宛先 IP アドレスフィールドの最初の 96 ビットは、IPv6 NAT プレフィクスとして設定されます。

Citrix ADC アプライアンスは、すべての受信 IPv6 パケットの宛先 IP アドレスの最初の 96 ビットを、構成済みのプレフィックスと比較します。一致するものがある場合、Citrix ADC アプライアンスは IPv4 パケットを生成し、一致した IPv6 パケットの宛先 IP アドレスの最後の 32 ビットとして宛先 IP アドレスを設定します。このプレフィックス宛ての IPv6 パケットは、Citrix ADC によって IPv6-IPv4 変換が行われるように、Citrix ADC にルーティングする必要があります。

次の図では、3ffe:: /96 が Citrix ADC NS1 上の IPv6 の NAT プレフィックスとして構成されています。IPv6 ホストは、宛先 IP アドレス 3ffe:: 74.125.91.105 を持つ IPv6 パケットを送信します。NS1 は、すべての着信 IPv6 パケットの宛先 IP アドレスの最初の 96 ビットを、設定されたプレフィクスと比較し、一致させます。次に、NS1 は IPv4 パケットを生成し、宛先 IP アドレスを 74.125.91.105 に設定します。

図 1: IPv6-IPv4 プレフィクスベースの変換



CLI を使用してプレフィクスベースの IPv6-IPv4 変換を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
set ipv6 [-natprefix <ipv6_addr                    *>]
```

-
- show ipv6

例:

```
1 > set ipv6 -natprefix 3ffe::/96
2  Done
3 <!--NeedCopy-->
```

GUI を使用してプレフィクスベースの IPv6-IPv4 変換を設定するには、次の手順を実行します。

システムに移動 > [ネットワーク] の [設定] グループで、[INAT パラメーターの構成] をクリックし、[プレフィックス] パラメーターを設定します。

IP プレフィクス NAT

October 7, 2021

Citrix ADC アプライアンスは、アプライアンスで受信したパケットの全アドレスではなく、送信元 IP アドレスの一部を変換します。IP プレフィクス NAT には、送信元 IP アドレスの 1 つ以上のオクテットまたはビットの変更が含まれます。

Citrix ADC アプライアンスは、ANY 型、UDP 型、DNS 型、TCP 型、HTTP 型の負荷分散構成用に IP プレフィックス NAT をサポートしています。

ユースケース: Citrix ADC アプライアンスと最適化デバイスを展開するためのクライアントのゾーニフィケーション

IP プレフィックスの NAT は、Citrix ADC アプライアンスと最適化デバイス (Citrix ByteMobile など) を含む展開で非常に便利です。このタイプの配置には、地理的に配置された異なるクライアントネットワークがあり、同じネットワークアドレスを共有します。Citrix ADC アプライアンスは、宛先に転送する前に、各クライアントネットワークから受信したトラフィックを最適化デバイスに送信する必要があります。

デバイスは、最適化されたトラフィックを Citrix ADC アプライアンスに送り返します。最適化要件はクライアントネットワークごとに異なるため、最適化デバイスは、受信する各パケットのクライアントネットワークを認識する必要があります。解決方法は、VLAN を使用して、各クライアントネットワークからのトラフィックを異なるゾーンに分

離することです。ゾーンごとに異なる設定の IP プレフィクス NAT が設定されます。Citrix ADC アプライアンスは、各パケットの送信元 IP アドレスの最後のオクテットを変換し、変換されたオクテット値はゾーンごとに異なります。

ネットワークアドレス 192.0.2.0/24 を共有する、Z1 と Z2 の 2 つのゾーンの例を考えてみましょう。Citrix ADC アプライアンスでは、これらの 2 つのゾーンに対して natrule-1 と natrule-2 という名前の IP プレフィックス NAT エンティティが構成されます。アプライアンスが Z1 からパケットを転送する前に、natrule-1 はパケットの送信元 IP アドレスの最後のオクテットを 100 に変換します。同様に、Z2 からのパケットの場合、natrule-2 は送信元 IP アドレスの最後のオクテットを 200 に変換します。ゾーン Z1 の CL1-Z1 とゾーン Z2 の CL1-Z2 の 2 つのクライアント（それぞれ IP アドレス 192.0.2.30 を持つ）の場合、Citrix ADC アプライアンスは CL1-Z1 のパケットの送信元 IP アドレスを 100.2.30 に変換し、CL1-Z2 のパケットの送信元 IP アドレスを 200.0.2.30 に変換します。Citrix ADC アプライアンスが変換されたパケットを送信する最適化デバイスは、パケットの送信元 IP アドレスを使用してゾーンを認識するように構成されているため、パケットの送信元ゾーンに対して構成された適切な最適化が適用されます。

構成の手順

IP プレフィクス NAT を設定する手順は、次のとおりです。

- ネットプロファイルを作成し、ネットプロファイルの **NAT** ルールパラメータを設定します。NAT 規則では、2 つの IP アドレスとネットマスクを指定します。最初の IP アドレス（IP Address パラメータで指定）は、2 番目の IP アドレス（IP Rewrite パラメータで指定）で変換される送信元 IP アドレスです。ネットマスクは、送信元 IP アドレスのうち、2 番目の IP アドレスの同じ部分で変換される部分を指定します。
- ネットプロファイルを仮想サーバーまたはサービスの負荷分散にバインドします。NAT ルール設定を持つネットプロファイルは、ANY、UDP、DNS、TCP、および HTTP タイプの仮想サーバーまたはサービスにバインドできます。ネットプロファイルを仮想サーバーまたはサービスにバインドすると、Citrix ADC アプライアンスは、仮想サーバーまたはサービスに関連する着信パケットの送信元 IP アドレスと NAT ルールの設定を照合します。その後、Citrix ADC は、NAT ルールに一致するパケットに対して IP プレフィックス NAT を実行します。

コマンドラインを使用して IP プレフィクス NAT 変換を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **bind netProfile** <name> (-**natRule** <ip_addr> <netmask> <rewriteIp>)
- **show netprofile** <name>

GUI を使用して IP プレフィクス NAT を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. NetProfiles の追加または変更時に、[NAT ルール] で次のパラメータを設定します。
    - IP アドレス
    - ネットマスク
    - IP を書き換え

構成例

次の設定例では、ネットプロファイル parti-NAT-1 には IP プレフィクス NAT 設定があり、タイプ ANY のロードバランシング仮想サーバー LBVS-1 にバインドされています。192.0.0.0/8 から LBVS-1 で受信したパケットの場合、Citrix ADC アプライアンスはパケットの送信元 IP アドレスの最後のオクテットを 100 に変換します。たとえば、送信元 IP アドレスが 192.0.2.30 のパケットが LBVS-1 で受信された場合、Citrix ADC アプライアンスは、送信元 IP アドレスを 100.0.2.30 に変換してから、バインドされたサーバーの 1 つを送信します。

```
> add netprofile PARTIAL-NAT-1
Done

> bind netprofile PARTIAL-NAT-1  - natrule 192.0.0.0 255.0.0.0 100.0.0.0
Done

> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1
Done
<!--NeedCopy-->
```

スタティック **ARP**

October 7, 2021

スタティック ARP エントリを ARP テーブルに追加したり、ARP テーブルからスタティック ARP エントリを削除したりできます。エントリを追加したら、設定を確認する必要があります。スタティック ARP エントリの作成後に IP アドレス、ポート、または MAC アドレスが変更された場合は、スタティックエントリを削除するか、手動で調整する必要があります。したがって、必要な場合を除き、スタティック ARP エントリを作成することはお勧めしません。

CLI を使用してスタティック ARP エントリを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add arp -IPAddress** <ip_addr> -**mac**<mac_addr> -**ifnum** <interface_name>
- **show arp** <IPAddress>

例**:**

```
> add arp -ip 10.102.29.6 -mac 00:24:e8:73:ca:ec -ifnum 1/1
 Done
<!--NeedCopy-->
```


CLI を使用してスタティック ARP エントリを削除するには、次の手順を実行します。

コマンドプロンプトで、**rm arp** コマンドと IP アドレスを入力します。

GUI を使用してスタティック ARP エントリを追加するには、次の手順を実行します。

[システム] > [ネットワーク] > [**ARP** テーブル] に移動し、スタティック ARP エントリを追加します。

スタティック **ARP** エントリでの **VLAN** の指定

スタティック ARP エントリでは、宛先デバイスにアクセスできる VLAN を指定できます。この機能は、スタティック ARP エントリで指定されたインターフェイスが複数のタグ付き VLAN の一部であり、宛先が VLAN の 1 つを介してアクセスできる場合に便利です。Citrix ADC アプライアンスは、スタティック ARP エントリと一致する発信パケットに指定された VLAN ID を含めます。ARP エントリで VLAN ID を指定せず、指定したインターフェイスが複数のタグ付き VLAN の一部である場合、アプライアンスはインターフェイスのネイティブ VLAN を ARP エントリに割り当てます。

たとえば、Citrix ADC インターフェイス 1/2 がネイティブ VLAN 2 の一部であり、タグ付き VLAN 3 および 4 の一部であり、VLAN 3 の一部であり、インターフェイス 1/2 を介してアクセス可能なネットワークデバイス A のスタティック ARP エントリを追加します。ネットワークデバイス A の ARP エントリに VLAN 3 を指定する必要があります。Citrix ADC アプライアンスは、ネットワークデバイス A 宛てのすべてのパケットにタグ付き VLAN 3 を含み、インターフェース 1/2 から送信します。

VLAN ID を指定しない場合、Citrix ADC アプライアンスは ARP エントリにネイティブ VLAN 2 を割り当てます。デバイス A 宛てのパケットは、デバイス A の VLAN であるタグ付き VLAN 3 を指定しないため、ネットワークパスでドロップされます。

CLI を使用してスタティック ARP エントリに VLAN を指定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add arp -IPAddress** <ip_addr> **-mac**<mac_addr> **-ifnum** <interface_name> [-**vlan** <positive_integer>]
- **show arp** <IPAddress>

例**:**

```
1 > add arp -ip 198.51.100.91 -mac 36:db:4b:f6:12:15 -ifnum 1/2 -vlan 3
2 Done
3 <!--NeedCopy-->
```

ダイナミック **ARP** エントリのタイムアウトの設定

October 7, 2021

ダイナミックに学習された ARP エントリのエージングタイム（タイムアウト値）をグローバルに設定できます。新しい値は、新しい値を設定した後にダイナミックに学習される ARP エントリだけに適用されます。以前に存在する ARP エントリは、以前に設定されたエージングタイムの後に期限切れになります。ARP タイムアウト値は、1～1200 秒で指定できます。

CLI を使用してダイナミック ARP エントリのタイムアウトを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set arpparam** -**timeout** <positive_integer>]
- **show arpparam**

例**:**

```
1 > set arpparam -timeout 500
2  Done
3 <!--NeedCopy-->
```

CLI を使用してダイナミック ARP エントリのタイムアウトをデフォルト値に設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **unset arpparam**
- **show arpparam**

例**:**

```
1 > unset arpparam
2  Done
3 <!--NeedCopy-->
```

GUI を使用してダイナミック ARP エントリのタイムアウトを設定するには、次の手順を実行します。

[システム] > [ネットワーク] に移動し、[設定] グループで [**ARP** グローバルパラメータの構成] をクリックし、[**ARP** テーブルエントリタイムアウト] パラメータを設定します。

ネイバー検出

October 7, 2021

近隣検出（ND）は、IPv6 の最も重要なプロトコルの 1 つです。これは、アドレス解決プロトコル (ARP)、インターネット制御メッセージプロトコル (ICMP)、およびルーター探索の機能を組み合わせたメッセージベースのプロトコ

ルです。ND を使用すると、ノードはリンク層アドレスをアドバタイズし、近隣ノードの MAC アドレスまたはリンク層アドレスを取得できます。このプロセスは、近隣探索プロトコル (ND6) によって実行されます。

近隣探索では、次の機能を実行できます。

- ルーター検出: ホストが接続されているリンク上のローカルルーターを検出し、デフォルトルーターを自動的に構成できるようにします。
- **[**プレフィックス検出]: ホストがローカル宛先のネットワークプレフィックスを検出できるようにします。

  注：Citrix ADC アプライアンスはプレフィックス検出をサポートしていません。
- パラメータ検出：ホストが、MTU や発信トラフィックのデフォルトのホップ制限など、追加の動作パラメータを検出できるようにします。
- アドレス自動構成: DHCPv6 などのステートフルアドレス構成サービスの有無にかかわらず、ホストが IP アドレスを自動的に構成できるようにします。Citrix ADC は、グローバル IPv6 アドレスのアドレス自動構成をサポートしていません。
- アドレス解決: IPv4 の ARP に相当し、ノードが隣接ノードの IPv6 アドレスをリンク層アドレスに解決できるようにします。
- ネイバー到達不能検出：ノードがネイバーの到達可能性状態を判別できるようにします。
- 重複アドレス検出：NSIP アドレスがネイバーノードによって既に使用されているかどうかをノードで判別できるようにします。
- リダイレクト：IPv4 ICMP リダイレクトメッセージと同等で、ルータはホストをより良いファーストホップ IPv6 アドレスにリダイレクトして、宛先に到達できるようにします。

注： Citrix ADC アプライアンスは、IPv6 リダイレクトをサポートしていません。

構成の手順

ネイバー検出の設定は、次の作業で構成されます。

- IPv6 ネイバーの追加
- （任意）IPv6 ネイバーの削除

CLI のプロシージャ

CLI を使用して IPv6 ネイバーを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add nd6** <neighbor> <mac> <ifnum> [-**vlan** <integer>]
- **sh nd6**

例**:**


```
> add nd6 2001::1 00:04:23:be:3c:06 1/1  - vlan 1
 Done

> show nd6
   Neighbor                 MAC-Address(Vlan, Interface)       State
              TIME
   --------                 ----------------------------       -----
            --------
1) ::1                      00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
        PERMANENT
2) fe80::2d0:68ff:fe0b:58da 00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
        PERMANENT
3) 2001::1                  00:04:23:be:3c:06(   1,   1/1)   REACHABLE
          STATIC
 Done
<!--NeedCopy-->
```

CLI を使用してネイバー探索エントリを削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **rm nd6** <Neighbor> -**vlan** <VLANID>

例**:**

```
 rm nd6 3ffe:100:100::1 -vlan 1
<!--NeedCopy-->
```

CLI を使用してすべてのネイバー探索エントリを削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **clear nd6**

GUI のプロシージャ

GUI を使用して IPv6 ネイバーを追加するには、次の手順を実行します。

[システム] > [ネットワーク] > [**IPv6** ネイバー] に移動し、新しい IPv6 ネイバーを追加します。

GUI を使用してネイバー探索エントリを削除するには、次の手順を実行します。

[システム] > [ネットワーク] > [**IPv6** ネイバー] に移動し、IPv6 ネイバーを削除します。

GUI を使用してすべてのネイバー探索エントリを削除するには、次の手順を実行します。

[システム] > [ネットワーク] > [**IPv6** ネイバー] に移動し、[クリア] をクリックします。

IP トンネル

October 7, 2021

IP トンネルは、ルーティングパスを持たない 2 つのネットワーク間で、カプセル化技術を使用して作成できる通信チャネルです。2 つのネットワーク間で共有されるすべての IP パケットは、別のパケット内にカプセル化され、トンネル経由で送信されます。

Citrix ADC アプライアンスは、次の方法で IP トンネリングを実装します。

- カプセル化機能としての **Citrix ADC**（**DSR** モードによる負荷分散）：異なる国にまたがる複数のデータセンターを持つ組織を考えてみましょう。Citrix ADC は 1 つの場所にあり、バックエンドサーバーは別の国にあります。本質的に、Citrix ADC とバックエンドサーバーは異なるネットワーク上にあり、ルーター経由で接続されています。

  この Citrix ADC でダイレクトサーバーリターン（DSR）を設定すると、送信元サブネットから送信されたパケットは Citrix ADC によってカプセル化され、ルーターとトンネルを経由して適切なバックエンドサーバーに送信されます。バックエンドサーバーはパケットをカプセル化解除し、クライアントに直接応答します。パケットは Citrix ADC 経由で渡されません。

- カプセル化解除機能としての **Citrix ADC**：複数のデータセンターがそれぞれ Citrix ADC とバックエンドサーバーを持つ組織を考えてみましょう。パケットがデータセンター A からデータセンター B に送信されると、通常、ルータや別の Citrix ADC を経由して送信されます。Citrix ADC はパケットを処理し、パケットをバックエンドサーバーに転送します。ただし、カプセル化されたパケットが送信される場合、Citrix ADC はパケットをバックエンドサーバーに送信する前にカプセル化を解除できる必要があります。Citrix ADC をカプセル解除機能として機能させるには、ルーターと Citrix ADC の間にトンネルを追加します。追加のヘッダー情報を含むカプセル化されたパケットが Citrix ADC に到達すると、データパケットのカプセル化が解除されます。つまり、追加のヘッダー情報が削除され、パケットは適切なバックエンドサーバーに転送されます。

  Citrix ADC は、負荷分散機能のカプセル解除機能として使用することもできます。特に、vserver 上の接続数がしきい値を超え、すべての新しい接続がバックアップ仮想サーバーに迂回される場合に使用します。

IP トンネルの設定

Citrix ADC アプライアンスで IP トンネルを構成するには、IP トンネルエンティティを作成します。IP トンネルエンティティは、ローカルおよびリモートトンネルエンドポイント IP アドレスと、IP トンネルに使用するプロトコルを指定します。

注：クラスタ設定で IP トンネルを設定する場合、ローカル IP アドレスはストライプ SNIP アドレスである必要があります。

CLI のプロシージャ

CLI を使用して IP トンネルを作成するには、次の手順を実行します。

コマンドプロンプトで次のように入力します。

- **add iptunnel** <name> <remote> <remoteSubnetMask> <local> -**type** -**protocol** (**ipoverip** | **GRE**)
- **show iptunnel**

CLI を使用して IP トンネルを削除するには、次の手順を実行します。

IP トンネルを削除するには、**rm iptunnel** コマンドとトンネルの名前を入力します。

CLI を使用して IPv6 トンネルを作成するには、次の手順を実行します。

コマンドプロンプトで次のように入力します。

- **add ip6tunnel** <name> <remoteIp> <local>
- **show ip6tunnel**

CLI を使用して IPv6 トンネルを削除するには、次の手順を実行します。

IPv6 トンネルを削除するには、**rm ip6tunnel** コマンドとトンネルの名前を入力します。

GUI のプロシージャ

GUI を使用して IP トンネルを作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [**IP** トンネル] に移動し、新しい IP トンネルを追加します。

GUI を使用して IPv6 トンネルを作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [**IP** トンネル] > [**IPv6** トンネル] に移動し、新しい IPv6 トンネルを追加します。

IP トンネルのグローバルなカスタマイズ

送信元 IP アドレスをグローバルに指定することで、すべてのトンネルに共通の送信元 IP アドレスを割り当てることができます。また、フラグメンテーションは CPU を大量に消費するため、Citrix ADC アプライアンスがフラグメンテーションを必要とするパケットをドロップするようにグローバルに指定できます。または、CPU しきい値に達していない限り、すべてのパケットをフラグメント化する場合は、CPU しきい値をグローバルに指定できます。

CLI のプロシージャ

CLI を使用して IP トンネルをグローバルにカスタマイズするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set ipTunnelParam** -**srcIP** <sourceIPAddress> -**srcIPRoundRobin** (**YES** | **NO**)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ipTunnelParam**

例**:**


```
> set iptunnelparam  -srcIP 12.12.12.22 -dropFrag Yes  -
    dropFragCpuThreshold 50
 Done

> set iptunnelparam -srcIPRoundRobin YES -dropFrag Yes  -
    dropFragCpuThreshold 50
 Done
<!--NeedCopy-->
```

CLI を使用して IPv6 トンネルをグローバルにカスタマイズするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set ip6tunnelparam** -**srcIP** <IPv6Address> -**srcIPRoundRobin** (**YES** | NO)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ip6tunnelparam**

GUI のプロシージャ

GUI を使用して IP トンネルをグローバルにカスタマイズするには、次の手順を実行します。

[システム] > [ネットワーク] に移動し、[設定] グループで [**IPv4** トンネルのグローバル設定] をクリックします。

1. [システム] > [ネットワーク] に移動し、[設定] グループで [**IPv6** トンネルのグローバル設定] をクリックします。
2. [**IP** トンネルグローバルパラメータの設定] ダイアログボックスで、パラメータを設定します。

GUI を使用して IPv6 トンネルをグローバルにカスタマイズするには、次の手順を実行します。

1. [システム] > [ネットワーク] に移動し、[設定] グループで [**IPv6** トンネルのグローバル設定] をクリックします。
2. [**IP** トンネルグローバルパラメータの設定] ダイアログボックスで、パラメータを設定します。

GRE IP トンネルの **GRE** ペイロードオプション

設定された GRE IP トンネルの場合、Citrix ADC アプライアンスはイーサネットヘッダーと VLAN ヘッダー（dot1q VLAN タグ）を含むレイヤー 2 パケット全体をカプセル化します。Citrix ADC アプライアンスと一部のサードパーティ製デバイス間の IP GRE トンネルが安定しない場合があります。これらのサードパーティ製デバイスは、一部またはレイヤー 2 のパケットヘッダーを処理するようにプログラムされていないためです。Citrix ADC アプライアンスとサードパーティ製デバイス間の安定した IP GRE トンネルを構成するには、GRE IP トンネルコマンドセットの GRE ペイロードパラメータを使用します。GRE ペイロード設定は、IPSec トンネルを持つ GRE にも適用できます。

GRE ペイロードパラメータを設定して、GRE トンネルを経由してパケットが送信される前に次のいずれかを実行できます。

- **DOT1Q** を使用したイーサネット。イーサネットヘッダーと VLAN ヘッダーを伝送します。これがデフォルトの設定です。ネットブリッジにバインドされたトンネルの場合、内部イーサネットヘッダーと VLAN ヘッダーには、Citrix ADC アプライアンスの ARP およびブリッジテーブルからの情報が含まれます。PBR ルールのネクストホップとして設定されたトンネルの場合、内部イーサネット宛先 MAC アドレスはゼロに設定され、VLAN ヘッダーはデフォルト VLAN を指定します。Citrix ADC トンネルのエンドポイントから送信されるカプセル化（GRE）パケットの形式は次のとおりです。

Outer Ethernet Header	Outer IP Header	GRE Header	Inner Ethernet	Inner VLAN header	Inner IP/IPv6/ARP header	Inner TCP/UDP Header	Payload

- イーサネット。イーサネットヘッダーを伝送しますが、VLAN ヘッダーをドロップします。パケットはトンネル内の VLAN 情報を伝送しないため、この設定でネットブリッジにバインドされたトンネルでは、適切な VLAN をネットブリッジにバインドする必要があります。これにより、トンネル上のパケットを受信すると、Citrix ADC はこれらのパケットを指定された VLAN に転送できます。トンネルが PBR ルールでネクストホップとして設定されている場合、Citrix ADC はトンネルで受信したパケットをルーティングします。Citrix ADC トンネルのエンドポイントから送信されるカプセル化（GRE）パケットの形式は次のとおりです。

Outer Ethernet header	Outer IP header	GRE Header	Inner Ethernet header	Inner IP/IPv6/ARP header	Inner TCP/UDP header	Payload

- 知的財産権です。イーサネットヘッダーと VLAN ヘッダーをドロップします。この設定を持つトンネルはレイヤ 2 ヘッダーを伝送しないため、これらのトンネルはネットブリッジにバインドできませんが、PBR ルールでネクストホップとして設定できます。パケットを受信するピアトンネルエンドポイントデバイスは、パケットを消費またはルーティングします。Citrix ADC トンネルのエンドポイントから送信されるカプセル化（GRE）パケットの形式は次のとおりです。

Outer Ethernet header	Outer IP header	GRE header	Inner IP/IPv6 header	Inner TCP/UDP header	Payload

CLI を使用して GRE IP トンネル内のパケットのレイヤ 2 ヘッダーをドロップするには、次の手順を実行します。

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> [-**protocol** <GRE> [-**vlan** <positive_integer>]] [-**grepayload** <grepayload>] [-**ipsecProfileName** <string>]
- **show iptunnel** <tunnelname>

例:

```
1 > add iptunnel IPTUNNEL-1 203.0.113.133 255.255.255.0 198.51.100.15 -
    protocol GRE - grepayload Ethernet -ipsecProfileName IPTUNNEL-IPSEC
    -1
2 Done
```



```
<!--NeedCopy-->
```

GRE IPV4 トンネルを介した IPv6 トラフィック

Citrix ADC アプライアンスは、IPV4 GRE トンネルを介した IPv6 トラフィックの転送をサポートします。この機能は、分離された IPv6 ネットワーク間の IPv4 インフラストラクチャをアップグレードすることなく、それらのネットワーク間の通信を有効にするために使用できます。

この機能を設定するには、Citrix ADC が IPv6 トラフィックの送受信に使用する構成済みの IPv4 GRE トンネルに PBR6 ルールを関連付けます。PBR6 規則の送信元 IPv6 アドレスおよび宛先 IPv6 アドレスパラメータは、トラフィックが IPv4 GRE トンネルを通過する IPv6 ネットワークを指定します。

注: IPSec プロトコルは、IPv6 パケットを転送するように設定された GRE IPv4 トンネルではサポートされません。

CLI を使用して GRE IPv4 トンネルを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol GRE**
- **show ipTunnel** <name>

CLI を使用して PBR6 ルールを GRE IPv4 トンネルに関連付けるには、次の手順を実行します。

- **add ns pbr6** <pbrName> **ALLOW** -**srcIPv6** <network-range> -**dstIPv6** <network-range> -**ipTunnel** <tunnelName>
- **show pbr**

構成例

次の設定例では、リモートトンネルエンドポイント IP アドレス 10.10.6.30 とローカルトンネルエンドポイント IP アドレス 10.10.5.30 を使用して、GRE IP トンネル TUNNEL-V6onV4 が作成されます。その後、トンネルは pbr6 PBR6-V6onV4 にバインドされます。srcIPv6 はローカルエンドポイントに接続されている IPv6 ネットワークを指定し、destIPv6 はリモートエンドポイントに接続されている IPv6 ネットワークを指定します。これらの Ipv6 ネットワークからのトラフィックは、GRE IPv4 トンネルを通過できます。

```
> add ipTunnel TUNNEL-V6onV4 10.10.6.30 255.255.255.255 10.10.5.30 -
   protocol GRE
-ipsecProfileName None
Done
> add ns pbr6 PBR6-V6onV4 ALLOW -srcIPv6 = 2001:0db8:1::1-2001:0db8
   :1::255 -destIPv6 =
1-2001:0db8:4::255 -ipTunnel TUNNEL-V6onV4
<!--NeedCopy-->
```


IP-IP トンネルを介した応答トラフィックの送信

Citrix ADC アプライアンスは、応答トラフィックを送信元にルーティングするのではなく、IP-IP トンネルを介して送信するように構成できます。デフォルトでは、アプライアンスが別の Citrix ADC またはサードパーティ製デバイスから IP-IP トンネルを介して要求を受信すると、その応答トラフィックをトンネル経由で送信するのではなく、ルーティングします。ポリシーベースルート（PBR）を使用するか、MAC ベースフォワーディング（MBF）を有効にして、トンネル経由で応答を送信できます。

PBR ルールでは、トラフィックがトンネルを通過する両方のエンドポイントのサブネットを指定します。また、ネクストホップをトンネル名として設定します。応答トラフィックが PBR ルールと一致すると、Citrix ADC アプライアンスはトンネルを介してトラフィックを送信します。

または、MBF を有効にしてこの要件を満たすこともできますが、この機能は、Citrix ADC アプライアンスがセッション情報を格納するトラフィック（負荷分散や RNAT 構成に関連するトラフィックなど）に制限されます。アプライアンスは、セッション情報を使用して、トンネルを介して応答トラフィックを送信します。

CLI のプロシージャ

CLI を使用して PBR ルールを作成し、PBR ルールに IP-IP トンネルを関連付けるには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns pbr** <pbr_name> **ALLOW** -**srcIP** = <local_subnet_range> -**destIP** = <remote_subnet_range> -**ipTunnel** <tunnel_name>
- **apply ns pbrs**
- **show ns pbr** <pbr_name>

CLI を使用して MAC ベースの転送を有効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **enable ns mode MBF**
- **show ns mode**

GUI のプロシージャ

GUI を使用して PBR ルールを作成し、PBR ルールに IP-IP トンネルを関連付けるには、次の手順を実行します。

1. **System** > **Network** > **PBRs** に移動します。[**PBR**] タブで、**PBR** ルールを作成します。
2. PBR の作成中に、ネクストホップタイプを **IP** トンネルに、**IP** トンネル名を設定した IP-IP トンネル名に設定します。

GUI を使用して MAC ベースの転送を有効にするには、次の手順を実行します。

1. [システム] **>** [設定] に移動し、[モードと機能] で [モードの構成] をクリックします。
2. [モードの設定] ページで、[**MAC** ベースの転送] を選択します。

設定例

たとえば、2 つの Citrix ADC アプライアンス NS1 と NS2 の間にセットアップされる IPIP トンネルである NS1-NS2-IP の例を考えてみましょう。

デフォルトでは、NS2 がトンネルを介して受信する要求に対して、NS2 はトンネルを介して（NS1 に）送信するのではなく、応答トラフィックを送信元にルーティングします。

NS2 でポリシーベースルート（PBR）を設定するか、または MAC-Based Forwarding（MBF; MAC ベースフォワーディング）を有効にして、トンネル経由で応答を送信できます。

次の NS2 での構成例では、NS1-NS2-IPIP は IPIP トンネルで、NS1-NS2-IP-PBR は PBR ルールです。トンネルを介して NS2 が受信した要求（内部送信元 IP アドレスが 10.102.147.0 ～10.102.147.255 で、内部宛先 IP アドレスが 10.102.147.0 ～10.102.147.255）の場合、NS2 は送信元にルーティングするのではなく、トンネルを介して（NS1 に）対応する応答を送信します。この機能は、PBR ルールに一致するトラフィックに限定されます。

```
> add iptunnel NS1-NS2-IPIP 192.0.2.99 255.255.255.255 203.0.113.99 -
    protocol IPIP

Done
> add pbr NS1-NS2-IPIP-PBR -srcIP 10.102.147.0-10.102.147.255  - destIP
    10.20.1.0-10.20.1.255  - ipTunnel NS1-NS2-IPIP

Done
> apply pbrs

Done
```

または、NS2 で MBF を有効にすることもできます。この機能は、NS2 がセッション情報を格納するトラフィック（たとえば、ロード・バランシングや RNAT 構成に関連するトラフィック）に限定されます。

```
> enable ns mode MBF

Done
```

クラス **E IPv4** パケット

October 7, 2021

デフォルトでは、Citrix ADC アプライアンスは、送信元 IP または宛先 IP フィールドにクラス E IPv4 アドレスが

含まれているパケットをドロップします。セットアップでクラス E IPv4 アドレスを使用している場合は、クラス E IPv4 パケットを処理するように Citrix ADC アプライアンスを構成できます。

はじめに

クラス E IPv4 パケットを処理するように Citrix ADC アプライアンスを構成する前に、次の点に注意してください。

- Citrix ADC アプライアンスは、クラス E の範囲内の Citrix ADC が所有する IPv4 アドレス（SNIP や VIP など）の構成をサポートしていません。Citrix ADC アプライアンスは、クラス E の IPv4 パケットの処理のみをサポートします。
- Citrix ADC アプライアンスは、IPv6 機能にクラス E IPv4 アドレスを内部的に使用します。Citrix ADC アプライアンスは、両方の機能（処理クラス E IPv4 パケットと IPv6 サポート）を同時に動作させることはできません。Citrix ADC アプライアンスでは、クラス E の IPv4 パケットの処理が有効になっているときに IPv6 機能を有効にしないという制限があります。逆の場合も同様です。

構成の手順

クラス E IPv4 パケットを処理するように Citrix ADC アプライアンスを構成するには、**IPv4** クラス **E** アドレスクライアント（**allowClassEIPv4** ）レイヤー 3 パラメータを有効にします。

CLI のプロシージャ

CLI を使用してクラス E IPv4 パケットを処理するように Citrix ADC アプライアンスを構成するには：

コマンドプロンプトで入力します。

- **set l3param** -**allowClassEIPv4** (**ENABLED**|**DISABLED**)
- **show l3param**

設定例**:**

```
1  > set l3param -allowClassEIPv4 ENABLED
2
3  Done
4
5  > sh l3param
6
7      Network L3 related Configuration Parameters
8
9      icmpgen_rate_threshold      : 100
10
11     srcnat                      : ENABLED
12
```



```
13     override_rnat                 : DISABLED
14
15     drop_df_flag                  : DISABLED
16
17     .
18
19     .
20
21     .
22
23     IPv6DynamicRouting            : DISABLED
24
25     allowClassEIPv4               : ENABLED
26
27 Done
28 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してクラス E の IPv4 パケットを処理するように Citrix ADC アプライアンスを構成するには:

1. [システム] > [ネットワーク] に移動し、**[**設定] セクションで [レイヤ **3** パラメータの設定] をクリックします。
2. [**IPv4** クラス **E** アドレスクライアント] を選択し、[**OK**] をクリックします。

インターフェイス

October 7, 2021

インターフェイスの設定を開始する前に、設定で MAC ベースの転送モードを使用できるかどうかを決定し、それに応じてこのシステム設定を有効または無効にします。構成内のインターフェイスの数は、Citrix ADC アプライアンスのモデルによって異なります。個々のインターフェイスを設定する以外に、インターフェイスを論理的にグループ化し、VLAN を使用して一連のインターフェイス内のデータフローを制限したり、リンクをチャネルに集約したりできます。高可用性セットアップでは、必要に応じて仮想 MAC アドレスを設定できます。L2 モードを使用する場合は、ブリッジテーブルのエージングを変更できます。

設定が完了したら、パス MTU ディスカバリのシステム設定を有効にするかどうかを決定します。Citrix ADC アプライアンスは、VRRP を使用してアクティブ-アクティブモードで展開できます。アクティブ-アクティブ展開では、ダウンタイムを防止するだけでなく、展開内のすべての Citrix ADC アプライアンスを効率的に使用できます。ネットワークビジュアライザーツールを使用して、Citrix ADC 展開のネットワーク構成を表示し、インターフェイス、チャネル、VLAN、およびブリッジグループを構成できます。

MAC ベースの転送の設定

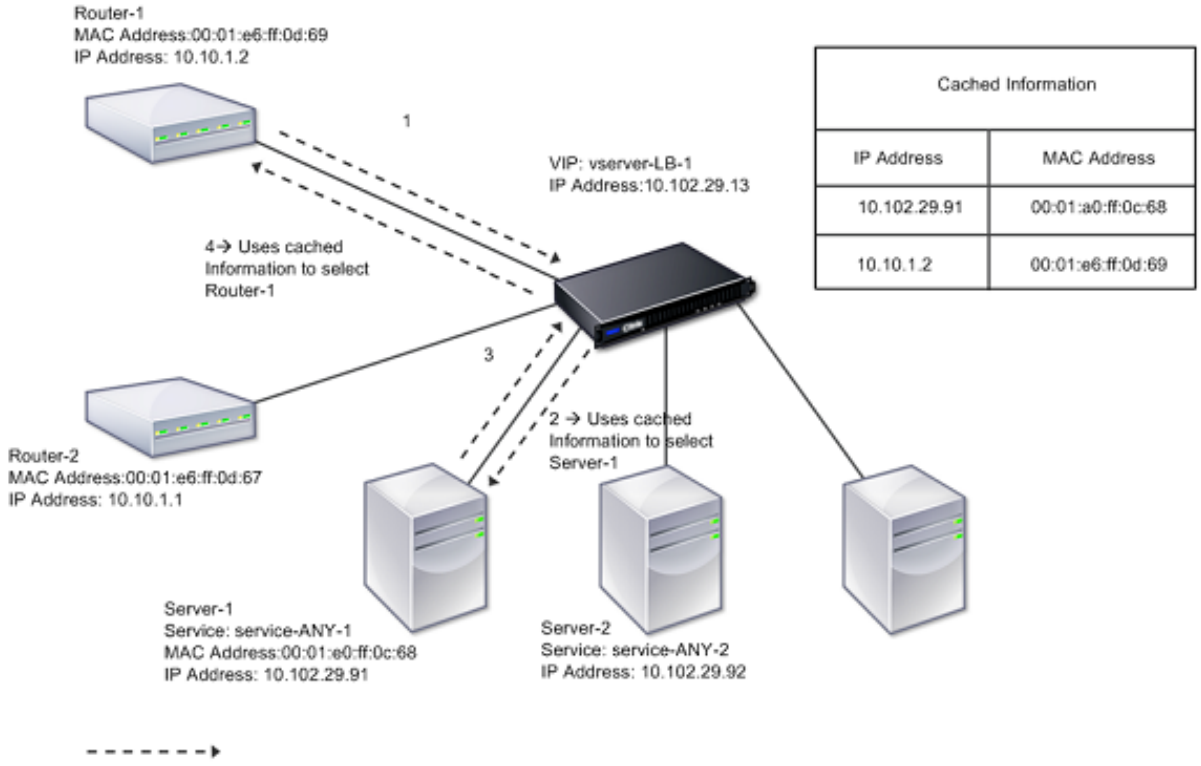
October 7, 2021

MAC ベース転送（MBF）を有効にすると、要求が Citrix ADC アプライアンスに届くと、アプライアンスはフレームの送信元 MAC アドレスを記憶し、結果の応答の宛先 MAC アドレスとして使用します。MAC ベースの転送を使用すると、複数ルートまたは ARP ルックアップを回避し、非対称のパケットフローを回避できます。Citrix ADC が VPN やファイアウォールなどの複数のステートフルデバイスに接続されている場合は、MAC ベースの転送が必要になることがあります。これは、リターントラフィックが初期トラフィックの送信元と同じデバイスに送信されるためです。

MAC ベースの転送は、VPN デバイスを使用する場合に便利です。これは、VPN を通過するすべてのトラフィックが同じ VPN デバイスを通過することを保証するためです。

次のトポロジ図は、MAC ベースの転送のプロセスを示しています。

図 1: MAC ベース転送モード



MAC ベース転送（MBF）を有効にすると、Citrix ADC は次の MAC アドレスをキャッシュします。

- 受信接続のソース（ルーター、ファイアウォール、VPN デバイスなどの通信デバイス）
- 要求に応答するサーバー

サーバーが Citrix ADC アプライアンスを介して応答する場合、アプライアンスは応答パケットの宛先 MAC アドレスをキャッシュされたアドレスに設定し、トラフィックが対称的に流れるようにして、クライアントに応答を転送します。このプロセスでは、ルートテーブルのルックアップ機能と ARP ルックアップ機能が回避されます。ただし、

Citrix ADC が接続を開始すると、ルックアップ機能にルートテーブルと ARP テーブルが使用されます。ダイレクトサーバリターン設定では、MAC ベースの転送を有効にする必要があります。

ダイレクトサーバーのリターン構成の詳細については、「負荷分散」を参照してください。

一部の展開トポロジでは、着信パスと発信パスが異なるルータを経由する必要があります。MAC ベースの転送は、このトポロジ設計を破ります。

MBF は、次の状況で無効にする必要があります。

- サーバーが **LACP (802.1ad** リンクアグリゲーション**)** を使用せずにネットワークインターフェイスカード **(NIC)** チーミングを使用する場合。このような状況で MAC ベースの転送を有効にするには、Citrix ADC とサーバー間でレイヤー 3 デバイスを使用する必要があります。
  注：仮想インターフェイスは 1 つの MAC アドレスを使用するため、サーバが LACP と NIC チーミングを使用する場合、MBF を有効にできます。
- ファイアウォールのクラスタリングを使用する場合。ファイアウォールクラスタリングでは、ARP を使用して着信トラフィックの MAC アドレスを解決することを前提としています。着信 MAC アドレスは、非クラスタ化 MAC アドレスになる場合があり、着信パケット処理には使用しないでください。

MBF が無効になっている場合、アプライアンスは L2 または L3 接続を使用して、サーバーからクライアントに応答を転送します。ルートテーブルによっては、発信接続と着信接続に使用されるルータが異なる場合があります。リバーストラフィックの場合（サーバからの応答）：

- 送信元と宛先が異なる IP サブネット上にある場合、アプライアンスはルート検索を使用して宛先を特定します。
- 送信元が宛先と同じサブネット上にある場合、Citrix ADC は ARP テーブルを検索してネットワークインターフェイスを特定し、トラフィックをそのテーブルに転送します。ARP テーブルが存在しない場合、Citrix ADC は ARP エントリを要求します。

CLI を使用して MAC ベースの転送を有効または無効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **enable ns mode MBF**
- **disable ns mode MBF**

GUI を使用して MAC ベースの転送を有効または無効にするには、次の手順を実行します。

1. [システム] **>** [設定] に移動し、**[**モードと機能] グループで [モードの構成] をクリックします。
2. **MAC** ベースの転送オプションを選択または選択解除します。

ロードバランシング設定のための **MAC** ベースの転送

一部の負荷分散設定では、Citrix ADC アプライアンスがこれらの設定に対してグローバル MBF（有効になっている場合）をバイパスし、代わりにルート/ARP ルックアップを使用してパケットを宛先に送信する必要があります。

ネットプロファイルの MBF パラメータは、特定のロードバランシング設定の MBF を有効または無効にするために使用されます。MBF は、ネットプロファイル（MBF の有効化または無効化）を仮想サーバーおよびサービスにバインドすることで、クライアント側および負荷分散構成のサーバー側に設定できます。

たとえば、MBF が無効になっているネットプロファイルが負荷分散構成の仮想サーバーにバインドされている場合、Citrix ADC アプライアンスはグローバル MBF をバイパスし（有効になっている場合）、代わりにルート/ARP ルックアップを使用してクライアントに応答パケットを送信します。

はじめに

ロードバランシング設定の MBF の設定を開始する前に、次の点に注意してください。

- 負荷分散構成では、クライアント側（仮想サーバー）とサーバー側（サービス/サービスグループ）が異なる MBF 設定を持つことができます。
- 仮想サーバおよびサービスにバインドされたネットプロファイルで MBF が明示的に設定されていない場合、ロードバランシング構成はグローバル MBF 設定を継承します。
- 負荷分散構成では、ネットプロファイルがサービスにバインドされていない場合、サーバー側（サービス）は仮想サーバーにバインドされたネットプロファイルのクライアント側 MBF 設定を継承します。
- ダイレクト・サーバー・リターン・モードのロード・バランシング構成では、クライアント側はサービスにバインドされたネット・プロファイルの MBF 設定を継承します。
- コンテンツスイッチング構成では、クライアント側は、ターゲット負荷分散仮想サーバーからではなく、コンテンツスイッチング仮想サーバーにバインドされたネットプロファイルの MBF 設定を取ります。

制限事項

ロードバランシング設定の MBF の設定を開始する前に、次の制限事項に注意してください。

- 負荷分散構成の MBF 設定は、クラスタセットアップではサポートされていません。
- MAC モードまたは L2Conn 設定のロードバランシング仮想サーバでは、仮想サーバへのバインドされたネットプロファイルの MBF 設定に関係なく、MBF が有効になります。
- Citrix ADC アプライアンスは、ネットプロファイルを使用した負荷分散モニターの MBF の設定をサポートしていません。つまり、ネットプロファイルの MBF 設定は、ネットプロファイルがバインドされているモニターには適用されません。グローバル MBF 設定は、バインドされたネットプロファイルの MBF 設定に関係なく、モニタに適用されます。

ロードバランシング設定用の **MBF** の構成

ロードバランシング設定の MBF の設定は、次の作業で構成されます。

- ネットプロファイルで MBF パラメータを有効にします。
- ネット・プロファイルをロード・バランシング仮想サーバーまたはサービスにバインドします。

CLI を使用してネットプロファイルで MBF を有効にするには、次の手順を実行します。

- ネットプロファイルの追加中に MBF を有効にするには、コマンドプロンプトで次のように入力します。
  - **add netProfile** <name> -**MBF** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>
- 既存のネットプロファイルで MBF を有効にするには、コマンドプロンプトで次のように入力します。
  - **set netProfile** <name> -**MBF** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>

GUI を使用してネットプロファイルで MBF を有効にするには **

1. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
2. ネットプロファイルを追加または変更するときに、**MBF** パラメータを有効にします。

次の設定例では、ネットプロファイル NETPROFILE-MBF-LBVS は MBF を有効にし、ロードバランシング仮想サーバー LBVS-1 にバインドされています。また、ネットプロファイル NETPROFILE-MBF-SVC では、MBF が有効になっており、ロードバランシングサービス SVC-1 にバインドされています。

```
> add netprofile NETPROFILE-MBF-LBVS -MBF ENABLED

Done

> add netprofile NETPROFILE-MBF-SVC -MBF ENABLED

Done

> set lb vserver LBVS-1 -netprofile NETPROFILE-MBF-LBVS

Done

> set service SVC-1 -netprofile NETPROFILE-MBF-SVC

Done

<!--NeedCopy-->
```

ネットワークインターフェイスの設定

October 7, 2021

Citrix ADC アプライアンスのネットワークインターフェイスは、<slot><port> 表記で番号が付けられています。インターフェイスを設定したら、インターフェイスとその設定を表示して設定を確認します。この情報を表示して、設定の問題をトラブルシューティングすることもできます。

ネットワークインターフェイスを管理するには、次の操作を行います。

- 一部のインターフェイスを有効にし、その他のインターフェイスを無効にします。
- インターフェイスをリセットして、設定を再ネゴシエートします。
- インターフェイスの累積統計情報を消去します。

設定を確認するには、インターフェイス設定を表示します。インターフェイスの健全性を評価するために、インターフェイスの統計情報を表示できます。

ネットワーク・インタフェース・パラメータの設定

ネットワークインターフェースの設定は、同期も伝達もされません。HA ペアについては、各装置で個別に設定を実行する必要があります。

CLI を使用してネットワークインターフェイスパラメータを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 -  set interface <id> [-speed <speed>] [-duplex <duplex>] [-flowControl
      <flowControl>] [-autoneg ( DISABLED | ENABLED )] [-haMonitor ( ON |
      OFF )] [ ( ON | OFF )] [-tagall ( ON | OFF )] [-lacpMode <lacpMode
     >] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>]
     [-lacpTimeout (LONG | SHORT )] [-ifAlias <string>] [-throughput <
     positive_integer>][-bandwidthHigh <positive_integer> [-
     bandwidthNormal <positive_integer>]]
2 -  show interface [<id>]
3 <!--NeedCopy-->
```

例:

```
1 > set interface 1/8 -duplex full
2  Done
3 <!--NeedCopy-->
```

GUI を使用してネットワークインターフェイスパラメータを設定するには、次の手順を実行します。

[システム**] > [**ネットワーク**] >**[インターフェイス] に移動し、変更するネットワークインターフェイス (たとえば、1/8) を選択し、[編集] をクリックしてパラメータを設定します。

インターフェイスの受信リングサイズおよびリングタイプの設定

Citrix ADC MPX および SDX プラットフォームでは、IX、F1X、F2X、または F4X インターフェイスの受信リングサイズとリングの種類を増やすことができます。

リングサイズを大きくすると、バーストトラフィックを処理するためのクッションが増えますが、パフォーマンスに影響を与える可能性があります。IX インターフェイスでは、最大 8192 のリングサイズがサポートされます。最大 4096 のリングサイズは、F1X、F2X、および F4X インターフェイスでサポートされます。デフォルトのリングサイズは 2048 のままです。

インターフェイスリングタイプはデフォルトで伸縮性があります。パケット到着率に基づいてサイズが増減します。リングタイプを「固定」に設定できます。この場合、リングサイズはトラフィックレートに基づいて変更されません。

注: この機能は、リリース 13.0 ビルド 41.x でサポートされ、IX、F1X、F2X、または F4X インターフェイスを持つプラットフォームでサポートされます。

`show hardware`コマンドを使用して、アプライアンスに IX、F1X、F2X、または F4X インターフェイスがあるかどうかを識別します。

例:

次のモデルには、16 の F1X（10G）インターフェイスと 4 つの F4X（40G）インターフェイスがあります。

```
> sh hardware
        Platform: NSMPX-25000-40G 20\*CPU+16\*F1X+4\*F4X+2\*E1K+2*CVM
            N3 250040
        Manufactured on: 12/16/2016
        CPU: 2800MHZ
        Host Id: 234913926
        Serial no: N43RJCRV3X
        Encoded serial no: N43RJCRV3X
        Netscaler UUID: 336a32d6-2cfa-11e8-bf01-00e0ed5dd23c
        BMC Revision: 4.08
 Done
<!--NeedCopy-->
```

次のモデルには、2 つの 1X（10G）インターフェイスがあります。

```
> sh hardware
        Platform: NSMPX-10500 8\*CPU+2\*E1K+8\*E1K+2\*IX+8*CVM 1620
            760100
        Manufactured on: 12/27/2010
        CPU: 2832MHZ
        Host Id: 1707114630
        Serial no: 7VZZV1ZXJ4
        Encoded serial no: 7VZZV1ZXJ4
        Netscaler UUID: eb1bfd72-5176-11e7-ba18-00e0ed1b0d12
 Done
<!--NeedCopy-->
```


CLI を使用して呼び出し音のサイズと呼び出し音の種類を構成するには
コマンドラインで次のように入力します。

```
set interface <id> -ringsize <positive_integer> -ringtype ( Elastic |
    Fixed )
<!--NeedCopy-->
```

パラメーター:

ringsize:

インターフェイスの受信リングサイズ。値が大きいほど、着信トラフィックを処理するためのバッファが多くなります。

デフォルト値:2048
最小値:512
最大値:16384

ringtype:

インターフェイスの受信リングタイプ。固定リングタイプは、トラフィックレートに関係なく、設定されたバッファ数を事前に割り当てます。対照的に、弾性リングは、着信トラフィックレートに基づいて拡大および縮小します。

設定可能な値: 弾性、固定

デフォルト値: 弾性

例:

```
> set interface 40/2 -ringsize 4096 -ringtype Fixed
 Done
> show interface 40/2

1)      Interface 40/2 (40G Ethernet, CR4, 40 Gbit) #21  flags=0xc020 <
    ENABLED, UP, UP, autoneg, HAMON, HEARTBEAT, 802.1q> MTU=1500, native
     vlan=10, MAC=00:e0:ed:75:14:2a, uptime 119h26m32s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
            throughput 0
        Actual: media UTP, speed 40000, duplex FULL, fctl OFF,
            throughput 40000
        LLDP Mode: NONE, LR Priority: 1024
        RX: Pkts(1443972660032) Bytes(1457207315336105) Errs(0) Drops
            (53319) Stalls(0)
        TX: Pkts(1452311431262) Bytes(1458534011197761) Errs(0) Drops
            (788) Stalls(0)
```



```
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
        Rx Ring: Configured size=4096, Actual size=4096, Type: Fixed
 Done
<!--NeedCopy-->
```

最後の行には、設定済みおよび実際のリングサイズ、およびリングタイプが表示されます。

GUI を使用してリングサイズとリングタイプを設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [インターフェイス] に移動します。
2. インターフェイスを選択し、[**Edit**] をクリックします。
3. [リングサイズ] で、次のいずれかを指定します。
   - **IX** インターフェース： 512,1024,2048,4096 または 8192 です。
   - **F1X**、**F2X**、または **F4X** インターフェイス： 512、1024、2048、または 4096。
4. [リングのタイプ] で、[弾性] または [固定] を選択します
5. [**OK**] をクリックします。

ネットワークインターフェイスの有効化と無効化

デフォルトでは、ネットワークインターフェイスは有効になっています。ネットワークに接続されていないネットワークインターフェイスを無効にして、パケットを送受信できないようにします。高可用性セットアップでネットワークに接続されているネットワークインターフェイスを無効にすると、フェールオーバーが発生する可能性があります。

高可用性の詳細については、「高可用性」を参照してください。

CLI を使用してネットワークインターフェイスを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
-  enable interface <interface_num>
-  show interface <interface_num>
-  disable interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

例:

```
> enable interface 1/8
 Done
```



```
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg,
            802.1q>
        MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h58m40s
        Requested: media UTP, speed AUTO, duplex FULL, fctl OFF,
            throughput 0
        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
<!--NeedCopy-->
```

GUI を使用してネットワークインターフェイスを有効または無効にするには、次の手順を実行します。

1. [システム] > [ネットワーク] > [インターフェイス] に移動します。
2. ネットワークインターフェイスを選択し、[操作] リストで [有効] または [無効] を選択します。

ネットワークインターフェースのリセット

ネットワークインターフェイス設定は、デュプレックスや速度などのプロパティを制御します。ネットワークインターフェイスの設定を再ネゴシエートするには、ネットワークインターフェイスをリセットする必要があります。

CLI を使用してネットワークインターフェイスをリセットするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
-  reset interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

例:

```
> reset interface 1/8
 Done
<!--NeedCopy-->
```

GUI を使用してネットワークインターフェイスをリセットするには、次の手順を実行します。

1. [システム] > [ネットワーク] > [インターフェイス] に移動します。
2. ネットワークインターフェイスを選択し、[操作] リストで [インターフェイスのリセット] を選択します。

ネットワークインターフェースを監視する

ネットワークインターフェイスの統計情報を表示してパラメータを監視し、その情報を使用してネットワークインターフェイスの健全性をチェックできます。送信パケットと受信パケット、スループット、リンク集約制御プロトコル（LACP）データユニット、エラーなどのパラメータを監視できます。ネットワークインターフェイスの統計情報をクリアして、統計情報をクリアした時点からの統計情報をモニタできます。

CLI を使用してネットワークインターフェイスの統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
-  stat interface <interface_num>
<!--NeedCopy-->
```

例:

```
> stat interface 1/8
 Done
<!--NeedCopy-->
```

CLI を使用してネットワークインターフェイスの統計情報をクリアするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
-  clear interface <interface_num>
<!--NeedCopy-->
```

例:

```
> clear interface 1/8
 Done
<!--NeedCopy-->
```

GUI を使用してインターフェイスの統計情報を表示するには、次の手順を実行します。

[システム**] > [**ネットワーク**] > [**インターフェイス] に移動し、ネットワークインターフェイスを選択し、[インターフェイス 統計] をクリックします。

GUI を使用してネットワークインターフェースの統計情報を消去するには、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [**インターフェイス] に移動します。
2. ネットワークインターフェイスを選択し、[操作] リストで [統計のクリア] を選択します。

転送セッション規則の設定

October 7, 2021

デフォルトでは、Citrix ADC アプライアンスは転送するトラフィックのセッションエントリを作成しません（L3 モード）。アプライアンスがサーバに転送するクライアント要求によって、同じパスで応答が返される場合は、転送セッションルールを作成できます。転送セッションルールは、特定のネットワークを発信する、または特定のネットワークを宛先とする、Citrix ADC によって転送されるトラフィックの転送セッションエントリを作成します。IPv4 トラフィックと IPv6 トラフィックの転送セッションルールを作成できます。

IPv4 転送セッション規則を設定する場合、転送セッションエントリを作成する IPv4 トラフィックを識別するための条件として、IPv4 ネットワークアドレスまたは拡張 ACL を指定できます。

- ネットワークアドレス。IPv4 ネットワークアドレスを指定すると、アプライアンスは、送信元または宛先がネットワークアドレスと一致する IPv4 トラフィックの転送セッションを作成します。
- 拡張 **ACL** ルール。拡張 ACL ルールを指定すると、アプライアンスは拡張 ACL ルールで指定された条件に一致する IPv4 トラフィックの転送セッションを作成します。

IPv6 転送セッション規則を設定する場合、転送セッションエントリを作成する IPv6 トラフィックを識別するための条件として、IPv6 プレフィクスまたは ACL6 のいずれかを指定できます。

- **IPv6** プレフィクス。IPv6 プレフィックスを指定すると、アプライアンスは、送信元または宛先が IPv6 プレフィックスと一致する IPv6 トラフィックの転送セッションを作成します。
- **ACL6** ルール。ACL6 ルールを指定すると、アプライアンスは、ACL6 ルールで指定された条件に一致する IPv6 トラフィックの転送セッションを作成します。

CLI を使用して IPv4 転送セッションルールを作成するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、転送セッションルールを作成し、構成を確認します。

- add forwardingSession <name> [<network> <netmask>] | [-aclname <string>] -connfailover (ENABLED | DISABLED)
- show forwardingSession

例**:**

```
A network address as the condition:

> add forwardingSession fs-nw-1 10.102.105.51 255.255.255.255
 Done

An ACL as the condition:

> add forwardingSession fs-acl-1 acl1
```



```
 Done
<!--NeedCopy-->
```

GUI を使用して IPv4 転送セッション規則を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [転送セッション] に移動するか、新しい IPv4 転送セッションを追加するか、既存の転送セッションを編集します。

CLI を使用して IPv6 転送セッションルールを作成するには、次の手順を実行します。

- コマンドプロンプトで次のコマンドを入力して、転送セッションルールを作成し、構成を確認します。
  - 転送セッションを追加 <name>[] <IPv6 prefix>| [-acl6name]<string>
  - show forwardingSession

  例:

```
    An IPv6 prefix as the condition:

    > add forwardingSession fsv6-pfx-1 3ffe::/64
     Done

    An ACL6 rule as the condition:

    > add forwardingSession fsv6-acl6-1  -acl6name ACL6-FS
     Done
<!--NeedCopy-->
```

GUI を使用して IPv6 転送セッション規則を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [転送セッション] に移動するか、新しい IPv6 転送セッションを追加するか、既存の転送セッションを編集します。

既存の転送セッション規則への **ACL** 規則の割り当て

ACL ルールは、ネットワークアドレス/IPv6 プレフィクスベースの転送セッションルールに割り当てることができます。この場合、ACL ベースの転送セッションルールになります。既存の ACL ルールを、ACL ベースの転送セッションルール内の別の ACL ルールに変更することもできます。既存の関連する転送セッションエントリ（存在する場合）がタイムアウトすると、ルールは新しく割り当てられた ACL を使用して、転送セッションエントリを作成する IPv4/IPv6 トラフィックを識別します。

CLI を使用して拡張 ACL ルールを既存の IPv4 転送セッションルールに割り当てるには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

- `set forwardingSession <name> [-aclname <string>]`

- `show forwardingSession <name>`

CLI を使用して、既存の IPv6 転送セッション規則に ACL6 規則を割り当てるには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

- `set forwardingSession <name> [-acl6name <string>]`
- `show forwardingSession <name>`

例:

```
1 > add forwardingSession FS-1 -aclname ACL-9
2 Done
3
4 > add forwardingSession FS6-1  - acl6name ACL6-9
5 Done
```

クラスタセットアップでの転送セッションのステアリングの無効化

Citrix ADC クラスターのデフォルトの動作では、トラフィックを受信するノード（フローレシーバー）が、トラフィックを処理する別のノード（フロープロセッサ）にトラフィックを転送します。フローレシーバからフロープロセッサへのトラフィックの誘導は、クラスタバックプレーン上で行われ、ステアリングと呼ばれます。

ステアリングは、リアルタイム処理や、セットアップに高レイテンシのリンクが含まれている場合にオーバーヘッドになる可能性があります。

転送セッションのステアリングを無効にして、処理がフローレシーバに対してローカルになるようになりました。つまり、フローレシーバがフロープロセッサになります。

はじめに

クラスタ設定で転送セッションルールを設定する前に、次の点に注意してください。

- 転送セッションに使用するリンクセットを設定する必要があります。
- クラスタセットアップで MAC ベースフォワーディング（MBF）を有効にする必要があります。

クラスタ設定での転送セッション規則の設定

クラスタ設定での転送セッション規則のステアリングを無効にするには、次の 2 つのレベルで実行できます。

- 特定の転送セッション規則レベル。新しい転送セッション規則を追加するか、既存の転送セッション規則を編集するときに、Process Local パラメータを有効にします。
- グローバルレベル。新しいクラスタインスタンスの追加時または既存のクラスタインスタンスの編集時に、Process Local パラメータを有効にします。グローバル設定は、転送セッション規則設定よりも優先されます。

CLI のプロシージャ

CLI を使用してクラスタ設定で転送セッション規則のステアリングを無効にするには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しい転送セッション規則を追加する場合:
  - **add forwardingSession** <name> ((<network> [<netmask>]) | -**acl6name** <string> | -**aclname** <string>) -**processLocal ENABLED**
  - **show forwardingSession** <name>
- 既存の転送セッション規則を再設定する場合:
  - **set forwardingSession** <name> -**processLocal ENABLED**
  - **show forwardingSession** <name>

CLI を使用して、クラスタ設定のすべての（グローバルレベル）転送セッション規則のステアリングを無効にするには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しいクラスタ・インスタンスを追加する場合:
  - **add cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>
- 既存のクラスタ・インスタンスを再構成する場合:
  - **set cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>

設定例:

次に、転送セッション規則レベルでステアリングを無効にする 2 つの例と、グローバルレベルでステアリングを無効にする例を示します。

```
1  An IPv4 forwarding session rule:
2
3  > add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV4-1 10.102.105.51
       255.255.255.255 -processLocal Enabled
4  Done
5
6  An IPv6 forwarding session rule:
7
8  > add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV6-1  - acl6name ACL6-
       FWD-SESSN-1 -processLocal Enabled
9  Done
10
```



```
A cluster setup, with an instance ID 10, has steering disabled at
   global level:

> set cluster instance 10 -processLocal Enabled
Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してクラスタセットアップで転送セッションルールのステアリングを無効にするには、次の手順を実行します。

[システム] > [ネットワーク] > [転送セッション] に移動し、[ローカルプロセス] を選択して、新しい転送セッションルールを追加するか、既存の転送セッションルールを編集します。

GUI を使用して、クラスタセットアップのすべての（グローバルレベル）転送セッションルールのステアリングを無効にするには、次の手順を実行します。

[システム] > [クラスタ] に移動し、クラスタ構成の追加または既存のクラスタ構成の変更中に [ローカル処理] を選択します。

VLAN について

October 7, 2021

Citrix ADC アプライアンスは、レイヤー 2 ポートおよび IEEE 802.1q タグ付き VLAN をサポートします。VLAN 構成は、トラフィックを特定のワークステーショングループだけに制限しなければならない場合に便利です。IEEE 802.1q タグ付けを使用して、ネットワークインターフェイスを複数の VLAN の一部として構成できます。

VLAN を設定し、IP サブネットにバインドできます。Citrix ADC は、これらの VLAN 間で IP 転送を実行します（これらのサブネット上のホストのデフォルトルーターとして設定されている場合）。

Citrix ADC では、次のタイプの VLAN がサポートされています。

- ポートベース **VLAN**。ポートベース VLAN のメンバシップは、共通の排他的なレイヤ 2 ブロードキャストドメインを共有する一連のネットワークインターフェイスによって定義されます。複数のポートベース VLAN を構成できます。デフォルトでは、Citrix ADC 上のすべてのネットワークインターフェイスは VLAN 1 のメンバーです。

  ポートに 802.1q タグ付けを適用すると、ネットワークインターフェイスはポートベース VLAN に属します。レイヤ 2 トラフィックはポートベースの VLAN 内でブリッジングされ、レイヤ 2 モードが有効の場合、レイヤ 2 ブロードキャストは VLAN のすべてのメンバーに送信されます。タグなしネットワークインターフェイスを新しい VLAN のメンバーとして追加すると、現在の VLAN から削除されます。

- デフォルト **VLAN**。デフォルトでは、Citrix ADC 上のネットワークインターフェイスは、タグなしのネットワークインターフェイスとして単一のポートベースの VLAN に含まれます。この VLAN がデフォルト VLAN です。VLAN ID（VID）は 1 です。この VLAN は永続的に存在します。デフォルト VLAN を削除したり、その VID を変更したりすることはできません。

  ネットワークインターフェイスを別の VLAN にタグなしメンバーとして追加すると、ネットワークインターフェイスはデフォルト VLAN から自動的に削除されます。現在のポートベース VLAN からネットワークインターフェイスをバインド解除すると、デフォルト VLAN に再び追加されます。

- タグ付き **VLAN**。 802.1q タグ付け（IEEE 802.1q 標準で定義）を使用すると、ネットワーキングデバイス（Citrix ADC など）がレイヤ 2 のフレームに情報を追加して、フレームの VLAN メンバーシップを識別できます。タグ付けにより、ネットワーク環境は複数のデバイスにまたがる VLAN を持つことができます。パケットを受信するデバイスは、タグを読み取り、フレームが属する VLAN を認識します。一部のネットワークデバイスでは、タグ付きパケットとタグなしパケットの両方を同じネットワークインターフェイス（特に Force10 スイッチ）で受信できないことがあります。そのような場合は、カスタマーサポートに連絡して支援を受ける必要があります。

  ネットワークインターフェイスは、VLAN のタグ付きメンバーまたはタグなしメンバーにすることができます。各ネットワークインターフェイスは、1 つの VLAN だけ（ネイティブ VLAN）のタグなしメンバーです。このネットワークインターフェイスは、ネイティブ VLAN のフレームをタグなしフレームとして送信します。他の VLAN にタグが付けられている場合、ネットワークインターフェイスは複数の VLAN の一部になることができます。

  タグ付けを設定する場合は、リンクの両端の VLAN の設定を必ず一致させてください。Citrix ADC が接続するポートは、Citrix ADC ネットワークインターフェイスと同じ VLAN 上にある必要があります。

  注：この VLAN 設定は同期化も伝播もされないため、HA ペアの各ユニットで個別に設定を実行する必要があります。

規則を適用してフレームを分類する

VLAN には、フレームを分類するための 2 種類のルールがあります。

- 入力ルール。入力ルールは、各フレームを 1 つの VLAN にだけ属するものとして分類します。ネットワークインターフェイスでフレームを受信すると、フレームを分類するために次の規則が適用されます。
  - フレームにタグが付けられていないか、タグ値が 0 の場合、フレームの VID は受信インターフェイスのポート VID（PVID）に設定されます。PVID はネイティブ VLAN に属するものとして分類されます。（PVID は IEEE 802.1q 標準で定義されています）。
  - フレームに FFF と等しいタグ値がある場合、フレームはドロップされます。
  - フレームの VID で、受信ネットワークインターフェイスがメンバーではない VLAN が指定されている場合、フレームはドロップされます。たとえば、VLAN ID 12 に関連付けられたサブネットから VLAN ID 10 に関連付けられたサブネットにパケットが送信された場合、パケットはドロップされます。VID

9 のタグなしパケットが VLAN ID 10 に関連付けられたサブネットからネットワークインターフェイス PVID 9 に送信された場合、パケットはドロップされます。

- 出力ルール。次の出力ルールが適用されます。
  - フレームの VID で、伝送ネットワークインターフェイスがメンバーではない VLAN が指定されている場合、フレームは廃棄されます。
  - 学習プロセス（IEEE 802.1q 規格で定義）では、Citrix ADC ブリッジルックアップテーブルを更新するために、Src MAC および VID が使用されます。
  - フレームの VID にメンバーを持たない VLAN が指定されている場合、フレームは廃棄されます。（メンバーを定義するには、ネットワークインターフェイスを VLAN にバインドします）。

Citrix ADC 上の **VLAN** とパケット転送

Citrix ADC アプライアンスの転送プロセスは、標準スイッチと同様です。ただし、Citrix ADC はレイヤー 2 モードがオンの場合にのみ転送を実行します。転送プロセスの主な機能は次のとおりです。

- トポロジの制限が適用されます。適用には、VLAN 内の各ネットワークインターフェイスを伝送ポート（ネットワークインターフェイスの状態による）、ブリッジング制限（受信ネットワークインターフェイスでは転送しない）、および MTU 制限として選択します。
- フレームは、Citrix ADC 転送データベース（FDB）テーブルのブリッジテーブルルックアップの情報に基づいてフィルタリングされます。ブリッジテーブルのルックアップは、宛先 MAC および VID に基づいて行われます。Citrix ADC MAC アドレス宛てのパケットは、上位層で処理されます。
- すべてのブロードキャストおよびマルチキャストフレームは、VLAN のメンバーである各ネットワークインターフェイスに転送されますが、転送は L2 モードが有効の場合だけ行われます。L2 モードが無効の場合、ブロードキャストパケットとマルチキャストパケットはドロップされます。これは、ブリッジングテーブルに現在存在しない MAC アドレスにも当てはまります。
- VLAN エントリには、タグなしメンバセットの一部であるメンバネットワークインターフェイスのリストがあります。これらのネットワークインターフェイスにフレームを転送する場合、タグはフレームに挿入されません。
- ネットワークインターフェイスがこの VLAN のタグ付きメンバーである場合、フレームが転送されるときにタグがフレームに挿入されます。

VLAN が識別されていないブロードキャストパケットまたはマルチキャストパケットを送信した場合、つまり、ルートのネクストホップの NSIP または ND6 の Duplicate Address Detection（DAD; 重複アドレス検出）中に、パケットはすべてのネットワークインターフェイスに送信され、入力規則と出力規則のいずれかに基づいて適切なタグ付けが行われます。ND6 は通常 VLAN を識別し、データパケットはこの VLAN 上でのみ送信されます。ポートベースの VLAN は、IPv4 および IPv6 に共通しています。IPv6 の場合、Citrix ADC はプレフィックスベースの VLAN をサポートします。

VLAN の設定

October 7, 2021

VLAN は、次の環境で実装できます。

- 単一サブネット
- 複数のサブネット
- シングル LAN
- VLAN（タグ付けなし）
- VLAN（802.1q タグ付け）

タグなしのネットワークインターフェイスのみをメンバーとして持つ VLAN を構成する場合、可能な VLAN の合計数は、Citrix ADC で使用可能なネットワークインターフェイスの数に制限されます。VLAN 設定でさらに多くの IP サブネットが必要な場合は、802.1q タグ付けを使用する必要があります。

ネットワークインターフェイスを VLAN にバインドすると、そのネットワークインターフェイスはデフォルト VLAN から削除されます。ネットワークインターフェイスが複数の VLAN の一部である必要がある場合は、ネットワークインターフェイスをタグ付きメンバーとして VLAN にバインドできます。

レイヤ 3 で VLAN 間でトラフィックを転送するように Citrix ADC を設定できます。この場合、VLAN は単一の IP サブネットに関連付けられます。単一サブネットに属する VLAN 内のホストは、同じサブネットマスクと、そのサブネットに接続されている 1 つ以上のデフォルトゲートウェイを使用します。VLAN のレイヤ 3 の設定はオプションです。レイヤ 3 は IP 転送（VLAN 間ルーティング）に使用されます。各 VLAN には、VLAN の IP サブネットを定義する一意の IP アドレスとサブネットマスクがあります。高可用性構成では、この IP アドレスは他の Citrix ADC アプライアンスと共有されます。Citrix ADC は、設定された IP サブネット（VLAN）間でパケットを転送します。

Citrix ADC を構成する場合、重複する IP サブネットを作成しないでください。これにより、レイヤ 3 機能が妨げられます。

各 VLAN は、一意のレイヤ 2 ブロードキャストドメインです。2 つの VLAN は、それぞれ別々の IP サブネットにバインドされ、1 つのブロードキャストドメインに結合できません。2 つの VLAN 間でトラフィックを転送するには、Citrix ADC アプライアンスなどのレイヤー 3 転送（ルーティング）デバイスが必要です。

HA セットアップでの VLAN の設定

高可用性セットアップ用の VLAN 構成では、Citrix ADC アプライアンスのハードウェア構成が同じであり、それらのアプライアンス上で構成された VLAN はミラーイメージである必要があります。

Citrix ADC アプライアンス間で設定が同期されると、正しい VLAN 構成が自動的に実装されます。その結果、すべてのアプライアンスで同じアクションが実行されます。たとえば、ネットワークインターフェイス 0/1 を VLAN2 に追加すると、このネットワークインターフェイスは、高可用性セットアップに参加するすべてのアプライアンスの VLAN 2 に追加されます。

注：HA セットアップでネットワークインターフェイス固有のコマンドを使用する場合、作成した構成は他の Citrix ADC アプライアンスに伝播されません。HA ペア内の 2 つのアプライアンスの設定が同期されたままになるように、HA ペアの各アプライアンスでこれらのコマンドを実行する必要があります。

VLAN の作成または変更

VLAN を設定するには、VLAN エンティティを作成し、ネットワークインターフェイスと IP アドレスを VLAN にバインドします。VLAN を削除すると、そのメンバーインターフェイスがデフォルト VLAN に追加されます。

CLI のプロシージャ

CLI を使用して VLAN を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- `add vlan <id> [-aliasName <string>] [-ipv6DynamicRouting (ENABLED|DISABLED)]`
- `sh vlan <id>`

例:

```
1   > add vlan 2  - aliasName ”Network A”   Done
2 <!--NeedCopy-->
```

CLI を使用してインターフェイスを VLAN にバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- `bind vlan <id> -ifnum <slot/port>`
- `sh vlan <id>`

例:

```
1   > bind vlan 2 -ifnum 1/8   Done
2 <!--NeedCopy-->
```

CLI を使用して IP アドレスを VLAN にバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- `bind vlan <id> -IPAddress <IPAddress> <netMask>`
- `sh vlan <id>`

例:

```
  > bind vlan 2 -IPAddress 10.102.29.54 255.255.255.0   Done
<!--NeedCopy-->
```

CLI を使用して VLAN を削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- `rm vlan <id>`

GUI のプロシージャ

GUI を使用して VLAN を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [VLAN] に移動するか、新しい VLAN を追加するか、既存の VLAN を編集します。
2. IP アドレスを VLAN にバインドするには、[IP バインディング] で、VLAN にバインドする IP アドレスに対応する [アクティブ] オプションを選択します（たとえば、10.102.29.54）。[Type] 列の [IP Address] 列には、各 IP アドレスの IP アドレスの種類（マップされた IP、仮想 IP、サブネット IP など）が [IP Address] 列に表示されます。
3. VLAN へのネットワークインターフェースをバインドするには、インターフェイスのバインディングの下で、あなたは VLAN にバインドすることをインターフェイスに対応する Active オプションを選択します。

VLAN のモニタリング

受信パケット、受信バイト、送信されたパケット、送信されたバイトなどの VLAN 統計情報を表示し、この情報を使用して異常を識別したり、VLAN をデバッグしたりできます。

CLI を使用して VLAN の統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- `stat vlan <vlanID>`

例:

```
  stat vlan 2
<!--NeedCopy-->
```

GUI を使用して VLAN の統計情報を表示するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [VLAN] に移動します。
2. VLAN を選択し、[統計情報] をクリックします。

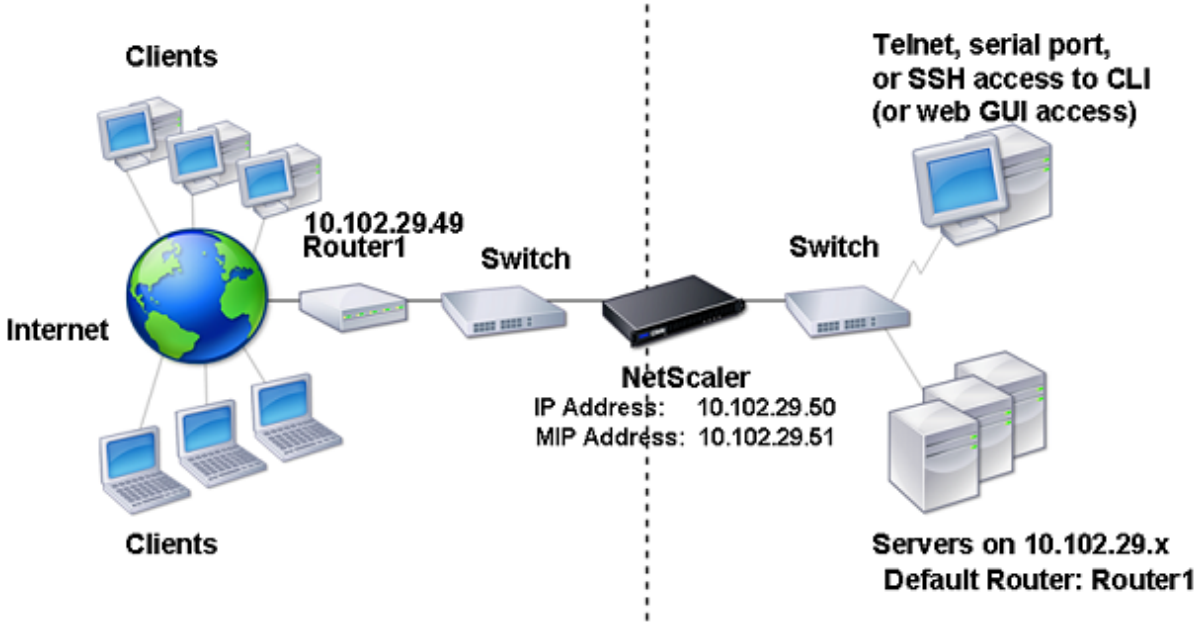
単一サブネットでの **VLAN** の設定

October 7, 2021

単一のサブネットに VLAN を設定する前に、レイヤ 2 モードが有効になっていることを確認してください。

次の図は、単一サブネット環境を示しています。

図 1: 単一サブネット上の VLAN



上の図では、次のようになります。

1. Citrix ADC およびサーバーのデフォルトのルーターは、ルーター 1 です。
2. Citrix ADC がサーバーに直接アクセスできるようにするには、Citrix ADC でレイヤー 2 モードを有効にする必要があります。
3. このサブネットでは、Citrix ADC アプライアンスの負荷分散用に仮想サーバーを構成できます。

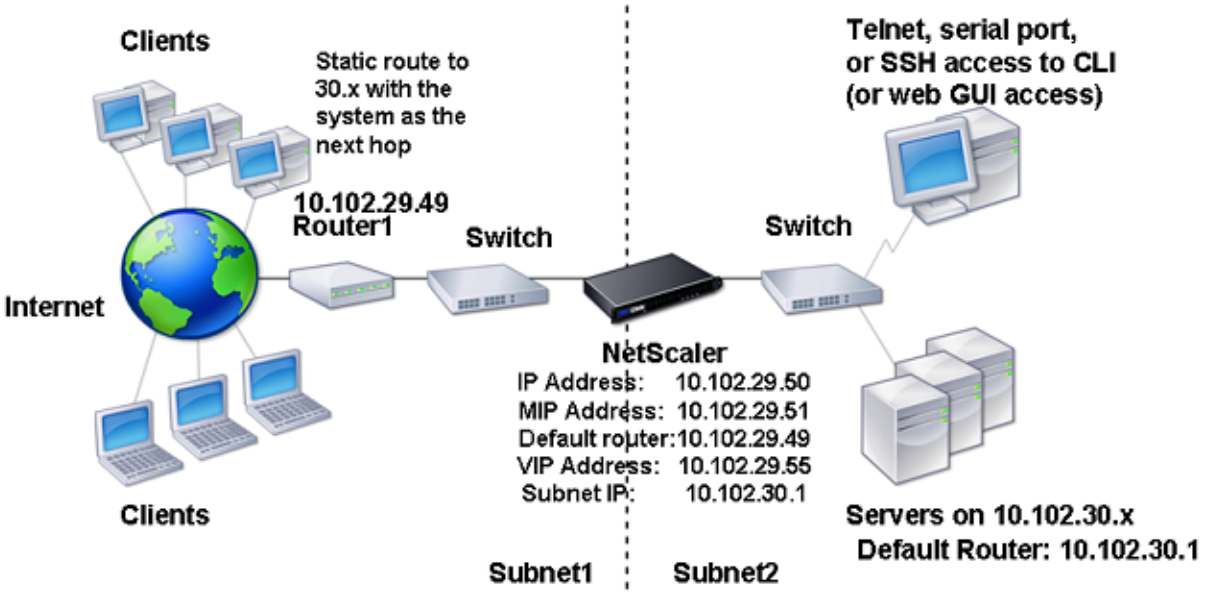
1 つのサブネットに VLAN を設定するには、VLAN の設定で説明されている手順に従います。

複数のサブネットでの **VLAN** の設定

October 7, 2021

複数のサブネットにまたがる 1 つの VLAN を設定するには、VLAN の VIP を追加し、ルーティングを適切に設定する必要があります。次の図は、複数のサブネットにまたがって設定された単一の VLAN を示しています。

図 1: 単一の VLAN 内の複数のサブネット



複数のサブネットにまたがる 1 つの VLAN を設定するには、次の作業を実行します。

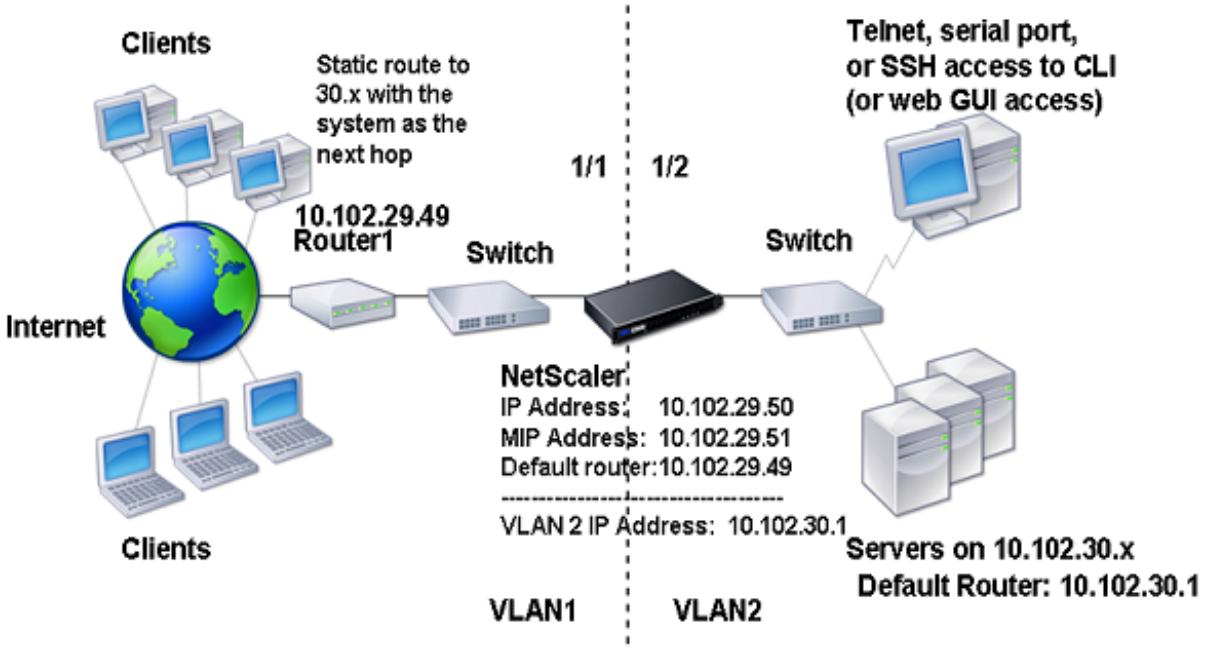
1. レイヤ 2 モードを無効にします。レイヤ 2 モードを無効にする手順については、パケット転送モードを参照してください。
2. VIP アドレスを追加します。VIP アドレスを追加する手順については、仮想 IP アドレス (VIP) の設定と管理を参照してください。
3. RNAT ルールを設定します。RNAT ID を設定する手順については、「RNAT の設定」を参照してください。

複数のサブネットにまたがる複数のタグなし **VLAN** の設定

October 7, 2021

複数のサブネットにまたがるタグなし VLAN が複数ある環境では、IP サブネットごとに 1 つの VLAN が設定されます。ネットワークインターフェイスは、1 つの VLAN にだけバインドされます。次の図は、この構成を示しています。

図 1: VLAN を使用した複数のサブネット-タグ付けなし



上の図に示す設定を実装するには、次のタスクを実行します。

1. VLAN 2 を追加します。
2. Citrix ADC 1/2 ネットワークインターフェイスを、タグなしのネットワークインターフェイスとして VLAN 2 にバインドします。
3. IP アドレスとサブネットマスクを VLAN 2 にバインドします。

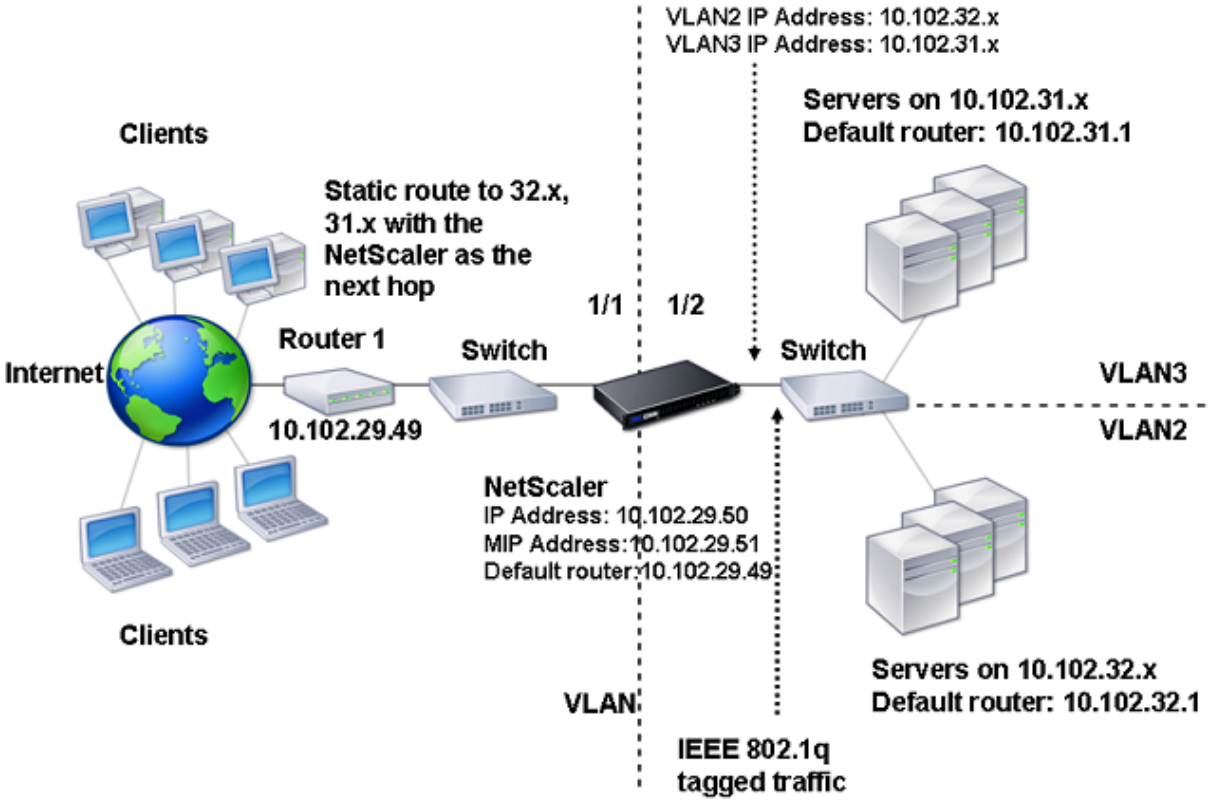
これらのタスクの手順については、VLAN の設定を参照してください。

802.1q タグ付けを使用した複数の VLAN の設定

October 7, 2021

802.1q タグ付けを持つ複数の VLAN の場合、各 VLAN には異なる IP サブネットが設定されます。各ネットワークインターフェイスは 1 つの VLAN にあります。VLAN の 1 つがタグ付きとして設定されています。次の図は、この構成を示しています。

図 1: IEEE 802.1q タグ付けを使用する複数の VLAN



上の図に示す設定を実装するには、次のタスクを実行します。

1. VLAN 2 を追加します。
2. Citrix ADC 1/2 ネットワークインターフェイスを、タグなしのネットワークインターフェイスとして VLAN 2 にバインドします。
3. IP アドレスとネットマスクを VLAN 2 にバインドします。
4. VLAN 3 を追加します。
5. Citrix ADC 1/2 ネットワークインターフェイスを、タグ付きネットワークインターフェイスとして VLAN 3 にバインドします。
6. IP アドレスとネットマスクを VLAN 3 にバインドします。

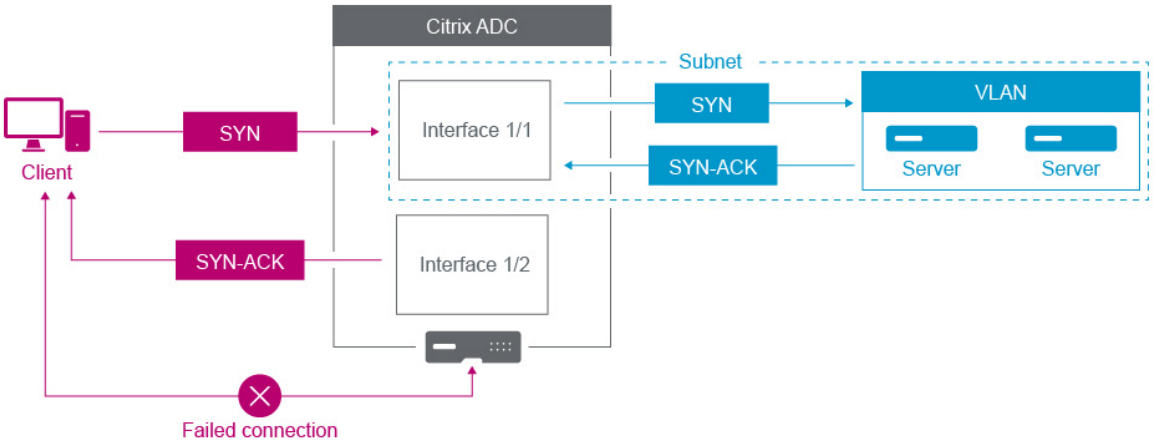
これらのタスクの手順については、VLAN の設定を参照してください。

VLAN を使用した IP サブネットと Citrix ADC インターフェイスの関連付け

October 7, 2021

デフォルトでは、Citrix ADC アプライアンスはネットワークインターフェースを区別しません。アプライアンスは、スイッチよりもネットワークハブのように機能します。これにより、重複したトラフィックが複数のインターフェイスで送信されるレイヤ 3 ネットワークループが発生する可能性があります。

このようなシナリオでは、ネットワーク設計によっては、要求が 1 つのインターフェイスで送信され、対応する応答が別のインターフェイスで受信される可能性があります。



たとえば、あるインターフェイスで送信された SYN パケットと、別のインターフェイスで SYN-ACK 応答を受信すると、アプライアンスは元の SYN パケットを送信したインターフェイスで SYN-ACK を受信すると予想されるため、接続に失敗する可能性があります。

このような問題を解決するために、アプライアンスは内部 VLAN または外部 VLAN を使用して、特定のサブネットをインターフェイスに関連付けることができます。

はじめに

VLAN を使用して IP サブネットと Citrix ADC インターフェイスとの関連付けを開始する前に、次の点に注意してください。

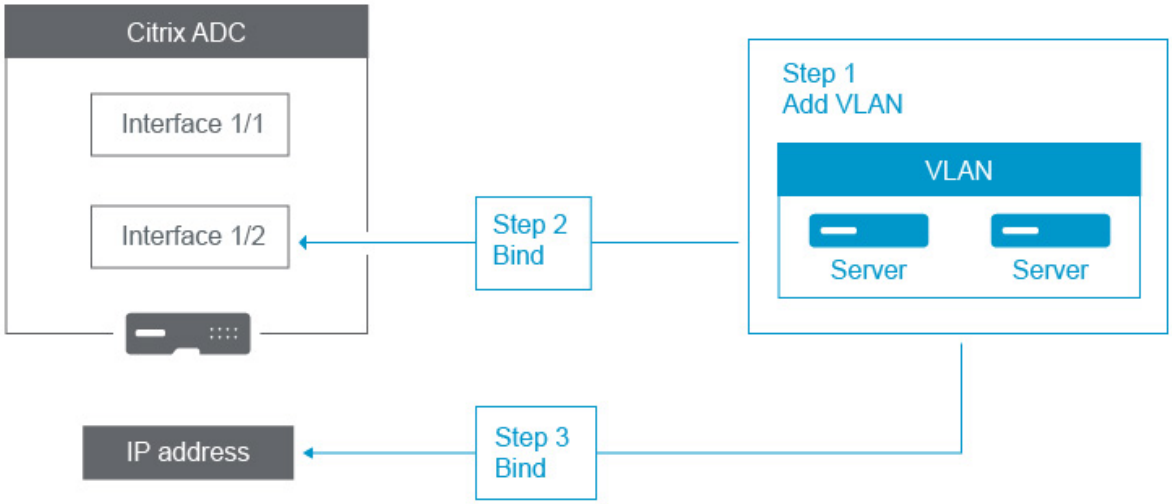
- Citrix ADC GUI またはコマンドラインインターフェイスにアクセスするために使用されているサブネットまたはインターフェイスに VLAN を関連付けると、ネットワーク接続が誤って失われることがあります。したがって、このようなシナリオでは、物理 Citrix ADC アプライアンスのシリアルコンソールまたは Citrix ADC VPX 仮想シリアルコンソールを介してコマンドラインインターフェイスにアクセスすることで、変更を行うことを強くお勧めします。
- Citrix ADC 管理インターフェイスには、特定のハードウェア最適化機能がないため、本番データトラフィックでの使用にはあまり望ましくありません。そのため、管理（NSIP）トラフィックの管理インターフェイスのみを使用するように Citrix ADC を構成することをお勧めします。デフォルト構成では、ハードウェア NetScaler 上の管理インターフェイスとデータインターフェイスの間に論理的な違いはありません。この目標を達成するには、NSIP をデータトラフィックとは別の VLAN に配置することをお勧めします。これにより、管理トラフィックは別のインターフェイス上に配置できます。

  概念は同じですが、NSIP アドレスを含むサブネットの VLAN アソシエーションを変更するには、次の手順ではなく NSVLAN を設定する必要があります。また、このような変更を有効にするには、Citrix ADC を再起動する必要があります。詳細については、NSVLAN の設定を参照してください。

- Citrix ADC SDX では、各インスタンスの NSIP を、SDX の SVM（管理サービス GUI）および XenServer と同じサブネットおよび VLAN 上に置くことを強くお勧めします。SVM はネットワーク経由でインスタンスと通信します。SVM、XenServer、およびインスタンスが同じ VLAN およびサブネット上にない場合、管理トラフィックは SDX の外側に流れる必要があります。この状況では、ネットワークの問題により、インスタンスの状態が黄色または赤で表示され、Citrix ADC インスタンスの管理と構成の変更が妨げられることがあります。

構成の手順

IP サブネットを Citrix ADC インターフェイスに関連付けるには、次のタスクがあります。



VLAN を追加します。VLAN を追加するときに、VLAN にタグを付ける場合は、関連するスイッチポートのネットワークスイッチで定義されている VLAN 番号を選択する必要があります。VLAN がタグなしで、アプライアンスの内部にある場合は、簡単に参照できるように、スイッチ構成で使用可能な VLAN 番号を選択することをお勧めします。

インターフェイスを **VLAN** にバインドします。バインディング中に、リンク集約を使用している場合は、VLAN を物理インターフェイスではなく LA チャネル（LA/1 など）に関連付けます。VLAN は、1 つのネットワークインターフェイスにのみ関連付ける必要があります。

インターフェイス上のトラフィックにタグを付ける場合は、tag（Tag）オプションを使用します。それ以外の場合は、トラフィックはアプライアンスからタグなしになり、スイッチポートのネイティブ VLAN に関連付けられます。

IP アドレスを **VLAN** にバインドします。バインド中に、同じサブネットから複数の IP アドレスをバインドすると、エラーが発生します。IP アドレスが VLAN に関連付けられると、そのサブネット内のすべての IP アドレスが VLAN に自動的に関連付けられます。

> 注:
>
> 高可用性（HA）セットアップでは、これらの VLAN 構成は、HA 同期中にプライマリノードからセカンダリノードに自動的に追加されます。高可用性設定の詳細については、「高可用性」を参照してください。

CLI のプロシージャ

CLI を使用して VLAN を追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add vlan** <id>
- **sh vlan** < id>

CLI を使用してインターフェイスを VLAN にバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **bind vlan** <id> -**ifnum** <slot/port>
- **sh vlan** < id>

CLI を使用して IP アドレスを VLAN にバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **bind vlan** <id> -**IPAddress** <IPAddress> <netMask>
- **sh vlan** < id>

例**:**

```
1 > add vlan 100
2
3 > bind vlan 100 -ifnum 1/1
4
5 > bind vlan 100 -ipAddress 10.0.1.0 255.255.255.0
6 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して VLAN を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**VLAN**] に移動し、新しい VLAN を追加します。
2. VLAN へのネットワークインターフェースをバインドするには、インターフェイスのバインディングの下で、あなたは VLAN にバインドすることをインターフェイスに対応する **Active** オプションを選択します。
3. IP アドレスを VLAN にバインドするには、[**IP** バインディング**]** で、VLAN にバインドする IP アドレスに対応する [アクティブ] オプションを選択します（たとえば、10.102.29.54）。**[**タイプ**]** 列の [IP アドレス] 列には、各 IP アドレスの **IP** アドレスタイプが表示されます。

Citrix ADC アプライアンスのネットワークと VLAN のベストプラクティス

October 7, 2021

Citrix ADC アプライアンスは、VLAN を使用して、どのインターフェイスをどのトラフィックに使用する必要があるかを決定します。また、Citrix ADC アプライアンスはスパニングツリーに参加しません。適切な VLAN 構成がなければ、Citrix ADC アプライアンスは使用するインターフェイスを判断できず、スイッチやルーターよりも HUB のように機能します。つまり、Citrix ADC アプライアンスは、各カンバセーションのすべてのインターフェイスを使用できます。

VLAN 設定ミスの症状

VLAN 構成ミスの問題は、パフォーマンスの問題、接続を確立できない、ランダムに切断されたセッション、重大な状況では、Citrix ADC アプライアンス自体とは無関係に見えるネットワークの中断など、さまざまな形で現れます。Citrix ADC アプライアンスは、ネットワークとの相互作用の正確な性質に応じて、MAC 移動、ミュートされたインターフェイス、管理インターフェイスの送受信バッファオーバーフローを報告することもあります。

MAC 移動（**counter nic_tot_bdg_mac_moved**）：この問題は、Citrix ADC アプライアンスが複数のインターフェイスを使用して同じデバイス（MAC アドレス）と通信していることを示しています。これは、使用するインターフェイスを正しく判断できなかったためです。

ミュートされたインターフェイス（**counter nic_err_bdg_muted**）：この問題は、Citrix ADC アプライアンスが VLAN 構成の問題によりルーティングループを作成していることを検出し、ネットワークを防止するために 1 つ以上の障害のあるインターフェイスをシャットダウンしたことを示します。停止します。

インターフェイスバッファオーバフロー。通常は管理インターフェイス（**counter nic_err_tx_overflow**）を参照します。この問題は、管理インターフェイス経由で送信されるトラフィックが多すぎる場合に発生する可能性があります。Citrix ADC アプライアンスの管理インターフェイスは、大量のトラフィックを処理するように設計されていません。これは、ネットワークおよび VLAN の構成ミスが原因で、Citrix ADC アプライアンスが本番データトラフィックに管理インターフェイスを使用するようにトリガーされる可能性があります。これは、Citrix ADC アプライアンスが、NSIP（NSVLAN）の VLAN /サブネット上のトラフィックを通常の本番トラフィックと区別する方法がないために発生します。NSIP は、ワークステーションやサーバなどの本番デバイスとは別の VLAN およびサブネット上に置くことを強く推奨します。

孤立した **ACK**（**counter tcp_err_orphan_ack**）：この問題は、Citrix ADC アプライアンスが受信した ACK パケットを、通常は ACK の送信元とは異なるインターフェイスで受信したことを示します。この状況は、Citrix ADC アプライアンスが Citrix ADC アプライアンスとの通信に通常使用するターゲットデバイスとは異なるインターフェイスで送信する VLAN 設定の誤りによって発生することがあります（多くの場合、MAC の移動と連動して見られます）。

高いレートの再送信または再送信のあきらめアップ（**counter**：**tcp_err_retransmit_giveups**、**tcp_err_7th_retransmit**、その他のさまざまな再送信カウンタ）：Citrix ADC アプライアンスは、TCP パケットを合計 7 回再送信しようとします。この状況はネットワークの状態によって発生することがありますが、VLAN およびインターフェイスの設定ミスによって発生することがよくあります。

高可用性スプリットブレイン：スプリットブレインは、両方の高可用性ノードがプライマリであると認識し、IP アドレスが重複し、Citrix ADC アプライアンスの機能が失われる状態です。これは、2 つの高可用性ノードが、NSIP を使用して UDP ポート 3003 で高可用性ハートビートを使用して相互に通信できない場合に発生します。これは、通常、Citrix ADC アプライアンスのインターフェイス上のネイティブ VLAN が Citrix ADC アプライアンス間で接続されていない VLAN の構成ミスが原因です。

VLAN およびネットワーク構成のベストプラクティス

1. 各サブネットは、VLAN に関連付ける必要があります。
2. 複数のサブネットを同じ VLAN に関連付けることができます（ネットワーク設計によって異なります）。
3. 各 VLAN は、1 つのインターフェイスだけに関連付ける必要があります（この説明では、LA チャネルは 1 つのインターフェイスとしてカウントされます）。
4. 1 つのインターフェイスに複数のサブネットを関連付ける必要がある場合は、サブネットにタグを付ける必要があります。
5. 一般的な考え方とは対照的に、Citrix ADC アプライアンスの Mac ベースフォワーディング（MBF）機能は、この種の問題を軽減するように設計されていません。MBF は、主に Citrix ADC アプライアンスの DSR（ダイレクト・サーバー・リターン）モード用に設計されています。これは、ほとんどの環境ではほとんど使用されません（バックエンドサーバーからのリターン・パスでトラフィックが意図的に Citrix ADC アプライアンスをバイパスできるように設計されています）。MBF は VLAN の問題を隠す場合がありますが、この種の問題を解決するために信頼すべきではありません。
6. Citrix ADC アプライアンスのすべてのインターフェイスにはネイティブ VLAN が必要です（ネイティブ VLAN がオプションである Cisco とは異なり）。ただし、インターフェイス上の TagAll 設定を使用すると、タグなしのトラフィックが問題のインターフェイスから送信されないようにすることができます。
7. ネイティブ VLAN は、ネットワーク設計に必要な場合にタグ付けできます（インターフェイスの [TagAll] オプション）。
8. Citrix ADC アプライアンスの NSIP のサブネットの VLAN は特殊なケースです。これを NSVLAN と呼びます。概念は同じですが、構成するコマンドは異なります。NSVLAN の変更を有効にするには、Citrix ADC アプライアンスを再起動する必要があります。NSIP と同じサブネットを共有する SNIP に VLAN をバインドしようとすると、「操作は許可されていません」というメッセージが表示されます。これは、代わりに NSVLAN コマンドを使用する必要があるためです。また、一部のファームウェアバージョンでは、`add VLAN` コマンドを使用して VLAN 番号が存在する場合、NSVLAN を設定できません。VLAN を削除してから、NSVLAN を再度設定します。
9. 高可用性ハートビートは、常にそれぞれのインターフェイスのネイティブ VLAN を使用します（インターフェイスで TagAll オプションが設定されている場合、オプションでタグ付けされます）。
10. 高可用性ペアの 2 つのノード上の少なくとも 1 つのネイティブ VLAN セット間には、通信が必要です（直接またはルータ経由のどちらでもかまいません）。ネイティブ VLAN は、高可用性ハートビートに使用されます。

Citrix ADC アプライアンスがインターフェイス上のネイティブ VLAN 間で通信できない場合、高可用性のフェイルオーバーが発生し、両方の Citrix ADC アプライアンスがプライマリであると判断するスプリットブレイン状況が発生する可能性があります（IP アドレスの重複を招くなど）。

11. Citrix ADC アプライアンスはスパニングツリーに参加しません。そのため、Citrix ADC アプライアンスを使用する場合、スパニングツリーを使用してインターフェイスの冗長性を確保することはできません。代わりに、リンクアグリゲーション（LACP または手動 LAG）の形式を使用します。

    注：複数の物理スイッチ間でリンク集約を行う場合は、Cisco の Switch Stack などの機能を使用して、スイッチを仮想スイッチとして設定する必要があります。

12. デフォルトでは、高可用性同期とコマンドプロパゲーションでは、NSIP/NSVLAN が使用されます。これらを別の VLAN に分離するには、`set HA node` コマンドの SyncVLAN オプションを使用します。

13. Citrix ADC アプライアンスのデフォルト構成には、管理インターフェイス（0/1 または 0/2）が管理トラフィックのみに制限されていることを示すものはありません。この制限は、エンドユーザが VLAN 設定を通じて実施する必要があります。管理インターフェイスはデータトラフィックを処理するように設計されていないため、ネットワーク設計ではこの点を考慮する必要があります。Citrix ADC アプライアンスのマザーボードに含まれる管理インターフェイスには、CRC オフロード、より大きなパケットバッファ、その他の最適化などのさまざまなオフロード機能がないため、大量のトラフィックを処理する際の効率が大幅に低下します。本番データと管理トラフィックを分離するには、NSIP をデータトラフィックと同じサブネット/VLAN 上に配置しないでください。

14. 管理インターフェイスを使用して管理トラフィックを伝送する場合は、デフォルトルートを NSIP（NSVLAN）のサブネット以外のサブネットに配置することをお勧めします。

    多くの設定では、デフォルトのルートはワークステーション通信に依存しています（インターネットの場合）。デフォルトルートが NSIP と同じサブネット上にある場合、ADC アプライアンスは管理インターフェイスを使用してデータトラフィックを送受信できます。このデータトラフィックの使用により、管理インターフェイスが過負荷になる可能性があります。

15. また、SDX-SVM、XenServer、およびすべての Citrix ADC インスタンスの NSIP は、同じ VLAN およびサブネット上に存在する必要があります。SDX アプライアンスには、SVM/Xen/インスタンス間の通信を可能にする バックプレーンはありません。同じ VLAN/サブネット/インターフェイス上にない場合、それらの間のトラフィックは物理ハードウェアから送信され、ネットワーク上でルーティングされ、戻ってくる必要があります。

    この設定により、インスタンスと SVM 間の接続の問題が明らかになる可能性があるため、推奨されません。この現象の一般的な症状は、問題の VPX インスタンスの SVM の黄色のインスタンス状態インジケータと、SVM を使用して VPX インスタンスを再構成できないことです。

16. 一部の VLAN がサブネットにバインドされ、一部の VLAN がサブネットにバインドされていない場合、高可用性フェールオーバー中に、VLAN にバインドされていないサブネットの IP アドレスに対して GARP パケットが送信されません。この設定により、高可用性のフェールオーバー中に接続が切断され、接続の問題が発生する可能性があります。この問題は、Citrix ADC アプライアンスがネットワークの MAC 所有権の IP アドレスを VMAC 構成以外の Citrix ADC アプライアンスで通知できないために発生します。

この現象は、高可用性フェイルオーバー中/後に、以前のプライマリ Citrix ADC アプライアンスで ip_tot_floating_ip_err カウンタが数秒以上増加します。これは、ネットワークが GARP パケットを受信または処理せず、ネットワークが新しいセカンダリ Citrix ADC アプライアンスに移行します。

NSVLAN の設定

September 2, 2022

NSVLAN は、Citrix ADC 管理 IP（NSIP）アドレスのサブネットがバインドされている VLAN です。NSIP サブネットは、NSVLAN に関連付けられたインターフェイスでのみ使用できます。デフォルトでは、NSVLAN は VLAN 1 ですが、別の VLAN を NSVLAN として指定できます。その場合、変更を有効にするために Citrix ADC アプライアンスを再起動する必要があります。再起動後、NSIP サブネットトラフィックは新しい NSVLAN に制限されます。

Citrix ADC IP サブネットからのトラフィックには、NSVLAN に指定された VLAN ID でタグ付け（802.1q）できます。接続されているスイッチインターフェイスに、接続されているインターフェイスでこれと同じ VLAN ID をタグ付けして許可するように設定する必要があります。NSVLAN 構成を削除すると、NSIP サブネットは自動的に VLAN1 にバインドされ、デフォルトの NSVLAN が復元されます。

CLI を使用して NSVLAN を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- set ns config -nsvlan <positive_integer> -ifnum <interface_name> … [-tagged (YES|NO)]
- show ns config

  注：この構成は、Citrix ADC アプライアンスが再起動された後に有効になります。

例：

```
1 > set ns config -nsvlan 300 -ifnum 1/1 1/2 1/3 -tagged YES
2  Done
3
4 > save config
5  Done
6 <!--NeedCopy-->
```

CLI を使用してデフォルトの NSVLAN 構成を復元するには、次の手順を実行します。

コマンドプロンプトで入力します。

- unset ns config -nsvlan
- show ns config

例:

```
> unset ns config -nsvlan
 Done
<!--NeedCopy-->
```

GUI を使用して NSVLAN を設定するには、次の手順を実行します。

[システム] > [設定] に移動し、[設定] グループで [NSVLAN 設定の変更] をクリックします。

NSVLAN で MTU を設定する

デフォルトでは、NSVLAN の MTU は 1500 バイトに設定されています。この設定を変更して、スループットとネットワークパフォーマンスを最適化できます。たとえば、ジャンボフレームを処理するように NSVLAN を設定できます。

CLI を使用して NSVLAN の MTU を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set vlan** <id> **-mtu** <positive_integer>
- **show vlan** <id>

GUI を使用して NSVLAN の MTU を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [**VLAN**] に移動し、NSVLAN を開き、[最大伝送単位] パラメータを設定します。

サンプル構成**:**

次の設定例では、VLAN 100 が NSVLAN です。

```
> set ns config -nsvlan 100 -ifnum 1/1 -tagged no

  Warning: The configuration must be saved and the system rebooted for
     these settings to take effect

> set vlan 100 -mtu 1600

  Done

> sh vlan

  1)  VLAN ID: 1

      Link-local IPv6 addr:
```



```
    fe80::947b:52ff:fead:12d5/64

      Interfaces : 1/2 LO/1

  2)  VLAN ID: 100    VLAN Alias Name:

      MTU: 1600

      Interfaces : 1/1

      IPs :

          10.102.53.114     Mask: 255.255.255.0

  Done

> save config

  Done
<!--NeedCopy-->
```

許可 **VLAN** リストの設定

October 7, 2021

Citrix ADC アプライアンスで VLAN が明示的に設定されていて、インターフェイスが VLAN にバインドされている場合、Citrix ADC はインターフェイス上の VLAN のタグ付きパケットを受け入れて送信します。一部の展開（たとえば、Bump in a Wire）では、Citrix ADC アプライアンスが透過デバイスとして機能し、多数の VLAN に関連するタグ付きパケットを受け入れて転送する必要があります。この要件では、多数の VLAN を設定および管理することは実現可能な解決策ではありません。

インターフェイス上の許可 VLAN リストは、VLAN のリストを指定します。インターフェイスは、特定の VLAN に関連するタグ付きパケットを透過的に受け入れ、送信します。これらの VLAN をアプライアンス上で明示的に設定する必要はありません。

許可 **VLAN** リストを設定する前に考慮すべきポイント

許可 VLAN リストを設定する前に、次の点を考慮してください。

- 高可用性セットアップでは、許可 VLAN リストが伝播または同期化されません。したがって、両方のノードで許可 VLAN リストを設定する必要があります。

- ネイティブ VLAN のトラフィックは、許可 VLAN リストでネイティブ VLAN を指定する非メンバーインターフェイスにリークする可能性があります。
- インターフェイスの許可 VLAN リストの一部として最大 60 の VLAN 範囲を指定できます。
- Citrix ADC アプライアンスは、リンク集約チャネルまたは冗長インターフェイスセットの一部であるインターフェイス上の許可された VLAN リストをサポートしていません。冗長インターフェイスセットの詳細については、冗長インターフェイスセットを参照してください。
- 許可された VLAN リストは、Citrix ADC クラスタ構成ではサポートされません。
- Citrix ADC アプライアンスは、ブリッジグループの許可された VLAN リストをサポートしていません。
- Citrix ADC アプライアンスは、VXLAN の許可された VLAN リストをサポートしていません。

許可 **VLAN** リストの設定

CLI を使用して許可 VLAN リストを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set interface** <id> **-trunkmode** (ON|OFF) **-trunkAllowedVlan** <int[-int]> …
- インターフェースを表示 < id>

GUI を使用して許可 VLAN リストを設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [インターフェイス] に移動し、ネットワークインターフェイスを選択し、[編集] をクリックして次のパラメータを設定します。

- トランクモード
- トランクの許可 VLAN

設定例:

次の設定例では、100-120、190-200、300-330 の範囲の VLAN が、インターフェイス 1/2 の許可 VLAN リストの一部として指定されています。

```
> set int 1/2 -trunkmode on -trunkallowedVlan 100-120 190-200 300-330

Done

> sh int 1/2

1)      Interface 1/2 (Gig Ethernet 10/100/1000 MBits) #6
        flags=0xc020

        <ENABLED, UP, UP, AUTONEG OFF, HEARTBEAT, 802.1q, trunkmode>

        Trunk Allowed Vlans:  100-120 190-200 300-330

```



```
14 Done
15
16 <!--NeedCopy-->
```

ブリッジグループの設定

October 7, 2021

通常、2 つ以上の VLAN を 1 つのドメインにマージする場合は、個別のドメイン内のすべてのデバイスで VLAN 設定を変更します。これは面倒な作業になる可能性があります。複数の VLAN を 1 つのブロードキャストドメインに簡単にマージするには、ブリッジグループを使用します。

ブリッジグループ機能は、VLAN と同じように動作します。複数の VLAN を 1 つのブリッジグループにバインドでき、同じブリッジグループにバインドされたすべての VLAN が 1 つのブロードキャストドメインを形成します。ブリッジグループにバインドできるのは、レイヤ 2 VLAN だけです。レイヤ 3 機能を使用するには、ブリッジグループに IP アドレスを割り当てる必要があります。

レイヤ 2 モードでは、特定の VLAN に属するインターフェイスで受信したブロードキャストパケットは、同じブリッジグループに属する他の VLAN にブリッジングされます。ユニキャストパケットの場合、Citrix ADC アプライアンスはブリッジテーブルで、同じブリッジグループに属するすべての VLAN の学習済み MAC アドレスを検索します。

レイヤ 3 転送モードでは、IP サブネットはブリッジグループにバインドされます。Citrix ADC は、バインドされたサブネットに属する着信パケットを受け入れ、ブリッジグループにバインドされた VLAN 上でのみパケットを転送します。

IPv6 ルーティングは、設定されたブリッジグループで有効にできます。

> 注
>
> ブリッジグループ機能とブリッジ BPDU モードは連携できません。

構成の手順

ブリッジグループを設定するには、次の手順を実行します。

- レイヤ 2 モードを有効にする
- ブリッジグループの追加と VLAN のブリッジグループへのバインド

CLI のプロシージャ

CLI を使用してレイヤ 2 モードを有効にするには、
コマンドプロンプトで次のように入力します。

- **enable ns mode** l2

- **show ns mode**

CLI を使用してブリッジグループを追加し、VLAN をバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add bridgegroup** <id> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]
- **bind bridgegroup** <id> -**vlan** <positive_integer>
- **show bridgegroup** <id>

例**:**

```
1 > add bridgegroup 12
2  Done
3 <!--NeedCopy-->
```

CLI を使用してブリッジグループを削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **rm bridgegroup** <id>

例**:**

```
1  rm bridgegroup 12
2 <!--NeedCopy-->
```

GUI の手順

GUI を使用してブリッジグループを設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [ブリッジグループ] に移動するか、新しいブリッジグループを追加して VLAN をブリッジグループにバインドするか、既存のブリッジグループを編集します。

仮想 **MAC** の設定

October 7, 2021

高可用性 (HA) セットアップのプライマリノードとセカンダリノードは、仮想 MAC アドレスのフローティングエンティティを共有します。プライマリノードは、フローティング IP アドレス（MIP、SNIP、VIP など）を所有し、これらの IP アドレスに対する ARP 要求に独自の MAC アドレスで応答します。したがって、アップストリームルータなどの外部デバイスの ARP テーブルは、プライマリノードの Floating IP アドレスと MAC アドレスで更新されます。

フェイルオーバーが発生すると、セカンダリノードが新しいプライマリノードとして引き継がれます。以前のセカンダリノードは Gratuitous ARP (GARP) を使用して、古いプライマリノードから学習したフローティング IP アドレスをアドバタイズします。新しいプライマリノードがアドバタイズする MAC アドレスは、自身のネットワークインターフェイスの MAC アドレスです。一部のデバイス（少数のルータ）は、これらの GARP メッセージを受け付けません。したがって、これらの外部デバイスは、古いプライマリノードがアドバタイズした IP アドレスと MAC アドレスのマッピングを保持します。これにより、GSLB サイトがダウンする可能性があります。

したがって、HA ペアの両方のノードで仮想 MAC を設定する必要があります。これは、両方のノードが同じ MAC アドレスを持っていることを意味します。フェールオーバーが発生しても、セカンダリノードの MAC アドレスは変更されず、外部デバイスの ARP テーブルを更新する必要はありません。

仮想 MAC を設定する手順については、仮想 MAC アドレスの設定を参照してください。

リンク集約の設定

October 7, 2021

リンクアグリゲーションは、複数のポートからのデータを 1 つの高速リンクに結合します。リンクアグリゲーションを設定すると、Citrix ADC アプライアンスと他の接続デバイス間の通信チャネルの容量と可用性が向上します。集約リンクは「チャネル」とも呼ばれます。チャネルは手動で設定することも、リンクアグリゲーション制御プロトコル（LACP）を使用することもできます。LACP は、手動で設定されたチャネルに適用することも、LACP によって作成されたチャネルを手動で設定することもできません。

ネットワークインターフェイスをチャネルにバインドした場合、チャネルのパラメーターは、ネットワークインターフェイスのパラメーターよりも優先されます。（つまり、ネットワークインターフェースパラメータは無視されます）。ネットワークインターフェイスは、1 つのチャネルにしかバインドできません。

ネットワークインターフェイスがチャネルにバインドされると、その VLAN 設定はドロップされます。ネットワークインターフェイスが手動または LACP によってチャネルにバインドされると、元々属していた VLAN から削除され、デフォルトの VLAN に追加されます。ただし、チャネルを元の VLAN や新しい VLAN にバインドすることができます。たとえば、ネットワークインターフェイスをバインドする場合 1/2 そして 1/3 ID 2 の VLAN に接続し、それらをチャネルにバインドします LA/1, ネットワークインターフェイスはデフォルトの VLAN に移動されますが、VLAN2 にバインドして戻すことができます。

リンクアグリゲーションを手動で構成する

リンクアグリゲーションチャネルを作成すると、アクティブなインターフェイスをバインドするまで、その状態は DOWN になります。チャネルはいつでも変更できます。チャネルを削除することも、enable/disable それら。

CLI のプロシージャ

CLI を使用してリンクアグリゲーションチャネルを作成するには：

コマンドプロンプトで入力します。

- add channel <id> [-ifnum <interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl <flowControl>] [-haMonitor (ON | OFF)][-tagall (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- チャンネルを表示

例:

```
> add channel LA/1 -ifnum 1/8
Done
<!--NeedCopy-->
```

CLI を使用して、既存のリンクアグリゲーションチャネルにインターフェイスをバインドまたはバインド解除するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- bind channel <id> <interfaceName>
- unbind channel <id> <interfaceName>

例:

```
bind channel LA/1 1/8
<!--NeedCopy-->
```

CLI を使用してリンクアグリゲーションチャネルを変更するには:

コマンドプロンプトで、
set channel コマンド、チャネル ID、および変更するパラメータを新しい値とともに入力します。

CLI を使用してリンクアグリゲーションチャネルを削除するには:

重要: チャネルが削除されると、それにバインドされたネットワークインターフェイスがネットワークループを引き起こし、ネットワークパフォーマンスが低下します。チャネルを削除する前に、ネットワークインターフェイスを無効にする必要があります。

コマンドプロンプトで入力します。

- rm channel <id>

例:


```
> rm channel LA/1
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してリンクアグリゲーションチャネルを設定するには:

システムに移動 > 通信網 > チャンネル、新しいチャンネルの追加、または既存のチャンネルの編集。

GUI を使用してリンクアグリゲーションチャネルを削除するには:

> 重要:
>
> チャネルが削除されると、それにバインドされたネットワークインターフェイスがネットワークループを引き起こし、ネットワークパフォーマンスが低下します。チャネルを削除する前に、ネットワークインターフェイスを無効にする必要があります。

システムに移動 > 通信網 > チャネルで、削除するチャネルを選択し、[削除] をクリックします。

Link Aggregation Control Protocol を使用したリンクアグリゲーションの構成

リンクアグリゲーション制御プロトコル（LACP）を使用すると、ネットワークデバイスは LACP データユニット（LACPDU）を交換することにより、リンクアグリゲーション情報を交換できます。したがって、手動で作成したチャネルのメンバーであるネットワークインターフェイスで LACP を有効にすることはできません。

LACP を使用してリンクアグリゲーションを設定する場合、リンクアグリゲーションチャネルの変更には、リンクアグリゲーションチャネルの作成とは異なるコマンドとパラメータを使用します。チャネルを削除するには、チャネルの一部であるすべてのインターフェイスで LACP を無効にする必要があります。

注：高可用性設定では、LACP 設定は伝播も同期もされません。

LACP システム優先度の構成

LACP システムの優先順位は、LACP LA チャネルのどのピアデバイスが LA チャネルを制御できるかを決定します。この番号は、アプライアンスのすべての LACP チャネルにグローバルに適用されます。この値が小さいほど優先度が高くなります。

CLI を使用して LACP システムの優先順位を設定するには:

コマンドプロンプトで次のコマンドを入力して、スタンドアロンアプライアンスの優先度を設定し、構成を確認します。

- set lacp -sysPriority <positive_integer>
- show lacp

例:

```
set lacp -sysPriority 50
<!--NeedCopy-->
```

特定のクラスターノードの優先度を設定するには、クラスター IP アドレスにログオンし、コマンドプロンプトで次のコマンドを入力します。

- set lacp -sysPriority <positive_integer> -ownerNode <positive_integer>
- show lacp

例:

```
set lacp -sysPriority 50 -ownerNode 2
<!--NeedCopy-->
```

GUI を使用して LACP システムの優先度を設定するには:

1. システムに移動 > 通信網 > インターフェイスと、[アクション] リストで [LACP の設定] を選択します。
2. システムの優先順位と所有者ノードを指定します（クラスター設定にのみ適用可能）。

リンクアグリゲーションチャネルの作成

LACP を使用してリンク集約チャネルを作成するには、LACP を有効にし、チャネルの一部にする各インターフェイスで同じ LACP キーを指定する必要があります。たとえば、インターフェイス 1/1 および 1/2 で LACP を有効にし、LACP キーを 3 に設定すると、リンク集約チャネル LA/3 が作成され、インターフェイス 1/1 および 1/2 が自動的にバインドされます。

注:

- ネットワークインターフェイスで LACP を有効にする場合は、LACP キーを指定する必要があります。
- デフォルトでは、LACP はすべてのネットワークインターフェイスで無効になっています。

CLI を使用して LACP チャネルを作成するには:

コマンドプロンプトで入力します。

- set interface <id> [-lacpMode <lacpMode>] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>] [-lacpTimeout (LONG | SHORT)]
- show interface [<id>]

GUI を使用して LACP チャネルを作成するには:

システムに移動 > 通信網 > インターフェイス、ネットワークインターフェイスを開き、パラメータを設定します。

リンクアグリゲーションチャネルの変更

インターフェイスを指定して LACP チャネルを作成した後、チャネルのプロパティを変更できます。

CLI を使用して LACP チャネルを変更するには:

コマンドプロンプトで入力します。

- set channel <id> [-ifnum <interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl <flowControl>] [-haMonitor (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-tagall (ON | OFF)] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- チャンネルを表示

例**:**

```
1 > set channel LA/3 -state ENABLED -speed 10000
2  Done
3 <!--NeedCopy-->
```

GUI を使用して LACP チャネルを変更するには:

システムに移動 > 通信網 > チャネル、および既存の LACP チャネルを変更します。

リンクアグリゲーションチャネルの削除

LACP を使用して作成されたリンク集約チャネルを削除するには、チャネルの一部であるすべてのインターフェイスで LACP を無効にする必要があります。

CLI を使用して LACP チャネルを削除するには:

コマンドプロンプトで入力します。

- set interface <id> -lacpMode Disable
- インターフェースを表示 [<id>]

GUI を使用して LACP チャネルを削除するには:

システムに移動 > 通信網 > インターフェイス、ネットワークインターフェイスを開き、[LACP を有効にする] オプションをオフにします。

LACP チャネルを使用したリンク冗長性

LACP チャネルを使用したリンク冗長性により、Citrix ADC は LACP チャネルを論理サブチャネルに分割し、1 つのサブチャネルをアクティブにし、他のサブチャネルをスタンバイモードにすることができます。アクティブなサブチ

ャネルがスループットの最小しきい値を満たしていない場合、スタンバイサブチャネルの 1 つがアクティブになり、引き継ぎます。

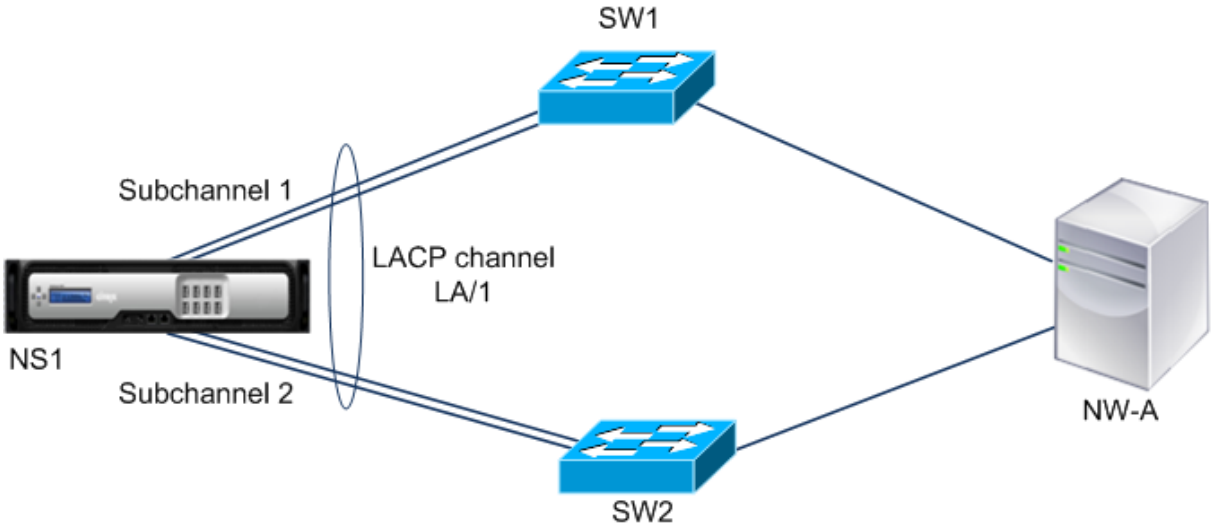
サブチャネルは、LACP チャネルの一部であり、特定のデバイスに接続されているリンクから作成されます。たとえば、Citrix ADC に 4 つのインターフェイスがあり、2 つのインターフェイスがデバイス A に接続され、他の 2 つがデバイス B に接続されている LACP チャネルの場合、ADC は 2 つの論理サブチャネルを作成します。1 つのサブチャネルにはデバイス A への 2 つのリンクがあります。デバイス B への 2 つのリンクを持つ別のサブチャネル。

LACP チャネルのリンク冗長性を設定するには、アクティブなサブチャネルが満たす最小スループットしきい値（Mbps 単位）を指定する lrMinThroughput パラメータを設定します。このパラメータを設定すると、サブチャネルが自動的に作成されます。アクティブチャネルでサポートされている最大スループットが lrMinThroughput 値を下回ると、リンクフェイルオーバーが発生し、スタンバイサブチャネルがアクティブになります。

LACP チャネルの lrMinThroughput パラメータを設定解除するか、値をゼロに設定した場合、そのチャネルのリンク冗長性は無効になります（デフォルト設定）。

例

Citrix ADC NS1 とスイッチ SW1 および SW2 の間に構成されたリンク冗長性の例を考えてみましょう。



NS1 は、SW1 および SW2 を介してネットワークデバイス NW-A に接続されます。
NS1 では、LACP チャネル LA/1 インターフェイスから作成されます 1/1, 1/2, 1/3, そして 1/4. インターフェイス 1/1 そして 1/2 NS1 のは SW1 に接続され、インターフェイス 1/3 そして 1/4 SW2 に接続されています。4 つのリンクはそれぞれ、1000Mbps の最大スループットをサポートします。
lrMinThroughput パラメータが何らかの値（たとえば 2000）に設定されている場合、NS1 はから 2 つの論理サブチャネルを作成します。LA/1, インターフェイスを使用する 1 つのサブチャネル（たとえばサブチャネル 1）1/1 そして 1/2 （SW1 に接続）、およびインターフェイスを使用する他のサブチャネル（サブチャネル 2）1/3 そして 1/4 （SW2 に接続）。

NS1 は、アルゴリズムを適用して、一方のサブチャネル（たとえば、サブチャネル 1）をアクティブにし、もう一方を

スタンバイにします。NS1 とネットワークデバイス NW-A は、アクティブなサブチャネルのみを介して相互にアクセスできます。

サブチャネル 1 がアクティブであり、サポートされる最大スループットが lrMinThroughput 値を下回ったとします（たとえば、リンクの 1 つに障害が発生し、サポートされる最大スループットが 1000 Mbps に低下します）。サブチャネル 2 がアクティブになり、引き継ぎます。

高可用性セットアップでの **LACP** チャネルを使用したリンク冗長性

高可用性（HA）設定で、LACP チャネルでスループット（スループットパラメータ）ベースの HA フェールオーバーおよびリンク冗長性（lrMinThroughput パラメータ）を設定する場合は、スループットパラメータを lrMinThroughput パラメータの値以下に設定する必要があります。

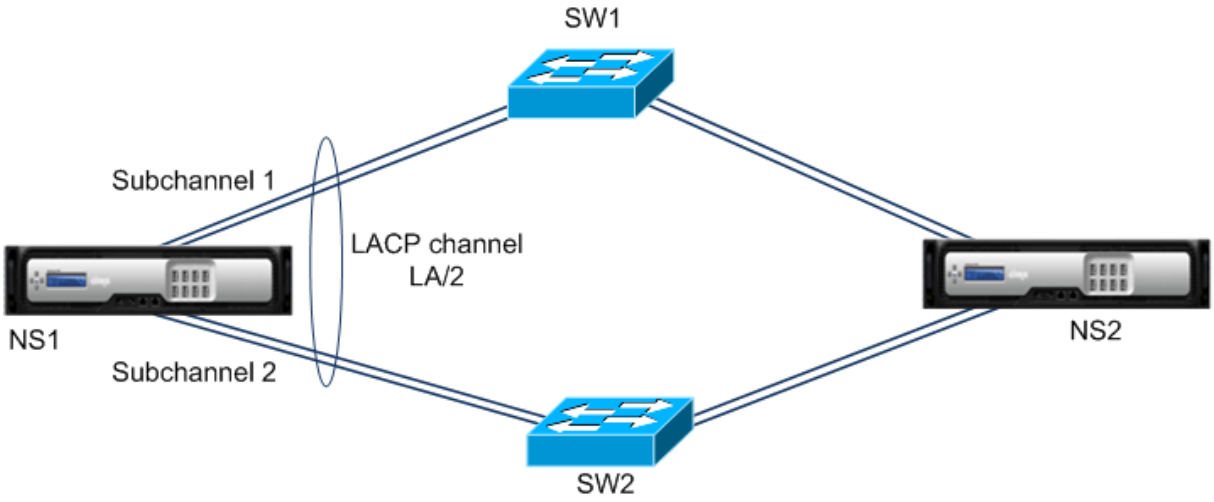
LACP チャネルのサポートされる最大スループットは、アクティブなサブチャネルのサポートされる最大スループットとして計算されます。

スループットパラメータ値が lrminthroughput パラメータ値以下の場合、次の両方の条件が同時に存在すると、HA フェイルオーバーが発生します。

- サポートされているサブチャネルの最大スループットのいずれも、lrMinThroughput パラメーター値を満たしていません。
- LACP チャネルでサポートされている最大スループットがスループットパラメータ値を満たしていません

スイッチ SW1 と SW2 を備えた Citrix ADCNS1 と NS2 を備えた HA セットアップの例を考えてみましょう。NS1 は SW1 と SW2 を介して NS2 に接続されています。

NS1 では、LACP チャネル LA/1 インターフェイスから作成されます 1/1, 1/2, 1/3, そして 1/4. インターフェイス 1/1 そして 1/2 NS1 のは SW1 に接続され、インターフェイス 1/3 そして 1/4 SW2 に接続されています。4 つのリンクはそれぞれ、1000Mbps の最大スループットをサポートします。



この例の LACP パラメーター設定は次のとおりです。

パラメーター	値
スループット	2000
lrminthroughput	2000

NS1 はから 2 つのサブチャネルを形成します LA/1, インターフェイスを使用する 1 つのサブチャネル（たとえばサブチャネル 1）1/1 そして 1/2 （SW1 に接続)、およびインターフェイスを使用する他のサブチャネル（サブチャネル 2）1/3 そして 1/4 （SW2 に接続)。2 つのサブチャネルはそれぞれ、2000Mbps の最大スループットをサポートします。NS1 はアルゴリズムを適用して、一方のサブチャネル（たとえば、サブチャネル 1）をアクティブにし、もう一方をスタンバイにします。

サブチャネル 1 がアクティブであり、サポートされる最大スループットが lrMinThroughput 値を下回ったとします（たとえば、リンクの 1 つに障害が発生し、サポートされる最大スループットが 1000 Mbps に低下します)。サブチャネル 2 がアクティブになり、引き継ぎます。LACP チャネルでサポートされている最大スループットがスループットパラメータ値以上であるため、HA フェイルオーバーは発生しません。

LACP チャネルでサポートされる最大スループット = アクティブチャネルでサポートされる最大スループット = サブチャネル 2 でサポートされる最大スループット = 2000 Mbps

サブチャネル 2 の最大サポートスループットも lrminthroughput 値を下回る場合（たとえば、リンクの 1 つに障害が発生し、最大サポートスループットが 1000 Mbps に低下する場合)、LACP チャネルの最大サポートスループットが以下になるため、HA フェイルオーバーが発生します。スループットパラメータ値:

LACP チャネルを使用してリンク冗長性を構成する

CLI を使用して LACP チャネルのリンク冗長性を設定するには:

コマンドプロンプトで、次のコマンドを入力してチャネルを構成し、構成を確認します。

- **set channel** <id> -lrMinThroughput <positive_integer>
- チャンネルを表示

例**:**

```
1 > set channel la/1 - lrMinThroughput  2000
2 Done
3 > set channel la/2 - throughput 2000 - lrMinThroughput  2000
4 Done
5 <!--NeedCopy-->
```

GUI を使用して LACP チャネルのリンク冗長性を設定するには

1. [システム] > [ネットワーク] > [チャネル] に移動します。

2. 詳細ペインで、リンクの冗長性を構成する LACP チャネルを選択し、[編集] をクリックします。
3. [LACP チャネルの構成] ダイアログボックスで、lrMinThroughput パラメーターを設定します。
4. [閉じる] をクリックします。

冗長インターフェイスセット

October 7, 2021

冗長インターフェイスセットは、インターフェイスの 1 つがアクティブで、残りのインターフェイスがスタンバイであるインターフェイスのセットです。アクティブインターフェイスに障害が発生すると、スタンバイインターフェイスの 1 つが引き継がれ、アクティブになります。

冗長インターフェイスセットを使用する主な利点は次のとおりです。

- 冗長インターフェースセットは、Citrix ADC アプライアンスとピアデバイス間のバックアップリンクを提供することで、Citrix ADC アプライアンスとピアデバイス間の接続の信頼性を保証します。
- LACP を使用したリンク冗長とは異なり、冗長インターフェイスセットに対してピアデバイス上で設定する必要はありません。ピアデバイスでは、冗長インターフェイスセットは個別のインターフェイスとして表示され、セットまたはコレクションとしては表示されません。
- 高可用性構成（HA）では、冗長インターフェイスセットによって HA フェールオーバーの数を最小限に抑えることができます。

> 注
>
> 冗長インターフェイスセットは、以前は 10.5 リリースで初めて導入された「NIC バンドリング」と呼ばれていました。

冗長インターフェイスセットの動作

冗長インターフェイスセットの場合、Citrix ADC アプライアンスは内部アルゴリズムに基づいて MAC アドレスを取得し、それを冗長インターフェイスセットに割り当てます。この MAC アドレスは、すべてのメンバインターフェイスで共有され、一度に使用されるのはアクティブインターフェイスだけです。アクティブインターフェイスは GARP メッセージをブロードキャストします。このメッセージには、インターフェイス自体の物理 MAC アドレスではなく、冗長インターフェイスセットに割り当てられた MAC アドレスが含まれます。現在のアクティブインターフェイスに障害が発生し、別のインターフェイスが引き継がれると、新しいアクティブインターフェイスは GARP メッセージを送信します。ピアデバイスは、新しいアクティブインターフェイス情報でフォワーディングテーブルを更新します。スタンバイインターフェイスは GARP メッセージを送信しません。スタンバイインターフェイスはパケットを送信せず、受信したパケットもドロップします。

冗長インターフェイスセットでは、メンバーインターフェイスをアクティブとして選択するのは、次のいずれかの要因に基づきます。

- 冗長インターフェイスプライオリティ。これはインターフェイスのパラメータで、アクティブなメンバー選択の冗長インターフェイスセット内のインターフェイスのプライオリティを定義します。このパラメータは正の整数を指定します。値を小さくすると、アクティブなメンバー選択の優先度が高くなります。プライオリティ（最低値）が最も高いメンバインターフェイスが、冗長インターフェイスセットのアクティブインターフェイスとして選択されます。
- メンバインターフェイスのバインド順序。すべてのメンバーインターフェイスが同じ冗長インターフェイスプライオリティを持つ場合、最初に冗長インターフェイスセットにバインドされたメンバーインターフェイスが、冗長インターフェイスセットのアクティブインターフェイスとして選択されます。

冗長インターフェイスセットでは、次のいずれかのイベントでアクティブなインターフェイス選択がトリガーされます。

- 現在のアクティブインターフェイスに障害が発生した場合、または無効の場合。
- スタンバイインターフェイスのプライオリティを現在のアクティブインターフェイスのプライオリティよりも低い値に設定した場合。スタンバイインターフェイスがアクティブインターフェイスとして引き継がれます。
- 現在のアクティブインターフェイスのプライオリティよりも低いインターフェイスをバインドする場合。新しくバインドされたインターフェイスがアクティブインターフェイスとして引き継がれます。

冗長インターフェイスセットの設定に関する考慮事項

冗長インターフェイスセットを設定する前に、次の点を考慮してください。

- スタンドアロンアプライアンスまたは高可用性セットアップのアプライアンスでは、リンク冗長セットが LR/X 表記で指定されます。X は 1 ～4 です。たとえば、LR/1 などです。
- 高可用性構成では、冗長インターフェイスセット構成は、セカンダリノードに伝播または同期化されません。
- Citrix ADC アプライアンスでは、最大 4 つの冗長インターフェイスセットを構成できます。
- 冗長インターフェイスセットには、最大 16 個のインターフェイスをバインドできます。
- 冗長インターフェイスセットのメンバーインターフェイスは、別の冗長インターフェイスセットにバインドできません。
- 冗長インターフェイスセットのメンバインターフェイスは、リンク集約（LA）チャネルにバインドできません。
- LA チャネルは冗長インターフェイスセットにバインドできません。
- 冗長インターフェイスセットは LA チャネルにバインドできません。
- クラスタ設定では、次の操作を行います。
  - 冗長インターフェイスセットをクラスタリンク集約にバインドすることはできません。
  - リンク冗長セットは、N/LR/X 表記で指定されます（たとえば、1/LR/3）。ここで、
    N は、冗長インターフェイスセットが作成されるクラスタノードの ID です。
    X は、クラスタノード上のリンク冗長セット識別子です。X の範囲は 1 ～4 です。
  - クラスタリンク集約は、冗長インターフェイスセットにバインドできません。
  - 冗長インターフェイスセットには、冗長インターフェイスセットが属するノードのインターフェイスだけを含めることができます。
  - スタンドアロンアプライアンス上の既存の elink 冗長セット構成は、アプライアンスがクラスタ設定に追加されると、自動的にクラスタ表記（N/LR/X）に変更されます。

構成の手順

Citrix ADC アプライアンスで冗長インターフェースセットを構成するには、次の作業を行います。

- 冗長インターフェイスセットを作成します。冗長インターフェイスセットを作成するには、channel コマンド操作を使用します。

  スタンドアロンアプライアンスまたは高可用性セットアップのアプライアンスでは、リンク冗長セットが LR/X 表記で指定されます。X は 1 ～4 です。たとえば、LR/1 などです。

  クラスタセットアップでは、リンク冗長セットが N/LR/X で指定されます（たとえば、1/LR/3）。ここで、N は冗長インターフェイスセットを作成するクラスタノードの ID、X はクラスタノード上のリンク冗長セット ID です。X の範囲は 1 ～4 です。

- インターフェイスを冗長インターフェイスセットにバインドします。必要なインターフェイスを冗長インターフェイスセットに関連付けます。1 つのインターフェイスを複数の冗長インターフェイスセットの一部にすることはできません。

- （任意）メンバーインターフェイスに冗長インターフェイスプライオリティを設定します。interface コマンド動作を使用して、冗長インターフェイスセットの目的のメンバインターフェイスに冗長インターフェイスプライオリティを設定します。

CLI を使用して冗長インターフェイスセットを作成するには、次の手順を実行します。

コマンドプロンプトで、次の操作を行います。

- add channel <ID>
- show channel <ID>

CLI を使用してインターフェイスを冗長インターフェイスセットにバインドするには、次の手順を実行します。

コマンドプロンプトで、次の操作を行います。

- bind channel <ID> <ifnum>
- show channel <ID>

CLI を使用してインターフェイスの冗長インターフェイスプライオリティを設定するには、次の手順を実行します。

コマンドプロンプトで、次の操作を行います。

- set interface <ID> -lrsetpriority <positive_integer>
- show interface <ID>

設定例 **1:**

次の例では、冗長インターフェイスセット LR/1 が作成され、インターフェイス 1/1、1/2、1/3、および 1/4 が LR/1 にバインドされます。冗長インターフェイスプライオリティは、これらすべてのメンバインターフェイスのデフォルト値の 1024 に設定されます。show channel コマンドの出力は、インターフェイス 1/1 が冗長インターフェイスセット lr/1 の現在のアクティブインターフェイスであることを示しています。


```
> add channel lr/1
 Done
> bind channel lr/1 1/1 1/2 1/3 1/4
 Done
> show channel
1)    Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                 throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Active Member
                1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
Done
<!--NeedCopy-->
```

設定例 **2:**

次の例では、メンバインターフェイス 1/4 の冗長インターフェイスプライオリティが 100 に設定されています。これは、LR/1 の他のすべてのメンバインターフェイスの冗長インターフェイスプライオリティの設定よりも低くなっています。

show channel コマンドの出力は、インターフェイス 1/4 が冗長インターフェイスセット LR/1 の現在のアクティブインターフェイスであることを示しています。

```
> set interface 1/4 -lrsetPriority 100
Done
> show channel
1)      Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
```



```
                  throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Active Member
Done
<!--NeedCopy-->
```

設定例 **3:**

4 つのノード N1、N2、N3、および N4 のクラスタセットアップを考えてみます。この例では、冗長インターフェイスセット 1/LR/3 がノード N1 上に作成され、インターフェイス 1/1/1、1/1/2、および 1/1/3 がこれにバインドされます。冗長インターフェイスプライオリティは、これらすべてのメンバインターフェイスのデフォルト値の 1024 に設定されます。show channel コマンドの出力は、インターフェイス 1/1/1 が冗長インターフェイスセット 1/LR/3 の現在のアクティブインターフェイスであることを示しています。

```
    > add channel 1/LR/3

    Done
    > bind channel 1/LR/3 1/1/1 1/1/2 1/1/3

    Done
    > show channel
    1)    Interface 1/LR/3 (Link Redundant) #14
            flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON,
               802.1q>
            MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0
               h00m00s
            Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
            throughput 0
            Actual: throughput 1000
            LLDP Mode: NONE,
            RX: Pkts(66) Bytes(4406) Errs(0) Drops(82) Stalls(0)
```



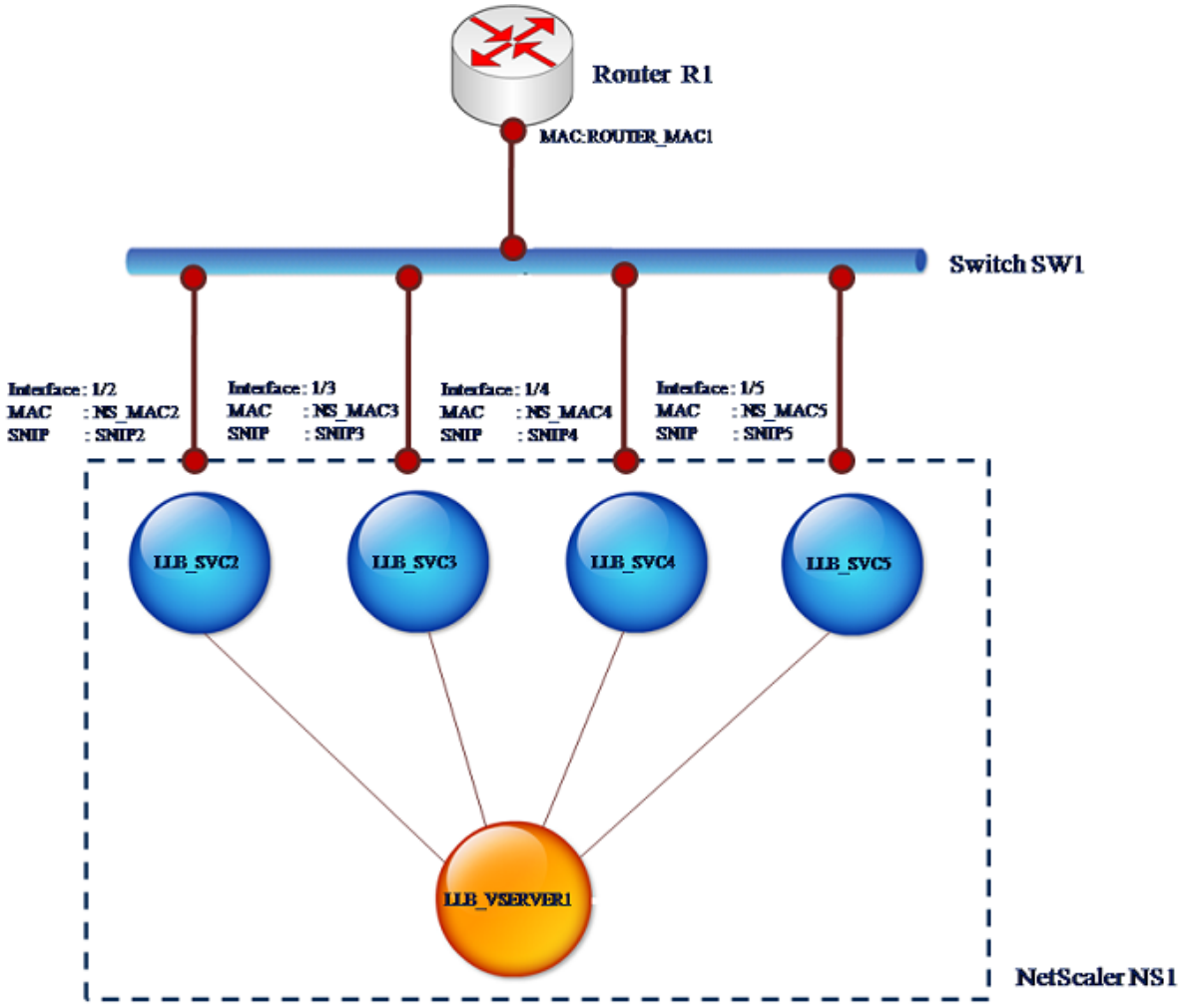
```
16              TX: Pkts(55) Bytes(2626) Errs(0) Drops(145) Stalls(0)
17              NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted
                    (0)
18              Bandwidth thresholds are not set.
19
20              1/1/1: UTP-1000-FULL-OFF UP  0h14m06s LR Active Member
21              1/1/2: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
22              1/1/3: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
23
24      Done
25  <!--NeedCopy-->
```

インターフェイスへの **SNIP** アドレスのバインド

October 7, 2021

レイヤ 3 VLAN を使用せずに、Citrix ADC が所有する SNIP アドレスをインターフェイスにバインドできるようになりました。SNIP アドレスに関連するパケットは、バインドされたインターフェイスだけを通過します。

この機能は、アップストリームスイッチがリンクアグリゲーションチャネルをサポートしていない場合に、次の図に示すように、サーバから発信されたトラフィックを、アップストリームスイッチへの 4 つのリンクにわたって Citrix ADC アプライアンスに負荷分散させる場合に役立ちます。



次の表に、シナリオの設定例を示します。

エンティティ	名前	値
NS1 上の SNIP アドレス	SNIP2（参考目的のみ）	10.10.10.2
	SNIP3（参考目的のみ）	10.10.10.3
	SNIP4（参考目的のみ）	10.10.10.4
	SNIP5（参考目的のみ）	10.10.10.5
NS1 上の LLB 仮想サーバ	LLB_VSERVER1	-
NS1 上の透過モニタ	TRANS_MON	-
NS1 上の LLB サービス	LLB_SVC2	10.10.10.240
	LLB_SVC3	10.10.10.120
	LLB_SVC4	10.10.10.60

エンティティ	名前	値
	LLB_SVC5	10.10.10.30
NS1 上のインタフェース 1/2 の MAC アドレス	NS_MAC_2（参照用のみ）	00:e0:ed:0f:bc:e0
NS1 上のインターフェイス 1/3 の MAC アドレス	NS_MAC_3（参照用のみ）	00:e0:ed:0f:bc:df
NS1 上のインターフェイス 1/4 の MAC アドレス	NS_MAC_4（参照用のみ）	00:e0:ed:0f:bc:de
NS1 上のインタフェース 1/5 の MAC アドレス	NS_MAC_5（参照用のみ）	00:e0:ed:1c:89:53
ルータ R1 の IP アドレス	ルータ IP（参照目的のみ）	10.10.10.1
R1 のインターフェースの MAC アドレス	ROUTER_MAC1 (参照用のみ)	00:21:a1:2d:db:cc

設定例を構成するには、次の手順を実行します。

1. 異なるサブネット範囲に 4 つの異なる SNIP を追加します。これは、ARP が 4 つの異なるリンクで解決されるためです。SNIP アドレスの作成の詳細については、「サブネット IP アドレス (SNIP) の設定」を参照してください。

   CLI の例:

```
> add ns ip 10.10.10.2 255.255.255.0 -type SNIP
 Done
> add ns ip 10.10.10.3 255.255.255.128  - type SNIP
Done
> add ns ip 10.10.10.4 255.255.255.192  - type SNIP
Done
> add ns ip 10.10.10.5 255.255.255.224  - type SNIP
Done
<!--NeedCopy-->
```

2. 追加された SNIP サブネットに 4 つの異なるダミーサービスを追加します。これは、4 つの設定済み SNIP の 1 つとして、送信元 IP を使用してトラフィックが送信されるようにするためです。サービスの作成の詳細については、「基本的な負荷分散の設定」を参照してください。

   CLI の例:


```
> add service LLB_SVC2 10.10.10.240 any *
 Done
> add service LLB_SVC3 10.10.10.120 any *
 Done
> add service LLB_SVC4 10.10.10.60 any *
 Done
> add service LLB_SVC5 10.10.10.30 any *
 Done
<!--NeedCopy-->
```

3. Gateway を監視するためのトランスペアレント ping モニタを追加します。設定された各ダミーサービスにモニタをバインドします。これは、サービスの状態を UP にすることです。トランスペアレントモニタの作成の詳細については、「ロードバランシングセットアップでのモニタの設定」を参照してください。

   CLI の例:

```
> add monitor TRANS_MON ping -destIP 10.10.10.1 -transparent YES
 Done
> bind monitor TRANS_MON LLB_SVC2
 Done
> bind monitor TRANS_MON LLB_SVC3
 Done
> bind monitor TRANS_MON LLB_SVC4
 Done
> bind monitor TRANS_MON LLB_SVC5
 Done
<!--NeedCopy-->
```

4. リンクロードバランシング (LLB) 仮想サーバーを追加し、ダミーサービスをバインドします。LLB 仮想サーバーの作成の詳細については、基本的な LLB セットアップの構成を参照してください。

   CLI の例:

```
> add lb vserver LLB_VSERVER1 any
 Done
> set lb vserver LLB_VSERVER1 -lbmethod ROUNDROBIN
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
```



```
10  Done
11 > bind lb vserver LLB_VSERVER1 LLB_SVC2
12  Done
13 <!--NeedCopy-->
```

5. LLB 仮想サーバをデフォルトの LLB ルートとして追加します。LLB ルートの作成の詳細については、基本的な LLB セットアップの設定を参照してください。

   CLI の例:

```
1 > add lb route 0.0.0.0 0.0.0.0  LLB_VSERVER1
2  Done
3 <!--NeedCopy-->
```

6. Gateway の MAC アドレスを使用して、各ダミーサービスの ARP エントリを追加します。このようにして、これらのダミーサービスを介して Gateway に到達できます。ARP エントリの追加の詳細については、スタティック ARP の設定を参照してください。

   CLI の例:

```
1 > add arp -ipaddress 10.10.10.240 -mac 00:21:a1:2d:db:cc -ifnum
     1/2
2  Done
3 > add arp -ipaddress 10.10.10.120 -mac 00:21:a1:2d:db:cc -ifnum
     1/3
4  Done
5 > add arp -ipaddress 10.10.10.60 -mac 00:21:a1:2d:db:cc -ifnum 1/4
6  Done
7 > add arp -ipaddress 10.10.10.30 -mac 00:21:a1:2d:db:cc -ifnum 1/5
8  Done
9 <!--NeedCopy-->
```

7. SNIP ごとに ARP エントリを追加して、特定のインターフェイスを SNIP にバインドします。これは、応答トラフィックが、要求が送信されたのと同じインターフェイスに到達するようにするためです。ARP エントリの追加の詳細については、スタティック ARP の設定を参照してください。

   CLI の例:

```
1 > add arp -ipAddress 10.10.10.2 -mac 00:e0:ed:0f:bc:e0 -ifnum 1/2
2  Done
3 > add arp -ipAddress 10.10.10.3 -mac 00:e0:ed:0f:bc:df -ifnum 1/3
```

```
 Done
> add arp -ipAddress 10.10.10.4 -mac 00:e0:ed:0f:bc:de -ifnum 1/4
 Done
> add arp -ipAddress 10.10.10.5 -mac 00:e0:ed:1c:89:53 -ifnum 1/5
 Done
<!--NeedCopy-->
```

ブリッジテーブルのモニタとエージングタイムの変更

October 7, 2021

Citrix ADC アプライアンスは、宛先 MAC アドレスと VLAN ID のブリッジテーブル検索に基づいてフレームをブリッジします。ただし、アプライアンスはレイヤ 2 モードが有効になっている場合にのみ転送を実行します。

ブリッジテーブルは動的に生成されますが、表示したり、ブリッジテーブルのエージングタイムを変更したり、ブリッジング統計情報を表示したりできます。ブリッジテーブルのすべての MAC エントリがエージングタイムで更新されます。

CLI を使用してブリッジテーブルエントリのエージングタイムを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set l2param -bridgeagetimeout** <positive_integer>
- **show l2param**

例**:**

```
> set l2param -bridgeagetimeout 90
 Done
<!--NeedCopy-->
```

CLI を使用してブリッジテーブルの統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat bridge**

GUI を使用してブリッジテーブルエントリのエージングタイムを設定するには、次の手順を実行します。

[システム] > [ネットワーク] に移動します。[ネットワーク] ページの [設定] セクションで、[レイヤ **2** パラメータの設定] をクリックし、**[**ブリッジテーブルエントリのタイムアウト値 **(**秒**)]** パラメータを設定します。

GUI を使用してブリッジテーブルの統計情報を表示するには、次の手順を実行します。

[システム] > [ネットワーク] > [ブリッジテーブル] に移動し、MAC アドレスを選択し、[統計] をクリックします。

VRRP を使用したアクティブ/アクティブモードの Citrix ADC アプライアンス

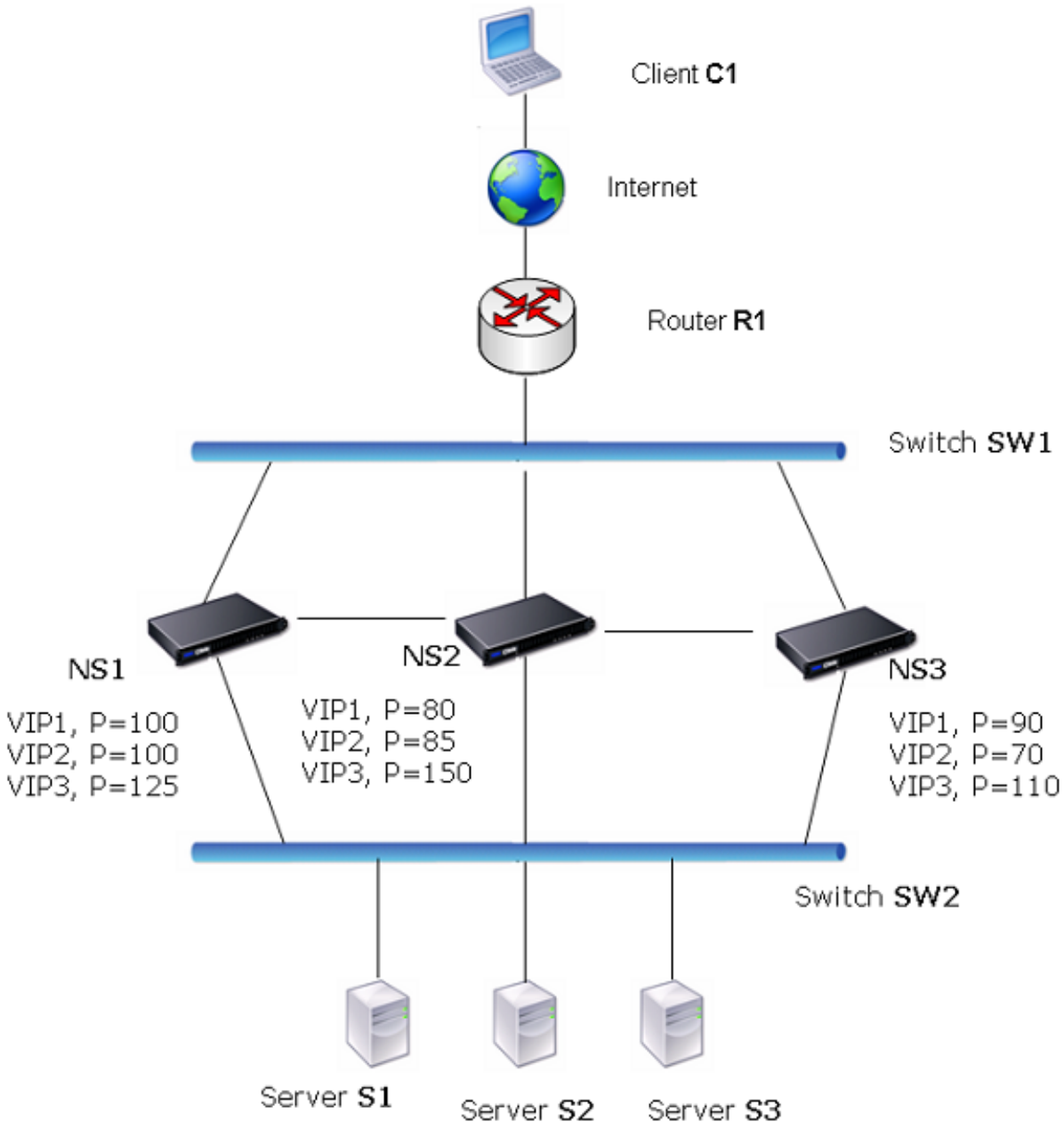
October 7, 2021

アクティブ-アクティブ展開では、ダウンタイムを防止するだけでなく、展開内のすべての Citrix ADC アプライアンスを効率的に使用できます。アクティブ-アクティブ展開モードでは、構成内のすべての Citrix ADC アプライアンスに同じ VIP が設定されますが、優先順位は異なります。これにより、特定の VIP は一度に 1 つのアプライアンスでのみアクティブになります。

アクティブ VIP はマスター VIP と呼ばれ、他の Citrix ADC アプライアンス上の対応する VIP はバックアップ VIP と呼ばれます。マスター VIP に障害が発生すると、プライオリティが最も高いバックアップ VIP が引き継がれ、マスター VIP になります。アクティブ/アクティブ展開のすべての Citrix ADC アプライアンスは、仮想ルーター冗長プロトコル（VRRP）プロトコルを使用して、VIP とそれに対応する優先順位を定期的にアドバタイズします。

アクティブ-アクティブモードの Citrix ADC アプライアンスは、Citrix ADC がアイドル状態でないように構成できます。この構成では、Citrix ADC ごとに異なる VIP セットがアクティブになります。たとえば、次の図では、VIP1、VIP2、VIP3、および VIP4 がアプライアンス NS1、NS2、および NS3 上で構成されています。これらの優先順位により、VIP1 と VIP 2 は NS1 上でアクティブになり、VIP3 は NS2 上でアクティブになり、VIP 4 は NS3 上でアクティブになります。たとえば、NS1 に障害が発生すると、NS3 上の VIP1 と NS2 上の VIP2 がアクティブになります。

図 1: アクティブ-アクティブ設定



上の図の Citrix ADC アプライアンスは、次のようにトラフィックを処理します。

1. クライアント C1 は VIP1 に要求を送信します。要求は R1 に到達します。
2. R1 には VIP1 用の ARP エントリがないため、VIP1 に対する ARP 要求をブロードキャストします。
3. VIP1 は NS1 でアクティブであるため、NS1 は VIP1 に関連付けられた仮想 MAC（仮想 MAC1 など）として送信元 MAC アドレス、および VIP1 を送信元 IP アドレスとして応答します。
4. SW1 は、ARP 応答から VIP1 のポートを学習し、ブリッジテーブルを更新します。
5. R1 は ARP エントリを仮想 MAC1 および VIP1 で更新します。

6. R1 は NS1 上の VIP1 にパケットを転送します。
7. NS1 の負荷分散アルゴリズムはサーバー S2 を選択し、NS1 はその SNIP アドレスの 1 つと S2 の間の接続を開きます。
8. S2 は、Citrix ADC の SNIP に応答します。
9. NS1 は、S2 の応答をクライアントに送信します。応答では、NS1 は物理インターフェイスの MAC アドレスを送信元 MAC アドレスとして、VIP1 を送信元 IP アドレスとして挿入します。
10. NS1 に障害が発生した場合、Citrix ADC アプライアンスは VRRP プロトコルを使用して、優先順位が最も高い VIP1 を選択します。この場合、NS3 上の VIP1 がアクティブになり、次の 2 つの手順でアクティブ-アクティブ設定を更新します。
11. NS3 は VIP1 の GARP メッセージをブロードキャストします。このメッセージでは、仮想 MAC1 は送信元 MAC アドレス、VIP1 は送信元 IP アドレスです。
12. SW1 は、仮想 MAC1 の新しいポートを GARP ブロードキャストから学習し、ブリッジテーブルを更新して VIP1 に対する後続のクライアント要求を NS3 に送信します。R1 は ARP テーブルを更新します。

VIP のプライオリティは、ヘルストラッキングによって変更できます。ヘルストラッキングを有効にする場合は、優先度が低い VIP が別の VIP によってプリエンプトされるように、プリエンプションも有効になっていることを確認する必要があります。

場合によっては、トラフィックがバックアップ VIP に到達することがあります。このようなトラフィックのドロップを回避するには、アクティブ/アクティブ設定を作成するときに、ノード単位で共有を有効にできます。または、マスターへのグローバル送信オプションを有効にすることもできます。共有が有効になっているノードでは、共有がマスターに送るよりも優先されます。

健全性の追跡

ベースプライオリティ（BP 範囲 1 ～255）は、通常、どの VIP がマスター VIP であるかを決定しますが、実効プライオリティ（EP）も決定に影響します。

たとえば、NS1 の VIP のプライオリティが 101 で、NS2 の同じ VIP のプライオリティが 99 の場合、NS1 の VIP はアクティブです。ただし、NS1 で 2 つの vServer が VIP を使用していて、そのうちの 1 つが DOWN すると、NS1 の VIP の EP がヘルストラッキングによって削減されます。次に、VRRP は NS2 上の VIP をアクティブな VIP にします。

EP を変更するためのヘルストラッキングオプションは次のとおりです。

- なし。追跡なし。EP = BP
- **ALL**。すべての仮想サーバが UP の場合は、EP = BP。それ以外の場合は、EP = 0 です。
- **ONE.** 少なくとも 1 つの仮想サーバが UP の場合、EP = BP。それ以外の場合は、EP = 0 です。
- **PROGRESSIVE.** すべての仮想サーバが UP の場合は、EP = BP。すべての仮想サーバが DOWN している場合、EP = 0。それ以外の場合、EP = BP（1-K/N）。ここで、N は VIP に関連付けられた仮想サーバの総数、k はダウンしている仮想サーバの数です。

注： NONE 以外の値を指定する場合は、優先度が最も高いバックアップ VIP がアクティブになるように、マスター VIP の優先度がダウングレードされる必要があります。

プリエンプション

より高いプライオリティを達成する別の VIP によるアクティブ VIP のプリエンプションは、デフォルトで有効であり、通常は有効にする必要があります。ただし、場合によっては、無効にすることもできます。プリエンプションは、各 VIP のノード単位の設定です。

プリエンプションは、次の状況で発生します。

- アクティブ VIP がダウンし、プライオリティの低い VIP が優先されます。プライオリティが高い VIP がオンラインに戻ると、現在アクティブな VIP がプリエンプトされます。
- ヘルストラッキングにより、バックアップ VIP のプライオリティがアクティブ VIP のプライオリティよりも高くなります。バックアップ VIP は、アクティブ VIP をプリエンプトします。

共有

トラフィックがバックアップ VIP に到達した場合、バックアップ VIP で共有オプションが有効になっていない限り、トラフィックはドロップされます。この動作は、各 VIP のノード単位の設定であり、デフォルトでは無効になっています。

図では、NS1 上の アクティブ/アクティブ構成 VIP1 がアクティブで、NS2 および NS3 上の VIP1 VIP はバックアップです。特定の状況下では、トラフィックが NS2 上の VIP1 に到達することがあります。NS2 で共有が有効になっている場合、このトラフィックはドロップされずに処理されます。

アクティブ/アクティブモードの設定

October 7, 2021

アクティブ-アクティブモードで展開する Citrix ADC アプライアンスごとに、仮想 MAC を追加し、その仮想 MAC を VIP にバインドする必要があります。特定の VIP の仮想 MAC は、各アプライアンスで同じである必要があります。たとえば、アプライアンスで VIP 10.102.29.5 を作成する場合、各 Citrix ADC で仮想ルーター ID（VRID）を作成し、各 Citrix ADC で VIP 10.102.29.5 にバインドする必要があります。仮想 MAC を VIP にバインドすると、アプライアンスはその VIP にバインドされている各 VLAN に VRRP アドバタイズメントを送信します。仮想 MAC は、同じ Citrix ADC 上で構成された異なる VIP によって共有できます。

IPv4 アクティブ/アクティブモードの設定

アクティブ-アクティブ構成に含める各 Citrix ADC アプライアンスで、次のタスクを実行します。

- 仮想 **MAC** アドレスを追加します。VRID を追加して、仮想 MAC アドレスを追加します。また、プライオリティを指定し、この VRID アドレスでプリエンプションと共有を有効または無効にすることもできます。
- **VIP** アドレスを追加し、仮想 **MAC** の **VRID** を関連付けます。VIP アドレスを追加し、VRID パラメータを新しく作成した VRID に設定します。VRID の属性（プライオリティやプリエンプションなど）は、この VIP ア

ドレスにバインドされます。
注：他のすべての Citrix ADC アプライアンスには、同じ VIP アドレスを追加する必要があります。

CLI を使用して仮想 MAC アドレスを追加するには

コマンドプロンプトで入力します。

- **add vrID** <id> [-**priority** <positive_integer>] [-**preemption** (**ENABLED**|**DISABLED**)][-**sharing** (**ENABLED**|**DISABLED**)] [-**tracking** <tracking>]
- **show vrid**

CLI を使用して VIP アドレスを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns ip** <IPv4Address> -type VIP -vrid <value>
- **show ns ip**

GUI を使用して仮想 MAC を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC**] タブで、新しい仮想 MAC を追加するか、既存の仮想 MAC を編集します。
2. 次のパラメーターを設定します。
   - 仮想ルータ ID
   - 優先度
   - 追跡
   - プリエンプション
   - 共有

GUI を使用して VIP アドレスを設定し、VRID をそのアドレスに関連付けるには、次の手順を実行します。

1. **[**システム**] > [**ネットワーク**] > [IP]** に移動し、**[**IPV4s**]** タブで **VIP** タイプの **IP** アドレスを追加します。
2. IP アドレスを追加するときに、[**Virtual Router Id**] ドロップダウンボックスから仮想ルータ ID を選択します。

設定例**:**

以下の構成例は、Citrix ADC アプライアンス NS1 および NS2 を IPv4 アクティブ-アクティブモードで展開するためのものです。VIP アドレス 203.0.113.10 は、NS1 と NS2 の両方で構成され、アプライアンスごとに異なるプライオリティ値が使用されます。各アプライアンスでは、この VIP アドレスは仮想 MAC アドレスにバインドされます。NS2 の優先度 (200) が NS1 (100) よりも高いため、203.0.113.10 は NS2 のマスターです。

```
1    Settings on NS1
2
3    > add vrid 10 -Priority 100 -Preemption Enabled -sharing Enabled
4
5    Done
```



```
6
7       > add ns ip 203.0.113.10  -type VIP  -vrid 10
8
9       Done
10
11      Settings on NS2
12
13      > add vrid 10  -Priority 200  -Preemption Enabled  -sharing Enabled
14
15      Done
16
17      > add ns ip 203.0.113.10  -type VIP  -vrid 10
18
19      Done
20  <!--NeedCopy-->
```

IPv6 アクティブ/アクティブモードの設定

アクティブ-アクティブ構成に含める各 Citrix ADC アプライアンスで、次のタスクを実行します。

- 仮想 **MAC6** アドレスを追加します。VRID6 を追加して、仮想 MAC6 アドレスを追加します。また、プライオリティを指定し、この VRID6 アドレスでプリエンプションと共有を有効または無効にすることもできます。
- **VIP6** アドレスを追加します。VIP6 アドレスを追加します。VRID6 パラメータを、新しく作成された仮想 MAC6 の VRID6 に設定します。仮想 MAC6 の属性（プライオリティやプリエンプションなど）は、この VIP6 アドレスにバインドされます。
  注：他のすべての Citrix ADC アプライアンスには、同じ VIP6 アドレスを追加する必要があります。

CLI を使用して仮想 MAC6 アドレスを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add vrID6** <id> [-**priority** <positive_integer>] [-**preemption** (**ENABLED** | **DISABLED**)] [-**sharing** (**ENABLED** | **DISABLED**)]
- **show vrID6**

CLI を使用して VIP6 アドレスを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns ip6** <IPv6Address> -**type** VIP -**vrid** <value>
- **show ns ip6**

GUI を使用して仮想 MAC6 を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC6**] タブで、新しい仮想 MAC6 を追加するか、既存の **VMAC6** を編集します。
2. 次のパラメーターを設定します。

- 仮想ルータ ID
- 優先度
- プリエンプション
- 共有

GUI を使用して VIP6 アドレスを設定し、VRID をそのアドレスに関連付けるには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**IP**] に移動し、[**IPV6s**] タブで VIP タイプの IPv6 アドレスを追加します。
2. VIP6 アドレスを追加するときに、[仮想ルータ **ID**] ドロップダウンボックスから VRID6 を選択します。

設定例**:**

以下の構成例は、Citrix ADC アプライアンスの NS1 および NS2 を IPv6 アクティブ-アクティブモードで展開するためのものです。VIP6 アドレス 2001:db8:: 5001 は、NS1 と NS2 の両方で構成され、アプライアンスごとに異なるプライオリティ値が設定されます。各アプライアンスでは、この VIP6 アドレスは仮想 MAC6 アドレスにバインドされます。NS2 の優先度 (200) が NS1 (100) よりも高いため、NS2 のマスターは 2001: db8:: 5001 は NS2 のマスターです。

```
    Settings on NS1
    > add vrid6 10 -Priority 100 -Preemption Enable -sharing Enable

    Done
    > add ns ip6 2001:db8::5001 -type VIP -vrid6 10

    Done
    Settings on NS2
    > add vrid6 10 -Priority 200 -Preemption Enable -sharing Enable

    Done
    > add ns ip6 2001:db8::5001 -type VIP -vrid6 10

    Done
<!--NeedCopy-->
```

マスターへの送信の設定

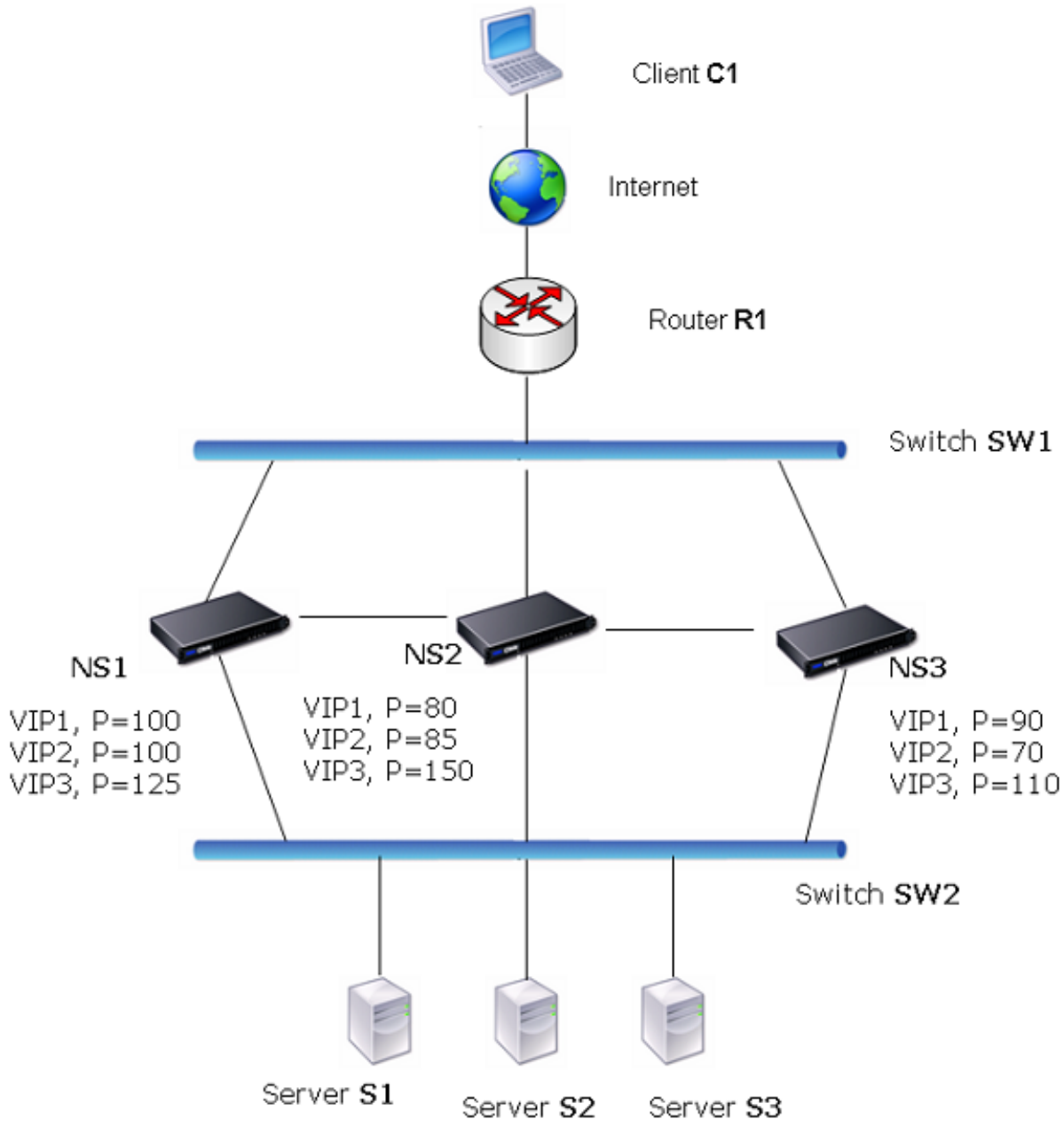
October 7, 2021

通常、VIP を宛先とするトラフィックは、VIP がアクティブである Citrix ADC アプライアンスに到達します。これは、VIP とそのアプライアンス上の仮想 MAC を含む ARP 要求がアップストリームルーターに到達するためです。ただし、VIP サブネットのアップストリームルーターで設定されたスタティックルートや、このルートをブロックするト

ポロジなど、VIP がバックアップ状態にある Citrix ADC アプライアンスにトラフィックが届くことがあります。VIP がアクティブなアプライアンスにデータパケットを転送する場合は、send to master オプションを有効にする必要があります。この動作はノード単位の設定で、デフォルトでは無効になっています。

たとえば、次の図では、VIP1 が NS1、NS2、および NS3 で構成され、NS1 でアクティブになっています。特定の状況下では、VIP1（NS1 でアクティブ）のトラフィックが NS3 の VIP1 に到達することがあります。マスターに送信オプションが NS3 で有効になっている場合、NS3 は NS1 のルートエントリを使用して NS2 経由でトラフィックを NS1 に転送します。

図 1: マスターへの送信オプションが有効になっているアクティブ/アクティブ設定



CLI を使用してマスターへの送信を有効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
set vrIDParam -sendToMaster (ENABLED          DISABLED)
```

例:


```
> set vrIDParam -sendToMaster ENABLED
 Done
<!--NeedCopy-->
```

GUI を使用してマスターへの送信を有効にするには、次の手順を実行します。

1. [システム] > [ネットワーク] に移動し、[設定] グループで [仮想ルーターパラメータ] をクリックします。
2. 「マスターに送信」オプションを選択します。

VRRP 通信インターバルの設定

October 7, 2021

アクティブ/アクティブ展開では、すべての Citrix ADC ノードは仮想ルーター冗長プロトコル（VRRP）を使用して、マスター VIP アドレスと VRRP アドバタイズメントパケット（hello メッセージ）内の対応する優先順位を定期的にアドバタイズします。

VRRP では、次の通信間隔が使用されます。

- **Hello** インターバル。マスター VIP アドレスのノードがピアノードに送信する VRRP hello メッセージ間の間隔。
- **Dead** インターバル。マスター VIP アドレスのノードから VRRP hello メッセージが受信されない場合、バックアップ VIP アドレスのノードがマスター VIP アドレスの状態を DOWN と見なすまでの時間。dead インターバルの後、バックアップ VIP アドレスが引き継ぎ、マスター VIP アドレスになります。

これらの間隔は、目的の値に変更できます。これらの通信間隔はどちらも、そのノード内のすべての VIP アドレスのノード単位の設定です。

CLI を使用して VRRP 通信間隔を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set vrIDParam** [-**helloInterval** <msecs>] [-**deadInterval** <secs>]
- **sh vrIDParam**

例**:**

```
> set vrIDParam -helloInterval 500 -deadInterval 2
 Done
<!--NeedCopy-->
```

GUI を使用して VRRP 通信間隔を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] に移動し、[設定] グループで [仮想ルーターパラメータ] をクリックします。

2. [仮想ルータパラメータの構成] で、[**Hello Interval**] および [**Dead Interval**] パラメータを設定します。
3. [**OK**] をクリックします。

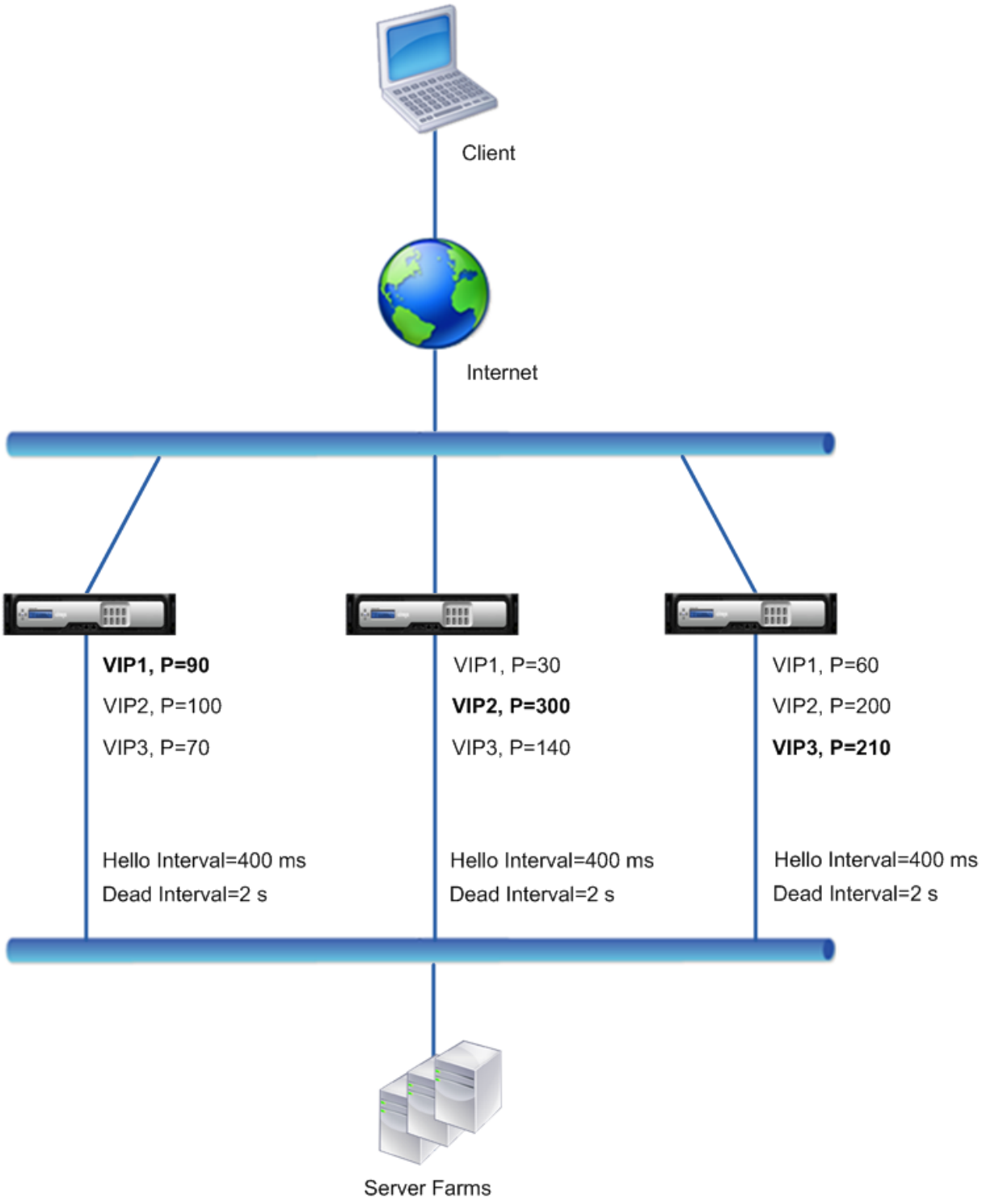
例 **1**: 同じ **VRRP Dead** インターバルを持つノード

Citrix ADC NS1、NS2、NS3 で構成されるアクティブ/アクティブ展開を考えてみましょう。仮想 IP アドレス VIP1、VIP2、VIP3 は、これらの各 ADC 上で構成されています。優先順位により、VIP1 は NS1 上でアクティブ、VIP2 は NS2 上でアクティブ、VIP3 は NS3 上でアクティブになります。

次の表に示すように、dead インターバルは 3 つのノードすべてで同じ値 (2 秒) に設定されます。ノードの VRRP 通信間隔（hello インターバルと dead インターバル）は、ノードに設定されているすべての VRID に適用され、ノード上の VRID に関連付けられたすべての VIP アドレスに適用されます。

各ノードで、そのノード上でアクティブ（マスター）の VIP アドレスは hello インターバルを使用し、dead インターバルは、そのノード上で非アクティブ（バックアップ）の VIP アドレスによって使用されます。3 つのノードすべての VIP アドレスのプリエンプションは無効です。

次の表は、この例で使用される設定の一覧です。VRRP 間隔の例 1 の設定。



実行フローは次のとおりです。

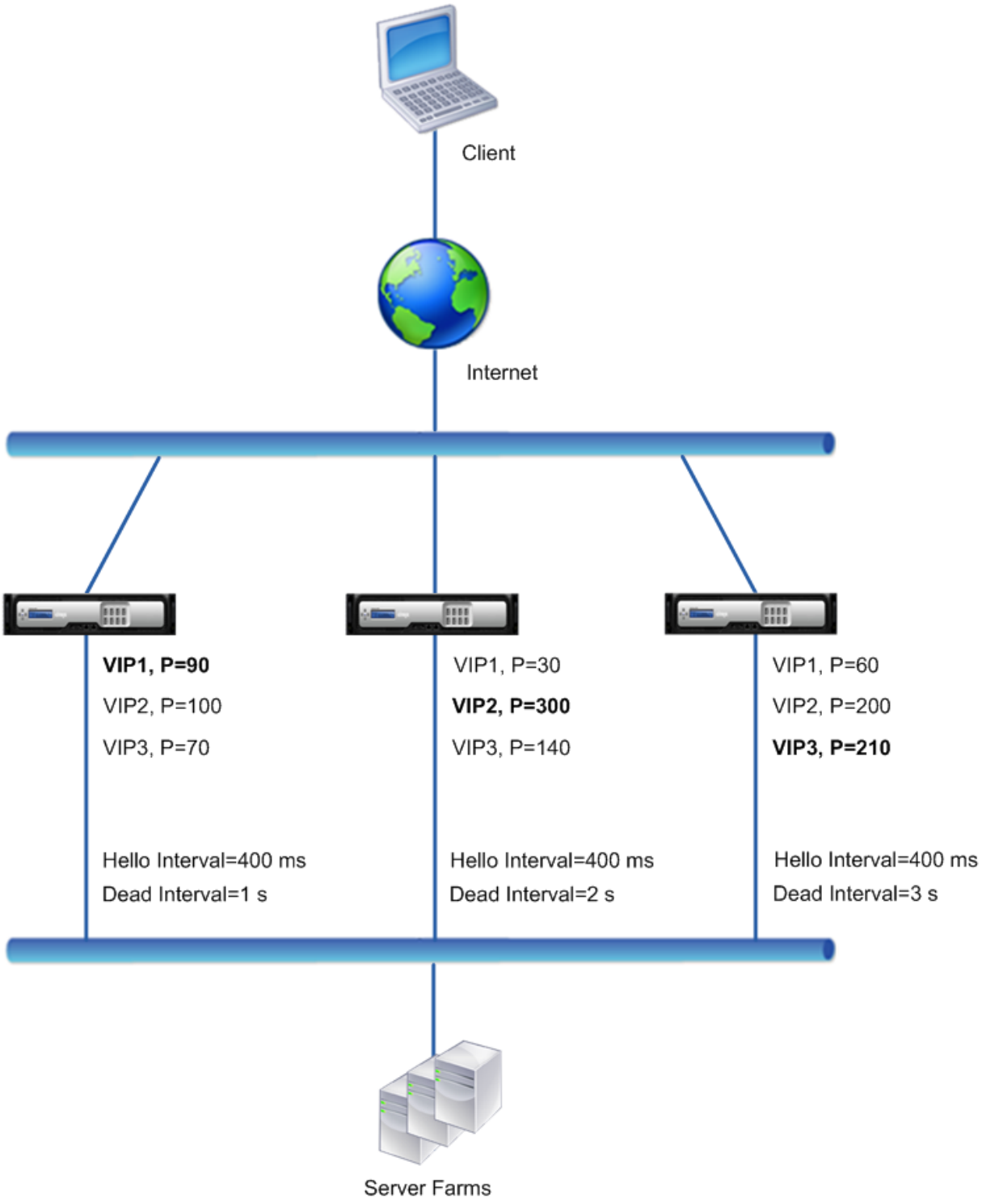
1. NS1 では、VIP1 がアクティブ（マスター）であるため、NS1 は、400 ミリ秒に設定された hello インターバルで VIP1 アドレスについて NS2 および NS3 に hello メッセージを送信します。同様に、NS2 は VIP2 の hello メッセージを送信し、NS3 は VIP3 の hello メッセージを送信します。

2. NS1 では、設定された dead インターバルは、NS1 上で非アクティブ（バックアップ）であるため、VIP2 および VIP3 に適用されます。同様に、NS2 では、設定された dead インターバルが VIP1 および VIP3 に適用され、NS3 では、設定された dead インターバルが VIP1 および VIP2 に適用されます。
3. NS1 がダウンした場合、NS2 と NS3 は NS1 から hello メッセージを 2 秒間受信しない場合（dead インターバル）、NS1 がダウンしていると見なします。NS3 上の VIP1 が引き継がれ、アクティブ（マスター）になります。これは、VRID の優先度（60）が NS2（30）の VIP1 よりも高いためです。

例 **2**: **VRRP dead** インターバルが異なるノード

例 1 で説明した配置と似ていますが、各ノード（NS1、NS2、および NS3）で dead インターバルが異なる VRRP 配置を検討してください。3 つのノードすべての VIP アドレスのプリエンプションは無効です。

次の表は、この例で使用される設定の一覧です。VRRP 間隔の例 2 の設定。



NS1 がダウンすると、実行フローは次のようになります。

1. NS2 は、NS1 から hello メッセージを 2 秒間受信しなかった後（NS2 の dead インターバル）、NS1 がダウンしていると見なします。
2. NS2 上の VIP1 が引き継がれ、アクティブ（マスター）になります。NS2 では、VIP1 に対する hello メッセ

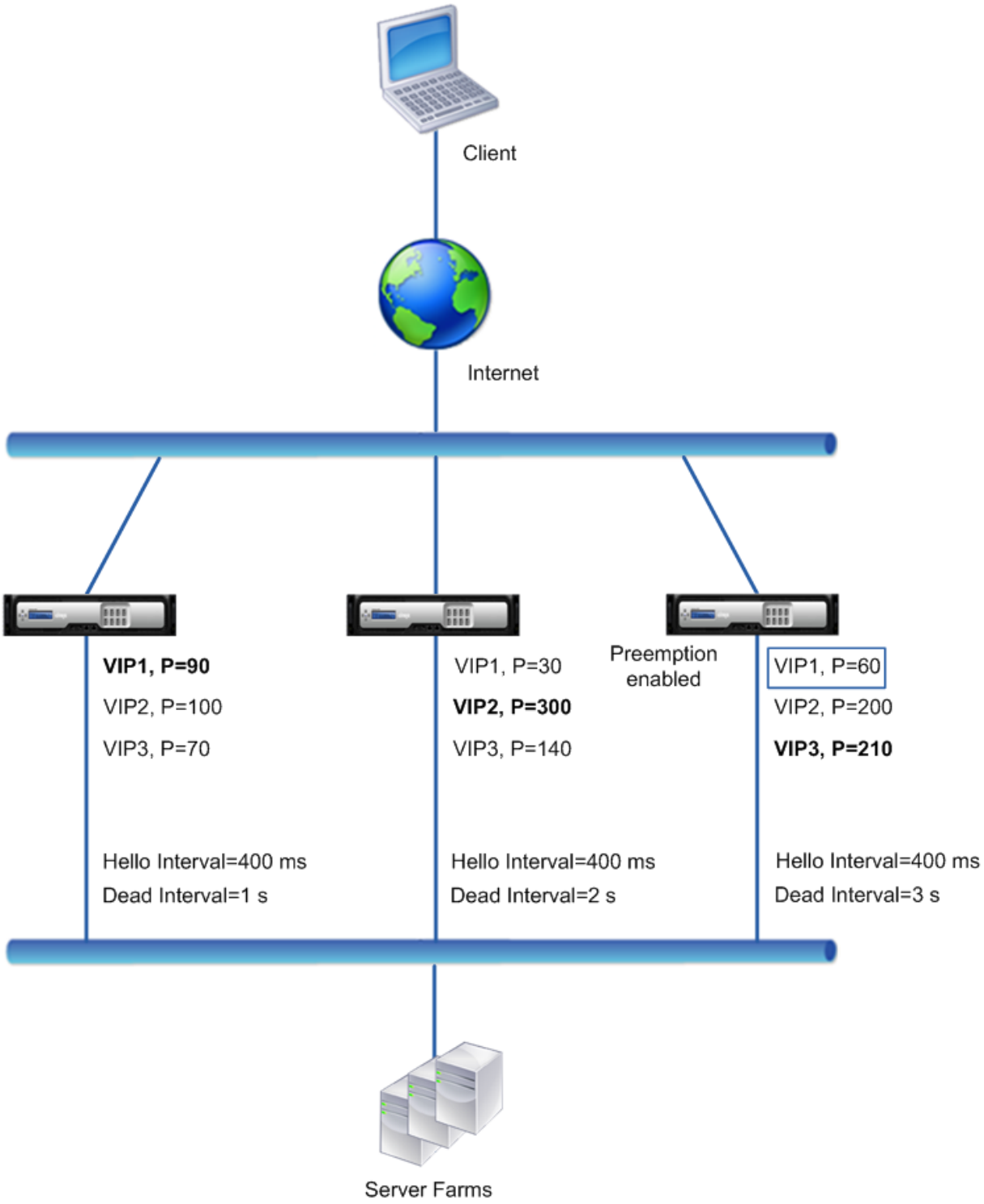
ージの送信が開始されます。

NS3 の VIP1 は NS2 の VIP1 よりも高い VRIP プライオリティ（60）ですが、NS3 の dead インターバル（3 秒、NS2 では 2 秒）が大きい場合、NS2 の VIP 1 がすでに実行されている前に NS3 の VIP1 を引き継ぐのを防ぎます。

例 **3: dead** インターバルが異なり、プリエンプションが有効なノード

例 1 で説明した配置と似ていますが、NS1、NS2、および NS3 の 3 つのノードで dead インターバルが異なり、NS3 の VIP1 アドレスに対してプリエンプションが有効になっている VRRP 配置を検討してください。

次の表は、この例で使用される設定の一覧です。VRRP 間隔の例 3 の設定。



NS1 がダウンすると、実行フローは次のようになります。

1. NS2 は、NS1 から hello メッセージを 2 秒間受信しなかった後、NS1 がダウンしていると見なします（NS2 によって設定された dead インターバル）。現時点では、停止間隔が 3 秒の NS3 では、NS1 がダウンしているとはみなされません。

2. NS2 上の VIP1 が引き継がれ、アクティブ（マスター）になります。NS2 では、VIP1 に対する hello メッセージの送信が開始されます。
3. NS2 から VIP1 用の hello メッセージを受信すると、NS3 は NS2 を VIP1 用にプリエンプトします。これは、NS3 の VIP1 に対してプリエンプトが有効であり、NS3 の VIP1 の VRID プライオリティ（60）が NS2 の VIP1 のプライオリティ（30）よりも大きいためです。
4. NS3 上の VIP1 が引き継ぎ、アクティブ（マスター）になります。NS3 が VIP1 の hello メッセージの送信を開始するようになりました。

インターフェイスステートに基づくヘルストラッキングの設定

October 7, 2021

現在のマスター VIP アドレスのノードが完全にダウンする前に、バックアップ VIP アドレスがマスター VIP として引き継がれるようにするには、ノードのインターフェイスの状態が変わったときに VIP アドレスのプライオリティを変更するようにノードを設定できます。たとえば、ノードは、インターフェイスの状態が DOWN に変わると VIP アドレスのプライオリティを下げ、インターフェイスの状態が UP に変わるとプライオリティを上げます。この機能は、各 VIP アドレスのノードごとの構成です。

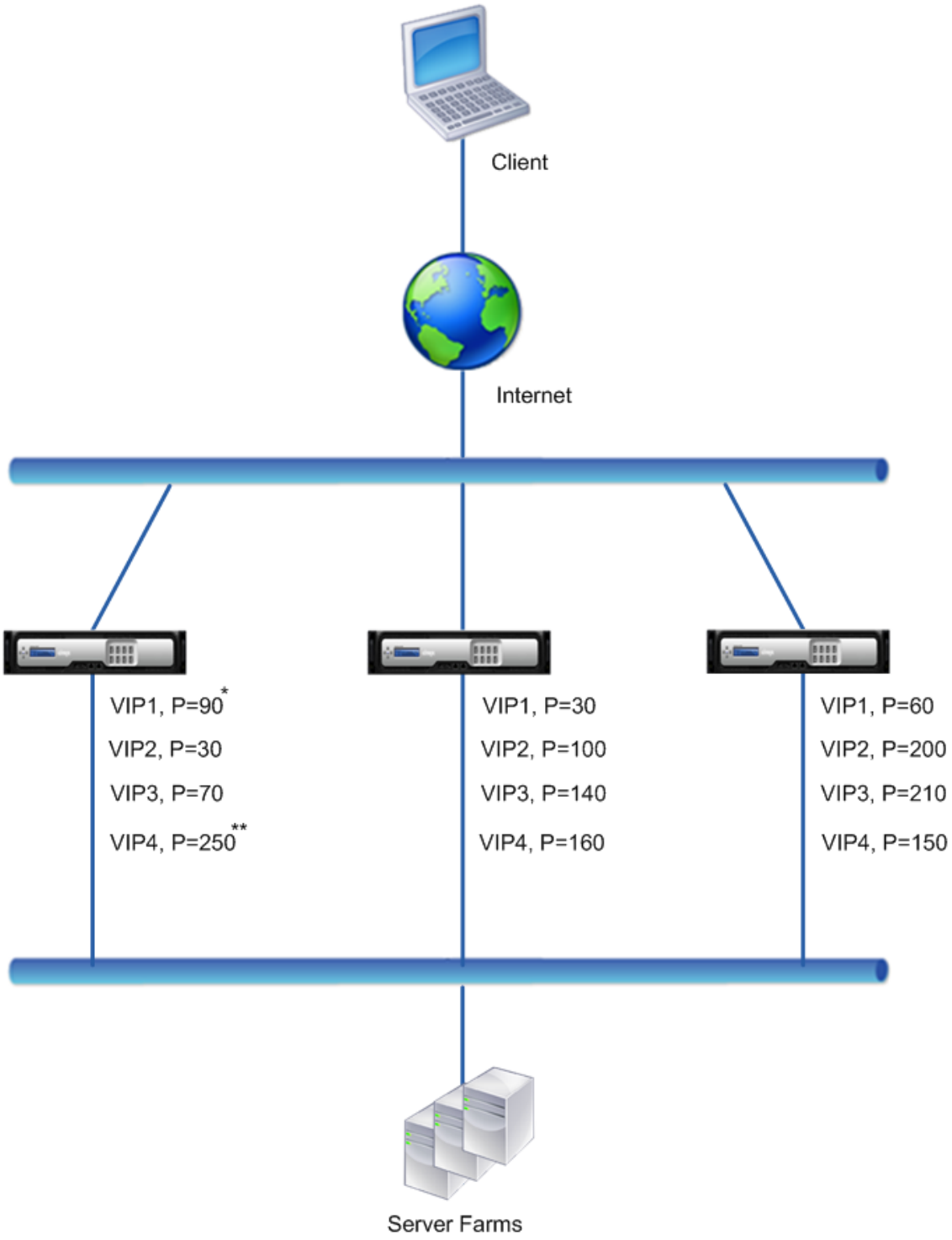
例

Citrix ADC NS1、NS2、NS3 で構成されるアクティブ/アクティブ展開を考えてみましょう。仮想 IP アドレス VIP1、VIP2、VIP3、および VIP4 は、これらの各 ADC 上で構成されます。優先順位により、VIP1 と VIP4 は NS1 上でアクティブになり、VIP2 は NS2 上でアクティブになり、VIP3 は NS3 上でアクティブになります。

NS1 が完全にダウンする前に、NS1 上のアクティブな VIP アドレスが NS2 または NS3 によって引き継がれるように、NS1 上の VIP1 および VIP4 アドレスに対してインターフェイスベースのヘルス・トラッキングが設定されます。VIP アドレスのインターフェイスベースのヘルストラッキングの設定には、必要なインターフェイスの関連付け、VIP アドレスの関連付けられた VRID の低プライオリティ（trackifNumPriority）パラメータの設定が含まれます。たとえば、NS1 では、インターフェイス 1/2、1/3、および 1/5 が VIP1 の VRID に関連付けられ、低優先度は 20 に設定されます。

3 つのノードすべてで、これらの VIP アドレスに対してプリエンプションが有効になります。

次の表に、この例で使用される設定を示します。ヘルストラッキングの設定例。



* Packet Interfaces = 1/2, 1/3, 1/5
Reduced Priority = 20

** Packet Interfaces = 1/5, 1/7
Reduced Priority = 55

NS1 上の複数のインターフェイスがダウンした場合、NS1 で実行フローは次のようになります。

1. インターフェイス 1/3 がダウンすると、インターフェイス 1/3 が VIP1 に関連付けられているため、アドレス VIP1 のプライオリティが 20（VIP1 の減少プライオリティ値）減少します。
   - VIP1 の有効優先度 = (現在の優先度-低優先度) = (90-20) = 70
2. 同様に、インターフェイス 1/5 がダウンすると、アドレス VIP1 のプライオリティがさらに低下します。
   - VIP1 の有効優先度 = (現在の優先度-低優先度) = (70-20) = 50
3. この時点で、NS1 での VIP1 の有効プライオリティは NS3 での VIP1 のプライオリティより小さくなります。NS3 は NS1 を VIP1 用にプリエンプトします。NS3 上の VIP1 が引き継ぎ、アクティブ（マスター）になります。
4. また、インターフェイス 1/5 も VIP4 に関連付けられているため、VIP4 のプライオリティは VIP4 の減少プライオリティ値（55）によって減少します。
   - VIP4 の有効プライオリティ =（250-55）= 195
5. インターフェイス 1/7 がダウンすると、VIP4 のプライオリティがさらに低下します。
   - VIP4 の有効優先度 = (現在の優先度-低優先度) = (195-55) = 145
6. この時点で、NS1 上の VIP4 の実効プライオリティは NS2 上の VIP4 のプライオリティよりも低くなります。NS2 は NS1 を VIP4 用にプリエンプトします。NS3 上の VIP4 が引き継ぎ、アクティブ（マスター）になります。この構成により、NS1 が完全にダウンする前に、4 つの VIP アドレスのいずれもアクティブになりません。

IPv4 アクティブ/アクティブモードの設定手順

VIP アドレスのノードでこの機能を設定するには、Reduced Priority（trackifNumPriority）パラメータを設定し、VIP アドレスのプライオリティを変更するためにステートを追跡するインターフェイスを関連付けます。関連付けられたインターフェイスのいずれかの状態が DOWN または UP に変化すると、ノードは VIP アドレスのプライオリティを、設定済みの Reduced Priority（trackifNumPriority）値だけ減少または増加させます。

CLI を使用して低プライオリティを設定し、インターフェイスを仮想ルータ ID にバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set vrID** <id> [-**trackifNumPriority** <positive_integer>]
- **bind vrID** <id> -**trackifNum** <interface_name>
- **show vrID** <id>

例**:**

```
1      > set vrID 125 -trackifNumPriority 10
2      Done
3
4      > bind vrID 125 -trackifNum 1/4 1/5
5      Done
6  <!--NeedCopy-->
```


GUI を使用して低プライオリティを設定し、インターフェイスを仮想ルータ ID にバインドするには、次の手順を実行します。

1. システムに移動 > 通信網 > **VMAC**。
2. [**VMAC**] タブで、仮想ルータ ID を選択し、[**Edit**] をクリックします。
3. [仮想 **MAC** の設定] で、[優先度の低下] パラメータを設定します。
4. **VRID** オプションで **[**追跡されるインターフェイス] を選択し、[**Associate Interfaces**] で仮想ルータ ID にインターフェイスを追加します。

IPv6 アクティブ/アクティブモードの設定手順

VIP6 アドレスのノードでこの機能を設定するには、Reduced Priority（trackifNumPriority）パラメータを設定し、VIP6 アドレスのプライオリティを変更するためにステートを追跡するインターフェイスを関連付けます。関連付けられたインターフェイスのいずれかの状態が DOWN または UP に変更されると、ノードは、設定された優先度の低下（trackifNumPriority）値によって、VIP6 アドレスの優先順位を下げたり増やしたりします。

CLI を使用して VIP アドレスのプライオリティを自動的に変更するには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しい仮想 MAC6 を追加する場合:
  - **vrID6** を追加 < id> [-**trackifNumPriority** <positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>
- 既存の仮想 MAC6 を再設定する場合:
  - **set vrID6** <id> [-**trackifNumPriority** <positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>

例**:**

```
1      > set vrID6 130 -trackifNumPriority 10
2      Done
3
4      > bind vrID6 130 -trackifNum 1/4 1/5
5      Done
6  <!--NeedCopy-->
```


優先の遅延

October 7, 2021

デフォルトでは、バックアップ VIP アドレスは、マスター VIP アドレスよりもプライオリティが高くなった直後にマスター VIP アドレスをプリエンプトします。バックアップ VIP アドレスを設定する場合、プリエンプションを遅延させる時間を指定できます。プリエンプション遅延時間は、各バックアップ VIP アドレスのノード単位の設定です。

バックアップ VIP のプリエンプション遅延設定は、次の条件では適用されません。

- マスター VIP のノードがダウンします。この場合、バックアップ VIP のノードに Dead インターバルが設定された後、バックアップ VIP がマスター VIP として引き継がれます。
- マスター VIP のプライオリティは 0 に設定されます。バックアップ VIP のノードで設定された Dead インターバルの後、バックアップ VIP がマスター VIP として引き継がれます。

例：プリエンプションの遅延

Citrix ADC アプライアンス NS1 と NS2 で構成されるアクティブ/アクティブ展開を考えてみましょう。仮想 IP アドレス VIP1 は、これらのアプライアンスごとに設定されます。VIP1 は優先順位があるため、NS2 ではマスターです。この 2 つのノードで VIP1 にプリエンプションが有効になり、プリエンプション遅延時間が設定されます。

次の表に、この例で使用される設定を示します。

エンティティとパラメータ	NS1 での設定	NS2 での設定
VIP1 (参照目的のみ)	**IP** アドレス**:** 192.0.1.10, **VRID:** 10, 優先度**:** 100, プリエンプション**:** 有効, プリエンプション遅延時間**:** 1000 秒	**IP** アドレス： 192.0.1.10、**VRID**: **10** 、優先度： 200、プリエンプション：有効、プリエンプション遅延時間： 2000 秒
Dead インターバル	1 秒	2 秒

この設定で発生する可能性のあるプリエンプト動作の例を次に示します。

- NS1 の VIP1 のプライオリティが NS2 の VIP1 よりも高い値（たとえば、210）に設定されている場合、NS1 の VIP1 は、設定されたプリエンプション遅延時間（1000 秒）後にマスターとして引き継ぎます。
- 次の VRRP 設定を持つ 3 番目のノード NS3 がこのデプロイメントに追加されると、設定されたプリエンプション遅延時間（3000 秒）の後、NS3 の VIP1 がマスターになります。
  - VIP1
    - VRID: 30
    - IP アドレス:
    - 優先度 = 300
    - プリエンプション遅延時間 = 3000 秒

- NS2 がダウンすると、NS1 の VIP1 が 1 秒後にマスターとして引き継がれます（NS1 で Dead インターバルが設定されます）。この場合、NS1 上の VIP1 のプリエンプション遅延時間は適用されません。
- NS2 が停止し、NS1 が再起動すると、NS1 が起動した後、NS1 上の VIP1 がマスター 1 秒になります（NS1 では Dead インターバルが設定されます）。この場合、NS1 上の VIP1 のプリエンプション遅延時間は適用されません。
- NS2 の VIP1 のプライオリティが 0 に設定されている場合、VIP1 はスタンバイモードになります。NS1 の VIP1 は、1 秒後にマスターとして引き継がれます（NS1 で Dead インターバルを設定）。この場合、NS1 上の VIP1 のプリエンプション遅延時間は適用されません。

IPv4 アクティブ/アクティブモードの遅延プリエンプションの設定

VIP アドレスのプリエンプション遅延時間を設定するには、関連付けられた仮想 MAC アドレスのプリエンプション遅延タイマーパラメータを設定します。このパラメータは、アドレスを追加するときに設定することも、既存の仮想 MAC アドレスを変更することもできます。

CLI を使用してプリエンプション遅延時間を設定するには、次の手順を実行します。

- 仮想 MAC の追加中にプリエンプト遅延時間を設定するには、コマンドプロンプトで次のように入力します。
  - **add vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**
- 仮想 MAC の変更中にプリエンプト遅延時間を設定するには、コマンドプロンプトで次のように入力します。
  - **set vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**

GUI を使用してプリエンプション遅延時間を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**VMAC**] に移動します。
2. [VMAC] タブで、[**VMAC**] タブをクリックします。新しい仮想 MAC を追加するとき、または既存の仮想 MAC を編集するときに、プリエンプション遅延タイマーパラメータを設定します。

設定例:

次の構成では、「例: プリエンプションの遅延」の表に示す設定を使用します。

```
Settings on NS1

> set vrid param  -deadInterval 1

Done

> add ns ip 192.0.1.10 255.255.255.255  -type VIP

Done

```



```
11      > add vrid 10  - Priority 100  - Preemption Enable  -
           preemptiondelaytimer 1000
12
13      Done
14
15      > bind ns ip 192.0.1.10 255.255.255.255  - vrid 10
16
17      Done
18
19      Settings on NS2
20
21      > set vrid param  - deadInterval 2
22
23      Done
24
25      > add ns ip 192.0.1.10 255.255.255.255  - type VIP
26
27      Done
28
29      > add vrid 20  - Priority 200  - Preemption Enable  -
           preemptiondelaytimer 2000
30
31      Done
32
33      > set ns ip 192.0.1.10 255.255.255.255  - vrid 10
34
35      Done
36  <!--NeedCopy-->
```

IPv6 アクティブ/アクティブモードの遅延プリエンプションの設定

VIP6 アドレスのプリエンプション遅延時間を設定するには、関連付けられた仮想 MAC6 アドレスのプリエンプション遅延タイマーパラメータを設定します。このパラメータは、仮想 MAC6 アドレスを追加するときに設定することも、既存の仮想 MAC6 アドレスを変更することもできます。

CLI を使用してプリエンプション遅延時間を設定するには、次の手順を実行します。

- 仮想 MAC6 の追加中にプリエンプト遅延時間を設定するには、コマンドプロンプトで次のように入力します。
  - **vrID6** を追加 < id>**-preemptiondelaytimer** < 秒 >
  - **show vrID6**
- 仮想 MAC6 の変更中にプリエンプト遅延時間を設定するには、コマンドプロンプトで次のように入力します。
  - **set vrID6** <id> -**preemptiondelaytimer** <secs>

- **show vrID6**

GUI を使用してプリエンプション遅延時間を設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**VMAC**] に移動します。
2. [**VMAC6**] タブで、仮想 MAC6 アドレスを追加する場合、または既存の仮想 MAC6 アドレスを編集する場合は、プリエンプション遅延タイマーパラメータを設定します。

VIP アドレスのバックアップ状態の維持

October 7, 2021

VIP アドレスは常にバックアップ状態に留まるように強制できます。この操作は、VRRP 配置のメンテナンスまたはテストに役立ちます。

VIP アドレスが強制的にバックアップステートを維持する場合、その VIP アドレスは VRRP ステート遷移に関与しません。また、他のすべてのノードがダウンしてもマスターになることはできません。

VIP アドレスを強制的にバックアップ状態に維持するには、関連付けられた仮想 MAC アドレスのプライオリティを 0 に設定します。ノードのメンテナンスプロセス中にノードの VIP アドレスがトラフィックを処理しないようにするには、すべての優先度をゼロに設定します。

アドレスの追加または変更時に、仮想 MAC アドレスのプライオリティを設定できます。

CLI を使用して VIP アドレスを強制的にバックアップステートのままにするには、次の手順を実行します。

- 仮想 MAC の追加時に優先順位を設定するには、コマンドプロンプトで次のように入力します。
  - **add vrID** <id> -**priority** 0
  - **show vrID**
- 仮想 MAC の変更中に優先順位を設定するには、コマンドプロンプトで次のように入力します。
  - **set vrID** <id> -**priority** 0
  - **show vrID**

GUI を使用して VIP アドレスを強制的にバックアップ状態に維持するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**VMAC**] に移動します。
2. [**VMAC**] タブで、新しい仮想 MAC の追加または既存の仮想 MAC の編集中に、**Priority** パラメータをゼロに設定します。

ネットワーク・ビジュアライザ

October 7, 2021

ネットワークビジュアライザーは、Citrix ADC アプライアンスのすべてのインターフェイス、チャネル、VLAN、IP アドレス、および VLAN へのバインディングをグラフィカルに表示します。有効になっているインターフェイスまたはチャネルには、黒いラベルが付いています。無効のインターフェイスまたはチャネルには、赤色のラベルが付いています。

このアプライアンスのネットワーク接続全体像は、ネットワーク設計における欠陥の検出やネットワークの最適化に役立ちます。また、新しい管理者がアプライアンスのネットワーク構成を簡単に理解するのに役立ちます。

ネットワークビジュアライザーを開くには:

[システム] > [ネットワーク] に移動します。[モニタ接続] で、[ネットワークビジュアライザー] をクリックします。

リンク層ディスカバリプロトコルの設定

October 7, 2021

Citrix ADC は、業界標準（IEEE 802.1AB）の Link Layer Discovery Protocol（LLDP）をサポートしています。LLDP は、Citrix ADC がその ID と機能を直接接続されたデバイスにアドバタイズし、これらの隣接デバイスの ID と機能を学習できるようにするレイヤー 2 プロトコルです。

> 注:
> Link Layer Discovery Protocol（LLDP）は、Citrix ADC MPX プラットフォームでのみサポートされます。

Citrix ADC は、LLDP を使用して、LLDP パケットデータユニット（LLDPU）と呼ばれる LLDP メッセージの形式で情報を送受信します。LLDPU は、タイプ、長さ、値（TLV）の情報要素のシーケンスです。各 TLV は、LLDPDU を送信するデバイスに関する特定のタイプの情報を保持します。Citrix ADC は、各 LLDPU で次の TLV を送信します。

- シャーシ ID
- ポート ID
- 存続時間の値
- システム名
- システムの説明
- ポートの説明
- システム機能
- 管理アドレス
- ポート VLAN ID
- リンクアグリゲーション

注: LLDP メッセージで送信する TLV を指定することはできません。

Citrix ADC インターフェースは、次の LLDP モードをサポートします。

- なし。インターフェイスは、直接接続されたデバイスとの間で LLDP メッセージを送受信しません。
- **TRANSMITTER**。インターフェイスは、直接接続されたデバイスに LLDP メッセージを送信しますが、直接接続されたデバイスから LLDP メッセージを受信しません。

- **RECEIVER**。インターフェイスは、直接接続されたデバイスから LLDP メッセージを受信しますが、直接接続されたデバイスには LLDP メッセージを送信しません。
- **TRANSCEIVER**。インターフェイスは、直接接続されたデバイスとの間で LLDP メッセージを送受信します。

インターフェイスの LLDP モードは、グローバルレベルとインターフェイスレベルで設定された LLDP モードによって異なります。次の表に、グローバルレベルとインターフェイスレベル設定の組み合わせによるモードを示します。インターフェイスレベルとグローバルレベル LLDP モード。

Citrix ADC によって送受信される LLDP メッセージに関連する次の点に注意してください。

- **LLDP** メッセージの送信。Citrix ADC は、TRANSMITTER または TRANSCEIVER LLDP モードのいずれかで動作しているインターフェイスから LLDPU を送信します。

  以下は、Citrix ADC のグローバル LLDP 送信パラメーターです。

  - タイマー。Citrix ADC が直接接続されたデバイスに送信する LLDPU 間の間隔（秒単位）。
  - ホールドタイム乗数。受信デバイスが LLDP 情報を破棄または削除する前にデータベースに保存する期間を計算するための乗数。持続時間は、[保持時間乗数] パラメータ値に [タイマー] パラメータ値を掛けた値として計算されます。

- **LLDP** メッセージの受信。Citrix ADC は、LLDPDU 情報を管理情報ベース（MIB）に保存します。格納されている LLDP 情報は、LLDPDU を受信したインターフェイスの ID で分類またはグループ化されます。Citrix ADC は、受信した LLDPU で指定された期間、この LLDP 情報を保持します。

  ADC がインターフェイス上で別の LLDPDU を受信した後、そのインターフェイスに保存されている LLDP 情報が破棄されると、ADC はそのインターフェイスに保存されている LLDP 情報を新しい LLDPDU の情報に置き換えます。

構成の手順

Citrix ADC アプライアンスで LLDP を構成するには、次の作業を行います。

1. グローバルレベルの **LLDP** パラメータを設定します。この作業では、LLDP タイマー、ホールドタイム乗数、LLDP モードなどのグローバル LLDP パラメータを設定します。
2. インターフェイスレベルの **LLDP** パラメータを設定します。この作業では、インターフェイスの LLDP モードを設定します。
3. （任意）ネイバーデバイス情報を表示します。Citrix ADC のすべてのインターフェイスで収集されたネイバーデバイス LLDP 情報を表示することも、指定したインターフェイスで収集された LLDP 情報だけを表示することもできます。インターフェイスを指定しない場合、すべてのインターフェイスの情報が表示されます。

次に、Citrix ADC で LLDP を構成するための前提条件を示します。

1. 標準 LLDP プロトコル（IEEE 802.1AB）を理解していることを確認します。
2. 目的の直接接続されたデバイスに LLDP が設定されていることを確認します。

CLI のプロシージャ

CLI を使用してグローバルレベルの LLDP パラメータを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- set lldp param [-[-holdtimeTxMult <positive_integer>][-timer <positive_integer>] [-Mode <Mode>]
- show lldp param

CLI を使用して LLDP のインターフェイスを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- set interface <id> -lldpmode <lldpmode>
- show interface <id>

CLI を使用してネイバーデバイス情報を表示するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- show lldp neighbors
- show lldp neighbors <ifnum>

GUI のプロシージャ

GUI を使用してグローバルレベルの LLDP パラメータを設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] に移動し、[LLDP パラメータの構成] をクリックします。
2. 次のパラメーターを設定します。
    - ホールド・タイマ乗数
    - タイマー
    - Mode

GUI を使用して LLDP のインターフェイスを設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [インターフェイス] に移動し、インターフェイスを開き、LLDP モードパラメータを設定します。

GUI を使用してネイバーデバイス情報を表示するには、次の手順を実行します。

[システム] > [ネットワーク] > [インターフェイス] に移動し、[アクション] リストで [LLDP ネイバーの表示] を選択します。

クラスタ設定における **LLDP** のサポート

クラスタセットアップでは、GUI または CLI に、クラスタ IP アドレス（CLIP）を介して GUI または CLI にアクセスすると、すべての、または特定のクラスタノードの LLDP ネイバー設定が表示されます。グローバルレベル LLDP モードに加えられた変更は、各クラスタノードのグローバルレベル LLDP モードに適用されます。

NS1、NS2、NSS3 という 3 つのノードで構成されるクラスタのセットアップの例を考えてみましょう。これらの各ノードは、ルータルータ 1 とルータ 2 の両方に接続されています。次の出力は、クラスタセットアップのクラスタ IP アドレス（CLIP）を介してアクセスされるクラスタ CLI で **show lldp neighbor-summary** 操作が実行された場合に表示されます。出力には、これらすべてのノードの LLDP ネイバー情報が表示されます。

```
> show lldp neighbor -summary

Node Id: 1
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     1/1/1      fe:c7:3b:13:bd:11    1/1         Router-1

2     1/1/2      12:68:7b:9e:4c:11    1/1         Router-2

Node Id: 2
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     2/1/1      fe:c7:3b:13:bd:12    1/2        Router-1

2     2/1/2      12:68:7b:9e:4c:12    1/2        Router-2

Node Id: 3
------------------------------------------------------------
    Interface    ChassisId            PortId     System name
------------------------------------------------------------

1     3/1/1       fe:c7:3b:13:bd:13    1/3        Router-1

2     3/1/2       12:68:7b:9e:4c:13    1/3        Router-2

Done
<!--NeedCopy-->
```

ジャンボフレーム

October 7, 2021

Citrix ADC アプライアンスは、最大 9216 バイトの IP データを含むジャンボフレームの送受信をサポートします。ジャンボフレームでは、標準の IP MTU サイズ（1500 バイト）を使用するよりも効率的に大きなファイルを送信す

ることができます。

Citrix ADC アプライアンスは、次の展開シナリオでジャンボフレームを使用できます。

- ジャンボで受信/ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それをジャンボフレームで送信します。
- 非ジャンボで受信/ジャンボで送信。アプライアンスがデータを通常のフレームで受信し、それをジャンボフレームで送信します。
- ジャンボで受信/非ジャンボで送信。アプライアンスがデータをジャンボフレームで受信し、それを通常のフレームで送信します。

Citrix ADC アプライアンスは、次のプロトコルの負荷分散構成でジャンボフレームをサポートします。

- TCP
- TCP を介した任意のプロトコル（HTTP など）
- SIP
- RADIUS

Citrix ADC アプライアンスでのジャンボフレームサポートの構成

October 7, 2021

Citrix ADC アプライアンスがジャンボフレームをサポートできるようにするには、インターフェイスまたは LA チャネル、および Citrix ADC アプライアンスがジャンボフレームをサポートするようにする VLAN で MTU を 1500 以上に設定します。

Citrix ADC アプライアンスのインターフェイス、LA チャネル、または VLAN の MTU を設定する前に考慮すべきポイント

1. LA チャネルを作成すると、チャネルに MTU が指定されていない場合、チャネルは最初にバインドされたインターフェイスの MTU を使用します。
2. チャネルの MTU は、すべてのバインドされたインターフェイスに伝播されます。
3. インターフェイスが MTU がインターフェイスの MTU と異なるチャネルにバインドされると、インターフェイスは非アクティブリストに移動します。
4. メンバーインターフェイスの MTU を変更すると、インターフェイスは非アクティブリストに移動します。
5. インターフェイスがチャネルからバインド解除されると、インターフェイスはチャネルの MTU 値を保持します。
6. インターフェイス、チャネル、または VLAN の MTU は、1500-9216 の範囲の値に設定できます。
7. デフォルト VLAN では MTU を設定できません。Citrix ADC アプライアンスは、デフォルトの VLAN との間でデータの送受信に使用するインターフェイスの MTU を使用します。

8. Citrix ADC アプライアンスの負荷分散構成で TCP ベースのトラフィックの場合、MSS はジャンボフレームをサポートするために各エンドポイントでそれに応じて設定されます。

    - クライアントと Citrix ADC アプライアンス上の負荷分散仮想サーバー間の接続では、Citrix ADC アプライアンス上の MSS が TCP プロファイルで設定され、負荷分散仮想サーバーにバインドされます。
    - Citrix ADC アプライアンスとサーバー間の接続の場合、NS1 の MSS は TCP プロファイルで設定され、このプロファイルは Citrix ADC アプライアンス上のサーバーを表すサービスにバインドされます。
    - デフォルトでは、TCP プロファイル nstcp_default_profile は、Citrix ADC アプライアンス上のすべての TCP ベースの負荷分散サーバーとサービスにバインドされます。
    - ジャンボフレームをサポートするには、TCP プロファイル nstcp_default_profile の MSS 値を変更するか、カスタム TCP プロファイルを作成し、それに応じて MSS を設定し、カスタム TCP プロファイルを目的のロードバランシング仮想サーバーおよびサービスにバインドします。
    - TCP プロファイルのデフォルトの MSS 値は 1460 です。

CLI のプロシージャ

CLI を使用してインターフェイスの MTU を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- set interface <id> -mtu <positive_integer>
- show interface <id>

例:

```
> set interface 10/1  - mtu 9000
Done
<!--NeedCopy-->
```

CLI を使用してチャネルの MTU を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- set channel <id> -mtu <positive_integer>
- show channel <id>

例:

```
> set channel LA/1  - mtu 9000
Done
<!--NeedCopy-->
```

CLI を使用して VLAN の MTU を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add vlan <id> -mtu <positive_integer>
- show vlan <id>

例**:**

```
> set vlan 20  - mtu 9000
Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してインターフェイスの MTU を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [インターフェイス] に移動し、インターフェイスを開き、[最大伝送ユニット] パラメータを設定します。

GUI を使用してチャネルの MTU を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [チャネル] に移動し、チャネルを開き、[最大伝送ユニット] パラメータを設定します。

GUI を使用して VLAN の MTU を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [VLAN] に移動し、VLAN を開き、[最大伝送ユニット] パラメータを設定します。

ユースケース **1 —** ジャンボからジャンボへのセットアップ

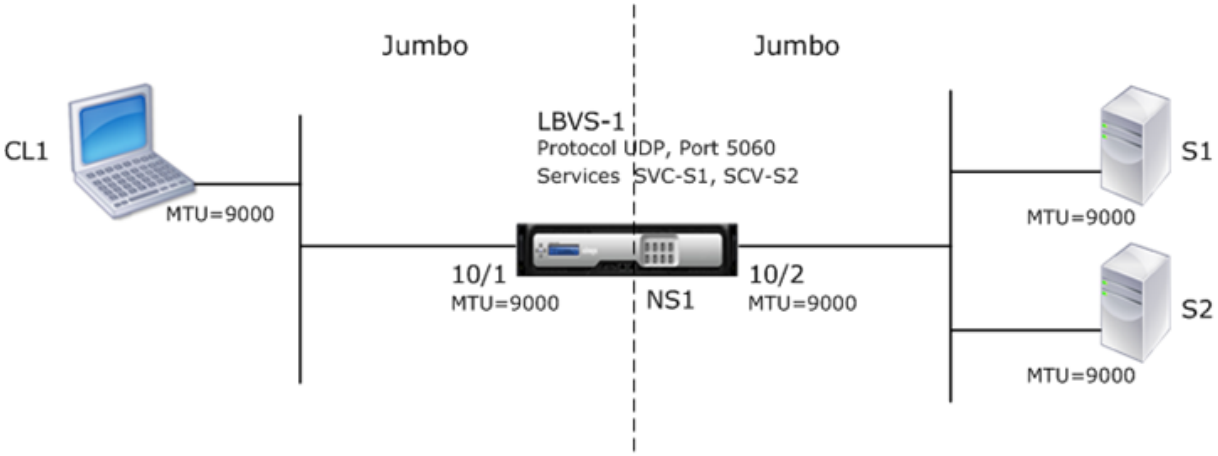
October 7, 2021

たとえば、Citrix ADC アプライアンス NS1 で構成された SIP 負荷分散仮想サーバー LBVS-1 を使用して、サーバー S1 と S2 間で SIP トラフィックの負荷分散を行うジャンボからジャンボへのセットアップの例を考えてみましょう。クライアント CL1 と NS1 の間の接続、および NS1 とサーバ間の接続は、ジャンボ・フレームをサポートします。

NS1 のインターフェイス 10/1 は、クライアント CL1 との間でトラフィックを受信または送信します。NS1 のインタフェース 10/2 は、サーバ S1 または S2 との間でトラフィックを受信または送信します。NS1 のインターフェイス 10/1 および 10/2 は、それぞれ VLAN 10 および VLAN 20 の一部です。

ジャンボフレームをサポートする場合、NS1 では、インターフェイス 10/1、10/2、VLAN 10、VLAN 20 に対して MTU が 9216 に設定されます。

このセットアップ例では、CL1、S1、S2 を含む他のすべてのネットワークデバイスも、ジャンボフレームをサポートするように設定されます。



次の表は、例で使用される設定の一覧です。

エンティティ	Name	詳細
クライアント CL1 の IP アドレス	-	192.0.2.10
サーバの IP アドレス	S1	198.51.100.19
	S2	198.51.100.20
NS1 上の SNIP アドレス		198.51.100.18
NS1 上のインターフェイスおよび VLAN に対して指定された MTU	10/1	9000
	10/2	9000
	VLAN 10	9000
	VLAN 20	9000
サーバを表す NS1 上のサービス	SVC-S1	**IP** アドレス**:** 198.51.100.19, プロトコル**:** SIP, ポート**:** 5060
	SVC-S2	**IP** アドレス**:** 198.51.100.20, プロトコル**:** SIP, ポート**:** 5060
VLAN 10 上の仮想サーバのロードバランシング	LBVS-1	**IP** アドレス**:** 203.0.113.15, プロトコル**:** SIP, ポート**:** 5060, バウンドサービス**:** SVC-S1, SVC-S2

次に、NS1 への CL1 要求のトラフィックフローを示します。

1. CL1 は、200 バイトの SIP 要求を作成して、NS1 の LBVS-1 に送信します。
2. CL1 は、IP フラグメント内の要求データを LBVS-1 に送信します。各 IP フラグメントのサイズは、CL1 がこれらのフラグメントを NS1 に送信するインターフェイスに設定された MTU（9000）以下になります。

- 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000
- 2 番目の IP フラグメントのサイズ = [IP ヘッダー + SIP データセグメント] = [20 + 8980] = 9000
- 最後の IP フラグメントのサイズ [IP ヘッダー + SIP データセグメント] = [20 + 2048] = 2068

3. NS1 は、インターフェイス 10/1 で要求 IP フラグメントを受信します。NS1 はこれらのフラグメントを受け入れます。これは、各フラグメントのサイズがインターフェイス 10/1 の MTU（9000）以下であるためです。
4. NS1 は、これらの IP フラグメントを再構成して、200 バイトの SIP 要求を形成します。NS1 はこの要求を処理します。
5. LBVS-1 の負荷分散アルゴリズムは、サーバー S1 を選択します。
6. NS1 は、要求データを IP フラグメントで S1 に送信します。各 IP フラグメントのサイズは、NS1 がこれらのフラグメントを S1 に送信するインターフェイス 10/2 の MTU（9000）と同じか小さくなります。IP パケットの送信元は NS1 の SNIP アドレスです。

- 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000
- 2 番目の IP フラグメントのサイズ [IP ヘッダー + SIP データセグメント] =[20 + 8980] = 9000
- 最後の IP フラグメントのサイズ [IP ヘッダー + SIP データセグメント] =[20 + 2048] = 2068

次に、この例の CL1 に対する S1 の応答のトラフィックフローを示します。

1. サーバ S1 は、30000 バイトの SIP 応答を作成して、NS1 の SNIP アドレスに送信します。
2. S1 は、IP フラグメントで応答データを NS1 の SNIP アドレスに送信します。各 IP フラグメントのサイズは、S1 がこれらのフラグメントを NS1 に送信するインターフェイスに設定された MTU（9000）以下になります。

- 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000
- 2 番目と 3 番目の IP フラグメントのサイズ [IP ヘッダー + SIP データセグメント] =[20 + 8980] = 9000
- 最後の IP フラグメントのサイズ [IP ヘッダー + SIP データセグメント] = [20 + 3068] = 3088

3. NS1 は、インターフェイス 10/2 で応答 IP フラグメントを受信します。NS1 はこれらのフラグメントを受け入れます。これは、各フラグメントのサイズがインターフェイス 10/2 の MTU（9000）以下であるためです。
4. NS1 は、これらの IP フラグメントを再構成して 30000 バイトの SIP 応答を形成します。NS1 はこの応答を処理します。
5. NS1 は、IP フラグメントで応答データを CL1 に送信します。各 IP フラグメントのサイズは、NS1 がこれらのフラグメントを CL1 に送信するインターフェイス 10/1 の MTU（9000）と同じか小さいかのいずれかです。IP フラグメントは、LBVS-1 の IP アドレスでソースされます。

- 最初の IP フラグメントのサイズ = [IP ヘッダー + UDP ヘッダー + SIP データセグメント] = [20 + 8 + 8972] = 9000

- 2 番目と 3 番目の IP フラグメントのサイズ [IP ヘッダー + SIP データセグメント] =[20 + 8980] = 9000
- 最後の IP フラグメントのサイズ [IP ヘッダー + SIP データセグメント] =[20 + 3068] = 3088

構成タスク

次の表に、Citrix ADC アプライアンスで必要な構成を作成するタスク、Citrix ADC コマンド、および例を示します。

タスク	Citrix ADC コマンド構文	例
ジャンボフレームをサポートするための目的のインターフェイスの MTU を設定します。	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1 -mtu 9000 set int 10/2 -mtu 9000
ジャンボフレームをサポートするために VLAN を作成し、目的の VLAN の MTU を設定します。	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 9000 add vlan 20 -mtu 9000
インターフェイスを VLAN にバインドする	bind vlan <id> -ifnum <interface_name>, show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
SNIP アドレスを追加する	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
SIP サーバを表すサービスの作成	add service <serviceName> <ip> SIP_UDP <port>, show service <name>	add service SVC-S1 198.51.100.19 SIP_UDP 5060 add service SVC-S2 198.51.100.20 SIP_UDP 5060
SIP ロードバランシング仮想サーバを作成し、そのサーバにサービスをバインドする	add lb vserver <name> SIP_UDP <ip> <port> bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 SIP_UDP 203.0.113.15 5060 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2
構成を保存します	save ns config, show ns config	

ユースケース **2** — ジャンボ以外のセットアップからジャンボへのセットアップ

October 7, 2021

Citrix ADC アプライアンス NS1 で構成された負荷分散仮想サーバー LBVS-1 を使用して、サーバー S1 と S2 間でトラフィックの負荷分散を行う定期的なジャンボセットアップの例を考えてみましょう。クライアント CL1 と NS1 の間の接続は通常のフレームをサポートし、NS1 とサーバ間の接続はジャンボ・フレームをサポートします。

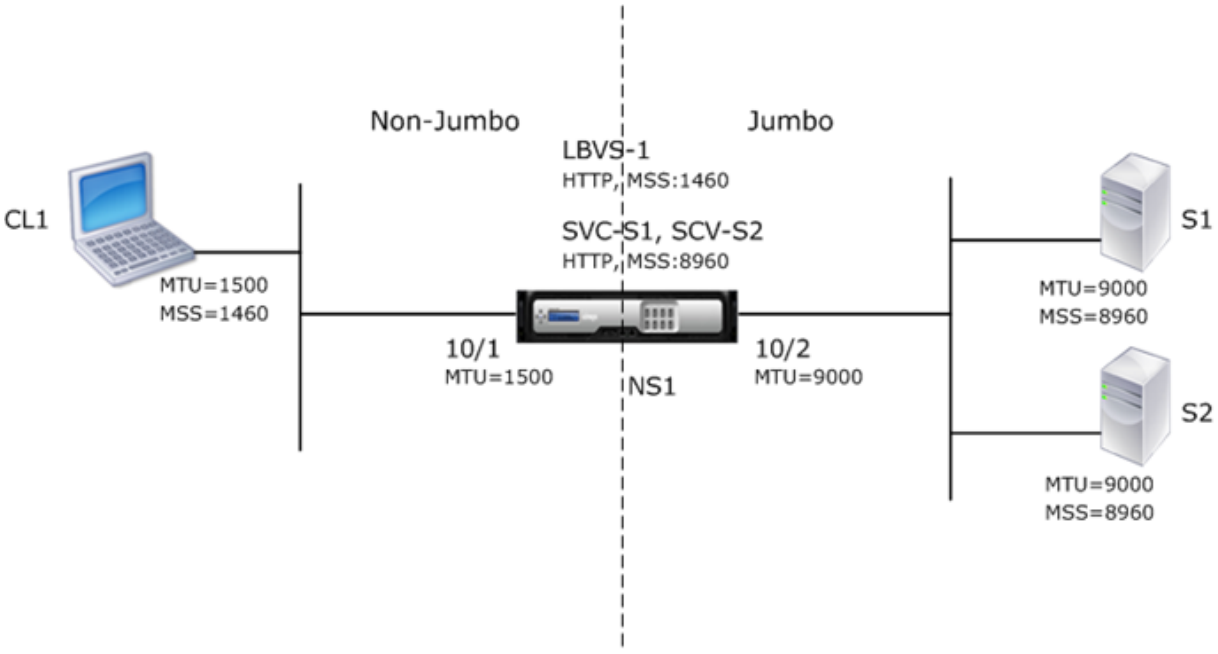
NS1 のインターフェイス 10/1 は、クライアント CL1 との間でトラフィックを受信または送信します。NS1 のインタフェース 10/2 は、サーバ S1 または S2 との間でトラフィックを受信または送信します。

NS1 のインターフェイス 10/1 および 10/2 は、それぞれ VLAN 10 および VLAN 20 の一部です。CL1 と NS1 の間の通常のフレームだけをサポートする場合、インターフェイス 10/1 と VLAN 10 の両方で MTU はデフォルト値の 1500 に設定されます。

NS1 とサーバ間のジャンボ・フレームをサポートするために、インターフェイス 10/2 と VLAN 20 の MTU は 9000 に設定されています。NS1 とサーバ間のサーバおよび他のすべてのネットワーク・デバイスも、ジャンボ・フレームをサポートするように構成されます。

HTTP トラフィックは TCP に基づいているため、MSS はジャンボフレームをサポートするために各エンドポイントでそれに応じて設定されます。

- NS1 と S1 または S2 の SNIP アドレス間の接続でジャンボフレームをサポートするために、NS1 の MSS は NS1 上の S1 および S2 を表すサービス（SVC-S1 および SVC-S1）にバインドされるカスタム TCP プロファイルでそれに応じて設定されます。
- CL1 と NS1 の仮想サーバー LBVS-1 間の接続で通常のフレームのみをサポートするには、デフォルトの TCP プロファイル nstcp_default_profile が使用されます。このプロファイルはデフォルトで LBVS-1 にバインドされ、MSS はデフォルト値の 1460 に設定されます。



次の表に、この例で使用される設定を示します。

エンティティ	Name	詳細
クライアント CL1 の IP アドレス		192.0.2.10
サーバの IP アドレス	S1	198.51.100.19
	S2	198.51.100.20
NS1 上の SNIP アドレス		198.51.100.18
NS1 上のインターフェイスおよび VLAN に対して指定された MTU	10/1	1500
	10/2	9000
	VLAN 10	1500
	VLAN 20	9000
デフォルトの TCP プロファイル	nstcp_default_profile	MSS:1460
カスタム TCP プロファイル	NS1-SERVERS-JUMBO	MSS: 8960
サーバを表す NS1 上のサービス	SVC-S1	IP address: 198.51.100.19, Protocol: HTTP, Port: 80, TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)
	SVC-S2	IP address: 198.51.100.20, Protocol: HTTP, Port: 80, TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)
VLAN 10 上の仮想サーバのロードバランシング	LBVS-1	IP address = 203.0.113.15, Protocol: HTTP, Port:80, Bound services: SVC-S1, SVC-S2, TCP Profile: nstcp_default_profile (MSS:1460)

次に、この例では、CL1 から S1 への要求のトラフィックフローを示します。

1. クライアント CL1 は、NS1 の仮想サーバー LBVS-1 に送信する 200 バイトの HTTP 要求を作成します。
2. CL1 は、NS1 の LBVS-1 への接続を開きます。CL1 と NS1 は、接続を確立しながら、それぞれの TCP MSS 値を交換します。
3. NS1 の MSS は HTTP 要求よりも大きいので、CL1 は 1 つの IP パケットで要求データを NS1 に送信します。

   要求パケットのサイズ = [IP ヘッダー + TCP ヘッダー + TCP 要求] = [20 + 20 + 200] = 240

4. NS1 は、インターフェイス 10/1 で要求パケットを受信し、パケット内の HTTP 要求データを処理します。
5. LBVS-1 のロード・バランシング・アルゴリズムはサーバ S1 を選択し、NS1 は SNIP アドレスの 1 つと S1 間の接続を開きます。NS1 と CL1 は、接続の確立中に、それぞれの TCP MSS 値を交換します。
6. S1 の MSS は HTTP 要求よりも大きいので、NS1 は 1 つの IP パケットで要求データを S1 に送信します。

   要求パケットのサイズ = [IP ヘッダー + TCP ヘッダー + [TCP 要求] = [20 + 20 + 200] = 240

次に、この例の CL1 に対する S1 の応答のトラフィックフローを示します。

1. サーバ S1 は、NS1 の SNIP アドレスに送信する 18,000 バイトの HTTP 応答を作成します。
2. S1 は、応答データを NS1 の MSS の倍数に分割し、これらのセグメントを IP パケットで NS1 に送信します。これらの IP パケットは、S1 の IP アドレスから送信され、NS1 の SNIP アドレスを宛先とします。
   - 最初の 2 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP セグメント =NS1 の MSS サイズ)] = [20 + 20 + 8960] = 9000
   - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー +（残りの TCP セグメント）] = [20 + 20 + 2080] = 2120
3. NS1 は、インターフェイス 10/2 で応答パケットを受信します。
4. NS1 は、これらの IP パケットから、すべての TCP セグメントをアセンブルして、18000 バイトの HTTP 応答データを形成します。NS1 はこの応答を処理します。
5. NS1 は、応答データを CL1 の MSS の倍数に分割し、これらのセグメントを IP パケットでインタフェース 10/1 から CL1 に送信します。これらの IP パケットは LBVS-1 の IP アドレスから発信され、CL1 の IP アドレスを宛先とします。
   - 最後のパケット以外のすべてのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP ペイロード =CL1 の MSS サイズ)] = [20 + 20 + 1460] = 1500
   - 最後のパケットのサイズ [IP ヘッダー + TCP ヘッダー +（残りの TCP セグメント）] = [20 + 20 + 480] = 520

構成タスク

次の表に、Citrix ADC アプライアンスで必要な構成を作成するタスク、Citrix ADC コマンド、および例を示します。

タスク	CLI の構文	例
ジャンボフレームをサポートするための目的のインターフェイスの MTU を設定します。	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1 -mtu 1500 set int 10/2 -mtu 9000
ジャンボフレームをサポートするために VLAN を作成し、目的の VLAN の MTU を設定します。	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 1500 add vlan 20 -mtu 9000

タスク	CLI の構文	例
インターフェイスを VLAN にバインドする	bind vlan <id> -ifnum <interface_name>, show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
SNIP アドレスを追加する	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
HTTP サーバーを表すサービスの作成	add service <serviceName> <ip> HTTP <port>, show service <name>	add service SVC-S1 198.51.100.19 http 80, add service SVC-S2 198.51.100.20 http 80
HTTP 負荷分散仮想サーバーを作成し、サービスをそれにバインドする	add lb vserver <name> HTTP <ip> <port>, bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 http 203.0.113.15 80, bind lb vserver LBVS-1 SVC-S1, bind lb vserver LBVS-1 SVC-S2
カスタム TCP プロファイルを作成し、ジャンボフレームをサポートするための MSS を設定します。	add tcpProfile <name> -mss <positive_integer>, show tcpProfile <name>	add tcpprofile NS1-SERVERS-JUMBO -mss 8960
カスタム TCP プロファイルを目的のサービスにバインドする	set service <Name> -tcpProfileName <string>, show service <name>	set service SVC-S1 -tcpProfileName NS1-SERVERS-JUMBO, set service SVC-S2 -tcpProfileName NS1-SERVERS-JUMBO
構成を保存します	save ns config, show ns config	

使用例 3：同じインターフェイスセットでのジャンボフローと非ジャンボフローの共存

October 7, 2021

たとえば、Citrix ADC アプライアンス NS1 上で負荷分散仮想サーバー LBVS-1 と LBVS-2 が構成されているとします。LBVS-1 はサーバー S1 および S2 間で HTTP トラフィックの負荷分散に使用され、LBVS-2 はサーバー S3 および S4 間でトラフィックの負荷分散に使用されます。

CL1 は VLAN 10、S1 と S2 は VLAN 20、CL2 は VLAN 30、S3 と S4 は VLAN 40 にあります。VLAN 10 および

VLAN 20 はジャンボフレームをサポートし、VLAN 30 および VLAN 40 は通常のフレームだけをサポートします。

つまり、CL1 と NS1 の間の接続、および NS1 とサーバ S1 または S2 間の接続は、ジャンボ・フレームをサポートします。CL2 と NS1 の間の接続、および NS1 とサーバ S3 または S4 間の接続は、通常のフレームのみをサポートします。

NS1 のインタフェース 10/1 は、クライアントとの間でトラフィックを受信または送信します。NS1 のインタフェース 10/2 は、サーバとの間でトラフィックを受信または送信します。

インターフェイス 10/1 は、タグ付きインターフェイスとして VLAN 10 と VLAN 30 の両方にバインドされ、インターフェイス 10/2 は VLAN 20 と VLAN 40 の両方にタグ付きインターフェイスとしてバインドされます。

ジャンボフレームをサポートする場合、インターフェイス 10/1 および 10/2 の MTU は 9216 に設定されます。

NS1 では、MTU は VLAN 10 で 9000 に設定され、ジャンボフレームをサポートするために VLAN 20 に設定され、MTU は VLAN 30 と VLAN 40 のデフォルト値 1500 に設定され、通常のフレームだけをサポートします。

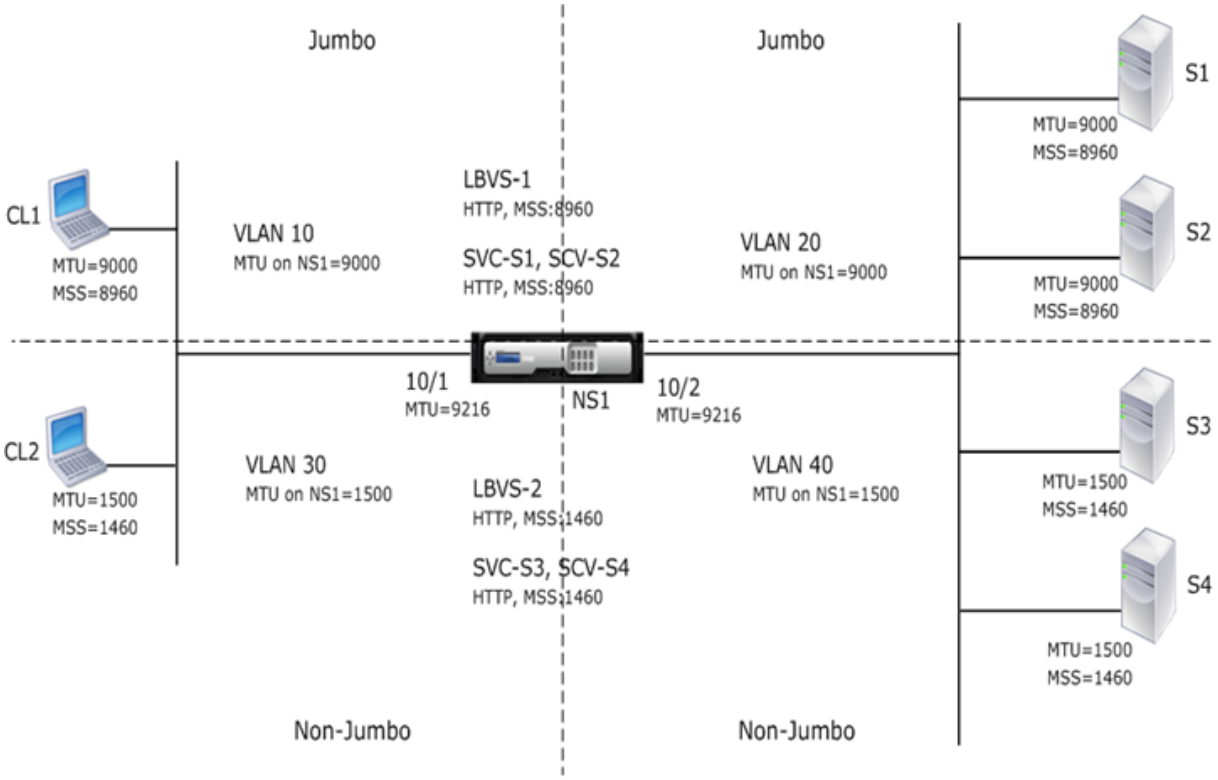
VLAN タグ付きパケットに対する Citrix ADC インターフェイスの有効な MTU は、インターフェイスの MTU または VLAN の MTU のいずれか小さい方になります。次に例を示します:

- インターフェイス 10/1 の MTU は 9216 です。VLAN 10 の MTU は 9000 です。インターフェイス 10/1 では、VLAN 10 タグ付きパケットの MTU は 9000 です。
- インターフェイス 10/2 の MTU は 9216 です。VLAN 20 の MTU は 9000 です。インターフェイス 10/2 では、VLAN 20 タグ付きパケットの MTU は 9000 です。
- インターフェイス 10/1 の MTU は 9216 です。VLAN 30 の MTU は 1500 です。インターフェイス 10/1 では、VLAN 30 タグ付きパケットの MTU は 1500 です。
- インターフェイス 10/2 の MTU は 9216 です。VLAN 40 の MTU は 1500 です。インターフェイス 10/2 では、VLAN 40 タグ付きパケットの MTU は 9000 です。

CL1、S1、S2、および CL1 と S1 または S2 の間にあるすべてのネットワークデバイスは、ジャンボフレーム用に設定されます。

HTTP トラフィックは TCP に基づいているため、MSS はジャンボフレームをサポートするために各エンドポイントでそれに応じて設定されます。

- CL1 と NS1 の仮想サーバー LBVS-1 の間の接続では、NS1 の MSS が TCP プロファイルに設定され、このプロファイルが LBVS-1 にバインドされます。
- NS1 と S1 の SNIP アドレス間の接続では、NS1 の MSS が TCP プロファイルに設定され、NS1 上の S1 を表すサービス（SVC-S1）にバインドされます。



次の表に、この例で使用される設定を示します。ジャンボフレームの使用例 3 の設定例。

次に、CL1 から S1 への要求のトラフィックフローを示します。

1. クライアント CL1 は、NS1 の仮想サーバー LBVS-1 に送信する 200 バイトの HTTP 要求を作成します。
2. CL1 は、NS1 の LBVS-1 への接続を開きます。CL1 と NS1 は、接続の確立中に TCP MSS 値を交換します。
3. NS1 の MSS 値は HTTP 要求よりも小さいので、CL1 は要求データを NS1 の MSS の倍数に分割し、VLAN 10 としてタグ付けされた IP パケットでこれらのセグメントを NS1 に送信します。
   - 最初の 2 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP セグメント =NS1 MSS)] = [20 + 20 + 8960] = 9000
   - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー +（残りの TCP セグメント）] = [20 + 20 + 2080] = 2120
4. NS1 は、これらのパケットをインタフェース 10/1 で受信します。NS1 では、これらのパケットのサイズが VLAN 10 タグ付きパケットのインターフェイス 10/1 の有効な MTU（9000）以下であるため、これらのパケットを受け入れます。
5. NS1 は IP パケットから、すべての TCP セグメントをアセンブルして、200 バイトの HTTP 要求を形成します。NS1 はこの要求を処理します。
6. LBVS-1 のロード・バランシング・アルゴリズムはサーバ S1 を選択し、NS1 は SNIP アドレスの 1 つと S1 間の接続を開きます。NS1 と CL1 は、接続の確立中に、それぞれの TCP MSS 値を交換します。
7. NS1 は、要求データを S1 の MSS の倍数に分割し、これらのセグメントを VLAN 20 とタグ付けされた IP パケットで S1 に送信します。
   - 最初の 2 つのパケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP ペイロード =S1 MSS)][20 + 20

+ 8960] = 9000
- 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー +（残りの TCP セグメント）] = [20 + 20 + 2080] = 2120

次に、CL1 に対する S1 の応答のトラフィックフローを示します。

1. サーバ S1 は、30000 バイトの HTTP 応答を作成して、NS1 の SNIP アドレスに送信します。
2. S1 は、応答データを NS1 の MSS の倍数に分割し、これらのセグメントを VLAN 20 とタグ付けされた IP パケットで NS1 に送信します。これらの IP パケットは、S1 の IP アドレスから送信され、NS1 の SNIP アドレスを宛先とします。
   - 最初の 3 パケットのサイズ = [IP ヘッダー + TCP ヘッダー + (TCP セグメント =NS1 の MSS サイズ)][20 + 20 + 8960] = 9000
   - 最後のパケットのサイズ [IP ヘッダー + TCP ヘッダー +（残りの TCP セグメント）] = [20 + 20 + 3120] = 3160
3. NS1 は、インターフェイス 10/2 で応答パケットを受信します。NS1 は、VLAN 20 タグ付きパケットのインターフェイス 10/2 の実効 MTU 値（9000）以下であるため、これらのパケットを受け入れます。
4. NS1 は、これらの IP パケットから、すべての TCP セグメントをアセンブルして 30000 バイトの HTTP 応答を形成します。NS1 はこの応答を処理します。
5. NS1 は、応答データを CL1 の MSS の倍数に分割し、これらのセグメントを VLAN 10 とタグ付けされた IP パケットでインターフェイス 10/1 から CL1 に送信します。これらの IP パケットは LBVS の IP アドレスから発信され、CL1 の IP アドレスを宛先とします。
   - 最初の 3 パケットのサイズ = [IP ヘッダー + TCP ヘッダー + [(TCP ペイロード =CL1 の MSS サイズ)][20 + 20 + 8960] = 9000
   - 最後のパケットのサイズ = [IP ヘッダー + TCP ヘッダー +（残りの TCP セグメント）] = [20 + 20 + 3120] = 3160

構成タスク

次の表に、Citrix ADC アプライアンスで必要な構成を作成するためのタスク、コマンド、および例を示します。ジャンボフレームのユースケース 3 構成タスク。

Microsoft Direct Access 展開における Citrix ADC サポート

October 7, 2021

Microsoft Direct Access は、リモートユーザーが社内のネットワークにシームレスかつ安全に接続できるようにするテクノロジです。VPN 接続を個別に確立する必要はありません。接続の開閉にユーザーの介入を必要とする VPN 接続とは異なり、Direct Access 対応クライアントは、クライアントがインターネットに接続するたびに企業の内部ネットワークに自動的に接続します。

Manage-Out は Microsoft Direct Access 機能の 1 つで、企業ネットワーク内の管理者は、ネットワーク外部の Direct Access クライアントに接続し、クライアントを管理できます (たとえば、サービスの更新のスケジュール設定やリモートサポートの提供など)。

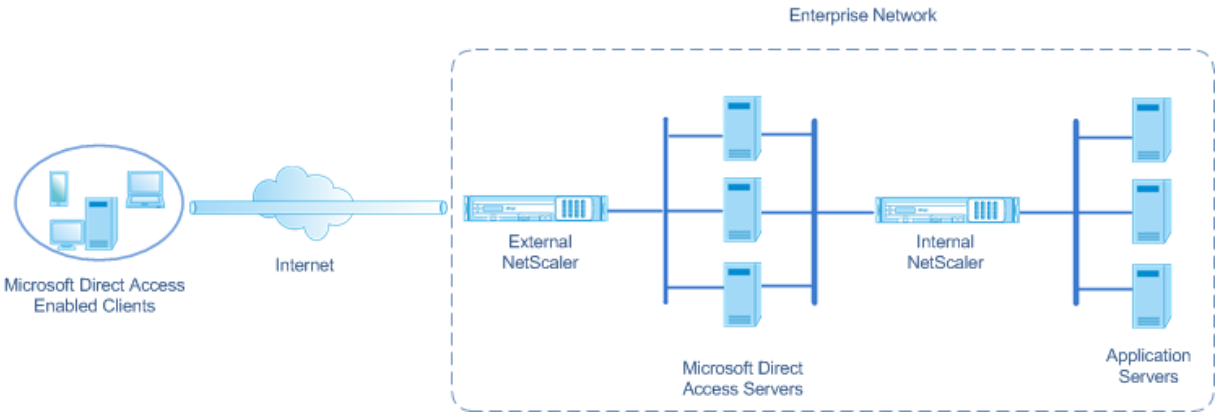
Direct Access 展開では、Citrix ADC アプライアンスは高可用性、拡張性、高パフォーマンス、およびセキュリティを提供します。Citrix ADC 負荷分散機能は、最適なサーバーを介してクライアントトラフィックを送信します。アプライアンスは、Manage-Out トラフィックを適切なパスで転送して、クライアントに到達することもできます。

アーキテクチャ

Microsoft Direct Access 展開のアーキテクチャは、Direct Access 対応のクライアント、Direct Access サーバー、アプリケーション・サーバー、および内部および外部の Citrix ADC アプライアンスで構成されています。クライアントは、Direct Access サーバーを経由してアプリケーションサーバーに接続します。外部 Citrix ADC アプライアンスは、クライアントトラフィックの Direct Access サーバーへの負荷分散を行います。内部 Citrix ADC アプライアンスは、クライアントトラフィックを Direct Access サーバーから宛先アプリケーションサーバーに転送します。Direct Access は、IPv4 ネットワーク経由でクライアントの IPv6 トラフィックをトンネリングするために使用されます。外部 Citrix ADC アプライアンス上の IPv4 負荷分散仮想サーバーは、クライアントのトンネリングされたトラフィックを Direct Access サーバーの 1 つに負荷分散します。Direct Access サーバーは、受信したクライアントの IPv4 パケットから IPv6 パケットを抽出し、内部の Citrix ADC アプライアンスを介して送信先アプリケーションサーバーに送信します。内部 Citrix ADC アプライアンスには、Direct Access サーバからのクライアントのトラフィックに関するレイヤ 2 およびレイヤ 3 接続情報を格納するための、ソースルートキャッシュオプションが有効になっている転送セッション規則があります。Citrix ADC アプライアンスは、ソースルートキャッシュテーブルと呼ばれるテーブルに次のレイヤー 2 およびレイヤー 3 情報を保存します。

- 受信パケットの送信元 IP アドレス
- パケットを送信した Direct Access サーバーの MAC アドレス
- パケットを受信した Citrix ADC アプライアンスの VLAN ID
- パケットを受信した Citrix ADC アプライアンスのインターフェイス ID

Citrix ADC アプライアンスは、クライアントに到達するためのトンネリング情報があるため、同じ Direct Access サーバーに応答を転送するためにソースルートキャッシュテーブル内の情報を使用します。また、内部アプライアンスは、ソースルートキャッシュテーブルを使用して、アプリケーションサーバの Manage-out トラフィックを適切な Direct Access サーバに転送し、特定のクライアントに到達します。



Microsoft Direct Access 展開での内部 **Citrix ADC** アプライアンスの構成

アプリケーションサーバーの応答トラフィックと管理アウトトラフィックを適切な Direct Access Gateway に転送するように内部 Citrix ADC アプライアンスを構成するには、転送セッションルールを構成します。各ルールで、ソースキャッシュパラメータを ENABLED に設定します。

CLI を使用して転送セッション規則を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add forwardingSession** <name> ((<network> [<netmask>]) | **-acl6name** <string> | **-aclname** <string>) **-sourceroutecache** (**ENABLED** | **DISABLED**]
- **show forwardingsession** <name>

設定例:

次の例では、転送セッションルール MS-DA-FW-1 が内部 Citrix ADC アプライアンス上に作成されます。転送セッションでは、送信元 IPv6 プレフィクス 2001:DB8:: /96 と一致するダイレクトアクセスサーバからの着信 IPv6 パケットのレイヤ 2 およびレイヤ 3 情報が格納されます。

```
1 > add forwardingSession MS-DA-FW-1 2001:DB8::/96 -sourceroutecache -
    ENABLED
2  Done
```

ソースルートキャッシュテーブルの表示

ソースルートキャッシュテーブルを表示して、ダイレクトアクセスサーバとアプリケーションサーバ間の不要な接続を監視または検出できます。

CLI を使用してソースルートキャッシュテーブルを表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **show sourceroutecachetable**

例**:**

```
1 > show sourceroutecachetable
2 SOURCEIP            MAC                 VLAN    INTERFACE
3 2001:DB8:5001:10    56:53:24:3d:02:eb    30        1/2
4 2001:DB8:5003:30    60:54:35:3e:04:bd    60        1/3
5 Done
```

送信元ルートキャッシュテーブルの消去

Citrix ADC アプライアンスのソースルートキャッシュテーブルからすべてのエントリをクリアできます。

CLI を使用してソースルートキャッシュテーブルを消去するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **flush ns sourceroutecachetable**

アクセス制御リスト

October 7, 2021

Access Control List（ACL; アクセスコントロールリスト）は IP トラフィックをフィルタリングし、ネットワークを不正アクセスから保護します。ACL は、Citrix ADC がアクセスを許可するかどうかを決定するために評価する一連の条件です。たとえば、財務部門は、人事部門やドキュメントなどの他の部門からリソースにアクセスすることを許可したくないと思っており、それらの部門はデータへのアクセスを制限したいと考えています。

Citrix ADC はデータパケットを受信すると、データパケット内の情報と ACL で指定された条件を比較し、アクセスを許可または拒否します。組織の管理者は、次の処理モードで機能するように ACL を設定できます。

- ALLOW：パケットを処理します。
- BRIDGE：パケットを処理せずに宛先にブリッジします。パケットは、レイヤ 2 およびレイヤ 3 転送によって直接送信されます。
- DENY：パケットをドロップします。

ACL ルールは、Citrix ADC の第 1 レベルの防御です。

Citrix ADC では、次のタイプの ACL がサポートされています。

- シンプルな **ACL** は、送信元 IP アドレス、およびオプションでプロトコル、宛先ポート、またはトラフィックドメインに基づいてパケットをフィルタリングします。ACL で指定された特性を持つパケットはすべてドロップされます。

- 拡張 **ACL** は、送信元 IP アドレス、送信元ポート、アクション、プロトコルなどのさまざまなパラメータに基づいてデータパケットをフィルタリングします。拡張 ACL は、Citrix ADC がパケットを処理したり、パケットをブリッジしたり、パケットをドロップしたりするためにパケットが満たす必要がある条件を定義します。

命名法

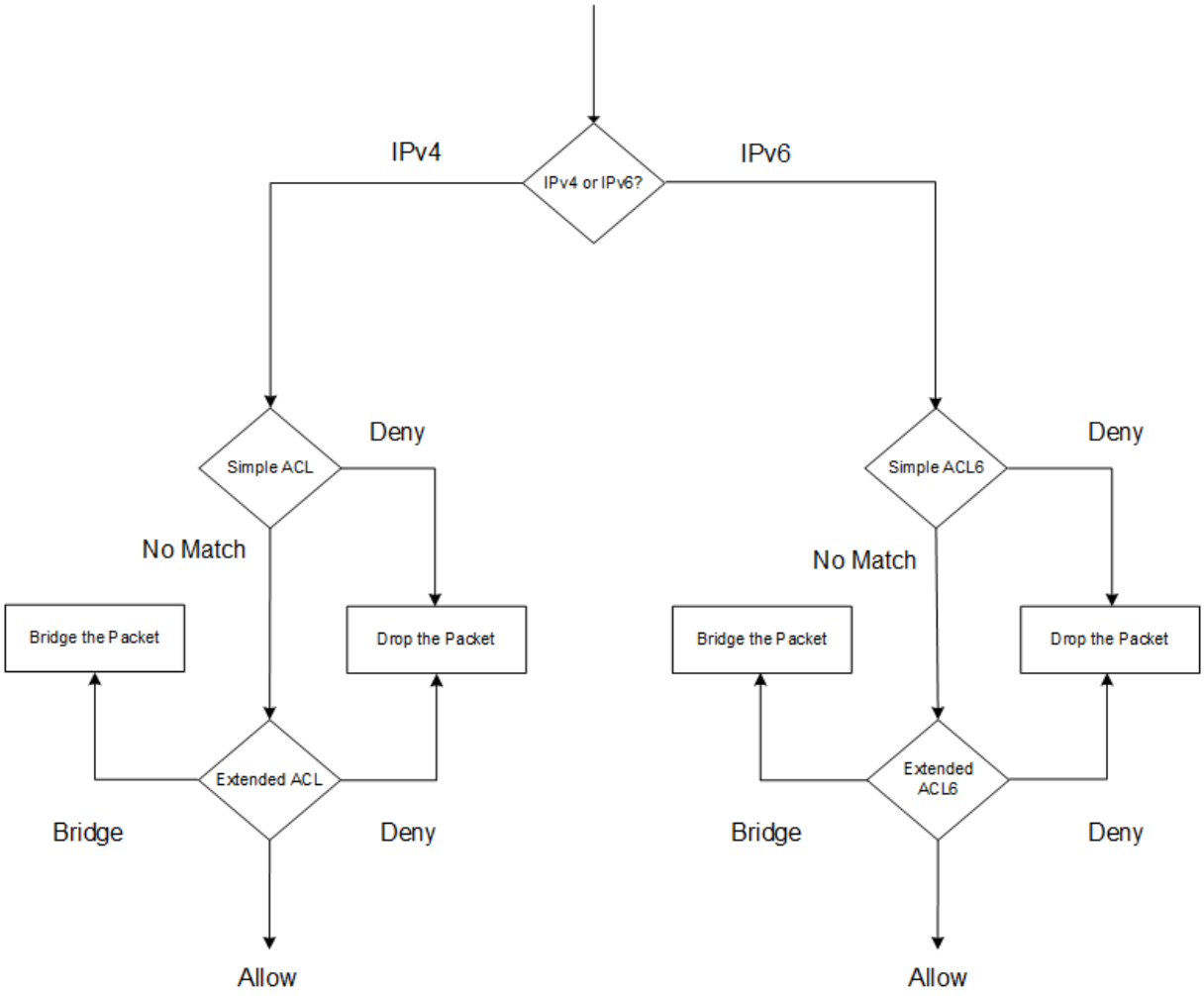
Citrix ADC ユーザーインターフェイスでは、単純 ACL と拡張 ACL という用語は、IPv4 パケットを処理する ACL を指します。IPv6 パケットを処理する ACL は、単純な ACL6 および拡張 ACL6 と呼ばれます。両方のタイプについて説明する場合、このドキュメントでは、両方を単純 ACL または拡張 ACL と呼びます。

ACL の優先順位

簡易 ACL と拡張 ACL の両方が設定されている場合、最初に着信パケットが簡易 ACL と比較されます。

Citrix ADC は、まず受信パケットが IPv4 パケットか IPv6 パケットかを判断し、パケットの特性を単純 ACL か単純 ACL6 と比較します。一致するものが見つかると、パケットはドロップされます。一致するものが見つからない場合、パケットは拡張 ACL または拡張 ACL6 と比較されます。この比較の結果が一致する場合、パケットは ACL で指定されたとおりに処理されます。パケットは、ブリッジング、ドロップ、または許可できます。一致するものが見つからない場合、パケットは許可されます。

図 1: 簡易 ACL および拡張 ACL フローシーケンス



シンプル **ACL** とシンプル **ACL 6**

October 7, 2021

単純な ACL または単純な ACL6 では、パラメータはほとんど使用されないため、IP パケットをドロップするようにだけ設定できます。パケットは、送信元 IP アドレス、およびオプションでプロトコル、宛先ポート、またはトラフィックドメインに基づいてドロップできます。

単純な ACL または単純な ACL6 を作成する場合、存続可能時間（TTL）を秒単位で指定できます。この時間が経過すると、ACL が期限切れになります。TTL を持つ ACL は、構成を保存するときに保存されません。簡易 ACL と簡易 ACL6 を表示して設定を確認したり、統計情報を表示したりできます。

簡易 **ACL** および簡易 **ACL6** の設定

Citrix ADC で単純 ACL または単純 ACL6 を構成するには、次の作業が含まれます。

- 単純な **ACL** または単純な **ACL6** を作成します。送信元 IP アドレス、およびオプションでプロトコル、宛先ポート、またはトラフィックドメインに基づいてパケットをドロップ（拒否）するシンプル ACL またはシンプルな ACL 6 を作成します。
- 単純な **ACL** または単純な **ACL6** を削除します。これらの ACL は、一度作成すると変更できません。単純な ACL または単純な ACL6 を変更する必要がある場合は、それを削除して作成する必要があります。

CLI のプロシージャ

CLI を使用して単純な ACL を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 -  ns simpleacl <aclname> DENY -srcIP <ip_addr> [-destPort <port> -
     protocol ( TCP | UDP )] [-TTL <positive_integer>]
2 -  show ns simpleacl [<aclname>]
3 <!--NeedCopy-->
```

例**:**

```
1 > add simpleacl rule1 DENY -srcIP 10.102.29.5 -TTL 600
2 Done
3 <!--NeedCopy-->
```

CLI を使用して単純な ACL6 を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 -  add ns simpleacl6 <aclname> DENY - srcIPv6 <ipv6_addr|null> [-
     destPort <port> -protocol ( TCP | UDP )] [-TTL <positive_integer>]
2 -  show ns simpleacl6 [<aclname>]
3 <!--NeedCopy-->
```

例**:**

```
1 >  add ns simpleacl6 rule1 DENY  - srcIPv6 3ffe:192:168:215::82 -
     destPort 80 -Protocol TCP -TTL 9000
2  Done
3 <!--NeedCopy-->
```


CLI を使用して単純な ACL を 1 つ削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **rm ns simpleacl** <aclname>
- **show ns simpleacl**

CLI を使用して単純な ACL6 を 1 つ削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **rm ns simpleacl6**<aclname>
- **show ns simpleacl6**

CLI を使用して単純な ACL をすべて削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **clear ns simpleacl**
- **show ns simpleacl**

CLI を使用して単純な ACL6 をすべて削除するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **clear ns simpleacl6**
- **show ns simpleacl6**

GUI のプロシージャ

GUI を使用して単純な ACL を作成するには、次の手順を実行します。

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL**] タブで、新しいシンプル ACL を追加します。

GUI を使用して単純な ACL6 を作成するには、次の手順を実行します。

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL6**] タブで、新しいシンプル ACL6 を追加します。

GUI を使用して単純な ACL を 1 つ削除するには、次の手順を実行します。

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL**] タブでシンプル ACL を削除します。

GUI を使用して単純な ACL6 を 1 つ削除するには、次の手順を実行します。

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL6**] タブで、単純な ACL6 を削除します。

GUI を使用して単純な ACL をすべて削除するには、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2. [簡易 **ACL**] タブの [操作] ボックスの一覧で、[クリア] をクリックします。

GUI を使用して単純な ACL6 をすべて削除するには、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2. [簡易 **ACL6s**] タブの [操作] ボックスの一覧で、[クリア] をクリックします。

簡易 **ACL** および簡易 **ACL6** 統計情報の表示

単純な ACL（または簡易 ACL6）統計情報を表示できます。これには、一致の数、ミス数、および設定された単純 ACL の数が含まれます。

次の表に、シンプル ACL およびシンプル ACL 6 について表示できる統計情報を示します。

統計	内容
ACL マッチ	ACL に一致するパケット
ACL misses	ACL に一致しないパケット
ACL count	設定された ACL の数

CLI のプロシージャ

CLI を使用して単純な ACL 統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat ns simpleacl**

例**:**

```
> stat ns simpleacl

SimpleACL Statistics

                                          Rate (/s)
   Total
SimpleACL hits                                  0
                       0
SimpleACL misses                                0
    51872
SimpleACLs count                               --
                       2
Done
<!--NeedCopy-->
```

CLI を使用して単純な ACL6 統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat ns simpleacl6**

GUI のプロシージャ

GUI を使用して単純な ACL 統計情報を表示するには、次の手順を実行します。

[システム] > [ネットワーク] > [**ACL**] に移動し、[シンプル **ACL**] タブで ACL を選択し、[統計] をクリックします。

GUI を使用して単純な ACL6 統計情報を表示するには、次の手順を実行します。

[システム**] > [**ネットワーク**] > [ACL**] に移動し、[シンプル **ACL6**] タブで単純な ACL6 を選択し、[統計] をクリックします。

確立された接続の終了

単純 ACL または単純 ACL6 の場合、Citrix ADC は、ACL で指定された条件に一致する新しい接続をブロックします。ACL が作成される前に確立された既存の接続に関連するパケットはブロックされません。既存の ACL と一致する以前に確立された接続を終了するには、CLI または GUI からフラッシュ操作を実行します。

フラッシュは、次の場合に役立ちます。

- ブラックリストに登録された IP アドレスのリストを受け取り、それらの IP アドレスが Citrix ADC にアクセスすることを完全にブロックしたい場合。この場合、単純な ACL または単純な ACL6 を作成して、これらの IP アドレスからの新しい接続をブロックし、それらのアドレスに関連付けられた既存の接続をフラッシュします。
- 特定のネットワークからの多数の接続を 1 つずつ終端する時間をとらずに終端したい。

はじめに

- フラッシュを実行すると、Citrix ADC は確立されているすべての接続を検索し、ADC で構成された単純な ACL のいずれかに指定された条件に一致する接続を終了します。
- 複数の単純な ACL を作成し、それらのいずれかに一致する既存の接続をフラッシュする場合は、まずすべての単純な ACL を作成し、フラッシュを 1 回だけ実行することで、パフォーマンスへの影響を最小限に抑えることができます。

CLI のプロシージャ

CLI を使用して、設定したシンプル ACL のいずれかに一致する確立済みの IPv4 接続をすべて終了するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **flush simpleacl -estSessions**

CLI を使用して、設定した単純な ACL6 のいずれかに一致する確立済みの IPv6 接続をすべて終了するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **flush simpleacl6 -estSessions**

GUI のプロシージャ

GUI を使用して、設定したシンプル ACL のいずれかに一致する確立済みの IPv4 接続をすべて終了するには、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2. [簡易 **ACL**] タブの [操作] ボックスの一覧で、[フラッシュ] をクリックします。

GUI を使用して、設定した単純な ACL6 のいずれかに一致する確立済みの IPv6 接続をすべて終了するには、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [ACL**] に移動します。
2. [簡易 **ACL6s**] タブの [操作] ボックスの一覧で、[フラッシュ] をクリックします。

拡張 **ACL** および拡張 **ACL 6**

October 7, 2021

拡張 ACL および拡張 ACL6 は、単純な ACL では使用できないパラメータとアクションを提供します。送信元 IP アドレス、送信元ポート、アクション、プロトコルなどのパラメータに基づいてデータをフィルタリングできます。パケットを許可する、パケットを拒否する、またはパケットをブリッジするタスクを指定できます。

拡張 ACL および ACL 6 は、作成後に変更できます。また、プライオリティを再番号付けして、評価の順序を指定できます。

注**:** シンプル ACL と拡張 ACL の両方を設定する場合、シンプル ACL は拡張 ACL よりも優先されます。

拡張 ACL および ACL 6 に対して次のアクションを実行できます。変更、適用、無効化、有効、削除、および番号付け(優先度)。拡張 ACL と ACL 6 を表示して設定を確認し、統計情報を表示できます。

拡張 ACL に一致するパケットの詳細を記録するように Citrix ADC を構成できます。

拡張 **ACL** と拡張 **ACL6** の適用：単純な ACL および ACL6 とは異なり、Citrix ADC で作成された拡張 ACL および ACL6 は、適用されるまで機能しません。また、拡張 ACL または ACL の無効化、優先度の変更、ACL の削除など、拡張 ACL または ACL 6 に変更を加えた場合は、拡張 ACL または ACL 6 を再適用する必要があります。ログを有効にした後に再適用する必要があります。拡張 ACL または ACL6 を適用する手順は、これらすべてを再適用します。たとえば、拡張 ACL ルール 1 ～10 を適用し、次にルール 11 を作成して適用すると、最初の 10 個のルールが新たに適用されます。

セッションに関連した DENY ACL がある場合、ACL を適用すると、そのセッションが終了します。

拡張 ACL と ACL 6 はデフォルトで有効になっています。それらが適用されると、Citrix ADC は受信パケットと受信パケットを比較し始めます。ただし、無効にした場合は、再適用された場合でも、再度有効にするまでは使用されません。

拡張 **ACL** および拡張 **ACL 6** のプライオリティの再番号付け：プライオリティ番号によって、拡張 ACL または ACL 6 がパケットに対して照合される順序が決まります。プライオリティ番号が低い ACL のプライオリティが高くなります。これは、プライオリティ番号が高い（優先度が低い）ACL の前に評価され、パケットに一致する最初の ACL によって、パケットに適用されるアクションが決まります。

拡張 ACL または ACL6 を作成すると、特に指定しない限り、Citrix ADC は 10 の倍数の優先度番号を自動的に割り当てます。たとえば、2 つの拡張 ACL のプライオリティがそれぞれ 20 と 30 で、3 番目の ACL にそれらの数値の間に値を持たせたい場合は、値 25 を割り当てます。ACL の評価順序を後で保持し、番号を 10 の倍数に復元する場合は、再番号付け手順を使用できます。

拡張 **ACL** および拡張 **ACL 6** の設定

Citrix ADC で拡張 ACL または ACL6 を構成するには、次のタスクがあります。

- 拡張 **ACL** または **ACL6** を作成します。拡張 ACL または ACL6 を作成して、パケットを許可、拒否、ブリッジします。パケットの送信元または宛先 IP アドレスと照合する IP アドレスまたは IP アドレスの範囲を指定できます。着信パケットのプロトコルと照合するプロトコルを指定できます。
- （任意）拡張 **ACL** または **ACL6** を変更します。以前に作成した拡張 ACL または ACL 6 を変更できます。または、一時的に使用を中止する場合は、無効にして後で再度有効にすることができます。
- 拡張 **ACL** または **ACL 6** を適用します。拡張 ACL または ACL6 を作成、変更、無効化、または再有効化、または削除した後、拡張 ACL または ACL 6 を適用してアクティブ化する必要があります。
- （任意）拡張 **ACL** または **ACL 6** のプライオリティを再番号付けします。10 の倍数ではないプライオリティで ACL を設定していて、番号付けを 10 の倍数に戻す場合は、再番号付け手順を使用します。

CLI のプロシージャ

CLI を使用して拡張 **ACL** を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns acl** <aclname> <aclaction> [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] [-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>] [-**icmpType** <positive_integer> [-**icmpCode** <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)] [-**logstate** (ENABLED | DISABLED) [-**ratelimit** <positive_integer>]]
- **show ns acl** [<aclName>]

CLI を使用して拡張 **ACL6** を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns acl6** <acl6name> <acl6action> [-**srcIPv6** [<operator>] <srcIPv6Val>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIPv6** [<operator>] <destIPv6Val>] [-**destPort** [<operator>] <destPortVal>] [-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>] [-**icmpType** <positive_integer> [-**icmpCode** <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)]
- **show ns acl6** [<aclName>]

CLI を使用して拡張 **ACL** を変更するには、次の手順を実行します。

拡張 ACL を変更するには、**set ns acl** コマンド、拡張 ACL の名前、および変更するパラメータを新しい値で入力します。

CLI を使用して拡張 **ACL6** を変更するには、次の手順を実行します。

拡張 ACL6 を変更するには、**set ns acl6** コマンド、拡張 ACL6 の名前、および変更するパラメータを新しい値で入力します。

CLI を使用して拡張 **ACL** を無効または有効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **disable ns acl** <aclname>
- **enable ns acl** <aclname>

CLI を使用して拡張 **ACL6** を無効または有効にするには:

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **disable ns acl6** <aclname>
- **enable ns acl6** <aclname>

CLI を使用して拡張 **ACL** を適用するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **apply ns acls**

CLI を使用して拡張 **ACL6** を適用するには:

コマンドプロンプトで入力します。

- **apply ns acls6**

CLI を使用して拡張 **ACL** の優先順位を変更するには:

コマンドプロンプトで入力します。

- **renumber ns acls**

CLI を使用して拡張 **ACL6** の優先順位を変更するには:

コマンドプロンプトで入力します。

- **renumber ns acls6**

GUI のプロシージャ

GUI を使用して拡張 **ACL** を設定するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブで、新しい拡張 ACL を追加するか、既存の拡張 ACL を編集します。既存の拡張 ACL を有効または無効にするには、その拡張 ACL を選択し、[アクション] リストから [有効] または [無効] を選択します。

GUI を使用して拡張 **ACL6** を設定するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL6**] タブで、新しい拡張 ACL6 を追加するか、既存の拡張 ACL6 を編集します。既存の拡張 ACL6 を有効または無効にするには、それを選択し、[操作] リストから [有効] または [無効] を選択します。

GUI を使用して拡張 **ACL** を適用するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブの [アクション] リストで、[適用] をクリックします。

GUI を使用して拡張 **ACL6** を適用するには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL 6**] タブの [アクション] リストで、[適用] をクリックします。

GUI を使用して拡張 **ACL** のプライオリティを再番号付けするには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブの [アクション] リストで、[優先度の再番号付け] をクリックします。

GUI を使用して拡張 **ACL6** の優先順位を再番号付けするには、次の手順を実行します。

- [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL 6**] タブの [アクション] リストで、[優先度の再番号付け] をクリックします。

設定例

次の表に、コマンドラインインターフェイスを介して拡張 ACL ルールを設定する例を示します。ACL の設定例。

拡張 **ACL** のロギング

拡張 ACL に一致するパケットの詳細を記録するように Citrix ADC を構成できます。

ACL 名に加えて、ログに記録される詳細には、送信元および宛先 IP アドレスなどのパケット固有の情報が含まれます。この情報は、有効になっているグローバルロギング（syslog or nslog）のタイプに応じて、syslog ファイルまたはnslogファイルに保存されます。

ロギングは、グローバルレベルと ACL レベルの両方で有効にする必要があります。グローバル設定が優先されます。

ロギングを最適化するために、同じフローからの複数のパケットが ACL に一致すると、最初のパケットの詳細だけがログに記録され、同じフローに属するすべてのパケットに対してカウンタが増加します。フローは、送信元 IP アドレス、宛先 IP アドレス、送信元ポート、宛先ポート、およびプロトコルパラメータに対して同じ値を持つパケットのセットとして定義されます。ログメッセージのフラッディングを避けるために、Citrix ADC は、同じフローに属するパケットが繰り返しログに記録されないように内部レート制限を実行します。任意の時点でログに記録できる異なるフローの総数は 10,000 に制限されています。

注**:** ロギングを有効にした後に ACL を適用する必要があります。

CLI のプロシージャ

CLI を使用して拡張 **ACL** ロギングを設定するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、ログを構成し、構成を確認します。

- **set ns acl** <aclName> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
- **apply acls**
- **show ns acl** [<aclName>]

GUI のプロシージャ

GUI を使用して拡張 **ACL** ロギングを設定するには、次の手順を実行します。

1. [システム**] > [**ネットワーク**] > [ACL**] に移動し、[拡張 **ACL**] タブで拡張 ACL を開きます。
2. 次のパラメーターを設定します。
   - [**Log State**]：拡張 ACL 規則に関連するイベントのロギングを有効または無効にします。ログメッセージは、構成されたsyslog or auditlogサーバに保存されます。
   - [**Log Rate Limit**]：1 秒間に生成されるログメッセージの最大数。このパラメータを設定する場合は、[Log State] パラメーターを有効にする必要があります。

構成例

```
> set ns acl restrict -logstate ENABLED -ratelimit 120
Warning: ACL modified, apply ACLs to activate change

> apply ns acls
Done
<!--NeedCopy-->
```


拡張 **ACL6** のロギング

拡張 ACL6 ルールに一致するパケットの詳細を記録するように Citrix ADC アプライアンスを構成できます。ACL6 名に加えて、ログに記録される詳細には、送信元および宛先 IP アドレスなどのパケット固有の情報が含まれます。この情報は、Citrix ADC アプライアンスで構成したログの種類（`syslog or nslog`）に応じて、syslog または`nslog`ファイルに保存されます。

ロギングを最適化するために、同じフローからの複数のパケットが ACL6 と一致する場合、最初のパケットの詳細だけがログに記録されます。カウンタは、同じフローに属する他のすべてのパケットに対して増分されます。フローは、次のパラメータに対して同じ値を持つパケットのセットとして定義されます。

- 接続元 IP
- 接続先 IP
- 送信元ポート
- Destination port
- プロトコル (TCP または UDP)

着信パケットが同じフローからのものでない場合、新しいフローが作成されます。任意の時点でログに記録できる異なるフローの総数は 10,000 に制限されています。

CLI のプロシージャ

CLI を使用して拡張 **ACl6** ルールのロギングを設定するには、次の手順を実行します。

- 拡張 ACL6 ルールの追加時にログを構成するには、コマンドプロンプトで次のように入力します。
  - **add acl6** <acl6Name> <acl6action> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
  - **apply acls6**
  - **show acl6** [<acl6Name>]
- 既存の拡張 ACL6 ルールのログを構成するには、コマンドプロンプトで次のように入力します。
  - **set acl6** <acl6Name> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
  - **show acl6** [<acl6Name>]
  - **apply acls6**

GUI のプロシージャ

GUI を使用して拡張 **ACL6** ロギングを設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL 6**] タブをクリックします。
2. 既存の拡張 ACL6 ルールの追加または変更時に、次のパラメータを設定します。
   - ログ状態：拡張 ACL6s ルールに関連するイベントのロギングを有効または無効にします。ログメッセージは、設定された syslog または`auditlog`サーバに保存されます。

- [**Log Rate Limit**]：1 秒間に生成されるログメッセージの最大数。このパラメータを設定する場合は、[**Log State**] パラメーターを有効にする必要があります。

構成例

```
> set acl6 ACL6-1 -logstate ENABLED -ratelimit 120
Done

> apply acls6
Done
<!--NeedCopy-->
```

拡張 **ACL** および拡張 **ACL 6** の統計情報の表示

拡張 ACL および ACL 6 の統計情報を表示できます。

次の表に、拡張 ACL および ACL 6 に関連する統計情報とその説明を示します。

統計情報	**Specifies**
ACL の一致を許可	処理モードが ALLOW に設定された ACL と一致するパケット。Citrix ADC はこれらのパケットを処理します。
NAT ACL が一致する	NAT ACL に一致するパケット。その結果、NAT セッションが発生します。
ACL の一致を拒否する	処理モードが DENY に設定された ACL と一致するため、ドロップされたパケット。
ブリッジ ACL マッチ	ブリッジ ACL に一致するパケット。トランスペアレントモードでは、サービス処理をバイパスします。
ACL マッチ	ACL に一致するパケット。
ACL ミス	どの ACL とも一致しないパケット。
ACL カウント	ユーザによって設定された ACL ルールの総数。
有効な ACL カウント	内部で設定された有効な ACL の総数。IP アドレスの範囲を持つ拡張 ACL の場合、Citrix ADC アプライアンスは各 IP アドレスに対して内部的に拡張 ACL を作成します。たとえば、1000 個の IPv4 アドレス（範囲またはデータセット）を持つ拡張 ACL の場合、Citrix ADC は内部的に 1000 の拡張 ACL を作成しました。

CLI のプロシージャ

CLI を使用してすべての拡張 **ACL** の統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat ns acl**

CLI を使用してすべての拡張 **ACL6** の統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **stat ns acl6**

GUI のプロシージャ

GUI を使用して拡張 **ACL** の統計情報を表示するには、次の手順を実行します。

- [システム**] > [**ネットワーク**] > [ACL**] に移動し、[拡張 **ACL**] タブで拡張 ACL を選択し、[統計情報] をクリックします。

GUI を使用して拡張 **ACL6** の統計情報を表示するには、次の手順を実行します。

- [システム**] > [**ネットワーク**] > [ACL**] に移動し、[拡張 **ACL 6**] タブで拡張 ACL を選択し、[統計情報] をクリックします。

ステートフル **ACL**

ステートフル ACL ルールは、要求がルールに一致したときにセッションを作成し、これらの応答が Citrix ADC アプライアンスの拒否 ACL ルールと一致していても結果の応答を許可します。ステートフル ACL は、これらの特定の応答を許可するために、より多くの ACL ルール/転送セッションルールを作成する作業をオフロードします。

ステートフル ACL は、次の要件を持つ Citrix ADC アプライアンスのエッジファイアウォール展開に最適です。

- Citrix ADC アプライアンスは、内部クライアントから開始された要求とインターネットからの関連する応答を許可する必要があります。
- アプライアンスは、クライアント接続に関連しないパケットをインターネットからドロップする必要があります。

はじめに

ステートフル ACL 規則を設定する前に、次の点に注意してください。

- Citrix ADC アプライアンスは、ステートフル ACL ルールとステートフル ACL6 ルールをサポートしています。
- 高可用性セットアップでは、ステートフル ACL ルールのセッションは 2 次ノードに同期されません。
- ルールが Citrix ADC NAT 構成にバインドされている場合、ACL ルールをステートフルとして構成することはできません。Citrix ADC NAT 構成の例を次に示します。

  - RNAT
  - 大規模 NAT（ラージスケール NAT44、DS-Lite、large scale NAT64）
  - NAT64
  - 転送セッション
- この ACL ルールに TTL パラメータと Established パラメータが設定されている場合、ACL ルールをステートフルとして設定することはできません。
- ステートフル ACL ルール用に作成されたセッションは、次の ACL 操作に関係なく、タイムアウトするまで存続します。
  - ACL の削除
  - ACL を無効にする
  - ACL をクリアする
- ステートフル ACL は、次のプロトコルではサポートされていません。
  - アクティブ FTP
  - TFTP

ステートフル **IPv4 ACL** ルールの設定

ステートフル ACL ルールの設定は、ACL ルールのステートフルパラメータを有効にすることで構成されます。

CLI を使用して **ACL** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

- ACL ルールの追加中にステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **add acl** <lname> ALLOW -**stateful** (ENABLED | DISABLED)
  - **apply acls**
  - **show acl** <name>
- 既存の ACL ルールのステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **set acl** <name> -**stateful** (ENABLED | DISABLED)
  - **apply acls**
  - **show acl** <name>

GUI を使用して **ACL** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL**] タブをクリックします。
2. 既存の ACL ルールの追加または変更中に **Stateful** パラメータを有効にします。

構成例

```
1 > add acl ACL-1 allow -srciP 1.1.1.1 -stateful Yes
2
```



```
Done

> apply acls

Done

> show acl

1)         Name: ACL-1

    Action: ALLOW                          Hits: 0

    srcIP = 1.1.1.1

    destIP

    srcMac:

    Protocol:

    Vlan:                                 Interface:

    Active Status: ENABLED                 Applied Status: NOTAPPLIED

    Priority: 10                           NAT: NO

    TTL:

    Log Status: DISABLED

    Forward Session: NO

    Stateful: YES
<!--NeedCopy-->
```

ステートフル **ACL6** ルールの設定

ステートフル ACL6 ルールの設定は、ACL6 ルールのステートフルパラメータを有効にすることで構成されます。

CLI を使用して **ACL6** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

- ACL6 ルールの追加中にステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **add acl6** <name> ALLOW -stateful (ENABLED | DISABLD)

  - **apply acls6**
  - **show acl6** <name>

- 既存の ACL6 ルールのステートフルパラメータを有効にするには、コマンドプロンプトで次のように入力します。
  - **set acl6** <name> -**stateful** (ENABLED | DISABLED)
  - **apply acls6**
  - **show acl6** <name>

GUI を使用して **ACL6** ルールのステートフルパラメータを有効にするには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**ACL**] に移動し、[拡張 **ACL6**] タブをクリックします。
2. 既存の ACL6 ルールの追加または変更中に **Stateful** パラメータを有効にします。

構成例

```
>  add acl6 ACL6-1 allow -srcipv6 1000::1  - stateful Yes

Done

>  apply acls6

Done

> show acl6

1)    Name: ACL6-1

    Action: ALLOW                          Hits: 0

    srcIPv6 = 1000::1

    destIPv6

    srcMac:

    Protocol:

    Vlan:                                 Interface:

    Active Status: ENABLED                 Applied Status: NOTAPPLIED

    Priority: 10                           NAT: NO

```



```
29      TTL:
30
31      Forward Session: NO
32
33      Stateful: YES
34  <!--NeedCopy-->
```

データセットベースの拡張 **ACL**

企業では多くの ACL が必要です。多くの ACL の設定と管理は、頻繁に変更が必要な場合、困難で面倒です。

Citrix ADC アプライアンスは、拡張 ACL のデータセットをサポートします。データセットは、Citrix ADC アプライアンスの既存の機能です。データセットは、数値（整数）、IPv4 アドレス、または IPv6 アドレスのインデックス付きパターンの配列です。

拡張 ACL でのデータセットのサポートは、共通の ACL パラメータを必要とする複数の ACL ルールを作成する場合に役立ちます。

ACL ルールの作成時に、共通パラメータを指定する代わりに、これらの共通パラメータを含むデータセットを指定できます。

データセットに加えられた変更は、このデータセットを使用している ACL ルールに自動的に反映されます。データセットを持つ ACL は、構成と管理が簡単です。また、従来の ACL よりも小さく、読みやすくなっています。

現在、Citrix ADC アプライアンスは、拡張 ACL の IPv4 アドレスタイプデータセットのみをサポートしています。

はじめに

データセットベースの拡張 ACL ルールを構成する前に、次の点に注意してください。

- Citrix ADC アプライアンスのデータセット機能に精通していることを確認してください。データセットの詳細については、「パターンセットとデータセット」を参照してください。
- Citrix ADC アプライアンスは、IPv4 拡張 ACL のデータセットのみをサポートします。
- Citrix ADC アプライアンスは、拡張 ACL の IPv4 タイプのデータセットのみをサポートします。
- Citrix ADC アプライアンスは、スタンドアロン、高可用性、クラスタのすべてのセットアップでデータセットベースの拡張 ACL をサポートしています。
- IP アドレスの範囲を持つ拡張 ACL の場合、Citrix ADC アプライアンスは各 IP アドレスに対して内部的に拡張 ACL を作成します。たとえば、データセットにバインドされた 1000 個の IPv4 アドレスを持つ IPv4 データセットベースの拡張 ACL の場合、Citrix ADC アプライアンスは内部的に 1000 の拡張 ACL を作成しました。
  - Citrix ADC アプライアンスは、最大 10K の拡張 ACL をサポートします。データセットにバインドされた IP アドレスの範囲を持つ IPv4 データセットベースの拡張 ACL の場合、拡張 ACL の合計数が上限に達すると、Citrix ADC アプライアンスは内部 ACL の作成を停止します。

  - 拡張 ACL 統計情報の一部として、次のカウンタがあります。

    * **ACL** カウント。ユーザによって設定された ACL ルールの総数。
    * 有効な **ACL** カウント。Citrix ADC アプライアンスが内部で構成する有効な ACL ルールの総数。

    詳細については、拡張 ACL および拡張 ACL6 の統計情報の表示を参照してください。

- Citrix ADC アプライアンスは、次の `set` および `unset` 操作をサポートしていません。associating/dissociating 拡張 ACL のパラメーターを持つデータセット。ACL パラメータをデータセットに設定できるのは、`add` 操作中に限られます。

データセットベースの拡張 **ACL** を構成する

データセットベースの拡張 ACL ルールの構成は、次のタスクで構成されます。

- データセットを追加します。データセットは、数値（整数）、IPv4 アドレス、または IPv6 アドレスのインデックス付きパターンの配列です。このタスクでは、IPv4 タイプのデータセットなど、データセットのタイプを作成します。
- 値をデータセットにバインドします。データセットの値または値の範囲を指定します。指定する値は、データセットタイプと同じタイプである必要があります。たとえば、IPv4 タイプのデータセットに IPv4 アドレスまたは IPv4 アドレスの範囲を指定できます。
- 拡張 **ACL** を追加し、**ACL** パラメーターをデータセットに設定します。拡張 ACL を追加し、必要な ACL パラメータをデータセットに設定します。この設定により、パラメータはデータセットで指定された値に設定されます。
- 拡張 **ACL** を適用します。ACL を適用して、新規または変更された拡張 ACL をアクティブにします。

CLI を使用してポリシーデータセットを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add policy dataset** <name> <type>
- **show policy dataset**

CLI を使用してパターンをデータセットにバインドするには:

コマンドプロンプトで入力します。

- **bind policy dataset** <name> <value> [-endRange <string>]
- **show policy dataset**

CLI を使用して拡張 **ACL** を追加し、**ACL** パラメーターをデータセットに設定するには:

コマンドプロンプトで入力します。

- **add ns acl** <aclname> <aclaction> [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] …
- **show acls**

CLI を使用して拡張 **ACL** を適用するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **apply acls**

構成例

次のデータセットベースの拡張 ACL の設定例では、IPv4 データセット DATASET-ACL-1 が作成されます。2 つの IPv4 アドレス: 192.0.2.30 と 192.0.2.60、および 2 つの IPv4 アドレス範囲 (198.51.100.15-45) と (203.0.113.60-90) が DATASET-ACL-1 にバインドされている。DATASET-ACL-1 は、拡張 ACL ACL-1 の srCip パラメータおよび deStIP パラメータに指定されます。

```
add policy dataset DATASET-ACL-1 IPV4

bind dataset DATASET-ACL-1 192.0.2.30

bind dataset DATASET-ACL-1 192.0.2.60

bind dataset DATASET-ACL-1 198.51.100.15 -endrange 198.51.100.45

bind dataset DATASET-ACL-1 203.0.113.60 -endrange 203.0.113.90

add ns acl ACL-1 ALLOW -srcIP DATASET-ACL-1 -destIP DATASET-ACL-1

apply acls
<!--NeedCopy-->
```

ACL の MAC アドレスワイルドカードマスク

October 7, 2021

拡張 ACL および ACL6 にワイルドカードマスクパラメータが導入され、送信元 MAC アドレスパラメータとともに使用され、着信パケットの送信元 MAC アドレスと照合する MAC アドレスの範囲を定義します。

ワイルドカードマスクは、使用する MAC アドレスの 16 進数と、無視する 16 進数を指定します。ワイルドカードマスクパラメータは、一連の 1 と 0 を指定し、長さは 12 桁です。各桁は、MAC アドレスの対応する 16 進数のマスクです。ワイルドカードマスクのゼロ桁は、MAC アドレスの対応する 16 進数を考慮する必要があることを示し、1 桁は対応する 16 進数を無視することを示します。

ワイルドカードマスクは、次の条件を満たす必要があります。

- ゼロのシリーズが 1 つしかありません
- 1 つのシリーズしか持たない
- 一連のゼロから始める

有効なワイルドカードマスクの例を次に示します。

- 000000111111
- 000000011111
- 000011111111

無効なワイルドカードマスクの例を次に示します。

- 000000111100
- 111110000000
- 010101010101

ACL の場合、MAC アドレス 96: FA: 95: FF: FF: FF: FF: FF: FF: FF のワイルドカードマスクは、MAC アドレス範囲 96:FA: 95:00:00:00-96: FA: 95: FF: FF: FF を定義します。この MAC アドレス範囲は、着信パケットの送信元 MAC アドレスと照合されます。

CLI を使用して ACL ルール内の送信元 MAC アドレスの範囲を指定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 -  add ns acl <name> <action> -srcMac <mac_addr> -srcMacMask <string>
2 -  show ns acl <aclname>
3 <!--NeedCopy-->
```

例:

```
1 add ns acl ACL-1 ALLOW  - protocol TCP  - srcport 2000-3000 -srcMac 96:fa
     :95:1d:67:4a
2   - srcMacMask 000000111111
3 Done
4 <!--NeedCopy-->
```

CLI を使用して ACL6 ルールで送信元 MAC アドレスの範囲を指定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
1 -  add ns acl6 <name> <action> -srcMac <mac_addr> -srcMacMask <string>
2 -  show ns acl6 <acl6name>
3 <!--NeedCopy-->
```


例:

```
1 > add ns acl6 ACL6-1 ALLOW -destIPv6 2001::45 -srcMac 96:fa:90:1d:67:4a
2  - srcMacMask 000000001111
3 Done
4 <!--NeedCopy-->
```

内部ポートでのトラフィックのブロック

September 26, 2022

デフォルトでは、Citrix ADC アプライアンスは、ACL ルールを使用しても一部の種類の内部トラフィックをブロックしません。

次の表は、Citrix ADC アプライアンスが ACL ルールを使用してもブロックしない内部トラフィックの種類を示しています。

Citrix ADC セットアップ	プロトコル	送信先ポート	宛先 IP アドレス
すべて	TCP	3008–3011	NSIP または SNIP
すべて	TCP	179	NSIP または SNIP
すべて	UDP	520	NSIP または SNIP
高可用性	UDP	3003	NSIP
高可用性	TCP	22	NSIP
クラスター	UDP	7000	NSIP

前述のタイプのトラフィックをブロックしないというこの機能は、グローバルレイヤ 3 `Implicit ACL Allow`（`implicitACLAllow`）パラメータのデフォルト設定で指定されます。

ACL ルールを使用して前述のトラフィックタイプをブロックする場合は、このパラメータを無効にできます。高可用性セットアップのアプライアンスは、そのパートナー（プライマリまたはセカンダリ）ノードの例外を作成します。そのノードからのトラフィックはブロックされません。

CLI を使用してこのパラメータを無効または有効にするには:

コマンドプロンプトで入力します。

- **l3param** を設定 -**implicitaClAllow** [有効 | 無効]
- **sh l3param**

注**:** パラメータ implicitaclAllow はデフォルトで有効になっています。

例:

```
> set l3param -implicitACLAllow DISABLED
Done
<!--NeedCopy-->
```

IP ルーティング

October 7, 2021

Citrix ADC アプライアンスは、動的ルーティングと静的ルーティングの両方をサポートします。簡易ルーティングは Citrix ADC の主な役割ではないため、動的ルーティングプロトコルの実行の主な目的は、ルートヘルス注入（RHI）を有効にすることです。これにより、アップストリームルーターは、地形的に分散した仮想サーバーへの複数のルートの中から最適なものを選択できます。

ほとんどの Citrix ADC 実装では、ルーティングのオーバーヘッドを減らすために、静的ルートを使用します。バックアップスタティックルートを作成し、スタティックルートをモニタして、スタティックルートがダウンした場合の自動スイッチオーバーを有効にすることができます。また、重みを割り当ててスタティックルート間のロードバランシングを容易にしたり、ヌルルートを作成してルーティングループを防止したり、IPv6 スタティックルートを設定したりすることもできます。ポリシーベースルート（PBR）を設定できます。PBR は、指定した基準に基づいてルーティングを決定します。

動的ルートの構成

October 7, 2021

ダイナミックルーティングプロトコルが有効の場合、対応するルーティングプロセスはルートアップデートを監視し、ルートをアドバタイズします。ルーティングプロトコルにより、アップストリームルーターは、等価コストマルチパス（ECMP）技術を使用して、2 つのスタンドアロン Citrix ADC アプライアンスでホストされる同一の仮想サーバーへのトラフィックの負荷分散を行うことができます。Citrix ADC アプライアンスの動的ルーティングでは、3 つのルーティングテーブルが使用されます。高可用性設定では、セカンダリアプライアンスのルーティングテーブルがプライマリアプライアンスのルーティングテーブルとミラーリングします。

ダイナミックルーティングプロトコルに関するコマンドリファレンスガイドおよびサポートされていないコマンドについては、ダイナミックルーティングプロトコルコマンドリファレンスガイドおよびサポートされていないコマンドを参照してください。

Citrix ADC では、次のプロトコルがサポートされています。

- Routing Information Protocol (RIP) version 2
- Open Shortest Path First (OSPF) version 2
- Border Gateway Protocol（BGP）
- Routing Information Protocol next generation (RIPng) for IPv6
- Open Shortest Path First (OSPF) version 3 for IPv6
- ISIS プロトコル

複数のプロトコルを同時に有効にできます。

Citrix ADC のルーティングテーブル

Citrix ADC アプライアンスでは、Citrix ADC カーネルルーティングテーブル、FreeBSD カーネルルーティングテーブル、および NSM FIB ルーティングテーブルはそれぞれ異なるルートのセットを保持し、異なる目的を果たします。これらは、UNIX ルーティングソケットを使用して相互に通信します。ルートアップデートは、あるルーティングテーブルから別のルーティングテーブルに自動的には伝播されません。ルーティングテーブルごとにルートアップデートの伝播を設定する必要があります。

NS カーネル・ルーティング・テーブル

NS カーネル・ルーティング・テーブルには、NSIP に対応するサブネット・ルートと、各 SNIP と MIP に対応するサブネット・ルートが格納されます。通常、VIP に対応するルートは NS カーネルルーティングテーブルには存在しません。例外は、add ns ip コマンドを使用して追加され、255.255.255.255 以外のサブネットマスクで設定された VIP です。同じサブネットに属する複数の IP アドレスがある場合、それらは単一のサブネットルートとして抽象化されます。さらに、このテーブルには、ループバックネットワーク（127.0.0.0）へのルート、および CLI（CLI）を介して追加されたスタティックルートが格納されます。この表のエントリは、パケット転送時に Citrix ADC によって使用されます。CLI から、show route コマンドを使用して検査できます。

フリー **BSD** ルーティングテーブル

FreeBSD ルーティングテーブルの唯一の目的は、管理トラフィック (telnet、ssh など) の開始と終了を容易にすることです。Citrix ADC アプライアンスでは、これらのアプリケーションは FreeBSD と緊密に結合されており、FreeBSD はこれらのアプリケーションとの間のトラフィックを処理するために必要な情報を持つことが不可欠です。このルーティングテーブルには、NSIP サブネットへのルートとデフォルトルートが含まれます。さらに、FreeBSD は、Citrix ADC がローカルネットワーク上のホストへの接続を確立するときに、WasCloned（W）タイプのルートを追加します。このルーティングテーブルのエントリには高度に特殊なユーティリティがあるため、NS カーネルと NSM FIB ルーティングテーブルからのその他のルート更新はすべて FreeBSD ルーティングテーブルをバイパスします。route コマンドで変更しないでください。FreeBSD ルーティングテーブルは、任意の UNIX シェルから netstat コマンドを使用して検査できます。

ネットワークサービスモジュール（**NSM**）**FIB**

NSM FIB ルーティングテーブルには、ダイナミックルーティングプロトコルによってネットワーク内のピアに配布されるアドバタイズ可能なルートが含まれています。これには、次のものが含まれます。

- 接続されたルート。Citrix ADC から直接到達可能な IP サブネット。通常、NSIP サブネットとルーティングプロトコルが有効になっているサブネットに対応するルートは、NSM FIB に接続ルートとして存在します。
- カーネルルート。-hostRoute オプションが有効になっているすべての VIP アドレスは、必要な RHI レベルを満たしている場合、カーネルルートとして NSM FIB に存在します。さらに、NSM FIB には、CLI で設定された-advertise オプションが有効になっている、すべてのスタティックルートが含まれます。また、Citrix ADC がスタティックルートアドバタイズ（SRADV）モードで動作している場合、CLI で設定されたすべてのスタティックルートが NSM FIB に存在します。これらのスタティックルートは NSM FIB でカーネルルートとしてマークされます。これは、実際には NS カーネルルーティングテーブルに属するためです。
- スタティックルート。通常、VTYSH で設定されたスタティックルートは NSM FIB に存在します。プロトコルのアドミニストレーティブディスタンスが変更された場合、必ずしもそうとは限りません。注意すべき重要な点は、これらのルートが NS カーネルルーティングテーブルに入らないことです。
- 学習ルート。Citrix ADC がルートを動的に学習するように設定されている場合、NSM FIB には、さまざまなダイナミックルーティングプロトコルによって学習されたルートが含まれます。ただし、OSPF によって学習されたルートには特別な処理が必要です。これらは、OSPF プロセスで fib-install オプションが有効になっている場合に限り、FIB にダウンロードされます。これは、VTYSH のルータ設定ビューから実行できます。

高可用性セットアップでのダイナミックルーティング

高可用性設定では、プライマリノードがルーティングプロセスを実行し、ルーティングテーブルの更新をセカンダリノードに伝達します。セカンダリノードのルーティングテーブルは、プライマリノードのルーティングテーブルをミラーリングします。

ノンストップフォワーディング

フェールオーバー後、セカンダリノードはプロトコルの開始、ルートの習得、ルーティングテーブルの更新に多少時間がかかります。ただし、セカンダリノードのルーティングテーブルは、プライマリノードのルーティングテーブルと同じであるため、ルーティングには影響しません。この動作モードは、ノンストップフォワーディングと呼ばれます。

ブラックホール回避メカニズム

フェールオーバー後、新しいプライマリノードはすべての VIP ルートをアップストリームルータに挿入します。ただし、このルーターは、古いプライマリノードのルートを 180 秒間保持します。ルータはフェールオーバーを認識しないため、2 つのノード間でトラフィックのロードバランシングを試みます。古いルートが期限切れになる 180 秒の間に、ルータはトラフィックの半分を古い非アクティブなプライマリノード（実際にはブラックホール）に送信します。

これを防ぐために、新しいプライマリノードは、ルートを挿入するときに、古いプライマリノードによって指定されたメトリックよりもわずかに低いメトリックを割り当てます。

ダイナミックルーティングを設定するためのインターフェイス

ダイナミックルーティングを設定するには、GUI またはコマンドラインインターフェイスのいずれかを使用します。Citrix ADC は、CLI と仮想テレタイプシェル（VTYSH）の 2 つの独立したコマンドラインインターフェイスをサポートしています。CLI はアプライアンスのネイティブシェルです。VTYSH は ZebOS によって公開されています。Citrix ADC ルーティングスイートは、GNU Zebra の商用バージョンである ZebOS に基づいています。

> 注:
>
> CLI でのみ設定できるコマンドを除き、すべてのコマンドに VTYSH を使用することをお勧めします。CLI の使用は、通常、ルーティングプロトコルの有効化、ホストルートアドバタイズメントの設定、およびパケット転送用のスタティックルートの追加を行うコマンドに限定する必要があります。

ダイナミックルーティングプロトコルコマンドリファレンスガイドおよびサポートされていないコマンド

次の表に、Citrix ADC アプライアンスでのさまざまな動的ルーティングプロトコル、およびサポートされていないコマンドに関するコマンドリファレンスガイドのリンクを示します：動的ルーティングプロトコルリファレンスガイドおよびサポートされていないコマンド。

RIP の設定

October 7, 2021

ルーティング情報プロトコル（RIP）は、ディスタンスベクトルプロトコルです。Citrix ADC は、RFC 1058 および RFC 2453 で定義されている RIP をサポートしています。RIP はどのサブネットでも実行できます。

RIP を有効にした後、RIP ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、RIP 伝播を制限できます。RIP 設定を表示して、設定を確認できます。

RIP の有効化および無効化

RIP を有効または無効にするには、次のいずれかの手順を使用します。RIP を有効にすると、Citrix ADC アプライアンスは RIP プロセスを開始します。RIP を無効にすると、アプライアンスは RIP プロセスを停止します。

CLI を使用して RIP ルーティングを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドを入力して、RIP を有効または無効にします。

- **enable ns feature RIP**
- **disable ns feature RIP**

GUI を使用して RIP ルーティングを有効または無効にするには、次の手順を実行します。

1. [システム] **>** [設定] に移動し、[モードと機能] で [高度な機能の変更] をクリックします。
2. [**RIP** ルーティング] オプションを選択または選択解除します。

アドバタイズルート

RIP を使用すると、アップストリームルーターは、2 つのスタンドアロン Citrix ADC アプライアンスでホストされている 2 つの同一の仮想サーバー間でトラフィックを負荷分散できます。ルートアドバタイズメントにより、アップストリームルーターは Citrix ADC 背後にあるネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用してルートをアドバタイズするように RIP を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router rip	RIP ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
redistribute static	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布します。

例**:**

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

RIP 伝播の制限

設定をトラブルシューティングする必要がある場合は、任意のインターフェイスでリスン専用モードを設定できます。

VTYSH コマンド・ラインを使用して RIP 伝播を制限するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。

コマンド	Specifies
router rip	RIP ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
passive-interface < vlan_name>	指定された VLAN にバインドされたインターフェイス上のルーティングアップデートを抑制します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

RIP 設定の確認

ルーティングテーブルと他の RIP 設定を表示できます。

VTYSH コマンドラインを使用して RIP 設定を表示するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
リップ	更新された RIP ルーティングテーブルを表示します。
sh rip interface <vlan_name>	指定された VLAN の RIP 情報を表示します。

例:

```
NS# VTYSH
NS# sh rip
NS# sh rip interface VLAN0
<!--NeedCopy-->
```


OSPF の設定

December 7, 2021

Citrix ADC は、オープン最短パスファースト（OSPF）バージョン 2（RFC 2328）をサポートしています。Citrix ADC 上の OSPF の機能は次のとおりです。

- vserver がアクティブな場合、vserver へのホストルートをルーティングプロトコルに挿入できます。
- OSPF はどのサブネットでも実行できます。
- ネイバー OSPF ルーターによってアドバタイズされたルートラーニングは、Citrix ADC で無効にできます。
- Citrix ADC は、すべてのルートに対して Type-1 または Type2 の外部メトリックをアドバタイズできます。
- Citrix ADC は、VIP ルートのユーザー指定のメトリック設定をアドバタイズできます。たとえば、特別なルートマップを使用せずに VIP ごとにメトリックを設定できます。
- Citrix ADC の OSPF エリア ID を指定できます。
- Citrix ADC は、それほどスタブではないエリア（NSSA）をサポートしています。NSSA は OSPF スタブエリアに似ていますが、外部ルートを限定的にスタブエリアに挿入できます。NSSA をサポートするために、リンクステートアドバタイズメント（LSA）領域の新しいオプションビット（N ビット）と新しいタイプ（タイプ 7）が定義されています。タイプ 7 LSA は、NSSA 内の外部ルート情報をサポートします。NSSA エリアボーダールータ（ABR）は、タイプ 7 LSA をタイプ 5 LSA に変換し、OSPF ドメインに伝播します。OSPF 仕様では、次の一般クラスのエリア設定のみが定義されています。
  - タイプ 5 LSA：エリア内部のルータから発信されたルータは、AS Boarder Router（ASBR）によってドメインにフラッディングされます。
  - スタブ：タイプ 5 LSA をエリア内またはエリア全体に伝搬することはできず、代わりに外部宛先へのデフォルトルーティングに依存します。

OSPF を有効にしたら、OSPF ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、OSPF の伝播を制限できます。OSPF 設定を表示して、設定を確認できます。

OSPF の有効化と無効化

OSPF を有効または無効にするには、CLI または GUI を使用する必要があります。OSPF が有効になっていると、Citrix ADC は OSPF プロセスを開始します。OSPF が無効になっていると、Citrix ADC は OSPF ルーティングプロセスを停止します。

CLI を使用して OSPF ルーティングを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

1. **enable ns feature OSPF**
2. **disable ns feature OSPF**

GUI を使用して OSPF ルーティングを有効または無効にするには

1. [システム**] > [**設定**]** に移動し、[モードと機能] グループで [詳細機能の変更] をクリックします。

2. [**OSPF** ルーティング] オプションをオンまたはオフにします。

OSPF ルートの宣伝

OSPF を使用すると、アップストリームルーターは、2 つのスタンドアロン Citrix ADC アプライアンスでホストされている 2 つの同一の仮想サーバー間でトラフィックの負荷を分散できます。ルートアドバタイズにより、アップストリームルーターは Citrix ADC 背後に位置するネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用してルートをアドバタイズするように OSPF を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router OSPF	OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
network A.B.C.D/M area <0-4294967295>	IP ネットワーク上でルーティングを有効にします。
redistribute static	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布する。

例**:**

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# network 10.102.29.0/24 area 0
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

OSPF 伝搬の制限

設定のトラブルシューティングが必要な場合は、任意の VLAN でリッスン専用モードを設定できます。

VTYSH コマンドラインを使用して OSPF 伝播を制限するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router OSPF	OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
passive-interface < vlan_name>	指定した VLAN にバインドされたインターフェイスのルーティングアップデートを抑制します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

OSPF 設定の確認

現在の OSPF ネイバーと OSPF ルートを表示できます。

VTYSH コマンドラインを使用して OSPF 設定を表示するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
sh OSPF neighbor	現在のネイバーを表示します。
sh OSPF route	OSPF ルートを表示します。

例:

```
>VTYSH
NS# sh ip OSPF neighbor
NS# sh ip OSPF route
<!--NeedCopy-->
```


OSPF のグレースフルリスタートの設定

ルーティングプロトコルが設定されている INC 以外の High Availability（HA; 高可用性）設定では、フェールオーバー後にルーティングプロトコルがコンバージされ、新しいプライマリノードと隣接ネイバールータ間のルートが学習されます。ルート学習が完了するまでに時間がかかります。この間、パケットの転送が遅れ、ネットワークパフォーマンスが低下し、パケットがドロップされる可能性があります。

グレースフルリスタートにより、フェールオーバー中の HA セットアップによって、古いプライマリノードの学習済みルートがルーティングデータベースから削除されないように隣接ルータに指示できます。新しいプライマリノードと隣接ルータは、古いプライマリノードのルーティング情報を使用して、ネットワークパフォーマンスを低下させることなく、直ちにパケットの転送を開始します。

> 注:
>
> INC モードの高可用性セットアップでは、グレースフルリスタートはサポートされていません。

VTYSH コマンドラインを使用して OSPF のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# configure terminal	グローバル構成モードを開始します。
router-id <id>	NS(config)# router-id 1.1.1.1	Citrix ADC アプライアンスのルーター識別子を設定します。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。HA セットアップでグレースフルリスタートが正しく機能するためには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
ospf restart grace-period <1-1800>	NS(config)# ospf restart grace-period 170	ヘルパーデバイスでルートが保持される猶予期間を秒単位で指定します。デフォルト値:120 秒。

コマンド	例	コマンドの説明
ospf restart helper max-grace-period <1-1800>	NS(config)# ospf restart helper max-grace-period 180	これは、Citrix ADC アプライアンスがヘルパーモードになる最大猶予期間を制限するオプションのコマンドです。Citrix ADC アプライアンスが設定されたヘルパー最大猶予期間よりも長い猶予期間を持つ不透明な LSA を受信した場合、LSA は破棄され、Citrix ADC はヘルパーモードにはなりません。
router ospf	NS(config)# router ospf	OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
network A.B.C.D/M area <0-4294967295>	NS(config-router)# network 192.0.2.0/24 area 0	IP ネットワーク上でルーティングを有効にします。
capability restart graceful	NS(config-router)# capability restart graceful	OSPF ルーティングプロセスでグレースフルリスタートを有効にします。
redistribute kernel	NS(config-router)# redistribute kernel	カーネルルートを再配布します。

BGP の設定

December 7, 2021

Citrix ADC アプライアンスは BGP（RFC 4271）をサポートしています。Citrix ADC での BGP の機能は次のとおりです。

- Citrix ADC は BGP ピアにルートをアドバタイズします。
- Citrix ADC は、基盤となる仮想サーバーの正常性によって決定される仮想 IP アドレス（VIP）にホストルートを挿入します。
- Citrix ADC は、HA 構成でのフェイルオーバー後、セカンダリノードで BGP を実行するための構成ファイルを生成します。
- このプロトコルは IPv6 ルート交換をサポートします。
- ボーダーゲートウェイプロトコルの As-Override サポート

BGP を有効にしたら、BGP ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、BGP 伝播を制限できます。BGP 設定を表示して、設定を確認できます。

IPv6 BGP の前提条件

IPv6 BGP の設定を開始する前に、次の操作を行います。

- IPv6 BGP プロトコルを理解していることを確認します。
- IPv6 機能を有効にします。

BGP の有効化と無効化

BGP を有効または無効にするには、CLI または GUI を使用する必要があります。BGP が有効になると、Citrix ADC アプライアンスは BGP プロセスを開始します。BGP が無効になっている場合、アプライアンスは BGP プロセスを停止します。

CLI を使用して BGP ルーティングを有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- ns 機能 BGP を有効にする
- ns 機能 BGP を無効にする

GUI を使用して BGP ルーティングを有効または無効にするには

1. [システム] > [設定] に移動し、[モードと機能] グループで [詳細機能の変更] をクリックします。
2. [BGP ルーティング] オプションをオンまたはオフにします。

IPv4 ルートの宣伝

Citrix ADC アプライアンスは、ホストルートを VIP にアドバタイズし、ルートをダウンストリームネットワークにアドバタイズするように構成できます。

VTYSH コマンドラインを使用して IPv4 ルートをアドバタイズするように BGP を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
端末の設定	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。指定可能な値:1 から 4,294,967,295 です。
< IPv4 address> ネイバーリモート AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーのリンクローカル IPv4 アドレスで更新します。

コマンド	Specifies
アドレスファミリ ipv4	アドレスファミリ構成モードを開始します。
< IPv4 address> 近隣アクティブ化	リンクローカルアドレスを使用して、ピアとローカルノード間で IPv4 ルータファミリのプレフィクスを交換します。
カーネルの再配布	カーネルルートを再配布する。
静的な再配布	スタティックルートを再配布します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor 10.102.29.170 remote-as 100
NS(config-router)# Address-family ipv4
NS(config-router-af)# Neighbor 10.102.29.170 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

IPv6 BGP の前提条件

IPv6 BGP の設定を開始する前に、次の操作を行います。

- IPv6 BGP プロトコルを理解していることを確認します。
- IPv6 機能を有効にします。

IPv6 BGP ルートの宣伝

Border Gateway Protocol（BGP）を使用すると、アップストリームルーターは、2 つのスタンドアロンの Citrix ADC アプライアンスでホストされている 2 つの同一の仮想サーバー間でトラフィックの負荷を分散できます。ルートアドバタイズにより、アップストリームルーターは Citrix ADC 背後に位置するネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用して IPv6 ルートをアドバタイズするように BGP を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
端末の設定	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。指定可能な値:1 から 4,294,967,295 です。
< IPv6 address> ネイバーリモート AS < as-number>	IPv6 BGP ネイバーテーブルを、指定した自律システム内のネイバーのリンクローカル IPv6 アドレスで更新します。
アドレスファミリ ipv6	アドレスファミリ構成モードを開始します。
< IPv6 address> 近隣アクティブ化	リンクローカルアドレスを使用して、ピアとローカルノード間で IPv6 ルーターファミリのプレフィクスを交換します。
カーネルの再配布	カーネルルートを再配布する。
静的な再配布	スタティックルートを再配布します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

BGP 設定の確認

VTYSH を使用して BGP 設定を表示できます。

VTYSH コマンドラインを使用して BGP 設定を表示するには

コマンドプロンプトで入力します。

```
VTYSH
```



```
You are now in the VTYSH command prompt. An output similar to the
    following appears:
NS170#
At the VTYSH command prompt, type:
NS170# sh ip BGP
NS170# sh BGP
NS170# sh ip BGP neighbors
NS170# sh ip BGP summary
NS170# sh ip BGP route-map <map-tag>
<!--NeedCopy-->
```

ボーダーゲートウェイプロトコルの **As-Override** サポート

BGP ループ防止機能の一部として、自律システム（AS）パスでルータの自律システム番号（ASN）を含む BGP パケットをルータが受信すると、ルータはそのパケットをドロップします。パケットはルータから発信され、発信元の場所に到達したと想定されます。

企業に同じ ASN を持つサイトが複数ある場合、BGP ループ防止により、同じ ASN を持つサイトは別の ASN によってリンクされません。ルーティングアップデート（BGP パケット）は、別のサイトが受信するとドロップされます。

この問題を解決するために、Citrix ADC ZeboS BGP ルーティングモジュールに BGP AS-Override 機能が追加されました。

ピアデバイスで AS-Override が有効になっている場合、Citrix ADC アプライアンスがピアに転送する BGP パケットを受信し、パケットの ASN がピアの ASN と一致すると、アプライアンスは BGP パケットの ASN を独自の ASN 番号に置き換えてからパケットを転送します。

VTYSH コマンドラインを使用して、特定のネイバーまたはネイバーのグループ（ピアグループ）に対して AS オーバーライドを有効にできます。

VTYSH コマンドラインを使用して IPv4 ネイバーの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	**Specifies**
端末の設定	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。
ネイバーリモート < IPv4 address> AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド <IPv4 address>	指定したネイバーの BGP as-override を有効にします。


```
    > VTYSH NS# configure terminal
    NS(config)# router BGP 200
    NS(config-router)# Neighbor 192.0.2.100 remote-as 100
    NS(config-router)# Neighbor 10.102.29.100 as-override
<!--NeedCopy-->
```

VTYSH コマンドラインを使用して IPv4 BGP ピアグループの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	**Specifies**
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。
** ネイバーピアグループ **<peer group name>	BGP ピアグループを作成します。
** ネイバー・ピア・グループ ** <IPv4 address><peer group name>	ネイバーを指定されたピアグループに関連付けます。
ネイバーリモート <peer group name>AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド <peer group name>	指定したピアグループに関連付けられているすべてのネイバーに対して、BGP as-override を有効にします。

```
    > VTYSH NS# configure terminal
    NS(config)# router BGP 200
    NS(config-router)# neighbor external-peers-1 peer-group
    NS(config-router)# neighbor 192.0.2.101 peer-group external-peers-1
    NS(config-router)# neighbor 192.0.2.102 peer-group external-peers-1
    NS(config-router)# neighbor 192.0.2.103 peer-group external-peers-1
    NS(config-router)# Neighbor external-peers-1 remote-as 100
    NS(config-router)# Neighbor external-peers-1 as-override
<!--NeedCopy-->
```

VTYSH コマンドラインを使用して IPv6 ネイバーの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	**Specifies**
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。
ネイバーリモート < IPv6 address> AS < as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド <IPv6 address>	指定したネイバーの BGP as-override を有効にします。
アドレスファミリ **ipv6**	アドレスファミリ構成モードを開始します。
近隣アクティブ化 < IPv6 address>	リンクローカルアドレスを使用して、指定されたネイバーと Citrix ADC との間で、IPv6 ルーターファミリのプレフィックスを交換します。
ネイバーとしてオーバーライド <IPv6 address>	指定したネイバーの BGP as-override を有効にします。

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# Neighbor a1bc::102 remote-as 100
4      NS(config-router)# Neighbor a1bc::102 as-override
5      NS(config-router)# Address-family ipv6
6      NS(config-router-af)# Neighbor a1bc::102 activate
7      NS(config-router)# Neighbor a1bc::102 as-override
8  <!--NeedCopy-->
```

VTYSH コマンドラインを使用して IPv6 ピアグループの BGP AS オーバーライドを設定するには、次の手順を実行します。

コマンド	**Specifies**
configure terminal	グローバル構成モードを開始します。
router BGP < ASnumber>	BGP 自律システム。< ASnumber> は必須パラメータです。
** ネイバーピアグループ **<peer group name>	BGP ピアグループを作成します。
** ネイバーピアグループ ** <IPv6 address><peer group name>	ネイバーを指定されたピアグループに関連付けます。

コマンド	**Specifies**
ネイバーリモート <peer group name>AS <as-number>	IPv4 BGP ネイバーテーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
ネイバーとしてオーバーライド <peer group name>	指定したピアグループに関連付けられているすべてのネイバーに対して、BGP as-override を有効にします。
アドレスファミリ **ipv6**	アドレスファミリ構成モードを開始します。
近隣アクティブ化 <peer group name>	リンクローカルアドレスを使用して、指定されたピアグループのネイバーと Citrix ADC 間で IPv6 ルーターファミリのプレフィックスを交換します。
ネイバーとしてオーバーライド <peer group name>	指定したピアグループに関連付けられているすべてのネイバーに対して、BGP as-override を有効にします。

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# neighbor external-peers-2 peer-group
4      NS(config-router)# neighbor 2001::1 peer-group external-peers-2
5      NS(config-router)# neighbor 2001::2 peer-group external-peers-2
6      NS(config-router)# Neighbor external-peers-2 remote-as 100
7      NS(config-router)# Neighbor external-peers-2 as-override
8      NS(config-router)# Address-family ipv6
9      NS(config-router-af)# Neighbor external-peers-2 activate
10     NS(config-router)# Neighbor external-peers-2 as-override
11 <!--NeedCopy-->
```

グレースフルリスタート

ルーティングプロトコルが設定されている INC 以外の High Availability（HA; 高可用性）設定では、フェールオーバー後にルーティングプロトコルがコンバージされ、新しいプライマリノードと隣接ネイバールータ間のルートが学習されます。ルート学習が完了するまでに時間がかかります。この間、パケットの転送が遅れ、ネットワークパフォーマンスが低下し、パケットがドロップされる可能性があります。

グレースフルリスタートにより、フェールオーバー中の HA セットアップによって、古いプライマリノードの学習済みルートがルーティングデータベースから削除されないように隣接ルータに指示できます。新しいプライマリノードと隣接ルータは、古いプライマリノードのルーティング情報を使用して、ネットワークパフォーマンスを低下させることなく、直ちにパケットの転送を開始します。

> 注:
>
> INC モードの高可用性セットアップでは、グレースフルリスタートはサポートされていません。

BGP のグレースフルリスタートの設定

VTYSH コマンドラインを使用して BGP のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# ターミナルを設定	グローバル構成モードを開始します。
router-id <ID>	NS (構成) # ルータ ID 1.1.1.1	Citrix ADC アプライアンスのルーター識別子。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。グレースフルリスタートを正しく機能させるには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
router bgp <AS-number>	NS(config)# router bgp 5	BGP 構成モードを開始します。
bgp graceful-restart	NS(config)# bgp graceful-restart	BGP ルーティングプロセスでグレースフルリスタートを有効にします。
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	フェールオーバー後、ヘルパールータが新しいプライマリノードからの TCP 接続を待機する猶予期間を秒単位で指定します。この時間の間、ヘルパールーターはルートを保持します。
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	ヘルパーモードの Citrix ADC アプライアンスがネイバールーターを再起動するための古いルートを保持する時間を秒単位で指定します。デフォルト値は 360 秒です。

コマンド	例	コマンドの説明
neighbor <IPv4 address of the peer router> remote-as <AS-number>	NS(config-router)# neighbor 192.0.2.30 remote-as 2	指定されたネイバールータデバイスとの BGP ピアリングを確立します。
neighbor <IPv4 address of the peer router> capability graceful-restart	NS(config-router)# neighbor 192.0.2.30 capability graceful-restart	指定したネイバーとのグレースフルリスタートを有効にします。
カーネルの再配布	NS (config-router) # カーネルの再配布	カーネルルートを再配布します。

IPv6 BGP のグレースフルリスタートの設定

VTYSH コマンドラインを使用して IPv6 BGP のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# configure terminal	グローバル構成モードを開始します。
ルータ ID <id>	NS(config)# router-id 1.1.1.1	Citrix ADC アプライアンスのルーター識別子を設定します。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。グレースフルリスタートを正しく機能させるには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
router bgp <AS-number>	NS(config)# router bgp 5	BGP プロトコルの構成モードを開始します。
bgp graceful-restart	NS(config)# bgp graceful-restart	BGP ルーティングプロセスでグレースフルリスタートを有効にします。

コマンド	例	コマンドの説明
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	フェールオーバー後、ヘルパールータが新しいプライマリノードからの TCP 接続を待機する猶予期間を秒単位で指定します。この時間の間、ヘルパールーターはルートを保持します。デフォルト値は 360 秒です。
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	ヘルパーモードの Citrix ADC アプライアンスがネイバールーターを再起動するための古いルートを保持する時間を秒単位で指定します。デフォルト値は 360 秒です。
neighbor <IPv6 address> remote-as <AS-number>	NS(config-router)# neighbor 2001:db8::10 remote-as 2	指定されたネイバールータデバイスとの BGP ピアリングを確立します。
address-family ipv6	NS(config-router)#address-family ipv6	アドレスファミリ構成モードを開始します。
neighbor <IPv6 address of the neighbor> activate	NS(config-router-af)#neighbor 2001:db8::10 activate	指定したネイバールータデバイスとのアドレスファミリルートの交換を有効にします。
neighbor <IPv6 address of the neighbor> capability graceful-restart	NS(config-router-af)#neighbor 2001:db8::10 capability graceful-restart	Enables graceful restart with the specified neighbor router device.
redistribute kernel	NS(config-router-af)#redistribute kernel	カーネルルートを再配布します。
exit-address-family	NS(config-router-af)#exit-address-family	アドレスファミリ構成モードを終了します。

IPv4 BGP 用の **MD5** 認証の設定

Citrix ADC アプライアンスは、Border Gateway Protocol（BGP）の MD5 認証をサポートしています。認証を有効にすると、Citrix ADC アプライアンスとそのピアデバイス間で交換される BGP に属する TCP セグメントは、認証が成功した場合にのみ検証され、受け入れられます。認証を成功させるには、両方のピアに同じ MD5 パスワード

を設定する必要があります。認証に失敗すると、BGP ネイバー関係は確立されません。Citrix ADC アプライアンスでの BGP に対する MD5 認証のサポートは、RFC 2385 に準拠しています。

はじめに

BGP MD5 認証の設定を開始する前に、次の点を考慮してください。

- RFC 2385 で説明されている BGP MD5 認証のさまざまなコンポーネントを理解していることを確認します。
- BGP MD5 認証は、Citrix ADC 管理パーティションではサポートされていません。
- BGP MD5 認証は IPv6 BGP 構成ではサポートされていません。
- BGP MD5 認証は、Citrix ADC クラスター構成と高可用性構成でサポートされています。
- FreeBSD には次のような問題があるため、レイヤ 2 の高可用性構成では、キープライブとホールドタイムの値を低く設定し（5 と 15 など）、BGP セッションのグレースフルリスタートを構成することをお勧めします。そうしないと、MD5 認証を有効にすると、フェールオーバー後に BGP がネイバーとの接続を再確立するのに時間がかかることがあります。
  - FreeBSD からの最後の ACK には md5 ダイジェストが含まれていません:
    * https://forums.freebsd.org/threads/11170/
    * http://support.pfsense.narkive.com/povrH5HI/bgp-md5-weird-behavior-when-connection-closes

IPv4 BGP 用の MD5 認証の設定

VTYSH コマンドラインを使用して IPv4 BGP の MD5 認証を設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	Specifies
vtysh	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <AS-number>	BGP プロトコルの構成モードを開始します。<AS-number> は BGP 自律システム番号で、は必須パラメータです。
neighbor <neighbour IPv4 address> remote-as < AS-number >	IPv4 BGP テーブルを、指定した自律システム内のネイバーの IPv4 アドレスで更新します。
neighbor < neighbour IPv4 address > password < password in double quotes>	指定された MD5 パスワードを使用して、指定されたネイバーの MD5 認証を設定します。MD5 認証を正常に実行するには、Citrix ADC アプライアンスとネイバーアプライアンスで同じ MD5 パスワードを構成する必要があります。


```
> vtysh

ns# configure terminal

ns(config)#router bgp 5

ns(config-router)#neighbor 20.20.20.138 remote-as 1

ns(config-router)#neighbor 20.20.20.138 password  “secret”

ns(config-router)#redistribute kernel

ns(config-router)#exit

<!--NeedCopy-->
```

IPv6 RIP の設定

October 7, 2021

IPv6 ルーティング情報プロトコル（RIP）または RIPng は、ディスタンスベクタープロトコルです。このプロトコルは、IPv6 をサポートするための RIP の拡張です。IPv6 RIP を有効にした後、IPv6 RIP ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、IPv6RIP の伝播を制限できます。IPv6 RIP 設定を表示して、構成を確認できます。

IPv6RIP の前提条件

IPv6 RIP の構成を開始する前に、以下を実行してください。

- IPv6RIP プロトコルを理解していることを確認してください。
- Citrix ADC アプライアンスに IPv6PT ライセンスをインストールします。
- IPv6 機能を有効にします。

IPv6RIP ルートのアドバタイズ

IPv6 RIP を使用すると、アップストリームルーターは、2 つのスタンドアロン Citrix ADC デバイスでホストされている 2 つの同一の vserver 間でトラフィックを負荷分散できます。ルートアドバタイズメントにより、アップストリームルーターは Citrix ADC 背後にあるネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用して IPv6 ルートをアドバタイズするように IPv6RIP を設定するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 rip	IPv6 RIP ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
redistribute static	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布します。

例:

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

IPv6RIP 伝搬の制限

設定のトラブルシューティングが必要な場合は、任意のインターフェイスでリッスン専用モードを設定できます。

VTYSH コマンドラインを使用して IPv6RIP の伝播を制限するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 rip	IPv6 RIP ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
passive-interface < vlan_name>	指定された VLAN にバインドされたインターフェイス上のルーティングアップデートを抑制します。

例:


```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

IPv6RIP 構成の確認

VTYSH を使用して、指定した VLAN の IPv6RIP ルーティングテーブルと IPv6RIP 情報を表示できます。

VTYSH コマンドラインを使用して IPv6RIP 設定を表示するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
sh ipv6 rip	更新された IPv6RIP ルーティングテーブルを表示します。
sh ipv6 rip interface <vlan_name>	指定した VLAN の IPv6RIP 情報を表示します。

例:

```
NS# VTYSH
NS# sh ipv6 rip
NS# sh ipv6 rip interface VLAN0
<!--NeedCopy-->
```

IPv6 OSPF の設定

December 7, 2021

IPv6 OSPF または OSPF バージョン 3（OSPF v3）は、IPv6 ルーティング情報の交換に使用されるリンクステートプロトコルです。IPv6 OSPF を有効にした後は、IPv6 OSPF ルートのアドバタイズメントを設定する必要があります。トラブルシューティングのために、IPv6 OSPF の伝播を制限できます。IPv6 OSPF 設定を表示して、設定を確認できます。

IPv6 OSPF の前提条件

IPv6 OSPF の設定を開始する前に、次の操作を行います。

- IPv6 OSPF プロトコルを理解していることを確認します。
- Citrix ADC アプライアンスに IPv6pt ライセンスをインストールします。
- IPv6 機能を有効にします。

IPv6 ルートの宣伝

IPv6 OSPF を使用すると、アップストリームルーターは、2 つのスタンドアロン Citrix ADC デバイスでホストされている 2 つの同一の vServer 間でトラフィックの負荷を分散できます。ルートアドバタイズにより、アップストリームルーターは Citrix ADC 背後に位置するネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用して IPv6 ルートをアドバタイズするように IPv6 OSPF を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 OSPF	IPv6 OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
静的な再配布	スタティックルートを再配布します。
redistribute kernel	カーネルルートを再配布する。

例:

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router ipv6 OSPF
4 NS(config-router)# redistribute static
5 NS(config-router)# redistribute kernel
6 <!--NeedCopy-->
```


IPv6 OSPF 伝搬の制限

設定のトラブルシューティングが必要な場合は、VTYSH を使用して、任意の VLAN でリッスン専用モードを設定します。

VTYSH コマンドラインを使用して IPv6 OSPF 伝播を制限するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 OSPF	IPv6 OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
passive-interface < vlan_name >	指定した VLAN にバインドされたインターフェイスのルーティングアップデートを抑制します。

例:

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router ipv6 OSPF
4 NS(config-router)# passive-interface VLAN0
5 <!--NeedCopy-->
```

IPv6 OSPF 設定の確認

VTYSH を使用して、IPv6 OSPF の現在のネイバーと IPv6 OSPF ルートを表示します。

VTYSH コマンドラインを使用して IPv6 OSPF 設定を表示するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
sh ipv6 OSPF neighbor	現在のネイバーを表示します。
sh ipv6 OSPF route	IPv6 OSPF ルートを表示します。

例:

```
>VTYSH
NS# sh ipv6 OSPF neighbor
NS# sh ipv6 OSPF route
<!--NeedCopy-->
```

OSPFv3 認証

OSPFv3 パケットの整合性、データ発信元認証、およびデータの機密性を確保するには、OSPFv3 ピアに OSPFv3 認証を設定する必要があります。

Citrix ADC アプライアンスは OSPFv3 認証をサポートし、RFC 4552 に部分的に準拠しています。OSPFv3 認証は、認証ヘッダー (AH) とカプセル化セキュリティペイロード (ESP) の 2 つの IPsec プロトコルに基づいています。Citrix ADC アプライアンスは、OSPFv3 認証用に AH プロトコルのみをサポートします。

OSPFv3 認証では、OSPFv3 ピア間で手動で定義された IPSec Security Associations（SA; セキュリティアソシエーション）が使用され、ダイナミック SA の形成に I 手動 SA は、ピア間で使用されるセキュリティパラメータ Index（SPI）値、アルゴリズム、およびキーを定義します。手動 SA では、ピア間でネゴシエーションを行う必要がないため、両方のピアで同じ SA を定義する必要があります。

OSPFv3 認証は、VLAN または OSPFv3 エリアに対して設定できます。VLAN に対してを設定すると、VLAN のメンバーであるすべてのインターフェイスに設定が適用されます。OSPF エリアに OSPFv3 認証を設定すると、そのエリア内のすべての VLAN に設定が適用されます。この設定は、これらの VLAN のメンバーであるすべてのインターフェイスに適用されます。これらの設定は、OSPFv3 認証を直接設定したメンバ VLAN には適用されません。

Citrix ADC アプライアンスで OSPFv3 認証を構成する前に、次の点と制限事項を考慮してください。

- RFC 4552 で説明されている OSPFv3 認証のさまざまなコンポーネントを理解していることを確認します。
- OSPFv3 認証では、認証ヘッダープロトコルのみがサポートされています。セキュリティペイロード (ESP) のカプセル化はサポートされていません。
- ピアインターフェイスには、同じ設定で SA を定義する必要があります。
- 手動キーのキー再生成はサポートされていません。

VTYSH コマンドラインを使用して VLAN 上で OSPFv3 認証を設定するには、次の手順を実行します。

コマンドプロンプトで、OSPFv3 認証 VLAN コマンドの順に次のコマンドを入力します。

例:

```
> VTYSH NS# configure terminal
NS(config)# interface vlan2
```



```
3 NS(config-if)# ipv6 ospf authentication ipsec spi 256 md5 123456789
     ABCDEF0123456789ABCDEF0
4 <!--NeedCopy-->
```

VTYSH コマンドラインを使用して OSPF エリアで OSPFv3 認証を設定するには、次の手順を実行します。

コマンドプロンプトで、OSPFv3 認証 OSPF エリアコマンドの順に次のコマンドを入力します。

例:

```
1 > VTYSH NS# configure terminal
2 ns(config)#router ipv6 ospf 30
3 ns(config-router)# area 1 authentication ipsec spi 256
     md5123456789ABCDEF0123456789ABCDEF0
4 <!--NeedCopy-->
```

IPv6 OSPF のグレースフルリスタートの設定

ルーティングプロトコルが設定されている INC 以外の High Availability（HA; 高可用性）設定では、フェールオーバー後にルーティングプロトコルがコンバージされ、新しいプライマリノードと隣接ネイバールータ間のルートが学習されます。ルート学習が完了するまでに時間がかかります。この間、パケットの転送が遅れ、ネットワークパフォーマンスが低下し、パケットがドロップされる可能性があります。

グレースフルリスタートにより、フェールオーバー中の HA セットアップによって、古いプライマリノードの学習済みルートがルーティングデータベースから削除されないように隣接ルータに指示できます。新しいプライマリノードと隣接ルータは、古いプライマリノードのルーティング情報を使用して、ネットワークパフォーマンスを低下させることなく、直ちにパケットの転送を開始します。

> 注:
>
> INC モードの高可用性セットアップでは、グレースフルリスタートはサポートされていません。

VTYSH コマンドラインを使用して IPv6 OSPF のグレースフルリスタートを設定するには、コマンドプロンプトで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
VTYSH	> VTYSH	VTYSH コマンドプロンプトに入ります。
configure terminal	NS# configure terminal	グローバル構成モードを開始します。

コマンド	例	コマンドの説明
router-id id>	NS(config)#router-id 1.1.1.1	Citrix ADC アプライアンスのルーター識別子を設定します。この ID は、すべてのダイナミックルーティングプロトコルに対して設定されます。HA セットアップでグレースフルリスタートが正しく機能するためには、高可用性セットアップの他のノードに同じ ID を指定する必要があります。
IPv6ospf restart grace-period <1-1800>	NS(config)# IPv6ospf restart grace-period 170	ヘルパーデバイスでルートが保持される猶予期間を秒単位で指定します。デフォルト値:120 秒。
IPv6 ospf restart helper max-grace-period <1-1800>	NS(config)# IPv6 ospf restart helper max-grace-period 180	これは、Citrix ADC アプライアンスがヘルパーモードになる最大猶予期間を制限するオプションのコマンドです。Citrix ADC アプライアンスが設定されたヘルパー最大猶予期間よりも長い猶予期間を持つ不透明な LSA を受信した場合、LSA は破棄され、Citrix ADC はヘルパーモードにはなりません。
interface <VLANID>	NS(config)#interface vlan3	VLAN 構成モードを開始します。
ipv6 router ospf area <area_id> tag <tag_id>	NS(config-if)#ipv6 router ospf area 0 tag 1	VLAN 上で IPv6 OSPF ルーティングプロセスを開始します。
exit	NS(config-if)#exit	VLAN 構成モードを終了します。
router ipv6 ospf	NS(config)# router ipv6 ospf 1	IPv6 OSPF ルーティングプロセスを開始し、ルーティングプロセスの構成モードを開始します。
capability restart graceful	NS(config-router)#capability restart graceful	IPv6 OSPF ルーティングプロセスでグレースフルリスタートを有効にします。
redistribute kernel	NS(config-router)# redistribute kernel	カーネルルートを再配布します。

ISIS の設定

October 7, 2021

Citrix ADC アプライアンスは、Intermediate System-to-Intermediate System（IS-IS または ISIS）動的ルーティングプロトコルをサポートします。このプロトコルは、IPv4 および IPv6 ルート交換をサポートします。ISIS はリンクステートプロトコルであるため、ルーティングループが発生しにくくなっています。より高速なコンバージェンスとより大規模なネットワークをサポートする機能の利点により、ISIS はインターネットサービスプロバイダー（ISP）ネットワークで非常に役立ちます。

ISIS を構成するための前提条件

ISIS の構成を開始する前に、次のことを行ってください。

- ISIS プロトコルを理解していることを確認してください。
- IPV6 ルートでは、以下を有効にします。
  - IPv6 プロトコル変換機能。
  - ISIS プロトコルを実行する VLAN の IPv6 動的ルーティングオプション。

ISIS の有効化

次のいずれかの手順を使用して、Citrix ADC アプライアンスで ISIS ルーティング機能を有効にします。

CLI を使用して ISIS ルーティングを有効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

enable ns feature ISIS

GUI を使用して ISIS ルーティングを有効にするには、次の手順を実行します。

1. [システム] > [設定] に移動し、[モードと機能] で [高度な機能の変更] をクリックします。
2. ISIS ルーティングオプションを選択またはクリアします。

ISIS ルーティングプロセスを作成し、VLAN で開始する

ISIS ルーティングプロセスを作成するには、VTYSH コマンドラインを使用する必要があります。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードに移行します。

コマンド	説明
router ISIS [tag]	ルーティングプロセスの ISIS ルーティングプロセスおよび構成モードを作成します。
net XX….XXXX.YYYY.YYYY.YYYY.00	ルーティングプロセスの NET 値を指定します。ここで、・**XX**。**.. .XXXX** はエリアアドレス（1～13 バイト）、・**YYYY.YYYY.YYYY** はシステム ID（6 バイト）、・**00** は N セレクター（1 バイト）です。
is-type (level-1|level-1-2|level-2-only)	ISIS ルーティングプロセスを指定されたルーティングレベルに設定します。デフォルト：レベル 1-2。
ns IPv6-routing	IPv6 動的ルーティングデーモンを開始します。
interface <vlan_name>	VLAN 構成モードを開始します。
ip router ISIS	IPv4 ルート交換用の VLAN 上で ISIS ルーティングプロセスを有効にします。
ipv6 router ISIS	IPv6 ルート交換用の VLAN 上で ISIS ルーティングプロセスを有効にします。

例**:**

```
> VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# net 15.aabb.ccdd.0097.00
NS(config-router)# is-type level-1
NS(config-router)# exit
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)#  ip router isis 11
NS(config-if)# ipv6 router isis 11
<!--NeedCopy-->
```

アドバタイズルート

ルートアドバタイズメントにより、アップストリームルーターは Citrix ADC アプライアンスの背後にあるネットワークエンティティを追跡できます。

VTYSH コマンドラインを使用してルートをアドバタイズするように ISIS を設定するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードに移行します。
router ISIS [tag]	ISIS ルーティングインスタンスを起動し、ルーティングプロセスの構成モードを開始します。
redistribute connected (level-1 or level-1-2 or level-2)	接続されたルートを再配布します。ここで、レベル **1**: 接続されたルートをレベル 1、レベル **1-2** に再配布します: 接続されたルートをレベル 1 とレベル 2、レベル 2 に再配布します: 接続されたルートをレベル 2 に再配布します。
redistribute kernel (level-1 or level-1-2 or level-2)	カーネルルートを再配布します。ここで、レベル **1**: カーネルルートをレベル 1、レベル **1-2** に再配布します: カーネルルートをレベル 1 とレベル 2、レベル 2 に再配布します: カーネルルートをレベル 2 に再配布します。

例**:**

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# redistribute connected level-1
NS(config-router)# redistribute kernel level-1
<!--NeedCopy-->
```

ISIS の伝播を制限する

設定のトラブルシューティングが必要な場合は、任意の VLAN でリッスン専用モードを設定できます。

VTYSH コマンドラインを使用して ISIS の伝播を制限するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードに移行します。

コマンド	説明
router isis [tag]	ルーティングプロセスの構成モードを開始します。
passive-interface <vlan_name>	指定された VLAN にバインドされたインターフェイスでのルーティング更新を抑制します。

例:

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router isis 11
4 NS(config-router)# passive-interface VLAN0
5 <!--NeedCopy-->
```

ISIS 構成の確認

VTYSH を使用して、指定した VLAN の ISIS ルーティングテーブルと ISIS 情報を表示できます。

VTYSH コマンドラインを使用して ISIS 設定を表示するには:

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	説明
VTYSH	VTYSH コマンドプロンプトを表示します。
show ip isis route	更新された IPv4ISIS ルーティングテーブルを表示します。
show ipv6 isis route	更新された IPv6ISIS ルーティングテーブルを表示します。
sh isis interface <vlan_name>	指定された VLAN の IPv6ISIS 情報を表示します。

例:

```
1 NS# VTYSH
2 NS#  show ip isis route
3 NS#  show ipv6 isis route
4 NS# sh isis interface VLAN0
5 <!--NeedCopy-->
```


Citrix ADC ルーティングテーブルへのルートのインストール

October 7, 2021

Citrix ADC アプライアンスは、アプライアンスのルーティングテーブルにルートをインストールした後、さまざまなルーティングプロトコルで学習したルートを使用できます。

VTYSH コマンドラインを使用して、内部ルーティングテーブルにさまざまなルートをインストールするには、次の手順を実行します。

CLI で、インストールするルートに応じて次のコマンドを入力します。

コマンド	Specifies
VTYSH	VTYSH コマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
ns route-install Default	IPv4 デフォルトルートを内部ルーティングテーブルにインストールします。
ns route-install RIP	IPv4 RIP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install BGP	IPv4 BGP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install OSPF	IPv4 OSPF 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 Default	IPv6 デフォルトルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 RIP	IPv6 RIP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 BGP	IPv6 BGP 固有のルートを内部ルーティングテーブルにインストールします。
ns route-install IPv6 OSPF	IPv6 OSPF 固有のルートを内部ルーティングテーブルにインストールします。

例**:**


```
>VTYSH
NS# configure terminal
NS# ns route-install Default
NS(config)# ns route-install RIP
NS(config)# ns route-install BGP
NS(config)# ns route-install OSPF
NS# ns route-install IPv6 Default
NS(config)# ns route-install IPv6 RIP
NS(config)# ns route-install IPv6 BGP
NS(config)# ns route-install IPv6 OSPF
<!--NeedCopy-->
```

選択エリアへの **SNIP** および **VIP** ルートのアドバタイズ

October 7, 2021

一部の SNIP アドレスを選択エリアにアドバタイズするには、DRADV モードを有効にしたり、接続 ZebOS を再配布したりすることはできません。これは、これらの操作が接続されたすべてのルートを ZebOS に送信するためです。また、必要なサブネットにダミーのスタティックルートを ZebOS に追加したり、不要な接続ルートをフィルタリングするために ZebOS に ACL を追加したりすることは、面倒で面倒な作業です。

ネットワークルートとタグオプションは、この問題に対処します。ネットワークルートオプションは、サブネットごとに 1 つの SNIP アドレスに対してのみ有効にできます。その SNIP アドレスの接続ルートは、カーネルルートとして ZebOS に送信されます。

VIP アドレスと SNIP アドレスの場合、タグには 1 ～4294967295 の整数を割り当てることができます。このパラメータを設定できるのは、VIP アドレスまたは SNIP アドレスに対してホストルートまたはネットワークルートが有効になっている場合だけです。VIP アドレスと SNIP アドレスに関連付けられたタグ値は、そのルートとともに ZebOS に送信されます。VIP ルートと SNIP ルートには、異なる値を持つタグを設定できます。これらのタグ値は、ZebOS のルートマップで照合され、選択エリアにアドバタイズされます。

選択エリアへの **SNIP** ルートのアドバタイズ

CLI を使用して SNIP アドレスのネットワークルートおよびタグパラメータを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- 新しい SNIP アドレスを追加する場合:
  - **add ns ip** <IPAddress>@ <netmask> -**type SNIP** -**networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

- 既存の SNIP アドレスを再設定する場合は、次の手順を実行します。
  - **set ns ip** <IPAddress>@ <netmask> -**type SNIP** - **networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

GUI を使用して SNIP アドレスのネットワークルートおよびタグパラメータを設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [**IP] > [IPV4**] に移動します。
2. サブネット IP (SNIP) アドレスを追加したり、既存のサブネット IP アドレスを変更したりするときに、ネットワークルートと タグパラメータを設定します。

選択エリアへの **VIP** ルートのアドバタイズ

CLI を使用して VIP アドレスのホストルートおよびタグパラメータを設定するには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

- 新しい VIP アドレスを追加する場合:
  - **add ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>
- 既存の VIP アドレスを再構成する場合は、次の手順を実行します。
  - **set ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

GUI を使用して VIP アドレスのネットワークルートおよびタグパラメータを設定するには、次の手順を実行します。

1. **System** > **Network** > **IPs** > **IPV4s** に移動します。
2. VIP アドレスの追加時または既存の VIP アドレスの変更時に、**Host Route** および **Tag** パラメータを設定します。

双方向フォワーディング検出の設定

October 7, 2021

双方向フォワーディング検出（BFD）プロトコルは、フォワーディングパスの障害を迅速に検出するためのメカニズムです。BFD は、パス障害をミリ秒単位で検出します。BFD は、動的ルーティングプロトコルで使用されます。

BFD 動作では、ルーティングピアはネゴシエートされた間隔で BFD パケットを交換します。ネゴシエートされた間隔に猶予間隔を加えた範囲内にピアからパケットを受信しなかった場合、ピアは Dead と見なされ、登録されたルーティングプロトコルのセットに通知が送信されます。次に、ルーティングプロトコルは最適パスを再計算し、ルーテ

ィングテーブルを再プログラムします。BFD では、ルーティングプロトコルによって提供されるタイマーと比較して、より短い時間間隔がサポートされるため、障害の検出が高速になります。

Citrix ADC アプライアンスは、BGP（IPv4 および IPv6）、OSPFv2（IPv4）、および OSPFv3（IPv6）のルーティングプロトコルに対して BFD をサポートします。Citrix ADC アプライアンスの BFD サポートは、RFC5880、5881、および 5883 に準拠しています。

双方向フォワーディング検出の設定で考慮すべきポイント

BFD の設定を開始する前に、次の点を考慮してください。

- RFC 5880、5881、および 5883 で説明されている BFD のさまざまなコンポーネントについて理解していることを確認してください。
- Citrix ADC アプライアンス上の BFD は、次のルーティングプロトコルでサポートされています。
  - BGP（IPv4 および IPv6）
  - OSPFv2 (IPv4)
  - OSPFv3 (IPv6)
- Citrix ADC アプライアンスの BFD は、次のルーティングプロトコルではサポートされていません。
  - ISIS
  - RIP (IPv4)
  - RIPng (IPv6)
- Citrix ADC アプライアンスでは、次の BFD 機能はサポートされていません。
  - BFD Echo mode
  - BFD 認証
  - BFD デマンド非同期モード
- BFD 間隔および BFDRx タイマーの最小値は 100 ミリ秒です。
- 共有 IP アドレスを使用するトポロジで BFD を使用する場合（たとえば、SNIP アドレスを使用するレイヤー 2 高可用性セットアップ、またはストライプ IP アドレスを使用するクラスターセットアップ）、BFD 障害検出時間（順序）により、フェイルオーバー中にアクティブセッションが停止します。ミリ秒単位）は、HA フェイルオーバー検出間隔（3〜4 秒）よりも短いです。したがって、フェイルオーバー処理中にルートが保持されるため、レイヤ 2 の HA トポロジでは正常な再起動を使用することをお勧めします。

構成の手順

Citrix ADC アプライアンスでの BFD の構成は、次のタスクで構成されます。

- BFD パラメータの設定
- ダイナミックルーティングプロトコルに対する BFD サポートの設定

BFD パラメータの設定

Citrix ADC アプライアンスは、シングルホップセッション、IPv4 マルチホップセッション、IPv6 マルチホップセッション用に個別の BFD セッションパラメータを提供します。セッションタイプに BFD パラメータを設定しない場合、デフォルト値がそのセッションに適用されます。

各 BFD パラメータのデフォルト値は、シングルホップセッション、IPv4 マルチホップセッション、および IPv6 マルチホップセッションで同じです。次の表に、各 BFD パラメータのデフォルト値を示します。

BFD パラメータ名	デフォルト値
Interval	750 ミリ秒
最小 Rx	500 ミリ秒
乗数	3

> 重要:
>
> Citrix アプライアンスの MellanoxNIC は、初期化に約 1500 ミリ秒かかります。MellanoxNIC を搭載した Citrix ADC アプライアンスでは、BFD タイマーを 1500 ミリ秒以上に設定する必要があります。BFD タイマーを 3000 ミリ秒に設定することをお勧めします。
>
> - 間隔 Tx = 600 ミリ秒
> - 最小受信 = 600 ミリ秒
> - マルチプラー = 5

シングルホップセッションの **BFD** パラメータの設定

VTYSHコマンドラインを使用してシングルホップセッションの BFD パラメータを設定するには、コマンドプロンプトで次のコマンドを順番に入力します。

コマンド	Specifies
`vtysh`	VTYSHコマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。
`interface` `vlan ID>`	インターフェイス構成モードを開始します。
`bfd singlehop-peer interval <num> minrx <num> multiplier <num>`	指定したインターフェイスで BFD パラメータを設定します。

設定例:


```
> vtysh

ns# configure terminal

ns(config)# interface vlan3

ns(config-if)# bfd singlehop-peer interval 200 minrx 200 multiplier 5

ns(config-if)# exit
<!--NeedCopy-->
```

IPv4 マルチホップセッションの **BFD** パラメータの設定

VTYSHコマンドラインを使用して IPv4 マルチホップセッションの BFD パラメータを設定するには、コマンドプロンプトで次のコマンドを順番に入力します。

コマンド	Specifies
`vtysh`	VTYSHコマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。
`bfd multihop-peer <ipv4addr> interval <num> minrx <num> multiplier <num>`	IPv4 マルチホップセッションの BFD パラメータを設定します。

設定例:

```
    > vtysh

    ns# configure terminal

    ns(config)# bfd multihop-peer 20.20.20.138 interval 300 minrx 300
       multiplier 5

    ns(config)# exit
<!--NeedCopy-->
```


IPv6 マルチホップセッションの **BFD** パラメータの設定

VTYSHコマンドラインを使用して IPv6 マルチホップセッションの BFD パラメータを設定するには、コマンドプロンプトで次のコマンドを、表示された順序で入力します。

コマンド	Specifies
`vtysh`	VTYSHコマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。
`bfd multihop-peer ipv6 <ipv6addr> interval <num> minrx <num> multiplier <num>`	IPv6 マルチホップセッションの BFD パラメータを設定します。

設定例:

```
1      > vtysh
2
3      ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx
           500 multiplier 5
4
5      ns(config)# exit
6  <!--NeedCopy-->
```

ダイナミックルーティングプロトコルに対する **BFD** サポートの設定

ピアとのセッションタイプのダイナミックルーティングプロトコルに対して BFD を有効にできます。たとえば、シングルホップとマルチホップ。Citrix ADC アプライアンスは、関連する BFD パラメータ設定をセッションに適用します。

IPv4 BGP シングルホップセッションの **BFD** の設定

VTYSHコマンドラインを使用して IPv4 BGP シングルホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを次のように入力します。

コマンド	Specifies
`vtysh`	VTYSHコマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。

コマンド	Specifies
router bgp <asnumber>	BGP 自律システム。asnumber は必須パラメーターです。
neighbor <ipv4addr> remote-as <num>	指定した自律システム内のネイバーの IPv4 アドレスで IPv4 BGP テーブルを更新します。
neighbor <ipv4addr> fall-over bfd	指定されたネイバーの BFD を有効にします。

設定例:

```
    > vtysh

    ns# configure terminal

    ns(config)#router bgp 1

    ns(config-router)#neighbor 20.20.20.138 remote-as 1

    ns(config-router)#neighbor 20.20.20.138 fall-over bfd

    ns(config-router)#redistribute kernel

    ns(config-router)#exit
<!--NeedCopy-->
```

IPv4 BGP マルチホップセッションの **BFD** の設定

VTYSHコマンドラインを使用して IPv4 BGP マルチホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを次のように入力します。

コマンド	Specifies
vtysh	VTYSHコマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <asnumber>	BGP 自律システム。asnumber は必須パラメーターです。
neighbor <ipv4addr> remote-as <num>	指定した自律システム内のネイバーの IPv4 アドレスで IPv4 BGP テーブルを更新します。

コマンド	Specifies
neighbor <ipv4addr> fall-over bfd multihop	指定されたネイバーの BFD を有効にします。

設定例:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router bgp 1
6
7      ns(config-router)#neighbor 20.20.20.138 remote-as 1
8
9      ns(config-router)#neighbor 20.20.20.138 fall-over bfd multihop
10
11     ns(config-router)#redistribute kernel
12
13     ns(config-router)#exit
14 <!--NeedCopy-->
```

IPv6 BGP シングルホップセッションの **BFD** の設定

VTYSHコマンドラインを使用して IPv6 BGP シングルホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを次のように入力します。

コマンド	Specifies
vtysh	VTYSHコマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <asnumber>	BGP 自律システム。asnumber は必須パラメーターです。
neighbor <ipv6addr> remote-as <num>	指定した自律システム内のネイバーのリンクローカル IPv6 アドレスで IPv6 BGP テーブルを更新します。
neighbor <ipv6addr> fall-over bfd	指定されたネイバーの BFD を有効にします。
address-family ipv6	アドレスファミリ構成モードを開始します。

コマンド	Specifies
neighbor <ipv6addr> activate	リンクローカルアドレスを使用して、ピアとローカルノード間の IPv6 ルータファミリのプレフィクスを交換します。

設定例:

```
> vtysh

ns# configure terminal ns(config)#router bgp 1

ns(config-router)#neighbor 30fe:123::124 remote-as 1

ns(config-router)#neighbor 30fe:123::124 fall-over bfd

ns(config-router)#address-family ipv6

ns(config-router-af)#neighbor 30fe:123::124 activate

ns(config-router-af)#redistribute kernel

ns(config-router-af)#exit

<!--NeedCopy-->
```

IPv6 BGP マルチホップセッションの **BFD** の設定

VTYSHコマンドラインを使用して IPv6 BGP マルチホップセッションの BFD を設定するには、コマンドプロンプトで次のコマンドを、表示された順序で入力します。

コマンド	Specifies
vtysh	VTYSHコマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router bgp <asnumber>	BGP 自律システム。asnumber は必須パラメーターです。
neighbor <ipv6addr> remote-as <num>	指定した自律システム内のネイバーのリンクローカル IPv6 アドレスで IPv6 BGP テーブルを更新します。

コマンド	Specifies
neighbor <ipv6addr> fall-over bfd multihop	指定されたネイバーの BFD を有効にします。
address-family ipv6	アドレスファミリ構成モードを開始します。
neighbor <ipv6addr> activate	リンクローカルアドレスを使用して、ピアとローカルノード間の IPv6 ルータファミリのプレフィクスを交換します。

設定例:

```
> vtysh

ns# configure terminal

ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx 500
     multiplier 5

ns(config)#router bgp 1

ns(config-router)#neighbor 20fe:125::138 remote-as 1

ns(config-router)#neighbor 20fe:125::138 fall-over bfd multihop

ns(config-router)#address-family ipv6

ns(config-router-af)#neighbor 20fe:125::138 activate

ns(config-router-af)#redistribute kernel

ns(config-router-af)#end

<!--NeedCopy-->
```

インターフェイスでの **OSPFv2**（**IPv4**）の **BFD** の設定

BFD は、すべて、または OSPFv2 プロトコルを使用する特定のインターフェイスで有効にできます。

VTYSHコマンドラインを使用して、すべてのインターフェイスで **OSPFv2** の **BFD** を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
vtysh	VTYSHコマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ospf <process tag>	OSPFv2 構成モードを開始します。
bfd all-interfaces	OSPFv2 を使用するすべてのインターフェイスで BFD を有効にします。

設定例:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router ospf 1
6
7      ns(config-router)#bfd all-interfaces
8
9      ns(config-router)#redistribute kernel
10
11     ns(config-router)#exit
12 <!--NeedCopy-->
```

VTYSHコマンドラインを使用して、特定のインターフェイスで **OSPFv2** の **BFD** を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
vtysh	VTYSHコマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
interface <vlan ID>	インターフェイス構成モードを開始します。
ip ospf bfd	OSPFv2 を使用する指定されたインターフェイスで BFD を有効にします。

設定例:


```
    > vtysh

    ns# configure terminal

    ns(config)# interface vlan5

    ns(config-if)# ip ospf bfd

    ns(config-if)# exit
<!--NeedCopy-->
```

インターフェイスでの **OSPFv3**（**IPv6**）の **BFD** の設定

BFD は、すべて、または OSPFv3 プロトコルを使用する特定のインターフェイスで有効にできます。

VTYSHコマンドラインを使用して、すべてのインターフェイスで **OSPFv3** の **BFD** を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
vtysh	VTYSHコマンドプロンプトを表示します。
configure terminal	グローバル構成モードを開始します。
router ipv6 ospf <process tag>	OSPFv3 構成モードを開始します。
bfd all-interfaces	OSPFv3 を使用するすべてのインターフェイスで BFD を有効にします。

設定例:

```
    > vtysh

    ns# configure terminal

    ns(config)#router ipv6 ospf 10

    ns(config-router)#bfd all-interfaces

    ns(config-router)#redistribute kernel
```



```
10
11      ns(config-router)#exit
12  <!--NeedCopy-->
```

VTYSHコマンドラインを使用して、特定のインターフェイスで **OSPFv3** の **BFD** を設定するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを次の順序で入力します。

コマンド	Specifies
`vtysh`	VTYSHコマンドプロンプトを表示します。
`configure terminal`	グローバル構成モードを開始します。
`interface` <vlan ID>	インターフェイス構成モードを開始します。
`ipv6 ospf bfd`	OSPFv3 を使用する指定されたインターフェイスで BFD を有効にします。

設定例:

```
1       > vtysh
2
3       ns# configure terminal
4
5       ns(config)# interface vlan15
6
7       ns(config-if)# ipv6 ospf bfd
8
9       ns(config-if)# exit
10  <!--NeedCopy-->
```

スタティックルートの設定

October 7, 2021

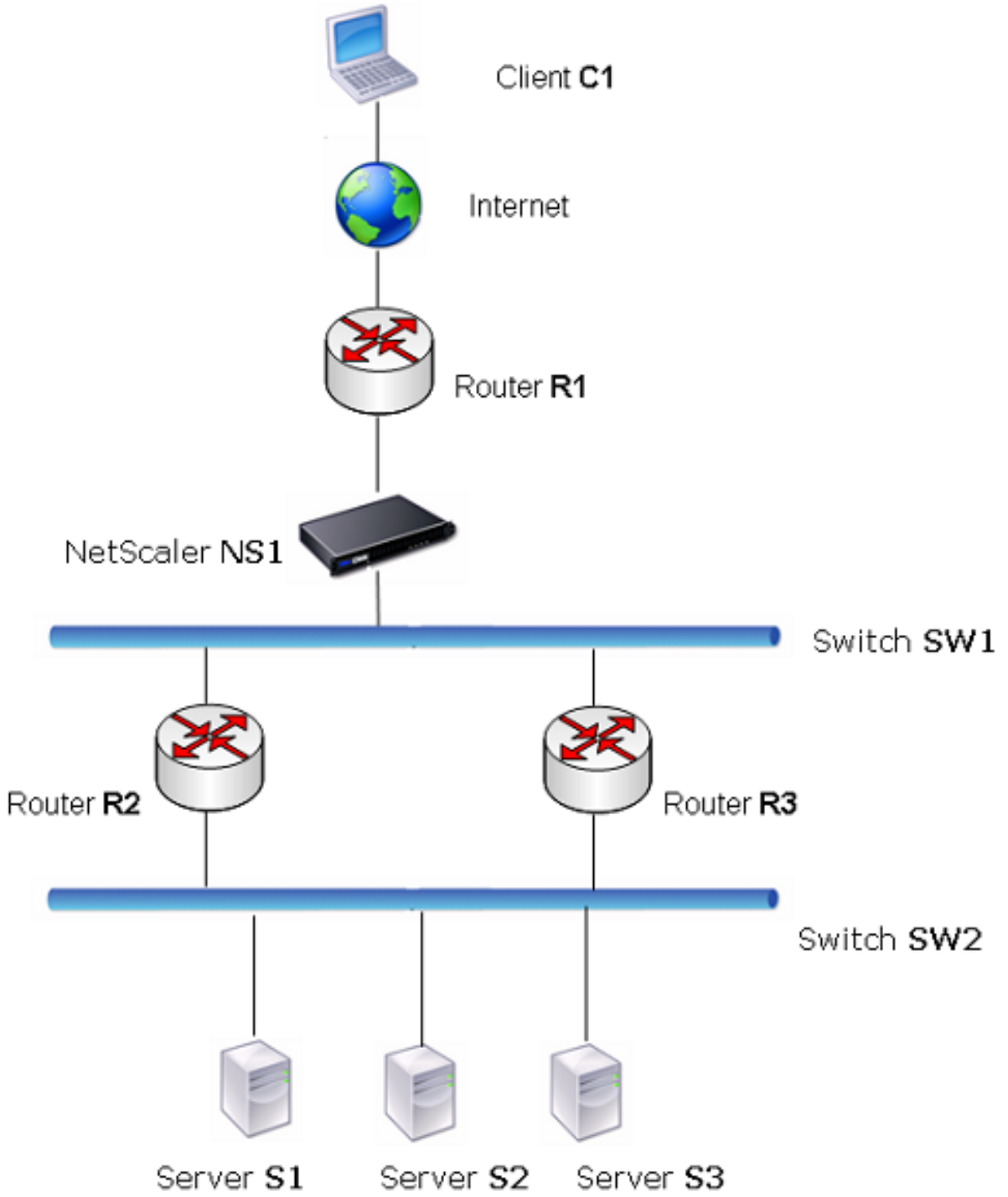
静的ルートは、ネットワークのパフォーマンスを向上させるために手動で作成されます。静的ルートを監視して、サービスの中断を回避できます。また、ECMP ルートに重みを割り当てたり、ヌルルートを作成してルーティングループを防止したりできます。

監視対象のスタティックルート。手動で作成された（静的）ルートがダウンした場合、バックアップルートは自動的にアクティブ化されません。非アクティブなプライマリスタティックルートを手動で削除する必要があります。ただし、静的ルートを監視対象ルートとして構成すると、Citrix ADC アプライアンスはバックアップルートを自動的にアクティブ化できます。

静的ルートの監視は、サブネットのアクセス可能性に基づくこともできます。サブネットは通常、単一のインターフェイスに接続されていますが、他のインターフェイスを介して論理的にアクセスできます。VLAN にバインドされたサブネットは、VLAN が稼働している場合にのみアクセスできます。VLAN は、Citrix ADC がパケットを送受信するための論理インターフェイスです。ネクストホップが到達不能なサブネット上にある場合、静的ルートは DOWN としてマークされます。

注：高可用性（HA）セットアップでは、セカンダリノードの監視対象状態ルート（MSR）のデフォルト値は UP です。この値は、フェイルオーバー時の状態遷移ギャップを回避するために設定されます。これにより、これらのルートでパケットがドロップされる可能性があります。

次の単純なトポロジについて考えてみます。このトポロジでは、Citrix ADC が複数のサーバーにまたがるサイトへのトラフィックを負荷分散しています。



ルーター R1 は、クライアントと Citrix ADC アプライアンス間でトラフィックを移動します。アプライアンスは、ルーター R2 または R3 を介してサーバー S1 および S2 に到達できます。サーバーのサブネットに到達するための 2 つ

の静的ルートがあります。1 つはゲートウェイとして R2 を使用し、もう 1 つはゲートウェイとして R3 を使用します。両方のルートでモニタリングが有効になっています。ゲートウェイ R2 を使用したスタティックルートのアドミニストレーティブディスタンスは、ゲートウェイ R3 を使用したスタティックルートのアドミニストレーティブディスタンスよりも短くなっています。したがって、サーバーにトラフィックを転送するには、R3 よりも R2 が優先されます。また、Citrix ADC のデフォルトルートは R1 を指しているため、すべてのインターネットトラフィックが適切に終了します。

R2 をゲートウェイとして使用する静的ルートで監視が有効になっているときに R2 に障害が発生した場合、Citrix ADC はそれを DOWN としてマークします。Citrix ADC は、ゲートウェイとして R3 を使用する静的ルートを使用し、R3 を介してトラフィックをサーバーに転送するようになりました。

Citrix ADC は、IPv4 および IPv6 静的ルートの監視をサポートしています。新しい ARP または PING モニターを作成するか、既存の ARP または PING モニターを使用して、IPv4 静的ルートを監視するように Citrix ADC を構成できます。IPv6（ND6）または PING モニターの新しい近隣探索を作成するか、既存の ND6 または PING モニターを使用して、IPv6 静的ルートを監視するように Citrix ADC を構成できます。

重み付けされたスタティックルート。Citrix ADC アプライアンスは、距離とコストが等しいルート、つまりコストが等しいマルチパス（ECMP）ルートを含むルーティング決定を行う場合、送信元 IP アドレスと宛先 IP アドレスに基づくハッシュメカニズムを使用して、ルート間の負荷を分散します。ただし、ECMP ルートの場合、重み値を構成できます。次に、Citrix ADC は、負荷を分散するために重みとハッシュ値の両方を使用します。

NULL ルート。ルーティング決定で選択されたルートが非アクティブの場合、Citrix ADC アプライアンスはバックアップルートを選択します。すべてのバックアップルートにアクセスできなくなると、アプライアンスはパケットを送信者に再ルーティングする可能性があり、その結果、ルーティングループが発生してネットワークが輻輳する可能性があります。このような状況を回避するには、NULL ルートを作成して、NULL インターフェイスを Gateway として追加します。NULL ルートは、他のスタティックルートよりもアドミニストレーティブディスタンスが高いため、優先ルートではありません。ただし、他の静的ルートにアクセスできなくなった場合に選択されます。その場合、アプライアンスはパケットをドロップし、ルーティングループを防ぎます。

IPv4 静的ルートの構成

いくつかのパラメータを設定することで、単純なスタティックルートまたは NULL ルートを追加できます。また、追加のパラメータを設定して、モニタ対象またはモニタ対象および重み付けスタティックルートを設定することもできます。静的ルートのパラメータを変更できます。たとえば、重み付けされていないルートに重みを割り当てたり、監視されているルートでモニタリングを無効にしたりできます。

CLI のプロシージャ

CLI を使用して静的ルートを作成するには:

コマンドプロンプトで入力します。

- add route <network> <netmask> <gateway>[-cost <positive_integer>] [-advertise (DISABLED | ENABLED)]

- show route [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

例:

```
1 > add route 10.102.29.0 255.255.255.0 10.102.29.2 -cost 2 -advertise
     ENABLED
2  Done
3 <!--NeedCopy-->
```

CLI を使用して監視対象の静的ルートを作成するには:

コマンドプロンプトで次のコマンドを入力して、監視対象のスタティックルートを作成し、設定を確認します。

- add route <network> <netmask> <gateway> [-distance <positive_integer>] [-weight <positive_integer>][-msr (ENABLED | DISABLED) [-monitor <string>]]
- ルートを表示 [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

例:

```
1 > add route 10.102.29.0 255.255.255.0 10.102.29.3 -distance 5 -weight 6
     -msr ENBLED -monitor PING
2  Done
3 <!--NeedCopy-->
```

CLI を使用して NULL ルートを作成するには、次の手順を実行します。

コマンドプロンプトで次のように入力します。

- add route <network> <netmask> null
- show route <network> <netmask>

例:

```
1 > add route 10.102.29.0 255.255.255.0 null
2  Done
3 <!--NeedCopy-->
```

CLI を使用して静的ルートを削除するには:

コマンドプロンプトで入力します。

rm route <network> <netmask> <gateway>

例:


```
> rm route 10.102.29.0 255.255.255.0 10.102.29.3
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して静的ルートを構成するには:

システムに移動 > 通信網 > ルートと、[基本] タブで、新しい静的ルートを追加するか、既存の静的ルートを編集します。

GUI を使用してルートを削除するには:

システムに移動 > 通信網 > ルートを選択し、[基本] タブで静的ルートを削除します。

IPv6 静的ルートの構成

最大 6 つのデフォルト IPv6 スタティックルートを設定できます。IPv6 ルートは、宛先デバイスの MAC アドレスが到達可能かどうかに基づいて選択されます。これは、IPv6 近隣探索機能を使用して判別できます。ルートは負荷分散されており、source/destination-based ハッシュメカニズムが使用されます。したがって、ラウンドロビンなどのルート選択メカニズムはサポートされていません。デフォルトルートのネクストホップアドレスは、NSIP サブネットに属している必要はありません。

CLI のプロシージャ

CLI を使用して IPv6 ルートを作成するには:

コマンドプロンプトで、次のコマンドを入力して IPv6 ルートを作成し、構成を確認します。

- add route6 <network> <gateway> [-vlan <positive_integer>]
- show route6 [<network> [<gateway>]

例**:**

```
> add route6 ::/0 FE80::67 -vlan 5
 Done
<!--NeedCopy-->
```

CLI を使用して監視対象の IPv6 静的ルートを作成するには:

コマンドプロンプトで、次のコマンドを入力して、監視対象の IPv6 静的ルートを作成し、構成を確認します。

- add route6 <network> <gateway> [-msr (ENABLED | DISABLED) [-monitor <string>]

- show route6 [<network> [<gateway>]

例**:**

```
> add route6 ::/0 2004::1 -msr ENABLED -monitor PING
 Done
<!--NeedCopy-->
```

CLI を使用して IPv6 ルートを削除するには:

コマンドプロンプトで入力します。

rm route6 <network> <gateway>

例**:**

```
> rm route6 ::/0 FE80::67
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して IPv6 ルートを構成するには:

システムに移動 > 通信網 > ルートと、[IPV6] タブで、新しい IPv6 ルートを追加するか、既存の IPv6 ルートを編集します。

GUI を使用して IPv6 ルートを削除するには:

システムに移動 > 通信網 > ルートを選択し、[IPV6] タブで IPv6 ルートを削除します。

仮想サーバ設定に基づくルートヘルスインジェクション

October 7, 2021

次のオプションとパラメーターは、VIP アドレスのルートをアドバタイズするための Citrix ADC アプライアンスのルートヘルスインジェクション（RHI）機能を制御するために導入されています。

- **VSVR_CNTRLD**。これは、VIP アドレスの（Vserver RHI レベル）パラメーターのオプションです。このオプションが VserverRHI Level パラメーターに設定されている場合、VIP アドレスのルートをアドバタイズするための RHI の動作は、VIP アドレスの関連するすべての仮想サーバーの RHISTATE パラメーター設定とその状態によって異なります。

- **RHI STATE**。仮想サーバーのパラメータです。RHISTATE パラメーターは PASSIVE または ACTIVE のいずれかに設定できます。デフォルトでは、RHISTATE パラメーターは PASSIVE に設定されています。

VIP アドレスの場合、RHI（Vserver RHI Level）パラメーターがに設定されている場合 VSVR_CNTRLD, 以下は、VIP アドレスに関連付けられた仮想サーバーの RHISTATE 設定に基づく VIP アドレスのさまざまな RHI 動作です。

- すべての仮想サーバーで RHISTATE を PASSIVE に設定すると、Citrix ADC は常に VIP アドレスのルートをアドバタイズします。
- すべての仮想サーバーで RHISTATE を ACTIVE に設定した場合、関連付けられた仮想サーバーの少なくとも 1 つが UP 状態であれば、Citrix ADC は VIP アドレスのルートをアドバタイズします。
- 一部で RHISTATE を ACTIVE に設定し、他で PASSIVE に設定した場合、RHI STATE が ACTIVE に設定されている関連する仮想サーバーの少なくとも 1 つが UP 状態であると、Citrix ADC は VIP アドレスのルートをアドバタイズします。

次の表は、VIP アドレスに関連付けられた仮想サーバーの RHI STATE 設定に基づいた、VIP アドレスの RHI 動作の例を示しています。Citrix ADC アプライアンスには、VIP アドレスに関連付けられた 2 つの仮想サーバー V1 と V2 があります。

VIP の関連付けられた仮想サーバー	状態 1	状態 2	状態 3	状態 4
すべての仮想サーバーで **RHI** 状態がパッシブに設定されている				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP アドレスのルートをアドバタイズしますか?	はい	はい	はい	はい
すべての仮想サーバーで **RHI** 状態が **ACTIVE** に設定されている				
V1	上へ	上へ	DOWN	DOWN
V2	上へ	DOWN	上へ	DOWN
この VIP アドレスのルートをアドバタイズしますか?	はい	はい	はい	いいえ

VIP の関連付けられた仮想サーバー	状態 1	状態 2	状態 3	状態 4
RHI 状態は、一方の仮想サーバーではアクティブに設定され、もう一方の仮想サーバーではパッシブに設定されます				
V1 (RHI State = ACTIVE)	上へ	上へ	DOWN	DOWN
V2 (RHI State = PASSIVE)	上へ	DOWN	上へ	DOWN
この VIP アドレスのルートをアドバタイズしますか?	はい	はい	いいえ	いいえ

関連する仮想サーバーの RHI（RHI 状態）パラメーター設定に基づいて VIP アドレスの RHI を構成するには、次の手順を実行します。

- RHI（Vserver RHI Level）パラメーターをに設定します VSVR_CNTRLD VIP アドレス用。
- VIP アドレスに関連付けられた各仮想サーバーの RHI 状態パラメーターを設定します。

CLI を使用して VIP アドレスの vServerRHI レベルを設定するには:

コマンドプロンプトで入力します。

- **set ns ip** <IPAddress> [-**vserverRHILevel** <vserverRHILevel>]

CLI を使用して仮想サーバの RHI State パラメータを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set lb vserver** <name> [-**RHIstate** (**PASSIVE** | **ACTIVE**)]

GUI を使用して VIP アドレスの vServer RHI レベルを設定するには

1. **System** > **Network** > **IPs** に移動します。
2. VIP アドレスを選択し、[編集] をクリックします。
3. **Vserver** の **RHI** レベルパラメーターを **VSVR_CNTRLD** に設定し、[**OK**] をクリックします。

GUI を使用して仮想サーバーの RHI 状態パラメーターを設定するには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 負荷分散仮想サーバーを選択し、[編集] をクリックします。

3. [**RHI** 状態] パラメータを設定し、[**OK**] をクリックします。

ポリシーベースルートの設定

October 7, 2021

ポリシーベースルーティングは、指定した基準に基づいてルーティングを決定します。ポリシーベースルート（PBR）は、パケットを選択するための基準を指定し、通常、選択したパケットの送信先となるネクストホップを指定します。たとえば、特定の IP アドレスまたは範囲から特定のネクストホップルーターに発信パケットをルーティングするように Citrix ADC アプライアンスを構成できます。各パケットは、一致が見つかるまで、指定された優先順位によって決定された順序で、構成された各 PBR と照合されます。一致するものが見つからない場合、または一致する PBR が DENY アクションを指定している場合、Citrix ADC はルーティングテーブルを通常の宛先ベースのルーティングに適用します。

PBR は、送信元 IP アドレス、送信元ポート、宛先 IP アドレス、宛先ポート、プロトコル、送信元 MAC アドレスなどのパラメーターに基づいてデータパケットのルーティングを決定します。PBR は、Citrix ADC がパケットをルーティングするためにパケットが満たさなければならない条件を定義します。これらのアクションは、「処理モード」と呼ばれます。処理モードは次のとおりです。

- **ALLOW**。アプライアンスは、指定されたネクストホップルーターにパケットを送信します。
- **DENY**。Citrix ADC は、通常の宛先ベースのルーティングにルーティングテーブルを適用します。

発信 IPv4 および IPv6 トラフィック用の PBR を作成できます。

多くのユーザーは、まず PBR を作成し、次にそれらを変更します。新しい PBR をアクティブ化するには、それを適用する必要があります。PBR を非アクティブ化するには、削除または無効にできます。PBR の優先順位番号を変更して、優先順位を高くしたり低くしたりできます。

IPv4 トラフィック用のポリシーベースルート（**PBR**）

October 7, 2021

PBR の設定には、次の作業が含まれます。

- PBR を作成します。
- PBR を適用します。
- （任意）PBR を無効または有効にします。
- （任意）PBR のプライオリティを再番号付けします。

PBR の作成または変更

同じパラメータで 2 つの PBR を作成することはできません。複製を作成しようとすると、エラーメッセージが表示されます。

PBR のプライオリティを設定できます。優先度（整数）は、Citrix ADC アプライアンスが PBR を評価する順序を定義します。優先順位を指定せずに PBR を作成すると、Citrix ADC は 10 の倍数の優先順位を自動的に割り当てます。

パケットが PBR で定義された条件と一致する場合、Citrix ADC はアクションを実行します。パケットが PBR で定義された条件と一致しない場合、Citrix ADC は次に高い優先順位を持つ PBR とパケットを比較します。

選択したパケットをネクストホップルータに送信する代わりに、複数のネクストホップをバインドしたリンクロードバランシング仮想サーバに送信するように PBR を設定できます。この設定は、ネクストホップリンクに障害が発生した場合にバックアップを提供できます。

次の例を考えてみましょう。2 つの PBR（p1 と p2）が Citrix ADC 上で構成され、優先順位 20 と 30 が自動的に割り当てられます。最初の PBR p1 の直後に評価される 3 番目の PBR p3 を追加する必要があります。新しい PBR p3 のプライオリティは 20 ～30 である必要があります。この場合、プライオリティを 25 に指定できます。

CLI のプロシージャ

CLI を使用して PBR を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>] <destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> |-protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]
- show ns pbr

例:

```
> add ns pbr pbr1  allow -srcip 10.102.37.252 -destip 10.10.10.2 -
    nexthop 10.102.29.77
 Done
<!--NeedCopy-->
```

CLI を使用して PBR のプライオリティを変更するには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、優先度を変更し、構成を確認します。

- set ns pbr <name> [-action (ALLOW | DENY)] [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>]

<destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> | -protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]

- show ns pbr [<name>]

例:

```
> set ns pbr pbr1 -priority 23
 Done
<!--NeedCopy-->
```

CLI を使用して 1 つまたはすべての PBR を削除するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- rm ns pbr <name>
- clear ns pbrs

例:

```
> rm ns pbr pbr1
 Done

> clear ns PBRs
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して PBR を作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブで、新しい PBR を追加するか、既存の PBR を編集します。

GUI を使用して 1 つまたはすべての PBR を削除するには、次の手順を実行します。

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブで PBR を削除します。

PBR の適用

PBR をアクティブにするには、PBR を適用する必要があります。次の手順では、無効にしていないすべての PBR を再適用します。PBR は、メモリツリー（ルックアップテーブル）を構成します。たとえば、10 個の PBR (p1-p10) を

作成し、別の PBR (p11) を作成して適用すると、すべての PBR (p1-p11) が新しく適用され、新しいルックアップテーブルが作成されます。セッションに DENY PBR が関連付けられている場合、セッションは破棄されます。

PBR に変更を加えるたびに、この手順を適用する必要があります。たとえば、PBR を無効にした後でこの手順を実行する必要があります。

注： Citrix ADC アプライアンスで作成された PBR は、適用されるまで機能しません。

CLI を使用して PBR を適用するには、次の手順を実行します。

コマンドプロンプトで入力します。

PBR の適用

GUI を使用して PBR を適用するには、次の手順を実行します。

1. System > Network > PBRs に移動します。
2. [PBR] タブで PBR を選択し、[アクション] リストで [適用] を選択します。

PBR の有効化または無効化

デフォルトでは、PBR は有効になっています。つまり、PBR が適用されると、Citrix ADC アプライアンスは着信パケットと構成済みの PBR を自動的に比較します。PBR がルックアップテーブルで必要ないが、構成内に保持する必要がある場合は、PBR を適用する前に無効にする必要があります。PBR が適用されると、Citrix ADC は着信パケットを無効にした PBR と比較しません。

CLI を使用して PBR を有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- enable ns pbr <name>
- nspbr を無効にする < 名前 >

例:

```
> enable ns PBR pbr1
 Done
> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                           Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                 Protocol:
        Vlan:                                   Interface:
        Active Status: ENABLED                  Applied Status: APPLIED
        Priority: 10
        NextHop: 10.102.29.77
```



```
13
14  Done
15
16 > disable ns PBR pbr1
17 Warning: PBR modified, use 'apply pbrs' to commit this operation
18
19 > apply pbrs
20  Done
21
22 > show ns PBR pbr1
23 1)      Name: pbr1
24         Action: ALLOW                          Hits: 0
25         srcIP = 10.102.37.252
26         destIP = 10.10.10.2
27         srcMac:                                Protocol:
28         Vlan:                                  Interface:
29         Active Status: DISABLED                Applied Status:
              NOTAPPLIED
30         Priority: 10
31         NextHop: 10.102.29.77
32 Done
33 <!--NeedCopy-->
```

GUI を使用して PBR を有効または無効にするには、次の手順を実行します。

1. System > Network > PBRs に移動します。
2. [PBR] タブで PBR を選択し、[アクション] リストで [有効] または [無効] を選択します。

PBR の再番号付け

PBR の番号を自動的に変更して、優先順位を 10 の倍数に設定できます。

CLI を使用して PBR の番号を変更するには、次の手順を実行します。

コマンドプロンプトで入力します。

- renumber ns pbrs

GUI を使用して PBR の番号を変更するには、次の手順を実行します。

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブの [アクション] リストで [優先順位の再番号付け] を選択します。

ユースケース-複数のホップを持つ PBR

2 つの PBR（PBR1 と PBR2）が Citrix ADC アプライアンス NS1 上で構成されているシナリオを考えてみましょう。PBR1 は、送信元 IP アドレスが 10.102.29.30 であるすべての発信パケットをネクストホップルータ R1 にルーティ

ングします。PBR2 は、送信元 IP アドレスが 10.102.29.90 であるすべての発信パケットをネクストホップルータ R2 にルーティングします。R3 は、NS1 に接続されたもう 1 つのネクストホップルータです。

ルータ R1 に障害が発生すると、PBR1 と照合されたすべての発信パケットがドロップされます。このような状況を回避するには、PBR の作成または変更中に、ネクストホップフィールドで Link Load Balancing（LLB; リンクロードバランシング）仮想サーバを指定します。複数のネクストホップは、サービスとして LLB 仮想サーバにバインドされます（たとえば、R1、R2、R3）。これで、R1 が失敗した場合、PBR1 と照合されたすべてのパケットが、LLB 仮想サーバに設定された LB 方式によって決定された R2 または R3 にルーティングされます。

次の場合に LLB 仮想サーバーをネクストホップとして持つ PBR を作成しようとすると、Citrix ADC アプライアンスがエラーをスローします。

- 同じ LLB 仮想サーバで別の PBR を追加する。
- 存在しない LLB 仮想サーバを指定する。
- バインドされたサービスがネクストホップではない LLB 仮想サーバの指定。
- LB 方式が次のいずれかに設定されていない LLB 仮想サーバを指定する。
  - ROUNDROBIN
  - DESTINATIONIPHASH
  - SOURCEIPHASH
  - SRCIPDESTIPHASH
  - LEASTPACKETS
  - LEASTBANDWIDTH
  - LTRM
  - CALLIDHASH
  - CUSTOM LOAD
- LB パーシステンスタイプが次のいずれかに設定されていない LLB 仮想サーバーの指定。
  - DESTIP
  - SOURCEIP
  - SRCDSTIP

次の表に、Citrix ADC アプライアンス上で構成されているエンティティの名前と値を示します。

エンティティの種類	名前	IP アドレス
リンク・ロード・バランシング仮想サーバ	LLB1	-
サービス（ネクストホップ）	Router1	1.1.1.254
	Router2	2.2.2.254
	Router3	3.3.3.254
PBR	PBR1	-
	PBR2	-

表 1. エンティティ作成のサンプル値

上記の設定を実装するには、以下を行う必要があります。

1. ネクストホップルータ R1、R2、および R3 を表すサービスルータ 1、ルータ 2、およびルータ 3 を作成します。
2. リンクロードバランシング仮想サーバ LLB1 を作成し、サービスルータ 1、ルータ 2、およびルータ 3 をバインドします。
3. PBR1 および PBR2 を作成し、ネクストホップフィールドを LLB1 および 2.2.254（ルータ R2 の IP アドレス）に設定します。

CLI を使用してサービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add service <name> <IP> <serviceType> <port>
- show service <name>

例:

```
> add service Router1 1.1.1.254 ANY *
 Done
> add service Router2 2.2.2.254 ANY *
 Done
> add service Router3 3.3.3.254 ANY *
 Done
<!--NeedCopy-->
```

GUI を使用してサービスを作成するには、次のステップを実行します。

Traffic Management > Load Balancing > Services に移動してサービスを作成します。

CLI を使用してリンクロードバランシング仮想サーバを作成し、サービスをバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

例:

```
> add lb vserver LLB1 ANY
 Done
> bind lb vserver LLB1 Router1 Router2 Router3
 Done
```



```
<!--NeedCopy-->
```

リンク・ロード・バランシング仮想サーバーを作成し、GUI を使用してサービスをバインドするには、次の手順を実行します。

1. トラフィック管理に移動します > 負荷分散 > 仮想サーバー、およびリンク負荷分散用の仮想サーバーを作成します。[プロトコル] フィールドに **ANY** を指定します。
   注:[
   直接アドレス可能] がオフになっていることを確認します。
2. [サービス] タブの [アクティブ] 列で、仮想サーバーにバインドするサービスのチェックボックスをオンにします。

CLI を使用して PBR を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-nextHop <nextHopVal>]
- show ns pbr

例**:**

```
> add pbr PBR1 ALLOW -srcIP 10.102.29.30 -nextHop LLB1
 Done
> add pbr PBR2 ALLOW -srcIP 10.102.29.90 -nextHop 2.2.2.254
 Done
<!--NeedCopy-->
```

GUI を使用して PBR を作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [PBR] に移動し、[PBR] タブで新しい PBR を追加します。

IPv6 トラフィックのポリシーベースルート（PBR6）

October 7, 2021

PBR6 の設定には、次の作業が含まれます。

- PBR6 を作成します。
- PBR6 を適用します。
- （任意）PBR6 を無効または有効にします。
- （任意）PBR6 のプライオリティを再番号付けします。

PBR6 の作成または変更

同じパラメータで 2 つの PBR6 を作成することはできません。複製を作成しようとすると、エラーメッセージが表示されます。

PBR6 のプライオリティを設定できます。優先度（整数）は、Citrix ADC アプライアンスが PBR6 を評価する順序を定義します。優先順位を指定せずに PBR6 を作成すると、Citrix ADC は 10 の倍数の優先順位を自動的に割り当てます。

パケットが PBR6 で定義された条件と一致する場合、Citrix ADC はアクションを実行します。パケットが PBR6 で定義された条件と一致しない場合、Citrix ADC は次に高い優先順位を持つ PBR6 とパケットを比較します。

CLI のプロシージャ

CLI を使用して PBR6 を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns pbr6** <name> <action> [-srcIPv6 [<operator>] <srcIPv6Val>] [-srcPort [<operator>] <srcPortVal>] [-destIPv6 [<operator>] <destIPv6Val>] [-destPort [<operator>] <destPortVal>] [-srcMac <mac_addr>] [-protocol <protocol> |-protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-state (ENABLED | DISABLED)] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-nextHop <nextHopVal>] [-nextHopVlan <positive_integer>]
- show ns pbr

CLI を使用して PBR6 を変更または削除するには、次の手順を実行します。

PBR6 を変更するには、**set pbr6** <name> コマンドと、変更するパラメータと新しい値を入力します。

CLI を使用して 1 つまたはすべての PBR6 を削除するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **rm ns pbr6** <name>
- **clear ns pbr6**

GUI のプロシージャ

GUI を使用して PBR6 を作成または変更するには、次の手順を実行します。

[システム] > [ネットワーク] > [PBR] に移動し、[PBR6s] タブで新しい PBR6 を追加するか、既存の PBR6 を編集します。

GUI を使用して 1 つまたはすべての PBR6 を削除するには、次の手順を実行します。

[システム] > [ネットワーク] > [PBR] に移動し、[PBR6s] タブで PBR6 を削除します。

PBR6 の適用

PBR6 をアクティブにするには、PBR6 を適用する必要があります。次の手順では、無効にしていないすべての PBR6 を再適用します。PBR6 は、メモリツリー（ルックアップテーブル）を構成します。たとえば、10 個の PBR6 (p6_1-p6_10) を作成し、別の PBR6 (p6_11) を作成して適用すると、すべての PBR6 (p6_1-p6_11) が新しく適用され、新しいルックアップテーブルが作成されます。セッションに DENY PBR6 が関連付けられている場合、セッションは破棄されます。

PBR6 に変更を加えるたびに、この手順を適用する必要があります。たとえば、PBR6 を無効にした後でこの手順を実行する必要があります。

注： Citrix ADC アプライアンス上に作成された PBR6 は、適用されるまで機能しません。

CLI を使用して PBR6 を適用するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **apply ns PBR6**

GUI を使用して PBR6 を適用するには、次の手順を実行します。

1. System > Network > PBRs に移動します。
2. [PBR6s] タブで PBR6 を選択し、[アクション] リストで [適用] を選択します。

PBR6 の有効化または無効化

デフォルトでは、PBR6 は有効になっています。つまり、PBR6 が適用されると、Citrix ADC アプライアンスは送信 IPv6 パケットと構成済みの PBR6 を自動的に比較します。PBR6 がルックアップテーブルで必要ないが、構成内に保持する必要がある場合は、PBR6 を適用する前に無効にする必要があります。PBR6 が適用されると、Citrix ADC は着信パケットを無効にした PBR6 と比較しません。

CLI を使用して PBR6 を有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- **enable ns pbr** <name>
- **nspbr** を無効にする < 名前 >

GUI を使用して PBR6 を有効または無効にするには、次の手順を実行します。

1. System > Network > PBRs に移動します。
2. [PBR6s] タブで PBR6 を選択し、[アクション] リストで [有効] または [無効] を選択します。

PBR6 の再番号付け

PBR6 の番号を自動的に変更して、優先順位を 10 の倍数に設定できます。

CLI を使用して PBR6 の番号を再設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **renumber ns pbr6**

GUI を使用して PBR6 の番号を変更するには、次の手順を実行します。

[システム] > [ネットワーク] > [PBR] に移動し、[PBR6s] タブの [アクション] リストで [優先順位の再番号付け] を選択します。

PBR の MAC アドレスワイルドカードマスク

October 7, 2021

拡張 PBR および PBR6 にワイルドカードマスクパラメータが導入され、送信元 MAC アドレスパラメータとともに使用され、発信パケットの送信元 MAC アドレスと照合する MAC アドレスの範囲を定義します。

ワイルドカードマスクは、使用する MAC アドレスの 16 進数と、無視する 16 進数を指定します。ワイルドカードマスクパラメータは、一連の 1 と 0 を指定し、長さは 12 桁です。各桁は、MAC アドレスの対応する 16 進数のマスクです。ワイルドカードマスクのゼロ桁は、MAC アドレスの対応する 16 進数を考慮する必要があることを示し、1 桁は対応する 16 進数を無視することを示します。

ワイルドカードマスクは、次の条件を満たす必要があります。

- ゼロのシリーズが 1 つしかありません
- 1 つのシリーズしか持たない
- 一連のゼロから始める

有効なワイルドカードマスクの例を次に示します。

- 000000111111
- 000000011111
- 000011111111

無効なワイルドカードマスクの例を次に示します。

- 000000111100
- 111110000000
- 010101010101

PBR ルールの場合、MAC アドレス 96: fa: 95:1 d: 67:4 のワイルドカードマスクは 0000001111 であり、MAC アドレス範囲 96:FA: 95:00:00-96: FA: 95: FF: FF: FF: FF を定義します。この MAC アドレス範囲は、発信パケットの送信元 MAC アドレスと照合されます。

CLI を使用して PBR ルール内の送信元 MAC アドレスの範囲を指定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns pbr** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns pbr** <pbrname>

例:

```
> add ns pbr PBR-1 ALLOW -srcip 192.0.2.34 -srcMac 96:fa:95:1d:67:4a
    - srcMacMask 000000111111 -nexthop 198.51.100.1

 Done
```

CLI を使用して PBR6 ルール内の送信元 MAC アドレスの範囲を指定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns pbr6** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show pbr6** <pbr6name>

例:

```
> add ns pbr6 PBR6-1 ALLOW  - srcipv6 2001:db8:0::7 -srcMac 96:fa:95:1d
   :67:4a   - srcMacMask 000000001111 -nexthop 2001:db8:0::1
 Done
```

NULL ポリシーベースのルートを使用した発信パケットのドロップ

October 7, 2021

場合によっては、Citrix ADC アプライアンスが特定の送信パケットをルーティングするのではなく、特定の送信パケットをドロップするように要求することがあります。たとえば、テストケースや展開の移行中などです。

NULL ポリシーベースのルートを使用して、特定の発信パケットをドロップできます。NULL PBR は、nexthop パラメータが NULL に設定されている PBR のタイプです。Citrix ADC アプライアンスは、NULL PBR と一致する発信パケットをドロップします。

IPv4 パケットの NULL PBR の設定

CLI を使用して NULL PBR を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns pbr** <name> ALLOW [**-td** <positive_integer>] [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] (-**nextHop NULL**) [**srcMac** <mac_addr> [-**srcMacMask** <string>]] [-**protocol** <protocol> | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer> | -**vxlan**

<positive_integer>] [-**interface** <interface_name>] [-**priority** <positive_integer>] [-**msr** (**ENABLED** | **DISABLED**) [-**monitor** <string>]] [-**state** (**ENABLED** | **DISABLED**)[-**ownerGroup** <string>]

- **apply ns pbrs**
- **show ns pbr**<id>

GUI を使用して NULL PBR を設定するには、次の手順を実行します。

[システム] > [ネットワーク] > [**PBR**] に移動し、[**PBR**] タブで、新しい **NULL PBR** を追加するか、既存の NULL PBR を編集します。

構成例

次の設定例では、インターフェイス 1/5 からの発信 IPv6 パケットをドロップするために、NULL PBR6 PBR6-NULL-EXAMPLE-1 が設定されています。

```
> add ns pbr PBR6-NULL-EXAMPLE-1 ALLOW  -nextHop NULL -interface 1/5
 Done

> apply ns pbr6
 Done
```

5 つのタプル情報に基づく複数のルートでのトラフィック分布

October 7, 2021

負荷分散のセットアップでは、Citrix ADC アプライアンスは、パケットを宛先に送信するための複数のルートを持つことができます。例：サーバーへ、およびクライアントへ。

Citrix ADC アプライアンスは、ハッシュアルゴリズムを使用して、パケットを宛先に送信するためのルートを選択します。

ハッシュアルゴリズムは、パケットの次の 2 つのタプルを使用してハッシュを計算し、それに基づいて Citrix ADC アプライアンスがパケットのルートを選択します。

- 送信元 IP アドレス
- 宛先 IP アドレス

2 つのタプル情報に基づいてルートを選択すると、使用可能なルートでトラフィックが不均一に分散される可能性があります。このトラフィックの不均一な分布は、一部のルートでトラフィックの過負荷につながります。

この問題を解決するために、ビルド 13.0 71.x 以降、Citrix ADC アプライアンスは、ハッシュアルゴリズムでパケットの次の 5 つのタプル情報を使用して、パケットのルートを選択します。

- 送信元 IP アドレス（クライアント IP）
- 送信元ポート（クライアントポート）
- 宛先 IP アドレス（サービス IP）
- 宛先ポート（サービスポート）
- プロトコル番号

5 つのタプル情報に基づいてルートを選択すると、使用可能なルートでトラフィックが均等に分散されます。このトラフィックの均等な分散により、ルート内のトラフィックの過負荷が防止されます。

クライアントが VIP アドレスにリクエストを送信する負荷分散設定の例を考えてみましょう。Citrix ADC アプライアンスは、次の 5 つのタプル情報を使用して、負荷分散サーバーに要求パケットを送信するルートを選択します。

- 送信元 IP アドレス（クライアント IP アドレス）
- 送信元ポート（クライアントポート）
- 宛先 IP アドレス（サービス IP アドレス）
- 宛先ポート（サービスポート番号）
- プロトコル番号

Citrix ADC 機能に基づく他のルート選択に関する優先順位

このセクションでは、Citrix ADC アプライアンスの 5 つのタプル機能およびその他のルート選択関連機能に基づくルート選択の優先順位について説明します。

- ポリシーベースのルート（**PBR**）。PBR ルールは、常に 5 つのタプルに基づくルート選択よりも優先されます。
- **Mac** ベースの転送（**MBF**）。負荷分散構成では、次の場合に 5 つのタプルに基づく MBF またはルート選択が優先されます。
  - Citrix ADC アプライアンスの負荷分散構成の VIP アドレスへのクライアント開始トラフィックの場合：
    * 負荷分散サーバー宛てのトラフィックを要求します。5 つのタプルに基づくルート選択は、MBF よりも優先されます。
    * クライアント宛ての応答トラフィック。MBF は、5 つのタプルに基づくルート選択よりも優先されます。
  - サーバーが Citrix ADC アプライアンスの SNIP アドレスへのトラフィックを開始した場合：
    * クライアント宛ての応答トラフィック。5 つのタプルに基づくルート選択は、MBF よりも優先されます。
    * 負荷分散サーバー宛てのトラフィックを要求します。MBF は、5 つのタプルに基づくルート選択よりも優先されます。

ルーティングの問題のトラブルシューティング

October 7, 2021

トラブルシューティングプロセスをできるだけ効率的にするには、まずネットワークに関する情報を収集します。Citrix ADC アプライアンスおよびネットワーク内のその他のシステムに関する以下の情報を入手する必要があります。

- インターフェイス接続および中間スイッチの詳細を含む、完全なトポロジ図。
- 構成を実行します。show running コマンドを使用して、ns.conf および ZebOS.conf の実行構成を取得できます。
- History コマンドの出力。問題が発生したときに設定が変更されたかどうかを判断します。
- Top コマンドと ps-ax コマンドの出力。ルーティングデーモンが CPU を過度に使用しているか、または誤動作しているかを判別します。
- /var/core 内のルーティング関連のコアファイル（nsm、bgpd、ospfd、または ripd）。タイムスタンプをチェックして、関連性があるかどうかを確認します。
- /var/log のファイルと dr_info.log ファイルのファイルを削除します。
- 関連するすべてのシステムの date コマンドと時刻の詳細の出力。ログメッセージの時刻をさまざまなイベントと関連付けることができるように、すべてのデバイスの日付を次々と出力します。
- 関連する ns.log ファイル、ニュースログファイル。
- アップストリームおよびダウンストリームルータからの構成ファイル、ログファイル、およびコマンド履歴の詳細。

汎用ルーティングに関する **FAQ**

October 7, 2021

一般的なルーティングの問題のトラブルシューティング方法について、ユーザーには通常、次の質問があります。

- 設定ファイルを保存するにはどうすればよいですか?

  VTYSH からの書き込みコマンドは、ZebOS.conf のみを保存します。CLI から save ns config コマンドを実行して、ns.conf ファイルと ZebOS.conf ファイルの両方を保存します。

- スタティックデフォルトルートとダイナミックに学習されたデフォルトルートの両方を設定した場合、これは優先デフォルトルートですか。

  ダイナミックに学習されたルートは、優先されるデフォルトルートです。この動作は、デフォルトルートに固有です。ただし、Network Services Module（NSM; ネットワークサービスモジュール）の場合、アドミニストレーティブディスタンスが変更されない限り、ダイナミックルートよりもスタティックに設定されたルートが優先されます。NSM FIB にダウンロードされるルートは、スタティックルートです。

- デフォルトルートのアドバタイズメントをブロックするにはどうすればよいですか。

  デフォルトルートは ZebOS に挿入されません。

- ネットワークデーモンのデバッグ出力を表示するにはどうすればよいですか?

  VTYSH のグローバル設定ビューから次の log file コマンドを入力すると、ネットワーキングデーモンのデバッグ出力をファイルに書き込むことができます。

```
ns(config)# log file /var/ZebOS.log
<!--NeedCopy-->
```

  VTYSH ユーザビューから terminal monitor コマンドを入力して、デバッグ出力をコンソールに送ることができます。

```
ns# terminal monitor
<!--NeedCopy-->
```

- 実行中のデーモンのコアを収集するにはどうすればよいですか?

  gcore ユーティリティを使用して、実行中のデーモンのコアを収集し、gdb で処理することができます。これは、ルーティング操作全体を停止させることなく、正しく動作しないデーモンをデバッグする場合に役立ちます。

```
gcore [-s] [-c core] [executable] pid
<!--NeedCopy-->
```

  -s オプションは、コアイメージの収集中にデーモンを一時的に停止します。これは、結果のイメージでコアが一貫した状態であることが保証されるため、推奨されるオプションです。

```
root@ns#gcore -s -c nsm.core /netscaler/nsm 342
<!--NeedCopy-->
```

- ZebOS コマンドのバッチ実行方法を教えてください。

  VTYSH-f コマンドを入力すると、ファイルから ZebOS<gile-name> コマンドのバッチを実行できます。これは実行構成を置き換えるのではなく、それに追加されます。ただし、バッチファイル内の既存の構成を削除し、新しい必要な構成に追加するためのコマンドを含めることで、このメカニズムを使用して特定の構成を置き換えることができます。


```
!
router bgp 234
network 1.1.1.1 255.255.255.0
!
route-map bgp-out2 permit 10
 set metric 9900
 set community 8602:300
!
<!--NeedCopy-->
```

OSPF 固有の問題のトラブルシューティング

October 7, 2021

OSPF 固有の問題のデバッグを開始する前に、Citrix ADC アプライアンスおよび影響を受ける LAN 内のすべてのシステム（アップストリームおよびダウンストリームルータを含む）から情報を収集する必要があります。開始するには、次のコマンドを入力します。

1. show interface from both nscli and VTYSH
2. show ip ospf interface
3. show ip ospf neighbor detail
4. show ip route
5. show ip ospf route
6. show ip ospf database summary
    - データベースに LSA が少数しかない場合は、show ip ospf データベースルータ、show ip ospf データベース A ネットワーク、show ip ospf データベース外部、およびその他のコマンドを入力して、LSA の完全な詳細を取得します。
    - データベース内に多数の LSA がある場合は、show ip ospf database 自己生成コマンドを入力します。
7. show ip ospf
8. show ns ip. これにより、関心のあるすべての VIP の詳細が確実に含められます。
9. ピアリングデバイスからログを取得し、次のコマンドを実行します。

```
gcore -s -c xyz.core /netscaler/ospfd <pid>
```

注**:** gcore コマンドは中断を伴いません。

以下の手順に従って、Citrix ADC から追加情報を収集します。

1. VTYSH のグローバル構成ビューから次のコマンドを入力して、エラーメッセージのロギングを有効にします。

```
ns(config)# log file /var/ospf.log
<!--NeedCopy-->
```

2. ospf イベントのデバッグを有効にし、次のコマンドを使用してログに記録します。

```
ns(config) #log file /var/ospf.log
<!--NeedCopy-->
```

データベース内の LSA の数が比較的少ない（500 未満）場合にだけ debug ospf lsa パケットを有効にします。

インターネットプロトコルバージョン 6 (IPv6)

October 7, 2021

Citrix ADC アプライアンスは、サーバー側とクライアント側の両方の IPv6 をサポートしているため、IPv6 ノードとして機能できます。IPv6 ノード（ホストとルーターの両方）と IPv4 ノードからの接続を受け入れ、トラフィックをサービスに送信する前にプロトコル変換（RFC 2765）を実行できます。

次の表に、Citrix ADC アプライアンスがサポートする IPv6 機能の一部を示します。

表 1. サポートされている IPv6 の機能

IPv6 の機能
SNIP の IPv6 アドレス（NSIP6、VIP6、および SNIP6）
近隣探索 (アドレス解決、重複アドレス検出、近隣到達不能検出、ルーター検出)
管理アプリケーション（ping6、telnet6、ssh6）
スタティックルーティングとダイナミックルーティング（OSPF、BGP、RIPng、および ISIS）
ポートベース VLAN
IPv6 アドレスのアクセスコントロールリスト（ACL6）
IPv6 プロトコル（TCP6、UDP6、ICMP6）
サーバ側のサポート（vserver、サービスの IPv6 アドレス）

IPv6 の機能
IPv6 用の USIP（ソース IP を使用）および DSR（ダイレクト・サーバー・リターン）
IPv6 用の SNMP および CVPN
ネイティブ IPv6 ノードアドレスを持つ HA
MIP の IPv6 アドレス
IPv6 のパス MTU ディスカバリ

IPv6 サポートの実装

Citrix ADC アプライアンスの IPv6 機能を使用または構成する前に、Citrix ADC アプライアンスで IPv6 機能を有効にする必要があります。IPv6 が無効になっている場合、Citrix ADC は IPv6 パケットを処理しません。サポートされていないコマンドを実行すると、次の警告が表示されます。

```
1 ”Warning: Feature(s) not enabled [IPv6PT]”
2 <!--NeedCopy-->
```

IPv6 を有効または無効にするには、次のいずれかの手順を使用します。

CLI のプロシージャ

CLI を使用して IPv6 を有効または無効にするには、次の手順を実行します。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- enable ns feature ipv6pt
- disable ns feature ipv6pt

GUI のプロシージャ

GUI を使用して IPv6 を有効または無効にするには、次の手順を実行します。

1. [システム] **>** [設定] に移動し、[モードと機能] グループで [高度な機能の設定] をクリックします。
2. [**IPv6** プロトコル変換] オプションを選択または選択解除します。

VLAN のサポート

VLAN を識別せずにブロードキャストまたはマルチキャストパケットを送信する必要がある場合（たとえば、DAD が NSIP の場合、または ND6 の場合はルートのネクストホップの場合）、Citrix ADC アプライアンスを構成して、すべ

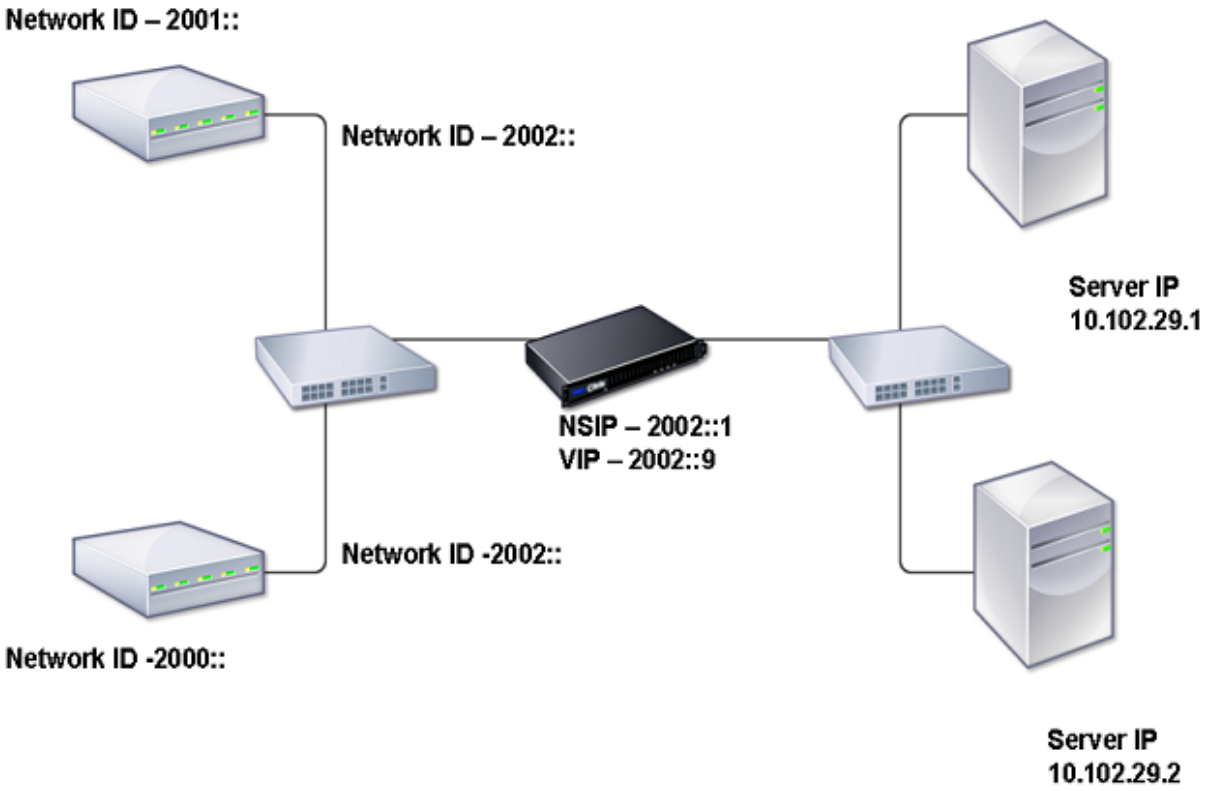
てのインターフェイスに適切なタグを付けてパケットを送信できます。VLAN は ND6 によって識別され、データパケットは VLAN 上でのみ送信されます。ND6 および VLAN の詳細については、ネイバー探索の設定を参照してください。

ポートベースの VLAN は、IPv4 および IPv6 に共通です。IPv6 では、プレフィクスベースの VLAN がサポートされています。

単純な展開シナリオ

次のトポロジ図に示すように、IPv6 vserver および IPv4 サービスで構成される単純なロードバランシングの設定例を次に示します。

図 1: IPv6 サンプルトポロジ



次の表は、Citrix ADC 上で構成する必要があるエンティティの名前と値をまとめたものです。

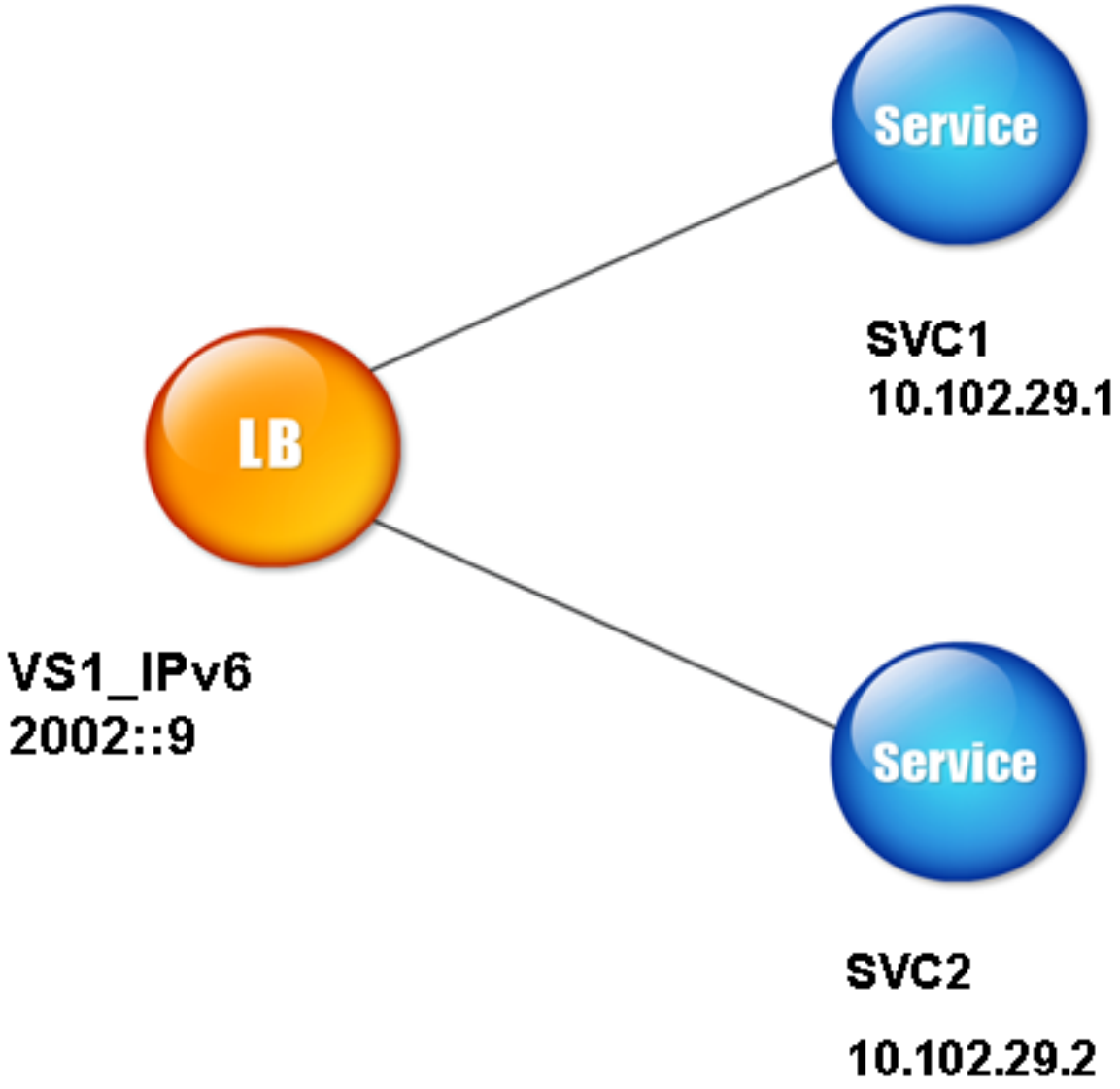
表 2. エンティティ作成のサンプル値

エンティティタイプ	名前	値
LB 仮想サーバー	VS1_IPv6	2002::9
サービス	SVC1	10.102.29.1

エンティティタイプ	名前	値
	SVC2	10.102.29.2

次の図は、Citrix ADC 上で構成するパラメータのエンティティと値を示しています。

図 2：IPv6 エンティティ図



この展開シナリオを構成するには、次の操作を行う必要があります。

1. IPv6 サービスを作成します。

2. IPv6 LB 仮想サーバーを作成します。
3. サービスを vserver にバインドします。

CLI のプロシージャ

CLI を使用して IPv4 サービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add service** <Name> <IPAddress> <Protocol> <Port>
- **sh service** <Name>

例**:**

```
1 > add service SVC1 10.102.29.1 HTTP 80
2 Done
3
4 >add service SVC2 10.102.29.2 HTTP 80
5 Done
6 <!--NeedCopy-->
```

CLI を使用して IPv6 仮想サーバを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add lb vserver** <Name> <IPAddress> <Protocol> <Port>
- **sh lb vserver** <Name>

例**:**

```
1 > add lb vserver VS1_IPv6 2002::9 HTTP 80
2 Done
3 <!--NeedCopy-->
```

CLI を使用してサービスを LB 仮想サーバーにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **bind lb vserver** <name> <service>
- **sh lb vserver** <name>

例**:**


```
> bind lb vserver VS1_IPv6 SVC1
Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して IPv4 サービスを作成するには、次の手順を実行します。

[トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックして次のパラメータを設定します。

- サービス名
- IP アドレス
- プロトコル
- ポート

GUI を使用して IPv6 仮想サーバを作成するには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックして、[**IPv6**] チェックボックスをオンにします。
2. 次のパラメーターを設定します。
   - 名前
   - プロトコル
   - IP アドレスの種類
   - IP アドレス
   - ポート

GUI を使用してサービスを LB 仮想サーバーにバインドするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. [負荷分散仮想サーバー] ページで、サービスをバインドする vserver (VS1_IPv6 など) を選択します。
3. [開く] をクリックします。
4. [仮想サーバーの構成 **(**負荷分散**)**] ダイアログボックスの [サービス] タブで、vserver にバインドするサービス (SVC1 など) に対応する [アクティブ] チェックボックスをオンにします。
5. [**OK**] をクリックします。
6. 手順 1 ～4 を繰り返して、サービスをバインドします（たとえば、SVC2 を vserver にバインドします)。

ホストヘッダーの変更

HTTP 要求のホストヘッダーに IPv6 アドレスが含まれており、サーバーが IPv6 アドレスを認識しない場合は、IPv6 アドレスを IPv4 アドレスにマッピングする必要があります。IPv4 アドレスは、vserver に送信される HTTP 要求のホストヘッダーで使用されます。

CLI のプロシージャ

CLI を使用して、ホストヘッダー内の IPv6 アドレスを IPv4 アドレスに変更するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set ns ip6** <IPv6Address> -**map** <IPAddress>
- **sh ns ip6** < IPv6 アドレス >

例**:**

```
1 > set ns ip6 2002::9 -map 200.200.200.200
2 Done
3 <!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して、ホストヘッダーの IPv6 アドレスを IPv4 アドレスに変更するには、次の手順を実行します。

1. **[**システム**] > [**ネットワーク**] > [IP]** に移動し、**[**IPV6s**]** タブで、マッピングされた **IP** アドレスを構成する **IP** アドレス **(**たとえば、**2002:0:0:0:0:0:0:9)** を選択し、**[**編集**]** をクリックします。
2. [マップされた **IP**] テキストボックスに、構成するマップされた IP アドレスを入力します。たとえば、200.200.200.200 と入力します。

VIP 挿入

IPv6 アドレスが IPv4 ベースのサーバーに送信されると、サーバーが HTTP ヘッダー内の IP アドレスを認識できず、エラーが発生する可能性があります。これを回避するには、IPv4 アドレスを IPv6 VIP にマッピングします。次に、VIP 挿入を有効にして、サーバに送信される HTTP 要求に IPv4 VIP アドレスとポート番号を挿入できるようにします。

CLI のプロシージャ

CLI を使用してマップ IPv6 アドレスを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

set ns ip6 <IPv6Address> -**map** <IPAddress>

例**:**

```
1 > set ns ip6 2002::9 -map 200.200.200.200
2  Done
```



```
<!--NeedCopy-->
```

CLI を使用して VIP 挿入を有効にするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **set lb vserver** <name> -**insertVserverIPPort** <Value>
- **sh lb vserver** <name>

例**:**

```
> set lb vserver VS1_IPv6 -insertVserverIPPort ON
 Done

<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用してマップ IPv6 アドレスを設定するには、次の手順を実行します。

1. **[**システム**] > [**ネットワーク**] > [IP]** に移動し、**[**IPV6s**]** タブで、マップ **IP** アドレスを構成する **IP** アドレス（たとえば、**2002:0:0:0:0:0:0:9**）を選択し、**[**編集**]** をクリックします。
2. [マップされた **IP**] テキストボックスに、構成するマップ IP アドレス（200.200.200.200 など）を入力します。

GUI を使用して VIP 挿入を有効にするには、次の手順を実行します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、ポートの挿入を有効にする仮想サーバーを選択して、[編集] をクリックします。
2. [詳細設定] タブの **[**トラフィックの設定**]** で、**[**仮想サーバー **IP** ポートの挿入] ドロップダウンリストボックスで [**VIPADDR**] を選択します。
3. [仮想サーバー **IP** ポートの挿入] テキストボックスに、vip ヘッダーを入力します。

トラフィックドメイン

October 7, 2021

> 警告
>
> トラフィックドメインではなく、管理パーティションを使用することをお勧めします。詳細については、「管理者パーティショニング」ページを参照してください。

トラフィックドメインは、さまざまなアプリケーションのネットワークトラフィックをセグメント化する方法です。トラフィックドメインを使用して、Citrix ADC アプライアンス内に複数の分離された環境を作成できます。特定のトラフィックドメインに属するアプリケーションは、エンティティと通信し、そのドメイン内のトラフィックを処理します。あるトラフィックドメインに属するトラフィックは、別のトラフィックドメインの境界を越えることはできません。

トラフィックドメインを使用する利点

Citrix ADC アプライアンスでトラフィックドメインを使用する主な利点は次のとおりです。

- ネットワーク内の重複した **IP** アドレスの使用。トラフィックドメインでは、ネットワーク上で重複する IP アドレスを使用できます。重複するアドレスが異なるトラフィックドメインに属していれば、ネットワーク上の複数のデバイス、または Citrix ADC アプライアンス上の複数のエンティティに同じ IP アドレスまたはネットワークアドレスを割り当てることができます。
- **Citrix ADC** アプライアンスでの重複エンティティの使用。トラフィックドメインでは、アプライアンスで重複する Citrix ADC 機能エンティティを使用することもできます。各エンティティが別のトラフィックドメインに割り当てられている限り、同じ設定でエンティティを作成できます。
  注: 同じ名前の重複エンティティはサポートされていません。
- マルチテナンシー。トラフィックドメインを使用すると、ネットワーク上の定義されたアドレス空間内で各顧客のタイプのアプリケーショントラフィックを分離することで、複数の顧客にホスティングサービスを提供できます。

トラフィックドメインは、整数値である ID によって一意に識別されます。各トラフィックドメインには、VLAN または VLAN のセットが必要です。トラフィックドメインの分離機能は、トラフィックドメインにバインドされた VLAN によって異なります。1 つのトラフィックドメインに複数の VLAN をバインドできますが、同じ VLAN を複数のトラフィックドメインの一部にすることはできません。したがって、作成できるトラフィックドメインの最大数は、アプライアンスに設定されている VLAN の数によって異なります。

デフォルトトラフィックドメイン

Citrix ADC アプライアンスには、
デフォルトのトラフィックドメインと呼ばれる事前構成済みのトラフィックドメインがあり、ID は 0 です。すべての工場出荷時の設定と設定は、デフォルトトラフィックドメインの一部です。他のトラフィックドメインを作成し、デフォルトのトラフィックドメインと他のトラフィックドメイン間でトラフィックをセグメント化することができます。Citrix ADC アプライアンスからデフォルトのトラフィックドメインを削除することはできません。トラフィックドメイン ID を設定せずに作成したフィーチャエンティティは、自動的にデフォルトのトラフィックドメインに関連付けられます。

注：一部の機能と設定は、デフォルトのトラフィックドメインでのみサポートされています。デフォルト以外のトラフィックドメインでは機能しません。すべてのトラフィックドメインでサポートされている機能のリストについては、「トラフィックドメインでサポートされる Citrix ADC 機能」を参照してください。

トラフィックドメインの仕組み

トラフィックドメインの例として、ID 1 と 2 を持つ 2 つのトラフィックドメインが Citrix ADC アプライアンスの NS1 上に構成されているとします。

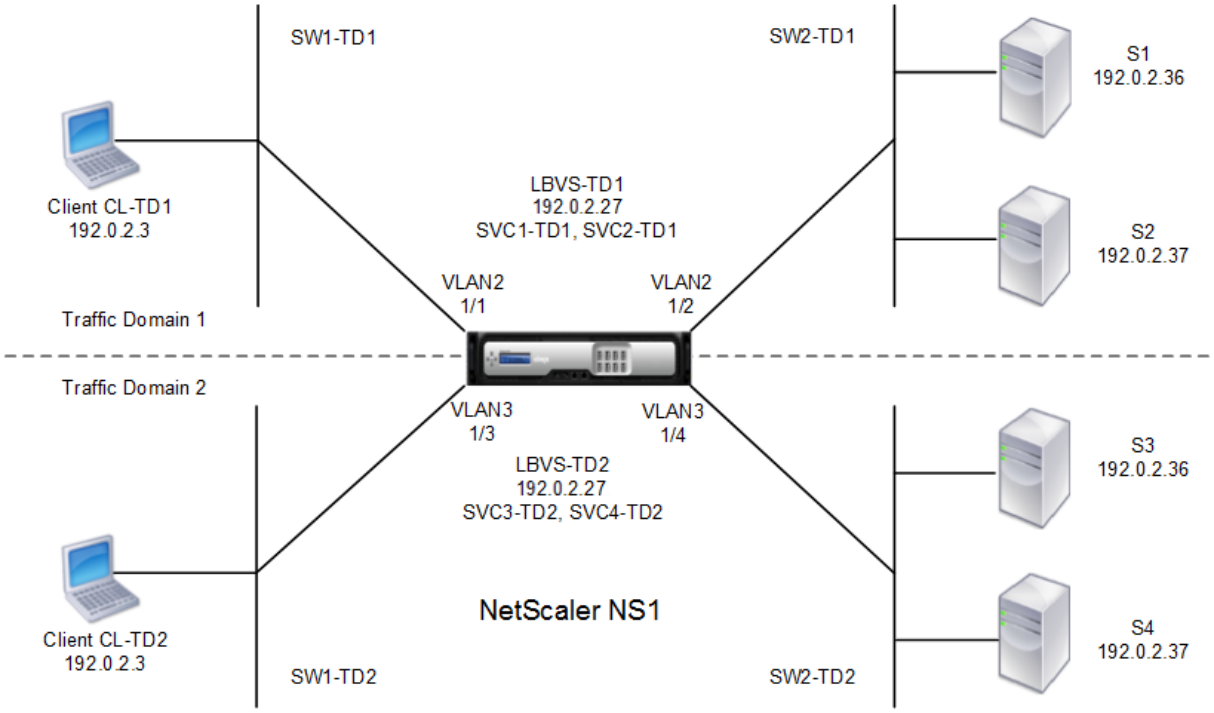
トラフィックドメイン 1 では、ロードバランシング仮想サーバ LBVS-TD1 が、サーバ S1 および S2 間でトラフィックをロードバランシングするように設定されます。Citrix ADC アプライアンスでは、サーバー S1 と S2 は、それぞれサービス SVC1-TD1 と SVC2-TD1 で表されます。サーバ S1 および S2 は、L2 スイッチ SW2-TD1 を介して NS1 に接続されています。クライアント CL-TD1 は、L2 スイッチ SW1-TD1 を介して NS1 に接続されたプライベートネットワーク上にある。SW1-TD1 および SW2-TD1 は、NS1 の VLAN 2 に接続されています。VLAN 2 はトラフィックドメイン 1 にバインドされます。つまり、クライアント CL-TD1 およびサーバ S1 および S2 はトラフィックドメイン 1 の一部です。

同様に、トラフィックドメイン 2 でも、ロードバランシング仮想サーバー LBVS-TD2 は、S3 と S4 間でトラフィックを負荷分散するように設定されます。Citrix ADC アプライアンスでは、サーバー S3 と S4 はサービス SVC3-TD2 と SVC4-TD2 で表されます。サーバ S3 および S4 は、L2 スイッチ SW2-TD2 を介して NS1 に接続されています。クライアント CL-TD2 は、L2 スイッチ SW1-TD2 を介して NS1 に接続されたプライベートネットワーク上にある。SW1-TD2 および SW2-TD2 は、NS1 の VLAN 3 に接続されています。VLAN 3 はトラフィックドメイン 2 にバインドされています。つまり、クライアント CL-TD2 およびサーバ S3 および S4 はトラフィックドメイン 2 の一部です。

Citrix ADC アプライアンスでは、エンティティ LBVS-TD1 と LBVS-TD2 は、IP アドレスを含む同じ設定を共有します。同じことが、SVC1-TD1 および SVC3-TD2、および SVC2-TD1 および SVC4-TD2 にも当てはまります。これは、これらのエンティティが異なるトラフィックドメインにあるために可能です。

同様に、サーバー S1 と S3、S2 および S4 は同じ IP アドレスを共有し、クライアント CL-TD1 と CL-TD2 はそれぞれ同じ IP アドレスを持っています。

図 1: トラフィックドメインの仕組み



次の表は、例で使用される設定の一覧です。

エンティティ	Name	詳細
トラフィックドメイン **1** の設定		
トラフィックドメイン 1 にバインドされた VLAN	VLAN 2	VLAN ID：2 つのインターフェイスにバインドされます。1/1、1/2
クライアントが TD1 に接続されている	CL-TD1 (参考目的のみ)	IP address: 192.0.2.3
TD1 の仮想サーバの負荷分散	LBVS-TD1	IP address: 192.0.2.27
仮想サーバー LBVS-TD1 にバインドされたサービス	SVC1-TD1	IP address: 192.0.2.36
仮想サーバー LBVS-TD1 にバインドされたサービス	SVC2-TD1	IP address: 192.0.2.37
SNIP	SNIP-TD1（参考目的のみ）	IP address: 192.0.2.27
トラフィックドメイン **2** の設定		
トラフィックドメイン 2 にバインドされた VLAN	VLAN 3	VLAN ID：3 つのインターフェイスにバインドされます。1/3、1/4
クライアントが TD2 に接続されている	CL-TD2 (参考目的のみ)	IP address: 192.0.2.3
TD2 の仮想サーバの負荷分散	LBVS-TD2	IP address: 192.0.2.27

エンティティ	Name	詳細
仮想サーバー LBVS-TD2 にバインドされたサービス	SVC3-TD2	IP address: 192.0.2.36
仮想サーバー LBVS-TD2 にバインドされたサービス	SVC4-TD2	IP address: 192.0.2.37
トリニティの対立法	SNIP-TD2（参考目的のみ）	IP address: 192.0.2.29

トラフィックドメイン 1 のトラフィックフローを次に示します。

1. クライアント CL-TD1 は、L2 スイッチ SW1-TD1 を介して 192.0.2.27 の IP アドレスに対する ARP 要求をブロードキャストします。
2. ARP 要求は、VLAN 2 にバインドされているインターフェイス 1/1 で NS1 に到達します。VLAN 2 はトラフィックドメイン 1 にバインドされているため、NS1 はクライアント CL-TD1 の IP アドレスのトラフィックドメイン 1 の ARP テーブルを更新します。
3. ARP 要求はトラフィックドメイン 1 で受信されるため、NS1 は、IP アドレスが 192.0.2.27 であるトラフィックドメイン 1 で構成されたエンティティを探します。NS1 は、負荷分散仮想サーバー LBVS-TD1 がトラフィックドメイン 1 に構成されており、IP アドレスが 192.0.2.27 であることを検出します。
4. NS1 は、インタフェース 1/1 の MAC アドレスを含む ARP 応答を送信します。
5. ARP 応答が CL-TD1 に到達します。CL-TD1 は、LBVS-TD1 の IP アドレスの ARP テーブルを NS1 のインターフェイス 1/1 の MAC アドレスで更新します。
6. クライアント CL-TD1 は 192.0.2.27 に要求を送信します。要求は、NS1 のポート 1/1 で LBVS-TD1 によって受信されます。
7. LBVS-TD1 のロードバランシングアルゴリズムはサーバ S2 を選択し、NS1 はトラフィックドメイン 1（192.0.2.27）と S2 の SNIP 間の接続を開きます。
8. S2 は、NS1 上の SNIP 192.0.2.27 に返信します。
9. NS1 は、クライアント CL-TD1 に S2 の応答を送信します。

トラフィックドメイン 2 のトラフィックフローを次に示します。

1. クライアント CL-TD2 は、L2 スイッチ SW1-TD2 を介して 192.0.2.27 の IP アドレスに対する ARP 要求をブロードキャストします。
2. ARP 要求は、VLAN 3 にバインドされているインターフェイス 1/3 で NS1 に到達します。VLAN 3 はトラフィックドメイン 2 にバインドされているため、同じ IP アドレスの ARP エントリ（CL-TD1）がすでにトラフィックドメイン 1 の ARP テーブルに存在していても、NS1 はトラフィックドメイン 2 の ARP テーブルエントリをクライアント CL-TD2 の IP アドレスに対して更新します。
3. ARP 要求はトラフィックドメイン 2 で受信されるため、NS1 はトラフィックドメイン 2 で、IP アドレスが 192.0.2.27 のエンティティを検索します。NS1 は、負荷分散仮想サーバー LBVS-TD2 がトラフィックドメイン 2 で構成されており、IP アドレスが 192.0.2.27 であることを検出します。NS1 は LBVS-TD2 と同じ IP アドレスを持っていても、トラフィックドメイン 1 の LBVS-TD1 を無視します。

4. NS1 は、インタフェース 1/3 の MAC アドレスを含む ARP 応答を送信します。
5. ARP 応答が CL-TD2 に到達します。CL-TD2 は、LBVS-TD2 の IP アドレスの ARP テーブルエントリを NS1 のインターフェイス 1/3 の MAC アドレスで更新します。
6. クライアント CL-TD2 は 192.0.2.27 に要求を送信します。要求は、NS1 のインターフェイス 1/3 で LBVS-TD2 によって受信されます。
7. LBVS-TD2 の負荷分散アルゴリズムはサーバー S3 を選択し、NS1 はトラフィックドメイン 2（192.0.2.29）の SNIP と S3 の間の接続を開きます。
8. S2 は、NS1 上の SNIP 192.0.2.29 に返信します。
9. NS1 は、クライアント CL-TD2 に S2 の応答を送信します。

トラフィックドメインでサポートされる **Citrix ADC** 機能

次のリストの Citrix ADC 機能は、すべてのトラフィックドメインでサポートされています。

> 重要
>
> 以下に記載されていない Citrix ADC 機能は、デフォルトのトラフィックドメインでのみサポートされます。

- ARP テーブル
- ND6 テーブル
- Bridge テーブル
- すべてのタイプの IPv4 アドレスと IPv6 アドレス
- IPv4 および IPv6 のルート
- ACL および ACL6
- PBR & PBR6
- INAT
- RNAT
- RNAT6
- MSR
- MSR6
- ネットプロファイル
- SNMP MIB
- フラグメンテーション
- モニタ（スクリプト可能なモニタはサポートされていません）
- コンテンツスイッチ
- キャッシュリダイレクト
- 永続性 (永続性グループはサポートされていません)
- サービス（ドメインベースのサービスはサポートされていません）
- サービスグループ (ドメインベースのサービスグループはサポートされていません)
- ポリシー (*)
- PING
- TRACEROUTE

- PMTU
- 高可用性（接続ミラーリングはサポートされていません）
- クラスタ（L2 クラスタでサポートされています。L3 クラスタではサポートされません）
- クッキーの永続性
- MSS
- ロギング（Syslog はサポートされていません）
- 優先度によるキューイング
- サージ保護
- HTTP DOSP (**)
- 負荷分散（次のタイプはサポートされていません。）
  - TFTP
  - RTSP
  - Diameter
  - SIP
  - SMPP
- NAT46
- NAT64
- DNS64
- 転送セッション規則
- SNMP

> 注
>
> - * ポリシーには、トラフィックドメインのグローバルバインディングポイントはありません。ただし、ポリシーは、トラフィックドメインの特定のロードバランシング仮想サーバにバインドできます。
> - ** HTTP DOSP ポリシーには、トラフィックドメインのグローバルバインディングポイントはありません。ただし、HTTP DOSP ポリシーは、トラフィックドメインの特定のロードバランシングサービスにバインドできます。
> - グローバルサーバー負荷分散（GSLB）と Citrix ADC NS 機能は、トラフィックドメインを認識しません。GSLB 設定をすべてのトラフィックドメインで共有する必要がある場合、GSLB メソッドの静的近接およびラウンドトリップ時間（RTT）は機能しません。このシナリオの回避策として、RTT および静的近接以外の GSLB メソッドを使用できます。詳しくは、http://support.citrix.com/article/CTX202277を参照してください。

トラフィックドメインの設定

Citrix ADC アプライアンスでトラフィックドメインを構成するには、次の作業を行います。

- **VLAN** を追加します。VLAN を作成し、指定したインターフェイスをそれらにバインドします。
- トラフィックドメインエンティティを作成し、**VLAN** をバインドします。これには、次の 2 つのタスクが含まれます。

  - ID（整数）で一意に識別されるトラフィックドメインエンティティを作成します。
  - 指定した VLAN をトラフィックドメインエンティティにバインドします。指定された VLAN にバインドされたすべてのインターフェイスは、トラフィックドメインに関連付けられます。1 つのトラフィックドメインに複数の VLAN をバインドできますが、1 つの VLAN を複数のトラフィックドメインの一部にすることはできません。
- トラフィックドメインにフィーチャエンティティを作成します。トラフィックドメインに必要なフィーチャエンティティを作成します。デフォルト以外のトラフィックドメインでサポートされるすべての機能の CLI コマンドおよび設定ダイアログボックスには、トラフィックドメイン *ID*（td）と呼ばれるパラメータが含まれています。機能エンティティを設定するときに、エンティティを特定のトラフィックドメインに関連付ける場合は、td を指定する必要があります。td を設定せずに作成したフィーチャエンティティは、自動的にデフォルトのトラフィックドメインに関連付けられます。

このトピックでは、フィーチャエンティティがトラフィックドメインにどのように関連付けられているかを理解するために、「トラフィックドメインの仕組み」という図で説明されているすべてのエンティティの構成手順について説明します。

CLI には、これら 2 つのタスクに対して 2 つのコマンドがありますが、GUI では 1 つのダイアログボックスにこれらのコマンドが組み合わされます。

CLI のプロシージャ

CLI を使用して VLAN を作成し、インターフェイスをバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add vlan** <id>
- **bind vlan** <id> -**ifnum** <slot/port>
- **show vlan** <id>

CLI を使用して、トラフィックドメインエンティティを作成し、VLAN をバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add ns trafficdomain** <td>
- **bind ns trafficdomain** <td> -**vlan** <id>
- **nstrafficdomain** を表示 <td>

CLI を使用してサービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add service** <name> <IP> <serviceType> <port> -**td** <id>
- ショーサービス < 名前 >

CLI を使用して負荷分散仮想サーバーを作成し、サービスをバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add lb vserver** <name> <serviceType> <IPAddress> <port> -**td** <id>
- **bind lb vserver** <name> <serviceName>
- **show lb vserver** < 名前 >

GUI のプロシージャ

GUI を使用して VLAN を作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [**VLAN**] に移動し、[追加] をクリックしてパラメータを設定します。

GUI を使用してトラフィックドメインエンティティを作成するには、次の手順を実行します。

[システム] > [ネットワーク] > [トラフィックドメイン**]** に移動し、**[**追加**]** をクリックし、**[**トラフィックドメインの作成] ダイアログボックスでパラメータを設定します。

GUI を使用してサービスを作成するには、次のステップを実行します。

[トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックしてパラメータを設定します。

GUI を使用してロード・バランシング仮想サーバーを作成するには、次の手順を実行します。

[トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、[追加] をクリックしてパラメーターを設定します。

トラフィックドメインエンティティ間バインディング

October 7, 2021

あるトラフィックドメインのサービスを、別のトラフィックドメインの仮想サーバにバインドできます。異なるトラフィックドメイン内の仮想サーバにバインドされるすべてのサービスは、同じトラフィックドメインに存在する必要があります。

このサポートを設定するには、既存の bind lb vserver コマンドまたは関連する GUI 手順を使用します。

この機能により、異なるトラフィックドメイン間の相互作用が容易になります。企業では、サーバーを異なるトラフィックドメインにグループ化できます。仮想サーバーは、インターネットに面するトラフィックドメイン内に作成されます。このトラフィックドメインの仮想サーバは、別のトラフィックドメインのサーバをロードバランシングするように設定できます。この仮想サーバーは、バインドされたサーバーに転送されるインターネットからの接続要求を受信します。

Citrix ADC をクラウドインフラストラクチャで使用する場合、各テナントに個別のトラフィックドメインを割り当てることができ、テナントのすべてのリソース（サーバーを含む）をテナントのトラフィックドメインにグループ化できます。テナントごとに、トラフィックドメイン内の負荷分散サーバー用に仮想サーバーが作成されます。これらの仮想サーバはすべて、インターネットに面する 1 つのトラフィックドメインにグループ化されます。

たとえば、クラウドサービスプロバイダーの Example-Cloud-A に、ID 10、20、30 の 3 つのトラフィックドメインがあり、Citrix ADC アプライアンス NS1 上で構成されているとします。

例-組織 A と例-組織 B は、例-クラウド A のテナントです。テナント A にはトラフィックドメイン 20、テナント B にはドメイン 30 が割り当てられます。サーバ S1 と S2 はトラフィックドメイン 20 にあり、サーバ S3 と S4 はトラフィックドメイン 30 にあります。

トラフィックドメイン 10 はインターネットに面しています。仮想サーバー LBVS-1 および LBVS-2 は、トラフィックドメイン 10 で作成されます。トラフィックドメイン 10 の LBVS-1 は、トラフィックドメイン 20 にあるサーバ S1 および S2 をロードバランシングするように設定されています。トラフィックドメイン 10 の LBVS-2 は、トラフィックドメイン 30 にあるサーバー S3 および S4 を負荷分散するように設定されています。

したがって、これらの仮想サーバーは、仮想サーバーとは異なるトラフィックドメインにあるサーバーのインターネット接続要求を受け入れます。

仮想 **MAC** ベースのトラフィックドメイン

October 7, 2021

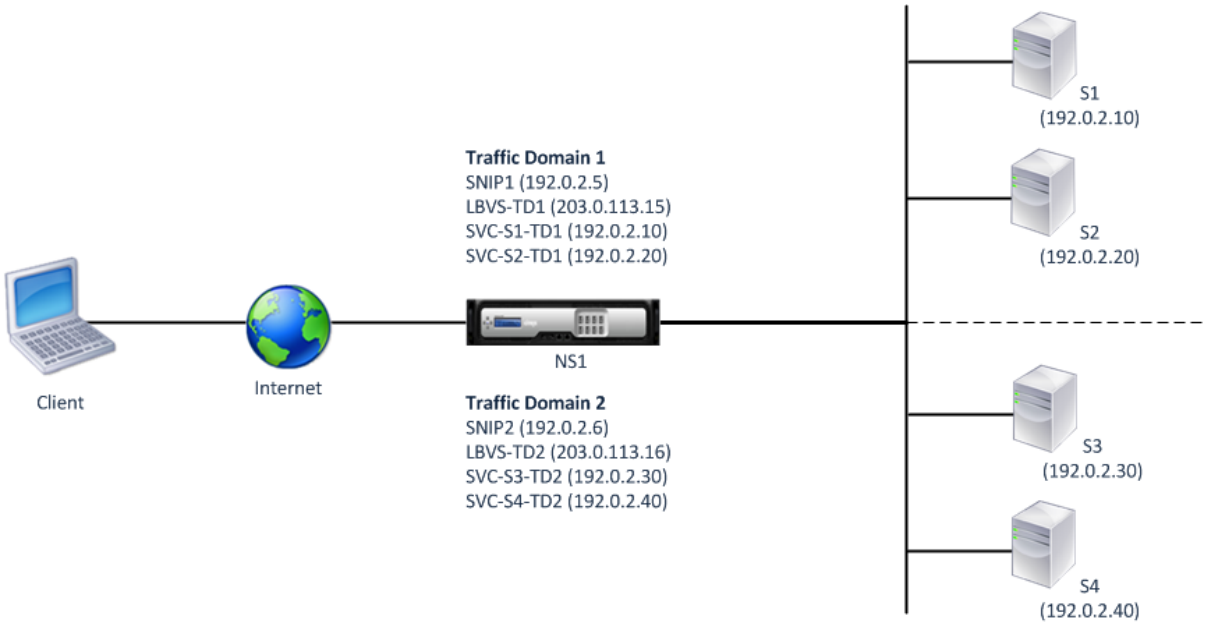
トラフィックドメインを VLAN ではなく仮想 MAC アドレスに関連付けることができます。Citrix ADC は、そのドメイン内のネットワークエンティティに対する ARP クエリに対するすべての応答で、トラフィックドメインの仮想 MAC アドレスを送信します。その結果、宛先 MAC アドレスはトラフィックドメインの仮想 MAC アドレスであるため、ADC は宛先 MAC アドレスに基づいて、異なるトラフィックドメインの後続の着信トラフィックを分離できます。トラフィックドメインにエンティティを作成したら、トラフィックドメインレベルの操作を実行することで、エンティティを簡単に管理および監視できます。

たとえば、ID 1 と 2 を持つ 2 つのトラフィックドメインが Citrix ADC アプライアンスの NS1 上に構成されているとします。Citrix ADC は、仮想 MAC アドレスの仮想 MAC1 を作成し、トラフィックドメイン 1 に関連付けます。同様に、Citrix ADC は別の仮想 MAC アドレス仮想 MAC2 を作成し、トラフィックドメイン 2 に関連付けます。

トラフィックドメイン 1 では、ロードバランシング仮想サーバ LBVS-TD1 が、サーバ S1 および S2 間でトラフィックをロードバランシングするように設定されます。Citrix ADC アプライアンスでは、サーバー S1 と S2 は、それぞれサービス SVC1-TD1 と SVC2-TD1 で表されます。サブネット IP アドレス（SNIP）SNIP1 は、Citrix ADC が S1 および S2 と通信できるように構成されています。仮想 MAC1 はトラフィックドメイン 1 に関連付けられているため、アプライアンスは LBVS-TD1 および SNIP1 に対するすべての ARP アナウンスメントおよび ARP 応答で、仮想 MAC1 を MAC アドレスとして送信します。

同様に、トラフィックドメイン 2 でも、ロードバランシング仮想サーバー LBVS-TD2 は、S3 と S4 間でトラフィックを負荷分散するように設定されます。Citrix ADC アプライアンスでは、サーバー S3 と S4 はサービス SVC3-TD2 と SVC4-TD2 で表されます。SNIP2 は、Citrix ADC が S3 および S4 と通信できるようにするために構成されています。仮想 MAC2 はトラフィックドメイン 2 に関連付けられているため、アプライアンスは、すべての ARP アナウンスメントおよび LBVS-TD2 および SNIP2 に対する ARP 応答で、仮想 MAC2 を MAC アドレスとして送信します。

Citrix ADC は、宛先 MAC アドレスが仮想 MAC1 または仮想 MAC2 の場合、宛先 MAC アドレスに基づいてトラフィックドメイン 1 または 2 に対する後続の着信トラフィックを分離します。



次の表に、例で使用される設定を示します。仮想 MAC ベースのトラフィックドメインの設定例。

はじめに

仮想 MAC ベースのトラフィックドメインを設定する前に考慮すべき点を次に示します。

1. 仮想 MAC ベースのトラフィックドメインは、ネットワークトラフィックの分離を実現する最も簡単な方法です。
2. 仮想 MAC ベースのトラフィックドメインは、VLANS ではなく仮想 MAC アドレスに基づいてネットワークトラフィックを分離するため、Citrix ADC 上の異なる仮想 MAC ベースのトラフィックドメインで重複 IP アドレスを作成することはできません。
3. Citrix ADC が L2 モードでのみ展開されている場合、仮想 MAC ベースのトラフィックドメインは機能しません。
4. VLAN ベースのトラフィックドメインと仮想 MAC ベースのトラフィックドメインは共存できます。仮想 MAC ベースのトラフィックドメインは、実際には VLAN ベースのトラフィックドメインにバインドされていないすべての VLAN で実行されます。

構成の手順

Citrix ADC アプライアンスで仮想 MAC ベースのトラフィックドメインを構成するには、次の作業を行います。

- トラフィックドメインエンティティを作成し、仮想 MAC オプションを有効にします。整数値である ID によって一意に識別されるトラフィックドメインエンティティを作成し、仮想 MAC オプションを有効にします。トラフィックドメインエンティティを作成した後、Citrix ADC は仮想 MAC アドレスを作成し、トラフィックドメインエンティティに関連付けます。

- トラフィックドメインにフィーチャエンティティを作成します。これらの機能エンティティを設定するときに、トラフィックドメイン ID（td）を指定して、必要な機能エンティティをトラフィックドメイン内に作成します。Citrix ADC が所有する仮想 MAC ベースのトラフィックドメインで作成されたネットワークエンティティは、トラフィックドメインに関連付けられた仮想 MAC アドレスに関連付けられます。Citrix ADC は、トラフィックドメインの仮想 MAC アドレスを ARP アナウンスメントとこれらのネットワークエンティティの ARP 応答で送信します。

CLI のプロシージャ

CLI を使用して仮想 MAC ベースのトラフィックドメインを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add ns trafficDomain <td> [-vmac ( ENABLED        DISABLED )]
```

-
- nstrafficdomain を表示 <td>

CLI を使用して SNIP アドレスを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add ns ip <IPAddress> <netmask> -type SNIP –td <id>
- show ns ip <IPAddress> -td <id>

CLI を使用してサービスを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- add service <name> <IP> <serviceType> <port> -td <id>
- show service <name> -td <id>

CLI を使用して負荷分散仮想サーバーを作成し、サービスをバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

- add lb vserver <name> <serviceType> <IPAddress> <port> -td <id>
- bind lb vserver <name> <serviceName>
- show lb vserver <name> -td <id>

例**:**

```
1 > add ns trafficDomain 1 -vmac ENABLED
2  Done
3 > add ns trafficDomain 2 -vmac ENABLED
```



```
 Done

> add ns ip 192.0.2.5 255.255.255.0 -type -SNIP -td 1
 Done
> add service  SVC-S1-TD1  192.0.2.10 HTTP 80 -td 1
 Done
> add service  SVC-S2-TD1  192.0.2.20 HTTP 80 -td 1
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.15 80 -td 1
 Done
> bind lb vserver  LBVS-TD1 SVC-S1-TD1
 Done
> bind lb vserver  LBVS-TD1 SVC-S2-TD1
 Done

> add ns ip 192.0.2.6 255.255.255.0 -type -SNIP -td 2
 Done
> add service  SVC-S3-TD2  192.0.2.30 HTTP 80 -td 2
 Done
> add service  SVC-S4-TD2  192.0.2.40 HTTP 80 -td 2
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.16 80 -td 1
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
<!--NeedCopy-->
```

GUI のプロシージャ

GUI を使用して仮想 MAC ベースのトラフィックドメインを作成するには、次の手順を実行します。

1. ［System］> ［Network］> ［Interfaces］に移動します。
2. 詳細ペインで、［Add］をクリックします。
3. [Create Traffic Domain] ページで、次のパラメータを設定します。
   - トラフィックドメイン ID*
   - Mac を有効にする
4. ［作成］をクリックします。

GUI を使用して SNIP アドレスを設定するには、次の手順を実行します。

1. [システム] > [ネットワーク] > [IP] > [IPv4] に移動します。
2. [ネットワーク] > [IP] > [IPv4] に移動します。

3. 詳細ウィンドウで、[追加] をクリックします。
4. [IP の作成] ページで、次のパラメータを設定します。パラメータの説明を参照するには、対応するフィールドにマウスカーソルを合わせます。
   - IP アドレス
   - ネットマスク
   - IP タイプ
   - トラフィックドメイン ID
5. ［作成］をクリックします。

GUI を使用してサービスを作成するには、次のステップを実行します。

1. ［Traffic Management］> ［Load Balancing］> ［Services］の順に移動します。
2. 詳細ペインで、［Add］をクリックします。
3. [基本設定] ページで、次のパラメータを設定します。パラメータの説明を参照するには、対応するフィールドにマウスカーソルを合わせます。
   - サービス名
   - サーバー
   - プロトコル
   - ポート
   - トラフィックドメイン ID
4. [続行] をクリックし、[完了] をクリックします。
5. 手順 2 ～4 を繰り返して、別のサービスを作成します。
6. ［閉じる］をクリックします。

負荷分散仮想サーバーを作成し、GUI を使用してサービスをバインドするには、次の手順を実行します。

1. ［Traffic Management］>［Load Balancing］>［Virtual Servers］の順に選択します。
2. [仮想サーバーの負荷分散] ウィンドウで、[追加] をクリックします。
3. [仮想サーバーの作成 (負荷分散)] ダイアログボックスで、次のパラメーターを設定します。パラメータの説明を参照するには、対応するフィールドにマウスカーソルを合わせます。
   - Name
   - IP アドレス
   - プロトコル
   - ポート
   - トラフィックドメイン ID
4. [続行] をクリックし、サービスペインで [>] をクリックします。
5. [サービス] ページで [挿入] をクリックし、仮想サーバーにバインドするサービスのチェックボックスをオンにします。
6. [続行] をクリックし、[完了] をクリックします。
7. 手順 2 ～5 を繰り返して、別の仮想サーバーを作成します。

VXLAN

October 7, 2021

Citrix ADC アプライアンスは、仮想拡張可能なローカルエリアネットワーク（VXLAN）をサポートしています。VXLAN は、レイヤ 2 フレームを UDP パケットにカプセル化することによって、レイヤ 2 ネットワークをレイヤ 3 インフラストラクチャにオーバーレイします。各オーバーレイネットワークは VXLAN セグメントと呼ばれ、VXLAN ネットワーク識別子（VNI）と呼ばれる一意の 24 ビット識別子によって識別されます。同じ VXLAN 内のネットワークデバイスだけが相互に通信できます。

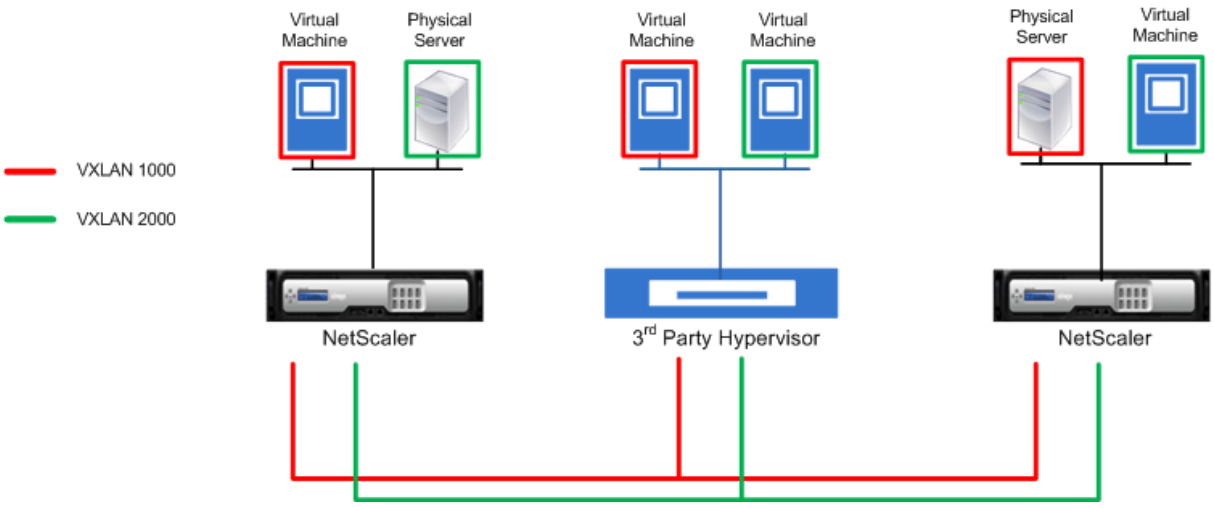
VXLAN は、VLAN と同じイーサネットレイヤ 2 ネットワークサービスを提供しますが、拡張性と柔軟性が向上します。VXLAN を使用する主な利点は、次の 2 つです。

- 拡張性の向上。サーバ仮想化とクラウドコンピューティングアーキテクチャは、データセンター内の分離されたレイヤ 2 ネットワークの需要を劇的に増加させました。VLAN 仕様では、12 ビット VLAN ID を使用してレイヤ 2 ネットワークを識別するため、4094 VLAN を超えて拡張することはできません。数千もの孤立したレイヤ 2 ネットワークが要求される場合は、この数が不十分になる可能性があります。24 ビット VNI は、同じ管理ドメイン内で最大 1,600 万の VXLAN セグメントに対応します。
- 高い柔軟性。VXLAN はレイヤ 3 パケットを介してレイヤ 2 データフレームを伝送するため、VXLAN は L2 ネットワークをデータセンターのさまざまな部分および地理的に離れたデータセンターに拡張します。データセンターの異なる部分と異なるデータセンターでホストされているが、同じ VXLAN の一部であるアプリケーションは、1 つの連続したネットワークとして表示されます。

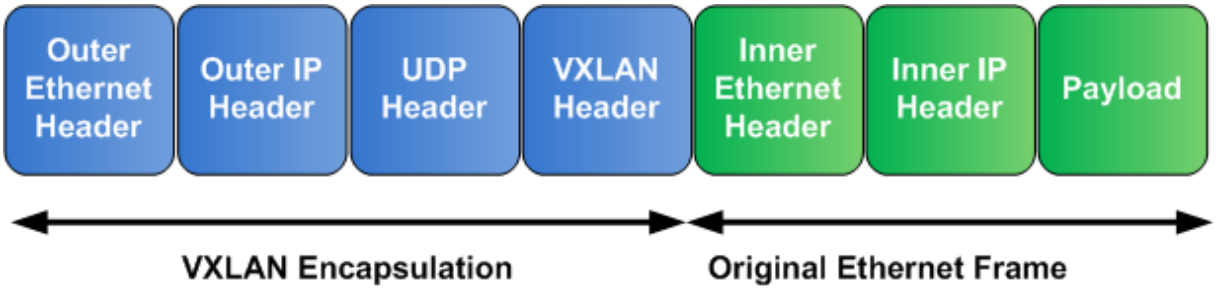
VXLAN のしくみ

VXLAN セグメントは、VXLAN トンネルエンドポイント（VTEP）間に作成されます。VTEP は VXLAN プロトコルをサポートし、VXLAN カプセル化およびカプセル化解除を実行します。VXLAN セグメントは、2 つの VTEP 間のトンネルと考えることができます。この場合、1 つの VTEP は UDP ヘッダーと IP ヘッダーを持つレイヤ 2 フレームをカプセル化し、トンネルを介して送信します。もう一方の VTEP は、パケットを受信してカプセル化を解除して、レイヤ 2 フレームを取得します。Citrix ADC は、VTEP の一例です。その他の例としては、サードパーティ製のハイパーバイザ、VXLAN 対応の仮想マシン、および VXLAN 対応スイッチがあります。

次の図は、VXLAN トンネルを介して接続された仮想マシンと物理サーバを示しています。



次の図は、VXLAN パケットの形式を示しています。



Citrix ADC 上の VXLAN は、ブロードキャスト、マルチキャスト、および未知のユニキャストフレームの送信にレイヤー 2 メカニズムを使用します。VXLAN は、これらの L2 フレームを送信するために、次のモードをサポートしています。

- ユニキャストモード：このモードでは、Citrix ADC で VXLAN を構成する際に、VTEP の IP アドレスを指定します。Citrix ADC は、ブロードキャスト、マルチキャスト、および未知のユニキャストフレームをレイヤー 3 経由でこの VXLAN のすべての VTEP に送信します。
- マルチキャストモード：このモードでは、Citrix ADC で VXLAN を設定する際に、マルチキャストグループの IP アドレスを指定します。Citrix ADC は、インターネットグループ管理プロトコル（IGMP）プロトコルをサポートしていません。Citrix ADC は、共通のマルチキャストグループ IP アドレスを共有するマルチキャストグループに参加するためにアップストリームルーターに依存します。Citrix ADC は、ブロードキャスト、マルチキャスト、および未知のユニキャストフレームをレイヤー 3 経由でこの VXLAN のマルチキャストグループ IP アドレスに送信します。

レイヤー 2 ブリッジテーブルと同様に、Citrix ADC は、受信した VXLAN パケットの内部ヘッダーと外部ヘッダーに基づいて VXLAN マッピングテーブルを維持します。このテーブルは、リモートホストの MAC アドレスを特定の VXLAN の VTEP IP アドレスにマッピングします。Citrix ADC は、VXLAN マッピングテーブルを使用して、レイヤー 2 フレームの宛先 MAC アドレスを検索します。この MAC アドレスのエントリが VXLAN テーブルに存在する場合、Citrix ADC は、VXLAN プロトコルを使用して、レイヤー 3 上のレイヤー 2 フレームを、VXLAN のマッピングエントリで指定されたマッピングされた VTEP IP アドレスに送信します。

VXLAN は VLAN と同様に機能するため、VLAN を分類パラメータとしてサポートする Citrix ADC 機能のほとんどは VXLAN をサポートしています。これらの機能には、VXLAN VNI を指定するオプションの VXLAN パラメータ設定が含まれます。

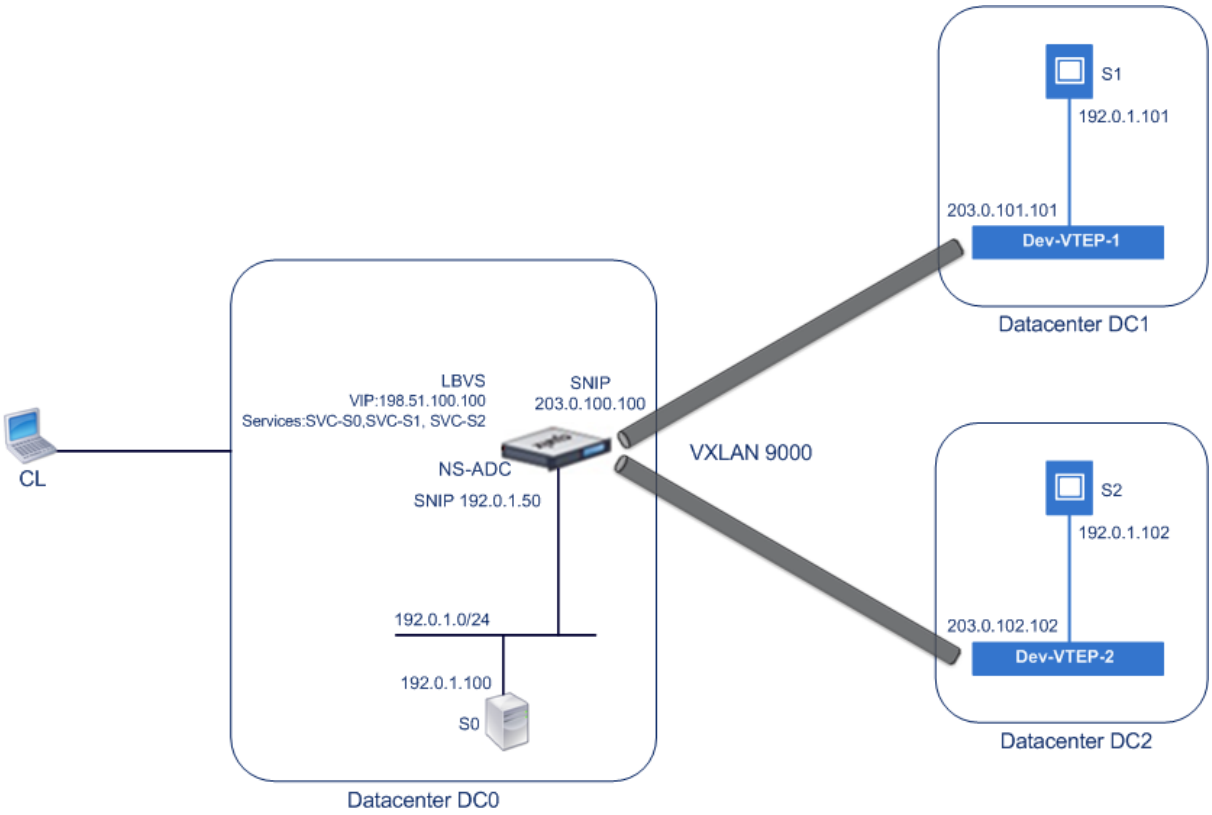
高可用性（HA）構成では、VXLAN 構成がセカンダリノードに伝播または同期されます。

VXLAN の使用例**:** データセンター間の負荷分散

Citrix ADC の VXLAN 機能を理解するために、サンプルコーポレーションが www.example.com のサイトをホストする例を考えてみます。アプリケーションの可用性を確保するため、サイトは S0、S1、S2 の 3 つのサーバーでホストされます。Citrix ADC NS-ADC 上の負荷分散仮想サーバー LBVS を使用して、これらのサーバーの負荷分散を行います。S0、S1、および S2 は、それぞれデータセンターの DC0、DC1、および DC2 に存在します。DC0 では、サーバ S0 が NS-ADC に接続されています。

S0 は物理サーバであり、S1 と S2 は仮想マシン（VM）です。S1 は、データセンター DC1 の仮想化ホストデバイス Dev-VTEP-1 で実行され、S2 は DC2 のホストデバイス Dev-VTEP-2 で実行されます。NS-ADC、開発 VTEP-1、および開発 VTEP-2 は、VXLAN プロトコルをサポートしています。

S0、S1、および S2 は、同じプライベートサブネット 192.0.1.0/24 の一部です。S0、S1、S2 は共通のブロードキャストドメインの一部であり、VXLAN 9000 は NS-ADC、Dev-VTEP-1、および Dev-VTEP-2 上で設定されます。サーバ S1 と S2 は、それぞれ開発-VTEP-1 と開発-VTEP-2 上の VXLAN9000 の一部で構成されています。



次の表に、この例で使用される設定を示します。

VXLAN 設定。

NS-ADC 上のサービス SVC-S0、SVC-S1、および SVC-S2 は、S0、S1、および S2 を表します。これらのサービスが構成されるとすぐに、NS-ADC は S0、S1、および S2 に対する ARP 要求をブロードキャストして IP-to-MAC マッピングを解決します。これらの ARP 要求は、VXLAN 9000 を介してデベロッパー VTEP-1 およびデベロッパー VTEP-2 に送信されます。

次に、S2 の ARP 要求を解決するためのトラフィックフローを示します。

1. NS-ADC は、S2 の ARP 要求をブロードキャストして、IP/MAC マッピングを解決します。このパケットには、次のものがあります。
   - 送信元 IP アドレス = サーバに対する SNIP サブネット IP アドレス (192.0.1.50)
   - 送信元 MAC アドレス = パケットの送信元 NS-ADC のインターフェイスの MAC アドレス = NS-MAC-1
2. NS-ADC は、パケットを次のヘッダーでカプセル化することによって、VXLAN 9000 経由で送信する ARP パケットを準備します。
   - ID（VNI）が 9000 の VXLAN ヘッダー
   - 標準 UDP ヘッダー、UDP チェックサムを 0×0000 に設定し、宛先ポートを 4789 に設定します。
3. NS-ADC は、生成されたカプセル化されたパケットを VXLAN-9000 上の開発 VTEP-1 と開発 VTEP-2 に送信します。カプセル化されたパケットには、次のものがあります。
   - 送信元 IP アドレス = SNIP-VTEP-0 (203.0.100.100)。
4. Dev-VTEP-2 は UDP パケットを受信し、UDP ヘッダーをカプセル化解除します。このヘッダーから Dev-VTEP-2 は、パケットが VXLAN 関連パケットであることを学習します。次に、Dev-VTEP-2 は VXLAN ヘッダーのカプセル化を解除し、パケットの VXLAN ID を学習します。結果のパケットは S2 の ARP 要求パケットです。これはステップ 1 と同じです。
5. VXLAN パケットの内部ヘッダーと外部ヘッダーから、VXLAN9000 の MAC アドレス（NS-MAC-1）と SNIP-VTEP-0（203.0.100.100）のマッピングを示す VXLAN マッピングテーブルにエントリを作成します。
6. 開発-VTEP-2 は ARP パケットを S2 に送信します。S2 の応答パケットは開発-VTEP-2 に到達します。Dev-VTEP-2 は VXLAN マッピングテーブル内で検索を実行し、宛先 MAC アドレス NS-MAC-1 の一致を取得します。このリリースでは、NS-MAC-1 が VXLAN 9000 経由で SNIP-VTEP-0（203.0.100.100）を介して到達可能であることを認識しています。
7. S2 は MAC アドレス（MAC-S2）で応答します。ARP 応答パケットには、次のものがあります。
   - 宛先 IP アドレス = サーバに対する SNIP サブネット IP アドレス (192.0.1.50)
   - 宛先 MAC アドレス = NS-MAC-1
8. S2 の応答パケットは開発-VTEP-2 に到達します。Dev-VTEP-2 は VXLAN マッピングテーブル内で検索を実行し、宛先 MAC アドレス NS-MAC-1 の一致を取得します。このリリースでは、NS-MAC-1 が VXLAN 9000 経由で SNIP-VTEP-0（203.0.100.100）を介して到達可能であることを認識しています。デベロッパー VTEP-2 は、ARP 応答を VXLAN ヘッダーと UDP ヘッダーでカプセル化し、その結果パケットを NS-ADC の SNIP-VTEP-0（203.0.100.100）に送信します。
9. NS-ADC は、パケットを受信すると、VXLAN および UDP ヘッダーを削除してパケットのカプセル化を解除します。結果のパケットは S2 の ARP 応答です。NS-ADC は、それが S2 の MAC アドレスのための VXLAN マッピングテーブルを更新します (MAC-S2) 開発 VTEP-2 の IP アドレス (203.0.102.102) VXLAN9000. NS-ADC はまた、S2 の MAC アドレス（MAC-S2）で S2 の IP アドレス（192.0.1.102）の ARP テーブルを

更新します。

この例では、仮想サーバー LBVS をロードバランシングするためのトラフィックフローを次に示します。

1. クライアント CL は NS-ADC の LBVS に要求パケットを送信します。要求パケットには次のものがあります。
   - 送信元 IP アドレス = クライアント CL の IP アドレス (198.51.100.90)
   - 宛先 IP アドレス = LBVS の IP アドレス (VIP) = 198.51.110.100
2. NS-ADC の LBVS は要求パケットを受信し、ロードバランシングアルゴリズムはデータセンター DC2 のサーバ S2 を選択します。
3. NS-ADC は要求パケットを処理し、宛先 IP アドレスを S2 の IP アドレスに変更し、送信元 IP アドレスを NS-ADC で設定されたサブネット IP（SNIP）アドレスのいずれかに変更します。要求パケットには次のものがあります。
   - 送信元 IP アドレス = NS-ADC 上のサブネット IP アドレス = サーバに対する SNIP（192.0.1.50）
   - 宛先 IP アドレス = S2 の IP アドレス (192.0.1.102)
4. NS-ADC は、ブリッジテーブル内で S2 の VXLAN マッピングエントリを検出します。このエントリは、S2 が VXLAN 9000 経由でデベロッパー VTEP-2 を介して到達可能であることを示しています。
5. NS-ADC は、パケットを次のヘッダーでカプセル化することによって、VXLAN 9000 経由で送信するパケットを準備します。
   - ID（VNI）が 9000 の VXLAN ヘッダー
   - 標準 UDP ヘッダー、UDP チェックサムを 0×0000 に設定し、宛先ポートを 4789 に設定します。
6. NS-ADC は、生成されたカプセル化されたパケットを Dev-VTEP-2 に送信します。要求パケットには次のものがあります。
   - 送信元 IP アドレス = SNIP アドレス = SNIP アドレス-VTEP-0 (203.0.100.100)
   - 宛先 IP アドレス = デベロッパー VTEP-2 の IP アドレス (203.0.102.102)
7. Dev-VTEP-2 は UDP パケットを受信し、UDP ヘッダーをカプセル化解除します。このヘッダーから Dev-VTEP-2 は、パケットが VXLAN 関連パケットであることを学習します。次に、Dev-VTEP-2 は VXLAN ヘッダーのカプセル化を解除し、パケットの VXLAN ID を学習します。結果のパケットは、手順 3 と同じパケットです。
8. その後、Dev-VTEP-2 はパケットを S2 に転送します。
9. S2 は要求パケットを処理し、NS-ADC の SNIP アドレスに応答を送信します。応答パケットには次のものがあります。
   - 送信元 IP アドレス = S2 の IP アドレス (192.0.1.102)
   - 宛先 IP アドレス = NS-ADC 上のサブネット IP アドレス = サーバに対する SNIP（192.0.1.50）
10. Dev-VTEP-2 は、NS-ADC がステップ 4 および 5 で要求パケットをカプセル化するのと同じ方法で、応答パケットをカプセル化します。その後、Dev-VTEP-2 は、カプセル化された UDP パケットを NS-ADC のサーバ向け SNIP アドレス（192.0.1.50）に送信します。
11. NS-ADC は、カプセル化された UDP パケットを受信すると、ステップ 7 で Dev-VTEP-2 がパケットをカプセル化解除するのと同じ方法で UDP ヘッダーと VXLAN ヘッダーを削除してパケットのカプセル化を解除します。結果のパケットは、ステップ 9 と同じ応答パケットです。
12. 次に、NS-ADC はセッションテーブルを使用して仮想サーバー LBVS をロードバランシングし、応答パケットをクライアント CL に転送します。応答パケットには次のものがあります。

- 送信元 IP アドレス = クライアント CL の IP アドレス (198.51.100.90)
- 宛先 IP アドレス = LBVS の IP アドレス (VIP) (198.51.110.100)

VXLAN の設定で考慮すべきポイント

Citrix ADC で VXLAN を構成する前に、次の点を考慮してください。

- Citrix ADC では、最大 2048 の VXLAN を構成できます。
- VXLAN は、クラスタではサポートされません。
- リンクローカル IPv6 アドレスは、各 VXLAN に対して設定できません。
- Citrix ADC は、マルチキャストグループを形成するためのインターネットグループ管理プロトコル（IGMP）プロトコルをサポートしていません。Citrix ADC は、アップストリームルーターの IGMP プロトコルを使用して、共通のマルチキャストグループ IP アドレスを共有するマルチキャストグループに参加します。VXLAN ブリッジテーブルエントリの作成時にマルチキャストグループ IP アドレスを指定できますが、マルチキャストグループはアップストリームルータで設定する必要があります。Citrix ADC は、ブロードキャスト、マルチキャスト、および未知のユニキャストフレームをレイヤー 3 経由でこの VXLAN のマルチキャストグループ IP アドレスに送信します。次に、アップストリームルータは、マルチキャストグループの一部であるすべての VTEP にパケットを転送します。
- VXLAN カプセル化により、各パケットに 50 バイトのオーバーヘッドが追加されます。

  外部イーサネットヘッダー (14) + UDP ヘッダー (8) + IP ヘッダー (20) + VXLAN ヘッダー (8) = 50 バイト

  フラグメンテーションとパフォーマンスの低下を回避するには、VXLAN パケットの 50 バイトのオーバーヘッドを処理するために、VXLAN VTEP デバイスを含む VXLAN パスウェイ内のすべてのネットワークデバイスの MTU 設定を調整する必要があります。

  重要: ジャンボフレームは、Citrix ADC VPX 仮想アプライアンス、Citrix ADC SDX アプライアンス、および Citrix ADC MPX 15000/17000 アプライアンスではサポートされていません。これらのアプライアンスは、わずか 1500 バイトの MTU サイズをサポートしており、VXLAN パケットの 50 バイトのオーバーヘッドを処理するように調整することはできません。これらのアプライアンスの 1 つが VXLAN 経路内にあるか、VXLAN VTEP デバイスとして機能している場合、VXLAN トラフィックが断片化したり、パフォーマンスが低下したりすることがあります。
- Citrix ADC SDX アプライアンスでは、VXLAN パケットに対して VLAN フィルタリングが機能しません。
- VXLAN では MTU 値を設定できません。
- インターフェイスを VXLAN にバインドすることはできません。

構成の手順

Citrix ADC アプライアンス上で VXLAN を構成するには、次の作業を行います。

- **VXLAN** エンティティを追加します。正の整数で一意に識別される VXLAN エンティティを作成します。これは VXLAN ネットワーク識別子 (VNI) とも呼ばれます。この手順では、VXLAN プロトコルが実行されているリモート VTEP の宛先 UDP ポートを指定することもできます。デフォルトでは、VXLAN エンティティの宛先 UDP ポートパラメータは 4789 に設定されています。この UDP ポート設定は、この VXLAN のすべてのリモート VTEP の設定と一致する必要があります。VLAN をこの VXLAN にバインドすることもできます。バインドされたすべての VLAN のトラフィック（ブロードキャスト、マルチキャスト、不明なユニキャストを含む）は、この VXLAN 上で許可されます。VXLAN にバインドされた VLAN がない場合、Citrix ADC は、この VXLAN 上の他の VXLAN の一部ではないすべての VLAN のトラフィックを許可します。
- ローカル **VTEP IP** アドレスと **VXLAN** エンティティにバインドします。設定済みの SNIP アドレスの 1 つを VXLAN にバインドし、発信 VXLAN パケットを送信します。
- ブリッジ可能なエントリを追加します。作成する VXLAN の VXLAN の ID とリモート VTEP IP アドレスを指定するブリッジ可能なエントリを追加します。
- （任意）設定された **VXLAN** に異なるフィーチャエンティティをバインドします。VXLAN は VLAN と同様に機能しますが、VLAN を分類パラメータとしてサポートする Citrix ADC 機能のほとんどは、VXLAN もサポートしています。これらの機能には、VXLAN VNI を指定するオプションの VXLAN パラメータ設定が含まれます。
- （任意）**VXLAN** マッピングテーブルを表示します。VXLAN マッピングテーブルを表示します。このテーブルには、特定の VXLAN の VTEP IP アドレスへのリモートホストの MAC アドレスのマッピングエントリが含まれます。つまり、VXLAN マッピングでは、ホストが特定の VXLAN 上の VTEP を介して到達可能であることを示します。Citrix ADC は VXLAN マッピングを学習し、受信した VXLAN パケットからマッピングテーブルを更新します。Citrix ADC は、VXLAN マッピングテーブルを使用して、レイヤー 2 フレームの宛先 MAC アドレスを検索します。この MAC アドレスのエントリが VXLAN テーブルに存在する場合、Citrix ADC は、VXLAN プロトコルを使用して、レイヤー 3 上のレイヤー 2 フレームを、VXLAN のマッピングエントリで指定されたマッピングされた VTEP IP アドレスに送信します。

CLI のプロシージャ

CLI を使用して VXLAN エンティティを追加するには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

- **vxlan** を追加する <id>
- **vxlan** を表示する <id>

CLI を使用してローカル VTEP IP アドレスを VXLAN にバインドするには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

- **bind vxlan** <id> -**SrcIP** <IPaddress>
- **show vxlan** <id>

CLI を使用してブリッジ可能なを追加するには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

- マック -**bridgetable** を追加 < macaddress>-**vxlan** < ID>-**vtep** < IP アドレス >

- **show bridgetable**

コマンドラインを使用して VXLAN 転送テーブルを表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **show bridgetable**

GUI のプロシージャ

GUI を使用して VXLAN エンティティを追加し、ローカル VTEPIP アドレスをバインドするには:

System > **Network** > **VXLANs** に移動して新しい VXLAN エンティティを追加するか、既存の VXLAN エンティティを変更します。

GUI を使用してブリッジ可能なを追加するには、次の手順を実行します。

[システム] > **[ネットワーク]** > **[ブリッジテーブル]** に移動し、VXLAN ブリッジテーブルエントリを追加または変更するときに、次のパラメータを設定します。

- MAC
- VTEP
- VXLAN ID

GUI を使用して VXLAN 転送テーブルを表示するには、次の手順を実行します。

[システム] > **[ネットワーク]**> **[ブリッジテーブル]** に移動します。

```
Example
> add vxlan 9000
Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
   203.0.101.101

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
   203.0.102.102

Done
```


VXLAN での **IPv6** ダイナミックルーティングプロトコルのサポート

Citrix ADC アプライアンスは、VXLAN 用の IPv6 動的ルーティングプロトコルをサポートしています。VTYSH コマンドラインから VXLAN 上でさまざまな IPv6 ダイナミックルーティングプロトコル（OSPFv3、RIPng、BGP など）を設定できます。VXLAN 上で IPv6 ダイナミックルーティングプロトコルを有効または無効にするために、オプションの IPv6 ダイナミックルーティングプロトコルが VXLAN コマンドセットに追加されました。VXLAN で IPv6 ダイナミックルーティングプロトコルを有効にした後、VTYSH コマンドラインを使用して VXLAN 上で IPv6 ダイナミックルーティングプロトコルに関連するプロセスを開始する必要があります。

CLI を使用して VXLAN で IPv6 ダイナミックルーティングプロトコルを有効にするには、次の手順を実行します。

- **add vxlan** <ID> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]
- **show vxlan**

```
In the following sample configuration, VXLAN-9000 is created and has
    IPv6 dynamic routing protocols enabled on it. Then, using the VTYSH
    command line, process for the IPv6 OSPF protocol is started on the
    VXLAN.

> add vxlan 9000 -ipv6DynamicRouting ENABLED

Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

Done
> VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface VXLAN-9000
NS(config-if)# ipv6 router OSPF area 3
```

VXLAN-VLAN マップを使用した複数の企業からクラウドへの **VLAN** の拡張

CloudBridge Connector トンネルは、エンタープライズの VLAN をクラウドに拡張するために使用されます。複数の企業から拡張された VLAN は、重複する VLAN ID を持つことができます。各企業の VLAN をクラウド内の一意の VXLAN にマッピングすることで、各企業の VLAN を分離できます。クラウド内の CloudBridge コネクタエンドポイントである Citrix ADC アプライアンスでは、企業の VLAN をクラウド内の一意の VXLAN にリンクする VXLAN-VLAN マップを構成できます。VXLAN は、CloudBridge Connector から同じ VXLAN に企業の複数の

VLAN を拡張するための VLAN タグ付けをサポートします。

複数の企業の VLAN をクラウドに拡張するには、次のタスクを実行します。

1. VXLAN VLAN マップを作成します。
2. クラウド上の Citrix ADC アプライアンス上のネットワークブリッジベースまたは PBR ベースの Cloud-Bridge Connector トンネル構成に VXLAN-VLAN マップをバインドします。
3.（任意）VXLAN 構成で VLAN タグ付けを有効にします。

CLI のプロシージャ

CLI を使用して VXLAN VLAN マップを追加するには、次の手順を実行します。

- **add vxlanVlanMap** <name>
- **show vxlanVlanMap** <name>

CLI を使用して VXLAN および VLAN を VXLAN VLAN マップにバインドするには、次の手順を実行します。

- **bind vxlanVlanMap** <name> [-**vxlan** <positive_integer> -**vlan** <int[-int]> …]
- **show vxlanVlanMap** <name>

CLI を使用して VXLAN-VLAN マップをネットワークブリッジベースの CloudBridge Connector トンネルにバインドするには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

新しいネットワークブリッジを追加する場合:

- **add netbridge** <name> [-**vxlanVlanMap** <string>]
- **show netbridge** <name>

既存のネットワークブリッジを再構成する場合:

- **set netbridge** <name> [-**vxlanVlanMap** <string>]
- **show netbridge** <name>

CLI を使用して VXLAN-VLAN マップを PBR ベースの CloudBridge Connector トンネルにバインドするには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

新しい PBR を追加する場合:

- **add pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** <name>])
- **show pbr** <name>

既存の PBR を再設定する場合:

- **set pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** <name>])
- **show pbr** <name>

CLI を使用して VXLAN に関連するパケットに VLAN タグを含めるには、次の手順を実行します。

コマンドプロンプトで、次のいずれかのコマンドセットを入力します。

新しい VXLAN を追加する場合:

- **add vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

既存の VXLAN を再設定する場合:

- **set vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

GUI のプロシージャ

GUI を使用して VXLAN VLAN マップを追加するには、次の手順を実行します。

[システム] > [ネットワーク] > [**VXLAN VLAN** マップ**]** に移動し、VXLAN VLAN マップを追加します。

GUI を使用して VXLAN-VLAN マップをネットブリッジベースの CloudBridge Connector トンネルにバインドするには、次の手順を実行します。

[システム] > [**CloudBridge Connector**] > [ネットワークブリッジ] に移動し、[VX **LAN VLAN]** ドロップダウンリストから **VXLAN-VLAN** マップを選択し、新しいネットワークブリッジを追加するか、既存のネットワークブリッジを再設定します。

GUI を使用して VXLAN-VLAN マップを PBR ベースの CloudBridge Connector トンネルにバインドするには、次の手順を実行します。

[システム] > [ネットワーク] > [**PBR**] に移動し、[ポリシーベースルーティング (PBR)] タブで [**VXLAN VLAN]** ドロップダウンリストから **VXLAN-VLAN** マップを選択し、新しい PBR を追加するか、既存の PBR を再設定します。

GUI を使用して VXLAN に関連するパケットに VLAN タグを含めるには、次の手順を実行します。

[システム] > [ネットワーク] > [**VXLAN**] に移動し、新しい VXLAN を追加するときに 内部 **VLAN** タグ付けを有効化するか、既存の VXLAN を再構成します。

```
> add vxlanVlanMap VXLANVLAN-DC1

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3000 -vlan 3

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3500 -vlan 4

```



```
Done

>add vxlanVlanMap VXLANVLAN-DC2

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 8000 -vlan 3 4

Done

> set pbr PBR-CBC-DC-1-CLOUD ALLOW  -ipTunnel  CBC-DC-1-CLOUD -
    vxlanVlanMap VXLANVLAN-DC1

Done

> set pbr PBR-CBC-DC-2-CLOUD ALLOW  -ipTunnel  CBC-DC-2-CLOUD -
    vxlanVlanMap VXLANVLAN-DC2

Done
```

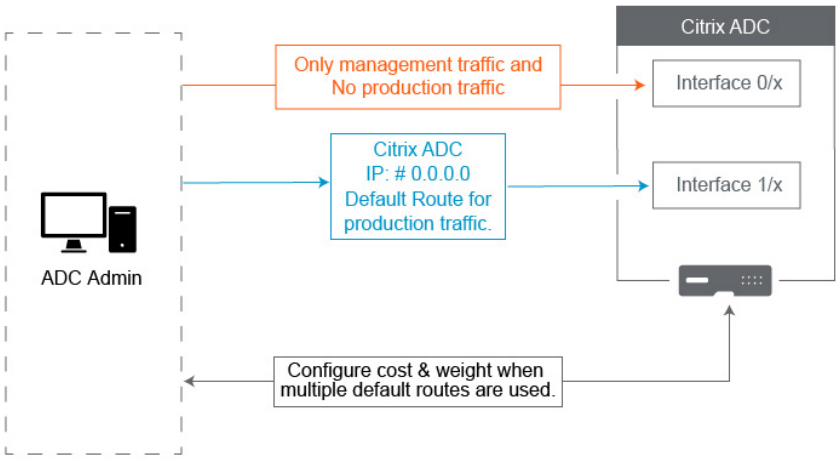
ネットワーク構成のベストプラクティス

October 7, 2021

以下のセクションでは、Citrix ADC アプライアンスのネットワーク機能を構成するためのベストプラクティスについて説明します。

ルーティングとデフォルトルート

Citrix ADC アプライアンスでレイヤー 3 機能を設定するためのベストプラクティスを以下に示します。



- **Citrix ADC** アプライアンスまたは **Citrix SDX** アプライアンスのインターフェイス**0/x** は、本番トラフィックに使用しないでください。MPX または SDX では、0/xという名前のインターフェイスは管理インターフェイスと呼ばれます。これは、管理にこれらのインターフェイスを使用する必要があるという意味ではありません。つまり、これらのインターフェイスは本番トラフィック用に設計されていないということです。1 Gbps の持続的なスループットを達成するために必要なハードウェアバッファと最適化はありません。したがって、デフォルトルートが NSIP と同じサブネットにある場合は、デフォルトルートを変更するか、管理ネットワークの1/xインターフェイスを使用する必要があります。これは、1/xインターフェイスが実稼働 1 Gbps トラフィック用に完全に最適化されているためです。

  > 注:
  >
  > これは、Citrix ADC VPX アプライアンスには適用されません。

  - オプション **1**。インターフェイスに接続しない0/x — インターフェイス0/1からケーブルを取り外します。NetScaler は、他のインターフェイスで NSIP をリッスンします。（注：SVM と XenServer は0/xインターフェイスに対してのみ話すことができるため、これは SDX ではオプションではありません）
  - オプション **2**：次のセクションで説明するように、デフォルトルートを別のインターフェイスに変更します。

- デフォルト **Gateway**（ルート **0.0.0.0**）は、**0/x**インターフェイスではなく、実稼動ネットワーク上に配置する必要があります。NetScaler を最初にセットアップするときに、NSIP、サブネットマスク、ゲートウェイアドレスの入力を求められます。管理者にとっての問題は、インターフェイス 0/1 を使用して管理ネットワーク上にデフォルトルートを設定したことです。

  - あなたのルートが何であるかを確認するには、CLI `show route` で実行します。デフォルト Gateway は、ネットワークとネットマスクが 0.0.0.0 である行の IP です。ゲートウェイがライン 1 にある例を次に示します。


```
> sh route
        Network          Netmask          Gateway/OwnedIP
           State   Traffic Domain  Type
        -------          -------          ---------------
           -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.25.213.65     UP
         0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        UP
         0               PERMANENT
3)      10.25.213.64     255.255.255.192  10.25.213.68     UP
         0               DIRECT
4)      172.16.0.0       255.255.255.0    172.16.0.1       UP
         0               DIRECT

<!--NeedCopy-->
```

- デフォルトゲートウェイに使用されるインターフェイスと VLAN を確認するには、CLI で`sh arp`を使用して ARP テーブルを確認します。`show arp | grep 10.25.213.65`を使用して特定の IP を検索することもできます。次に、ゲートウェイ 10.25.213.65 がインターフェイス 1/1 および VLAN 1 を使用している例を示します。

```
> sh arp
        IP               MAC                Iface VLAN
           Origin     TTL     Traffic Domain
        --               ---                ----- ----
           ------     ---     --------------
1)      127.0.0.1        02:00:18:a4:00:1e  LO/1  1
   PERMANENT  N/A    0
2)      10.25.213.70     02:00:0f:46:00:28  1/1   1
   DYNAMIC    967    0
3)      10.25.213.68     02:00:18:a4:00:1e  LO/1  1
   PERMANENT  N/A    0
4)      10.25.213.67     02:00:0f:46:00:28  1/1   1
   DYNAMIC    641    0
5)      10.25.213.65     00:08:e3:ff:fd:90  1/1   1
   DYNAMIC    483    0
<!--NeedCopy-->
```

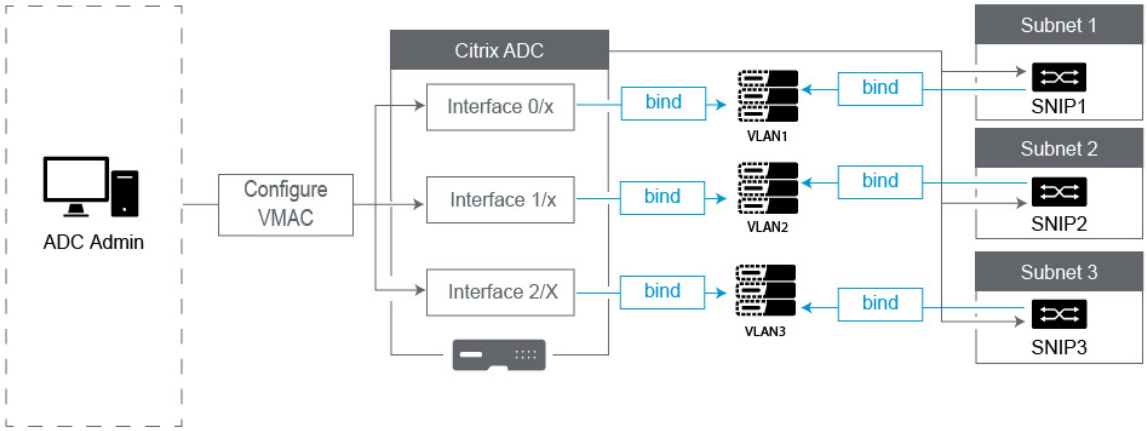
- 実稼働サブネットとインターフェイスでゲートウェイを使用するようにデフォルトルートを変更します。管理ネットワークが Gateway 10.0.0.1 で 10.0.0.0/24 で、運用ネットワークが Gateway 10.1.1.1 で 10.1.1.0/24 であると仮定します。次のように設定してください:

* SNIP:（管理アクセス無効）10.1.1.2
* NSIP:（管理アクセスが有効になっている）10.0.0.2
* デフォルトルート:0.0.0.0 0.0.0.0 10.1.1.1 ([システム] > [ネットワーク] > [ルート])。これは、NSIP ネットワークではなく、SNIP ネットワーク上のルーターを使用します。

> 注:
>
> デフォルト Gateway を変更すると、スタティックルート、ポリシーベースルートを設定するか、MAC ベースフォワーディングを有効にしない限り、管理トラフィックが切断される可能性があります。

インターフェイス、チャネル、**VLAN**

Citrix ADC アプライアンスでレイヤー 2 機能を設定するためのベストプラクティスを以下に示します。



- **VLAN 1** を含む複数のインターフェイス/チャネルを同じ **VLAN** に接続しないでください。
  - VLAN を適切に設定しないと、ネットワーク内で予期しないパケットルーティングが発生し、同じ VLAN（ネイティブまたはタグ付き）を持つアクティブインターフェイスが複数あるときはいつでも、レイヤ 2 ループが発生する可能性があります。
  - デフォルトでは、すべてのインターフェイスおよびチャネルはネイティブ VLAN 1 上にあります。これにより、次の 2 つの問題が発生する可能性があります。
    * NetScaler は、受信したすべてのトラフィックが同じネットワーク上にあると考えているため、任意のインターフェイスを使用してトラフィックを送信します。データを送信したインターフェイスに異なるネイティブ VLAN がある場合、トラフィックは予想どおりにルーティングされません。
    * NetScaler が 1 つのポートでブロードキャストパケットを受信すると、別のポートで再送信することがあります。両方のスイッチポートが同じ VLAN 上にある場合は、レイヤ 2 ループを作成しました。

- VLAN 1 からインターフェイス/チャネルを削除するには、次の手順を実行します。
  - スイッチインターフェイス/ポートチャネルでネイティブ VLAN を使用していない場合。NetScaler インターフェイス/チャネルのネイティブ VLAN を、999 などの未使用の VLAN 番号に変更します。レイヤ 2 ループが発生するため、複数のチャネルまたはインターフェイスに同じ未使用の VLAN 番号を使用しないでください。
  - スイッチインターフェイス/ポートチャネルでネイティブ VLAN を使用している場合。NetScaler インターフェイス/チャネルのネイティブ VLAN を一致するように変更します。ただし、同じ VLAN 上に複数のアクティブなインターフェイスまたはチャネルが存在しないように注意してください。そうすると、レイヤ 2 ループが作成されます。
  - ネイティブ VLAN は削除できません。代わりに、インターフェイスまたはチャネルの TagAll を変更したり、TagAll を設定したりできます。スイッチポートにタグなしのネイティブ VLAN が設定されていない場合は、インターフェイスで tagall を有効にして、高可用性ハートビートパケットにタグを付けます。
- インターフェイス上のネイティブ VLAN を表示するには、CLI で`sh interface`を実行します。これにより、インターフェイスが TAGALL オプションを使用しているかどうかも通知されます。

- インターフェイスを **VLAN** にバインドする -NetScaler は、デフォルトでは、新しい VLAN をインターフェイスに接続しません。つまり、VLAN はインターフェイスにバインドするまで使用されません。新しい VLAN がインターフェイスにバインドされておらず、その VLAN がタグ付けされている場合、NetScaler はその VLAN からのすべてのインバウンドトラフィックをドロップします。また、同じ VLAN を複数のインターフェイスにバインドしないでください。
  - サブネットを VLAN にバインドします。NetScaler は、一般的なルーターのように動作しません。ほとんどのルータは IP をインターフェイスに接続します。NetScaler では、特に構成されていない限り、IP はインターフェイス上でフローティングします。したがって、NetScaler が特定の VLAN 経由で送信することを保証するサブネット（特に NetScaler がそのトラフィックを開始しているとき）は、そのサブネット内の SNIP を VLAN にバインドする必要があります。
  - これに対して一般的な議論は、以前は正常に動作していましたが、サブネットを VLAN にバインドしないと解決できないということです。これは、NetScaler がトラフィックを送信する VLAN を学習するためによく発生しますが、ARP テーブルの構築には時間がかかることがあります。リブートまたはファームウェアアップグレード後、ARP テーブルの構築が再び開始されると、最初は学習されるため、デフォルトルートなど、希望とは異なるパスを使用している可能性があります。SNIP を VLAN にバインドすることで、どのパスを取るべきかを指示するのが最善です。SNIP が VLAN にバインドされると、その SNIP のサブネット全体が VLAN にバインドされます。
  - すべての SNIP が 1 つの VLAN にバインドされていること（ただし、サブネット内に複数の SNIP がある場合は、1 つの SNIP だけをバインドする必要があります）、VLAN が 1 つのインターフェイスまたはチャネルにバインドされていることを確認します。また、すべてのサブネットに SNIP を設定するのが最善ですが、SNIP を持たない宛先サブネットには、最も具体的なルートが使用されるため必要ありません。

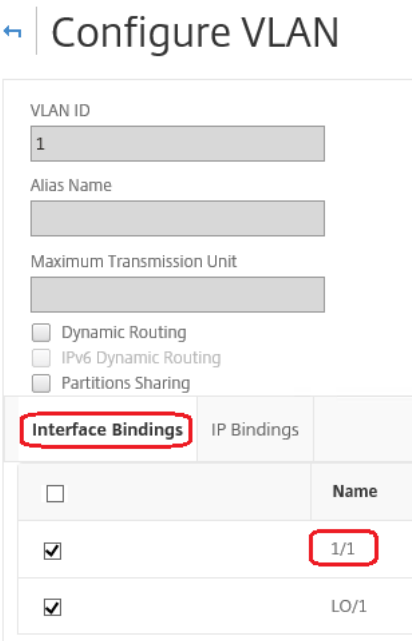
- サブネットで使用される VLAN とインターフェイスを識別するには、次の手順を実行します。
  1. [システム] **>** [ネットワーク] **>** **[VLAN]** に移動します。
  2. 次の手順で説明する正しい IP アドレスが見つかるまで、設定済みの各 VLAN を編集します。
  3. [IP Bindings] タブをクリックして、どの IP、つまりどのサブネットがバインドされ、この VLAN を使用しているかを確認します。
  4. IP がバインドされている VLAN（デフォルトルートのサブネット内）を特定したら、[Interface Bindings] をクリックします。この VLAN にバインドされた各インターフェイスまたはチャネルが使用されます。

例

デフォルトルートは`0.0.0.0 0.0.0.0 10.1.1.1`であると仮定します。

10.0.0.5 と 10.1.1.69 の 2 つの SNIP があるとします。10.1.1.69 はデフォルトルートのサブネットにあるので、それが探したいものです。以下のスクリーンショットでは、VLAN 1 を確認しており、IP 10.1.1.69 がこの VLAN にバインドされていることがわかっているので、正しい VLAN を確認しています。

次に、[インターフェイスのバインディング] をクリックします。VLAN インターフェイスバインディングでは、インターフェイス`1/1`がこのサブネットに使用されているため、デフォルトルートに使用されます。



> 注:
>
> VLAN にバインドされた IP がない場合は、デフォルトで VLAN 1 が送信されます。この場合は、VLAN 1 にバインドされているインターフェイスを調べます。これは、新しい VLAN に IP をバインドしない限り、NetScaler は開始トラフィックに構成された VLAN を使用しないことを意味します。

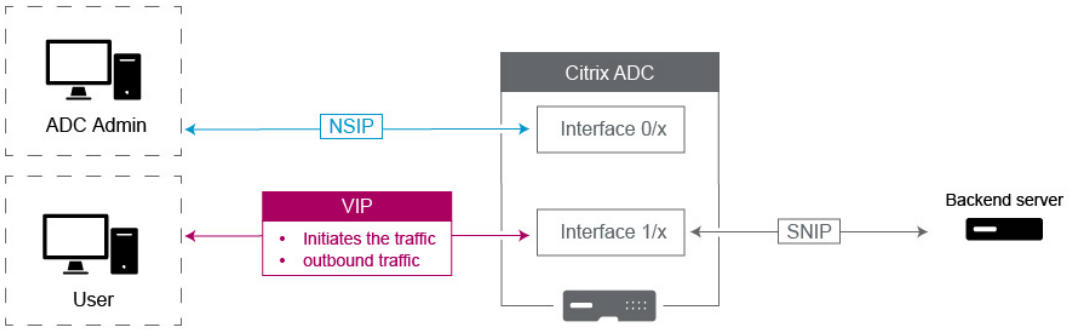
ARP の無償着信

GARP が機能しない場合は、VMAC を使用します。デフォルトでは、NetScaler は GARP を使用して IP と MAC アドレスのバインディングを他のネットワークデバイスに通知します。これは通常問題なく動作しますが、NetScaler でさらに多くのサービスを作成すると、HA ペアでフェイルオーバーするときに問題が発生する可能性があります。最も一般的な問題は、一部のネットワークデバイスが新しい MAC アドレスで ARP テーブルを更新していないために、フェイルオーバーした NetScaler でサービスがダウンしたままになることです。これは、ARP テーブルをチェックして、MAC アドレスが現在のプライマリ NetScaler の MAC アドレスと一致しているかどうかを確認することで簡単に確認できます。この現象が発生すると、ネットワークデバイスの一部が優先する GARP アドバタイズメントの数を制限している可能性が高くなります。この場合、すべてのアクティブなインターフェイスまたはチャネルで VMAC を設定する必要があります。NetScaler で大規模な構成を行うことが予想される場合は、初期展開時にすべてのインターフェイスとチャネルに対して VMAC を構成することをお勧めします。

> 注:
>
> デフォルトルートで使用されるインターフェイスまたはチャネルに VMAC を設定することを忘れないでください。

Citrix ADC が所有する IP アドレス

このセクションでは、Citrix ADC が所有する IP アドレスを構成するためのベストプラクティスについて説明します。



- **Citrix ADC IP**（**NSIP**）：一般的にこの IP は、高可用性またはクラスタ環境における個々の NetScaler に固有の唯一の IP であるため、管理に使用されます。また、LDAP、RADIUS、およびユーザースクリプトモニターのトラフィック（LDAP モニターや StoreFront モニターなど）が NSIP から送信されるため、NSIP がバインドされている VLAN とインターフェイス（デフォルトのネイティブ VLAN 1）を介してルーティングされることに注意してください。SNIP から発信する LDAP および RADIUS トラフィックが必要な場合は、バックエンドサーバー用の LB 仮想サーバーを作成します。

- サブネット **IP (SNIP)**: この IP アドレスは、バックエンドサーバーとの通信を開始するために使用され、常にトラフィックを開始します。つまり、次の場合にトラフィックの宛先になる可能性があります。
  - NetScaler でレイヤー 3 ルーティングを行うときに、他のデバイスのゲートウェイアドレスとして使用できます。
  - 有効にすると、GUI、SSH、SNMP へのアクセスなどの管理サービスを受け入れることができます。
- 仮想 **IP (VIP)**: VIP は、発信トラフィックの開始に使用されないという点で一意です。これは、トラフィックのみを受信することを目的としています。トラフィックを受信すると、クライアントに応答してアウトバウンドのトラフィックを送信します。つまり、VIP アドレスは発信トラフィックを開始しません。

これは、LB 仮想サーバーなどで使用されるバックエンドサーバーと通信するためのソースとして使用されていないことを意味します。

SNIP アドレスから Citrix ADC FreeBSD データトラフィックをソースするように構成します

October 7, 2021

一部の Citrix ADC データ機能は、Citrix ADC OS ではなく基盤となる FreeBSD OS で実行されます。このため、これらの機能は、SNIP アドレスからではなく、Citrix ADC IP（NSIP）アドレスから送信されたトラフィックを送信します。セットアップにすべての管理トラフィックとデータトラフィックを分離する構成がある場合、NSIP アドレスからデータトラフィックを調達することは望ましくありません。

以下の Citrix ADC データ機能は、基盤となる FreeBSD OS で実行され、Citrix ADC IP（NSIP）アドレスから送信されたトラフィックを送信します。

- スクリプト可能なモニターの負荷分散
- GSLB 自動同期

この問題を解決するには、グローバルレイヤー 2 パラメーター `useNetprofileBSDtraffic`を使用できます。このパラメーターを有効にすると、Citrix ADC 機能は、機能に関連付けられたネットプロファイル内の SNIP アドレスの 1 つから送信されたトラフィックを送信します。

はじめに

SNIP アドレスから Citrix ADC 機能に関連するトラフィックを送信するように Citrix ADC アプライアンスを構成する前に、次の点に注意してください。

- 現在、グローバルレイヤー 2 パラメーター `useNetprofileBSDtraffic` は、スクリプト可能なモニターの負荷分散でのみサポートされています。

  SNIP アドレスから GSLB 自動同期トラフィックを送信するように Citrix ADC アプライアンスを構成する場合、回避策として拡張 ACL ルールと RNAT ルールを使用できます。

- 負荷分散スクリプト可能モニターの useNetprofileBSDtraffic サポートは、関連サービスにバインドされたネットプロファイルにのみ適用されます。useNetprofileBSDtraffic サポートは、関連するサービスグループにバインドされたネットプロファイルには適用されません。

  つまり、Citrix ADC アプライアンスは、サービスグループにバインドされたネットプロファイルの SNIP アドレスを使用して、負荷分散のスクリプト可能なモニタートラフィックを調達しません。

- useNetprofileBSDtraffic このサポートは SSL サービスには適用されません。

  つまり、Citrix ADC アプライアンスは、負荷分散スクリプト可能なモニタトラフィックをソースするために、SSL サービスにバインドされたネットプロファイルからの SNIP アドレスを使用しません。

SNIP アドレスからスクリプト可能なモニタートラフィックを送信するように **Citrix ADC** アプライアンスを構成する

SNIP アドレスからスクリプト可能なモニタートラフィックを送信するように Citrix ADC アプライアンスを構成することは、次のタスクで構成されます。

- グローバルレイヤー 2 パラメーター useNetprofileBSDtrafficを有効にします。
- ネットプロファイルを作成し、少なくとも 1 つの SNIP アドレスをそれにバインドします。
- スクリプト可能なモニターを使用している負荷分散サービスにネットプロファイルをバインドします。

CLI を使用してレイヤー **2** パラメーター **useNetprofileBSDtraffic** を有効にするには:

コマンドプロンプトで入力します。

- **set l2param** -**useNetprofileBSDtraffic** (**ENABLED** / **DISABLED**)
- **show l2param**

CLI を使用してネットプロファイルを作成し、**SNIP** アドレスをそれにバインドするには:

コマンドプロンプトで入力します。

- **add netProfile** <name> -**srcIP** <string>
- **show netProfile**

CLI を使用してネットプロファイルを負荷分散サービスにバインドするには:

コマンドプロンプトで入力します。

- セットサービス < 名前 >**-netProfile** < 文字列 >
- ショーサービス < 名前 >

構成例

次の設定例では、Citrix ADC アプライアンスが SNIP アドレスからスクリプト可能なモニタートラフィックを送信できるようにします。ネットプロファイル NETPROFILE-1 は、SNIP アドレス 198.51.100.20 がバインドされた状態で構成されています。A user/scriptable モニター USER-MONITOR-1 が作成され、負荷分散サービス SERVICE-1 に

バインドされます。NETPROFILE-1 は SERVICE-1 にバインドされています。Citrix ADC アプライアンスは、SNIP アドレス 198.51.100.20 から USER-MONITOR-1 のすべてのスクリプト可能なモニターパケットを送信します。

```
set l2param -useNetprofileBSDtraffic ENABLED

set netprofile NETPROFILE-1 -srcip 198.51.100.20

add lb monitor USER-MONITOR-1 USER -scriptName nsftp.pl -scriptArgs ”
    file=Index.png;user=nsroot;password=nsroot” -dispatcherIP 127.0.0.1
    -dispatcherPort 3013 -destIP 203.0.113.90 -destPort 21

bind service SERVICE-1 -monitorName USER-MONITOR-1

set service SERVICE-1 -netProfile NETPROFILE-1

<!--NeedCopy-->
```

SNIP アドレスから **GSLB** 自動同期トラフィックを送信するように **Citrix ADC** アプライアンスを構成します

SNIP アドレスから GSLB 自動同期トラフィックを送信するように Citrix ADC アプライアンスを構成することは、次の回避策タスクで構成されます。

- 拡張 **ACL** ルールを作成します。拡張 ACL ルールは、GSLB 自動同期パケットを識別します。この識別は、送信元 IP アドレスと宛先 IP アドレスに基づいています。
- **ACL** を適用します。ACL を適用すると、新しく作成された ACL ルールがアクティブになります。
- **ACL** ベースの **RNAT** ルールを作成します。RNAT ルールは、これらのパケットの送信元 IP アドレスを NSIP アドレスから SNIP アドレスに変更します。

> 注:
>
> 高可用性またはクラスターセットアップでは、セットアップのすべての NSIP アドレスに ACL および RNAT ルールを追加する必要があります。

CLI を使用して拡張 **ACL** を作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **add acl** <aclname> **ALLOW** -**srcIP** = <NSIP address> -**destIP** = <destination IP address of the packets>
- **show acl** <aclName>

CLI を使用して拡張 **ACL** を適用するには、次の手順を実行します。

コマンドプロンプトで入力します。

- **apply acls**

CLI を使用して **ACL** ベースの **RNAT** ルールを作成するには:

コマンドプロンプトで入力します。

- **add rnat** <name> <aclname>
- **bind rnat** <name> -**natIP** <SNIP address - source IP address for the packets>
- **show rnat** <name>

構成例

次の設定例では、Citrix ADC アプライアンスが SNIP アドレスから GSLB 自動同期トラフィックを送信できるようにします。ACL-2 は、NSIP アドレス 192.0.1.20 から送信され、GSLB サイト IP アドレス 203.0.113.20 宛ての GSLB 自動同期パケットを識別します。RNAT-2 は、これらの識別されたパケットの送信元 IP アドレスを SNIP アドレス 198.51.100.20 に変更します。

```
add acl ACL-2 ALLOW -srcIP = 192.0.1.20  -destIP = 203.0.113.20

apply acls

add rnat RNAT-2 ACL-2

bind rnat RNAT-2 -natIP 198.51.100.20
<!--NeedCopy-->
```

優先負荷分散

October 7, 2021

プライオリティ負荷分散機能を使用すると、プライオリティ負荷分散仮想サーバにバインドされている各サービスまたはサービスグループにプライオリティ番号を割り当てることができます。番号が最も小さいサービスまたはサービスグループが最も高いプライオリティを持ちます。アプリケーショントラフィックは、このサービスまたはサービスグループが UP である限り、このサービスまたはサービスグループにだけ分散されます。次のプライオリティ番号が割り当てられたサービスまたはサービスグループは、プライオリティが最も高いサービスグループ内のすべてのサービスまたはメンバーが DOWN の場合にだけ動作可能になります。ただし、優先度が最も高いサービスまたはサービスグループのメンバーのいずれかが再び使用可能になると、トラフィックはそのサービスまたはサービスグループにリダイレクトされます。

たとえば、優先負荷分散仮想サーバーにバインドされているサービスグループ SVG1、SVG2、および SVG3 について考えてみます。プライオリティグループの最大数は 3 に設定されています。各グループの優先順位は、次のように

割り当てます。

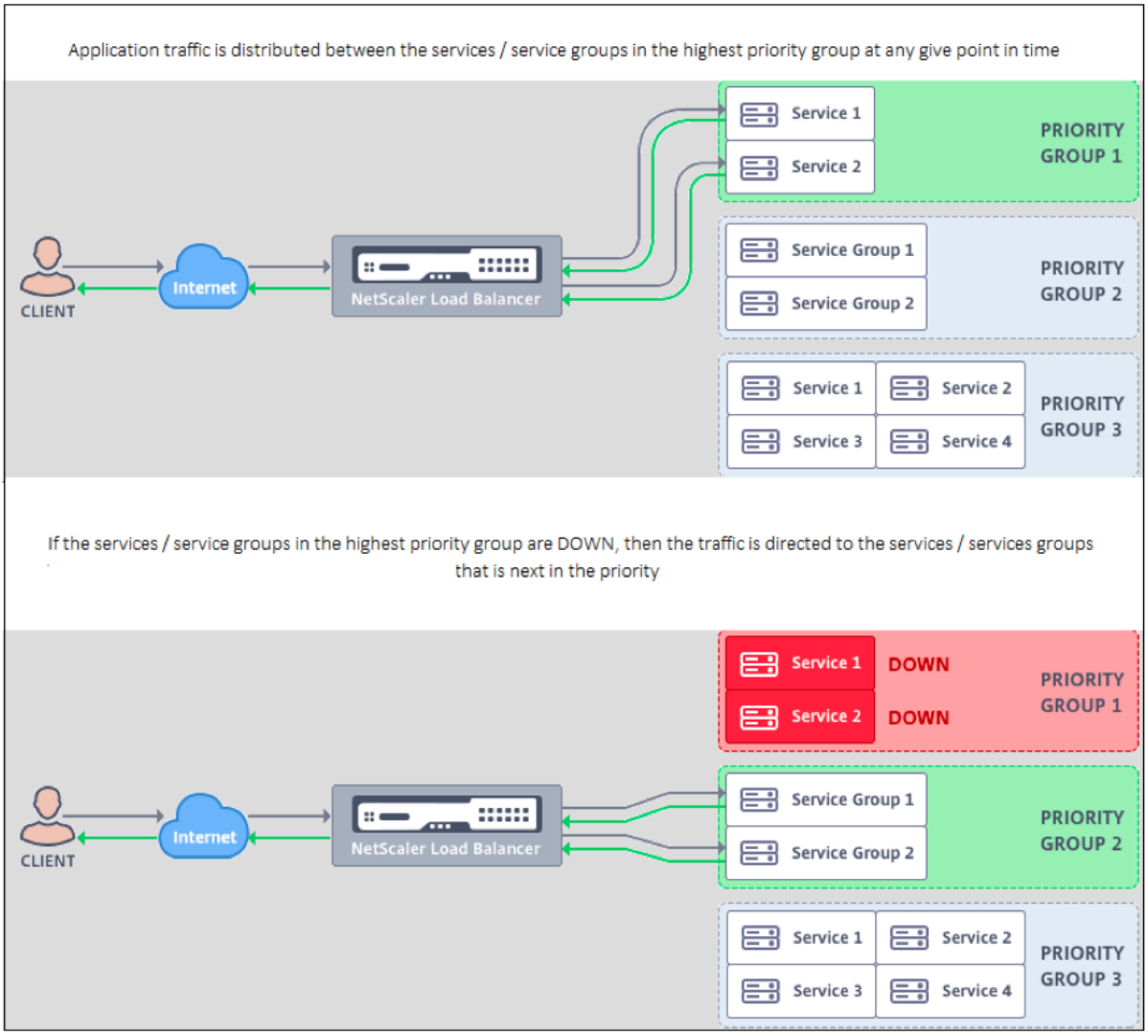
- SVG1：プライオリティ 1
- SVG2：プライオリティ 2
- SVG3：プライオリティ 3

このシナリオでは、アプリケーショントラフィックはサービスグループ SVG1 に送信されます。これは、このグループに最も低いプライオリティ番号が割り当てられているためです。SVG1 のすべてのメンバーが DOWN の場合、トラフィックはサービスグループ SVG2 に分散されます。このグループには、次に低いプライオリティ番号が割り当てられます。SVG2 のすべてのメンバーが DOWN の場合、トラフィックは SVG3 に分散されます。ただし、SVG1 のメンバーのいずれかが UP の場合、トラフィックは SVG1 にリダイレクトされます。これは、SVG1 には最小の数が割り当てられ、プライオリティが最も高いためです。

サービスまたはサービスグループに優先度を割り当てて、本番トラフィックへの影響を最小限に抑えるか、まったく影響を与えずに、必要に応じて、優先度が最も高い特定のサービスまたはサービスグループをアップグレードできます。

また、アップグレードが正常に行われなかった場合は、運用トラフィックへの影響を最小限に抑えたり、プライオリティの次のサービスまたはサービスグループに安全に切り替えることができます。

次の図は、プライオリティ負荷分散機能を示しています。



優先負荷分散を構成する

> 注
>
> Citrix ADC 優先負荷分散構成は、GUI 経由でのみサポートされます。CLI を使用してプライオリティ負荷分散を設定することはできません。

1. トラフィック管理に移動します > 優先負荷分散 > バーチャル *Servers 仮想サーバーのプロトコル、IP アドレス、および仮想サーバーのポート番号を指定します。
2. [最大優先度グループ] ボックスに、この仮想サーバーにバインドできる優先度サービスまたはサービスグループの数を入力します。デフォルト値は 2 で、設定可能な最大プライオリティは 10 です。このパラメータは、設定後は編集できません。

   注:

   優先度グループの最大数を指定して [**OK**] をクリックすると、コンテンツスイッチ仮想サーバーと「n」数のバ

ックアップ負荷分散仮想サーバーが作成されます。アルファベット「n」は、優先グループの最大数を表します。

たとえば、仮想サーバー名を vs1 として入力し、最大優先度グループを 5 に設定した場合、_Pri.LB##vs1##MaxPri=5 という名前のコンテンツスイッチング仮想サーバーと次の 5 つの負荷分散仮想サーバーが作成されます。

- _Pri.LB##vs1##MaxPri=5_LB1
- _Pri.LB##vs1##MaxPri=5_LB2
- _Pri.LB##vs1##MaxPri=5_LB3
- _Pri.LB##vs1##MaxPri=5_LB4
- _Pri.LB##vs1##MaxPri=5_LB5

3. 優先度グループの最大数を指定して [**OK**] をクリックすると、このコンテンツスイッチ仮想サーバーにバインドする必要があるサービスまたはサービスグループを選択するように求められます。

   - サービスを仮想サーバにバインドするには、［サービス］セクションの［挿入］をクリックします。次に、既存のサービスを選択するか、サービスを作成し、このサービスの優先度を設定します。また、このサービスをバインドする必要がある優先順位番号を設定します。
   - サービスグループを仮想サーバにバインドするには、[Service Groups] セクションの [**Insert**] をクリックします。次に、既存のサービスグループを選択するか、サービスグループを作成し、このサービスグループの優先順位を設定します。また、このサービスグループをバインドする必要がある優先順位番号を設定します。

   入力したプライオリティグループの最大数に応じて、ステップ 3 を繰り返します。

   注:

   - プライオリティが最も高いサービスまたはサービスグループは、プライオリティが最も高い負荷分散仮想サーバにバインドされます。

   たとえば、優先度 1 と 2 をそれぞれサービスグループSG_App1 and SG_App2に割り当てた場合、SG_App1はvirtual server _Priにバインドされます。LB##vs1##MaxPri=5_LB1 and SG_App2はvirtual server _Priにバインドされています。LB##vs1##MaxPri=5_LB2はステップ 2 で作成しました。

   - サービス・グループまたはサービスの優先度を変更するには、優先度ロード・バランシング仮想サーバー・ページの編集アイコンをクリックし、必要に応じて優先度を変更します。
   - すべての負荷分散仮想サーバーの構成が同じであるため、各仮想サーバーの負荷分散方法と永続性を明示的に設定することはできません。

4. [詳細設定] セクションで、要件に合ったその他の構成を完了します。

> 重要:
>
> 優先負荷分散の構成中に作成されたエンティティは、GUI の他のタブや CLI から変更しないでください。優先負荷分散エンティティは、[優先負荷分散] タブからのみ変更することをお勧めします。

Citrix ADC 拡張機能

October 7, 2021

Citrix ADC 拡張機能は、拡張コードを記述して Citrix ADC アプライアンスをカスタマイズするために使用できます。現在、ポリシー拡張とプロトコル拡張がサポートされています。ポリシー拡張は、ポリシー言語を拡張するために使用できます。プロトコル拡張を使用すると、Citrix ADC アプライアンスでカスタムプロトコルのサポートを追加できます。

Citrix ADC 拡張機能は、Citrix ADC CPX でもサポートされています。

このドキュメントでは、次の内容について説明します。

- Citrix ADC 拡張機能-言語の概要
- Citrix ADC 拡張機能-ライブラリリファレンス
- Citrix ADC 拡張機能 API リファレンス
- プロトコル拡張
- ポリシー拡張

Citrix ADC 拡張機能-言語の概要

October 7, 2021

拡張言語は、Lua 5.2 プログラミング言語に基づいています。Lua は、Citrix ADC ソフトウェアなどの C プログラムに埋め込むために設計された優れたパフォーマンスを備えたコンパクトな実行エンジンを提供します。

拡張言語は動的に型付けされます。つまり、各オブジェクトは独自の型情報を保持します。どの変数も実行中にいつでも任意の型を保持できるため、変数型は宣言されません。

言語は自由形式でもあり、トークン間の空白は無視されます。ステートメントはセミコロンで区切ることができますが、これは必須ではなく、通常は行われません。ステートメントのブロックは、通常、終了によって終了します。C や Java では、{と} のようなブロックの周りに括弧はありません。

識別子は、文字のシーケンス（a ～z および A ～Z）、数字（0 ～9）、およびアンダースコア（_）で、数字で始まりません。識別子は大文字と小文字が区別されるため、var、VAR、および Var はすべて異なる識別子です。

コメントは — によって開始されます。— 以降のすべては、行の最後まで無視されます。例:

```
-- This is a comment.
```

単純型

October 7, 2021

この言語では、次の単純な型の値を使用できます。

- 数字
- 文字列
- ブーリアン型
- Nil
- その他のタイプ

数字

すべての数値（整数）は、IEEE 754 浮動小数点値で表されます。2 ^ 54 までの整数は正確な表現を持っています。数値は次のように表すことができます。

- 符号付きおよび符号なしの 10 進整数 (例:10,-5)
- 小数点を含む実数 (10.5, 3.14159)
- 指数を持つ実数 (1.0e+10)
- 16 進数 (0xffff0000)

Citrix ADC ポリシー式には、次の 3 つの数値タイプがあります。

- 32 ビット整数 (num_at)
- 64 ビット整数 (unsigned_long_at)
- 64 ビット浮動小数点 (double_at)

これらのすべては、拡張関数に渡されたときに数値型に変換され、返されるときに数値は予想されるポリシー数値型に変換されます。

文字列

文字列は、任意の長さのバイトシーケンスです。これらは、ポリシー **text_at** タイプに対応します。文字列には、null（0x00）バイトを含めることができます。任意のバイナリデータは、任意の文字コード表現（例えば、UTF-8 や完全な Unicode など）を含む文字列で保持することができます。ただし、文字列関数 **likestring.upper ()** は 8 ビットの ASCII を想定しています。

文字列は、使用時に自動的に割り当てられます。文字列のバッファを明示的に割り当てる必要はありません（または方法）。文字列は、使用されなくなったときにガベージコレクションによって自動的に割り当て解除されます。明示的に文字列を解放する必要はありません（または方法でも）。この自動割り当てと割り当て解除は、メモリリークやダングリングポインタなど、C 言語の一般的な問題を回避します。

文字列リテラルは、二重引用符または一重引用符で囲まれた文字列です。2 つのタイプの引用符には違いはありません。「文字列リテラル」は「文字列リテラル」と同じです。通常のバックスラッシュエスケープを使用できます：s（ベル）、b（バックスペース）、f（フォームフィード）、n（改行/改行）、t（水平タブ）、（バックスラッシュ）、”（二重引用符）、’（一重引用符）。10 進数のバイト値は、バックスラッシュと 1 ～3 桁 (d, dd, ddd) で入力できます。16 進数のバイト値は、バックスラッシュ、x、および 2 桁の 16 進数 (xhh) で入力できます。

長い括弧表記という特殊な構文呼び出しは、長い複数行の文字列リテラルに使用できます。この表記法は、文字列を二重角括弧で囲み、括弧の間にゼロ以上の等号を付けます。文字列にない括弧と等号の組み合わせを考え出すことです。文字列ではエスケープシーケンスは考慮されません。例:

[[これは長いかっこの注釈を使用した不空数行の文字列です。]]

[= [これは、[と] とエスケープされていない長い記法を使用した複数行の文字列です]。] =]

長い括弧表記は、複数行のコメントを作成するために使用することができます。例:

—[[
これは複数行のコメントです。
–]]

ブーリアン型

通常の true および false のブール値が提供されます。ブール値は、ゼロが偽であると仮定され、ゼロ以外の値が真である C とは対照的に、数値とは異なることに注意してください。

Nil

nil は「値なし」を意味する特別な値です。NULL がゼロに定義されている C とは対照的に、独自の型であり、他の値と同等ではありません。

その他のタイプ

ユーザーデータとスレッドの 2 つのタイプがあります。これらは高度なトピックであり、ここでは説明しません。

変数

October 7, 2021

変数は、拡張の実行中に変更される可能性のある値を保持します。動的型付けのため、どの変数も任意の型の値を保持できます。変数の型宣言はありません。代わりに、変数の型は実行時に決定されます。実際、変数の値の型は実行中に変更される可能性がありますが、これは推奨されません。変数の初期値は nil です。

変数名は識別子であるため、文字、数字、およびアンダースコアが数字で始まらない文字列です。例: ヘッダー、結合ヘッダー。

グローバル変数

Lua では、そうでなければ宣言されていない変数は、プログラム内でグローバルです。ただし、ポリシー拡張関数ではグローバル変数を使用できません。これは、関数を実行できるパケットエンジンが複数あり、各パケットエンジンには独自のメモリがあるためです。

拡張モジュールでグローバル変数を使用すると、実行時エラーが発生します。**/var/log/ns.log** で報告されたグローバル変数を更新または作成しようとします。

タイプミスを持つ変数は、別のグローバル変数として解釈され、C や Java のような言語のように構文エラーを引き起こさないため、変数名のタイプミスは潜在的な問題です。上記のように、代わりにランタイムエラーが発生します。

局所変数

変数は、関数などのステートメントのブロックに対してローカルであると宣言できます。これは、ローカル変数名によって行われます。変数はブロックにスコープされます。つまり、ブロック内にのみ存在します。ローカル宣言は、オプションで変数に値を代入することができます。

例:

```
local headers = {}
local combined_headers = {}
```

式

October 7, 2021

式は、変数とリテラル値から値を計算します。

- 算術演算
- 関係演算
- 論理演算
- 連結
- 長さ
- 優先順位

算術演算

算術演算は、数値に対して実行されます。算術演算で文字列値が使用されると、数値に変換されます。これが失敗すると、エラーが返されます。

a + b	a と b を加算する
a-b	a から b を引く
a * b	a と b を乗算する
a/b	a を b で除算する
a% b	modulo = a - math.floor(a/b)*b
a^b	a を b の累乗します。b は任意の数になります。
-a	否定する

関係演算

関係演算は、2 つの値を比較し、関係が満たされている場合は true を返し、そうでない場合は false を返します。関係演算は、任意の型の値間で実行できます。値が同じ型でない場合は、false が返されます。数字は通常の方法で比較されます。文字列は、現在のロケールの照合順序を使用して比較されます。

a == b	a は b に等しい
a ~= b	a が b と等しくない
a < b	a が b より小さい
a > b	a が b より大きい
a <= b	a が b 以下である
a >= b	a が b 以上である

論理演算

論理演算は伝統的にブール値に対して実行されますが、この言語では、任意の 2 つの値に対して実行できます。nil と false は false とみなされ、他の値は true とみなされます。論理演算はショートカット評価を使用します。最初の値が操作の結果を決定した場合、2 番目の値は評価されません。

a and b	a が偽または nil の場合、他の戻り b を返します
a or b	a が偽でなく、nil でない場合は、他の戻り b を返します

not a	a が false または nil でない場合は false を返し、それ以外の場合は true を返します

and および or 演算は、式内の条件付き評価に使用できます。

a or b	a が初期化されていない（nil）場合、デフォルト値 b を提供するために使用することができます。これは、関数のオプションのパラメータに便利です。
a and b or c	条件 a に基づいて非 nil b または c を選択するために使用できます。a が真の場合、a と b は b を返し、b または c は a が偽の場合、a と b は偽を返し、または c は c を返します。これは、? b: C プログラミング言語の c。

連結

文字列の連結は s1.. s2 です。これにより、s1 と s2 の内容を保持するのに十分な大きさの新しい文字列が作成され、その内容が新しい文字列にコピーされます。s1 または s2 が文字列でない場合は、エラーが発生します。連結を繰り返すと、かなりのコピーオーバーヘッドが生じる可能性があることに注意してください。一度に 1 バイトを連結して n バイトの文字列を作成すると、n*（n+1）/2 バイトをコピーします。パフォーマンスを向上させるには、文字列の一部を連結してテーブル（後述）に配置し、table.concat（）関数を使用できます。この例は、COMBINE_HEADERS () の例に示されています。

長さ

文字列 s の長さは #s によって返されます。＃演算子は、後で説明するように、配列テーブルでも使用されます。

優先順位

演算子の優先順位は、式内で実行される演算の順序を決定します。優先順位の高い演算は、優先順位の低い演算よりも優先されます。優先順位の順序は、いつものように括弧で上書きすることができます。たとえば、`a + b \* c`では、* は + よりも優先順位が高いため、式は`a + (b \* c)`として評価されます。

最高	^
-	not # - (unary)
-	* / %
-	..
-	= ~= < > <= >=
-	および
最低	または

優先順位が同じ操作は左から右 (左結合) に実行されます。ただし、^ と.. は右から左 (右結合) に実行されます。したがって、^ b ^ c は a^（b ^ c）として評価されます。

割り当て

October 7, 2021

代入文は式を評価し、結果の値を変数に代入します。

```
variable = expression
```

前述のように、任意の型の値を任意の変数に割り当てることができるので、次のことが許可されます。

```
local v1 = ”a string literal”
v1 = 10
```

代入文は、実際にフォームを使用して、複数の変数を設定することができます

variable1, variable2, … = expression1, expression2, …

式よりも多くの変数がある場合、余分な変数には nil が割り当てられます。変数よりも多くの式がある場合、余分な式値は破棄されます。式はすべて代入の前に評価されるため、これを使用して 2 つの変数の値を簡潔に交換できます。

v1, v2 = v2, v1

は次と等価です。

tmp = v1
v2 = v1
v1 = tmp

テーブル

October 7, 2021

テーブルは、キーと値を持つエントリのコレクションです。これらは、提供される唯一の集約データ構造です。他のすべてのデータ構造（配列、リスト、セットなど）はテーブルから構築されます。テーブルのキーと値は、他のテーブルを含む任意の型にすることができます。同じテーブル内のキーと値は、型を混在させることができます。

- テーブルコンストラクタ
- テーブルの使用状況
- 配列としてのテーブル
- レコードとしてのテーブル

テーブルコンストラクタ

テーブルコンストラクタを使用すると、キーと関連する値を持つテーブルを指定できます。構文は次のとおりです。

{[key1] = value1, [key2] = value2, …}

キーと値は式です。キーが予約語でない文字列の場合、キーの前後の角かっこと引用符は省略できます。例:

{key1 = “value1”, key2 = “value2”, key3 = “value3”}

空のテーブルは単に {} で指定されます。

テーブルコンストラクタは、テーブルを参照する変数を設定するために代入で使用することができます。例:

local t1 = {} – set t1 to an empty table
local t2 = {key1 = “value1”, key2 = “value2”, key3 = “value3”}

テーブル自体は匿名であることに注意してください。複数の変数が同じテーブルを参照することがあります。上記の例を続けます。

ローカル t3 = t2-t2 と t3 の両方が同じテーブルを参照しています

テーブルの使用状況

期待どおりに、キーを使用してテーブル内の値を検索できます。構文は table[key] で、table はテーブル参照 (通常はテーブルに割り当てられた変数) で、key はキーを提供する式です。これが式で使用され、キーがテーブルに存在する場合、キーに関連付けられた値が返されます。キーがテーブルにない場合は、nil を返します。これが代入の変数として使用され、キーがテーブルに存在しない場合、キーと値の新しいエントリが作成されます。キーがすでにテーブルに存在する場合は、キーの値を新しい値で置き換えます。例:

local t = {} – sets t to an empty table
t[“k1”] = “v1” – creates an entry for key “k1” and value “v1”
v1 = t[“k1”] – sets v1 to the value for key “k1” = “v1”
t[“k1”] = “new_v1” – sets the value for key “k1” to “new_v1”

配列としてのテーブル

従来の配列は、インデックスとして整数キーを持つテーブルを使用して実装することができます。配列は、負のものを含む任意のインデックスを持つことができますが、慣例はインデックス 1 で配列を開始することです（C や Java のような言語の場合のように 0 ではありません）。そのような配列のための特別な目的のテーブルコンストラクタがあります：

{value1, value2, value3, … }

[配列参照は配列インデックスです]。

長さ演算子＃は、1 から始まる連続したインデックスを持つ配列の要素の数を返します。例:

local a = {“value1”, “value2”, “value3”}
local length = #a – sets length to the length of array a = 3

配列は、定義された要素のみが割り当てられているスパースにすることができます。しかし、＃は、非連続のインデックスを持つ疎配列では使用できません。例:

ローカル sparse_array = {} — 空の配列を設定する
sparse_array[1] =「値 1」— インデックス 1 に要素を追加する
sparse_array[99] =「value99」— インデックス 99 に要素を追加する

多次元配列は、テーブルのテーブルとして設定できます。たとえば、3x3 マトリックスは次のように設定できます。

local m = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9} }
local v22 = m[2][2] – sets v22 to 5

レコードとしてのテーブル

フィールドを持つレコードは、フィールド名キーを持つテーブルとして実装できます。参照フォーム table.field は、テーブル [「フィールド」] に使用できます。例:

local person = {name = “John Smith”, phone = “777-777-7777”}
local name = person.name – sets name to “John Smith”

テーブルの配列は、レコードのシーケンスに使用できます。例:

local people = {
{name = “John Smith”, phone = “777-777-7777”},
{name = “Jane Doe”, phone = “888-888-8888”}
…
}

名前 = 人 [2].name — 名前を「ジェーン・ドー」に設定します

制御構造

October 7, 2021

拡張関数言語は、プログラムの実行を制御するための通常のステートメントを提供します。

- If Then Else
- While Do and Repeat Until
- Numeric For
- Break
- Goto

If Then Else

If ステートメントは、1 つ以上の条件に基づいて実行するステートメントのブロックを選択します。次の 3 つの形式があります。

If then Form

```
if expression then
     statements to execute if expression is not false or nil
end
<!--NeedCopy-->
```

If then else Form

```
if expression then
     statements to execute if expression is not false or nil
else
     statements to execute if expression is false or nil
end
<!--NeedCopy-->
```

If then elseif else Form

```
if expression1 then
     statements to execute if expression1 is not false or nil
     elseif expression2 then
          statements to execute if expression2 is not false or nil
```



```
. . .
else
     statements to execute if all expressions are false or nil
end
<!--NeedCopy-->
```

例:

```
if headers[name] then

     local next_value_index = #(headers[name]) + 1
     headers[name][next_value_index] = value

else

     headers[name] = {
 name .. ":" .. value }


end
<!--NeedCopy-->
```

注:

- 式は、C および Java の場合と同様に、括弧で囲まれていません。
- C/Java スイッチ文に相当するものはありません。等価を行うには、一連の if elseif ステートメントを使用する必要があります。

While Do and Repeat Until

while 文と **repeat** 文は、式によって制御ループを提供します。

```
while expression do
     statements to execute while expression is not false or nil
end

repeat

     statements to execute until expression is not false or nil

until expression
<!--NeedCopy-->
```


while の例:

```
local a = {
 1, 2, 3, 4 }

local sum, i = 0, 1 -- multiple assignment initializing sum and i
while i <= #a do -- check if at the end of the array
     sum = sum + a[i] -- add array element with index i to sum
     i = i + 1 -- move to the next element
end
<!--NeedCopy-->
```

repeat の例**:**

```
sum, i = 0, 1 -- multiple assignment initializing sum and i
repeat
     sum = sum + a[i] -- add array element with index i to sum
     i = i + 1 -- move to the next element
until i > #a -- check if past the end of the array
<!--NeedCopy-->
```

もちろん、終了しないループを書くことは可能です。たとえば、これらの例のいずれかで i = i + 1 ステートメントを省略した場合などです。このような関数が実行されると、Citrix ADC は関数が妥当な時間内に完了しなかったことを検出し、実行時エラーで強制終了します。

```
Cpu limit reached. Terminating extension execution in [[string "function extension function..."]]: line line-number.
```

は、/var/log/ns.log に報告されます。

Numeric For

for ループには 2 種類あります。最初は、のための数値です。これは、C および Java での for ステートメントの通常の使用に似ています。数値の for 文は、変数が最終的な値を渡したかどうかをテストし、そうでない場合は、文のブロックを実行し、変数をインクリメントし、繰り返し、変数を初期化します。for ループの数値の構文は次のとおりです。

```
for variable = initial, final, increment do

     statements in the loop body

end
```



```
<!--NeedCopy-->
```

ここで、initial、final、および increment は、数値を生成する（または変換できる）すべての式です。変数は、for ループステートメントブロックに対してローカルであるとみなされ、ループの外側では使用できません。increment は省略できます。デフォルトは 1 です。式は、ループの先頭で 1 回評価されます。終了条件は、増分が正の場合は変数 > final で、増分が負の場合は変数 < final です。増分が 0 の場合、ループは直ちに終了します。

例 (前のセクションの while ループと繰り返しのループと同等):

```
sum = 0
for i = 1, #a do -- increment defaults to 1
     sum = sum + a[i]
end
<!--NeedCopy-->
```

for ループの 2 番目のタイプは、ループのより柔軟なタイプに使用することができ、のための汎用です。これは、関数の使用を伴うので、関数が導入された後に後で説明します。

Break

break 文は、しばらくの間、繰り返し、または for ループ内で使用されています。これは、ループを終了し、ループの後の最初の文で実行を再開します。例（前述の while、repeat、および for ループと同等）:

```
sum, i = 0, 1
while true do
     if i > #a then
          break
     end
     sum = sum + a[i]
     i = i + 1
end
<!--NeedCopy-->
```

Goto

goto 文は、ラベルに前方または後方にジャンプするために使用することができます。ラベルは識別子であり、その構文は:: label:: です。goto 文は goto ラベルです。例（前述のループと同等）:


```
sum, i = 0, 1
::start_loop::
     if i > #a then
          goto end_loop -- forward jump
     end
     sum = sum + a[i]
     i = i + 1
     goto start_loop -- backwards jump
::end_loop::
. . .
<!--NeedCopy-->
```

プログラミングで gotos を使用することについては、長い間論争がありました。一般に、関数をより読みやすく信頼できるものにするために、他の制御構造を使用してください。しかし、時折 gotos の賢明な使用は、より良いプログラムにつながる可能性があります。特に、gotos はエラーの処理に役立ちます。

関数

October 7, 2021

関数はプログラミングの基本的な構成要素であり、タスクを実行するステートメントをグループ化するための便利で強力な方法です。これらは、Citrix ADC アプライアンスと拡張コードの間のインターフェイスです。ポリシーの場合は、ポリシー拡張関数を定義します。プロトコルの場合は、プロトコルの動作のコールバック関数を実装します。関数は、関数との間で渡される値と関数に対して実行されるステートメントを指定する関数定義と、特定の入力データを持つ関数を実行し、関数から結果を取得する関数呼び出しで構成されます。

プロトコル動作コールバック関数

TCP クライアントの動作は、TCP クライアントデータストリームイベントを処理するコールバック関数 (on_data) で構成されます。TCP ベースのプロトコルに Message Based Load Balancing (MBLB) を実装するには、このコールバック関数のコードを追加して、クライアントからの TCP データストリームを処理し、バイトストリームをプロトコルメッセージに解析します。

ビヘイビア内のコールバック関数は、処理モジュールの状態であるコンテキストで呼び出されます。コンテキストは、処理モジュールのインスタンスです。たとえば、TCP クライアント動作コールバックは、クライアント TCP 接続ごとに異なるコンテキストで呼び出されます。

コンテキストに加えて、ビヘイビアコールバックは他の引数を持つことができます。通常、残りの引数はペイロードとして渡されます。ペイロードは、すべての引数のコレクションです。したがって、プログラマブル処理モジュールのインスタンスは、インスタンス状態とイベントコールバック関数、つまりコンテキストと動作の組み合わせとして見ることができます。トラフィックはイベントペイロードとしてパイプラインを通過します。

TCP クライアントコールバック関数のプロトタイプ:

```
1             Function                client on_data (ctxt, payload)
2
3                                             //.code
4
5             end
```

各項目の意味は次のとおりです。

- ctxt -TCP クライアント処理コンテキスト
- payload： イベントペイロード
  - payload.d ata-受信した TCP データ。バイトストリームとして利用できます。

ポリシー拡張関数

NetScaler ポリシー表現言語は厳密に型付けされているため、拡張関数の定義では、入力の種類と戻り値を指定する必要があります。Lua 関数定義は、次の型を含むように拡張されました。

```
1 function self-type:function-name(parameter1: parameter1-type, etc.):
     return-type
2      statements
3 end
4
5 where,
6
7 the types are NSTEXT, NSNUM, NSBOOL, or NSDOUBLE.
8 <!--NeedCopy-->
```

self 型は、関数に渡される暗黙の self パラメータの型です。Citrix ADC ポリシー式で拡張関数を使用する場合、これは関数の左側にある式によって生成された値です。これを表示するもう 1 つの方法は、この関数によって Citrix ADC ポリシー言語でそのタイプが拡張されることです。

パラメータ型は、ポリシー式で拡張関数呼び出しで指定された各パラメータの型です。拡張関数は、0 個以上のパラメータを持つことができます。

return-type は、拡張関数呼び出しによって返される値の型です。ポリシー式の部分（ある場合）への入力であり、関数の右側に入力されます。それ以外の場合は、式の結果の値です。

例**:**

```
function NSTEXT:COMBINE_HEADERS(): NSTEXT
```


ポリシー表現での拡張関数の使用:

```
HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1rn").COMBINE_HEADERS()
```

ここで自己パラメータは、テキスト値である HTTP.REQ.FULL_HEADER.AFTER_STR（「HTTP/1.1rn」）の結果です。COMBINE_HEADERS () 呼び出しの結果はテキストであり、この呼び出しの右側には何もないため、式全体の結果はテキストになります。

ローカル関数定義

拡張関数のほかに、拡張ファイルにグローバル関数を定義することはできません。しかし、ローカル関数は、通常の Lua の関数文を使用して拡張関数内で定義することができます。これは、関数の名前とそのパラメータの名前（引数とも呼ばれます）を宣言し、Lua のすべての宣言と同様に、任意の型を指定しません。この構文は次のとおりです。

```
local function function-name(parameter1-name, parameter2-name, etc.)
     statements
end
<!--NeedCopy-->
```

関数名とパラメータ名はすべて識別子です。(関数名は実際には変数であり、関数文はローカル関数名 = 関数（パラメータ 1 など）の略語ですが、関数を使用するためにこの微妙さを理解する必要はありません)。

等は、通常の... の代わりにパラメータ名のパターンを継続するためにここで使用されることに注意してください。これは... それ自体が実際には変数パラメータリストを意味するためです。これはここでは説明しません。

関数本体と戻り値

関数と end 文の間の文のブロックは、関数本体です。関数本体では、関数パラメータはローカル変数のように動作し、前述のように、関数呼び出しによって指定された値を持ちます。

return 文は、関数の呼び出し元に返される値を提供します。それはブロックの最後に現れなければなりません（関数内で、もしそうなら、for ループなど; それはそれ自身のブロックにあることができます戻り値... 終了)。これは、なし、1、または複数の戻り値を指定することができます。

```
return -- returns nil
return expression -- one return value
return expression1, expression2, ... -- multiple return values
<!--NeedCopy-->
```

例:


```
local function fsum(a)
     local sum = 0
     for i = 1, #a do
          sum = sum + a[i]
     end
     return sum
end

local function fsum_and_average(a)
     local sum = 0
     for i = 1, #a do
          sum = sum + a[i]
     end
     return sum, sum/#a
end
<!--NeedCopy-->
```

関数呼び出し

関数呼び出しは、関数の本体を実行し、そのパラメータの値を指定し、結果を受け取ります。関数呼び出しの構文は、関数名（式 1、式 2 など）で、関数パラメータは対応する式に設定されます。式とパラメータの数は同じである必要はありません。式の数がパラメータよりも少ない場合、残りのパラメータは nil に設定されます。したがって、呼び出しの最後に 1 つ以上のパラメータをオプションにすることができ、関数が nil でないかどうかをチェックすることによって指定されているかどうかを確認できます。これを行う一般的な方法は、または操作です。

```
function f(p1, p2) -- p2 is optional
     p2 = p2 or 0 -- if p2 is nil, set to a default of 0
     . . .
end
<!--NeedCopy-->
```

パラメータよりも多くのエクスプレッションがある場合、残りのエクスプレッション値は無視されます。

前述のように、関数は複数の値を返すことができます。これらのリターンは、複数の代入ステートメントで使用できます。例:

```
local my_array = {
 1, 2, 3, 4 }

local my_sum, my_ave = sum_and_average(my_array)
```



```
<!--NeedCopy-->
```

反復子関数と汎用 **for** ループ

関数を導入したので、一般的な for ループについて話すことができます。一般的な for ループ（1 つの変数を持つ）の構文は次のとおりです。

```
for variable in iterator(parameter1, parameter2, etc.) do
     statements in the for loop body
end
<!--NeedCopy-->
```

ここで、iterator () は、ループ本体の各反復で変数の値を提供するゼロ以上のパラメータを持つ関数です。イテレータ関数は、クロージャと呼ばれるテクニックを使用して、イテレーションのどこにあるかを追跡します。これはここで心配する必要はありません。これは、nil を返すことによって、反復の終了を知らせます。イテレータ関数は、複数の代入で使用するために、複数の値を返すことができます。

イテレータ関数の記述は、本書の範囲を超えていますが、概念を説明する便利な組み込みイテレータがいくつかあります。1 つは、テーブル内のエントリを反復処理し、2 つの値、キーと次のエントリの値を返すペア () イテレータです。

例**:**

```
local t = {
 k1 = ”v1”, k2 = ”v2”, k3 = ”v3” }

local a = {
  }
  -- array to accumulate key-value pairs
local n = 0 -- number of key-value pairs
for key, value in pairs(t) do
     n = n + 1
     a[n] = key .. ” = ” .. value -- add key-value pair to the array
end
local s = table.concat(a, ”; ”) -- concatenate all key-value pairs into
    one string
<!--NeedCopy-->
```

もう 1 つの便利なイテレータは、次の COMBINE_HEADERS () の例で使用される string.gmatch () 関数です。

Citrix ADC 拡張機能-ライブラリ参照

October 7, 2021

ポリシー拡張でサポートされるライブラリのリスト。

- 基本ライブラリ
- String ライブラリ
- 正規表現パターン-文字クラス
- 正規表現パターン-パターンアイテム
- Table ライブラリ
- Math ライブラリ
- ビット演算ライブラリ
- オペレーティングシステムライブラリ
- Citrix ADC ライブラリ

基本ライブラリ

assert(v[,message])	v が false の場合、オプションのメッセージとともにエラーを発行します。
error(message)	関数を終了し、エラーメッセージを報告します。
ipairs(a)	配列 a のためのイテレータ。各反復のインデックスと値を返します。
pairs(t)	テーブル t のイテレータ。各反復のキーと値を返します。
tonumber(e[,base])	e を数値に変換し、オプションの基数を指定します。
tostring(v)	v を文字列に変換します
type(v)	数値、文字列、ブール値、テーブルなど、v のタイプを返します。
getmetatable (object)	オブジェクトにメタータブルがない場合は nil を返します。それ以外の場合、オブジェクトのメタレコードに「__metatable」フィールドがある場合は、関連付けられた値を返します。それ以外の場合は、指定されたオブジェクトのメタレコードを返します。

setmetatable (table, metatable)	指定されたテーブルのメタテーブルを設定します。（他のタイプのメタータブルを Lua から変更することはできません。C からのみ変更してください）メタテーブルが nil の場合、指定されたテーブルのメタテーブルを削除します。元のメタレコードに「__metatable」フィールドがある場合、エラーが発生します。
select (index, ···)	引数番号インデックスの後のすべての引数を返します。インデックスが文字列「#」の場合は、それが受け取った余分な引数の合計数を返します。
pcall (f [, arg1, ···])	プロテクトモードで指定された引数で関数 f を呼び出します。これは、呼び出しが成功したかどうかを示す最初の結果としてステータスコードを返します。呼び出しが成功した場合、ステータスコードとともに、それはまた、それ以外の場合はエラーメッセージを返し、呼び出しからすべての結果を返します。
xpcall (f, msgh [, arg1, ···])	この関数は pcall に似ていますが、エラー処理の引数も取ります。
_VERSION	現在のインタプリタのバージョンを返します。

String ライブラリ

string.byte(s[,i[,j]])	s[i] から S[j] のバイト値を返します。デフォルト i = 1 および j = i
string.char (…)	整数パラメータで構成された文字列を返します。
string.find(s,pattern[,init[,plain])	s 内の正規表現パターンが最初に一致するものを検索します。マッチまたは nil の最初と最後のインデックスを返します。init は開始するインデックス、デフォルト 1 プレーン = true はパターンが正規表現でないことを意味します。
string.format(form,…)	パラメータのフォーマットされたバージョンを返します。
string.gmatch(s,pattern)	正規表現パターンで「s」を検索するための反復子。一致する値を返します。

string.gsub(s,pattern,repl[,n])	パターンのすべての（または n）出現が repl に置き換えられた s のコピーを返します。
string.len(s)	文字列の長さを返します。
string.lower(s)	小文字に変換された文字列のコピーを返します。
string.match(s,pattern[,init])	s の正規表現パターンの最初の一致を検索し、キャプチャまたはパターン全体を返します。init は開始するインデックス、デフォルトは 1 です。
string.rep(s,n[,sep])	セパレータ sep、デフォルトなしセパレータで、s の n 個のコピーである文字列を返します
string.reverse(s)	逆にされた文字列を返します。
string.sub (s, i[, j])	s から s[i] への s の [j] 部分文字列を返します。デフォルト j は文字列の終わりです。
string.upper(s)	大文字に変換された文字列のコピーを返します。
string.dump (function)	指定された関数のバイナリ表現を含む文字列を返します。

正規表現パターン-文字クラス

x	文字 x (マジック文字 ^$ ()% を除く。[]*+-?)
。	任意の文字
A	任意の文字
%c	任意の制御文字
%d	任意の数字
%g	スペース以外の印刷可能な文字
%l	任意の小文字
%p	任意の句読点
%s	任意の空白文字
%u	任意の大文字
%w	任意の英数字
%x	エスケープされたマジック文字 x (例:%%)

[set]	文字のセット：個々の文字のシーケンス、範囲 x-y、および%クラス
[^set]	文字セットに含まれていません。

正規表現パターン-パターンアイテム

⍰	文字クラス
X*	X の 0 文字以上の最長繰り返し
X+	X の文字を 1 回以上繰り返す
X-	X で 0 文字以上の最短繰り返し
X?	X の 0 文字または 1 文字
%n	n=1〜9; n 番目のキャプチャされた文字列にマッチします
%bxy	は、バランスの取れた 2 つの文字 x と y の間の部分文字列に一致します。例%b () は、2 つのカッコ間で部分文字列に一致します。
%f[set]	は、次の文字が set に属し、前の文字が set に属さないような任意の位置で空の文字列に一致します。

パターンは、パターンアイテムのシーケンスです。^pattern は文字列の先頭に一致し、pattern $ は文字列の末尾に一致します。

マッチした部分文字列は（パターン）を使用してキャプチャすることができます。pattern () のない括弧 () は、現在の文字列位置 (数値) を取得します。

Table ライブラリ

table.concat(list[,sep[,i[,j]]])	文字列リストを返します [i]。.. sep.. list[i+1].. sep.[j]. デフォルト sep は空の文字列です。既定値 i は 1、j は #list です。

table.insert(list,[pos,]value)	インデックス pos のリストに値を挿入します。pos のデフォルトは #list (リストの最後) です。
table.pack (…)	インデックス 1 から始まるパラメータと、パラメータの総数を示すキー n を含む配列を返します。
table.remove(list[,pos])	位置を埋めるために要素をシフトし、位置 pos で要素をリストから削除します。削除された要素を返します。posis #list のデフォルト (リストの最後)
table.sort(list[,comp])	リストの要素をソートします。comp は、使用する比較関数です。comp のデフォルトは < です。
table.unpack(list[,i[,j]])	list[i]～list[j] を返します。i の既定値は 1、j は #list です <c/ode>。

Math ライブラリ

さまざまな三角関数と対数関数は表示されません。

math.abs(x)	x の絶対値を返します。
math.ceil(x)	最小の整数 >= x を返します。
math.floor(x)	最大整数 <= x を返します。
math.fmod(x,y)	0 に向かって商を丸める x/y の余りを返します。
math.huge	値 >= その他の数値。
math.max(x,…)	最大引数を返します。
math.min(x,…)	最小引数を返します。
math.modf(x)	x の整数部と小数部を返します。
math.random()	0 と 1 の間の擬似乱数を返します。
math.random(m)	1 と m の間の擬似ランダム整数を返します。
math.random(m, n)	m と n の間の擬似乱数整数を返します。
math.randomseed(x)	擬似乱数ジェネレータを x に設定します。
math.sqrt(x)	x (x ^ 0.5) の平方根を返します。
math.acos(x)	x のアークコサインを返します（ラジアン単位）。
math.asin(x)	x のアークサインを返します（ラジアン単位）。

math.atan(x)	x のアークタンジェントを返します（ラジアン単位）。
math.atan2(y, x)	y/x のアークタンジェントを返します (ラジアン単位)。
math.cos(x)	x の余弦を返します。
math.cosh(x)	x の双曲線余弦を返します。
math.sin(x)	x の正弦を返します。
math.sinh(x)	x の双曲線正弦を返します。
math.tan(x)	x のタンジェントを返します。
math.tanh(x)	x の双曲線正接を返します。
math.deg(x)	角度 x (ラジアンで指定) を度単位で返します。
math.exp(x)	値 e^x を返します。
math.frexp (x)	x = m2e、e が整数であり、m の絶対値が範囲 [0.5、1）にあるように m と e を返します。
math.ldexp (m, e)	m2e (e は整数でなければなりません) を返します。
math.log (x [, base])	指定された底の x の対数を返します。base のデフォルトは e です。
math.pow (x, y)	x^y を返します。
math.rad (x)	角度 x (度単位) をラジアンで返します。
math.pi	π の値。

ビット演算ライブラリ

特に明記されていない限り：

- すべての関数は、範囲（-2^51、+2^51）の数値引数を受け入れます。
- 各引数は、その除算の余りに 2^32 で正規化され、整数に切り捨てられ、最終的な値が [0,2^32-1] の範囲になります。
- すべての結果が範囲 [0,2^32-1] 内にある。

bit32.arshift(x,disp)	数値 x を右 (+disp) または左 (-disp) に算術的にシフトした disp ビットを返します。
bit32.band (…)	引数のビット単位 AND を返します。

bit32.bnot(x)	x のビット単位 NOT を返します。
bit32.bor (…)	引数のビット単位 OR を返します。
bit32.btest(…)	引数のビット単位 AND がゼロでない場合は true を返します。
bit32.bxor (…)	引数のビット単位排他的論理和を返します。
bit32.extract(n,field[,width])	n の field から field + width - 1 までのビットを返します（最上位から最下位までのビット数）。width のデフォルトは 1 です。
bit32.replace(n,v,field[,width])	field から field + width -1 へのビットを v で置き換えられた n のコピーを返します。デフォルトの width は 1 です。
bit32.lrotate(x,disp)	x を disp ビット左（+disp）または右（-disp）に回転した数値を返します。
bit32.lshift(x,disp)	x を disp ビット左（+disp）または右（-disp）にシフトした数値を返します。
bit32.rrotate(x,disp)	x を disp ビット右（+disp）または左（-disp）に回転した数値を返します。
bit32.rshift(x,disp)	x を disp ビット右（+disp）または左（-disp）にシフトした数値を返します。

オペレーティングシステムライブラリ

os.clock ()	CPU 時間の秒単位の近似値を返します。
os.date ([format [, time]])	指定された文字列フォーマットに従ってフォーマットされた文字列または日付と時刻を含むテーブルを返します。
os.time ([table])	引数なしで呼び出されたときの現在の時刻、または指定されたテーブルで指定された日付と時刻を表す時刻を返します。
os.difftime (t2, t1)	時間 t1 から時間 t2 までの秒数を返します。

Citrix ADC ライブラリ

ns.logger:level(message)	レベルが緊急、アラート、クリティカル、エラー、警告、通知、情報、デバッグであるメッセージをログに記録します。パラメータは、C printf () 関数と同じです。書式文字列と、書式文字列の% 指定子に値を指定するための可変数の引数です。

Citrix ADC 拡張モジュールの **API** リファレンス

October 7, 2021

動作は、Citrix ADC アプライアンスで使用できる一般的なプログラマブルパターンの形式化です。たとえば、TCP 仮想サーバーは TCP クライアントの動作と TCP サーバーの動作をサポートします。ビヘイビアは、あらかじめ定義されたコールバック関数のセットです。コールバック関数を提供することで、ビヘイビアを実装できます。たとえば、TCP クライアントの動作は、TCP データストリームを処理する on_data 関数で構成できます。

TCP クライアントの動作

on_data -TCP クライアントデータイベントの関数コールバック。コールバックは、次の 2 つの引数を取ります。

- **ctxt** -TCP クライアント処理コンテキスト
- **payload**： イベントペイロード
  - **payload.d** ata-受信した TCP データ。バイトストリームとして利用できます。

TCP サーバの動作

on_data -TCP サーバデータイベントの関数コールバックの場合、コールバックは 2 つの引数を取ります。

- **ctxt** -TCP サーバの処理コンテキスト
- **payload**： イベントペイロード
  - **payload.data** -受信した tcp データ。バイトストリームとして利用可能

TCP クライアントコンテキスト

TCP クライアントイベントコールバックに渡されるコンテキスト。

- **ctxt.output** -パイプライン内の次の処理コンテキスト。拡張コールバックハンドラは、プロトコルメッセージの終わりを意味する部分的なメッセージまたは EOM を意味するイベント DATA を使用して、ctxt.output

に ns.tcp.stream タイプのデータを送信することができます。EOM イベントには、TCP データが含まれている場合とそうでない場合があります。TCP データを持つ EOM イベントは、先行する DATA イベントなしで送信して、プロトコルメッセージデータ全体を送信し、メッセージの終わりをマークできます。ロードバランシングの決定は、最初に受信したデータに基づいて、ロードバランシング仮想サーバによって下流に行われます。EOM メッセージの受信後、新しいロードバランシングの決定が行われます。したがって、プロトコルメッセージデータをストリーミングするには、最後のイベントを EOM として複数の DATA イベントを送信します。すべての連続する DATA イベントと次の EOM イベントは、シーケンスの最初の DATA イベントのロードバランシングの決定によって選択された同じサーバー接続に送信されます。

- **ctxt.input** -TCP ストリームデータが送信されるパイプライン内の以前の処理コンテキスト。
- **ctxt**：**Hold**（データ） -将来の処理のためにデータを格納する関数。データで保留を呼び出すと、データはコンテキストに格納されます。後で、同じコンテキストでより多くのデータが受信されると、新しく受信したデータが以前に格納されたデータに追加され、結合されたデータストリームが on_data コールバック関数に渡されます。保留を呼び出した後、データ参照は使用できなくなり、使用時にエラーが発生します。
- **ctxt.vserver** -仮想サーバーのコンテキスト。
- **ctxt.client**： クライアント接続処理コンテキスト。この処理コンテキストは、クライアントにデータを送信し、IP アドレス、送信元および宛先ポートなどの接続関連情報を取得するために使用することができます。
- **ctxt: close ()** — クライアントに FIN を送信して、クライアント接続を閉じます。この API を呼び出すと、クライアント処理コンテキストは使用できなくなり、使用時にエラーが発生します。

TCP サーバコンテキスト

TCP サーバイベントコールバックに渡されるコンテキスト。

- **ctxt.output** — パイプライン内の次の処理コンテキスト。拡張コールバックハンドラは、プロトコルメッセージの終わりを意味する部分的なメッセージまたは EOM を意味するイベント DATA を使用して、ctxt.output に ns.tcp.stream タイプのデータを送信することができます。
- **ctxt.input** -TCP ストリームデータが送信されるパイプライン内の以前の処理コンテキスト。
- **ctxt**：**Hold**（データ） -将来の処理のためにデータを格納する関数。データで保留を呼び出すと、データはコンテキストに格納されます。後で、同じコンテキストでより多くのデータが受信されると、新しく受信したデータが以前に格納されたデータに追加され、結合されたデータストリームが on_data コールバック関数に渡されます。保留を呼び出した後、データ参照は使用できなくなり、使用時にエラーが発生します。
- **ctxt.vserver** -仮想サーバーのコンテキスト。
- **ctxt.server** -サーバー接続処理コンテキスト。この処理コンテキストは、サーバーにデータを送信し、IP アドレス、送信元および宛先ポートなどの接続関連情報を取得するために使用することができます。
- **ctxt**：**reuse_server_connection**（） -この API は、サーバー接続がサーバーコンテキスト内の他のクライアント接続に対してのみ再利用できるようにするために使用されます。この API は、クライアントコンテキ

ストでデータを送信するために (ns.send () API で) EOM イベントが使用されている場合にのみ使用できます。それ以外の場合、ADC アプライアンスはエラーをスローします。

サーバー接続を他のクライアントで再利用できるようにするには、各応答メッセージの最後にこの API を呼び出す必要があります。この API を呼び出した後、このサーバー接続でより多くのデータが受信された場合、これはエラーとして扱われ、サーバー接続は閉じられます。この API を使用しない場合、サーバー接続は開かれたクライアントに対してのみ使用できます。また、そのクライアントの別の負荷分散決定に同じサーバーが選択されている場合は、同じサーバー接続を使用してクライアントデータを送信します。この API を使用すると、サーバー接続が開かれたクライアント接続に関連付けられなくなり、他のクライアント接続に対する新しい負荷分散決定に再利用できます。この API を呼び出すと、サーバーコンテキストは使用できなくなり、使用時にエラーがスローされます。

注：この API は、Citrix ADC 12.1 ビルド 49.xx 以降で使用できます。

- **ctxt: close ()** — サーバーに FIN を送信して、サーバー接続を閉じます。この API を呼び出すと、クライアント処理コンテキストは使用できなくなり、使用時にエラーが表示されます。

  注：この API は、Citrix ADC 12.1 ビルド 50.xx 以降で使用できます。

仮想サーバーコンテキスト

コールバックに渡されるコンテキストを通じて利用可能なユーザー仮想サーバーコンテキスト:

- **vserver: counter_increment (**カウンタ **_name)** -引数として渡された仮想サーバカウンタの値をインクリメントします。現在、次の組み込みカウンタがサポートされています。
  - **-invalid_messages** — この仮想サーバー上の無効な要求/応答の数。
  - **-invalid_messages_drop** — この仮想サーバによって削除された無効な要求/応答の数。
- **vserver.params** -ユーザー仮想サーバー用に構成されたパラメータ。パラメーターは、拡張機能の設定機能を提供します。拡張コードは、CLI で指定されたパラメータにアクセスして、ユーザ仮想サーバを追加できます。

クライアント接続コンテキスト

接続関連の情報を取得するためのクライアント接続処理コンテキスト。

- **client.ssl** – SSL コンテキスト
- **client.tcp** – TCP コンテキスト
- **client.is_ssl** — クライアント接続が SSL ベースの場合は True

サーバ接続コンテキスト

接続関連の情報を取得するためのサーバー接続処理コンテキスト。

- **server.ssl** – SSL コンテキスト

- **server.tcp** – TCP コンテキスト
- **server.is_ssl** — サーバー接続が SSL ベースの場合は True

TCP コンテキスト

TCP コンテキストは TCP プロトコル上で動作します。

- **tcp.srcport** — 数値の送信元ポート
- **tcp.dstport** -番号としての宛先ポート

IP コンテキスト

IP コンテキストは、IP または IPv6 プロトコルデータで動作します。

- **ip.src**： 送信元 IP アドレスのコンテキスト。
- **ip.dst** -宛先 IP アドレスのコンテキスト。

注：この API は、Citrix ADC 12.1 ビルド 51.xx 以降で使用できます。

IP アドレスコンテキスト

IP アドレスコンテキストは、IP アドレスまたは IPv6 アドレスデータに対して機能します。

- **<address>.to_s**-適切な ASCII 表記でのアドレス文字列。
- **<address>.to_n**-ネットワーク順のバイト文字列としてのアドレスの数値 (IPv4 の場合は 4 バイト、IPv6 の場合は 16 バイト)。
- **<address>.version**-IPv4 の場合は 4 を返し、IPv6 の場合は 6 を返します。
- **<address>:subnet(<prefix value>)**-プレフィックス番号を適用した後、サブネットアドレス文字列を返します。
  - IPv4 アドレスの場合、値は 0 から 32 の間でなければなりません
  - IPv6 アドレスの場合、値は 0 から 128 の間である必要があります。
- **<address>:apply_mask(<mask string>)**-マスク文字列を適用した後のアドレス文字列を返します。API は、引数のバージョンを検証し、適切なエラーチェックを行います。
- **address>:eq(<address string>)**-引数がアドレスオブジェクトと同等であるかどうかに基づいて true または false を返します。API は引数のバージョンを検証します。

注：この API は、Citrix ADC 12.1 ビルド 51.xx 以降で使用できます。

SSL コンテキスト

SSL コンテキストは、フロントエンド SSL 接続に関連する情報を提供します。

- **ssl.cert** — SSL 証明書コンテキスト。クライアント接続の場合は、クライアント証明書コンテキストを提供し、サーバー接続の場合は、サーバー証明書コンテキストを提供します。

- **ssl.version** -現在のトランザクションの SSL プロトコルのバージョンを表す数値。次のようになります。

  - - 0: The transaction is not SSL-based
  - - 0x002: The transaction is SSLv2
  - - 0x300: The transaction is SSLv3
  - - 0x301: The transaction is TLSv1
  - - 0x302: The transaction is TLSv1.1
  - - 0x303: The transaction is TLSv1.2

- **ssl.cipher_name** -SSL 接続から呼び出された場合、SSL 暗号名を文字列として返します。それ以外の場合は、NULL 文字列を返します。
- **ssl.cipher_bits** — 暗号化キーのビット数。

SSL 証明書コンテキスト

- **Cert.version**： 証明書のバージョン番号。接続が SSL ベースでない場合は、0 を返します。
- **Cert.valid_not_before** — 文字列形式の日付で、証明書が無効になります。
- **Cert.valid_not_after** — 文字列形式の日付。この日付を超えると、証明書は無効になります。
- **Cert.days_to_expire** — 証明書が有効になるまでの日数。期限切れの証明書の場合は-1 を返します。
- **Cert.to_pem** — バイナリ形式の証明書。
- **cert.issuer** -証明書内の発行者の識別名 (DN) を名前/値リストとして指定します。等号 (「=」) は名前と値の区切り文字で、スラッシュ (「/」) は名前と値のペアを区切る区切り文字です。

  返される DN の例を次に示します。
  /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@n

- **cert.auth_keyid** — X.509 V3 証明書の認証機関キー識別子拡張のコンテキスト。
  - **auth_keyid.exists** -証明書に認証局キー識別子の拡張子が含まれている場合は TRUE。
  - **auth_keyid.issuer_name** -証明書内の発行者の識別名 (名前/値リスト)。
    等号 (「=」) は名前と値の区切り文字で、スラッシュ (「/」) は名前と値のペアを区切る区切り文字です。

  以下に例を示します。
  /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@n
  - **auth_keyid.keyid** -ブロブとしての権限キー識別子のキー識別子フィールド
  - **auth_keyid.cert_serialnumber** -BLOB としての権限キー識別子のシリアル番号フィールド。
- **cert.pk_algorithm** -証明書で使用される公開鍵アルゴリズムの名前。
- **cert.pk_size** -証明書で使用される公開キーのサイズ。
- **cert.serialnumber** -クライアント証明書のシリアル番号。これが SSL 以外のトランザクションである場合、または証明書にエラーがある場合、空の文字列が返されます。

- **cert.signature_algorithm** -この証明書に署名するために CA が使用する暗号化アルゴリズムの名前。
- **cert.subject_keyid** -クライアント証明書のサブジェクトキー ID。Subject KeyID がない場合は、長さ 0 のテキストオブジェクトが返されます。
- **cert.subject** -名前-値としてのサブジェクトの識別名。等号（「=」）は名前と値を区切り、スラッシュ（「/」）は名前と値のペアを区切ります。

以下に例を示します。
/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycom

Citrix ADC ライブラリ

- **ns.tcp.stream** -TCP データをバイトストリームとして処理するためのライブラリのような文字列。これらの API が動作できる TCP ストリームデータの最大サイズは 128 KB です。ns.tcp.stream ライブラリ関数は、通常の拡張オブジェクト指向の呼び出しスタイルでも呼び出すことができます。たとえば、データ:len () は、ns.tcp.stream.len (データ) と同じです。
  - **ns.tcp.stream.len(data)** -Lua の string.len に似たバイト単位のデータの長さを返します
  - **ns.tcp.stream.find(data, pattern [, init])**-Lua の文字列.match に似た機能。また、データの最後に部分一致も行います。部分一致すると、開始インデックスが返され、終了インデックスが nil になります。
  - **ns.tcp.stream.split(data, length)** -データを 2 つのチャンクに分割します。最初のチャンクは指定された長さです。分割が成功すると、元のデータは TCP データストリームとして使用できなくなります。そのように使用しようとすると、エラーが発生します。
  - **ns.tcp.stream.byte(data[, i [, j]])** -Lua の string.byte に似た機能。文字データ [i]、データ [i+1]、...、データ [j] の内部数値コードを返します。
  - n**s.tcp.stream.sub(data, i [, j])** -Lua の string.sub に似た機能。i から始まり、j まで続く s の部分文字列を返します。
  - **ns.tcp.stream.match(data, pattern, [, init])**-Lua の string.match に似た機能。文字列 s のパターンが最初に 一致しているかどうかを調べます。
- **ns.send(processing_ctxt, event_name, event_data)** -処理コンテキストにイベントを送信するための汎用関数。イベントデータは、任意のコンテンツを持つことができる Lua テーブルです。内容はイベントによって異なります。ns.send () API が呼び出されると、データ参照は使用できなくなります。それを使用しようとすると、エラーが発生します。
- **ns.pipe**（**src_ctxt**、**dest_ctxt**） -pipe() API の呼び出しを使用して、拡張コードはソースコンテキストを宛先コンテキストに接続できます。パイプへの呼び出しの後、ソースコンテキストからパイプライン内の次のモジュールに送信されるすべてのイベントは、宛先コンテキストに直接移動します。この API は、通常、pipe() 呼び出しを行うモジュールによって使用され、パイプラインから自身を削除します。
- **ns.inet** — インターネットアドレスのライブラリ。

- **ns.inet.apply_mask(address_str, mask_str)** -マスク文字列を適用した後にアドレス文字列を返します。
- **ns.inet.aton (address_str)** -アドレスの数値をネットワーク順のバイト文字列として返します (IPv4 の場合は 4 バイト、IPv6 の場合は 16 バイト)。
- **ns.inet.ntoa (byte_str)** -数値バイト値をバイト文字列としてアドレス文字列に変換します。
- **ns.inet.ntohs(number)** -指定されたネットワークバイトオーダーをホストバイトオーダーに変換します。入力が 2 ^ 16-1 より大きい場合、エラーがスローされます。
- **ns.inet.htons(number)** -指定されたホストのバイト順序をネットワークのバイト順序に変換します。入力が 2 ^ 16-1 より大きい場合、エラーがスローされます。
- **ns.inet.ntohl(number)** -指定されたネットワークバイトオーダーをホストバイトオーダーに変換します。入力が 2 ^ 32-1 より大きい場合、エラーがスローされます。
- **ns.inet.htonl(number)** -指定されたホストのバイト順序をネットワークのバイト順序に変換します。入力が 2 ^ 32-1 より大きい場合、エラーがスローされます。
- **ns.inet.subnet(address_str, subnet_value)** –指定されたサブネットを適用した後、サブネットアドレス文字列を返します。

プロトコル拡張

October 7, 2021

Citrix ADC アプライアンスは、HTTP などのプロトコルをネイティブでサポートしています。これに加えて、プロトコル拡張を使用して、カスタムプロトコルのサポートを追加できます。現在、TCP ベースのカスタムプロトコルのみがサポートされています。たとえば、メッセージキューテレメトリトランスポート (MQTT) プロトコルです。セキュリティで保護されたトランザクションの場合、TCP over SSL もサポートされています。

Citrix ADC アプライアンスのプロトコル拡張は、Citrix ADC アプライアンスで使用可能な高レベルのスクリプトインフラストラクチャの一部です。スクリプト言語は Lua 5.2 プログラミング言語に基づいています。Citrix ADC アプライアンスにカスタムプロトコルを追加するには、該当する動作を実装するための拡張コードを記述する必要があります。たとえば、ns.tcp.client および ns.tcp.server の動作は、TCP ベースのプロトコルに適用できます。ビヘイビアを実装するには、カスタマイズするコールバックのみを実装します。コールバックが実装されていない場合、そのデフォルトが有効になります。スクリプト言語の詳細については、Citrix ADC 拡張機能-言語の概要を参照してください。動作の詳細については、Citrix ADC 拡張機能の API リファレンスを参照してください。

Citrix ADC プロトコル拡張は、次の用途に使用できます。

- Citrix ADC アプライアンスの新しいプロトコルサポートを、拡張機能を使用してプログラムで追加します。
- プロトコルトラフィックを解析し、プロトコル固有のメッセージベースのロードバランシング (MBLB) を実行します。
- ユーザ定義のロードバランシングの永続性を設定します。

プロトコル拡張 - アーキテクチャ

October 7, 2021

トラフィックレベルの拡張性を実現するために、Citrix ADC アプライアンスでのトラフィック処理は、個別の処理モジュールのパイプラインとして公開されます。トラフィックは、入力から出力にトラフィックを処理するにつれて、トラフィックがフローします。パイプライン内のこれらのモジュールは、共有なしモデルに従います。メッセージパッシングは、パイプライン内の 1 つのモジュールから次のモジュールにトラフィックデータを送信するために使用されます。

トラフィック処理パイプラインの特定のポイントは拡張可能であるため、コードを追加して Citrix ADC の動作をカスタマイズできます。

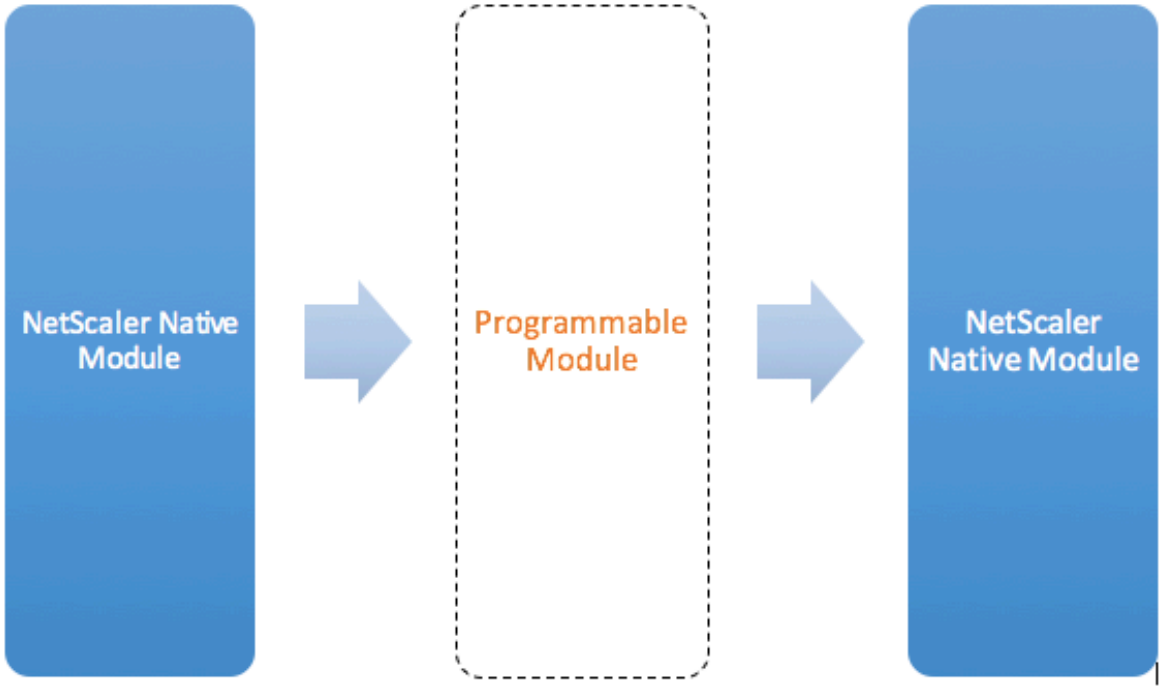


Figure: A Programmable Module In the Traffic Pipeline

デフォルトでは、トラフィックはコードを追加しないプログラマブルモジュールをバイパスします。

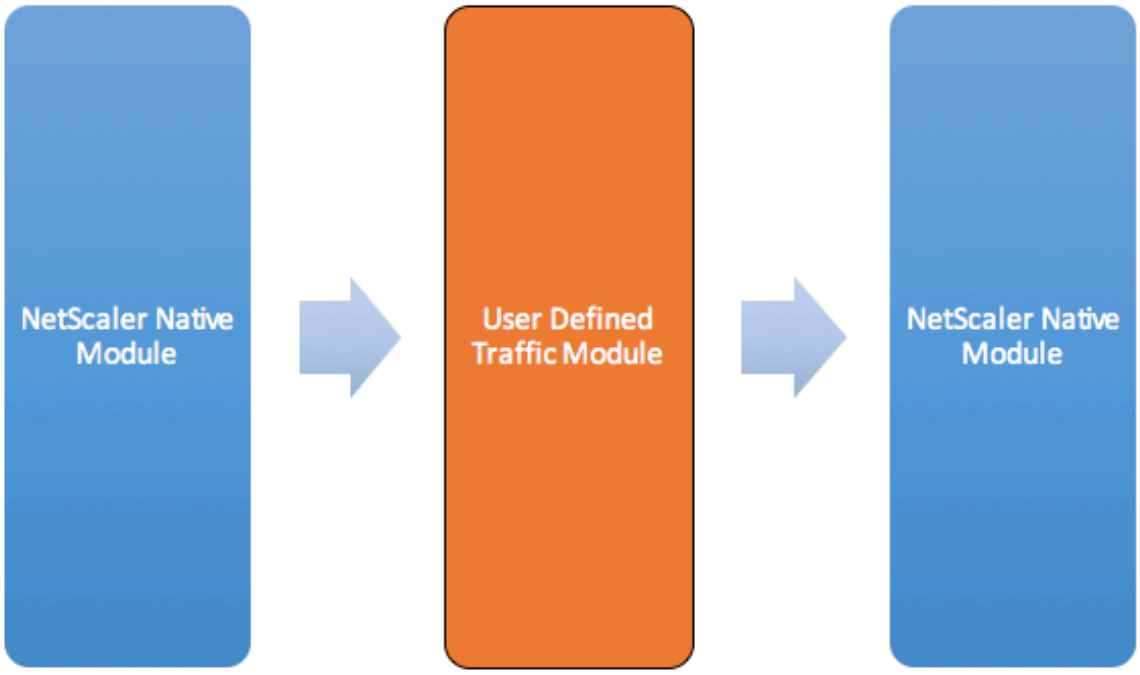


Figure: User Defined Traffic Module

ビヘイビア

トラフィック処理をカスタマイズするためのプログラマブルインターフェイスは、動作と呼ばれます。動作は、基本的に、Citrix ADC アプライアンスで使用できる一般的なプログラマブルパターンの形式化です。ビヘイビアは、事前に定義された一連のイベントコールバック関数で構成されます。動作に適合するコールバック関数を提供することで、動作を実装できます。

たとえば、TCP クライアントの動作は、TCP クライアントデータストリームイベントを処理するコールバック関数 (on_data) で構成されます。TCP ベースのプロトコルに Message Based Load Balancing (MBLB) を実装するには、このコールバック関数のコードを追加して、クライアントからの TCP データストリームを処理し、バイトストリームをプロトコルメッセージに解析します。

コンテキスト**:**

ビヘイビア内のコールバック関数は、処理モジュールの状態であるコンテキストで呼び出されます。コンテキストは、処理モジュールのインスタンスです。たとえば、TCP クライアント動作コールバックは、クライアント TCP 接続ごとに異なるコンテキストで呼び出されます。

ペイロード**:**

コンテキストに加えて、ビヘイビアコールバックは他の引数を持つことができます。通常、残りの引数はペイロードとして渡されます。ペイロードは、すべての引数のコレクションです。

したがって、プログラマブル処理モジュールのインスタンスは、インスタンス状態とイベントコールバック関数、つまりコンテキストと動作の組み合わせとして見ることができます。トラフィックはイベントペイロードとしてパイプラインを通過します。

Citrix ADC API 拡張機能については、Citrix ADC 拡張 API リファレンスを参照してください。

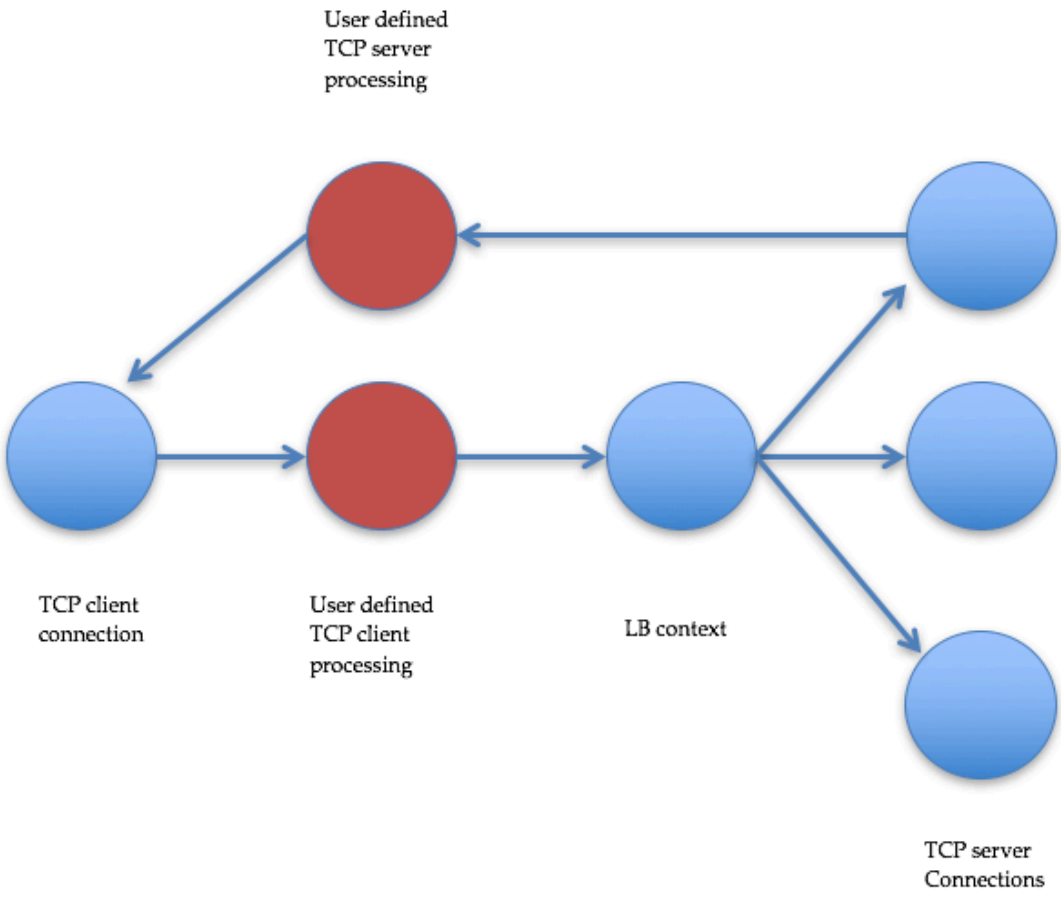
次のコードスニペットは、TCP クライアントデータストリームイベントを処理するユーザー定義関数を示しています。コンテキストとペイロードは、Citrix ADC コードによって関数に渡されます。このコードは、すべての呼び出しで受信した TCP データを、パイプライン内の次の処理モジュールコンテキストに転送するだけです。この場合、次のモジュールは負荷分散（LB）コンテキストであり、これは Citrix ADC ネイティブモジュールです。

```
function client.on_data(ctxt, payload)
    ns.send(ctxt.output, ”DATA”, {
 data = payload.data }
 )
end
<!--NeedCopy-->
```

プロトコル拡張 - ユーザー定義 **TCP** クライアントとサーバーの動作のトラフィックパイプライン

October 7, 2021

次の図は、プロトコル拡張の例を示しています。ユーザー定義の TCP クライアントとサーバー動作のトラフィックパイプライン



Traffic Pipeline For User Defined TCP Client And Server Behaviors

プロトコル拡張を使用してカスタムプロトコルを追加する

カスタムプロトコルの Command Line Interface（CLI; コマンドラインインターフェイス）コマンドは、キーワード「user」を使用して、基礎となる設定エンティティのユーザ定義の性質を示します。拡張コードの助けを借りて、新しいユーザプロトコルをシステムに追加し、ユーザ定義プロトコルのユーザ仮想サーバを追加することができます。ユーザー仮想サーバーは、パラメータを設定することによって順番に設定できます。仮想サーバパラメータの設定値は、拡張コードで使用できます。

次に、新しいプロトコルのサポートを追加するためのユーザフローの例を示します。この例では、MQTT プロトコルのサポートをシステムに追加します。MQTT は、マシン間の「モノのインターネット」接続プロトコルです。これは、軽量なパブリッシュ/サブスクライブメッセージングトランスポートです。リモートロケーションとの接続に便利なこのプロトコルは、クライアントおよびブローカツールを使用して、サブスクライバにメッセージをパブリッシュします。

1. MQTT プロトコル拡張実装ファイルを Citrix ADC システムにインポートします。mqtt.lua のコードリスト

は以下の通りです。以下の例では、ウェブサーバー上でホストされている MQTT 拡張ファイルをインポートしています。

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
```

2. 拡張を使用して、新しいユーザ TCP ベースのプロトコルをシステムに追加します。

```
add user protocol MQTT -transport TCP -extension mqtt_code
```

3. ユーザロードバランシング vserver を追加し、それにバックエンドサービスをバインドします。

```
1 add service mqtt_svr1 10.217.24.48 USER_TCP 1501
2 add service mqtt_svr2 10.217.24.48 USER_TCP 1502
3 add lb vserver mqtt_lb USER_TCP  - lbmethod USER_TOKEN
4 bind lb vserver mqtt_lb mqtt_svr1
5 bind lb vserver mqtt_lb mqtt_svr2
6 <!--NeedCopy-->
```

4. 新しく追加されたプロトコルのユーザー vserver を追加します。defaultlb を、上記で設定した LB 仮想サーバに設定します。

```
add user vserver mqtt_vs MQTT 10.217.24.28 8765 -defaultLb mqtt_lb
```

5. オプションで、ClientID に基づいて MQTT セッションパーシステンスを有効にし、パーシステンスタイプを USERSESSION に設定します。

```
set lb vserver mqtt_lb -persistenceType USERSESSION
```

プロトコル拡張 - ユースケース

October 7, 2021

プロトコル拡張は、次のユースケースで使用できます。

- メッセージベースの負荷分散（MBLB）
- ストリーミング
- トークン・ベースのロード・バランシング
- ロード・バランシングの永続性
- TCP 接続ベースのロードバランシング
- コンテンツ・ベースのロード・バランシング
- SSL
- トラフィックを変更する
- クライアントまたはサーバへのトラフィックの発信元
- 接続確立のデータを処理する

メッセージ・ベースのロード・バランシング

プロトコル拡張では、メッセージベース負荷分散（MBLB）がサポートされています。MBLB（メッセージベース負荷分散）は、Citrix ADC アプライアンス上の任意のプロトコルを解析し、1 つのクライアント接続で受信したプロトコルメッセージを負荷分散します。MBLB は、クライアント TCP データストリームを解析するユーザーコードによって実現されます。

TCP データストリームは、クライアントとサーバーの動作のために on_data コールバックに渡されます。TCP データストリームは、インタフェースのような Lua 文字列を介して拡張関数に利用可能です。Lua 文字列 API に似た API を使用して、TCP データストリームを解析できます。

有用な API には、次のものがあります。

```
data:len()
```

```
data:find()
```

```
data:byte()
```

```
data:sub()
```

```
data:split()
```

TCP データストリームがプロトコルメッセージに解析されると、ユーザーコードは、クライアント用の on_data コールバックに渡されたコンテキストから利用可能な次のコンテキストにプロトコルメッセージを送信するだけで負荷分散を実現します。

ns.send () API は、他の処理モジュールにメッセージを送信するために使用されます。送信先コンテキストに加えて、send API はイベント名とオプションのペイロードを引数として受け取ります。イベント名とビヘイビアのコールバック関数名には、1 対 1 の対応があります。イベントのコールバックは on_ と呼ばれます <event_name>。コールバック名は小文字のみを使用します。

たとえば、TCP クライアントおよびサーバーの on_data コールバックは、「DATA」という名前のイベントのユーザー定義ハンドラーです。1 回の送信コールでプロトコルメッセージ全体を送信するには、EOM イベントが使用されます。EOM は、メッセージの終わりを表し、LB コンテキストのダウンストリームへのプロトコルメッセージの終わりを示します。したがって、このメッセージの後に続くデータに対して新しいロードバランシングの決定が行われます。

拡張コードは、on_data イベントでプロトコルメッセージ全体を受信しないことがあります。このような場合には、ctxt: hold () API を使用してデータを保持することができます。保留 API は、TCP クライアントコンテキストとサーバコールバックコンテキストの両方で使用できます。「データで保持」が呼び出されると、データはコンテキストに格納されます。同じコンテキストでより多くのデータが受信されると、新しく受信したデータが以前に格納されたデータに追加され、on_data コールバック関数が結合されたデータで再び呼び出されます。

注：使用される負荷分散方法は、負荷分散コンテキストに対応する負荷分散仮想サーバーの構成によって異なります。

次のコードスニペットは、send API を使用して解析されたプロトコルメッセージを送信する方法を示しています。

例**:**


```
    function client.on_data(ctxt, payload)
        --
        -- code to parse payload.data into protocol message comes here
        --
        -- sending the message to lb
        ns.send(ctxt.output, ”EOM”, {
 data = message }
 )
    end -- client.on_data

    function server.on_data(ctxt, payload)
        --
        -- code to parse payload.data into protocol message comes here
        --
        -- sending the message to client
        ns.send(ctxt.output, ”EOM”, {
 data = message }
 )

    end -- server.on_data
<!--NeedCopy-->
```

ストリーミング

シナリオによっては、プロトコルメッセージ全体が収集されるまで TCP データストリームを保持する必要がない場合があります。実際には、それが必要でない限り、それは助言されません。データを保持すると、Citrix ADC アプライアンスのメモリ使用量が増加し、多くの接続で不完全なプロトコルメッセージで Citrix ADC アプライアンスのメモリを使い果たすことにより、アプライアンスが DDoS 攻撃を受けやすくなります。

ユーザーは、send API を使用して、拡張コールバックハンドラで TCP データのストリーミングを実現できます。メッセージ全体が収集されるまでデータを保持する代わりに、データをチャンクで送信できます。DATA イベントを使用して ctxt.output にデータを送信すると、部分的なプロトコルメッセージが送信されます。これは、より多くの DATA イベントが続くことができます。プロトコルメッセージの終わりをマークするには、EOM イベントを送信する必要があります。ロードバランシングコンテキストは、最初に受信したデータに対してロードバランシングの決定を行います。EOM メッセージの受信後、新しいロードバランシングの決定が行われます。

プロトコルメッセージデータをストリーミングするには、複数の DATA イベントの後に EOM イベントを送信します。連続する DATA イベントと次の EOM イベントは、シーケンスの最初の DATA イベントのロードバランシングの決定によって選択された同じサーバー接続に送信されます。

クライアントコンテキストに送信する場合、EOM イベントと DATA イベントは事実上同じです。これは、クライアントコンテキストのダウンストリームによる EOM イベントに対する特別な処理がないためです。

トークン・ベースのロード・バランシング

ネイティブでサポートされるプロトコルの場合、Citrix ADC アプライアンスは、PI 式を使用してトークンを作成するトークンベースの負荷分散方式をサポートします。拡張の場合、プロトコルは事前に知られていないため、PI 式は使用できません。トークンベースの負荷分散では、USER_TOKEN 負荷分散メソッドを使用するようにデフォルトの負荷分散仮想サーバーを設定し、user_token フィールドで送信 API を呼び出して拡張コードからトークン値を指定する必要があります。トークン値が送信 API から送信され、USER_TOKEN ロードバランシング方式がデフォルトのロードバランシング仮想サーバーで構成されている場合、トークンの値に基づいてハッシュを計算することによってロードバランシングの決定が行われます。トークン値の最大長は 64 バイトです。

```
add lb vserver v\\_mqttlb USER\\_TCP -lbMethod USER\\_TOKEN
```

次の例のコードスニペットは、送信 API を使用して LB トークン値を送信します。

例:

```
        -- send the message to lb



        -- user_token is set to do LB based on clientID



        ns.send(ctxt.output, "EOM", {
 data = message,

                                user_token = token_info }
 )
<!--NeedCopy-->
```

ロード・バランシングの永続性

ロードバランシングの永続性は、トークンベースのロードバランシングと密接に関連しています。ユーザーは、永続性セッション値をプログラムで計算し、永続性を負荷分散するために使用できる必要があります。送信 API は、永続性パラメータを送信するために使用されます。負荷分散の永続性を使用するには、デフォルトの負荷分散仮想サーバーで USERSESSION 永続性タイプを設定し、user_session フィールドを指定して送信 API を呼び出して、拡張コードから永続性パラメータを指定する必要があります。永続性パラメータ値の最大長は 64 バイトです。

カスタムプロトコルに複数のタイプの永続性が必要な場合は、ユーザー永続性タイプを定義して設定する必要があります。仮想サーバの設定に使用されるパラメータの名前は、プロトコル実装者によって決定されます。パラメータの

設定値は、拡張コードでも使用できます。

次の CLI とコードスニペットは、ロードバランシングの永続性をサポートする送信 API の使用方法を示しています。mqtt.lua のコードリストのセクションにあるコードリストでは 、user_session フィールドの使用方法も示しています。

永続性については、負荷分散仮想サーバーで USERSESSION 永続性タイプを指定し、ns.send API から user_session 値を渡す必要があります。

```
add lb vserver v\\_mqttlb USER\\_TCP -persistencetype USERSESSION
```

ペイロードで user_session フィールドを clientID に設定して、MQTT メッセージをロードバランサーに送信します。

例:

```
1 -- send the data so far to lb
2
3 -- user_session is set to clientID as well (it will be used to persist
     session)
4
5 ns.send(ctxt.output, “DATA” , {
6  data = data, user_session = clientID }
7  )
8 <!--NeedCopy-->
```

TCP 接続ベースのロードバランシング

一部のプロトコルでは、MBLB は必要ない場合があります。代わりに、TCP 接続ベースの負荷分散が必要になる場合があります。たとえば、MQTT プロトコルは TCP ストリームの最初の部分を解析して、負荷分散用のトークンを決定する必要があります。また、同じ TCP 接続上のすべての MQTT メッセージは、同じサーバー接続に送信する必要があります。

TCP 接続ベースのロードバランシングは、DATA イベントのみで send API を使用し、EOM を送信しないことで実現できます。このようにして、ダウンストリームの負荷分散コンテキストは、最初に受信したデータに基づいて負荷分散を決定し、その後のすべてのデータを負荷分散の決定によって選択された同じサーバー接続に送信します。

また、ユースケースによっては、ロードバランシングの決定後に拡張処理をバイパスする機能が必要になる場合があります。拡張呼び出しをバイパスすると、トラフィックが純粋にネイティブコードで処理されるため、パフォーマンスが向上します。バイパスは、ns.pipe () API を使用して行うことができます。pipe () API 拡張コードを呼び出すと、入力コンテキストを出力コンテキストに接続できます。pipe () の呼び出しの後、入力コンテキストからのすべてのイベントが直接出力コンテキストに移動します。実際には、pipe () 呼び出しが行われたモジュールはパイプラインから削除されます。

次のコードスニペットは、ストリーミングと pipe () API を使用してモジュールをバイパスする方法を示しています。mqtt.lua のコードリストのセクションにあるコードリストは 、ストリーミングを行う方法と pipe () API を使用して接続上の残りのトラフィックに対してモジュールをバイパスする方法も示しています。

例:

```
        -- send the data so far to lb
        ns.send(ctxt.output, "DATA", {
 data = data,
                                        user_token = clientID }
 )
        -- pipe the subsequent traffic to the lb - to bypass the client
            on_data handler
        ns.pipe(ctxt.input, ctxt.output)
<!--NeedCopy-->
```

コンテンツ・ベースのロード・バランシング

ネイティブプロトコルの場合、プロトコル拡張のコンテンツスイッチングと同様の機能がサポートされます。この機能を使用すると、デフォルトのロードバランサーにデータを送信する代わりに、選択したロードバランサーにデータを送信できます。

プロトコル拡張のコンテンツスイッチング機能は、ctxt: lb_connect (<lbname>) API を使用することによって実現されます。この API は、TCP クライアントコンテキストで使用できます。拡張コードは、この API を使用して、すでに設定されている負荷分散仮想サーバーに対応する負荷分散コンテキストを取得できます。その後、取得した負荷分散コンテキストで送信 API を使用できます。

lb コンテキストは NULL になることがあります。

- 仮想サーバが存在しません
- 仮想サーバーがユーザープロトコルタイプではありません
- 仮想サーバの状態が UP ではありません
- 仮想サーバは、負荷分散仮想サーバではなく、ユーザーの仮想サーバです。

ターゲットの負荷分散仮想サーバーの使用中に削除すると、その負荷分散仮想サーバーに関連付けられているすべての接続がリセットされます。

次のコードスニペットは、lb_connect () API の使用方法を示しています。このコードは、Lua テーブル lb_map を使用してクライアント ID を負荷分散仮想サーバー名 (lbname) にマップし、lb_connect () を使用して lbname の LB コンテキストを取得します。最後に、send API を使用して LB コンテキストに送信します。

```
    local lb_map = {
```



```
2
3          ["client1*"] = "lb_1",
4          ["client2*"] = "lb_2",
5          ["client3*"] = "lb_3",
6          ["client4*"] = "lb_4"
7       }
8
9
10      -- map the clientID to the corresponding LB vserver and connect to
           it
11      for client_pattern, lbname in pairs(lb_map) do
12         local match_idx = string.find(clientID, client_pattern)
13         if (match_idx == 1) then
14        lb_ctxt = ctxt:lb_connect(lbname)
15        if (lb_ctxt == nil) then
16           error("Failed to connect to LB vserver: " .. lbname)
17        end
18        break
19         end
20      end
21      if (lb_ctxt == nil) then
22      -- If lb context is NULL, the user can raise an error or send data
           to default LB
23         error("Failed to map LB vserver for client: " .. clientID)
24      end
25  -- send the data so far to lb
26  ns.send(lb_ctxt, "DATA", {
27   data = data }
28
29  <!--NeedCopy-->
```

SSL

拡張機能を使用するプロトコルの SSL は、ネイティブプロトコルの SSL がサポートされる方法と同様の方法でサポートされます。カスタムプロトコルを作成するために同じ解析コードを使用して、TCP または SSL を介してプロトコルインスタンスを作成し、仮想サーバーを構成するために使用することができます。同様に、TCP または SSL 経由でユーザーサービスを追加できます。

詳細については、MQTT の SSL オフロードの設定およびエンドツーエンド暗号化を使用したMQTT の SSL オフロードの設定を参照してください。

サーバー接続の多重化

場合によっては、クライアントは一度に 1 つの要求を送信し、最初の要求の応答がサーバーから受信された後に次の要求を送信します。この場合、サーバー接続は、応答がクライアントに送信された後、他のクライアント接続および同じ接続上の次のメッセージに再利用できます。他のクライアント接続によるサーバー接続の再利用を許可するには、サーバー側のコンテキストで ctxt: reuse_server_connection () API を使用する必要があります。

注：この API は、Citrix ADC 12.1 ビルド 49.xx 以降で使用できます。

トラフィックを変更する

要求または応答のデータを変更するには、高度なポリシー PI 式を使用するネイティブの書き換え機能を使用する必要があります。拡張で PI 式を使用できないため、次の API を使用して TCP ストリームデータを変更できます。

```
data:replace(offset, length, new_string)
data:insert(offset, new_string)
data:delete(offset, length)
data:gsub(pattern, replace [,n]))
```

次のコードスニペットは、replace () API の使用方法を示しています。

```
-- Get the offset of the pattern, we want to replace
   local old_pattern =  “pattern to repalace”
local old_pattern_length = old_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
    goto send_data
   end
  -- If the data we want to modify is not completely present, then
  -- wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
data:replace(pat_off, old_pattern_length,  “new pattern” )
::send_data::
ns.send(ctxt.output,  “EOM” , {
 data = data }
 )
::done::
```


次のコードスニペットは、insert () API の使用方法を示しています。

```
data:insert(5, “pattern to insert” )
```

次のコードスニペットは、いくつかのパターンの前または後に挿入したいときに、insert () API の使用を示しています。

```
-- Get the offset of the pattern, after or before which we want to
    insert
   local pattern = “pattern after/before which we need to insert”
local pattern_length = pattern:len()
   local pat_off, pat_end = data:find(pattern)
-- pattern is not present
   if (not pat_off) then
    goto send_data
   end
  -- If the pattern after which we want to insert is not
  -- completely present, then wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
-- Insert after the pattern
data:insert(pat_end + 1, “pattern to insert” )
   -- Insert before the pattern
data:insert(pat_off, “pattern to insert” )
::send_data::
    ns.send(ctxt.output, “EOM” , {
 data = data }
 )
::done::
```

次のコードスニペットは、delete () API の使用方法を示しています。

```
-- Get the offset of the pattern, we want to delete
   local delete_pattern = “pattern to delete”
local delete_pattern_length = delete_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
```



```
            goto send_data
     end
    -- If the data we want to delete is not completely present,
    -- then wait for more data
    if (not pat_end) then
          ctxt:hold(data)
          data = nil
      goto done
    end
data:delete(pat_off, delete_pattern_length)
::send_data::
ns.send(ctxt.output, "EOM" , {
 data = data }
 )
::done::
```

次のコードスニペットは、gsub () API の使用を示しています。

```
    -- Replace all the instances of the pattern with the new string
data:gsub( "old pattern" , "new string" )
-- Replace only 2 instances of "old pattern"
data:gsub( "old pattern" , "new string" , 2)
-- Insert new_string before all instances of "http"
data:gsub( "input data" , "(http)" , "new_string%1" )
-- Insert new_string after all instances of "http"
data:gsub( "input data" , "(http)" , "%1new_string" )
-- Insert new_string before only 2 instances of "http"
data:gsub( "input data" , "(http)" , "new_string%1" , 2)
```

注: この API は、Citrix ADC 12.1 ビルド 50.xx 以降で使用できます。

クライアントまたはサーバへのトラフィックの発信元

ns.send () API を使用して、拡張コードから生成されたデータをクライアントとバックエンドサーバーに送信できます。クライアントコンテキストからクライアントと直接応答を送受信するには、ctxt.client をターゲットとして使用する必要があります。サーバーコンテキストからバックエンドサーバーと直接応答を送受信するには、ctxt.server をターゲットとして使用する必要があります。ペイロードのデータは、TCP ストリームデータまたは Lua 文字列です。

接続でのトラフィック処理を停止するには、クライアントまたはサーバコンテキストから ctxt: close () API を使用できます。この API は、クライアント側の接続またはそれにリンクされているサーバー接続を閉じます。

ctxt: close () API を呼び出すと、拡張コードがクライアントとサーバー接続に TCP FIN パケットを送信します。この接続でクライアントまたはサーバーからより多くのデータが受信されると、アプライアンスは接続をリセットしま

す。

次のコードスニペットは、ctxt.client と ctxt の使用方法を示しています。クローズ () API。

```
    -- If the input packet is not MQTT CONNECT type, then
-- send some error response to the client.
function client.on_data(ctxt, payload)
    local data = payload.data
    local offset = 1
    local msg_type = 0
    local error_response =  “Missing MQTT Connect packet.”
    byte = data:byte(offset)
msg_type = bit32.rshift(byte, 4)
if (msg_type ~= 1) then
-- Send the error response
   ns.send(ctxt.client,  “DATA” , {
 data = error_response }
 )
-- Since error response has been sent, so now close the connection
    ctxt:close()
end
```

次のコードスニペットは、ユーザーが通常のトラフィックフローにデータを注入できる例を示しています。

```
-- After sending request, send some log message to the server.
function client.on_data(ctxt, payload)
local data = payload.data
local log_message =  “client id :  “..data:sub(3, 7)..”  user name :  “
    data:sub(9, 15)
-- Send the request we get from the client to backend server
ns.send(ctxt.output,  “DATA” , {
 data = data }
 )
After sending the request, also send the log message
ns.send(ctxt.output,  “DATA” , {
 data = log_message”  }
 )
end
```

次のコードスニペットは、ctxt.to_server API の使用方法を示しています。


```
-- If the HTTP response status message is  “Not Found” ,
-- then send another request to the server.
function server.on_data(ctxt, payload)
    local data = payload.data
    local request  “GET /default.html HTTP/1.1\r\n\r\n” ss
    local start, end = data:find( “Not Found” )
    if (start) then
    -- Send the another request to server
        ns.send(ctxt.server,  “DATA” , {
 data = request }
 )
end
```

注：この API は、Citrix ADC 12.1 ビルド 50.xx 以降で使用できます。

接続確立でのデータ処理

接続確立時にデータを送信したいユースケースがあるかもしれません（最終的な ACK を受信したとき）。たとえば、プロキシプロトコルでは、接続確立時にクライアントの送信元と宛先 IP アドレスとポートをバックエンドサーバーに送信できます。この場合、client.init () コールバックハンドラを使用して、接続確立時にデータを送信することができます。

次のコードスニペットは、client.init () コールバックの使用方法を示しています。

```
-- Send a request to the next processing context
-- on the connection establishment.
function client.init(ctxt)
   local request  “PROXY TCP4”  + ctxt.client.ip.src.to_s +  “  “ +
       ctxt.client.ip.dst.to_s +  “  “ + ctxt.client.tcp.srcport +  “  “
        +     ctxt.client.tcp.dstport
-- Send the another request to server
   ns.send(ctxt.output,  “DATA” , {
 data = request }
 )
  end
```

注：この API は、Citrix ADC 13.0 ビルド xx.xx 以降で使用できます。

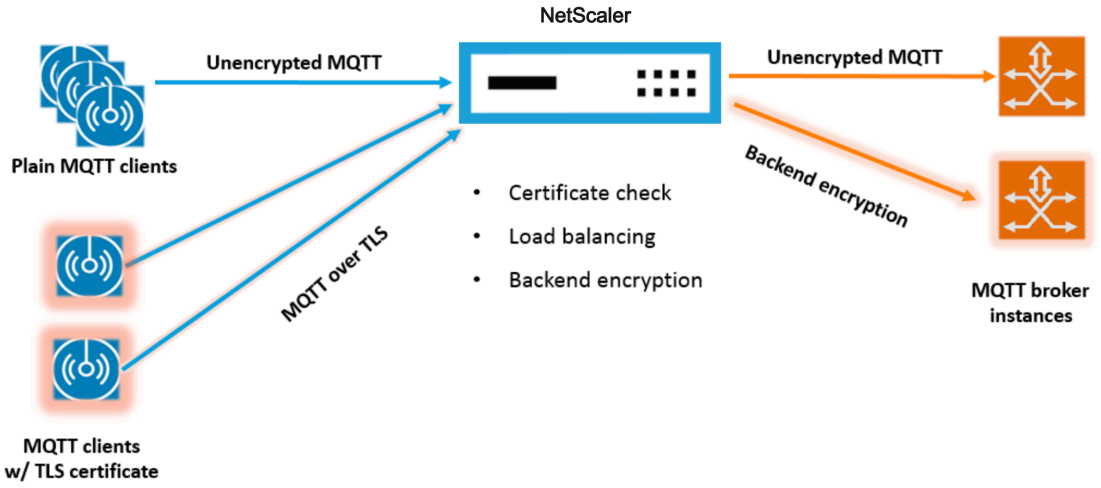
チュートリアル — プロトコル拡張を使用して **MQTT** プロトコルを **Citrix ADC** アプライアンスに追加する

October 7, 2021

カスタムプロトコルの Command Line Interface（CLI; コマンドラインインターフェイス）コマンドは、キーワード「user」を使用して、基礎となる設定エンティティのユーザ定義の性質を示します。拡張コードの助けを借りて、新しいユーザプロトコルをシステムに追加し、ユーザ定義プロトコルのユーザ仮想サーバを追加することができます。ユーザー仮想サーバーは、パラメータを設定することによって順番に設定できます。仮想サーバパラメータの設定値は、拡張コードで使用できます。

MQTT プロトコルは説明目的で使用されます。

次の図は、Citrix ADC アプライアンスと MQTT クライアントとブローカーツールを示しています。



mqtt.lua のコードリスト

October 7, 2021

以下のコードである mqtt.lua は、プロトコル拡張を使用して Citrix ADC で MQTT プロトコルを実装するためのコードを示します。このコードには、TCP クライアントデータコールバック関数 (client.on_data ()) のみが定義されています。サーバーデータの場合、コールバック関数を追加せず、サーバーからクライアントへの高速ネイティブパスを取得します。クライアントデータの場合、コードは CONNECT MQTT プロトコルのメッセージを解析し、ClientID を抽出します。次に、user_token の ClientID を使用します。この値は、LB 仮想サーバーの LB メソッドを USER_TOKEN に設定することで、クライアント ID に基づいて接続のすべてのクライアントトラフィックの負荷分散に使用されます。ユーザーセッション値にもクライアント ID を使用します。これは、LB 仮想サーバーの永続性タイプを USERSESSION に設定することで、LB 永続性に使用できます。このコードでは、ns.send () を使用して

LB を行い、初期データを送信します。ns.pipe () API を使用して、残りのクライアントトラフィックをサーバー接続に直接送信し、拡張コールバックハンドラへの呼び出しをバイパスします。

```
--[[

  MQTT event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

    - parse the client ID from the connect message - the first message
       should be connect

    - send the data to LB with ClientID as user token and session

    - pipe the subsequent data to LB directly. This way the subsequent
       MQTT traffic will

      bypass the tcp client on_data handler

    - if a parse error is seen, throw an error so the connection is
       reset

--]]

function client.on_data(ctxt, payload)

    local data = payload.data

    local data_len = data:len()

    local offset = 1

    local byte = nil

    local utf8_str_len = 0

    local msg_type = 0

    local multiplier = 1

    local max_multiplier = 128 * 128 * 128
```



```
38
39      local rem_length = 0
40
41      local clientID = nil
42
43      -- check if MQTT fixed header is present (fixed header length is
            atleast 2 bytes)
44
45      if (data_len < 2) then
46
47          goto need_more_data
48
49      end
50
51      byte = data:byte(offset)
52
53      offset = offset + 1
54
55         -- check for connect packet - type value 1
56
57      msg_type = bit32.rshift(byte, 4)
58
59      if (msg_type ~= 1) then
60
61          error("Missing MQTT Connect packet.")
62
63      end
64
65      -- parse the remaining length
66
67      repeat
68
69          if (multiplier > max_multiplier) then
70
71             error("MQTT CONNECT packet parse error - invalid Remaining
                   Length.")
72
73         end
74
75         if (data_len < offset) then
76
77            goto need_more_data
78
79         end
80
```



```
81          byte = data:byte(offset)
82
83          offset = offset + 1
84
85          rem_length = rem_length + (bit32.band(byte, 0x7F) * multiplier)
86
87          multiplier = multiplier * 128
88
89       until (bit32.band(byte, 0x80) == 0)
90
91       -- protocol name
92
93       -- check if protocol name length is present
94
95       if (data_len < offset + 1) then
96
97          goto need_more_data
98
99       end
100
101      -- protocol name length MSB
102
103      byte = data:byte(offset)
104
105      offset = offset + 1
106
107      utf8_str_len = byte * 256
108
109      -- length LSB
110
111      byte = data:byte(offset)
112
113      offset = offset + 1
114
115      utf8_str_len = utf8_str_len + byte
116
117         -- skip the variable header for connect message
118
119      -- the four required fields (protocol name, protocol level, connect
           flags, keep alive)
120
121      offset = offset + utf8_str_len + 4
122
123      -- parse the client ID
124
```



```
125     --
126
127     -- check if client ID len is present
128
129     if (data_len < offset + 1) then
130
131        goto need_more_data
132
133     end
134
135        -- client ID length MSB
136
137     byte = data:byte(offset)
138
139     offset = offset + 1
140
141     utf8_str_len = byte * 256
142
143     -- length LSB
144
145     byte = data:byte(offset)
146
147     offset = offset + 1
148
149     utf8_str_len = utf8_str_len + byte
150
151     if (data_len < (offset + utf8_str_len - 1)) then
152
153         goto need_more_data
154
155     end
156
157     clientID = data:sub(offset, offset + utf8_str_len - 1)
158
159     -- send the data so far to lb, user_token is set to do LB based on
           clientID
160
161     -- user_session is set to clientID as well (it will be used to
           persist session)
162
163     ns.send(ctxt.output, ”DATA”, {
164  data = data,
165
166                                 user_token = clientID,
167
```



```
                                    user_session = clientID }
  )

        -- pipe the subsequent traffic to the lb - to bypass the
           extension handler

    ns.pipe(ctxt.input, ctxt.output)

    goto parse_done

    ::need_more_data::

    ctxt:hold(data)

    ::parse_done::

    return

end
<!--NeedCopy-->
```

プロトコル拡張を使用した **MQTT** の構成

October 7, 2021

以下の手順では、Citrix ADC アプライアンスに MQTT プロトコルを追加します。

Web サーバー（HTTP を使用）またはローカルワークステーションから、拡張子ファイルを Citrix ADC アプライアンスにインポートします。拡張ファイルのインポートの詳細については、「拡張機能のインポート」を参照してください。

```
import ns extension local:mqtt_generic_fs.lua mqtt_code
```

拡張を使用して、新しいユーザ TCP ベースのプロトコルをシステムに追加します。

```
add user protocol MQTT -transport TCP -extension mqtt_code
```

USER_TCP タイプのサービスを追加して、これがユーザー定義プロトコルであることを示します。

```
add service s1 10.102.90.112 USER_TCP 80
```

ユーザロードバランシング vserver を追加し、それにバックエンドサービスをバインドします。

```
add lb vs mysv USER_TCP
```

```
bind lb vs mysv s1
```


新しく追加したプロトコルのユーザー仮想サーバーを追加し、前のステップで設定した負荷分散仮想サーバーをデフォルトのロードバランサーにします。

```
add user vs v_mqtt MQTT 10.217.24.28 80 -defaultlb mysv
```

オプションで、ClientID に基づいて MQTT セッションパーシステンスを有効にし、パーシステンスタイプを USERSESSION に設定します。

```
set lb vserver mqtt_lb -persistenceType USERSESSION
```

MQTT の SSL オフロードの設定

October 7, 2021

プロトコルの SSL インスタンスを追加することで、ユーザープロトコルの SSL オフロードを実装できます。以下の例は、ユーザープロトコルに対して SSL オフロードを行う方法を示しています。この設定では、バックエンドサービスへのトラフィックは暗号化されません。

注意: この例では、証明書とキーのペアを追加または更新し、それを仮想サーバーにバインドすることに関する詳細は提供していません。詳細については、SSL 証明書を参照してください。

次のコマンドは、mqtt.lua をトランスポート値「SSL」に含めて MQTT_SSL プロトコルを追加します。

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
add user protocol MQTT_SSL -transport SSL -extension mqtt_code
<!--NeedCopy-->
```

次のコマンドは、ユーザー負荷分散仮想サーバーを追加し、それにバックエンドサービスをバインドします。

```
add service mqtt_svr1 10.217.24.48 USER_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_TCP 1502
add lb vserver mqtt_lb USER_TCP  - lbMethod ROUNDROBIN
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

次のコマンドは、新しく追加されたプロトコル MQTT_SSL 用のユーザー仮想サーバーを追加します。MQTT_SSL を使用すると、Citrix ADC アプライアンスで SSL オフロードを実行することになります。これは、MQTT_SSL が SSL トランスポートで設定されているためです。また、このコマンドは、前の手順で設定した負荷分散仮想サーバーに defaultlb を設定します。

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb mqtt_lb
```


SSL オフロードでは、SSL 機能を有効にして、認証キーをユーザー仮想サーバーにバインドする必要もあります。詳しくは、次のトピックを参照してください:

証明書とキーのペアの追加または更新

証明書とキーのペアを SSL 仮想サーバーにバインドする

例:

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```

MQTT のエンドツーエンド暗号化による SSL オフロードの設定

October 7, 2021

次の例は、エンドツーエンド暗号化で MQTT の SSL オフロードを行う方法を示しています。

注意: この例では、証明書とキーのペアを追加または更新し、それを仮想サーバーにバインドすることに関する詳細は提供していません。詳細については、SSL 証明書を参照してください。

次のコマンドは、拡張ファイルをインポートし、SSL トランスポートで MQTT_SSL プロトコルを追加します。

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
add user protocol MQTT_SSL -transport SSL -extension mqtt_code
<!--NeedCopy-->
```

次のコマンドは、ユーザー負荷分散仮想サーバーを追加し、それにバックエンドサービスをバインドします。負荷分散仮想サーバーとサービスの両方が、サービスタイプ USER_SSL_TCP 用に構成されています。

```
add service mqtt_svr1 10.217.24.48 USER_SSL_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_SSL_TCP 1502
add lb vserver mqtt_lb USER_SSL_TCP  - lbmethod RR
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```


次のコマンドは、新しく追加されたプロトコル MQTT_SSL 用のユーザー仮想サーバーを追加します。MQTT_SSL を使用すると、Citrix ADC アプライアンスで SSL オフロードを実行することになります。これは、MQTT_SSL が SSL トランスポートで設定されているためです。また、このコマンドは、前の手順で設定した負荷分散仮想サーバーをデフォルトのロードバランサーにします。

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb mqtt_lb
```

エンドツーエンドの暗号化では、SSL 機能を有効にして、証明書キーをユーザーおよびデフォルトの負荷分散仮想サーバーにバインドする必要もあります。詳しくは、次のトピックを参照してください:

証明書とキーのペアの追加または更新

証明書とキーのペアを SSL 仮想サーバーにバインドする

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_lb  -certkeyName mqtt_svr_cert_key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```

チュートリアル-プロトコル拡張を使用した **syslog** メッセージのロードバランシング

October 7, 2021

Citrix ADC アプライアンスで使用可能な Syslog プロトコルは、Citrix ADC アプライアンスで生成されたメッセージに対してのみ機能します。外部ノードからのメッセージの負荷分散は行われません。このようなメッセージをロードバランシングするには、プロトコル拡張機能を使用し、Lua 5.2 プログラミング言語を使用して syslog メッセージ解析ロジックを記述する必要があります。

syslog メッセージを解析するためのコード

このコードには、TCP クライアントデータコールバック関数 (client.on_data ()) のみが定義されています。サーバーデータの場合、コールバック関数を追加せず、サーバーからクライアントへの高速ネイティブパスを取得します。このコードは、末尾の文字に基づいてメッセージの境界を識別します。TCP パケットに複数の syslog メッセージが含まれている場合は、末尾の文字に基づいてパケットを分割し、各メッセージをロードバランシングします。


```
--[[

  Syslog event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

--]]

function client.on_data(ctxt, payload)

                  local message = nil

                  local data_len

                  local data = payload.data

                  local trailing_character = "\n"

                  ::split_message::

                                    -- Get the offset of trailing
                                       character

                                    local new_line_character_offset =
                                       data:find(trailing_character)

                                    -- If trailing character is not
                                       found, then wait for more data.

                                    if (not new_line_character_offset)
                                       then

                                                      goto
                                                         need_more_data

                                    end

                                    -- Get the length of the current
                                       message

                                    data_len = data:len()

```



```
39                                  -- Check whether we have more than
                                       one message
40
41                                  -- by comparing trailing character
                                       offset and
42
43                                  -- current data length
44
45                                  if (data_len >
                                       new_line_character_offset) then
46
47                                                  -- If we have
                                                       more than one
                                                       message, then
                                                       split
48
49                                                  -- the data into
                                                       two parts such
                                                        that first
                                                       part
50
51                                                  -- will contain
                                                       message upto
                                                       trailing
                                                       character
52
53                                                  -- offset and
                                                       second part
                                                       will contain
54
55                                                  -- remaining
                                                       message.
56
57                                                  message, data =
                                                       data:split(
                                                       new_line_character_offse
                                                       )
58
59                                  else
60
61                                                  message = data
62
63                                                  data = nil
64
65                                  end
```



```
66
67 -- Send the data to the backend server.
68
69                                     ns.send(ctxt.output, ”EOM”, {
70  data = message }
71  )
72
73                                     goto done
74
75                                     ::need_more_data::
76
77                                                     -- Wait for more
                                                        data
78
79                                                     ctxt:hold(data)
80
81                                                     data = nil
82
83                                                     goto done
84
85                                     ::done::
86
87                                                     -- If we have
                                                        more data to
                                                        parse,
88
89                                                     -- then do
                                                        parsing again.
90
91                                                     if (data) then
92
93                                                                     goto
                                                                        split_
94
95                                                     end
96
97 end
98 <!--NeedCopy-->
```


プロトコル拡張を使用した **syslog** プロトコルの設定

October 7, 2021

以下の手順では、Citrix ADC アプライアンスにユーザーの SYSLOG プロトコルを追加します。

Web サーバー（HTTP を使用）またはローカルワークステーションから、拡張子ファイルを Citrix ADC アプライアンスにインポートします。拡張ファイルのインポートの詳細については、拡張機能のインポートを参照してください。

```
import ns extension local:syslog_parser.lua syslog_parser_code
```

拡張を使用して、新しいユーザー TCP ベースのプロトコルをシステムに追加します。

```
add user protocol USER_SYSLOG -transport TCP -extension syslog_parser_code
```

USER_TCP タイプのサービスを追加して、これがユーザー定義プロトコルであることを示します。

```
add service s1 10.102.90.112 USER_TCP 80
```

ユーザロードバランシング vserver を追加し、それにバックエンドサービスをバインドします。

```
1 add lb vs mysv USER_TCP
2
3 bind lb vs mysv s1
4 <!--NeedCopy-->
```

新しく追加したプロトコルのユーザー仮想サーバーを追加し、前のステップで設定した負荷分散仮想サーバーをデフォルトのロードバランサーにします。

```
add user vs v_syslog USER_SYSLOG 10.217.24.28 80 -defaultlb mysv
```

プロトコル拡張 **-** コマンドリファレンス

October 7, 2021

次の表は、カスタムプロトコルに対して追加された新しいコマンドと、カスタムプロトコル用に変更された既存のコマンドの一覧です。

```
show lb persistentSessions [<vserv-name>]
```

- **CLI** コマンド：

  ```
  add user protocol <name> -transport ( TCP | SSL )-extension <string> -
  comment <string>]>
  ```


- 説明:

  拡張機能を使用して、Citrix ADC アプライアンスに新しいユーザープロトコルを追加します。現在、トランスポート値 TCP または SSL を持つユーザプロトコルのみがサポートされています。

  > 例:
  >
  > add user protocol MQTT -transport TCP -extension mqtt_code

- **CLI** コマンド:

  ```
  rm user protocol <name>
  ```

- 説明:

  Citrix ADC アプライアンスに追加したユーザープロトコルを削除します。

  > 例:
  >
  > rm user protocol mqtt

- **CLI** コマンド:

  ```
  set user protocol <name> -comment <string>
  ```

- 説明:

  Citrix ADC アプライアンスに追加したユーザープロトコルの設定を変更します。

  > 例:
  >
  > set user protocol mqtt -comment “MQTT protocol implementation”

- **CLI** コマンド:

  ```
  unset user protocol <name> -comment
  ```

- 説明:

  Citrix ADC アプライアンスに追加したユーザープロトコルの設定を削除します。

  > 例:
  >
  > unset user protocol mqtt -comment “MQTT protocol implementation”

- **CLI** コマンド:

  ```
  update ns extension <extension name>
  ```

- 説明:

  拡張機能を使用して、以前に追加されたユーザープロトコルの実装を更新します。

  プロトコルの実装を更新できるのは、そのプロトコルがユーザーの仮想サーバーによって使用されていない場合だけです。

> 例:
>
> update ns extension my-extension

- **CLI** コマンド:

```
add lb vserver <name> [USER_TCP | USER_SSL_TCP] [-lbmethod USER_TOKEN]
[-persistencetype USERSESSION] [-timeout <value>]
```

- 説明:

  負荷分散仮想サーバーを Citrix ADC アプライアンスに追加します。これは既存の CLI コマンドです。

  負荷分散ユーザー仮想サーバーの場合、使用するサービスタイプは USER_TCP または USER_SSL_TCP です。IP アドレスとポートは、ユーザーロードバランシング仮想サーバーでは許可されません。

  ユーザー負荷分散仮想サーバーでは、ROUNDROBIN 負荷分散方式のみが許可され、トークン値は拡張コードによって提供されます。同様に、USERSESSION 永続性のみが許可され、永続性設定は拡張コードによって提供されます。

  > 例:
  >
  > add lb vserver mysv USER_TCP –lbmethod ROUNDROBIN

- **CLI** コマンド:

```
add user vserver <name> <userProtocol> <IPAddress> <port> -defaultLB <
string> [-params <string>] [-comment <string>]
```

- 説明:

  拡張機能を使用して、ユーザープロトコルの仮想サーバーを追加します。設定されたデフォルトのユーザロードバランシング仮想サーバは、ctxt.output として TCP クライアントデータ拡張ハンドラで使用できます。仮想サーバーの場合、拡張パラメータは、名前と値のペアを指定して-params オプションを使用して設定できます。対応するパラメータ値は、拡張ハンドラで ctxt.vserver.params.<paramName> として利用できます。

  > 例:
  >
  > add user vs v_mqtt MQTT 10.217.24.28 80 -defaultlb mysv

- **CLI** コマンド:

```
rm user vserver <name>
```

- 説明:

  Citrix ADC アプライアンスに追加済みのユーザー仮想サーバーを削除します。

  > 例:
  >
  > rm user vserver v_mqtt

- **CLI** コマンド:

```
set user vserver <name> [-IPAddress <ip_addr|ipv6_addr|*>] [-defaultLB
<string>] [-params <string>] [-comment <string>]
```

- 説明:

  Citrix ADC アプライアンスに追加したユーザー仮想サーバーの設定を変更します。拡張パラメータに-params オプションによって新しい値が割り当てられると、古い値は上書きされます。

  > 例:
  >
  > set user vs v_mqtt MQTT 10.217.24.28 -defaultlb mysv -comment “MQTT protocol implementation”

- **CLI** コマンド:

```
unset user vserver <name> [-params] [-comment]
```

- 説明:

  Citrix ADC アプライアンスに追加したユーザー仮想サーバーの設定を削除します。—params オプションを使用して拡張パラメータの設定を解除すると、拡張ハンドラで使用できる対応するパラメータ値が nil に変更されます。

  > 例:
  >
  > unset user vs v_mqtt MQTT 10.217.24.28 -defaultlb mysv -comment “MQTT protocol implementation”

- **CLI** コマンド:

```
show user protocol [<name>]
```

- 説明:

  拡張機能やコールバックなど、ユーザプロトコルに関する情報を表示します。

  > 例:
  >
  > show user protocol mqtt

- **CLI** コマンド:

```
show user vserver [<name>]
```

- 説明:

  ユーザー仮想サーバーに関する情報を表示します。

  > 例:
  >
  > show user vserver vs_mqtt

- **CLI** コマンド:

```
stat user vserver [<name>]
```

- 説明:

  ユーザー仮想サーバーに関する統計情報を表示します。

  > 例:
  >
  > stat user vserver vs_mqtt

- **CLI** コマンド:

```
show lb persistentSessions [<vserv-name>]
```

- 説明:

  永続セッションに関する情報を表示します。これは既存の CLI です。ユーザープロトコルの場合、持続性タイプは USERSESSION と表示されます。

- **CLI** コマンド:

```
rm lb vserver <name>
```

- 説明:

  Citrix ADC アプライアンスに追加したユーザー LB 仮想サーバーを削除します。

  > 例:
  >
  > rm lb vserver mysv

- **CLI** コマンド:

```
add service <name> <IPAddr> (USER_TCP | USER_SSL_TCP)<Port>
```

- 説明:

  ユーザプロトコルに使用するバックエンドサービスを追加します。これは、新しいサービスタイプ USER_TCP および USER_SSL_TCP を持つ既存の CLI コマンドです。

  > 例:
  >
  > add service mqtt_svr1 10.217.24.48 USER_TCP 1501
  >
  > 注意: 既存の「サービスの設定およびサービスの設定解除」コマンドを使用して、ユーザープロトコルに対して以前に追加したサービスの設定を削除または変更することができます。

- **CLI** コマンド:

```
bind lb vserver <name> <serviceName>
```

- 説明:

  サービスをユーザー LB vserver にバインドします。サービスタイプは、タイプが USER_TCP/USER_SSL_TCP の LB 仮想サーバーにバインドする場合は、USER_TCP/USER_SSL_TCP である必要があります。

> 例:
>
> bind lb vserver mysv mqtt_svr1

- **CLI** コマンド:

  ```
  unbind lb vserver <name> <serviceName>
  ```

- 説明:

  以前にバインドされたサービスをユーザー LB vserver にバインド解除します。

  > 例:
  >
  > unbind lb vserver mysv mqtt_svr1

- **CLI** コマンド:

  ```
  rm service <name>
  ```

- 説明:

  ユーザープロトコルに対して以前に追加されたサービスを削除します。

  > 例:
  >
  > rm service mqtt_svr1

プロトコル拡張のトラブルシューティング

October 7, 2021

拡張関数が期待どおりに動作しない場合は、拡張トレース機能を使用して、拡張関数の動作を確認できます。カスタムログ機能を使用して、拡張機能にログを追加することもできます。カスタムログ機能では、Citrix ADC アプライアンスでキャプチャするログレベルを定義できます。

カスタムロギング

拡張関数に独自のロギングを追加することもできます。これを行うには、組み込みの ns.logger: level () 関数を使用します。level は、緊急、アラート、クリティカル、エラー、警告、通知、情報、デバッグです。パラメータは、C printf () 関数と同じです。フォーマット文字列と、フォーマット文字列で指定された% の値を指定する可変数の引数です。たとえば、COMBINE_HEADERS 関数に次のコードを追加して、呼び出しの結果を記録できます。

```
1 local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"
2
3 ns.logger:info("Result: %s", result_str)
```



```
4
5 return result_str
6 <!--NeedCopy-->
```

上記の関数は、上記の「拡張トレース」セクションの省略されたログメッセージの例に示されたサンプル入力について、次のメッセージを /var/log/ns.log に記録します。

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M
 H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4)libcurl
/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: \*/\*^M H2: h2val1, h2val2,
h2val3^M ^M"
```

ポリシー拡張

October 7, 2021

ポリシー拡張機能を使用すると、組み込みのポリシータイプの拡張機能を記述できます。拡張機能は、組み込み関数と同様に、ポリシー式で使用できます。これらは、対応するポリシー式が評価されるときに実行されます。この機能は次の場合に役立ちます。

- カスタマイズした機能を既存のポリシーに追加する。
- 複雑な顧客要件のための論理構造の実装。

ポリシー拡張機能は、組み込みのポリシータイプの拡張機能をユーザが記述できるようにすることで、これらの制限に対処します。拡張機能は、組み込み関数と同様に、ポリシー式で使用できます。これらは、対応するポリシー式が評価されるときに実行されます。

次の表は、拡張機能の記述時に使用できるポリシータイプと、関連するマッピングの一覧です。

ポリシーの種類	マッピングされたポリシータイプ	出力
TEXT_T	NSTEXT	文字列
BOOL_AT	NSBOOL	ブーリアン型
NUM_AT	NSNUM	数値 (倍精度浮動小数点)
DOUBLE_AT	NSDOUBLE	数値 (倍精度浮動小数点)

ポリシー拡張を使用するための前提条件

インポートされた関数は、既存のポリシー標準に準拠している必要があります。したがって、次のようになります。

- 関数名は文字で始まり、数字またはアンダースコアを含むことができます。

- Citrix ADC ポリシーでは、関数名の大文字と小文字は区別されません。
- 拡張言語が複数の値を返す場合でも、関数は 1 つの値を返す必要があります。
- 可変数の引数を持つ関数はサポートされていません。

ポリシー拡張機能の仕組み

Citrix ADC アプライアンスの既存のポリシーは、インタープリタを使用して機能を評価します。これらの機能は、ポリシー拡張ファイルにインポートされます。ユーザーがポリシー拡張ファイルに新しい関数をインポートすると、次のようになります。

1. 拡張子ファイルは、構文やその他の条件について検証されます。
2. 検証が失敗した場合、エラーはユーザーに報告されます。
3. 検証に成功すると、拡張ファイルは Citrix ADC アプライアンスにインポートされ、その内容は組み込みのポリシー機能と同様に、ポリシー式で使用できます。
   a) ポリシー式の評価が実行中にエラーを返した場合、その評価は undef イベントとして報告され、関連するエラーカウンタが増分されます。
   注： policy undef イベントが発生し、ポリシー規則に 1 つ以上のポリシー拡張機能が含まれている場合、`show ns extension <name>` コマンドはそれらのポリシー拡張に適用されると、undef ヒットを表示します。拡張機能が中断されると、中止カウンタの値が増加します。
   b) ポリシー式の評価が成功すると、式全体が評価されるまで、またはエラーのために中止されるまで、式の評価が再開されます。

拡張関数の実行に時間がかかりすぎると中止され、その拡張関数に関連するエラーカウンタが増加します。拡張関数はサンドボックス化され、次のことを防ぎます。

- Citrix ADC アプライアンスで過剰な CPU 使用率。
- Citrix ADC アプライアンスのメモリ使用量が多すぎます。
- 有害な組み込みライブラリまたはサードパーティのライブラリまたはバイナリの使用。
- Citrix ADC アプライアンスが再起動する可能性がある長時間実行されるスクリプト。

ポリシー拡張の設定

October 7, 2021

ポリシー拡張ファイルの準備ができたら、Citrix ADC アプライアンスにインポートします。インポートプロセスでは、拡張子ファイルが Citrix ADC アプライアンス上のディレクトリにコピーされ、構文エラーがないかどうかがチェックされます。

インポート後、拡張ファイルをポリシー式で使用できるようにする必要があります。

注： import コマンドは、外部ソース \<src\>または内部ソースから Citrix ADC ファイルシステムにファイルコンテンツをダウンロードするために使用されます。このファイルコンテンツを 1 つ以上のパケットエンジンに初めて

ロードするには、add コマンドを使用します。ファイルコンテンツに更新がある場合、overwrite 引数を指定してインポートコマンドを発行することで、更新されたコンテンツを Citrix ADC ファイルシステムにダウンロードできます。このコマンドは、ファイルシステムの内容を更新します。更新されたコンテンツを 1 つ以上のパケットエンジンにロードするには、update コマンドを使用します。

CLI を使用したポリシー拡張の設定

1. Web サーバー（HTTP を使用）またはローカルワークステーションから、ポリシー拡張ファイルを Citrix ADC アプライアンスにインポートします。

   a) HTTP インポート

   使用可能な Web サーバーがある場合は、拡張ファイルを Webserver ディレクトリに保存し、Citrix ADC アプライアンスにインポートできます。

   ```
   import ns extension <src> <name> [-comment<string>] [-
      overwrite]
   <!--NeedCopy-->
   ```

   例:

   ```
   import ns extension http://myhost/path/to/extension
      myextension -comment  “Custom crc calculation
   <!--NeedCopy-->
   ```

   b) ローカルインポート

   SSH クライアントを使用して、ワークステーションから Citrix ADC アプライアンスの/var/tmp ディレクトリに拡張ファイルをコピーできます。

   ```
   scp extension-file-name <ns-userid@ns-ip-addr>:/var/tmp
   <!--NeedCopy-->
   ```

   各項目の意味は次のとおりです。

   - extension-file-name は、クライアントマシン上の拡張ファイルの名前です。
   - ns-userid Citrix ADC アプライアンスユーザーには、/var/tmp への書き込み権限が付与されているか。
   - ns-ip-addr は、Citrix ADC IP アドレスです。

ファイルを Citrix ADC アプライアンスにコピーした後、Citrix ADC アプライアンス上でインポートコマンドを実行します。

```
import ns extension local:<extension-file-name extension-name>
<!--NeedCopy-->
```

注**:** ローカル拡張ファイルをインポートするには、
import コマンドを実行して CLI を使用する必要があります。

2. 評価用にポリシー拡張をパケットエンジンに追加します。

```
add ns extension <name> [-comment <string>]
<!--NeedCopy-->
```

例**:**

```
add ns extension myextension
<!--NeedCopy-->
```

拡張ファイルをインポートした後、import コマンドに-overwrite パラメータを指定した場合は、拡張ファイルを更新したり、削除したりできます。インポートした拡張ファイルの詳細を表示することもできます。

Citrix ADC アプライアンス上の拡張ファイルをソースから更新する

コマンドプロンプトで入力します。

```
update ns extension <name>
<!--NeedCopy-->
```

注：拡張子ファイルは、-overwrite パラメーターを使用して指定した拡張子ファイルを Citrix ADC アプライアンスにインポートした後にのみ更新できます。

例**:**

```
update ns extension myextension
<!--NeedCopy-->
```


Citrix ADC アプライアンスからの拡張ファイルの削除

コマンドプロンプトで次のように入力します。

```
rm ns extension <name>
<!--NeedCopy-->
```

例**:**

```
rm ns extension myextension
<!--NeedCopy-->
```

Citrix ADC アプライアンス上で指定した拡張機能の詳細を表示します

コマンドプロンプトで入力します。

```
show ns extension <name>
<!--NeedCopy-->
```

例**:**

```
show ns extension myextension
<!--NeedCopy-->
```

GUI を使用したポリシー拡張の設定

1. Web サーバー（HTTP を使用）またはローカルワークステーションから、ポリシー拡張ファイルを Citrix ADC アプライアンスにインポートします。

   a) [**AppExpert**] > [ポリシー拡張] に移動し、[ポリシー拡張] をクリックし、[インポート元**]** ドロップダウンリストから、インポートする拡張ファイルの場所の URL を選択します。

   b) **AppExpert** > ポリシー拡張機能、ポリシー拡張に移動し、[インポート元] ドロップダウンリストで [ファイル] を選択して拡張ファイルをインポートします。

2. 評価用にポリシー拡張をパケットエンジンに追加します。

   [**AppExpert**] > [ポリシー拡張] に移動し、[ポリシー拡張] タブで拡張ファイルを追加します。

Citrix ADC アプライアンス上の拡張ファイルをソースから更新する

[**AppExpert**] > [ポリシー拡張] に移動し、[ポリシー拡張] タブで拡張ファイルを更新します。

Citrix ADC アプライアンスからの拡張ファイルの削除

AppExpert > ポリシー拡張に移動し、[ポリシー拡張] タブで、拡張ファイルを削除します。

Citrix ADC アプライアンス上で指定した拡張機能の詳細を表示します

AppExpert > ポリシー拡張に移動し、[ポリシー拡張機能] タブで、詳細を表示する拡張機能の [クリック] ドロップダウンリスト矢印をクリックします。

ポリシー拡張 - ユースケース

October 7, 2021

特定のお客様のアプリケーションには、既存のポリシーや表現では対応できない要件があります。ポリシー拡張機能を使用すると、要件に合わせてカスタマイズした機能をアプリケーションに追加できます。

以下のユースケースは、Citrix ADC アプライアンスのポリシー拡張機能を使用した新しい機能の追加を示しています。

- ケース 1: カスタムハッシュ
- ケース 2: URL の二重スラッシュを折りたたむ
- ケース 3: ヘッダーを結合する

ケース **1:** カスタムハッシュ

CUSTOM_HASH 関数は、クライアントに送信される応答に任意のタイプのハッシュ値を挿入するメカニズムを提供します。このユースケースでは、ハッシュ関数を使用して、書き換え HTTP リクエストのクエリ文字列のハッシュを計算し、計算された値を持つ CUSTOM_HASH という名前の HTTP ヘッダーを挿入します。CUSTOM_HASH 関数は DJB2 ハッシュアルゴリズムを実装します。

CUSTOM_HASH の使用例**:**

```
1 > add rewrite action test_custom_hash insert_http_header ”CUSTOM_HASH”
    ”HTTP.REQ.URL.QUERY.CUSTOM_HASH”
2 <!--NeedCopy-->
```


CUSTOM_HASH () のサンプル定義**:**

```
    -- Extension function to compute custom hash on the text

    -- Uses the djb2 string hash algorithm
    function NSTEXT:CUSTOM_HASH() : NSTEXT

        local hash = 5381

        local len = string.len(self)

        for i = 1, len do

             hash = bit32.bxor((hash * 33), string.byte(self, i))

        end

        return tostring(hash)

    end
<!--NeedCopy-->
```

上記のサンプルの行ごとの説明:

```
function NSTEXT:CUSTOM_HASH() : NSTEXT

Defines the CUSTOM_HASH() function, with text input and a text return
   value.

local hash = 5381
local len = string.len(self)

Declares two local variables:

 - hash. Accumulates the compute hash value and is seeded with the
    number 5381

 - len. Sets to the length of the self input text string, using the
    built-in string.len() function.

for i = 1, len do
      hash = bit32.bxor((hash * 33), string.byte(self, i))
```



```
16 end
17
18 Iterates through each byte of the input string and adds the byte to the
       hash. It uses the built-in string.byte() function to get the byte
      and the built-in bit32.bxor() function to compute the XOR of the
      existing hash value (multiplied by 33) and the byte.
19
20 return tostring(hash)
21
22 Calls the built-in tostring() function to convert the numeric hash
      value to a string and returns the string as the value of the
      function.
23 <!--NeedCopy-->
```

ケース **2: URL** の二重スラッシュを折りたたむ

URL で二重スラッシュを折りたたむと、ブラウザが単一のスラッシュの URL をより効率的に解析できるため、Web サイトのレンダリング時間が短縮されます。単一のスラッシュ URL は、二重スラッシュを受け入れないアプリケーションとの互換性を維持するためにも使用します。ポリシー拡張機能を使用すると、URL 内の二重スラッシュを 1 つのスラッシュに置き換える関数を追加できます。次の例は、URL 内の二重スラッシュを折りたたむポリシー拡張関数の追加を示しています。

定義例: **COLLAPSE_DOUBLE_SLASHES():**

```
1     -- Collapse double slashes in URL to a single slash and return the
         result
2     function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT
3
4         local result = string.gsub(self, "//", "/")
5
6         return result
7
8     end
9 <!--NeedCopy-->
```

上記のサンプルの行ごとの説明:

```
1 function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT
2
3 Declares the COLLAPSE_DOUBLE_SLASHES() function with text input and
      return.
```



```
4
5  local result = string.gsub(self, "//", "/")
6
7  Declares a local variable named result and uses the built-in string.
       gsub() function to replace all double slashes with single slashes in
        the self input text.
8
9  The second parameter of string.gsub() is actually a regular expression
       pattern, although here a simple string is used for the pattern.
10
11 return result
12
13 Returns the resulting string.
14 <!--NeedCopy-->
```

ケース 3: ヘッダーを結合する

特定の顧客アプリケーションは、リクエストで複数のヘッダーを処理できません。また、同じヘッダー値を持つ重複ヘッダー、または要求内の同じ名前で異なる値を持つ複数のヘッダーの解析は、時間とネットワークリソースを消費します。ポリシー拡張機能を使用すると、これらのヘッダーを元の値を組み合わせた値を持つ単一のヘッダーに結合する関数を追加できます。たとえば、ヘッダー H1 と H2 の値を組み合わせます。

元のリクエスト:

```
1  GET /combine_headers HTTP/1.1
2  User-Agent: amigo unit test
3  Host: myhost
4  H2: h2val1
5  H1: abcd
6  Accept: \*/\*
7  H2: h2val2
8  Content-Length: 0
9  H2: h2val3
10 H1: 1234
11 <!--NeedCopy-->
```

変更されたリクエスト:

```
1  GET /combine_headers HTTP/1.1
2  User-Agent: amigo unit test
3  Host: myhost
```



```
H2: h2val1, h2val2, h2val3
H1: abcd, 1234
Accept: \*/\*
Content-Length: 0
<!--NeedCopy-->
```

一般に、このタイプの要求の変更は、ポリシー式を使用して、変更される要求の一部（ターゲット）と実行する変更（文字列ビルダー式）を描写します。ただし、ポリシー式には、任意の数のヘッダーを反復処理する機能はありません。

この問題を解決するには、ポリシー機能への拡張が必要です。これを行うには、COMBINE_HEADERS と呼ばれる拡張関数を定義します。この機能を使用すると、次の書き換えアクションを設定できます。

```
> add rewrite action combine_headers_act replace 'HTTP.REQ.FULL_HEADER
.AFTER_STR(”HTTP/1.1rn”)''HTTP.REQ.FULL_HEADER.AFTER_STR(”HTTP/1.1rn”).
COMBINE_HEADERS'
```

ここでは、書き換えターゲットは HTTP.REQ.FULL_HEADER.AFTER_STR（「HTTP/1.1rn」）です。FULL_HEADER には HTTP リクエストの最初の行が含まれているため、AFTER_STR（「HTTP/1.1rn」）が必要です（例：GET /結合ヘッダー HTTP/1.1）。

文字列ビルダー式は HTTP.REQ.FULL_HEADER.AFTER_STR（「HTTP/1.1rn」）.COMBINE_HEADERS です。ここで、ヘッダー（最初の行を引いたもの）は、ヘッダーの値を結合して返します。

COMBINE_HEADERS() のサンプル定義**:**

```
    -- Extension function to combine multiple headers of the same name
       into one header.


    function NSTEXT:COMBINE_HEADERS(): NSTEXT

        local headers = {
}
-- headers

        local combined_headers = {
}
-- headers with final combined values
        -- Iterate over each header (format ”name:valuer\r\n”)

        -- and build a list of values for each unique header name.

        for name, value in string.gmatch(self, ”([^:]+):([^\r\n]*)\r\n”
           ) do
```



```
19
20              if headers[name] then
21
22                  local next_value_index = #(headers[name]) + 1
23
24                  headers[name][next_value_index] = value
25
26              else
27
28                  headers[name] = {
29  name .. ":" .. value }
30
31
32              end
33
34          end
35
36
37
38          -- iterate over the headers and concat the values with
                separator ","
39
40          for name, values in pairs(headers) do
41
42              local next_header_index = #combined_headers + 1
43
44              combined_headers[next_header_index] = table.concat(values,
                    ",")
45
46          end
47
48
49
50          -- Construct the result headers using table.concat()
51
52          local result_str = table.concat(combined_headers, "\r\n") .. "\
                r\n\r\n"
53
54          return result_str
55
56      end
57  <!--NeedCopy-->
```

上記のサンプルの行ごとの説明:


```
function NSTEXT:COMBINE_HEADERS(): NSTEXT

Defines the COMBINE_HEADERS extension function, with the text input
    into the function from the policy expression and a text return type
    to the policy expression.

local headers = {
  }
  -- headers
local combined_headers = {
  }
  -- headers with final combined values

Declares local variables headers and combined_headers and initialize
    these variables to empty tables. headers will be a table of arrays
    of strings, where each array holds one or more values for a header.
    combined_headers will be an array of strings, where each array
    element is a header with its combined values.

for name, value in string.gmatch(self, "([^:]+):([^\r\n]*)\r\n") do
. . .
end
<!--NeedCopy-->
```

この汎用的な for ループは、入力の各ヘッダーを解析します。イテレータは、組み込みの string.gmatch () 関数です。この関数は、検索する文字列と、文字列の一部を一致させるために使用するパターンの 2 つのパラメータを取ります。検索する文字列は、関数に入力されたヘッダーのテキストである暗黙の self パラメータによって提供されます。

パターンは、正規表現（短い場合は正規表現）を使用して表現されます。この正規表現は、HTTP 標準では name: value rn として定義されている、各ヘッダーのヘッダー 名と ** 値に一致します。正規表現の括弧は抽出する一致する部分を指定するので、正規表現の回路図は（*match-name*）:（*match-value*）rn です。*match-name* パターンは、コロン以外のすべての文字と一致する必要があります。これは [^: + ([^:] :] 以外の任意の文字、+ は 1 つ以上の繰り返し) と書かれています。同様に、マッチ値パターンは \ r\ n 以外の任意の文字と一致する必要があるため、[^\ r\ n (\ r と\ n]*[^\ r\ n (\ r と\ n]* 以外の任意の文字に一致し、はゼロ以上の繰り返しです) と書かれます。これにより、完全な正規表現（[^: + (^: :]+）:（[^\ r\ n (\ r と\ n]*）\r\n になります。

for 文は、string.gmatch () イテレータによって返された 2 つの一致に名前と値を設定するために、複数の割り当てを使用しています。これらは、for ループの本体内でローカル変数として暗黙的に宣言されています。

```
if headers[name] then
     local next_value_index = #(headers[name]) + 1
     headers[name][next_value_index] = value
```



```
else
     headers[name] = {
 name .. ":" .. value }

end
<!--NeedCopy-->
```

for ループ内のこれらのステートメントは、ヘッダー名と値をヘッダーテーブルに入れます。ヘッダー名を初めて解析する場合 (たとえば、入力例では H2: h2val1)、名前のヘッダーエントリはなく、[ヘッダー名は] nil です。

nil は false として扱われるので、else 節が実行されます。これは、name のヘッダエントリを 1 つの文字列値 *name*:*value* を持つ配列に設定します。

注意**:** else ループの配列コンストラクタは {[1] = name と同じです。「:」.. value}。配列の最初の要素を設定します。) 最初の H2 ヘッダーには、headers[“H2”] = {“H2:h2val1”} を設定します。

ヘッダーの後続のインスタンス（たとえば、入力例では H2: h2val2)。headers[ヘッダー名は] は nil ではないので、then 節が実行されます。これは、headers[ヘッダー名は] の配列値の中で次に使用可能なインデックスを決定し、そのインデックスにヘッダー値を置きます。2 番目の H2 ヘッダーについては、headers[“H2”] ={“H2:h2val1”, “h2val2”} を設定します。

```
for name, values in pairs(headers) do
    local next_header_index = #combined_headers + 1
    combined_headers[next_header_index] = table.concat(values, ",")
end
<!--NeedCopy-->
```

元のヘッダーが解析され、ヘッダーテーブルが入力された後、このループは combined_headers 配列を構築します。これは、for ループイテレータとして pairs() 関数を使用しています。

pairs() を呼び出すたびに、ヘッダーテーブル内の次のエントリの名前と値を返します。

次の行は、combined_headers 配列内の次に使用可能なインデックスを決定し、次の行は、結合されたヘッダーにその配列要素を設定します。組み込みの table.concat () 関数を使用します。この関数は、引数として文字列の配列と文字列をセパレータとして使用し、セパレータで区切られた配列文字列を連結した文字列を返します。

たとえば、値 = {“H2:h2val1”, “h2val2”} の場合、これは”H2:h2val1, h2val2” を生成します

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"
<!--NeedCopy-->
```

combined_headers 配列が構築された後、要素を 1 つの文字列に連結し、HTTP ヘッダーを終了する二重の rn を追加します。


```
return result_str
<!--NeedCopy-->
```

COMBINE_HEADERS 拡張関数の結果として文字列を返します。

ポリシー拡張のトラブルシューティング

October 7, 2021

拡張関数が期待どおりに動作しない場合は、拡張トレース機能を使用して、拡張関数の動作を確認できます。カスタムログ機能を使用して、拡張機能にログを追加することもできます。カスタムログ機能では、Citrix ADC アプライアンスでキャプチャするログレベルを定義できます。

このトピックでは、次の情報について説明します。

- 拡張トレース
- カスタムロギング

拡張トレース

拡張機能の実行内容を表示するには、拡張トレース機能によって関数の実行が Citrix ADC システムログ (/var/log/ns.log) に記録されます。トレースログでは、DEBUG ログレベルが使用されますが、通常は有効になりません。したがって、すべてのログレベルを有効にする必要があります。次に、set ns 拡張コマンドの-trace オプションを設定して、トレースを有効にできます。使用可能な設定は次のとおりです。

- off トレースをオフにします (ns 拡張子-trace を設定解除した場合と同等)。
- トレース関数呼び出しを引数で呼び出し、関数は最初の戻り値で返します。
- 行は、実行された行の上記のプラス行番号をトレースします。
- は、上記に加えて、実行された行によって変更されたローカル変数をトレースします。

例:

```
set audit syslogParams -loglevel ALL

set ns extension combine_headers -trace all
<!--NeedCopy-->
```

各トレースメッセージは次の形式です。

```
log-header : default NSEXTENSION Message message-number 0 : "TRACE function
-name CALL call-number: event"
```


各項目の意味は次のとおりです。

- log-header は、タイムスタンプ、Citrix ADC IP アドレス、パケットエンジン ID を提供します。
- message-number は、ログメッセージを識別する連番です。
- function-name は、拡張関数名です。
- call-number は、各拡張関数呼び出しの連番です。これは、拡張関数呼び出しのすべてのトレースメッセージをグループ化するために使用できます。
- イベントは次のいずれかです。
  - CALL function-name。パラメータ値は、関数が指定されたパラメータで呼び出されたことを示します。
  - RETURN FROM function-name; 戻り値 = 関数は、指定された（最初の）値を返したことを示します。（追加の戻り値は報告されません）。
  - LINE line-number。変数値は、行が実行されたことを示し、変更された値を持つ変数をリストします。

各項目の意味は次のとおりです。

- 値が次の値です。
  - 小数点の有無にかかわらず、数値、
  - 二重引用符で囲み、前述のようにエスケープ文字で囲まれた文字列、
  - 真または偽のブール値
  - nil、
  - {[key1 = 値 1, key2][= 値 2],...} の形式のテーブルコンストラクタ。
- parameter-values is parameter1 = value1 ; parameter2 = value2 , ...
- variable-values is variable1 = value1 ; variable2 = value2 , ...

短縮ログメッセージの例:

```
1  >shell tail -f /var/log/ns.log | grep TRACE | more
2
3  ... NSEXTENSION Message 3035 0 : ”TRACE combine_headers CALL 30 : CALL
      COMBINE_HEADERS; self = ”User-Agent: curl/7.24.0 (amd64-portbld-
      freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nHost:
      10.217.24.7\r\nAccept: \*/\*\r\nH2: h2val1\r\nH1: abcd\r\nH2: h2val2
      \r\nH2: h2val3\r\n\r\n””
4
5  ... NSEXTENSION Message 3036 0 : ”TRACE combine_headers CALL 30 : LINE
      4; headers = {
6    }
7   ”
8
9  ... NSEXTENSION Message 3037 0 : ”TRACE combine_headers CALL 30 : LINE
      5; combined_headers = {
10   }
```



```
 ”
... NSEXTENSION Message 3038 0 : ”TRACE combine_headers CALL 30 : CALL gmatch”
... NSEXTENSION Message 3039 0 : ”TRACE combine_headers CALL 30 : RETURN FROM gmatch; return = function 0x2bee5a80”
... NSEXTENSION Message 3040 0 : ”TRACE combine_headers CALL 30 : CALL for iterator”
... NSEXTENSION Message 3041 0 : ”TRACE combine_headers CALL 30 : RETURN FROM for iterator; return = ” curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3””
... NSEXTENSION Message 3042 0 : ”TRACE combine_headers CALL 30 : LINE 9; name = ”User-Agent”; value = ” curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3””
... NSEXTENSION Message 3043 0 : ”TRACE combine_headers CALL 30 : LINE 10”
... NSEXTENSION Message 3044 0 : ”TRACE combine_headers CALL 30 : LINE 14; headers = {
 [”User-Agent”]={
 [1]=”User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
     OpenSSL/0.9.8y zlib/1.2.3” }
  }
 ”
. . .
... NSEXTENSION Message 3117 0 : ”TRACE combine_headers CALL 30 : CALL for iterator”
... NSEXTENSION Message 3118 0 : ”TRACE combine_headers CALL 30 : RETURN FROM for iterator; return = nil”
... NSEXTENSION Message 3119 0 : ”TRACE combine_headers CALL 30 : LINE 19”
... NSEXTENSION Message 3120 0 : ”TRACE combine_headers CALL 30 : CALL concat”
... NSEXTENSION Message 3121 0 : ”TRACE combine_headers CALL 30 :
```



```
    RETURN FROM concat; return = "User-Agent: curl/7.24.0 (amd64-portbld
    -freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
    nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3""
    ... NSEXTENSION Message 3122 0 : "TRACE combine_headers CALL 30 :
    LINE 25; result_str = "User-Agent: curl/7.24.0 (amd64-portbld-
    freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
    nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3\r\
    n\r\n""
42
43  ... NSEXTENSION Message 3123 0 : "TRACE combine_headers CALL 30 :
    RETURN FROM COMBINE_HEADERS; return = "User-Agent: curl/7.24.0 (
    amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r
    \nH1: abcd\r\nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1,
    h2val2, h2val3\r\n\r\n""
44  <!--NeedCopy-->
```

カスタムロギング

拡張関数に独自のロギングを追加することもできます。これを行うには、組み込みの ns.logger:*level ()* 関数を使用します。level** は、緊急、アラート、クリティカル、エラー、警告、通知、情報、デバッグです。パラメータは、C printf () 関数と同じです。フォーマット文字列と、フォーマット文字列で指定された% の値を指定する可変数の引数です。たとえば、COMBINE_HEADERS 関数に次のコードを追加して、呼び出しの結果を記録できます。

```
1  local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"
2
3  ns.logger:info("Result: %s", result_str)
4
5  return result_str
6  <!--NeedCopy-->
```

上記の関数は、上記の「拡張トレース」セクションの省略されたログメッセージの例に示されたサンプル入力について、次のメッセージを /var/log/ns.log に記録します。

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M
 H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4)libcurl
/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: */*^M H2: h2val1, h2val2,
h2val3^M ^M"
```


最適化

October 7, 2021

Citrix ADC 最適化機能は、クライアントとサーバー間のトランザクション時間を短縮し、帯域幅の消費を減らします。また、一部のタスクをオフロードし、他のタスクを効率化することで、サーバーのパフォーマンスを向上させます。

機能	説明
クライアントの Keep-Alive	単一のクライアント接続で複数の要求を処理します。クライアントは、サーバーへの要求ごとに新しい接続をネゴシエートする必要はありません。
HTTP 圧縮	サーバーから圧縮対応ブラウザに送信される HTTP 応答を圧縮します。応答が小さくなると、ダウンロード時間が短縮され、帯域幅が節約されます。
統合キャッシング	クライアント要求に対する応答を格納します。同じコンテンツに対する後続の要求は、元のサーバーに転送されるのではなく、Citrix ADC キャッシュから処理されます。
フロントエンドの最適化	クライアントブラウザに提供されるコンテンツを簡素化および最適化することにより、Web ページのロードおよびレンダリング時間を短縮します。注: NetScaler 10.5 以降でサポートされています。
コンテンツアクセラレータ	サーバー応答は、Citrix ByteMobile T2100 アプライアンスに保存します。注: NetScaler 10.1 以降でサポートされています。
SPDY (Speedy)	クライアントとサーバー間の SPDYGateway として機能し、サーバー上で SPDY を設定/アップグレードすることなく、SPDY サポートを提供します。注: NetScaler 10.1 以降でサポートされています。

クライアントのキープアライブ

October 7, 2021

クライアントキープアライブ機能を使用すると、単一の接続で複数のクライアント要求を送信できます。この機能は、トランザクション管理に役立ちます。アプライアンスでクライアントキープアライブモードが有効で、クライアント

要求に対するサーバ応答に Connection が含まれている場合：HTTP ヘッダーを閉じて、次のタスクを実行します。

- ヘッダー名の文字をシャッフルして、既存の Connection ヘッダー名の名前を変更します。
- Keep-Alive をヘッダーの値として新しい Connection: ヘッダーを追加します。

クライアントキープアライブモードを使用すると、Citrix ADC アプライアンスは同じソケット接続を使用して複数の要求と応答を処理できます。この機能は、サーバーがアプライアンスとの接続を閉じた後でも、クライアントとアプライアンスの間の接続（クライアント側接続）を開いたままにします。これにより、単一の接続を使用して複数のクライアント要求が可能になり、接続の開閉に関連するラウンドトリップが節約されます。クライアントのキープアライブは、SSL セッションで最も有益です。

クライアントのキープアライブは、次のシナリオで役立ちます。

- サーバーがクライアントのキープアライブをサポートしていない場合。
- サーバーはサポートしているが、サーバー上のアプリケーションはクライアントのキープアライブをサポートしていない場合。

> 注:
> クライアントのキープアライブは、HTTP および SSL トラフィックに適用できます。クライアントキープアライブは、すべてのトラフィックを処理するようにグローバルに構成できます。また、特定のサービスでアクティブ化することもできます。

クライアントのキープアライブ環境では、構成されたサービスがクライアントトラフィックをインターセプトし、クライアント要求がオリジンサーバーに送信されます。サーバーは応答を送信し、サーバーとアプライアンス間の接続を閉じます。サーバーの応答に「Connection: Close」ヘッダーが存在する場合、アプライアンスはクライアント側の応答でこのヘッダーを破損し、クライアント側の接続は開いたままになります。その結果、クライアントは次のリクエストのために新しい接続を開く必要がありません。代わりに、サーバーへの接続が再開されます。

> 注:
> サーバーが 2 つの「接続：閉じる」ヘッダーを送り返す場合、編集されるのは 1 つだけです。これにより、クライアントは接続が閉じられるまでオブジェクトが完全に配信されたとは想定しないため、オブジェクトのクライアントレンダリングに大幅な遅延が発生します。

クライアントのキープアライブを構成する

クライアントキープアライブは、デフォルトでは、Citrix ADC 上でグローバルかつサービスレベルの両方で無効になっています。したがって、必要なスコープで機能を有効にする必要があります。

> 注:
> クライアントのキープアライブをグローバルに有効にすると、サービスレベルで有効にするかどうかに関係なく、すべてのサービスで有効になります。また、以下を指定するためにいくつかの HTTP パラメーターを構成する必要があります。
>
> - 接続再利用プールに保持される HTTP 接続の最大数。

- 接続の多重化を有効にし、永続性を有効にします Etag。

注:
永続 ETag が有効になっている場合、ETag ヘッダーには、コンテンツを提供したサーバーに関する情報が含まれます。これにより、そのコンテンツに対するキャッシュ検証条件付き要求またはブラウザ要求が常に同じサーバーに到達することが保証されます。

Citrix ADC コマンドインターフェイスを使用してクライアントのキープアライブを構成する

コマンドプロンプトで、次の操作を行います。

1. Citrix ADC でクライアント Keep-Alive を有効にします。
   - グローバルレベルで- `enable ns mode cka`
   - サービスレベルで- `set service <name> -CKA YES`

   注:

   クライアントのキープアライブは、HTTP サービスおよび SSL サービスに対してだけ有効にできます。
2. 1 つ以上のサービスにバインドされている HTTP プロファイルに HTTP パラメーターを構成します。

```
set ns httpProfile <name> -maxReusePool <value> -conMultiplex
    ENABLED -persistentETag ENABLED
<!--NeedCopy-->
```

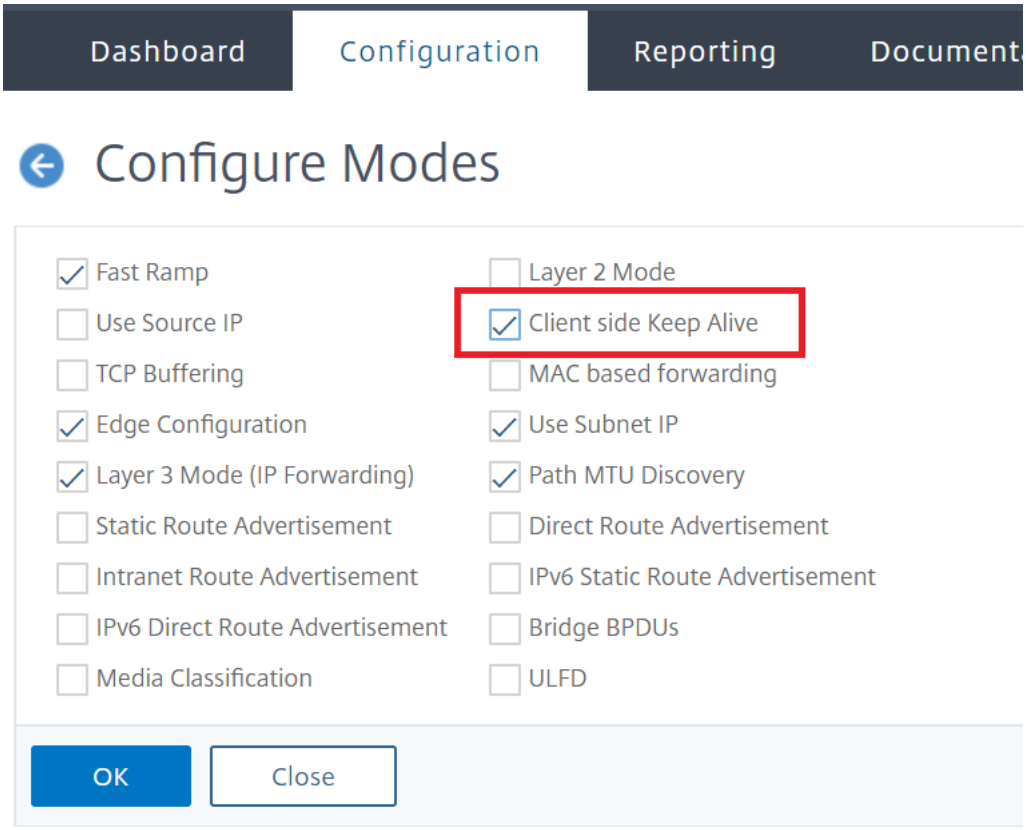
注:

これらのパラメータをnshttp_default _profile HTTPプロファイルに設定して、グローバルに使用できるようにします。

Citrix ADC GUI を使用してクライアントのキープアライブを構成する

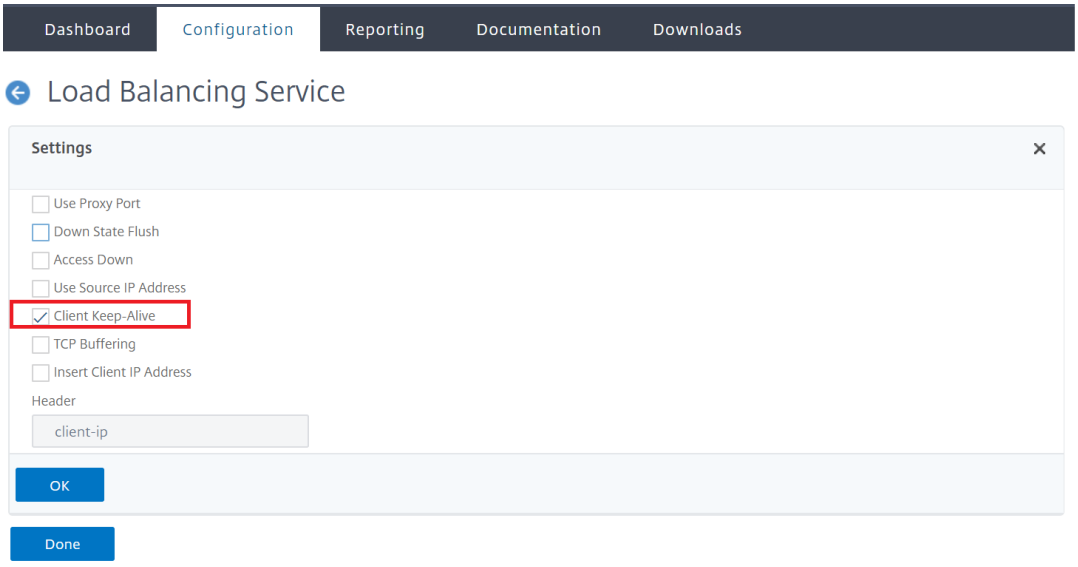
1. Citrix ADC でクライアント Keep-Alive を有効にします。
   - グローバルレベル

     [システム] **>** [設定] に移動し、[モードの構成] をクリックし、[クライアント側のキープアライブ] を選択します。



- サービスレベル

  [トラフィック管理] > [負荷分散] > [サービス] に移動し、必要なサービスを選択します。[設定] セクションで、[クライアントの **Keep-Alive**] チェックボックスをオンにします。



2. 1 つ以上のサービスにバインドされている HTTP プロファイルに必要な HTTP パラメーターを構成します。

3. [システム] > [プロファイル] に移動し、[**HTTP** プロファイル] タブで必要なプロファイルを選択し、必要な HTTP パラメータを更新します。

HTTP 圧縮

October 7, 2021

圧縮可能なコンテンツを持つ Web サイトの場合、HTTP 圧縮機能は可逆圧縮を実装し、サーバーから圧縮対応ブラウザに送信される HTTP 応答を圧縮することで、待ち時間、長いダウンロード時間、その他のネットワークパフォーマンスの問題を軽減します。処理負荷の高い圧縮タスクをサーバーから Citrix ADC アプライアンスにオフロードすることで、サーバーのパフォーマンスを向上させることができます。

次の表に、HTTP 圧縮機能の機能を示します。

機能	説明
圧縮率	圧縮率は応答内のファイルの種類によって異なりますが、常に重要であり、ネットワークを介して送信されるデータの量が著しく減少します。
ブラウザ認識	Citrix ADC は、圧縮データを圧縮対応のブラウザにのみ提供し、クライアントとサーバー間のトランザクション時間を短縮します。最新の Web ブラウザのほとんどは、HTTP 圧縮をサポートしています。
圧縮ブロック	組み込みアクションを適用することで、コンテンツフィルタを定義して圧縮を選択的にブロックできます。
圧縮キャッシュ	統合キャッシュ機能を有効にすると、同じコンテンツに対する後続のリクエストがローカルキャッシュから処理されるため、サーバーへの往復回数が削減され、トランザクション時間が短縮されます。
HTTPS のサポート	圧縮は、サーバーまたは Citrix ADC アプライアンスのいずれかで暗号化し、クライアントで復号化する必要があるコンテンツの量を減らすため、SSL 接続で役立ちます。

機能	説明
インテリジェントな応答フィルタリング	Citrix ADC 圧縮エンジンは、定義された圧縮パラメーターに基づいてサーバーの応答をインテリジェントにフィルタリングします。たとえば、圧縮エンジンはコンテンツ長がゼロの応答と圧縮された応答を検出し、圧縮しません。圧縮された応答の検出により、オリジンサイトは Citrix ADC 圧縮機能を備えたサーバーベースの圧縮を使用できます。
圧縮スイッチング	Citrix ADC アプライアンスは、圧縮対応クライアントからの要求を圧縮対応サーバーに透過的に送信します。これにより、これらのクライアントへの応答は圧縮され、他のクライアントへの応答は圧縮処理によって遅延されません。

HTTP 圧縮の仕組み

Citrix ADC では、静的データと動的に生成されたデータの両方を圧縮できます。GZIP または DEFLATE 圧縮アルゴリズムが適用されることで、無関係で反復的な情報がサーバー応答から削除され、より簡潔で効率的な形式で元の情報が表されます。この圧縮データは、クライアントのブラウザに送信され、ブラウザでサポートされているアルゴリズム（GZIP または DEFLATE）によって決定された圧縮解除されます。

Citrix ADC 圧縮では、静的コンテンツと動的コンテンツの扱いが異なります。

- 静的ファイルは一度だけ圧縮され、圧縮されたコピーはローカルメモリに保存されます。キャッシュされたファイルに対する後続のクライアント要求は、そのメモリから処理されます。
- 動的ページは、クライアントが要求するたびに動的に作成されます。

クライアントがサーバーに要求を送信すると、次のようになります。

1. クライアント要求が Citrix ADC に到着します。ADC はヘッダーを調べ、ブラウザがサポートする圧縮の種類に関する情報を格納します。
2. ADC は要求をサーバーに転送し、応答を受信します。
3. Citrix ADC 圧縮エンジンは、サーバー応答をポリシーと照合して、圧縮性を検査します。
4. 応答が圧縮アクションに関連付けられたポリシーと一致し、クライアントブラウザーがアクションで指定された圧縮アルゴリズムをサポートしている場合、Citrix ADC はそのアルゴリズムを適用し、圧縮された応答をクライアントブラウザーに送信します。
5. クライアントは、サポートされている圧縮アルゴリズムを適用して応答を解凍します。

HTTP 圧縮の構成

デフォルトでは、Citrix ADC では圧縮が無効になっています。この機能を設定する前に、有効にする必要があります。この機能が有効の場合、ADC は圧縮ポリシーで指定されたサーバ要求を圧縮します。

CLI を使用して HTTP 圧縮を有効にするには

圧縮は、HTTP サービスおよび SSL サービスに対してのみ有効にできます。すべての HTTP サービスおよび SSL サービスに適用されるように、グローバルに有効にすることも、特定のサービスに対してだけ有効にすることもできます。

コマンドプロンプトで、次のいずれかのコマンドを入力して、グローバルまたは特定のサービスに対して圧縮を有効にします。

- `enable ns feature cmp`
  または
- `set service \<name\> -CMP YES`

GUI を使用して圧縮を構成するには

次のいずれかを行います:

グローバルに圧縮を有効にするには、[システム] に移動します > [設定] で、[基本機能の構成] をクリックし、[HTTP 圧縮] を選択します。

特定のサービスの圧縮を有効にするには、トラフィック管理に移動します > 負荷分散 > サービスは、サービスを選択し、[編集] をクリックします。[設定] グループで鉛筆アイコンをクリックし、[圧縮] を有効にします。

圧縮アクションの設定

圧縮アクションは、要求または応答が、アクションが関連付けられているポリシーの規則（式）と一致した場合に実行するアクションを指定します。たとえば、特定のサーバーに送信される要求を識別する圧縮ポリシーを構成し、そのポリシーをサーバーの応答を圧縮するアクションに関連付けることができます。

次の 4 つの組み込み圧縮アクションがあります。

- 圧縮: GZIP アルゴリズムを使用して、GZIP または GZIP と DEFLATE の両方をサポートするブラウザからデータを圧縮します。DEFLATE アルゴリズムを使用して、DEFLATE アルゴリズムのみをサポートするブラウザからデータを圧縮します。ブラウザがどちらのアルゴリズムもサポートしていない場合、ブラウザの応答は圧縮されません。
- NOCOMPRESS: データを圧縮しません。
- GZIP: GZIP アルゴリズムを使用して、GZIP 圧縮をサポートするブラウザーのデータを圧縮します。ブラウザが GZIP アルゴリズムをサポートしていない場合、ブラウザの応答は圧縮されません。
- DEFLATE: DEFLATE アルゴリズムを使用して、DEFLATE アルゴリズムをサポートするブラウザーのデータを圧縮します。ブラウザが DEFLATE アルゴリズムをサポートしていない場合、ブラウザの応答は圧縮されません。アクションを作成したら、アクションを 1 つ以上の圧縮ポリシーに関連付けます。

コマンドプロンプトで次のコマンドを入力して、圧縮アクションを作成します。

```
add cmp action <name> <cmpType> [-addVaryHeader <addVaryHeader> -varyHeaderValue
<string>]
```

CLI を使用して圧縮ポリシーを設定するには

圧縮ポリシーにはルールが含まれています。ルールは、Citrix ADC アプライアンスが圧縮する必要があるトラフィックを識別できるようにする論理式です。

Citrix ADC はサーバーから HTTP 応答を受信すると、組み込みの圧縮ポリシーとカスタム圧縮ポリシーを評価して、応答を圧縮するかどうか、圧縮する場合は適用する圧縮の種類を決定します。ポリシーに割り当てられた優先順位によって、ポリシーがリクエストに対して照合される順序が決まります。

コマンドプロンプトで次のコマンドを入力して、圧縮ポリシーを作成します。

```
add cmp policy <name> -rule <expression> -resAction <string>
```

GUI を使用して圧縮アクションを作成するには

[最適化] > [**HTTP** 圧縮] > [アクション] に移動し、[追加] をクリックし、圧縮アクションを作成して、HTTP 応答で実行する圧縮のタイプを指定します。

圧縮ポリシーの設定

圧縮ポリシーにはルールが含まれています。ルールは、Citrix ADC アプライアンスが圧縮する必要があるトラフィックを識別できるようにする論理式です。

Citrix ADC はサーバーから HTTP 応答を受信すると、組み込みの圧縮ポリシーとカスタム圧縮ポリシーを評価して、応答を圧縮するかどうか、圧縮する場合は適用する圧縮の種類を決定します。ポリシーに割り当てられた優先順位によって、ポリシーがリクエストに対して照合される順序が決まります。

次の表に、組み込みの HTTP 圧縮ポリシーを示します。これらのポリシーは、圧縮を有効にすると、グローバルにアクティブになります。

組み込みのクラシックまたはデフォルトの構文ポリシー	説明
ns_nocmp_mozilla_47, ns_adv_nocmp_mozilla_	Mozilla 4.7 ブラウザからリクエストが送信されたときに CSS ファイルの圧縮を防ぎます。
cmp_mscss	Internet Explorer ブラウザから要求が送信されたときに、CSS ファイルを圧縮します。
cmp_msapp, ns_adv_cmp_msapp	次のアプリケーションによって生成されるファイルを圧縮します。:Microsoft Office Word、Microsoft Office Excel、Microsoft Office パワーポイント。
ns_cmp_content_type, ns_adv_cmp_content_type	応答に Content-Type ヘッダーが含まれており、テキストが含まれているときにデータを圧縮します。

組み込みのクラシックまたはデフォルトの構文ポリシー	説明
ns_nocmp_xml_ie, ns_adv_nocmp_xml_ie	Microsoft Internet Explorer ブラウザから要求が送信され、応答に Content-Type ヘッダーが含まれており、テキストまたは xml が含まれているときの圧縮を防止します。

圧縮ポリシーのバインド

圧縮ポリシーを有効にするには、Citrix ADC を通過するすべてのトラフィックに適用されるように、グローバルにバインドするか、特定の仮想サーバーにバインドする必要があります。これにより、宛先がその仮想サーバーの VIP アドレスの要求にのみポリシーが適用されます。

ポリシーをバインドするときは、ポリシーに優先度を割り当てます。優先順位によって、定義したポリシーが評価される順序が決まります。プライオリティは任意の正の整数に設定できます。

CLI を使用して圧縮ポリシーをバインドするには

コマンドプロンプトで、次のいずれかのコマンドを入力して、圧縮ポリシーをグローバルにバインドするか、特定の仮想サーバーにバインドします。

- `bind cmp global <policyName> [-priority <positive_integer>] [-state ( ENABLED|DISABLED)]...`
- `bind lb vserver <vserverName> -policyName <policyName> -priority < positive_integer>`。

圧縮ポリシーをバインドする仮想サーバごとに、このコマンドを繰り返します。

GUI を使用して圧縮ポリシーをバインドするには

次のいずれかを行います:

グローバルレベルで [最適化] > [**HTTP** 圧縮] > [ポリシー] に移動し、[ポリシーマネージャ] をクリックし、関連するバインドポイントと接続タイプ (要求/応答) を指定して、必要なポリシーをバインドします。

仮想サーバ・レベルで

負荷分散仮想サーバーの場合は、トラフィック管理 > 負荷分散 > 仮想サーバーに移動し、必要な仮想サーバーを選択し、[ポリシー] をクリックして、関連するポリシーをバインドします。

コンテンツスイッチング仮想サーバーの場合は、トラフィック管理に移動します > コンテンツの切り替え > 仮想サーバーで、必要な仮想サーバーを選択し、[ポリシー] をクリックして、関連するポリシーをバインドします。

最適なパフォーマンスを実現するためのグローバル圧縮パラメータの設定

多くのユーザーはグローバル圧縮パラメータのデフォルト値を受け入れますが、これらの設定をカスタマイズすることで、より効果的な圧縮を提供できる場合があります。

> 注:
> グローバル圧縮パラメータを設定した後は、アプライアンスを再起動する必要はありません。それらはすぐに新しいフローに適用されます。

次の表では、Citrix ADC で設定できる圧縮パラメータについて説明します。

圧縮パラメータ	説明
量子サイズ	サーバ応答を累積するために維持されるバッファのサイズ（KB)。バッファサイズがこの値を超えると、応答は圧縮されます。たとえば、量子サイズを 50 KB に設定すると、Citrix ADC はバッファのサイズが 50 KB を超えるとバッファの内容を圧縮します。最小値: 1。最大値:63488。デフォルトは 57344 です。
圧縮レベル	サーバの応答に適用する圧縮レベル。可能な値: 最高速度、最高圧縮、最適値。
HTTP 応答の最小サイズ	圧縮される HTTP 応答の最小サイズ（バイト単位)。このパラメータで指定された値より小さい応答は、圧縮されずに送信されます。
CPU 使用率の圧縮をバイパスする	Citrix ADC の CPU 使用率（比率）で、圧縮は行われません。デフォルトは 100 です。
ポリシーの種類 *	圧縮に使用されるポリシーのタイプ。使用可能な値: クラシック、デフォルト構文。デフォルト: クラシック。
サーバー側の圧縮を許可する	サーバーから Citrix ADC への圧縮データの送信を許可します。
プッシュパケットの圧縮	TCP PUSH フラグを持つパケットを受信すると、量子バッファがいっぱいになるのを待たずに、蓄積されたパケットをただちに圧縮します。
外部キャッシュ	応答メッセージが単一のユーザー向けであり、共有キャッシュまたはプロキシキャッシュによってキャッシュされてはならないことを示すプライベート応答ディレクティブを発行します。

GUI を使用して HTTP 圧縮を構成するには

次のいずれかを行います:

- 圧縮をグローバルに有効にするには、[システム] > [設定] に移動し、[基本機能の構成] をクリックして、[**HTTP** 圧縮] を選択します。

- 特定のサービスの圧縮を有効にするには、[トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを選択し、[編集] をクリックします。
- [設定] グループで鉛筆アイコンをクリックし、[圧縮] を有効にします。

GUI を使用して圧縮アクションを作成するには

[最適化] > [**HTTP** 圧縮] > [アクション] に移動し、[追加] をクリックし、圧縮アクションを作成して、HTTP 応答で実行する圧縮のタイプを指定します。

GUI を使用して圧縮ポリシーを作成するには

最適化に移動 > **HTTP** 圧縮 > ポリシーをクリックし、[追加] をクリックして、実行する条件と対応するアクションを指定して圧縮ポリシーを作成します。

圧縮構成の評価

圧縮統計情報は、ダッシュボードユーティリティまたは SNMP モニタで表示できます。ダッシュボード・ユーティリティは、サマリー統計と詳細な統計を表形式およびグラフィック形式で表示します。

オプションで、ポリシーベースの圧縮中にポリシーカウンターが増分する要求の数など、圧縮ポリシーの統計を表示することもできます。

> 注
>
> - 統計情報とグラフの詳細については、Citrix ADC アプライアンスのダッシュボードのヘルプを参照してください。
> - SNMP の詳細については、SNMP のトピックを参照してください。

CLI を使用して圧縮統計を表示するには

コマンドプロンプトで次のコマンドを入力して、圧縮の統計情報を表示します。

1. 圧縮統計情報のサマリーを表示します。

```
stat cmp
```

> 注**:**
>
> stat cmp policy コマンドは、既定の構文圧縮ポリシーの統計情報のみを表示します。

1. 圧縮ポリシーのヒットと詳細を表示するには

   ```
   show cmp policy <name>
   ```

2. 詳細な圧縮統計情報を表示するには

   ```
   stat cmp -detail
   ```

ダッシュボードを使用して圧縮統計を表示する手順は、次のとおりです。

ダッシュボード・ユーティリティでは、次のタイプの圧縮統計を表示できます。

- 圧縮統計のサマリーを表示するには、[圧縮] を選択します。

- プロトコルの種類別に詳細な圧縮統計を表示するには、[詳細] をクリックします。
- 圧縮機能によって処理された要求の割合を表示するには、「グラフィカル・ビュー」タブをクリックします。

SNMP を使用して圧縮統計情報を表示するには

SNMP ネットワーク管理アプリケーションを使用して、次の圧縮統計を表示できます。

- 圧縮リクエストの数 (OID:
- 送信された圧縮バイト数（OID：1.3.6.1.4.1.5951.1.1.50.2）
- 受信した圧縮可能なバイト数 (OID: 1.3.6.1.4.1.5951.1.1.50.3)
- 送信された圧縮性パケットの数 (OID: 1.3.6.1.4.1.5951.1.1.50.4)
- 受信した圧縮性パケットの数（OID：1.3.6.1.4.1.5951.1.50.5）
- 受信した圧縮性データおよび送信した圧縮性データの比率 (OID: 1.3.6.1.4.1.5951.4.1.1.50.6)
- 受信した合計データと、送信されたデータ合計の比率 (OID: 1.3.6.1.4.1.5951.1.1.50.7)

GUI を使用してより多くの圧縮統計を表示するには

1. HTTP 圧縮統計情報を表示するには、次の手順を実行します。

[最適化] > [**HTTP** 圧縮] に移動し、[統計] をクリックします。

1. 圧縮ポリシーの統計情報を表示する。

[最適化] > [**HTTP** 圧縮] > [ポリシー] に移動し、ポリシーを選択し、[統計] をクリックします。

1. 圧縮ポリシー・ラベルの統計情報を表示するには
2. [最適化] > [**HTTP** 圧縮] > [ポリシー] に移動し、ポリシーラベルを選択し、[統計] をクリックします。

HTTP 圧縮のオフロード

サーバー上で圧縮を実行すると、サーバーのパフォーマンスに影響を与える可能性があります。Web サーバーの前に配置され、HTTP 圧縮用に構成された Citrix ADC は、静的コンテンツと動的コンテンツの両方の圧縮をオフロードし、サーバーの CPU サイクルとリソースを節約します。

Web サーバーから圧縮をオフロードするには、次の 2 つの方法があります。

Web サーバー上の圧縮を無効にし、グローバルレベルで Citrix ADC 圧縮機能を有効にし、圧縮のためのサービスを構成します。

Web サーバーで圧縮機能を有効にしたまま、すべての HTTP クライアント要求から「Accept Encoding」ヘッダーを削除するように Citrix ADC アプライアンスを構成します。その後、サーバーは圧縮されていない応答を送信します。Citrix ADC は、クライアントに送信する前にサーバーの応答を圧縮します。

> 注**:**
> サーバーが自動的にすべての応答を圧縮する場合、2 番目のオプションは機能しません。Citrix ADC は、すでに圧縮されている応答を圧縮しようとしません。

Servercmp パラメーターを使用すると、Citrix ADC アプライアンスはオフロードの HTTP 圧縮を処理できます。デフォルトでは、サーバーが Citrix ADC アプライアンスに圧縮データを送信するために、このパラメータは ON に設定されています。HTTP 圧縮をオフロードするには、servercmp パラメータを OFF に設定する必要があります。コマンドプロンプトで、次のコマンドを入力します。

```
set service <service name> -CMP YES
```

圧縮を有効にする各サービスに対して、このコマンドを繰り返します。

```
show service <service name>
```

各サービスに対してこのコマンドを繰り返し、圧縮が有効になっていることを確認します。

```
Save config
```

```
set cmp parameter -serverCmp OFF
```

> 注:
>
> Servercmp パラメータがオンになっていて、アプライアンスがサーバーから圧縮された応答を受信した場合、アプライアンスはそれ以上データを圧縮しません。代わりに、圧縮された応答をクライアントに転送します。

統合されたキャッシング

October 7, 2021

統合キャッシュは、Citrix ADC アプライアンスのメモリ内ストレージを提供し、オリジンサーバーへの往復を必要とせずにユーザーに Web コンテンツを提供します。静的コンテンツの場合、統合キャッシュの初期設定はほとんど必要ありません。統合キャッシュ機能を有効にして基本セットアップ（たとえば、キャッシュが使用を許可されている Citrix ADC アプライアンスメモリの量を決定する）を実行すると、統合キャッシュは組み込みポリシーを使用して、次のような特定のタイプの静的コンテンツを保存および提供します。シンプルなウェブページと画像ファイル。統合キャッシュを構成して、Web サーバーおよびアプリケーションサーバーによってキャッシュ不可としてマークされた動的コンテンツ（データベースレコードや株価など）を保存および提供することもできます。

> 注:
>
> 統合キャッシュという用語は、AppCache と同じ意味で使用できます。機能の観点から、両方の用語は同じ意味であることに注意してください。

要求または応答が、組み込みポリシーまたは作成したポリシーで指定されたルール（論理式）と一致する場合。Citrix ADC アプライアンスは、ポリシーに関連付けられたアクションを実行します。デフォルトでは、すべてのポリシーはキャッシュされたオブジェクトをデフォルトのコンテンツグループに保存し、そこから取得します。さまざまなタイプのコンテンツ用に独自のコンテンツグループを作成できます。

アプライアンスがコンテンツグループ内のキャッシュされたオブジェクトを検索できるようにするには、セレクターを構成できます。セレクターは、キャッシュされたオブジェクトを式と照合します。または、コンテンツグループ内

のオブジェクトを検索するためのパラメーターを指定できます。Citrix が推奨するセレクターを使用する場合は、最初にセレクターを構成して、コンテンツグループを構成するときにセレクターを指定できるようにします。次に、追加するコンテンツグループを設定し、ポリシーを構成するときにそれらを使用できるようにします。初期構成を完了するには、各ポリシーをグローバルバインドポイントまたは仮想サーバーにバインドして、ポリシーバンクを作成します。または、他のポリシーバンクから呼び出すことができるラベルをバインドできます。

統合キャッシュは、有効期限が切れる前にキャッシュオブジェクトメソッドをプリロードすることで改善できます。キャッシュされたデータの処理を管理するために、応答に挿入されるキャッシュ関連のヘッダーを構成できます。統合キャッシュは、他のキャッシュサーバーのフォワードプロキシとしても機能します。

> 注:
>
> 統合キャッシュには、HTTP 要求と応答にある程度精通している必要があります。
> HTTP データの構造の詳細については、”<http://livehttpheaders.mozdev.org/>.”の *Live HTTP Headers* を参照してください。

統合キャッシュの仕組み

統合キャッシュは、Citrix ADC アプライアンスを通過する HTTP および SQL 要求を監視し、その要求を保存されたポリシーと比較します。結果に応じて、統合キャッシュ機能はキャッシュで応答を検索するか、要求をオリジンサーバーに転送します。HTTP リクエストの場合、統合キャッシュは、シングルバイト範囲およびマルチパートバイト範囲リクエストに応答して、キャッシュからの部分的なコンテンツとして機能します。

クライアントが圧縮されたコンテンツを受け入れる場合、キャッシュされたデータは圧縮されます。コンテンツグループの有効期限を設定したり、コンテンツグループのエントリを選択して期限切れにすることができます。

次の表に示すように、統合キャッシュから提供されるデータは要求であり、オリジンから提供されるデータはキャッシュミスです。

トランザクション・タイプ	仕様
キャッシュヒット	Citrix ADC アプライアンスがキャッシュから提供する応答：静的オブジェクト（画像ファイルや静的 Web ページなど）、200 OK ページ、203 非権限応答ページ、300 複数選択ページ、301 永久移動ページ、302 検出ページ、304 変更されていないページ、これらの応答は肯定応答として知られています。Citrix ADC アプライアンスでは、307 の一時リダイレクトページ、403 禁止ページ、404 ページが見つかりませんページ、410 ページという否定的な応答もキャッシュします。パフォーマンスをさらに向上させるために、より多くの種類のコンテンツをキャッシュするように Citrix ADC アプライアンスを構成できます。

トランザクション・タイプ	仕様
ストレージ可能キャッシュミス	保存可能なキャッシュミスの場合、Citrix ADC アプライアンスはオリジンサーバーからレスポンスをフェッチし、クライアントに送信する前にレスポンスをキャッシュに保存します。
保存不可能なキャッシュミス	保存不可能なキャッシュミスは、キャッシュには不適切です。既定では、次のステータスコードを含む応答は、格納不可能なキャッシュミスです。201、202、204、205、206 ステータスコード、403、404 および 410 を除くすべての 4xx コード、5xx ステータスコード

> 注:
> 動的キャッシュをアプリケーションインフラストラクチャと統合するには、NITRO API を使用してキャッシュコマンドをリモートで発行します。たとえば、データベーステーブルが更新されたときにキャッシュされた応答を期限切れにするトリガーを設定できます。

キャッシュされたレスポンスをオリジンサーバー上のデータと確実に同期させるには、有効期限方法を設定します。Citrix ADC アプライアンスは、期限切れの応答と一致する要求を受信すると、オリジナル・サーバーからの応答を更新します。

> 注:
> Citrix ADC アプライアンスと 1 つ以上のバックエンドサーバーで時刻を同期することをお勧めします。

動的キャッシュの仕組み

動的キャッシュは、パラメータ値ペア、文字列、文字列パターン、またはその他のデータに基づいて HTTP 要求と応答を評価します。たとえば、ユーザーがバグ報告アプリケーションで Bug 31231 を検索するとします。ブラウザは、ユーザーに代わって次の要求を送信します。

```
GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
    Template=view&TableId=1000

Host: mycompany.net

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
     Gecko/2008052906 Firefox/3.0

Accept: text/html,application/xhtml+xml,application/xml;q
    =0.9,\*/\*;q=0.8
```



```
8
9       Accept-Language: en-us,en;q=0.5
10  <!--NeedCopy-->
```

この例では、このバグ報告アプリケーションの GET リクエストには、常に次のパラメータが含まれます。

- IssuePage
- RecordID
- テンプレート
- TableId

GET リクエストはデータを更新または変更しないため、次のように、キャッシュポリシーおよびセレクタでこれらのパラメータを設定できます。

- 文字列 mybugreportingsystem および HTTP 要求の GET メソッドを検索するキャッシュポリシーを構成します。このポリシーは、バグのコンテンツグループに一致するリクエストを転送します。
- バグのコンテンツグループで、IssuePage、RecordID などのさまざまなパラメーターと値のペアに一致する `hit` セレクターを構成します。

> 注
>
> ブラウザは、1 つのユーザーアクションに基づいて複数の GET リクエストを送信できます。以下は、ユーザーがバグ ID に基づいてバグを検索したときにブラウザが発行する 3 つの個別の GET リクエストのシリーズです。

```
1       GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
            Template=view&TableId=1000
2
3       GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
            viewbtns&RecordId=31231&TableId=1000
4
5       GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
            viewbody&RecordId=31231&tableid=1000
6   <!--NeedCopy-->
```

これらの要求を満たすために、複数の応答がユーザーのブラウザーに送信され、ユーザーに表示される Web ページは応答のアセンブリです。

ユーザーがバグレポートを更新した場合、キャッシュ内の対応する応答をオリジンサーバーからのデータで更新する必要があります。バグ報告アプリケーションは、ユーザーがバグ報告を更新したときに HTTP POST リクエストを発行します。この例では、POST 要求がキャッシュ内の無効化をトリガーするように、以下を設定します。

- 文字列 mybugreportingsystem および POST HTTP リクエストメソッドを検索し、一致するリクエストをバグレポート用にコンテンツグループに誘導する、リクエスト時の無効化ポリシー。
- RecordID パラメータに基づいてキャッシュされたコンテンツを期限切れにするバグレポートのコンテンツグループの無効化セレクタ。このパラメーターはすべての応答に表示されるため、無効化セレクターはキャッシ

ュ内の関連するすべてのアイテムを期限切れにすることができます。

次の抜粋は、バグレポートのサンプルを更新する POST リクエストを示しています。

```
1     POST /mybugreportingsystem/mybugreport.dll?TransitionForm HTTP/1.1\
          r\n
2
3     User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
          Opera 7.23 [en]\r\n
4
5     Host: mybugreportingsystem\r\n
6
7     Cookie:ttSearch.134=%23options%3Afalse%23active%23owner%3Afalse%23
          unowned%3Afalse%23submitter%3Afalse%23incsub%3Atrue;
8
9     Cookie2: $Version=1\r\n
10
11    . . .
12
13    \r\n
14
15    ProjectId=2&RecordId=31231&TableId=1000&TransitionId=1&Action=
          Update&CopyProjectId=0&ReloadForm=0&State=&RecordLockId=49873+
          issues+in+HTTP&F43. . .
16 <!--NeedCopy-->
```

Citrix ADC アプライアンスは、この要求を受信すると、次の処理を行います。

- リクエストを無効化ポリシーと照合します。
- ポリシーで指定されているコンテンツグループを検索します。
- このコンテンツグループの無効化セレクタを適用し、RecordID=31231 に一致するすべての応答を期限切れにします。

ユーザーがこのバグレポートに対して新しい要求を発行すると、Citrix ADC アプライアンスは、レポートインスタンスに関連付けられているすべての応答の更新されたコピーを求めてオリジンサーバーに移動します。応答をコンテンツグループに保存し、ユーザーのブラウザに提供します。ブラウザはレポートを再構成して表示します。

統合キャッシュの構成

統合キャッシュを使用するには、ライセンスをインストールし、機能を有効にする必要があります。統合キャッシュを有効にすると、Citrix ADC® アプライアンスは、組み込みポリシーで指定された静的オブジェクトを自動的にキャッシュし、キャッシュの動作に関する統計を生成します。(組み込みのポリシーには、ポリシー名の初期位置にアンダースコアがあります)。

組み込みポリシーが自分の状況に適している場合でも、グローバル属性を変更することもできます。たとえば、統合キャッシュに割り当てられる Citrix ADC アプライアンスのメモリ容量を変更できます。

設定を変更する前にキャッシュ操作を確認する場合は、「キャッシュされたオブジェクトとキャッシュ統計の表示」を参照してください。

> 注:
>
> Citrix ADC キャッシュは、アプライアンスを再起動すると削除されるメモリ内ストアです。

統合キャッシュライセンスをインストールするには

- 統合キャッシュライセンスが必要です。
- Citrix からライセンスコードを取得し、コマンドラインインターフェイスにアクセスしてログインします。

コマンドラインインターフェイスで、ライセンスファイルを /nsconfig/license フォルダーにコピーします。

- 次のコマンドを使用して、Citrix ADC アプライアンスを再起動します。

```
reboot
```

統合キャッシュを有効にするには:
統合キャッシュを有効にすると、Citrix ADC アプライアンスはサーバー応答のキャッシュを開始します。ポリシーまたはコンテンツグループを構成していない場合、組み込みポリシーはキャッシュされたオブジェクトをデフォルトのコンテンツグループに保存します。

コマンドプロンプトで、次のいずれかのコマンドを入力して、統合キャッシュを有効または無効にします。

```
enable ns feature IC
```

キャッシュのグローバル属性を構成する

グローバル属性は、すべてのキャッシュされたデータに適用されます。ヘッダー挿入を介して、統合キャッシュに割り当てられる Citrix ADC メモリの量を指定できます。キャッシュされたオブジェクトを提供する必要があることを確認するための基準。キャッシュで許可される POST 本体の最大長、HTTP GET 要求のポリシー評価をバイパスするかどうか、およびポリシーを評価できない場合に実行するアクション。

キャッシュメモリの容量は、ハードウェアアプライアンスのメモリによってのみ制限されます。また、nCore Citrix ADC アプライアンスのパケットエンジン（すべての着信 TCP 要求の中央配信ハブ）は、nCore Citrix ADC アプライアンスの他のパケットエンジンによってキャッシュされたオブジェクトを認識します。

> 注:
>
> デフォルトのグローバルメモリ制限が 0 に設定され、統合キャッシュ（IC）機能が有効になっている場合、アプライアンスはオブジェクトをキャッシュしません。キャッシュの場合は、グローバルメモリ制限を明示的に設定する必要があります。ただし、「認証、承認、および監査パラメータの設定 enableStaticPageCaching」オプションを有効にすると、アプライアンスにいくつかのデフォルトメモリが構成されます。このメモリは大きなオブジェクトのキャッシュには不十分なため、IC には高いメモリ制限を割り当てる必要があります。これを実行するには、「set cache パラメータ —memLimit」コマンドを設定します。新しい設定は、設定を保存

> してアプライアンスを再起動した後にのみ適用されます。

オブジェクトのキャッシュ用に設定されたグローバルメモリ制限を変更できます。ただし、グローバルメモリ制限を既存の値よりも低い値（たとえば、10GB から 4GB）に更新すると、アプライアンスは引き続きメモリ制限を使用します。

これは、統合キャッシュ制限がある値に構成されている場合でも、実際に使用される制限が高くなる可能性があることを意味します。ただし、この過剰なメモリは、オブジェクトがキャッシュから削除されると解放されます。

show cache parameter コマンドの出力は、設定された値（メモリ使用量の制限）と実際に使用されている値（メモリ使用量の制限（アクティブな値））を示します。

コマンドプロンプトで入力します。

```
1     set cache parameter [-memLimit <MBytes>] [-via <string>] [-
          verifyUsing <criterion>] [-maxPostLen <positiveInteger>] [-
          prefetchMaxPending <positiveInteger>] [-enableBypass(YES|NO)] [-
          undefAction (NOCACHE|RESET)]
2 <!--NeedCopy-->
```

Citrix ADC GUI による統合キャッシュの有効化

[システム] **>** [設定] に移動し、[基本機能の設定] をクリックし、[統合キャッシュ] を選択します。

Citrix ADC GUI を使用してキャッシュのグローバル設定を構成する

[最適化] > [統合キャッシュ] に移動し、[キャッシュ設定の変更] をクリックして、キャッシュのグローバル設定を構成します。

統合キャッシュの組み込みコンテンツグループ、パターンセット、およびポリシーを設定する

Citrix ADC アプライアンスには、コンテンツのキャッシュに使用できる統合キャッシュ構成が組み込まれています。構成は、ctx_cg_poc というコンテンツグループ、ctx_file_extensions というパターンセット、および統合キャッシュポリシーのセットで構成されます。コンテンツグループ ctx_cg_poc では、500 KB 以下のオブジェクトのみがキャッシュされます。コンテンツは 86000 秒間キャッシュされ、コンテンツグループのメモリ制限は 512 MB です。パターンセットは、ファイルタイプマッチング用の一般的な拡張子のインデックス付き配列です。

次の表に、組み込みの統合キャッシュポリシーを示します。デフォルトでは、ポリシーはバインドポイントにバインドされません。Citrix ADC アプライアンスでポリシーに対してトラフィックを評価する場合は、ポリシーをバインドポイントにバインドする必要があります。ポリシーは、ctx_cg_poc コンテンツグループのオブジェクトをキャッシュします。

統合キャッシュポリシー名	ポリシールール
_cacheVPNStaticObjects	HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS_IN
_cacheTCPVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(“.css”)
_cacheOCVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(“.pdf”)
_cacheWFStaticObjects	HTTP.REQ.URL.ENDSWITH(“.js”)
mayNoCacheReq	HTTP.RES.HEADER(“Content-Type”).CONTAINS(“application/x-javascript”)
noCacheRest	TRUE

キャッシュ構成のフラッシュ

キャッシュグループ、キャッシュグループ、またはキャッシュオブジェクトロケーターをフラッシュできます。キャッシュオブジェクトをフラッシュするコマンドは次のとおりです。

コマンドプロンプトで入力します。

```
flush cache contentgroup all
```

例

```
1          0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hello
2          0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hi
3
4          Flush cache contentGroup all
5          done
6
7  `flush cache contentgroup <content group name>`
8  <!--NeedCopy-->
```

例**:**

```
1          0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hello
2          0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hi
3
```



```
        Flush cache ob -| 0x00000089bae000000004
        done

`flush cache object (-locator <positive_integer> | (-url <URL> (-host <
   string> [-port <port>] [-groupName <string>] [-httpMethod ( GET |
   POST )]))))`
<!--NeedCopy-->
```

例**:**

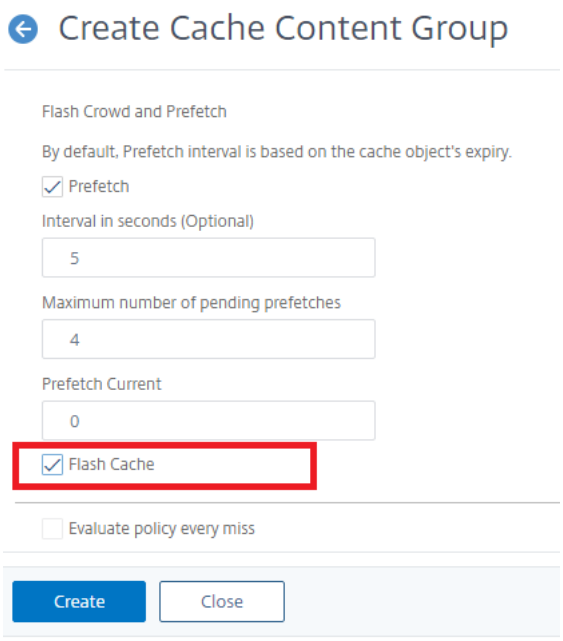
```
        0x00000089bae000000006 DEFAULT GET //1.1.1.1:80/html/index.html

        flush cache ob -URL /html/index.html -host 1.1.1.1 -groupName
           DEFAULT
        done
<!--NeedCopy-->
```

Citrix ADC GUI を使用してキャッシュ構成をフラッシュする

Citrix ADC GUI を使用してキャッシュフラッシュを構成する手順を完了します

1. **Optimization** > **Content Groups** に移動します。
2. [コンテンツグループ] の詳細ペインで、[追加] をクリックします。
3. [キャッシュコンテンツグループの作成] ページで、[その他] タブで次のパラメーターを設定します。
   a) キャッシュをフラッシュします。チェックボックスを選択して、キャッシュオブジェクトをフラッシュします。
4. **[**作成**]** して **[**閉じる**]** をクリックします。



さまざまなシナリオの統合キャッシュを構成する

次のセクションでは、さまざまなシナリオでの NetScaler アプライアンスでの統合キャッシュの構成について説明します。

NetScaler 9.2 リリース以降、統合キャッシュにはキャッシュ用のメモリが増えています。統合キャッシュメモリは、ハードウェアアプライアンスで使用可能なメモリによってのみ制限されます。使用可能なメモリの最大 50%を統合キャッシュ機能に割り当てることができます。

CLI を使用してキャッシュのメモリ割り当てを設定するには

コマンドプロンプトで入力します。

```
set cache parameter -memlimit <value>
```

> 注:
>
> 統合キャッシュのデフォルトのグローバルメモリ制限はゼロです。したがって、統合キャッシュ機能を有効にしても、グローバルメモリ制限が明示的に設定されるまで、NetScaler アプライアンスはオブジェクトをキャッシュしません。

次のセクションでは、さまざまなシナリオで統合キャッシュを構成する方法について説明します。

> 注:
>
> NetScaler アプライアンスのメモリ制限は、アプライアンスの起動時に識別されます。したがって、メモリ制限を変更する場合は、アプライアンスを再起動して、パケットエンジン全体に変更を適用する必要があります。

統合キャッシュが有効になっていて、キャッシュメモリ制限がゼロ以外に設定されている

アプライアンスを起動し、統合キャッシュ機能が有効になっていて、グローバルメモリ制限が正の数に設定されているシナリオを考えてみます。以前に設定したメモリは、ブートプロセス中に統合キャッシュ機能に割り当てられます。アプライアンスで使用可能なメモリに応じて、メモリ制限を別の値に変更することをお勧めします。

CLI を使用した構成

1. キャッシュパラメータを表示します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 MBytes
4                  Memory usage limit (active value): 500 MBytes
5                  Maximum value for Memory usage limit: 843 MBytes
6                  Via header: NS-CACHE-9.3: 18
7                  Verify cached object using: HOSTNAME_AND_IP
8                  Max POST body size to accumulate: 0 bytes
9                  Current outstanding prefetches: 0
10                 Max outstanding prefetches: 4294967295
11                 Treat NOCACHE policies as BYPASS policies: YES
12                 Global Undef Action: NOCACHE
13 <!--NeedCopy-->
```

1. ゼロ以外のメモリ制限を設定する

```
set cache parameter -memlimit 600
```

> 注:
>
> 上記のコマンドは、次の警告メッセージを表示します。警告: 新しい統合キャッシュのメモリ制限を使用するには、構成を保存して、**NetScaler** アプライアンスを再始動します。

1. 構成を保存します

```
save config
```

1. シェルプロンプトから、次のコマンドを実行して構成ファイルで確認します。

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. アプライアンスを再起動します


```
root@ns## reboot
```

1. メモリ制限の新しい値を確認します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 600 MBytes
4                  Memory usage limit (active value): 600 MBytes
5                  Maximum value for Memory usage limit: 843 MBytes
6                  Via header: NS-CACHE-9.3: 18
7                  Verify cached object using: HOSTNAME_AND_IP
8                  Max POST body size to accumulate: 0 bytes
9                  Current outstanding prefetches: 0
10                 Max outstanding prefetches: 4294967295
11                 Treat NOCACHE policies as BYPASS policies: YES
12                 Global Undef Action: NOCACHE
13 <!--NeedCopy-->
```

すべてのパケットエンジンが正常に起動すると、統合キャッシュ機能が構成済みのメモリをネゴシエートします。アプライアンスが構成済みのメモリを使用できない場合は、それに応じてメモリが割り当てられます。使用可能なメモリが割り当てたメモリよりも少ない場合、アプライアンスはより少ない数を推奨します。統合キャッシュ機能は、アクティブな値と同じものを使用します。

統合キャッシュが無効になり、キャッシュメモリ制限がゼロ以外に設定されている

このシナリオでは、アプライアンスを起動すると、統合キャッシュ機能が無効になり、グローバルメモリ制限が正の数に設定されます。したがって、ブートプロセス中に統合キャッシュにメモリが割り当てられることはありません。

CLI を使用した構成

1. キャッシュパラメータを表示します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 600 MBytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
```



```
10                  Treat NOCACHE policies as BYPASS policies: YES
11                  Global Undef Action: NOCACHE
12  <!--NeedCopy-->
```

1. 新しいメモリ制限を設定する

```
set cache parameter -memlimit 500
```

> 注:
>
> 上記のコマンドは、次の警告メッセージを表示します。警告：機能が有効になっていない **[IC]**。

1. 構成を保存します

```
save config
```

1. シェルプロンプトから、次のコマンドを実行して構成ファイルで確認します

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. メモリ制限の新しい値を確認します

```
1           > show cache parameter
2                   Integrated cache global configuration:
3                   Memory usage limit: 500 MBytes
4                   Maximum value for Memory usage limit: 843 MBytes
5                   Via header: NS-CACHE-9.3: 18
6                   Verify cached object using: HOSTNAME_AND_IP
7                   Max POST body size to accumulate: 0 bytes
8                   Current outstanding prefetches: 0
9                   Max outstanding prefetches: 4294967295
10                  Treat NOCACHE policies as BYPASS policies: YES
11                  Global Undef Action: NOCACHE
12  <!--NeedCopy-->
```

1. 統合キャッシュ機能を有効にする

```
enable ns feature IC
```

1. メモリ制限の新しい値を確認します

```
1           > show cache parameter
```



```
                Integrated cache global configuration:
                Memory usage limit: 500 Mbytes
                Memory usage limit (active value): 500 Mbytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3: 18
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> 注:
>
> 500MB のメモリが統合キャッシュ機能に割り当てられます。

1. 構成を保存して、アプライアンスの再起動時にメモリが機能に自動的に割り当てられるようにします。

統合キャッシュが有効になり、キャッシュメモリがゼロに設定されます

このシナリオでは、アプライアンスを起動すると、統合キャッシュ機能が有効になり、グローバルメモリ制限がゼロに設定されます。したがって、ブートプロセス中に統合キャッシュにメモリが割り当てられることはありません。

CLI を使用した構成

1. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns## cat ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

1. メモリ制限の値を確認します

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 0 Mbytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3: 18
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
```



```
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> 注:
>
> メモリ制限は 0MB に設定されており、統合キャッシュ機能にメモリは割り当てられていません。

1. 統合キャッシュ機能がオブジェクトを確実にキャッシュするようにメモリ制限を設定します

```
set cache parameter -memLimit 600
```

上記のコマンドを実行すると、アプライアンスは統合キャッシュ機能のメモリをネゴシエートし、使用可能なメモリが機能に割り当てられます。その結果、アプライアンスを再起動せずに、アプライアンスがオブジェクトをキャッシュします。

1. メモリ制限の値を確認します

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 600 Mbytes
                Memory usage limit (active value): 600 Mbytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3:
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> 注:
>
> 600MB のメモリが統合キャッシュ機能に割り当てられます。

1. 構成を保存します。アプライアンスの再起動時に、メモリが機能に自動的に割り当てられることを確認します。
2. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```


統合キャッシュが無効になり、キャッシュメモリがゼロに設定されます

このシナリオでは、アプライアンスを起動すると、統合キャッシュ機能が無効になり、グローバルメモリ制限がゼロに設定されます。したがって、ブートプロセス中に統合キャッシュにメモリが割り当てられることはありません。

CLI を使用した構成

1. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

1. メモリ制限の値を確認します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 0 Mbytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

> 注:
>
> メモリ制限は 0MB に設定されており、統合キャッシュ機能にメモリは割り当てられていません。また、キャッシュ構成コマンドを実行すると、「警告**:** 機能は有効ではありません **[IC]**」という警告メッセージが表示されます。

1. 統合キャッシュ機能を有効にする

```
enable ns feature IC
```

> 注:
>
> この段階で、統合キャッシュ機能を有効にすると、アプライアンスはその機能にメモリを割り当てません。その結果、オブジェクトはメモリにキャッシュされません。また、キャッシュ構成コマンドを実行すると、次の警告メッセージが表示されます。**IC** 用にメモリが設定されていません。**set cache parameter** コマンドを使用し

> て、メモリ制限を設定します。

1. 統合キャッシュ機能がオブジェクトを確実にキャッシュするようにメモリ制限を設定します

```
set cache parameter -memLimit 500
```

上記のコマンドを実行すると、アプライアンスは統合キャッシュ機能のメモリをネゴシエートし、使用可能なメモリが機能に割り当てられます。その結果、アプライアンスを再起動せずに、アプライアンスがオブジェクトをキャッシュします。

> 注:
>
> この機能を有効にしてメモリ制限を設定する順序は重要です。機能を有効にする前にメモリ制限を設定すると、「警告**:** 機能は有効ではありません **[IC]**」という警告メッセージが表示されます。

1. メモリ制限の値を確認します

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 Mbytes
4                  Memory usage limit (active value): 500 Mbytes
5                  Maximum value for Memory usage limit: 843 MBytes
6                  Via header: NS-CACHE-9.3:
7                  Verify cached object using: HOSTNAME_AND_IP
8                  Max POST body size to accumulate: 0 bytes
9                  Current outstanding prefetches: 0
10                 Max outstanding prefetches: 4294967295
11                 Treat NOCACHE policies as BYPASS policies: YES
12                 Global Undef Action: NOCACHE
13 <!--NeedCopy-->
```

> 注:
>
> 500MB のメモリが統合キャッシュ機能に割り当てられます。

1. 構成を保存します

```
save config
```

1. シェルプロンプトから ns.conf ファイルに設定されているメモリ制限を確認します

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. メモリ制限を変更する

```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```


セレクタと基本コンテンツグループの構成

October 7, 2021

セレクターを構成し、コンテンツグループに適用できます。1 つ以上のコンテンツグループにセレクターを追加する場合、セレクターをキャッシュ要求の識別に使用するか、無効化（期限切れ）するキャッシュオブジェクトの識別に使用するかを指定します。セレクタはオプションです。または、`hit` パラメーターと無効化パラメーターを使用するようにコンテンツグループを構成することもできます。ただし、セレクタを構成することをお勧めします。

セレクタを設定した後、または代わりにパラメータを使用することを決定したら、基本的なコンテンツグループを設定する準備が整います。基本コンテンツグループを作成したら、キャッシュからオブジェクトを期限切れにする方法を決定し、キャッシュの有効期限を設定する必要があります。「キャッシュパフォーマンスの向上」および「Cookie、ヘッダー、およびポーリングの設定」の説明に従って、キャッシュをさらに変更できますが、最初にキャッシュポリシーを構成する必要がある場合があります。

> 注
>
> コンテンツグループのパラメータとセレクタは要求時にのみ使用され、通常は MAY_CACHE または MAY_NOCACHE アクションを使用するポリシーに関連付けます。

セレクタの利点

セレクタは、コンテンツグループ内の特定のオブジェクトを検索するフィルタです。セレクタを構成しない場合、Citrix® ADC アプライアンスはコンテンツグループ内で完全に一致するものを探します。これにより、同じオブジェクトの複数のコピーが 1 つのコンテンツグループ内に存在する可能性があります。たとえば、セレクタを持たないコンテンツグループでは、ホスト 1. ドメイン.comypage.htm、ホスト 2. ドメイン.commypage.htm、およびホスト 3. ドメイン.commypage.htm の URL を保存する必要があります。これとは対照的に、セレクタは URL（mypage.html、式 http.req.url を使用）とドメイン（.com、式 http.req.hostname.domain を使用）のみに一致することができ、リクエストは同じ URL によって満たされます。

セレクタ式は、パラメータの簡単なマッチングを実行できます（たとえば、いくつかのクエリ文字列パラメータとその値に一致するオブジェクトを検索する場合など）。セレクター式では、ブール論理、算術演算、属性の組み合わせを使用して、オブジェクトを識別できます (たとえば、URL ステムのセグメント、クエリ文字列、POST リクエスト本体の文字列、HTTP ヘッダー内の文字列、Cookie など)。セレクタは、要求内の情報を分析するためのプログラム機能を実行することもできます。例えば、セレクタは、POST 本文内のテキストを抽出し、テキストをリストに変換し、リストから特定の項目を抽出することができます。

式および式で指定できる内容の詳細については、「ポリシーと式」を参照してください。

セレクタの代わりにパラメータを使用する

Citrix では、コンテンツグループでセレクターを使用することをお勧めしますが、代わりに `hit` パラメーターと無効化パラメーターを構成できます。たとえば、バグレポートのコンテンツグループに、BugID、Issuer、Assignee の

3 つの hit パラメータを設定するとします。リクエストに BugID=456 が含まれており、発行者は RohitV、担当者はロバートである場合、Citrix ADC アプライアンスはこれらのパラメーターと値のペアに一致する応答を提供できます。

コンテンツグループの無効化パラメーターは、キャッシュされたエントリを期限切れにします。たとえば、BugID が無効化パラメータであり、ユーザーが POST リクエストを発行してバグレポートを更新するとします。無効化ポリシーはこのコンテンツグループにリクエストを転送し、コンテンツグループの無効化パラメーターは、BugID 値と一致するすべてのキャッシュされた応答を期限切れにします。(次回ユーザーがこのレポートに対して GET リクエストを発行すると、キャッシュポリシーにより、Citrix ADC アプライアンスがレポートのキャッシュされたエントリをオリジンサーバーから更新できるようになります)。

同じパラメータを hit パラメータまたは無効化パラメータとして使用できることに注意してください。

コンテンツグループは、次の順序でリクエストパラメータを抽出します。

- URL クエリ
- ポスト本文
- クッキーのヘッダー

パラメータが最初に出現した後、リクエストで発生した場所に関係なく、後続のすべての出現は無視されます。たとえば、URL クエリと POST 本文の両方にパラメータが存在する場合、URL クエリの 1 つだけが考慮されます。

コンテンツグループに対してヒットおよび無効化パラメータを使用する場合は、コンテンツグループを設定するときにパラメータを設定します。

注: セレクタは柔軟性が高く、より多くの種類のデータに対応できるため、パラメータ化されたコンテンツグループではなくセレクタを使用することをお勧めします。

セレクタの設定

コンテンツグループは、ヒットセレクタを使用してキャッシュヒットを取得したり、無効セレクターを使用して期限切れのキャッシュオブジェクトを取得したり、オリジンサーバーから新しいオブジェクトをフェッチしたりできます。

セレクタには、高度な式と呼ばれる名前と論理式が含まれています。

高度な式の詳細については、「ポリシーと式」を参照してください。

セレクタを設定するには、セレクタに名前を割り当て、1 つ以上の式を入力します。ベストプラクティスとして、セレクタ式には URL ステムとホストを含める必要があります（省略する強い理由がない限り）。

CLI を使用してセレクタを設定するには

コマンドプロンプトで入力します。

```
add cache selector \<selectorName\> ( \<rule\> ... )
```

式を設定する方法については、「コマンドラインインターフェイスを使用してセレクタ式を設定するには」を参照してください。


```
>add cache selector product_selector "http.req.url.query.value("ProductId")" "http.req.url.query.value("BatchNum")" "http.req.url.query.value("depotLocation")"

> add cache selector batch_selector "http.req.url.query.value("ProductId")" "http.req.url.query.value("BatchId")" "http.req.url.query.value("depotLocation")"

> add cache selector product_id_selector "http.req.url.query.value("ProductId")"

> add cache selector batchnum_selector "http.req.url.query.value("BatchNum")" "http.req.url.query.value("depotLocation")"

> add cache selector batchid_selector "http.req.url.query.value("depotLocation")" "http.req.url.query.value("BatchId")"

<!--NeedCopy-->
```

GUI を使用してセレクタを設定するには

[最適化] > [統合キャッシュ] > [キャッシュセレクタ] に移動し、キャッシュセレクタを追加します。

コンテンツグループ

コンテンツグループは、レスポンスで提供できるキャッシュされたオブジェクトのコンテナです。統合キャッシュを初めて有効にすると、キャッシュ可能なオブジェクトは Default という名前のコンテンツグループに保存されます。固有のプロパティを持つコンテンツグループを作成できます。たとえば、画像データ、バグレポート、株価情報用に個別のコンテンツグループを定義し、株価情報コンテンツグループを他のグループよりも頻繁に更新するように設定できます。

コンテンツグループ全体、またはコンテンツグループ内の選択したエントリの有効期限を構成できます。

コンテンツグループのデータは、次のように静的または動的にすることができます。

- 静的コンテンツグループ。リクエストの URL ステムとホスト名、レスポンスの URL ステムとホスト名との完全一致を検索します。
- 動的コンテンツグループ。特定のパラメーターと値のペア、任意の文字列、または文字列パターンを含むオブジェクトを検索します。動的コンテンツグループは、頻繁に更新されるデータ（バグレポートや株価など）をキャッシュする場合に便利です。

コンテンツグループからのリクエストを処理する

1. ユーザーがバグレポートなどの項目の検索条件を入力し、HTML フォームの [検索] ボタンをクリックします。

2. ブラウザは、1 つ以上の HTTP GET 要求を発行します。これらのリクエストには、パラメータ（バグの所有者、バグ ID など）が含まれます。
3. Citrix ADC アプライアンスは要求を受信すると、一致するポリシーが検索され、これらの要求に一致するキャッシュポリシーが見つかると、要求がコンテンツグループに転送されます。
4. コンテンツグループは、セレクターで構成した基準に基づいて、コンテンツグループ内の適切なオブジェクトを探します。

たとえば、コンテンツグループは `NameField=username and BugID=ID`に一致する応答を取得できます。

1. 一致するオブジェクトが見つかった場合、Citrix ADC アプライアンスはそれらをユーザーのブラウザに提供し、そこで完全な応答（バグレポートなど）にアセンブルされます。

コンテンツグループ内のオブジェクトを無効にする

1. ユーザーがデータを変更します（たとえば、ユーザーがバグレポートを変更し、[Submit] ボタンをクリックした場合など）。
2. ブラウザは、このデータを 1 つ以上の HTTP 要求の形式で送信します。たとえば、バグの所有者とバグ ID に関する情報を含む複数の HTTP POST リクエストの形式でバグレポートを送信できます。
3. Citrix ADC アプライアンスは、無効ポリシーに対して要求を照合します。通常、これらのポリシーは HTTP POST メソッドを検出するように設定されます。
4. 要求が無効ポリシーと一致する場合、Citrix ADC アプライアンスは、このポリシーに関連付けられているコンテンツグループを検索し、無効化のための構成された条件に一致する応答を期限切れにします。

たとえば、無効化セレクターは `NameField=username and BugID=ID`に一致する応答を見つけることができます。

1. Citrix ADC アプライアンスは、次回これらの応答に対する GET 要求を受信すると、オリジンサーバーからリフレッシュされたバージョンを取得し、リフレッシュされた応答をキャッシュし、ユーザーのブラウザーにこれらの応答を提供します。そこで完全なバグレポートが作成されます。

基本コンテンツグループを設定する

デフォルトでは、すべてのキャッシュされたデータは、デフォルトのコンテンツグループに格納されます。より多くのコンテンツグループを構成し、これらのコンテンツグループを 1 つ以上のポリシーで指定できます。

静的コンテンツのコンテンツグループを構成できます。動的コンテンツのコンテンツグループを構成する必要があります。デフォルトグループを含め、任意のコンテンツグループの構成を変更できます。

コマンドラインインターフェイスを使用して基本的なコンテンツグループをセットアップするには

コマンドプロンプトで入力します。

```
add cache contentgroup <name> (-hitSelector <hitSelectorName> -invalSelector
 <invalidationSelectorName> | -hitParams <hitParamName> -invalParams<
invalidationParamName>)-type <type> [-relExpiry <sec> | -relExpiryMilliSec
<msec>] [-heurExpiryParam <positiveInteger>]
```



```
add cache contentgroup Products_Details -hitSelector product_selector -
invalSelector id_selector

add cache contentgroup bugrep -hitParams IssuePage RecordID Template
TableId -invalParams RecordID -relExpiry 864000
```

GUI を使用して基本的なコンテンツグループを設定するには

[最適化**] > [**統合キャッシュ**] > [**コンテンツグループ**]** に移動し、コンテンツグループを作成します。

キャッシュされたオブジェクトを期限切れまたはフラッシュする

応答に Expires ヘッダーまたは有効期限が設定された Cache-Control ヘッダー (Max-Age または Smax-Age) がない場合は、次のいずれかの方法を使用して、コンテンツグループ内のオブジェクトを期限切れにする必要があります。

- コンテンツグループの有効期限設定を構成して、オブジェクトを保持するかどうか、および保持期間を決定します。
- コンテンツグループの無効化ポリシーとアクションを構成します。詳細については、「キャッシュと無効化のポリシーの設定」を参照してください。
- コンテンツグループまたはコンテンツグループ内のオブジェクトを手動で期限切れにします。

キャッシュされたレスポンスの有効期限が切れると、Citrix ADC アプライアンスは、クライアントが次回レスポンスのリクエストを発行したときにレスポンスを更新します。デフォルトでは、キャッシュがいっぱいになると、Citrix ADC アプライアンスは最も最近使用されていない応答を最初に置き換えます。

次のリストは、コンテンツグループの設定を使用して、キャッシュされた応答を期限切れにする方法を示しています。通常、これらのメソッドはパーセントまたは秒単位で指定します。

- **[**手動**]:** コンテンツグループ内のすべての応答またはキャッシュ内のすべての応答を手動で無効にします。
- 応答ベース。肯定応答と否定応答の特定の有効期限間隔。応答ベースの有効期限は、Last-Modified ヘッダーが応答に欠落している場合にのみ考慮されます。
- ヒューリスティック有効期限。Last-Modified ヘッダーを持つ応答の場合、ヒューリスティック有効期限は、応答が変更されてからの時間のマウントを指定します（現在の時刻から Last-Modified 時間を差し引き、ヒューリスティック有効期限の値を掛けて計算されます）。たとえば、Last-Modified ヘッダーが応答が 2 時間前に更新されたことを示し、ヒューリスティック有効期限設定が 10% の場合、キャッシュされたオブジェクトは 0.2 時間後に期限切れになります。この方法は、頻繁に更新される応答をより頻繁に期限切れにする必要があることを前提としています。
- 絶対または相対。応答が毎日期限切れになる正確な（絶対）時間を指定します。HH:MM 形式、現地時間、または GMT。ローカル時刻は、すべてのタイムゾーンで動作しない場合があります。

相対有効期限は、キャッシュミスによってオリジンサーバーへのトリップが発生してから応答が期限切れになるまでの数秒またはミリ秒を指定します。相対的な有効期限をミリ秒単位で指定する場合は、10 の倍数を入力します。この形式の有効期限は、すべての肯定的な応答に対して機能します。レスポンス内の最終変更ヘッダー、失効ヘッダー、キャッシュ制御ヘッダーは無視されます。

絶対的および相対的な有効期限は、応答自体の有効期限情報を上書きします。

- ダウンロード時。[完全な応答を受信した後に期限切れ] オプションは、ダウンロード時に応答を期限切れにします。これは、株価など、頻繁に更新される応答に役立ちます。デフォルトでは、このオプションは無効になっています。

[Flash Cache] と [応答完了後に期限切れにする] の両方を有効にすると、動的アプリケーションのパフォーマンスが高速になります。両方のオプションを有効にすると、Citrix ADC アプライアンスは同時要求のブロックに対して 1 つの応答のみをフェッチします。

- ピン固定。デフォルトでは、キャッシュがいっぱいになると、Citrix ADC アプライアンスは最も最近使用されていない応答を最初に置き換えます。Citrix ADC アプライアンスは、ピン留めとしてマークされたコンテンツグループにはこの動作を適用しません。

コンテンツグループの有効期限設定を構成しない場合、グループ内のオブジェクトを期限切れにするためのその他のオプションは次のとおりです。

- コンテンツグループに適用される INVAL アクションを使用してポリシーを設定します。
- INVAL アクションを使用するポリシーを設定するときに、コンテンツグループの名前を入力します。

有効期限メソッドの適用方法

肯定応答と否定的な応答では、有効期限が異なります。正と負の応答については、以下の表「正と負の応答の 有効期限」で説明しています。

コンテンツグループに適用される有効期限方法を理解するための経験則を次に示します。

- オブジェクトを期限切れにするかどうかを決定するときに、Citrix ADC アプライアンスが応答ヘッダーを評価するかどうかを制御できます。
- 絶対的および相対的な有効期限により、Citrix ADC アプライアンスは応答ヘッダーを無視します（応答内の有効期限情報を上書きします）。
- ヒューリスティック有効期限の設定と「弱い陽性」と「弱い負」の有効期限（構成ユーティリティで デフォルト値）により、Citrix ADC アプライアンスは応答ヘッダーを調べます。これらの設定は、次のように連携して機能します。
  - Expires ヘッダーまたは Cache-Control ヘッダーの値は、これらのコンテンツグループ設定よりも優先されます。
  - Expires または Cache-Control ヘッダーがなく、Last-Modified ヘッダーがある肯定的な応答の場合、Citrix ADC アプライアンスはヒューリスティックの有効期限設定とヘッダー値を比較します。
  - Expires、Cache-Control、または Last-Modified ヘッダーがない肯定応答の場合、Citrix ADC アプライアンスは「弱い肯定」値を使用します。
  - Expires または Cache-Control ヘッダーがない否定応答の場合、Citrix ADC アプライアンスは「弱い否定」値を使用します。

次の表では、これらのメソッドがどのように適用されるかを説明します。

応答タイプ	有効期限ヘッダータイプ	コンテンツグループの設定	オブジェクトがキャッシュに残る期間
Positive	任意のヘッダー	他の設定なしでコンテンツを期限切れにする（relExpiry）	**Expire ContentAfter** 設定の値を使用します。
Positive	任意のヘッダー	他の設定なしでコンテンツを期限切れにする（absExpiry）	**Expire ContentAt** 設定の値から現在の日付を引きます。
Positive	任意のヘッダー	コンテンツの期限を過ぎる (relExpiry) と、コンテンツの期限を過ぎる (absExpiry)	コンテンツグループ設定には、2 つの値のうち小さい方を使用します。この表の前の行を参照してください。
Positive	Last-Modified（他のヘッダーも含む）	ヒューリスティック（heurExpiry パラメータ）とその他の設定	現在の日付から Last-Modified 日を減算し、ヒューリスティック有効期限の設定の値で結果を乗算し、100 で除算します。
Positive	Last-Modified（他のヘッダーも含む）	デフォルト（positive)(weakPosRel 有効期限）と他の設定なし	[既定 (positive) 有効期限設定] の値を使用します。
Positive	Expires または Cache-Control: Max-Age ヘッダーが存在する	最終変更されたヘッダーがない、ヒューリスティック（heurExpiry Param)、デフォルト（正）（weakPosRel Expiry)、またはその両方	有効期限または `Cache-Control:Max-Age` 日付から現在の日付を引きます。
Positive	キャッシングヘッダーなし	デフォルト（positive)(weakPosRel Expiry）およびその他の有効期限設定	[既定 (positive)] の設定値を使用します。

応答タイプ	有効期限ヘッダータイプ	コンテンツグループの設定	オブジェクトがキャッシュに残る期間
Positive	キャッシングヘッダーなし	ヒューリスティック（heurExpiry Param）が存在し、デフォルト（正）（weakPosRel Expiry）設定が存在しません。	Last-Modified ヘッダーが存在しない場合、応答はキャッシュされないか、すでに期限切れの状態でキャッシュされます。Last-Modified ヘッダーが存在する場合は、ヒューリスティック有効期限の値を使用します。
Negative	有効期限または Cache-Control:Max-Age	[コンテンツの期限を過ぎる] (relExpiry)、[期限切れ] (absExpiry)、またはその両方の設定	Expires ヘッダーの値から現在の日付を減算するか、キャッシュコントロール:Max-Age ヘッダーの値を使用します。
Negative	Expires または Cache-Control ヘッダーがない	[コンテンツの期限を過ぎる] (relExpiry)、[期限切れ] (absExpiry)、またはその両方の設定	応答がキャッシュされていないか、Already Expired 状態でキャッシュされています。
Negative	有効期限または Cache-Control:Max-Age	任意の設定	有効期限または Cache-Control:Max-Age 日付から現在の日付を引きます。
Negative	Expires および Cache-Control:Max-Age ヘッダーが存在しない	デフォルト (negativwe) (weakNegRel Expiry)	[既定 (negative)] の設定値を使用します。
Negative	Expires および Cache-Control:Max-Age ヘッダーが存在しない	デフォルト（negative）以外の設定（weakNegRel Expiry）	オブジェクトはキャッシュされていないか、既に期限切れの状態でキャッシュされています。

手動の方法でコンテンツグループを期限切れにする

コンテンツグループ内のすべてのエントリを手動で期限切れにすることができます。

コマンドラインインターフェイスを使用して、コンテンツグループ内のすべての応答を手動で期限切れにするにはコマンドプロンプトで入力します。


```
expire cache contentGroup <name>
```

GUI を使用してコンテンツグループ内のすべての応答を手動で期限切れにするには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択し、[無効] をクリックして、コンテンツグループ内のすべての応答を期限切れにします。

GUI を使用してキャッシュ内のすべての応答を手動で期限切れにするには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、[すべて無効] をクリックして、キャッシュ内のすべての応答を期限切れにします。

コンテンツグループの定期的な有効期限を構成する

コンテンツグループは、エントリの選択的または完全な有効期限を実行するように設定できます。有効期限間隔は、固定または相対的に指定できます。

コマンドラインインターフェイスを使用してコンテンツグループの有効期限を構成するには

コマンドプロンプトで入力します。

```
set cache contentgroup \<name> (-relExpiry|-relExpiryMilliSec|-absExpiry
|-absExpiryGMT| -heurExpiryParam|-weakPosRelExpiry|-weakNegRelExpiry| -
expireAtLastBye)\<expirationValue>
```

GUI を使用してコンテンツグループの有効期限を構成するには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択し、有効期限方法を指定します。

個々の回答を期限切れにする

応答の有効期限が切れると、Citrix ADC アプライアンスは、更新されたコピーをオリジナル・サーバーからフェッチします。ETag や Last-Modified ヘッダーなど、バリデーターがない応答は再検証できません。その結果、これらの応答をフラッシュすることは、それらの応答を期限切れにする場合と同じ効果があります。

静的データのコンテンツグループのキャッシュされた応答を期限切れにするには、保存された URL と一致する必要がある URL を指定できます。キャッシュされた応答がパラメータ化されたコンテンツグループの一部である場合は、グループ名と正確な URL ステムを指定する必要があります。ホスト名とポート番号は、キャッシュされた応答のホスト HTTP 要求ヘッダーと同じである必要があります。ポートが指定されていない場合は、ポート 80 が想定されます。

コマンドラインインターフェイスを使用してコンテンツグループ内の個々の応答を期限切れにするには

コマンドプロンプトで入力します。

```
expire cache object -url <URL> -host <hostName> [-port <port>] [-groupName<
contentGroupName>] [-httpMethod GET|POST]
```

CLI を使用してコンテンツグループの個々の応答を期限切れにするには

コマンドプロンプトで、次のコマンドを入力します。

```
expire cache object -locator <positiveInteger>
```

GUI を使用してキャッシュされた応答を期限切れにするには

最適化に移動 > 統合キャッシング > キャッシュされたオブジェクト、キャッシュされた応答を選択し、期限切れにします。

GUI を使用して応答を期限切れにするには

最適化に移動 > 統合キャッシング > キャッシュされたオブジェクトで、[検索] をクリックし、検索条件を設定して、必要なキャッシュされた応答を見つけて期限切れにします。

コンテンツグループ内の応答のフラッシュ

コンテンツグループ内のすべてのレスポンス、グループ内の一部のレスポンス、またはキャッシュ内のすべてのレスポンスを削除またはフラッシュできます。キャッシュされたレスポンスをフラッシュすると、新しいキャッシュされたレスポンスのメモリが解放されます。

> 注:
>
> 一度に複数のオブジェクトの応答をフラッシュするには、構成ユーティリティメソッドを使用します。コマンドラインインターフェイスでは、このオプションは提供されません。

コマンドラインインターフェイスを使用してコンテンツグループからの応答をフラッシュするには

コマンドプロンプトで入力します。

```
flush cache contentGroup <name> [-query <queryString> | [-selectorValue <
selectorExpressionIDList> -host <hostName>]]
```

GUI を使用してコンテンツグループからの応答をフラッシュするには

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動します。
2. 詳細ペインで、次のように応答をフラッシュします。
   - すべてのコンテンツグループ内のすべての回答をフラッシュするには、[すべて無効化] をクリックし、すべての回答をフラッシュします。
   - 特定のコンテンツグループ内の応答をフラッシュするには、コンテンツグループを選択し、[無効化] をクリックして、すべての応答をフラッシュします。

> 注意:
> このコンテンツ・グループがセレクタを使用する場合、「セレクタ値」テキスト・ボックスに文字列を入力し、「ホスト」テキスト・ボックスにホスト名を入力することで、応答を選択してフラッシュできます。次に、[フラッシュ] と **[OK]** をクリックします。Selector 値には、パラメータ化された無効化に使用される最大 2319 文字のクエリ文字列を指定できます。

コンテンツグループが無効化パラメータを使用している場合は、[クエリ] フィールドに文字列を入力することで、応答を選択的にフラッシュできます。

コンテンツグループが無効化パラメーターを使用し、ターゲットホストに属する Invalidate オブジェクトが構成されている場合は、[クエリ] フィールドと [ホスト] フィールドに文字列を入力します。

コマンドラインインターフェイスを使用してキャッシュされた応答をフラッシュするには

コマンドプロンプトで入力します。

```
flush cache object -locator <positiveInteger> | -url <URL> -host <hostName>
 [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET|POST]
```

GUI を使用してキャッシュされた応答をフラッシュするには

[最適化] > [統合キャッシュ] **>** [キャッシュされたオブジェクト] に移動し、キャッシュされたオブジェクトを選択してフラッシュします。

コンテンツグループの削除

コンテンツグループは、応答をキャッシュに保存するポリシーによって使用されていない場合は、削除できます。コンテンツグループがポリシーにバインドされている場合は、最初にポリシーを削除する必要があります。コンテンツグループを削除すると、そのグループに保存されているすべての応答が削除されます。

Default、BASEFILE、または Deltas グループを削除することはできません。Default グループは、他のコンテンツグループに属さないキャッシュされたレスポンスを格納します。

コマンドラインインターフェイスを使用してコンテンツグループを削除するには

コマンドプロンプトで入力します。

```
rm cache contentgroup <name>
```

GUI を使用してコンテンツグループを削除するには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択して削除します。

キャッシュと無効化のポリシーを構成する

October 7, 2021

ポリシーにより、統合キャッシュは、キャッシュまたはオリジンからの応答の提供を試みるかどうかを判断できます。NetScaler アプライアンスは、統合キャッシュ用の組み込みポリシーを提供し、さらに多くのポリシーを構成できます。ポリシーを設定するときは、アクションに関連付けます。アクションは、ポリシーが適用されるオブジェクトをキャッシュするか、オブジェクトを無効化（期限切れ）します。通常、キャッシュポリシーは GET リクエストと POST リクエストの情報に基づいています。通常、無効化ポリシーは、リクエスト内の POST メソッドの存在と他の情報に基づきます。キャッシュポリシーまたは無効化ポリシーでは、GET または POST リクエスト内の任意の情報を使用できます。

構成ユーティリティーの統合キャッシュの [ポリシー] ノードで、組み込みポリシーの一部を表示できます。組み込みのポリシー名は、アンダースコア（_）で始まります。

アクションによって、トラフィックがポリシーに一致した場合の NetScaler アプライアンスの動作が決まります。実行できるアクションは次のとおりです：

- キャッシングアクション。CACHE アクションに関連付けるポリシーは、応答をキャッシュに保存し、キャッシュから処理します。
- 無効化アクション。INVAL アクションに関連付けるポリシーは、キャッシュされた応答をただちに期限切れにし、オリジンサーバーから更新します。Web ベースのアプリケーションの場合、無効化ポリシーは POST リクエストを評価することがよくあります。
- 「キャッシュしない」アクション。NOCACHE アクションに関連付けるポリシーは、オブジェクトをキャッシュに保存しません。
- アクションを暫定的にキャッシュします。MAYCACHE または MAYNOCACHE アクションに関連付けるポリシーは、より多くのポリシー評価の結果によって異なります。

統合キャッシュは LOCK メソッドで指定されたオブジェクトを格納しませんが、`LOCK`リクエストの受信時にキャッシュされたオブジェクトを無効にすることができます。無効化ポリシーの場合のみ、式`http.req.method.eq(“lock”)`を使用して`LOCK`をメソッドとして指定できます。NetScaler アプライアンスはこのメソッド名を文字列として認識するため、`GET`および`POST`要求のポリシーとは異なり、LOCK メソッドを引用符で囲む必要があります。

ポリシーを作成したら、リクエストとレスポンスの全体的な処理における特定のポイントにポリシーをバインドします。ポリシーをバインドする前にポリシーを作成しますが、ポリシーを作成する前に、バインドポイントが処理の順序にどのように影響するかを理解しておく必要があります。

特定のバインドポイントにバインドされたポリシーは、ポリシーバンクを構成します。goto 式を使用して、ポリシーバンクでの実行順序を変更できます。また、他のポリシーバンクでポリシーを呼び出すことができます。また、ラベルを作成し、ラベルにポリシーをバインドすることもできます。このようなラベルは処理ポイントに関連付けられませんが、そのラベルにバインドされているポリシーは、他のポリシーバンクから呼び出すことができます。

統合キャッシュポリシーに関連付けるアクション

次の表では、統合キャッシュポリシーのアクションについて説明します。

操作（アクション）	仕様
CACHE	応答の有効期限が切れていない場合は、キャッシュからの応答を提供します。応答を元のサーバーからフェッチする必要がある場合、NetScaler アプライアンスは応答を提供する前に応答をキャッシュします。頻繁に更新され、アクセスされるデータもキャッシュできます。たとえば、株価は頻繁に更新されますが、複数のユーザーにすばやく提供できるようにキャッシュすることができます。必要に応じて、キャッシュされたデータは、ダウンロードされた直後に更新できます。CACHE アクションは、組み込みポリシーによって上書きできます。
NOCACHE	常にオリジンサーバーからの応答をフェッチし、応答を非保存としてマークします。通常、機密データまたはパーソナライズされたデータに対して NOCACHE ポリシーを構成します。
MAY_CACHE	この設定は要求時間ポリシーで使用され、応答時間ポリシーの評価を保留して、応答をコンテンツグループに保存できるようにします。以下が可能です：1。一致する応答時間ポリシーに CACHE アクションがあり、コンテンツグループが指定されていない場合、組み込みポリシーがこのポリシーを上書きしない限り、応答は Default グループに格納されます。一致する応答時間ポリシーに CACHE アクションがあり、要求時間ポリシーと同じコンテンツグループが指定されている場合、組み込みポリシーがこのポリシーを上書きしない限り、応答は名前付きコンテンツグループに保存されます。3. 一致する応答時間ポリシーに CACHE アクションがあり、要求時間ポリシーとは異なるコンテンツグループが指定されている場合は、NOCACHE アクションが適用されます。4. 一致する応答時間ポリシーに NOCACHE アクションがある場合は、NOCACHE アクションを実行します。5. 一致する応答時間ポリシーがない場合、組み込みポリシーがこのポリシーを上書きしない限り、CACHE アクションが適用されます。

操作（アクション）	仕様
MAY_NOCACHE	要求時間ポリシーの場合、この設定は応答のキャッシュを暫定的に防止します。応答時に、次のいずれかのアクションが実行されます。-要求に一致する応答時間ポリシーがない場合、最後のアクションは NOCACHE です。-一致する応答時間ポリシーに CACHE アクションが含まれている場合、組み込みポリシーがこのポリシーを上書きしない限り、最終的なアクションは CACHE です。-一致する応答時間ポリシーに NOCACHE アクションが含まれている場合、最後のアクションは NOCACHE です。-一致する応答時間ポリシーに CACHE アクションがあり、コンテンツグループが指定されていない場合、組み込みポリシーがこのポリシーを上書きしない限り、最終的なアクションは Default コンテンツグループの応答を CACHE することです。
INVAL	キャッシュされた応答を期限切れにします。ポリシーとコンテンツグループの構成方法に応じて、1 つ以上のコンテンツグループ内のすべての応答が期限切れになるか、またはコンテンツグループ内の選択したオブジェクトが期限切れになります。注意:INVAL アクションは、リクエスト時間ポリシーでのみ指定できます。

ポリシーのバインドポイント

ポリシーは、次のいずれかのバインドポイントにバインドできます。

- グローバルポリシーバンク。これらは、「ポリシー評価の順序」で説明されているように、リクエスト時間のデフォルト、リクエスト時間のオーバーライド、応答時間のデフォルト、および応答時間オーバーライドポリシーバンクです。
- 仮想サーバ。仮想サーバにバインドするポリシーは、「ポリシー評価の順序」で説明されているように、グローバルオーバーライドポリシーの後およびグローバルデフォルトポリシーの前に処理されます。ポリシーを仮想サーバーにバインドするときは、ポリシーを要求時または応答時の処理にバインドします。
- アドホックポリシーラベル。ポリシーラベルは、ポリシーバンクに割り当てられた名前です。統合キャッシュには、グローバルラベルに加えて、次の 2 つの組み込みカスタムポリシーラベルがあります。
  - _reqBuiltinDefaults. このポリシーラベルは、デフォルトでは、要求時のデフォルトポリシーバンクから呼び出されます。
  - _resBuiltinDefaults. このポリシーラベルは、デフォルトでは、応答時間のデフォルトポリシーバンク

から呼び出されます。

新しいポリシーラベルを定義することもできます。ユーザー定義のポリシーラベルにバインドされたポリシーは、いずれかの組み込みバインドポイントのポリシーバンク内から呼び出す必要があります。

> 重要:
> INVAL アクションを使用して、ポリシーを要求時オーバーライドまたは応答時オーバーライドバインド・ポイントにバインドする必要があります。ポリシーを削除するには、まずポリシーをバインド解除する必要があります。

ポリシー評価の順序

高度なポリシーを有効にするには、NetScaler アプライアンスのトラフィック処理中のある時点でポリシーが呼び出されるようにする必要があります。呼び出し時間を指定するには、ポリシーをバインドポイントに関連付けます。以下は、評価順にリストされたバインドポイントです。

- リクエスト時のオーバーライド。要求が要求時オーバーライドポリシーに一致する場合、デフォルトでは要求時ポリシーの評価が終了し、NetScaler アプライアンスは一致するポリシーに関連付けられているアクションを保存します。
- 要求時の負荷分散仮想サーバ。すべての要求時間上書きポリシーの評価後にポリシー評価を完了できない場合、NetScaler アプライアンスは、負荷分散仮想サーバーにバインドされた要求時間ポリシーを処理します。リクエストがこれらのポリシーのいずれかに一致する場合、評価は終了し、NetScaler アプライアンスは一致するポリシーに関連付けられているアクションを保存します。
- 要求時のコンテンツスイッチング仮想サーバー。このバインドポイントにバインドされているポリシーは、負荷分散仮想サーバーにバインドされている要求時ポリシーの後に評価されます。
- リクエスト時のデフォルト。すべての要求時の仮想サーバー固有のポリシーが評価された後、ポリシーの評価を完了できない場合、NetScaler アプライアンスは要求時のデフォルトポリシーを処理します。要求が要求時のデフォルトポリシーと一致する場合、デフォルトでは要求時ポリシーの評価が終了し、NetScaler アプライアンスは一致するポリシーに関連付けられているアクションを保存します。
- 応答時間のオーバーライド。要求時間上書きポリシー評価に似ています。
- 応答時間の負荷分散仮想サーバ。要求時の仮想サーバポリシーの評価に似ています。
- 応答時間のコンテンツスイッチング仮想サーバー。要求時の仮想サーバポリシーの評価に似ています。
- 応答時間のデフォルト。要求時のデフォルトポリシー評価に似ています。

各バインドポイントに複数のポリシーを関連付けることができます。バインドポイントに関連付けられたポリシーの評価順序を制御するには、プライオリティレベルを設定します。他のフロー制御情報がない場合、ポリシーは、プライオリティレベルに従って評価されます。プライオリティレベルは、最も低い数値から開始します。

> 注:
> POST データまたは Cookie ヘッダーのリクエスト時間ポリシーは、統合キャッシュ内の組み込みのリクエスト時ポリシーが POST リクエストのNOCACHEアクションと Cookie を使用したリクエストに対するMAY_NOCACHEアクションを返すため、リクエスト時のオーバーライド評価時に呼び出す必要があります。

> パラメータ化されたコンテンツグループを指す要求時ポリシーに、MAY_CACHEまたはMAY_NOCACHEアクションを関連付けます。応答時間ポリシーによって、トランザクションがキャッシュに格納されるかどうかが決まります。

統合キャッシュのポリシーを構成する

組み込みポリシーで処理できないデータを処理するには、新しいポリシーを構成します。キャッシュ、キャッシュの発生の防止、およびキャッシュされたデータの無効化には、個別のポリシーを構成します。統合キャッシュのポリシーの主なコンポーネントは次のとおりです。

- ルール:HTTP リクエストまたはレスポンスを評価する論理式。
- アクション: ポリシーとアクションを関連付けて、ポリシー規則に一致するリクエストまたはレスポンスの処理を決定します。

コンテンツグループ: ポリシーを 1 つ以上のコンテンツグループに関連付けて、アクションを実行する場所を特定します。

コマンドラインインターフェイスを使用してキャッシュのポリシーを構成するには

コマンドプロンプトで入力します。

```
add cache policy <policyName> -rule <expression> -actionCACHE|MAY_CACHE
|NOCACHE|MAY_NOCACHE [-storeInGroup <contentGroupName>] [-undefAction
NOCACHE|RESET]
> add cache policy image_cache -rule ”http.req.url.contains(\”jpg\”)|| http
.req.url.contains(\”jpeg\”)”-action CACHE -storeingroup myImages_group -
undefaction NOCACHE

> add cache policy bugReportPolicy -rule ”http.req.url.query.contains(\”
IssuePage\”)”-action CACHE -storeInGroup bugReportGroup

> add cache policy my_form_policy -rule ”http.req.header(\”Host\”)contains
(\”my.company.com\”)&& http.req.method.eq(\”GET\”)&& http.req.url.query.
contains(\”v=7\”)”-action CACHE -storeInGroup my_form_event

> add cache policy viewproducts_policy -rule ”http.req.url.contains(\”
viewproducts.aspx\”)”-action CACHE -storeInGroup Product_Details
```

コマンドラインインターフェイスを使用して無効化ポリシーを構成するには

コマンドプロンプトで入力します。

```
1 add cache policy <policyName> -rule <expression> -action INVAL [-
    invalObjects ”<contentGroupName1>[,<selectorName1>”]. . .]] | [-
    invalGroup <contentGroupName1>[, <contentGroupName2>. . .]] [-
    undefAction NOCACHE|RESET]
```



```
<!--NeedCopy-->
```

```
> add cache policy invalidation_events_policy -rule ”http.req.header(”Host”)contains(”my.company.com”) && http.req.method.eq(”GET”) && http.req.url.query.contains(”v=8”) -action INVAL -invalObjects my_form_event -undefaction NOCACHE
<!--NeedCopy-->
```

```
> add cache policy inval_all -rule ”http.req.method.eq(”POST”) && http.req.url.contains(”jpeg”)” -action INVAL -invalGroups myImages_group myApps_group PDF_group
<!--NeedCopy-->
```

```
> add cache policy bugReportInvalidationPolicy -rule ”http.req.url.query.contains(”TransitionForm”)” -action INVAL -invalObjects bugReport`
`> add cache policy editproducts_policy  - rule ”http.req.url.contains(”editproducts.aspx”)”  - action INVAL -invalObjects ”Product_Details, batchnum_sel” ”Products_In_Depots,batchid_sel”
<!--NeedCopy-->
```

GUI を使用してキャッシュまたは無効化のポリシーを構成するには

[最適化] > [統合キャッシュ] > [ポリシー] に移動し、新しいポリシーを作成します。

統合キャッシュポリシーのグローバルバインド

ポリシーをグローバルにバインドすると、NetScaler アプライアンス上のすべての仮想サーバーでポリシーを使用できます。

コマンドラインインターフェイスを使用して統合キャッシュポリシーをグローバルにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind cache global <policy> -priority <positiveInteger> [-typeREQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] [-gotoPriorityExpression <expression>] [-invoke <labelType> <labelName>]
```



```
<!--NeedCopy-->
```

```
> bind cache global myCachePolicy -priority 100 -type req_default
<!--NeedCopy-->
```

> 注:
>
> type 引数は、以前のバージョンの NetScaler アプライアンスを使用して定義したポリシーとの下位互換性を維持するために、グローバルにバインドされたポリシーの場合はオプションです。タイプを省略すると、ポリシールールが応答時間式か要求時間式かに応じて、ポリシーは REQ_DEFAULT または RES_DEFAULT にバインドされます。ルールに要求時間と応答時間のパラメータの両方が含まれている場合は、RES_DEFAULT にバインドされます。以下は、型を省略するバインディングの例です

次に、型を省略するバインディングの例を示します。

```
> bind cache global myCache Policy 200
```

構成ユーティリティを使用して統合キャッシュポリシーをグローバルにバインドするには

[最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイントと接続タイプ (要求/応答) を指定して、ポリシーをバインドします。

統合キャッシュポリシーを仮想サーバーにバインドする

ポリシーを仮想サーバにバインドすると、ポリシーに一致し、関連する仮想サーバを通過する要求と応答に対してのみ使用できます。

GUI を使用する場合、仮想サーバの設定ダイアログボックスを使用してポリシーをバインドできます。これにより、この仮想サーバーにバインドされているすべての Citrix ADC モジュールのすべてのポリシーを表示できます。統合キャッシュの PolicyManager 構成ダイアログを使用することもできます。これにより、仮想サーバーにバインドされている統合キャッシュポリシーのみを表示できます。

コマンドラインインターフェイスを使用して統合キャッシュポリシーを仮想サーバーにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind lb vserver <name>@ -policyName <policyName> -priority <
    positiveInteger> -type(REQUEST|RESPONSE)
<!--NeedCopy-->
```



```
bind cs vserver <name>@ -policyName <policyName> -priority <
    positiveInteger> -type(REQUEST|RESPONSE)
<!--NeedCopy-->
```

構成ユーティリティを使用して統合キャッシュポリシーを仮想サーバーにバインドするには (仮想サーバーメソッド)

- CS 仮想サーバー-トラフィック 管理に移動します > コンテンツの切り替え > 仮想サーバー、仮想サーバーを選択し、関連するキャッシュポリシーをバインドします。
- LB 仮想サーバー - Traffic **Management > Load Balancing > Virtual Servers**, に移動して仮想サーバーを選択し、関連キャッシュポリシーをバインドします。

GUI を使用して統合キャッシュポリシーを仮想サーバにバインドするには（Policy Manager 方式)。

[最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイントと接続タイプを指定してキャッシュポリシーをバインドします。

> 注:
> 適切なバインドポイントを選択すると、キャッシュポリシーを負荷分散仮想サーバーとコンテンツスイッチング仮想サーバーの両方にバインドできます。

ファイルの圧縮および非圧縮バージョンをキャッシュする方法

デフォルトでは、圧縮を処理できるクライアントは、圧縮されていない応答または圧縮された応答を gzip、deflate、compress、および pack200-gzip 形式で提供できます。クライアントが圧縮を処理する場合、`Accept-Encoding:compression` 形式のヘッダーがリクエストで送信されます。クライアントによって受け入れられる圧縮タイプは、キャッシュされたオブジェクトの圧縮タイプと一致する必要があります。たとえば、`Accept-Encoding:deflate` ヘッダーのあるリクエストに応答して `cached.gzip` ファイルを提供することはできません。

圧縮を処理できないクライアントは、キャッシュされた応答が圧縮されている場合、キャッシュミスを処理します。

動的キャッシュの場合、同じデータの圧縮データ用と非圧縮バージョン用の 2 つのコンテンツグループを構成する必要があります。次に、圧縮を処理できないクライアントにキャッシュから非圧縮ファイルを提供し、圧縮を処理できる同じファイルの圧縮バージョンをクライアントに提供するためのセレクタ、コンテンツグループ、およびポリシーを設定する例を示します。

```
add cache selector uncompressed_response_selector http.req.url ”http.req.
header(\”Host\”)”

add cache contentGroup uncompressed_group -hitSelector uncompressed_responst_selector
 -invalSelector uncomp_resp_sel

add cache policy cache_uncompressed -rule ”HTTP.REQ.URL.CONTAINS(\”xyz\”)&&
 !HTTP.REQ.HEADER(\”Accept-Encoding\”).EXISTS”-action CACHE -storeInGroup
uncompressed_group
```



```
bind cache global cache_uncompressed -priority 100 -gotoPriorityExpression
END -type REQ_OVERRIDE

add cache selector compressed_response_selector HTTP.REQ.URL ”HTTP.REQ.
HEADER(\”Host\”)””HTTP.REQ.HEADER(\”Accept-Encoding\”)”

add cache contentGroup compressed_group -hitSelector compressed_response_selector


add cache policy cache_compressed -rule ”HTTP.REQ.URL.CONTAINS(\”xyz\”)&&
 HTTP.REQ.HEADER(\”Accept-Encoding\”).EXISTS”-action CACHE -storeInGroup
compressed_group

bind cache global cache_compressed -priority 200 -gotoPriorityExpression
END -type REQ_OVERRIDE
```

キャッシュ用のポリシーバンクの構成

特定のバインドポイントに関連付けられているすべてのポリシーは、まとめてポリシーバンクと呼ばれます。銀行のポリシーの優先度レベルを構成することに加えて、Goto 式を構成することにより、銀行の評価の順序を変更できます。現在のポリシーバンク内から外部ポリシーバンクを呼び出すことによって、評価順序をさらに変更できます。また、独自のラベルを割り当てる新しいポリシーバンクを構成することもできます。このようなポリシーバンクは、処理サイクルのどの時点にも拘束されないため、他のポリシーバンク内からのみ呼び出すことができます。便宜上、組み込みのバインドポイントに対応しないラベルを持つポリシーバンクをポリシーラベルと呼びます。

「ポリシーのバインド」で説明されているように、ポリシーをバインドしてプライオリティレベルを割り当てることによってポリシー評価の順序を制御するだけでなく、Goto 式を設定することで、ポリシーバンク内のフローを確立できます。Goto 式は、優先度レベルによって決定されるフローを上書きします。また、現在のバンクのエントリを評価した後に外部ポリシーバンクを呼び出すことによって、評価フローを制御することもできます。評価の完了後、評価は常に現在のバンクに戻ります。

次の表は、ポリシーバンクで評価を制御するためのエントリをまとめたものです。

属性	**Specifies**
Name	ポリシーの名前。ポリシーを評価せずに別のポリシーバンクを呼び出す場合は、キーワード NOPOLICY。ポリシーバンクでは NOPOLICY を複数回指定できますが、名前付きポリシーは一度しか指定できません。
優先度	整数。整数値が小さいほど、優先度が高くなります。

属性	**Specifies**
Goto 式	次に評価するポリシーまたはポリシーバンクを決定します。次のいずれかの値を指定できます。1. NEXT: 次に高い優先度を持つポリシーに移動します。2. END: 評価を停止します。3. USE_INVOCATION_RESULT: このエントリが別のポリシー・バンクを呼び出す場合に適用されます。呼び出されたバンクの最後の Goto の値が END の場合、評価は停止します。最後の Goto が END 以外のものである場合、現在のポリシーバンクは NEXT を実行します。4. 正の数：評価される次のポリシーの優先度番号。5. 数値式：評価される次のポリシーの優先順位番号を生成する式。Goto は、ポリシーバンクでのみ進めることができます。Goto 式を省略することは、END を指定することと同じです。
呼び出しタイプ	ポリシーバンクタイプを指定します。値は次のいずれかになります-1。仮想サーバーの要求：仮想サーバーに関連付けられている要求時ポリシーを呼び出します。2.2。応答仮想サーバー：仮想サーバーに関連付けられている応答時間ポリシーを呼び出します。3.3。ポリシーラベル: バンクのポリシーラベルで識別される、別のポリシーバンクを起動します。
呼び出し名	［呼び出しタイプ］に指定した値に応じて、仮想サーバまたはポリシー・ラベルの名前。

統合キャッシュには 2 つの組み込みポリシーラベルがあり、さらに多くのポリシーラベルを設定できます。

_reqBuiltInDefaults：このポリシーラベルは、リクエスト時のデフォルトのバインドポイントから呼び出されます。
_resBuiltInDefaults：このポリシーラベルは、応答時のデフォルトのバインドポイントから呼び出されます。

コマンドラインインターフェイスを使用してキャッシュポリシーバンクでポリシーラベルを呼び出すには

コマンドプロンプトで入力します。

```
1 bind cache policylabel <labelName> -policname<policyName> -priority<
     priority> [-gotoPriorityExpression <gotopriorityExpression>] [-
     invoke <labelType> <labelName>]
2 <!--NeedCopy-->
```


GUI を使用してキャッシュポリシーバンク内のポリシーラベルを呼び出すには、次の手順を実行します。

1. [最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイント ([グローバル上書き] または [デフォルトグローバル]) と接続タイプを指定して、このバインドポイントにバインドされているポリシーのリストを表示します。
2. ポリシーを評価せずにポリシー・ラベルを呼び出す場合は、[**NOPOLICY**] をクリックします。

> 注:
>
> 外部ポリシーバンクを呼び出すには、[呼び出しタイプ] 列のフィールドをクリックし、ポリシーバンクでこの時点で呼び出すポリシーバンクのタイプを選択します。これは、グローバルラベルまたは仮想サーババンクです。[呼び出し名] フィールドに、ラベルまたは仮想サーバ名を入力します。

コマンドラインインターフェイスを使用して仮想サーバーポリシーバンクでキャッシュポリシーラベルを呼び出すには

コマンドプロンプトで入力します。

```
bind lb vserver <name>@ -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

```
bind cs vserver <name> -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

GUI を使用して仮想サーバポリシーバンクでキャッシュポリシーラベルを呼び出すには

1. [トラフィック管理] > [負荷分散/コンテンツスイッチング] > [仮想サーバー] に移動し、仮想サーバーを選択して [ポリシー] をクリックします。
2. このバンクに既存のエントリを設定する場合は、この手順をスキップします。このポリシーバンクに新しいポリシーを追加する場合、または「ダミー」NOPOLICY エントリを使用する場合は、[追加] をクリックして、次のいずれかの操作を行います。
   - 新しいポリシーを設定するには、「統合キャッシュでのポリシーの設定」の説明に従って、[キャッシュ] をクリックし、新しいポリシーを設定します。
   - ポリシールールを処理せずにポリシーバンクを呼び出すには、NOPOLICY-CACHEオプションを選択します。

> 注:
>
> 外部ポリシーバンクを呼び出すには、[呼び出しタイプ] 列のフィールドをクリックし、ポリシーバンクでこの時点で呼び出すポリシーバンクのタイプを選択します。これは、グローバルラベルまたは仮想サーババンクです。［呼び出し名］フィールドに、ラベルまたは仮想サーバ名を入力します。

統合キャッシュでのポリシーラベルの設定

組み込みバインドポイントまたは仮想サーバーのポリシーバンクでポリシーを構成する以外に、キャッシュポリシーラベルを作成し、これらの新しいラベルのポリシーバンクを構成できます。

統合キャッシュのポリシーラベルは、統合キャッシュの詳細ペインの Policy Manager で表示できるバインドポイントの 1 つ（リクエストの上書き、リクエストのデフォルト、応答の上書き、または応答のデフォルト）または組み込みポリシーラベル_reqBuiltinDefaultsおよび_resBuiltinDefaultsからのみ呼び出すことができます。ポリシーラベルは、一度だけ呼び出すことができるポリシーとは異なり、何度でも呼び出すことができます。

Citrix ADC GUI には、ポリシーラベルの名前を変更するオプションがあります。ポリシー・ラベルの名前を変更しても、ラベルにバインドされているポリシーの評価プロセスには影響しません。

> 注:
>
> NOPOLICY「ダミー」ポリシーを使用して、別のポリシーバンクからポリシーラベルを呼び出すことができます。NOPOLICYエントリは、ルールを処理しないプレースホルダです。

コマンドラインインターフェイスを使用してキャッシュ用のポリシーラベルを構成するには

コマンド・プロンプトで次のコマンドを入力して、ポリシー・ラベルを作成し、構成を確認します。

- `add cache policylabel <labelName> -evaluates (REQ|RES)`
- `show cache policylabel <labelName>`

このポリシーラベルは、ポリシーバンクから呼び出します。

GUI を使用してキャッシュのポリシーラベルを設定するには、次の手順を実行します。

[最適化] > [統合キャッシュ] > [ポリシーラベル] に移動し、ポリシーラベルを追加して、キャッシュされたポリシーをバインドします。

> 注:
> Citrix ADC が適切なタイミングでポリシーラベルを処理できるようにするには、組み込みのバインドポイントに関連付けられているポリシーバンクのいずれかでこのラベルの呼び出しを構成します。

GUI を使用してポリシーラベルの名前を変更するには、次の手順を実行します。

[最適化] > [統合キャッシュ] > [ポリシーラベル] に移動して、ポリシーラベルを選択し、名前を変更します。

統合キャッシュポリシーとポリシーラベルのバインド解除と削除

ポリシーバンクからポリシーをバインド解除したり、削除したりできます。ポリシーを削除するには、まずポリシーをバインド解除する必要があります。ポリシーラベルの呼び出しを削除し、ポリシーラベルを削除することもできます。ポリシーラベルを削除するには、まずラベルに設定した呼び出しをすべて削除する必要があります。

組み込みのバインドポイント (要求のデフォルト、要求のオーバーライド、応答のデフォルト、応答の上書き) のラベルのバインドを解除または削除することはできません。

コマンドラインインターフェイスを使用してグローバルキャッシュポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
unbind cache global <policy>
```

コマンドラインインターフェイスを使用して仮想サーバー固有のキャッシュポリシーをバインド解除するには

コマンドプロンプトで入力します。

```
(unbind lb vserver|unbind cs vserver)<vserverName> -policyName <policyName>
 -type(REQUEST|RESPONSE)
```

コマンドラインインターフェイスを使用してキャッシュポリシーを削除するには

コマンドプロンプトで入力します。

```
rm cache policy <policyName>
```

GUI を使用してキャッシュポリシーをバインド解除するには、次の手順を実行します。

[最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイントと接続タイプ (要求/応答) を指定してポリシーをバインド解除します。

GUI を使用してポリシーラベルの呼び出しを削除するには、次の手順を実行します。

1. [最適化] > [統合キャッシュ] に移動し、[キャッシュポリシーマネージャ] をクリックし、関連するバインドポイント (負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバー) と接続タイプを指定して、この仮想サーバーにバインドされたキャッシュポリシーのリストを表示します。
2. ポリシーの [呼び出し] 列で、エントリをクリアします。

データベース・プロトコルのキャッシュ・サポート

October 7, 2021

統合キャッシュ機能は、キャッシュポリシーによって決定されたデータベース要求を監視し、キャッシュします。Citrix ADC アプライアンスはデフォルトのポリシーを提供しないため、ユーザーは MYSQL および MSSQL プロトコルのキャッシュポリシーを構成する必要があります。デフォルトのプロトコルを構成するときは、要求ベースのポリシーは CACHE アクションと INVAL アクションのみをサポートし、応答ベースのポリシーは「NOCACHE」アク

ションのみをサポートすることに注意してください。ポリシーを構成した後、それらを仮想サーバーにバインドする必要があります。MYSQL および MSSQL ポリシーは、要求と応答の両方で、仮想サーバーにのみバインドされます。

キャッシュポリシーを作成する前に、タイプ MYSQL または MSSQL のキャッシュコンテンツグループを作成する必要があります。キャッシュコンテンツグループを作成するときは、少なくとも 1 つの選択セレクターを関連付けます。キャッシュコンテンツグループの設定については、「基本コンテンツグループの設定」を参照してください。

次の例では、SQL プロトコルのキャッシュサポートを構成および検証する方法について説明します。

```
1  > enable feature IC
2  > set cache parameter -memlimit 100
3  > add cache selector sel1 mssql.req.query.text
4
5  > add cache contentgroup cg1  -type ”MSSQL” -hitselector ”sel1” -
       invalselector ”inval_sel” -relExpiry ”500” -maxResSize
6   ”100”
7  > add cache policy cp1  -rule ”mssql.req.query.command.contains(”select
       ”)” -action ”CACHE” -storeInGroup ”cg1”
8  > add cache policy cp2  -invalObjects ”cg1” -rule ”mssql.req.query.text
       .contains(”insert”)” -action ”INVAL”
9  > add db user user1  -password ”Pass1”
10 > add service svc_sql_1 10.102.147.70 mssql 64834 -healthMonitor ”NO” -
       downstateflush ”ENABLED”
11 > add lb vserver lb_mssql1 mssql 10.102.147.77 1433  -lbmethod ”
       roundrobin”
12 > bind lb vserver lb_mssql1 svc_sql_1
13 > bind lb vserver lb_mssql1 -policyName cp1  -type ”REQUEST” -priority
       ”2”
14 > bind lb vserver lb_mssql1 -policyName cp2  -type ”REQUEST” -priority
       ”1”
15
16 > show cache selector sel1
17      Name:sel1
18                      Expressions:
19                      1)mssql.req.query.text
20 > show cache policy cp1
21                      Name:cp1
22      Rule:mssql.req.query.command.contains(”select”)
23                      CacheAction:CACHE
24                      Stored in group: cg1
25                      UndefAction:Use Global
26                  Hits:2
27                      Undef Hits:0
28                      Policy is bound to following entities
```



```
                        1) Bound to:
                                    REQ VSERVER lb_mssql1
                                    Priority:2
                                    GotoPriorityExpression: END
<!--NeedCopy-->
```

> 注:
>
> フラッシュクラウドの削減で説明されているように、フラッシュクラウドを削減する方法は、MYSQL および MSSQL プロトコルではサポートされていません。

キャッシュポリシーとセレクタの式を設定する

October 7, 2021

リクエスト時間式は、リクエストタイムトランザクションのデータを検査し、応答時間式は応答時間トランザクション内のデータを検査します。キャッシュのポリシーで、式が要求または応答のデータと一致する場合、Citrix ADC アプライアンスはポリシーに関連付けられたアクションを実行します。セレクタでは、リクエスト時間式を使用して、コンテンツグループに格納されている一致する応答を検索します。

統合キャッシュのポリシーとセレクタを構成する前に、少なくとも HTTP 要求および応答 URL に表示されるホスト名、パス、および IP アドレスを知っておく必要があります。そして、おそらく HTTP リクエストとレスポンス全体の形式を知る必要があります。Live HTTP ヘッダーhttp://livehttpheaders.mozdev.org/)or HTTPFox https://addons.mozilla.org/en-US/firefox/addon/6647などのプログラムは、組織が操作する HTTP データの構造を調べるのに役立ちます。

株価プログラムに対する HTTP GET リクエストの例を次に示します。

```
GET /quote.dll?page=dynamic&mode=data&mode=stock&symbol=CTXS&page=multi
    &selected=CTXS&random=0.00792039478975548 HTTP/1.1

Host: quotes.mystockquotes.com

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
    Gecko/2008052906 Firefox/3.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q
    =0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate,compress,pack200-gzip
```



```
12
13 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
14
15 Keep-Alive: 300
16
17 Connection: keep-alive
18
19 Referer: http://quotes.mystockquotes.com/quote.dll?mode=stock&symbol=
       CTXS&page=multi&selected=CTXS
20
21 Cookie: __qca=1210021679-72161677-10297606
22 <!--NeedCopy-->
```

式を設定するときは、次の制限事項に注意してください。

式タイプ	制限
リクエスト	CACHE アクションまたは NOCACHE アクションを使用して、ポリシーにリクエスト時間式を設定しないでください。代わりに MAY_CACHE または MAY_NOCACHE を使用してください。
応答	キャッシュポリシーでのみ応答時間式を構成します。セレクタはリクエスト時間式のみを使用できます。INVAL アクションを使用してポリシーで応答時間式を設定しないでください。注:CACHE アクションとパラメータ化されたコンテンツグループを使用して、ポリシーで応答時間式を設定しないでください。MAY_CACHE アクションを使用します。

> 注:
> 高度な式の包括的な説明については、「ポリシーと式」を参照してください。

式の構文

構文の基本的なコンポーネントは次のとおりです。

- 次のように、キーワードをピリオド (.) で区切ります。

```
http.req.url
```

- 次のように、文字列値を括弧と引用符で囲みます。

```
http.req.url.query.contains("this")
```


- コマンドラインから式を構成するときは、内部引用符 (式を区切る引用符ではなく、式の値を区切る引用符) をエスケープする必要があります。1 つの方法は、次のようにスラッシュを使用することです。

```
\”abc\”
```

セレクタ式は外観順に評価され、セレクタ定義内の複数の式は論理 AND で結合されます。セレクタ式とは異なり、ブール演算子を指定し、ポリシールールの高度な式の優先順位を変更できます。

キャッシュポリシーまたはセレクタで式を設定する

> 注:
> ポリシー式の構文は、セレクタ式とは異なります。高度な表現に関する包括的な説明については、「ポリシーと式」を参照してください。

コマンドラインインターフェイスを使用してポリシー式を設定するには

1. 「統合キャッシュポリシーのグローバルバインド」の説明に従って、ポリシー定義を開始します。
2. ポリシールールを設定するには、ルール全体を引用符で囲み、ルール内の文字列値をエスケープ引用符で囲みます。

以下はその例です:

「http.req.url.contains (「jpg」)」

ブール値を追加するには、&&、 、または! 演算子。

1.

次に例を示します。

```
”http.req.url.contains(\”jpg\”)|| http.req.url.contains(\”jpeg\”)”
```

```
”http.req.url.query.contains(\”IssuePage\”)”
```

```
”http.req.header(\”Host\”)contains(\”my.company.com\”)&& http.req.method.eq(\”GET\”)&& http.req.url.query.contains(\”v=7\”)”
```

1. コンパウンドの構成部分の評価順序を設定するには

```
”http.req.url.contains(\”jpg\”)|| (http.req.url.contains(\”jpeg\”)&& http.req.method.eq(\”GET\”))”
```

コマンドラインインターフェイスを使用してセレクタ式を設定するには、次の手順を実行します。

1. 「コンテンツ・グループについて」の説明に従って、セレクタ定義を開始します。
2. セレクタ式を設定するには、ルール全体を引用符で囲み、ルール内の文字列値をエスケープ引用符で囲みます。

以下はその例です:

```
”http.req.url.contains(\”jpg\”)”
```


ブール値を追加したり、&&、	、！を挿入することはできません。演算子。各式エレメントを引用符で囲んで入力します。定義内の複数の式は、論理 AND で結合された複合式として扱われます。

1.

次に例を示します。

```
1 ”http.req.url.query.value(”ProductId”)” ”http.req.url.query.value(”
     BatchNum”)” ”http.req.url.query.value(”depotLocation”)”
2 <!--NeedCopy-->
```

GUI を使用してポリシーまたはセレクタ式を設定するには

1. 「構成ユーティリティを使用してキャッシュまたは無効化のポリシーを構成するには」または「構成ユーティリティを使用してセレクタを構成するには」の説明に従って、ポリシーまたはセレクタ定義を開始します。
2. 「式」フィールドに、「クラシック構文に切り替え」をクリックしてデフォルトの構文を手動で入力するか、式エディタを使用して新しい式を作成できます。

複合式の 2 つの部分の間に演算子を挿入するには、[演算子] ボタンをクリックし、オペレータタイプを選択します。次に、ブール値 OR (二重縦棒、	）が設定された式の例を示します。

3.
4. [頻繁に使用する式] ドロップダウンリストをクリックして、よく使用される式を挿入します。
5. 式をテストするには、[評価] をクリックします。[式エバリュエータ] ダイアログボックスで、式と一致するフロータイプを選択します。データフィールドに、式を使用して解析する HTTP リクエストまたはレスポンスを貼り付け、[評価] をクリックします。

キャッシュされたオブジェクトとキャッシュ統計を表示する

特定のキャッシュされたオブジェクトを表示したり、キャッシュリクエスト、ミス、メモリ使用量に関するサマリー統計を表示できます。この統計は、キャッシュから提供されるデータの量、最大のパフォーマンス上の利点の原因と

なる項目、およびキャッシュのパフォーマンスを向上させるために調整できる内容に関する洞察を提供します。

ここでは、次の詳細について説明します。

- キャッシュされたオブジェクトの表示
- 特定のキャッシュされたレスポンスの検索
- キャッシュ統計の表示

キャッシュされたオブジェクトの表示

キャッシュを有効にすると、キャッシュされたオブジェクトの詳細を表示できます。たとえば、次の項目を表示できます。

- レスポンスサイズとヘッダーサイズ
- ステータスコード
- コンテンツグループ
- ETag、最終変更ヘッダー、およびキャッシュ制御ヘッダー
- URL をリクエストする
- ヒットパラメータ
- 宛先 IP アドレス
- リクエストと応答時間

コマンドラインインターフェイスを使用してキャッシュされたオブジェクトのリストを表示するには

コマンドプロンプトで入力します。

```
show cache object
```

プロパティ	説明
応答サイズ (バイト)	レスポンスヘッダーとボディのサイズ。
レスポンスヘッダーのサイズ (バイト)	レスポンスのヘッダー部分のサイズ。
レスポンスステータスコード	レスポンスとともに送信されるステータスコード。
ETag	応答に挿入された ETag ヘッダー。通常、このヘッダーは、応答が最近変更されたかどうかを示します。
最終変更日	レスポンスに挿入された Last-Modified ヘッダー。このヘッダーは、レスポンスが最後に変更された日付を示します。
キャッシュコントロール	レスポンスに挿入された Cache-Control ヘッダー。
日付	応答がいつ送信されたかを示す Date ヘッダー。
コンテンツグループ	レスポンスが格納されるコンテンツグループ。

プロパティ	説明
複合マッチ	このオブジェクトがパラメータ化された値に基づいてキャッシュされている場合、このフィールド値は YES です。
ホスト	このレスポンスを要求した URL で指定されたホスト。
ホストポート	このレスポンスを要求した URL で指定されたホストのリスニングポート
URL	格納されたレスポンスに対して発行された URL。
接続先 IP	このレスポンスが取得されたサーバーの IP アドレス。
Destination port	宛先サーバーのリスニングポート。
ヒットパラメータ	レスポンスを格納するコンテンツグループがヒットパラメータを使用している場合は、このフィールドに表示されます。
ヒットセレクタ	このコンテンツグループがヒットセレクタを使用している場合は、このフィールドに表示されます。
無効なセレクタ	このコンテンツグループが無効化セレクタを使用している場合は、このフィールドに表示されます。
セレクタ式	このコンテンツグループがセレクタを使用している場合、このフィールドには選択ルールを定義する式が表示されます。
要求時間	リクエストが発行されてからの時間（ミリ秒）。
Response time	キャッシュがレスポンスの受信を開始してから経過した時間（ミリ秒）。
年齢	オブジェクトがキャッシュ内にある時間。
有効期限	オブジェクトが期限切れとしてマークされるまでの時間。
フラッシュされた	有効期限が切れた後に応答がフラッシュされたかどうか。
プリフェッチ	このコンテンツグループに対してプリフェッチが設定されている場合、オブジェクトがオリジンからフェッチされる有効期限までの時間。プリフェッチはネガティブオブジェクトには適用されません (たとえば、404「オブジェクトが見つかりません」というレスポンス)。

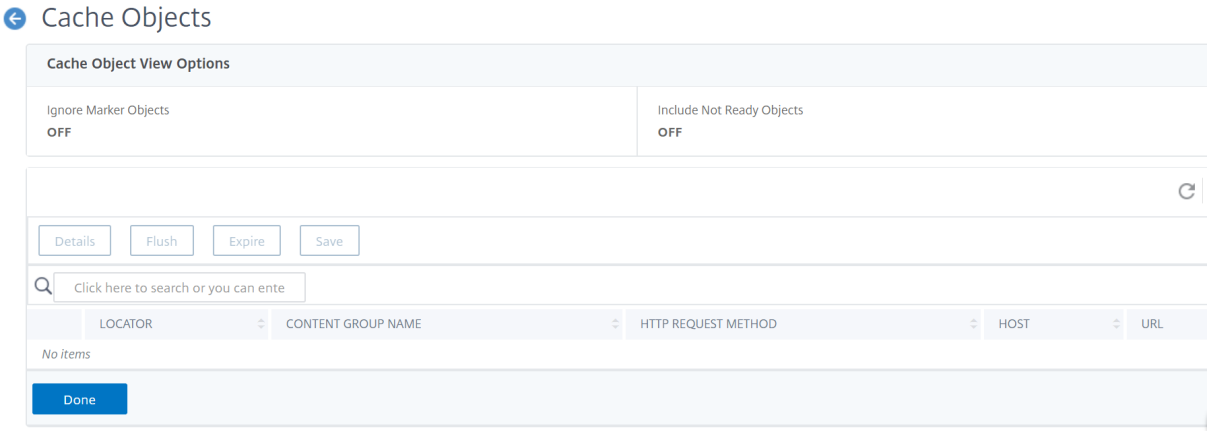
プロパティ	説明
現在の読者	現在処理されているリクエストの概数。Content-Length ヘッダーオブジェクトを含むレスポンスがダウンロードされている場合、現在のミスおよび現在のリーダー値はそれぞれ 1 になります。チャンクレスポンスオブジェクトがダウンロードされている場合、現在のミス値は通常 1 になりますが、クライアントに対して提供されるチャンク応答は統合キャッシュバッファから送信されないため、現在のリーダー値は通常 0 になります。
現在のミス	キャッシュミスおよびオリジンサーバーからのフェッチの結果となったリクエストの現在の数。この値は、通常 0 または 1 です。コンテンツグループに対して [毎回ポーリング] が有効になっている場合、カウントは 1 より大きくなる可能性があります。
ヒット数	このオブジェクトのキャッシュヒット数。
ミス	このオブジェクトのキャッシュミス数
圧縮形式	このオブジェクトに適用される圧縮のタイプ。圧縮形式としては、gzip、deflate、compress、pack200-gzip などがある。
応答中の HTTP バージョン	レスポンスの送信に使用された HTTP のバージョン。
応答に弱い etag が存在する	エンティティのビットが変更された場合、強力な etag ヘッダーが変わります。強力なヘッダーは、オブジェクトのオクテット値に基づいています。エンティティの意味が変わると、弱い etag ヘッダーが変わります。弱な etag 値は、セマンティックアイデンティティに基づいています。弱い etags の値は「W」で始まります。
ネガティブマーカーセル	マーカーオブジェクトはキャッシュ可能ですが、キャッシュされるための条件をすべて満たしているわけではありません。たとえば、オブジェクトがコンテンツグループの最大応答サイズを超えている可能性があります。このタイプのオブジェクトに対してマーカーセルが作成されます。ユーザーが次回このオブジェクトに対するリクエストを送信すると、キャッシュミスが提供されます。

プロパティ	説明
理由マーカーが作成されました	マーカーセルが作成された理由 (たとえば、「minhit を待っている」、「コンテンツ長の応答データがグループサイズの制限にありません」など)。
毎回オートポーリング	統合キャッシュがバリデーター（Last-Modified または eTag レスポンスヘッダーのいずれか）で既に期限切れの 200 OK レスポンスを受け取った場合、レスポンスを保存し、Auto-PET（毎回自動的にポーリング）としてマークします。
Citrix ADC Etag がレスポンスで挿入されました	Citrix ADC アプライアンスによって生成される ETag ヘッダーのバリエーション。Citrix ADC が応答に Etag を挿入すると、値 YES が表示されます。
キャッシュに完全なレスポンスが存在	これが完全な応答であるかどうかを示します。
DNS によって検証された宛先 IP	オブジェクトの格納時に DNS 解決が実行されたかどうかを示します。
キャッシュフォワードプロキシを介して格納されたオブジェクト	統合キャッシュに設定されたフォワードプロキシが原因で、この応答が格納されたかどうかを示します。
オブジェクトはデルタベースファイルです	デルタ圧縮された応答。
minhits 待ち	レスポンスをキャッシュする前に、このコンテンツグループで最小数のオリジンサーバーヒットが必要かどうかを示します。
ミニヒットカウント	オブジェクトをキャッシュする前に、このコンテンツグループが最小数のオリジンサーバー要求を必要とする場合、このフィールドにはこれまでに受信したリクエストの数が表示されます。
HTTP リクエストメソッド	このオブジェクトを取得したリクエストで使用されるメソッド GET または POST。
ポリシー別に格納	このオブジェクトが格納された原因となったキャッシュポリシーの名前。[利用不可] の値は、ポリシーが非アクティブ化または削除されたことを示します。 NONE の値は、オブジェクトが可視ポリシーと一致せず、キャッシュの内部基準に従って格納されたことを示します。

プロパティ	説明
アプリケーションファイアウォールのメタデータが存在します	このパラメータは、アプリケーションファイアウォールと統合キャッシュの両方が有効になっている場合に使用されます。アプリケーションファイアウォールは、レスポンスページのコンテンツを分析し、そのメタデータ（ページに含まれる URL やフォームなど）を保存し、レスポンスを含むメタデータをキャッシュにエクスポートします。キャッシュはページとメタデータを格納し、キャッシュがページを提供すると、メタデータをリクエストのセッションに送り返します。
HTTP コールアウトオブジェクト、名前、タイプ、応答	これらのセルは、HTTP コールアウト式の結果としてこのデータが格納されたかどうかを示し、コールアウトのさまざまな側面と対応する応答に関する情報を提供します。HTTP コールアウトの詳細については、「HTTP コールアウト」を参照してください。

GUI を使用してキャッシュされたオブジェクトを表示するには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動します。キャッシュされたすべてのオブジェクトを表示し、必要に応じて並べ替えることができます。



キャッシュされた特定のレスポンスを見つける

検索条件に基づいて、キャッシュ内の個々のアイテムを検索できます。キャッシュされたアイテムを検索するには、データを含むコンテンツグループがヒットセレクターと無効化セレクターを使用するかどうかに応じて、次のようにさまざまな方法があります。

- コンテンツグループがセレクターを使用している場合、キャッシュされたアイテムのロケータ ID を使用してのみ検索を実行できます。

- コンテンツグループがセレクタを使用しない場合は、URL、ホスト、コンテンツグループ名などの基準を使用して検索を実行します。

キャッシュされたレスポンスを検索するときに、URL とホストでいくつかの項目を見つけることができます。レスポンスがセレクタを使用するコンテンツグループ内にある場合は、ロケータ番号（たとえば、0x00000000ad7af00000050）を使用してのみ検索できます。後で使用するためにロケータ番号を保存するには、エントリを右クリックし、[コピー] を選択します。セレクタの詳細については、「セレクタと基本コンテンツグループの設定」を参照してください。

コマンドラインインターフェイスを使用して、セレクタを持たないコンテンツグループのキャッシュされたレスポンスを表示するには

コマンドプロンプトで入力します。

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-host <
hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET
 | POST ]))| [-httpStatus <positive integer>] | -group <contentGroupName> |
 -ignoreMarkerObjects (ON | OFF)| -includeNotReadyObjects (ON | OFF)]
```

コマンドラインインターフェイスを使用してセレクタを持つコンテンツグループのキャッシュされたレスポンスを表示するには

コマンドプロンプトで入力します。

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )| -
includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer>]
```

構成ユーティリティを使用して、セレクタを持たないコンテンツグループのキャッシュされた応答を表示するには

[最適化] **>** [統合キャッシュ] **>** [キャッシュオブジェクト] に移動し、[検索] をクリックし、必要なキャッシュされた応答を表示するように検索条件を設定します。

コンテンツグループをまだ構成していない場合は、すべてのオブジェクトが Default グループに属します。

キャッシュ統計の表示

次の表に、表示できる詳細なキャッシュ統計情報をまとめたものです。

カウンター	説明
ヒット数	統合キャッシュで見つかり、統合キャッシュから提供されるレスポンス。イメージファイルなどの静的オブジェクト、ステータスコード 200、203、300、301、302、304、307、403、404、410、および CACHE アクションを使用したユーザー定義のポリシーに一致する応答を含むページが含まれます。

カウンター	説明
ミス	レスポンスが最終的にオリジンサーバーからフェッチされた HTTP リクエストをインターセプトしました。
リクエスト	キャッシュリクエストの合計とキャッシュミスの合計。
304 ヒット以外	ユーザーがアイテムを複数回要求し、Citrix ADC アプライアンスが最後にアイテムを提供してからキャッシュ内のアイテムが変更されていない場合、Citrix ADC アプライアンスはキャッシュされたオブジェクトの代わりに 304 の応答を提供します。
この統計は、304 の応答を除いて、Citrix ADC アプライアンスがキャッシュから処理したアイテムの数を示します。	
304 ヒット	Citrix ADC アプライアンスがキャッシュから処理した 304（オブジェクトが変更されていない）応答の数。
304 ヒット率 (%)	Citrix ADC アプライアンスが提供した 304 の応答のうち、他の応答に対する割合。
ヒット率 (%)	キャッシュから提供できなかった応答に対する Citrix ADC アプライアンスがキャッシュ（キャッシュリクエスト）から処理したレスポンスの割合。
オリジン帯域幅の保存 (%)	キャッシュからの応答を提供するため、Citrix ADC アプライアンスがオリジンサーバーに保存した処理能力の推定。
Citrix ADC によって処理されるバイト数	Citrix ADC アプライアンスがオリジンサーバーとキャッシュから処理した合計バイト数。
キャッシュによって処理されるバイト数	Citrix ADC アプライアンスがキャッシュから処理した合計バイト数。
バイトヒット率（%）	Citrix ADC アプライアンスがキャッシュから提供したデータの割合。すべての応答に含まれるすべてのデータに対する相対値。
キャッシュから圧縮されたバイト	Citrix ADC アプライアンスが圧縮形式で処理したデータの量（バイト単位）。

カウンター	説明
ストレージ可能なミス	Citrix ADC アプライアンスがキャッシュ内で要求されたオブジェクトを見つけない場合、オリジンサーバーからオブジェクトを取得します。これは、キャッシュミスと呼ばれます。保存可能なキャッシュミスはキャッシュに格納できます。
格納不能なミス	格納不可能なキャッシュミスをキャッシュに保存できません。
Misses	すべてのキャッシュミス。
Revalidations	Cache-Control ヘッダーの Max-Age 設定により、ユーザーに提供する前に、中間キャッシュが統合キャッシュでコンテンツを再検証する必要があるタイミングを秒数で決定します。
詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。	
正常に再検証	実行された再検証の数。
詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。	
条件付き Req への変換	キャッシュされた PET オブジェクトに対するユーザーエージェントリクエストは、常に条件付きリクエストに変換され、オリジンサーバーに送信されます。
詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。	
ストレージ可能なミス率 (%)	格納可能なキャッシュミスを格納できないキャッシュミスのパーセンテージで表す。
成功したリヴァール率 (%)	すべての再検証試行に対するパーセンテージで表した再検証の成功率。
詳しい情報については、「キャッシュコントロールヘッダーの挿入」を参照してください。	
最後のバイトで期限切れ	最後のボディバイトを受け取った直後にキャッシュのコンテンツが期限切れになった回数。表の「キャッシュヒットとミス」で説明されているように、ポジティブレスポンスにのみ適用されます。

カウンター	説明
詳細については、「パフォーマンス最適化の例」を参照してください。	
Flashcache misses	Flash Cache を有効にした場合、キャッシュはサーバーへのリクエストを 1 つだけ許可し、フラッシュの群れを排除します。この統計は、キャッシュミスであったフラッシュキャッシュ要求の数を示します。
詳細は、「キャッシュへのリクエストのキューイング。」	
Flashcache ヒット	キャッシュヒットしたフラッシュキャッシュリクエストの数。
詳細は、「キャッシュへのリクエストのキューイング」を参照してください。	
パラメータ化されたインバルリクエスト	無効化 (INVAL) アクション、および無効化セレクタまたはパラメータを使用してグループ内のキャッシュされたオブジェクトを選択的に期限切れにするコンテンツグループを持つポリシーに一致するリクエスト。
完全インバル要求	invalGroups パラメータが設定され、1 つ以上のコンテンツグループが期限切れになる無効化ポリシーに一致するリクエスト。
Inval Request	無効化ポリシーに一致し、特定のキャッシュされたレスポンスまたはコンテンツグループ全体の有効期限が切れるリクエスト。
パラメータ化されたリクエスト	パラメータ化されたコンテンツグループのポリシーを使用して処理されたキャッシュリクエストの数。
パラメータ化された非 304 ヒット	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。フルキャッシュされたレスポンスが検出され、応答が 304（オブジェクトが更新されていない）応答ではありませんでした。
Parameterized 304 ヒット	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。キャッシュされたオブジェクトが見つかり、オブジェクトが 304（オブジェクトが更新されていない）応答でした。

カウンター	説明
パラメータ化されたヒット数の合計	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。キャッシュされたオブジェクトが見つかった。
パラメータ化された 304 ヒット率 (%)	パラメータ化されたポリシーを使用して検出された 304 (オブジェクトが更新されていない) 応答のうち、すべてのキャッシュヒットに対する割合。
リクエストのたびにポーリング	毎回ポーリングが有効になっている場合、Citrix ADC アプライアンスは、保存されたオブジェクトを提供する前に、常にオリジンサーバーを参照します。
詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。	
ヒットするたびにポーリング	毎回ポーリングメソッドを使用してキャッシュヒットが検出された回数。
詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。	
毎回ポーリングヒット率 (%)	Poll Every Time メソッドを使用したキャッシュヒットのパーセンテージ。[毎回ポーリング] を使用したキャッシュオブジェクトに対するすべての検索に対する相対値。詳しい情報については、「リクエストを受信するたびにオリジンサーバーのポーリングを行うこと」を参照してください。
最大メモリ（KB）	キャッシュに割り当てられている Citrix ADC アプライアンスの最大メモリ容量。詳細は、「キャッシュのグローバル属性の設定」を参照してください。
最大メモリアクティブ値 (KB)	メモリをキャッシュに割り当てた後に設定される最大メモリ量 (アクティブ値)。詳細については、「さまざまなシナリオで Citrix ADC アプライアンスの統合キャッシュ機能を構成する方法」を参照してください。
使用済みメモリ (KB)	実際に使用されているメモリの量。
メモリ割り当て失敗	キャッシュにレスポンスを格納する目的でメモリの使用に失敗した回数。

カウンター	説明
これまでの最大の応答	キャッシュまたはオリジンサーバーのいずれかで見つかり、クライアントに送信された最大レスポンス (バイト単位)。
キャッシュされたオブジェクト	キャッシュ内のオブジェクトの数。まだ完全にダウンロードされていないレスポンスや、有効期限が切れているがまだフラッシュされていないレスポンスを含みます。
Marker オブジェクト	マーカーオブジェクトは、応答がコンテンツグループの最大または最小応答サイズを超えている場合、またはコンテンツグループの最小ヒット数をまだ受け取っていない場合に作成されます。
処理中のヒット	キャッシュから配信されたヒットの数。
処理ミス	オリジンサーバーからフェッチされ、キャッシュに保存され、処理された応答。保管可能なミスの数を概算する必要があります。保存不可能なミスは含まれません。

コマンドラインインターフェイスを使用してサマリーキャッシュ統計を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
stat cache
```

コマンドラインインターフェイスを使用して特定のキャッシュ統計情報を表示するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
stat cache -detail
```

```
1 > stat cache -detail
2
3 Integrated Cache Statistics - Detail
4 Integrated Cache Statistics - Summary
5                                          Rate (/s)
                                              Total
6
7
8 Hits                                              0
                        0
9
```



```
Misses                                                  0
                        0

Requests                                                0
                        0

Hit ratio(%)                                           --
                        0

Origin bandwidth saved(%)                              --
                        0
Cached objects                                         --
                        0

Marker objects                                         --
                        0
                                                Rate (/s)
                                                   Total

Requests                                                0
                        0


Hit Statistics

                                                Rate (/s)
                                                   Total


Non-304 hits                                            0
                        0

304 hits                                                0
                        0


Sql hits                                                0
                        0


Hits                                                    0
                        0

304 hit ratio(%)                                       --
                        0
```



```
41
42 Hit ratio(%)                                          --
                        0
43
44 Origin bandwidth saved(%)                            --
                        0
45 Byte Statistics
46                                                 Rate (/s)
                                                      Total
47
48
49 Bytes served by Citrix ADC                            648
    55379204
50
51 Bytes served by cache                                   0
                        0
52 Byte hit ratio(%)                                    --
                        0
53 Compressed bytes from cache                             0
                        0
54
55 Miss Statistics
56
57                                                 Rate (/s)
                                                      Total
58
59
60 Storable misses                                         0
                        0
61
62 Non-storable misses                                     0
                        0
63
64 Misses                                                  0
                        0
65
66 Revalidations                                           0
                        0
67
68 Successful revalidations                                0
                        0
69
70 Conversions to conditional req                          0
                        0
71
```



```
72
73 Storable miss ratio(%)                                --
                        0
74 Successful reval ratio(%)                            --
                        0
75
76 Flashcache Statistics
77                                                  Rate (/s)
                                                       Total
78
79
80 Expire at last byte                                     0
                        0
81
82 Flashcache misses                                       0
                        0
83 Flashcache hits                                         0
                        0
84
85 Invalidation Statistics
86
87                                                  Rate (/s)
                                                       Total
88
89 Parameterized inval requests                            0
                        0
90
91
92 Full inval requests                                     0
                        0
93
94
95
96 Inval requests                                          0
                        0
97
98 Parameterized Caching Statistics
99
100                                                 Rate (/s)
                                                       Total
101
102
103 Parameterized requests                                 0
                        0
104
```



```
105 Parameterized non-304 hits                          0
                        0
106
107 Parameterized 304 hits                              0
                        0
108
109
110 Total parameterized hits                            0
                        0
111
112 Parameterized 304 hit ratio(%)                      --
                        0
113
114 Poll Every Time (PET) Statistics
115
116                                             Rate (/s)
                                                   Total
117
118
119 Poll every time requests                             0
                        0
120
121 Poll every time hits                                 0
                        0
122
123 Poll every time hit ratio(%)                        --
                        0
124
125 Memory Usage Statistics
126                                                 Total
127
128 Maximum memory(KB)                                   0
129
130 Maximum memory active value(KB)                      0
131
132 Utilized memory(KB)                                  0
133
134 Memory allocation failures                           0
135
136 Largest response so far(B)                           0
137
138 Cached objects                                       0
139
140 Marker objects                                       0
141
```



```
Hits being served                                    0
Misses being handled                                 0
 Done
<!--NeedCopy-->
```

GUI を使用してサマリーキャッシュ統計を表示するには

1. ページの上部にある [ダッシュボード] タブをクリックします。
2. ウィンドウの [統合キャッシュ] セクションまでスクロールします。
3. 詳細な統計を表示するには、表の下部にある [More...] リンクをクリックします。

GUI を使用して特定のキャッシュ統計を表示するには

1. ページの上部にある [レポート] タブをクリックします。
2. [組み込みレポート] で、[統合キャッシュ] を展開し、表示する統計情報を含むレポートをクリックします。
3. レポートをテンプレートとして保存するには、[名前を付けて保存] をクリックし、レポートの名前を指定します。保存したレポートが [カスタムレポート] の下に表示されます。

キャッシュされたオブジェクトとキャッシュ統計を表示する

October 7, 2021

特定のキャッシュされたオブジェクトを表示したり、キャッシュヒット、ミス、メモリ使用量に関するサマリー統計を表示できます。この統計は、キャッシュから提供されるデータの量、最大のパフォーマンス上の利点の原因となる項目、およびキャッシュのパフォーマンスを向上させるために調整できる項目に関する洞察を提供します。

ここでは、次の詳細について説明します。

- キャッシュされたオブジェクトの表示
- キャッシュされた特定のレスポンスの検索
- キャッシュ統計情報の表示

キャッシュされたオブジェクトの表示

キャッシュを有効にすると、キャッシュされたオブジェクトの詳細を表示できます。たとえば、次のアイテムを表示できます。

- 応答サイズとヘッダーサイズ
- ステータスコード
- コンテンツグループ
- ETag、Last-Modified、および Cache-Control ヘッダー
- 要求 URL
- ヒットパラメーター

- 送信先 IP アドレス
- 要求と応答の時間

コマンドラインインターフェイスを使用してキャッシュされたオブジェクトのリストを表示するには

コマンドプロンプトで入力します。

```
show cache object
```

プロパティ	仕様
応答サイズ（バイト）	レスポンスヘッダーと本文のサイズ。
応答ヘッダーサイズ（バイト）	レスポンスのヘッダー部分のサイズ。
応答ステータスコード	応答とともに送信されるステータスコード。
ETag	応答に挿入された ETag ヘッダー。通常、このヘッダーは、応答が最近変更されたかどうかを示します。
Last-Modified	レスポンスに挿入された最終変更ヘッダー。このヘッダーは、応答が最後に変更された日付を示します。
Cache-Control	応答に挿入されたキャッシュ制御ヘッダー。
日付	レスポンスがいつ送信されたかを示す Date ヘッダー。
Contentgroup	応答が格納されるコンテンツグループ。
複雑さの一致	このオブジェクトがパラメーター化された値に基づいてキャッシュされた場合、このフィールド値は YES です。
ホスト	この応答を要求した URL で指定されたホスト。
ホストポート	このレスポンスを要求した URL で指定されたホストのリスンポート
URL	保存された応答に対して発行された URL。
接続先 IP	この応答が取得されたサーバーの IP アドレス。
Destination port	宛先サーバーのリッスンポート。
ヒットパラメーター	レスポンスを格納するコンテンツグループがヒットパラメータを使用する場合、それらがこのフィールドにリストされます。
ヒットセレクタ	このコンテンツグループがヒットセレクタを使用する場合、このフィールドにリストされます。
Inval selector	このコンテンツグループが無効化セレクターを使用する場合、このフィールドに表示されます。

プロパティ	仕様
セレクタ式	このコンテンツグループがセレクタを使用する場合、このフィールドには選択ルールを定義する式が表示されます。
要求時間	リクエストが発行されてからの時間（ミリ秒）。
Response time	キャッシュがレスポンスの受信を開始してから経過した時間（ミリ秒）。
年齢	オブジェクトがキャッシュに保持された時間。
期限	オブジェクトが期限切れとしてマークされるまでの時間。
フラッシュ済み	応答が満了後にフラッシュされたかどうか。
プリフェッチ	このコンテンツ・グループに対してプリフェッチが構成されている場合、有効期限が切れる前にオブジェクトがオリジンからフェッチされる時間。プリフェッチは、ネガティブ・オブジェクトには適用されません（たとえば、404「オブジェクトが見つかりません」というレスポンス）。
現在のリーダー	現在提供されているヒット数の概算です。Content-Length ヘッダーオブジェクトを含むレスポンスがダウンロードされるとき、現在のミスおよび現在のリーダーの値はそれぞれ通常 1 です。チャンク応答オブジェクトがダウンロードされている場合、現在のミス値は通常 1 ですが、クライアントに提供されるチャンク応答は、統合キャッシュバッファから来ていないため、現在のリーダーの値は通常 0 です。
現在のミス	キャッシュミスが発生し、オリジンサーバーからのフェッチが発生したリクエストの現在の数。この値は、通常 0 または 1 です。コンテンツグループに対して [毎回ポーリング] が有効になっている場合、カウントは 1 より大きくなる可能性があります。
ヒット数	このオブジェクトのキャッシュヒット数。
ミス	このオブジェクトのキャッシュミスの数。
圧縮形式	このオブジェクトに適用される圧縮のタイプ。圧縮フォーマットには、gzip、デフレート、圧縮、および pack200-gzip が含まれます。

プロパティ	仕様
レスポンスの HTTP バージョン	レスポンスの送信に使用された HTTP のバージョン。
応答に弱い etag が存在する	エンティティのビットが変更されると、強力な etag ヘッダーが変更されます。強力なヘッダーは、オブジェクトのオクテット値に基づいています。エンティティの意味が変わると、弱い etag ヘッダーも変わります。弱い etag 値は、セマンティックアイデンティティに基づいています。弱い etags 値は「W」で始まります。
負のマーカーセル	マーカーオブジェクトはキャッシュ可能ですが、まだキャッシュされる条件をすべて満たしていません。たとえば、オブジェクトがコンテンツグループの最大応答サイズを超える場合があります。このタイプのオブジェクトには、マーカーセルが作成されます。次回ユーザーがこのオブジェクトに対するリクエストを送信すると、キャッシュミスが処理されます。
作成された理由マーカー	マーカーセルが作成された理由 (例:「minhit を待機中」、「コンテンツ長の応答データがグループサイズ制限にありません」)。
毎回自動ポーリング	統合キャッシュは、バリデーター（Last-Modified または ETag 応答ヘッダー）を使用してすでに期限切れの 200 OK 応答を受信すると、応答を格納し、Auto-PET としてマークします（毎回自動的にポーリングします)。
応答として挿入された Citrix ADC タグ	Citrix ADC アプライアンスによって生成される ETag ヘッダーのバリエーション。Citrix ADC が応答に Etag を挿入すると、値 YES が表示されます。
キャッシュに完全な応答が存在	これが完全な応答かどうかを示します。
DNS 検証済み送信先 IP	オブジェクトを格納するときに DNS 解決が実行されたかどうかを示します。
キャッシュフォワードプロキシ経由で格納されるオブジェクト	統合キャッシュに設定されているフォワードプロキシが原因で、この応答が格納されたかどうかを示します。
オブジェクトはデータベースファイル	デルタ圧縮されたレスポンス。
最小ヒットを待機	このコンテンツグループが、応答をキャッシュする前にヒットするオリジンサーバーの最小数を必要とするかどうかを示します。

プロパティ	仕様
最小ヒット数	このコンテンツグループがオブジェクトをキャッシュする前に最小数のオリジンサーバーヒットを必要とする場合、このフィールドにはこれまでに受信したヒット数のカウントが表示されます。
HTTP 要求メソッド	このオブジェクトを取得したリクエストで使用されるメソッド GET または POST。
ポリシーで格納	このオブジェクトを格納する原因となったキャッシュポリシーの名前。NOT AVAILABLE の値は、ポリシーが非アクティブ化または削除されたことを示します。NONE の値は、オブジェクトが可視ポリシーと一致しなかったが、キャッシュの内部基準に従って格納されたことを示します。
アプリケーションファイアウォールメタデータが存在	このパラメータは、アプリケーションファイアウォールと統合キャッシュの両方が有効になっている場合に使用されます。アプリケーションファイアウォールは、応答ページの内容を分析し、そのメタデータ（ページに含まれる URL やフォームなど）を格納し、応答とともにメタデータをキャッシュにエクスポートします。キャッシュはページとメタデータを格納し、キャッシュがページを提供するとき、メタデータをリクエストのセッションに送り返します。
HTTP コールアウトオブジェクト、名前、タイプ、応答	これらのセルは、このデータが HTTP コールアウト式の結果として格納されたかどうかを示し、コールアウトと対応する応答のさまざまな側面に関する情報を提供します。HTTP コールアウトの詳細については、「HTTP コールアウト」を参照してください。

キャッシュされた特定のレスポンスを検索する

検索条件に基づいて、キャッシュ内の個々のアイテムを検索できます。キャッシュされた項目を検索する方法は、データを含むコンテンツグループがヒットセレクタと無効セレクタを使用するかどうかによって異なります。

コンテンツグループがセレクターを使用する場合、キャッシュされたアイテムのロケーター ID を使用してのみ検索を実行できます。

コンテンツグループがセレクターを使用しない場合は、URL、ホスト、コンテンツグループ名などの条件を使用して検索を実行します。

キャッシュされたレスポンスを検索するときは、URL とホストでいくつかの項目を見つけることができます。応答がセレクタを使用するコンテンツグループ内にある場合は、ロケーター番号 (0x0000000000ad7af000050 など) を使用してのみ検索できます。後で使用するためにロケーター番号を保存するには、エントリを右クリックして [コピー] を選択します。セレクタの詳細は、「セレクタおよび基本コンテンツグループの構成」を参照してください。

コマンドラインインターフェイスを使用して、セレクタを持たないコンテンツグループにキャッシュされたレスポンスを表示するには

コマンドプロンプトで入力します。

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-host <
hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET
 | POST ]))| [-httpStatus<positive integer>] | -group <contentGroupName> |
-ignoreMarkerObjects (ON | OFF)| -includeNotReadyObjects (ON | OFF)]
```

コマンドラインインターフェイスを使用してセレクタを持つコンテンツグループにキャッシュされた応答を表示するには

コマンドプロンプトで入力します。

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )| -
includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer>]
```

GUI を使用してセレクタを持たないコンテンツグループにキャッシュされた応答を表示するには

[最適化] > [統合キャッシュ] > [キャッシュオブジェクト] に移動し、[検索] をクリックし、必要なキャッシュされた応答を表示するように検索条件を設定します。

コンテンツグループをまだ構成していない場合は、すべてのオブジェクトが Default グループに含まれます。

GUI を使用してセレクタを持つコンテンツグループにキャッシュされた応答を表示するには

「最適化」>「統合キャッシュ」>「キャッシュオブジェクト」に移動し、「検索」をクリックし、セレクタ検索条件を設定して、必要なキャッシュされたレスポンスを表示します。

キャッシュ統計の表示

次の表は、キャッシュの統計情報をまとめたものです。

Counter

仕様

キャッシュ統計情報の表示

更新済み：2013-10-28

次の表は、表示できる詳細なキャッシュ統計をまとめたものです。

Counter	Specifies
ヒット数	統合キャッシュで検出され、統合キャッシュから提供される応答。イメージファイル、ステータスコードが 200、203、300、301、302、304、307、403、404、410、および CACHE アクションを持つユーザー定義ポリシーと一致する応答などの静的オブジェクトが含まれます。
ミス	応答が最終的にオリジンサーバーからフェッチされた HTTP リクエストをインターセプトしました。
要求	キャッシュ・ヒットの合計とキャッシュ・ミスの合計。
304 件以外	ユーザーがアイテムを複数回要求し、最後に Citrix ADC アプライアンスがアイテムを配信してからキャッシュ内のアイテムが変更されない場合、Citrix ADC アプライアンスはキャッシュされたオブジェクトの代わりに 304 レスポンスを返します。この統計は、Citrix ADC アプライアンスがキャッシュから処理したアイテムの数（304 レスポンスを除く）を示します。
ヒット回数 304 件	Citrix ADC アプライアンスがキャッシュから処理した 304（オブジェクトが変更されていない）応答の数。
304 hit ratio (%)	Citrix ADC アプライアンスが処理した 304 応答の割合（他の応答に対する割合）。
Hit ratio (%)	Citrix ADC アプライアンスがキャッシュから処理したレスポンス（キャッシュヒット）のうち、キャッシュから処理できなかったレスポンスの割合。
Origin bandwidth saved (%)	キャッシュからの応答を提供するために Citrix ADC アプライアンスがオリジナル・サーバーに保存した処理キャパシティの見積もり。
Citrix ADC によって処理されるバイト数	Citrix ADC アプライアンスがオリジナル・サーバーとキャッシュから処理した合計バイト数。
Bytes served by cache	Citrix ADC アプライアンスがキャッシュから処理した合計バイト数。
Byte hit ratio(%)	Citrix ADC アプライアンスがキャッシュから処理したデータの割合。すべての応答に含まれるすべてのデータに対する相対的な割合。
Compressed bytes from cache	Citrix ADC アプライアンスが圧縮形式で処理したデータの量（バイト単位）。

Counter	Specifies
Storable misses	Citrix ADC アプライアンスは、要求されたオブジェクトをキャッシュ内で検出しない場合、オリジナル・サーバーからオブジェクトをフェッチします。これは、キャッシュミスと呼ばれます。保存可能なキャッシュミスをキャッシュに保存できます。
Non-storable misses	保存不可能なキャッシュミスはキャッシュに保存できません。
ミス	すべてのキャッシュミス。
Revalidations	Cache-Control ヘッダーの Max-Age 設定は、介在するキャッシュが統合キャッシュでコンテンツをユーザーに提供する前に再検証する必要があるタイミングを秒単位で決定します。詳細は、「キャッシュ制御ヘッダーの挿入」を参照してください。
Successful revalidations	実行された再検証の数。詳細は、「キャッシュ制御ヘッダーの挿入」を参照してください。
Conversions to conditional req	キャッシュされた PET オブジェクトに対するユーザーエージェント要求は、常に条件付き要求に変換され、オリジンサーバーに送信されます。詳細は、「要求を受信するたびにオリジンサーバーをポーリングする」を参照してください。
Storable miss ratio (%)	格納可能なキャッシュ・ミスの割合として、格納不可能なキャッシュ・ミスを示します。
Successful reval ratio (%)	再検証の成功率（すべての再検証試行の割合）。詳細は、「キャッシュ制御ヘッダーの挿入」を参照してください。
Expire at last byte	最後のボディバイトを受信した直後にキャッシュがコンテンツを期限切れにした回数。「キャッシュ・ヒットとミス」の表で説明されているように、ポジティブな応答にのみ適用されます。詳細については、「パフォーマンス最適化の例」を参照してください。

Counter	Specifies
フラッシュキャッシュミス	Flash Cache を有効にすると、キャッシュはサーバーへのリクエストを 1 つだけ許可し、フラッシュクラウドを排除します。この統計は、キャッシュミスした Flash Cache リクエストの数を示します。詳細については、「キャッシュへのリクエストのキューイング」を参照してください。
Flashcache ヒット曲	キャッシュヒットであったフラッシュキャッシュリクエストの数。詳細については、「キャッシュへの要求のキューイング」を参照してください。
Parameterized inval requests	無効化（INVAL）アクションを持つポリシーと、無効化セレクタまたはパラメータを使用してグループ内のキャッシュされたオブジェクトを選択的に期限切れにするコンテンツグループと一致するリクエスト。
Full inval requests	invalGroups パラメータが設定され、1 つ以上のコンテンツグループが期限切れになる無効ポリシーと一致するリクエスト。
Inval requests	無効化ポリシーに一致し、特定のキャッシュされた応答またはコンテンツグループ全体の有効期限が切れるリクエスト。
Parameterized requests	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。
パラメータ化された 304 以外のヒット	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュリクエストの数。フルキャッシュされた応答が見つかり、応答が 304 (オブジェクトが更新されていない) 応答ではなかった。
Parameterized 304 hits	パラメータ化されたコンテンツグループを持つポリシーを使用して処理されたキャッシュ要求の数。キャッシュされたオブジェクトが見つかり、オブジェクトが 304（オブジェクトが更新されていない）応答でした。
Total parameterized hits	パラメータ化されたコンテンツグループを持つポリシーを使用して処理され、キャッシュされたオブジェクトが見つかったキャッシュ要求の数。
Parameterized 304 hit ratio (%)	すべてのキャッシュヒットに対する、パラメータ化されたポリシーを使用して検出された 304（オブジェクトが更新されていない）応答の割合。

Counter	Specifies
Poll every time requests	［毎回ポーリング］が有効になっている場合、Citrix ADC アプライアンスは、保存されているオブジェクトを提供する前に、常にオリジナル・サーバーに問い合わせます。詳細は、「要求を受信するたびにオリジンサーバーをポーリングする」を参照してください。
Poll every time hits	[毎回ポール] メソッドを使用してキャッシュヒットが検出された回数。詳細は、「要求を受信するたびにオリジンサーバーをポーリングする」を参照してください。
Poll every time hit ratio (%)	[Poll Every Time] メソッドを使用したキャッシュヒットの割合。[Poll Every Time] を使用したキャッシュオブジェクトのすべての検索に対する、[Poll Every Time] を使用したキャッシュヒットの割合。詳細は、「要求を受信するたびにオリジンサーバーをポーリングする」を参照してください。
Maximum memory (KB)	キャッシュに割り当てられる Citrix ADC アプライアンスの最大メモリ容量。詳細は、「キャッシュ用のグローバル属性の構成」を参照してください。
Maximum memory active value (KB)	メモリが実際にキャッシュに割り当てられた後に設定されるメモリの最大量（アクティブな値）。詳細については、「Citrix ADC アプライアンスの統合キャッシュ機能をさまざまなシナリオで構成する方法」を参照してください。
Utilized memory (KB)	実際に使用されているメモリの量。
Memory allocation failures	応答をキャッシュに格納する目的でメモリを使用しようとして失敗した試行回数。
Largest response so far	キャッシュまたはオリジンサーバーのいずれかで検出され、クライアントに送信される最大のレスポンス（バイト単位）。
Cached objects	まだ完全にダウンロードされていないレスポンス、期限切れだがフラッシュされていないレスポンスを含む、キャッシュ内のオブジェクトの数。
Marker objects	マーカーオブジェクトは、応答がコンテンツグループの最大応答サイズまたは最小応答サイズを超えた場合、またはコンテンツグループの最小ヒット数をまだ受信していない場合に作成されます。

Counter	Specifies
Hits being served	キャッシュから提供されたヒット数。
Misses being handled	オリジンサーバーからフェッチされ、キャッシュに格納されてから処理されたレスポンス。保存可能なミスの数に近似する必要があります。保管不可能なミスは含まれません。

コマンドラインインターフェイスを使用してサマリーキャッシュ統計を表示するには

コマンドプロンプトで入力します。

```
stat cache
```

コマンドラインインターフェイスを使用して特定のキャッシュ統計を表示するには

コマンドプロンプトで入力します。

```
stat cache -detail

    > stat cache -detail
    Integrated Cache Statistics - Detail
    Integrated Cache Statistics - Summary
                                                Rate (/s)
                                                   Total
    Hits                                              0
                           0
    Misses                                            0
                           0
    Requests                                          0
                           0
    Hit ratio(%)                                     --
                           0
    Origin bandwidth saved(%)                        --
                           0
    Cached objects                                   --
                           0
    Marker objects                                   --
                           0
                                                Rate (/s)
                                                   Total
    Requests                                          0
                           0
```



```
16      Hit Statistics
17                                                  Rate (/s)
                                                       Total
18      Non-304 hits                                        0
                            0
19      304 hits                                            0
                            0
20      Sql hits                                            0
                            0
21      Hits                                                0
                            0
22      304 hit ratio(%)                                   --
                            0
23      Hit ratio(%)                                       --
                            0
24      Origin bandwidth saved(%)                          --
                            0
25
26      Byte Statistics
27                                                  Rate (/s)
                                                       Total
28      Bytes served by Citrix ADC                        648
           55379204
29      Bytes served by cache                               0
                            0
30      Byte hit ratio(%)                                  --
                            0
31      Compressed bytes from cache                         0
                            0
32      Miss Statistics
33                                                  Rate (/s)
                                                       Total
34      Storable misses                                     0
                            0
35      Non-storable misses                                 0
                            0
36      Misses                                              0
                            0
37      Revalidations                                       0
                            0
38      Successful revalidations                            0
                            0
39      Conversions to conditional req                      0
                            0
40      Storable miss ratio(%)                             --
```



```
                            0
41      Successful reval ratio(%)                          --
                            0
42      Flashcache Statistics
43                                                  Rate (/s)
                                                       Total
44      Expire at last byte                                 0
                            0
45      Flashcache misses                                   0
                            0
46      Flashcache hits                                     0
                            0
47
48      Invalidation Statistics
49                                                  Rate (/s)
                                                       Total
50      Parameterized inval requests                        0
                            0
51      Full inval requests                                 0
                            0
52      Inval requests                                      0
                            0
53
54      Parameterized Caching Statistics
55                                                  Rate (/s)
                                                       Total
56      Parameterized requests                              0
                            0
57      Parameterized non-304 hits                          0
                            0
58      Parameterized 304 hits                              0
                            0
59      Total parameterized hits                            0
                            0
60      Parameterized 304 hit ratio(%)                     --
                            0
61
62      Poll Every Time (PET) Statistics
63                                                  Rate (/s)
                                                       Total
64      Poll every time requests                            0
                            0
65      Poll every time hits                                0
                            0
66      Poll every time hit ratio(%)                       --
```



```
                                0
67      Memory Usage Statistics
68                                                      Total
69      Maximum memory(KB)                                  0
70      Maximum memory active value(KB)                     0
71      Utilized memory(KB)                                 0
72      Memory allocation failures                          0
73      Largest response so far(B)                          0
74      Cached objects                                      0
75      Marker objects                                      0
76      Hits being served                                   0
77      Misses being handled                                0
78       Done
79  <!--NeedCopy-->
```

GUI を使用してサマリーキャッシュ統計情報を表示するには

1. ページ上部の [ダッシュボード] タブをクリックします。
2. ウィンドウの [統合キャッシュ] セクションまで下にスクロールします。
3. 詳細な統計情報を表示するには、表の下部にある [More...] リンクをクリックします。

GUI を使用して特定のキャッシュ統計を表示するには

1. ページ上部の [レポート] タブをクリックします。
2. [組み込みレポート] で、[統合キャッシュ] を展開し、表示する統計情報を含むレポートをクリックします。
3. レポートをテンプレートとして保存するには、[名前を付けて保存] をクリックして、レポートに名前を付けます。保存されたレポートは、カスタムレポートの下に表示されます。

キャッシュ・パフォーマンスの向上

October 7, 2021

同じキャッシュデータに対する同時要求の処理、オリジンサーバーからのキャッシュされた応答の更新に関連する遅延の回避、キャッシュする価値のある十分な頻度で応答が要求されるようにするなど、統合キャッシュのパフォーマンスを向上させることができます。

フラッシュクラウドを減らす

フラッシュクラウドは、多くのユーザーが同時に同じデータを要求したときに発生します。オブジェクト全体がダウンロードされた後にのみヒットを提供するようにキャッシュを構成した場合、フラッシュクラウド内の要求はキャッシュミスになる可能性があります。

フラッシュクラウドを減少または排除するには、次の手法を使用します。

- **PREFETCH**: 期限切れになる前に肯定応答をリフレッシュして、古くなったり非アクティブになったりしないようにします。詳細については、「有効期限が切れる前に応答を更新する」を参照してください。
- キャッシュバッファリング: 応答全体がダウンロードされるのを待つのではなく、オリジンサーバーから応答ヘッダーを受信すると、複数のクライアントへの応答の提供を開始します。応答を同時にダウンロードできるクライアント数の唯一の制限は、使用可能なシステムリソースです。Citrix ADC アプライアンスは、ダウンロードを開始したクライアントがダウンロードが完了する前に停止した場合でも、応答をダウンロードして処理します。応答がキャッシュサイズを超えた場合、または応答がチャンク化された場合、キャッシュは応答の保存を停止しますが、クライアントへのサービスは中断されません。
- **Flash Cache**: Flash Cache はリクエストをキャッシュにキューイングし、一度に 1 つのリクエストだけがサーバーに到達できるようにします。

詳細については、「キャッシュへの要求のキューイング」を参照してください。

有効期限が切れる前にレスポンスを更新する

キャッシュされた応答が必要なときに常に最新の状態になるように、PREFETCH オプションは、計算された有効期限前に応答を更新します。プリフェッチ間隔は、最初のクライアント要求を受信した後に計算されます。それ以降、Citrix ADC アプライアンスは、PREFETCH パラメータで設定した時間間隔でキャッシュされた応答を更新します。

この設定は、要求間で頻繁に更新されるデータに便利です。否定的な応答（404 メッセージなど）には適用されません。

コマンドラインインターフェイスを使用してコンテンツグループのプリフェッチを構成するには

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -prefetch YES [-prefetchPeriod <seconds> | -
prefetchPeriodMilliSec <milliseconds>] [-prefetchMaxPending <positiveInteger
>]
```

*GUI を使用してコンテンツ・グループのプリフェッチを構成するには

最適化に移動 > 統合キャッシング > コンテンツグループ、および コンテンツグループを選択します。

[その他] タブの [フラッシュ群集とプリフェッチ] グループで、[プリフェッチ] オプションを選択し、[保留中のプリフェッチの間隔と最大数] テキストボックスに値を指定します。

リクエストをキャッシュにキューイングする

Flash Cache オプションは、同時に到着するリクエスト（フラッシュ群集）をキューに入れ、レスポンスを取得して、リクエストがキューにあるすべてのクライアントに配信します。この処理中に応答がキャッシュ不能になると、Citrix ADC アプライアンスはキャッシュからの応答の提供を停止し、代わりにキューに入れられたクライアントに対するオリジンサーバーの応答を提供します。応答が利用できない場合、クライアントはエラーメッセージを受け取ります。

フラッシュキャッシュはデフォルトで無効になっています。同じコンテンツグループで毎回ポーリング（PET）とフラッシュキャッシュを有効にすることはできません。

Flash Cache の欠点の 1 つは、サーバーがエラー（たとえば、迅速に修正された 404）で応答した場合、そのエラーが待機中のクライアントにファンアウトされることです。

> 注:
> Flash Cache が有効になっている場合、Citrix ADC アプライアンスは、クライアントリクエストの Accept-Encoding ヘッダーとレスポンスの Content-Encoding ヘッダーを正しく一致させることができない場合があります。Citrix ADC アプライアンスは、これらのヘッダーが一致し、誤ってヒットしたと見なすことができます。回避策として、適切な Accept-Encoding ヘッダーを持たないクライアントにヒットを提供できないように、統合キャッシュポリシーを構成できます。

コマンドラインインターフェイスを使用して Flash Cache を有効にするには

コマンドプロンプトで入力します。

```
set cache contentgroup <contentGroupName> -flashcache yes
```

GUI を使用してフラッシュキャッシュを有効にするには

[最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。

[その他] タブの [フラッシュ群集とプリフェッチ] グループで、[プリフェッチ] オプションを選択します。

クライアントがダウンロードを停止した後にレスポンスをキャッシュする

Quick Abort パラメータを設定すると、応答がキャッシュ内に格納される前にクライアントが要求を停止した場合でも、応答のキャッシュを続行できます。

ダウンロードされた応答サイズがクイックアボートサイズ以下の場合、Citrix ADC アプライアンスは応答のダウンロードを停止します。Quick Abort パラメータを 0 に設定すると、すべてのダウンロードが停止します。

コマンドラインインターフェイスを使用してクイックアボートサイズを構成するには

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -quickAbortSize <integerInKBytes>
```

GUI を使用してクイックアボートサイズを設定するには

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [メモリ] タブで、[クイック中止:] テキストボックス以上の場合はキャッシュを続行する] で関連する値を設定します。

キャッシュする前に最低限のサーバーヒット数を要求する

応答がキャッシュされる前に、オリジンサーバーで検出される必要のある最小回数を設定できます。キャッシュメモリがすぐにいっぱいになり、ヒット率が予想よりも低い場合は、最小ヒット数を増やすことを検討する必要があります。

最小ヒット数のデフォルト値は 0 です。この値は、最初のリクエストの後にレスポンスをキャッシュします。

コマンドラインインターフェイスを使用してキャッシュする前に必要な最小ヒット数を構成するには

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -minhits <positiveInteger>
```

GUI を使用してキャッシュする前に必要な最小ヒット数を設定するには

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [メモリ] タブで、ヒットがテキストボックスより少ない場合は、[キャッシュしない] で関連する値を設定します。

パフォーマンスの最適化の例

この例では、クライアントが株価にアクセスします。株価は非常に動的です。統合キャッシュは、オリジンサーバーに複数のリクエストを送信せずに、同時クライアントに対して同じ株価を提供するように設定します。株価はクライアントにダウンロードされた後に期限切れになり、次のリクエストはオリジンサーバーからフェッチされます。これにより、見積もりは常に最新の状態になります。

次のタスクの概要では、株価アプリケーションのキャッシュを構成する手順について説明します。

株価アプリケーションのキャッシュを構成する

株価のコンテンツグループの作成

詳細については、「コンテンツグループについて」を参照してください。

このコンテンツグループに対して以下を設定します。

1. [有効期限の方法] タブで、[応答完了後に失効する] チェックボックスをオンにします。
2. 「その他」タブで「**Flash Cache** 」チェックボックスをオンにして、「作成」をクリックします。
3. 株価をキャッシュするキャッシュポリシーを追加します。

詳細は、「統合キャッシュでのポリシーの構成」を参照してください。

ポリシーに対して以下を設定します。

1. [アクション**]** リストと **[**グループに格納**]** リストで、[**CACHE**] を選択し、前の手順で定義したグループを選択します。
2. [追加] をクリックし、[式の 追加] ダイアログボックスで、株価見積依頼を識別する式を設定します。例:http.req.url.contains (「cgi-bin/stock-quote.pl」)
3. ポリシーをアクティブ化します。

詳細については、「統合キャッシュポリシーのグローバルバインド」を参照してください。この例では、このポリシーを request-time 上書き処理にバインドし、プライオリティを低い値に設定します。

Cookie、ヘッダー、およびポーリングを構成する

December 7, 2021

このトピックでは、Cookie、HTTP ヘッダー、およびオリジンサーバーのポーリングを管理するキャッシュを構成する方法について説明します。これには、キャッシュが文書化された標準から外れるようなデフォルトの動作の変更、キャッシュ可能なコンテンツがキャッシュに保存されなくなる可能性がある HTTP ヘッダーのオーバーライド、更新されたコンテンツについてオリジンに常にポーリングするようにキャッシュを設定することが含まれます。

標準からのキャッシュ動作の相違について

デフォルトでは、統合キャッシュは次の RFC 標準に準拠しています。

- RFC 2616「HTTP HTTP/1.1」
- RFC 2617「HTTP 認証: 基本認証とダイジェストアクセス認証」で説明されているキャッシュ動作
- RFC 2965「HTTP 状態管理メカニズム」で説明されているキャッシュ動作

組み込みのポリシーと [Default] コンテンツグループ属性により、これらの標準のほとんどに準拠していることが保証されます。

デフォルトの統合キャッシュの動作は、以下のように仕様とは違います。

- ヴァリヘッダは限定的にサポートされています。デフォルトでは、ヴァリヘッダーを含むレスポンスは、圧縮されない限りキャッシュ不可とみなされます。圧縮された応答には、コンテンツエンコーディング:gzip、コンテンツエンコーディング:deflate、またはコンテンツエンコーディング:pack200-gzip が含まれており、Vary: Accept-encoding ヘッダーが含まれていてもキャッシュ可能です。
- 統合キャッシュは、ヘッダーの cache-control (no-cache および cache-control: private) の値を無視します。たとえば、cache-control: no-cache= “set-cookie” を含むレスポンスは、レスポンスに Cache-Control: no-cache が含まれているかのように扱われます。デフォルトでは、レスポンスはキャッシュされません。
- イメージレスポンスに set-cookie または set-cookie2 ヘッダーが含まれている場合や、イメージリクエストに Cookie ヘッダーが含まれている場合でも、イメージ（content-type = image/*）は常にキャッシュ可能と見なされます。統合キャッシュは、set-cookie と set-cookie2 ヘッダーをキャッシュする前にレスポンスから削除します。これは RFC 2965 とは違います。RFC 準拠の動作は次のように設定できます。

```
1 add cache policy rfc_compliant_images_policy -rule ”http.res.header.set
    -cookie2.exists || http.res.header.set-cookie.exists” -action
    NOCACHE
2
3
4 bind cache global rfc_compliant_images_policy -priority 100 -type
    REQ_OVERRIDE
```



```
<!--NeedCopy-->
```

- リクエスト内の次のキャッシュ制御ヘッダーは、RFC 準拠のキャッシュにオリジンサーバーからキャッシュされたレスポンスをリロードするように強制します。

```
Cache-control: max-age=0

Cache-control: no-cache
```

サービス拒否攻撃を防ぐため、この動作はデフォルトではありません。

- デフォルトでは、キャッシュモジュールは、レスポンスヘッダーの状態がそうでない場合を除き、レスポンスはキャッシュ可能であるとみなします。この動作を RFC 2616 に準拠させるには、すべてのコンテンツグループに対して-weakPosRelExpiryと-weakNegResExpiryを 0 に設定します。

レスポンスからクッキーを削除する

クッキーはユーザーに合わせてカスタマイズされることが多く、通常はキャッシュすべきではありません。Remove Response Cookiesパラメーターは、レスポンスをキャッシュする前にSet-Cookie and Set-Cookie2ヘッダーを削除します。既定では、コンテンツグループのRemove Response Cookiesオプションでは、Set-Cookieヘッダーまたは Set-Cookie2ヘッダーを含む応答のキャッシュは禁止されています。

> 注:
> イメージがキャッシュされる場合、組み込みの動作では、コンテンツグループがどのように設定されていても、キャッシュの前にSet-Cookieヘッダーと Set-Cookie2ヘッダーが削除されます。
>
> 画像などの埋め込みレスポンスを保存するすべてのコンテンツグループに対して、デフォルトRemove Response Cookiesを使用することをお勧めします。

コマンドラインインターフェイスを使用してコンテンツグループに対してRemove Response Cookiesを設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -removeCookies YES
```

Citrix ADC GUI を使用してコンテンツグループの応答 **Cookie** の削除を構成する

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [その他] タブの [設定] グループで、[レスポンス Cookie の削除] オプションを選択します。

レスポンスタイムに **HTTP** ヘッダを挿入する

統合キャッシュは、キャッシュリクエストから返されるレスポンスに HTTP ヘッダーを挿入できます。Citrix ADC アプライアンスは、キャッシュミスによる応答のヘッダーを変更しません。

次の表に、レスポンスに挿入できるヘッダーについて説明します。

ヘッダー	仕様
年齢	オリジンサーバーでレスポンスが生成された時間から計算された、レスポンスの経過時間（秒）を提供します。デフォルトでは、キャッシュはキャッシュから提供されるすべてのレスポンスに Age ヘッダーを挿入します。
経由で	要求または応答の開始点と終了点の間にあるプロトコルと受信者を一覧表示します。Citrix ADC アプライアンスは、キャッシュから処理されるすべての応答に Via ヘッダーを挿入します。挿入されるヘッダーのデフォルト値は、NS-CACHE-10.0Citrix ADC IP アドレスの最後のオクテットです。詳しい情報については、「キャッシュ用のグローバル属性の設定」を参照してください。
Tag	キャッシュは、Last-Modified ヘッダーとTagヘッダーを使用したレスポンスの検証をサポートし、レスポンスが古くなっているかどうかを判断します。キャッシュは、レスポンスをキャッシュし、オリジンサーバーが独自のTagヘッダーを挿入していない場合にのみ、Tagをレスポンスに挿入します。Tag 値は任意の一意の数値です。レスポンスのTag値は、オリジンサーバーから更新されると変更されますが、サーバーが 304 (object Not updated) レスポンスを送信した場合は同じままです。動的コンテンツはキャッシュ不可と見なされるため、通常、オリジンサーバーは動的コンテンツのバリデータを生成しません。この動作はオーバーライドできます。Tag ヘッダーを挿入すると、キャッシュは完全なレスポンスを処理しないことが許可されます。代わりに、ユーザーエージェントは、統合キャッシュによって最初に送信された動的応答をキャッシュする必要があります。ユーザーエージェントにレスポンスを強制的にキャッシュさせるには、統合キャッシュを設定してTagヘッダーを挿入し、オリジンが提供する Cache-Control ヘッダーを置き換えます。

ヘッダー	仕様
キャッシュコントロール	Citrix ADC アプライアンスは通常、オリジンサーバーから配信される応答のキャッシュ可能性ヘッダーを変更しません。オリジンサーバーがキャッシュ不可とラベル付けされた応答を送信すると、Citrix ADC アプライアンスが応答をキャッシュしても、クライアントは応答をキャッシュ不可として扱います。動的レスポンスをユーザーエージェントにキャッシュするには、オリジンサーバーの Cache-Control ヘッダーを置き換えることができます。これは、ユーザーエージェントとその他の介在するキャッシュにのみ適用されます。統合キャッシュには影響しません。

ヘッダー	仕様
年齢	オリジンサーバーでレスポンスが生成された時間から計算された、レスポンスの経過時間（秒）を提供します。デフォルトでは、キャッシュはキャッシュから提供されるすべてのレスポンスに Age ヘッダーを挿入します。
経由で	要求または応答の開始点と終了点の間にあるプロトコルと受信者を一覧表示します。Citrix ADC アプライアンスは、キャッシュから処理されるすべての応答に Via ヘッダーを挿入します。挿入されるヘッダーのデフォルト値は「NS-CACHE-9.2：Citrix ADC IP アドレスの最後のオクテット」です。詳しい情報については、「キャッシュ用のグローバル属性の設定」を参照してください。

ヘッダー	仕様
Tag	キャッシュは、Last-Modified ヘッダーと Tag ヘッダーを使用した応答検証をサポートし、応答が古くなっているかどうかを判断します。キャッシュは、レスポンスをキャッシュし、オリジンサーバーが独自のTagヘッダーを挿入していない場合にのみ、Tagをレスポンスに挿入します。Tag 値は任意の一意の数値です。レスポンスのTag値は、オリジンサーバーから更新されると変更されますが、サーバーが 304 (object Not updated) レスポンスを送信した場合は同じままです。動的コンテンツはキャッシュ不可と見なされるため、通常、オリジンサーバーは動的コンテンツのバリデータを生成しません。この動作はオーバーライドできます。Tag ヘッダーを挿入すると、キャッシュは完全なレスポンスを処理しないことが許可されます。代わりに、ユーザーエージェントは、統合キャッシュによって最初に送信された動的応答をキャッシュする必要があります。ユーザーエージェントにレスポンスを強制的にキャッシュさせるには、統合キャッシュを設定してTagヘッダーを挿入し、オリジンが提供する Cache-Control ヘッダーを置き換えます。
キャッシュコントロール	Citrix ADC アプライアンスは通常、オリジンサーバーから配信される応答のキャッシュ可能性ヘッダーを変更しません。オリジンサーバーがキャッシュ不可とラベル付けされた応答を送信すると、Citrix ADC アプライアンスが応答をキャッシュしても、クライアントは応答をキャッシュ不可として扱います。動的レスポンスをユーザーエージェントにキャッシュするには、オリジンサーバーの Cache-Control ヘッダーを置き換えることができます。これは、ユーザーエージェントとその他の介在するキャッシュにのみ適用されます。統合キャッシュには影響しません。

年齢、経由、またはタグヘッダーの挿入

次の手順では、Age、Via、およびETagヘッダーを挿入する方法について説明します。

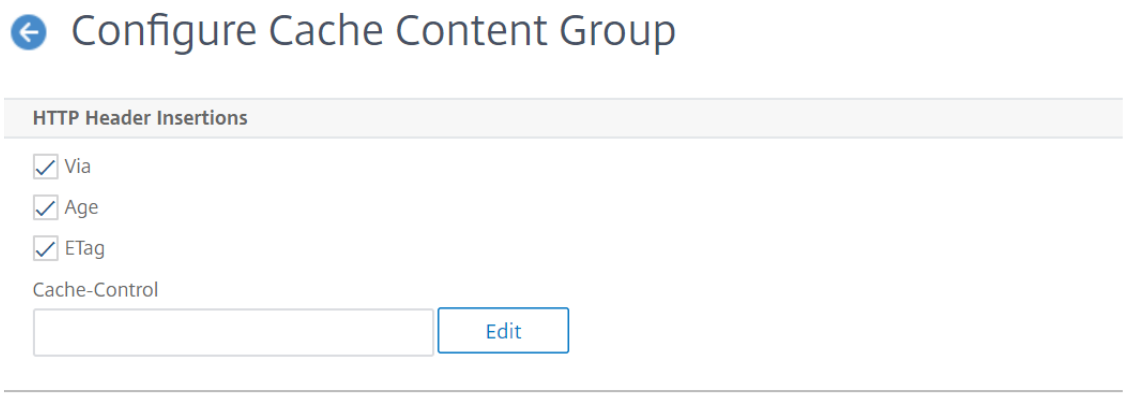
Citrix ADC コマンドインターフェイスを使用して、**Age**、**Via**、または **Etag** ヘッダーを挿入します。

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -insertVia YES -insertAge YES -insertETag YES
```

Citrix ADC GUI を使用して、**Age**、**Via**、または **Etag** ヘッダーを構成します

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [その他] タブの [HTTP ヘッダー挿入] グループで、必要に応じて [経由]、[経過日数]、または [**ETag**] オプションを選択します。
3. その他のヘッダータイプの値は自動的に計算されます。Via の値は、キャッシュのメイン設定で設定します。



キャッシュコントロールヘッダを挿入する

統合キャッシュが、オリジンサーバーが挿入した Cache-Control ヘッダーを置き換える場合、Expires ヘッダーも置き換えられます。新しい Expires ヘッダーには、過去の有効期限が含まれています。これにより、HTTP/1.0 クライアントと (Cache-Control ヘッダーを認識しない) キャッシュがコンテンツをキャッシュしないようにします。

Citrix ADC コマンドインターフェイスを使用してキャッシュ制御ヘッダーを挿入する

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -cacheControl <value>
```

Citrix ADC GUI を使用してキャッシュ制御ヘッダーを挿入する

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、
   a) [有効期限方法] をクリックして、ヒューリスティックおよび既定の有効期限設定をクリアし、[コンテンツの有効期限が切れるまでの期間] テキストボックスで適切な値を設定します。
   b) [その他] タブをクリックし、[Cache-Control] テキストボックスに挿入するヘッダーを入力します。または、[Configure] をクリックして、キャッシュされたレスポンスに Cache-Control ディレクティブを設定します。

リクエスト内のキャッシュコントロールとプラグマヘッダーを無視する

デフォルトでは、キャッシュモジュールは Cache-Control ヘッダーと Pragma ヘッダーを処理します。Cache-Control ヘッダー内の次のトークンは、RFC 2616 の説明に従って処理されます。

- 最大年齢
- 最大失効しました
- キャッシュされた場合のみ有る
- キャッシュなし

リクエスト内の Pragma: no-cache ヘッダーは、Cache-Control: no-cache ヘッダーと同じように扱われます。

Cache-Control ヘッダーと Pragma ヘッダーを無視するようにキャッシュモジュールを構成すると、Cache-Control: No-Cache ヘッダーを含む要求により、Citrix ADC アプライアンスはオリジンサーバーから応答を取得しますが、キャッシュされた応答は更新されません。キャッシュモジュールが Cache-Control ヘッダーと Pragma ヘッダーを処理すると、キャッシュされたレスポンスが更新されます。

次の表に、これらのヘッダーのさまざまな設定と [ブラウザのリロード要求を無視] 設定の影響をまとめています。

キャッシュコントロールとプラグマヘッダーを無視するための設定	ブラウザのリロード要求を無視する設定	結果
はい	はいまたはいいえ	Cache-Control: no-cache ディレクティブを含め、クライアントからの Cache-Control ヘッダーと Pragma ヘッダーは無視してください。
いいえ	はい	Cache-Control: no-cache ヘッダーはキャッシュミスを生成しますが、すでにキャッシュにあるレスポンスは更新されません。
いいえ	いいえ	Cache-Control: no-cache ヘッダーを含むリクエストはキャッシュミスを引き起こし、保存されたレスポンスはリフレッシュされます。

コマンドラインインターフェイスを使用してリクエストの Cache-Control および Pragma ヘッダーを無視するには

コマンドプロンプトで入力します。

```
set cache contentgroup <name> -ignoreReqCachingHdrs YES
```

コマンドラインインターフェイスを使用してブラウザのリロード要求を無視するには

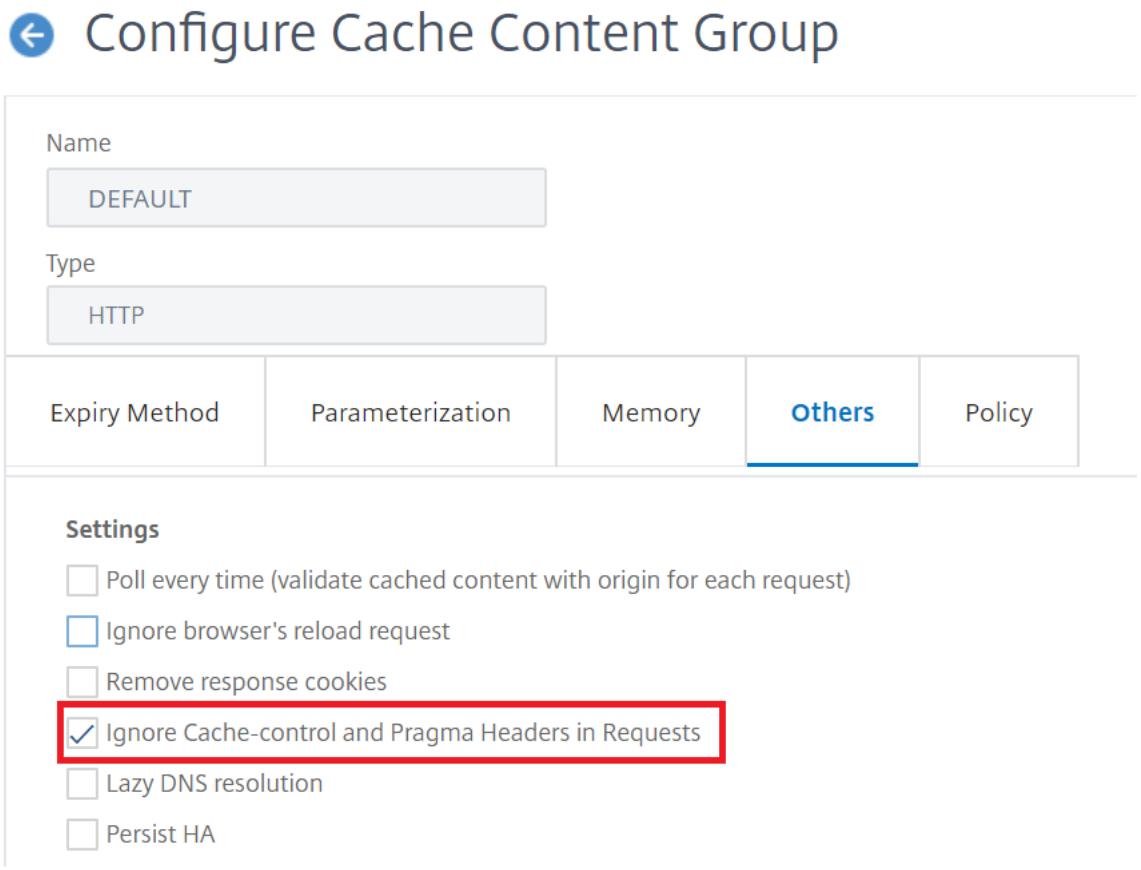
コマンドプロンプトで入力します。

```
set cache contentgroup <name> -ignoreReloadReq NO
```

> メモ:
> デフォルトでは、-ignoreReloadReq パラメータは YES に設定されています。

GUI を使用してリクエストの **Cache-Control** ヘッダーと **Pragma** ヘッダーを無視する

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [その他] タブの [設定] グループで、[****** 要求内のキャッシュコントロールとプラグマヘッダーを無視 ******] オプションを選択します。



Cache-Control ヘッダーを無視するポリシーの例:

次の例では、Content-type: image/* を含むレスポンスを、レスポンスの Cache-Control ヘッダーに関係なく、キャッシュするようにリクエスト時間オーバーライドポリシーを設定します。

image/* を含むすべてのレスポンスをキャッシュするようにリクエスト時間オーバーライドポリシーを設定するには

[すべて無効化] オプションを使用してキャッシュをフラッシュします。

新しいキャッシュポリシーを設定し、そのポリシーを特定のコンテンツグループに向けます。詳しい情報については、「統合キャッシュのポリシーの設定」を参照してください。

「リクエストでの Cache-Control ヘッダーと Pragma ヘッダーを無視する」の説明に従って、ポリシーが使用するコンテンツグループが Cache-Control ヘッダーを無視するように設定されていることを確認します。

ポリシーをリクエスト時間上書きポリシーバンクにバインドします。

詳細については、「統合キャッシュポリシーのグローバルバインド」トピックを参照してください。

リクエストを受信するたびにオリジンサーバーをポーリングします

保存された応答を提供する前に、常にオリジンサーバーを参照するように Citrix ADC アプライアンスを構成できます。これは、毎回ポーリング (PET) と呼ばれます。Citrix ADC アプライアンスがオリジンサーバーを参照し、PET 応答の有効期限が切れていない場合、オリジンサーバーからの完全な応答によってキャッシュされたコンテンツは上書きされません。このプロパティは、クライアント固有のコンテンツを提供する場合に便利です。

PET 応答の有効期限が切れると、元のサーバーから最初の完全な応答が到着すると、Citrix ADC アプライアンスは PET 応答を更新します。

毎回ポーリング (PET) 関数は次のように機能します。

Tag または Last-Modified ヘッダーの形式のバリデータを含むキャッシュされたレスポンスの場合、レスポンスの有効期限が切れると、自動的に PET とマークされ、キャッシュされます。

コンテンツグループに PET を設定できます。

コンテンツグループを PET として設定すると、コンテンツグループ内のすべての応答に PET とマークされます。PET コンテンツグループは、バリデータを持たないレスポンスを保存できます。自動的に PET とマークされた応答は、常に期限切れとなります。PET コンテンツグループに属する応答は、コンテンツグループの設定方法に基づいて、遅延後に期限切れになることがあります。

ポーリングは、次の 2 種類のリクエストに影響します。

- 条件付きリクエスト: クライアントは、応答が最新のコピーであることを確認するために、条件付きリクエストを発行します。キャッシュされた PET 応答に対するユーザーエージェントリクエストは、常に条件付きリクエストに変換され、オリジンサーバーに送信されます。条件付きリクエストでは、If-Modified-Since ヘッダーまたは If-None-Match ヘッダーにバリデータがあります。If-Modified-SINSE ヘッダーには、最終変更ヘッダーからの時刻が含まれます。If-None-Match ヘッダーには、レスポンスの Tag ヘッダー値が含まれます。クライアントのレスポンスのコピーが新鮮な場合、オリジンサーバーは 304 Not Modified と応答します。コピーが古くなっている場合、条件付き応答では 200 OK が生成され、応答全体が格納されます。
- 非条件付きリクエスト: 非条件リクエストでは、レスポンス全体を含む 200 OK しか生成できません。

オリジンサーバーのレスポンス	操作（アクション）
完全な回答を送信	オリジンサーバーはレスポンスをそのままクライアントに送信します。キャッシュされたレスポンスの有効期限が切れた場合、そのレスポンスはリフレッシュされます。
304 変更されていない	304 レスポンスのヘッダー値がキャッシュされたレスポンスとマージされ、キャッシュされたレスポンスがクライアントに提供されます。Date、Expires、Age、Cache-Control ヘッダー Max-Age、および S-Maxage トークン
401 無許可、400 不正な要求、405 メソッドは許可されない、406 は受け入れられない、407 プロキシ認証が必要	オリジンの応答は、そのままクライアントに提供されます。キャッシュされたレスポンスは変更されません。
その他のエラー応答 (404 Not Found など)	オリジンの応答は、そのままクライアントに提供されます。キャッシュされたレスポンスは削除されます。

> 注:
> Poll Every Time パラメータは、影響を受けるレスポンスを保存不可として扱います。

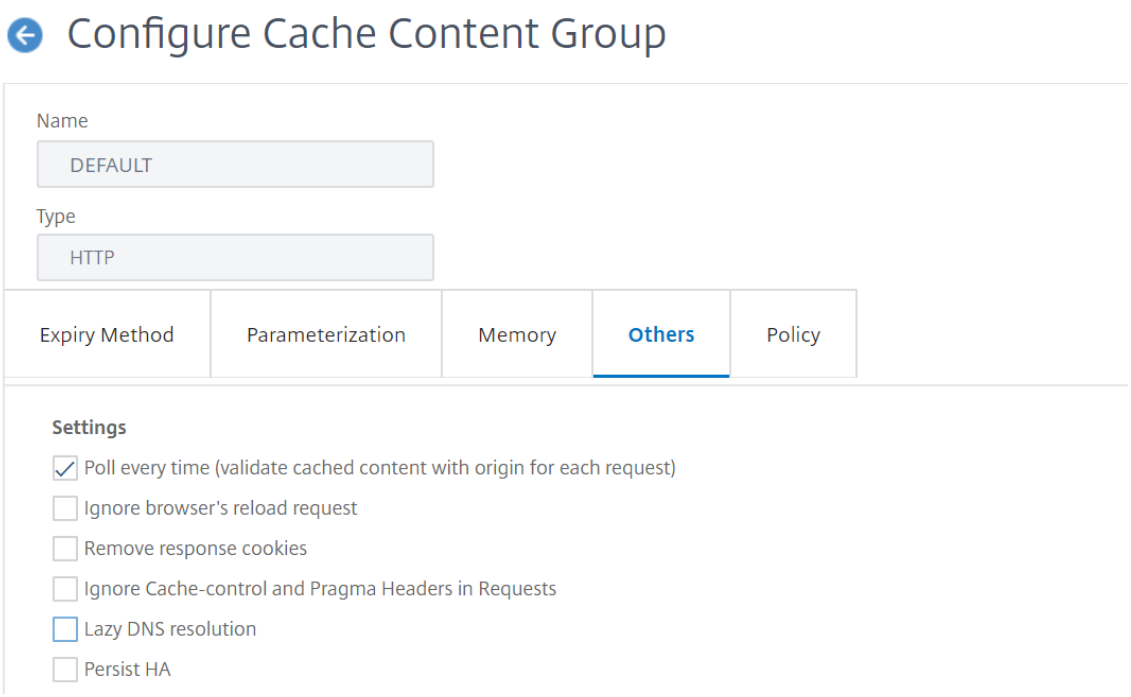
コマンドラインインターフェイスを使用してポーリングを毎回設定するには

コマンドプロンプトで入力します。

```
add cache contentgroup <contentGroupName> -pollEveryTime YES
```

GUI を使用してポーリングする

1. [最適化] > [統合キャッシュ] > [コンテンツグループ] に移動し、コンテンツグループを選択します。
2. [その他] タブの [設定] グループで、[毎回ポーリング (リクエストごとにキャッシュされたコンテンツをオリジンで検証)] オプションを選択します。



PET およびクライアント固有のコンテンツ

PET 機能を使用すると、クライアント向けにコンテンツを確実にカスタマイズできます。たとえば、複数の言語でコンテンツを提供する Web サイトでは、Accept-Language リクエストヘッダーを調べて、配信するコンテンツの言語を選択します。英語が主言語である多言語の Web サイトでは、すべての英語コンテンツを PET コンテンツグループにキャッシュできます。これにより、すべてのリクエストがオリジンサーバーに送信され、レスポンスの言語が決定されます。応答が英語で、コンテンツが変更されていない場合、オリジンサーバーは 304 Not Modified をキャッシュに提供できます。

次に、英語による応答を PET コンテンツグループにキャッシュし、キャッシュ内の英語の応答を識別する名前付き式を設定し、このコンテンツグループと名前付き式を使用するポリシーを設定するコマンドの例を示します。太字は強調のために使われます。

```
1 add cache contentgroup EnglishLanguageGroup -pollEveryTime YES
2 add expression containsENExpression  - rule ”http.res.header(\\”Content-
     Language\\”).contains(\\”en\\”)”
3 add cache policy englishPolicy -rule containsENExpression -action CACHE
      -storeInGroup englishLanguageGroup
4 bind cache policy englishPolicy -priority 100 -precedeDefRules NO
5 <!--NeedCopy-->
```


PET と認証、承認、監査

Outlook Web Access (OWA) は、PET の恩恵を受ける動的に生成されたコンテンツの好例です。すべてのメールレスポンス (*.EML オブジェクト) には、PET ETag レスポンスとして保存できるバリデータがあります。

メールレスポンスのリクエストはすべて、レスポンスがキャッシュされていても、オリジンサーバーに送信されます。オリジンサーバーは、リクエスタが認証され、承認されているかどうかを判断します。また、オリジンサーバーに応答が存在するかどうかも検証します。すべての結果が肯定的である場合、オリジンサーバーは 304 Not Modified レスポンスを送信します。

統合キャッシュをフォワードプロキシとして構成する

October 7, 2021

統合キャッシュは、他の Citrix ADC アプライアンスまたは他のタイプのキャッシュサーバーに要求を渡すフォワードプロキシデバイスとして機能します。統合キャッシュをフォワードプロキシとして構成するには、キャッシュサーバの IP アドレスを指定します。転送プロキシを構成した後、Citrix ADC アプライアンスは、統合されたキャッシュを使用する代わりに、構成済みの IP アドレスを含む要求をキャッシュサーバーに送信します。

コマンドラインインターフェイスを使用して Citrix ADC をフォワードキャッシュプロキシとして構成するには

コマンドプロンプトで入力します。

```
add cache forwardProxy <IPAddress> <port>
```

GUI を使用して Citrix ADC をフォワードキャッシュプロキシとして構成するには

1. [最適化] > [統合キャッシュ] > [フォワードプロキシ] に移動し、IP アドレスとポート番号を指定してフォワードプロキシを追加します。

統合キャッシュのデフォルト設定

October 7, 2021

Citrix ADC 統合キャッシュ機能は、デフォルトのコンテンツグループのデフォルト設定と初期設定を備えた組み込みポリシーを提供します。このセクションの情報は、組み込みポリシーと Default コンテンツグループのパラメータを定義します。

デフォルトのキャッシュ・ポリシー

統合キャッシュには、ポリシーが組み込まれています。Citrix ADC アプライアンスは、次のセクションで説明するように、特定の順序でポリシーを評価します。

これらの組み込みポリシーは、要求時間上書きまたは応答時間上書きポリシーバンクにバインドされたユーザー定義ポリシーで上書きできます。

> 注:
> リリース 9.0 より前のポリシーを構成し、ポリシーのバインド時に-precedeDefRules パラメータを指定した場合、移行中に上書き時のバインドポイントに自動的に割り当てられます。

デフォルトポリシーの表示

組み込みのポリシー名は、アンダースコア（_）で始まります。組み込みポリシーは、コマンドラインおよび管理コンソールから show cache policy コマンドを使用して表示できます。

デフォルトのリクエストポリシー

次の組み込み要求時間ポリシーを上書きするには、新しいポリシーを設定し、それらを要求時間上書き処理ポイントにバインドします。次のポリシーでは、MAY_NOCACHE アクションは、応答時にユーザー設定または組み込みの CACHE ディレクティブがある場合にのみトランザクションがキャッシュされることを規定していることに注意してください。

次のポリシーは、_reqBuiltinDefaults ポリシーラベルにバインドされています。これらの項目は、優先順に表示されます。

GET 以外のメソッドを使用するリクエストに対するレスポンスをキャッシュしないでください。

ポリシー名は _nonGetReq です。次に、ポリシールールを示します。

```
!HTTP.REQ.METHOD.eq(GET)
```

ヘッダー値が If-Match または If-Unmodified-Since を含むリクエストに対して NOCACHE アクションを設定します。

ポリシー名は _advanced 条件付き必須です。次に、ポリシールールを示します。

```
HTTP.REQ.HEADER("If-Match").EXISTS || HTTP.REQ.HEADER("If-Unmodified-Since")
).EXISTS
```

を設定します MAY_NOCACHE 次のヘッダー値を持つ要求に対するアクション：Cookie、Authorization、Proxy-authorization、または NTLM または Negotiate ヘッダーを含む要求。

ポリシー名は _personalizedReq です。次に、ポリシールールを示します。

```
HTTP.REQ.HEADER("Cookie").EXISTS || HTTP.REQ.HEADER("Authorization").EXISTS
 || HTTP.REQ.HEADER("Proxy-Authorization").EXISTS || HTTP.REQ.IS_NTLM_OR_NEGOTIATE
```


デフォルトのレスポンス・ポリシー

次のデフォルトの応答時間ポリシーを上書きするには、新しいポリシーを設定し、応答時間の上書き処理ポイントにバインドします。

次のポリシーは、**_resBuiltinDefaults** ポリシーラベルにバインドされ、リストされている順序で評価されます。

1. HTTP 応答がタイプ 200、304、307、203 の場合、またはタイプが 400 ～499 の場合、または 300 ～302 の場合以外は HTTP 応答をキャッシュしないでください。

   ポリシー名は **_uncacheableStatusRes** です。次に、ポリシールールを示します。

   ```
   !((HTTP.RES.STATUS.EQ(200))|| (HTTP.RES.STATUS.EQ(304))|| (HTTP.RES.
   STATUS.BETWEEN(400,499))|| (HTTP.RES.STATUS.BETWEEN(300, 302))|| (HTTP.
   RES.STATUS.EQ(307))|| (HTTP.RES.STATUS.EQ(203)))
   ```

2. Accept-Encoding 以外の値を持つ Vary ヘッダーがある場合は、HTTP レスポンスをキャッシュしないでください。

   圧縮モジュールは、Vary: Accept_Encoding ヘッダーを挿入します。この式の名前は、**_unacheable-VaryRes** です。次に、ポリシールールを示します。

   ```
   ((HTTP.RES.HEADER("Vary").EXISTS)&& ((HTTP.RES.HEADER("Vary").INSTANCE
   (1).LENGTH > 0)|| (!HTTP.RES.HEADER("Vary").STRIP_END\\_WS.SET_TEXT_MODE
   (IGNORECASE).eq("Accept-Encoding"))))
   ```

3. Cache-Control ヘッダー値が [キャッシュしない]、[ストアしない]、または [プライベート] の場合、またはキャッシュ制御ヘッダーが有効でない場合は、レスポンスをキャッシュしないでください。

   ポリシー名は **_uncacheable** キャッシュコントロールです。次に、ポリシールールを示します。

   ```
   ((HTTP.RES.CACHE\\_CONTROL.IS\\_PRIVATE)|| (HTTP.RES.CACHE\\_CONTROL.IS
   \\_NO\\_CACHE)|| (HTTP.RES.CACHE\\_CONTROL.IS\\_NO\\_STORE)|| (HTTP.RES
   .CACHE\\_CONTROL.IS\\_INVALID))
   ```

4. キャッシュ制御ヘッダーに次のいずれかの値がある場合、応答をキャッシュします。パブリック、再確認する必要があります、プロキシ再確認、最大有効期間、S-Maxage。

   ポリシー名は _ キャッシュ可能キャッシュコントロールです。次に、ポリシールールを示します。

   ```
   ((HTTP.RES.CACHE_CONTROL.IS_PUBLIC)|| (HTTP.RES.CACHE_CONTROL.IS_MAX_AGE
   )|| (HTTP.RES.CACHE_CONTROL.IS_MUST_REVALIDATE)|| (HTTP.RES.CACHE_CONTROL
   .IS_PROXY_REVALIDATE)|| (HTTP.RES.CACHE_CONTROL.IS_S_MAXAGE))
   ```

5. Pragma ヘッダーを含むレスポンスをキャッシュしないでください。

   ポリシーの名前は **_uncacheable** プラグマレです。次に、ポリシールールを示します。

   ```
   HTTP.RES.HEADER("Pragma").EXISTS
   ```


6. Expires ヘッダーを含むレスポンスをキャッシュします。

   ポリシーの名前は _ キャッシュ可能有効期限です。次に、ポリシールールを示します。

   ```
   HTTP.RES.HEADER("Expires").EXISTS
   ```

7. レスポンスに Image の値を持つ Content-Type ヘッダーが含まれている場合は、ヘッダー内の Cookie をすべて削除してキャッシュします。

   ポリシーの名前は **_imageRes** です。次に、ポリシールールを示します。

   ```
   HTTP.RES.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).STARTSWITH("image/")
   ```

   このポリシーで機能するように、次のコンテンツグループを構成できます。

   ```
   add cache contentgroup nocookie -group -removeCookies YES
   ```

8. Set-Cookie ヘッダーを含むレスポンスをキャッシュしないでください。

   ポリシーの名前は、_ パーソナライズされた **Res** です。次に、ポリシールールを示します。

HTTP.RES.HEADER("Set-Cookie").EXISTS	HTTP.RES.HEADER("Set-Cookie2").EXISTS

デフォルトポリシーの制限

次の組み込み要求時間ポリシーをユーザー定義ポリシーで上書きすることはできません。

これらのポリシーは、優先順に表示されます。

1. 対応する HTTP リクエストに GET または POST メソッドがない場合は、応答をキャッシュしないでください。
2. HTTP 要求 URL の長さとホスト名が 1744 バイトを超える場合は、要求に対する応答をキャッシュしないでください。
3. If-Match ヘッダーを含むリクエストのレスポンスをキャッシュしないでください。
4. If-Unmodified-Since ヘッダーを含むリクエストをキャッシュしないでください。

> 注:
> これは、変更-以来ヘッダーとは異なります。

1. サーバーが期限切れヘッダーを設定していない場合は、応答をキャッシュしないでください。

次の組み込み応答時間ポリシーは上書きできません。これらのポリシーは、リストされている順に評価されます。

1. HTTP 応答ステータスコードが 201、202、204、205、または 206 の応答をキャッシュしないでください。
2. ステータスコード 403、404、および 410 を除き、4xx の HTTP 応答ステータスコードを持つ応答をキャッシュしないでください。

3. 応答タイプが FIN 終了の場合、または応答に Content-Length または Transfer-Encoding: Chunked のいずれかの属性がない場合は、応答をキャッシュしないでください。
4. キャッシュモジュールが Cache-Control ヘッダーを解析できない場合は、応答をキャッシュしないでください。

既定のコンテンツグループの初期設定

統合キャッシュを初めて有効にすると、Citrix ADC アプライアンスは、デフォルトのコンテンツグループという名前の定義済みコンテンツグループを提供します。詳細については、「既定のコンテンツグループ設定テーブル」を参照してください。

トラブルシューティング

October 7, 2021

統合キャッシュ機能を構成した後に期待どおりに機能しない場合は、一般的なツールを使用して Citrix ADC リソースにアクセスし、問題を診断できます。

トラブルシューティングのリソース

トラブルシューティングに使用できるリソースと設定例の詳細については、「PDF ファイルのトラブルシューティング用リソース」を参照してください。

フロントエンドの最適化

October 7, 2021

> 注：フロントエンドの最適化は、高度なライセンスまたはプレミアム Citrix ADC ライセンスがあり、Citrix ADC リリース 10.5 以降を実行している場合に利用できます。

Web アプリケーションの基礎となる HTTP プロトコルは、元々、単純な Web ページの送信とレンダリングをサポートするために開発されました。JavaScript やカスケードスタイルシート (CSS) などの新しいテクノロジや、Flash ビデオやグラフィックが豊富な画像などの新しいメディアタイプでは、フロントエンドのパフォーマンス、つまりブラウザーレベルでのパフォーマンスに多大な要求が課せられます。

Citrix ADC フロントエンド最適化（FEO）機能は、このような問題に対処し、次の方法で Web ページの読み込み時間とレンダリング時間を短縮します。

- 要求の数を減らす。
- 各ページのレンダリングに必要です。

- ページレスポンスのバイト数を減らす。

クライアントブラウザに提供されるコンテンツを簡素化し、最適化します。

FEO 構成をカスタマイズして、ユーザーに最適な結果を提供することができます。Citrix ADC は、デスクトップユーザーとモバイルユーザーの両方に対して、Web コンテンツの最適化を多数サポートしています。次の表では、FEO 機能によって提供されるフロントエンドの最適化と、さまざまな種類のファイルに対して実行される操作について説明します。

FEO 機能によって実行される最適化

ウェブ最適化	問題	Citrix ADC FEO 機能の機能	長所
インライン化	クライアントブラウザは、多くの場合、Web ページに関連付けられた外部 CSS、画像、および JavaScript をロードするためにサーバーに複数の要求を送信します。	CSS インライン、JavaScript インライン、CSS の組み合わせ	外部 CSS、画像、および JavaScript を HTML ファイルにインラインでロードすると、ページレンダリング時間が短縮されます。この最適化は、一度だけ表示されるコンテンツや、キャッシュサイズが制限されているモバイルデバイスに役立ちます。
ミニフィケーション	サーバから取得したデータには、空白、コメント、改行文字などの重要でない文字が含まれます。ブラウザがそのようなデータの処理に費やす時間は、ウェブサイトのレイテンシーを生成します。	CSS の縮小、JavaScript の縮小、HTML コメントの削除	縮小されたファイルは、帯域幅を消費し、特殊な処理によるレイテンシーを回避します。

ウェブ最適化	問題	Citrix ADC FEO 機能の機能	長所
画像最適化	モバイルブラウザでは、接続速度が遅く、キャッシュメモリが限られていることがよくあります。モバイルクライアントに画像をダウンロードすると、より多くの帯域幅、処理時間、およびキャッシュスペースが消費され、Web サイトの待ち時間が発生します。	JPEG 最適化, CSS 画像のインライン化, 画像の縮小属性, GIF から PNG への変換, HTML 画像のインライン化, WebP 画像変換, JPEG, GIF, PNG から JPEG-XR 画像変換	画像を Citrix ADC によって画像タグに示されているサイズに縮小し、クライアントブラウザが画像をより高速にロードできるようにします。
再配置	外部 CSS、画像、および JavaScript の非効率的な処理により、ページの読み込み時間が長くなります。	画像の遅延読み込み、CSS のヘッドへの移動、JavaScript の終了への移動	HTML 要素を再配置して、Web ページのレンダリング時間を短縮し、クライアントブラウザがオブジェクトをより高速にロードできるようにします。
接続管理	多くのブラウザでは、単一のドメインに対して確立できる同時接続の数に制限が設定されています。これにより、ブラウザーが Web ページのリソースを一度に 1 つずつダウンロードし、ブラウザーに時間がかかる可能性があります。	ドメインシャーディング	接続制限を克服します。これにより、クライアントブラウザがより多くのリソースを並行してダウンロードできるようになり、ページレンダリング時間が短縮されます。

さまざまなファイルタイプでの **Web** 最適化:

Citrix ADC は、CSS、画像、Javascript、および HTML 上で Web 最適化を実行できます。詳細については、Web 最適化 PDF を参照してください。

> 注:
> フロントエンド最適化機能は ASCII 文字のみをサポートします。Unicode 文字セットはサポートしていません。

フロントエンドの最適化の仕組み

Citrix ADC がサーバーから応答を受信した後、次の手順を実行します。

1. ページの内容を解析し、キャッシュにエントリを作成し (該当する場合)、FEO ポリシーを適用します。

   たとえば、Citrix ADC は次の最適化ルールを適用できます。

   - CSS または JavaScript 内の空白やコメントを削除します。
   - 1 つまたは複数の CSS ファイルを 1 つのファイルに結合します。
   - PNG 形式に GIF 画像形式を変換します.

2. 埋め込みオブジェクトを書き換え、最適化されたコンテンツをキャッシュに保存します。初期キャッシュエントリに使用されたものとは異なる署名を使用します。
3. 後続の要求では、サーバからではなく、キャッシュから最適化されたオブジェクトをフェッチし、応答をクライアントに転送します。

**

空白やコメントなどの無関係な情報を削除します。

サーバー上で新しいコンテンツが利用可能かどうかをチェックせずに、ブラウザがキャッシュされたリソースを使用できる期間。

フロントエンド最適化の構成

オプションで、フロントエンド最適化のグローバル設定の値を変更できます。それ以外の場合は、まず、埋め込みオブジェクトに適用される最適化ルールを指定するアクションを作成します。

アクションを設定したら、ポリシーを作成し、それぞれにレスポンスを最適化するリクエストのタイプを指定するルールを指定し、アクションをポリシーに関連付けます。

> 注: Citrix ADC は、リクエスト時にのみフロントエンド最適化ポリシーを評価し、応答時間ではなく評価します。

ポリシーを有効にするには、ポリシーをバインドポイントにバインドします。ポリシーをグローバルにバインドして、Citrix ADC を通過するすべてのトラフィックに適用することも、HTTP または SSL タイプの負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにポリシーをバインドすることもできます。ポリシーをバインドするときは、プライオリティを割り当てます。プライオリティ番号が小さいほど、値が大きいことを示します。Citrix ADC は、優先度の順にポリシーを適用します。

前提条件

フロントエンドの最適化では、Citrix ADC 統合キャッシュ機能を有効にする必要があります。また、次の統合キャッシュ構成を実行する必要があります。

- キャッシュ・メモリを割り当てます。

- デフォルトのキャッシュコンテンツグループの最大応答サイズとメモリ制限を設定します。

統合キャッシュの構成の詳細については、統合キャッシュを参照してください。

注：統合キャッシュという用語は、AppCache と同じ意味で使用できます。機能の観点から、両方の用語は同じ意味であることに注意してください。

Citrix ADC コマンドインターフェイスを使用してフロントエンドの最適化を構成する

コマンドプロンプトで、次の操作を行います。

1. フロントエンド最適化機能を有効にします。

```
enable ns feature FEO
```

1. 1 つ以上のフロントエンド最適化アクションを作成します。

```
add feo action <name> [-imgShrinkToAttrib] [-imgGifToPng] ...
```

例: GIF 形式の画像を PNG 形式に変換するためのフロントエンド最適化アクションを追加し、キャッシュの有効期限を延長するには:

```
add feo action allact -imgGifToPng -pageExtendCache
```

1. [オプション:] フロントエンド最適化のグローバル設定にデフォルト以外の値を指定します。

```
set feo parameter [-cacheMaxage <integer>] [-JpegQualityPercent <integer>]
[-cssInlineThresSize <integer>] [-inlineJsThresSize <integer> [-inlineImgThresSize
 <integer>]
```

例: キャッシュの最大有効期限を指定するには:

```
set feo parameter -cacheMaxage 10
```

1. 1 つ以上のフロントエンド最適化ポリシーを作成します。

```
add feo policy <name> <rule> <action>
```

例: フロントエンド最適化ポリシーを追加し、上記で指定した allact アクションに関連付けるには、次の手順を実行します。

```
1 >add feo policy pol1 TRUE all act
2 >add feo policy pol1 ”(HTTP.REQ.URL.CONTAINS(”testsite”))” allact1
3 <!--NeedCopy-->
```

1. ポリシーを負荷分散またはコンテンツスイッチング仮想サーバーにバインドするか、グローバルにバインドします。


```
bind lb vserver <name> -policyName <string> -priority <num>

bind cs vserver <name> -policyName <string> -priority <num>

bind feo global <policyName> <priority> -type <type> <gotoPriorityExpression
>
```

例:「abc」という名前の仮想サーバーにフロントエンド最適化ポリシーを適用するには、次のようにします。

```
> bind lb vserver abc -policyName pol1 -priority 1 -type NONE
```

例: ADC に到達するすべてのトラフィックに対してフロントエンドの最適化ポリシーを適用するには、次のようにします。

```
> bind feo global pol1 100 -type REQ_DEFAULT
```

1. 構成を保存します。構成を保存します。

GUI を使用したフロントエンドの最適化の構成

1. [最適化] > [フロントエンド最適化] > [アクション] に移動し、[追加] をクリックし、関連する詳細を指定してフロントエンド最適化アクションを作成します。
2. [オプション:] フロントエンド最適化のグローバル設定を指定します。
3. [最適化] > [フロントエンド最適化] に移動し、右側のペインの [設定] で [フロントエンド最適化設定の変更] をクリックし、フロントエンド最適化のグローバル設定を指定します。
4. フロントエンド最適化ポリシーを作成します。
5. [最適化] > [フロントエンド最適化] > [ポリシー] に移動し、[追加] をクリックし、関連する詳細を指定してフロントエンド最適化ポリシーを作成します。
6. 負荷分散またはコンテンツスイッチング仮想サーバーにポリシーをバインドします。
   a) [最適化] > [フロントエンド最適化] > [ポリシー] に移動します。
   b) フロントエンド最適化ポリシーを選択し、[**Policy Manager**] をクリックします。
   c) フロントエンド最適化ポリシーマネージャーで、フロントエンド最適化ポリシーを負荷分散またはコンテンツスイッチング仮想サーバーにバインドします。

フロントエンド最適化構成の確認

ダッシュボード・ユーティリティは、サマリー統計と詳細統計を表形式およびグラフィック形式で表示します。FEO 統計情報を表示して、FEO 構成を評価できます。

オプションで、ポリシーベースの FEO 中にポリシーカウンターが増分する選択の数など、FEO ポリシーの統計を表示することもできます。

> 注:
> 統計とグラフの詳細については、Citrix ADC アプライアンスのダッシュボードヘルプを参照してください。

CLI を使用して FEO 統計を表示する

コマンドプロンプトで次のコマンドを入力して、FEO 統計の概要、FEO ポリシーの選択と詳細、および詳細な FEO 統計をそれぞれ表示します。

- `stat feo` 注:stat feo policy コマンドは、高度な FEO ポリシーの統計情報のみを表示します。
- `show feo policy name`
- `stat feo -detail`

Citrix ADC ダッシュボードで FEO 統計を表示する

ダッシュボード GUI では、次の操作を実行できます。

- FEO 統計の要約を表示するには、[フロントエンドの最適化] を選択します。
- 「グラフィカルビュー」**(Graphical View**) タブをクリックして、FEO 機能によって処理されたリクエストのレートを表示します。

サンプルの最適化:

HTML コンテンツおよび HTML コンテンツ内の埋め込みオブジェクトに適用されるコンテンツ最適化アクションの例については、サンプル PDF を参照してください。

コンテンツアクセラレータ

October 7, 2021

> 重要:
>
> コンテンツアクセラレータ機能は、Citrix ADC アプライアンスではサポートされなくなりました。

コンテンツアクセラレータは、Citrix ByteMobile T1100 展開で使用できる Citrix ADC 機能で、Citrix ByteMobile T2100 アプライアンスにデータを格納します。

T2100 アプライアンスにデータを保存すると、帯域幅が節約され、応答時間が短縮されます。これは、同じデータを繰り返し要求するために Citrix ADC がサーバーに接続する必要がないためです。

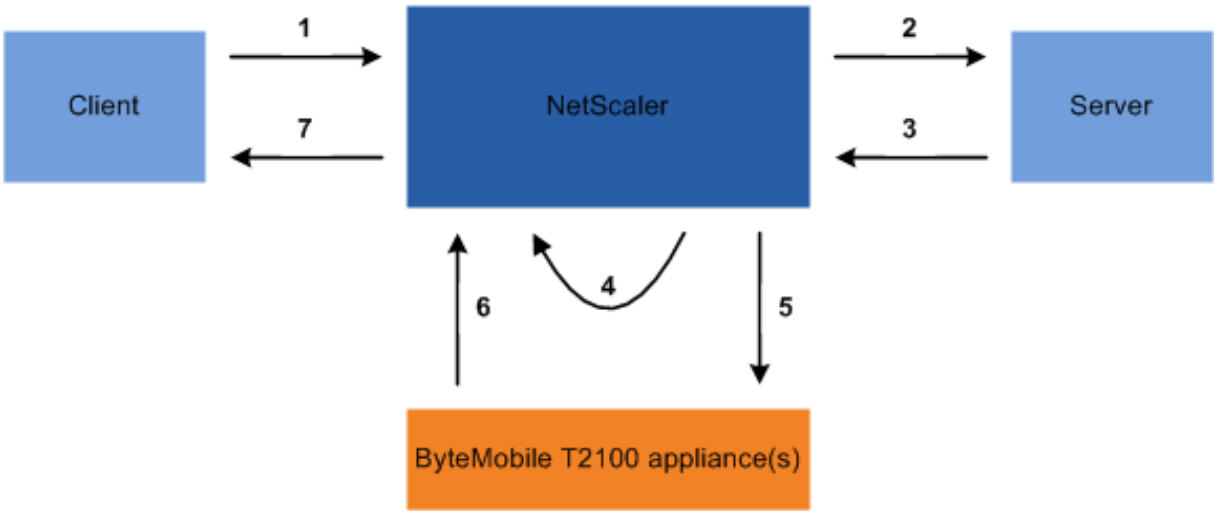
> 注:コンテンツアクセラレータは、Citrix ByteMobile Premium ライセンスで動作します。詳細およびライセンスの取得については、カスタマーサポートにお問い合わせください。

コンテンツアクセラレータのしくみ

負荷分散またはコンテンツスイッチング仮想サーバーがクライアント要求を受信すると、Citrix ADC アプライアンスは、仮想サーバーにバインドされたコンテンツアクセラレータポリシーを評価します。ポリシーは、コンテンツアクセラレータ機能を適用する要求を識別する要求をフィルタリングします。

> 注:
>
> HTTP リクエストの場合、コンテンツアクセラレータ機能は、単一バイト範囲のリクエストに応答して部分的なコンテンツを提供できます。

次の図は、コンテンツアクセラレータ機能を使用するように構成された仮想サーバにクライアント要求が到着したときにアプライアンスが実行する操作を示しています。



プロセス・フローは次のとおりです。

1. クライアントが要求を送信します。
2. Citrix ADC は要求をサーバーに転送します。
3. サーバは、事前定義された応答サイズ（add ca action コマンドの accumResSize パラメータによって指定）で応答します。
4. Citrix ADC は、サーバーから送信された応答のハッシュを計算します。
5. Citrix ADC は、T2100 アプライアンス上のハッシュを検索します。
6. 検索に成功すると、データが利用可能になり、T2100 アプライアンスが Citrix ADC にデータを送信します。

   > 注:
   > データベースルックアップが成功しない場合、アプライアンスは要求されたデータをサーバーからフェッチし、そのデータをクライアントに提供し、T2100 アプライアンスのデータを更新します。
   > T2100 アプライアンスは、データをキャッシュする要求の数を指定するように構成できます。

7. Citrix ADC は、クライアントに応答を送信します。

コンテンツアクセラレータを構成する

コンテンツアクセラレータ機能を設定する前に、Citrix ADC アプライアンスで有効にする必要があります。

1 つ以上の T2100 アプライアンスを使用するようにコンテンツアクセラレータ機能を構成する必要があります。各 T2100 アプライアンスをサービスとして追加し、これらのサービスを、構成された T2100 アプライアンス間で負荷を分散するための専用の負荷分散仮想サーバーにバインドする必要があります。

T2100 アプライアンスでデータを検索するには、コンテンツアクセラレータアクションを設定する必要があります。このアクションでは、T2100 負荷分散仮想サーバーと、ハッシュを計算するためにサーバーからフェッチするデータのサイズ（KB 単位）を指定する必要があります。

アクションは、コンテンツアクセラレータを実行するトラフィックを定義するコンテンツアクセラレータポリシーにバインドする必要があります。コンテンツアクセラレータポリシーは、クライアントトラフィックを受信するコンテンツスイッチまたは負荷分散仮想サーバーにバインドする必要があります。または、ポリシーを該当するすべての仮想サーバーにグローバルにバインドすることもできます。

コマンドラインインターフェイスを使用してコンテンツアクセラレータを構成するには

コマンドプロンプトで、次の操作を行います。

1. コンテンツアクセラレータ機能を有効にします。

   ```
   enable ns feature ca
   ```

2. T2100 アプライアンスを特定し、それぞれを Citrix ADC アプライアンスにサービスとして追加します。

   ```
   add service <name> <IPAddress> <serviceType> <port>
   ```

   例:

   ```
   > add service T2100-A 10.102.29.61 HTTP 30
   > add service T2100-B 10.102.29.62 HTTP 40
   > add service T2100-C 10.102.29.63 HTTP 50
   <!--NeedCopy-->
   ```

   > 注:
   > サービスは HTTP タイプのみである必要があります。

3. T2100 アプライアンス用の負荷分散仮想サーバーを作成します。トークン・ロード・バランシング方法と、次の構文に示すルールを指定します。

   ```
   add lb vserver <name> <serviceType> <IPAddress> <port> -lbMethod
       TOKEN -rule "http.req.url.after_str("/lookup/") alt http.req.
       url.path.SKIP(1).PREFIX(64)"
   <!--NeedCopy-->
   ```

   例:


```
1 add lb vserver T2100-lbvserver HTTP 10.102.29.64 99 -lbMethod
     TOKEN -rule "http.req.url.after_str("/lookup/") alt http.req.
     url.path.SKIP(1).PREFIX(64)"
2 <!--NeedCopy-->
```

4. T2100 サービスを、作成した負荷分散仮想サーバーにバインドします。

   ```
   bind lb vserver <name> <serviceName>
   ```

   例:

   ```
   1 > bind lb vserver T2100-lbvserver T2100-A
   2 > bind lb vserver T2100-lbvserver T2100-B
   3 > bind lb vserver T2100-lbvserver T2100-C
   4 <!--NeedCopy-->
   ```

5. コンテンツアクセラレータアクションを定義します。

   ```
   add ca action <name> accumResSize <KBytes> -lbvserver <string> -type
   lookup
   ```

   例:

   ```
   > add ca action ca_action1 -type lookup -lbvserver T2100-lbvserver -
   accumResSize 60
   ```

6. コンテンツアクセラレータポリシーを定義します。

   ```
   add ca policy <name> -rule <expression> -action <name>
   ```

   例:

   すべてのビデオ形式をキャッシュするコンテンツアクセラレータポリシーを作成します。

   ```
   > add ca policy ca_mp4_pol -rule ns_video -action ca_action1
   ```

   ns_video は組み込み式です。

7. コンテンツアクセラレータポリシーをトラフィックを受信する仮想サーバーまたは Citrix ADC システムにグローバルにバインドします。

   ```
   bind lb vserver <name> -policyName <string>
   ```

   ```
   bind cs vserver <name> -policyName <string>
   ```

   ```
   bind ca global -policyName <string> -priority <num> -type <type>
   ```

   例: コンテンツアクセラレータポリシーを「traf_rec」という名前の仮想サーバーに適用するには

   ```
   bind lb vserver traf_rec -policyName ca_mp4_pol
   ```


例： Citrix ADC に到達するすべてのトラフィックにコンテンツアクセラレータポリシーを適用します。

```
bind ca global -policyName ca_mp4_pol -priority 100 -type RES_DEFAULT
```

8. 構成を保存します。

```
save ns config
```

GUI を使用したコンテンツアクセラレータの構成

1. [システム] **>** [設定] **>** [高度な機能の設定] に移動し、[コンテンツアクセラレータ] を選択します。
2. 各 T2100 アプライアンスのサービスを作成します。
   a) [トラフィック管理] > [負荷分散] > [サービス] に移動します。
   b) [追加] をクリックし、関連する詳細を指定します。[**Server**] フィールドで、T2100 アプライアンスの IP アドレスを指定していることを確認します。[プロトコル] フィールドで [HTTP] を選択します。
3. 仮想サーバーを作成し、T2100 サービスをバインドします。
   a) [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
   b) [追加] をクリックし、関連する詳細を指定します。
   c)「メソッドと持続性」タブで、メソッドを トークンとして指定します。
   d)「ポリシー」タブで、ルールを http.req.url.after_str(“/lookup/”) alt http.req.url.path.SKIP(1).PREFIX(64) で指定します。
   e)［サービス］ タブで、仮想サーバにバインドする T2100 サービスを選択します。
4. コンテンツアクセラレータアクションを作成します。
   a) [最適化] > [コンテンツアクセラレータ] > [アクション] に移動します。
   b) 関連する詳細を指定します。
5. コンテンツアクセラレータポリシーを作成します。
   a) [最適化] > [コンテンツアクセラレータ] > [ポリシー] に移動します。
   b) [追加] をクリックし、ポリシールールを指定し、コンテンツアクセラレータアクションを関連付けます。
6. コンテンツアクセラレータポリシーをグローバルにバインドするか、仮想サーバーにバインドします。
   a) [最適化] > [コンテンツアクセラレータ] に移動します。
   b) [[コンテンツアクセラレータポリシーマネージャの要求]] セクションまたは [[コンテンツアクセラレータポリシーマネージャの応答]] セクションで、コンテンツアクセラレータポリシーをグローバルにバインドするか、仮想サーバにバインドします。

メディア分類

October 7, 2021

ネットワーク内のトラフィックのタイプを理解すると、ネットワーク管理者は帯域幅消費を管理して最適なネットワークパフォーマンスを得ることができます。メディア分類モードは、Citrix ADC アプライアンスを通過するメディアトラフィックの統計を監視し、表示します。

このモードを有効にすると、ネットワーク管理者は、アクセスされたデータの量、およびメディアファイルにアクセスされたデバイスのタイプを示す統計を収集できます。Citrix ADC アプライアンスは、このモードでのバイト範囲要求もサポートします。

現在、Citrix ADC アプライアンスは、次のメディアファイルタイプの統計情報を監視および表示できます。

メディア	ファイルタイプ
Microsoft Smooth Streaming	ビデオ
Apple Live Streaming	ビデオ
Audio Data Transport Stream (ADTS)	オーディオ
Advanced Audio Coding (AAC)	オーディオ
Flash Video (FLV)	オーディオとビデオ
3GP	オーディオとビデオ

アプライアンスは、次のデバイスの統計情報を表示できます。

デバイスプラットフォーム	デバイスの種類
iOS	アイポッド
Android	携帯電話とタブレット
ノートパソコンまたはデスクトップ	Windows ラップトップコンピュータおよびデスクトップコンピュータ
その他	その他のモバイル機器（モバイル・タブレット）

ネットワーク管理者は、以下の統計カウンタをチェックして、さまざまな種類のメディアトラフィックについてCitrix ADC アプライアンスを介してアクセスされるデータ量を知ることができます。

メディアファイル名	統計カウンタ
Microsoft Smooth Streaming	mcmssmthstrmvid—このカウンターは、Citrix ADC によって提供される MicrosoftSmoothStreaming ビデオの総数を記録します appliance; Mcmssmthstrvidpl—このカウンターは、Citrix ADC によって提供される MicrosoftSmoothStreaming ビデオプレイリストの総数を記録します appliance; Mcmssmthstrmvidbytes：このカウンターは、Citrix ADC で MicrosoftSmoothStreaming メディアトラフィックに提供されるデータバイトの総数を記録します appliance; Mcmssmthstrmplvidbytespl：このカウンターは、Citrix ADC アプライアンスによって提供される MicrosoftSmoothStreaming プレイリストバイトの総数を記録します。
Apple Live Streaming	mccaplelivestrmngvid-このカウンターは、Citrix ADC アプライアンスによって提供される Apple ライブストリーミングビデオの総数を記録します。Mccapplelivestrmngvidpl-このカウンタは、Citrix ADC アプライアンスによって提供される Apple ライブストリーミングビデオプレイリストの総数を記録します。 Mcapplelivestreamingvidbytes：このカウンターは、Citrix ADC アプライアンスで AppleLiveStreaming メディアトラフィックに提供されるデータバイトの総数を記録します。 Mcapplelivestreamingplaylistvidbytespl—このカウンターは、Citrix ADC アプライアンスによって提供される AppleLivePlaylist バイトの総数を記録します。
Audio Data Transport Stream (ADTS)	mcadtsaudio：このカウンターは、Citrix ADC アプライアンスによって提供される ADTS オーディオクリップの総数を記録します。Mcadtsaudiobytes：このカウンターは、Citrix ADC アプライアンスで ADTS メディアトラフィックに提供されるデータバイトの総数を記録します。

メディアファイル名	統計カウンタ
Advanced Audio Coding (AAC)	Mcaacaudio：このカウンターは、Citrix ADC アプライアンスによって提供される AAC オーディオクリップの総数を記録します。Mcaacaudiobytes：このカウンターは、Citrix ADC アプライアンスで AAC メディアトラフィックに提供されるデータバイトの総数を記録します。
Flash Video (FLV)	Mcflvvid—このカウンターは、Citrix ADC アプライアンスによって提供されるフラッシュビデオの総数を記録します。Mcflvvidbytes：このカウンターは、Citrix ADC アプライアンスでフラッシュビデオに提供されるデータバイトの総数を記録します。
3GP	mc3gpvidbytes-このカウンタは、Citrix ADC アプライアンスで 3GP メディアトラフィックに提供されたデータバイトの総数を記録します。

Citrix ADC アプライアンスは、レスポンスの 最初の本文バイトの署名によってメディアファイルの種類を検出します。たとえば、mp4 ファイルの最初の本文バイトの応答には次の署名があります。

```
**....ftypmp42** ....isommp42....moov...lmvhd.....c.\!.c.\!..
```

Citrix ADC アプライアンスは、クライアントデバイスが HTTP GET 要求に含める ユーザーエージェント文字列によってクライアントデバイスの種類を検出します。たとえば、UC ブラウザを使用する Windows Phone では、HTTP GET リクエストに次のユーザーエージェント文字列が含まれています。

```
User-Agent: **UCWEB**/2.0 (**Windows**; U; wds 8.10; en-US; HTC; 8X by HTC)
U2/1.0.0
```

メディアの分類を有効にする

デフォルトでは、Citrix ADC アプライアンスではメディア分類が無効になっています。使用する前に、モードを有効にする必要があります。

コマンドラインインターフェイスを使用してメディア分類を有効にするには

コマンドプロンプトで入力します。

```
enable ns mode Mediaclassification
```

GUI を使用してメディア分類を有効にするには

Citrix ADC アプライアンスでメディア分類を有効にする

[システム] > [設定] > [モードの構成] に移動し、[メディアの分類] を選択します。

Citrix ADC アプライアンスでメディアトラフィックの統計情報を表示するには

[最適化] に移動し、[メディアの分類] をクリックして、メディアトラフィックの統計情報を表示します。

メディア分類統計の確認

ダッシュボードユーティリティまたはコマンドラインインターフェイスを使用して、メディアトラフィックの統計を表示できます。ダッシュボード・ユーティリティは、サマリー統計と詳細な統計を表形式およびグラフィック形式で表示します。

> 注:
> 統計情報とグラフの詳細については、Citrix ADC アプライアンスのダッシュボードのヘルプを参照してください。

コマンドラインインターフェイスを使用してメディア分類の統計情報を表示するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、メディア分類の統計情報の概要を表示したり、詳細な統計情報を表示したり、表示をクリアします。

```
stat Mediaclassification

stat Mediaclassification -detail

stat Mediaclassification -clearstats
```

ダッシュボードでメディア分類の統計を表示するには

ダッシュボード・ユーティリティでは、次のタイプのメディア分類統計を表示できます。

1. メディアトラフィック統計情報のサマリーを表示するには、[メディア分類] を選択します。
2. 詳細なメディアトラフィック統計情報を表示するには、[**Details**] をクリックします。
3. メディアトラフィック統計情報をクリアするには、[**Clear**] をクリックします。

評価

October 7, 2021

Citrix は、レピュテーションベースのセキュリティを提供します。レピュテーション評価を使用してリクエストを処理するリスクを判断し、特定のリクエストをブロックまたはドロップするなどのアクションを実行して、アプリケーションのパフォーマンスを向上させることができます。

Citrix ADC IP レピュテーション機能は、IP レピュテーションチェックを使用して、ゼロデイ攻撃を防ぎ、Web 攻撃、フィッシング活動、または Web スキャンに関連する悪意のあるソースからの保護を提供します。

詳細については、IP レピュテーションを参照してください。

IP レピュテーション

December 7, 2021

IP レピュテーションは、不要な要求を送信する IP アドレスを識別するツールです。IP レピュテーションリストを使用すると、レピュテーションの悪い IP アドレスからのリクエストを拒否できます。処理しない要求をフィルタリングして、Web アプリケーションファイアウォールのパフォーマンスを最適化します。リセット、要求のドロップ、またはレスポンダーポリシーを構成して、特定のレスポンダーアクションを実行します。

次に、IP レピュテーションを使用して防止できる攻撃をいくつか挙げます。

- ウイルスに感染したパーソナルコンピュータ。（自宅の PC）は、インターネット上のスパムの唯一の最大のソースです。IP レピュテーションは、不要な要求を送信している IP アドレスを識別できます。IP レピュテーションは、既知の感染元からの大規模な DDoS 攻撃、DoS 攻撃、または異常な SYN フラッド攻撃をブロックする場合に特に役立ちます。
- 一元管理および自動化されたボットネット。攻撃者はパスワードを盗むことで人気を博しています。何百ものコンピュータが連携してパスワードを解読するのに時間がかからないからです。ボットネット攻撃を開始して、よく使われる辞書の単語を使用するパスワードを見つけ出すのは簡単です。
- 侵害された **Web** サーバー。攻撃はそれほど一般的ではありません。なぜなら、意識とサーバーのセキュリティが高まっているため、ハッカーやスパマーはより簡単な標的を探します。ハッカーが妥協してスパム（ウイルスやポルノなど）を送信するために使用できる Web サーバーやオンラインフォームはまだあります。このようなアクティビティは、SpamRats などのレピュテーションリストを使用して検出してすばやくシャットダウンしたり、ブロックしたりするのが簡単です。
- **Windows** エクスプロイト。（マルウェア、シェルコード、ルートキット、ワーム、またはウイルスを提供または配布するアクティブ IP など）。
- 既知のスパマーとハッカー。
- マスメールマーケティングキャンペーン。
- フィッシングプロキシ（フィッシングサイトをホストしている IP アドレス、および広告クリック詐欺やゲーム詐欺などのその他の詐欺）。
- 匿名プロキシ **(**プロキシおよび匿名化サービスを提供する **IP** (オニオンルーター別名 TOR を含む)。

Citrix ADC アプライアンスは、動的に生成された悪意のある **IP** データベースとそれらの **IP** アドレスのメタデータのサービスプロバイダーとして **Webroot** を使用します。メタデータには、位置情報の詳細、脅威カテゴリ、脅威数などが含まれます。Webroot 脅威インテリジェンスエンジンは、数百万のセンサーからリアルタイムデータを受信します。高度な機械学習と行動分析を使用して、データを自動的かつ継続的にキャプチャ、スキャン、分析、スコアリングします。脅威に関するインテリジェンスは継続的に更新されます。

Citrix ADC アプライアンスは、Webroot の使用 IP レピュテーションデータベースを使用して、受信リクエストの評判が悪いかどうかを検証します。データベースには、IP アドレス分類に基づく IP 脅威カテゴリの膨大なコレクションがあります。次に、IP 脅威のカテゴリとその説明を示します。

- スパムソース。スパム送信元には、プロキシを介したスパムメッセージのトンネリング、異常な SMTP アクティビティ、フォーラムスパムアクティビティが含まれます。

- Windows エクスプロイト。Windows エクスプロイトのカテゴリには、マルウェア、シェルコード、ルートキット、ワーム、ウイルスを提供または配布するアクティブ IP アドレスが含まれます。
- ウェブ攻撃。Web 攻撃カテゴリには、クロスサイトスクリプティング、iFrame インジェクション、SQL インジェクション、クロスドメインインジェクション、ドメインパスワードブルートフォース攻撃が含まれます
- ボットネット。ボットネットカテゴリには、ボットネットの C&C チャネル、およびボットマスターが制御する感染したゾンビマシンが含まれます
- スキャナー。スキャナのカテゴリには、プローブ、ホストスキャン、ドメインスキャン、パスワードブルートフォース攻撃など、すべての偵察が含まれます。
- サービス拒否。サービス拒否カテゴリには、DOS、DDOS、異常同期フラッド、異常トラフィックの検出が含まれます
- 評判。マルウェアに感染していることが現在わかっている IP アドレスからのアクセスを拒否します。このカテゴリには、Webroot レピュテーションインデックススコアが平均的に低い IP も含まれます。このカテゴリを有効にすると、マルウェアの配布ポイントに連絡するために特定されたソースからのアクセスが防止されます
- フィッシング。フィッシングカテゴリには、フィッシングサイトをホストする IP アドレス、広告クリック詐欺、ゲーム詐欺などのその他の種類の詐欺行為が含まれます。
- プロキシ。プロキシカテゴリには、プロキシおよびデフサービスを提供する IP アドレスが含まれます。
- モバイルの脅威。モバイル脅威カテゴリには、悪意のあるモバイルアプリケーションや望ましくないモバイルアプリケーションの IP アドレスが含まれます。このカテゴリは、Webroot モバイル脅威調査チームのデータを活用します。
- Tor プロキシ Tor プロキシカテゴリには、Tor ネットワークの出口ノードとして動作する IP アドレスが含まれます。終了ノードはプロキシチェーンに沿った最後のポイントで、オリジネータの意図したデスティネーションに直接コネクトします。

ネットワーク内の任意の場所で脅威が検出されると、IP アドレスに悪意のあるフラグが付けられ、ネットワークに接続されているすべてのアプライアンスが即座に保護されます。IP アドレスの動的な変更は、高度な機械学習を使用して高速かつ正確に処理されます。

Webroot のデータシートに記載されているように、Webroot のセンサーネットワークは、スパムソース、Windows エクスプロイト、ボットネット、スキャナーなど、多くの主要な IP 脅威タイプを識別します。(データシートのフロー図を参照してください。)

Citrix ADC アプライアンスは、`iprep`クライアントプロセスを使用して Webroot からデータベースを取得します。`iprep`クライアントは HTTP GET メソッドを使用して、Webroot から絶対 IP リストを初めて取得します。その後、デルタの変更を 5 分ごとに 1 回チェックします。

> 重要:
>
> - IP レピュテーション機能を使用する前に、Citrix ADC アプライアンスにインターネットアクセスがあり、DNS が構成されていることを確認してください。
> - ウェブルートデータベースにアクセスするには、Citrix ****ADC** アプライアンスがポート **443** で **api.bcti.brightcloud.com** に接続できる必要があります **。HA またはクラスタ展開の各ノードは、Webroot からデータベースを取得し、この完全修飾ドメイン名（FQDN）にアクセスできる必要があり

ます。

- Webroot は現在 AWS でレピュテーションデータベースをホストしています。したがって、Citrix ADC は、レピュテーションデータベースをダウンロードするために AWS ドメインを解決できる必要があります。また、ファイアウォールは AWS ドメインに対して開いている必要があります。

注:

IP レピュテーション機能が有効の場合、各パケットエンジンが正しく機能するには、少なくとも 4 GB が必要です。

高度なポリシー式。Web Application Firewall やレスポンダーなど、サポートされているモジュールにバインドされたポリシーで高度なポリシー式（デフォルトの構文式）を使用して、IP レピュテーション機能を設定します。次に、クライアント IP アドレスが悪意があるかどうかを検出するために使用できる式を示す 2 つの例を示します。

1. **CLIENT.IP.SRC.IPREP_IS_MALICIALICE**: この式は、クライアントが悪意のある IP リストに含まれている場合に TRUE と評価されます。
2. **CLIENT.IP.SRC.IPREP_THREAT_CATEGORY (**カテゴリ**)**: この式は、クライアント IP が悪意のある IP であり、指定された脅威カテゴリにある場合に TRUE と評価されます。

脅威カテゴリに指定できる値は次のとおりです。

SPAM_SOURCES, WINDOWS_EXPLOITS, WEB_ATTACKS, ボットネット, スキャナ, DOS, レピュテーション, フィッシング, プロキシ, ネットワーク, CLOUD_PROVIDERS, MOBILE_ 脅威, TOR_PROXY。

注:

IP レピュテーション機能は、送信元と宛先 IP アドレスの両方をチェックします。ヘッダー内の悪意のある IP を検出します。ポリシー内の PI 式が IP アドレスを識別できる場合、IP レピュテーションチェックはそれが悪意があるかどうかを判断します。

iPrep ログメッセージ。/var/log/iprep.logファイルには、Webroot データベースとの通信に関する情報をキャプチャする便利なメッセージが含まれています。この情報には、Webroot 通信中に使用される資格情報、Webroot との接続の失敗、更新に含まれる情報（データベース内の IP アドレス数など）が含まれます。

ポリシーデータセットを使用して **IP** のブロックリストまたは許可リストを作成する。許可リストを維持して、Webroot データベースでブロックリストに登録されている特定の IP アドレスへのアクセスを許可できます。また、Webroot レピュテーションチェックを補完するために、IP アドレスのカスタマイズされたブロックリストを作成することもできます。これらのリストは、ポリシーデータセットを使用して作成できます。データセットは、IPv4 アドレスのマッチングに最適な特殊なパターンセットです。データセットを使用するには、まずデータセットを作成し、そのデータセットに IPv4 アドレスをバインドします。パケット内の文字列を比較するポリシーを設定する場合は、適切な演算子を使用し、パターンセットまたはデータセットの名前を引数として渡します。

IP レピュテーション評価中に例外として処理するアドレスの許可リストを作成するには、次の手順を実行します。

- 許可リストのアドレスが Webroot (または任意のサービスプロバイダー) によって悪意のあるアドレスとしてリストされている場合でも、PI 式が False と評価されるようにポリシーを構成します。

IP レピュテーションの有効化または無効化。IP レピュテーションは、ライセンスベースの一般レピュテーション機能の一部です。レピュテーション機能を有効または無効にすると、IP レピュテーションが有効または無効になります。

一般的な手順。IP レピュテーションの展開には、次のタスクが含まれます。

- Citrix ADC アプライアンスにインストールされているライセンスに IP レピュテーションがサポートされていることを確認します。プレミアムおよびスタンドアロンのアプリケーションファイアウォールライセンスは、IP レピュテーション機能をサポートします。
- IP レピュテーションおよびアプリケーションファイアウォール機能を有効にします。
- アプリケーションファイアウォールプロファイルを追加します。
- PI 式を使用して、IP レピュテーションデータベース内の悪意のある IP アドレスを識別するアプリケーションファイアウォールポリシーを追加します。
- アプリケーションファイアウォールポリシーを適切なバインドポイントにバインドします。
- 悪意のあるアドレスから受信した要求が`ns.log`ファイルに記録され、要求がプロファイルの指定どおりに処理されたことを示します。

CLI を使用して **IP** レピュテーション機能を設定する

コマンドプロンプトで入力します。

- `enable feature reputation`
- `disable feature reputation`

次の例は、PI 式を使用して悪意のあるアドレスを識別するアプリケーションファイアウォールポリシーを追加する方法を示しています。組み込みプロファイルを使用するか、プロファイルを追加するか、既存のプロファイルを設定して、リクエストがポリシー一致と一致したときに目的のアクションを呼び出すことができます。

例 3 と 4 は、IP アドレスのブロックリストまたは許可リストを生成するポリシーデータセットを作成する方法を示しています。

例 **1**:

次のコマンドは、悪意のある IP アドレスを識別し、一致がトリガーされた場合に要求をブロックするポリシーを作成します。

```
add appfw policy pol1 CLIENT.IP.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK
```

例 **2**:

次のコマンドは、レピュテーションサービスを使用して`X-Forwarded-For`ヘッダー内のクライアント IP アドレスをチェックし、一致がトリガーされた場合は接続をリセットするポリシーを作成します。

```
> add appfw policy pol1 ”HTTP.REQ.HEADER(\”X-Forwarded-For\”).TYPECAST_IP_ADDRESS_AT
.IPREP_IS_MALICIOUS”APPFW_RESET**
```

例 **3:**

次に、リストを追加して、指定した IP アドレスを許可する例外を追加する例を示します。


```
> add policy dataset Allow_list1 ipv4
> bind policy dataset Allow_list1 10.217.25.17 -index 1
> bind policy dataset Allow_list1 10.217.25.18 -index 2
```

例 **4:**

次に、カスタマイズリストを追加して、指定した IP アドレスに悪意のあるフラグを付ける例を示します。

```
> add policy dataset Block_list1 ipv4
> bind policy dataset Block_list1 10.217.31.48 -index 1
> bind policy dataset Block_list1 10.217.25.19 -index 2
```

例 **5:**

次の例は、次の条件でクライアント IP をブロックするポリシー式を示しています。

- カスタマイズされた block_list1 で設定された IP アドレスと一致します（例 4）
- allow_list1 に含めることによって緩和されない限り、Webroot データベースにリストされている IP アドレスと一致します（例 3）。

```
1 > add appfw policy ”Ip_Rep_Policy” ”((CLIENT.IP.SRC.IPREP_IS_MALICIOUS
     || CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(”Block_list1”)) && ! (
     CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(”Allow_list1”)))”
     APPFW_BLOCK
2 <!--NeedCopy-->
```

プロキシサーバーの使用**:**

Citrix ADC アプライアンスがインターネットに直接アクセスせず、プロキシに接続されている場合は、プロキシに要求を送信するように IP レピュテーションクライアントを構成します。

コマンドプロンプトで入力します。

```
set reputation settings -proxyServer <proxy server ip> -proxyPort <proxy
server port>
```

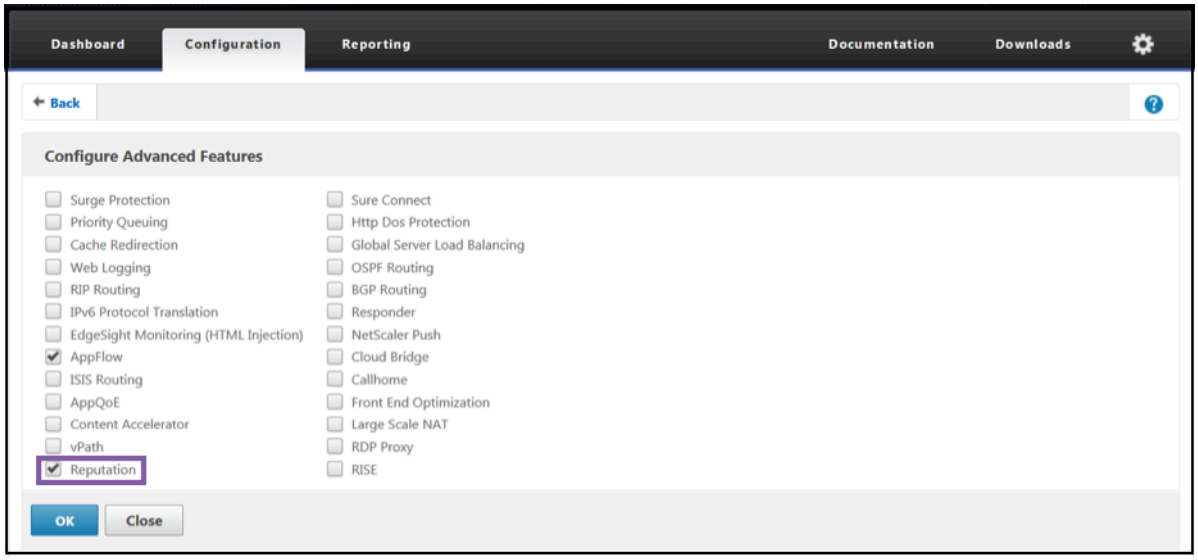
例**:**

```
> set reputation settings proxyServer 10.102.30.112 proxyPort 3128
> set reputation settings -proxyServer testproxy.citrite.net -proxyPort 3128
> unset reputation settings -proxyserver -proxyport
> sh reputation settings
```


> 注:
>
> プロキシサーバー IP には、IP アドレスまたは完全修飾ドメイン名 (FQDN) を指定できます。

Citrix ADC GUI を使用して IP レピュテーションを構成する

1. [システム] > [設定] に移動します。[モードと機能] セクションで、リンクをクリックして [高度な機能の構成] ウィンドウにアクセスし、[レピュテーション] チェックボックスをオンにします。
2. [**OK**] をクリックします。

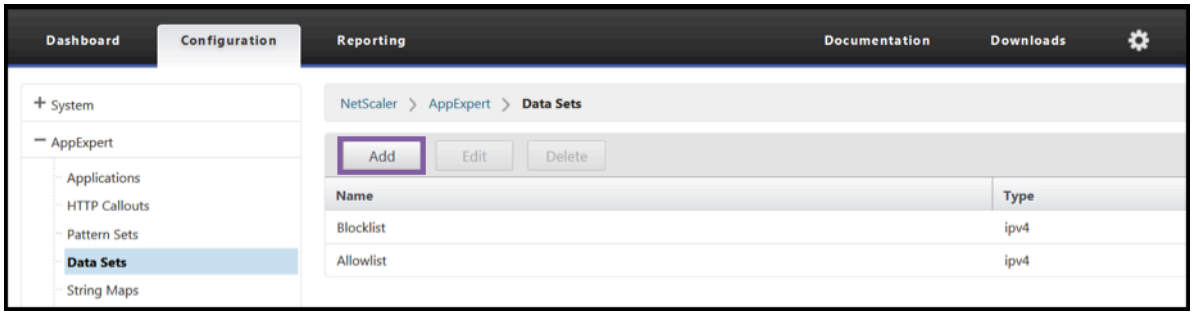


Citrix ADC GUI を使用してプロキシサーバーを構成するには

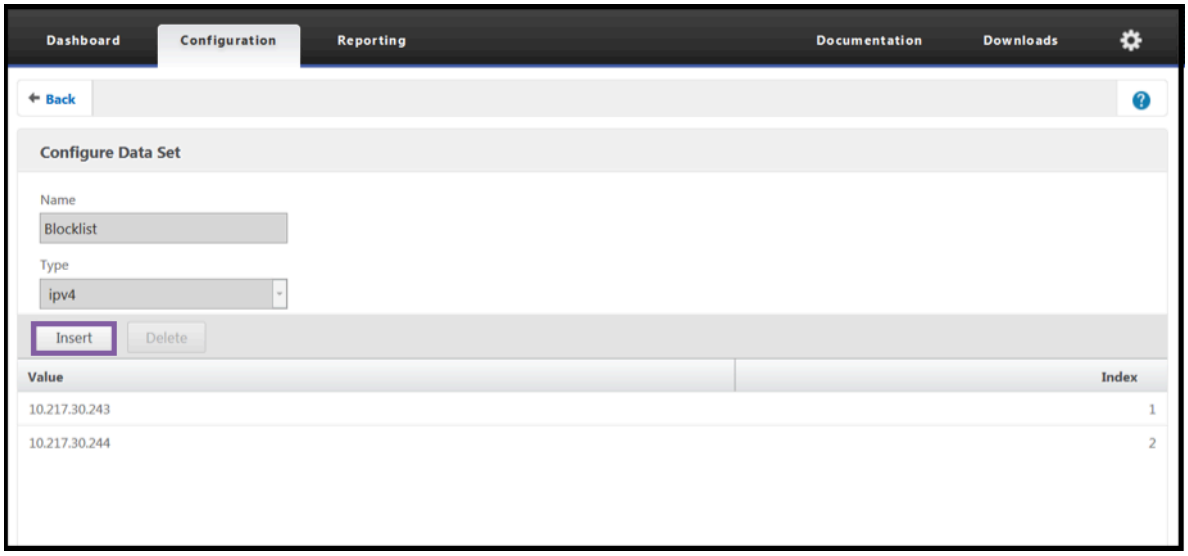
1. [設定] タブで、[セキュリティ] > [レピュテーション] に移動します。[設定] で、[レピュテーション設定の変更] をクリックしてプロキシサーバーを構成します。レピュテーション機能を有効または無効にすることもできます。プロキシサーバーには、IP アドレスまたは完全修飾ドメイン名 (FQDN) を指定できます。プロキシポートは [1 ～65535] の値を受け入れます。

GUI を使用してクライアント **IP** アドレスの許可リストとブロックリストを作成する

1. [構成] タブで、[**AppExpert] > [**データセット] に移動します。
2. [追加] をクリックします。



- [データセットの作成] (または [データセットの構成]) ペインで、IP アドレスのリストに意味のある名前を入力します。名前は、リストの目的を反映している必要があります。
- [タイプ] を **IPv4** として選択します。
- [**Insert**] をクリックしてエントリを追加します。



- [ポリシーデータセットのバインドの設定] ウィンドウで、[値] 入力ボックスに IPv4 形式の IP アドレスを追加します。
- インデックスを指定します。
- リストの目的を説明するコメントを追加します。この手順はオプションですが、説明的なコメントがリストの管理に役立つため、推奨されています。

同様に、ブロックリストを作成し、悪意のあると見なされる IP アドレスを追加できます。

データセットの使用およびデフォルトの構文ポリシー式の構成の詳細については、「パターンセットとデータセット」も参照してください。

Citrix ADC GUI を使用してアプリケーションファイアウォールポリシーを構成する

1. [構成] タブで、[セキュリティ] **>** [アプリケーションファイアウォール] **>** [ポリシー] **>** [ファイアウォール] に移動します。[追加（Add）] をクリックして、IP レピュテーションを使用する PI 式を使用するポリシーを追加します。

式エディタを使用して、独自のポリシー式を作成することもできます。このリストには、脅威カテゴリを使用して式を設定するのに役立つ、構成済みのオプションが表示されます。

ハイライト

- さまざまな種類の脅威をもたらす既知の悪意のある IP アドレスから、ネットワークのエッジで悪質なトラフィックを迅速かつ正確に阻止します。本文を解析せずにリクエストをブロックできます。
- 複数のアプリケーションの IP レピュテーション機能を動的に設定します。
- パフォーマンス・ペナルティなしでデータ侵害からネットワークを保護し、迅速かつ簡単な導入を使用して保護を単一のサービスファブリックに統合します。
- 送信元と宛先 IP で IP レピュテーションチェックを実行できます。
- ヘッダーを検査して、悪意のある IP を検出することもできます。
- IP レピュテーションチェックは、フォワードプロキシとリバースプロキシの両方の展開でサポートされています。
- IP レピュテーションプロセスは Webroot に接続し、5 分ごとにデータベースを更新します。
- 高可用性 (HA) またはクラスタ展開の各ノードは、Webroot からデータベースを取得します。
- IP レピュテーションデータは、管理パーティション展開のすべてのパーティションで共有されます。
- AppExpert データセットを使用して IP アドレスのリストを作成し、Webroot データベースでブロックリストに登録されている IP の例外を追加できます。独自にカスタマイズしたブロックリストを作成して、特定の IP を悪意のあるものとして指定することもできます。
- iprep.db ファイルが/var/nslog/iprepフォルダ内に作成されます。一度作成すると、フィーチャが無効になっていても削除されません。
- レピュテーション機能が有効になると、Citrix ADC Webroot データベースがダウンロードされます。その後、5 分ごとに更新されます。
- Webroot データベースのバージョンは 1 です。
- マイナーバージョンは毎日更新されます。更新バージョンは 5 分ごとにインクリメントされ、マイナーバージョンがインクリメントされると 1 にリセットされます。
- PI 式を使用すると、応答側や書き換えなどの他の機能で IP レピュテーションを使用できます。
- データベース内の IP アドレスは 10 進表記です。

デバッグに関するヒント

- GUI にレピュテーション機能が表示されない場合は、適切なライセンスがあることを確認します。
- デバッグのために、var/log/iprep.logのメッセージをモニタします。
- **Webroot** 接続: ns iprep: Not able to connect/resolve WebRootメッセージが表示された場合は、アプライアンスにインターネットアクセスがあり、DNS が設定されていることを確認します。

- プロキシサーバー: `ns iprep: iprep_curl_download: 88 curl_easy_perform failed . Error code: 5 Err msg:couldnt resolve proxy name`メッセージが表示された場合は、プロキシサーバーの設定が正確であることを確認してください。
- **IP** レピュテーション機能が動作しない: レピュテーション機能を有効にすると、IP レピュテーションプロセスが開始するまでに約 5 分かかります。IP レピュテーション機能はその期間動作しない場合があります。
- データベースのダウンロード: IP レピュテーション機能を有効にした後に IP DB データのダウンロードが失敗した場合、ログに次のエラーが表示されます。

```
iprep: iprep_curl_download:86 curl_easy_perform failed. Error code:7 Err
msg:Couldn't connect to server
```

解決方法: 次の URL へのアウトバウンドトラフィックを許可するか、プロキシを設定して問題を解決します。

```
1 localdb-ip-daily.brightcloud.com:443
2 localdb-ip-rtu.brightcloud.com:443
3 api.bcti.brightcloud.com:443
4 <!--NeedCopy-->
```

SSL オフロードおよび SSL アクセラレーション

October 7, 2021

SSL アクセラレーション用に構成された Citrix ADC アプライアンスは、SSL 処理をサーバーからオフロードすることで、SSL トランザクションを透過的に高速化します。SSL オフロードを設定するには、SSL トランザクションを代行受信して処理し、復号化されたトラフィックをサーバに送信するように仮想サーバを設定します（エンドツーエンドの暗号化を設定しない限り、トラフィックは再暗号化されます）。サーバーから応答を受信すると、アプライアンスはクライアントとの安全なトランザクションを完了します。クライアントの観点からは、トランザクションはサーバーと直接関係しているようです。SSL アクセラレーション用に構成された Citrix ADC は、負荷分散などの他の構成された機能も実行します。

SSL オフロードを設定するには、SSL 証明書とキーペアが必要です。SSL 証明書がない場合は、この証明書を取得する必要があります。その他の SSL 関連のタスクには、証明書の管理、証明書失効リストの管理、クライアント認証の構成、SSL アクションとポリシーの管理などがあります。

FIPS 以外の Citrix ADC アプライアンスは、サーバーの秘密鍵をハードディスクに保存します。FIPS アプライアンスでは、キーはハードウェアセキュリティモジュール (HSM) と呼ばれる暗号化モジュールに格納されます。

FIPS カードをサポートしないすべての Citrix ADC アプライアンス（仮想アプライアンスを含む）は、Thales nShield® Connect および SafeNet 外部 HSM をサポートしています。（MPX アプライアンスは、外部 HSM をサポートしていません）。

注：このドキュメントで説明されている一部の SSL 構成手順の FIPS 関連オプションは、FIPS 対応の Citrix ADC アプライアンスに固有のものです。

SSL オフロード設定

March 8, 2022

SSL オフロードを構成するには、Citrix ADC アプライアンスで SSL 処理を有効にし、SSL ベースの仮想サーバーを構成する必要があります。仮想サーバは SSL トラフィックをインターセプトし、トラフィックを復号化し、仮想サーバにバインドされたサービスに転送します。メディアストリーミングなどの時間的制約のあるトラフィックを保護するために、DTLS 仮想サーバーを構成できます。SSL オフロードを有効にするには、有効な証明書とキーをインポートし、そのペアを仮想サーバーにバインドする必要があります。

SSL を有効にする

SSL トラフィックを処理するには、SSL 処理を有効にする必要があります。SSL 処理を有効にしなくても、仮想サーバやサービスなどの SSL ベースのエンティティを設定できます。ただし、SSL 処理が有効になるまでは機能しません。

CLI を使用して SSL 処理を有効にする

コマンドプロンプトで入力します。

```
enable ns feature ssl

show ns feature
<!--NeedCopy-->
```

例:

```
enable ns feature SSL
 Done
show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
```



```
3)      Load Balancing                  LB                      ON
.
.
.
9)      SSL Offloading                  SSL                     ON
.
.
.
24)     NetScaler Push                  push                    OFF
Done
<!--NeedCopy-->
```

GUI を使用して **SSL** 処理を有効にする

[システム] **>** [設定] に移動し、[モードと機能] グループで [基本機能の構成] をクリックし、[**SSL** オフロード] をクリックします。

サービスの構成

Citrix ADC アプライアンスでは、サービスは物理サーバーまたは物理サーバー上のアプリケーションを表します。設定が完了すると、アプライアンスがネットワーク上の物理サーバに到達してそのステータスを監視できるようになるまで、サービスは無効状態になります。

CLI を使用してサービスを追加する

コマンドプロンプトで次のコマンドを入力してサービスを追加し、構成を確認します。

```
add service <name> (<IP> | <serverName>) <serviceType> <port>
show service <serviceName>
<!--NeedCopy-->
```

例**:**

```
add service sslsvc 198.51.100.225 SSL 443

Done

sh ssl service sslsvc

            Advanced SSL configuration for Back-end SSL Service sslsvc:
```



```
8               DH: DISABLED
9               DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                    RSA: DISABLED
10              Session Reuse: ENABLED          Timeout: 300 seconds
11              Cipher Redirect: DISABLED
12              SSLv2 Redirect: DISABLED
13              ClearText Port: 0
14              Server Auth: DISABLED
15              SSL Redirect: DISABLED
16              Non FIPS Ciphers: DISABLED
17              SNI: DISABLED
18              OCSP Stapling: DISABLED
19              SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                    ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
20              Send Close-Notify: YES
21              Strict Sig-Digest Check: DISABLED
22              Zero RTT Early Data: ???
23              DHE Key Exchange With PSK: ???
24              Tickets Per Authentication Context: ???
25
26              ECC Curve: P_256, P_384, P_224, P_521
27
28      1)      Cipher Name: DEFAULT_BACKEND
29              Description: Default cipher list for Backend SSL session
30       Done
31  <!--NeedCopy-->
```

CLI を使用してサービスを変更または削除する

サービスを変更するには、set service コマンドを使用します。これは add service コマンドと同様ですが、既存のサービスの名前を入力する点が異なります。

サービスを削除するには、rm service コマンドを使用します。このコマンドは\ <name\ > 引数のみを受け付けます。

```
1  rm service <servicename>
2  <!--NeedCopy-->
```

例**:**

```
1  rm service sslsvc
2  <!--NeedCopy-->
```


サービスを変更するには、set service コマンドを使用し、任意のパラメータを選択してその設定を変更します。

```
set service <name> (<IP> | <serverName>) <serviceType> <port>
<!--NeedCopy-->
```

例:

```
set service sslsvc 198.51.100.225 SSL 443
<!--NeedCopy-->
```

GUI を使用してサービスを構成する

[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを作成して、プロトコルを SSL として指定します。

SSL 仮想サーバ構成

セキュアセッションでは、Citrix ADC アプライアンス上のクライアントと SSL ベースの仮想サーバー間の接続を確立する必要があります。SSL 仮想サーバは、SSL トラフィックをインターセプトし、復号化して処理してから、仮想サーバにバインドされたサービスに送信します。

注：有効な証明書/キーのペアと少なくとも 1 つのサービスがバインドされるまで、Citrix ADC アプライアンスでは SSL 仮想サーバーがダウンしているとマークされます。SSL ベースの仮想サーバーは、プロトコルタイプが SSL または SSL_TCP の負荷分散仮想サーバーです。Citrix ADC アプライアンスで負荷分散機能を有効にする必要があります。

CLI を使用して **SSL** ベースの仮想サーバーを追加します

コマンドプロンプトで次のコマンドを入力し、SSL ベースの仮想サーバーを追加して構成を確認します。

```
add lb vserver <name> (serviceType) <IPAddress> <port>
show ssl vserver <name>
<!--NeedCopy-->
```

例:


```
add lb vserver sslvs SSL 192.0.2.240 443
Done

sh ssl vserver sslvs

            Advanced SSL configuration for VServer sslvs:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: ENABLED          Refresh Count: 0
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
            HSTS Max-Age: 0
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Push Encryption Trigger: Always
            Send Close-Notify: YES
            Strict Sig-Digest Check: DISABLED
            Zero RTT Early Data: DISABLED
            DHE Key Exchange With PSK: NO
            Tickets Per Authentication Context: 1
            ECC Curve: P_256, P_384, P_224, P_521

    1)      Cipher Name: DEFAULT
            Description: Default cipher list with encryption strength
                >= 128bit
     Done
<!--NeedCopy-->
```

CLI を使用して **SSL** ベースの仮想サーバを変更または削除する

SSL 仮想サーバの負荷分散プロパティを変更するには、`set lb vserver` コマンドを使用します。set コマンドは`add lb vserver`コマンドと似ていますが、既存の仮想サーバの名前を入力する点が異なります。**SSL** ベースの仮想サーバの **SSL** プロパティを変更するには、`set ssl vserver` コマンドを使用します。詳細については、このページの「SSL 仮想サーバーパラメーター」を参照してください。

SSL 仮想サーバを削除するには、<name>引数のみを受け入れるrm lb vserverコマンドを使用します。

GUI を使用して **SSL** ベースの仮想サーバを構成する

[トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを作成し、プロトコルを SSL として指定します。

サービスの **SSL** 仮想サーバーへのバインド

ADC アプライアンスは、復号化された SSL データをネットワーク内のサーバに転送します。データを転送するには、これらの物理サーバを表すサービスを、SSL データを受信する仮想サーバにバインドする必要があります。

通常、ADC アプライアンスと物理サーバ間のリンクはセキュアです。したがって、アプライアンスと物理サーバ間のデータ転送を暗号化する必要はありません。ただし、アプライアンスとサーバー間のデータ転送を暗号化することで、エンドツーエンド暗号化を提供できます。詳細については、エンドツーエンドの暗号化を使用した SSL オフロードの構成を参照してください。

注： SSL ベースの仮想サーバーにサービスをバインドする前に、ADC アプライアンスで負荷分散機能を有効にしてください。

CLI を使用してサービスを仮想サーバーにバインドします

コマンドプロンプトで次のコマンドを入力して、サービスを仮想サーバーにバインドし、構成を確認します。

```
bind lb vserver <name> <serviceName>
show lb vserver <name>
<!--NeedCopy-->
```

例:

```
bind lb vserver sslvs sslsvc
     Done

sh lb vserver sslvs

            sslvs (192.0.2.240:443) - SSL       Type: ADDRESS
            State: DOWN[Certkey not bound]
            Last state change was at Wed May  2 11:43:04 2018
            Time since last state change: 0 days, 00:13:21.150
            Effective State: DOWN
            Client Idle Timeout: 180 sec
```



```
12              Down state flush: ENABLED
13              Disable Primary Vserver On Down : DISABLED
14              Appflow logging: ENABLED
15              No. of Bound Services :  1 (Total)       0 (Active)
16              Configured Method: LEASTCONNECTION      BackupMethod:
                   ROUNDROBIN
17              Mode: IP
18              Persistence: NONE
19              Vserver IP and Port insertion: OFF
20              Push: DISABLED  Push VServer:
21              Push Multi Clients: NO
22              Push Label Rule: none
23              L2Conn: OFF
24              Skip Persistency: None
25              Listen Policy: NONE
26              IcmpResponse: PASSIVE
27              RHIstate: PASSIVE
28              New Service Startup Request Rate: 0 PER_SECOND, Increment
                   Interval: 0
29              Mac mode Retain Vlan: DISABLED
30              DBS_LB: DISABLED
31              Process Local: DISABLE
32              Traffic Domain: 0
33              TROFS Persistence honored: ENABLED
34              Retain Connections on Cluster: NO
35      1) sslsvc (198.51.100.225: 443) - SSL State: DOWN       Weight: 1
36       Done
37  <!--NeedCopy-->
```

CLI を使用して仮想サーバからサービスをバインド解除する

コマンドプロンプトで、次のコマンドを入力します。

```
1  unbind lb vserver <name> <serviceName>
2  <!--NeedCopy-->
```

例:

```
1  unbind lb vserver sslvs sslsvc
2       Done
3  <!--NeedCopy-->
```


GUI を使用してサービスを仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバーを開き、[サービスとサービスグループ] セクションの **[**** 負荷分散仮想サーバーサービスバインディング ******] タイルをクリックします。
3. [負荷分散仮想サーバーサービスバインド] ページで、**[**バインドの追加] タブをクリックし、**[**** サービスの選択**]** の [クリックして選択 ******] をクリックし、バインドするサービスの横にあるチェックボックスをオンにします。
4. [選択**]** をクリックして [バインド] をクリックします

複数のサイトを安全にホストするためのサーバー名表示 **(SNI)** 仮想サーバーを構成する

仮想ホスティングは、同じ IP アドレスを持つ複数のドメイン名をホストするために Web サーバーによって使用されます。アプライアンスは、透過的な SSL サービスまたは仮想サーバベースの SSL オフロードを使用して Web サーバから SSL 処理をオフロードすることにより、複数のセキュアドメインのホスティングをサポートします。ただし、複数の Web サイトが同じ仮想サーバーでホストされている場合は、想定されるホスト名が仮想サーバーに送信される前に SSL ハンドシェイクが完了します。その結果、アプライアンスは、接続の確立後にクライアントに提示する証明書を判断できません。この問題は、仮想サーバで SNI を有効にすることで解決します。SNI は、クライアントがハンドシェイクの開始時にホスト名を提供するために使用するトランスポート層セキュリティ (TLS) 拡張です。ADC アプライアンスはこのホスト名を共通名と比較し、一致しない場合はサブジェクト代替名 (SAN) と比較します。名前が一致すると、アプライアンスは対応する証明書をクライアントに提示します。

ワイルドカード SSL 証明書は、同じ組織がこれらのドメインを管理していて、第 2 レベルのドメイン名が同じ場合に、複数のサブドメインで SSL 暗号化を有効にするのに役立ちます。たとえば、一般名「*.sports.net」を使用してスポーツネットワークに発行されたワイルドカード証明書は、「login.sports.net」や「help.sports.net」などのドメインを保護するために使用できます。「login.ftp.sports.net」ドメインをセキュリティで保護することはできません。

注:
ADC アプライアンスでは、**SAN** フィールドのドメイン名、URL、電子メール ID DNS エントリのみが比較されます。

-snicert オプションを使用すると、複数のサーバー証明書を 1 つの SSL 仮想サーバーまたはトランスペアレントサービスにバインドできます。仮想サーバーまたはサービスで SNI が有効になっている場合、仮想サーバーまたはサービスはこれらの証明書を発行します。SNI はいつでも有効にできます。

CLI を使用して複数のサーバ証明書を **1** つの **SSL** 仮想サーバにバインドする

コマンドプロンプトで次のコマンドを入力して SNI を構成し、構成を確認します。

```
set ssl vserver <vServerName>@ [-SNIEnable ( ENABLED | DISABLED )]

```



```
bind ssl vserver <vServerName>@ -certkeyName <string> -SNICert

show ssl vserver <vServerName>
<!--NeedCopy-->
```

CLI を使用して複数のサーバ証明書を 1 つのトランスペアレントサービスにバインドするには、前述のコマンドで service を`vserver`に、service name を`vservername`に置き換えます。

注**:**`-clearTextPort 80` オプションを使用して SSL サービスを作成します。

GUI を使用して複数のサーバ証明書を **1** つの **SSL** 仮想サーバにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL 仮想サーバーを開き、[証明書] で [サーバー証明書] を選択します。
3. 証明書を追加するか、一覧から証明書を選択して、[**SNI** のサーバー証明書] をクリックします。
4. **[**詳細設定**]** で [**SSL** パラメータ] を選択します。
5. [**SNI** 有効化] をクリックします。

バックエンドサービスでの **SNI** のサポート

> 注:SNI は DTLS バックエンドサービスではサポートされていません。

Citrix ADC アプライアンスは、バックエンドでサーバー名表示（SNI）をサポートしています。つまり、ハンドシェイクが正常に完了するために、共通名がクライアント hello のサーバ名としてバックエンドサーバに送信されます。このサポートは、連邦政府システムインテグレータのお客様のセキュリティ要件を満たすのに役立ちます。また、SNI には、ファイアウォールで数百もの異なる IP アドレスやポートを開くのではなく、1 つのポートしか使用できないという利点があります。

連邦システムインテグレーターのお客様のセキュリティ要件には、2012R2 および WAP サーバーにおける Active Directory フェデレーションサービス (ADFS) 3.0 のサポートが含まれます。この要件を満たすには、Citrix ADC アプライアンスのバックエンドで SNI をサポートする必要があります。

> 注:
>
> SNI を機能させるには、クライアント hello のサーバ名が、SSL 仮想サーバにバインドされたバックエンドサービスに設定されたホスト名と一致する必要があります。たとえば、バックエンドサーバーのホスト名が www.mail.example.com の場合、SNI 対応バックエンドサービスはサーバー名をhttps://www.mail.example.comとして構成する必要があります。このホスト名は、クライアント hello のサーバ名と一致する必要があります。

バックエンドサービスでの動的 **SNI** のサポート

Citrix ADC アプライアンスは、バックエンド TLS 接続で動的 SNI をサポートします。つまり、アプライアンスはクライアント接続で SNI を学習し、サーバー側接続で SNI を使用します。SSL サービス、サービスグループ、または

プロファイルに共通名を指定する必要がなくなりました。Client Hello メッセージの SNI 拡張で受信した共通名は、バックエンド SSL 接続に転送されます。

以前は、SSL サービス、サービスグループ、および SSL プロファイルに静的 SNI を設定する必要がありました。その結果、設定された静的 SNI 拡張だけがサーバに送信されました。クライアントが同時に複数のドメインにアクセスする必要がある場合、ADC アプライアンスはクライアントから受け取った SNI をバックエンドサービスに送信できませんでした。代わりに、設定された静的な共通名が送信されました。バックエンドサーバが複数のドメインに対して設定されている場合、サーバは、アプライアンスからの Client Hello メッセージで受信した SNI に基づいて、正しい証明書で応答できます。

注意点**:**

- フロントエンドで SNI を有効にし、正しい SNI 証明書を SSL 仮想サーバーにバインドする必要があります。フロントエンドで SNI を有効にしないと、SNI 情報はバックエンドに渡されません。
- サーバ認証を有効にすると、サーバ証明書が CA 証明書によって検証され、サーバ証明書のコモンネーム/SAN エントリが SNI と照合されます。そのため、CA 証明書をサービスにバインドする必要があります。
- ダイナミック SNI が有効な場合、バックエンド接続と SSL セッションの再利用は SNI に基づきます。

動的 SNI が有効な場合、SSL モニターは SNI を送信しません。SNI ベースのプロービングでは、静的 SNI が設定されているバックエンドプロファイルを SSL モニターにアタッチします。モニターは SNI と同じカスタムヘッダーで構成する必要があります。

CLI を使用してバックエンドサービスで **SNI** を設定する

コマンドプロンプトで入力します。

```
add service <name>  <IP>  <serviceType>  <port>

add lb vserver <name>  <IPAddress> <serviceType>  <port>

bind lb vserver <name> <serviceName>

set ssl service <serviceName> -SNIEnable ENABLED -commonName <string>

set ssl profile <name> -SNIEnable ENABLED
<!--NeedCopy-->
```

例**:**

```
    add service service_ssl 198.51.100.100 SSL 443

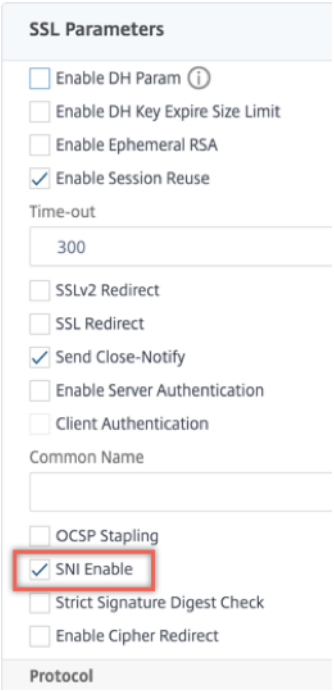
```



```
3     add lb vserver ssl-vs 203.0.113.200 SSL 443
4
5     bind lb vserver ssl-vs service_ssl
6
7     set ssl service service_ssl -SNIEnable ENABLED  - commonName www.
          example.com
8
9     set ssl profile sslprof -SNIEnable ENABLED
10 <!--NeedCopy-->
```

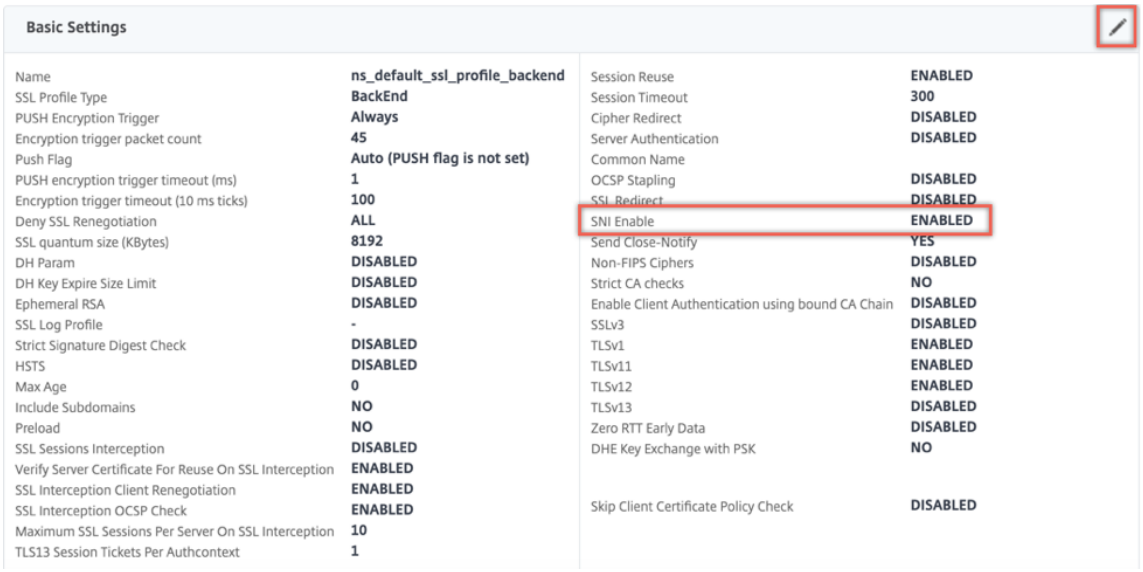
GUI を使用してバックエンドサービスで **SNI** を設定する

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. SSL サービスを選択し、[詳細設定] で [**SSL** パラメータ] をクリックします。
3. [**SNI** 有効化] をクリックします。



GUI を使用して **SSL** プロファイルで **SNI** を設定する

1. システム **>** プロファイル **> SSL** プロファイルに移動します。
2. [追加] をクリックします。
3. **[**基本設定**]** で [**SNI** 有効] を選択します。



4. [**OK**] をクリックします。

セキュアモニタを **SNI** 対応バックエンドサービスにバインドする

HTTP、HTTP-ECV、TCP、または TCP-ECV タイプのセキュアモニタを、SNI をサポートするバックエンドサービスおよびサービスグループにバインドできます。ただし、動的 SNI が有効な場合、モニタプローブは SNI 拡張を送信しません。SNI プローブを送信するには、バックエンド SSL プロファイルで静的 SNI を有効にし、プロファイルをモニタにバインドします。モニタのカスタムヘッダーを、モニタプローブの client hello で SNI 拡張として送信されるサーバ名に設定します。

CLI を使用して、セキュアモニタを設定して **SNI** 対応バックエンドサービスにバインドする

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> -secure YES
add ssl profile <name> -sslProfileType BackEnd
set lb monitor <monitorName> <type> -customHeaders <string> -sslprofile
    <backend ssl profile>
set ssl profile <name> -sniEnable ENABLED -commonName <string>
bind service <name> -monitorName <string>
<!--NeedCopy-->
```

例**:**

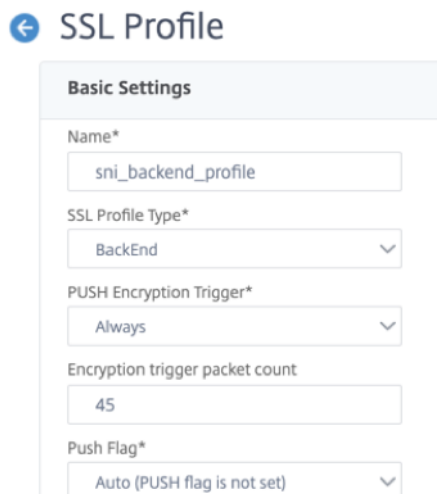
```
add ssl profile sni_backend_profile -sslProfileType BackEnd
```



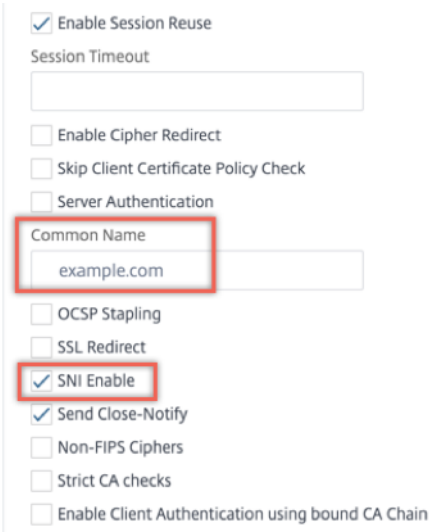
```
2 set ssl profile sni_backend_profile -sniEnable ENABLED -commonName
    example.com
3 add lb monitor http-ecv-mon HTTP-ECV -secure YES
4 set monitor http-ecv-mon HTTP-ECV -customHeaders "Host: example.com\r\n
    " -sslprofile sni_backend_profile
5 bind service ssl_service  - monitorName http-ecv-mon
6 <!--NeedCopy-->
```

GUI を使用して、セキュアモニタを構成し、**SNI** 対応バックエンドサービスにバインドする

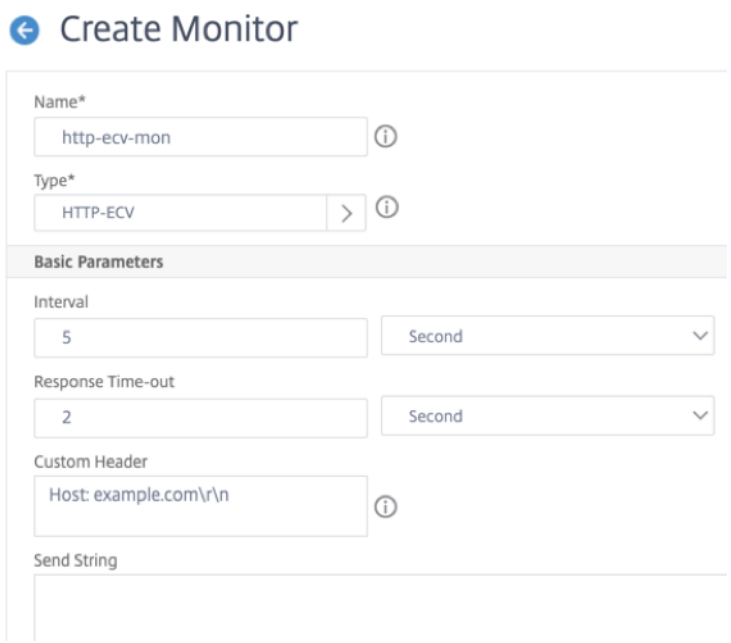
1. システム **>** プロファイル **> SSL** プロファイルに移動します。
2. [追加] をクリックします。
3. プロファイルの名前を指定し、[**SSL** プロファイルタイプ] で [バックエンド] を選択します。



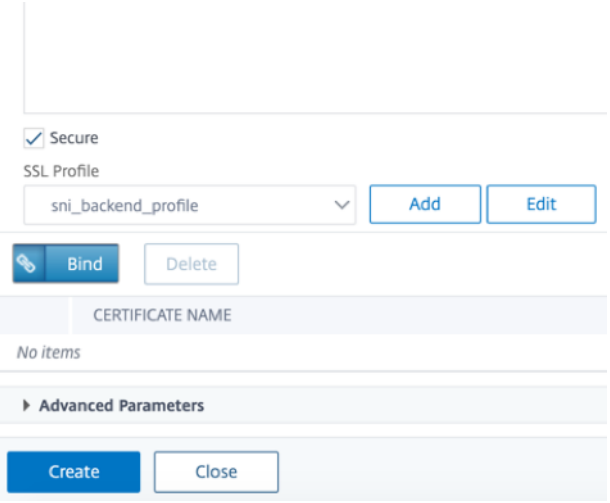
4. 共通名 (ホストヘッダーと同じ) を指定し、[**SNI Enable**] を選択します。



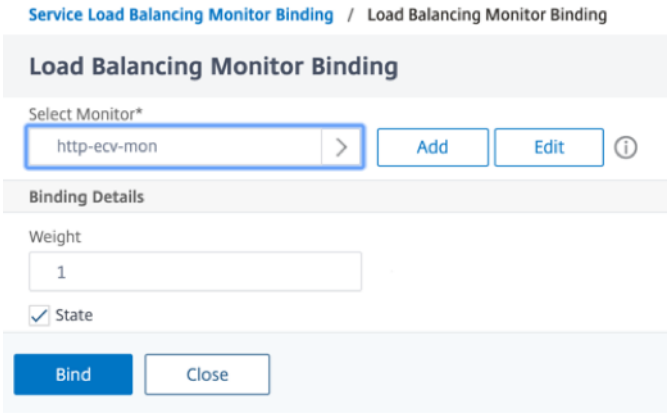
5. [**OK**] をクリックします。
6. [トラフィック管理] **>** [負荷分散] **>** [モニタ] に移動します。
7. [追加] をクリックします。
8. モニターの名前を指定します。[タイプ] で、[HTTP]、[HTTP-ECV]、[TCP]、または [TCP-ECV] を選択します。
9. カスタムヘッダーを指定します。



10. [安全] を選択します。
11. [**SSL** プロファイル] で、前の手順で作成したバックエンド SSL プロファイルを選択します。
12. [**Create**] をクリックします。



13. [**Traffic Management**] > [**Load Balancing**] > [**Services**] の順に移動します。
14. SSL サービスを選択し、[編集] をクリックします。
15. [モニター] で、[バインドの追加] をクリックし、前の手順で作成したモニターを選択して [バインド] をクリックします。



GUI を使用してセキュアモニタを構成し、**SNI** 対応バックエンドサービスにバインドする

1. トラフィック管理 > 負荷分散 > モニタに移動します。
2. **HTTP-ECV** または TCP-ECV**** タイプのモニターを追加し、**[**** カスタムヘッダー] を指定します。
3. [**Create**] を選択します。
4. トラフィック管理 > 負荷分散 > サービスに移動します。
5. SSL サービスを選択し、[編集] をクリックします。
6. [モニター] で [バインドの追加] をクリックし、手順 3 で作成したモニターを選択して [バインド] をクリックします。

証明書とキーのペアを追加または更新する

> 注:
>
> 既存の証明書とキーがない場合は、「証明書の作成」を参照してください。
>
> ECDSA [証明書とキーのペアを作成するには、ECDSA 証明書とキーのペアの作成] をクリックします。
>
> ビルド 41.x から、63 文字までの証明書名がサポートされます。
>
> リリース 13.0 ビルド 79.x からは、パスワードで保護された証明書とキーのペアが常に正常に追加されます。以前は、Citrix ADC アプライアンスで強力なパスワードオプションを有効にすると、パスワードで保護された証明書とキーのペアが追加されないことがありました。ただし、以前のビルドにダウングレードすると、証明書キーの設定は失われます。また、証明書とキーのペアの NITRO API レスポンスでは、`passplain`変数ではなく`passcrypt`変数が送信されます。

SSL トランザクションの場合、サーバーには有効な証明書と、対応する秘密鍵と公開鍵のペアが必要です。SSL データはサーバーの公開鍵で暗号化されます。公開鍵はサーバーの証明書から入手できます。復号化には、対応する秘密

鍵が必要です。SSL 証明書とキーのペアの追加時に使用される秘密キーのパスワードは、Citrix ADC アプライアンスごとに一意の暗号化キーを使用して保存されます。

ADC アプライアンスは、SSL トランザクションをサーバからオフロードします。したがって、サーバの証明書と秘密キーがアプライアンスに存在し、証明書が対応する秘密キーとペアになっている必要があります。この証明書とキーのペアは、SSL トランザクションを処理する仮想サーバーにバインドする必要があります。

注： Citrix ADC アプライアンスのデフォルトの証明書は 2048 ビットです。以前のビルドでは、デフォルトの証明書は 512 ビットまたは 1024 ビットでした。リリース 11.0 にアップグレードした後、”ns-”で始まる古い証明書とキーのペアをすべて削除し、アプライアンスを再起動して 2048 ビットのデフォルト証明書を自動的に生成する必要があります。

証明書とキーの両方をアプライアンスに追加するには、Citrix ADC アプライアンスのローカルストレージに存在する必要があります。証明書またはキーファイルがアプライアンス上にない場合は、ペアを作成する前に証明書またはキーファイルをアプライアンスにアップロードします。

重要: 証明書とキーは、デフォルトで /nsconfig/ssl ディレクトリに保存されます。証明書またはキーが他の場所に保存されている場合は、Citrix ADC アプライアンス上のファイルへの絶対パスを指定する必要があります。Citrix ADC FIPS アプライアンスは外部キー（非 FIPS キー）をサポートしていません。FIPS アプライアンスでは、ハードディスクやフラッシュメモリなどのローカルストレージデバイスからキーをロードすることはできません。FIPS キーは、アプライアンスのハードウェアセキュリティモジュール (HSM) に存在する必要があります。

Citrix ADC アプライアンスでは RSA キーのみがサポートされています。

通知期間を設定し、有効期限モニターが証明書が期限切れになる前にプロンプトを発行できるようにします。

Citrix ADC アプライアンスは、証明書と秘密キーファイルの次の入力形式をサポートしています。

- PEM-プライバシー強化メール
- DER-識別符号化規則
- PFX-個人情報交換

ソフトウェアはフォーマットを自動的に検出します。そのため、inform パラメーターで形式を指定する必要はなくなりました。フォーマット (正しいか正しくない) を指定した場合、ソフトウェアはそのフォーマットを無視します。証明書と鍵ファイルの形式は同じである必要があります。

注: 証明書は、次のいずれかのハッシュアルゴリズムを使用して署名する必要があります。

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

MPX アプライアンスは、次のサイズまで、512 ビット以上の証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書

- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）
- バックエンドサーバー上の 4096 ビット証明書
- 4096 ビットのクライアント証明書 (仮想サーバーでクライアント認証が有効になっている場合)

VPX 仮想アプライアンスは、次のサイズまで、512 ビット以上の証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）
- バックエンドサーバー上の 4096 ビット証明書
- 4096 ビットのクライアント証明書 (仮想サーバーでクライアント認証が有効になっている場合)

> 注
>
> Citrix ADC SDX アプライアンスは、512 ビット以上の証明書をサポートします。アプライアンスでホストされている各 Citrix ADC VPX インスタンスは、VPX 仮想アプライアンスの前述の証明書サイズをサポートします。ただし、SSL チップがインスタンスに割り当てられている場合、そのインスタンスは MPX アプライアンスでサポートされる証明書サイズをサポートします。

CLI を使用した証明書とキーのペアの追加

コマンドプロンプトで次のコマンドを入力して、証明書とキーのペアを追加し、構成を確認します。

```
add ssl certKey <certkeyName> -cert <string>[(-key <string> [-password
    ]) | -fipsKey <string>] [-inform ( DER | PEM )] [<passplain>] [-
    expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]]

show ssl certKey [<certkeyName>]
<!--NeedCopy-->
```

例:

```
add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -
    password ssl -expiryMonitor ENABLED -notificationPeriod 30
 Done
Note: For FIPS appliances, replace -key with -fipskey

show ssl certKey sslckey
        Name: sslckey           Status: Valid,   Days to expiration
            :8418
```



```
        Version: 3
        Serial Number: 01
        Signature Algorithm: md5WithRSAEncryption
        Issuer:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.root.com
        Validity
                Not Before: Jul 15 02:25:01 2005 GMT
                Not After : Nov 30 02:25:01 2032 GMT
        Subject:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.server.com
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
 Done
<!--NeedCopy-->
```

CLI を使用して証明書とキーのペアを更新または削除する

証明書とキーのペアの有効期限モニタまたは通知期間を変更するには、`set ssl certkey` コマンドを使用します。証明書とキーのペアの証明書またはキーを置き換えるには、`update ssl certkey` コマンドを使用します。`update ssl certkey`コマンドには、ドメインチェックをオーバーライドするための追加パラメータがあります。どちらのコマンドでも、既存の証明書とキーのペアの名前を入力します。SSL 証明書とキーのペアを削除するには、`rm ssl certkey`コマンドを使用します。このコマンドでは<certkeyName>引数のみを受け付けます。

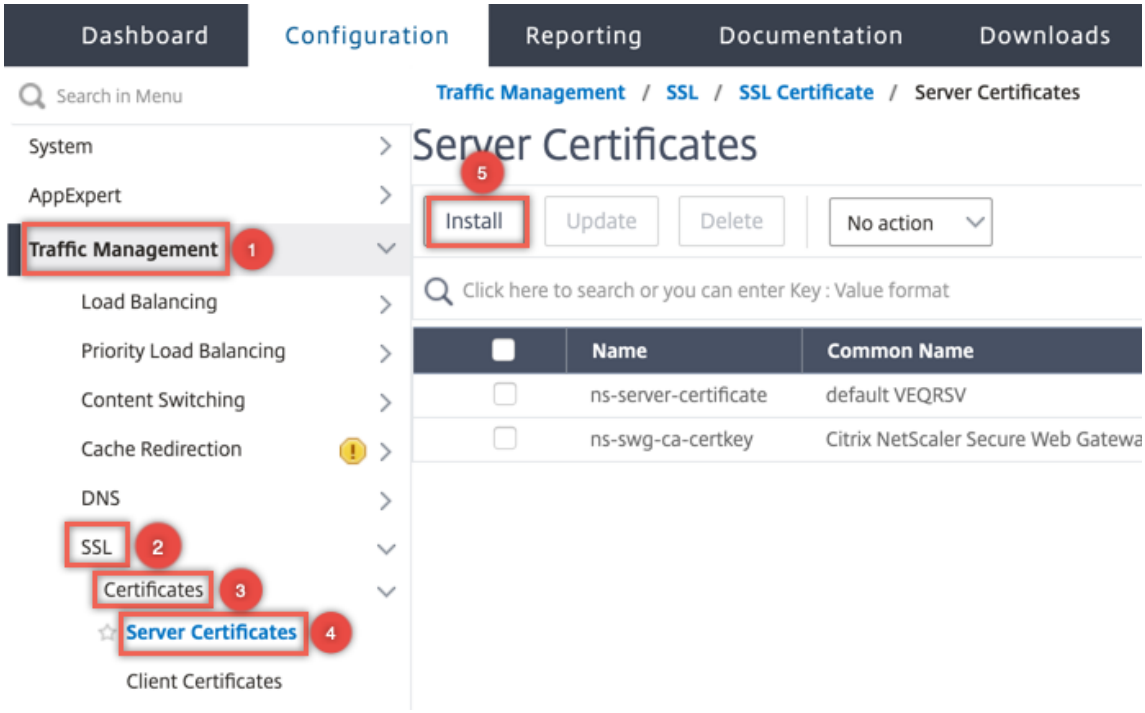
例**:**

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED )
       [-notificationPeriod <positive_integer>]]

update ssl certKey <certkeyName> [-cert <string> [-password]] [-key
       <string> | -fipsKey <string>] [-inform <inform>] [-noDomainCheck
          ]
<!--NeedCopy-->
```

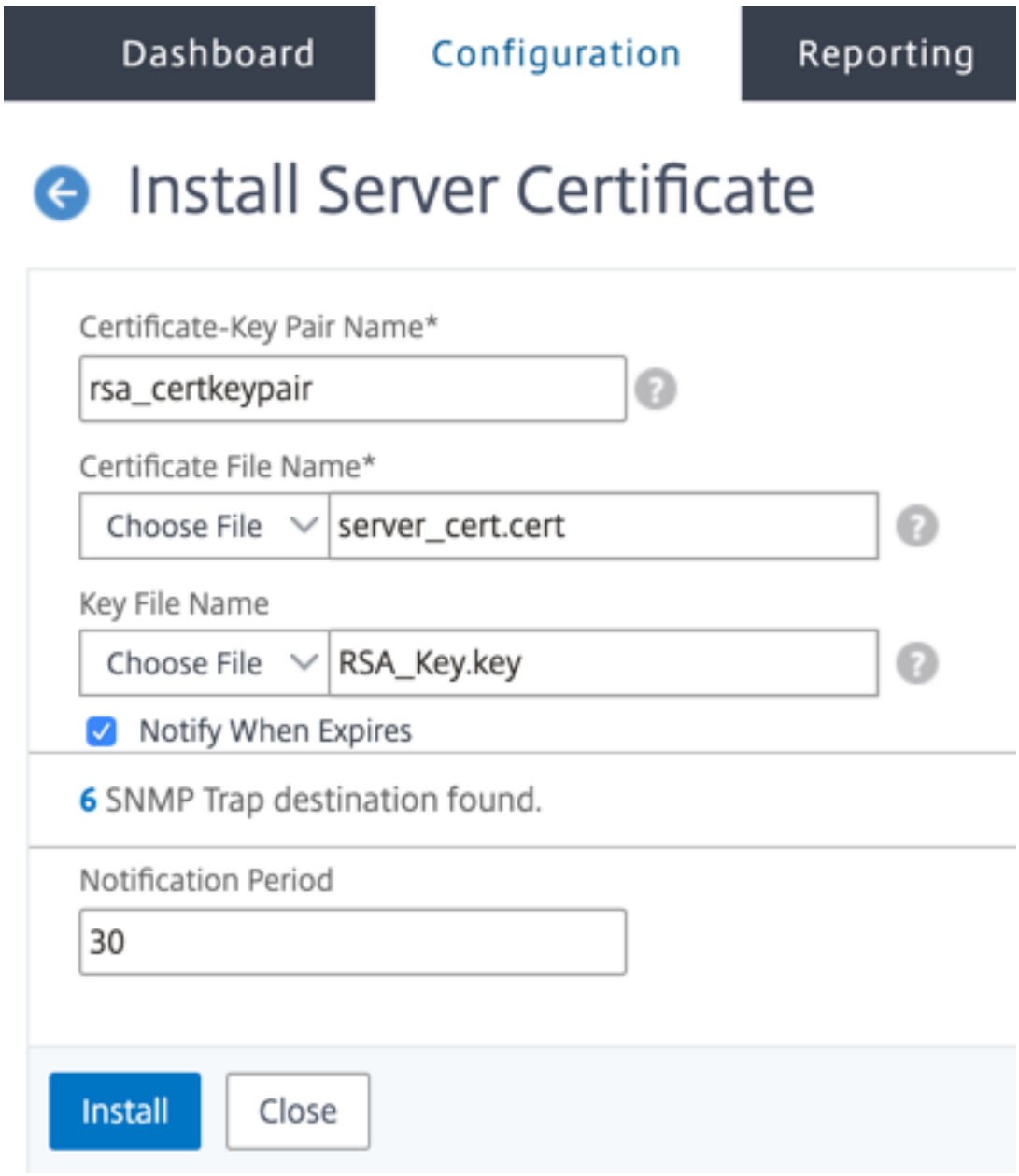
GUI を使用して証明書とキーのペアを追加または更新する

1. トラフィック管理 **>** **SSL >** 証明書 **>** サーバに移動します。



2. 次のパラメータの値を入力し、[**Install**] をクリックします。

   - 証明書とキーのペア名-証明書と秘密キーのペアの名前。
   - 証明書ファイル名-認証局から受け取った署名付き証明書。
   - [Key File Name]：証明書とキーのペアを形成するために使用される秘密キーファイルの名前と、オプションでパスです。



証明書とキーのペアを **SSL** 仮想サーバーにバインドする

重要: 証明書を SSL 仮想サーバーにバインドする前に、中間証明書をこの証明書にリンクします。証明書のリンクの詳細については、「証明書のチェーンを作成する」を参照してください。

SSL トランザクションの処理に使用される証明書は、SSL データを受信する仮想サーバーにバインドする必要があります。SSL データを受信する仮想サーバが複数ある場合は、有効な証明書とキーのペアをそれぞれにバインドする必要があります。

Citrix ADC アプライアンスにアップロードした有効な既存の SSL 証明書を使用します。テスト目的の代わりに、アプライアンスに独自の SSL 証明書を作成します。アプライアンスで FIPS キーを使用して作成された中間証明書は、

SSL 仮想サーバーにバインドできません。

SSL ハンドシェイク中、クライアント認証中の証明書要求メッセージに、サーバはサーバにバインドされているすべての認証局（CA）の識別名（DN）を一覧表示します。サーバーは、このリストからのみクライアント証明書を受け入れます。特定の CA 証明書の DN 名を SSL クライアントに送信したくない場合は、skipCA フラグを設定します。この設定は、特定の CA 証明書の識別名を SSL クライアントに送信してはならないことを示します。

独自の証明書を作成する方法の詳細については、「証明書の管理」を参照してください。

注：信頼できる認証局が発行する有効な SSL 証明書のみを使用することをお勧めします。

CLI を使用して **SSL** 証明書とキーのペアを仮想サーバにバインドする

コマンドプロンプトで次のコマンドを入力して、SSL 証明書とキーのペアを仮想サーバーにバインドし、構成を確認します。

```
1 -  bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName
     > -CA -skipCAName
2 -  show ssl vserver <vServerName>
3 <!--NeedCopy-->
```

例:

```
1 bind ssl vs vs1 -certkeyName cert2 -CA -skipCAName
2  Done
3 sh ssl vs vs1
4
5  Advanced SSL configuration for VServer vs1:
6
7  DH: DISABLED
8
9  Ephemeral RSA: ENABLED Refresh Count: 0
10
11  Session Reuse: ENABLED Timeout: 120 seconds
12
13  Cipher Redirect: DISABLED
14
15  SSLv2 Redirect: DISABLED
16
17  ClearText Port: 0
18
19  Client Auth: DISABLED
20
```



```
 SSL Redirect: DISABLED

 Non FIPS Ciphers: DISABLED

 SNI: DISABLED

 OCSP Stapling: DISABLED

 HSTS: DISABLED

 IncludeSubDomains: NO

 HSTS Max-Age: 0

 SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED
    TLSv1.2: DISABLED

 Push Encryption Trigger: Always

 Send Close-Notify: YES

 Strict Sig-Digest Check: DISABLED

ECC Curve: P_256, P_384, P_224, P_521

 1) CertKey Name: cert1 CA Certificate OCSPCheck: Optional CA_Name Sent
 2) CertKey Name: cert2 CA Certificate OCSPCheck: Optional CA_Name
    Skipped
 1) Cipher Name: DEFAULT

Description: Default cipher list with encryption strength >= 128bit
Done
<!--NeedCopy-->
```

CLI を使用して **SSL** 証明書とキーのペアを仮想サーバからバインド解除する

`unbind ssl certKey <certkeyName>` コマンドを使用して証明書とキーのペアを仮想サーバからバインド解除しようとすると、エラーメッセージが表示されます。このエラーは、コマンドの構文が変更されたために表示されます。コマンドプロンプトで、次のコマンドを入力します。

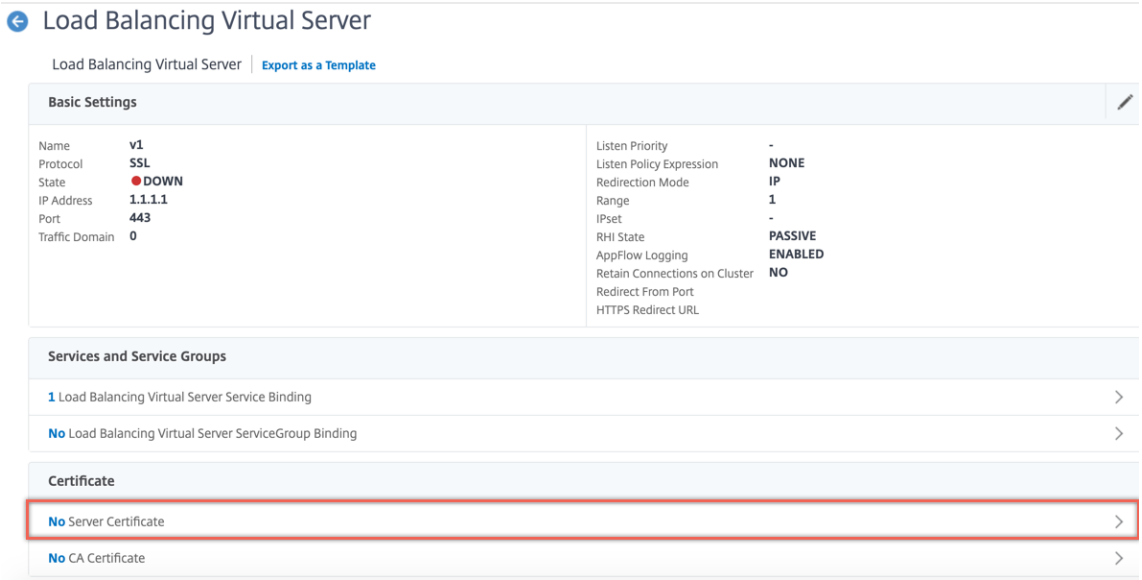
```
unbind ssl vserver <vServerName> -certkeyName <string>
<!--NeedCopy-->
```


例:

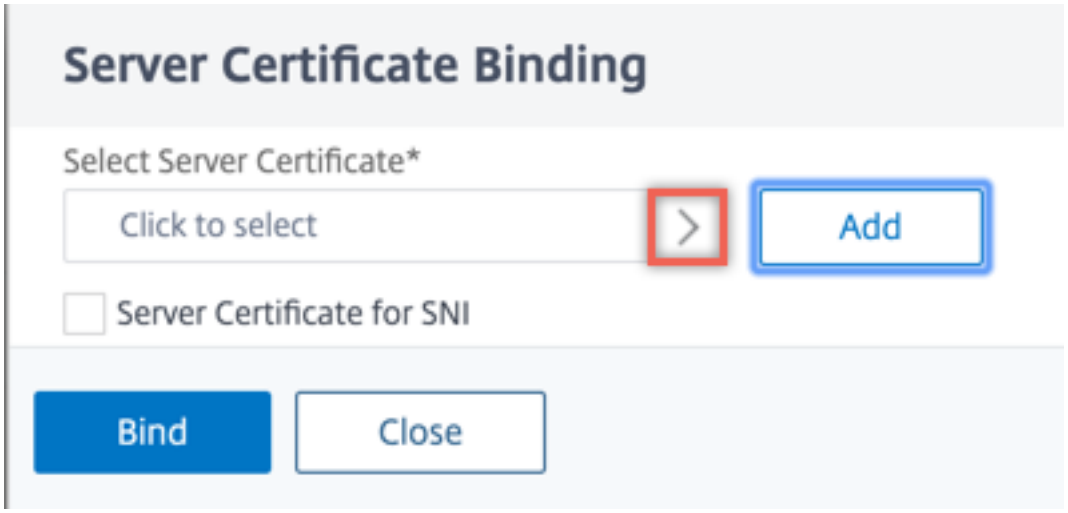
```
unbind ssl vserver vssl -certkeyName sslckey
<!--NeedCopy-->
```

GUI を使用して **SSL** 証明書とキーのペアを仮想サーバにバインドする

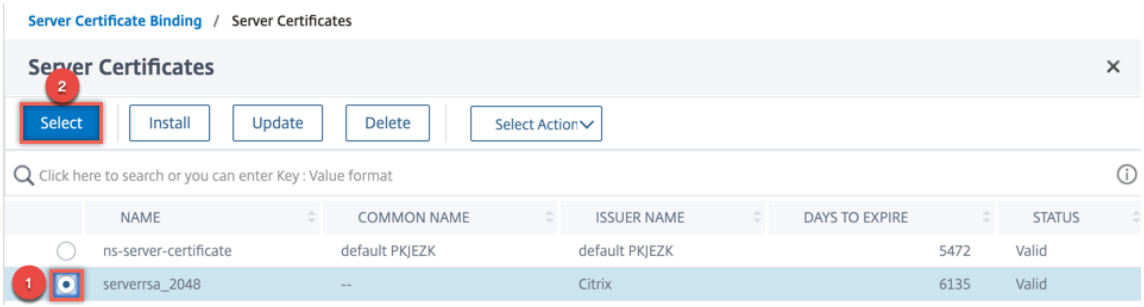
1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。[証明書] セクション内をクリックします。



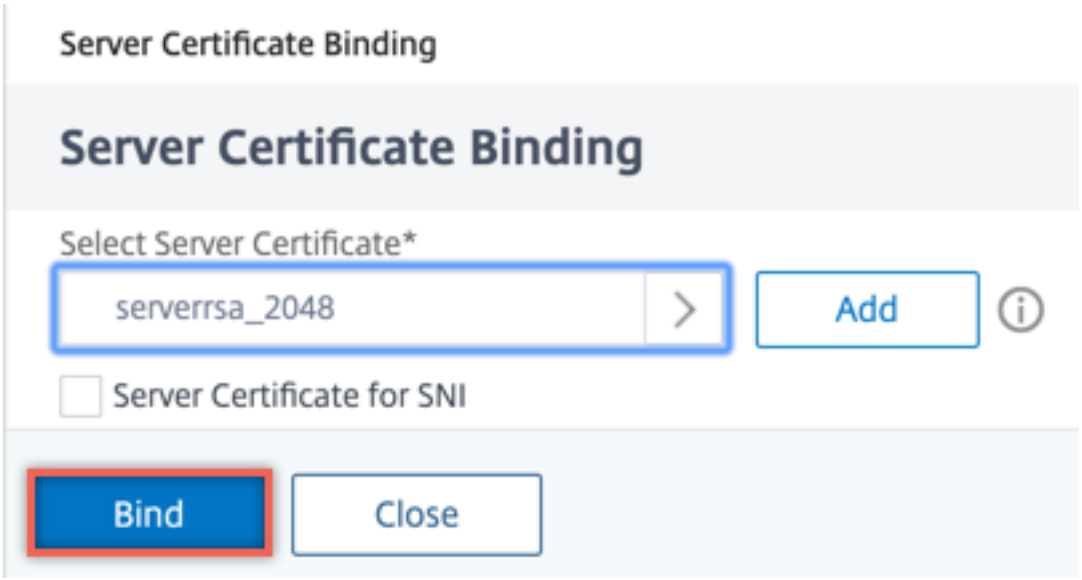
2. 矢印をクリックして、証明書とキーのペアを選択します。



3. リストから証明書とキーのペアを選択します。



4. 証明書とキーのペアを仮想サーバーにバインドします。サーバー証明書を SNI 証明書として追加するには、SNI の [サーバー証明書] を選択します。



SSL 仮想サーバパラメータ

SSL 仮想サーバの高度な SSL 構成を設定します。SSL プロファイルでは、これらのパラメータの多くを設定することもできます。SSL プロファイルで設定できるパラメータの詳細については、SSL プロファイルパラメータを参照してください。

CLI を使用した SSL 仮想サーバーのパラメータの設定

コマンドプロンプトで入力します。

```
set ssl vserver <vServerName>@ [-clearTextPort <port>] [-dh ( ENABLED |
    DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-
    dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED |
    DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED |
    DISABLED )[-sessTimeout <positive_integer>]] [-cipherRedirect (
```



```
      ENABLED | DISABLED ) [-cipherURL <URL>]] [-sslv2Redirect ( ENABLED |
       DISABLED )[-sslv2URL <URL>]] [-clientAuth ( ENABLED | DISABLED ) [-
      clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
      DISABLED )][-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl2 (
      ENABLED | DISABLED )] [-ssl3 ( ENABLED | DISABLED )] [-tls1 (
      ENABLED | DISABLED )] [-tls11 ( ENABLED | DISABLED )] [-tls12 (
      ENABLED | DISABLED )][-tls13 ( ENABLED | DISABLED )] [-SNIEnable (
      ENABLED | DISABLED )][-ocspStapling ( ENABLED | DISABLED )] [-
      pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES | NO )] [-
      dtlsProfileName <string>] [-sslProfile <string>] [-HSTS ( ENABLED |
      DISABLED )][-maxage <positive_integer>] [-IncludeSubdomains ( YES |
      NO )][-strictSigDigestCheck ( ENABLED | DISABLED )] [-
      zeroRttEarlyData (ENABLED | DISABLED )] [-
      tls13SessionTicketsPerAuthContext <positive_integer>] [-
      dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

Diffie-Hellman (DH) パラメーター

SSL トランザクションを設定するために DH キー交換を必要とする暗号をアプライアンスで使用するには、アプライアンスで DH キーエクスチェンジを有効にします。ネットワークに基づいて他の設定を構成します。

CLI を使用して DH パラメータを設定する必要がある暗号の一覧を表示するには、sh cipher DH と入力します。

設定ユーティリティを使用して DH パラメータを設定する必要がある暗号を一覧表示するには、[トラフィック管理] **> [SSL] >** [暗号グループ] に移動し、[**DH**] をダブルクリックします。

DH キー交換を有効にする方法の詳細については、Diffie-Hellman (DH) キーの生成を参照してください。

CLI を使用した **DH** パラメータの設定

コマンドプロンプトで次のコマンドを入力して DH パラメータを構成し、構成を確認します。

```
-  `set ssl vserver <vserverName> -dh <Option> -dhCount <
   RefreshCountValue> -filepath <string>
-  show ssl vserver <vServerName>`
<!--NeedCopy-->
```

例:

```
set ssl vserver vs-server -dh ENABLED  -dhFile /nsconfig/ssl/ns-server.
   cert -dhCount 1000
```



```
Done

show ssl vserver vs-server

            Advanced SSL configuration for VServer vs-server:
            DH: ENABLED
            Ephemeral RSA: ENABLED          Refresh Count: 1000
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
            HSTS Max-Age: 0
            SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
                ENABLED  TLSv1.2: ENABLED

    1)      Cipher Name: DEFAULT
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用して **DH** パラメータを設定する

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[**DH Param** を有効にする] を選択し、更新回数とファイルパスを指定します。

エフェメラル **RSA**

Ephemeral RSA を使用すると、サーバー証明書がエクスポートクライアント (1024 ビット証明書) をサポートしていない場合でも、エクスポートクライアントはセキュアサーバーと通信できます。エクスポートクライアントがセキュア Web オブジェクトまたはリソースにアクセスできないようにするには、エフェメラル RSA キー交換を無効にする必要があります。

デフォルトでは、この機能は Citrix ADC アプライアンスで有効になっており、更新カウントはゼロ（無限使用）に設定されています。

注:

エクスポート暗号を SSL または TCP ベースの SSL 仮想サーバーまたはサービスにバインドすると、エフェメラル RSA キーが自動的に生成されます。エクスポート暗号を削除しても、ErSA キーは削除されません。後で別のエクスポート暗号が SSL または TCP ベースの SSL 仮想サーバーまたはサービスにバインドされるときに再利用されます。ErSA キーは、システムの再起動時に削除されます。

CLI を使用してエフェメラル **RSA** を構成する

コマンドプロンプトで次のコマンドを入力して、エフェメラル RSA を構成し、構成を確認します。

```
set ssl vserver <vServerName> -eRSA (enabled | disabled) -eRSACount <
   positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set ssl vserver vs-server -eRSA ENABLED -eRSACount 1000
 Done

show ssl vserver vs-server

        Advanced SSL configuration for VServer vs-server:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
            ENABLED  TLSv1.2: ENABLED

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
```



```
<!--NeedCopy-->
```

GUI を使用してエフェメラル **RSA** を構成する

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[エフェメラル **RSA** を有効にする] を選択し、更新回数を指定します。

セッション再利用

SSL トランザクションの場合、初期 SSL ハンドシェイクを確立するには、CPU を集中的に使用する公開キー暗号化操作が必要です。ほとんどのハンドシェイク操作は、SSL セッションキー (クライアントキー交換メッセージ) の交換に関連付けられています。クライアントセッションがしばらくアイドル状態になってから再開されると、通常、SSL ハンドシェイクが最初からやり直されます。セッション再利用を有効にすると、クライアントから受け取ったセッション再開要求に対するセッション鍵の交換が回避されます。

セッションの再利用は、Citrix ADC アプライアンスではデフォルトで有効になっています。この機能を有効にすると、サーバーの負荷が軽減され、応答時間が短縮され、サーバーがサポートできる 1 秒あたりの SSL トランザクション (TPS) の数が増加します。

CLI を使用してセッションの再利用を設定する

コマンドプロンプトで次のコマンドを入力して、セッションの再利用を構成し、構成を確認します。

```
set ssl vserver <vServerName> -sessReuse ( ENABLED | DISABLED ) -
   sessTimeout <positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例**:**

```
set ssl vserver vs-ssl -sessreuse enabled -sesstimeout 600
 Done

show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: DISABLED
```



```
11          SSLv2 Redirect: DISABLED
12          ClearText Port: 0
13          Client Auth: DISABLED
14          SSL Redirect: DISABLED
15          Non FIPS Ciphers: DISABLED
16          SNI: DISABLED
17          OCSP Stapling: DISABLED
18          HSTS: DISABLED
19          HSTS IncludeSubDomains: NO
20          HSTS Max-Age: 0
21          SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
                ENABLED  TLSv1.2: ENABLED
22
23  1)      CertKey Name: Auth-Cert-1       Server Certificate
24
25  1)      Cipher Name: DEFAULT
26          Description: Predefined Cipher Alias
27   Done
28  <!--NeedCopy-->
```

GUI を使用してセッションの再利用を構成する

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[セッション再利用を有効にする] を選択し、セッションをアクティブにしておく時間を指定します。

SSL プロトコル設定

Citrix ADC アプライアンスは、SSLv3、TLSv1、TLSv1.1、および TLSv1.2 プロトコルをサポートしています。これらの各プロトコルは、導入環境およびアプライアンスに接続するクライアントのタイプに応じて、アプライアンス上で設定できます。

TLS プロトコルバージョン 1.0、1.1、および 1.2 は、古いバージョンの TLS/SSL プロトコルよりも安全性が高くなっています。ただし、レガシーシステムをサポートするために、多くの TLS 実装では SSLv3 プロトコルとの下位互換性が維持されています。SSL ハンドシェイクでは、Citrix ADC アプライアンスで構成されたクライアントと SSL 仮想サーバーに共通する最新のプロトコルバージョンが使用されます。

最初のハンドシェイク試行では、TLS クライアントはサポートしている最も高いプロトコルバージョンを提供します。ハンドシェイクが失敗した場合、クライアントはより低いプロトコルバージョンを提供します。たとえば、TLS バージョン 1.1 のハンドシェイクが成功しなかった場合、クライアントは TLSv1.0 プロトコルを提供して再ネゴシエーションを試みます。この試行が失敗した場合、クライアントは SSLv3 プロトコルを使用して再試行します。「Man in the Middle」（MITM）攻撃者は、最初のハンドシェイクを破り、SSLv3 プロトコルとの再ネゴシエーションをトリガーし、SSLv3 の脆弱性を悪用する可能性があります。このような攻撃を軽減するには、SSLv3 を無効にするか、

ダウングレードされたプロトコルを使用した再ネゴシエーションを許可しないようにします。ただし、導入にレガシーシステムが含まれている場合、この方法は実用的ではない可能性があります。別の方法として、クライアント要求内のシグナリング暗号スイート値 (TLS_FALLBACK_SCSV) を認識する方法があります。

クライアントの hello メッセージの TLS_FALLBACK_SCSV 値は、クライアントが以前に上位のプロトコルバージョンで接続しようとしたことがあり、現在の要求がフォールバックであることを仮想サーバーに示します。仮想サーバがこの値を検出し、クライアントが指定したバージョンよりも新しいバージョンをサポートしている場合、接続は拒否され、致命的な警告が表示されます。ハンドシェイクは、次のいずれかの条件が満たされた場合に成功します。

- TLS_FALLBACK_SCSV 値は、クライアントの hello メッセージには含まれません。
- クライアント hello のプロトコルバージョンは、仮想サーバでサポートされる最も高いプロトコルバージョンです。

CLI を使用して **SSL** プロトコルサポートを設定する

コマンドプロンプトで次のコマンドを入力して SSL プロトコルサポートを構成し、構成を確認します。

```
set ssl vserver <vServerName> -ssl2 ( ENABLED | DISABLED ) -ssl3 (
    ENABLED | DISABLED ) -tls1 ( ENABLED | DISABLED ) -tls11 ( ENABLED |
     DISABLED ) -tls12 ( ENABLED | DISABLED )

show ssl vserver <vServerName>
<!--NeedCopy-->
```

例**:**

```
set ssl vserver vs-ssl -tls11 ENABLED -tls12 ENABLED
Done

sh ssl vs vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
            DH: DISABLED
            Ephemeral RSA: ENABLED                              Refresh
               Count: 0
            Session Reuse: ENABLED                                Timeout
               : 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
```



```
15              Non FIPS Ciphers: DISABLED
16              SNI: DISABLED
17              SSLv2: DISABLED        SSLv3: ENABLED    TLSv1.0: ENABLED
                   TLSv1.1: ENABLED  TLSv1.2: ENABLED
18              Push Encryption Trigger: Always
19              Send Close-Notify: YES
20              1 bound certificate:
21
22      1)      CertKey Name: mycert  Server Certificate
23              1 configured cipher:
24
25      1)      Cipher Name: DEFAULT
26              Description: Predefined Cipher Alias
27
28  Done
29  <!--NeedCopy-->
```

GUI を使用して **SSL** プロトコルサポートを構成する

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、有効にするプロトコルを選択します。

閉じる通知

クローズ通知は、SSL データ伝送の終了を示す安全なメッセージです。クローズ通知設定はグローバルレベルで必要です。この設定は、すべての仮想サーバ、サービス、およびサービスグループに適用されます。グローバル設定の詳細については、このページの「グローバル SSL パラメータ」を参照してください。

グローバル設定に加えて、close-notify パラメータを仮想サーバ、サービス、またはサービスグループレベルで設定できます。したがって、1 つのエンティティに対してパラメータを設定し、別のエンティティに対してはパラメータを設定解除できる柔軟性があります。ただし、このパラメータは必ずグローバルレベルで設定してください。そうしないと、エンティティレベルの設定は適用されません。

CLI を使用して、エンティティレベルでクローズ通知を設定する

コマンドプロンプトで、次のいずれかのコマンドを入力して close-notify 機能を構成し、構成を確認します。

1. 仮想サーバレベルで構成するには、以下のように入力します。

```
1  set ssl vserver <vServerName> -sendCloseNotify ( YES | NO )
2  show ssl vserver <vServerName>
3  <!--NeedCopy-->
```


1. をサービスレベルで設定するには、次のように入力します。

```
set ssl service <serviceName> -sendCloseNotify ( YES | NO )
show ssl service <serviceName>
<!--NeedCopy-->
```

1. サービスグループレベルで設定するには、次のように入力します。

```
set ssl serviceGroup <serviceGroupName> -sendCloseNotify ( YES | NO )
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

例**:**

```
set ssl vserver sslvsvr -sendCloseNotify YES

Done
<!--NeedCopy-->
```

GUI を使用してエンティティレベルでクローズ通知機能を設定する

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動し、仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[クローズ通知を送信] を選択します。

グローバル **SSL** パラメータ

SSL 設定を高度にカスタマイズすることで、特定の問題に対処できます。`set ssl parameter` コマンドまたは構成ユーティリティを使用して、次の項目を指定できます。

- SSL トランザクションに使用される量子サイズ。
- CRL メモリサイズ。
- OCSP キャッシュサイズ。
- SSL 再ネゴシエーションを拒否する
- 復号化されたレコード、暗号化されたレコード、またはすべてのレコードに PUSH フラグを設定します。
- クライアントが 1 つのドメインに対してハンドシェイクを開始し、別のドメインに対して HTTP 要求を送信した場合は、要求をドロップします。
- 暗号化がトリガーされるまでの時間を設定します。
  注: 指定した時刻は、

set ssl vserver コマンドまたは構成ユーティリティを使用してタイマーベースの暗号化を設定する場合にのみ適用されます。

- NDCPP 準拠証明書チェック — アプライアンスがクライアント (バックエンド接続) として動作する場合に適用されます。SSL 証明書に SAN が存在する場合は、証明書の検証時にコモンネームを無視します。
- MPX 14000 などの Cavium チップベースのアプライアンスの異種クラスタと、パケットエンジン数が異なる MPX 15000 アプライアンスなどの Intel Coleto チップベースのアプライアンスの異種クラスタを有効にします。(リリース 13.0 ビルド 47.x でサポートが追加されました)。
- バックエンドで安全な再ネゴシエーションを有効にします（リリース 1.0 ビルド 58.x からサポートが追加されました)。
- 適応型 SSL トラフィック制御（リリース 13.0 ビルド 58.x で追加されたサポート)。

CLI を使用してグローバル **SSL** パラメータを設定する

コマンドプロンプトで次のコマンドを入力して、SSL の詳細設定を構成し、構成を確認します。

```
set ssl parameter [-quantumSize <quantumSize>] [-crlMemorySizeMB <
    positive_integer>] [-strictCAChecks (YES | NO)] [-sslTriggerTimeout
    <positive_integer>] [-sendCloseNotify (YES | NO)] [-
    encryptTriggerPktCount <positive_integer>] [-denySSLReneg <
    denySSLReneg>] [-insertionEncoding (Unicode|UTF-8)] [-ocspCacheSize
    <positive_integer>][- pushFlag <positive_integer>] [-
    dropReqWithNoHostHeader (YES | NO)] [-pushEncTriggerTimeout <
    positive_integer>] [-ndcppComplianceCertCheck ( YES | NO)] [-
    heterogeneousSSLHW (ENABLED | DISABLED )]
show ssl parameter
<!--NeedCopy-->
```

例**:**

```
set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks
    no -ssltriggerTimeout 100 -sendClosenotify no -
    encryptTriggerPktCount 45 -denySSLReneg NONSECURE -insertionEncoding
     unicode -ocspCacheSize 10 -pushFlag 3 -dropReqWithNoHostHeader YES
     -pushEncTriggerTimeout 100 ms -ndcppComplianceCertCheck YES
Done

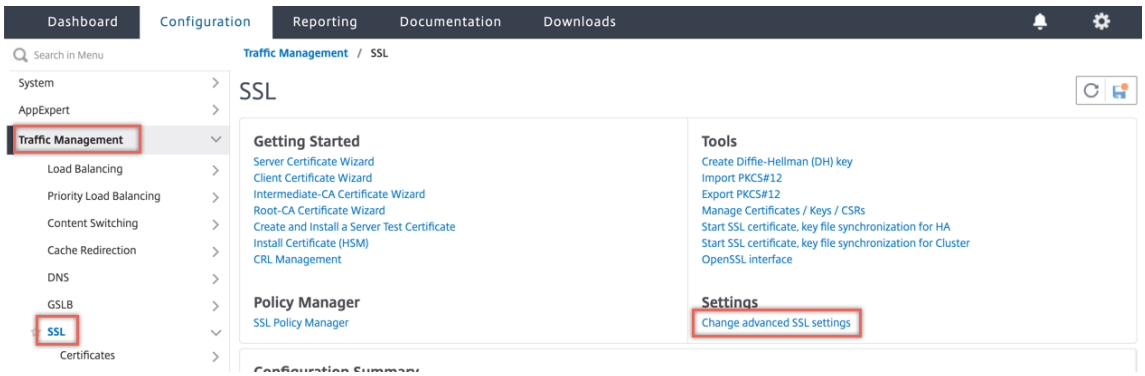
show ssl parameter
Advanced SSL Parameters
-----------------------
    SSL quantum size                                    : 8 KB
    Max CRL memory size                                 : 256 MB
```



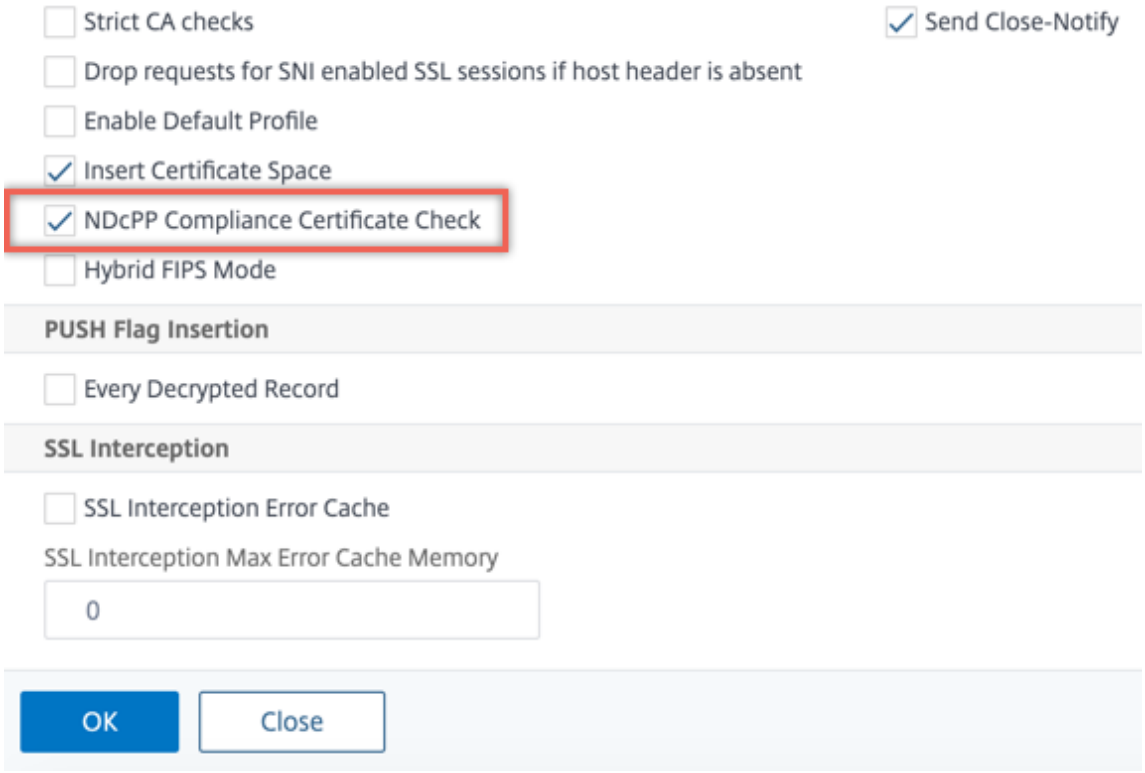
```
    Strict CA checks                                    : NO
    Encryption trigger timeout                          : 100 ms
    Send Close-Notify                                   : NO
    Encryption trigger packet count                     : 45
    Deny SSL Renegotiation                              : NONSECURE
    Subject/Issuer Name Insertion Format                : Unicode
    OCSP cache size                                     : 10 MB
    Push flag                                           : 0x3 (On
       every decrypted and encrypted record)
    Strict Host Header check for SNI enabled SSL sessions : YES
    PUSH encryption trigger timeout                     : 100 ms
    Crypto Device Disable Limit                         : 0
    Global undef action for control policies            : CLIENTAUTH
    Global undef action for data policies               : NOOP
    Default profile                                     : DISABLED
    SSL Insert Space in Certificate Header              : YES
    Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors  : NO
    Disable TLS 1.1/1.2 for dynamic and VPN services    : NO
    Software Crypto acceleration CPU Threshold          : 0
    Hybrid FIPS Mode                                    : DISABLED
    Signature and Hash Algorithms supported by TLS1.2   : ALL
    SSL Interception Error Learning and Caching         : DISABLED
    SSL Interception Maximum Error Cache Memory         : 0 Bytes
    NDCPP Compliance Certificate Check                  : YES
    Heterogeneous SSL HW (Cavium and Intel Based)       : ENABLED
 Done
<!--NeedCopy-->
```

GUI を使用して **ndCPP** 準拠証明書チェックを構成する

1. [トラフィック管理] **> [SSL**] に移動し、[設定] グループで [**SSL** の詳細設定の変更] を選択します。



2. [**NDCPP** 準拠証明書チェック] を選択します。[**OK**] をクリックします。



Citrix ADC アプライアンスのバックエンドでの安全な再ネゴシエーションのサポート

注**:** この機能は、リリース 13.0 ビルド 58.x 以降でサポートされています。以前のリリースおよびビルドでは、非セキュアな再ネゴシエーションのみがバックエンドでサポートされていました。

この機能は、次のプラットフォームでサポートされています。
• VPX
• N2 または N3 チップを含む MPX プラットフォーム
• インテル Coleto SSL チップベースのプラットフォーム

この機能は FIPS プラットフォームではまだサポートされていません。

ADC アプライアンスのバックエンドでは、セキュアな再ネゴシエーションはデフォルトで拒否されます。つまり、denySSLReneg パラメータは ALL (デフォルト) に設定されます。

バックエンドでセキュアな再ネゴシエーションを許可するには、次のいずれかのdenySSLRenegパラメータ設定を選択します。

- いいえ
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

CLI を使用してセキュアな再ネゴシエーションを有効にする

コマンドプロンプトで入力します。

```
set ssl parameter -denySSLReneg <denySSLReneg>
```

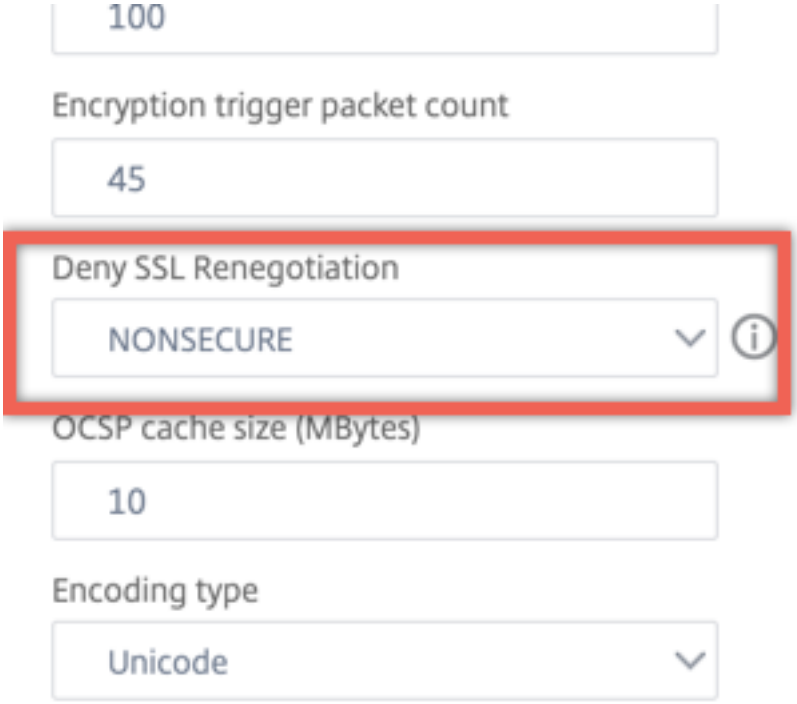
例:

```
set ssl parameter -denySSLReneg NONSECURE
 Done

sh ssl parameter
Advanced SSL Parameters
-----------------------
    SSL quantum size                                  : 8 KB
    Max CRL memory size                               : 256 MB
    Strict CA checks                                  : NO
    Encryption trigger timeout                        : 100 ms
    Send Close-Notify                                 : YES
    Encryption trigger packet count                   : 45
    Deny SSL Renegotiation                            : NONSECURE
    Subject/Issuer Name Insertion Format              : Unicode
    OCSP cache size                                   : 10 MB
    Push flag                                         : 0x0 (Auto)
    Strict Host Header check for SNI enabled SSL sessions : NO
    Match HTTP Host header with SNI                   : CERT
    PUSH encryption trigger timeout                   : 1 ms
    Crypto Device Disable Limit                       : 0
    Global undef action for control policies          : CLIENTAUTH
    Global undef action for data policies             : NOOP
    Default profile                                   : ENABLED
    SSL Insert Space in Certificate Header            : YES
    Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
    Disable TLS 1.1/1.2 for dynamic and VPN services      : NO
    Software Crypto acceleration CPU Threshold        : 0
    Hybrid FIPS Mode                                  : DISABLED
    Signature and Hash Algorithms supported by TLS1.2     : ALL
    SSL Interception Error Learning and Caching       : DISABLED
    SSL Interception Maximum Error Cache Memory       : 0 Bytes
    NDCPP Compliance Certificate Check                : NO
    Heterogeneous SSL HW (Cavium and Intel Based)         : DISABLED
    Crypto Operation Queue Limit                      : 150%
 Done
<!--NeedCopy-->
```


GUI を使用してセキュアな再ネゴシエーションを有効にする

1. [トラフィック管理**] > [SSL] > [SSL** の詳細設定の変更] に移動します。
2. [**SSL** 再ネゴシエーションの拒否] を ALL 以外の値に設定します。



アダプティブ **SSL** トラフィック制御

注**:** この機能は、リリース 13.0 ビルド 58.x 以降でサポートされています。

アプライアンスで大量のトラフィックが受信され、暗号アクセラレーションキャパシティがいっぱいになると、アプライアンスは後で処理するために接続のキューイングを開始します。現在、このキューのサイズは 64 K に固定されており、この値を超えるとアプライアンスは接続のドロップを開始します。

リリース 13.0 ビルド 58.x から、ユーザーは実際のキャパシティに対するパーセンテージの値を設定できます。この機能強化により、キュー内の要素の数が、適応的かつ動的に計算された制限を超えると、アプライアンスは新しい接続をドロップします。このアプローチは、着信 SSL 接続を制御し、アプライアンスでの過剰なリソース消費やその他の障害（負荷分散モニタリングの障害やセキュアなアプリケーションへの応答の遅延など）を防止します。

キューが空の場合、アプライアンスは引き続き接続を受け入れることができます。キューが空でない場合、暗号システムは容量に達しており、アプライアンスは接続のキューイングを開始します。

制限は次に基づいて計算されます。

- アプライアンスの実際の容量。
- 実際のキャパシティに対するパーセンテージとしてユーザーが設定した値。デフォルト値は 150% に設定されています。

たとえば、アプライアンスの実際のキャパシティが 1 秒あたり 1000 オペレーションで、デフォルトのパーセンテージが設定されている場合、アプライアンスが接続を切断するまでの制限は 1500（1000 の 150%）です。

CLI を使用してオペレーションキュー制限を設定するには

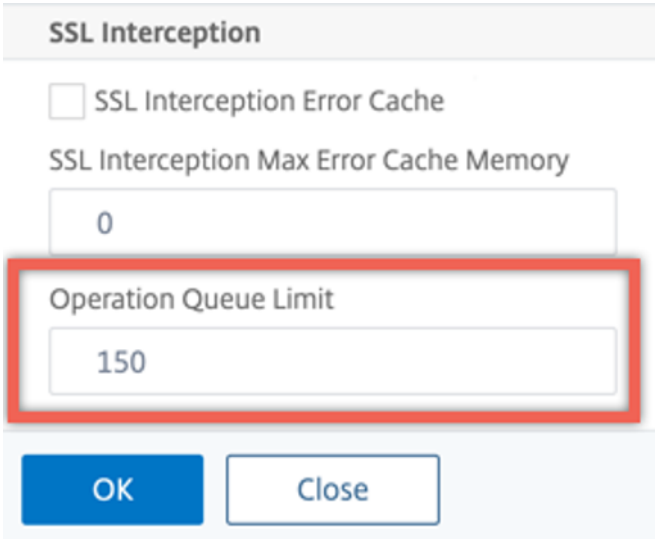
コマンドプロンプトで入力します。

```
set ssl parameter -operationQueueLimit <positive_integer>
```

Operation Queue Limit：暗号化操作キューの容量に対するパーセンテージの制限。この値を超えると、キューが削減されるまで新しい SSL 接続が受け入れられません。デフォルト値:150。最小値:0。最大値は 10000 です。

GUI を使用して操作キュー制限を設定するには

1. [トラフィック管理] **> [SSL**] に移動します。
2. [設定] で、[**SSL** の詳細設定の変更] をクリックします。
3. [操作キューの制限] に値を入力します。デフォルトは 150 です。
4. [**OK**] をクリックします。



異機種混在クラスタの展開

リリース 13.0 ビルド 47.x から、SSL パラメータ「異機種間 SSL HW」を「ENABLED」に設定することで、異なる数のパケットエンジンで Citrix ADC MPX アプライアンスの異機種混在クラスタ展開を形成できます。たとえば、Cavium チップベースのアプライアンス (MPX 14000 またはそれに類似) とインテル Coleto チップベースのアプライアンス (MPX 15000 またはそれに類似) のクラスタを形成するには、SSL パラメーター「異種 SSL HW」を有効にします。同じチップを使用してプラットフォームのクラスタを形成するには、このパラメータのデフォルト値（DISABLED）を維持します。

> 注:
>
> 異機種混在クラスタでは、次の機能はサポートされていません。
>
> - Citrix ADC SDX アプライアンスでホストされている VPX インスタンス。
> - 仮想サーバー、サービス、サービスグループ、内部サービスなどの SSL エンティティに対する SSLv3 プロトコル。
> - ソフトウェア暗号加速 CPU しきい値（ハードウェアとソフトウェアを使用して ECDSA および ECDHE 暗号のパフォーマンスを向上させます）。

異機種混在クラスタでサポートされるプラットフォームの詳細については、「https://docs.citrix.com/en-us/citrix-adc/13/clustering/support-for-heterogeneous-cluster.html」を参照してください。

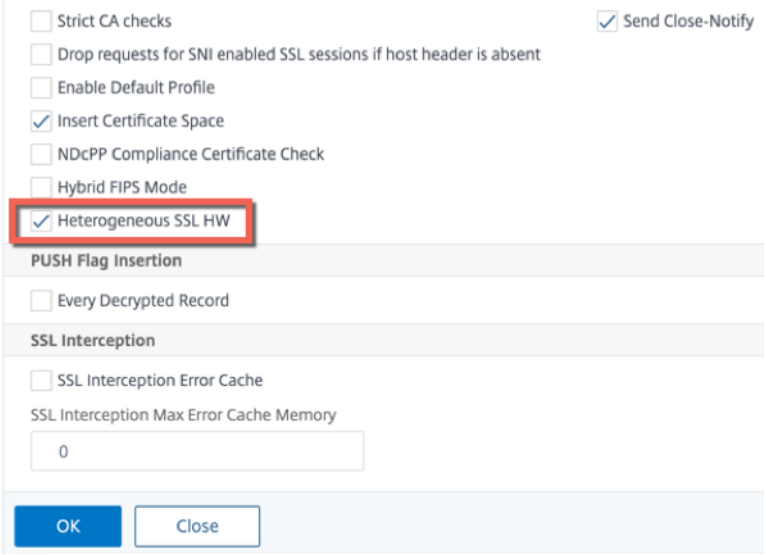
CLI を使用した異機種クラスターの有効化

コマンドプロンプトで入力します。

```
set ssl parameter -heterogeneousSSLHW ENABLED
```

GUI を使用した異機種混在クラスタの有効化

1. [トラフィック管理] **> [SSL**] に移動し、[設定] グループで [**SSL** の詳細設定の変更] を選択します。
2. [異機種混在 **SSL HW**] を選択します。[**OK**] をクリックします。



PUSH フラグベースの暗号化トリガーメカニズム

PSH TCP フラグに基づく暗号化トリガメカニズムにより、次のことが可能になりました。

- PSH フラグが設定された連続したパケットを 1 つの SSL レコードにマージするか、PSH フラグを無視します。

- `set ssl parameter -pushEncTriggerTimeout <positive_integer>` コマンドを使用してタイムアウト値をグローバルに設定する、タイマーベースの暗号化を実行します。

CLI を使用して **PUSH** フラグベースの暗号化を設定する

コマンドプロンプトで次のコマンドを入力して PUSH フラグベースの暗号化を構成し、構成を確認します。

```
set ssl vserver <vServerName> [-pushEncTrigger <pushEncTrigger>]

show ssl vserver
<!--NeedCopy-->
```

例:

```
set ssl vserver vserver1 -pushEncTrigger always

Done

sh ssl vserver vserver1

            Advanced SSL configuration for VServer vserver1:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: ENABLED

                Refresh Count: 0
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
            HSTS Max-Age: 0
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Push Encryption Trigger: Always
            Send Close-Notify: YES
```



```
25              Strict Sig-Digest Check: DISABLED
26              Zero RTT Early Data: DISABLED
27              DHE Key Exchange With PSK: NO
28              Tickets Per Authentication Context: 1
29              ECC Curve: P_256, P_384, P_224, P_521
30
31      1)      Cipher Name: DEFAULT
32              Description: Default cipher list with encryption strength
                    >= 128bit
33  Done
34  <!--NeedCopy-->
```

GUI を使用して **PUSH** フラグベースの暗号化を構成する

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションの [**PUSH** 暗号化トリガー] リストから値を選択します。

TLS1.2 署名ハッシュアルゴリズムのサポート

Citrix ADC アプライアンスは、TLS1.2 署名ハッシュ拡張に完全に準拠しています。

SSL ハンドシェイクでは、クライアントは、サポートされている署名ハッシュアルゴリズムのリストを送信します。クライアントは、「signature_algorithm」拡張を使用して、SSL ハンドシェイクメッセージ (SKE と CCV) でどのシグニチャハッシュアルゴリズムペアを使用できるかをサーバに指示します。この拡張モジュールの「拡張データ」フィールドには、クライアントの Hello メッセージに「サポートされた署名のアルゴリズム」値が含まれています。SSL ハンドシェイクは、サーバがこれらのシグニチャハッシュアルゴリズムのいずれかをサポートしている場合に処理されます。サーバーがこれらのアルゴリズムをサポートしていない場合、接続は切断されます。

同様に、サーバーがクライアント認証のためにクライアント証明書を要求した場合、証明書要求メッセージには「supported_signature_algorithms」の値が含まれます。クライアント証明書は、この署名ハッシュアルゴリズムに基づいて選択されます。

> 注:
>
> Citrix ADC アプライアンスは、クライアントに対してはサーバーとして、バックエンドサーバーに対してはクライアントとして機能します。

このアプライアンスは、フロントエンドでは RSA-SHA1 と RSA-SHA256 のみをサポートし、バックエンドでは RSA-MD5、RSA-SHA1、RSA-SHA256 のみをサポートします。

MPX/SDX/VPX アプライアンスは、次の署名ハッシュの組み合わせをサポートしています。SDX アプライアンスでは、SSL チップが VPX インスタンスに割り当てられている場合、MPX アプライアンスの暗号サポートが適用されます。それ以外の場合は、VPX インスタンスの通常の暗号サポートが適用されます。

- VPX インスタンスおよび N3 チップのない MPX/SDX アプライアンスでは、次のようになります。

  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
- N3 チップを搭載した MPX/SDX アプライアンスでは、次の手順を実行します。
  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
  - ECDSA-SHA1
  - ECDSA-SHA224
  - ECDSA-SHA256
  - ECDSA-SHA384
  - ECDSA-SHA512

デフォルトでは、すべてのシグニチャハッシュアルゴリズムが有効になっています。ただし、次のコマンドを使用して有効にできるシグニチャハッシュアルゴリズムはごくわずかです。

```
set ssl parameter -sigDigestType <sigDigestType>

Parameters

sigDigestType

Signature digest algorithms supported by the appliance. The platform
   determines the list of algorithms supported by default.

             On VPX: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384
                  RSA-

             SHA512

             On MPX with N3 cards: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-

             SHA256 RSA-SHA384 RSA-SHA512 ECDSA-SHA1 ECDSA-SHA224
                ECDSA-

             SHA256 ECDSA-SHA384 ECDSA-SHA512
```



```
18
19                Other MPX Platforms: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-
                      SHA256 RSA-SHA384 RSA-
20
21                SHA512.
22
23      set ssl parameter -sigDigestType RSA-SHA224 RSA-SHA256 RSA-SHA384
            RSA-SHA512
24  <!--NeedCopy-->
```

ピア証明書を検証する

RFC 5246 によると、ピア証明書は Client Hello 拡張に含まれる署名ハッシュアルゴリズムのいずれかを使用して署名される必要があります。strictSigDigestCheck パラメータを使用できます。クライアントから送信されたシグニチャハッシュリストに応じて、strictSigDigestCheckを有効にすると、アプライアンスは Client Hello 拡張機能に記載されているシグニチャハッシュアルゴリズムのいずれかによって署名された証明書を返します。ピアに適切な証明書がない場合、接続は切断されます。このパラメータを無効にすると、ピア証明書でシグニチャハッシュはチェックされません。

SSL 仮想サーバおよびサービスでは、厳密なシグニチャダイジェストチェックを設定できます。SSL 仮想サーバーでこのパラメーターを有効にする場合、サーバーから送信されるサーバー証明書は、Client Hello 拡張機能にリストされている署名ハッシュアルゴリズムのいずれかによって署名されている必要があります。クライアント認証が有効な場合、サーバーが受信したクライアント証明書は、サーバーから送信された証明書要求にリストされている署名ハッシュアルゴリズムのいずれかを使用して署名されている必要があります。

SSL サービスでこのパラメーターを有効にする場合、クライアントが受信するサーバー証明書は、Client Hello 拡張機能にリストされている署名ハッシュアルゴリズムのいずれかによって署名されている必要があります。クライアント証明書は、証明書要求メッセージに記載されている署名ハッシュアルゴリズムのいずれかを使用して署名されている必要があります。

デフォルトプロファイルが有効な場合は、このプロファイルを使用して、SSL 仮想サーバ、SSL サービス、および SSL プロファイルに対して厳密なシグニチャダイジェストチェックを設定できます。

CLI を使用して **SSL** 仮想サーバ、サービス、またはプロファイルに厳密なシグニチャダイジェストチェックを設定する

コマンドプロンプトで入力します。

```
1  set ssl vserver <vServerName> -strictSigDigestCheck ( ENABLED |
       DISABLED )
2
```



```
set ssl service <serviceName> -strictSigDigestCheck ( ENABLED |
    DISABLED )

set ssl profile <name>-strictSigDigestCheck ( ENABLED | DISABLED )

Parameters

strictSigDigestCheck

               Check whether peer entity certificate is signed using one
                    of the signature-hash algorithms supported by the
                   Citrix ADC appliance.

               Possible values: ENABLED, DISABLED

               Default: DISABLED
<!--NeedCopy-->
```

例:

```
set ssl vserver v1  -strictSigDigestCheck Enabled
set ssl service s1  -strictSigDigestCheck Enabled
set ssl profile p1  -strictSigDigestCheck Enabled
<!--NeedCopy-->
```

> 重要:
>
> DH、ECDHE、または ECDSA 暗号がアプライアンスに設定されている場合、SKE メッセージは、クライアントリストとアプライアンスに設定されたリストに共通するシグニチャハッシュの 1 つを使用して署名する必要があります。共通の署名ハッシュがない場合、接続は切断されます。

RFC 8446 で定義されている TLSv1.3 プロトコルサポート

January 25, 2022

Citrix ADC VPX および Citrix ADC MPX アプライアンスは、RFC 8446 で指定されている TLSv1.3 プロトコルをサポートするようになりました。

> メモ:
>
> - リリース 13.0 ビルド 71.x 以降では、TLS1.3 ハードウェアアクセラレーションが次のプラットフォームでサポートされています。

> - MPX 5900
> - MPX/SDX 8900
> - MPX/SDX 15000
> - MPX/SDX 15000-50G
> - MPX/SDX 26000
> - MPX/SDX 26000-50S
> - MPX/SDX 26000-100G
>
> - TLSv1.3 プロトコルのソフトウェアのみのサポートは、Citrix ADC FIPS アプライアンスを除く他のすべての Citrix ADC MPX および SDX アプライアンスで利用できます。
>
> - TLSv1.3 は、拡張プロファイルでのみサポートされます。拡張プロファイルを有効にするには、拡張プロファイルの有効化を参照してください。
> - TLS1.3 を使用するには、RFC 8446 仕様に準拠したクライアントを使用する必要があります。

サポートされている **Citrix ADC** 機能

次の SSL 機能がサポートされています。

1. TLSv1.3 暗号スイート:
   - TLS1.3-AES256-GCM-SHA384 (0x1302)
   - TLS1.3_CHACHA20_POLY1305_SHA256 (0x1303)
   - TLS1.3-AES128_GCM-SHA256 (0x1301)
2. エフェメラルな Diffie-Hellman キー交換のための ECC カーブ:
   - P_256
   - P_384
   - P_521
3. チケットベースのセッション再開が有効な場合のハンドシェイクの簡略化
4. 0-RTT アーリーアプリケーションデータ
5. クライアント証明書の OCSP および CRL 検証をサポートする、オプションまたは必須の証明書ベースのクライアント認証
6. サーバー名拡張:SNI を使用したサーバー証明書の選択
7. application_level_protocol_negotiation 拡張機能を使用した、アプリケーションプロトコルネゴシエーション (ALPN)。
8. OCSP ホチキス
9. TLSv1.3 ハンドシェイク用のログメッセージと AppFlow レコードが生成されます。
10. `nstrace` パケットキャプチャユーティリティによる TLS 1.3 トラフィックシークレットのロギング（オプション）。

11. RFC 8446 を実装する TLS クライアントとの相互運用性。たとえば、Mozilla Firefox、Google Chrome、OpenSSL などです。

サポートされているブラウザー

次のブラウザバージョンがサポートされ、TLS 1.3 プロトコルの Citrix ADC 実装と互換性があります。

- Google Chrome -バージョン 72.0.3626.121 (公式ビルド) (64 ビット)
- Mozilla Firefox-65.0.2 (64 ビット)
- Opera -バージョン:58.0.3135.79

構成

TLSv1.3 は SSL プロファイルではデフォルトで無効になっています。

CLI を使用して **SSL** プロファイルを追加する

コマンドプロンプトで入力します。

```
add ssl profile <tls13-profile-name>
<!--NeedCopy-->
```

例:

```
add ssl profile tls13profile

sh ssl profile tls13profile
1)  Name: tls13profile           (Front-End)
    SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1.1: ENABLED
       TLSv1.2: ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED                      Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA:
       ENABLED                             Refresh Count: 0
    Deny SSL Renegotiation                                ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
```



```
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:             1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions: NO
    Push flag:            0x0 (Auto)
    SSL quantum size:                            8 kB
    Encryption trigger timeout           100 mS
    Encryption trigger packet count:               45
    Subject/Issuer Name Insertion Format: Unicode

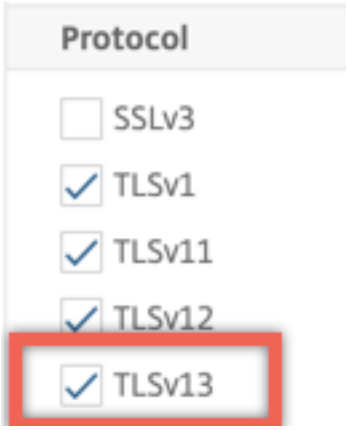
    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:  10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0

    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT Priority :1
    Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用して **SSL** プロファイルを追加する

1. [システム**] > [**プロファイル] に移動します。[**SSL** プロファイル] を選択します。
2. [追加] をクリックし、プロファイルの名前を指定します。
3. [プロトコル] で **TLSv13** を選択します。



4. [**OK**] をクリックします。

CLI を使用して **SSL** プロファイルを **SSL** 仮想サーバにバインドする

コマンドプロンプトで入力します。

```
set ssl vserver <vServerName> -sslProfile <tls13-profile-name>
<!--NeedCopy-->
```

例:

```
set ssl vserver ssl-vs -sslProfile tls13profile
```

GUI を使用して **SSL** プロファイルを **SSL** 仮想サーバにバインドする

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを選択します。
2. [詳細設定] で [**SSL** プロファイル] をクリックします。
3. 以前に作成した TLSv1.3 プロファイルを選択します。
4. [**OK**] をクリックします。
5. [完了] をクリックします。

TLSv1.3 プロトコルの **SSL** プロファイルパラメータ

1. SSL プロファイルで TLS1.3 パラメータを有効または無効にします。

   tls13: SSL プロファイルに対する TLSv1.3 プロトコルサポートの状態。

   設定可能な値: ENABLED, DISABLED

   デフォルト値: 無効


```
set ssl profile tls13profile -tls13 enable
<!--NeedCopy-->
```

```
set ssl profile tls13profile -tls13 disable
<!--NeedCopy-->
```

2. 発行されるセッションチケットの数を設定します。

   TLS13SessionTicketsPerAuthContext: TLS1.3 がネゴシエートされ、チケットベースの再開が有効になっていて、(1) ハンドシェイクが完了した、または (2) ハンドシェイク後にクライアント認証が完了したときに、SSL 仮想サーバーが発行するチケットの数。

   この値を増やすと、クライアントは接続ごとに新しいチケットを使用して複数のパラレル接続を開くことができます。

   再開が無効になっている場合、チケットは送信されません。

   デフォルト値:1

   最小値:1

   最大値:10

```
set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 1

set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 10
<!--NeedCopy-->
```

3. DH キー交換の設定

   dheKeyExchangeWithPsk: TLS 1.3 セッション再開ハンドシェイク中に事前共有キーが受け入れられたときに、SSL 仮想サーバで DHE キー交換が必要かどうかを指定します。DHE キー交換は、チケットキーが侵害された場合でも、**DHE** キー交換の実行に必要な追加リソースを犠牲にして、前方秘匿性を確保します。

   セッションチケットが有効な場合、使用可能な設定は次のように機能します。

   はい:DHE キー交換は、クライアントがキー交換をサポートしているかどうかにかかわらず、事前共有キーが受け入れられる場合に必要です。事前共有キーを提供するときにクライアントが DHE キー交換をサポートしていない場合、ハンドシェイクは致命的なアラートで中止されます。

   NO: DHE キー交換は、事前共有キーが受け入れられたときに、クライアントから要求された場合にのみ実行されます。

   可能な値: はい、いいえ

デフォルト値:NO

```
set ssl profile tls13profile dheKeyExchangeWithPsk yes

set ssl profile tls13profile dheKeyExchangeWithPsk no
<!--NeedCopy-->
```

4. 0-RTT アーリーデータ受け入れの有効化または無効化

   zeroRttEarlyData: TLS 1.3 の初期アプリケーションデータの状態。適用可能な設定は次のように機能します。

   ENABLED: ハンドシェイクが完了する前に、初期のアプリケーションデータが処理される可能性があります。
   DISABLED: 初期のアプリケーションデータは無視されます。

   設定可能な値: ENABLED, DISABLED

   デフォルト値: 無効

```
set ssl profile tls13profile -zeroRttEarlyData ENABLED

set ssl profile tls13profile -zeroRttEarlyData DISABLED
<!--NeedCopy-->
```

既定の暗号グループ

デフォルトの暗号グループには TLS1.3 暗号が含まれています。

```
sh cipher DEFAULT
1) Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035

2)  Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
...
...
27) Cipher Name: TLS1.3-AES256-GCM-SHA384         Priority : 27
    Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(256) Mac=AEAD
         HexCode=0x1302
```

```
11
12 28) Cipher Name: TLS1.3_CHACHA20_POLY1305_SHA256     Priority : 28
13     Description: TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256)
           Mac=AEAD   HexCode=0x1303
14
15 29) Cipher Name: TLS1.3-AES128_GCM-SHA256        Priority : 29
16     Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(128) Mac=AEAD
           HexCode=0x1301
17 Done
18 <!--NeedCopy-->
```

制限事項

- TLSv1.3 はバックエンドではサポートされていません。
- TLSv1.3 は、Citrix Secure Web Gateway アプライアンスおよび Citrix ADC FIPS アプライアンスではサポートされていません。

セキュリティ制限

TLSv1.3 サーバオペレータは、RFC 8446 で概説されている下位互換性のために、次のセキュリティ制限に留意する必要があります。NetScaler アプライアンスのデフォルト構成は、これらの制限に準拠しています。ただし、NetScaler アプライアンスでは、これらのルールの遵守は強制されません。

- RC4 暗号スイートのセキュリティは、RFC7465 で説明されているように不十分であると考えられています。実装では、TLS のどのバージョンでも RC4 暗号スイートを提供したり、ネゴシエートしたりしてはなりません。
- 古いバージョンの TLS では、低強度の暗号を使用できました。強度が 112 ビット未満の暗号は、どのバージョンの TLS でも提供またはネゴシエートしてはなりません。
- SSL 3.0 [SSLv3] のセキュリティは、RFC7568 で説明されているように不十分であるとみなされ、ネゴシエートしてはなりません。TLSv1.3 が有効になっている場合は、SSLv3 を無効にします（SSLv3 はデフォルトで無効になっています）。
- SSL 2.0 [SSLv2] のセキュリティは、RFC6176 で説明されているように不十分と見なされ、ネゴシエートしてはなりません。TLS 1.3 が有効な場合は、SSLv2 を無効にします (SSLv2 はデフォルトで無効になっています)。

> 注:
>
> TLS1.3 で実行されるプロトコルのトラブルシューティングについては、パケットトレースからの TLS1.3 トラフィックの復号化を参照してください。

ハウツー記事

October 7, 2021

ハウツー記事は、一般的な展開の構成手順を含むシンプルで使いやすい記事です。リンクをクリックして記事を表示します。

Citrix ADC アプライアンスで証明書署名リクエストを作成して SSL 証明書を使用します

クライアントトラフィックを転送するように SSL アクションを設定します

ADC で暗号がサポートされていない場合にクライアントトラフィックを転送するように SSL アクションを設定します

ディレクトリごとのクライアント認証の構成

Outlook Web Access に対するサポートの構成

SSL ベースのヘッダー挿入の設定

エンドツーエンド暗号化による SSL オフロードの構成

トランスペアレント SSL アクセラレーションの設定

フロントエンドで HTTP を使用し、バックエンドで SSL を使用して SSL アクセラレーションを構成する

他の TCP プロトコルによる SSL オフロードの構成

SSL ブリッジングの設定

バックエンドサービスでクライアント認証が有効になっている場合の SSL モニタリングの設定

セキュリティで保護されたコンテンツスイッチングサーバーを構成する

HTTP トラフィックを受け入れるように HTTPS 仮想サーバーを構成する

SSL セッションのグレースフルクリーンアップの構成

HTTP 厳密なトランスポートセキュリティ (HSTS) のサポートを構成します

SSLv2 リダイレクトの構成

高可用性セットアップでのファイルの同期の構成

NSIP で TLS1.0 および TLS1.1 を無効にする

Citrix ADC アプライアンスで使用される証明書を PFX ファイルとしてエクスポートする

SSL 証明書

October 7, 2021

SSL トランザクションの一部である SSL 証明書は、会社（ドメイン）または個人を識別するデジタルデータフォーム（X509）です。この証明書には、サーバーとの安全なトランザクションを開始しようとするすべてのクライアントが確認できる公開キーコンポーネントが含まれます。対応する秘密キーは、Citrix ADC アプライアンスに安全に配置され、非対称キー（または公開キー）の暗号化と復号化に使用されます。

SSL 証明書およびキーは、次のいずれかの方法で入手できます。

- 認証された認証機関 (CA) (Verisign など) から
- Citrix ADC アプライアンスで新しい SSL 証明書とキーを生成する

または、アプライアンスで既存の SSL 証明書を使用することもできます。

証明書は、Citrix ADC アプライアンスによって 4 つのタイプに分類されます。

- サーバー証明書：サーバー証明書は、クライアントに対してサーバーの ID を認証します。フロントエンドでは、ADC アプライアンスはサーバーとして機能します。サーバー証明書と秘密鍵を ADC アプライアンスの SSL 仮想サーバーにバインドします。
- クライアント証明書：クライアント証明書は、サーバーに対するクライアントの ID を認証します。バックエンドでは、ADC アプライアンスはクライアントとして機能します。クライアント証明書と秘密鍵を ADC アプライアンスの SSL サービスまたはサービスグループにバインドします。
- **CA** 証明書：CA 証明書は、エンドユーザー証明書（クライアント証明書とサーバー証明書）を発行します。CA 証明書は、信頼されたルート CA（認証局によって自己署名された）または中間 CA（信頼されたルート CA によって署名された）にすることができます。通常、CA 証明書には秘密鍵は必要ありません。
- 不明な証明書：他のすべての証明書はこのカテゴリに分類されます。

重要：すべての SSL トランザクションには、Verisign などの認証された CA から取得した証明書を使用することをお勧めします。Citrix ADC アプライアンスで生成された証明書は、テスト目的でのみ使用し、ライブ展開では使用しないでください。

- 証明書とキーのペアを追加するときに、既存の証明書ファイルと同じ名前の証明書ファイルを追加すると、元の証明書ファイルが警告なしに上書きされます。元の証明書ファイルが `/nsconfig/ssl`ディレクトリで使用できなくなったため、アプライアンスの再起動後に問題が発生する可能性があります。
- クラスター環境で証明書またはキーファイルを削除すると、ADC アプライアンスでのそれ以上の構成が制限されます。同じ場所にファイルを追加して戻し、構成を変更します。

注： SSL 証明書の管理を容易にするために ADM SSL ダッシュボードを使用し、未使用または間もなく期限切れになる証明書の通知を設定できます。詳細については、「SSL 証明書管理」を参照してください。

証明書の作成

April 7, 2022

認証局 (CA) は、公開キー暗号化で使用するデジタル証明書を発行するエンティティです。SSL トランザクションを実行するアプリケーション（Web ブラウザなど）は、認証局が発行または署名した証明書を信頼します。これらのア

プリケーションでは、信頼する CA のリストが保持されます。信頼できる CA のいずれかが、セキュアなトランザクションに使用されている証明書に署名すると、アプリケーションはトランザクションを続行します。

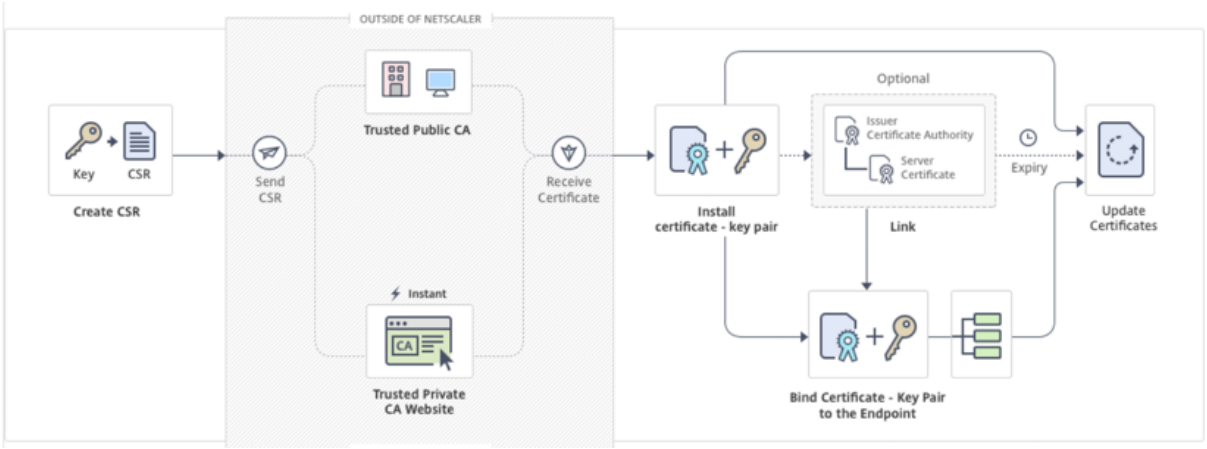
注意：すべての SSL トランザクションには、Verisign などの承認された CA から取得した証明書を使用することをお勧めします。Citrix ADC アプライアンスで生成された証明書は、テスト目的でのみ使用し、ライブ展開では使用しないでください。

既存の証明書とキーをインポートするには 、証明書のインポートを参照してください。

証明書を作成し、SSL 仮想サーバーにバインドするには、次の手順を実行します。ファイル名に使用できる特殊文字は、アンダースコアとドットだけです。

- プライベートキーを作成します。
- 証明書署名要求 (CSR) を作成します。
- CSR を認証局に送信します。
- 証明書とキーのペアを作成します。
- 証明書とキーのペアを SSL 仮想サーバーにバインドする

次の図は、このワークフローを示しています。



「新しい証明書を作成してインストールするにはどうすればよいですか」へのビデオリンク。

秘密キーの作成

> メモ：
>
> - リリース 12.1 ビルド 49.x から、PEM キー形式の AES256 アルゴリズムを使用して、アプライアンスの秘密キーを暗号化できます。256 ビットキーの AES は、データ暗号化規格 (DES) の 56 ビットキーに比べて、数学的に効率的で安全です。
> - リリース 12.1 ビルド 50.x から、PKCS #8 形式で RSA キーを作成できます。

秘密キーはデジタル証明書で最も重要な部分です。定義上、このキーは誰とも共有されるべきではなく、Citrix ADC アプライアンス上で安全に保管する必要があります。公開鍵で暗号化されたデータは、秘密鍵を使用することによってのみ復号化できます。

CA から受け取った証明書は、CSR の作成に使用された秘密キーでのみ有効です。このキーは、証明書を Citrix ADC アプライアンスに追加するために必要です。

アプライアンスは、秘密キーの作成に RSA 暗号化アルゴリズムのみをサポートします。どちらのタイプの秘密キーも認証局 (CA) に送信できます。CA から受け取った証明書は、CSR の作成に使用された秘密キーでのみ有効です。このキーは、証明書を Citrix ADC アプライアンスに追加するために必要です。

重要:

> - 秘密鍵へのアクセスは必ず制限してください。プライベートキーにアクセスできる人なら誰でも、SSL データを復号できます。
> - 許可される SSL キー名の長さには、パスがキー名に含まれている場合、絶対パス名の長さも含まれます。

SSL 証明書とキーはすべて、アプライアンス上の/nsconfig/sslフォルダに保存されます。セキュリティを強化するために、DES アルゴリズムまたはトリプル DES（3DES）アルゴリズムを使用して、アプライアンスに保存されている秘密キーを暗号化できます。

CLI を使用して **RSA** 秘密キーを作成する

コマンドプロンプトで入力します。

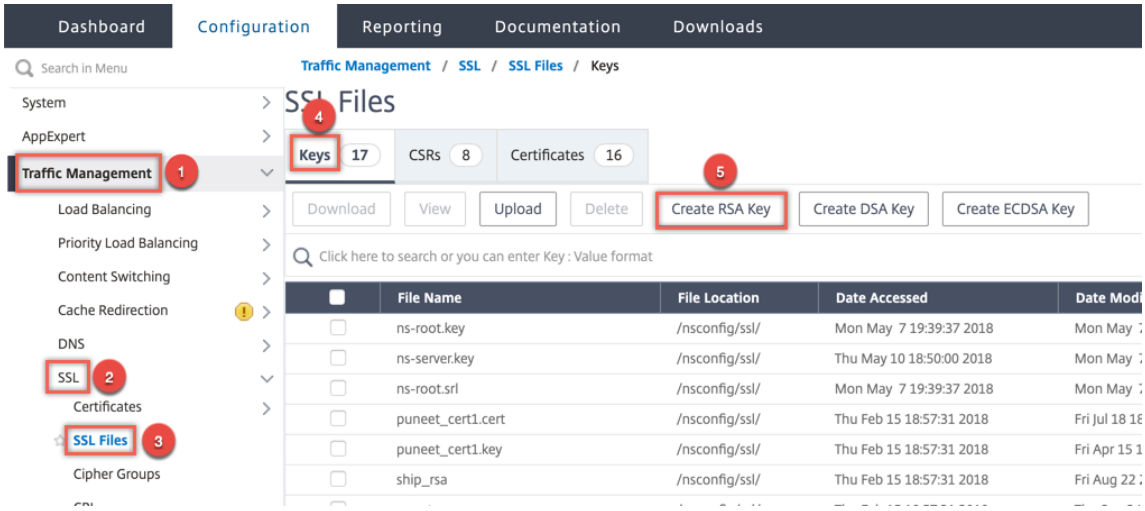
```
create ssl rsakey <keyFile> <bits> [-exponent ( 3 | F4 )] [-keyform (
    DER | PEM )] [-des | -des3 | -aes256] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

例:

```
create rsakey testkey 2048 -aes256 -password 123456 -pkcs8
<!--NeedCopy-->
```

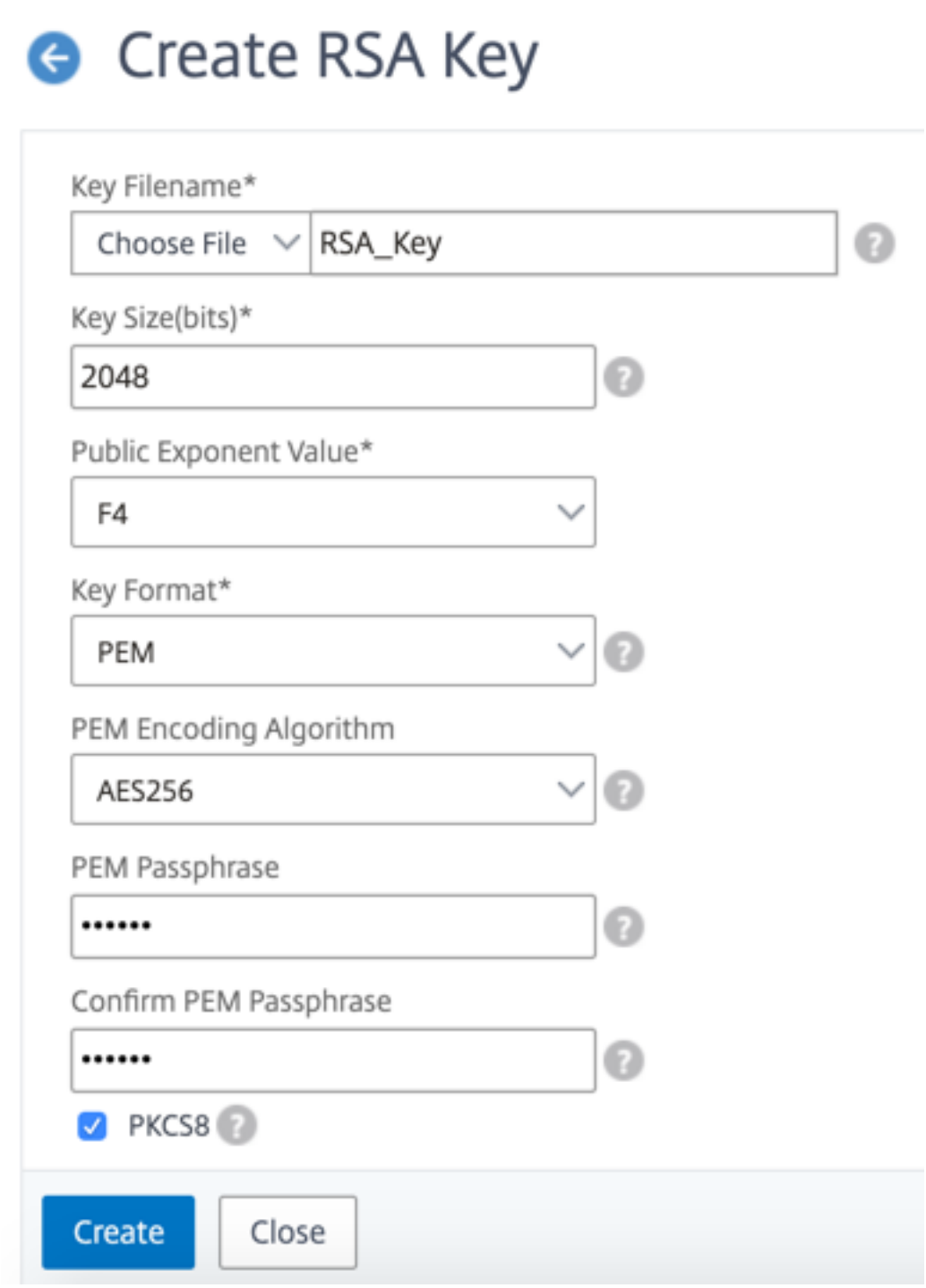
GUI を使用して **RSA** 秘密キーを作成する

1. [トラフィック管理] **> [SSL] > [SSL** ファイル] に移動します。
2. [キー] タブで、[**RSA** キーの作成] を選択します。



3. 次のパラメータに値を入力し、[**Create**] をクリックします。

   - **Key Filename** : RSA キーファイルの名前と、オプションで RSA キーファイルのパス。/nsconfig/ssl/ がデフォルトパスです。
   - キーサイズ: RSA キーのサイズ（ビット単位）。範囲は 512 ビットから 4096 ビットです。
   - 公開指数値 -RSA キーの公開指数。指数は暗号アルゴリズムの一部であり、RSA キーの作成に必要です。
   - **Key Format** : RSA キー・ファイルがアプライアンスに保存される形式。
   - **PEM** エンコードアルゴリズム -AES 256、DES、またはトリプル DES (DES3) アルゴリズムを使用して、生成された RSA キーを暗号化します。デフォルトでは、秘密鍵は暗号化されません。
   - **PEM Passphrase** : 秘密キーが暗号化されている場合は、キーのパスフレーズを入力します。



GUI を使用して **RSA** キーで **AES256** エンコードアルゴリズムを選択する

1. トラフィック管理 **> SSL > SSL** ファイル **> RSA** キーの作成に移動します。
2. [キー形式] で [**PEM**] を選択します。

3. [**PEM** エンコードアルゴリズム] で [**AES256**] を選択します。
4. **PKCS8** を選択します。

CLI を使用して証明書署名要求を作成する

コマンドプロンプトで入力します。

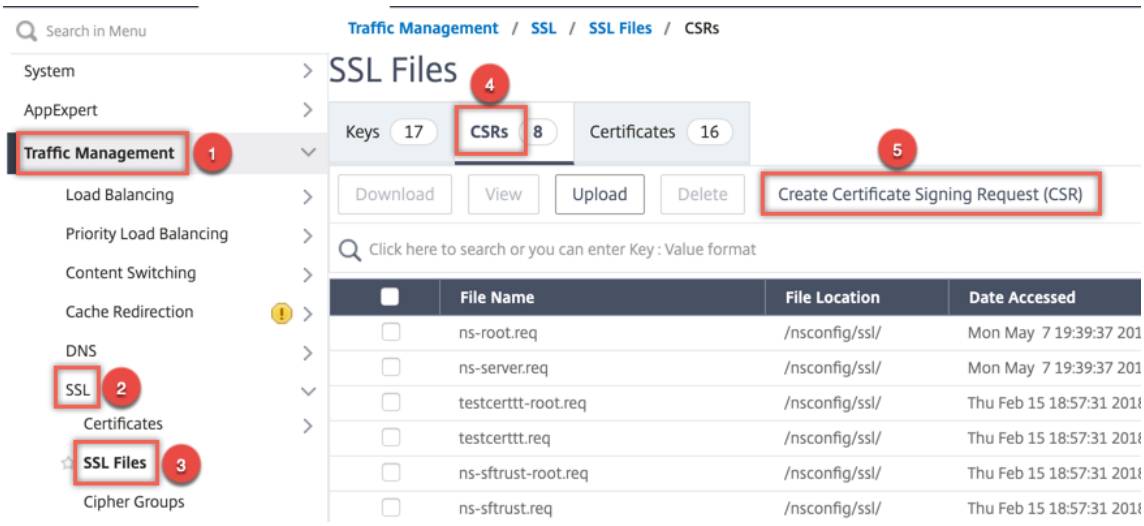
```
create ssl certreq <reqFile> -keyFile <input_filename> | -fipsKeyName <
   string>) [-keyForm (DER | PEM) {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string>
     -organizationUnitName <string> -localityName <string> -commonName
    <string> -emailAddress <string> {
 -challengePassword  }
  -companyName <string> -digestMethod ( SHA1 | SHA256 )
<!--NeedCopy-->
```

例**:**

```
create ssl certreq priv_csr_sha256 -keyfile priv_2048_2 -keyform PEM -
   countryName IN -stateName Karnataka -localityName Bangalore -
   organizationName Citrix -organizationUnitName NS -digestMethod
   SHA256
<!--NeedCopy-->
```

GUI を使用して証明書署名要求を作成する

1. [トラフィック管理**] > [SSL**] に移動します。
2. [**SSL** 証明書] で、[証明書署名要求 **(CSR)** の作成] をクリックします。



3. 「ダイジェスト方式」で「**SHA256**」を選択します。

詳細については、CSR の作成を参照してください。

証明書署名要求でのサブジェクト代替名のサポート

証明書のサブジェクト代替名 (SAN) フィールドを使用すると、ドメイン名や IP アドレスなどの複数の値を 1 つの証明書に関連付けることができます。つまり、www.example.com、www.example1.com、www.example2.com などの複数のドメインを 1 つの証明書で保護できます。

Google Chrome などの一部のブラウザでは、証明書署名要求 (CSR) で共通名がサポートされなくなりました。公的に信頼されたすべての証明書に SAN を適用します。

Citrix ADC アプライアンスは、CSR の作成時に SAN 値を追加することをサポートしています。SAN エントリを含む CSR を認証局に送信して、その SAN エントリを含む署名付き証明書を取得できます。アプライアンスは要求を受信すると、サーバ証明書の SAN エントリで一致するドメイン名があるかどうかを確認します。一致が見つかると、証明書がクライアントに送信され、SSL ハンドシェイクが完了します。CLI または GUI を使用して、SAN 値を持つ CSR を作成できます。

> 注: Citrix ADC アプライアンスは、DNS ベースの SAN 値のみを処理します。

CLI を使用してサブジェクトの別名で **CSR** を作成する

```
1 create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
     <string>) [-subjectAltName <string>] [-keyform ( DER | PEM )  {
2  -PEMPassPhrase  }
3  ] -countryName <string> -stateName <string> -organizationName <string>
      [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
4  -challengePassword  }
```



```
5    [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
6  <!--NeedCopy-->
```

パラメーター:

subjectAltName: サブジェクト代替名 (SAN) は X.509 の拡張であり、subjectAltName フィールドを使用してさまざまな値をセキュリティ証明書に関連付けることができます。これらの値は「サブジェクトの別名」(SAN) と呼ばれます。名前には以下が含まれます。

1. IP アドレス（「IP:」が付いたプレフィックス例:IP: 198.51.10.5 IP: 192.0.2.100）
2. DNS 名 (プレフィックスに「DNS:」が付く例:DNS: www.example.com dns: www.example.org dns: www.example.net)

コマンドラインで、引用符で囲んで値を入力します。2 つの値はスペースで区切ります。GUI では引用符は必要ありません。
最大長: 127

例**:**

```
1  create certReq test1.csr -keyFile test1.ky -countryName IN -stateName
       Kar -organizationName citrix -commonName ctx.com -subjectAltName ”
       DNS:*.example.com DNS:www.example.org DNS:www.example.net”
2  <!--NeedCopy-->
```

> 注:
>
> FIPS アプライアンスで、アプライアンスで FIPS キーを直接作成する場合は、キーファイル名を FIPS キー名に置き換える必要があります。
>
> ```
> 1  create certReq <csrname> -fipsKeyName fipskey.ky -countryName IN -
>        stateName Kar -organizationName citrix -commonName ctx.com -
>        subjectAltName ”DNS:www.example.com DNS:www.example.org DNS:www.
>        example.net”
> 2  <!--NeedCopy-->
> ```

GUI を使用して **CSR** を作成する

1. トラフィック管理 > **SSL** > **SSL** ファイルに移動します。
2. [**CSR**] タブで、[証明書署名要求 **(CSR)** の作成] をクリックします。
3. 値を入力し、[**Create**] をクリックします。

制限事項

SSL 証明書の作成時に SAN を使用するには、SAN 値を明示的に指定する必要があります。値は CSR ファイルから自動的には読み込まれません。

CSR を認証局に提出する

ほとんどの認証局 (CA) は、電子メールによる証明書の提出を受け付けています。CA は、CSR の送信元の電子メールアドレスに有効な証明書を返します。

CSR は/nsconfig/sslフォルダに保存されます。

テスト証明書の生成

> 注:
>
> サーバーテスト証明書を生成するには、サーバーテスト証明書の生成を参照してください。

Citrix ADC アプライアンスには、テスト用に自己署名証明書を作成するために使用できる CA ツールスイートが組み込まれています。

注意: Citrix ADC アプライアンスは実際の CA ではなくこれらの証明書に署名するため、実稼働環境では使用しないでください。実稼働環境で自己署名証明書を使用しようとすると、仮想サーバーにアクセスするたびに「証明書が無効です」という警告がユーザーに表示されます。

アプライアンスでは、次のタイプの証明書の作成がサポートされています。

- ルート CA 証明書
- 中間 CA 証明書
- エンドユーザー証明書
  - サーバー証明書
  - クライアント証明書

証明書を生成する前に、秘密キーを作成し、その秘密キーを使用してアプライアンスに証明書署名要求 (CSR) を作成します。次に、CSR を CA に送信する代わりに、Citrix ADC CA Tools を使用して証明書を生成します。

ウィザードを使用して証明書を作成する

1. [トラフィック管理] **> [SSL**] に移動します。
2. 詳細ウィンドウの [はじめに] で、作成する証明書の種類に応じたウィザードを選択します。
3. 画面の指示に従って操作します。

CLI を使用してルート CA 証明書を作成する

コマンドプロンプトで、次のコマンドを入力します。


```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
<!--NeedCopy-->
```

次の例では、csreq1 が CSR で、rsa1 が以前に作成された秘密キーです。

例:

```
create ssl cert cert1 csreq1 ROOT_CERT -keyFile rsa1 -keyForm PEM -days
     365

  Done
<!--NeedCopy-->
```

CLI を使用して中間 **CA** 証明書を作成する

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
     [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
     DER | PEM )] [-CAkey <input_filename>] [-CAkeyForm ( DER | PEM )]
    [-CAserial <output_filename>]
<!--NeedCopy-->
```

次の例では、csr1 は以前に作成された CSR です。cert1 と rsakey1 は、自己署名 (ルート CA) 証明書の証明書および対応するキーであり、pvtkey1 は中間 CA 証明書の秘密キーです。

例:

```
create ssl cert certsy csr1 INTM_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23
Done

create ssl rsakey pvtkey1 2048 -exponent F4 -keyform PEM
Done
<!--NeedCopy-->
```

GUI を使用してルート **CA** 証明書を作成する

[トラフィック管理**] > [SSL**] に移動し、[はじめに] グループで [ルート **CA** 証明書ウィザード] を選択して、ルート

CA 証明書を構成します。

GUI を使用して中間 **CA** 証明書を作成する

[トラフィック管理**] > [SSL**] に移動し、[はじめに] グループで [中間 **CA** 証明書ウィザード] を選択して、中間 CA 証明書を設定します。

エンドユーザー証明書を作成する

エンドユーザー証明書は、クライアント証明書でもサーバー証明書でもかまいません。テストエンドユーザ証明書を作成するには、中間 CA 証明書または自己署名ルート CA 証明書を指定します。

> **注:** 本番環境で使用するエンドユーザー証明書を作成するには、信頼できる CA 証明書を指定し、その CSR を認証局 (CA) に送信します。

コマンドラインインターフェイスを使用してテスト用エンドユーザー証明書を作成する

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days<positive_integer>]
    [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
    DER | PEM )] [-CAkey<input_filename>] [-CAkeyForm ( DER | PEM )] [-
    CAserial <output_filename>]
<!--NeedCopy-->
```

中間証明書がない場合は、CAcertおよびCAkeyのルート CA 証明書の証明書（cert1）と秘密キー（rsakey1）の値を使用します。

例:

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23

Done
<!--NeedCopy-->
```

中間証明書がある場合は、CAcertとCAkeyの中間証明書の certificate (certsy) と秘密キー (pvtkey1) の値を使用します。

例:


```
create ssl cert cert12 csr1 SRVR_CERT -CAcert certsy -CAkey pvtkey1 -
    CAserial 23

Done
<!--NeedCopy-->
```

OpenSSL を使用して自己署名 SAN 証明書を作成する

複数のサブジェクト代替名を持つ自己署名 SAN 証明書を作成するには、次の手順を実行します。

1. 会社の要件に従って関連するフィールドを編集して、ローカルコンピューターに OpenSSL 設定ファイルを作成します。

   注**:** 次の例では、設定ファイルは「**req.conf**」です。

   ```
   [req]
   distinguished_name = req_distinguished_name
   x509_extensions = v3_req
   prompt = no
   [req_distinguished_name]
   C = US
   ST = VA
   L = SomeCity
   O = MyCompany
   OU = MyDivision
   CN = www.company.com
   [v3_req]
   keyUsage = keyEncipherment, dataEncipherment
   extendedKeyUsage = serverAuth
   subjectAltName = @alt_names
   [alt_names]
   DNS.1 = www.company.net
   DNS.2 = company.com
   DNS.3 = company.net
   <!--NeedCopy-->
   ```

2. Citrix ADC アプライアンスの /nsconfig/ssl ディレクトリにファイルをアップロードします。
3. Citrix ADC CLI にユーザーnsrootとしてログオンし、シェルプロンプトに切り替えます。
4. 次のコマンドを実行して証明書を作成します。


```
cd /nsconfig/ssl
openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout cert.
   pem -out cert.pem -config req.conf -extensions 'v3_req'
<!--NeedCopy-->
```

5. 次のコマンドを実行して証明書を検証します。

```
openssl x509 -in cert.pem -noout  - text
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
ed:90:c5:f0:61:78:25:ab
Signature Algorithm: md5WithRSAEncryption
Issuer: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
   www.company.com
Validity
Not Before: Nov 6 22:21:38 2012 GMT
Not After : Nov 6 22:21:38 2014 GMT
Subject: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
   www.company.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Modulus (2048 bit):
…
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Key Usage:
Key Encipherment, Data Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 Subject Alternative Name:
DNS:www.company.net, DNS:company.com, DNS:company.net
Signature Algorithm: md5WithRSAEncryption …
<!--NeedCopy-->
```

証明書のインストール、リンク、および更新

April 7, 2022

証明書をインストールするには、「証明書とキーのペアの追加または更新」を参照してください。

リンク証明書

多くのサーバー証明書は、複数の階層的な認証局 (CA) によって署名されています。つまり、証明書は次のようなチェーンを形成します。

```
Root CA
|
+------------ Intermediate CA
|
+------------------------ End-Entity (Server) Certificate
```

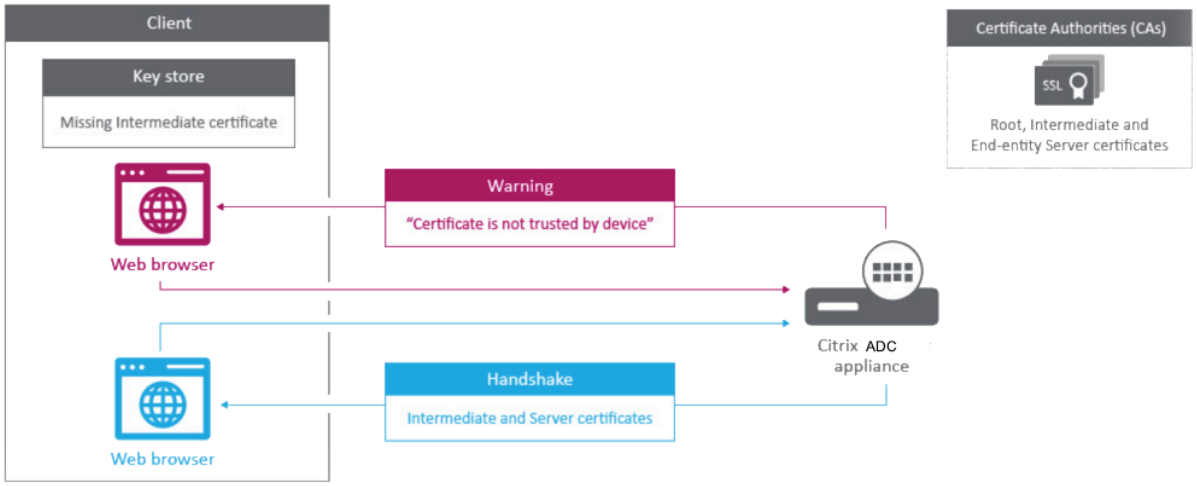
中間 CA がプライマリとセカンダリの中間 CA 証明書に分割されることがあります。証明書は次のようなチェーンを形成します。

```
Root CA
|
+------- Primary Intermediate CA
|+--------------- Secondary Intermediate CA
|
+------------------------ End-Entity (Server) Certificate
```

通常、クライアントマシンのローカル証明書ストアにはルート CA 証明書が含まれていますが、1 つ以上の中間 CA 証明書は含まれていません。ADC アプライアンスは、1 つ以上の中間 CA 証明書をクライアントに送信する必要があります。

注**:** アプライアンスはルート CA 証明書をクライアントに送信してはなりません。公開キー基盤 (PKI) の信頼関係モデルでは、ルート CA 証明書をアウトオブバンド方式でクライアントにインストールする必要があります。たとえば、証明書はオペレーティングシステムや Web ブラウザに含まれています。クライアントは、アプライアンスから送信されたルート CA 証明書を無視します。

標準的な Web ブラウザが信頼できる CA として認識しない中間 CA が、サーバ証明書を発行することがあります。この場合、1 つ以上の CA 証明書をサーバ独自の証明書とともにクライアントに送信する必要があります。そうしないと、サーバ証明書の認証に失敗するため、ブラウザは SSL セッションを終了します。



中間機関証明書をリンクするにはどうしたらいいですか? へのビデオリンク。

サーバ証明書と中間証明書を追加するには、次のセクションを参照してください。

- 手動による証明書リンク
- 証明書リンクの自動化
- 証明書のチェーンを作成する

手動による証明書リンク

注：この機能は、Citrix ADC FIPS プラットフォームおよびクラスターセットアップではサポートされていません。

個々の証明書を追加してリンクする代わりに、サーバ証明書と最大 9 個の中間証明書を 1 つのファイルにグループ化できるようになりました。証明書とキーのペアを追加するときに、ファイル名を指定できます。その前に、次の前提条件が満たされていることを確認してください。

- ファイル内の証明書は次の順序になっています。
  - サーバー証明書 (ファイル内の最初の証明書であることが必要)
  - オプションで、サーバーキー
  - 中間証明書 1 (ic1)
  - 中間証明書 2 (ic2)
  - 中間証明書 3 (ic3) など
    注: 中間証明書ファイルは、中間証明書ごとに「<certificatebundlename>.pem_ic<*n*>」という名前で作成されます。n は 1 ～9 です。たとえば、bundle.pem_ic1 と指定します。**bundle** は証明書セットの名前で、ic1 は証明書セット内の最初の中間証明書です。
- [バンドル] オプションが選択されている。
- このファイルには中間証明書が 9 つまで存在しません。

ファイルが解析され、サーバ証明書、中間証明書、およびサーバキー (存在する場合) が識別されます。まず、サーバ証明書とキーが追加されます。次に、中間証明書がファイルに追加された順序で追加され、それに応じてリンクされます。

次の条件のいずれかに当てはまる場合、エラーが報告されます。

- アプライアンスには、中間証明書の 1 つの証明書ファイルがあります。
- この鍵は、ファイル内のサーバ証明書の前に置かれます。
- 中間証明書はサーバ証明書の前に置かれます。
- 中間証明書は、作成時と同じ順序でファイル内に配置されません。
- このファイルには証明書がありません。
- 証明書が適切な PEM 形式ではありません。
- ファイル内の中間証明書の数が 9 個を超えています。

CLI を使用して証明書セットを追加する

コマンドプロンプトで次のコマンドを入力して証明書セットを作成し、構成を確認します。


```
add ssl certKey <certkeyName> -cert <string> -key <string> -bundle (YES
    | NO)

show ssl

show ssl certlink
<!--NeedCopy-->
```

次の例では、証明書セット (bundle.pem) に次のファイルが含まれています。

bundle_ic1 にリンクされたサーバー証明書 (バンドル)

bundle_ic2 にリンクされた最初の中間証明書 (bundle_ic1)

bundle_ic3 にリンクされた 2 つ目の中間証明書 (bundle_ic2)

3 番目の中間証明書 (bundle_ic3)

```
add ssl certKey bundletest -cert bundle9.pem -key bundle9.pem -bundle
   yes

sh ssl certkey

1)      Name: ns-server-certificate
        Cert Path: ns-server.cert
        Key Path: ns-server.key
        Format: PEM
        Status: Valid,   Days to expiration:5733
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
           Internal,CN=default OULLFT
        Validity
                Not Before: Apr 21 15:56:16 2016 GMT
                Not After : Mar  3 06:30:56 2032 GMT
        Subject:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
           Internal,CN=default OULLFT
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

2)      Name: servercert
```



```
        Cert Path: complete/server/server_rsa_1024.pem
        Key Path: complete/server/server_rsa_1024.ky
        Format: PEM
        Status: Valid,   Days to expiration:7150
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 1F
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:  C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix
        Validity
                Not Before: Sep  2 09:54:07 2008 GMT
                Not After : Jan 19 09:54:07 2036 GMT
        Subject:  C=IN,ST=KAR,O=Citrix Pvt Ltd,CN=Citrix
        Public Key Algorithm: rsaEncryption
        Public Key size: 1024

3)      Name: bundletest
        Cert Path: bundle9.pem
        Key Path: bundle9.pem
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Validity
                Not Before: Nov 28 06:43:11 2014 GMT
                Not After : Nov 25 06:43:11 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=Server9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

4)      Name: bundletest_ic1
        Cert Path: bundle9.pem_ic1
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
```



```
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Validity
                Not Before: Nov 28 06:42:56 2014 GMT
                Not After : Nov 25 06:42:56 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

5)      Name: bundletest_ic2
        Cert Path: bundle9.pem_ic2
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Validity
                Not Before: Nov 28 06:42:55 2014 GMT
                Not After : Nov 25 06:42:55 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

6)      Name: bundletest_ic3
        Cert Path: bundle9.pem_ic3
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA6
        Validity
                Not Before: Nov 28 06:42:53 2014 GMT
                Not After : Nov 25 06:42:53 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
```



```
115
116 7)       Name: bundletest_ic4
117         Cert Path: bundle9.pem_ic4
118         Format: PEM
119         Status: Valid,   Days to expiration:3078
120         Certificate Expiry Monitor: ENABLED
121         Expiry Notification period: 30 days
122         Certificate Type: Intermediate CA
123         Version: 3
124         Serial Number: 01
125         Signature Algorithm: sha256WithRSAEncryption
126         Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA5
127         Validity
128                 Not Before: Nov 28 06:42:51 2014 GMT
129                 Not After : Nov 25 06:42:51 2024 GMT
130         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA6
131         Public Key Algorithm: rsaEncryption
132         Public Key size: 2048
133
134 8)       Name: bundletest_ic5
135         Cert Path: bundle9.pem_ic5
136         Format: PEM
137         Status: Valid,   Days to expiration:3078
138         Certificate Expiry Monitor: ENABLED
139         Expiry Notification period: 30 days
140         Certificate Type: Intermediate CA
141         Version: 3
142         Serial Number: 01
143         Signature Algorithm: sha256WithRSAEncryption
144         Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA4
145         Validity
146                 Not Before: Nov 28 06:42:50 2014 GMT
147                 Not After : Nov 25 06:42:50 2024 GMT
148         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA5
149         Public Key Algorithm: rsaEncryption
150         Public Key size: 2048
151
152 9)       Name: bundletest_ic6
153         Cert Path: bundle9.pem_ic6
154         Format: PEM
155         Status: Valid,   Days to expiration:3078
156         Certificate Expiry Monitor: ENABLED
157         Expiry Notification period: 30 days
158         Certificate Type: Intermediate CA
159         Version: 3
```



```
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA3
        Validity
                Not Before: Nov 28 06:42:48 2014 GMT
                Not After : Nov 25 06:42:48 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA4
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

10)     Name: bundletest_ic7
        Cert Path: bundle9.pem_ic7
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA2
        Validity
                Not Before: Nov 28 06:42:46 2014 GMT
                Not After : Nov 25 06:42:46 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA3
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

11)     Name: bundletest_ic8
        Cert Path: bundle9.pem_ic8
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA1
        Validity
                Not Before: Nov 28 06:42:45 2014 GMT
                Not After : Nov 25 06:42:45 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA2
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
```



```
205
206 12)      Name: bundletest_ic9
207         Cert Path: bundle9.pem_ic9
208         Format: PEM
209         Status: Valid,   Days to expiration:3078
210         Certificate Expiry Monitor: ENABLED
211         Expiry Notification period: 30 days
212         Certificate Type: Intermediate CA
213         Version: 3
214         Serial Number: 01
215         Signature Algorithm: sha256WithRSAEncryption
216         Issuer:  C=IN,ST=ka,O=sslteam,CN=RootCA4096
217         Validity
218                 Not Before: Nov 28 06:42:43 2014 GMT
219                 Not After : Nov 25 06:42:43 2024 GMT
220         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA1
221         Public Key Algorithm: rsaEncryption
222         Public Key size: 2048
223 Done
224
225 sh ssl certlink
226
227 1)      Cert Name: bundletest    CA Cert Name: bundletest_ic1
228 2)      Cert Name: bundletest_ic1        CA Cert Name: bundletest_ic2
229 3)      Cert Name: bundletest_ic2        CA Cert Name: bundletest_ic3
230 4)      Cert Name: bundletest_ic3        CA Cert Name: bundletest_ic4
231 5)      Cert Name: bundletest_ic4        CA Cert Name: bundletest_ic5
232 6)      Cert Name: bundletest_ic5        CA Cert Name: bundletest_ic6
233 7)      Cert Name: bundletest_ic6        CA Cert Name: bundletest_ic7
234 8)      Cert Name: bundletest_ic7        CA Cert Name: bundletest_ic8
235 9)      Cert Name: bundletest_ic8        CA Cert Name: bundletest_ic9
236 Done
237 <!--NeedCopy-->
```

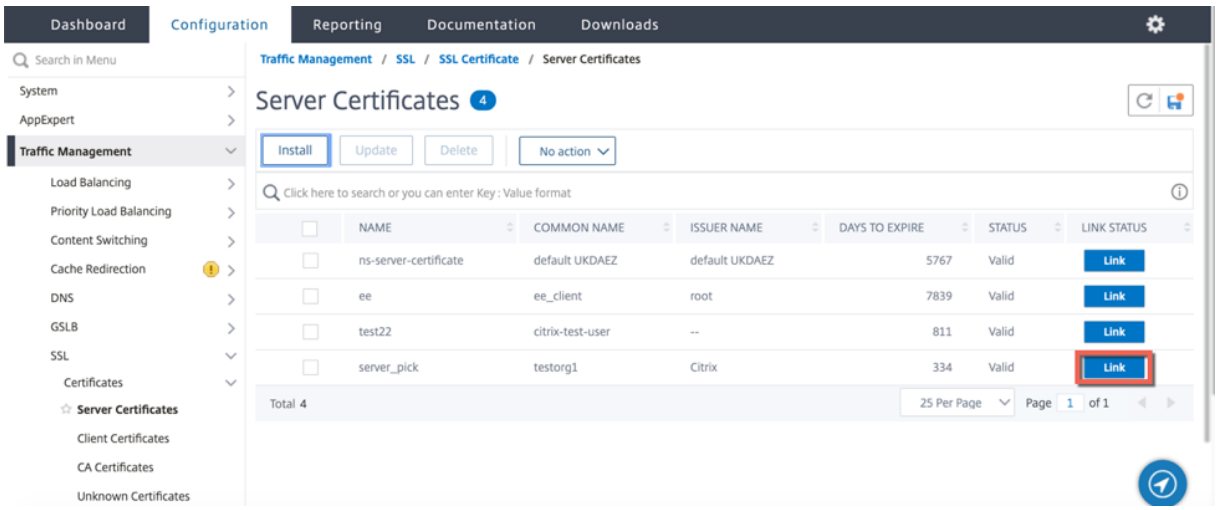
GUI を使用して証明書セットを追加する

1. トラフィック管理 > **SSL** > 証明書 > **CA** 証明書に移動します。
2. 詳細ウィンドウで、[インストール] をクリックします。
3. [証明書のインストール] ダイアログボックスで、証明書やキーファイル名などの詳細を入力し、[証明書バンドル] を選択します。
4. [インストール] をクリックし、[閉じる] をクリックします。

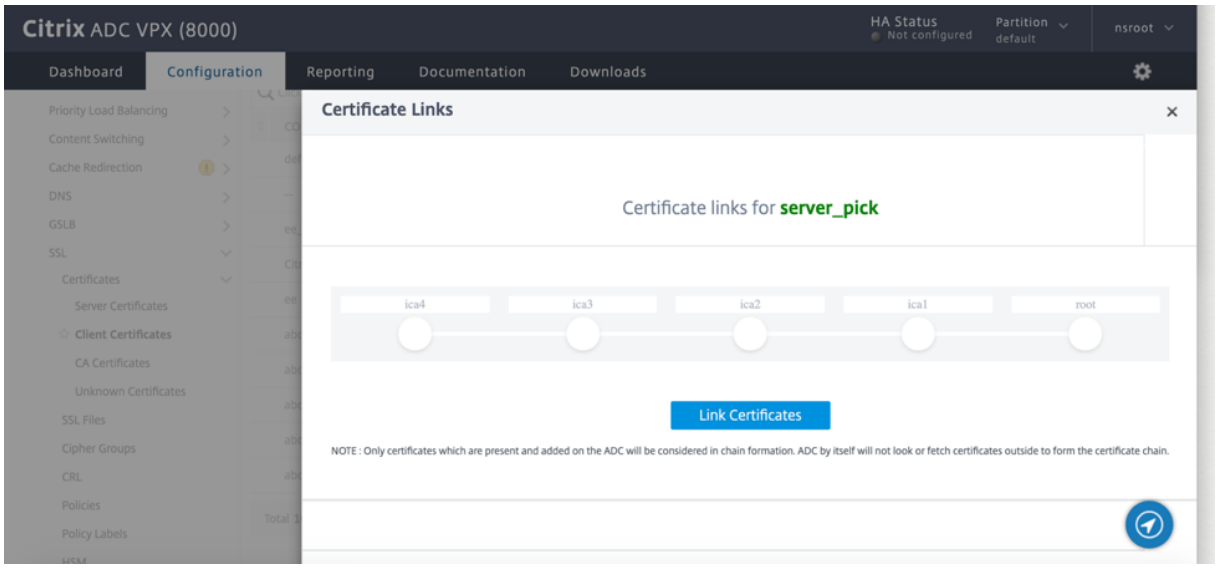
証明書リンクの自動化

注: この機能は、リリース 13.0 ビルド 47.x から利用できます。

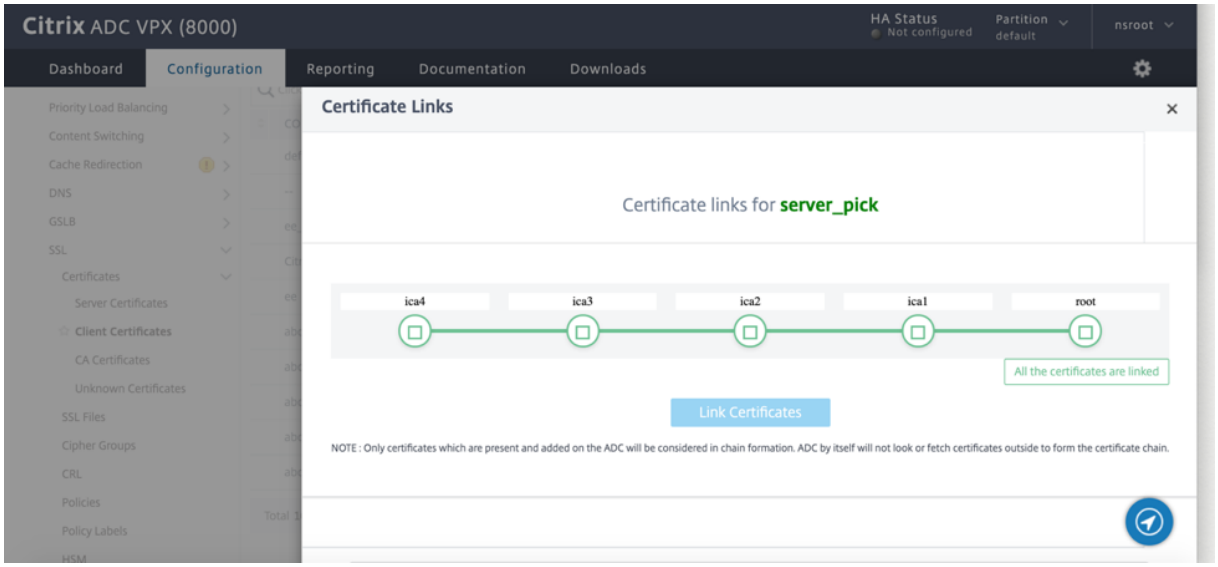
ルート証明書に至るまで、証明書をその発行者に手動でリンクする必要がなくなりました。中間 CA 証明書とルート証明書がアプライアンスに存在する場合は、エンドユーザ証明書の [リンク（Link）] ボタンをクリックできます。



ポテンシャルチェーンが表示されます。



[証明書をリンク] をクリックして、すべての証明書をリンクします。



証明書のチェーンを作成する

証明書のセット (1 つのファイル) を使用する代わりに、証明書のチェーンを作成できます。チェーンはサーバー証明書をその発行者 (中間 CA) にリンクします。この方法では、中間 CA 証明書ファイルが ADC アプライアンスにインストールされ、クライアントアプリケーションがチェーン内の証明書の 1 つを信頼する必要があります。たとえば、Cert-Intermediate-A を Cert-Intermediate-B にリンクします。ここで、Cert-Intermediate-B は、クライアントアプリケーションによって信頼される証明書である Cert-Intermediate-C にリンクされます。

注：アプライアンスは、クライアントに送信される証明書のチェーン内で最大 10 個の証明書（1 つのサーバ証明書と 9 つの CA 証明書）の送信をサポートします。

CLI を使用して証明書チェーンを作成する

コマンドプロンプトで次のコマンドを入力して証明書チェーンを作成し、構成を確認します。(チェーン内の新しいリンクごとに 1 つ目のコマンドを繰り返します)。

```
link ssl certkey <certKeyName> <linkCertKeyName>
show ssl certlink
<!--NeedCopy-->
```

例:

```
link ssl certkey siteAcertkey CAcertkey
Done

```



```
show ssl certlink

linked certificate:
       1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
Done
<!--NeedCopy-->
```

GUI を使用して証明書チェーンを作成する

1. トラフィック管理 > **SSL** > 証明書に移動します。
2. サーバー証明書を選択し、［アクション］リストで［リンク］を選択し、CA 証明書名を指定します。

既存のサーバー証明書を更新する

既存のサーバ証明書を手動で変更するには、次の手順を実行する必要があります。

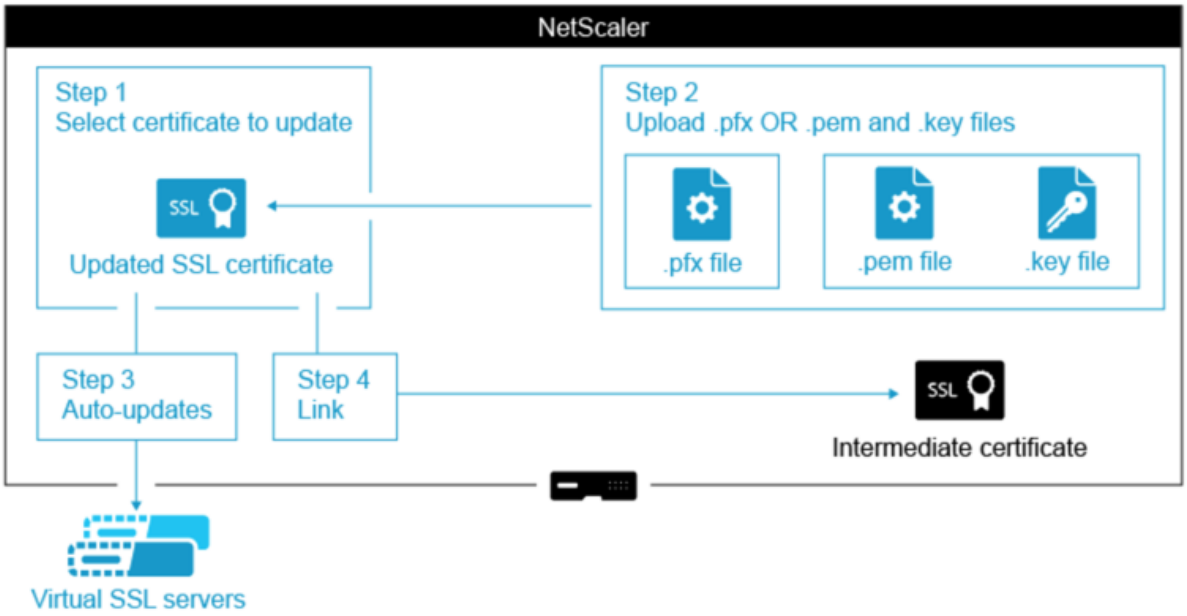
1. 仮想サーバから古い証明書のバインドを解除します。
2. 証明書をアプライアンスから削除します。
3. 新しい証明書をアプライアンスに追加します。
4. 新しい証明書を仮想サーバにバインドします。

証明書とキーのペアを置き換える際のダウンタイムを短縮するために、既存の証明書を更新できます。証明書を別のドメインに発行された証明書に置き換える場合は、証明書を更新する前にドメインチェックを無効にする必要があります。

有効期限が切れる証明書に関する通知を受け取るには、有効期限モニターを有効にします。

構成済みの SSL 仮想サーバーまたはサービスから証明書を削除またはバインド解除すると、その仮想サーバーまたはサービスは非アクティブになります。新しい有効な証明書がバインドされると、アクティブになります。ダウンタイムを減らすために、更新機能を使用して、SSL 仮想サーバーまたは SSL サービスにバインドされている証明書とキーのペアを置き換えることができます。

Citrix ADC アプライアンスで SSL 証明書を更新する方法の概要図。



既存の証明書を更新するにはどうしたらいいですか? へのビデオリンク。

CLI を使用して既存の証明書とキーペアを更新する

コマンドプロンプトで次のコマンドを入力して、既存の証明書とキーのペアを更新し、構成を確認します。

```
update ssl certkey <certkeyName> -cert <string> -key <string>

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

例:

```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
    nsconfig/ssl/pkey.pem

Done

show ssl certkey siteAcertkey

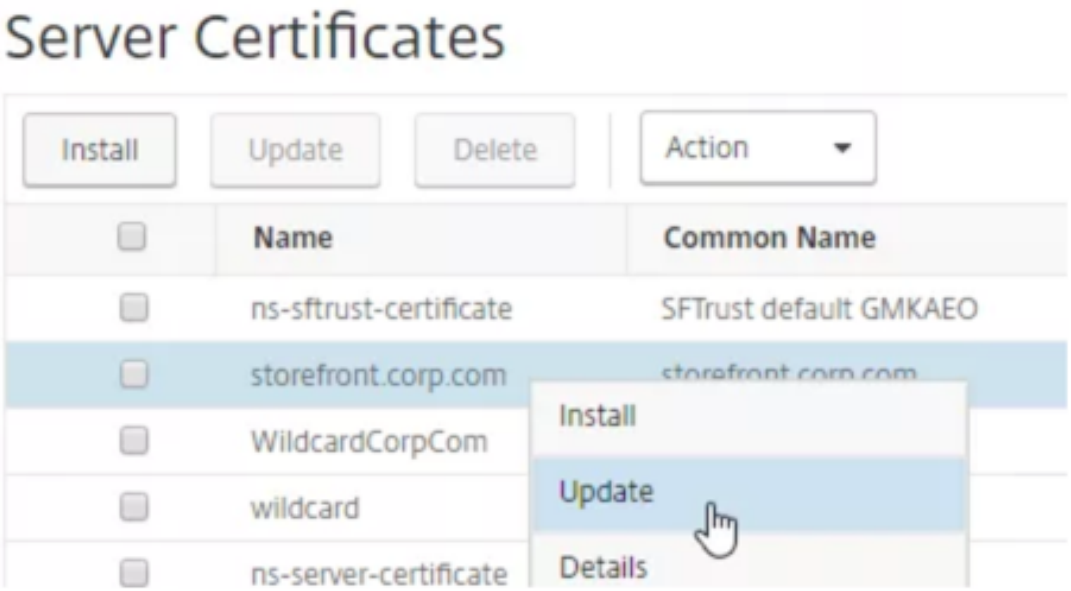
Name: siteAcertkey       Status: Valid
           Version: 3
           Serial Number: 02
           Signature Algorithm: md5WithRSAEncryption
           Issuer: /C=US/ST=CA/L=Santa Clara/O=siteA/OU=Tech
```



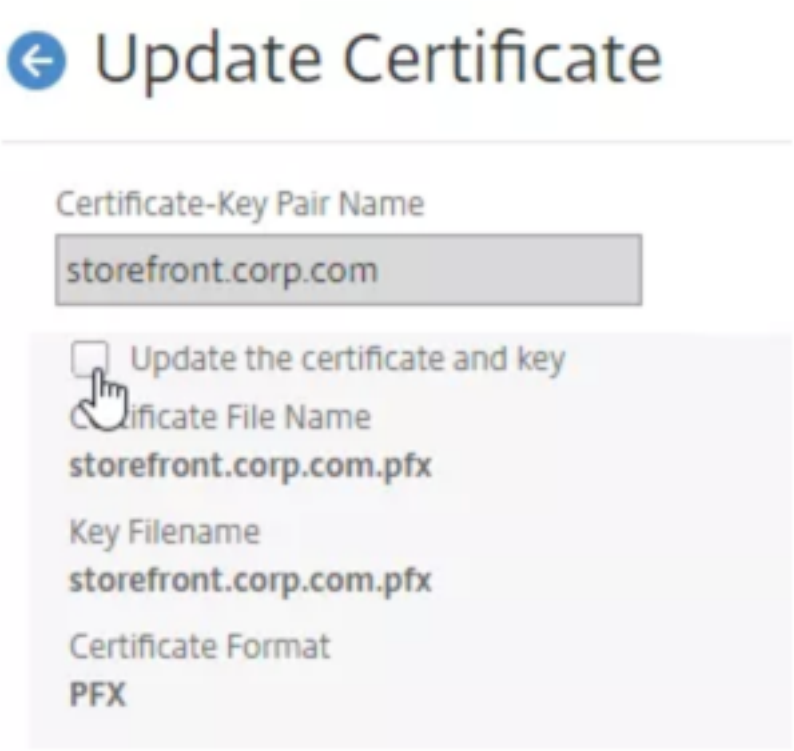
```
            Validity
                 Not Before: Nov 11 14:58:18 2001 GMT
                 Not After: Aug 7 14:58:18 2004 GMT
            Subject: /C=US/ST-CA/L=San Jose/O=CA/OU=Security
            Public Key Algorithm: rsaEncryption
            Public Key size: 2048
Done
<!--NeedCopy-->
```

GUI を使用して既存の証明書とキーのペアを更新する

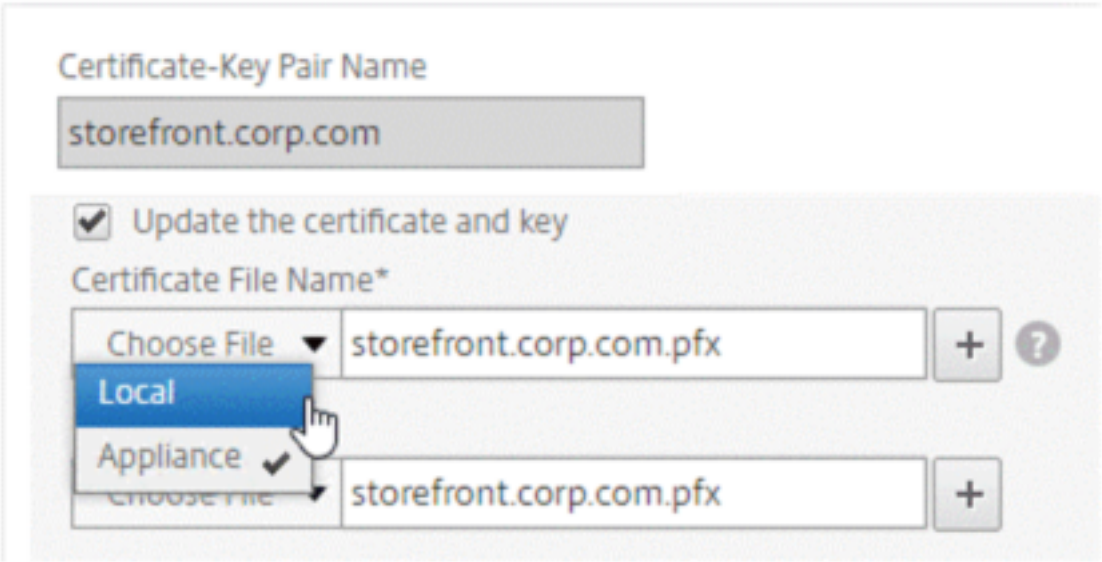
1. トラフィック管理 > **SSL** > 証明書 > サーバ証明書に移動します。
2. 更新する証明書を選択し、[**Update**] をクリックします。



3. [証明書とキーを更新] を選択します。



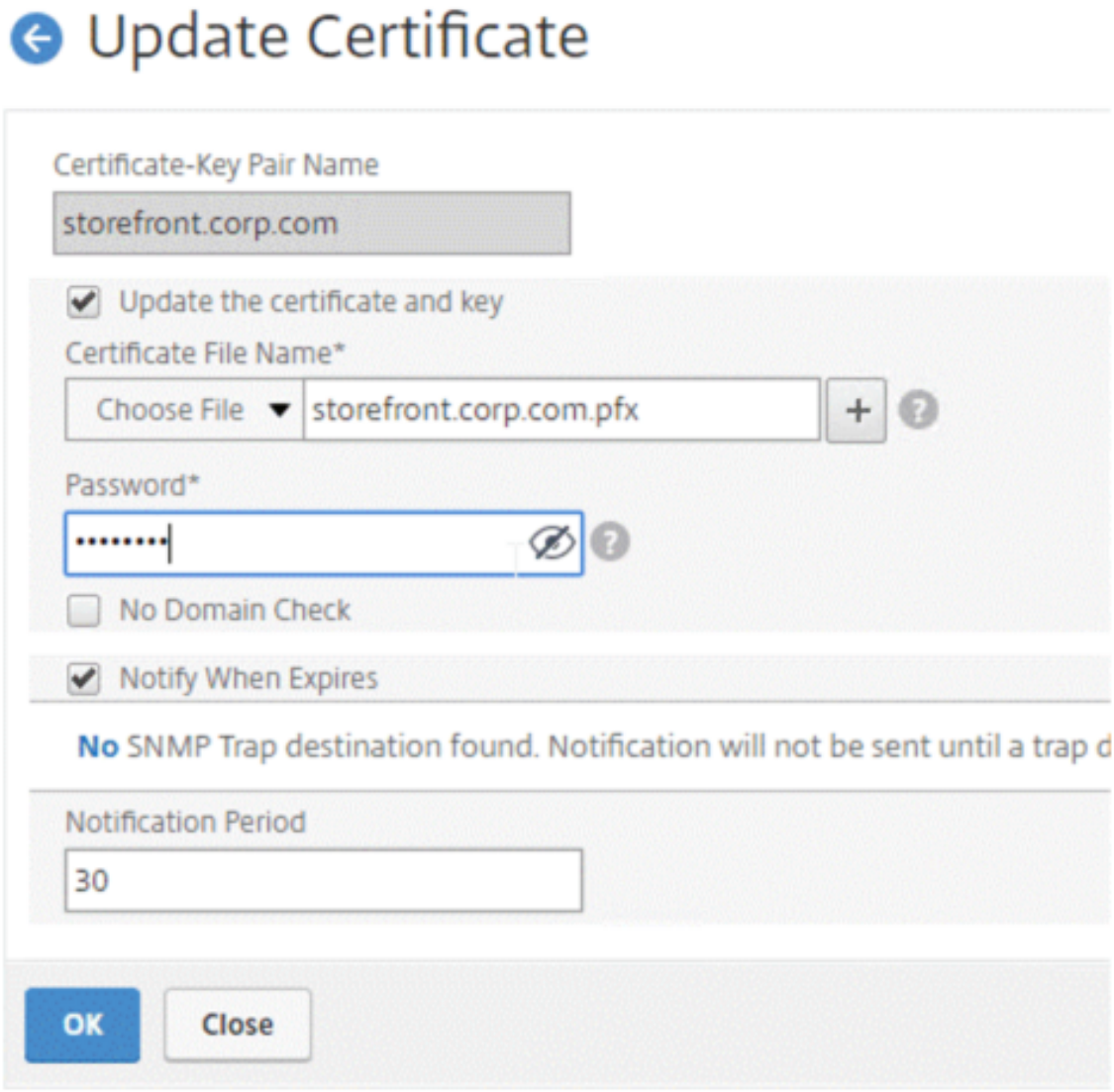
4. [証明書ファイル名] で、[ファイルの選択] > [ローカル] をクリックし、更新された.pfx ファイルまたは証明書 PEM ファイルを参照します。



- .pfx ファイルをアップロードすると、.pfx ファイルのパスワードを指定するよう求められます。
- 証明書 pem ファイルをアップロードする場合は、証明書キーファイルもアップロードする必要があります。キーが暗号化されている場合は、暗号化パスワードを指定する必要があります。

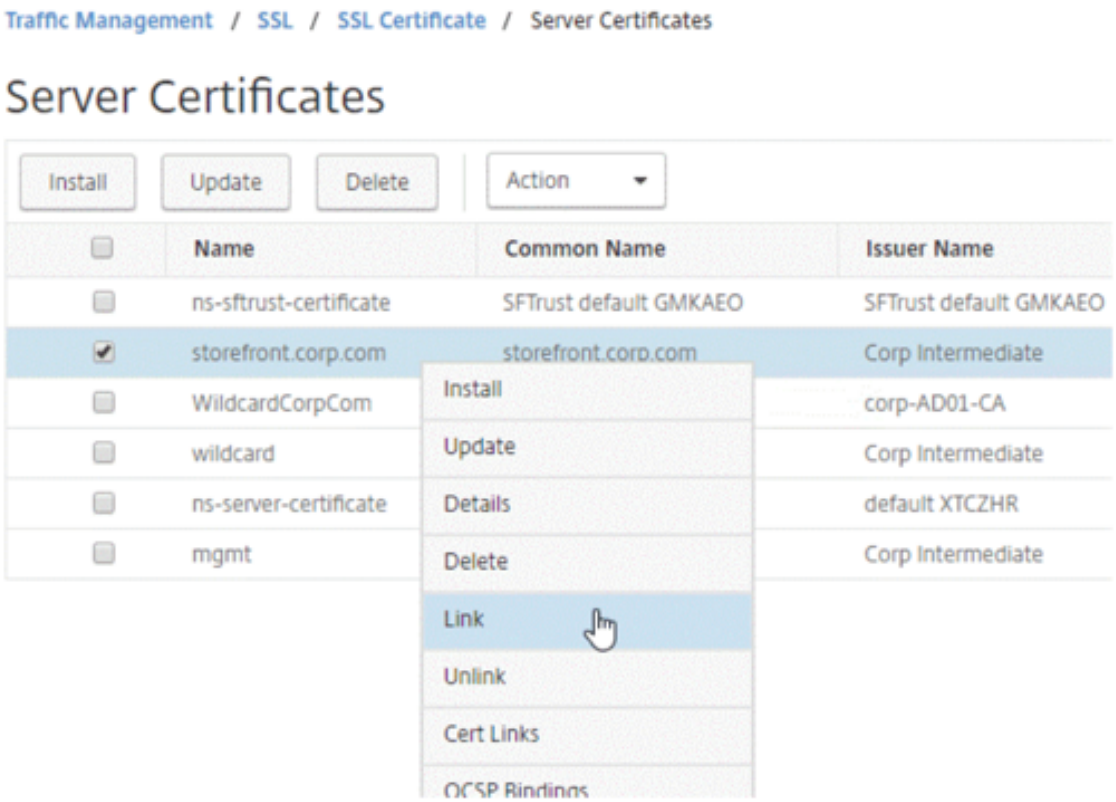
5. 新しい証明書の共通名が古い証明書と一致しない場合は、[**No Domain Check**] を選択します。

6. [**OK**] をクリックします。この証明書がバインドされているすべての SSL 仮想サーバーは自動的に更新されます。



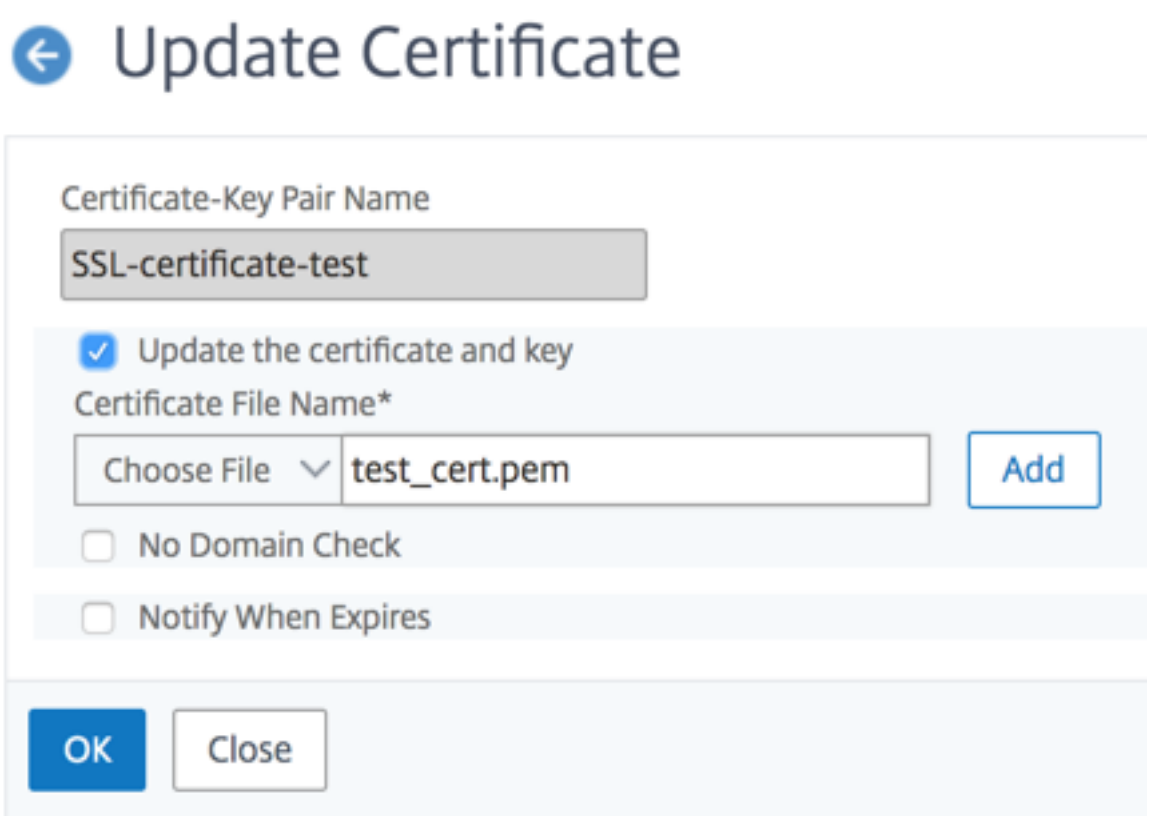
7. 証明書を置き換えた後、証明書リンクを新しい中間証明書に更新しなければならない場合があります。リンクを壊さずに中間証明書を更新する方法の詳細については、「リンクを壊さずに中間証明書を更新する」を参照してください。

   - 更新された証明書を右クリックし、[証明書リンク] をクリックして、中間証明書にリンクされているかどうかを確認します。
   - 証明書がリンクされていない場合は、更新された証明書を右クリックし、[リンク] をクリックして中間証明書にリンクします。リンクするオプションが表示されない場合は、まず、新しい中間証明書をアプライアンスの [**CA Certificates**] ノードにインストールする必要があります。



既存の **CA** 証明書を更新する

既存の CA 証明書を更新する手順は、既存のサーバ証明書を更新する手順と同じです。唯一の違いは、CA 証明書の場合、キーは必要ないということです。



ドメインチェックを無効にする

アプライアンスで SSL 証明書が置き換えられる場合、新しい証明書に記載されているドメイン名は、置き換えられる証明書のドメイン名と一致する必要があります。たとえば、abc.com に発行された証明書があり、def.com に発行された証明書で更新すると、証明書の更新は失敗します。

ただし、特定のドメインをホストしていたサーバーで新しいドメインをホストする場合は、証明書を更新する前にドメインチェックを無効にします。

CLI を使用して証明書のドメインチェックを無効にする

コマンドプロンプトで次のコマンドを入力して、ドメインチェックを無効にし、構成を確認します。

```
update ssl certKey <certkeyName> -noDomainCheck

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

例:


```
update ssl certKey sv -noDomainCheck

Done

show ssl certkey sv

Name: sv
Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
Format: PEM
Status: Valid,   Days to expiration:9349
Certificate Expiry Monitor: DISABLED
Done
<!--NeedCopy-->
```

GUI を使用して証明書のドメインチェックを無効にする

1. [トラフィック管理] **> [SSL] > [**証明書] に移動し、証明書を選択して [更新] をクリックします。
2. [ドメインチェックなし] を選択します。

ADC アプライアンスのデフォルト証明書を、アプライアンスのホスト名と一致する信頼できる **CA** 証明書に置き換えます

次の手順では、デフォルトの証明書 (ns-server-certificate) が内部サービスにバインドされていることを前提としています。

1. トラフィック管理 **> SSL > SSL** 証明書 **>** 証明書要求の作成に移動します。
2. [共通名] に、test.citrixadc.comと入力します。
3. CSR を信頼できる認証局に送信します。
4. 信頼できる CA から証明書を受け取ったら、このファイルを/nsconfig/sslディレクトリにコピーします。
5. トラフィック管理 **> SSL >** 証明書 **>** サーバ証明書に移動します。
6. デフォルトのサーバ証明書 (ns-server-certificate) を選択し、[**Update**] をクリックします。
7. [証明書の更新] ダイアログボックスの [証明書ファイル名] で、署名後に CA から受け取った証明書を参照します。
8. [**Key File Name**] フィールドで、デフォルトのプライベートキーファイル名 (ns-server.key) を指定します。
9. [ドメインチェックなし] を選択します。
10. [**OK**] をクリックします。

有効期限モニターを有効にする

SSL 証明書は一定期間有効です。一般的な展開には、SSL トランザクションを処理する複数の仮想サーバーが含まれ、それらにバインドされた証明書は異なる時間に期限切れになる可能性があります。アプライアンスに設定された有効期限モニタは、設定された証明書の有効期限が切れると、アプライアンスの syslog および ns 監査ログにエントリを作成します。

証明書の期限切れに関する SNMP アラートを作成する場合は、別途設定する必要があります。

CLI を使用して証明書の有効期限モニターを有効にする

コマンドプロンプトで次のコマンドを入力して、証明書の有効期限モニターを有効にし、構成を確認します。

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED ) [-
    notificationPeriod <positive_integer>]]

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

例**:**

```
set ssl certKey sv -expiryMonitor ENABLED  - notificationPeriod 60
Done
<!--NeedCopy-->
```

GUI を使用して証明書の有効期限モニターを有効にする

1. [トラフィック管理**] > [SSL] > [**証明書] に移動し、証明書を選択して [更新] をクリックします。
2. [有効期限が切れたら通知する] を選択し、必要に応じて通知期間を指定します。

リンクを解除せずに中間証明書を更新する

既存のリンクを切断することなく、中間証明書を更新できるようになりました。置き換えられる証明書によって発行されたリンクされた証明書の AuthorityKeyIdentifier 拡張には、認証局証明書のシリアル番号 (‘AuthorityCertSerialNumber’) フィールドを含めないでください。「AuthorityKeyIdentifier」拡張にシリアル番号フィールドが含まれる場合、古い証明書と新しい証明書のシリアル番号が同じである必要があります。上記の条件が満たされれば、リンク内の証明書をいくつでも 1 つずつ更新できます。以前は、中間証明書が更新されるとリンクが壊れました。

たとえばCertA、CertB、CertC、CertDの 4 つの証明書があります。証明書CertAはCertBの発行者、CertBはCertCの発行者、などです。リンクを解除せずに中間証明書CertBをCertB_newに置き換える場合は、次の条件を満たす必要があります。

次の両方の条件が満たされる場合、CertBの証明書シリアル番号はCertB_newの証明書シリアル番号と一致する必要があります。

- AuthorityKeyIdentifier拡張子はCertCにあります。
- この拡張機能にはシリアル番号フィールドが含まれます。

証明書内の共通名が変更された場合は、証明書の更新中にnodomaincheckを指定します。

前述の例では、CertDの「www.example.com」を「*.example.com」に変更するには、「ドメインチェックなし」パラメータを選択します。

CLI を使用して証明書を更新する

コマンドプロンプトで入力します。

```
update ssl certKey <certkeyName> -cert <string> [-password] -key <
    string>  [-noDomainCheck]
<!--NeedCopy-->
```

例**:**

```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
    nsconfig/ssl/pkey.pem -noDomainCheck
<!--NeedCopy-->
```

証明書チェーンを表示する

証明書には、発行機関の名前と、証明書が発行されたサブジェクトが含まれます。証明書を検証するには、その証明書の発行者を調べて、その発行者を信頼しているかどうかを確認する必要があります。発行者を信頼できない場合は、発行者証明書の発行者を確認する必要があります。ルート CA 証明書または信頼できる発行者に到達するまで、チェーンを上に移動します。

SSL ハンドシェイクの一環として、クライアントが証明書を要求すると、アプライアンスは証明書と、アプライアンスに存在する発行者証明書のチェーンを提示します。管理者は、アプライアンスに存在する証明書の証明書チェーンを表示し、不足している証明書をインストールできます。

CLI を使用して、アプライアンスに存在する証明書の証明書チェーンを表示する

コマンドプロンプトで入力します。

```
show ssl certchain <cert_name>
<!--NeedCopy-->
```

例

c1、c2、および c3 の 3 つの証明書があります。証明書 c3 はルート CA 証明書で、c2 に署名し、c2 は c1 に署名します。次の例は、さまざまなシナリオでの
`show ssl certchain c1` コマンドの出力を示しています。

シナリオ **1:**

証明書 c2 は c1 にリンクされ、c3 は c2 にリンクされています。

証明書 c3 はルート CA 証明書です。

次のコマンドを実行すると、ルート CA 証明書までの証明書リンクが表示されます。

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate name: c2               linked; not a root
       certificate
    2) Certificate name: c3               linked; root certificate
Done
<!--NeedCopy-->
```

シナリオ **2:**

証明書 c2 は c1 にリンクされています。

証明書 c2 はルート CA 証明書ではありません。

次のコマンドを実行すると、証明書 c3 はルート CA 証明書であるが、c2 にリンクされていないという情報が表示されます。

```
show ssl certchain c1

Certificate chain  details of certificate name c1 are:
```



```
4       1) Certificate Name: c2                linked; not a root
           certificate
5       2) Certificate Name: c3                not linked; root certificate
6       Done
7   <!--NeedCopy-->
```

シナリオ **3:**

証明書 c1、c2、および c3 はリンクされていませんが、アプライアンス上に存在します。

次のコマンドを実行すると、証明書 c1 の発行元から始まるすべての証明書に関する情報が表示されます。また、証明書がリンクされていないことも指定されています。

```
1   show ssl certchain c1
2
3   Certificate chain details of certificate name c1 are:
4       1) Certificate Name: c2                not linked; not a root
           certificate
5       2) Certificate Name: c3                not linked; root certificate
6       Done
7   <!--NeedCopy-->
```

シナリオ **4:**

証明書 c2 は c1 にリンクされています。

証明書 c3 はアプライアンスに存在しません。

次のコマンドを実行すると、c1 にリンクされた証明書に関する情報が表示されます。c2 で指定したサブジェクト名の証明書を追加するよう求められます。この場合、ユーザーはルート CA 証明書 c3 を追加するように求められます。

```
1   show ssl certchain c1
2
3   Certificate chain details of certificate name c1 are:
4       1) Certificate Name: c2                linked; not a root
           certificate
5       2) Certificate Name: /C=IN/ST=ka/O=netscaler/CN=test
6          Action: Add a certificate with this subject name.
7   Done
8   <!--NeedCopy-->
```

シナリオ **5:**

証明書が証明書 c1 にリンクされておらず、c1 の発行者証明書がアプライアンスに存在しません。

次のコマンドを実行すると、証明書 c1 にサブジェクト名を持つ証明書を追加するよう求められます。

```
sh ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate Name: /ST=KA/C=IN
       Action: Add a certificate with this subject name.
<!--NeedCopy-->
```

サーバーテスト証明書を生成する

October 7, 2021

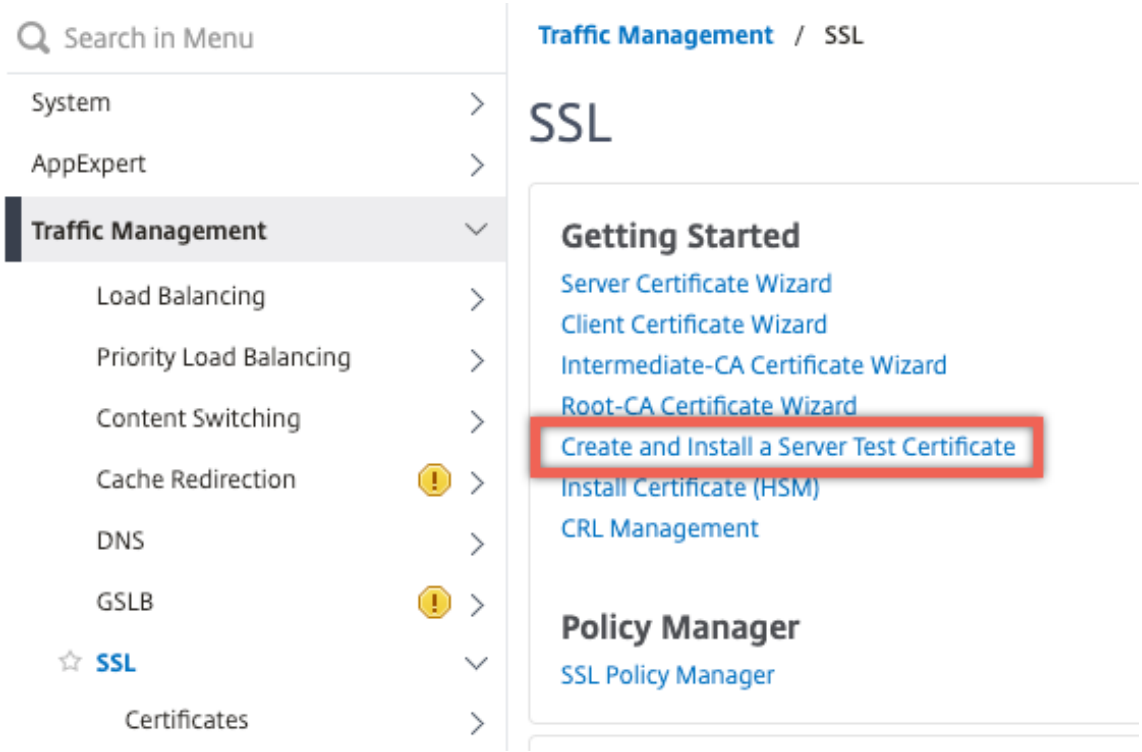
Citrix ADC アプライアンスでは、構成ユーティリティーの GUI ウィザードを使用して、サーバー認証用のテスト証明書を作成できます。サーバー証明書は、SSL ハンドシェイクでサーバーを認証および識別するために使用されます。通常、信頼できる CA はサーバー証明書を発行します。サーバーは、証明書を使用してサーバーを認証するクライアントに証明書を送信します。

サーバー・テスト証明書を発行する場合、アプライアンスは証明機関として動作します。この証明書は、クライアントとの SSL ハンドシェイクで認証するために SSL 仮想サーバーにバインドできます。この証明書はテストのみを目的としています。実稼働環境では使用しないでください。

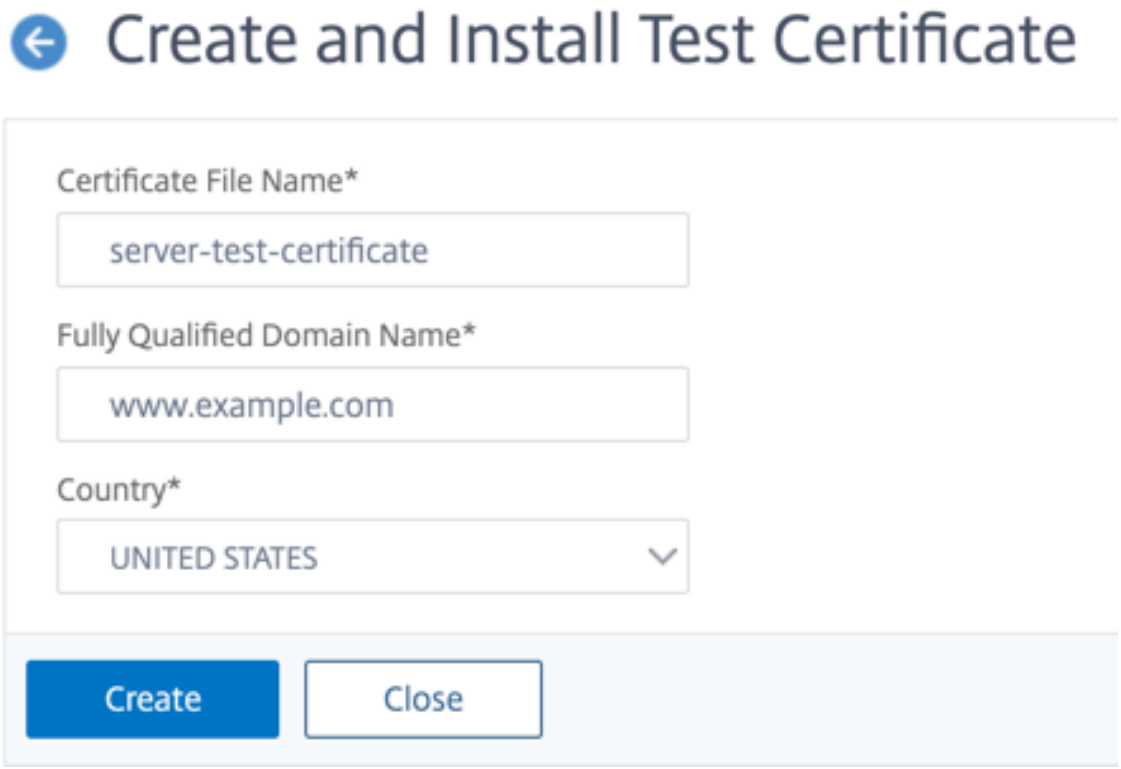
サーバーテスト証明書は、SSL または SSL_TCP プロトコルを使用する任意の仮想サーバーにインストールできます。

GUI を使用してサーバーテスト証明書を生成する

1. **Traffic Management > SSL** に移動して **SSL Certificates group** で **Create and Install a Server Test Certificate** を選択します。



2. 次のパラメータの値を入力し、【作成】をクリックします。



SSL ファイルのインポートと変換

October 7, 2021

これらのホストへの FTP アクセスが使用できない場合でも、リモートホストから証明書、秘密キー、CRL、DH キーなどの SSL リソースをインポートできるようになりました。この機能は、リモートホストへのシェルアクセスが制限されている環境で特に役立ちます。デフォルトのフォルダは、/nsconfig/ssl に次のように作成されます。

- 証明書ファイルの場合：/nsconfig/ssl/certfile
- 秘密鍵の場合：/nsconfig/ssl/keyfile
- CRL の場合：/var/netscaler/ssl/crlfile
- DH キーの場合：/nsconfig/ssl/dhfile

HTTP サーバと HTTPS サーバの両方からのインポートがサポートされています。ただし、ファイルがアクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。

> 注:
>
> 再起動後にファイルを再インポートするとエラーが発生する可能性があるため、import コマンドは構成ファイル（ns.conf）には保存されません。

証明書ファイルをインポートする

CLI および GUI を使用して、リモートホストからファイル（リソース）をインポートできます。

CLI を使用してリモートホストから証明書ファイルをインポートする

コマンドプロンプトで入力します。

```
import ssl certFile [<name>] [<src>]
<!--NeedCopy-->
```

例:

```
import ssl certfile my-certfile http://www.example.com/file_1
<!--NeedCopy-->
```

```
show ssl certfile
      Name : my-certfile
      URL : http://www.example.com/file_1
```



```
4 <!--NeedCopy-->
```

証明書ファイルを削除するには、「name」引数のみを受け入れる `rm ssl certFile` コマンドを使用します。

CLI を使用してリモートホストからキーファイルをインポートする

コマンドプロンプトで入力します。

```
1 import ssl keyFile [<name>] [<src>]
2 <!--NeedCopy-->
```

例:

```
1 import ssl keyfile my-keyfile http://www.example.com/key_file
2 <!--NeedCopy-->
```

```
1 show ssl keyfile
2       Name : my-keyfile
3       URL : http://www.example.com/key_file
4 <!--NeedCopy-->
```

キーファイルを削除するには、`rm ssl keyFile` コマンドを使用します。このコマンドは「name」引数のみを受け入れます。

CLI を使用してリモートホストから **CRL** ファイルをインポートする

コマンドプロンプトで入力します。

```
1 import ssl crlFile [<name>] [<src>]
2 <!--NeedCopy-->
```

CRL ファイルを削除するには、`rm ssl crlFile` コマンドを使用します。このコマンドでは、<name> 引数だけを指定できます。

例:


```
import ssl crlfile my-crlfile http://www.example.com/crl_file

show ssl crlfile

    Name : my-crlfile
    URL : http://www.example.com/crl_file
<!--NeedCopy-->
```

CLI を使用してリモートホストから **DH** ファイルをインポートする

コマンドプロンプトで入力します。

```
import ssl dhFile [<name>] [<src>]
<!--NeedCopy-->
```

例**:**

```
import ssl dhfile my-dhfile http://www.example.com/dh_file
show ssl dhfile
     Name : my-dhfile
     URL : http://www.example.com/dh_file
<!--NeedCopy-->
```

DH ファイルを削除するには、`rm ssl dhFile`コマンドを使用します。このコマンドでは、<name> 引数だけを指定できます。

GUI を使用した **SSL** リソースのインポート

Traffic Management > SSL > Imports に移動して適切なタブを選択します。

PKCS #8 証明書および **PKCS #12** 証明書をインポートする

ネットワーク内の他のセキュアなサーバーまたはアプリケーションにすでに存在する証明書とキーを使用する場合は、それらをエクスポートし、Citrix ADC アプライアンスにインポートします。エクスポートした証明書とキーを Citrix ADC アプライアンスにインポートする前に、その証明書とキーを変換する必要があります。

ネットワーク内のセキュアなサーバまたはアプリケーションから証明書をエクスポートする方法の詳細については、エクスポート元のサーバまたはアプリケーションのマニュアルを参照してください。

> 注:
>
> Citrix ADC アプライアンスにインストールする場合、キー名と証明書名に、UNIX ファイルシステムでサポートされている文字以外のスペースや特殊文字を含めることはできません。エクスポートされたキーと証明書を保存するときは、適切な命名規則に従います。

証明書と秘密キーのペアは、通常 PKCS #12 形式で送信されます。アプライアンスは、証明書とキーに対して PEM および DER 形式をサポートしています。PKCS #12 を PEM または DER に変換するか、PEM または DER を PKCS #12 に変換するには、このページの「インポートまたはエクスポート用に SSL 証明書を変換する」を参照してください。

Citrix ADC アプライアンスは、PKCS #8 形式の PEM キーをサポートしていません。ただし、OpenSSL インターフェイスを使用して、これらのキーをサポートされている形式に変換できます。OpenSSL インターフェイスは CLI または設定ユーティリティからアクセスできます。キーを変換する前に、秘密キーが PKCS #8 形式であることを確認する必要があります。PKCS #8 形式のキーは通常、次のテキストで始まります。

```
-----BEGIN ENCRYPTED PRIVATE KEY-----


leuSSZQZKgrgUQ==


-----END ENCRYPTED PRIVATE KEY-----
<!--NeedCopy-->
```

CLI から **OpenSSL** インターフェイスを開きます

1. PuTTY などの SSH クライアントを使用して、アプライアンスへの SSH 接続を開きます。
2. 管理者の資格情報を使用して、アプライアンスにログオンします。
3. コマンドプロンプトで shell と入力します。
4. シェルプロンプトで`openssl`と入力します。

GUI から **OpenSSL** インターフェースを開く

Traffic Management > SSL に移動してツールグループで **OpenSSL interface** を選択します。

OpenSSL インターフェイスを使用して、サポートされていない **PKCS #8** キー形式を暗号化されたサポートされているキー形式に変換する

OpenSSL プロンプトで、サポートされていないキー形式が RSA または ECDSA のどちらであるかに応じて、次のいずれかのコマンドを入力します。

```
1 OpenSSL>rsa- in <PKCS#8 Key Filename> -des3 -out <encrypted Key
     Filename>
2
3 OpenSSL>ec -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename
     >
4 <!--NeedCopy-->
```

サポートされていないキー形式をサポートされているキー形式に変換するためのパラメータ

- **PKCS #8** キーファイル名**:** 互換性のない PKCS #8 秘密キーの入力ファイル名。
- 暗号化キーファイル名**:** 互換性のある暗号化プライベートキーの出力ファイル名 (PEM 形式)。
- 暗号化されていないキーファイル名**:** 互換性のある暗号化されていない秘密キーの出力ファイル名 (PEM 形式)。

インポートまたはエクスポート用の **SSL** 証明書の変換

Citrix ADC アプライアンスは、SSL 証明書の PEM および DER 形式をサポートしています。クライアントブラウザや一部の外部セキュリティで保護されたサーバーなど、他のアプリケーションには、さまざまな公開キー暗号規格 (PKCS) 形式が必要です。アプライアンスは変換できます PKCS#12 アプライアンスに証明書をインポートするための PEM または DER 形式にフォーマットし、証明書をエクスポートするため PEM または DER を PKCS#12 に変換できます。セキュリティを強化するために、インポート用のファイルの変換には、DES または DES3 アルゴリズムを使用した秘密鍵の暗号化を含めることができます。

> 注:
>
> GUI を使用してインポートする場合 PKCS#12 証明書、およびパスワードにドル記号が含まれている ($), 逆引用 (‘), またはエスケープ () 文字を使用すると、インポートが失敗する可能性があります。その場合は、「エラー: パスワードが無効です」というメッセージが表示されます。パスワードに特殊文字を使用する必要がある場合は、CLI を使用してすべてのインポートを実行しない限り、必ずエスケープ文字 () を先頭に付けてください。

CLI を使用して証明書の形式を変換する

コマンドプロンプトで、次のコマンドを入力します。


```
convert ssl pkcs12 <outfile> [-import [-pkcs12File <inputFilename>] [-
    des | -des3] [-export [-certFile <inputFilename>] [-keyFile <
    inputFilename>]]
<!--NeedCopy-->
```

操作中に、インポートパスワードまたはエクスポートパスワードの入力を求められます。暗号化されたファイルの場合は、パスフレーズの入力も求められます。

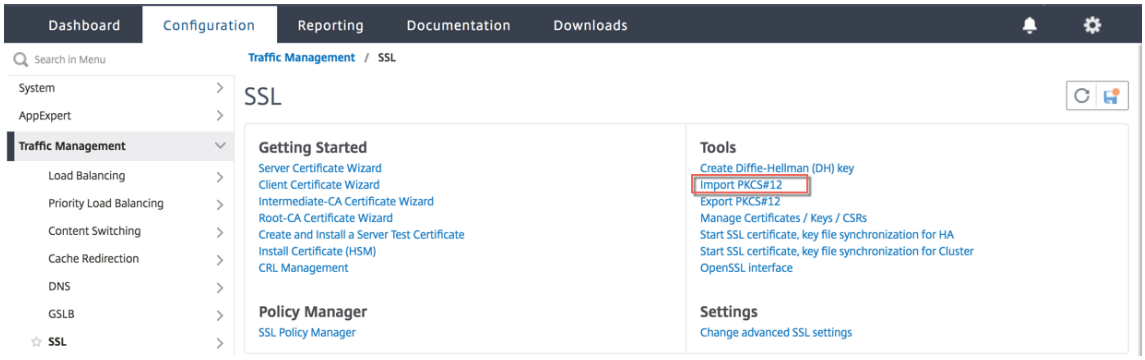
例**:**

```
convert ssl pkcs12 Cert-Import-1.pem -import -pkcs12File Cert-Import-1.
    pfx -des

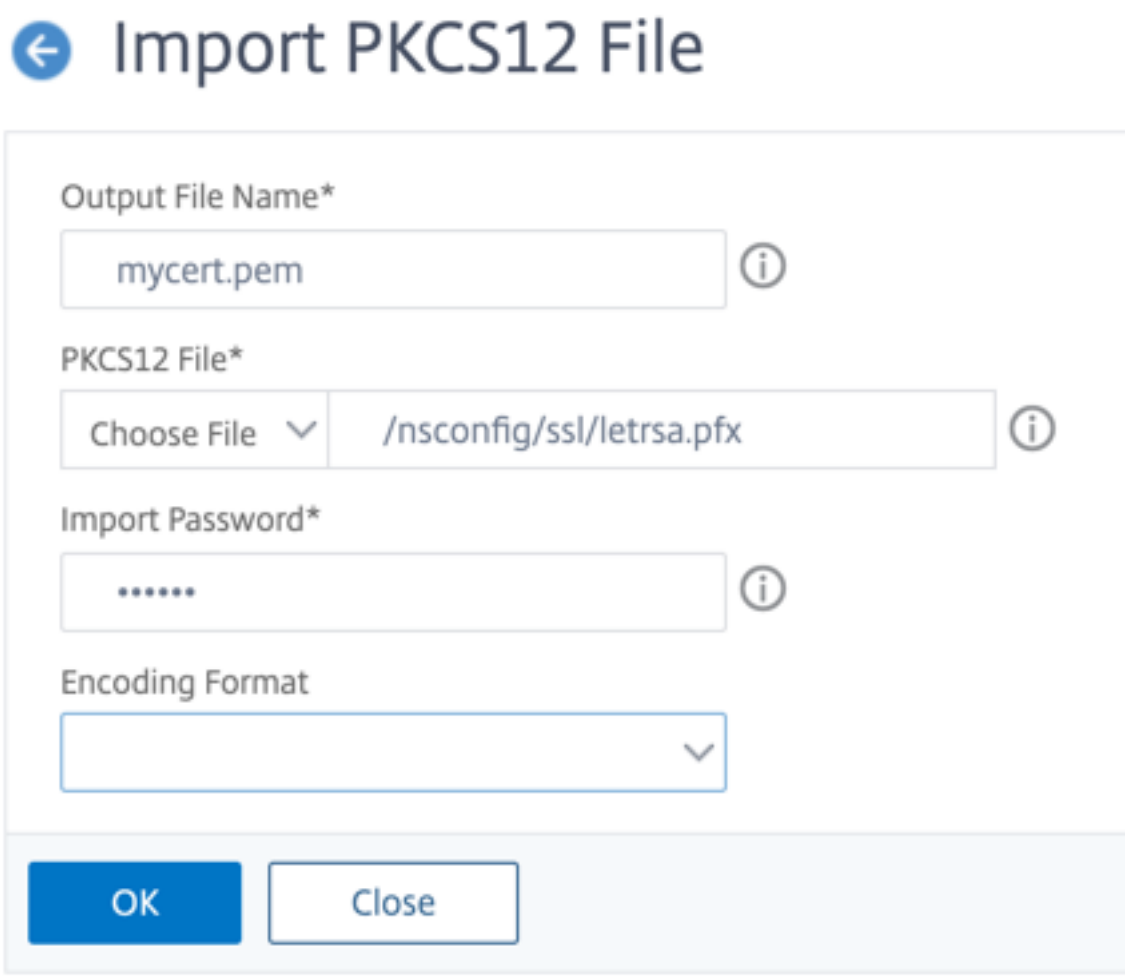
convert ssl pkcs12 Cert-Client-1.pfx -export -certFile Cert-Client-1 -
    keyFile Key-Client-1
<!--NeedCopy-->
```

GUI を使用して証明書の形式を変換する

1. **Traffic Management > SSL** に移動して **Tools** グループで **Import PKCS#12** を選択します。

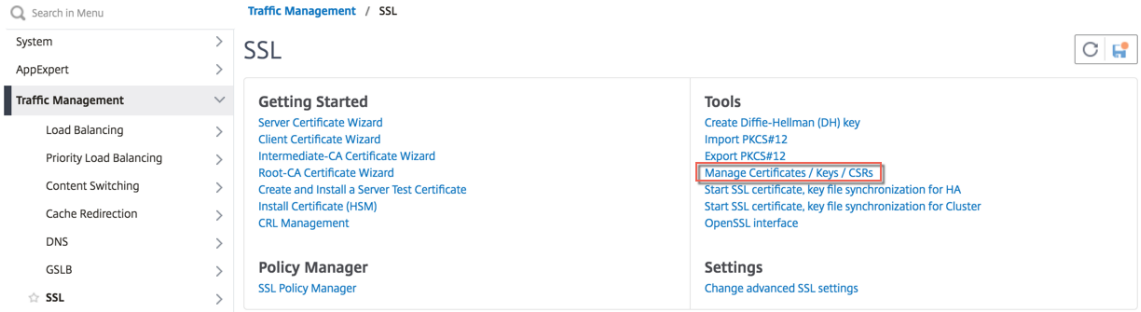


2. [出力ファイル名] フィールドに PEM 証明書名を指定します。
3. ローカルコンピュータまたはアプライアンス上の PFX 証明書の場所を参照します。



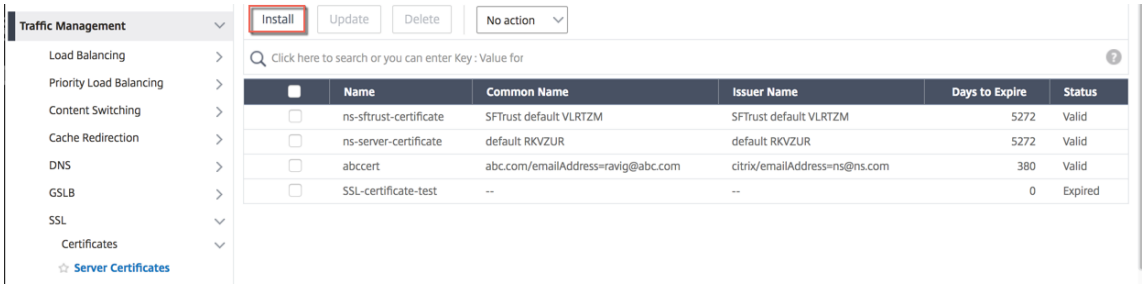
4. [**OK**] をクリックします。

5. [証明書、キー、**CSR** の管理] をクリックして、変換された PEM ファイルを表示します。

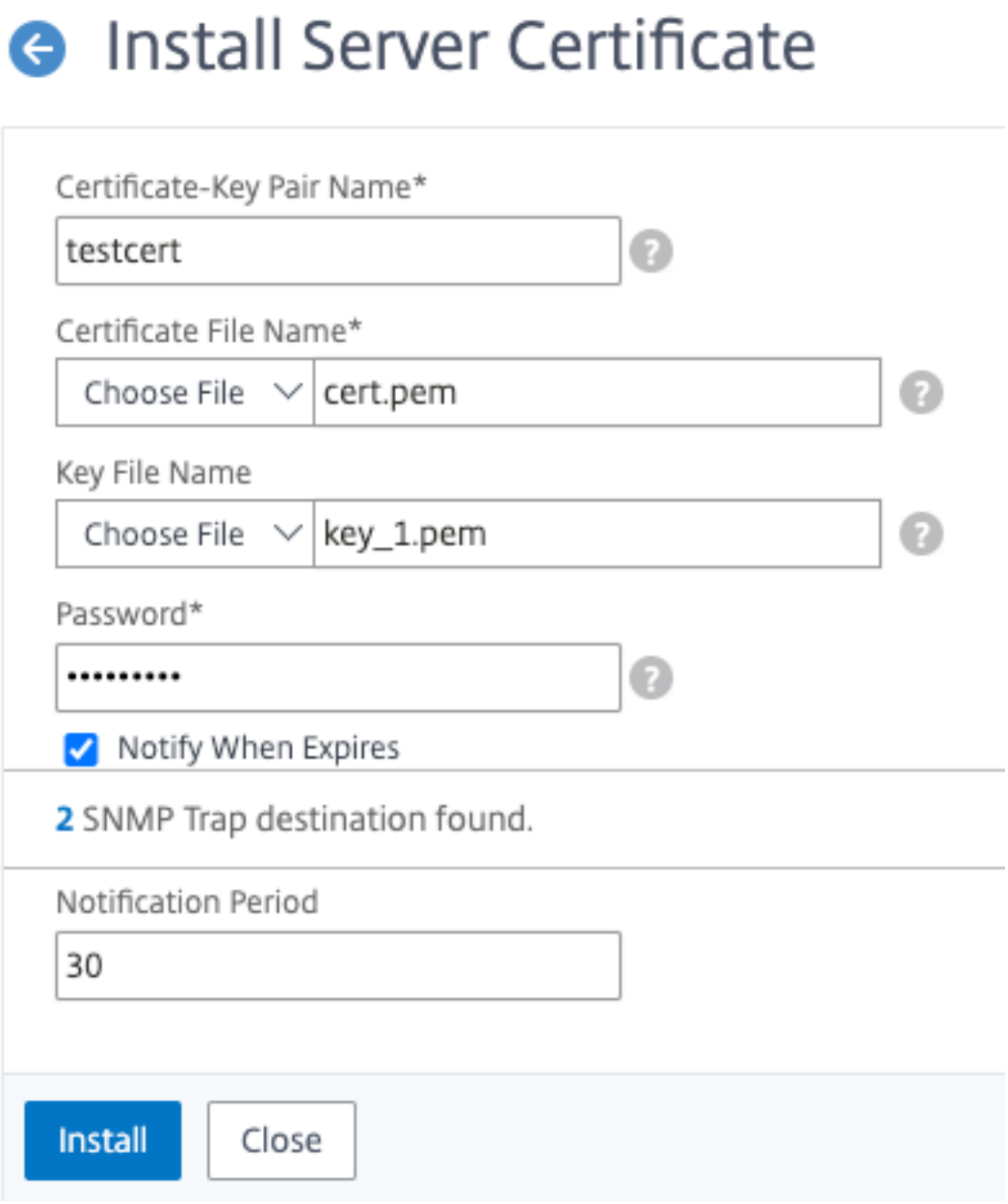


6. アップロードされた PFX ファイルと変換された PEM ファイルを表示できます。

7. [**SSL**] > [証明書] > [サーバー証明書] に移動し、[インストール] をクリックします。



8. 証明書とキーのペア名を指定します。
9. PEM ファイルの場所を参照します。
10. プロンプトが表示されたら、パスワードを指定します。
11. ［インストール］をクリックします。



12. 証明書とキーのペアを SSL 仮想サーバにバインドします。

Bind an SSL certificate to a virtual server on the Citrix ADC appliance

December 7, 2021

An SSL certificate is an essential part of SSL encryption and decryption processes. The certificate is used during an SSL handshake to establish the identity of the SSL server, which is the Citrix ADC appliance as it acts as the SSL termination point for the clients.

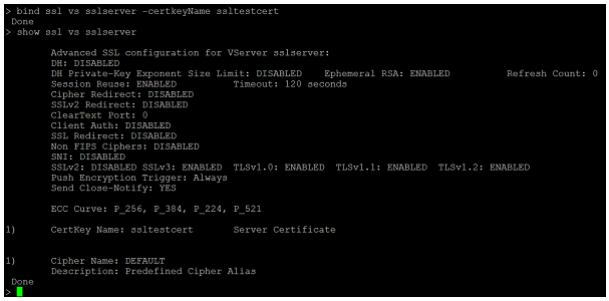
The certificate used for processing the SSL transactions must be bound to the virtual server (SSL) that receives the SSL data.

To bind an SSL certificate to an SSL virtual server using the command line interface

At the command prompt, type:

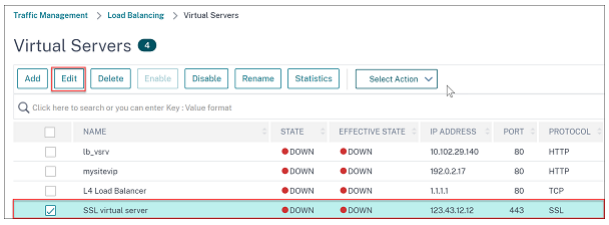
```
bind ssl vs <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vs <vServerName>
<!--NeedCopy-->
```

Example:



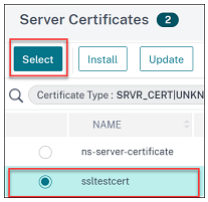
To bind an SSL certificate to an SSL virtual server using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Select a virtual server of type SSL and click **Edit**.



3. In the **Load Balancing Virtual Server** page, under the **Certificates** section, click **No Server Certificate**.

4. In the **Server Certificate Binding** page, click **Click to select**.
5. Select the SSL certificate and click **Select**.



6. Click **Bind** to bind the SSL certificate to the virtual server.
7. Click **Done**.

   You have completed binding the SSL certificate to the virtual server.

SSL プロファイル

October 7, 2021

SSL プロファイルとは、SSL エンティティの設定の集まりです。これは、構成と柔軟性の容易さを提供します。各エンティティの設定を構成する代わりに、プロファイル内で設定を構成し、その設定が適用されるすべてのエンティティにプロファイルをバインドできます。

SSL プロファイルインフラストラクチャは、最新の暗号とプロトコルを使用するように拡張されました。レガシープロファイル（古いプロファイル）と拡張 SSL プロファイル（新しいプロファイル）の相違点が強調表示されます。

古い **SSL** プロファイルインフラストラクチャと新しい **SSL** プロファイルインフラストラクチャの違い

残領域	古いプロファイル	新しいプロファイル
プロファイルに含まれる暗号と ECC カーブ	いいえ	はい
既存のリストの途中での暗号または暗号グループの挿入	すべての暗号をアンバインドし、必要な優先順位の順序で再度バインドします。	暗号を追加し、優先順位を割り当てます。プライオリティが指定されていない場合、暗号にはリスト内で最も低いプライオリティが割り当てられます。

残領域	古いプロファイル	新しいプロファイル
すべての暗号のバインド解除	`unbind ssl vserver \<name\> ciphername -ALL`	`unbind ssl profile -cipherName FlushAllCiphers`（リリース 11.0 ビルド 64.x 以降には、ALL が暗号グループのように扱われるため、プロファイルからすべての暗号または暗号グループのバインドを解除するための FlushAllCiphers パラメーターが含まれています。）
SSLv3 の状態	なし	デフォルトのフロントエンド・プロファイル（ns_default_ssl_ プロファイル _ フロントエンド）では無効になっています。注：このプロファイルを有効にする前に、SSLv3 がグローバルに有効になります。プロファイルを有効にすると、フロントエンドのデフォルトプロファイルで SSLv3 が無効になります。

SSL プロファイルインフラストラクチャ

October 7, 2021

SSLv3 および RC4 の実装の脆弱性により、ネットワーク接続のセキュリティ設定をネゴシエートするために最新の暗号とプロトコルを使用する必要性が強調されています。数千の SSL エンドポイントで SSLv3 を無効にするなど、設定への変更を実装するのは面倒なプロセスです。したがって、SSL エンドポイント設定の一部であった設定は、デフォルトの暗号とともに SSL プロファイルに移動されました。暗号サポートを含む設定の変更を実装するには、エンティティにバインドされているプロファイルを変更するだけで済みます。

デフォルトのフロントエンドおよびデフォルトのバックエンド SSL プロファイルには、古いプロファイルの一部であった設定に加えて、すべてのデフォルトの暗号と ECC 曲線が含まれています。デフォルトプロファイルの出力例は、付録に記載されています。[Enable Default Profile] 操作では、既定のフロントエンドプロファイルがすべてのフロントエンドエンティティに自動的にバインドされ、既定のバックエンドプロファイルがすべてのバックエンドエンティティにバインドされます。デフォルトのプロファイルは、デプロイメントに合わせて変更できます。カスタムプロファイルを作成し、SSL エンティティにバインドすることもできます。

フロントエンドプロファイルには、フロントエンドエンティティに適用可能なパラメータが含まれています。つまり、クライアントからの要求を受信するエンティティに適用されます。通常、このエンティティは、Citrix ADC アプライアンス上の SSL 仮想サーバーまたは透過 SSL サービスです。バックエンドプロファイルには、バックエンドエンティティに適用可能なパラメータが含まれています。つまり、クライアント要求をバックエンドサーバーに送信する ADC アプライアンス上のエンティティに適用されます。通常、このエンティティは Citrix ADC アプライアンスの SSL サービスです。サポートされていないパラメータを設定しようとすると、エラー `ERROR: Specified parameters are not applicable for this type of SSL profile` が表示されます。

> 重要:
>
> - SSL プロファイルは SSL パラメーターよりも優先されます。つまり、`set ssl parameter` コマンドを使用して SSL パラメータを設定し、後でプロファイルを SSL エンティティにバインドすると、プロファイルの設定が優先されます。
> - アップグレード後、デフォルトプロファイルを有効にすると、変更を元に戻すことはできません。つまり、プロファイルを無効にすることはできません。プロファイルを有効にする前に、構成を保存し、構成ファイル（ns.conf）のコピーを作成します。ただし、デフォルトプロファイルの機能を使用しない場合は、引き続き古い SSL プロファイルを使用できます。これらのプロファイルの詳細については、レガシー SSL プロファイルを参照してください。
> - リリース 11.1 51.x 以降、GUI および CLI で、デフォルトプロファイルを有効にすると、誤って有効にならないように確認プロンプトが追加されます。

コマンド:

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

デフォルトでは、グローバルパラメータと呼ばれる一部の *SSL* パラメータは、すべての SSL エンドポイントに適用されます。ただし、プロファイルが SSL エンドポイントにバインドされている場合、グローバルパラメータは適用されません。プロファイルに指定された設定が代わりに適用されます。

注意事項

1. 1 つのプロファイルを複数の仮想サーバにバインドできますが、1 つの仮想サーバにバインドできるプロファイルは 1 つだけです。
2. 仮想サーバーにバインドされているプロファイルを削除するには、まずプロファイルのバインドを解除します。
3. 暗号グループまたは暗号グループは、異なる優先順位で複数のプロファイルにバインドできます。

4. プロファイルは、異なる優先順位でバインドされた複数の暗号および暗号グループを持つことができます。
5. 暗号グループへの変更は、すべてのプロファイルとプロファイルのいずれかがバインドされているすべての仮想サーバーに即座に反映されます。
6. 暗号スイートが暗号グループの一部である場合、プロファイルから暗号スイートを削除する前に、暗号グループを編集してその暗号スイートを削除します。
7. プロファイルにアタッチされている暗号スイートまたは暗号グループにプライオリティを割り当てない場合は、プロファイル内で最も低いプライオリティが割り当てられます。
8. 既存の暗号グループおよび暗号スイートからカスタム暗号グループ（ユーザ定義暗号グループとも呼ばれる）を作成できます。暗号グループ A を作成し、既存の暗号グループ X と Y をこの順序で追加すると、Y は X よりも低い優先順位で割り当てられます。つまり、最初に追加されるグループの優先順位が高くなります。
9. 暗号スイートが同じプロファイルに接続された 2 つの暗号グループの一部である場合、暗号スイートは 2 番目の暗号グループの一部として追加されません。トラフィックが処理されると、プライオリティの高い暗号スイートが有効になります。
10. 暗号グループは、プロファイル内で展開されません。その結果、構成ファイル (ns.conf) の行数が大幅に削減されます。たとえば、それぞれ 15 個の暗号を含む 2 つの暗号グループが 1000 個の SSL 仮想サーバにバインドされている場合、拡張は設定ファイルに 30*1000 個の暗号関連のエントリを追加します。新しいプロファイルでは、プロファイルにバインドされた暗号グループごとに 1 つずつ、2 つのエントリしかありません。
11. 既存の暗号および暗号グループからユーザー定義暗号グループを作成することは、コピー＆ペースト操作です。元のグループの変更は、新しいグループに反映されません。
12. ユーザ定義の暗号グループには、それが属するすべてのプロファイルがリストされます。
13. プロファイルには、バインドされているすべての SSL 仮想サーバ、サービス、およびサービスグループが一覧表示されます。
14. デフォルトの SSL プロファイル機能が有効になっている場合は、プロファイルを使用して SSL エンティティの属性を設定または変更します。たとえば、仮想サーバー、サービス、サービスグループ、または内部サービスです。

CLI を使用して設定を保存する

コマンドプロンプトで入力します。

```
save config

shell

root@ns# cd /nsconfig

root@ns# cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber
    >
<!--NeedCopy-->
```


例:

```
save config
shell
root@ns# cd /nsconfig
root@ns# cp ns.conf ns.conf.NS.11.0.jun.16
<!--NeedCopy-->
```

既定のプロファイルを有効にする

> 重要:
>
> ソフトウェアをアップグレードし、デフォルトプロファイルを有効にする前に、設定を保存します。
>
> リリース 11.1 ビルド 51.x から、GUI および CLI で、デフォルトプロファイルを有効にすると、誤って有効にしないように確認プロンプトが表示されます。

コマンド: 次のコマンドは、デフォルトのプロファイルを有効にし、プロファイルが既にバインドされている SSL エンティティにこのプロファイルをバインドします。つまり、プロファイル（たとえば、P1）がすでに SSL エンティティにバインドされている場合、デフォルトのフロントエンドプロファイルまたはデフォルトのバックエンドプロファイルが P1 を置き換えます。古いプロファイル (P1) は削除されません。これで、拡張 SSL プロファイルになり、以前の設定、および暗号と ECC カーブが含まれています。デフォルトプロファイルを使用しない場合は、P1 を SSL エンティティに明示的にバインドできます。

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

拡張プロファイルインフラストラクチャをサポートするビルドにソフトウェアをアップグレードし、既定のプロファイルを有効にします。

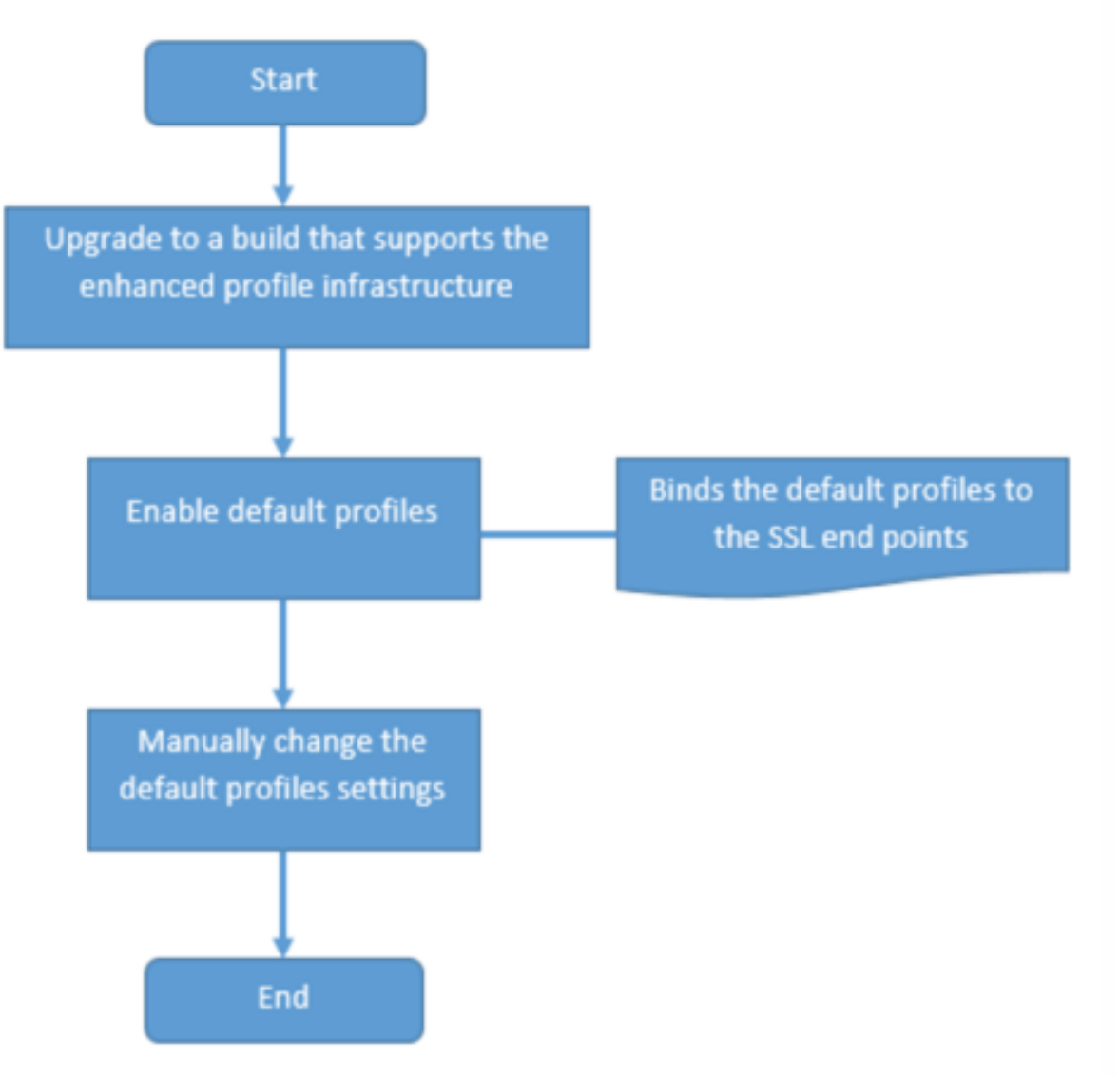
> メモ:
>
> - レガシープロファイル（P1）がすでに SSL エンティティにバインドされていて、デフォルトプロファイルを有効にすると、デフォルトプロファイルによって以前のバインディングが上書きされます。つまり、デフォルトのプロファイルは SSL エンティティにバインドされます。デフォルトプロファイルをバインドしない場合は、P1 を SSL エンティティに再度バインドする必要があります。
> - 1 回の操作 ([Enable Default Profile] または`set ssl parameter -defaultProfile`

> ENABLED) で、既定のフロントエンドプロファイルと既定のバックエンドプロファイルの両方が有効になります (バインド)。

使用例

デフォルトプロファイルを有効にすると、すべての SSL エンドポイントにバインドされます。既定のプロファイルは編集可能です。展開でほとんどのデフォルト設定が使用され、少数のパラメータしか変更されない場合は、デフォルトのプロファイルを編集できます。変更内容は、すべてのエンドポイントに即座に反映されます。カスタムパラメータとデフォルトパラメータを含むカスタム SSL プロファイルを作成し、SSL エンティティにバインドすることもできます。

次のフローチャートでは、実行する必要がある手順について説明します。



1. ソフトウェアのアップグレードの詳細については、システムソフトウェアのアップグレードを参照してくださ

い。

2. CLI または GUI を使用して、デフォルトプロファイルを有効にします。

- コマンドラインで、次を入力します：`set ssl parameter -defaultProfile ENABLED`。
- GUI を使用する場合は、[トラフィック管理] > [**SSL] > [**高度な **SSL** 設定の変更] に移動し、下にスクロールして [デフォルトプロファイルの有効化] を選択します。

アップグレード前にプロファイルがエンドポイントにバインドされていない場合、デフォルトプロファイルが SSL エンドポイントにバインドされます。アップグレード前にプロファイルがエンドポイントにバインドされている場合は、アップグレード後に同じプロファイルがバインドされ、デフォルトの暗号がプロファイルに追加されます。

1.（オプション）デフォルトプロファイルの設定を手動で変更します。

- コマンドラインで、変更するパラメータに続けて`set ssl profile <name>`を入力します。
- GUI を使用する場合は、[システム] > [プロファイル] に移動します。「**SSL** プロファイル」でプロファイルを選択し、「編集」をクリックします。

SSL プロファイルパラメータ

SSL プロファイルでは、次の SSL パラメータを設定できます。これらのパラメータの一部は、SSL 仮想サーバーで設定できます。SSL 仮想サーバーパラメーターの詳細については、「SSL 仮想サーバーパラメータ」を参照してください。

Citrix ADC アプライアンスのバックエンドでの安全な再ネゴシエーションのサポート

注：このパラメーターは、リリース 13.0 ビルド 58.x 以降で導入されました。以前のリリースとビルドでは、非セキュアな再ネゴシエーションのみがバックエンドでサポートされていました。

これらのプラットフォームでは、次の 3 つの LOM 機能がサポートされています。

- VPX
- N2 または N3 チップを含む MPX プラットフォーム
- インテル Coleto SSL チップ・ベースのプラットフォーム

この機能は FIPS プラットフォームではまだサポートされていません。

ADC アプライアンスのバックエンドでは、デフォルトで安全な再ネゴシエーションが拒否されます。つまり、denySSLReneg パラメーターは ALL（デフォルト）に設定されます。

バックエンドで安全な再ネゴシエーションを許可するには、denySSLReneg パラメータの次の設定のいずれかを選択します。

- いいえ
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

CLI を使用して安全な再ネゴシエーションを有効にする

コマンドプロンプトで入力します。

```
set ssl profile <name> -denySSLReneg <denySSLReneg>
```

例:

```
set ssl profile ns_default_ssl_profile_backend -denySSLReneg NONSECURE
 Done

sh ssl profile ns_default_ssl_profile_backend
1)  Name: ns_default_ssl_profile_backend    (Back-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Server Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:       NO
    Session Reuse: ENABLED      Timeout: 300 seconds
    DH: DISABLED
    Ephemeral RSA: DISABLED
    Deny SSL Renegotiation      NONSECURE
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:      NO
    Push flag:  0x0 (Auto)
    SSL quantum size:       8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45

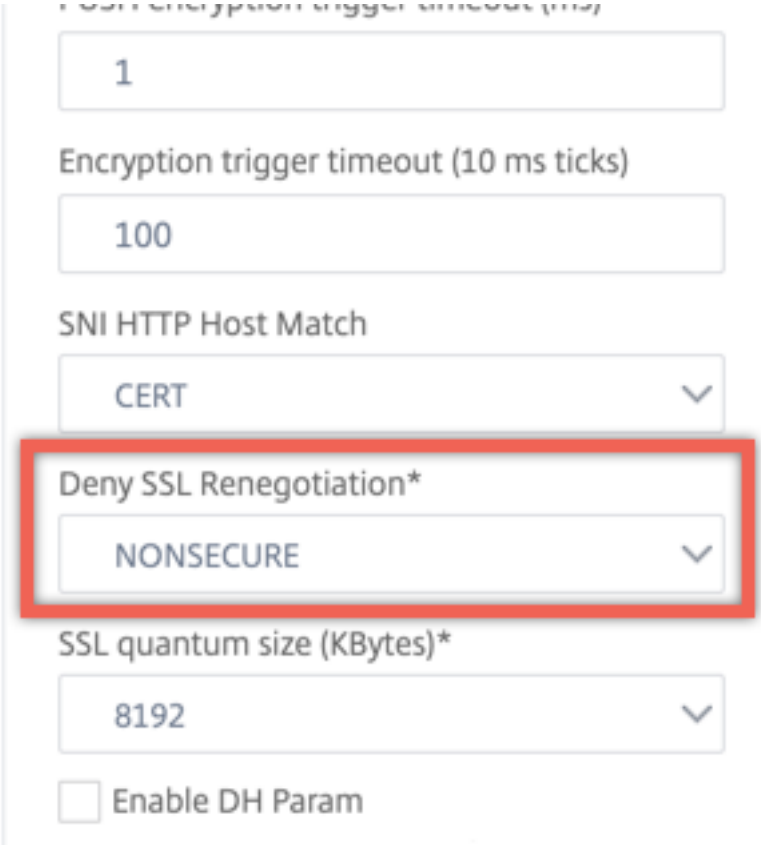
    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT_BACKEND        Priority :2
    Description: Predefined Cipher Alias

1)  Service Name: s187
 Done
<!--NeedCopy-->
```


GUI を使用して安全な再ネゴシエーションを有効にする

1. **[**システム**] > [**プロファイル**] > [SSL** プロファイル**]** に移動します。
2. プロファイルを追加または編集します。
3. [**SSL** 再ネゴシエーションの拒否**]** を ALL 以外の任意の値に設定します。



ホストヘッダーの検証

注：このパラメーターは、リリース 13.0 ビルド 52.x で導入されました。

と HTTP/1.1, クライアントは、複数の要求を処理するために複数の接続を使用する必要がありました。と HTTP/2, クライアントは、同じ証明書でカバーされているドメイン間で接続を再利用できます。SNI 対応セッションの場合、ADC アプライアンスは、この変更に対応するために HTTP ホストヘッダーを検証する方法を制御できる必要があります。以前のビルドでは、パラメーターが有効になっていて（「はい」に設定されている）、要求に SNI が有効なセッションのホストヘッダーが含まれていない場合、要求はドロップされました。パラメータが無効になっている（「いいえ」に設定されている）場合、アプライアンスは検証を実行しませんでした。新しいパラメータ `SNIHTTPHostMatch` が SSL プロファイルと SSL グローバルパラメータに追加され、この検証をより適切に制御できるようになりました。

このパラメーターは 3 つの値を取ることができます。CERT、STRICT、および NONE。これらの値は、SNI 対応セッションでのみ次のように機能します。SNI は、SSL 仮想サーバーまたは仮想サーバーにバインドされたプロファイルで有効にする必要があり、HTTP 要求にはホストヘッダーが含まれている必要があります。

- CERT- 要求のホストヘッダー値がこの SSL セッションの確立に使用される証明書でカバーされている場合、接続が転送されます。
- STRICT-接続は、リクエストのホストヘッダー値が SSL 接続の ClientHello メッセージで渡されたサーバー名の値と一致する場合にのみ転送されます。
- NO-ホストヘッダー値は検証されません。

可能な値：NO、CERT、STRICT
デフォルト値：CERT

新しいパラメーターSNIHTTPHostMatchの導入により、dropReqWithNoHostHeader パラメーターの動作が変更されます。dropReqWithNoHostHeaderパラメータの設定は、ホストヘッダーが SNI 証明書に対して検証される方法に影響しなくなりました。

CLI を使用した SSL プロファイルパラメータの設定

コマンドプロンプトで入力します。

```
set ssl profile <name> [-ssllogProfile <string>] [-dh ( ENABLED |
    DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-
    dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED |
    DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED |
    DISABLED )
[-sessTimeout <positive_integer>]] [-cipherRedirect ( ENABLED |
    DISABLED ) [-cipherURL <URL>]] [-clientAuth ( ENABLED | DISABLED )[-
    clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
DISABLED )] [-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl3 (
    ENABLED | DISABLED )] [-tls1 ( ENABLED | DISABLED )] [-tls11 (
    ENABLED| DISABLED )] [-tls12 ( ENABLED | DISABLED )] [-tls13 (
    ENABLED |DISABLED )] [-SNIEnable ( ENABLED | DISABLED )] [-
    ocspStapling (ENABLED | DISABLED )] [-serverAuth ( ENABLED |
    DISABLED )] [-commonName <string>] [-pushEncTrigger <pushEncTrigger
    >] [-sendCloseNotify ( YES |
NO )] [-clearTextPort <port|*>] [-insertionEncoding ( Unicode | UTF-8)]
     [-denySSLReneg <denySSLReneg>] [-quantumSize <quantumSize>]
[-strictCAChecks ( YES | NO )] [-encryptTriggerPktCount <
    positive_integer>] [-pushFlag <positive_integer>][-
    dropReqWithNoHostHeader ( YES | NO )] [-SNIHTTPHostMatch <
    SNIHTTPHostMatch>] [-pushEncTriggerTimeout <positive_integer>]
```

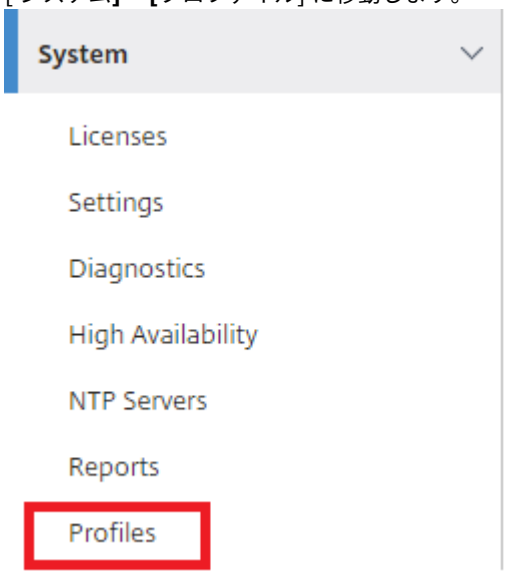


```
6  [-sslTriggerTimeout <positive_integer>] [-clientAuthUseBoundCAChain (
       ENABLED | DISABLED )] [-sslInterception ( ENABLED | DISABLED )][-
       ssliReneg ( ENABLED | DISABLED )] [-ssliOCSPCheck ( ENABLED |
       DISABLED )] [-ssliMaxSessPerServer <positive_integer>] [-HSTS (
       ENABLED| DISABLED )] [-maxage <positive_integer>] [-
       IncludeSubdomains ( YES | NO )] [-preload ( YES | NO )] [-
       sessionTicket ( ENABLED | DISABLED )][-sessionTicketLifeTime <
       positive_integer>] [-sessionTicketKeyRefresh (ENABLED | DISABLED )]
       {
7   -sessionTicketKeyData  }
8    [-sessionKeyLifeTime <positive_integer>] [-prevSessionKeyLifeTime <
         positive_integer>]
9  [-cipherName <string> -cipherPriority <positive_integer>][-
       strictSigDigestCheck ( ENABLED | DISABLED )]
10 [-skipClientCertPolicyCheck ( ENABLED | DISABLED )] [-zeroRttEarlyData
       ( ENABLED | DISABLED )] [-tls13SessionTicketsPerAuthContext
11 <positive_integer>] [-dheKeyExchangeWithPsk ( YES | NO )]
12 <!--NeedCopy-->
```

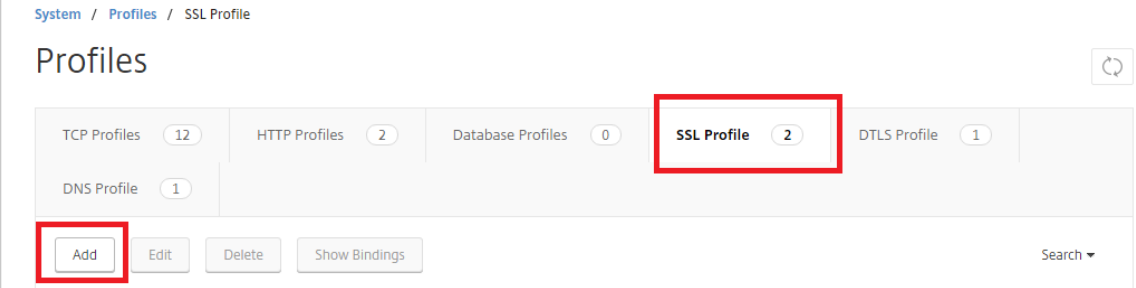
GUI を使用して SSL プロファイルパラメータを設定する

プロファイルを追加するには:

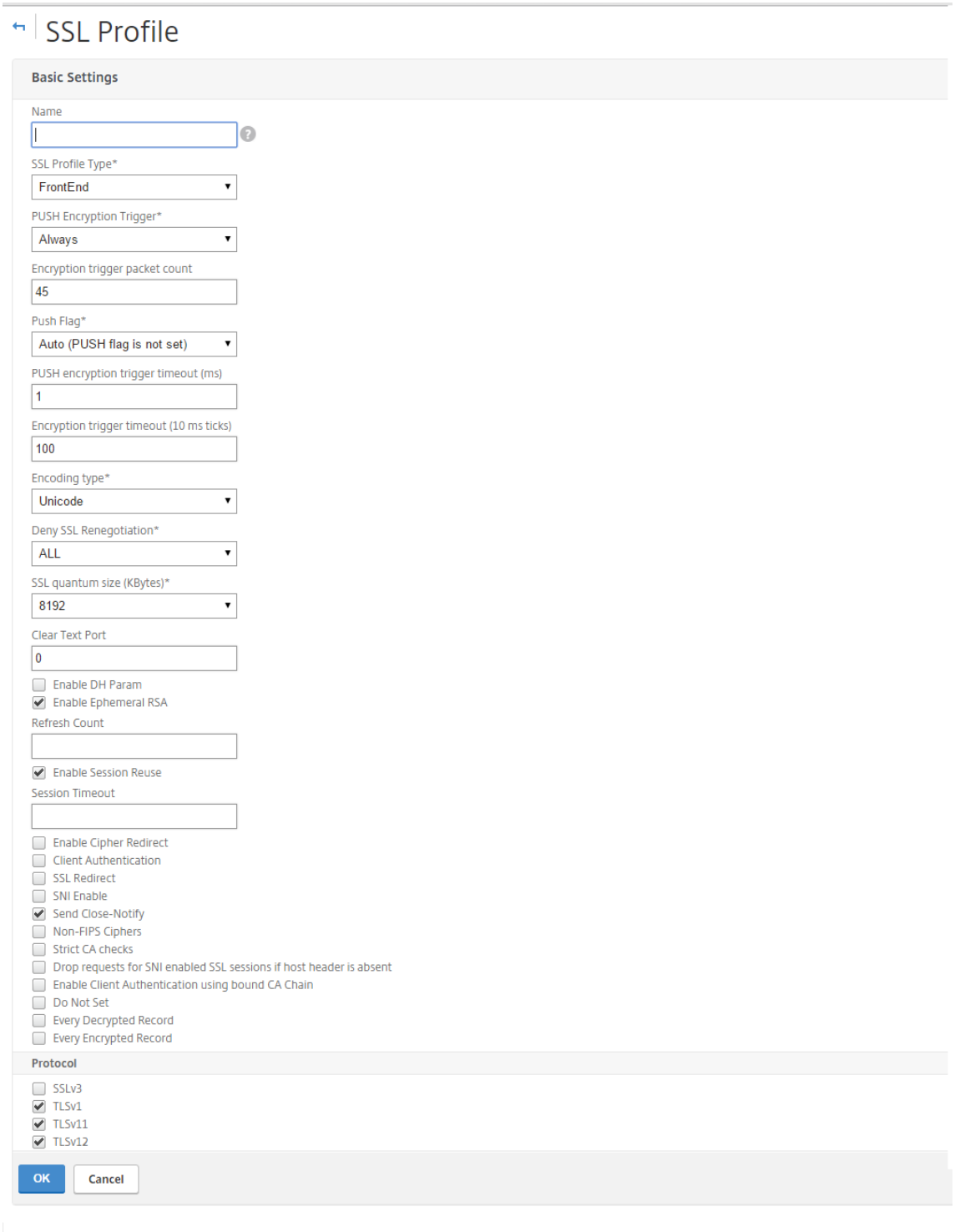
1. [システム] > [プロファイル] に移動します。



2. [**SSL** プロファイル] を選択します。[追加] をクリックします。



3. さまざまなパラメーターの値を指定します。



4. [**OK**] をクリックします。
5. [完了] をクリックします。

既存の SSL プロファイルを再利用するには:

1. [システム**] > [**プロファイル] に移動します。
2. 既存のプロファイルを選択し、[追加] をクリックします。

3. 別の名前を指定し、パラメータを変更して、[**OK**] をクリックします。
4. [完了] をクリックします。

TLS セッション・チケット拡張

SSL ハンドシェイクは CPU を大量に消費する操作です。セッションの再利用が有効な場合、既存のクライアントに対するサーバー/クライアントのキー交換操作はスキップされます。彼らはセッションを再開することができます。このアクションにより、応答時間が向上し、サーバーがサポートできる 1 秒あたりの SSL トランザクション数が増加します。ただし、サーバーは各セッション状態の詳細を保存する必要があり、メモリを消費し、要求がサーバー間で負荷分散される場合、複数のサーバー間で共有することは困難です。

Citrix ADC アプライアンスは、セッションチケット TLS 拡張をサポートしています。この拡張機能を使用すると、セッションの詳細がサーバーではなくクライアントに格納されます。クライアントは、クライアント Hello メッセージにセッションチケット TLS 拡張を含めることによって、このメカニズムをサポートしていることを示す必要があります。新しいクライアントの場合、この拡張機能は空です。サーバーは、NewSessionTicket ハンドシェイクメッセージで新しいセッションチケットを送信します。セッションチケットは、サーバーのみが認識するキーペアを使用して暗号化されます。サーバーが新しいチケットを発行できない場合は、通常のハンドシェイクを完了します。

この機能は、フロントエンド SSL プロファイルでのみ使用でき、アプライアンスがサーバーとして機能し、セッション・チケットを生成する通信のフロントエンドでのみ使用できます。

制限事項

- この機能は FIPS プラットフォームではサポートされていません。
- この機能は、TLS バージョン 1.1 および 1.2 でのみサポートされます。
- SSL セッション ID の永続性は、セッションチケットではサポートされていません。

CLI を使用して **TLS** セッションチケット拡張を有効にする

コマンドプロンプトで入力します。

```
set ssl profile <name> -sessionTicket (ENABLED | DISABLED ) [-
    sessionTicketLifeTime <positive_integer>
<!--NeedCopy-->
```

引数**:**

sessionTicket: TLS セッションチケット拡張の状態。この拡張機能を使用すると、RFC 5077 で定義されているように、セッションの詳細がサーバーではなくクライアントに格納されます。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

sessionTicketLifeTime：セッションチケットの有効期限が切れ、新しい SSL ハンドシェイクを開始する必要があるまでの時間を秒単位で指定します。

デフォルト値:300

最小値:0

最大値:172800

例**:**

```
add ssl profile profile1 -sessionTicket ENABLED -sessionTicketlifeTime
    300
Done
<!--NeedCopy-->
```

GUI を使用して **TLS** セッションチケット拡張を有効にする

1. **System** > **Profiles** に移動します。[**SSL** プロファイル] を選択します。
2. [追加] をクリックし、プロファイルの名前を指定します。
3.「セッション・チケット」を選択します。
4. オプションで、セッション・チケットの有効期間 **(**秒**)** を指定します。

セッション・チケットの安全な実装

TLS セッションチケットを使用すると、クライアントは省略されたハンドシェイクを使用して、サーバーへの再接続を高速化できます。ただし、セッションチケットが暗号化または長期間変更されないと、セキュリティリスクが生じる可能性があります。セッション・チケットは、対称鍵で暗号化することで保護できます。転送秘密を実現するために、セッションチケットキーがリフレッシュされる時間間隔を指定できます。

アプライアンスは、デフォルトでセッション・チケット・キーを生成します。ただし、デプロイメント内の複数のアプライアンスが互いのセッションチケットを復号化する必要がある場合は、すべて同じセッションチケットキーを使用する必要があります。したがって、すべてのアプライアンスで同じセッションチケットキーデータを手動で設定（追加またはロード）する必要があります。セッション・チケット・キー・データには、次の情報が含まれます。

- セッション・チケット名。
- チケットの暗号化または復号化に使用されるセッション AES キー。
- チケットのダイジェストを計算するために使用されるセッション HMAC キー。

RFC 5077 で推奨されているように 256 ビットの HMAC キーをサポートするために、長さ 64 バイトのセッションチケットキーデータを構成できるようになりました。下位互換性のために 48 バイトのキー長もサポートされています。

> 注:
>
> セッションチケットキーデータを手動で入力するときは、HA セットアップまたはクラスタセットアップのすべての Citrix ADC アプライアンスの構成が同じであることを確認してください。

sessionTicketKeyLifeTime パラメータは、セッションチケットキーをリフレッシュする頻度を指定します。prevSessionTicketKeyLifeTime パラメータを設定して、新しいキーが生成された後、そのキーを使用してチケットを復号化するために以前のセッションチケットキーを維持する期間を指定できます。prevSessionTicketKeyLifeTime この設定により、クライアントが省略されたハンドシェイクを使用して再接続できる時間が延長されます。たとえば、sessionTicketKeyLifeTimeが 10 分、prevSessionTicketKeyLifeTimeが 5 分に設定されている場合、10 分後に新しいキーが生成され、すべての新しいセッションで使用されます。ただし、以前に接続したクライアントにはさらに 5 分間あり、以前に発行されたチケットは省略されたハンドシェイクに対して優先されます。

CLI を使用した **SSL** セッションチケットデータの設定

コマンドプロンプトで入力します。

```
set ssl profile <name> -sessionTicket ENABLED -sessionTicketLifeTime <
    positive_integer> -sessionTicketKeyRefresh ( ENABLED | DISABLED )] -
    sessionTicketKeyLifeTime <positive_integer> [-
    prevSessionTicketKeyLifeTime <positive_integer>]
<!--NeedCopy-->
```

引数**:**

sessionTicket: RFC 5077 の記述に従ってセッションチケットを使用します。初期ハンドシェイクを確立するには、CPU を大量に消費する公開キー暗号化操作が必要です。**ENABLED** 設定では、サーバーはクライアントにセッションチケットを発行し、クライアントは省略されたハンドシェイクを実行するために使用できます。

設定可能な値: ENABLED, DISABLED。デフォルト: DISABLED

sessionTicketLifeTime: セッションチケットの有効期間（秒単位）。この時間が経過すると、クライアントはこのチケットを使用してセッションを再開できません。

最大値:172800 です。最小値: 0。デフォルトは 300 です。

sessionTicketKeyRefresh: セッションチケットキーの有効期間パラメータで指定された時間が経過すると、セッションチケットの暗号化または復号化に使用されるセッションチケットキーを再生成します。セッションチケットが有効になっている場合、自動的に有効になります。管理者がセッション・チケット・データを入力すると無効になります。

設定可能な値: ENABLED, DISABLED。デフォルト:ENABLED

sessionKeyLifeTime: Citrix ADC アプライアンスによって発行されたセッションチケットの暗号化に使用される対称キーの有効期間（秒単位）。

最大値:86400。最小値:600。デフォルト: 3000

prevSessionKeyLifeTime: セッションチケットキーの有効期間が終了した後も、セッションチケットの暗号化に使用された以前の対称キーが、既存のクライアントに対して有効なままになる時間（秒単位）。この時間内に、既存のクライアントは、以前のセッションチケットキーを使用してセッションを再開できます。新しいクライアントのセッションチケットは、新しいキーを使用して暗号化されます。

最大値:172800 です。最小値: 0。デフォルト:0

例**:**

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED
    -sessionTicketlifeTime 120 -sessionTicketKeyRefresh ENABLED -
    sessionTicketKeyLifeTime 100 -prevSessionTicketKeyLifeTime 60

Done

show ssl profile ns_default_ssl_profile_frontend

     Session Ticket: ENABLED
     Session Ticket Lifetime: 120 (secs)
     Session Key Auto Refresh: ENABLED
     Session Key Lifetime: 100 (secs)
     Previous Session Key Lifetime: 60 (secs)
<!--NeedCopy-->
```

GUI を使用して **SSL** セッションチケットデータを構成する

1. [システム] > [プロファイル] に移動し、[**SSL** プロファイル] を選択します。
2. 「**ns_default_ssl_** プロファイル **_** フロントエンド」を選択し、「編集」をクリックします。
3. [基本設定**]** セクションで、鉛筆アイコンをクリックし、次のパラメータを設定します。
   - セッションチケット
   - セッションチケットの有効期間 (秒)
   - セッションチケットキーの自動更新
   - セッションチケットキーの有効期間 (秒)
   - 前のセッションチケットキーの有効期間 (秒)
4. [**OK**] をクリックします。

CLI を使用して **SSL** セッションチケットデータを手動で入力します

コマンドプロンプトで入力します。

```
set ssl profile <name> -sessionTicket ENABLED

set ssl profile <name> -sessionTicketKeyData

show ssl profile ns_default_ssl_profile_frontend
<!--NeedCopy-->
```

引数**:**

sessionTicket: RFC 5077 によって記述されたセッションチケットの使用。初期ハンドシェイクを確立するには、CPU を大量に消費する公開キー暗号化操作が必要です。**ENABLED** 設定では、サーバーはクライアントにセッションチケットを発行し、クライアントは省略されたハンドシェイクを実行するために使用できます。

設定可能な値: ENABLED, DISABLED。デフォルト: DISABLED

sessionTicketKeyData: Contains the session ticket name (0-15 bytes), the session AES key used to encrypt or decrypt the session ticket (16-31 bytes), and the session HMAC key used to compute the digest of the ticket (32-63 bytes). Externally generated by an administrator and added to a Citrix ADC appliance.

最大長:64 バイト

例**:**

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED

Done

set ssl profile ns_default_ssl_profile_frontend -sessionTicketKeyData
    11111111111111111111111111111111111111111111111111

Done

show ssl profile ns_default_ssl_profile_frontend

     1) Name: ns_default_ssl_profile_frontend (Front-End)
     SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1.2: ENABLED
     Client Auth: DISABLED
     Use only bound CA certificates: DISABLED
```



```
15      Strict CA checks: NO
16      Session Reuse: ENABLED Timeout: 120 seconds
17      DH: DISABLED
18      DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA: ENABLED
            Refresh Count: 0
19      Deny SSL Renegotiation ALL
20      Non FIPS Ciphers: DISABLED
21      Cipher Redirect: DISABLED
22      SSL Redirect: DISABLED
23      Send Close-Notify: YES
24      Push Encryption Trigger: Always
25      PUSH encryption trigger timeout: 1 ms
26      SNI: DISABLED
27      OCSP Stapling: DISABLED
28      Strict Host Header check for SNI enabled SSL sessions: NO
29      Push flag: 0x0 (Auto)
30      SSL quantum size: 8 kB
31      Encryption trigger timeout 100 mS
32      Encryption trigger packet count: 45
33      Subject/Issuer Name Insertion Format: Unicode
34      Session Ticket: ENABLED
35      Session Ticket Lifetime: 300 (secs)
36      Session Key Auto Refresh: DISABLED
37      Session Key Lifetime: 3000 (secs)
38      Previous Session Key Lifetime: 0 (secs)
39      Session Key Data: 84
           dad1afc6d56b0deeb0a7fd7f299a207e8d8c15cdd087a5684a11a329fd732e87a0535d9088

40      47
           e8c181ba266f5c8838ae472cb3ab9255b683bf922fad32cee816c329989ef7cdeb278e93ac

41
42      ECC Curve: P_256, P_384, P_224, P_521
43
44      1) Cipher Name: DEFAULT Priority :4
45      Description: Predefined Cipher Alias
46
47      1) Internal Service Name (Front-End): nsrnatsip-127.0.0.1-5061
48      2) Internal Service Name (Front-End): nskrpcs-127.0.0.1-3009
49      3) Internal Service Name (Front-End): nshttps-::1l-443
50      4) Internal Service Name (Front-End): nsrpcs-::1l-3008
51      5) Internal Service Name (Front-End): nshttps-127.0.0.1-443
52      6) Internal Service Name (Front-End): nsrpcs-127.0.0.1-3008
53      7) Vserver Name: v1
54
```



```
55 Done
56 <!--NeedCopy-->
```

GUI を使用して **SSL** セッションチケットデータを手動で入力します

1. **System > Profiles** に移動して **SSL Profile** を選択します。
2. 「**ns_default_ssl_** プロファイル _ フロントエンド」を選択し、「編集」をクリックします。
3. [基本設定] セクションで、鉛筆アイコンをクリックし、次のパラメータを設定します。
   - セッションチケット
   - セッション・チケット・キー・データ
   - セッション・チケット・キー・データの確認
4. [**OK**] をクリックします。

Citrix ADC 非 **FIPS** プラットフォームでの **SSL** ハンドシェイクでの拡張マスターシークレットのサポート

注：このパラメーターは、リリース 13.0 ビルド 61.x で導入されました。

拡張マスターシークレット（EMS）は、トランスポート層セキュリティ（TLS）プロトコルのオプションの拡張機能です。Citrix ADC アプライアンスで EMS をサポートするために、フロントエンドとバックエンドの両方の SSL プロファイルに適用される新しいパラメーターが追加されました。パラメータが有効で、ピアが EMS をサポートしている場合、ADC アプライアンスは EMS 計算を使用します。ピアが EMS をサポートしていない場合、アプライアンスでパラメータが有効になっていても、EMS 計算は接続に使用されません。EMS について詳しくは、RFC 7627 を参照してください。

注： EMS は、TLS プロトコルバージョン 1.0、1.1、または 1.2 を使用するハンドシェイクにのみ適用できます。

EMS のプラットフォームサポート

- CaviumN3 チップまたは IntelColetoCreek 暗号カードのいずれかを含む MPX および SDX プラットフォーム。次のプラットフォームには、Intel Coleto チップが同梱されています。
- MPX 5900
- MPX/SDX 8900
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPS/SDX 26000-100G
- MPX/SDX 15000-50G

「showhardware」コマンドを使用して、アプライアンスに Coleto（COL）または N3 チップがあるかどうかを識別することもできます。

- 暗号カードのない MPX および SDX プラットフォーム（ソフトウェアのみ）。
- ソフトウェアのみのプラットフォーム：VPX、CPX、および BLX。

次のプラットフォームでは EMS を有効にできません。

- MPX 9700FIPS および MPX14000FIPS プラットフォーム。
- CaviumN2 暗号チップを含む MPX および SDX プラットフォーム。

パラメータが有効になっている場合、ADC アプライアンスは TLS 1.2、TLS 1.1、および TLS 1.0 接続で EMS を使用しようとします。この設定は、TLS1.3 または SSLv3 接続には影響しません。

EMS がピアとネゴシエートできるようにするには、仮想サーバー（フロントエンド）またはサービス（バックエンド）にバインドされた SSL プロファイルの設定を有効にします。

CLI を使用して **EMS** を有効にする

コマンドプロンプトで入力します。

```
set ssl profile <profile name> [-allowExtendedMasterSecret (YES | NO)]
```

例

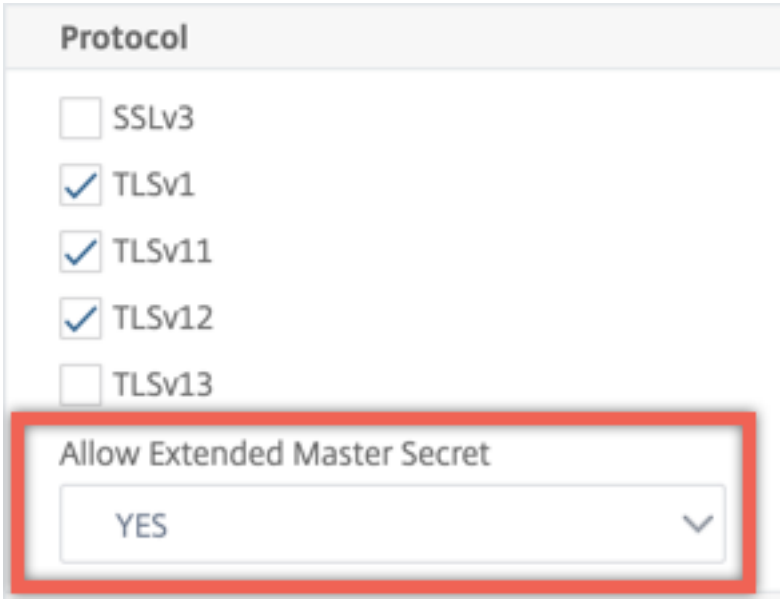
```
1 set ssl profile ns_default_ssl_profile_frontend  -
     allowExtendedMasterSecret YES
2
3 set ssl profile ns_default_ssl_profile_backend  -
     allowExtendedMasterSecret YES
4 <!--NeedCopy-->
```

次の表は、さまざまなデフォルトプロファイルとユーザー定義プロファイルの `allowExtendedMasterSecret` パラメーターのデフォルト値を示しています。

Profile	デフォルト設定
デフォルトのフロントエンドプロファイル	いいえ
デフォルトのフロントエンドセキュアプロファイル	はい
デフォルトのバックエンドプロファイル	いいえ
ユーザー定義のプロファイル	いいえ

GUI を使用して **EMS** を有効にする

1. **[**システム**] > [**プロファイル**] > [SSL** プロファイル**]** に移動します。
2. プロファイルを追加するか、プロファイルを編集します。
3. **[**拡張マスターシークレットを許可する**]** を [はい] に設定します。



クライアント **hello** メッセージでの **ALPN** 拡張の処理のサポート

注: この機能は、リリース 11.0 ビルド 64.x 以降でサポートされています。

パラメータ `alpnProtocol` がフロントエンド SSL プロファイルに追加され、ALPN 拡張機能で処理される接続のアプリケーションプロトコルをネゴシエートします。SSL_TCP 仮想サーバー。クライアント hello メッセージの ALPN 拡張で同じプロトコルが受信された場合、SSL プロファイルで指定されたプロトコルのみがネゴシエートされます。

注： `alpnProtocol`パラメーターはフロントエンド SSL プロファイルでのみサポートされ、によって処理される SSL 接続に適用されます。SSL_TCP タイプ仮想サーバー。

CLI を使用してフロントエンド **SSL** プロファイルにプロトコルを設定します

コマンドプロンプトで入力します。

```
set ssl profile ns_default_ssl_profile_frontend -alpnProtocol <protocol_name
>
```

`alpnProtocol`パラメータには 3 つの値を指定できます。最大長：4096 バイト。

- **NONE**：アプリケーションプロトコルネゴシエーションは行われません。この設定がデフォルトです。
- **HTTP1**： HTTP1 はアプリケーションプロトコルとしてネゴシエートできます。

- **HTTP2**: HTTP2 はアプリケーションプロトコルとしてネゴシエートできます。

例:

```
set ssl profile ns_default_ssl_profile_frontend -ALPNProtocol HTTP2
> sh ssl profile ns_default_ssl_profile_frontend
1)  Name: ns_default_ssl_profile_frontend   (Front-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED  Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED  Refresh Count: 0
    Deny SSL Renegotiation  ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:  NO
    Match HTTP Host header with SNI:    CERT
    Push flag:  0x0 (Auto)
    SSL quantum size:   8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45
    Subject/Issuer Name Insertion Format:   Unicode

    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server: 10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
```

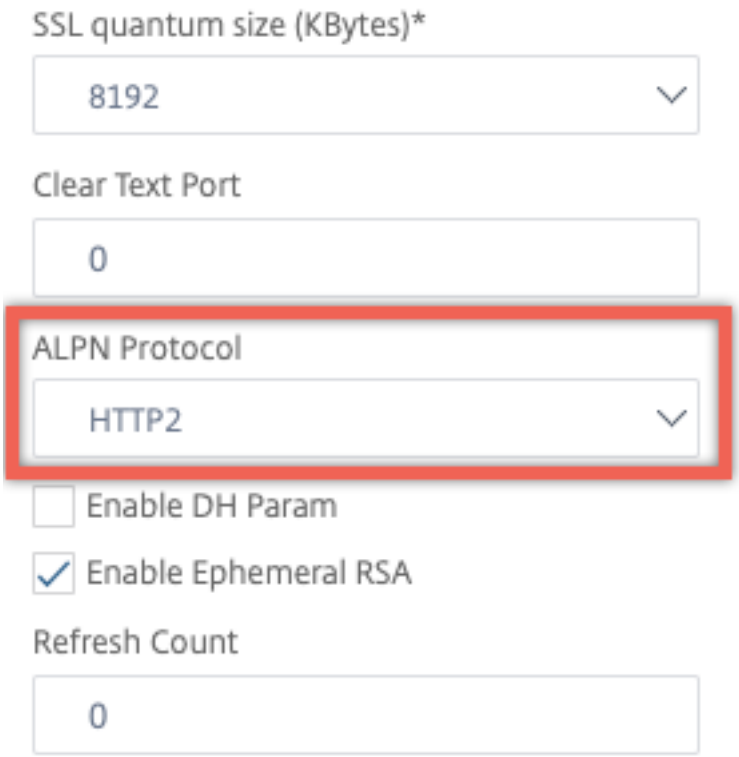


```
    HSTS Max-Age: 0
    HSTS Preload: NO
    Allow Extended Master Secret: NO
    Send ALPN Protocol: HTTP2

Done
<!--NeedCopy-->
```

GUI を使用してフロントエンド **SSL** プロファイルでプロトコルを設定します

1. **System > Profiles** に移動して **SSL Profile** を選択します。
2. 「**ns_default_ssl_** プロファイル _ フロントエンド」を選択し、「編集」をクリックします。
3. [**ALPN** プロトコル] リストで、[**HTTP2**] を選択します。



古い設定をロードする

デフォルトのプロファイルを有効にしても、元に戻すことはできません。ただし、展開で既定のプロファイルが必要でないと判断した場合は、既定のプロファイルを有効にする前に保存した古い設定を読み込むことができます。変更は、アプライアンスを再起動した後に有効になります。

CLI を使用して古い設定をロードする

コマンドプロンプトで入力します。

```
shell

root@ns# clear config

root@ns# cd /nsconfig

root@ns# cp ns.conf.NS.11.0.jun.16 ns.conf

root@ns# reboot
<!--NeedCopy-->
```

安全なフロントエンド・プロファイル

October 7, 2021

デフォルトのフロントエンドとデフォルトのバックエンド・プロファイルの他に、リリース 12.1 から新しいデフォルトのセキュア・フロントエンド・プロファイルを使用できます。Qualys SSL Labs の A+ レーティング（2018 年 5 月現在）に必要な設定は、このプロファイルにプリロードされています。以前は、SSL フロントエンドプロファイルまたは SSL 仮想サーバーの A+ 評価に必要な各パラメータを明示的に設定する必要がありました。これで、ns_default_ssl_profile_secure_frontend プロファイルを SSL 仮想サーバーにバインドでき、必要なパラメーターが SSL 仮想サーバーに自動的に設定されます。

> 注:
>
> セキュアフロントエンドプロファイルは編集できません。

既定のプロファイルを有効にすると、既定のフロントエンドプロファイルがすべての SSL 仮想サーバーに自動的にバインドされます。A+ 評価を取得するには、ns_default_ssl_profile_secure_ フロントエンドプロファイルを明示的にバインドし、SHA2/SHA256 サーバー証明書を SSL 仮想サーバーにバインドする必要があります。

セキュアなフロントエンド・プロファイル・パラメータ

デフォルト設定のパラメーターは次のとおりです。

```
SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2: ENABLED
      TLSv1.3: DISABLED
```



```
2
3  Deny SSL Renegotiation: NONSECURE
4
5  HSTS: ENABLED
6
7  HSTS IncludeSubDomains: YES
8
9  HSTS Max-Age: 15552000
10
11 Cipher Name: SECURE    Priority :1
12 <!--NeedCopy-->
```

安全な暗号エイリアス

新しいセキュア暗号エイリアスが追加され、セキュアフロントエンドプロファイルにバインドされます。このエイリアスの一部である暗号を一覧表示するには、コマンドプロンプトで「show cipher SECURE」と入力します。

```
1  show cipher SECURE
2
3      1) Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 1
4             Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256)
                  Mac=AEAD   HexCode=0xc030
5      2) Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 2
6             Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128)
                  Mac=AEAD   HexCode=0xc02f
7      3) Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
          Priority : 3
8             Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256)
                  Mac=AEAD   HexCode=0xc02c
9      4) Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
          Priority : 4
10            Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128)
                  Mac=AEAD   HexCode=0xc02b
11 Done
12 <!--NeedCopy-->
```

構成

次の手順を実行します。

1. SSL タイプの負荷分散仮想サーバーを追加します。
2. SHA2/SHA256 証明書をバインドします。

3. デフォルトプロファイルを有効にします。
4. セキュリティで保護されたフロントエンドプロファイルを SSL 仮想サーバーにバインドします。

CLI を使用して **SSL** 仮想サーバーの **A+** 評価を取得する

コマンドプロンプトで入力します。

```
add lb vserver <name> <serviceType> <IPAddress> <port>
bind ssl vserver <vServerName> -certkeyName <string>
set ssl parameter -defaultProfile ENABLED
set ssl vserver <vServerName> -sslProfile
   ns_default_ssl_profile_secure_frontend
show ssl vserver [<vServerName>]
<!--NeedCopy-->
```

例**:**

```
add lb vserver ssl-vsvr SSL 192.0.2.240 443

bind ssl vserver ssl-vsvr -certkeyName letrsa

set ssl parameter -defaultProfile ENABLED

Save your configuration before enabling the Default profile. You cannot
    undo the changes. Are you sure you want to enable the Default
   profile? [Y/N]y

set ssl vserver ssl-vsvr -sslProfile
   ns_default_ssl_profile_secure_frontend
<!--NeedCopy-->
```

```
sh ssl vserver ssl-vsvr

    Advanced SSL configuration for VServer ssl-vsvr:
    Profile Name :ns_default_ssl_profile_secure_frontend
    1) CertKey Name: letrsa     Server Certificate
Done
<!--NeedCopy-->
```



```
sh ssl profile ns_default_ssl_profile_secure_frontend

    1) Name: ns_default_ssl_profile_secure_frontend (Front-End)
    SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:                  NO
    Session Reuse: ENABLED   Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED     Refresh Count: 0
    Deny SSL Renegotiation                NONSECURE
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:  1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:
                           NO
    Push flag:              0x0 (Auto)
    SSL quantum size:                8 kB
    Encryption trigger timeout       100 mS
    Encryption trigger packet count: 45
    Subject/Issuer Name Insertion Format:  Unicode
    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:   10
    Session Ticket: DISABLED
    HSTS: ENABLED
    HSTS IncludeSubDomains: YES
    HSTS Max-Age: 15552000
    ECC Curve: P_256, P_384, P_224, P_521
    1) Cipher Name: SECURE    Priority :1
    Description: Predefined Cipher Alias
    1) Vserver Name: v2
```



```
42 Done
43 <!--NeedCopy-->
```

GUI を使用して **SSL** 仮想サーバーの **A+** 評価を取得する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して SSL 仮想サーバーを選択します。
2. Advanced Settings で SSL Profile をクリックします。
3. デフォルト _ssl_ プロファイル _ セキュリティ _ フロントエンドを選択します。
4. [OK] をクリックします。
5. [完了] をクリックします。

付録 **A:** アップグレード後の **SSL** 設定の移行例

October 7, 2021

注**:** 新しいデフォルトプロファイルの SSL 移行スクリプトはサポートされなくなったため、このコンテンツは削除されました。

付録 **B**: フロントエンドおよびバックエンド **SSL** プロファイルのデフォルト設定

October 7, 2021

既定のフロントエンドプロファイルには、次の設定があります。

```
1  sh ssl profile ns_default_ssl_profile_frontend
2
3  1)Name: ns_default_ssl_profile_frontend
4
5       Configuration for Front-End SSL profile
6       DH: DISABLED
7       Ephemeral RSA: ENABLED          Refresh Count: 0
8       Session Reuse: ENABLED          Timeout: 120 seconds
9       Non FIPS Ciphers: DISABLED
10      Cipher Redirect: ENABLED   Redirect URL: http://10.102.28.212/
            redirect.html
11      Client Auth: DISABLED
12      SSL Redirect: DISABLED
```



```
13       SNI: DISABLED
14       SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
             ENABLED
15       Push Encryption Trigger: Always
16       PUSH encryption trigger timeout:     1 ms
17       Send Close-Notify: YES
18       Push flag: 0x0 (Auto)
19       Deny SSL Renegotiation          NO
20       SSL quantum size:          8 kB
21       Strict CA checks:          NO
22       Encryption trigger timeout 100 mS
23       Encryption trigger packet count:     45
24       Use only bound CA certificates: DISABLED
25       Subject/Issuer Name Insertion Format: Unicode
26       Strict Host Header check for SNI enabled SSL sessions:          NO
27
28       ECC Curve: P_256, P_384, P_521
29
30   1)   Cipher Name: AES     Priority :2
31       Description: Predefined Cipher Alias
32
33   1)   Vserver Name: v1
34   2)   Vserver Name: nshttps-::1l-443
35   3)   Vserver Name: nsrpcs-::1l-3008
36   4)   Vserver Name: nskrpcs-127.0.0.1-3009
37   5)   Vserver Name: nshttps-127.0.0.1-443
38   6)   Vserver Name: nsrpcs-127.0.0.1-3008
39   Done
40   <!--NeedCopy-->
```

デフォルトのバックエンドプロファイルには、次の設定があります。

```
1  sh ssl profile ns_default_ssl_profile_backend
2
3  1)Name: ns_default_ssl_profile_backend
4
5       Configuration for Back-End SSL profile
6       Session Reuse: ENABLED          Timeout: 300 seconds
7       Non FIPS Ciphers: DISABLED
8       Server Auth: DISABLED
9       SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2:
            DISABLED
10      Push Encryption Trigger: Always
```



```
      PUSH encryption trigger timeout:     1 ms
      Send Close-Notify: YES
      Push flag: 0x0 (Auto)
      Deny SSL Renegotiation          ALL
      SSL quantum size:          8 kB
      Strict CA checks:          NO
      Encryption trigger timeout 100 mS
      Encryption trigger packet count:     45
      Use only bound CA certificates: DISABLED

      ECC Curve: P_256, P_224, P_521

1)    Cipher Name: AES     Priority :1
      Description: Predefined Cipher Alias

2)    Cipher Name: RC4     Priority :2
      Description: Predefined Cipher Alias

1)    Service Name: s2
2)    Service Name: s1
Done
<!--NeedCopy-->
```

レガシー **SSL** プロファイル

October 7, 2021

> 注:
>
> 従来のプロファイルではなく、拡張プロファイルの使用をお勧めします。拡張プロファイルインフラストラクチャの詳細については、「SSL プロファイルインフラストラクチャ」を参照してください。

重要:

> SSL プロファイルを SSL 仮想サーバーにバインドします。DTLS プロファイルを SSL 仮想サーバーにバインドしないでください。DTLS プロファイルの詳細については、DTLS プロファイルを参照してください。

SSL プロファイルを使用して、Citrix ADC が SSL トラフィックを処理する方法を指定できます。このプロファイルは、仮想サーバ、サービス、サービスグループなどの SSL エンティティの SSL パラメータ設定の集まりであり、設定の容易さと柔軟性を提供します。グローバルパラメータのセットを 1 つだけ設定することに限定されません。グローバルパラメータの複数のセット（プロファイル）を作成し、異なる SSL エンティティに異なるセットを割り当てることができます。SSL プロファイルは、次の 2 つのカテゴリに分類されます。

- フロントエンドプロファイル。フロントエンドエンティティに適用可能なパラメータを含みます。つまり、クライアントからの要求を受信するエンティティに適用されます。
- バックエンドプロファイル。バックエンドエンティティに適用可能なパラメータを含みます。つまり、クライアント要求をサーバーに送信するエンティティに適用されます。

TCP または HTTP プロファイルとは異なり、SSL プロファイルはオプションです。したがって、デフォルトの SSL プロファイルはありません。複数のエンティティ間で同じプロファイルを再利用できます。図形に縦断がアタッチされていない場合は、グローバルレベルで設定された値が適用されます。動的に学習されたサービスには、現在のグローバル値が適用されます。

次の表に、各プロファイルの一部であるパラメータを示します。

Front end profile	Back-end profile
cipherRedirect, cipherURL	denySSLReneg
clearTextPort*	encryptTriggerPktCount
clientAuth, clientCert	nonFipsCiphers
denySSLReneg	pushEncTrigger
dh, dhFile, dhCount	pushEncTriggerTimeout
dropReqWithNoHostHeader	pushFlag
encryptTriggerPktCount	quantumSize
eRSA, eRSACount	serverAuth
insertionEncoding	commonName
nonFipsCiphers	sessReuse, sessTimeout
pushEncTrigger	`SNIEnable`
pushEncTriggerTimeout	ssl3
pushFlag	sslTriggerTimeout
quantumSize	strictCAChecks
redirectPortRewrite	tls1
sendCloseNotify	-
sessReuse, sessTimeout	-
`SNIEnable`	-
ssl3	-
sslRedirect	-
sslTriggerTimeout	-

Front end profile	Back-end profile
strictCAChecks	-
tls1, tls11, tls12	-

* clearTextPort パラメーターは、SSL 仮想サーバーにのみ適用されます。

プロファイルの一部ではないパラメータを設定しようとすると、エラーメッセージが表示されます。たとえば、バックエンドプロファイルで clientAuth パラメータを設定しようとした場合です。

CRL メモリサイズ、OCSP キャッシュサイズ、UndefAction Control、UndefAction Data などの一部の SSL パラメータは、エンティティから独立しているため、前述のプロファイルのいずれにも含まれていません。

SSL プロファイルでは、次の操作がサポートされます。

- 追加：Citrix ADC 上に SSL プロファイルを作成します。プロファイルがフロントエンドかバックエンドかを指定します。フロントエンドがデフォルトです。
- 設定-既存のプロファイルの設定を変更します。
- 設定解除-指定したパラメータをデフォルト値に設定します。パラメータを指定しない場合は、エラーメッセージが表示されます。エンティティのプロファイルを設定解除すると、そのプロファイルはエンティティからバインド解除されます。
- 除去-プロファイルを削除します。どのエンティティでも使用されているプロファイルも削除できません。設定をクリアすると、すべてのエンティティが削除されます。その結果、プロファイルも削除されます。
- [表示]：Citrix ADC で使用可能なすべてのプロファイルを表示します。プロファイル名を指定すると、そのプロファイルの詳細が表示されます。図形を指定すると、その図形に関連付けられた縦断が表示されます。

CLI を使用した SSL プロファイルの作成

- SSL プロファイルを追加するには、次のように入力します。

```
add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )]
<!--NeedCopy-->
```

- 既存のプロファイルを変更するには、次のように入力します。

```
set ssl profile <name>
<!--NeedCopy-->
```

- 既存のプロファイルを設定解除するには、次のように入力します。

```
unset ssl profile <name> [-dh] [-dhFile] [-dhCount] [-eRSA] …
<!--NeedCopy-->
```

- エンティティから既存のプロファイルを設定解除するには、次のように入力します。

```
unset ssl vserver <vServerName>  - sslProfile
<!--NeedCopy-->
```

- 既存のプロファイルを削除するには、次のように入力します。

```
rm ssl profile <name>
<!--NeedCopy-->
```

- 既存のプロファイルを表示するには、次のように入力します。

```
sh ssl profile <name>
<!--NeedCopy-->
```

GUI を使用した **SSL** プロファイルの作成

[システム] > [プロファイル] に移動し、[SSL プロファイル] タブを選択して SSL プロファイルを作成します。

クライアント証明書の検証の厳密な制御を可能にする

Citrix ADC アプライアンスは、単一のルート CA が証明書を発行した場合、有効な中間 CA 証明書を受け入れます。つまり、ルート CA 証明書のみが仮想サーバーにバインドされており、ルート CA がクライアント証明書とともに送信された中間証明書のいずれかを検証する場合、アプライアンスは証明書チェーンを信頼し、ハンドシェイクは成功します。

ただし、クライアントがハンドシェイクで証明書のチェーンを送信する場合、その証明書が SSL 仮想サーバーにバインドされている場合にのみ、CRL または OCSP レスポンダーを使用して中間証明書を検証できます。したがって、中間証明書の 1 つが失効しても、ハンドシェイクは成功します。ハンドシェイクの一部として、SSL 仮想サーバーは、バインドされている CA 証明書のリストを送信します。より厳密に制御するために、SSL 仮想サーバーを構成して、その仮想サーバーにバインドされている CA 証明書の 1 つが署名した証明書のみを受け入れるようにすることができます。これを行うには、仮想サーバーにバインドされた SSL プロファイルで `ClientAuthUseBoundCAChain` 設定を有効にする必要があります。仮想サーバーにバインドされている CA 証明書の 1 つがクライアント証明書に署名していない場合、ハンドシェイクは失敗します。

たとえば、clientcert1 と clientcert2 の 2 つのクライアント証明書が、それぞれ中間証明書 Int-CA-A と Int-CA-B によって署名されているとします。中間証明書は、ルート証明書 Root-CA によって署名されます。Int-CA-A とルート CA は SSL 仮想サーバーにバインドされます。既定の場合 (クライアント認証の境界キャッシュが無効の場合)、クライアント 1 とクライアント 2 の両方が受け入れられます。ただし、ClientAuthUseBoundCAChain が有効になっている場合、Citrix ADC アプライアンスは clientcert1 のみを受け入れます。

CLI を使用したクライアント証明書の検証の厳密な制御の有効化

コマンドプロンプトで、次のように入力します。`set ssl profile <name> -ClientAuthUseBoundCAChain Enabled`

GUI を使用したクライアント証明書の検証の厳密な制御の有効化

1. [システム] > [プロファイル] に移動し、[**SSL** プロファイル] タブを選択して SSL プロファイルを作成するか、既存のプロファイルを選択します。
2. [バインドされた **CA** チェーンを使用したクライアント認証を有効にする] を選択します。

証明書失効リスト

October 7, 2021

CA によって発行された証明書は、通常、有効期限まで有効です。ただし、状況によっては、CA は有効期限が切れる前に発行された証明書を取り消す場合があります。たとえば、所有者の秘密鍵が危険にさらされると、会社または個人の名前が変更されたり、サブジェクトと CA の関連付けが変更されたりします。

証明書失効リスト (CRL) は、シリアル番号と発行者によって無効な証明書を識別します。

証明機関は CRL を定期的に発行します。CRL を使用して無効な証明書を示すクライアント要求をブロックするように Citrix ADC アプライアンスを構成できます。

CA からの CRL ファイルがすでに存在する場合は、そのファイルを Citrix ADC アプライアンスに追加します。更新オプションを設定できます。また、Web の場所または LDAP の場所から、指定した間隔で CRL ファイルを自動的に同期するように Citrix ADC を構成することもできます。アプライアンスは、PEM または DER ファイル形式の CRL をサポートします。Citrix ADC アプライアンスに追加される CRL ファイルのファイル形式を必ず指定してください。

ADC を CA として使用して SSL 展開で使用される証明書を作成した場合は、CRL を作成して特定の証明書を取り消すこともできます。この機能を使用すると、たとえば、Citrix ADC で作成された自己署名証明書が、実稼働環境や特定の日付以降に使用されないようにできます。

注:

デフォルトでは、CRL は Citrix ADC アプライアンス上の/var/netscaler/ssl ディレクトリに保存されます。

ADC アプライアンスで CRL を作成する

ADC アプライアンスを使用して CA として機能し、自己署名証明書を作成できるため、次の証明書を取り消すこともできます。

- 作成した証明書。
- CA 証明書を所有している証明書。

アプライアンスは、これらの証明書の CRL を作成する前に、無効な証明書を取り消す必要があります。アプライアンスは、失効した証明書のシリアル番号をインデックス・ファイルに保存し、証明書が失効するたびにファイルを更新します。インデックスファイルは、証明書が最初に失効したときに自動的に作成されます。

CLI を使用して証明書を取り消すか、CRL を作成する

コマンドプロンプトで、次のコマンドを入力します。

```
create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-revoke <
    input_filename> | -genCRL <output_filename>)
<!--NeedCopy-->
```

例:

```
create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -revoke Invalid-1

create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -genCRL CRL-1
<!--NeedCopy-->
```

GUI を使用して証明書を取り消すか、CRL を作成する

1. [トラフィック管理] > [SSL] に移動し、[はじめに] グループで [CRL 管理] を選択します。
2. 証明書の詳細を入力し、[操作の選択] リストで [証明書の取り消し] または [CRL の生成] を選択します。

既存の CRL を ADC に追加する

Citrix ADC アプライアンスで CRL を構成する前に、CRL ファイルが Citrix ADC アプライアンスにローカルに保存されていることを確認してください。HA セットアップでは、CRL ファイルが両方の ADC アプライアンスに存在する必要があり、ファイルへのディレクトリパスが両方のアプライアンスで同じである必要があります。

CLI を使用して、**Citrix ADC** で **CRL** を追加します

コマンドプロンプトで次のコマンドを入力して、Citrix ADC に CRL を追加し、構成を確認します。

```
add ssl crl <crlName> <crlPath> [-inform (DER | PEM)]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

例:

```
> add ssl crl crl-one /var/netscaler/ssl/CRL-one -inform PEM

Done

> show ssl crl crl-one

            Name: crl-one   Status: Valid,  Days to expiration: 29
            CRL Path: /var/netscaler/ssl/CRL-one
            Format: PEM     CAcert: samplecertkey
            Refresh: DISABLED
            Version: 1
            Signature Algorithm: sha1WithRSAEncryption
            Issuer:  C=US,ST=California,L=Santa Clara,O=NetScaler Inc.,
               OU=SSL Acceleration,CN=www.ns.com/emailAddress=
               support@Citrix ADC appliance.com
            Last_update:Jun 15 10:53:53 2010 GMT
            Next_update:Jul 15 10:53:53 2010 GMT

    1)      Serial Number: 00
            Revocation Date:Jun 15 10:51:16 2010 GMT
     Done
<!--NeedCopy-->
```

GUI を使用して **Citrix ADC** に **CRL** を追加する

[トラフィック管理] > [**SSL**] > [**CRL**] に移動し、CRL を追加します。

CRL 更新パラメータの構成

CRL は、特定の証明書が失効した直後に、定期的に、または場合によっては、認証局によって生成および発行されます。Citrix ADC アプライアンスで CRL を定期的に更新して、無効な証明書で接続しようとするクライアントから保護することをお勧めします。

Citrix ADC アプライアンスは、Web ロケーションまたは LDAP ディレクトリから CRL を更新できます。更新パラメーターと Web の場所または LDAP サーバーを指定する場合、コマンドの実行時に CRL がローカルハードディスクドライブに存在している必要はありません。最初の更新では、CRL File パラメーターで指定されたパスに、ローカルハードディスクドライブにコピーが格納されます。CRL を保存するためのデフォルトのパスは /var/netscaler/sl です。

注：リリース 10.0 以降では、CRL をリフレッシュするメソッドはデフォルトでは含まれていません。HTTP または LDAP メソッドを明示的に指定します。以前のリリースからリリース 10.0 以降にアップグレードする場合は、メソッドを追加して、コマンドを再度実行する必要があります。

CLI を使用した CRL 自動リフレッシュの設定

コマンドプロンプトで次のコマンドを入力して、CRL 自動更新を構成し、構成を確認します。

```
set ssl crl <crlName> [-refresh ( ENABLED | DISABLED )] [-CAcert <
    string>] [-server <ip_addr|ipv6_addr|*> | -url <URL>] [-method (
    HTTP | LDAP )] [-port <port>] [-baseDN <string>] [-scope ( Base |
    One )] [-interval <interval>] [-day <positive_integer>] [-time <HH:
    MM>][-bindDN <string>] {
 -password  }
  [-binary ( YES | NO )]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

例:

```
    set CRL crl1  -refresh enabled -method ldap -inform DER -CAcert ca1
        -server 10.102.192.192 -port 389 -scope base -baseDN "cn=
       clnt_rsa4_multicert_der,ou=eng,o=ns,c=in" -time 00:01

    set ssl crl crl1 -refresh enabled -method http -cacert ca1 -port 80
        -time 00:10 -url http://10.102.192.192/crl/ca1.crl


    > sh crl
```



```
7
8      1)          Name: crl1         Status: Valid,     Days to expiration:
           355
9                   CRL Path: /var/netscaler/ssl/crl1
10                  Format: PEM       CAcert: ca1
11                  Refresh: ENABLED           Method: HTTP
12                  URL: http://10.102.192.192/crl/ca1.crl
                       Port:80
13                  Refresh Time: 00:10
14                  Last Update: Successful, Date:Tue Jul  6 14:38:13 2010
15     Done
16 <!--NeedCopy-->
```

GUI を使用して **LDAP** または **HTTP** を使用して **CRL** 自動リフレッシュを構成する

1. [トラフィック管理] **> [SSL] > [CRL]** に移動します。
2. CRL を開き、[**CRL** 自動更新を有効にする] を選択します。

> 注:
>
> 新しい CRL が、CRL の「最終更新時刻」フィールドで指定された実際の更新時刻より前に外部リポジトリでリフレッシュされた場合は、
>
> Citrix ADC アプライアンスで CRL をただちに更新する必要があります。

最終更新時刻を表示するには、CRL を選択し、[**Details**] をクリックします。

CRL の同期

Citrix ADC アプライアンスは、最新の分散 CRL を使用して、失効した証明書を持つクライアントがセキュリティで保護されたリソースにアクセスすることを防ぎます。

CRL が頻繁に更新される場合、Citrix ADC アプライアンスは、リポジトリから最新の CRL を取得するための自動メカニズムを必要とします。指定した更新間隔で CRL を自動的に更新するようにアプライアンスを設定できます。

アプライアンスは、定期的に更新する必要がある CRL の内部リストを維持します。これらの指定された間隔で、アプライアンスは更新が必要な CRL のリストをスキャンします。次に、リモート LDAP サーバーまたは HTTP サーバーに接続し、最新の CRL を取得してから、ローカル CRL リストを新しい CRL で更新します。

> 注:
>
> CA 証明書が仮想サーバーにバインドされているときに CRL チェックが「必須」に設定されていて、初期 CRL リフレッシュが失敗した場合、次のアクションが接続で実行されます:
>
> CRL と同じ発行者を持つすべてのクライアント認証接続は、CRL が正常にリフレッシュされるまで REVOKED として拒否されます。

CRL リフレッシュを実行する間隔を指定できます。正確な時刻を指定することもできます。

CLI を使用した **CRL** 自動更新の同期

コマンドプロンプトで、次のコマンドを入力します。

```
set ssl crl <crlName> [-interval <interval>] [-day <integer>] [-time <
    HH:MM>]
<!--NeedCopy-->
```

例**:**

```
set ssl crl CRL-1 -refresh ENABLE -interval MONTHLY -days 10 -time
    12:00
<!--NeedCopy-->
```

GUI を使用した **CRL** 更新の同期

1. [トラフィック管理**] > [SSL] > [CRL**] に移動します。
2. CRL を開き、[**CRL** 自動更新を有効にする] を選択して、間隔を指定します。

証明書失効リストを使用してクライアント認証を実行する

証明書失効リスト（CRL）が Citrix ADC アプライアンスに存在する場合、CRL チェックの実行が必須かオプションかに関係なく、CRL チェックが実行されます。

ハンドシェイクの成功または失敗は、次の要因の組み合わせによって決まります。

- CRL チェックのルール
- クライアント証明書チェックの規則
- CA 証明書に設定された CRL の状態

次の表に、失効した証明書を含むハンドシェイクの可能な組み合わせの結果を示します。

表 1. 失効した証明書を使用したクライアントとのハンドシェイクの結果

CRL チェックのルール	クライアント証明書チェックの規則	CA 証明書に設定された CRL の状態	失効した証明書を使用したハンドシェイクの結果
オプション	オプション	行方不明	成功
オプション	固定	行方不明	成功

CRL チェックのルール	クライアント証明書チェックの規則	CA 証明書に設定された CRL の状態	失効した証明書を使用したハンドシェイクの結果
オプション	固定	存在する	失敗
固定	オプション	行方不明	成功
固定	固定	行方不明	失敗
固定	オプション	存在する	成功
固定	固定	存在する	失敗
オプション/必須	オプション	期限切れ	成功
オプション/必須	固定	期限切れ	失敗

> 注:
>
> - CRL チェックは、デフォルトではオプションです。オプションから必須へ、またはその逆に変更するには、まず SSL 仮想サーバーから証明書をバインド解除し、オプションを変更した後に再度バインドする必要があります。
> - `sh ssl vserver`コマンドの出力では、OCSP check: optional は、CRL チェックもオプションであることを意味します。CRL チェックの設定は、CRL チェックが必須に設定されている場合に限り、`sh ssl vserver` コマンドの出力に表示されます。CRL チェックがオプションに設定されている場合、CRL チェックの詳細は表示されません。

CLI を使用して CRL チェックを設定するには

コマンドプロンプトで、次のコマンドを入力します。

```
bind ssl vserver <vServerName> -certkeyName <string> [(-CA -crlCheck (
    Mandatory | Optional ))]
sh ssl vserver
<!--NeedCopy-->
```

例:

```
bind ssl vs v1 -certkeyName ca -CA -crlCheck mandatory
> sh ssl vs v1

Advanced SSL configuration for VServer v1:

```



```
DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1
    .2: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca CA Certificate CRLCheck: Mandatory CA_Name Sent

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用した **CRL** チェックの設定

1. **[**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動し、SSL 仮想サーバーを開きます。
2. [証明書] セクションをクリックします。
3. 証明書を選択し、[**OCSP** および **CRL** チェック] リストで [**CRL** 必須] を選択します。

失効または有効な証明書を使用したハンドシェイクの結果

CRL チェックのルール	クライアント証明書チェックの規則	CA 証明書に設定された CRL の状態	失効した証明書を使用したハンドシェイクの結果	有効な証明書を使用したハンドシェイクの結果
固定	固定	存在する	失敗	成功

CRL チェックのルール	クライアント証明書チェックの規則	CA 証明書に設定された CRL の状態	失効した証明書を使用したハンドシェイクの結果	有効な証明書を使用したハンドシェイクの結果
固定	固定	期限切れ	失敗	失敗
固定	固定	行方不明	失敗	失敗
固定	固定	未定義	失敗	失敗
オプション	固定	存在する	失敗	成功
オプション	固定	期限切れ	成功	成功
オプション	固定	行方不明	成功	成功
オプション	固定	未定義	成功	成功
固定	オプション	存在する	成功	成功
固定	オプション	期限切れ	成功	成功
固定	オプション	行方不明	成功	成功
固定	オプション	未定義	成功	成功
オプション	オプション	存在する	成功	成功
オプション	オプション	期限切れ	成功	成功
オプション	オプション	行方不明	成功	成功
オプション	オプション	未定義	成功	成功

OCSP による証明書ステータスの監視

October 7, 2021

オンライン証明書状態プロトコル (OCSP) は、クライアントの SSL 証明書の状態を決定するために使用されるインターネットプロトコルです。Citrix ADC アプライアンスは、RFC 2560 で定義されている OCSP をサポートしています。OCSP には、タイムリーな情報という点で、証明書失効リスト (CRL) よりも大きな利点があります。クライアント証明書の最新の失効ステータスは、多額の金銭や価値の高い株式取引を含む取引で特に役立ちます。また、使用するシステムリソースとネットワークリソースも少なくなります。Citrix ADC の OCSP 実装には、要求のバッチ処理と応答のキャッシュが含まれます。

OCSP の実装

Citrix ADC アプライアンスでの OCSP 検証は、SSL ハンドシェイク中にアプライアンスがクライアント証明書を受信したときに開始されます。証明書を検証するために、アプライアンスは OCSP 要求を作成し、それを OCSP 応答

側に転送します。そのために、アプライアンスはローカルに設定された URL を使用します。アプライアンスがサーバからの応答を評価し、トランザクションを許可するか拒否するかを決定するまで、トランザクションは中断状態になります。サーバーからの応答が構成された時間を超えて遅延し、他のレスポンダーが構成されていない場合、アプライアンスは、OCSP チェックがオプションに設定されているか必須に設定されているかに応じて、トランザクションを許可するか、エラーを表示します。

アプライアンスは、OCSP 要求のバッチ処理と OCSP 応答のキャッシュをサポートし、OCSP 応答側の負荷を軽減し、応答を高速化します。

OCSP 要求のバッチ処理

アプライアンスは、クライアント証明書を受信するたびに、OCSP レスポンダに要求を送信します。OCSP レスポンダが過負荷にならないように、アプライアンスは同じ要求内の複数のクライアント証明書のステータスを照会できます。この機能を効率的に機能させるには、バッチの形成を待機している間に単一の証明書の処理が過度に遅延しないように、タイムアウトを定義する必要があります。

OCSP 応答キャッシュ

OCSP レスポンダから受信した応答をキャッシュすると、クライアントへの応答が高速になり、OCSP レスポンダの負荷が軽減されます。OCSP レスポンダからクライアント証明書の失効ステータスを受け取ると、アプライアンスは応答をローカルにキャッシュします。SSL ハンドシェイク中にクライアント証明書を受信すると、アプライアンスはまずローカルキャッシュにこの証明書のエントリをチェックします。（キャッシュのタイムアウト制限内で）有効なエントリが見つかった場合、そのエントリが評価され、クライアント証明書が受け入れられるか拒否されます。証明書が見つからない場合、アプライアンスは OCSP レスポンダに要求を送信し、設定された時間だけ応答をローカルキャッシュに保存します。

> 注: リリース 12.1 ビルド 49.x から、キャッシュのタイムアウト制限は最大 43200 分 (30 日) に増加しました。以前は 1440 分（1 日）でした。この制限を大きくすると、OCSP サーバー上のルックアップを減らし、ネットワークやその他の問題によって OCSP サーバーに到達できない場合に SSL/TLS 接続障害を回避できます。

OCSP レスポンダの設定

OCSP の構成には、OCSP レスポンダーの追加、OCSP レスポンダーの証明機関 (CA) 証明書へのバインド、証明書の SSL 仮想サーバーへのバインドが含まれます。すでに設定されている OCSP 応答側に別の証明書をバインドする必要がある場合は、まず応答側のバインドを解除してから、応答側を別の証明書にバインドする必要があります。

CLI を使用した **OCSP** レスポンダーの追加

コマンドプロンプトで、次のコマンドを入力して OCSP を構成し、構成を確認します。


```
add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [ -batchingDepth <
    positive_integer>][-batchingDelay <positive_integer>] [-resptimeout
    <positive_integer>] [-responderCert <string> | -trustResponder] [-
    producedAtTimeSkew <positive_integer>][-signingCert <string>][-
    useNonce ( YES | NO )][ -insertClientCert( YES | NO )]
<!--NeedCopy-->
```

```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]
<!--NeedCopy-->
```

```
bind ssl vserver <vServerName>@ (-certkeyName <string> ( CA [-ocspCheck
     ( Mandatory | Optional )]))
<!--NeedCopy-->
```

```
show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

例:

```
add ssl ocspResponder ocsp_responder1 -url ”http:// www.myCA.org:80/
    ocsp/” -cache ENABLED -cacheTimeout 30 -batchingDepth 8 -
    batchingDelay 100 -resptimeout 100 -responderCert responder_cert -
    producedAtTimeSkew 300 -signingCert sign_cert  -insertClientCert YES
<!--NeedCopy-->
```

```
bind ssl certKey ca_cert -ocspResponder ocsp_responder1 -priority 1
<!--NeedCopy-->
```

```
bind ssl vserver vs1 -certkeyName ca_cert -CA -ocspCheck Mandatory
<!--NeedCopy-->
```



```
sh ocspResponder ocsp_responder1

    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done
<!--NeedCopy-->
```

```
show certkey ca_cert

    Name: ca_cert     Status: Valid,   Days to expiration:8907
    Version: 3
    …

    1)  VServer name: vs1      CA Certificate
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
Done
<!--NeedCopy-->
```

```
sh ssl vs vs1

    Advanced SSL configuration for VServer vs1:
    DH: DISABLED
    …

    1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory
    1) Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
    Done
<!--NeedCopy-->
```


CLI を使用した **OCSP** レスポンダーの変更

レスポンダ名は変更できません。他のすべてのパラメータは、`set ssl ocspResponder`コマンドを使用して変更できます。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED)
    ] [-cacheTimeout <positive_integer>] [-batchingDepth <
    positive_integer>] [-batchingDelay <positive_integer>] [-resptimeout
     <positive_integer>] [ -responderCert <string> | -trustResponder][-
    producedAtTimeSkew <positive_integer>][-signingCert <string>] [-
    useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

GUI を使用して **OCSP** レスポンダを構成する

1. [トラフィック管理] > [**SSL**] > [**OCSP** レスポンダー] に移動し、OCSP レスポンダーを設定します。
2. [トラフィック管理] > [**SSL**] > [証明書] に移動し、証明書を選択し、[アクション] リストで [**OCSP** バインディング] を選択します。OCSP レスポンダをバインドします。
3. 「トラフィック管理」>「負荷分散」>「仮想サーバー」に移動し、仮想サーバーを開き、「証明書」セクションをクリックして CA 証明書をバインドします。
4. 必要に応じて、**select OCSP Mandatory** を選択します。

OCSP ホチキス止め

October 7, 2021

CRL と OCSP の Citrix ADC 実装では、クライアント証明書の失効状態のみが報告されます。SSL ハンドシェイク中に受信したサーバー証明書の失効ステータスを確認するには、クライアントが認証局に要求を送信する必要があります。

トラフィックの多いウェブサイトでは、多くのクライアントが同じサーバー証明書を受け取ります。各クライアントがサーバー証明書の失効状態に関するクエリを送信した場合、証明書の有効性を確認するための OCSP 要求が証明機

関に浸水されます。

OCSP ホチキス止めソリューション

不要な輻輳を避けるために、Citrix ADC アプライアンスは OCSP ホチキス止めをサポートするようになりました。つまり、アプライアンスは、OCSP レスポンダから証明書ステータスを検証した後、SSL ハンドシェイク時に、サーバ証明書の失効ステータスをクライアントに送信できるようになりました。サーバ証明書の失効ステータスは、SSL ハンドシェイクの一部としてアプライアンスがクライアントに送信する応答に対して「ホチキス止め」されます。OCSP ホチキス止め機能を使用するには、SSL 仮想サーバーでこの機能を有効にし、アプライアンスに OCSP レスポンダを追加する必要があります。

注:

- Citrix ADC アプライアンスは、RFC 6066 で定義されている OCSP ホチキス止めをサポートします。
- OCSP ホチキス止めは、Citrix ADC アプライアンスのフロントエンドでのみサポートされます。

> 重要:
>
> Citrix ADC による OCSP ホチキス止めのサポートは、TLS プロトコルバージョン 1.0 以降を使用したハンドシェイクに限定されます。

サーバ証明書の **OCSP** 応答キャッシュ

SSL ハンドシェイク中に、クライアントがサーバー証明書の失効ステータスを要求すると、アプライアンスはまずローカルキャッシュにこの証明書のエントリをチェックします。有効なエントリが見つかった場合、そのエントリが評価され、サーバ証明書とそのステータスがクライアントに表示されます。失効ステータス・エントリが見つからない場合、アプライアンスはサーバー証明書の失効ステータスの要求を OCSP レスポンダに送信します。応答を受信すると、証明書と失効ステータスをクライアントに送信します。次のアップデートフィールドが OCSP 応答に存在する場合、応答は設定された時間（timeout フィールドに指定された値）だけキャッシュされます。

> 注：リリース 12.1 ビルド 49.x から、タイムアウトが切れる前であっても、OCSP レスポンダからサーバ証明書のキャッシュされた応答をクリアできます。以前は、設定されたタイムアウトが過ぎるまで、証明書とキーのペアのキャッシュされたステータスを廃棄できませんでした。

CLI を使用してキャッシュされたステータスをクリアするには、コマンドプロンプトで次のように入力します。

```
clear ssl certKey <certkey name> -ocspstaplingCache
<!--NeedCopy-->
```

例:


```
clear ssl certKey s1 -ocspstaplingCache
<!--NeedCopy-->
```

GUI を使用してキャッシュされた状態をクリアするには

1. GUI で、[トラフィック管理] > [**SSL**] > [証明書] > [**CA** 証明書] に移動します。
2. 詳細ペインで、証明書を選択します。
3. [アクションの選択] リストで、[クリア] を選択します。確認メッセージが表示されたら、[はい] をクリックします。

OCSP ホチキス止めの設定

OCSP ホチキス止めの設定には、機能の有効化と OCSP の設定が含まれます。OCSP を設定するには、OCSP レスポンダを追加し、OCSP レスポンダを CA 証明書にバインドし、証明書を SSL 仮想サーバにバインドする必要があります。

> 注:
>
> HTTP ベースの URL のみを持つ OCSP レスポンダがサポートされます。

CLI を使用して **OCSP** ホチキス止めを有効にします

コマンドプロンプトで入力します。

```
set ssl vserver <name> -ocspstapling [ENABLED | DISABLED]
<!--NeedCopy-->
```

例:

```
set ssl vserver vip1 -ocspStapling ENABLED
Done

sh ssl vserver vip1

      Advanced SSL configuration for VServer vip1:
      DH: DISABLED
      DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA:
          ENABLED Refresh Count: 0
      Session Reuse: ENABLED Timeout: 120 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
```



```
12       ClearText Port: 0
13       Client Auth: DISABLED
14       SSL Redirect: DISABLED
15       Non FIPS Ciphers: DISABLED
16       SNI: ENABLED
17       OCSP Stapling: ENABLED
18       SSLv2: DISABLED SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED
              TLSv1.2: ENABLED
19       Push Encryption Trigger: Always
20       Send Close-Notify: YES
21
22       ECC Curve: P_256, P_384, P_224, P_521
23
24      1) CertKey Name: server_certificate1 Server Certificate
25
26      1) Cipher Name: DEFAULT
27       Description: Default cipher list with encryption strength >= 128
            bit
28  Done
29  <!--NeedCopy-->
```

注：デフォルト（拡張）プロファイルが有効になっている場合は、`set ssl profile <profile name> -ocspStapling [ENABLED | DISABLED]` コマンドを使用して OCSP を有効または無効にします。

GUI を使用して **OCSP** ホチキス止めを有効にします

1. [トラフィック管理] > [**SSL**] > [仮想サーバー] に移動します。
2. 仮想サーバーを開き、**[SSL** パラメータ**]** で [**OCSP** ホチキス止め] を選択します。

OCSP 構成

OCSP 応答側は、OCSP ホチキス止め要求を送信するために、動的にまたは手動で追加されます。内部応答側は、サーバー証明書の OCSP URL に基づいてサーバー証明書とその発行者証明書を追加すると、動的に追加されます。手動の OCSP 応答側が CLI または GUI から追加されます。サーバー証明書の OCSP 要求を送信するために、Citrix ADC アプライアンスは、発行者証明書にバインドするときに、割り当てられた優先順位に基づいて OCSP レスポンダーを選択します。応答側が OCSP ホチキス止め要求の送信に失敗した場合、次に優先順位の高い応答側が要求の送信対象として選択されます。たとえば、手動で構成された応答側が 1 つだけ失敗し、動的にバインドされた応答側が存在する場合、OCSP 要求の送信対象として選択されます。

OCSP URL が HTTP 以外の場合、内部 OCSP レスポンダは作成されません。

> 注
>
> 手動で追加した OCSP 応答側は、動的に追加された応答側よりも優先されます。

手動で作成された **OCSP** 応答側と内部で作成された **OCSP** 応答側の違い

手動で作成された **OCSP** レスポンダ	内部的に（動的に）作成された **OCSP** レスポンダ
手動で明示的に作成し、優先度で発行者証明書にバインドします。	サーバー証明書とその発行者証明書（CA 証明書）を追加する際に、デフォルトで作成およびバインドされます。名前は「ns_internal_」で始まります。
1 ～127 のプライオリティは、設定された応答側用に予約されています。	プライオリティは 128 以降から自動的に割り当てられます。
URL とバッチの深さは変更できます。	URL とバッチの深さは変更できません。
直接削除されました。	サーバー証明書または CA 証明書を削除した場合にのみ削除されます。
任意の CA 証明書にバインドできます。	デフォルトでは、1 つの CA 証明書にバインドされます。他の CA 証明書にはバインドできません。
構成（ns.conf）に保存されます。	Add コマンドは構成に保存されません。set コマンドだけが保存されます。
3 つの OCSP レスポンダを、プライオリティ 1、2、3 の同じ発行者証明書にバインドし、後でプライオリティ 2 をバインド解除しても、その他のプライオリティは影響を受けません。	3 つの OCSP レスポンダが、それぞれプライオリティ 128、129、および 130 の発行者証明書に自動的にバインドされます。優先度 129 でバインドされた応答側を作成するために使用されたサーバー証明書を削除すると、その応答側は削除されます。また、次の応答側の優先度（優先度 130）は自動的に 129 に変更されます。

> リクエスト処理の例:

1. 仮想サーバー (VIP1) を追加します。
2. 発行者証明書 (CA1) を追加し、VIP1 にバインドします。
3. 3 つの証明書 S1、S2、および S3 を追加します。内部応答側の resp1、resp2、resp3 はそれぞれデフォルトで作成されます。
4. S3 を VIP1 にバインドします。
5. 要求が VIP1 に送信されます。レスポンダー resp3 が選択されています。

内部 OCSP レスポンダを動的に作成するには、アプライアンスに次のものが必要です。

- サーバー証明書の発行者の証明書（通常は CA 証明書）。

- サーバ証明書の証明書とキーのペア。この証明書には、証明機関が提供する OCSP URL が含まれている必要があります。URL は、動的に追加された内部応答者の名前として使用されます。

内部 OCSP 応答側のデフォルト値は、手動で設定された応答側と同じです。

> 注:
>
> 内部応答側では、キャッシュはデフォルトで無効になっています。キャッシュを有効にするには、`set ssl ocspResponder`コマンドを使用します。

CLI を使用した **OCSP** の設定

コマンドプロンプトで、次のコマンドを入力して OCSP を構成し、構成を確認します。

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]] [-bundle ( YES | NO )]

add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [-resptimeout <positive_integer
    >] [-responderCert <string> | -trustResponder] [-producedAtTimeSkew
    <positive_integer>][-signingCert <string>][-useNonce ( YES | NO )][
    -insertClientCert ( YES | NO )]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

パラメーター:

httpMethod:

OCSP 要求の送信に使用する HTTP メソッド。要求の長さが 255 バイト未満の場合は、OCSP サーバへのクエリーに対して HTTP GET メソッドを設定できます。GET メソッドを指定しても、長さが 255 バイトを超える場合、アプライアンスはデフォルトの方法（POST）を使用します。

可能な値：GET、POST

デフォルト値：POST

ocspUrlResolveTimeout:

OCSP URL 解決を待機する時間（ミリ秒単位）。この時間が経過すると、次に高い優先度を持つレスポンダが選択されます。すべてのレスポンダが失敗した場合は、仮想サーバの設定に応じて、エラーメッセージが表示されるか、接続が切断されます。

最小値:100

最大値:2000

例**:**

```
add ssl certkey root_ca1  - cert root_cacert.pem
add ssl ocspResponder ocsp_responder1 -url ”http:// www.myCA.org:80/
    ocsp/” -cache ENABLED -cacheTimeout 30  -resptimeout 100 -
    responderCert responder_cert -producedAtTimeSkew 300 -signingCert
    sign_cert  -insertClientCert YES
bind ssl certKey root_ca1 -ocspResponder ocsp_responder1 -priority 1
sh ocspResponder ocsp_responder1
    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done

show certkey root_ca1
    Name: root_ca1     Status: Valid,   Days to expiration:8907
    Version: 3
    …
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
    Done
<!--NeedCopy-->
```

CLI を使用した **OCSP** の変更

OCSP レスポンダの名前は変更できませんが、`set ssl ocspResponder`コマンドを使用して他のパラメータを変更できます。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。


```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED)
    ] [-cacheTimeout <positive_integer>]  [-resptimeout <
    positive_integer>] [ -responderCert <string> | -trustResponder][-
    producedAtTimeSkew <positive_integer>][-signingCert <string>] [-
    useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

GUI を使用した **OCSP** の設定

1. [トラフィック管理] > [**SSL**] > [**OCSP** レスポンダー] に移動し、OCSP レスポンダーを設定します。
2. [トラフィック管理] > [**SSL**] > [証明書] に移動し、証明書を選択し、[アクション] リストで [**OCSP** バインディング] を選択します。**OCSP** レスポンダをバインドします。
3. 「トラフィック管理」>「負荷分散」>「仮想サーバー」に移動し、仮想サーバーを開き、「証明書」セクションをクリックして CA 証明書をバインドします。
4. 必要に応じて、**OCSP Mandatory** を選択します。

> 注:
>
> `add ssl ocspResponder`コマンドと`set ssl ocspResponder`コマンドでクライアント証明書の挿入パラメーターが無効になりました。つまり、パラメータは設定時に無視されます。

Citrix ADC アプライアンスで利用可能な暗号

September 26, 2022

Citrix ADC アプライアンスには、事前定義された一連の暗号グループが付属しています。DEFAULT 暗号グループに含まれない暗号を使用するには、SSL 仮想サーバーに明示的にバインドする必要があります。また、ユーザー定義の暗号グループを作成して、SSL 仮想サーバーにバインドすることもできます。ユーザー定義の暗号グループの作成の詳細については、ADC アプライアンスでのユーザー定義の暗号グループの構成を参照してください。

> 注
>
> RC4 暗号は、Citrix ADC アプライアンスのデフォルトの暗号グループには含まれません。ただし、N3 ベースのアプライアンス上のソフトウェアではサポートされています。ハンドシェイクを含む RC4 暗号化はソフトウ

> ェアで行われます。
>
> この暗号は安全ではないと見なされ、RFC 7465 で非推奨になっているため、使用しないことをお勧めします。
>
> アプライアンスに N3 チップが搭載されているかどうかを確認するには、「show hardware」コマンドを使用します。

```
sh hardware

Platform: NSMPX-22000 16\*CPU+24\*IX+12\*E1K+2\*E1K+4*CVM N3 2200100

Manufactured on: 8/19/2013

CPU: 2900MHZ

Host Id: 1006665862

Serial no: ENUK6298FT

Encoded serial no: ENUK6298FT
<!--NeedCopy-->
```

- フロントエンド (仮想サーバー) でデフォルトでバインドされている暗号スイートに関する情報を表示するには、次のように入力します。`sh cipher DEFAULT`
- バックエンドで (サービスに) デフォルトでバインドされている暗号スイートに関する情報を表示するには、以下のように入力します。`sh cipher DEFAULT_BACKEND`
- アプライアンスに定義されているすべての暗号グループ (エイリアス) に関する情報を表示するには、次のように入力します。`sh cipher`
- 特定の暗号グループに含まれるすべての暗号スイートに関する情報を表示するには、`sh cipher <alias name>`と入力します。たとえば、sh 暗号 ECDHE です。

以下のリンクは、さまざまな Citrix ADC プラットフォームおよび外部ハードウェアセキュリティモジュール（HSM）でサポートされている暗号スイートの一覧です。

- **Citrix ADC MPX/SDX**（**N3**）アプライアンス：Citrix ADC MPX/SDX（N3）アプライアンスでの暗号サポート
- **Citrix ADC MPX/SDX** インテル **Coleto** アプライアンス：Citrix ADC MPX/SDX インテルコレト SSL チップベースアプライアンスでの暗号サポート
- **Citrix ADC VPX** アプライアンス：Citrix ADC VPX アプライアンスでの暗号サポート
- **Citrix ADC MPX/SDX 14000 FIPS** アプライアンス：Citrix ADC MPX/SDX 14000 FIPS アプライアンスでの暗号サポート
- 外部 **HSM (Thales/**セーフネット**):** 外部 HSM でサポートされる暗号 (Thales/Safenet)
- **Citrix ADC MPX/SDX**（**N2**）アプライアンス：Citrix ADC MPX/SDX（N2）アプライアンスでの暗号サポート

- **Citrix ADC MPX 9700 FIPS** アプライアンス：ファームウェア 2.2 を搭載した Citrix ADC MPX 9700 FIPS での暗号サポート
- **Citrix ADC VPX FIPS** および **MPX FIPS** アプライアンス：Citrix ADC VPX FIPS および MPX FIPS 認定アプライアンスでの暗号サポート

> 注：
>
> DTLS 暗号のサポートについては、Citrix ADC VPX、MPX、および SDX アプライアンスでの DTLS 暗号のサポートを参照してください。

表 **1-**仮想サーバー**/**フロントエンドサービス**/**内部サービスのサポート：

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
TLS 1.3	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	未サポート	13.0 すべてのビルド
	12.1—50.x (TLS1.3-CHACHA20-POLY1305-SHA256 を除く)	12.1—50.x (TLS1.3-CHACHA20-POLY1305-SHA256 を除く)	12.1–50.x	未サポート	12.1–50.x
TLS 1.1/1.2	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x
	11.0 全ビルド	11.0 全ビルド	11.0 全ビルド	11.0 全ビルド	11.0—70.x (MPX 5900/8900 でのみ)
	10.5 全ビルド	10.5 全ビルド	10.5–57.x	10.5–59.1359.e	10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
ECDHE/DHE (例 TLS1-ECDHE-RSA-AES128-SHA)	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x
	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	11.1–51.x	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x
	11.0 全ビルド	11.0 全ビルド	11.0 全ビルド		11.0—70.114 (MPX 5900/8900 でのみ)
	10.5–53.x	10.5–53.x	10.5 全ビルド		10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
AES-GCM (例 TLS1.2-AES128-GCM-SHA256)	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 〜 50.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	11.1—51.x (注を参照)	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x
	11.0 全ビルド	11.0 全ビルド	11.0–66.x		11.0—70.114 (MPX 5900/8900 でのみ)
	10.5–53.x	10.5–53.x			10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
SHA-2 暗号 (例 TLS1.2-AES-128-SHA256)	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x
	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	11.1–52.x	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.0 全ビルド	11.0 全ビルド	11.0–66.x		11.0 ～72.x、11.0-70.114 (MPX 5900/8900 でのみ)
	10.5–53.x	10.5–53.x			10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
ECDSA (例 TLS1-ECDHE-ECDSA-AES256-SHA)	未サポート	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	未サポート	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	未サポート	12.0 全ビルド	12.0-57.x	未サポート	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
		11.1 全ビルド			11.1 ～56.x、11.1 ～54.126 (ECC カーブ P_256 と P_384 のみがサポートされています)
CHACHA20	未サポート	13.0 すべてのビルド	13.0 すべてのビルド	未サポート	13.0 すべてのビルド
	未サポート	未サポート	12.1 全ビルド	未サポート	12.1 ～49.x (MPX 5900/8900 でのみ)
	未サポート	未サポート	12.0–56.x	未サポート	未サポート

表 **2-**バックエンドサービスのサポート**:**

TLS 1.3 はバックエンドではサポートされていません。

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
TLS 1.1/1.2	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x
	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	11.1 全ビルド	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.0–50.x	11.0–50.x	11.0–66.x		11.0—70.119 (MPX 5900/8900 でのみ)
	10.5–59.x	10.5–59.x		10.5–59.1359.e	10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)
ECDHE/DHE (例 TLS1-ECDHE-RSA-AES128-SHA)	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 〜 50.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.0 全ビルド	12.0 全ビルド	12.0–56.x	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
	11.1 全ビルド	11.1 全ビルド		11.1–51.x	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x
	11.0–50.x	11.0–50.x			11.0—70.119 (MPX 5900/8900 でのみ)
	10.5–58.x	10.5–58.x			10.5—67.x、10.5-63.47 (MPX 5900/8900 のみ)

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
AES-GCM (例 TLS1.2-AES128-GCM-SHA256)	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x
	12.0 全ビルド	12.0 全ビルド	未サポート	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.1 全ビルド	11.1 全ビルド		11.1–51.x	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x
SHA-2 暗号 (例 TLS1.2-AES-128-SHA256)	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x
	12.0 全ビルド	12.0 全ビルド	未サポート	12.0 全ビルド	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.1 全ビルド	11.1 全ビルド		11.1–52.x	MPX 5900/8900 および MPX 15000-50G の場合は 11.1-56.x、MPX 26000-100G の場合は 11.1-60.x
ECDSA (例 TLS1-ECDHE-ECDSA-AES256-SHA)	未サポート	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド	13.0 すべてのビルド
	未サポート	12.1 全ビルド	12.1 全ビルド	12.1 全ビルド	12.1 MPX 5900/8900 のすべてのビルド、MPX 15000-50G および MPX 26000-100G の場合は 12.1 ～ 50.x

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	未サポート	12.0 全ビルド	12.0–57.x	未サポート	MPX 5900/8900 の場合はすべて 12.0、MPX 15000-50G の場合は 12.0-57.x、MPX 26000-100G の場合は 12.0-60.x
		11.1–51.x			MPX 5900/8900 および MPX 15000-50G の場合は 11.1 ～ 56.x、MPX 26000-100G の場合は 11.1 ～ 60.x (ECC カーブ P_256 および P_384 のみがサポートされます)。
CHACHA20	未サポート	13.0 すべてのビルド	13.0 すべてのビルド	未サポート	13.0 すべてのビルド

プロトコル/プラットフォーム	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	未サポート	未サポート	12.1 全ビルド	未サポート	MPX 5900/8900 の場合は 12.1 ～ 49.x、MPX 15000-50 G および MPX 26000-100G の場合は 12.1 ～ 50.x
	未サポート	未サポート	12.0–56.x	未サポート	未サポート

サポートされる ECDSA 暗号の詳細なリストについては、ECDSA 暗号スイートのサポートを参照してください。

注

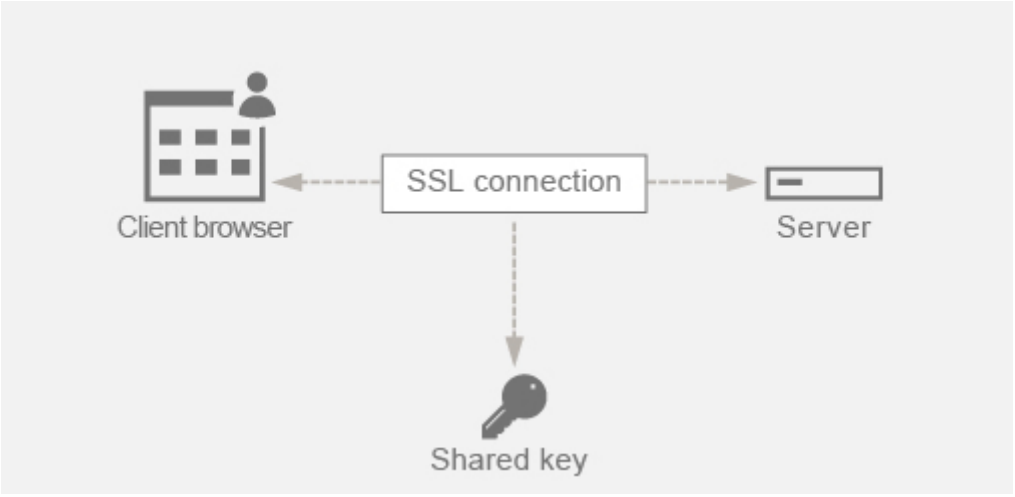
- TLS fallback_SCSV 暗号スイートは、リリース 10.5 ビルド 57.x 以降のすべてのアプライアンスでサポートされています。
- HTTP 厳密なトランスポートセキュリティ (HSTS) のサポートはポリシーベースです。
- すべての SHA-2 署名付き証明書（SHA256、SHA384、SHA512）は、すべてのアプライアンスのフロントエンドでサポートされています。リリース 11.1 ビルド 54.x 以降では、これらの証明書はすべてのアプライアンスのバックエンドでもサポートされています。リリース 11.0 以前では、すべてのアプライアンスのバックエンドで SHA256 署名付き証明書のみがサポートされています。
- リリース 11.1 ビルド 52.x 以前では、次の暗号は MPX9700 のフロントエンドでのみサポートされています。MPX/SDX 14000 FIPS アプライアンス:
  - TLS1.2-ECDHE-RSA-AES-256-SHA384
  - TLS1.2-ECDHE-RSA-AES256-GCM-SHA384. From release 11.1 build 53.x, and in release 12.0, these ciphers are also supported on the back end.
- すべての ChaCha20-Poly1035 暗号は、SHA-256 ハッシュ関数と TLS 擬似ランダム関数（PSF）を使用します。

Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy は、ウェブサーバーのセッションキーが後の時点で侵害された場合でも、現在の SSL 通信を確実に保護します。

なぜ **Perfect Forward Secrecy (PFS)** が必要なのですか

SSL 接続は、クライアントとサーバー間で渡されるデータを保護するために使用されます。この接続は、クライアントのブラウザと接続した Web サーバーとの間で行われる SSL ハンドシェイクから始まります。このハンドシェイク中に、ブラウザとサーバーが特定の情報を交換してセッションキーに到達します。セッションキーは、通信の残りの部分を通してデータを暗号化する手段として機能します。



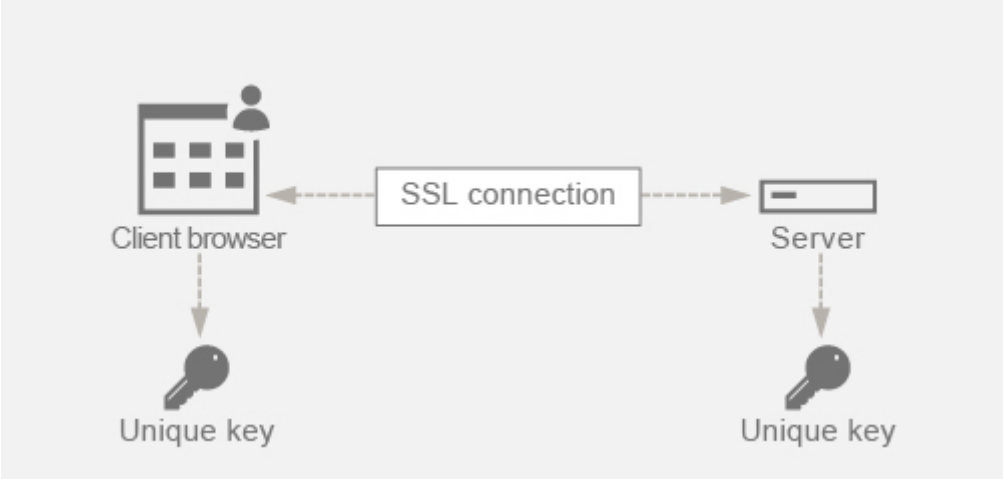
RSA は、鍵交換で最も一般的に使用されるアルゴリズムです。ブラウザは、サーバーの公開鍵を使用して暗号化し、プリマスターシークレットを介してサーバーに送信します。この事前マスターシークレットは、セッションキーに到達するために使用されます。RSA キー交換アプローチの問題は、攻撃者が将来の任意の時点でサーバーの秘密鍵を保持することができた場合、攻撃者がセッションキーを取得できる事前マスターシークレットを保持することです。攻撃者がこのセッションキーを使用して、すべての SSL カンバセーションを復号化できるようになりました。つまり、過去の SSL 通信は以前セキュリティで保護されていましたが、サーバーの盗まれた秘密鍵を使用してセッションキーに到達し、保存された履歴カンバセーションも復号化できるため、セキュリティがなくなることを意味します。

サーバの秘密キーが侵害された場合でも、過去の SSL 通信を保護できるようにする必要があります。ここで Perfect Forward Secrecy (PFS) の設定が助けになります。

PFS はどのように役立つのですか

Perfect Forward Secrecy（PFS）は、クライアントとサーバーをセッションごとに新しいキーに同意させ、このセッションキーの計算を秘密にすることで、過去の SSL 通信を保護します。これは、サーバーキーの侵害によってセッションキーが侵害されてはならないという基準で機能します。セッションキーは両端で別々に派生し、ワイヤを介して転送されることはありません。セッションキーは、通信が完了すると破棄されます。これらの事実により、誰かが

サーバーの秘密鍵にアクセスしても、セッションキーに到達できず、過去のデータを復号化できないことが保証されます。



例を挙げた説明

PFS の達成に DHE を使用していると仮定します。DH アルゴリズムは、ハッカーがサーバーの秘密鍵を保持していても、セッションキーと乱数（セッションキーに到達するために使用される）が両端で秘密に保たれ、ワイヤを介して交換されないため、ハッカーがセッションキーに到達できないようにします。
PFS は、SSL セッションごとに新しい一時キーを作成するエフェメラル Diffie-Hellman キー交換を使用することで実現できます。

セッションごとにキーを作成する反対側は、追加の計算が必要ですが、キーサイズが小さい楕円曲線を使用することでこれを克服できます。

Citrix ADC アプライアンスで **PFS** を構成する

PFS は、DHE または ECDHE 暗号を構成することで、Citrix ADC 上で構成できます。これらの暗号により、作成されたシークレットセッションキーがワイヤ上で共有されず（DH アルゴリズム）、セッションキーが短時間だけ存続することが保証されます（エフェメラル）。次のセクションでは、両方の構成について説明します。

注**:** DHE の代わりに ECDHE 暗号を使用すると、小さなキーサイズで通信がより安全になります。

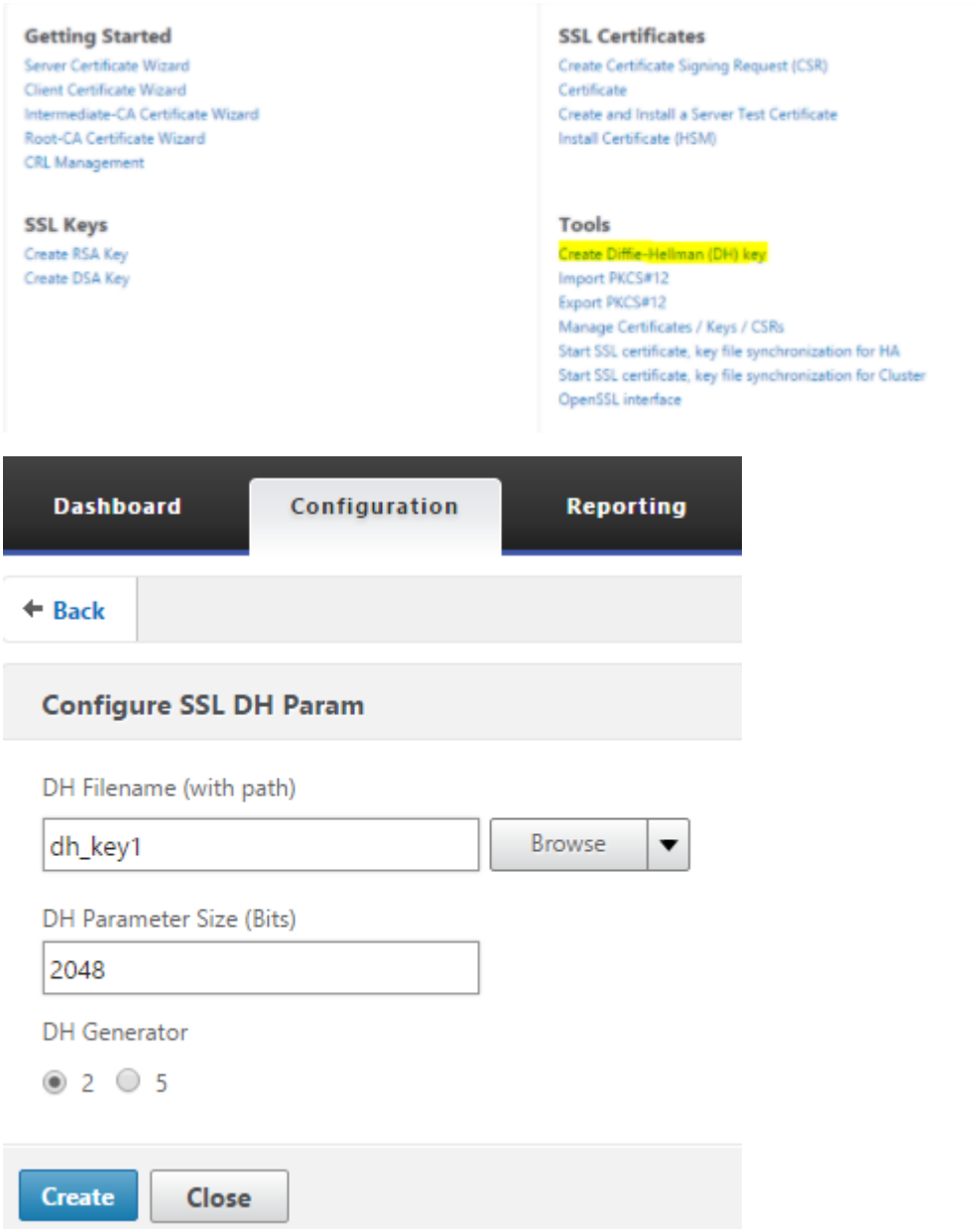
GUI を使用して **DHE** を構成する

1. DH キーを生成します。

   a. [トラフィック管理**] > [SSL] > [**ツール**]** に移動します。

   b. 「**Diffie Helman (DH)** キーを作成」をクリックします。

   注**:** 2048 ビット DH キーの生成には最大 30 分かかることがあります。

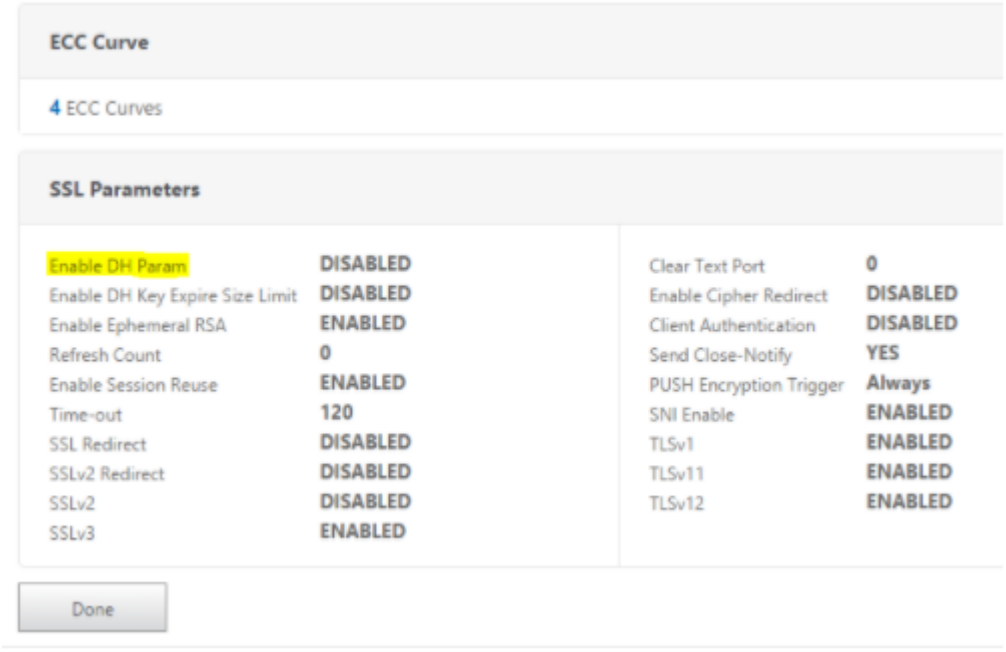


2. SSL 仮想サーバーの DH Param を有効にし、SSL 仮想サーバーに DH キーを接続します。

   a. [設定] **>** [トラフィック管理] **>** [仮想サーバ] に移動します。

   b. DH を有効にする仮想サーバーを選択します。

   c. [編集] をクリックし、[**SSL** パラメータ] をクリックし、[**DH** パラメータを有効にする] をクリックします。



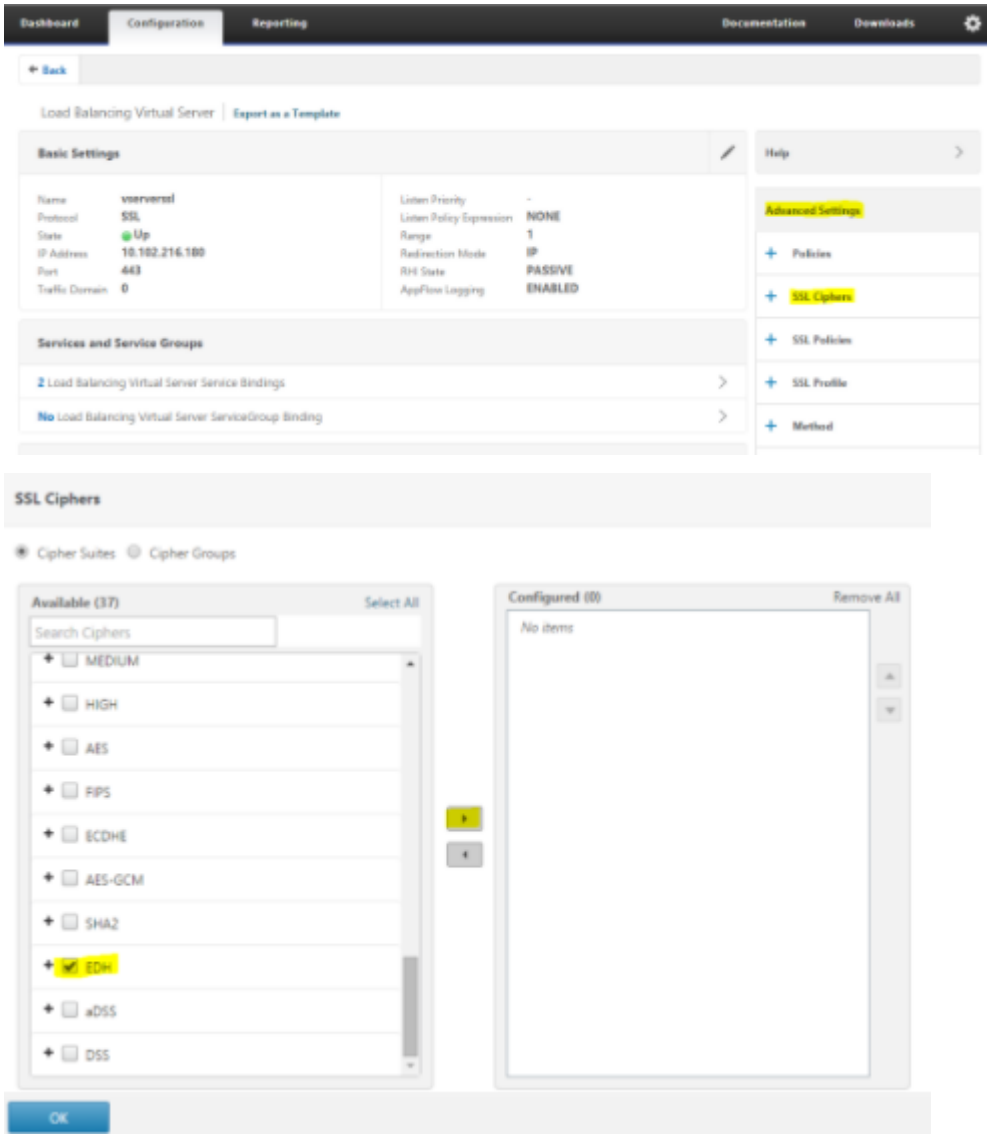
3. DHE 暗号を仮想サーバーにバインドします。

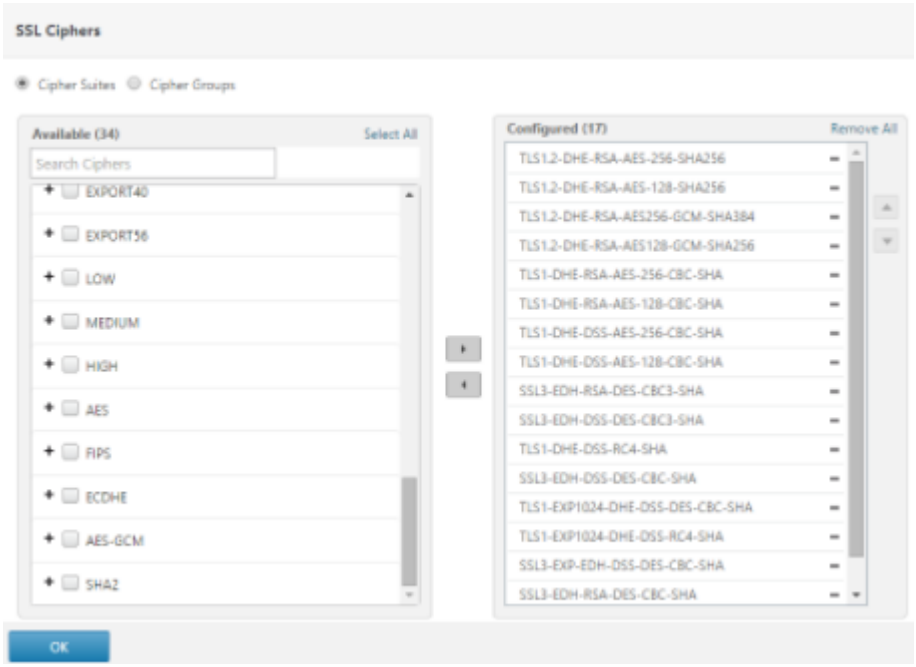
   a. [設定**] > [**トラフィック管理**] > [**仮想サーバ] に移動します。

   b. DH を有効にする仮想サーバーを選択し、鉛筆アイコンをクリックして編集します。

   c. ［詳細設定］で、［**SSL Ciphers** ］の横にあるプラスアイコンをクリックし、DHE 暗号グループを選択し、［**OK**］をクリックしてバインドします。

   注**:** DHE 暗号が仮想サーバにバインドされた暗号リストの一番上に配置されていることを確認してください。





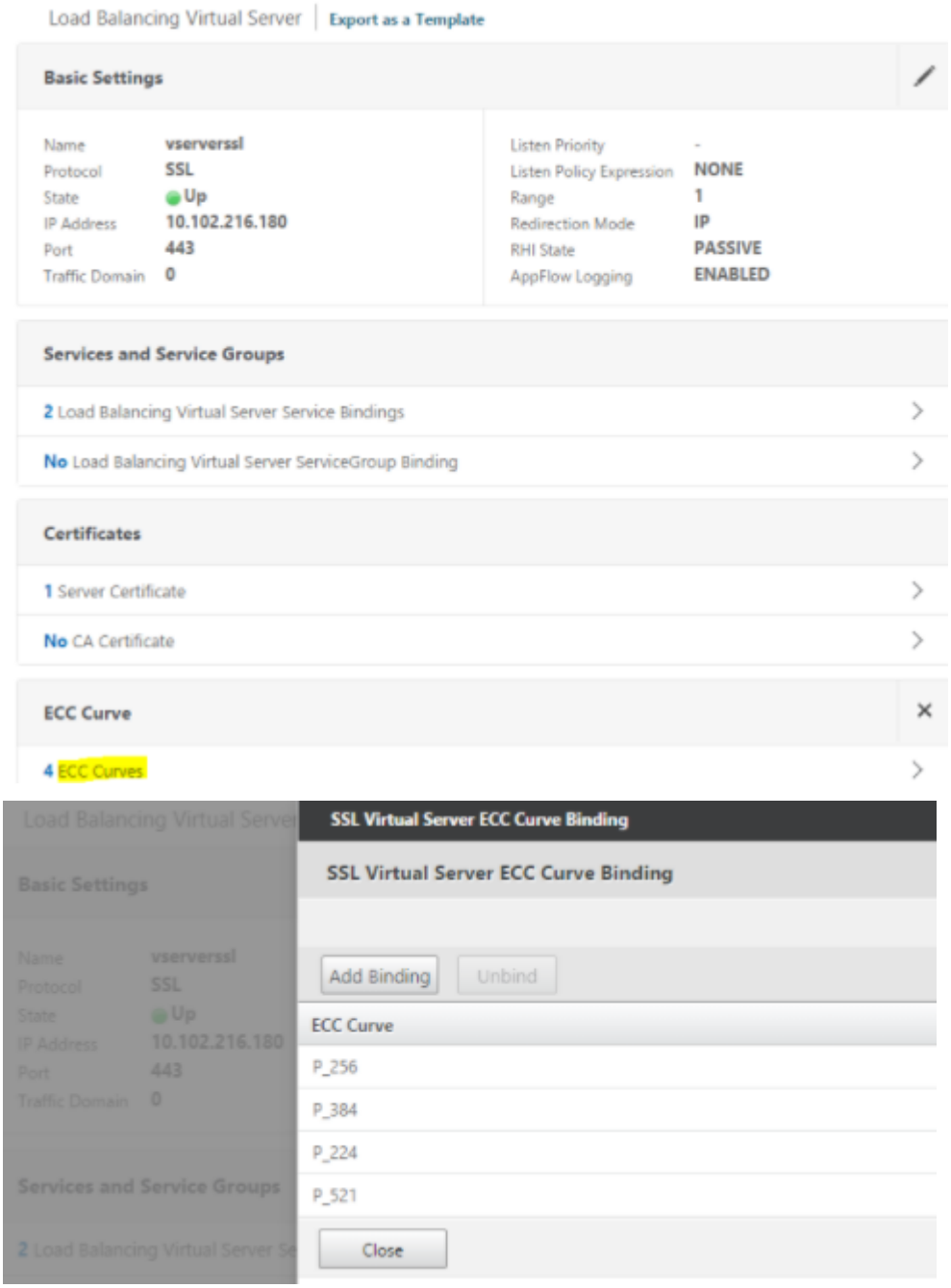
GUI を使用した **ECDHE** の設定

1. ECC カーブを SSL 仮想サーバーにバインドします。

   a.**[**設定**] > [**トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動します。

   b. 編集する SSL 仮想サーバーを選択し、[**ECC Curve**] をクリックして、[バインドの追加] をクリックします。

   c. 必要な ECC カーブを仮想サーバにバインドします。

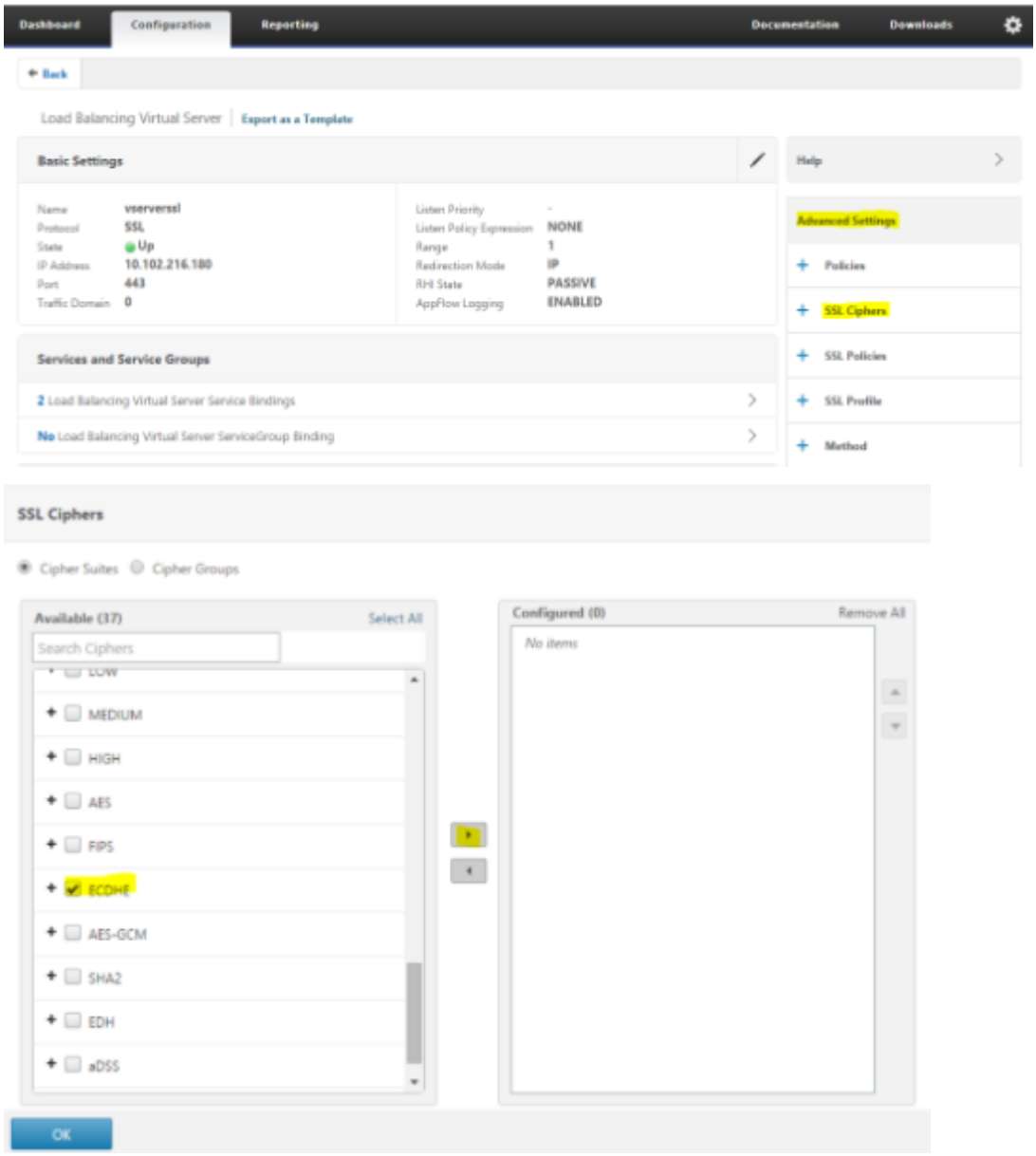


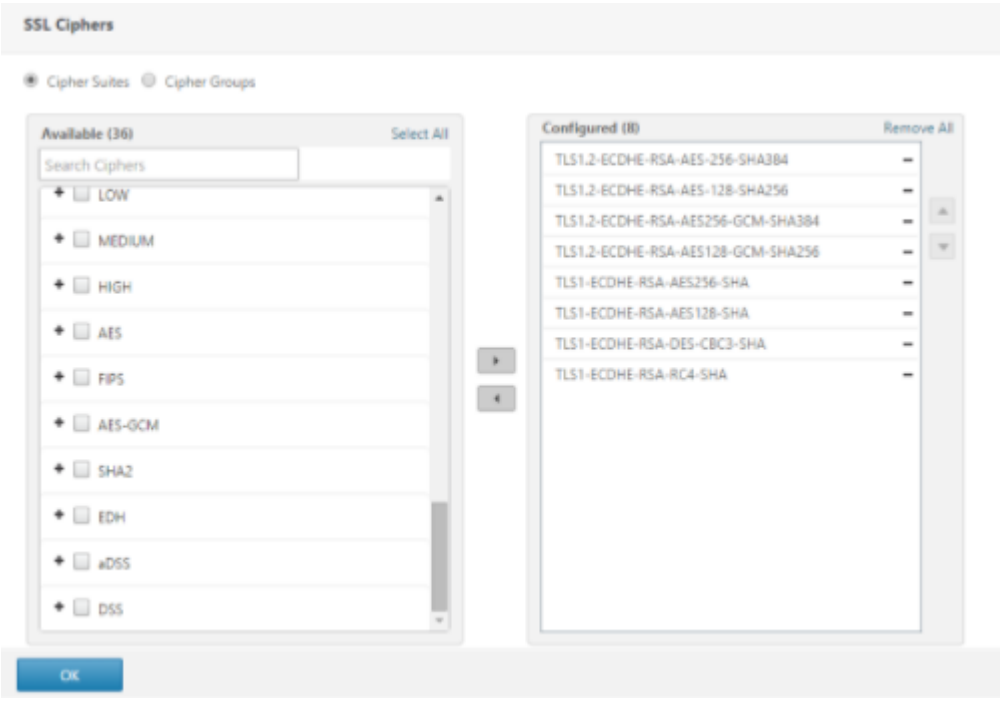
2. ECDHE 暗号を仮想サーバにバインドします。

   a. [構成] **>** [トラフィック管理] **>** [仮想サーバー] に移動し、DH を有効にする仮想サーバーを選択します。

   b. [編集] **>** [**SSL** 暗号] をクリックし、ECDHE 暗号グループを選択して [バインド] をクリックします。

   注**:** ECDHE 暗号が、仮想サーバーにバインドされた暗号リストの一番上にあることを確認してください。





注：それぞれのケースで、Citrix ADC アプライアンスが通信に使用する暗号をサポートしていることを確認します。

SSL プロファイルを使用した **PFS** の設定

注**:** SSL プロファイルを使用して PFS（暗号または ECC）を設定するオプションは、11.0 64.x リリース以降から導入されます。古いバージョンの場合は、次のセクションを無視してください。

SSL プロファイルを使用して PFS を有効にするには、仮想サーバ上で直接設定するのではなく、SSL プロファイルで同様の設定（前の設定セクションで説明した）を実行する必要があります。

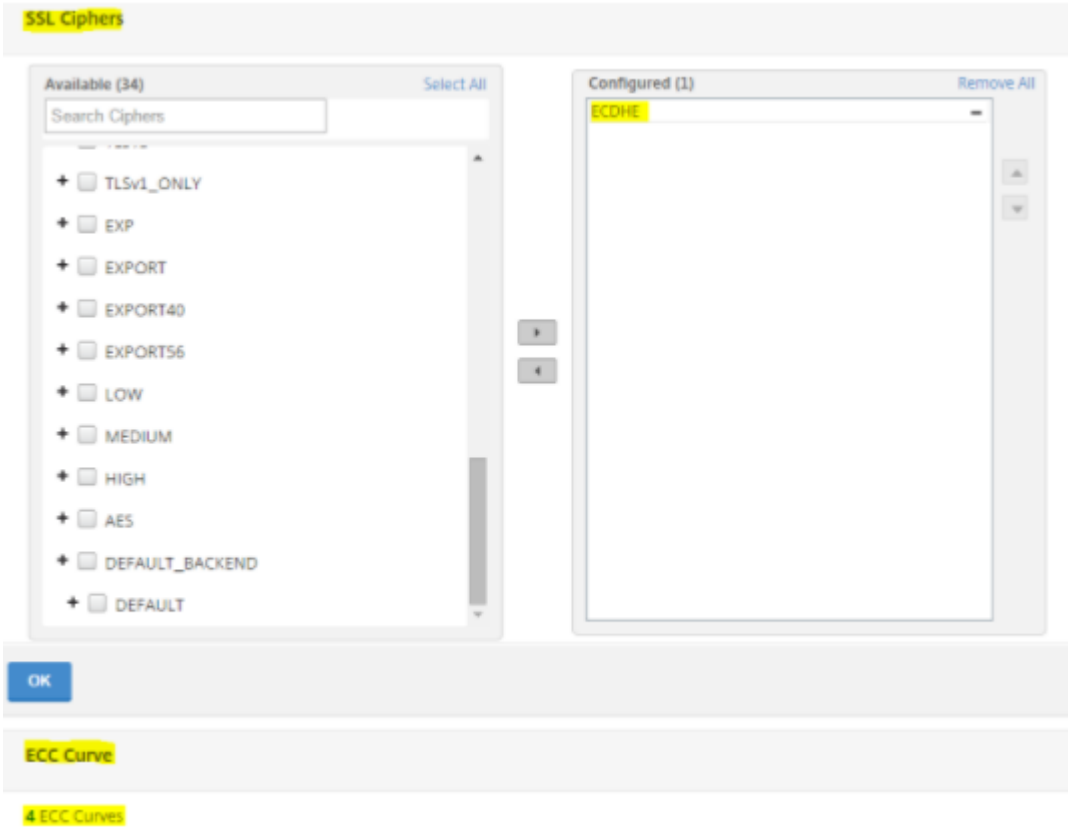
GUI を使用して **SSL** プロファイルを使用した **PFS** の設定

1. SSL プロファイルで ECC カーブと ECDHE 暗号をバインドします。

   注**:** ECC カーブは、デフォルトですべての SSL プロファイルに既にバインドされています。

   a. [システム**] > [**プロファイル**] > [SSL** プロファイル] に移動し、PFS を有効にするプロファイルを選択します。

   b. ECDHE 暗号をバインドします。

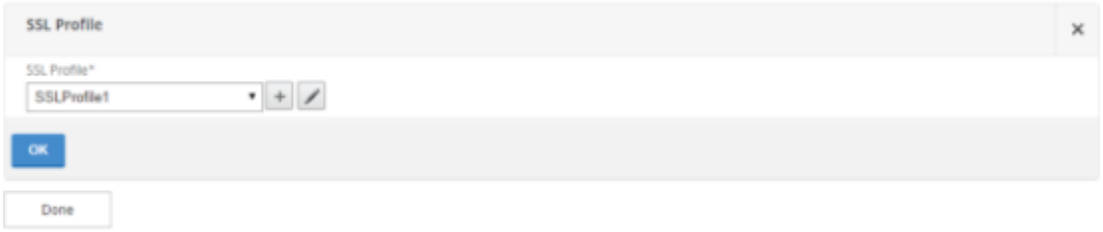


2. SSL プロファイルを仮想サーバーにバインドします。

   a. [構成] **>** **[**トラフィック管理**] >** **[**仮想サーバー] に移動し、仮想サーバーを選択します。

   b. 鉛筆アイコンをクリックして SSL プロファイルを編集します。

   c. 「**OK**」をクリックし、「完了」をクリックします。



CLI を使用した **SSL** を使用した **PFS** の設定

コマンドプロンプトで入力します。

1. ECC カーブを SSL プロファイルにバインドします。

```
bind sslprofile <SSLProfileName> -eccCurveName <Name_of_curve>
<!--NeedCopy-->
```


2. ECDHE 暗号グループをバインドします。

```
bind sslprofile <SSLProfileName> cipherName <ciphergroupName>
<!--NeedCopy-->
```

3. ECDHE 暗号のプライオリティを 1 に設定します。

```
set sslprofile <SSLProfileName> cipherName <ciphergroupName>
   cipherPriority <positive_integer>
<!--NeedCopy-->
```

4. SSL プロファイルを仮想サーバーにバインドします。

```
set SSL vserver <vservername> sslProfile <SSLProfileName>
<!--NeedCopy-->
```

ECDHE の暗号の組み合わせ

October 7, 2021

すべての Citrix ADC アプライアンスは、フロントエンドとバックエンドで ECDHE 暗号グループをサポートします。SDX アプライアンスでは、SSL チップが VPX インスタンスに割り当てられている場合、MPX アプライアンスの暗号サポートが適用されます。それ以外の場合は、VPX インスタンスの通常の暗号サポートが適用されます。

これらの暗号をサポートするビルドとプラットフォームの詳細については、Citrix ADC アプライアンスで使用可能な暗号を参照してください。

ECDHE 暗号スイートは、楕円曲線暗号（ECC）を使用します。キーサイズが小さいため、ECC は、モバイル（無線）環境またはミリ秒ごとに重要な対話型音声応答環境で特に役立ちます。キーサイズを小さくすると、電力、メモリ、帯域幅、計算コストを節約できます。

Citrix ADC アプライアンスは、次の ECC カーブをサポートしています。

- P_256
- P_384
- P_224
- P_521

注: リリース 10.1 ビルド 121.10 より前のビルドからアップグレードする場合は、ECC カーブを既存の SSL 仮想サーバーまたはサービスに明示的にバインドする必要があります。曲線は、デフォルトで、アップグレード後に作成するすべての仮想サーバーおよびサービスにバインドされます。

ECC 曲線を SSL フロントエンドおよびバックエンドエンティティにバインドできます。既定では、4 つのカーブはすべて P_256、P_384、P_224、P_521 の順序でバインドされます。順序を変更するには、まずすべてのカーブのバインドを解除してから、目的の順序でバインドする必要があります。

CLI を使用して **ECC** 曲線を **SSL** 仮想サーバーにバインドします

コマンドプロンプトで入力します。

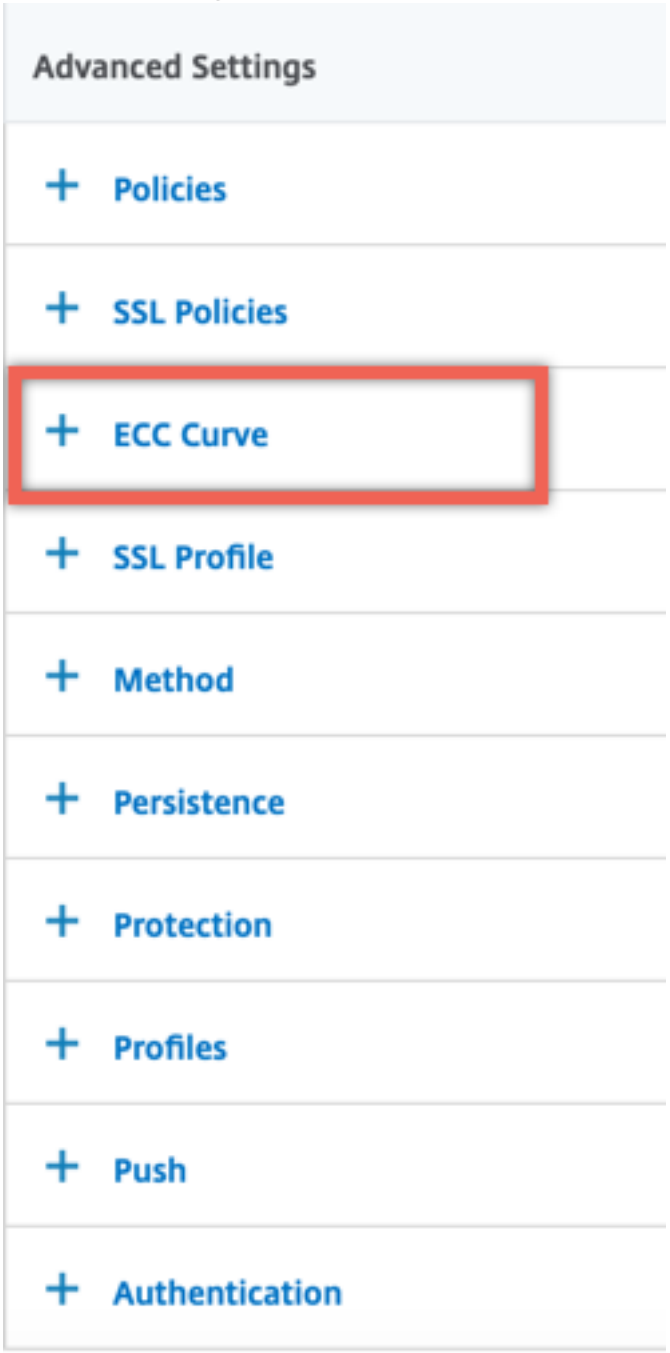
```
bind ssl vserver <vServerName > -eccCurveName <eccCurveName >
```

例:

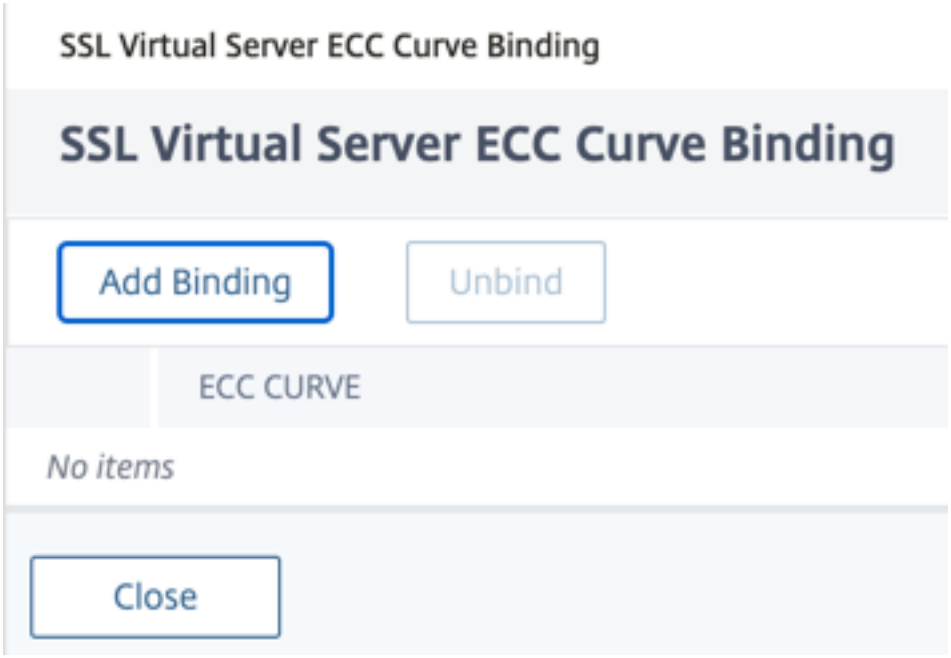
```
1  bind ssl vserver  v1 -eccCurveName P_224
2
3  sh ssl vserver v1
4
5  Advanced SSL configuration for VServer v1:
6  DH: DISABLED
7  Ephemeral RSA: ENABLED Refresh Count: 0
8  Session Reuse: ENABLED Timeout: 120 seconds
9  Cipher Redirect: DISABLED
10 SSLv2 Redirect: DISABLED
11 ClearText Port: 0
12 Client Auth: DISABLED
13 SSL Redirect: DISABLED
14 Non FIPS Ciphers: DISABLED
15 SNI: DISABLED
16 SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED
      TLSv1.2: DISABLED
17 Push Encryption Trigger: Always
18 Send Close-Notify: YES
19 ECC Curve: P_224
20
21 1) Cipher Name: DEFAULT
22 Description: Predefined Cipher Alias
23 Done
24 <!--NeedCopy-->
```


GUI を使用して **ECC** 曲線を **SSL** 仮想サーバーにバインドする

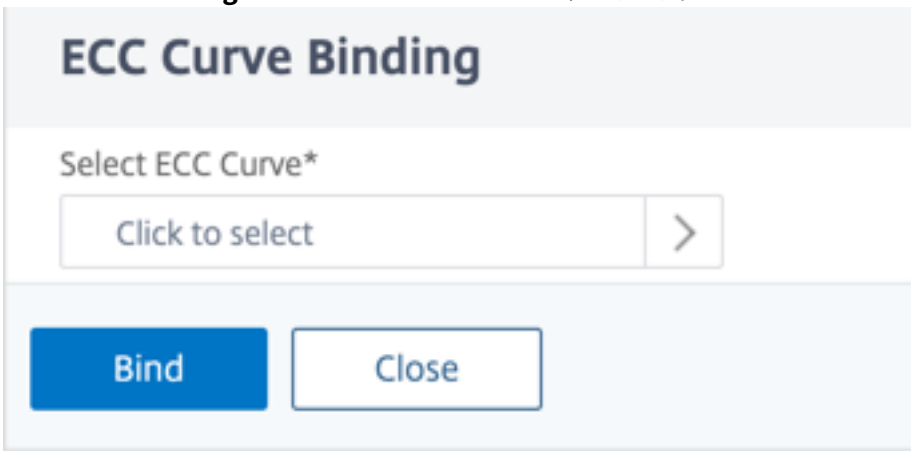
1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL 仮想サーバーを選択し、[編集] をクリックします。
3. **Advanced Settings** で **ECC Curve** をクリックします。



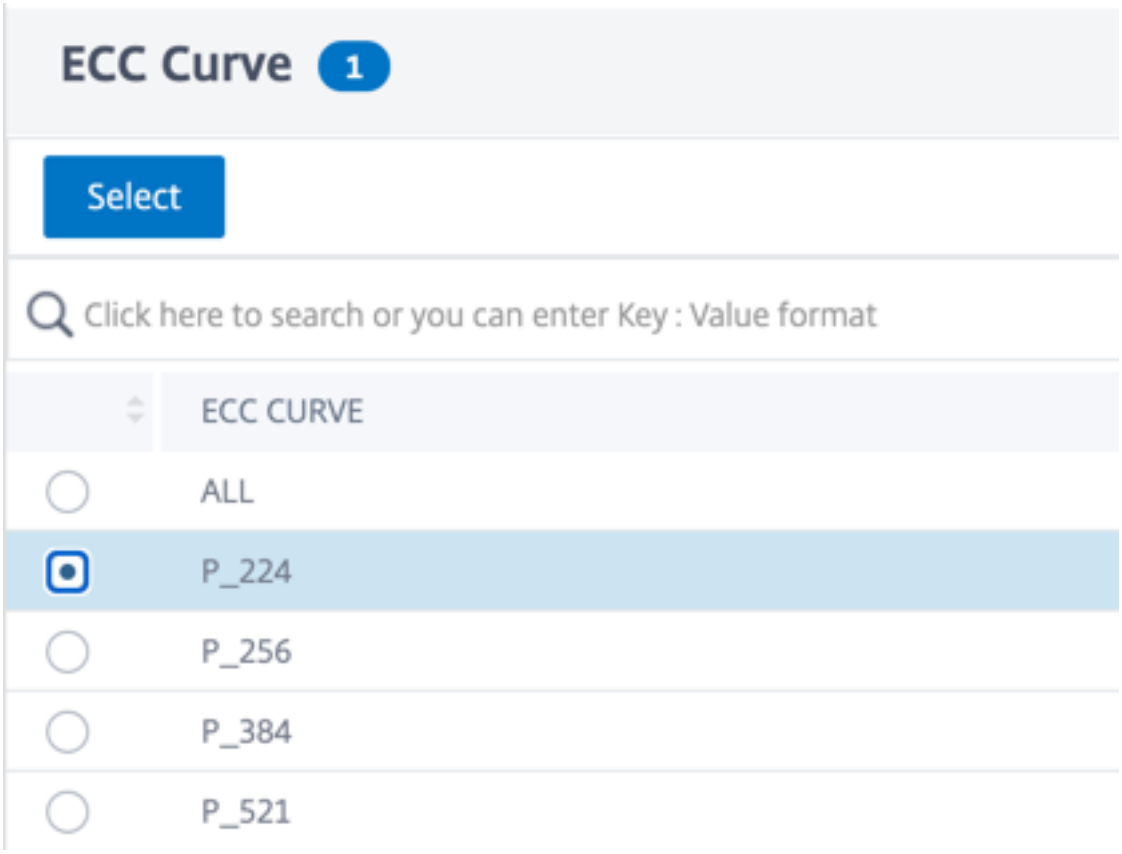
4. ECC 曲線セクションの内側をクリックします。
5. [**SSL** 仮想サーバー **ECC** 曲線バインディング] ページで、[バインディングの追加] をクリックします。



6. **ECC Curve Binding** で **Select ECC Curve** をクリックします。



7. 値を選択して **Select** をクリックします。



8. [バインド] をクリックします。
9. [閉じる] をクリックします。
10. [完了] をクリックします。

CLI を使用して ECC 曲線を SSL サービスにバインドします

コマンドプロンプトで入力します。

```
bind ssl service <vServerName > -eccCurveName <eccCurveName >
```

例:

```
> bind ssl service sslsvc -eccCurveName P_224
 Done
> sh ssl service sslsvc

    Advanced SSL configuration for Back-end SSL Service sslsvc:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       DISABLED
    Session Reuse: ENABLED      Timeout: 300 seconds
```



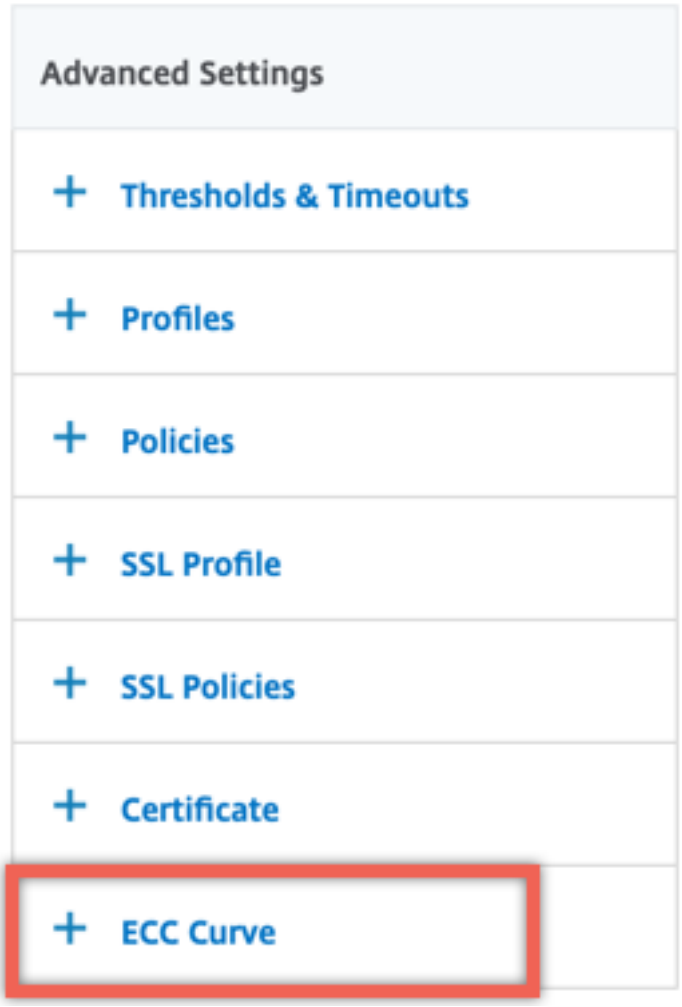
```
    Cipher Redirect: DISABLED
    ClearText Port: 0
    Server Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    OCSP Stapling: DISABLED
    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: ???
    DHE Key Exchange With PSK: ???
    Tickets Per Authentication Context: ???

    ECC Curve: P_224

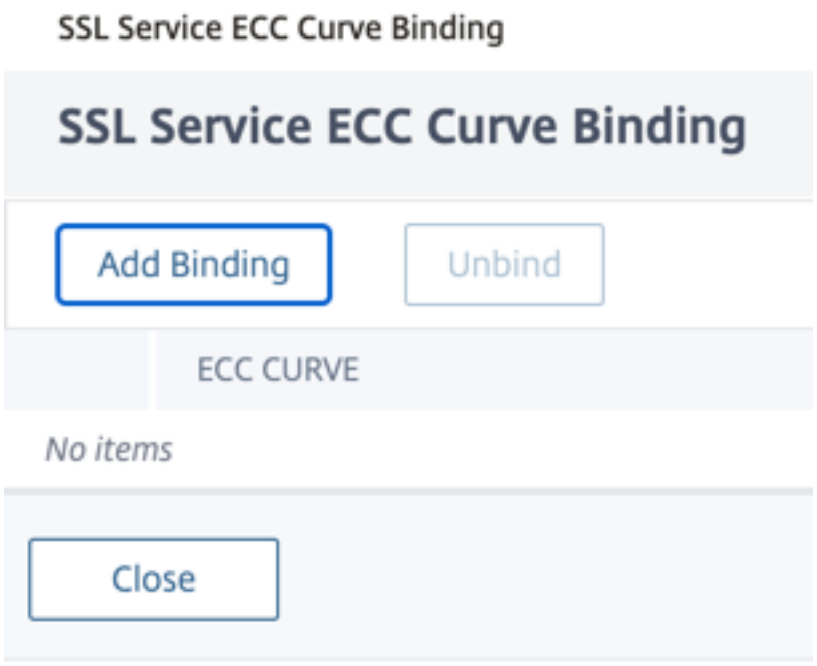

1)  Cipher Name: DEFAULT_BACKEND
    Description: Default cipher list for Backend SSL session
 Done
<!--NeedCopy-->
```

GUI を使用して **ECC** 曲線を **SSL** サービスにバインドする

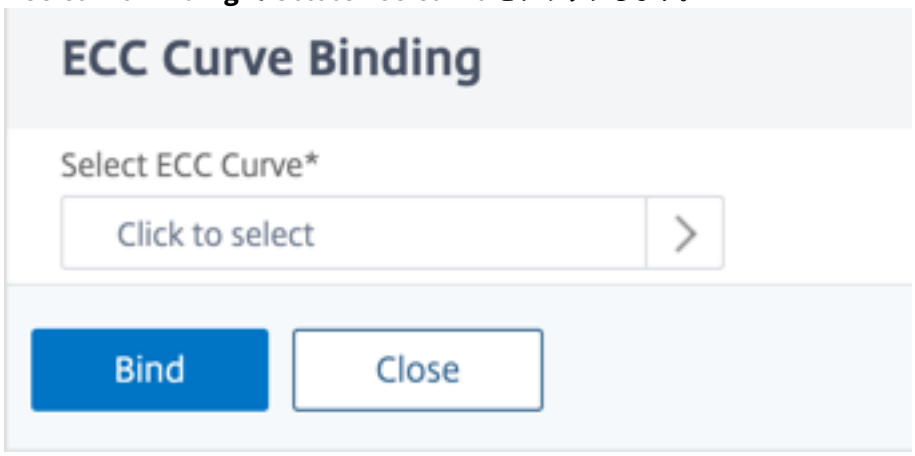
1. **Traffic Management > Load Balancing > Services** に移動します。
2. SSL サービスを選択し、[編集] をクリックします。
3. **Advanced Settings** で **ECC Curve** をクリックします。



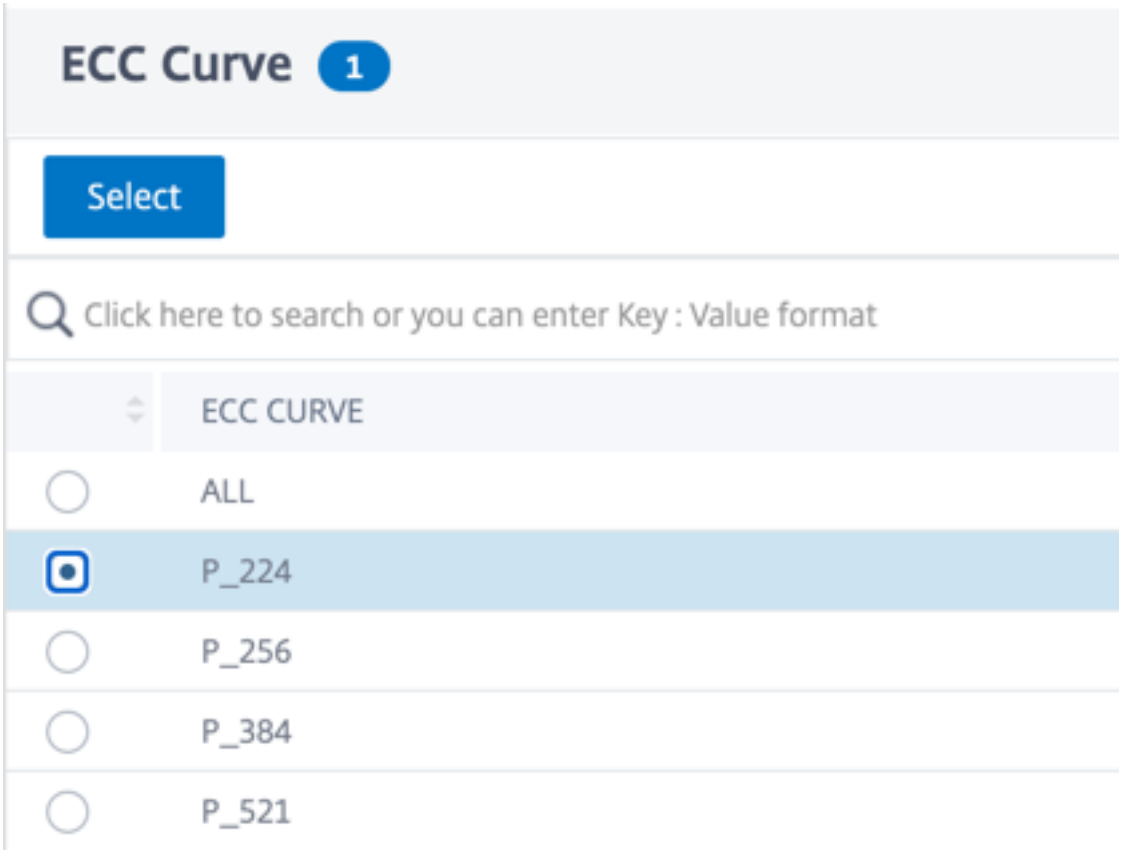
4. ECC 曲線セクションの内側をクリックします。
5. [**SSL** サービス **ECC** 曲線バインディング] ページで、[バインディングの追加] をクリックします。



6. **ECC Curve Binding** で **Select ECC Curve** をクリックします。



7. 値を選択して **Select** をクリックします。



8. ［バインド］をクリックします。
9. ［閉じる］をクリックします。
10. ［完了］をクリックします。

Diffie-Hellman パラメータの生成と **DHE** による **PFS** の実現

October 7, 2021

Diffie-Hellman（DH）キー交換は、SSL トランザクションに関与する 2 つの当事者が安全でないチャネルを介して共有秘密に同意する方法です。これらの当事者は、お互いについての事前知識を持っていません。このシークレットは、このようなキー交換を必要とする対称キー暗号アルゴリズムの暗号キー生成情報に変換できます。

この機能はデフォルトでは無効になっています。キー交換アルゴリズムとして DH を使用する暗号をサポートするように機能を設定しました。

> 注:
>
> 2048 ビットの DH パラメータの生成には長い時間がかかることがあります (最大 30 分)。

CLI を使用して DH パラメータを生成する

コマンドプロンプトで、次のコマンドを入力します。

```
create ssl dhparam <dhFile> [<bits>] [-gen (2 | 5)]
<!--NeedCopy-->
```

例:

```
create ssl dhparam Key-DH-1 512 -gen 2
<!--NeedCopy-->
```

GUI を使用して DH パラメータを生成する

「トラフィック管理」>「**SSL** 」に移動し、「ツール」グループで「**Diffie-Hellman (DH)** キーの作成」を選択し、「**SSL DH** パラメータの設定」を選択します。

> 注:
>
> DH パラメーターの詳細については、Diffie-Hellman パラメーターを参照してください。

DHE で完全な前方秘密を達成する

DH パラメータの生成は、CPU を大量に消費する操作です。以前のリリースでは、VPX アプライアンスでのパラメータ生成はソフトウェアで実行されているため、時間がかかりました。dhKeyExpSizeLimitパラメータを設定することで、パラメータの生成が最適化されます。SSL 仮想サーバーまたは SSL プロファイルに対してこのパラメーターを設定し、プロファイルを仮想サーバーにバインドできます。

DH カウントをゼロに設定することで、Citrix ADC MPX アプライアンスで Perfect Forward Secrecy（PFS）を維持できます。その結果、Citrix ADC MPX アプライアンスでは、トランザクションごとに DH パラメータが生成されます（最小値DHcountは 0）。操作が最適化されているため、パラメータはパフォーマンスを大幅に低下させることなく生成されます。以前は、許容される最小 DH カウントは 500 でした。つまり、最大 500 個のトランザクションに対してキーを再生成できませんでした。

Citrix ADC VPX アプライアンスでは、最低でも 500 トランザクションごとに DH パラメーターを生成できます (DHcount = 500)。DHcountを 0 に設定すると、DH パラメータは再生成されません。

> 制限事項:
>
> 現在、DH 暗号を使用して VPX で PFS を達成することはできません。

CLI を使用した DH パラメータ生成の最適化

コマンドプロンプトで、コマンド 1 と 2 を入力するか、コマンド 3 を入力します。

```
1. add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )] [-dhCount <positive_integer>] [-dh ( ENABLED | DISABLED) -dhFile <string>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED)]
2. set ssl vserver <vServerName> [-sslProfile <string>]
<!--NeedCopy-->
```

```
3. set ssl vserver <vServerName> [-dh ( ENABLED | DISABLED) -dhFile <string>] [-dhCount <positive_integer>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

GUI を使用した DH パラメータ生成の最適化

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [**SSL** パラメータ] セクションで、[**DH** キー有効期限のサイズ制限を有効にする] を選択します。

暗号リダイレクト

October 7, 2021

SSL ハンドシェイク中に、SSL クライアント（通常は Web ブラウザ）は、サポートする暗号スイートを、設定済みの順序で通知します。このリストから、SSL サーバは設定された暗号の独自のリストと一致する暗号を選択します。

クライアントによってアナウンスされた暗号が SSL サーバーで構成された暗号と一致しない場合、SSL ハンドシェイクは失敗します。失敗は、ブラウザに表示される不可解なエラーメッセージによって通知されます。これらのメッセージには、エラーの正確な原因が記載されることはほとんどありません。

暗号リダイレクションを使用すると、SSL ハンドシェイクが失敗したときに正確で意味のあるエラーメッセージを配信するように SSL 仮想サーバを設定できます。SSL ハンドシェイクが失敗すると、Citrix ADC アプライアンスは以前に構成された URL にユーザーをリダイレクトします。URL が設定されていない場合は、内部で生成されたエラーページが表示されます。

CLI を使用した暗号リダイレクションの構成

コマンドプロンプトで次のコマンドを入力して、暗号リダイレクトを構成し、構成を確認します。


```
-  set ssl vserver <vServerName> -cipherRedirect < ENABLED | DISABLED>
   -cipherURL < URL>
-  show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set ssl vserver vs-ssl -cipherRedirect ENABLED -cipherURL http://
   redirectURl

Done

show ssl vserver vs-ssl

Advanced SSL configuration for VServer vs-ssl:
DH: DISABLED
Ephemeral RSA: ENABLED          Refresh Count: 1000
Session Reuse: ENABLED          Timeout: 600 seconds
Cipher Redirect: ENABLED        Redirect URL: http://redirectURl
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED
   TLSv1.2: ENABLED
    1)      CertKey Name: Auth-Cert-1       Server Certificate
    1)      Cipher Name: DEFAULT
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用した暗号リダイレクトの構成

1. **Traffic Management > Load Balancing > Virtual Servers** に移動して仮想サーバーを開きます。

2. [**SSL** パラメータ] セクションで、[暗号リダイレクトを有効にする] を選択し、リダイレクト URL を指定します。

ハードウェアとソフトウェアを使用して、**ECDHE** および **ECDSA** 暗号のパフォーマンスを向上させます

October 7, 2021

> 注:
>
> この拡張機能は、次のプラットフォームにのみ適用されます。
>
> - MPX/SDX 11000
> - MPX/SDX 14000
> - MPX 22000, MPX 24000, and MPX 25000
> - MPX/SDX 14000 FIPS

以前のリリースでは、Citrix ADC アプライアンスでの ECDHE と ECDSA の計算はハードウェア（Cavium chips）でのみ実行され、SSL セッションの数はいつでも制限されていました。この機能拡張により、一部の操作もソフトウェアで実行されます。つまり、処理はキャビウムチップと CPU コアの両方で行われ、ECDHE と ECDSA 暗号の性能が向上します。

処理は、設定されたソフトウェア暗号しきい値まで、最初にソフトウェアで実行されます。このしきい値に達すると、操作はハードウェアにオフロードされます。したがって、このハイブリッドモデルは、ハードウェアとソフトウェアの両方を使用して SSL パフォーマンスを向上させます。要件に合わせて「softwareCryptoThreshold」パラメーターを設定することで、ハイブリッドモデルを有効にできます。ハイブリッドモデルを無効にするには、このパラメータを 0 に設定します。

CPU しきい値は ECDHE および ECDSA 計算に排他的ではないため、現在の CPU 使用率が高すぎない場合に最大のメリットがあります。たとえば、アプライアンスの現在のワークロードが CPU サイクルの 50%を消費し、しきい値がに設定されている場合 80%, ECDHE および ECDSA の計算では、30%. 設定されたソフトウェア暗号しきい値 80% に達すると、さらに ECDHE および ECDSA 計算がハードウェアにオフロードされます。この場合、ハードウェアで ECDHE および ECDSA 計算を実行すると CPU サイクルが消費されるため、実際の CPU 使用率が 80% を超える可能性があります。

CLI を使用してハイブリッドモデルを有効にする

コマンドプロンプトで入力します。

```
set ssl parameter -softwareCryptoThreshold <positive_integer>

```



```
Synopsis:

softwareCryptoThreshold:

Citrix ADC CPU utilization threshold (as a percentage) beyond which
    crypto operations are not done in software. A value of zero implies
    that CPU is not utilized for doing crypto in software.

Default = 0

Min = 0

Max = 100
<!--NeedCopy-->
```

例:

```
set ssl parameter - softwareCryptoThreshold 80
Done

show ssl parameter
Advanced SSL Parameters

SSL quantum size                  : 8 KB
Max CRL memory size               : 256 MB
Strict CA checks                  : NO
Encryption trigger timeout        : 100 ms
Send Close-Notify                 : YES
Encryption trigger packet c       : 45
Deny SSL Renegotiation            : ALL
Subject/Issuer Name Insertion Format : Unicode
OCSP cache size                   : 10 MB
Push flag                         : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout   : 1 ms
Crypto Device Disable Limit       : 0
Global undef action for control policies : CLIENTAUTH
Global undef action for data policies : NOOP
Default profile                   : DISABLED
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
Disable TLS 1.1/1.2 for dynamic and VPN services : NO
Software Crypto acceleration CPU Threshold : 80
Signature and Hash Algorithms supported by TLS1.2 : ALL
```



```
27 <!--NeedCopy-->
```

GUI を使用してハイブリッドモデルを有効にする

1. [トラフィック管理] > [**SSL**] > [詳細設定 **SSL** 設定の変更] に移動します。
2. [ソフトウェア暗号しきい値 **(%)**] に値を入力します。

ECDHE 為替レートの SNMP アラームを設定します

ECDHE ベースのキー交換により、アプライアンスでの 1 秒あたりのトランザクション数が減少する可能性があります。リリース 13.0 ビルド 52.x から、ECDHE ベースのトランザクションの SNMP アラームを設定できます。このアラームでは、ECDHE 為替レートのしきい値と通常の制限を設定できます。新しいカウンター `nsssl_tot_sslInfo_ECDHE_Tx` が追加されました。このカウンターは、アプライアンスのフロントエンドとバックエンドにあるすべての ECDHE ベースのトランザクションカウンターの合計です。ECDHE ベースの鍵交換が設定された制限を超えると、SNMP トラップが送信されます。値が構成された通常の値に戻ると、別のトラップが送信されます。

CLI を使用して ECDHE 為替レートの SNMP アラームを設定します

コマンドプロンプトで入力します。

```
1 set snmp alarm ECDHE-EXCHANGE-RATE -logging ( ENABLED | DISABLED ) -
     severity <severity>
2  -state ( ENABLED | DISABLED ) -thresholdValue <positive_integer>  [-
      normalValue <positive_integer>] -time <secs>
3 <!--NeedCopy-->
```

例**:**

```
1 set snmp alarm ECDHE-EXCHANGE-RATE -logging eNABLED -severity critical
     -state eNABLED -thresholdValue 100 -normalValue 50
2 <!--NeedCopy-->
```

ECDSA 暗号の組み合わせのサポート

October 7, 2021

ECDSA 暗号スイートは、楕円曲線暗号化（ECC）を使用します。サイズが小さいため、処理能力、ストレージ領域、帯域幅、消費電力に制約がある環境で役立ちます。

ECDHE_ECDSA 暗号グループを使用する場合、サーバーの証明書に ECDSA 対応の公開鍵が含まれている必要があります。

次の表は、N3 チップ、Citrix ADC VPX アプライアンス、MPX 5900/26000、MPX/SDX 8900/15000 アプライアンスを搭載したアプライアンスでサポートされる ECDSA 暗号の一覧です。

暗号名	優先度	説明	キー交換アルゴリズム	認証アルゴリズム	暗号化アルゴリズム (鍵サイズ)	メッセージ認証コード (MAC) アルゴリズム	16 進コード
TLS1-ECDHE-ECDSA-AES128-SHA	1	SSLv3	ECC-DHE	ECDSA	AES(128)	SHA1	0xc009
TLS1-ECDHE-ECDSA-AES256-SHA	2	SSLv3	ECC-DHE	ECDSA	AES(256)	SHA1	0xc00a
TLS1.2-ECDHE-ECDSA-AES128-SHA256	3	TLSv1.2	ECC-DHE	ECDSA	AES(128)	SHA-256	0xc023
TLS1.2-ECDHE-ECDSA-AES256-SHA384	4	TLSv1.2	ECC-DHE	ECDSA	AES(256)	SHA-384	0xc024
TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256	5	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(128)	SHA-256	0xc02b

暗号名	優先度	説明	キー交換アルゴリズム	認証アルゴリズム	暗号化アルゴリズム (鍵サイズ)	メッセージ認証コード (MAC) アルゴリズム	16 進コード
TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384	6	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(256)	SHA-384	0xc02c
TLS1-ECDHE-ECDSA-RC4-SHA	7	SSLv3	ECC-DHE	ECDSA	RC4(128)	SHA1	0xc007
TLS1-ECDHE-ECDSA-DES-CBC3-SHA	8	SSLv3	ECC-DHE	ECDSA	3DES(168)	SHA1	0xc008
TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305	9	TLSv1.2	ECC-DHE	ECDSA	CHACHA20/	AEAD	cca9

ECDSA/RSA 暗号と証明書の選択

ECDSA および RSA サーバ証明書の両方を SSL 仮想サーバに同時にバインドできます。ECDSA 証明書と RSA 証明書の両方が仮想サーバーにバインドされると、クライアントに対して提示する適切なサーバー証明書が自動的に選択されます。クライアント暗号リストに RSA 暗号が含まれていても、ECDSA 暗号が含まれていない場合、仮想サーバは RSA サーバ証明書を提示します。両方の暗号がクライアントのリストに存在する場合、提示されるサーバ証明書は、仮想サーバに設定されている暗号の優先順位によって異なります。つまり、RSA のプライオリティが高い場合は、RSA 証明書が表示されます。ECDSA の優先順位が高い場合は、ECDSA 証明書がクライアントに提示されます。

ECDSA または RSA 証明書を使用したクライアント認証

クライアント認証では、仮想サーバにバインドされた CA 証明書を ECDSA または RSA 署名にすることができます。アプライアンスは、混合証明書チェーンをサポートします。たとえば、次の証明書チェーンがサポートされています。

クライアント証明書（ECDSA）<-> CA 証明書（RSA）<-> 中間証明書（RSA）<-> ルート証明書（RSA）

次の表に、ECDSA 暗号グループおよび ECDSA 証明書を使用するさまざまな Citrix ADC アプライアンスでサポートされている楕円曲線を示します。

楕円曲線	サポートされているプラットフォーム
prime256v1	FIPS を含むすべてのプラットフォーム。
secp384r1	FIPS を含むすべてのプラットフォーム。
secp521r1	MPX 5900、MPX/SDX 8900、MPX/SDX 15000、MPX/SDX 26000、VPX
secp224r1	MPX 5900、MPX/SDX 8900。MPX/SDX 15000、MPX/SDX 26000、VPX

ECDSA 証明書とキーペアの作成

CLI または GUI を使用して、Citrix ADC アプライアンスで直接 ECDSA 証明書とキーのペアを作成できます。以前は、アプライアンスに ECC 証明書とキーのペアをインストールしてバインドすることができましたが、証明書とキーのペアを作成するには OpenSSL を使用する必要がありました。

P_256 および P_384 曲線のみがサポートされています。

> 注
>
> このサポートは、MPX 9700/1050/12500/15500 を除くすべてのプラットフォームで利用できます。

CLI を使用して **ECDSA** 証明書とキーペアを作成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
create ssl ecdsaKey <keyFile> -curve ( P_256 | P_384 ) [-keyform ( DER | PEM )] [-des | -des3] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

例:


```
create ecdsaKey ec_p256.ky -curve P_256 -pkcs8
Done
create ecdsaKey ec_p384.ky -curve P_384
Done
<!--NeedCopy-->
```

GUI を使用して **ECDSA** 証明書とキーペアを作成するには、次の手順を実行します。

1. [トラフィック管理] **> [SSL] > [SSL** ファイル] **> [**キー] に移動し、[**ECDSA** キーの作成] をクリックします。
2. PKCS #8 形式でキーを作成するには、[**PKCS8**] を選択します。

ADC アプライアンスでのユーザー定義の暗号グループの構成

October 7, 2021

暗号グループは、Citrix ADC アプライアンス上の SSL 仮想サーバー、サービス、またはサービスグループにバインドする一連の暗号スイートです。暗号スイートは、プロトコル、鍵交換（Kx）アルゴリズム、認証（Au）アルゴリズム、暗号化（Enc）アルゴリズム、およびメッセージ認証コード（Mac）で構成されています。アルゴリズムを使用します。アプライアンスには、事前定義された一連の暗号グループが付属しています。SSL サービスまたは SSL サービスグループを作成すると、ALL 暗号グループが自動的にバインドされます。ただし、SSL 仮想サーバーまたは透過的 SSL サービスを作成すると、DEFAULT 暗号グループが自動的にバインドされます。また、ユーザ定義の暗号グループを作成し、SSL 仮想サーバ、サービス、またはサービスグループにバインドすることもできます。

注**:** MPX アプライアンスにライセンスがない場合は、EXPORT 暗号のみが SSL 仮想サーバー、サービス、またはサービスグループにバインドされます。

ユーザ定義の暗号グループを作成するには、まず暗号グループを作成し、次に暗号グループまたは暗号グループをこのグループにバインドします。暗号エイリアスまたは暗号グループを指定すると、暗号エイリアスまたはグループ内のすべての暗号がユーザ定義の暗号グループに追加されます。また、個々の暗号（暗号スイート）をユーザー定義グループに追加することもできます。ただし、定義済みの暗号グループは変更できません。暗号グループを削除する前に、グループ内のすべての暗号スイートのバインドを解除します。

暗号グループを SSL 仮想サーバー、サービス、またはサービスグループにバインドすると、エンティティにバインドされている既存の暗号に暗号が追加されます。エンティティに特定の暗号グループをバインドするには、まずエンティティにバインドされている暗号または暗号グループのバインドを解除する必要があります。次に、特定の暗号グループをエンティティにバインドします。たとえば、AES 暗号グループだけを SSL サービスにバインドするには、次の手順を実行します。

1. サービスの作成時にデフォルトでサービスにバインドされているデフォルトの暗号グループ ALL をバインド解除します。


```
unbind ssl service <service name> -cipherName ALL
<!--NeedCopy-->
```

2. AES 暗号グループをサービスにバインドする

```
bind ssl service <Service name> -cipherName AE
<!--NeedCopy-->
```

AES に加えて暗号グループ DES をバインドする場合は、コマンドプロンプトで次のように入力します。

```
bind ssl service <service name> -cipherName DES
<!--NeedCopy-->
```

注：無料の Citrix ADC 仮想アプライアンスは、DH 暗号グループのみをサポートします。

CLI を使用したユーザ定義の暗号グループの設定

コマンドプロンプトで、次のコマンドを入力して、暗号グループを追加するか、以前に作成したグループに暗号を追加し、設定を確認します。

```
add ssl cipher <cipherGroupName>
bind ssl cipher <cipherGroupName> -cipherName <cipherGroup/cipherName>
show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

例:

```
add ssl cipher test

Done

bind ssl cipher test -cipherName ECDHE

Done

sh ssl cipher test

1)      Cipher Name: TLS1-ECDHE-RSA-AES256-SHA  Priority : 1
```



```
12 Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1   HexCode
      =0xc014
13 2)      Cipher Name: TLS1-ECDHE-RSA-AES128-SHA  Priority : 2
14 Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1   HexCode
      =0xc013
15 3)      Cipher Name: TLS1.2-ECDHE-RSA-AES-256-SHA384    Priority : 3
16 Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA-384
      HexCode=0xc028
17 4)      Cipher Name: TLS1.2-ECDHE-RSA-AES-128-SHA256    Priority : 4
18 Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA-256
      HexCode=0xc027
19 5)      Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 5
20 Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256) Mac=AEAD
      HexCode=0xc030
21 6)      Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 6
22 Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128) Mac=AEAD
      HexCode=0xc02f
23 7)      Cipher Name: TLS1-ECDHE-ECDSA-AES256-SHA        Priority : 7
24 Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA1
      HexCode=0xc00a
25 8)      Cipher Name: TLS1-ECDHE-ECDSA-AES128-SHA        Priority : 8
26 Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA1
      HexCode=0xc009
27 9)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-SHA384   Priority : 9
28 Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA-384
      HexCode=0xc024
29 10)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-SHA256   Priority : 10
30 Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA-256
      HexCode=0xc023
31 11)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
      Priority : 11
32 Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256) Mac=AEAD
      HexCode=0xc02c
33 12)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
      Priority : 12
34 Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128) Mac=AEAD
      HexCode=0xc02b
35 13)     Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA        Priority : 13
36 Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1   HexCode
      =0xc012
37 14)     Cipher Name: TLS1-ECDHE-ECDSA-DES-CBC3-SHA      Priority : 14
38 Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=3DES(168) Mac=SHA1
      HexCode=0xc008
39 15)     Cipher Name: TLS1-ECDHE-RSA-RC4-SHA     Priority : 15
40 Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=RC4(128)  Mac=SHA1   HexCode
```



```
    =0xc011
16)     Cipher Name: TLS1-ECDHE-ECDSA-RC4-SHA   Priority : 16
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=RC4(128)  Mac=SHA1
    HexCode=0xc007
17)     Cipher Name: TLS1.2-ECDHE-RSA-CHACHA20-POLY1305 Priority : 17
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=CHACHA20/POLY1305(256) Mac
    =AEAD   HexCode=0xcca8
18)     Cipher Name: TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
    Priority : 18
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=CHACHA20/POLY1305(256)
    Mac=AEAD   HexCode=0xcca9
Done

bind ssl cipher test -cipherName TLS1-ECDHE-RSA-DES-CBC3-SHA
<!--NeedCopy-->
```

CLI を使用して暗号グループから暗号をバインド解除する

コマンドプロンプトで次のコマンドを入力して、ユーザー定義の暗号グループから暗号のバインドを解除し、設定を確認します。

```
show ssl cipher <cipherGroupName>

unbind ssl cipher <cipherGroupName> -cipherName <string>

show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

CLI を使用した暗号グループの削除

注：組み込みの暗号グループは削除できません。ユーザ定義の暗号グループを削除する前に、暗号グループが空であることを確認してください。

コマンドプロンプトで、次のコマンドを入力して、ユーザー定義の暗号グループを削除し、構成を確認します。

```
rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]
show ssl cipher <cipherGroupName>

<!--NeedCopy-->
```


例:

```
rm ssl cipher test Done

sh ssl cipher test ERROR: No such resource [cipherGroupName, test]
<!--NeedCopy-->
```

GUI を使用したユーザー定義の暗号グループの構成

1. **Traffic Management > SSL > Cipher Groups** に移動します。
2. [追加] をクリックします。
3. 暗号グループの名前を指定します。
4. [追加] をクリックして、使用可能な暗号と暗号グループを表示します。
5. 暗号または暗号グループを選択し、矢印ボタンをクリックしてそれらを追加します。
6. [作成] をクリックします。
7. [閉じる] をクリックします。

CLI を使用して、暗号グループを **SSL** 仮想サーバ、サービス、またはサービスグループにバインドするには、次の手順を実行します。

コマンドプロンプトで、次のいずれかを入力します。

```
bind ssl vserver <vServerName> -cipherName <string>

bind ssl service <serviceName> -cipherName <string>

bind ssl serviceGroup <serviceGroupName> -cipherName <string>

<!--NeedCopy-->
```

例:

```
bind ssl vserver ssl_vserver_test -cipherName test
Done

bind ssl service  nshttps -cipherName test
Done

bind ssl servicegroup  ssl_svc  -cipherName test
Done
```



```
<!--NeedCopy-->
```

GUI を使用して、暗号グループを **SSL** 仮想サーバ、サービス、またはサービスグループにバインドするには、次の手順を実行します。

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。

   サービスの場合は、仮想サーバをサービスに置き換えます。サービスグループの場合は、仮想サーバをサービスグループに置き換えます。

   仮想サーバ、サービス、またはサービスグループを開きます。

2. [詳細設定] で、[**SSL** 暗号] を選択します。
3. 仮想サーバ、サービス、またはサービスグループに暗号グループをバインドします。

SSL 仮想サーバーまたはサービスへの個々の暗号のバインド

また、暗号グループの代わりに個々の暗号を仮想サーバーまたはサービスにバインドすることもできます。

CLI を使用して暗号をバインドするには**:**
コマンドプロンプトで次のように入力します。

```
bind ssl vserver <vServerName> -cipherName <string>
bind ssl service <serviceName> -cipherName <string>
<!--NeedCopy-->
```

例**:**

```
bind ssl vserver v1 -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done

bind ssl service sslsvc -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done
<!--NeedCopy-->
```

GUI を使用して **SSL** 仮想サーバに暗号をバインドするには、次の手順を実行します。

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL 仮想サーバーを選択し、[編集] をクリックします。
3. [詳細設定] で、[**SSL** 暗号] を選択します。
4. 「暗号スイート」で、「追加」を選択します。
5. 使用可能なリストで暗号を検索し、矢印をクリックして設定済みリストに追加します。

6. [**OK**] をクリックします。
7. [完了] をクリックします。

暗号を SSL サービスにバインドするには、仮想サーバをサービスに置き換えた後、上記の手順を繰り返します。

ADC アプライアンスのサーバー証明書サポートマトリックス

January 25, 2022

リリース 13.0 ビルド 41.x から、ADC アプライアンスは、合計サイズが 32 KB 以内の場合に複数のレコードにフラグメント化されるサーバ証明書メッセージをサポートします。以前は、サポートされる最大サイズは 16 KB で、フラグメンテーションはサポートされていませんでした。

Citrix ADC アプライアンスは、次のサーバー証明書をサポートしています。

表 1: フロントエンド (FE) サービスとバックエンド (BE) サービスのサポート

サーバー証明書/プラットフォーム	MPX/SDX (N2 チップ) FE	MPX/SDX (N2 チップス) が	MPX/SDX (N3 チップス)	MPX/SDX (N3 チップス) が	VPX FE	VPX BE
MD5	☒	☒	☒	☒	☒	☒
SHA1	☒	☒	☒	☒	☒	☒
SHA224	☒	☒	☒	☒	☒	☒
SHA256	☒	☒	☒	☒	☒	☒
SHA384	☒	☒	☒	☒	☒	☒
SHA512	☒	☒	☒	☒	☒	☒
RSA キー	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット
DH キー	1024 ビットと 2048 ビット	1024 ビットと 2048 ビット	1024 ビットと 2048 ビット	1024 ビットと 2048 ビット	1024、2048、3072、および 4096 ビット	1024、2048、3072、および 4096 ビット

サーバー証明書/プラットフォーム	MPX 9700/10500/12500/15500 FIPS with FW 2.2 FE	MPX 9700/10500/12500/15500 FIPS with FW 2.2 BE	MPX/SDX 14030/14060/14080 FIPS FE	MPX/SDX 14030/14060/14080 FIPS BE
MD5	☒	☒	☒	☒
SHA1	☒	☒	☒	☒
SHA224	☒	☒	☒	☒
SHA256	☒	☒	☒	☒
SHA384	☒	☒	☒	☒
SHA512	☒	☒	☒	☒
RSA キー	2048 ビット	2048 ビット	2048 ビットと 3072 ビット	2048 ビットと 3072 ビット
DH キー	×	×	×	×

メモ

- 4k 証明書はより高い CPU サイクルを必要とし、ローエンドアプライアンスのパフォーマンスに影響を与える可能性があります。
- リリース 11.1 以前では、Citrix ADC アプライアンスは、バックエンドクライアントのハローメッセージで、RSA-MD5、RSA-SHA1、および RSA-SHA256 の「署名アルゴリズム」拡張をサポートします。Citrix ADC アプライアンスは、SHA 384 および SHA 512 署名アルゴリズム拡張をサポートしていません。そのため、Windows IIS サーバーなどの一部のサーバーでは、接続がリセットされます。
- リリース 12.0 以降、Citrix ADC アプライアンスはすべての signature_algorithms 拡張をサポートします。

クライアント認証または相互 **TLS (mTLS)**

January 25, 2022

通常の SSL トランザクションでは、セキュリティで保護された接続を介してサーバーに接続しているクライアントが、サーバーの有効性を確認します。そのためには、SSL トランザクションを開始する前にサーバーの証明書をチェックします。ただし、サーバーに接続しているクライアントを認証するようにサーバーを構成したい場合があります。

注：リリース 13.0 ビルド 41.x 以降、Citrix ADC アプライアンスは、合計サイズが 32 KB 以内であれば、複数のレコードにフラグメント化された証明書要求メッセージをサポートします。以前は、サポートされる最大サイズは 16 KB で、フラグメンテーションはサポートされていませんでした。

SSL 仮想サーバーでクライアント認証を有効にすると、Citrix ADC アプライアンスは SSL ハンドシェイク中にクライアント証明書を要求します。アプライアンスは、発行者の署名や有効期限などの通常の制約について、クライアントから提示された証明書をチェックします。

> （注）
>
> アプライアンスが発行者の署名を検証するには、クライアント証明書を発行した CA の証明書が次の条件を満たしている必要があります。
>
> - アプライアンスにインストールされている。
> - クライアントがトランザクションを行っている仮想サーバにバインドされます。

証明書が有効な場合、アプライアンスはクライアントにすべてのセキュアなリソースへのアクセスを許可します。ただし、証明書が無効な場合、アプライアンスは SSL ハンドシェイク中にクライアント要求を破棄します。

アプライアンスは、まずクライアント証明書から始まり、クライアントのルート CA 証明書（Verisign など）で終わる証明書のチェーンを作成して、クライアント証明書を検証します。ルート CA 証明書には、1 つまたは複数の中間 CA 証明書が含まれる場合があります (ルート CA がクライアント証明書を直接発行しない場合)。

Citrix ADC アプライアンスでクライアント認証を有効にする前に、有効なクライアント証明書がクライアントにインストールされていることを確認してください。次に、トランザクションを処理する仮想サーバのクライアント認証を有効にします。最後に、クライアント証明書を発行した CA の証明書をアプライアンス上の仮想サーバーにバインドします。

注： Citrix ADC MPX アプライアンスは、512 ビットから 4096 ビットの証明書とキーのペアのサイズをサポートしています。証明書は、次のいずれかのハッシュアルゴリズムを使用して署名する必要があります。

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

SDX アプライアンスでは、SSL チップが VPX インスタンスに割り当てられている場合、MPX アプライアンスの証明書とキーのペアサイズのサポートが適用されます。それ以外の場合は、VPX インスタンスの通常の証明書とキーのペアサイズのサポートが適用されます。

Citrix ADC 仮想アプライアンス（VPX インスタンス）は、次のサイズまで、512 ビット以上の証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書
- 物理サーバー上の 4096 ビット証明書

注：リリース 13.0 ビルド 79.x 以降、VPX プラットフォームでの SSL ハンドシェイク中、4096 ビット RSA クライアント証明書によるクライアント認証がサポートされています。

メモ:

- MPX FIPS の制限については、MPX FIPS の制限を参照してください。
- SDX FIPS の制限については、SDX FIPS の制限を参照してください。

クライアント証明書の提供

クライアント認証を構成する前に、有効なクライアント証明書をクライアントにインストールしておく必要があります。クライアント証明書には、Citrix ADC アプライアンスとの安全なセッションを作成する特定のクライアントシステムに関する詳細が含まれています。各クライアント証明書は一意であり、1 つのクライアントシステムでのみ使用する必要があります。

CA からクライアント証明書を取得するか、既存のクライアント証明書を使用するか、Citrix ADC アプライアンスでクライアント証明書を生成するかにかかわらず、証明書を正しい形式に変換する必要があります。Citrix ADC アプライアンスでは、証明書は PEM または DER 形式で保存され、クライアントシステムにインストールする前に PKCS #12 形式に変換する必要があります。証明書を変換してクライアントシステムに転送したら、証明書がそのシステムにインストールされ、クライアントアプリケーション用に構成されていることを確認します。Web ブラウザーなどのアプリケーションは、SSL トランザクションの一部である必要があります。

証明書を PEM または DER 形式から PKCS #12 形式に変換する方法については、「SSL ファイルのインポートと変換」を参照してください。

クライアント証明書を生成する方法については、「証明書の作成」を参照してください。

クライアント証明書ベースの認証を有効にする

デフォルトでは、Citrix ADC アプライアンスではクライアント認証が無効になっており、すべての SSL トランザクションはクライアントを認証せずに続行されます。SSL ハンドシェイクの一部として、クライアント認証をオプションまたは必須のいずれかに設定できます。

クライアント認証がオプションの場合、アプライアンスはクライアント証明書を要求しますが、クライアントが無効な証明書を提示した場合でも SSL トランザクションは続行されます。クライアント認証が必須の場合、SSL クライアントが有効な証明書を提供しない場合、アプライアンスは SSL ハンドシェイクを終了します。

注意：クライアント証明書ベースの認証チェックをオプションに変更する前に、適切なアクセス制御ポリシーを定義することをお勧めします。

注: クライアント認証は、グローバルではなく個々の SSL 仮想サーバーに対して構成されます。

CLI を使用してクライアント証明書ベースの認証を有効にする

コマンドプロンプトで次のコマンドを入力して、クライアント証明書ベースの認証を有効にし、構成を確認します。


```
set ssl vserver <vServerName> [-clientAuth (ENABLED | DISABLED)] [-
    clientCert (MANDATORY | OPTIONAL)]
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set ssl vserver vssl -clientAuth ENABLED -clientCert Mandatory
Done
show ssl vserver vssl

Advanced SSL configuration for VServer vssl:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.2: ENABLED TLSv1
    .2: ENABLED

1) CertKey Name: sslckey Server Certificate

1) Policy Name: client_cert_policy Priority: 0

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用してクライアント証明書ベースの認証を有効にする

1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、仮想サーバーを開きます。

2. [SSL パラメータ] セクションで [クライアント認証] を選択し、[クライアント証明書] リストで [必須] を選択します。

> 注:
>
> クライアント認証が必須に設定されていて、クライアント証明書にポリシー拡張が含まれていると、証明書の検証は失敗します。リリース 12.0-56.x からは、フロントエンド SSL プロファイルにパラメータを設定して、このチェックをスキップできます。このパラメーターはデフォルトでは無効になっています。つまり、このチェックはデフォルトで実行されます。

CLI を使用してクライアント認証中にポリシー拡張チェックをスキップする

コマンドプロンプトで入力します。

```
set ssl profile ns_default_ssl_profile_frontend -clientauth ENABLED -
    skipClientCertPolicyCheck ENABLED

Parameter

skipClientCertPolicyCheck

              Control policy extension check, if present inside the
                 X509 certificate chain. Applicable only if client
                 authentication is enabled and client certificate is
                 set to mandatory. Possible values functions as follows
                 :

- ENABLED: Skip the policy check during client authentication.

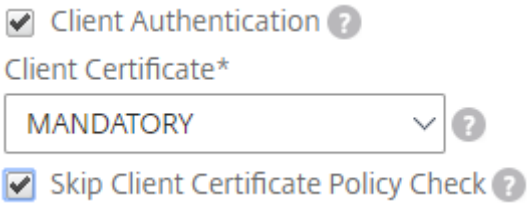
- DISABLED: Perform policy check during client authentication.

Possible values: ENABLED, DISABLED

Default: DISABLED
<!--NeedCopy-->
```

GUI を使用してクライアント認証中にポリシー拡張チェックをスキップする

1. [システム] > [プロファイル] > [**SSL** プロファイル] に移動します。
2. 新しいフロントエンドプロファイルを作成するか、既存のフロントエンドプロファイルを編集します。
3. クライアント認証が有効になっていて、クライアント証明書が必須に設定されていることを確認します。
4. [クライアント証明書ポリシーチェックをスキップ] を選択します。



CA 証明書を仮想サーバにバインドする

Citrix ADC アプライアンスに証明書が存在する CA は、クライアント認証に使用されるクライアント証明書を発行する必要があります。この証明書を、クライアント認証を実行する Citrix ADC 仮想サーバーにバインドします。

アプライアンスがクライアント証明書を検証するときに完全な証明書チェーンを形成できるように、CA 証明書を SSL 仮想サーバーにバインドします。そうしないと、証明書チェーンの形成に失敗し、証明書が有効であってもクライアントはアクセスを拒否されます。

CA 証明書は SSL 仮想サーバに任意の順序でバインドできます。アプライアンスは、クライアント証明書の検証中に正しい順序を形成します。

たとえば、クライアントが **CA_A** によって発行された証明書を提示する場合、**CA_A** は証明書が **CA_B** によって発行される中間 CA であり、証明書は信頼されたルート CA **Root_CA** によって発行されます。この証明書は、これらの 3 つの証明書はすべて、Citrix ADC アプライアンス上の仮想サーバーにバインドする必要があります。

1 つ以上の証明書を仮想サーバーにバインドする手順については、SSL 仮想サーバーへの証明書とキーのペアのバインドを参照してください。

証明書のチェーンを作成する手順については、「証明書のチェーンを作成する」を参照してください。

クライアント証明書の検証の厳密な制御

Citrix ADC アプライアンスは、単一のルート CA によって発行された場合、有効な中間 CA 証明書を受け入れます。つまり、ルート CA 証明書のみが仮想サーバーにバインドされ、そのルート CA がクライアント証明書とともに送信された中間証明書を検証すると、アプライアンスは証明書チェーンを信頼し、ハンドシェイクは成功します。

ただし、クライアントがハンドシェイクで一連の証明書を送信する場合、証明書が SSL 仮想サーバーにバインドされていない限り、CRL または OCSP レスポンダーを使用して中間証明書を検証することはできません。したがって、中間証明書の 1 つが失効しても、ハンドシェイクは成功します。ハンドシェイクの一部として、SSL 仮想サーバはバインドされている CA 証明書のリストを送信します。より厳密に制御するために、その仮想サーバーにバインドされた CA 証明書の 1 つによって署名された証明書のみを受け入れるように SSL 仮想サーバーを構成できます。これを行うには、仮想サーバーにバインドされている SSL プロファイルで **ClientAuthUseBoundCAChain** 設定を有効にする必要があります。仮想サーバーにバインドされている CA 証明書の 1 つがクライアント証明書に署名していない場合、ハンドシェイクは失敗します。

たとえば、clientcert1 と clientcert2 の 2 つのクライアント証明書が、それぞれ中間証明書 int-CA-A と int-CA-B によって署名されているとします。中間証明書は、ルート証明書 root-CA によって署名されます。int-CA-A と

ルート CA は SSL 仮想サーバにバインドされます。デフォルトの場合 (ClientAuthuseBoundCachain は無効)、clientcert1 と clientcert2 の両方が受け入れられます。ただし、ClientAuthUseBoundCachain が有効な場合、Citrix ADC アプライアンスは clientcert1 のみを受け入れます。

CLI を使用して、クライアント証明書の検証をより厳密に制御できるようにする

コマンドプロンプトで入力します。

```
set ssl profile <name> -ClientAuthUseBoundCAChain Enabled
<!--NeedCopy-->
```

GUI を使用してクライアント証明書の検証をより厳密に制御できるようにする

1. [システム] > [プロファイル] に移動し、[**SSL** プロファイル] タブを選択して SSL プロファイルを作成するか、既存のプロファイルを選択します。
2. [バインドされた **CA** チェーンを使用したクライアント認証を有効にする] を選択します。

サーバー認証

October 7, 2021

Citrix ADC アプライアンスは、Web サーバーに代わって SSL オフロードとアクセラレーションを実行するため、通常、アプライアンスは Web サーバーの証明書を認証しません。ただし、エンドツーエンドの SSL 暗号化を必要とする展開では、サーバーを認証できます。

このような状況では、アプライアンスは SSL クライアントになり、SSL サーバーとの安全なトランザクションを実行します。SSL サービスにバインドされている証明書を持つ CA がサーバー証明書に署名していることを確認し、サーバー証明書の有効性をチェックします。

サーバーを認証するには、サーバー認証を有効にし、サーバーの証明書に署名した CA の証明書を ADC アプライアンスの SSL サービスにバインドします。証明書をバインドするときは、[CA としてバインド] オプションを指定する必要があります。

サーバー証明書認証の有効化（または無効化）

CLI および GUI を使用して、サーバ証明書認証を有効または無効にできます。

CLI を使用したサーバ証明書認証の有効化（または無効化）

コマンドプロンプトで次のコマンドを入力して、サーバー証明書の認証を有効にし、構成を確認します。

```
set ssl service <serviceName> -serverAuth ( ENABLED | DISABLED )
show ssl service <serviceName>
<!--NeedCopy-->
```

例**:**

```
set ssl service ssl-service-1 -serverAuth ENABLED
<!--NeedCopy-->
```

```
show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-
                service-1:`
            DH: DISABLED
            Ephemeral RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            Server Auth: ENABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
    1)      Cipher Name: ALL
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用したサーバー証明書認証の有効化 **(**または無効化**)**

1. **Traffic Management > Load Balancing > Services** に移動して SSL サービスを開きます。
2. [SSL パラメータ] セクションで、[サーバー認証を有効にする] を選択し、[共通名] を指定します。
3. [詳細設定] で [証明書] を選択し、CA 証明書をサービスにバインドします。

CLI を使用して **CA** 証明書をサービスにバインドする

コマンドプロンプトで次のコマンドを入力して、CA 証明書をサービスにバインドし、構成を確認します。

```
bind ssl service <serviceName> -certkeyName <string> -CA

show ssl service <serviceName>
<!--NeedCopy-->
```

例:

```
bind ssl service ssl-service-1 -certkeyName samplecertkey -CA
<!--NeedCopy-->
```

```
show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-
               service-1:
            DH: DISABLED
            Ephemeral RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            Server Auth: ENABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
    1)      CertKey Name: samplecertkey     CA Certificate
       CRLCheck: Optional
    1)      Cipher Name: ALL
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

サーバー証明書認証の共通名を構成する

サーバ認証を有効にしたエンドツーエンドの暗号化では、SSL サービスまたはサービスグループの設定に共通名を含めることができます。指定した名前は、SSL ハンドシェイク中にサーバー証明書の共通名と比較されます。2 つの名前が一致すると、ハンドシェイクは成功します。

共通名が一致しない場合、サービスまたはサービスグループに指定された共通名が、証明書のサブジェクト代替名 (SAN) フィールドの値と比較されます。これらの値のいずれかに一致すると、ハンドシェイクは成功します。この設定は、たとえば、ファイアウォールの内側に 2 台のサーバがあり、一方のサーバが他方のサーバの ID をスプーフィングする場合に特に便利です。共通名がチェックされていない場合、IP アドレスが一致すれば、いずれかのサーバから提示された証明書が受け入れられます。

注**:** SAN フィールドのドメイン名、URL、電子メール ID DNS エントリのみが比較されます。

CLI を使用した **SSL** サービスまたはサービスグループの共通名検証の設定

コマンドプロンプトで次のコマンドを入力して、コモンネーム検証によるサーバー認証を指定し、構成を確認します。

1. サービスで共通名を構成するには、次のように入力します。

```
set ssl service <serviceName> -commonName <string> -serverAuth
    ENABLED
show ssl service <serviceName>
<!--NeedCopy-->
```

2. サービスグループの共通名を構成するには、次のように入力します。

```
set ssl serviceGroup <serviceGroupName> -commonName <string> -
    serverAuth ENABLED
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

例**:**

```
> set ssl service svc1 -commonName xyz.com -serverAuth ENABLED
<!--NeedCopy-->
```

```
show ssl service svc

     Advanced SSL configuration for Back-end SSL Service svc1:
     DH: DISABLED
     Ephemeral RSA: DISABLED
     Session Reuse: ENABLED Timeout: 300 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
```



```
      Server Auth: ENABLED Common Name: www.xyz.com
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SNI: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
     1) CertKey Name: cacert CA Certificate OCSPCheck: Optional
     1) Cipher Name: ALL
      Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

GUI を使用して **SSL** サービスまたはサービスグループの共通名検証を構成する

1. **Traffic Management > Load Balancing > Services** に移動するか、**Traffic Management > Load Balancing > Service Groups** に移動してサービスまたはサービスグループを開きます。
2. [SSL パラメータ] セクションで、[サーバー認証を有効にする] を選択し、共通名を指定します。

SSL アクションとポリシー

October 7, 2021

SSL ポリシーは、着信トラフィックを評価し、ルール（式）に一致する要求に事前定義されたアクションを適用します。ポリシーを作成する前にアクションを構成して、ポリシーの作成時にアクションを指定できるようにします。ポリシーを有効にするには、次のいずれかを実行します。

- ポリシーをアプライアンス上の仮想サーバーにバインドして、その仮想サーバーを流れるトラフィックにのみ適用されるようにします。
- ポリシーをグローバルにバインドして、アプライアンスを通過するすべてのトラフィックに適用されるようにします。

SSL アクションは、選択した要求に適用できる SSL 設定を定義します。1 つ以上のポリシーにアクションを関連付けます。クライアント接続要求または応答のデータは、ポリシーで指定された規則と比較され、規則に一致する接続にアクションが適用されます (式)。

クラシック式を使用してクラシックポリシーを設定し、SSL のデフォルト構文式を使用してデフォルト構文ポリシーを設定できます。

注： CLI でポリシーを構成する経験がないユーザは、通常、設定ユーティリティを使用するとかなり簡単になります。

ユーザー定義のアクションまたは組み込みアクションをデフォルトの構文ポリシーに関連付けることができます。クラシックポリシーでは、ユーザー定義アクションのみを許可します。デフォルトの構文ポリシーでは、ポリシーラベルの下にポリシーをグループ化することもできます。この場合、別のポリシーから呼び出された場合にのみ適用されます。

SSL アクションとポリシーの一般的な用途には、ディレクトリごとのクライアント認証、Outlook Web アクセスのサポート、SSL ベースのヘッダーの挿入などがあります。SSL ベースのヘッダーの挿入には、SSL 処理が Citrix ADC アプライアンスにオフロードされたサーバーに必要な SSL 設定が含まれます。

SSL ポリシー

October 7, 2021

Citrix ADC アプライアンスのポリシーは、処理する特定の接続を識別するのに役立ちます。処理は、その特定のポリシーに対して設定されているアクションに基づきます。ポリシーを作成してそのアクションを構成したら、次のいずれかを実行する必要があります。

- ポリシーをアプライアンス上の仮想サーバーにバインドして、その仮想サーバーを流れるトラフィックにのみ適用されるようにします。
- ポリシーをグローバルにバインドして、Citrix ADC アプライアンスで構成された仮想サーバーを通過するすべてのトラフィックに適用されるようにします。

Citrix ADC アプライアンスの SSL 機能は、デフォルトの構文（詳細）ポリシーをサポートします。デフォルトの構文式、それらの動作方法、およびそれらを手動で設定する方法については、ポリシーと式を参照してください。

> 注:
>
> CLI でのポリシーの設定に慣れていないユーザは、通常、設定ユーティリティの使用がかなり簡単になります。

SSL ポリシーでは、ポリシーを作成するときにアクションを指定できるように、ポリシーを作成する前にアクションを作成する必要があります。
SSL のデフォルト構文ポリシーでは、組み込みアクションを使用することもできます。組み込みアクションの詳細については、SSL 組み込みアクションとユーザー定義アクションを参照してください。

SSL のデフォルト構文ポリシー

SSL デフォルト構文ポリシー（詳細ポリシーとも呼ばれる）は、要求に対して実行される制御またはデータアクションを定義します。したがって、SSL ポリシーは、制御ポリシーおよびデータポリシーとして分類できます。

- 制御ポリシー。制御ポリシーは、クライアント認証の強制などの制御アクションを使用します。
  注意：リリース 10.5 以降では、SSL 再ネゴシエーションの拒否（denySSLReneg）はデフォルトで ALL に設定されています。ただし、CLIENTAUTH などの制御ポリシーは、再ネゴシエーションハンドシェイクをトリガーします。このようなポリシーを使用する場合は、denySSLReneg を NO に設定する必要があります。
- データポリシー。データポリシーは、リクエストにデータを挿入するなどのデータアクションを使用します。

ポリシーの重要なコンポーネントは、式とアクションです。式は、アクションが実行される要求を識別します。

組み込みアクションまたはユーザー定義アクションを使用して、デフォルトの構文ポリシーを設定できます。別のアクションを作成しなくても、組み込みアクションを使用してポリシーを設定できます。ただし、ユーザ定義のアクシ

ョンを使用してポリシーを設定するには、まずアクションを設定してから、ポリシーを設定します。

UNDEF アクションと呼ばれる追加のアクションを指定できます。このアクションは、要求に式を適用すると未定義の結果が生じた場合に実行されます。

SSL ポリシー構成

CLI および GUI を使用して、SSL デフォルト構文ポリシーを設定できます。

CLI を使用した SSL ポリシーの設定

コマンドプロンプトで入力します。

```
1 add ssl policy <name> -rule <expression> -Action <string> [-undefAction
      <string>] [-comment <string>]
2 <!--NeedCopy-->
```

GUI を使用した SSL ポリシーの構成

Traffic Management > SSL > Policies に移動して **Polices** タブで *Add* をクリックします。

SSL ポリシーと TLS1.3 プロトコルのサポート

リリース 13.0 ビルド 71.x 以降では、TLS1.3 プロトコルを使用した SSL ポリシーのサポートが追加されています。TLSv1.3 プロトコルが接続についてネゴシエートされると、クライアントから受信した TLS データを検査するポリシールールが設定されたアクションをトリガーするようになりました。

たとえば、次のポリシールールが true を返す場合、トラフィックはアクションで定義された仮想サーバーに転送されます。

```
1 add ssl action action1 -forward vserver2
2 add ssl policy pol1 -rule client.ssl.client_hello.sni.contains( “xyz” )
      -action action1
3 <!--NeedCopy-->
```

制限事項

- 制御ポリシーはサポートされていません。
- これらの操作はサポートされていません。

- DOCLIENTAUTH
- NOCLIENTAUTH
- caCertGrpName
- clientCertVerification
- ssllogProfile

SSL 組み込みアクションとユーザー定義アクション

October 7, 2021

ポリシーに組み込みアクションのみが必要な場合を除き、ポリシーを作成する前にアクションを作成する必要があります。その後、ポリシーを作成するときにアクションを指定できます。組み込みアクションには、制御アクションとデータアクションの 2 種類があります。コントロールポリシーではコントロールアクションを使用し、データポリシーではデータアクションを使用します。

組み込みの制御アクションは次のとおりです。

- DOCLIENTAUTH-クライアント証明書認証を実行します。(TLS1.3 ではサポートされていません)
- NOCLIENTAUTHH：クライアント証明書認証を実行しません。(TLS1.3 ではサポートされていません)

組み込みのデータアクションは次のとおりです。

- RESET：クライアントに RST パケットを送信して接続を閉じます。
- DROP：クライアントからのすべてのパケットをドロップします。接続は、クライアントが閉じるまで開いたままです。
- NOOP：処理を実行せずにパケットを転送します。

注： clientCertVerification や ssllogProfile など、クライアント認証に依存するアクションは、TLS1.3 プロトコルではサポートされていません。

ユーザー定義のデータアクションを作成できます。クライアント認証を有効にすると、要求を Web サーバに転送する前に、要求ヘッダーにクライアント証明書データを挿入する SSL アクションを作成できます。

ポリシー評価の結果が未定義の状態になった場合は、UNDEF アクションが実行されます。データポリシーまたは制御ポリシーのいずれの場合も、UNDEF アクションとして RESET、DROP、または NOOP を指定できます。制御ポリシーの場合は、DOCLIENTAUTH または NOCLIENTAUTH を指定することもできます。

ポリシー内の組み込みアクションの例

次の例では、クライアントが EXPORT カテゴリ以外の暗号を送信した場合、Citrix ADC アプライアンスはクライアント認証を要求します。クライアントは、トランザクションを成功させるには、有効な証明書を提供する必要があります。


```
add ssl policy pol1 -rule CLIENT.SSL.CIPHER_EXPORTABLE.NOT -reqAction
    DOCLIENTAUTH
<!--NeedCopy-->
```

次の例では、クライアント認証が有効であることを前提としています。

ユーザによって提供された証明書のバージョンが、ポリシーのバージョンと一致する場合、アクションは実行されず、パケットが転送されます。

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction NOOP
<!--NeedCopy-->
```

ユーザーによって提供された証明書のバージョンが、ポリシーのバージョンと一致する場合、接続は切断されます。

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction DROP
<!--NeedCopy-->
```

ユーザーによって提供された証明書のバージョンが、ポリシーのバージョンと一致する場合、接続はリセットされます。

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction RESET
<!--NeedCopy-->
```

ポリシーベースのクライアント認証によるクライアント証明書の検証

ポリシーベースのクライアント認証を設定している場合は、クライアント証明書の検証を「必須」または「オプション」に設定できます。デフォルトは必須です。

CLI を使用してクライアント証明書の検証をオプションに設定する

コマンドプロンプトで入力します。

```
add ssl action <name> ((-clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) [-
    clientCertVerification ( Mandatory | Optional )]
```



```
<!--NeedCopy-->
```

例**:**

```
add ssl action sslact -clientauth DOCLIENTAUTH -clientcertverification
    OPTIONAL
<!--NeedCopy-->
```

GUI を使用してクライアント証明書の検証をオプションに設定する

1. [**Traffic Management**] **>** [**SSL**] **>** [**Policies**] に移動します。
2. [**SSL** アクション] タブで、[追加] をクリックします。
3. 名前を指定し、[クライアント証明書の検証] リストで [オプション] を選択します。

ユーザ定義の **SSL** アクション

組み込みアクションに加えて、展開に応じて他の SSL アクションを構成することもできます。これらのアクションをユーザー定義アクションと呼びます。

CLI を使用したユーザ定義の **SSL** アクションの設定

コマンドプロンプトで次のコマンドを入力して、アクションを構成し、構成を確認します。

```
add SSL action <name> -clientAuth(DOCLIENTAUTH | NOCLIENTAUTH) -
    clientCert (ENABLED | DISABLED) certHeader <string> -clientHeader <
    string> -clientCertSerialNumber (ENABLED | DISABLED) -
    certSerialHeader <string> -clientCertSubject (ENABLED | DISABLED) -
    certSubjectHeader <string> -clientCertHash (ENABLED | DISABLED) -
    certHashHeader <string> -clientCertIssuer (ENABLED | DISABLED) -
    certIssuerHeader <string> -sessionID (ENABLED | DISABLED) -
    sessionIDheader <string> -cipher (ENABLED | DISABLED) -cipherHeader
    <string> -clientCertNotBefore (ENABLED | DISABLED) -
    certNotBeforeHeader <string> -clientCertNotAfter (ENABLED | DISABLED
    ) -certNotAfterHeader <string> -OWASupport (ENABLED | DISABLED)
<!--NeedCopy-->
```



```
show ssl action [<name>]
<!--NeedCopy-->
```

例:

```
add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X
   -Client-Cert"
<!--NeedCopy-->
```

```
show ssl action Action-SSL-ClientCert

1)      Name: Action-SSL-ClientCert
        Data Insertion Action:
        Cert Header: ENABLED            Cert Tag: X-Client-Cert
Done
<!--NeedCopy-->
```

GUI を使用したユーザ定義の **SSL** アクションの設定

[トラフィック管理] **> [SSL] > [**ポリシー] に移動し、[アクション] タブで [追加] をクリックします。

クライアントトラフィックを別の仮想サーバーに転送するための **SSL** アクションの設定

管理者は、SSL のオフロードを回避するために、SSL 仮想サーバーで受信したクライアントトラフィックを別の仮想サーバーに転送するように SSL アクションを設定できます。または、ADC アプライアンスの接続を終了させる場合に使用します。この仮想サーバーの種類は、SSL、TCP、または SSL_BRIDGE です。たとえば、管理者は、次のいずれかの場合に接続を終了する代わりに、別の仮想サーバーに要求を転送してさらにアクションを実行できます。

- アプライアンスには証明書がありません。
- アプライアンスは特定の暗号をサポートしていません。

上記を実現するために、新しいバインドポイント「CLIENTHELLO_REQ」を追加して、クライアントハローを受信したときにクライアントトラフィックを評価します。86379 クライアントトラフィックを受信する仮想サーバにバインドされたポリシーが、クライアント hello の解析後に true と評価された場合、トラフィックは別の仮想サーバに転送されます。この仮想サーバーのタイプが SSL の場合は、ハンドシェイクを実行します。この仮想サーバーのタイプが TCP または SSL_BRIDGE の場合、バックエンドサーバーはハンドシェイクを実行します。

リリース 12.1-49.x では、CLIENTHELLO_REQ バインドポイントではフォワードアクションとリセットアクションだけがサポートされています。次の式接頭辞を使用できます。

- CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE
- CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION
- CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE
- CLIENT.SSL.CLIENT_HELLO.IS_REUSE
- CLIENT.SSL.CLIENT_HELLO.IS_SCSV
- CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET
- CLIENT.SSL.CLIENT_HELLO.LENGTH
- CLIENT.SSL.CLIENT_HELLO.SNI
- CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPROTOCOL (13.0 ビルド 61.x から)

これらのプレフィクスの説明については、「高度なポリシー式:SSL の解析」を参照してください。

`add SSL action`コマンドにパラメータ`forward`が追加され、新しいバインドポイント`CLIENTHELLO_REQ`が`bind ssl vserver`コマンドに追加されます。

CLI を使用した設定

コマンドプロンプトで入力します。

```
add ssl action <name> -forward <virtual server name>

add ssl policy <name> -rule <expression> -action <string>

bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type <type>
<!--NeedCopy-->
```

例:

```
add ssl action act1 -forward v2

add ssl policy pol1 -rule client.ssl.client_hello.ciphers.has_hexcode(0
   x002f) -action act1

bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

GUI を使用した設定

[**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。

SSL アクションの作成:

1. [**SSL** アクション] で、[追加] をクリックします。
2.「**SSL** アクションの作成」で、アクションの名前を指定します。
3.「転送アクション仮想サーバー」で、既存の仮想サーバーを選択するか、トラフィックを転送する新しい仮想サーバーを追加します。
4. 必要に応じて、他のパラメータを設定します。
5. [作成] をクリックします。

SSL ポリシーを作成します。

1. [**SSL** ポリシー] で、[追加] をクリックします。
2.「**SSL** ポリシーの作成」で、ポリシーの名前を指定します。
3.「アクション」で、以前に作成したアクションを選択します。
4. 式エディタで、評価するルールを入力します。
5. [作成] をクリックします。

仮想サーバーおよびバインドポリシーを作成または追加します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを追加または選択します。
3. [詳細設定] で、[**SSL** ポリシー] をクリックします。
4. [SSL ポリシー] セクションをクリックします。
5. [ポリシーの選択] で、前に作成したポリシーを選択します。
6.「ポリシー・バインディング」で、ポリシーの優先度を指定します。
7.「タイプ」で「**CLIENTHELLO_REQ**」を選択します。
8. [バインド] をクリックします。
9. [完了] をクリックします。

最も一般的なユースケースのエンドツーエンド構成については、次のトピックを参照してください。

- アプライアンスにドメイン固有（SNI）証明書がない場合は、クライアントトラフィックを転送するように SSL アクションを設定します。
- クライアント hello メッセージの ALPN 拡張のプロトコルに基づいてクライアントトラフィックを転送するように SSL アクションを設定します。
- ADC で暗号がサポートされていない場合に、クライアントトラフィックを転送するように SSL アクションを設定します。

クライアント認証用に **SNI** に基づいて **CA** を選択的に選択する **SSL** アクション

SSL 仮想サーバーにバインドされたすべての CA のリストではなく、クライアント証明書要求の SNI (ドメイン) に基づく CA のリストのみを送信できます。たとえば、クライアント hello を受信すると、SSL ポリシー表現に基づく CA 証明書（SNI など）だけが送信されます。特定の一連の証明書を送信するには、CA 証明書グループを作成する

必要があります。次に、このグループを SSL アクションにバインドし、アクションを SSL ポリシーにバインドします。クライアントトラフィックを受信する仮想サーバにバインドされたポリシーが、クライアント hello の解析後に true と評価された場合、特定の CA 証明書グループだけがクライアント要求証明書で送信されます。

以前は、CA 証明書を SSL 仮想サーバーにバインドする必要がありました。この機能拡張により、CA 証明書グループを追加し、SSL アクションに関連付けることができます。

注： SSL 仮想サーバーでクライアント認証と SNI を有効にします。正しい SNI 証明書を仮想サーバーにバインドします。

次の手順を実行します。

1. CA 証明書グループを追加します。
2. 証明書とキーのペアを追加します。
3. 証明書とキーのペアをこのグループにバインドします。
4. SSL アクションを追加します。
5. SSL ポリシーを追加します。ポリシーでアクションを指定します。
6. ポリシーを SSL 仮想サーバーにバインドします。バインドポイントを「クライアントセル」(CLIENTHELLO_REQ) として指定します。

CLI を使用した設定

コマンドプロンプトで、次のコマンドを順番に入力します。

```
add ssl caCertGroup <caCertGroupName>
add ssl certkey <certkey_name> -cert <cert> -key <key>
bind ssl caCertGroup <caCertGroupName> <certkey_name>
add ssl action <name> -caCertGrpName <string>
add ssl policy <name> -rule <expression> -action <string>
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

例**:**

```
add ssl cacertGroup ca_cert_group

add ssl certkey ca_certkey1 -cert cacert1 -key cakey1
add ssl certkey ca_certkey2 -cert cacert2 -key cakey2
add ssl certkey snicert -cert snicert -key snikey
```



```
6
7 bind ssl cacertGroup ca_cert_group ca_certkey1
8 bind ssl caCertGroup ca_cert_group ca_certkey2
9 <!--NeedCopy-->
```

```
1 sh ssl caCertGroup ca_cert_group
2
3 CA GROUP NAME:     ca_cert_group
4 ACTIONS REFERRING: 1
5
6 1) CertKey Name: ca_certkey1   CA Certificate   CRLCheck: Optional
     CA_Name Sent
7 2) CertKey Name: ca_certkey2   CA Certificate   CRLCheck: Optional
     CA_Name Sent
8 <!--NeedCopy-->
```

```
1 add ssl action pick_ca_group -cacertGrpName ca_cert_group
2 <!--NeedCopy-->
```

```
1 sh ssl action pick_ca_group
2 1) Name: pick_ca_group
3    Type: Data Insertion
4    PickCaCertGroup: ca_cert_group
5    Hits: 0
6    Undef Hits: 0
7    Action Reference Count: 1
8 <!--NeedCopy-->
```

```
1 add ssl policy snipolicy -rule client.ssl.client_hello.sni.contains("
     abc") -action pick_ca_group
2 bind ssl vserver v_SSL -policyName snipolicy -type CLIENTHELLO_REQ -
     priority 10
3 <!--NeedCopy-->
```

```
1 sh ssl policy snipolicy
```



```
    Name: snipolicy
    Rule: client.ssl.client_hello.sni.contains(”abc”)
    Action: pick_ca_group
    UndefAction: Use Global
    Hits: 0
    Undef Hits: 0


    Policy is bound to following entities
1)  Bound to: CLIENTHELLO_REQ VSERVER v_SSL
    Priority: 10
<!--NeedCopy-->
```

```
set ssl vserver v_SSL -clientauth ENABLED -SNIEnable ENABLED
bind ssl vserver v_SSL -certkeyName snicert -sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v_SSL

    Advanced SSL configuration for VServer v_SSL:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA:
       ENABLED   Refresh Count: 0
    Session Reuse: ENABLED   Timeout: 120 seconds
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    ClearText Port: 0
    Client Auth: ENABLED   Client Cert Required: Mandatory
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: ENABLED
    OCSP Stapling: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED
         TLSv1.2: ENABLED  TLSv1.3: DISABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
```



```
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1

    ECC Curve: P_256, P_384, P_224, P_521

1)  CertKey Name: snicert   Server Certificate for SNI


    Data policy
1)  Policy Name: snipolicy  Priority: 10



1)  Cipher Name: DEFAULT
    Description: Default cipher list with encryption strength >= 128bit
<!--NeedCopy-->
```

GUI を使用した設定

CA 証明書グループを作成し、そのグループに証明書をバインドします。

1. [トラフィック管理] **>** **[SSL]** **>** **[CA** 証明書グループ] に移動します。
2. [追加] をクリックし、グループの名前を指定します。
3. [作成] をクリックします。
4. **CA** 証明書グループを選択し、[バインドの表示] をクリックします。
5. [バインド] をクリックします。
6. [**CA** 証明書のバインド] ページで、既存の証明書を選択するか、[追加] をクリックして新しい証明書を追加します。
7. [選択] をクリックし、[バインド] をクリックします。
8. 別の証明書をバインドするには、手順 5 ～7 を繰り返します。
9. [閉じる] をクリックします。

[**Traffic Management**] **>** [**SSL**] **>** [**Policies**] に移動します。

SSL アクションの作成:

1. [**SSL** アクション] で、[追加] をクリックします。
2. 「**SSL** アクションの作成」で、アクションの名前を指定します。
3. 「転送アクション仮想サーバー」で、既存の仮想サーバーを選択するか、トラフィックを転送する仮想サーバーを追加します。
4. 必要に応じて、他のパラメータを設定します。
5. [作成] をクリックします。

SSL ポリシーを作成します。

1. [**SSL** ポリシー] で、[追加] をクリックします。
2.「**SSL** ポリシーの作成」で、ポリシーの名前を指定します。
3.「アクション」で、前に作成したアクションを選択します。
4. 式エディタで、評価するルールを入力します。
5.［作成］をクリックします。

仮想サーバーおよびバインドポリシーを作成または追加します。

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
2. 仮想サーバーを追加または選択します。
3. [詳細設定**]** で、[**SSL** ポリシー] をクリックします。
4. [SSL ポリシー] セクションをクリックします。
5. [ポリシーの選択] で、前に作成したポリシーを選択します。
6.「ポリシー・バインディング」で、ポリシーの優先度を指定します。
7.「タイプ」で「**CLIENTHELLO_REQ**」を選択します。
8.［バインド］をクリックします。
9.［完了］をクリックします。

GUI を使用して **CA** 証明書グループをバインド解除する

1. [トラフィック管理**] > [SSL] > [CA** 証明書グループ] に移動します。
2. 証明書グループを選択し、[バインドの表示] をクリックします。
3. グループから削除する証明書を選択し、[バインド解除] をクリックします。
4. 確認メッセージが表示されたら、[** はい] をクリックします。••。
5.［閉じる］をクリックします。

GUI を使用して **CA** 証明書グループを削除する

1. [トラフィック管理**] > [SSL] > [CA** 証明書グループ] に移動します。
2. 証明書グループを選択し、[**Delete**] をクリックします。
3. 確認メッセージが表示されたら、[はい] をクリックします。

SSL ポリシーバインディング

October 7, 2021

SSL ポリシーは、グローバルに、または SSL 仮想サーバにのみバインドできます。グローバルにバインドされたポリシーは、サービス、仮想サーバー、またはその他の Citrix ADC バインドポイントにバインドされたすべてのポリシーが評価された後に評価されます。着信データが SSL ポリシーで設定された規則のいずれかに一致する場合、ポリシーがトリガーされ、ポリシーに関連付けられたアクションが実行されます。

SSL ポリシーを仮想サーバーにバインドする場合は、次のいずれかのバインドポイントを選択する必要があります。

- REQUEST (デフォルトのバインドポイント。ポリシー評価は、SSL ハンドシェイクの完了後に HTTP レイヤーで行われます)。
- INTERCEPT_REQ_REQ（このオプションは、Citrix Secure Web Gateway のセットアップに適用されます。詳細については、「SSL インターセプション用の SSL ポリシーインフラストラクチャ」を参照してください。
- CLIENTHELLO.REQ

同様に、仮想サーバからポリシーをバインド解除する場合は、バインドポイントを指定する必要があります。

CLIENTHELLO_REQ をバインドポイントとして指定すると、クライアントの hello メッセージを受信したときにポリシーが評価されます。許可されるアクションは、RESET、FORWARD、および`caCertGrpName`です。reset アクションは、接続を終了します。転送アクションは、要求をロードバランシング仮想サーバに転送して処理します。`caCertGrpName` アクションは、クライアント認証のために SNI に基づいて CA を選択的に選択します。SSL アクションの詳細については、SSL 組み込みアクションとユーザー定義アクションを参照してください。

注：アクション caCertGrpName は、TLS1.3 プロトコルではサポートされていません。

CLI を使用して SSL ポリシーをグローバルにバインドする

コマンドプロンプトで次のコマンドを入力して、グローバル SSL ポリシーをバインドし、構成を確認します。

```
bind ssl global - policyName <string> [- priority <positive_integer>]
show ssl global
<!--NeedCopy-->
```

例:

```
bind ssl global -policyName Policy-SSL-2 -priority 90
Done

sh ssl global

      1) Name: Policy-SSL-2 Priority: 90
      2) Name: Policy-SSL-1 Priority: 100
      Done
<!--NeedCopy-->
```


GUI を使用して SSL ポリシーをグローバルにバインドする

1. **Traffic Management** > **SSL** > **Policies** に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックします。
3. [**SSL** ポリシーをグローバルにバインド/バインド解除] ダイアログボックスで、[ポリシーの挿入] をクリックします。
4. [ポリシー名] リストで、ポリシーを選択します。
5. 必要に応じて、エントリをポリシーバンク内の新しい位置にドラッグして、優先度レベルを自動的に更新します。
6. [**OK**] をクリックします。ポリシーが正常にバインドされたことを示すメッセージがステータスバーに表示されます。

CLI を使用して SSL ポリシーを仮想サーバーにバインドまたはバインド解除する

コマンドプロンプトで次のコマンドを入力して、SSL ポリシーを仮想サーバーにバインドし、構成を確認します。

```
bind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

unbind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

<!--NeedCopy-->
```

例:

```
bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
unbind ssl vserver v1 -policyName pol1 -priority 1 -type
    CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
show ssl vserver vs-server

Advanced SSL configuration for VServer vs-server:

```



```
DH: DISABLED

Ephemeral RSA: ENABLED          Refresh Count: 1000

Session Reuse: ENABLED          Timeout: 120 seconds

Cipher Redirect: DISABLED

SSLv2 Redirect: DISABLED

ClearText Port: 80

Client Auth: DISABLED

SSL Redirect: ENABLED

SSL-REDIRECT Port Rewrite: ENABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Policy Name: ssl-policy-1        Priority: 10

1)      Cipher Name: DEFAULT

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーを仮想サーバーにバインドする

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。
2. [詳細設定] で [**SSL** ポリシー] を選択します。[**SSL policy**] セクションをクリックして、ポリシーを仮想サーバにバインドします。
3. [ポリシーバインディング] ページで、既存のポリシーを選択するか、新しいポリシーを追加します。
4. ポリシーのプライオリティとタイプ（バインドポイント）を指定します。
5. [バインド] を選択します。
6. [完了] を選択します。

SSL ポリシーラベル

October 7, 2021

ポリシーラベルはポリシーの所有者です。ポリシー・ラベルは、別のポリシーから呼び出すことができるポリシーバンク（ポリシーバンク）と呼ばれるポリシーのグループを管理するのに役立ちます。SSL ポリシーラベルは、ポリシーラベルに含まれるポリシーのタイプに応じて、コントロールラベルまたはデータラベルにすることができます。データポリシーラベルにはデータポリシーのみを追加し、コントロールポリシーラベルにはコントロールポリシーのみを追加できます。ポリシーバンクを作成するには、ラベルにポリシーをバインドし、ポリシーラベルのポリシーバンク内の他のポリシーと比較して、各ポリシーの評価順序を指定します。CLI で、2 つのコマンドを入力して、ポリシーラベルを作成し、ポリシーラベルにポリシーをバインドします。設定ユーティリティでは、ダイアログボックスからオプションを選択します。

注：タイプ制御のポリシーラベルは、TLS1.3 プロトコルではサポートされていません。

CLI を使用して SSL ポリシーラベルを作成し、ラベルにポリシーをバインドする

コマンドプロンプトで入力します。

```
add ssl policylabel <labelName> -type ( CONTROL | DATA )

bind ssl policylabel <labelName> <policyName> <priority> [<
   gotoPriorityExpression>] [-invoke (<labelType> <labelName>) ]
<!--NeedCopy-->
```

例:

```
add ssl policylabel cpl1 -type CONTROL

add ssl policylabel dpl1 -type DATA

bind ssl policylabel cpl1 -policyName ctrlpol -priority 1

bind ssl policylabel dpl1 -policyName datapol -priority 1
<!--NeedCopy-->
```

GUI を使用して SSL ポリシーラベルを構成し、ラベルにポリシーをバインドする

Traffic Management > SSL > Policy Labels に移動して SSL ポリシーラベルを構成します。

選択的な **SSL** ロギング

October 7, 2021

数千台の仮想サーバで構成される大規模な展開では、SSL 関連のすべての情報がログに記録されます。以前は、いくつかの重要な仮想サーバーのクライアント認証と SSL ハンドシェイクの成功と失敗をフィルタリングすることは容易ではありませんでした。ログ全体を調べてこの情報を取得することは、インフラストラクチャがログをフィルタリングする制御を提供していないため、時間がかかり、面倒な作業でした。これで、特定の仮想サーバまたは仮想サーバのグループについて、`ns.log`に SSL 関連情報を記録できるようになりました。この情報は、特に障害のデバッグに役立ちます。この情報をログに記録するには、SSL ログプロファイルを追加する必要があります。

正常なクライアント認証については、このページの最後の ns.log 出力例を参照してください。

重要： syslog ログレベルを DEBUG に設定します。コマンドプロンプトで入力します。

```
set audit syslogParams -logLevel DEBUG
```

SSL ログプロファイル

SSL ログプロファイルは、仮想サーバーまたは仮想サーバーのグループについて、次のイベントのロギングを制御できます。

- クライアント認証の成功と失敗、または失敗のみ。
- SSL ハンドシェイクの成功と失敗、または失敗のみ。

デフォルトでは、すべてのパラメータは無効になっています。

SSL ログプロファイルは、SSL プロファイルまたは SSL アクションで設定できます。SSL プロファイルが設定されている場合、クライアント認証と SSL ハンドシェイクの成功と失敗の両方の情報をログに記録できます。SSL アクションに設定すると、ポリシーが評価される前にハンドシェイクが完了するため、クライアント認証の成功と失敗の情報のみをログに記録できます。

SSL ログプロファイルを設定しない場合でも、クライアント認証と SSL ハンドシェイクの成功と失敗が記録されます。ただし、選択ログは、SSL ログプロファイルが使用されている場合にのみ可能です。

> 注:
>
> SSL ログプロファイルは、高可用性およびクラスタのセットアップでサポートされます。

CLI を使用した **SSL** ログプロファイルの追加

コマンドプロンプトで入力します。


```
add ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )] [-
    ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
    DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

パラメーター:

名前**:**

SSL ログプロファイルの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。プロファイルの作成後は変更できません。

Name は必須の引数です。最大長: 127

sslLogClAuth:

すべてのクライアント認証イベントをログに記録します。成功イベントと失敗イベントの両方が含まれます。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

ssllogClAuthFailures:

すべてのクライアント認証失敗イベントをログに記録します。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

sslLogHS:

SSL ハンドシェイク関連のイベントをすべてログに記録します。成功イベントと失敗イベントの両方が含まれます。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

sslLogHSfailures:

SSL ハンドシェイク関連のすべての失敗イベントをログに記録します。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

例**:**

```
> add ssl logprofile ssllog10 -sslLogClAuth ENABLED -sslLogHS ENABLED

 Done
```



```
4
5 sh ssllogprofile ssllog10
6
7 1)       Name: ssllog10
8
9         SSL log ClientAuth [Success/Failures] : ENABLED
10
11        SSL log ClientAuth [Failures] : DISABLED
12
13        SSL log Handshake [Success/Failures] : ENABLED
14
15        SSL log Handshake [Failures] : DISABLED
16
17  Done
18 <!--NeedCopy-->
```

GUI を使用した **SSL** ログプロファイルの追加

[システム] **>** [プロファイル] **>** [**SSL** ログプロファイル] に移動し、プロファイルを追加します。

CLI を使用した **SSL** ログプロファイルの変更

コマンドプロンプトで次のように入力します。

```
1 set ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )][-
     ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
     DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
2 <!--NeedCopy-->
```

例**:**

```
1 set ssllogprofile ssllog10 -ssllogClAuth en -ssllogClAuthFailures en -
     ssllogHS en -ssllogHSfailures en
2
3 Done
4
5 sh ssllogprofile ssllog10
6
7     1)            Name: ssllog10
8
9                     SSL log ClientAuth [Success/Failures] : ENABLED
```



```
10                      SSL log ClientAuth [Failures] : ENABLED
11                      SSL log Handshake [Success/Failures] : ENABLED
12                      SSL log Handshake [Failures] : ENABLED
13      Done
14  <!--NeedCopy-->
```

GUI を使用した **SSL** ログプロファイルの変更

1. **[システム] > [プロファイル] > [SSL** ログプロファイル] に移動し、プロファイルを選択して [編集] をクリックします。
2. 変更を行い、[**OK**] をクリックします。

CLI を使用してすべての **SSL** ログプロファイルを表示する

コマンドプロンプトで入力します。

```
1  sh ssl logprofile
2  <!--NeedCopy-->
```

例**:**

```
1  sh ssl logprofile
2
3      1)            Name: ssllogp1
4                      SSL log ClientAuth [Success/Failures] : ENABLED
5                      SSL log ClientAuth [Failures] : ENABLED
6                      SSL log Handshake [Success/Failures] : DISABLED
7                      SSL log Handshake [Failures] : ENABLED
8
9      2)            Name: ssllogp2
10                     SSL log ClientAuth [Success/Failures] : DISABLED
11                     SSL log ClientAuth [Failures] : DISABLED
12                     SSL log Handshake [Success/Failures] : DISABLED
13                     SSL log Handshake [Failures] : DISABLED
14
15     3)            Name: ssllogp3
16                     SSL log ClientAuth [Success/Failures] : DISABLED
17                     SSL log ClientAuth [Failures] : DISABLED
18                     SSL log Handshake [Success/Failures] : DISABLED
19                     SSL log Handshake [Failures] : DISABLED
```



```
20
21      4)              Name: ssllog10
22                        SSL log ClientAuth [Success/Failures] : ENABLED
23                        SSL log ClientAuth [Failures] : ENABLED
24                        SSL log Handshake [Success/Failures] : ENABLED
25                        SSL log Handshake [Failures] : ENABLED
26  Done
27  <!--NeedCopy-->
```

GUI を使用してすべての **SSL** ログプロファイルを表示する

[システム] **>** [プロファイル] **>** [**SSL** ログプロファイル] に移動します。すべてのプロファイルが一覧表示されます。

SSL ログプロファイルへの **SSL** ログプロファイルのアタッチ

SSL プロファイルの作成時に SSL ログプロファイルに SSL ログプロファイルをアタッチ（設定）することも、後で SSL プロファイルを編集することもできます。クライアント認証とハンドシェイクの成功と失敗の両方を記録できます。

> 重要:
>
> SSL ログプロファイルをアタッチする前に、デフォルトの SSL プロファイルを有効にする必要があります。

CLI を使用して **SSL** プロファイルに **SSL** ログプロファイルをアタッチする

コマンドプロンプトで入力します。

```
1  set ssl profile <name> [-ssllogProfile <string>]
2  <!--NeedCopy-->
```

例**:**

```
1  set ssl profile fron_1 -ssllogProfile ssllog10
2  <!--NeedCopy-->
```

GUI を使用して **SSL** ログプロファイルに **SSL** ログプロファイルをアタッチする

1. [システム] **>** [プロファイル] **>** [**SSL** プロファイル] に移動します。
2. 「編集」をクリックし、「**SSL** ログ・プロファイル」でプロファイルを指定します。

SSL ログプロファイルを SSL アクションにアタッチする

SSL ログプロファイルを設定できるのは、SSL アクションの作成時だけです。SSL アクションを変更してログプロファイルを設定することはできません。アクションをポリシーに関連付けます。ログに記録できるのは、クライアント認証の成功と失敗だけです。

CLI を使用して SSL ログプロファイルを SSL アクションにアタッチする

コマンドプロンプトで入力します。

```
add ssl action <name> -clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) -
    ssllogProfile <string>
<!--NeedCopy-->
```

例:

```
> add ssl action act1 -clientAuth DoCLIENTAUTH -ssllogProfile ssllog10

Done

> sh ssl action act1

    1)            Name: act1
                    Type: Client Authentication (DOCLIENTAUTH)
                    Hits: 0
                    Undef Hits: 0
                    Action Reference Count: 0
                    SSLlogProfile: ssllog10
Done
<!--NeedCopy-->
```

GUI を使用して SSL ログプロファイルを SSL アクションにアタッチする

1. [トラフィック管理] > [SSL] > [ポリシー] に移動し、[SSL アクション] をクリックします。
2. [追加] をクリックします。
3. [クライアント認証] で、[ENABLED] を選択します。
4. [SSL ログプロファイル] で、リストからプロファイルを選択するか、[+] をクリックしてプロファイルを作成します。
5. [作成] をクリックします。

ログファイルからのサンプル出力

次に、成功したクライアント認証のns.logからのログ出力例を示します。

```
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 158 0 :  SPCBId 671 - ClientIP 10.102.1.98 - ClientPort 49451 - VserverServiceIP 10.102.57.82 - VserverServicePort 443 - ClientVersion TLSv1.2 - CipherSuite ”AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit” - Session New - CLIENT_AUTHENTICATED -SerialNumber ”2A” - SignatureAlgorithm ”sha1WithRSAEncryption” - ValidFrom ”Sep 22 09:15:20 2008 GMT” - ValidTo ”Feb  8 09:15:20 2036 GMT” - HandshakeTime 10 ms
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 159 0 :  SPCBId 671 - IssuerName ” C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix”
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 160 0 :  SPCBId 671 - SubjectName ” C=IN,ST=KAR,O=Citrix Pvt Ltd,OU=A,CN=B”
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 161 0 :  Backend SPCBId 674 - ServerIP 10.102.57.85 - ServerPort 443 - ProtocolVersion TLSv1.2 - CipherSuite ”AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit” - Session Reuse - SERVER_AUTHENTICATED -SerialNumber ”3E” - SignatureAlgorithm ”sha1WithRSAEncryption” - ValidFrom ”Sep 24 06:40:37 2008 GMT” - ValidTo ”Feb 10 06:40:37 2036 GMT” - HandshakeTime 1 ms
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 162 0 :  SPCBId 674 - IssuerName ” C=IN,ST=KAR,O=Citrix Pvt Ltd”
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 163 0 :  SPCBId 674 - SubjectName ” C=IN,ST=P,L=Q,O=R”
<!--NeedCopy-->
```

DTLS プロトコルのサポート

February 21, 2022

> 注:
>
> - DTLSv1.0 プロトコルは、Citrix ADC MPX/SDX（N2 および N3 ベース）、VPX、および MPX 14000

FIPS アプライアンスでサポートされています。外部 HSM ではサポートされていません。

- DTLS 1.0 プロトコルは、インテル Coleto SSL チップ（リリース 12.1 ビルド 50.x 以降）を搭載した Citrix ADC アプライアンスでサポートされています。
- DTLSv1.2 プロトコルは、Citrix ADC VPX アプライアンス（リリース 13.0 ビルド 47.x 以降）のフロントエンドでサポートされています。
- DTLS 1.2 プロトコルは、Intel Coleto SSL チップ（リリース 13.0 ビルド 52.x 以降）を含む Citrix ADC アプライアンスのフロントエンドでサポートされています。インテル Coleto SSL チップを搭載したプラットフォームの詳細については、「インテル Coleto SSL チップベースのプラットフォームのサポート」を参照してください。
- DTLS タイプのサービスグループはサポートされません。
- DTLSv1.2 プロトコルは、Citrix ADC MPX（N3 ベース）アプライアンスのフロントエンドでサポートされています（リリース 13.0 ビルド 58.x 以降）。
- Citrix Gateway の Enlightened Data Transport（EDT）サポートの詳細については、「HDX 啓発データトランスポートのサポート」を参照してください。
- リリース 13.0 ビルド 79.x から DTLS プロファイルに変更が加えられました。詳細については、DTLS プロファイルを参照してください。
- リリース 13.0 ビルド 82.x から、DTLS セッションで受信した不良 MAC レコードを無視する新しいパラメータ`maxBadmacIgnorecount`が DTLS プロファイルに追加されました。詳細については、DTLS プロファイルを参照してください。
- サポートされているプラットフォームとビルドについて詳しくは、「Citrix ADC MPX ハードウェアとソフトウェアの互換性マトリックス」を参照してください。

SSL プロトコルと TLS プロトコルは、従来、ストリーミングトラフィックの保護に使用されてきました。これらのプロトコルはどちらも TCP をベースにしており、これは低速です。また、TLS は、損失または並べ替えられたパケットを処理できません。

UDP は、Lync、Skype、iTunes、YouTube、トレーニングビデオ、フラッシュなどのオーディオおよびビデオアプリケーションに適したプロトコルです。ただし、UDP は安全でも信頼性もありません。DTLS プロトコルは、UDP を介してデータを保護するように設計されており、メディアストリーミング、VOIP、通信用のオンラインゲームなどのアプリケーションに使用されます。DTLS では、各ハンドシェイクメッセージには、そのハンドシェイク内の特定のシーケンス番号が割り当てられます。ピアは、ハンドシェイクメッセージを受信すると、そのメッセージが次に予想されるメッセージであるかどうかを迅速に判断できます。そうであれば、ピアはメッセージを処理します。そうでない場合、メッセージは以前のすべてのメッセージを受信した後に処理のためにキューに入れられます。

DTLS 仮想サーバーと UDP タイプのサービスを作成します。デフォルトでは、DTLS プロファイル (ns-dtls_default_profile) は仮想サーバーにバインドされます。オプションで、ユーザー定義の DTLS プロファイルを作成し、仮想サーバーにバインドできます。

注:RC4 暗号は DTLS 仮想サーバーではサポートされていません。

DTLS 構成

コマンドライン（CLI）または構成ユーティリティ（GUI）を使用して、ADC アプライアンスで DTLS を設定できます。

注: リリース 13.0 ビルド 47.x から、DTLS 1.2 プロトコルは Citrix ADC VPX アプライアンスのフロントエンドでサポートされています。DTLSv1.2 仮想サーバーを構成するときに、DTLS12 を指定します。デフォルトは DTLS1 です。

コマンドプロンプトで入力します。

```
set ssl vserver DTLS [-dtls1 ( ENABLED | DISABLED )] [-dtls12 ( ENABLED | DISABLED )]
```

CLI を使用して **DTLS** 設定を作成する

コマンドプロンプトで入力します。

```
1 add lb vserver <vserver_name> DTLS <IPAddress>  <port>
2 add service  <service_name>  <IPAddress> UDP 443
3 bind lb vserver  <vserver_name>  <udp_service_name>
4 <!--NeedCopy-->
```

次の手順はオプションです。

```
1 add dtlsProfile dtls-profile -maxretryTime <positive_integer>
2 set ssl vserver <vserver_name> -dtlsProfileName <dtls_profile_name>
3 <!--NeedCopy-->
```

GUI を使用して **DTLS** 設定を作成する

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. DTLS タイプの仮想サーバーを作成し、UDP サービスを仮想サーバーにバインドします。
3. デフォルトの DTLS プロファイルは DTLS 仮想サーバーにバインドされます。別のプロファイルをバインドするには、[SSL パラメータ] で別の DTLS プロファイルを選択します。プロファイルを作成するには、[DTLS プロファイル] の横のプラス (+) をクリックします。

DTLS 仮想サーバでの **SNI** のサポート

SNI の詳細については、「複数のサイトのセキュアなホスティングのための SNI 仮想サーバーを構成する」を参照してください。

CLI を使用して **DTLS** 仮想サーバで **SNI** を構成する

コマンドプロンプトで入力します。


```
set ssl vserver <vServerName> -SNIEnable ENABLED
bind ssl vserver <vServerName> -certkeyName <string> -SNICert
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
set ssl vserver v1 -sniEnable ENABLED
bind ssl vserver  v1 -certkeyName san2 -sniCert
bind ssl vserver  v1 -certkeyName san13  - sniCert
bind ssl vserver  v1 -certkeyName san17  - sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v1

Advanced SSL configuration for VServer v1:
DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
Ephemeral RSA: ENABLED
Refresh Count: 0
Session Reuse: ENABLED
Timeout: 1800 seconds
Cipher Redirect: DISABLED

ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: ENABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
DTLSv1: ENABLED
Send Close-Notify: YES
Strict Sig-Digest Check: DISABLED
Zero RTT Early Data: DISABLED
DHE Key Exchange With PSK: NO
Tickets Per Authentication Context: 1

DTLS profile name: nsdtls_default_profile
```



```
29
30 ECC Curve: P_256, P_384, P_224, P_521
31
32 1) CertKey Name: ca
33 CA Certificate  OCSPCheck: OptionalCA_Name Sent
34 2) CertKey Name: san2 Server Certificate for SNI
35 3) CertKey Name: san17 Server Certificate for SNI
36 4) CertKey Name: san13 Server Certificate for SNI
37
38
39 1) Cipher Name: DEFAULT
40 Description: Default cipher list with encryption strength >= 128bit
41 Done
42 <!--NeedCopy-->
```

GUI を使用した **DTLS** 仮想サーバでの **SNI** の設定

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. DTLS 仮想サーバーを開き、[証明書] で [サーバー証明書] をクリックします。
3. 証明書を追加するか、リストから証明書を選択して、[**SNI** のサーバー証明書] を選択します。
4. [詳細設定] で、[**SSL** パラメータ] をクリックします。
5. [**SNI** 有効] を選択します。

DTLS 仮想サーバーでサポートされていない機能

次のオプションは、DTLS 仮想サーバーでは有効にできません。

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- プッシュ暗号化トリガー
- SSLv2Redirect
- SSLv2URL

DTLS 仮想サーバーによって使用されないパラメータ

DTLS 仮想サーバーは、設定されている場合でも、次の SSL パラメータを無視します。

- 暗号化トリガーパケット数
- プッシュ暗号化トリガータイムアウト

- SSL 量子サイズ
- 暗号化トリガータイムアウト
- 件名/発行者名の挿入形式

DTLS サービスでの再ネゴシエーションの設定

非セキュア再ネゴシエーションは、DTLS サービスでサポートされています。CLI または GUI を使用して、この設定を構成できます。

CLI を使用した DTLS サービスでの再ネゴシエーションの設定

コマンドプロンプトで入力します。

```
set ssl parameter -denysslreneg NONSECURE
<!--NeedCopy-->
```

例:

```
set ssl parameter -denysslreneg NONSECURE


sh ssl parameter
Advanced SSL Parameters
-----------------------
SSL quantum size                                    : 8 KB
Max CRL memory size                                 : 256 MB
Strict CA checks                                    : NO
Encryption trigger timeout                          : 100 ms
Send Close-Notify                                   : YES
Encryption trigger packet count                     : 45
Deny SSL Renegotiation                              : NONSECURE
Subject/Issuer Name Insertion Format                : Unicode
OCSP cache size                                     : 10 MB
Push flag                                           : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout                     : 1 ms
Crypto Device Disable Limit                         : 0
Global undef action for control policies            : CLIENTAUTH
Global undef action for data policies               : NOOP
Default profile                                     : DISABLED
SSL Insert Space in Certificate Header              : YES
```



```
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
Disable TLS 1.1/1.2 for dynamic and VPN services      : NO
Software Crypto acceleration CPU Threshold            : 0
Hybrid FIPS Mode                                      : DISABLED
Signature and Hash Algorithms supported by TLS1.2     : ALL
SSL Interception Error Learning and Caching           : DISABLED
SSL Interception Maximum Error Cache Memory           : 0 Bytes
Done
<!--NeedCopy-->
```

GUI を使用した **DTLS** サービスでの再ネゴシエーションの設定

1. [**Traffic Management**] **>** [**Load Balancing**] **>** [**Services**] の順に移動します。
2. DTLS サービスを選択し、[編集（Edit）] をクリックします。
3. [**SSL**] > [詳細設定**]** に移動します。
4. [**SSL** 再ネゴシエーションを拒否する] を選択します。

DTLS サービスでサポートされていない機能

次のオプションは、DTLS サービスでは有効にできません。

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- プッシュ暗号化トリガー
- SSLv2Redirect
- sslv2URL
- SNI
- 安全な再ネゴシエーション

DTLS サービスによって使用されないパラメータ

DTLS サービスは、設定されている場合でも、次の SSL パラメータを無視します。

- 暗号化トリガーパケット数
- プッシュ暗号化トリガータイムアウト
- SSL 量子サイズ
- 暗号化トリガータイムアウト
- 件名/発行者名の挿入形式

> 注:
>
> DTLS サービスでセッションの再利用が現在サポートされていないため、SSL セッション再利用ハンドシェイクは DTLS サービスで失敗します。
>
> 回避策: DTLS サービスでセッションの再利用を手動で無効にします。CLI で、次のように入力します。
>
> `set ssl service <dtls-service-name> -sessReuse DISABLED`

DTLS プロファイル

デフォルト設定の DTLS プロファイルは、自動的に DTLS 仮想サーバーにバインドされます。ただし、要件に合わせて特定の設定で DTLS プロファイルを作成できます。

DTLS 仮想サーバーまたは VPN DTLS 仮想サーバーで DTLS プロファイルを使用します。DTLS 仮想サーバでは SSL プロファイルを使用できません。

> 注:
>
> MTU およびパケットサイズの変更に基づいて DTLS プロファイルの最大レコードサイズ設定を変更して下さい。たとえば、デフォルトの最大レコードサイズ 1459 バイトは、IPv4 アドレスヘッダーサイズに基づいて計算されます。IPv6 レコードでは、ヘッダーサイズが大きくなります。したがって、次の基準を満たすには、最大レコードサイズを小さくする必要があります。

```
max record size + UDP header(8bytes)+ IP header size < MTU
```

例:

```
Default DTLS profile
     1) Name: nsdtls_default_profile
     PMTU Discovery: DISABLED
     Max Record Size: 1459 bytes
     Max Retry Time: 3 sec
     Hello Verify Request: ENABLED
     Terminate Session: DISABLED
     Max Packet Count: 120 bytes

Custom DTLS profile
     1) Name: ns_dtls_profile_ipv6_1
     PMTU Discovery: DISABLED
     Max Record Size: 1450 bytes
     Max Retry Time: 3 sec
     Hello Verify Request: ENABLED
     Terminate Session: DISABLED
     Max Packet Count: 120 bytes
<!--NeedCopy-->
```


CLI を使用した DTLS プロファイルの作成

> 注:
> リリース 13.0 ビルド 79.x から、DTLS プロファイルの変更点は次のとおりです。
>
> - helloverifyrequestこのパラメーターはデフォルトで有効になっています。このパラメータを有効にすると、攻撃者またはボットがネットワークのスループットを圧倒するリスクを軽減し、アウトバウンド帯域幅の枯渇につながる可能性があります。つまり、DTLS DDoS 増幅攻撃を軽減するのに役立ちます。
> - maxHoldQlenパラメータが追加されます。このパラメーターは、DTLS レイヤーで処理するためにキューに入れられるデータグラムの数を定義します。UDP 多重化が高い UDP トラフィックを送信している場合、maxHoldQlen パラメータの値が大きいと、DTLS レイヤでメモリが蓄積する可能性があります。したがって、低い値を設定することをお勧めします。最小値は 32、最大値は 65535、デフォルト値は 32 です。
>
> リリース 13.0 ビルド 82.x から、DTLS セッションで受信した不良 MAC レコードを無視する新しいパラメータmaxBadmacIgnorecountが DTLS プロファイルに追加されました。このパラメータを使用すると、パラメータに設定された値までの不良レコードは無視されます。アプライアンスは、制限に達した後にのみ、セッションを終了し、アラートを送信します。
> このパラメーター設定は、terminateSession パラメーターが有効な場合にのみ有効です。

```
ssl dtlsProfile <name> -maxRetryTime <positive_integer> -
    helloVerifyRequest ( ENABLED | DISABLED ) -terminateSession (ENABLED
     | DISABLED ) -maxHoldQLen <positive_integer> -maxBadmacIgnorecount
    <positive_integer>

helloVerifyRequest
               Send a Hello Verify request to validate the client.
               Possible values: ENABLED, DISABLED
               Default value: ENABLED

terminateSession
               Terminate the session if the message authentication code
                   (MAC)
               of the client and server do not match.
               Possible values: ENABLED, DISABLED
               Default value: DISABLED

maxHoldQLen
               Maximum number of datagrams that can be queued at DTLS
                   layer for
               processing
               Default value: 32
```



```
                Minimum value: 32
                Maximum value: 65535

maxBadmacIgnorecount
                Maximum number of bad MAC errors to ignore for a
                    connection prior disconnect. Disabling parameter
                    terminateSession
terminates session immediately when bad MAC is detected in the
    connection.
                Default value: 100
                Minimum value: 1
                Maximum value: 65535
<!--NeedCopy-->
```

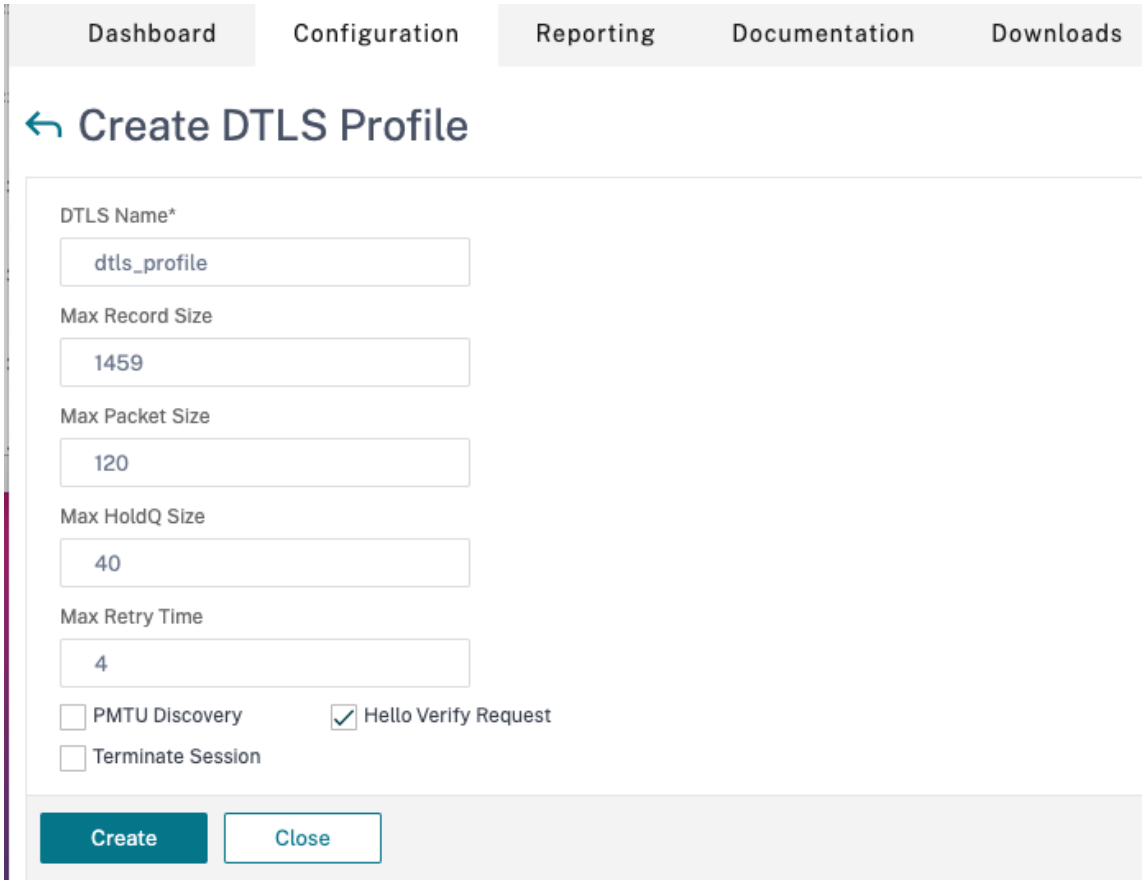
例:

```
> add ssl dtlsprofile dtls_profile -maxRetryTime 4 -helloVerifyRequest
    ENABLED -terminateSession ENABLED -maxHoldQLen 40 -
    maxBadmacIgnorecount 150
 Done
> sh dtlsprofile dtls_profile
1)  Name: dtls_profile
    PMTU Discovery: DISABLED
    Max Record Size: 1459 bytes
    Max Retry Time: 4 sec
    Hello Verify Request: ENABLED
    Terminate Session: ENABLED
    Max Packet Count: 120 bytes
    Max HoldQ Size: 40 datagrams
    Max bad-MAC Ignore Count: 150

 Done
<!--NeedCopy-->
```

GUI を使用した **DTLS** プロファイルの作成

1. [システム] **>** [プロファイル] **>** [**DTLS** プロファイル] に移動し、[追加] をクリックします。
2. [**DTLS** プロファイルの作成] ページで、さまざまなパラメータの値を入力します。



3. [**Create**] をクリックします。

エンドツーエンド **DTLS** 設定の例

```
enable ns feature SSL LB

add server s1 198.51.100.2

en ns mode usnip

add service svc_dtls s1 DTLS 443

add lb vserver v1 DTLS 10.102.59.244 443

bind ssl vserver v1 -ciphername ALL

add ssl certkey servercert -cert servercert_aia_valid.pem -key
    serverkey_aia.pem

bind ssl vserver v1 -certkeyname servercert
```



```
16
17 bind lb vserver lb1 svc_dtls
18
19 sh lb vserver v1
20
21                     v1 (10.102.59.244:4433) - DTLS     Type: ADDRESS
22                     State: UP
23                     Last state change was at Fri Apr 27 07:00:27 2018
24                     Time since last state change: 0 days, 00:00:04.810
25                     Effective State: UP
26                     Client Idle Timeout: 120 sec
27                     Down state flush: ENABLED
28                     Disable Primary Vserver On Down : DISABLED
29                     Appflow logging: ENABLED
30                     No. of Bound Services :  1 (Total) 0 (Active)
31                     Configured Method: LEASTCONNECTION
32                     Current Method: Round Robin, Reason: A new service
                          is bound         BackupMethod: ROUNDROBIN
33                     Mode: IP
34                     Persistence: NONE
35                     L2Conn: OFF
36                     Skip Persistency: None
37                     Listen Policy: NONE
38                     IcmpResponse: PASSIVE
39                     RHIstate: PASSIVE
40                     New Service Startup Request Rate: 0 PER_SECOND,
                          Increment Interval: 0
41                     Mac mode Retain Vlan: DISABLED
42                     DBS_LB: DISABLED
43                     Process Local: DISABLED
44                     Traffic Domain: 0
45                     TROFS Persistence honored: ENABLED
46                     Retain Connections on Cluster: NO
47
48     1) svc_dtls (10.102.59.190: 4433) - DTLS State: UP  Weight: 1
49 Done
50
51
52 sh ssl vserver v1
53
54                     Advanced SSL configuration for VServer v1:
55                     DH: DISABLED
56                     DH Private-Key Exponent Size Limit: DISABLED
                                    Ephemeral RSA: ENABLED
                                                          Refresh Count: 0
```



```
                      Session Reuse: ENABLED                  Timeout:
                         1800 seconds
                      Cipher Redirect: DISABLED
                      ClearText Port: 0
                      Client Auth: DISABLED
                      SSL Redirect: DISABLED
                      Non FIPS Ciphers: DISABLED
                      SNI: DISABLED
                      OCSP Stapling: DISABLED
                      HSTS: DISABLED
                      HSTS IncludeSubDomains: NO
                      HSTS Max-Age: 0
                      DTLSv1: ENABLED
                      Send Close-Notify: YES
                      Strict Sig-Digest Check: DISABLED
                      Zero RTT Early Data: DISABLED
                      DHE Key Exchange With PSK: NO
                      Tickets Per Authentication Context: 1
                      DTLS profile name: nsdtls_default_profile

                      ECC Curve: P_256, P_384, P_224, P_521

    1)            CertKey Name: servercert               Server
       Certificate

    1)            Cipher Name: DEFAULT
                    Description: Default cipher list with encryption
                       strength >= 128bit

    2)            Cipher Name: ALL
                    Description: All ciphers supported by NetScaler,
                       excluding NULL ciphers
     Done

sh service svc_dtls

                      svc_dtls (10.102.59.190:4433) - DTLS
                      State: UP
                      Last state change was at Fri Apr 27 07:00:26 2018
                      Time since last state change: 0 days, 00:00:22.790
                      Server Name: s1
                      Server ID : None                  Monitor Threshold
                         : 0
                      Max Conn: 0        Max Req: 0           Max
                         Bandwidth: 0 kbits
```



```
                        Use Source IP: NO
                        Client Keepalive(CKA): NO
                        Access Down Service: NO
                        TCP Buffering(TCPB): NO
                        HTTP Compression(CMP): NO
                        Idle timeout: Client: 120 sec           Server: 120
                            sec
                        Client IP: DISABLED
                        Cacheable: NO
                        SC: OFF
                        SP: OFF
                        Down state flush: ENABLED
                        Monitor Connection Close : NONE
                        Appflow logging: ENABLED
                        Process Local: DISABLED
                        Traffic Domain: 0

    1)            Monitor Name: ping-default
                                      State: UP               Weight: 1
                                                          Passive: 0
                                    Probes: 5               Failed [Total
                                       : 0 Current: 0]
                                     Last response: Success - ICMP echo
                                        reply received.
                          Response Time: 2.77 millisec
     Done

sh ssl service svc_dtls

                        Advanced SSL configuration for Back-end SSL Service
                            svc_dtls:
                        DH: DISABLED
                        DH Private-Key Exponent Size Limit: DISABLED
                                  Ephemeral RSA: DISABLED
                        Session Reuse: ENABLED                  Timeout:
                           1800 seconds
                        Cipher Redirect: DISABLED
                        ClearText Port: 0
                        Server Auth: DISABLED
                        SSL Redirect: DISABLED
                        Non FIPS Ciphers: DISABLED
                        SNI: DISABLED
                        OCSP Stapling: DISABLED
                        DTLSv1: ENABLED
                        Send Close-Notify: YES
```



```
                      Strict Sig-Digest Check: DISABLED
                      Zero RTT Early Data: ???
                      DHE Key Exchange With PSK: ???
                      Tickets Per Authentication Context: ???
                      DTLS profile name: nsdtls_default_profile
                      ECC Curve: P_256, P_384, P_224, P_521
    1)            Cipher Name: DEFAULT_BACKEND
                      Description: Default cipher list for Backend SSL
                         session
     Done


> sh dtlsProfile nsdtls_default_profile
1)  Name: nsdtls_default_profile
    PMTU Discovery: DISABLED
    Max Record Size: 1459 bytes
    Max Retry Time: 3 sec
    Hello Verify Request: DISABLED
    Terminate Session: ENABLED
    Max Packet Count: 120 bytes
    Max HoldQ Size: 32 datagrams
    Max bad-MAC Ignore Count: 10

 Done
<!--NeedCopy-->
```

IPv6 アドレスの DTLS サポート

DTLS は IPv6 アドレスでもサポートされています。ただし、IPv6 アドレスで DTLS を使用するには、DTLS プロファイルで最大レコードサイズを調整する必要があります。

最大レコードサイズにデフォルト値を使用すると、初期 DTLS 接続が失敗することがあります。DTLS プロファイルを使用して最大レコードサイズを調整します。

DTLS 暗号サポート

デフォルトでは、DTLS 仮想サーバーまたはサービスの作成時に DTLS 暗号グループがバインドされます。DEFAULT_DTLS には、フロントエンド DTLS エンティティがサポートする暗号が含まれています。このグループは、DTLS 仮想サーバーの作成時にデフォルトでバインドされます。DEFAULT_DTLS_BACKEND には、バックエンド DTLS エンティティでサポートされる暗号が含まれています。このグループは、デフォルトでは DTLS バックエンドサービスにバインドされています。DTLS_FIPS には、Citrix ADC FIPS プラットフォームでサポートされている暗号が含まれています。このグループは、デフォルトでは、FIPS プラットフォームで作成された DTLS 仮想サーバーまたはサービスにバインドされます。

Citrix ADC VPX、**MPX/SDX**（**N2** および **N3** ベース）アプライアンスでの **DTLS** 暗号のサポート

テーブルの読み方**:**

ビルド番号が指定されていない限り、暗号スイートはリリースのすべてのビルドでサポートされます。

例**:**

- **10.5**、**11.0**、**11.1**、**12.0**、**12.1**、**13.0**: 10.5、11.0、11.1、12.0、12.1、13.0 リリースのすべてのビルド。
- **-NA-**: 該当なし。

Citrix ADC VPX、MPX/SDX（N2 および N3 ベース）アプライアンスでの DTLS 暗号のサポート

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (フロントエンド)	サポートされているビルド (バックエンド)
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AI	11.0, 11.1, 12.0, 12.1, 13.0	12.0, 12.1, 13.0
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	11.0、11.1、12.0、12.1、13.0	12.0、12.1、13.0
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_DI	11.0、11.1、12.0、12.1、13.0	-なし-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	11.0、11.1、12.0、12.1、13.0	12.0、12.1、13.0
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WI	11.0、11.1、12.0、12.1、13.0	-なし-
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	11.0、11.1、12.0、12.1、13.0	-なし-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_	12.1, 13.0	12.1、13.0
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1、13.0	13.0
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_	12.1、13.0	-なし-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1、13.0	12.1、13.0

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (フロントエンド)	サポートされているビルド (バックエンド)
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WI	12.1、13.0	12.1、13.0

フロントエンドでサポートされているデフォルトの暗号の一覧を表示するには、コマンドプロンプトで次のように入力します。

```
show ssl cipher DEFAULT_DTLS
1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0039
6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x0033
7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```

バックエンドでサポートされているデフォルトの暗号の一覧を表示するには、コマンドプロンプトで次のように入力します。


```
show ssl cipher DEFAULT_DTLS_BACKEND
1)  Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0039
6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x0033
7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```

Citrix ADC MPX 14000 FIPS プラットフォームでの DTLS 暗号のサポート

注: 次の条件が満たされる場合、FIPS プラットフォームでエンライトンドデータサポート（EDT）がサポートされます。

- StoreFront で設定された UDT MSS の値は 900 です。
- Windows クライアントのバージョンは 4.12 以降です。
- DTLS が有効な VDA のバージョンは 7.17 以降です。
- 非 DTLS VDA のバージョンは 7.15 LTSR CU3 以降である。

テーブルの読み方**:**

ビルド番号が指定されていない限り、暗号スイートはリリースのすべてのビルドでサポートされます。

例**:**

- **10.5**、**11.0**、**11.1**、**12.0**、**12.1**、**13.0**: 10.5、11.0、11.1、12.0、12.1、13.0 リリースのすべてのビルド。
- **-NA-**: 該当なし。

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (フロントエンド)	サポートされているビルド (バックエンド)
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AI	11.0, 11.1, 12.0, 12.1–49.x, 13.0	12.0, 12.1–49.x, 13.0
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	11.0、11.1、12.0、12.1—49.x、13.0	12.0、12.1—49.x、13.0
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_DI	11.0、11.1、12.0、12.1—49.x、13.0	-なし-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	11.0、11.1、12.0、12.1—49.x、13.0	12.0、12.1—49.x、13.0
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WI	11.0、11.1、12.0、12.1—49.x、13.0	-なし-
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	11.0、11.1、12.0、12.1—49.x、13.0	-なし-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_	12.1–49.x, 13.0	12.1—49.x、13.0
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1—49.x、13.0	12.1—49.x、13.0
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_	12.1—49.x、13.0	-なし-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1—49.x、13.0	12.1—49.x、13.0
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WI	12.1—49.x、13.0	12.1—49.x、13.0

Citrix ADC FIPS アプライアンスでサポートされているデフォルトの暗号の一覧を表示するには、コマンドプロンプトで次のように入力します。

```
1 show ssl cipher DTLS_FIPS
2 1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
```



```
3      Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
          HexCode=0x0035
4  2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
5      Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
          HexCode=0x002f
6  3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
7      Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
          HexCode=0xc014
8  4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
9      Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
          HexCode=0xc013
10 5)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA  Priority : 5
11     Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
          HexCode=0xc012
12 6)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 6
13     Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
          HexCode=0x000a
14 <!--NeedCopy-->
```

フロントエンド **VPX** アプライアンス、**MPX/SDX**（**Coleto** および **N3** ベース）アプライアンスでの **DTLSv1.2** 暗号サポート

次の表に、DTLSv1.2 プロトコルでサポートされる追加の暗号を示します。

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (VPX フロントエンド)	サポートされているビルド (Coleto ベース)	ビルドサポート (N3 ベース)
TLS1.2-AES256-GCM-SHA384	0x009D	TLS_RSA_WITH	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-AES128-GCM-SHA256	0x009C	TLS_RSA_WITH_AES_128_GCM_SHA256	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384	0xc030	TLS_ECDHE_RS	13.0—47.x	13.0–52.x	13.0—58.x

暗号スイート名	16 進コード	Wireshark 暗号スイート名	サポートされているビルド (VPX フロントエンド)	サポートされているビルド (Coleto ベース)	ビルドサポート (N3 ベース)
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256	0xc02f	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-DHE-RSA-AES256-GCM-SHA384	0x009f	TLS_DHE_RSA_	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-DHE-RSA-AES128-GCM-SHA256	0x009e	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-AES-256-SHA256	0x003d	TLS_RSA_WITH	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-AES-128-SHA256	0x003C	TLS_RSA_WITH_AES_128_CBC_SHA256	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-ECDHE-RSA-AES-256-SHA384	0xc028	TLS_ECDHE_RS	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-ECDHE-RSA-AES-128-SHA256	0xc027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-DHE-RSA-AES-256-SHA256	0x006b	TLS_DHE_RSA_	13.0—47.x	13.0–52.x	13.0—58.x
TLS1.2-DHE-RSA-AES-128-SHA256	0x0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	13.0—47.x	13.0–52.x	13.0—58.x

インテル **Coleto SSL** チップ・ベースのプラットフォームのサポート

October 7, 2021

次のアプライアンスには、インテル Coleto チップが同梱されています。

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 15000
- MPX/SDX 15000-50G
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPX/SDX 26000-100G

「show hardware」コマンドを使用して、アプライアンスに Coleto（COL）チップがあるかどうかを識別します。

```
> sh hardware

    Platform: NSMPX-8900 8\*CPU+4\*F1X+6\*E1K+1\*E1K+1*COL 8955 30010
    Manufactured on: 10/18/2016
    CPU: 2100MHZ
    Host Id: 0
    Serial no: CRAC5CR8UA
    Encoded serial no: CRAC5CR8UA
 Done
<!--NeedCopy-->
```

注：セキュアな再ネゴシエーションは、これらのプラットフォームのバックエンドでサポートされています。

制限事項:

- DH 512 暗号はサポートされていません。
- SSLv3 プロトコルはサポートされていません。
- ハードウェアセキュリティモジュール（HSM）はサポートされていません。
- GnuTLS はサポートされていません。
- ECC 曲線 P_224 と P521 と ECDSA 証明書はサポートされていません (また、キャビウムチップを持つプラットフォームではサポートされていません。)
- DNSSEC オフロードはサポートされていません。(DNSSEC はソフトウェアでサポートされていますが、ハードウェアへのオフロードはサポートされていません)。

Citrix ADC MPX プラットフォームでの SSL チップ使用率の表示

リリース 13.0 ビルド 47.x 以降、Intel Coleto チップに同梱されている MPX プラットフォームでの SSL チップの使用率を表示できます。この機能は、SDX プラットフォームおよび MPX クラスターではサポートされていません。

コマンドプロンプトで入力します。

```
> stat ssl


SSL Summary


# SSL cards present                                4

# SSL cards UP                                     4

SSL engine status                                  1

SSL sessions (Rate)                              0

SSL Crypto Utilization Asym (%)                    67

SSL Crypto Utilization Symm (%)                    19
<!--NeedCopy-->
```

MPX 14000 FIPS アプライアンス

April 25, 2022

> 重要:
>
> - MPX 9700/10500/12500/15500 FIPS プラットフォームは終焉を迎えている。
> - NetScaler MPX 14000FIPS および NetScalerMPX の構成手順 9700/10500/12500/15500 FIPS アプライアンスは異なります。MPX 14000FIPS アプライアンスはファームウェア v2.2 を使用しません。MPX 9700 プラットフォームのハードウェアセキュリティモジュール (HSM) で作成された FIPS キーは、MPX 14000 プラットフォームの HSM に転送できません。もう一方のラウンドもサポートされていません。ただし、RSA キーを FIPS キーとしてインポートした場合は、RSA キーを MPX 14000 プラットフォームにコピーできます。次に、FIPS キーとしてインポートします。2048 ビットと 3072 ビットのキーのみがサポートされています。

> - Citrix ADC ダウンロードページの「Citrix ADC リリース 12.1-FIPS」および「Citrix ADC リリース 12.1-ndCPP」にリストされているファームウェアのバージョンは、MPX 14000 FIPS または SDX 14000 FIPS プラットフォームではサポートされていません。これらのプラットフォームでは、ダウンロードページで入手できる他の最新の Citrix ADC ファームウェアバージョンを使用できます。

FIPS アプライアンスには、改ざん防止（改ざん防止）暗号化モジュール（Cavium CNN3560-NFBE-G）が装備されており、FIPS 140-2 レベル 3 仕様（リリース 12.0 ビルド 56.x 以降）に準拠するように設計されています。クリティカルセキュリティパラメータ（CSP）は、主にサーバーの秘密鍵であり、HSM とも呼ばれる暗号化モジュール内に安全に格納され、生成されます。CSP は、HSM の境界外でアクセスされることはありません。スーパーユーザー (`nsroot`) だけが、HSM 内に格納されているキーに対して操作を実行できます。

FIPS アプライアンスを設定する前に、FIPS カードの状態を確認し、カードを初期化する必要があります。FIPS キーとサーバー証明書を作成し、追加の SSL 構成を追加します。

サポートされている FIPS 暗号の詳細については、「FIPS 承認アルゴリズムと暗号」を参照してください。

HA セットアップでの FIPS アプライアンスの構成の詳細については、HA セットアップのアプライアンスでの FIPS の構成を参照してください。

制限事項

1. SSLv3 プロトコルを使用した SSL 再ネゴシエーションは、MPX FIPS アプライアンスのバックエンドではサポートされません。
2. 1024 ビットおよび 4096 ビットのキーと指数値 3 はサポートされていません。
3. 4096 ビットサーバー証明書はサポートされていません。
4. 4096 ビットのクライアント証明書はサポートされていません (バックエンドサーバーでクライアント認証が有効になっている場合)。

HSM の構成

MPX 14000 FIPS アプライアンスで HSM を構成する前に、FIPS カードの状態をチェックして、ドライバが正しくロードされていることを確認します。次に、カードを初期化します。

コマンドプロンプトで入力します。

```
1 show fips
2
3 FIPS Card is not configured
4
5 <!--NeedCopy-->
```

ドライバが正しくロードされていない場合は、「エラー：操作が許可されていません-システムに FIPS カードがありません」というメッセージが表示されます。

FIPS カードの初期化

FIPS カードを適切に初期化するには、アプライアンスを 3 回再起動する必要があります。

> 重要
>
> - アプライアンスで/nsconfig/fipsディレクトリが正常に作成されたことを確認します。
> - アプライアンスを 3 回目に再起動する前に、設定を保存しないでください。

FIPS カードを初期化するには、次の手順に従います。

1. FIPS カードをリセットします (reset fips)。
2. アプライアンスを再起動します (reboot)。
3. パーティション 0 と 1 にはセキュリティ担当者のパスワードを設定し、パーティション (set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS
) にはユーザーパスワードを設定します。

   Note: The set or reset command takes more than 60 seconds to run.
4. 設定を保存します (saveconfig)。
5. メインパーティション (master_pek.key) のパスワードで暗号化されたキーが/nsconfig/fips/ディレクトリに作成されていることを確認します。
6. アプライアンスを再起動します (reboot)。
7. デフォルトパーティション (default_pek.key) のパスワードで暗号化されたキーが/nsconfig/fips/ディレクトリに作成されていることを確認します。
8. アプライアンスを再起動します (reboot)。
9. FIPS カードが稼働していることを確認します（show fips)。

CLI を使用して FIPS カードを初期化します

set fipsコマンドは、FIPS カードのハードウェアセキュリティモジュール (HSM) を初期化し、新しいセキュリティ担当者のパスワードとユーザーパスワードを設定します。

注意：このコマンドは、FIPS カード上のすべてのデータを消去します。コマンドの実行を続行する前に、プロンプトが表示されます。変更を適用するには、このコマンドの実行前と実行後に再起動が必要です。このコマンドを実行した後、アプライアンスを再起動する前に、設定を保存します。

コマンドプロンプトで、次のコマンドを入力します。

```
1 reset fips
2
```



```
3
4 reboot
5
6 set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS
7
8 This command will erase all data on the FIPS card. You must save the
     configuration (saveconfig) after executing this command. Do you want
      to continue?(Y/N)y
9
10 <!--NeedCopy-->
```

注:set fips コマンドを実行すると、次のメッセージが表示されます。

```
1 This command will erase all data on the FIPS card. You must save the
     configuration (saveconfig) after executing this command. [Note: On
     MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
     default, and the -initHSM Level-2 option is internally converted to
     Level-3]  Do you want to continue?(Y/N)y
2
3 saveconfig
4
5 reboot
6
7 reboot
8
9 show fips
10
11         FIPS HSM Info:
12                 HSM Label               : NetScaler FIPS
13                 Initialization          : FIPS-140-2 Level-3
14                 HSM Serial Number       : 3.1G1836-ICM000136
15                 HSM State               : 2
16                 HSM Model               : NITROX-III CNN35XX-NFBE
17                 Hardware Version        : 0.0-G
18                 Firmware Version        : 1.0
19                 Firmware Build          : NFBE-FW-1.0-48
20                 Max FIPS Key Memory     : 102235
21                 Free FIPS Key Memory    : 102231
22                 Total SRAM Memory       : 557396
23                 Free SRAM Memory        : 262780
24                 Total Crypto Cores      : 63
25                 Enabled Crypto Cores    : 63
26
```



```
<!--NeedCopy-->
```

FIPS キーを作成する

MPX 14000 FIPS アプライアンスで FIPS キーを作成するか、既存の FIPS キーをアプライアンスにインポートできます。MPX 14000 FIPS アプライアンスは、2048 ビットと 3072 ビットのキーと、指数値 F4 (その値は 65537) のみをサポートします。PEM キーの場合、指数は不要です。FIPS キーが正しく作成されていることを確認します。証明書署名要求とサーバー証明書を作成します。最後に、証明書とキーのペアをアプライアンスに追加します。

キータイプ (RSA または ECDSA) を指定します。ECDSA キーの場合は、カーブだけを指定します。カーブ P_256 および P_384 の ECDSA キーの作成がサポートされています。

> 注:
>
> 1024 ビットおよび 4096 ビットのキー、および指数値 3 はサポートされていません。

CLI を使用して FIPS キーを作成する

コマンドプロンプトで入力します。

```
create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent ( 3 | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
<!--NeedCopy-->
```

Example1:

```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4

show ssl fipskey f1

FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 ( Hex: 0x10001)

<!--NeedCopy-->
```

Example2:

```
> create fipskey f2 -keytype ECDSA -curve P_256

> sh fipskey f2
```



```
    FIPS Key Name: f2   Key Type: ECDSA Curve: P_256

<!--NeedCopy-->
```

GUI を使用して **FIPS** キーを作成する

1. [トラフィック管理] **> [SSL] > [FIPS]** に移動します。
2. 詳細ウィンドウの [FIPS キー] タブで、[追加] をクリックします。
3. [FIPS キーの作成] ダイアログボックスで、次のパラメータの値を指定します。
   - FIPS キー名 *—fipsKeyName
   - モジュラス *-モジュラス
   - 指数 *-指数

   * 必須パラメータ
4. [**Create**] をクリックしてから、[**Close**] をクリックします。
5. [FIPS キー] タブで、作成した FIPS キーに対して表示された設定が正しいことを確認します。

FIPS キーをインポートする

既存の FIPS キーを FIPS アプライアンスで使用するには、アプライアンスのハードディスクから HSM に FIPS キーを転送する必要があります。

注**:** FIPS キーのインポート時のエラーを回避するには、インポートされたキーの名前が、作成時の元のキー名と同じであることを確認してください。

CLI を使用して **FIPS** キーをインポートする

コマンドプロンプトで入力します。

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
   wrapKeyName <string>] [-iv<string>] -exponent  F4 ]
<!--NeedCopy-->
```

例**:**

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4

import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
```



```
4
5 <!--NeedCopy-->
```

`show fipskey`コマンドを実行して、FIPS キーが正しく作成またはインポートされていることを確認します。

```
1 show fipskey
2 1)      FIPS Key Name: Key-FIPS-2
3
4 <!--NeedCopy-->
```

GUI を使用して **FIPS** キーをインポートする

1. [トラフィック管理**] > [SSL] > [FIPS**] に移動します。
2. 詳細ウィンドウの [FIPS キー] タブで、[インポート] をクリックします。
3. [FIPS キーとしてインポート] ダイアログボックスで、[FIPS キーファイル] を選択し、次のパラメータの値を設定します。
   - FIPS キー名 *
   - Key File Name* — デフォルト以外の場所にファイルを配置するには、完全なパスを指定するか、[参照] をクリックして場所に移動します。
   - 指数 *

   * 必須パラメータ
4. [インポート] をクリックし、[閉じる] をクリックします。
5. [FIPS キー] タブで、インポートした FIPS キーに対して表示される設定が正しいことを確認します。

FIPS キーをエクスポートする

FIPS HSM で作成されたキーのバックアップを作成することをお勧めします。HSM 内のキーが削除された場合、同じキーを再度作成することはできず、それに関連付けられているすべての証明書は役に立たなくなります。

キーをバックアップとしてエクスポートするだけでなく、別のアプライアンスに転送するためにキーをエクスポートする必要がある場合があります。

次の手順では、FIPS キーをアプライアンスの CompactFlash 上の`/nsconfig/ssl`フォルダにエクスポートし、強力な非対称キー暗号化方式を使用してエクスポートされたキーを保護する方法について説明します。

CLI を使用して **FIPS** キーをエクスポートする

コマンドプロンプトで入力します。


```
export ssl fipsKey <fipsKeyName> -key <string>
<!--NeedCopy-->
```

例**:**

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
<!--NeedCopy-->
```

GUI を使用して **FIPS** キーをエクスポートする

1. [トラフィック管理**] > [SSL] > [FIPS**] に移動します。
2. 詳細ウィンドウの [FIPS キー] タブで、[エクスポート] をクリックします。
3. [FIPS キーをファイルにエクスポート] ダイアログボックスで、次のパラメータの値を指定します。
   - FIPS キー名 *—fipsKeyName
   - ファイル名 *-キー (デフォルト以外の場所にファイルを配置するには、完全パスを指定するか、[参照] ボタンをクリックして場所を指定します)。

   * 必須パラメータ
4. **[**エクスポート**]** をクリックし、**[**閉じる**]** をクリックします。

外部キーをインポートする

Citrix ADC アプライアンスの HSM 内に作成された FIPS キーを転送できます。外部秘密鍵（標準の Citrix ADC、Apache、または IIS で作成された鍵など）を Citrix ADC FIPS アプライアンスに転送することもできます。外部キーは、OpenSSL などのツールを使用して HSM の外部で作成されます。外部キーを HSM にインポートする前に、/nsconfig/sslの下にあるアプライアンスのフラッシュドライブにコピーします。

MPX 14000 FIPS アプライアンスでは、外部キーのインポート時に、**import** ssl fipskeyコマンドの-exponent パラメータは不要です。キーのインポート時に正しい公開指数が自動的に検出され、-exponent パラメータの値は無視されます。

Citrix ADC FIPS アプライアンスは、3 または F4 以外の公開指数を持つ外部キーをサポートしていません。

MPX 14000 FIPS アプライアンスにはラップキーは必要ありません。

外部の暗号化された FIPS キーを MPX 14000 FIPS アプライアンスに直接インポートすることはできません。キーをインポートするには、まずキーを復号化してからインポートする必要があります。キーを復号化するには、シェルプロンプトで次のように入力します。


```
openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>
<!--NeedCopy-->
```

注： RSA キーを FIPS キーとしてインポートする場合は、セキュリティ上の理由から RSA キーをアプライアンスから削除することをお勧めします。

CLI を使用して外部キーを **FIPS** キーとしてインポートする

1. 外部キーをアプライアンスのフラッシュドライブにコピーします。
2. キーが.pfx 形式の場合は、まず PEM 形式に変換する必要があります。コマンドプロンプトで入力します。

```
convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx
    file name> -password <password>
<!--NeedCopy-->
```

3. コマンドプロンプトで次のコマンドを入力して、外部キーを FIPS キーとしてインポートし、設定を確認します。

```
import ssl fipsKey <fipsKeyName> -key <string> -informPEM
show ssl fipskey<fipsKeyName>
<!--NeedCopy-->
```

例:

```
convert ssl pkcs12 iis.pem -password 123456 -import -pkcs12File iis.pfx

import fipskey Key-FIPS-2 -key iis.pem -inform PEM

show ssl fipskey key-FIPS-2

FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0
    x10001)
<!--NeedCopy-->
```

GUI を使用して外部キーを **FIPS** キーとしてインポートする

1. キーが.pfx 形式の場合は、まず PEM 形式に変換する必要があります。

a) [トラフィック管理**] > [SSL**] に移動します。
b) 詳細ウィンドウの [ツール] で、[**PKCS #12** のインポート] をクリックします。
c) [PKCS12 ファイルのインポート] ダイアログボックスで、次のパラメータを設定します。
- 出力ファイル名 *
- PKCS12 ファイル名 *-.pfx ファイル名を指定します。
- パスワードのインポート *
- エンコード形式

* 必須パラメータ

2. [トラフィック管理**] > [SSL] > [FIPS**] に移動します。
3. 詳細ウィンドウの [FIPS キー] タブで、[インポート] をクリックします。
4. [FIPS キーとしてインポート] ダイアログボックスで、[PEM ファイル] を選択し、次のパラメータの値を設定します。
   - FIPS キー名 *
   - Key File Name*-デフォルト以外の場所にファイルを配置するには、完全パスを指定するか、「参照」(Browse) をクリックして場所に移動します。

   * 必須パラメータ
5. [インポート] をクリックし、[閉じる] をクリックします。
6. [FIPS キー] タブで、インポートした FIPS キーに対して表示される設定が正しいことを確認します。

HA セットアップのアプライアンスで **FIPS** を構成する

1 つの HA ペアに 2 つのアプライアンスを FIPS アプライアンスとして設定できます。

前提条件

- 両方のアプライアンスで Hardware Security Module (HSM) を構成する必要があります。詳細については、「HSM の設定」を参照してください。
- GUI を使用する場合、アプライアンスで HA セットアップ済みであることを確認します。HA セットアップの構成の詳細については、「高可用性」を参照してください。

> 注:
>
> この手順には、構成ユーティリティ（GUI）を使用することをお勧めします。コマンドライン（CLI）を使用する場合は、手順に記載されている手順に注意深く従ってください。ステップの順序を変更したり、誤った入力ファイルを指定したりすると、アプライアンスの再起動を必要とする不整合が発生する可能性があります。また、CLI を使用する場合、`create ssl fipskey`コマンドはセカンダリノードに伝播されません。2 つの異なる FIPS アプライアンスでモジュラスサイズと指数に同じ入力値を指定してコマンドを実行すると、生成されるキーは同じではありません。いずれかのノードで FIPS キーを作成し、もう一方のノードに転送します。ただ

> し、構成ユーティリティを使用して HA セットアップで FIPS アプライアンスを構成すると、作成した FIPS キーが自動的にセカンダリノードに転送されます。FIPS キーを管理および転送するプロセスは、セキュア情報管理 (SIM) と呼ばれます。

重要： HA セットアップは 6 分以内に完了する必要があります。いずれかの手順で手順が失敗した場合は、次の操作を行います。

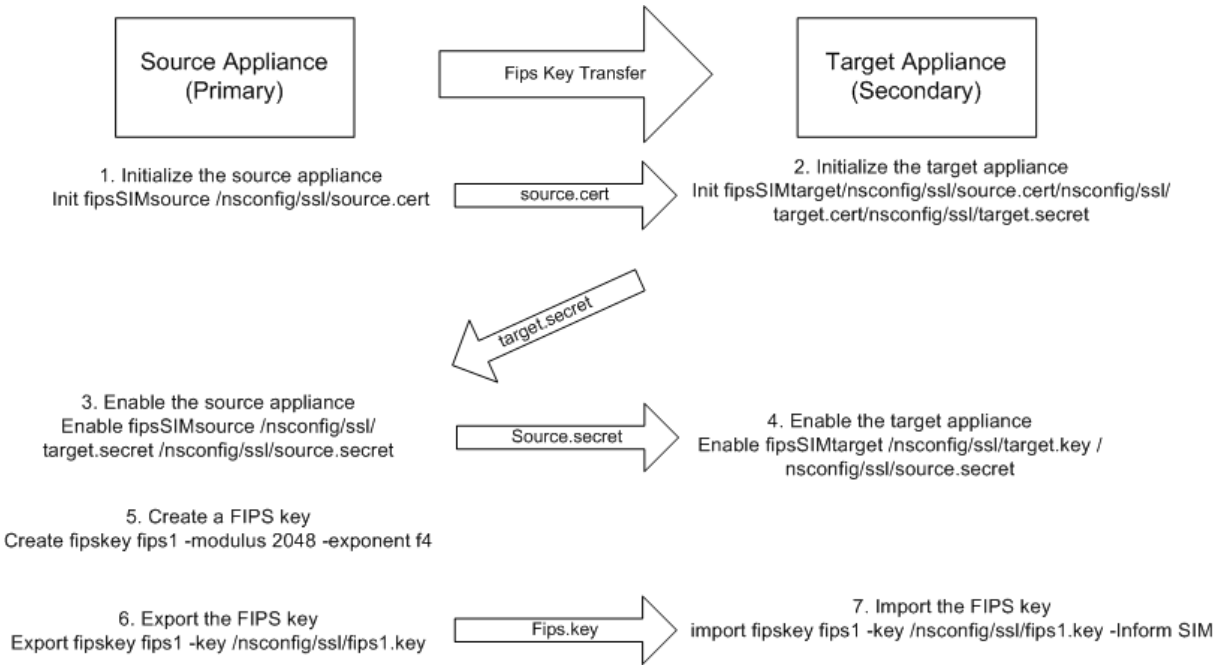
> 1. アプライアンスを再起動するか、10 分間待ちます。
> 2. プロシージャで作成されたすべてのファイルを削除します。
> 3. HA セットアップ手順を繰り返します。
>
> 既存のファイル名を再利用しないでください。

次の手順では、アプライアンス A がプライマリノード、アプライアンス B がセカンダリノードです。

CLI を使用して **HA** セットアップのアプライアンスで **FIPS** を構成する

次の図は、CLI で転送プロセスをまとめたものです。

図 1: FIPS キーサマリーを転送する



1. アプライアンス **A** で、PuTTY などの SSH クライアントを使用してアプライアンスへの SSH 接続を開きます。
2. 管理者の資格情報を使用して、アプライアンスにログオンします。
3. アプライアンス A をソースアプライアンスとして初期化します。コマンドプロンプトで入力します。

```
1 init ssl fipsSIMsource <certFile>
2 <!--NeedCopy-->
```


例:

```
init fipsSIMsource /nsconfig/ssl/nodeA.cert
```

4. この<certFile>ファイルをアプライアンス B の /nconfig/ssl フォルダにコピーします。

   例:

   ```
   scp /nsconfig/ssl/nodeA.cert nsroot@198.51.100.10:/nsconfig/ssl
   ```

5. アプライアンス **B** で、PuTTY などの SSH クライアントを使用してアプライアンスへの SSH 接続を開きます。
6. 管理者の資格情報を使用して、アプライアンスにログオンします。
7. アプライアンス B をターゲットアプライアンスとして初期化します。コマンドプロンプトで入力します。

   ```
   1 init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
   2 <!--NeedCopy-->
   ```

   例:

   ```
   init fipsSIMtarget /nsconfig/ssl/nodeA.cert /nsconfig/ssl/nodeB.key /
   nsconfig/ssl/nodeB.secret
   ```

8. この<targetSecret>ファイルをアプライアンス A にコピーします。

   例:

   ```
   scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.20:/nsconfig/
   ssl
   ```

9. アプライアンス **A** で、アプライアンス A をソースアプライアンスとして有効にします。コマンドプロンプトで入力します。

   ```
   1 enable ssl fipsSIMSource <targetSecret> <sourceSecret>
   2 <!--NeedCopy-->
   ```

   例:

   ```
   enable fipsSIMsource /nsconfig/ssl/nodeB.secret /nsconfig/ssl/nodeA.
   secret
   ```

10. この<sourceSecret>ファイルをアプライアンス B にコピーします。

    例:

    ```
    scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.10:/nsconfig/
    ssl
    ```


11. アプライアンス **B** で、アプライアンス B をターゲットアプライアンスとして有効にします。コマンドプロンプトで入力します。

```
enable ssl fipsSIMtarget <keyVector> <sourceSecret>
<!--NeedCopy-->
```

例**:**

```
enable fipsSIMtarget /nsconfig/ssl/nodeB.key /nsconfig/ssl/nodeA.secret
```

12. アプライアンス **A** で、FIPS キーの作成の説明に従ってFIPS キーを作成します。
13. FIPS キーのエクスポートの説明に従って、FIPS キーをアプライアンスのハードディスクにエクスポートします。
14. SCP などの安全なファイル転送ユーティリティを使用して、セカンダリアプライアンスのハードディスクに FIPS キーをコピーします。
15. アプライアンス **B** で、FIPS キーのインポートの説明に従って、FIPS キーをハードディスクからアプライアンスの HSM にインポートします。

GUI を使用して **HA** セットアップのアプライアンスで **FIPS** を構成する

1. ソースアプライアンスとして設定するアプライアンスで、トラフィック管理 **> SSL > FIPS** に移動します。
2. 詳細ウィンドウの [FIPS 情報] タブで、[**SIM** を有効にする] をクリックします。
3. [**HA** ペアの **SIM** を有効にする] ダイアログボックスの [証明書ファイル名] テキストボックスに、ファイル名を入力します。ファイル名には、ソースアプライアンス上の FIPS 証明書を保存する必要がある場所へのパスが含まれている必要があります。
4. [キーベクトルファイル名] テキストボックスに、ファイル名を入力します。ファイル名には、ソースアプライアンス上の FIPS キーベクトルを格納する必要がある場所へのパスが含まれている必要があります。
5. [ターゲットシークレットファイル名］テキストボックスに、ターゲットアプライアンス上のシークレットデータを格納する場所を入力します。
6. [ソースシークレットファイル名（Source Secret File Name）] テキストボックスに、ソースアプライアンス上のシークレットデータを格納する場所を入力します。
7. [セカンダリシステムのログイン資格情報] で、[ユーザー名] と [パスワード] の値を入力します。
8. **OK** をクリックします。これで、FIPS アプライアンスが HA モードに設定されます。

注**:** HA でアプライアンスを設定したら、FIPS キーの作成の説明に従ってFIPS キーを作成します。FIPS キーは、プライマリからセカンダリアプライアンスに自動的に転送されます。

CLI を使用して証明書署名要求を作成する

コマンドプロンプトで入力します。


```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string>
     [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

例:

```
>create certreq f1.req  - fipsKeyName  f1 -countryName US  -stateName CA
     -organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com

<!--NeedCopy-->
```

CLI を使用してサーバー証明書を作成する

コマンドプロンプトで入力します。

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
     input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
     input_filename>][-CAkeyForm ( DER | PEM )] [-CAserial <
     output_filename>]
<!--NeedCopy-->
```

例:

```
create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
    root.key -CAserial ns-root.srl -days 1000

<!--NeedCopy-->
```


前の例では、アプライアンスのローカルルート CA を使用してサーバ証明書を作成します。

CLI を使用した証明書とキーのペアの追加

コマンドプロンプトで入力します。

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>][-
    expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

例:

```
add certkey cert1 -cert f1.cert -fipsKey f1

<!--NeedCopy-->
```

FIPS キーとサーバー証明書を作成したら、汎用 SSL 設定を追加できます。展開に必要な機能を有効にします。サーバー、サービス、および SSL 仮想サーバーを追加します。証明書とキーのペアとサービスを SSL 仮想サーバーにバインドします。構成を保存します。

```
enable ns feature SSL LB

add server s1 10.217.2.5

add service sr1 s1 HTTP 80

add lb vserver v1 SSL 10.217.2.172 443

bind ssl vserver v1  - certkeyName cert1

bind lb vserver v1 sr1

saveconfig
<!--NeedCopy-->
```

これで、MPX 14000 FIPS アプライアンスの基本構成が完了しました。

セキュア HTTPS の構成の詳細については、「FIPS の構成」をクリックしてください。

セキュア RPC の構成の詳細については、[FIPS の構成] を初めてクリックしてください。

MPX 14000 FIPS アプライアンスのライセンスを更新する

このプラットフォームでライセンスを更新するには、2 回再起動する必要があります。

1. /nsconfig/licenseフォルダ内のライセンスを更新します。
2. アプライアンスを再起動します。
3. アプライアンスにログオンします。
4. アプライアンスを再起動します。
   注**:** 2 回目の再起動の前に、新しいコマンドを追加したり、設定を保存したり、システム状態をチェックしたりしないでください。
5. アプライアンスにログオンし、`show ssl fips` コマンドを実行して FIPS が初期化されていることを確認します。

MPX 14000 FIPS および SDX 14000 FIPS プラットフォームでのハイブリッド FIPS モードのサポート

> 注:
>
> この機能は、1 つのプライマリ FIPS カードと 1 つ以上のセカンダリカードを含む新しい MPX/SDX 14000 FIPS プラットフォームでのみサポートされます。VPX プラットフォームや、1 種類のハードウェアカードのみを含むプラットフォームではサポートされていません。

FIPS プラットフォームでは、セキュリティ上の理由から、非対称および対称の暗号化と復号化が FIPS カードで実行されます。ただし、FIPS カードでこのアクティビティの一部（非対称）を実行し、キーのセキュリティを損なうことなく、バルク暗号化と復号化（対称）を別のカードにオフロードできます。

新しい MPX/SDX 14000 FIPS プラットフォームには、1 枚のプライマリカードと 1 つ以上のセカンダリカードが含まれています。ハイブリッド FIPS モードを有効にすると、秘密キーがこのカードに保存されるため、プリマスターシークレット復号化コマンドがプライマリカードで実行されます。ただし、バルク暗号化と復号化はセカンダリカードにオフロードされます。このオフロードにより、非ハイブリッド FIPS モードおよび既存の MPX 9700/10500/12500/15000 FIPS プラットフォームと比較して、MPX/SDX 14000 FIPS プラットフォームでのバルク暗号化スループットが大幅に向上します。ハイブリッド FIPS モードを有効にすると、このプラットフォームでの 1 秒あたりの SSL トランザクションも向上します。

> メモ:
>
> - ハイブリッド FIPS モードは、すべての暗号計算を FIPS 認定モジュール内で行う必要がある厳格な認証要件を満たすために、デフォルトで無効になっています。ハイブリッドモードを有効にして、バルク暗号化と復号化をセカンダリカードにオフロードします。
> - SDX 14000 FIPS プラットフォームでは、ハイブリッドモードを有効にする前に、まず VPX インスタンスに SSL チップを割り当てる必要があります。

CLI を使用してハイブリッド **FIPS** モードを有効にする

コマンドプロンプトで入力します。

```
set SSL parameter -hybridFIPSMode {
 ENABLED|DISABLED }


Arguments

hybridFIPSMode

When this mode is enabled, system will use additional crypto hardware
   to accelerate symmetric crypto operations.

Possible values: ENABLED, DISABLED

Default value: DISABLED
<!--NeedCopy-->
```

例**:**

```
    set SSL parameter -hybridFIPSMode ENABLED
    show SSL parameter
    Advanced SSL Parameters
    -----------------------
    . . . . . .
    Hybrid FIPS Mode    : ENABLED
    . . . . . .

<!--NeedCopy-->
```

GUI を使用してハイブリッド **FIPS** モードを有効にする

1. [トラフィック管理**] > [SSL**] に移動します。
2. 詳細ウィンドウの [設定] で、[**SSL** の詳細設定の変更] をクリックします。
3. [**SSL** の詳細設定の変更] ダイアログボックスで、[ハイブリッド **FIPS** モード] を選択します。

制限事項:

1. 再ネゴシエーションはサポートされていません。

2. SDX 14000 プラットフォームの`stat ssl parameter`コマンドは、正しいセカンダリカードの使用率を表示しません。常に 0.00% の使用率が表示されます。

```
stat ssl

SSL Summary
# SSL cards present 1
# SSL cards UP  1
# Secondary SSL cards present   4
# Secondary SSL cards UP    4
SSL engine status      1
SSL sessions (Rate)     963
Secondary card utilization (%)     0.00
<!--NeedCopy-->
```

SDX 14000 FIPS アプライアンス

April 25, 2022

> 注:
>
> Citrix ADC のダウンロードページの「Citrix ADC リリース 12.1-FIPS」および「Citrix ADC リリース 12.1-ndCPP」にリストされているファームウェアのバージョンは、MPX 14000 FIPS または SDX 14000 FIPS プラットフォームではサポートされていません。これらのプラットフォームでは、ダウンロードページで入手できる他の最新の Citrix ADC ファームウェアバージョンを使用できます。

Citrix ADC SDX アプライアンスは、複数の仮想 Citrix ADC インスタンスをプロビジョニングおよび管理できるマルチテナントプラットフォームです。SDX アプライアンスは、単一の管理者がアプライアンスを構成および管理し、各ホストされたインスタンスの管理をテナントに委任できるようにすることで、クラウドコンピューティングおよびマルチテナンシーの要件に対応します。

Citrix ADC SDX 14030/14060/14080 FIPS アプライアンスは、FIPS 機能を備えた SDX アプライアンスの機能を提供します。FIPS 140-2 レベル 3 仕様（リリース 12.0 ビルド 56.x 以降）に準拠するように設計された、改ざん防止（改ざん防止）暗号化モジュール Cavium CNN3560-NFBE-G が装備されています。クリティカルセキュリティパラメータ (CSP) は、主にサーバの秘密キーで、暗号化モジュール内で安全に保存および生成されます。このモジュールは、ハードウェアセキュリティモジュール (HSM) とも呼ばれます。CSP は、HSM の境界外でアクセスされることはありません。スーパーユーザー (`nsroot`) だけが、HSM 内に格納されているキーに対して操作を実行できます。

Citrix ADC SDX 14030/14060/14080 FIPS アプライアンスには、63 コアの FIPS HSM モジュールが 1 つ含まれています。FIPS HSM モジュールは、最大 32 個のパーティションまでパーティション化できます。SDX 管理者は、専用のキーストレージ、暗号化リソース、および暗号化 SSL FIPS コアの数を各パーティションに割り当てることが

できます。パーティションに割り当てられたキーとリソースは専用で安全であり、他のパーティションはそれらにアクセスしたり共有したりすることはできません。

作成した FIPS HSM パーティションは、インスタンスのプロビジョニング時、または後でインスタンスを編集して VPX インスタンスに割り当てたり、アタッチしたりできます。作成され、インスタンスにアタッチされた FIPS パーティションは、そのインスタンスの仮想 HSM モジュールのように動作します。

SDX 14030/14060/14080 FIPS アプライアンスの VPX インスタンスには FIPS 仮想機能 (VF) パーティションが割り当てられ、分離された FIPS 仮想カードまたは HSM として扱われます。したがって、VPX インスタンス内で FIPS パーティションを構成する手順は、MPX FIPS アプライアンスを構成する手順と似ています。コンプライアンスの詳細については、米国国立標準技術研究所 (NIST) の Web サイトにあるセキュリティポリシーの詳細を参照してください。

高可用性セットアップでの FIPS アプライアンスの構成の詳細については、「高可用性セットアップでの FIPS アプライアンス」を参照してください。

> 重要
>
> 各キーには、プライベートキーとパブリックキーが含まれます。その結果、2 つのキースペースを占有します。したがって、キーの最大数は、キーストアサイズの半分未満の 1 つに制限されます。

SDX 14000 FIPS プラットフォームは、ハイブリッド FIPS モードをサポートしています。このモードでは、暗号化および復号化アクティビティの一部を FIPS 以外のカードにオフロードできます。詳細については、「ハイブリッド FIPS モード」を参照してください。

制限事項

October 7, 2021

1. SDX FIPS アプライアンスのバックエンドでは、SSLv3 プロトコルを使用した SSL 再ネゴシエーションはサポートされていません。
2. 1024 ビットおよび 4096 ビットのキーおよび指数値 3 はサポートされません。
3. バックアップと復元はサポートされていません。
4. クラスタおよび管理ドメインはサポートされていません。
5. 1 つのインスタンスにアタッチできる FIPS パーティションは 1 つだけです。
6. FIPS パーティションを持つインスタンスに割り当てることができる CPU コアは 1 つだけです。
7. FIPS パーティションまたは SSL コアをインスタンスに割り当てることはできますが、両方を割り当てることはできません。
8. 4096 ビットサーバ証明書はサポートされていません。
9. 4096 ビットのクライアント証明書はサポートされていません (バックエンドサーバーでクライアント認証が有効になっている場合)。

用語

October 7, 2021

ゼロ化: HSM をリセットします。HSM 上のすべてのデータが削除されます。この手順は、HSM を初期化する前に必須です。

初期化: HSM 機能を設定します。Citrix ADC SDX FIPS アプライアンスは、FIPS-140-2 レベル 2 に準拠しています。チップを初期化した後で、パーティションを作成できます。

キーストアサイズ: パーティションに格納できるキーの数。最大 102235 個のキーを指定できます。格納できるキーの最大数は、指定された数の半分以下です。たとえば、100 を指定した場合、作成できるキーは 49 個だけです。これは、キーの 1 つが RSA キーペアで 2 つのキーストアを消費するためです。

暗号コア容量：パーティションに割り当てられた暗号コアの数。最大 63 個のコアを使用できます。

SSL コンテキスト: パーティション上に作成できる同時 SSL 接続の数。

HSM の初期化

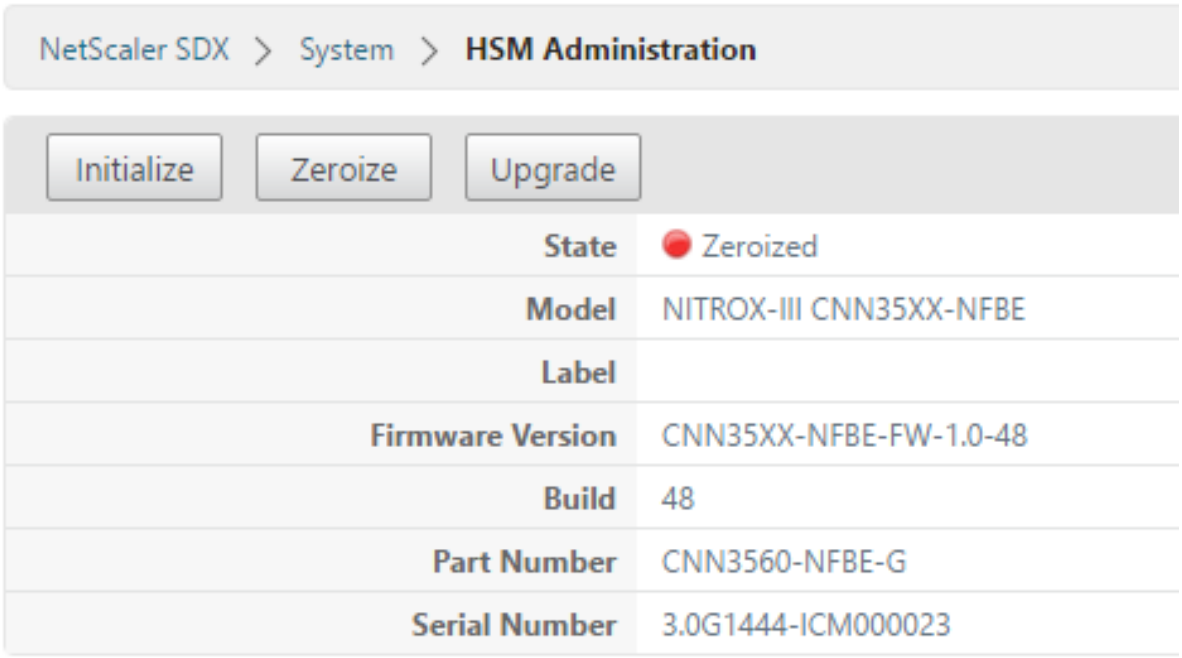
October 7, 2021

HSM を初期化する前に、まず HSM をゼロ化する必要があります。

管理サービスを使用して **HSM** をゼロにする

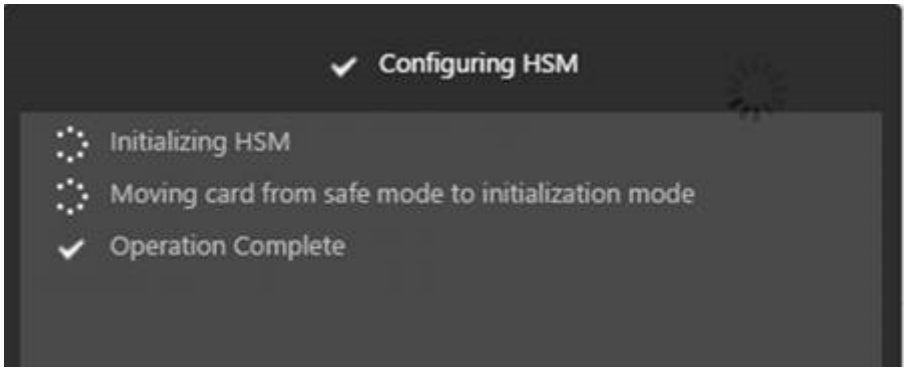
1. ブラウザを開き、アプライアンスにログオンします。
2. **[**構成**]** タブで、**[**システム**] > [HSM** 管理**]** に移動し、詳細プレーンで **[**ゼロ化**]** をクリックします。

すべてのデータは FIPS チップから消去され、状態は「ゼロ化」と表示されます。以前に作成された HSM パーティションはすべて削除されます。

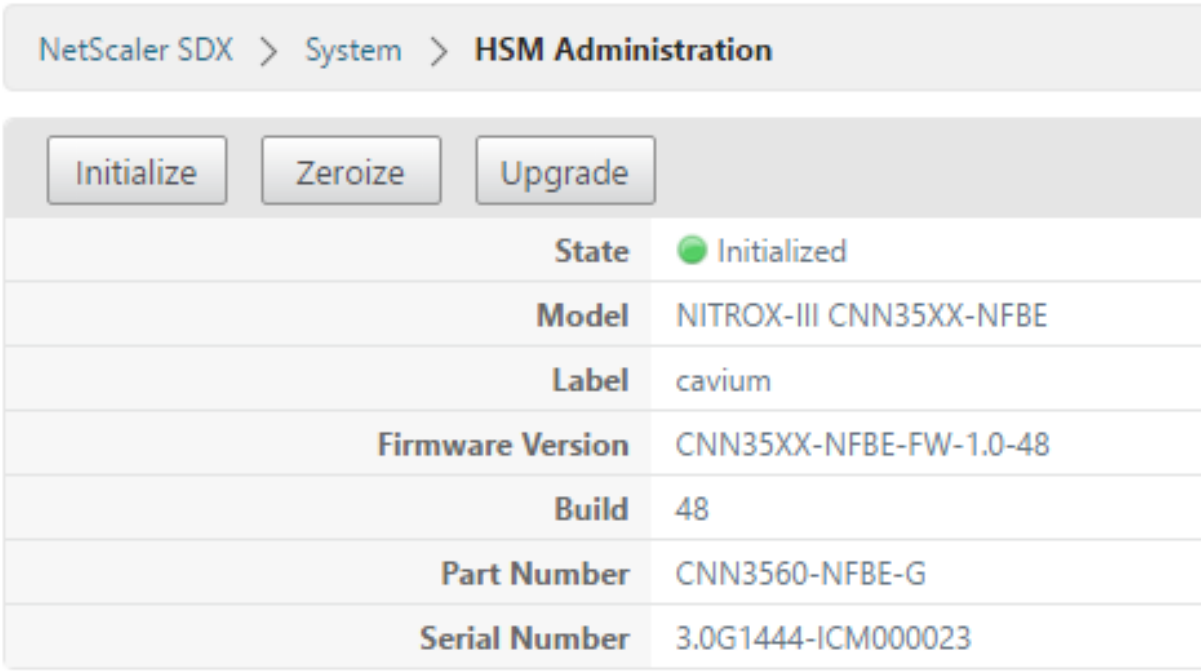


管理サービスを使用した **HSM** の初期化

1. **[**構成**]** タブで、**[**システム**] > [HSM** 管理**]** に移動し、詳細プレーンで **[**初期化**]** をクリックします。
2. 新しいユーザー名を入力し、パスワードを指定して、「**OK**」をクリックします。



カードの状態は「初期化」と表示されます。



パーティションの作成

October 7, 2021

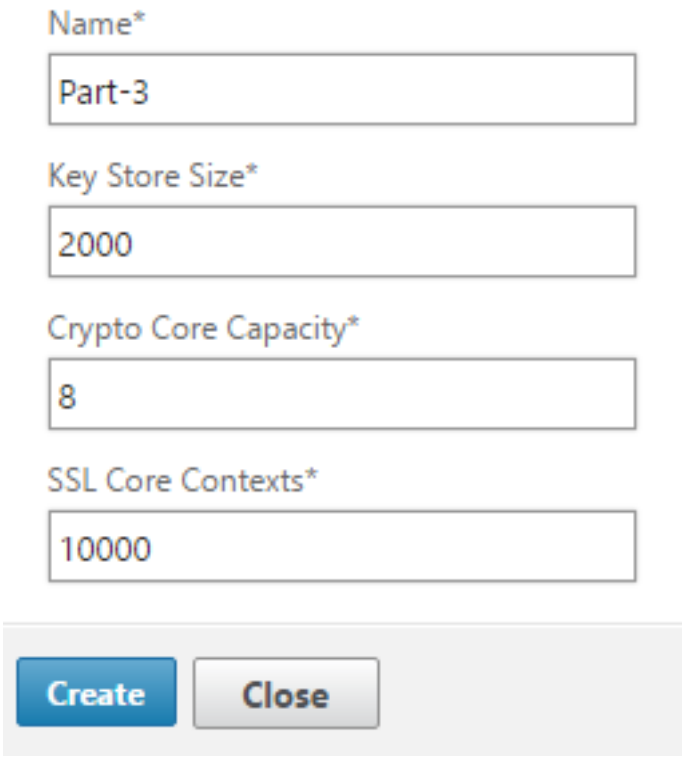
異なるテナントのパーティションを作成し、各パーティションの暗号化リソースを指定します。各インスタンスには 1 つのパーティションが割り当てられ、1 つのパーティションは 1 つのインスタンスにのみ割り当てられます。インスタンスを削除すると、そのインスタンスに割り当てられたパーティションが削除されます。その結果、パーティションデータも削除され、セキュリティで保護されていないままになったり、後でアクセスしたりすることはありません。キーの数と SSL コンテキストの割り当ては、アプリケーションによって異なります。割り当てるコアの数については、Citrix ADC のデータシートを参照してください。

> 重要
>
> キーストアのサイズとコアを HSM パーティションに割り当てた後は、実行時に変更することはできません。まず、パーティションをインスタンスからデタッチします。

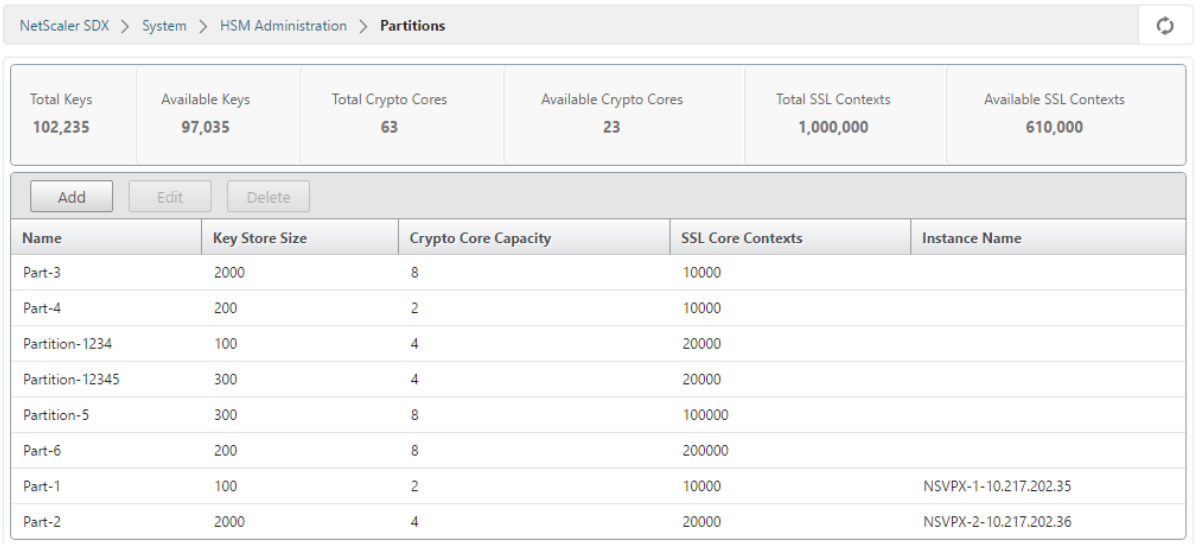
管理サービスを使用してパーティションを作成する

1. [構成] タブで、[システム] > [**HSM** 管理] > [パーティション] に移動し、詳細プレーンで [追加] をクリックします。

2. パーティションの名前と、このパーティションに割り当てるリソースを指定します。
3. [**OK**] をクリックします。



サマリページには、作成されたすべてのパーティションが表示されます。一部のパーティションはインスタンスに割り当てられ、一部は空きパーティションです。



Name	Key Store Size	Crypto Core Capacity	SSL Core Contexts	Instance Name
Part-3	2000	8	10000	
Part-4	200	2	10000	
Partition-1234	100	4	20000	
Partition-12345	300	4	20000	
Partition-5	300	8	100000	
Part-6	200	8	200000	
Part-1	100	2	10000	NSVPX-1-10.217.202.35
Part-2	2000	4	20000	NSVPX-2-10.217.202.36

新しいインスタンスをプロビジョニングするか、既存のインスタンスを変更してパーティションを割り当てる

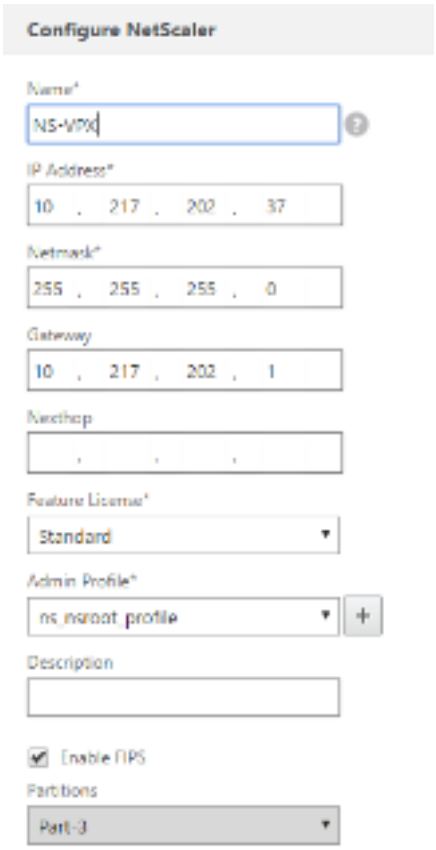
October 7, 2021

パーティションを作成したら、それらをインスタンスに割り当てる必要があります。

> 重要:
>
> - 1 つのインスタンスにアタッチできる FIPS パーティションは 1 つだけです。
> - FIPS パーティションを持つインスタンスに割り当てることができる CPU コアは 1 つだけです。

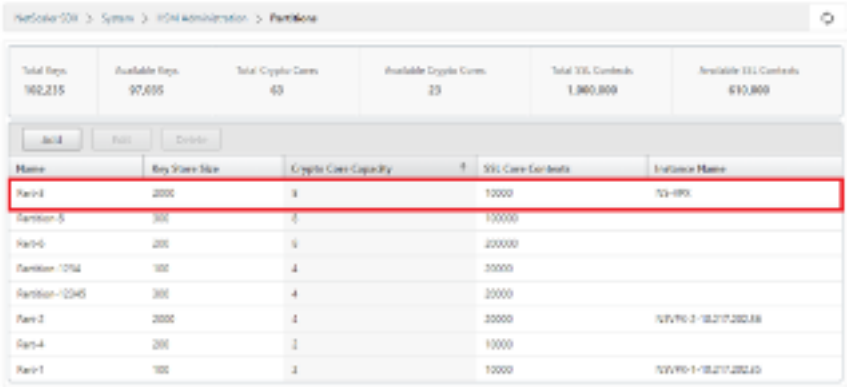
新しいインスタンスのプロビジョニングまたは既存のインスタンスの変更

1. [構成] タブで、[**NetScaler**] > [インスタンス] に移動し、インスタンスを追加または変更します。
2. [**Enable FIPS**] を選択し、[パーティション] リストから、このインスタンスにアタッチするパーティションを選択します。



パーティションがインスタンスにアタッチされていることを確認するには、GUI または CLI を使用します。

GUI で、[システム] > [**HSM** 管理] > [パーティション] に移動します。パーティションにアタッチされているインスタンス名が表示されます。



FIPS パーティションの割り当てを解除するには、[**NetScaler**] > [インスタンス] に移動します。インスタンスを編集し、[**Enable FIPS**] チェックボックスをオフにします。

CLI で、コマンドプロンプトで次のコマンドを入力します。

```
1 show fips
2
3 FIPS Card is not configured
4  Done
5 <!--NeedCopy-->
```

次の出力が表示された場合は、デバッグのトラブルシューティングのセクションを参照してください。

エラー: 操作は許可されていません。システムに FIPS カードがありません。

SDX 14030/14060/14080 FIPS アプライアンス上のインスタンスの HSM を設定します

December 7, 2021

まず FIPS カードの状態をチェックして、ドライバが正しくロードされたことを確認してから、カードを初期化します。

コマンドプロンプトで入力します。

```
1 show fips
2
3 FIPS Card is not configured
4
5 Done
```



```
<!--NeedCopy-->
```

ドライバが正しくロードされていない場合は、「エラー: 操作は許可されていません-システムに FIPS カードがありません」というメッセージが表示されます。

FIPS カードの初期化

> 重要:
>
> /nsconfig/fips ディレクトリがアプライアンスに正常に作成されたことを確認します。

アプライアンスを 3 回目に再起動する前に、設定を保存しないでください。

FIPS カードを初期化するには、次の手順に従います。

1. FIPS カードをリセットします (reset fips)。
2. アプライアンスを再起動します (reboot)。
3. パーティション 0 と 1 にはセキュリティ担当者のパスワードを設定し、partition (set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS) にはユーザーパスワードを設定します。

   注:set または reset コマンドの実行には 60 秒以上かかります。
4. 設定を保存します (saveconfig)。
5. メインパーティション (master_pek.key) のパスワード暗号化キーが /nsconfig/fips/ ディレクトリに作成されたことを確認します。
6. アプライアンスを再起動します (reboot)。
7. FIPS カードが稼働していることを確認します（show fips)。

CLI を使用して FIPS カードを初期化します

コマンドプロンプトで、次のコマンドを入力します。

```
reset fips

reboot

set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -
    hsmLabel <string>
<!--NeedCopy-->
```


> 注: **set fips** コマンドを実行すると、次のメッセージが表示されます。

```
This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. [Note: On
    MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
    default, and the -initHSM Level-2 option is internally converted to
    Level-3]  Do you want to continue?(Y/N)y

saveconfig

reboot

show fips
<!--NeedCopy-->
```

例:

```
reset fips

Done

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. [Note: On
    MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
    default, and the -initHSM Level-2 option is internally converted to
    Level-3]  Do you want to continue?(Y/N)y

Done

saveconfig

Done

reboot

show fips

    FIPS HSM Info:
```



```
22      HSM Label : NSFIPS
23      Initialization : FIPS-140-2 Level-2
24      HSM Serial Number : 3.0G1532-ICM000228
25      HSM State : 2
26      HSM Model : NITROX-III CNN35XX-NFBE
27      Hardware Version : 0.0-G
28      Firmware Version : 1.0
29      Firmware Build : NFBE-FW-1.0-48
30      Max FIPS Key Memory : 1000
31      Free FIPS Key Memory : 1000
32      Total SRAM Memory : 557396
33      Free SRAM Memory : 238088
34      Total Crypto Cores : 4
35      Enabled Crypto Cores : 4
36  Done
37  <!--NeedCopy-->
```

SDX 14030/14060/14080 FIPS アプライアンス上のインスタンスに対して FIPS キーを作成します

October 7, 2021

インスタンスに FIPS キーを作成することも、既存の FIPS キーをインスタンスにインポートすることもできます。SDX 14030/14060/14080 FIPS アプライアンスは、2048 ビットおよび 3072 ビットのキーと F4 の指数値のみをサポートします。PEM キーの場合、指数は必要ありません。FIPS キーが正しく作成されていることを確認します。証明書署名要求とサーバー証明書を作成します。最後に、証明書とキーのペアをインスタンスに追加します。

> 注:
>
> 1024 ビットおよび 4096 ビットのキーおよび指数値 3 はサポートされません。

CLI を使用して FIPS キーを作成する

コマンドプロンプトで入力します。

```
1  create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent (3
       | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
2  <!--NeedCopy-->
```

例:


```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4

Done

show ssl fipskey ddvws

FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (
   Hex: 0x10001)

Done
<!--NeedCopy-->
```

CLI を使用して FIPS キーをインポートする

コマンドプロンプトで入力します。

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
   wrapKeyName <string>] [-iv<string>] [-exponent F4 ]
<!--NeedCopy-->
```

例:

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
Done
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
Done
<!--NeedCopy-->
```

show fipskey コマンドを実行して、FIPS キーが正しく作成またはインポートされていることを確認します。

```
show fipskey
1)      FIPS Key Name: Key-FIPS-2
Done
<!--NeedCopy-->
```

CLI を使用して証明書署名要求を作成する

コマンドプロンプトで入力します。


```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string>
     [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

例:

```
create certreq f1.req  - fipsKeyName f1 -countryName US -stateName CA -
    organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com`
`Done
<!--NeedCopy-->
```

CLI を使用したサーバー証明書の作成

コマンドプロンプトで入力します。

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
     input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
     input_filename>] [-CAkeyForm ( DER | PEM )] [-CAserial <
     output_filename>]
<!--NeedCopy-->
```

例:

```
create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
    root.key -CAserial ns-root.srl -days 1000
Done
<!--NeedCopy-->
```


前の例では、アプライアンスのローカルルート CA を使用してサーバー証明書を作成します。

CLI を使用した証明書とキーのペアの追加

コマンドプロンプトで入力します。

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod   <
    positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

例:

```
add certkey cert1 -cert f1.cert -fipsKey f1
Done
<!--NeedCopy-->
```

FIPS キーおよびサーバー証明書を作成したら、汎用 SSL 設定を追加できます。展開に必要な機能を有効にします。サーバー、サービス、SSL 仮想サーバーを追加します。証明書とキーのペアとサービスを SSL 仮想サーバにバインドし、設定を保存します。

```
enable ns feature SSL LB
Done
add server s1 10.217.2.5
Done
add service sr1 s1 HTTP 80
Done
add lb vserver v1 SSL 10.217.2.172 443
Done
bind ssl vserver v1  - certkeyName cert1
Done
bind lb vserver v1 sr1
Done
saveconfig
Done
<!--NeedCopy-->
```

セキュア HTTPS とセキュア RPC の設定については、ここをクリックしてください。

VPX インスタンスの FIPS ファームウェアのアップグレード

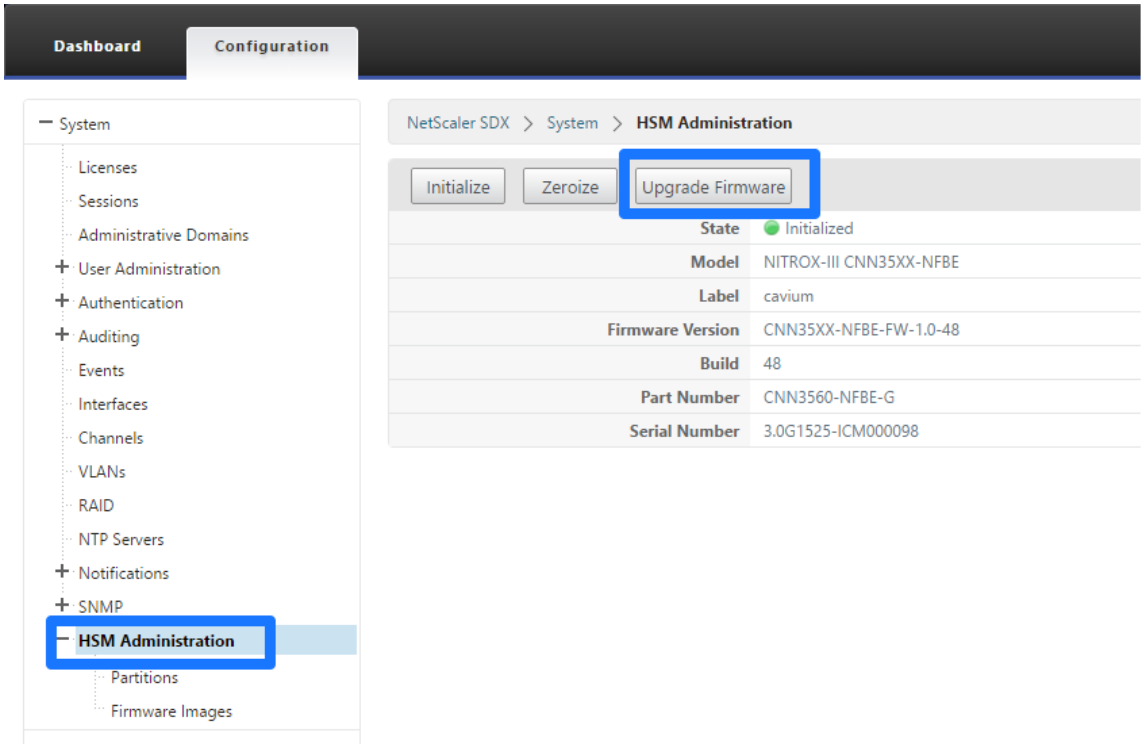
October 7, 2021

FIPS ファームウェアのアップデートは随時リリースされます。Citrix のダウンロードページから最新のファームウェアをダウンロードし、アプライアンスにアップロードします。アップグレード処理が完了するまでに最大で 10 分かかる場合があります。インスタンスは、アップグレード後に再起動されます。

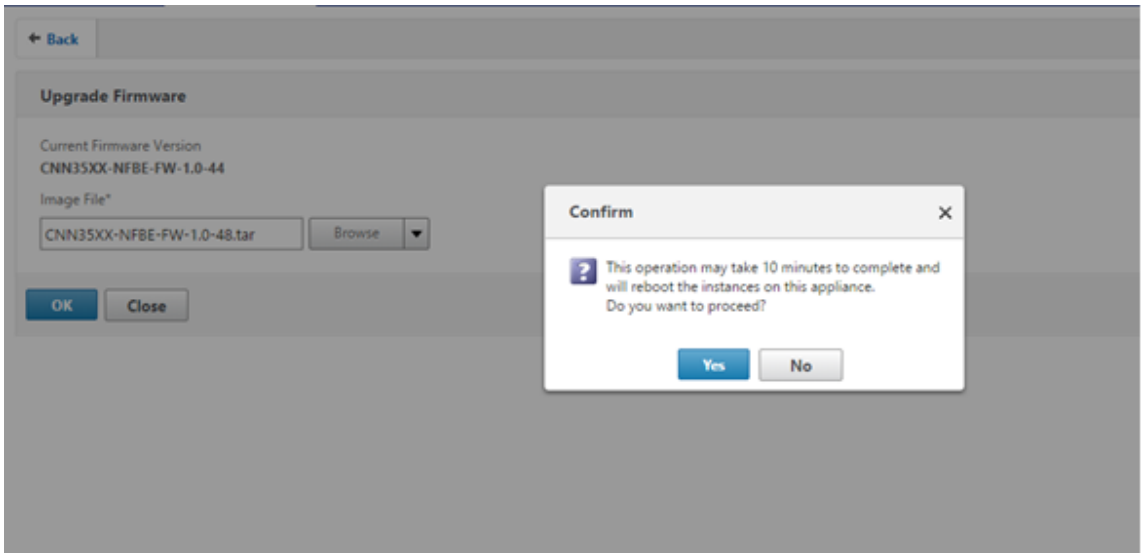
FIPS ファームウェアのアップグレード

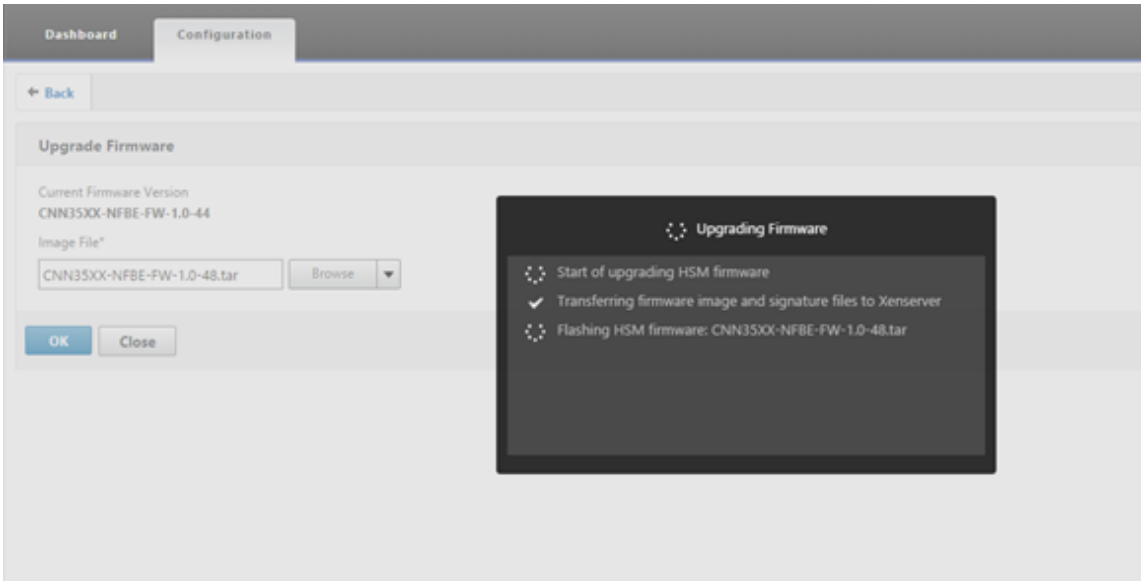
1. **[**システム**] > [HSM** 管理**] > [**ファームウェアイメージ**]** に移動します。
2. [アップロード] を選択します。

3. ファームウェアイメージを含むフォルダに移動し、ファイルを選択します。
4. **[**システム**] > [HSM** 管理**]** に移動し、**[**ファームウェアのアップグレード**]** を選択します。



5. アップグレード先のファームウェアイメージを選択し、[**OK**] をクリックします。





nShield Connect ハードウェアセキュリティモジュール（**HSM**）のサポート

October 7, 2021

FIPS 以外の Citrix ADC アプライアンスは、サーバーの秘密鍵をハードディスクに保存します。FIPS アプライアンスでは、キーは HSM と呼ばれる暗号化モジュールに格納されます。キーを HSM に保管すると、物理攻撃やソフトウェア攻撃から保護されます。さらに、キーは特別な FIPS 承認暗号を使用して暗号化されます。

FIPS カードをサポートしているのは、Citrix ADC MPX 9700/10500/12500/15500 アプライアンスのみです。FIPS のサポートは、他の MPX アプライアンス、または SDX アプライアンスおよび VPX アプライアンスでは使用できません。この制限は、MPX 9700/10500/12500/15500 FIPS アプライアンスを除くすべての Citrix ADC MPX、SDX、および VPX アプライアンスで nShieldConnect 外部 HSM をサポートすることで対処されます。

nShield®Connect は、外部の FIPS 認定ネットワーク接続 HSM です。nShield HSM では、キーはアプリケーションキートークンとしてリモートファイルサーバー (RFS) に安全に保存され、nShield HSM 内でのみ再構成できます。

nShield HSM をすでに使用している場合は、Citrix ADC を使用して、すべてのエンタープライズおよびクラウドサービスの配信を最適化、セキュリティ保護、制御できます。

注:

- nShield HSM は FIPS 140-2 レベル 3 仕様に準拠し、MPX FIPS アプライアンスはレベル 2 仕様に準拠しています。
- nShield HSM を使用している間は、トレースを復号化できません。HSM から Citrix ADC アプライアンスへの応答を読み取ることができるのは、`hardserver`だけです。これは、暗号化されているためです。

サポートされているバージョンのマトリックス

Citrix ADC のバージョン	nShield Client Version	Hardserver バージョン	nShield Firmware Version
10.5e, 11.0, 11.1, 12.0, 12.1	11.70, 11.72	2.71.2	2.50.16, 2.51.10

アーキテクチャの概要

October 7, 2021

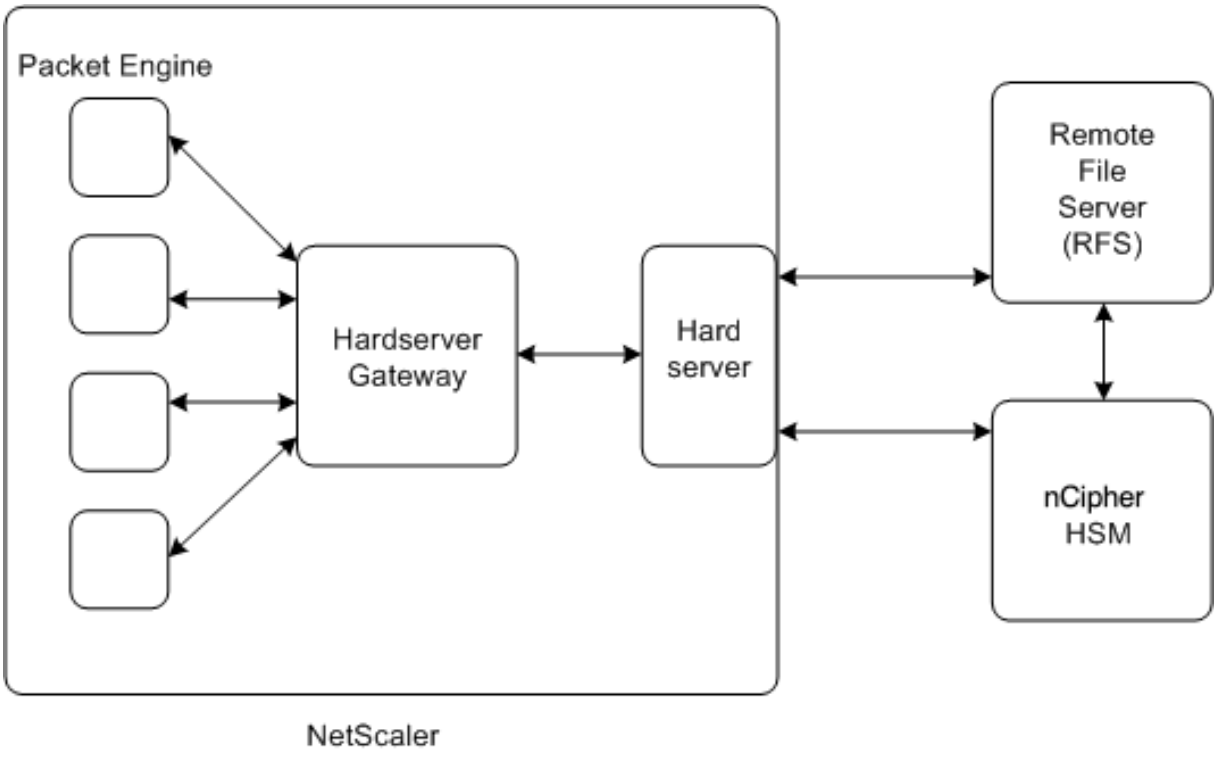
Citrix ADC-entrust 展開の一部である 3 つのエンティティは、Entrust nShield Connect モジュール、リモートファイルサーバー（RFS）、および Citrix ADC です。

Entrust nShield Connect は、ネットワークに接続されたハードウェアセキュリティモジュールです。RFS は、HSM を構成し、暗号化されたキーファイルを格納するために使用されます。

Hardserverは、Entrust が提供する独自のデーモンであり、クライアント (ADC)、Entrust HSM、RFS 間の通信に使用されます。IMPATH セキュア通信プロトコルを使用します。Hardserver Gatewayと呼ばれるゲートウェイデーモンは、Citrix ADC パケットエンジンと Hardserverの間の通信に使用されます。

注**:** このドキュメントでは、Entrust nShield Connect、Entrust HSM、および HSM という用語は同じ意味で使用されます。

次の図は、さまざまなコンポーネント間の相互作用を示しています。

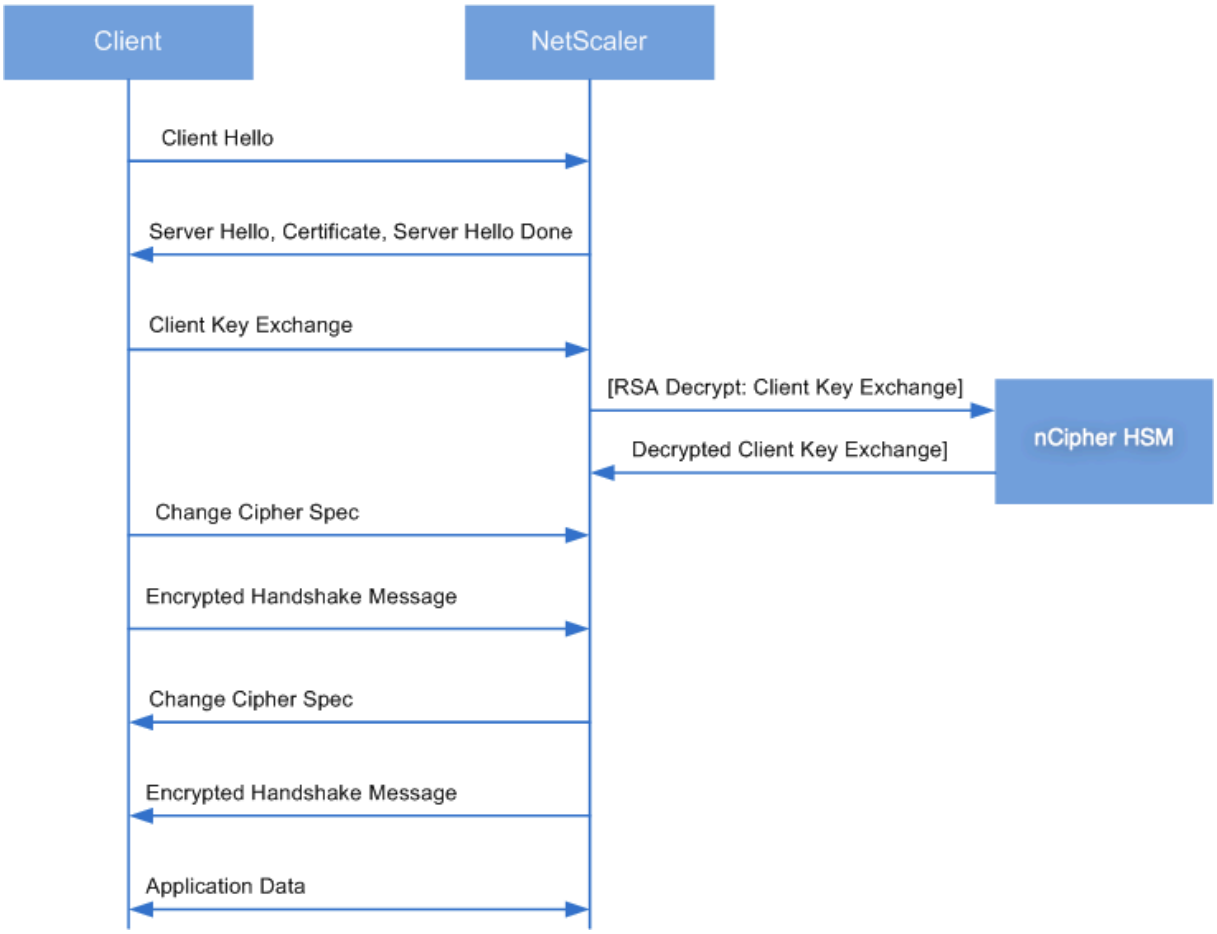


一般的なデプロイメントでは、RFS を使用して HSM によって生成されたキーを安全に保存します。キーを生成したら、それらを ADC に安全に転送し、GUI またはコマンドラインを使用してキーを HSM にロードできます。ADC 上の仮想サーバーは、Entrust を使用してクライアントキー交換を復号化し、SSL ハンドシェイクを完了します。その後、すべての SSL 操作が ADC 上で実行されます。

注：このドキュメントでは、キーとアプリケーションキートークンという用語は同じ意味で使用されています。

次の図は、Entrust HSM による SSL ハンドシェイクでのパケットフローを示しています。

図 1: Entrust HSM を使用した Citrix ADC での SSL ハンドシェイクパケットのフロー図



注**:** ADC と HSM の間の通信は、IMPATH と呼ばれる Entrust 独自の通信プロトコルを使用します。

前提条件

October 7, 2021

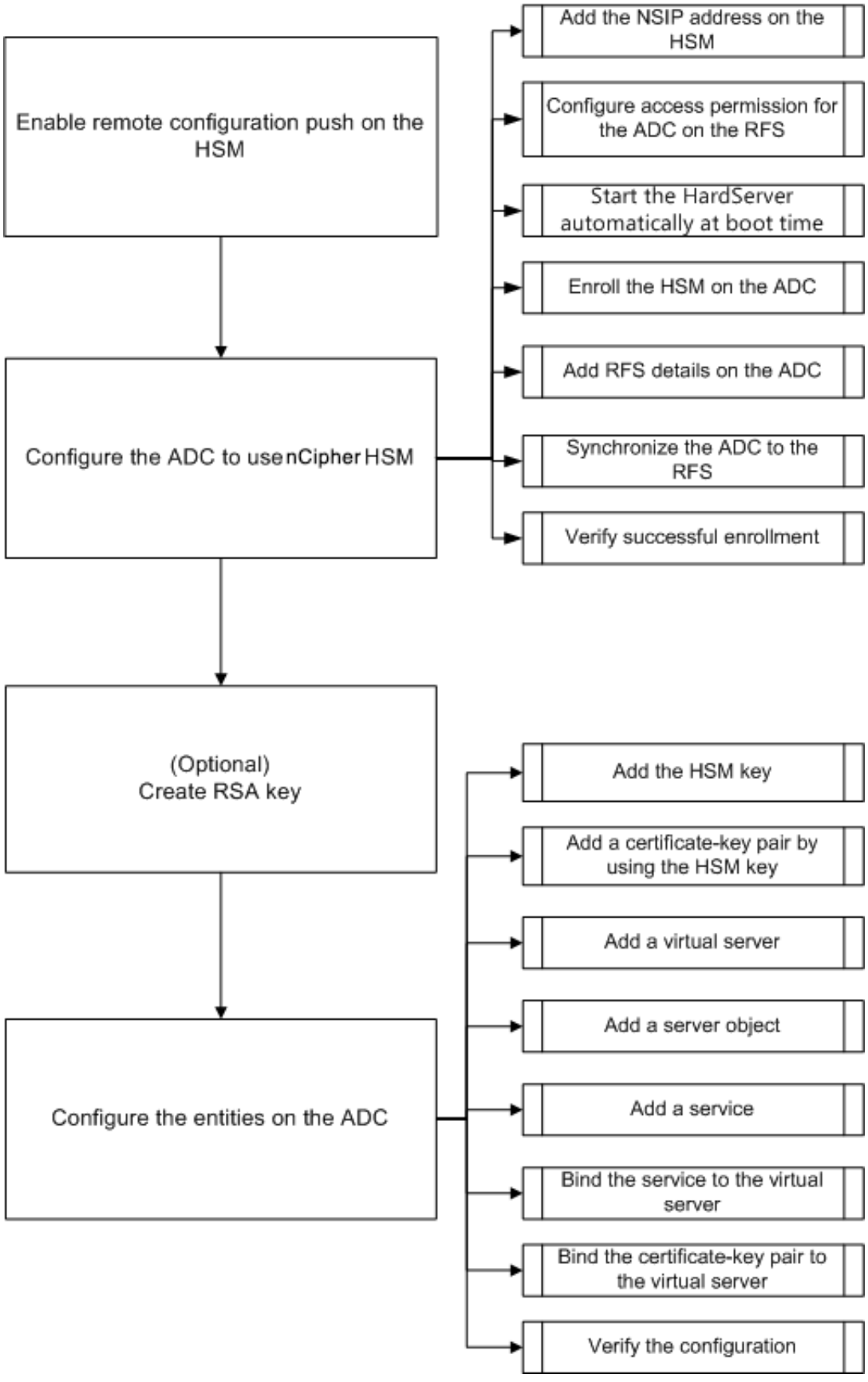
Citrix ADC で Entrust nShield Connect を使用する前に、次の前提条件が満たされていることを確認してください。

- Entrust nShield Connect デバイスがネットワークにインストールされ、すぐに使用でき、Citrix ADC からアクセスできます。つまり、NSIP アドレスは、承認されたクライアントとして HSM に追加されます。
- 使用可能なセキュリティワールドが存在します。Security World は、HSM の Entrust nShield ラインで使用されるユニークなキー管理アーキテクチャです。アプリケーションキートークンとしてキーを保護および管理し、無制限のキー容量と自動キーのバックアップとリカバリを可能にします。セキュリティワールドの作成の詳細については、『Entrust の nShield Connect クイックスタートガイド』を参照してください。このガイドは、Entrust HSM モジュールに付属する CD にも記載されています。CipherTools-Linux-dev-xx.xxx/document/nshield_connect_quick_Start_Guide.pdf.
  注： `Softcard` または token/OCS 保護されたキーは現在、Citrix ADC ではサポートされていません。
- ライセンスは、Entrust HSM に接続されているクライアントの数をサポートするために利用できます。ADC とリモートファイルサーバー（RFS）は、HSM のクライアントです。
- RFS はネットワークにインストールされており、Citrix ADC からアクセスできます。
- Entrust nShield Connect デバイス、RFS、および Citrix ADC は、ポート 9004 を介して互いに接続を開始できます。
- NetScaler リリース 10.5 ビルド 52.1115.e 以降を使用しています。
- Citrix ADC アプライアンスには、FIPS キャビウムカードは含まれていません。
  重要：委託 HSM は MPX 9700/10500/12500/15500 FIPS アプライアンスではサポートされていません。

ADC-Entrust 統合を構成する

October 7, 2021

次のフローチャートは、Entrust HSM を Citrix ADC で使用するために実行する必要があるタスクを示しています。



上記のフローチャートに示すように、次のタスクを実行します。

1. HSM でリモート設定プッシュを有効にします。
2. Entrust HSM を使用するように ADC を設定します。
    - HSM に NSIP アドレスを追加します。
    - RFS の ADC へのアクセス許可を設定します。
    - 起動時に `Hardserver` の自動開始を設定します。
    - ADC に HSM を登録します。
    - ADC に RFS の詳細を追加します。
    - ADC を RFS に同期させます。
    - Entrust HSM が ADC に正常に登録されていることを確認します。
3.（任意）HSM RSA キーを作成します。
4. Citrix ADC でエンティティを構成します。
    - HSM キーを追加します。
    - HSM キーを使用して証明書とキーのペアを追加します。
    - 仮想サーバーを追加します。
    - サーバーオブジェクトを追加します。
    - サービスを追加します。
    - サービスを仮想サーバーにバインドします。
    - 証明書とキーのペアを仮想サーバにバインドします。
    - 設定を確認します。

委託 **HSM** の設定

Entrust HSM 上の RFS の IP アドレスを指定して、RFS がプッシュする設定を受け入れるようにします。Entrust HSM の nShield Connect 前面パネルを使用して、次の手順を実行します。

Entrust HSM 上のリモートコンピュータの **IP** アドレスを指定します

1. [システム構成**] > [**設定ファイルオプション**] > [**自動プッシュを許可] に移動します。
2. **[ON]** を選択し、設定を受け入れるコンピュータ (RFS) の IP アドレスを指定します。

HSM でのリモート設定プッシュを有効にする

Entrust HSM 上の RFS の IP アドレスを指定して、RFS がプッシュする設定を受け入れるようにします。Entrust HSM の nShield Connect 前面パネルを使用して、次の手順を実行します。

Entrust HSM 上のリモートコンピュータの **IP** アドレスを指定します

1. [システム構成**] > [**設定ファイルオプション**] > [**自動プッシュを許可] に移動します。
2. **[ON]** を選択し、設定を受け入れるコンピュータ (RFS) の IP アドレスを指定します。

Entrust HSM を使用するように **ADC** を構成する

このドキュメントで使用されているサンプル値:

NSIP アドレス =10.217.2.43

HSM IP アドレスを委託 = 10.217.2.112

RFS IP アドレス =10.217.2.6

HSM に **NSIP** アドレスを追加します

通常は、nShield Connect フロントパネルを使用して、クライアントを HSM に追加します。詳細については、「nShield Connect クイックスタートガイド」を参照してください。

または、RFS を使用して、ADC をクライアントとして HSM に追加します。ADC を追加するには、RFS の HSM 構成に NSIP アドレスを追加してから、構成を HSM にプッシュする必要があります。構成をプッシュする前に、HSM の電子シリアル番号（ESN）を知っている必要があります。

HSM の ESN を取得するには、RFS で次のコマンドを実行します。

```
root@ns# /opt/nfast/bin/anonkneti <Entrust HSM IP address>
<!--NeedCopy-->
```

例**:**

```
root@ns# /opt/nfast/bin/anonkneti 10.217.2.112
BD17-C807-58D9 5e30a698f7bab3b2068ca90a9488dc4e6c78d822
<!--NeedCopy-->
```

ESN 番号は BD17-C807-58D9 です。

ESN 番号を確認したら、vi などのエディタを使用して RFS の HSM 設定ファイルを編集します。

```
vi /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

hs_clients セクションに、次のエントリを追加します。

```
# Amount of data in bytes to encrypt with a session key before session
    key# renegotiation, or 0 for unlimitied. (default=1024\*1024\*8b=8Mb
    ).
```



```
#  datalimit=INT
addr=10.217.2.43
clientperm=unpriv
keyhash=0000000000000000000000000000000000000000
esn=
timelimit=86400
datalimit=8388608
-----
<!--NeedCopy-->
```

注**:** 1 つ以上のハイフンを区切り文字として含めて、同じセクションに複数のエントリを追加します。

設定を HSM にプッシュするには、RFS で次のコマンドを実行します。

```
/opt/nfast/bin/cfg-pushnethsm --address=<Entrust HSM IP address> --
   force /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

例**:**

```
/opt/nfast/bin/cfg-pushnethsm --address=10.217.2.112 --force
                /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

RFS の **ADC** のアクセス許可を設定します

RFS で ADC のアクセス許可を設定するには、RFS で次のコマンドを実行します。

```
/opt/nfast/bin/rfs-setup --force -g --write-noauth <NetScaler IP
   address>
<!--NeedCopy-->
```

例**:**

```
[root@localhost bin]# /opt/nfast/bin/rfs-setup --force -g --write-
   noauth 10.217.2.43
Adding read-only remote_file_system entries
Ensuring the directory /opt/nfast/kmdata/local exists
Adding new writable remote_file_system entries
```



```
Ensuring the directory /opt/nfast/kmdata/local/sync-store exists
Saving the new config file and configuring the hardserver
Done
<!--NeedCopy-->
```

ADC がポート 9004 を使用して RFS と Entrust HSM の両方に到達できることを確認します。

起動時に **hardserver** の自動開始を構成する

ファイルを作成し、アプライアンスを再起動します。これで、アプライアンスを再起動するたびに、このファイルが見つかると、Hardserver が自動的に開始されます。

シェルプロンプトで、次のように入力します。

```
touch /var/opt/nfast/bin/thales_hsm_is_enrolled
<!--NeedCopy-->
```

コマンドプロンプトで入力します。

```
reboot
<!--NeedCopy-->
```

ADC に **HSM** を登録する

ディレクトリを /var/opt/nfast/bin に変更します。

HSM の詳細を ADC 構成に追加するには、ADC で次のコマンドを実行します。

```
nethsmenroll --force <Thales_nShield_Connect_ip_address> $(anonkneti <Thales_nShield_Connect_ip_address>)
```

例**:**

```
root@ns# ./nethsmenroll --force 10.217.2.112 $(anonkneti 10.217.2.112)
OK configuring hardserver's nethsm imports
<!--NeedCopy-->
```

この手順では、/var/opt/nfast/kmdata/config/config ファイルのnethsm_importsセクションにある行 # ntoken_esn=ESN の後に次のエントリを追加します。


```
...
local_module=0
remote_ip=10.217.2.112
remote_port=9004
remote_esn=BD17-C807-58D9
keyhash=5e30a698f7bab3b2068ca90a9488dc4e6c78d822
timelimit=86400
datalimit=8388608
privileged=0
privileged_use_high_port=0
ntoken_esn=
<!--NeedCopy-->
```

ディレクトリを/var/opt/nfast/bin に変更して ADC で次のコマンドを実行します。

```
touch ”thales_hsm_is_enrolled”
<!--NeedCopy-->
```

注: ADC に登録されている HSM を削除するには、次のように入力します。

```
./nethsmenroll - -remove <NETHSM-IP>
<!--NeedCopy-->
```

ADC に **RFS** の詳細を追加する

RFS の詳細を追加するには、ディレクトリを/var/opt/nfast/bin/に変更します。次に、次のコマンドを実行します。

```
./rfs-sync --no-authenticate --setup <rfs_ip_address>
<!--NeedCopy-->
```

例:

```
./rfs-sync --no-authenticate --setup 10.217.2.6
No current RFS synchronization configuration.
Configuration successfully written; new config details:
Using RFS at 10.217.2.6:9004: not authenticating.
<!--NeedCopy-->
```


この手順では、/var/opt/nfast/kmdata/コンフィグ/コンフィグ/コンフィグファイルの rfs_sync_clientセクションの # local_esn=ESN 行の後に次のエントリを追加します。

```
… …
remote_ip=10.217.2.6
remote_port=9004
use_kneti=no
local_esn=
<!--NeedCopy-->
```

注**:** ADC に登録されている RFS を削除するには、次のように入力します。

```
./rfs_sync  - remove
<!--NeedCopy-->
```

ADC を **RFS** に同期させる

すべてのファイルを同期するには、ディレクトリを次のように変更します。/var/opt/nfast/bin 次に、ADC で次のコマンドを実行します。

```
./rfs-sync  - -update
<!--NeedCopy-->
```

このコマンドは、RFS の /opt/nfast/kmdata/local ディレクトリからすべてのワールドファイル、モジュールファイル、キーファイルをフェッチし、ADC の /var/opt/nfast/kmdata/local ディレクトリに格納します。ワールドファイル、module_XXXX_XXXX_XXXX ファイル（XXXX_XXXX_XXXX は登録されている HSM の ESN）、および必要な RSA キーと証明書ファイルのみを手動でコピーすることをお勧めします。です。

Entrust HSM が **ADC** に正常に登録されていることを確認します

ADC を RFS に同期したら、次の操作を行います。

- ローカルHardserverが稼働していることを確認します。(サーバーの実行を委託)。
- 構成された HSM の状態を取得し、n_modules（モジュール数）フィールドと km 情報フィールドの値がゼロ以外であることを確認します。
- HSM が正しく登録され、ADC が使用できること（0x2 Usable 状態）を確認します。
- sigtest を使用した負荷テストは正しく実行されます。

ディレクトリを/var/opt/nfast/bin に変更し、シェルプロンプトで次のコマンドを実行します。

```
root@ns# ./chkserv root@ns# ./nfkminfo root@ns# ./sigtest
<!--NeedCopy-->
```

例については、付録を参照してください 。

HSM RSA キーの作成

HSM キーとしてサポートされているのは RSA キーのみです。

注： RFS の/opt/nfast/kmdata/localフォルダに既にキーが存在する場合は、この手順をスキップしてください。

RSA キー、自己署名証明書、および証明書署名要求 (CSR) を作成します。CSR を認証局に送信して、サーバ証明書を取得します。

次の例では、次のファイルが作成されます。

- RSA キーを埋め込む:key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
- 自己署名証明書:example_selfcert
- 証明書署名リクエスト:example_req

注:generatekeyコマンドは、厳密な FIPS 140-2 Level 3 Security World でサポートされています。キーや OCS の作成など、多くの操作を制御するには、管理者カードセット (ACS) またはオペレータカードセット (OCS) が必要です。このgeneratekeyコマンドを実行すると、ACS カードまたは OCS カードを挿入するように求められます。厳格な FIPS 140-2 Level 3 Security World の詳細については、nShield Connect ユーザーガイドを参照してください。

次の例では、Level-2 Security World を使用します。この例では、コマンドは太字です。

例:

```
[root@localhost bin]# ./generatekey embed
size: Key size? (bits, minimum 1024) [1024] > 2048
OPTIONAL: pubexp: Public exponent for RSA key (hex)? []
>
embedsavefile: Filename to write key to? []
> example
plainname: Key name? [] > example
x509country: Country code? [] > US
x509province: State or province? [] > CA
x509locality: City or locality? [] > Santa Clara
x509org: Organisation? [] > Citrix
```



```
x509orgunit: Organisation unit? [] > NS
x509dnscommon: Domain name? [] > www.citrix.com
x509email: Email address? [] > example@citrix.com
nvram: Blob in NVRAM (needs ACS)? (yes/no) [no] >
digest: Digest to sign cert req with? (md5, sha1, sha256, sha384,
    sha512)
  [default sha1] > sha512
key generation parameters:
 operation      Operation to perform               generate
 application    Application                        embed
 verify         Verify security of key             yes
 type           Key type                           RSA
 size           Key size                           2048
 pubexp         Public exponent for RSA key (hex)
 embedsavefile  Filename to write key to           example
 plainname      Key name                           example
 x509country    Country code                       US
 x509province   State or province                  CA
 x509locality   City or locality                   Santa Clara
 x509org        Organisation                       Citrix
 x509orgunit    Organisation unit                  NS
 x509dnscommon  Domain name                        www.citrix.com
 x509email      Email address                      example@citrix.com
 nvram          Blob in NVRAM (needs ACS)          no
 digest         Digest to sign cert req with       sha512
Key successfully generated.
Path to key: /opt/nfast/kmdata/local/
    key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
You have new mail in /var/spool/mail/root
<!--NeedCopy-->
```

結果:

CSR (例 _req)、自己署名証明書 (例 _selfcert)、および埋め込み形式のアプリケーションキートークンファイル (/opt/nfast/kmdata/ローカル/キー埋め込み _2ed5428ae159bdbd63f25292c7113ec2c78) を作成しました。

ADC は単純な形式のキーのみをサポートしているため、埋め込みキーを単純なキーに変換する必要があります。

埋め込みキーを単純なキーに変換するには、**RFS** で次のコマンドを実行します。

```
[root@localhost bin]# ./generatekey -r simple
from-application: Source application? (embed, simple) [embed] > embed
from-ident: Source key identifier? (
    c6410ca00af7e394157518cb53b2db46ff18ce29,
```



```
4                                          2
                                               ed5428aaeae1e159bdbd63f25292c7113ec2c78
                                               )
5    [default c6410ca00af7e394157518cb53b2db46ff18ce29]
6  > 2ed5428aaeae1e159bdbd63f25292c7113ec2c78
7  ident: Key identifier? [] > examplersa2048key
8  plainname: Key name? [] > examplersa2048key
9  key generation parameters:
10  operation         Operation to perform    retarget
11  application       Application             simple
12  verify            Verify security of key  yes
13  from-application  Source application      embed
14  from-ident        Source key identifier   2
       ed5428aaeae1e159bdbd63f25292c7113ec2c78
15  ident             Key identifier          examplersa2048key
16  plainname         Key name                examplersa2048key
17 Key successfully retargetted.
18 Path to key: /opt/nfast/kmdata/local/key_simple_examplersa2048key
19 <!--NeedCopy-->
```

> 重要:
>
> ソースキー識別子の入力を求められたら、埋め込みキーとして「**2ed5428aae1e159bd63f25292c7113ec2c78**」と入力します。

結果:

プレフィックス key_simple を持つキー (たとえば、key_simple_examplersa2048key) が作成されます。

注: examplersa2048key はキー識別子（ID）であり、ADC では HSM キー名と呼ばれます。キー識別子は一意です。すべての単純なファイルには、プレフィックス key_simple があります。

ADC でエンティティを構成する

ADC がトラフィックを処理する前に、次のことを行う必要があります。

1. 機能を有効にします。
2. サブネット IP（SNIP）アドレスを追加します。
3. ADC に HSM キーを追加します。
4. HSM キーを使用して証明書とキーのペアを追加します。
5. 仮想サーバーを追加します。
6. サーバーオブジェクトを追加します。
7. サービスを追加します。
8. サービスを仮想サーバーにバインドします。
9. 証明書とキーのペアを仮想サーバにバインドします。

10. 設定を確認します。

ADC の機能を有効にする

機能を有効にする前に、ADC にライセンスが存在している必要があります。

CLI を使用した機能の有効化

コマンドプロンプトで、次のコマンドを実行します。

```
enable feature lb
enable feature ssl
<!--NeedCopy-->
```

GUI を使用した機能の有効化

System > **Settings** に移動して **Modes and Features** グループで **Configure basic features** を選択し、**SSL Offloading** を選択します。

サブネット **IP** アドレスの追加

サブネット IP アドレスの詳細については、「サブネット IP アドレスの構成」を参照してください。

CLI を使用して **SNIP** アドレスを追加し、設定を確認します

コマンドプロンプトで、次のコマンドを実行します。

```
add ns ip <IPAddress> <netmask> -type SNIP
show ns ip
<!--NeedCopy-->
```

例**:**

```
add ns ip 192.168.17.253 255.255.248.0 -type SNIP
Done
show ns ip
        Ipaddress        Traffic Domain  Type            Mode     Arp
                Icmp     Vserver  State
```



```
5          ---------         ---------------  ----              ----     ---
                  ----     -------  ------
6  1)      192.168.17.251    0               NetScaler IP     Active
      Enabled  Enabled  NA       Enabled
7  2)      192.168.17.252    0               VIP              Active
      Enabled  Enabled  Enabled  Enabled
8  3)      192.168.17.253    0               SNIP              Active
      Enabled  Enabled  NA       Enabled
9   Done
10 <!--NeedCopy-->
```

GUI を使用して **SNIP** アドレスを追加し、構成を確認します

[システム**] > [**ネットワーク**] > [IP]** に移動し、IP アドレスを追加して、**[IP** タイプ**]** を **[**サブネット **IP]** として選択します。

HSM キーと証明書を **ADC** にコピーします

セキュアなファイル転送ユーティリティを使用して、キー (key_simple_examplersa2048key) を/var/opt/nfast/kmdata/localローカルフォルダーにコピーし、証明書 (example_selfcert) を ADC の/nsconfig/sslフォルダーに安全にコピーします。

ADC にキーを追加します

すべてのキーは、キー単純な接頭辞を持っています。ADC にキーを追加する場合は、HSM キー名として ident を使用します。たとえば、追加したキーが key_simple_XXXX の場合、HSM キー名は XXXX になります。

重要:

- HSM キー名は、埋め込みキーを単純なキー形式に変換したときに指定した ident と同じである必要があります。
- キーは、ADC の/var/opt/nfast/kmdata/local/ディレクトリに存在する必要があります。

CLI を使用した **HSM** キーの追加

シェルプロンプトで、次のコマンドを実行します。

```
1 add ssl hsmKey <hsmKeyName> -key <string>
2 <!--NeedCopy-->
```

例**:**


```
add ssl hsmKey examplersa2048key  - key key_simple_examplersa2048key
Done
<!--NeedCopy-->
```

GUI を使用して **HSM** キーを追加する

[トラフィック管理**] > [SSL] > [HSM**] に移動し、HSM キーを追加します。

ADC に証明書とキーのペアを追加する

証明書とキーのペアについては、「証明書とキーのペアの追加または更新」を参照してください。

CLI を使用した証明書とキーのペアの追加

コマンドプロンプトで、次のコマンドを実行します。

```
add ssl certKey <certkeyName> -cert <string> -hsmKey <string>
<!--NeedCopy-->
```

例**:**

```
add ssl certKey key22 -cert example_selfcert -hsmKey examplersa2048key
Done
<!--NeedCopy-->
```

GUI を使用した証明書とキーのペアの追加

Traffic Management > SSL > Certificates 証明書キーペアを追加します。

仮想サーバーの追加

仮想サーバーの詳細については、「SSL 仮想サーバー構成」を参照してください。

CLI を使用した **SSL** ベースの仮想サーバーの構成

コマンドプロンプトで、次のコマンドを実行します。


```
add lb vserver <name> <serviceType> <IPAddress> <port>
<!--NeedCopy-->
```

例:

```
add lb vserver v1 SSL 192.168.17.252 443
<!--NeedCopy-->
```

GUI を使用して **SSL** ベースの仮想サーバーを構成する

[トラフィック管理**] > [**負荷分散**] > [**仮想サーバー**]** に移動し、仮想サーバーを作成し、プロトコルを SSL として指定します。

サーバーオブジェクトの追加

ADC にサーバーオブジェクトを追加する前に、バックエンドサーバーが作成されていることを確認してください。次の例では、Linux システムで組み込みの Python HTTP サーバーモジュールを使用しています。

例:

```
%python  -m SimpleHTTPServer 80
<!--NeedCopy-->
```

CLI を使用したサーバーオブジェクトの追加

コマンドプロンプトで、次のコマンドを実行します。

```
add server <name> <IPAddress>
<!--NeedCopy-->
```

例:

```
add server s1 192.168.17.246
<!--NeedCopy-->
```


GUI を使用したサーバーオブジェクトの追加

Traffic Management > Load Balancing > Servers に移動してサーバーを追加します。

サービスを追加する

詳細については、「サービスの設定」を参照してください。

CLI を使用したサービスの設定

コマンドプロンプトで、次のコマンドを実行します。

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

例:

```
add service sr1 s1 HTTP 80
<!--NeedCopy-->
```

GUI を使用したサービスの構成

[トラフィック管理] **>** [負荷分散] **>** [サービス] に移動し、サービスを作成します。

サービスを仮想サーバーにバインドする

詳細については、「SSL 仮想サーバーへのサービスのバインド」を参照してください。

CLI を使用してサービスを仮想サーバーにバインドします

コマンドプロンプトで、次のコマンドを実行します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

例:


```
bind lb vserver v1 sr1
<!--NeedCopy-->
```

GUI を使用してサービスを仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 仮想サーバを開き、[Services] ペインをクリックして、サービスを仮想サーバにバインドします。

証明書とキーのペアを **ADC** の仮想サーバーにバインドします

詳細については、SSL 仮想サーバーに証明書とキーのペアをバインドするを参照してください。

CLI を使用して証明書とキーのペアを仮想サーバーにバインドする

コマンドプロンプトで、次のコマンドを実行します。

```
bind ssl vserver <vServerName> -certkeyName <string>
<!--NeedCopy-->
```

例**:**

```
bind ssl vserver v1 -certkeyName key22
Warning: Current certificate replaces the previous binding
<!--NeedCopy-->
```

GUI を使用して証明書とキーのペアを仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. SSL 仮想サーバーを開き、[詳細設定] で [**SSL** 証明書] をクリックします。
3. サーバー証明書を仮想サーバーにバインドします。

構成を確認します

CLI を使用して設定を表示するには、次の手順を実行します。

コマンドプロンプトで、次のコマンドを実行します。


```
show lb vserver <name>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

例:

```
show lb vserver v1
        v1 (192.168.17.252:443) - SSL   Type: ADDRESS
        State: UP
        Last state change was at Wed Oct 29 03:11:11 2014
        Time since last state change: 0 days, 00:01:25.220
        Effective State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: Bound service's state
           changed to UP
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
        L2Conn: OFF
        Skip Persistency: None
        IcmpResponse: PASSIVE
        RHIstate: PASSIVE
        New Service Startup Request Rate: 0 PER_SECOND, Increment
           Interval: 0
        Mac mode Retain Vlan: DISABLED
        DBS_LB: DISABLED
        Process Local: DISABLED
        Traffic Domain: 0

1) sr1 (192.168.17.246: 80) - HTTP State: UP    Weight: 1
Done
<!--NeedCopy-->
```



```
sh ssl vserver v1
        Advanced SSL configuration for VServer v1:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 0
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        SSLv2: DISABLED  SSLv3: DISABLED  TLSv1.0: ENABLED  TLSv1.1:
           DISABLED  TLSv1.2: DISABLED
        Push Encryption Trigger: Always
        Send Close-Notify: YES

        ECC Curve: P_256, P_384, P_224, P_521

1)      CertKey Name: key22       Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

GUI を使用して設定を表示するには、次の手順を実行します。

Traffic Management > Load Balancing > Virtual Servers に移動して SSL 仮想サーバーをダブルクリックして開いて、構成を表示します。

制限事項

October 7, 2021

- SSL バージョン 3（SSLv3）は MPX アプライアンスではサポートされませんが、VPX 仮想アプライアンスではサポートされています。SDX アプライアンスでプロビジョニングされた VPX インスタンスは、SSL チップがインスタンスに割り当てられていない場合にのみ SSLv3 をサポートします。
- Export 暗号はサポートされていません。
- HSM キーを使用した SSL サーバキー交換はサポートされていません。
- 最後に設定を保存した後にキーを追加または削除した場合は、ウォームリスタートを実行する前に設定を保存する必要があります。設定を保存しない場合、ADC と HSM の間にキーの不一致があります。

- HSM キーを DTLS 仮想サーバにバインドすることはできません。
- HSM キーを使用して作成された証明書とキーのペアを SSL サービスにバインドすることはできません。
- GUI を使用して、ADC を HSM のクライアントとして登録したり、設定ユーティリティから HSM のステータスを確認したりすることはできません。
- リリース 11 ビルド 62.x から、SSL 再ネゴシエーションがサポートされています。
- HSM キーを使用して作成された証明書とキーのペアを使用して OCSP 要求に署名することはできません。
- HSM キーを使用した証明書バンドルはサポートされていません。
- HSM キーと証明書が一致しない場合、エラーは表示されません。したがって、証明書とキーのペアを追加するときは、HSM キーと証明書が一致していることを確認する必要があります。
- クラスタリングおよび管理パーティションはサポートされていません。

付録

October 7, 2021

例:

注: 次の例では、コマンドは太字です。

```
root@ns# ./chkserv
nCipher server running
root@ns# ./nfkminfo
World
 generation  2
 state       0x17a70000 Initialised Usable Recovery PINRecovery !
    ExistingClient RTC NVRAM FTO !AlwaysUseStrongPrimes SEEDebug
 n_modules   1
 hknso       cbec8c0c56c6b5e76b73147ef02d34a661eaa044
 hkm     bbb8d4839da5782be4d092735a7535538834dc91 (type Rijndael)
 hkmwk       1d572201be533ebc89f30fdd8f3fac6ca3395bf0
 hkre        01f21ecf43933ffdd45e74c3883525176c5c439c
 hkra        ac8ec5ee6bce00991bd97adce2091d9739b9b452
 hkmc        cf1b509abaad91995ed202d8f36613fc99433155
 hkp         c20910b2ed1ca62d6a2b0db67052a05f7bbfeb43
 hkrtc       bd811020a7c2f8df435a481c3767a89c2e13bc4f
 hknv        278b8012e48910d518a9ee91cff57233fb0c9093
 hkdsee      12230b0e31e3cec66324c0815f782cfb9249edd5
 hkfto       89dd6250b3d6149bcd15606f4553085e2fd6271a
 hkmnull     0100000000000000000000000000000000000000
 ex.client   none
 k-out-of-n  1/2
```



```
 other quora m=1 r=1 p=1 nv=1 rtc=1 dsee=1 fto=1
 createtime  2014-02-28 21:05:32
 nso timeout 10 min
 ciphersuite DLf1024s160mRijndael

Module #1
 generation 2
 state      0x2 Usable
 flags      0x10000 ShareTarget
 n_slots    2
 esn        BD17-C807-58D9
 hkml       70289a6edba00ddc7e3f6d6f5a49edc963e822f2

Module #1 Slot #0 IC 0
 generation    1
 phystype      SmartCard
 slotlistflags 0x2 SupportsAuthentication
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

Module #1 Slot #1 IC 0
 generation    1
 phystype      SoftToken
 slotlistflags 0x0
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

No Pre-Loaded Objects

root@ns# ./sigtest
 Hardware module #1 speed index 5792 recommended minimum queue 19
Found 1 module; using 19 jobs
Making 1024-bit RSAPrivate key on module #1;
  using Mech_RSApPKCS1 and PlainTextType_Bignum.
Generated and exported key from module #1.
Imported keys on module #1
 1,       3059 1223.6, 3059 overall
```



```
2,       8698 2989.76, 4349 overall
3,      14396 4073.06, 4798.67 overall
4,      20091 4721.83, 5022.75 overall
5,      25799 5116.3, 5159.8 overall
6,      31496 5348.58, 5249.33 overall
7,      37192 5487.55, 5313.14 overall
8,      42780 5527.73, 5347.5 overall
9,      45777 4515.44, 5086.33 overall
10,      51457 4981.26, 5145.7 overall
11,      57151 5266.36, 5195.55 overall
12,      62813 5424.61, 5234.42 overall
13,      68496 5527.97, 5268.92 overall
14,      74182 5591.18, 5298.71 overall
15,      79832 5614.71, 5322.13 overall
16,      85518 5643.23, 5344.88 overall
17,      88412 4543.54, 5200.71 overall
18,      94086 4995.72, 5227 overall
19,      99778 5274.23, 5251.47 overall
20,     105469 5440.94, 5273.45 overall
21,     111133 5530.16, 5292.05 overall
22,     116838 5600.1, 5310.82 overall
23,     122522 5633.66, 5327.04 overall
24,     128175 5641.4, 5340.62 overall
25,     131072 4543.64, 5242.88 overall
26,     136762 5002.18, 5260.08 overall
27,     142415 5262.51, 5274.63 overall
28,     148125 5441.51, 5290.18 overall
29,     153816 5541.3, 5304 overall
30,     159414 5563.98, 5313.8 overall
<!--NeedCopy-->
```

Thales Luna Network ハードウェアセキュリティモジュールのサポート

May 9, 2022

FIPS 以外の Citrix ADC アプライアンスは、サーバーの秘密キーをハードディスクに保存します。FIPS アプライアンスでは、キーはハードウェアセキュリティモジュール (HSM) と呼ばれる暗号化モジュールに格納されます。HSM にキーを保存すると、物理攻撃やソフトウェア攻撃からキーを保護できます。さらに、キーは FIPS 承認の特殊な暗号で暗号化されます。

Citrix ADC MPX/SDX 14000 FIPS アプライアンスのみが FIPS カードをサポートします。FIPS のサポートは、他の MPX/SDX アプライアンスまたは Citrix ADC VPX アプライアンスでは使用できません。この制限は、MPX/SDX

14000 FIPS アプライアンスおよびインテル Coleto SSL チップベースプラットフォームのサポートに記載されているアプライアンスを除くすべての Citrix ADC MPX、SDX、および VPX アプライアンスで Thales Luna ネットワーク HSM をサポートすることで解決されます。

Thales Luna ネットワーク HSM は、重要な暗号化キーを保護し、幅広いセキュリティアプリケーションにわたって機密性の高い暗号化操作を高速化するように設計されています。

サポートされているバージョンのマトリックス

Citrix ADC バージョン	ソフトウェアアプライアンスバージョン	ファームウェアバージョン	クライアントのバージョン
11.1, 12.0, 12.1	5.2.3-1	6.2.1	6.0.0
11.1, 12.0, 12.1	6.2.2-5	6.10.9	6.2.2
13.0	7.2.0-220	7.0.3	7.2.2 (7.2.0-220)

前提条件

October 7, 2021

Citrix ADC で Thales Luna ネットワーク HSM を使用する前に、次の前提条件が満たされていることを確認してください。

- Thales Luna ネットワーク HSM がネットワークにインストールされ、すぐに使用でき、Citrix ADC からアクセスできます。つまり、NSIP アドレスまたは SNIP アドレスが、HSM で許可されたクライアントとして追加されます。
- ライセンスは、HSM 上で必要な数のパーティションをサポートするために利用できます。
- Thales Luna ネットワーク HSM と Citrix ADC は、ポート 1792 を介して相互に接続を開始できます。
- NetScaler リリース 11.1 以降を使用しています。
- Citrix ADC アプライアンスには、FIPS キャビウムカードは含まれていません。

> 重要
>
> Thales Luna ネットワーク HSM は MPX 9700/10500/12500/15500 FIPS アプライアンスではサポートされていません。

ADC で Thales Luna クライアントを構成する

June 24, 2022

Thales Luna HSM を構成し、必要なパーティションを作成したら、クライアントを作成してパーティションに割り当てる必要があります。まず、Citrix ADC で Thales Luna クライアントを構成し、Thales Luna クライアントと Thales Luna HSM の間のネットワークトラストリンク（NTL）を設定します。設定例については、付録に記載されています。

1. ディレクトリを /var/safenet に変更し、Thales Luna クライアントをインストールします。シェルプロンプトで、次のように入力します。

```
cd /var/safenet
<!--NeedCopy-->
```

Thales Luna クライアントバージョン 6.0.0 をインストールするには、次のように入力します。

```
install_client.sh -v 600
<!--NeedCopy-->
```

Thales Luna クライアントバージョン 6.2.2 をインストールするには、次のように入力します。

```
install_client.sh -v 622
<!--NeedCopy-->
```

Thales Luna クライアントバージョン 7.2.2 をインストールするには、次のように入力します。

```
install_client.sh -v 722
<!--NeedCopy-->
```

2. Thales Luna クライアント（ADC）と HSM の間の NTL を設定します。

‘/var/safenet/’ ディレクトリを作成したら、ADC で次のタスクを実行します。

a) ディレクトリを ‘/var/safenet/config/’ に変更し、’safenet_config’ スクリプトを実行します。シェルプロンプトで、次のように入力します。

```
cd /var/safenet/config

sh safenet_config
<!--NeedCopy-->
```


このスクリプトは「chrystoki.conf」ファイルを /etc/ ディレクトリにコピーします。また、'/usr/lib/' ディレクトリにシンボリックリンク 'libCryptoki2_64.so' を生成します。

b) ADC と Thales Luna HSM の間で証明書とキーを作成し、転送します。

安全に通信するには、ADC と HSM が証明書を交換する必要があります。ADC で証明書とキーを作成し、HSM に転送します。HSM 証明書を ADC にコピーします。

i) ディレクトリを /var/safenet/safenet/lunaclient/bin に変更します。

ii) ADC で証明書を作成します。シェルプロンプトで、次のように入力します。

```
./vtl createCert -n <ip address of Citrix ADC>
<!--NeedCopy-->
```

このコマンドはまた、証明書とキーパスを「/etc/chrystoki.conf」ファイルに追加します。

iii) この証明書を HSM にコピーします。シェルプロンプトで、次のように入力します。

```
scp /var/safenet/safenet/lunaclient/cert/client/<ip address of NS
   >.pem <LunaSA_HSM account>@<IP address of Luna SA>
<!--NeedCopy-->
```

iv) HSM 証明書を Citrix ADC にコピーします。シェルプロンプトで、次のように入力します。

```
scp <HSM account>@<HSM IP>:server.pem  /var/safenet/safenet/
   lunaclient/server_<HSM ip>.pem
<!--NeedCopy-->
```

3. Citrix ADC をクライアントとして登録し、Thales Luna HSM 上のパーティションを割り当てます。

   HSM にログオンし、クライアントを作成します。クライアント IP として NSIP を入力します。このアドレスは、HSM への証明書の転送元の ADC の IP アドレスである必要があります。クライアントが正常に登録されたら、パーティションを割り当てます。HSM で次のコマンドを実行します。

   a) SSH を使用して Thales Luna HSM に接続し、パスワードを入力します。

   b) Thales Luna HSM に Citrix ADC を登録します。クライアントは HSM 上に作成されます。IP アドレスはクライアントの IP アドレスです。つまり、NSIP アドレスです。

   プロンプトで、次のように入力します。


```
client register  - client <client name> -ip <Citrix ADC ip>
<!--NeedCopy-->
```

c) パーティションリストからクライアントにパーティションを割り当てます。使用可能なパーティションを表示するには、次のように入力します。

```
<luna_sh> partition list
<!--NeedCopy-->
```

このリストからパーティションを割り当てます。タイプ:

```
<lunash:> client assignPartition -client <Client Name> -par <
   Partition Name>
<!--NeedCopy-->
```

4. HSM とその証明書を Citrix ADC に登録します。

   ADC で、ディレクトリを「/var/safenet/safenet/lunaclient/bin」に変更し、シェルプロンプトで次のように入力します。

```
./vtl addserver -n <IP addr of HSM> -c /var/safenet/safenet/
   lunaclient/server_<HSM_IP>.pem
<!--NeedCopy-->
```

   ADC に登録されている HSM を削除するには、次のように入力します。

```
./vtl deleteServer -n <HSM IP> -c <cert path>
<!--NeedCopy-->
```

   ADC に設定されている HSM サーバーを一覧表示するには、次のように入力します。

```
./vtl listServer
<!--NeedCopy-->
```


> 注:
>
> vtlを使用して HSM を削除する前に、その HSM のすべてのキーがアプライアンスから手動で削除されていることを確認してください。HSM サーバーを削除すると、HSM キーは削除できません。

5. ADC と HSM の間のネットワーク信頼リンク (NTL) 接続を確認します。シェルプロンプトで、次のように入力します。

```
./vtl verify
<!--NeedCopy-->
```

検証に失敗した場合は、すべての手順を確認します。エラーは、クライアント証明書の IP アドレスが正しくないことが原因です。

6. 構成を保存します。

前の手順では、「/etc/chrystoki.conf」設定ファイルを更新します。このファイルは、ADC が起動すると削除されます。構成をデフォルト構成ファイルにコピーします。このファイルは、ADC の再起動時に使用されます。

シェルプロンプトで、次のように入力します。

```
root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

推奨される方法は、Thales Luna 関連の設定が変更されるたびにこのコマンドを実行することです。

7. Thales Luna ゲートウェイプロセスを開始します。

シェルプロンプトで、次のように入力します。

```
sh /var/safenet/gateway/start_safenet_gw
<!--NeedCopy-->
```

8. 起動時に Gateway デーモンの自動起動を設定します。

この ADC で Thales Luna HSM が設定されていることを示す「safenet_is_reglarbed」ファイルを作成します。ADC が再起動し、このファイルが見つかるたびに、Gateway が自動的に起動します。

シェルプロンプトで、次のように入力します。

```
touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```


9. Citrix ADC アプライアンスを再起動します。コマンドプロンプトで入力します。

```
reboot
<!--NeedCopy-->
```

ADC の高可用性セットアップで Thales Luna HSM を構成する

October 7, 2021

Thales Luna HSM をハイアベイラビリティ（HA）に設定することで、いずれかのデバイスをすべて使用できない場合でも、サービスが中断されないようにします。HA 設定では、各 HSM はアクティブ-アクティブモードで HA グループに加入します。HA セットアップの Thales Luna HSM は、すべてのグループメンバーのロードバランシングを提供し、パフォーマンスと応答時間を向上させながら、高可用性サービスを保証します。詳細については、Thales Luna のセールスおよびサポートにお問い合わせください。

前提条件：

- 最低 2 台の Thales Luna HSM デバイス。HA グループ内のすべてのデバイスには、PED（信頼できるパス）認証またはパスワード認証が必要です。HA グループでの信頼できるパス認証とパスワード認証の組み合わせはサポートされていません。
- ラベル（名前）が異なる場合でも、各 HSM デバイスのパーティションには同じパスワードが必要です。
- HA 内のすべてのパーティションをクライアント（Citrix ADC アプライアンス）に割り当てる必要があります。

ADC での Thales Luna クライアントの設定の説明に従って、ADC で Thales Luna クライアントを構成した後、次の手順を実行して HA で Thales Luna HSM を設定します。

1. Citrix ADC のシェルプロンプトで、`lunacm`（/usr/safenet/lunaclient/bin）を起動します。

   例：

```
root@ns# cd /var/safenet/safenet/lunaclient/bin/

root@ns# ./lunacm
<!--NeedCopy-->
```

2. パーティションのスロット ID を識別します。使用可能なスロット（パーティション）を一覧表示するには、次のように入力します。

```
lunacm:> slot list
<!--NeedCopy-->
```


例:

```
Slot Id ->              0
HSM Label ->            trinity-p1
HSM Serial Number ->    481681014
HSM Model ->            LunaSA 6.2.1
HSM Firmware Version -> 6.10.9
HSM Configuration ->    Luna SA Slot (PED) Signing With
   Cloning Mode
HSM Status ->           OK

Slot Id ->              1
HSM Label ->            trinity-p2
HSM Serial Number ->    481681018
HSM Model ->            LunaSA 6.2.1
HSM Firmware Version -> 6.10.9
HSM Configuration ->    Luna SA Slot (PED) Signing With
   Cloning Mode
HSM Status ->           OK

 Slot Id ->              2
 HSM Label ->            neo-p1
 HSM Serial Number ->    487298014
 HSM Model ->            LunaSA 6.2.1
 HSM Firmware Version -> 6.10.9
 HSM Configuration ->    Luna SA Slot (PED) Signing With
    Cloning Mode
HSM Status ->           OK

Slot Id ->              3
HSM Label ->            neo-p2
HSM Serial Number ->    487298018
HSM Model ->            LunaSA 6.2.1
HSM Firmware Version -> 6.10.9
HSM Configuration ->    Luna SA Slot (PED) Signing With
   Cloning Mode
HSM Status ->           OK

Slot Id ->              7
HSM Label ->            hsmha
HSM Serial Number ->    1481681014
HSM Model ->            LunaVirtual
HSM Firmware Version -> 6.10.9
```



```
38      HSM Configuration ->    Luna Virtual HSM (PED) Signing With
           Cloning Mode
39      HSM Status ->           N/A - HA Group
40
41      Slot Id ->              8
42      HSM Label ->            newha
43      HSM Serial Number ->    1481681018
44      HSM Model ->            LunaVirtual
45      HSM Firmware Version -> 6.10.9
46      HSM Configuration ->    Luna Virtual HSM (PED) Signing With
           Cloning Mode
47      HSM Status ->           N/A - HA Group
48
49      Current Slot Id: 0
50  <!--NeedCopy-->
```

3. HA グループを作成します。最初のパーティションは、プライマリパーティションと呼ばれます。2 つ以上のセカンダリパーティションを追加できます。

```
1  lunacm:> hagroup createGroup -slot <slot number of primary
       partition> -label <group name> -password <partition password >
2
3  lunacm:> hagroup createGroup -slot 1 -label gp12 -password ******
4  <!--NeedCopy-->
```

4. セカンダリメンバー（HSM パーティション）を追加します。HA グループに追加するすべてのパーティションに対して、この手順を繰り返します。

```
1  lunacm:> hagroup addMember -slot <slot number of secondary
       partition to be added> -group <group name> -password <partition
        password>
2  <!--NeedCopy-->
```

コード:

```
1  lunacm:> hagroup addMember -slot 2 -group gp12 -password ******
2  <!--NeedCopy-->
```

5. HA 専用モードを有効にします。


```
lunacm:> hagroup HAOnly  -enable
<!--NeedCopy-->
```

6. アクティブリカバリモードを有効にします。

```
lunacm:.>hagroup recoveryMode  -mode active
<!--NeedCopy-->
```

7. 自動リカバリ間隔を秒単位で設定します。デフォルト値は 60 秒です。

```
lunacm:.>hagroup interval  -interval <value in seconds>
<!--NeedCopy-->
```

例:

```
lunacm:.>hagroup interval  -interval 120
<!--NeedCopy-->
```

8. リカバリの再試行回数を設定します。値が-1 の場合、再試行回数は無限になります。

```
lunacm:> hagroup retry -count <xxx>
<!--NeedCopy-->
```

例:

```
lunacm:> hagroup retry -count 2
<!--NeedCopy-->
```

9. 設定を`Chrystoki.conf`から SafeNet 設定ディレクトリにコピーします。

```
cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

10. ADC アプライアンスを再起動します。


```
reboot
<!--NeedCopy-->
```

HA で Thales Luna HSM を設定した後、ADC での詳細設定については、その他の ADC の設定を参照してください。

Other ADC configuration

October 7, 2021

1. HSM でキーを生成します。

   サードパーティ製のツールを使用して、HSM 上にキーを作成します。

2. ADC に HSM キーを追加します。

   重要 # 文字は、キー名ではサポートされていません。キー名にこの文字が含まれている場合、キーのロード操作は失敗します。

   CLI を使用して **Thales Luna HSM** キーを追加するには、次の手順を実行します。

   コマンドプロンプトで入力します。

   ```
   add ssl hsmkey <KeyName> -hsmType SAFENET -serialNum <serial #> -
       password
   <!--NeedCopy-->
   ```

   各項目の意味は次の通りです：

   -keyName は、サードパーティのツールを使用して HSM 上に作成されるキーです。

   -serialNum は、キーが生成される HSM 上のパーティションのシリアル番号です。

   注：高可用性設定の HSM の場合は、高可用性グループのシリアル番号を使用します。

   -password は、キーが存在するパーティションのパスワードです。

   GUI を使用して **Thales Luna HSM** キーを追加するには、次の手順を実行します。

   [トラフィック管理] > [**SSL**] > [**HSM**] に移動し、HSM キーを追加します。HSM タイプを **SAFENET** として指定する必要があります。

3. ADC に証明書とキーのペアを追加します。まず、サードパーティのツールを使用して、キーに関連付けられた証明書を生成します。次に、証明書を ADC の /nsconfig/ssl/ ディレクトリにコピーします。

   注**:** キーは HSM キーである必要があります。

   CLI を使用して **ADC** に証明書キー・ペアを追加するには、次のようにします。

コマンドプロンプトで入力します。

```
add ssl certkey <CertkeyName> -cert <cert name> -hsmkey <KeyName>
<!--NeedCopy-->
```

GUI を使用して **ADC** に証明書キー・ペアを追加するには:

a) [トラフィック管理] > [**SSL**] に移動します。
b) [はじめに] で、[証明書のインストール **(HSM)**] を選択し、HSM キーを使用して証明書とキーのペアを作成します。

4. 仮想サーバーを作成し、証明書とキーのペアをこの仮想サーバーにバインドします。

仮想サーバーの作成の詳細については、「SSL 仮想サーバー構成」をクリックしてください。

証明書とキーのペアの追加の詳細については、[証明書とキーのペアの追加または更新] をクリックします。

証明書とキーのペアを SSL 仮想サーバーにバインドする方法については、「証明書とキーのペアを SSL 仮想サーバーにバインドする」をクリックします。

高可用性セットアップの **Citrix ADC** アプライアンス

October 7, 2021

Thales Luna HSM 構成を使用して Citrix ADC アプライアンスで高可用性（HA）セットアップを構成するには、次のいずれかの方法があります。

- まず、同じ HSM とパーティションを使用して、2 つのノードで Thales Luna HSM を構成します。次に、HA ペアを作成します。最後に、キー、証明書とキーのペア、仮想サーバーなどの Citrix ADC 構成をプライマリノードに追加します。
- Thales Luna HSM が Citrix ADC 構成のあるノードで既に構成されている場合は、他のノードに同様の構成を追加します。最初のノードから別のノードに「/var/safenet/sfgw_ident_file」をコピーし、safenet_gw バイナリを再起動します。Gateway が起動して実行したら、HA セットアップでノードを追加します。

制限事項

October 7, 2021

1. HSM の追加や削除、高可用性セットアップの作成など、既存のセットアップで HSM 関連の設定を変更する場合は、「/etc/Chrystoki.conf」を「/var/safenet/config」にコピーする必要があります。

2. HSM を追加、削除、または再起動した後、'/var/safenet/gateway/safenet_gw' バイナリを再起動する必要があります。Gateway バイナリを再起動しない場合、HSM は追加後または再起動後にトラフィックを処理しません。
3. 現在の '/var/safenet/gateway/safenet_gw' バイナリを再起動または停止するには、次を使用します

```
kill  - SIGTERM <PID>
kill  - SIGINT <PID>
<!--NeedCopy-->
```

   重要`kill -9 <PID>` または`kill -6 <PID>` を使用しないでください。

4. 既存の HSM を ADC から削除する前に、その HSM に関連付けられているすべてのキーと証明書とキーのペアを ADC から削除します。これらのファイルは、HSM を削除した後で ADC から削除できません。
5. スタンドアロンの Citrix ADC アプライアンスでは、HA の Thales Luna HSM が Luna バージョン 6.2 以降でサポートされています。
6. EXPORT 暗号はサポートされていません。
7. 証明書とキーのペアの更新操作はサポートされていません。
8. サードパーティのツールで HSM キーを生成する場合、秘密キーと公開キーの名前は同じである必要があります。アプライアンスに HSM キーを追加するときは、この名前をキー名として指定します。
9. ## 文字は、キー名ではサポートされていません。
10. クラスタパーティションと管理パーティションはサポートされていません。

付録

October 7, 2021

出力を含むサンプルコマンド:

スクリプトの実行

```
     root@ns# pwd
    /var/safenet/config
    root@ns# sh safenet_config
<!--NeedCopy-->
```


証明書の作成

```
1     root@ns# cd /var/safenet/safenet/lunaclient/bin
2     root@ns# ./vtl createcert -n 10.102.59.175
3     Private Key created and written to: /var/safenet/safenet/lunaclient
         /cert/client/10.102.59.175Key.pem
4     Certificate created and written to: /var/safenet/safenet/lunaclient
         /cert/client/10.102.59.175.pem
5  <!--NeedCopy-->
```

証明書を **HSM** にコピーします

```
1     root@ns# scp /var/safenet/safenet/lunaclient/cert/client
         /10.102.59.175.pem admin@10.217.2.7:
2     admin@10.217.2.7's password:
3
4     10.102.59.175.pem          100%  818     0.8KB/s   00:00
5  <!--NeedCopy-->
```

証明書とキーを **HSM** から **Citrix ADC** アプライアンスにコピーします

```
1     root@ns# scp admin@10.217.2.7:server.pem /var/Thales Luna/safenet/
         lunaclient/server.2.7.pem
2     admin@10.217.2.7's password:
3
4     server.pem            100% 1164     1.1KB/s   00:01
5  <!--NeedCopy-->
```

SSH を使用して **Thales Luna HSM** に接続する

```
1     ssh admin@10.217.2.7
2     Connecting to 10.217.2.7:22...
3     Connection established.
4     To escape to local shell, press 'Ctrl+Alt+]'.
5
6     Last login: Thu Jun 23 02:20:29 2016 from 10.252.243.11
7
8     Luna SA 5.2.3-1 Command Line Shell - Copyright (c) 2001-2014
         SafeNet, Inc. All rights reserved.
```



```
9
10      [Safenet1] lunash:>hsm login
11
12
13        Please enter the HSM Administrators' password:
14        > *******
15
16      'hsm login' successful.
17
18
19      Command Result : 0 (Success)
20      [Safenet1] lunash:>
21  <!--NeedCopy-->
```

Thales Luna HSM に **Citrix ADC** を登録する

```
1       [Safenet1] lunash:>client register -client ns175 -ip 10.102.59.175
2
3       'client register' successful.
4
5
6       Command Result : 0 (Success)
7       [Safenet1] lunash:>
8   <!--NeedCopy-->
```

パーティション・リストからクライアントにパーティションを割り当てる

```
1       [Safenet1] lunash:>client assignPartition -client ns175 -partition
           p2
2
3       'client assignPartition' successful.
4
5
6       Command Result : 0 (Success)
7       [Safenet1] lunash:>
8   <!--NeedCopy-->
```

Citrix ADC 上の証明書を使用して **HSM** を登録します


```
    root@ns# ./vtl addserver -n 10.217.2.7 -c /var/safenet/safenet/
       lunaclient/server.2.7.pem

    New server 10.217.2.7 successfully added to server list.
<!--NeedCopy-->
```

ADC と **HSM** の間のネットワーク信頼リンク（**NTL**）の接続を確認します

```
    root@ns# ./vtl verify

    The following Luna SA Slots/Partitions were found:

    Slot        Serial #                  Label
    ====      ================          =====
       0              477877010          p2
<!--NeedCopy-->
```

構成を保存します

```
    root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

起動時に **Gateway** デーモンの自動起動を設定する

```
    touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```

よくある質問

October 7, 2021

- **Thales Luna** プロセスが実行されていることを確認するにはどうすればよいですか?

  Citrix ADC シェルプロンプトで、次のように入力します。


```
ps  - aux | grep safenet_gw
<!--NeedCopy-->
```

- **ADC** と **HSM** の間のネットワーク信頼リンク（**NTL**）接続を確認するにはどうすればよいですか**?**

  Thales Luna を設定したら、ディレクトリを「/var/safenet/safenet/lunaclient/bin」に変更し、次のように入力します。

```
./vtl verify
<!--NeedCopy-->
```

Azure Key Vault のサポート

December 7, 2021

Citrix ADC アプライアンスは、オンプレミス展開用に外部 HSM（SafeNet および Thales）と統合されます。クラウドデプロイの場合、ADC アプライアンスは Azure Key Vault と統合されます。アプライアンスは、パブリッククラウドドメインでの秘密キーの管理とセキュリティを容易にするために、秘密キーを Key Vault に保存します。複数のデータセンターやクラウドプロバイダーに展開されている ADC アプライアンスのキーを異なる場所に保管して管理する必要がなくなりました。

HSM でバックアップされたキーが提供された Azure Key Vault Premium の料金範囲で ADC を使用すると、FIPS 140-2 レベル 2 のコンプライアンスが提供されます。

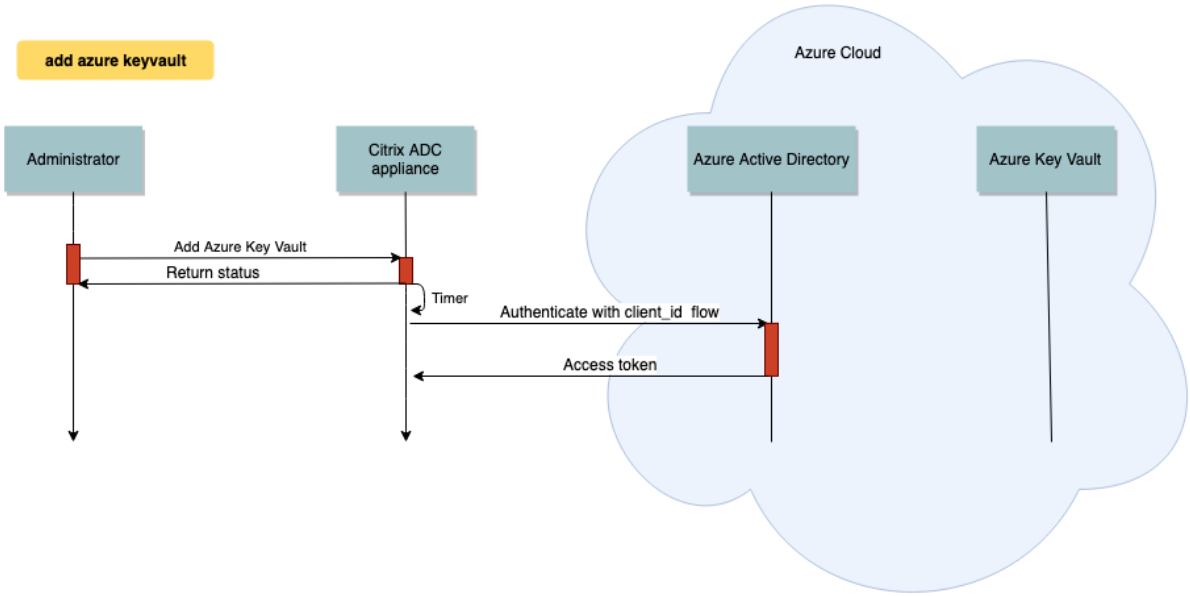
Azure Key Vault は、Microsoft が提供する標準サービスです。Azure キーボールトの詳細については、Microsoft Azure のドキュメントを参照してください。

注: Citrix ADC と Azure Key Vault との統合は、TLS 1.3 プロトコルでサポートされています。

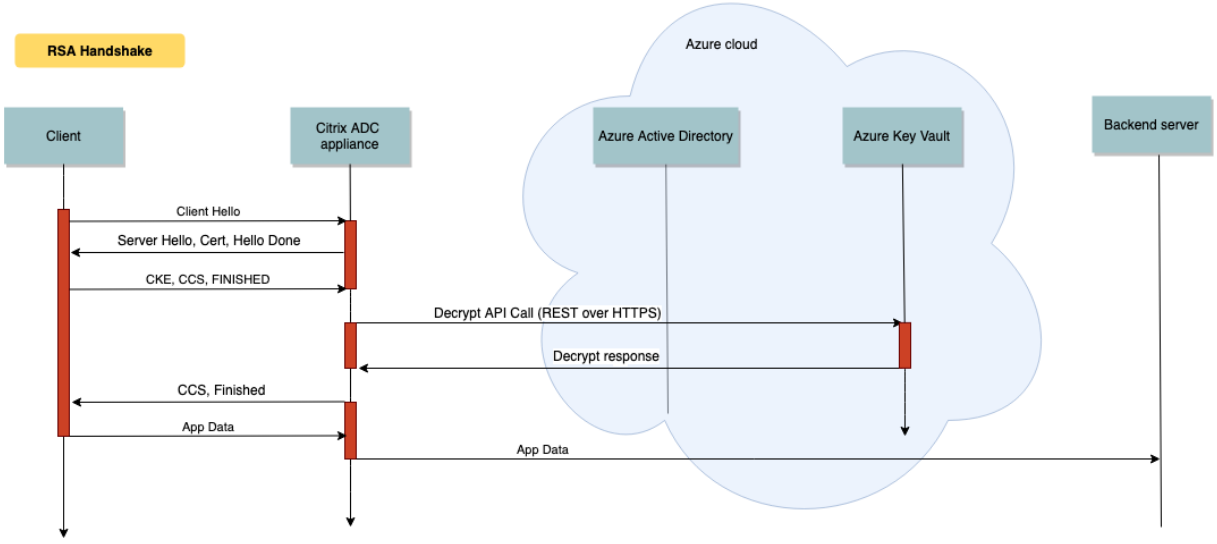
アーキテクチャの概要

Azure Key Vault は、Azure クラウドにシークレットを安全に保存するためのサービスです。キーを Azure Key Vault に保存することで、キーが盗まれる可能性を減らすことができます。Key Vault の設定が完了したら、キーを保管できます。Key Vault で秘密鍵操作を実行するように ADC アプライアンス上の仮想サーバーを構成します。ADC アプライアンスは、SSL ハンドシェイクごとにキーにアクセスします。

次の図は、認証後に Azure Active Directory からアクセストークンを取得するプロセスを示しています。このトークンは、秘密鍵を使用する暗号操作の REST API 呼び出しで使用されます。



次の図は、一般的な RSA ハンドシェイクを示しています。公開キーを使用して暗号化されたクライアントキー交換 (CKE) メッセージは、Key Vault に保存されている秘密キーを使用して復号されます。



ECDHE ハンドシェイクでは、Citrix ADC アプライアンスから送信されたサーバーキー交換（SCE）メッセージは、Key Vault に保存されている秘密キーを使用して署名されます。

前提条件

1. Azure サブスクリプションが必要です。

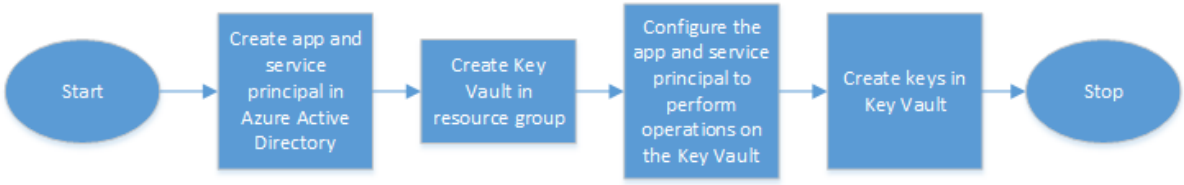
2. (オプション) Linux マシンに Azure CLI をインストールします。手順については、Azure のドキュメントhttps://docs.microsoft.com/en-us/cli/azure/install-azure-cli-apt?view=azure-cli-latestを参照してください。
3. ADC アプライアンスでエンティティを構成する前に、Azure Portal で構成を完了します。

ADC Azure キーボールト統合を構成する

最初に Azure Portal で構成を実行し、続いて ADC アプライアンスで構成を実行します。

Azure ポータルで次の手順を実行します

次のフローチャートは、Azure Portal で必要な構成の概要を示しています。

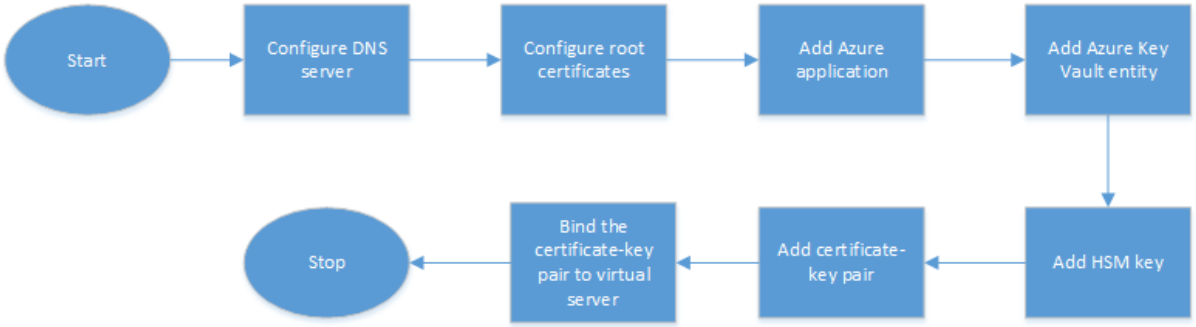


1. Azure Active Directory にアプリとサービスのプリンシパルを作成します
2. リソースグループに Key Vault を作成します。
3. Key Vault で署名および復号化操作を実行するようにアプリとサービスプリンシパルを設定します。
4. 次のいずれかの方法で Key Vault にキーを作成します。
    a) キーファイルをインポートする。
    b) 証明書を生成する。

上記の手順を構成するコマンドについては、https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principalsの Azure のドキュメントを参照してください。

ADC アプライアンスで次の手順を実行します

次のフローチャートは、ADC アプライアンスで必要な設定の概要を示しています。



1. DNS サーバーを構成します。
2. ルート証明書を構成して、Azure によって提示された証明書を検証します。
3. Azure アプリケーションを作成します。
4. Azure キーボールトエンティティを作成します。
5. HSM キーを作成します。
6. 証明書とキーのペアを作成します。
7. 証明書とキーのペアを仮想サーバーにバインドします。

DNS サーバーを構成する

キーボールトホストと Azure Active Directory エンドポイントの名前解決には、DNS サーバーが必要です。

CLI を使用して DNS サーバーを設定するには

コマンドプロンプトで入力します。

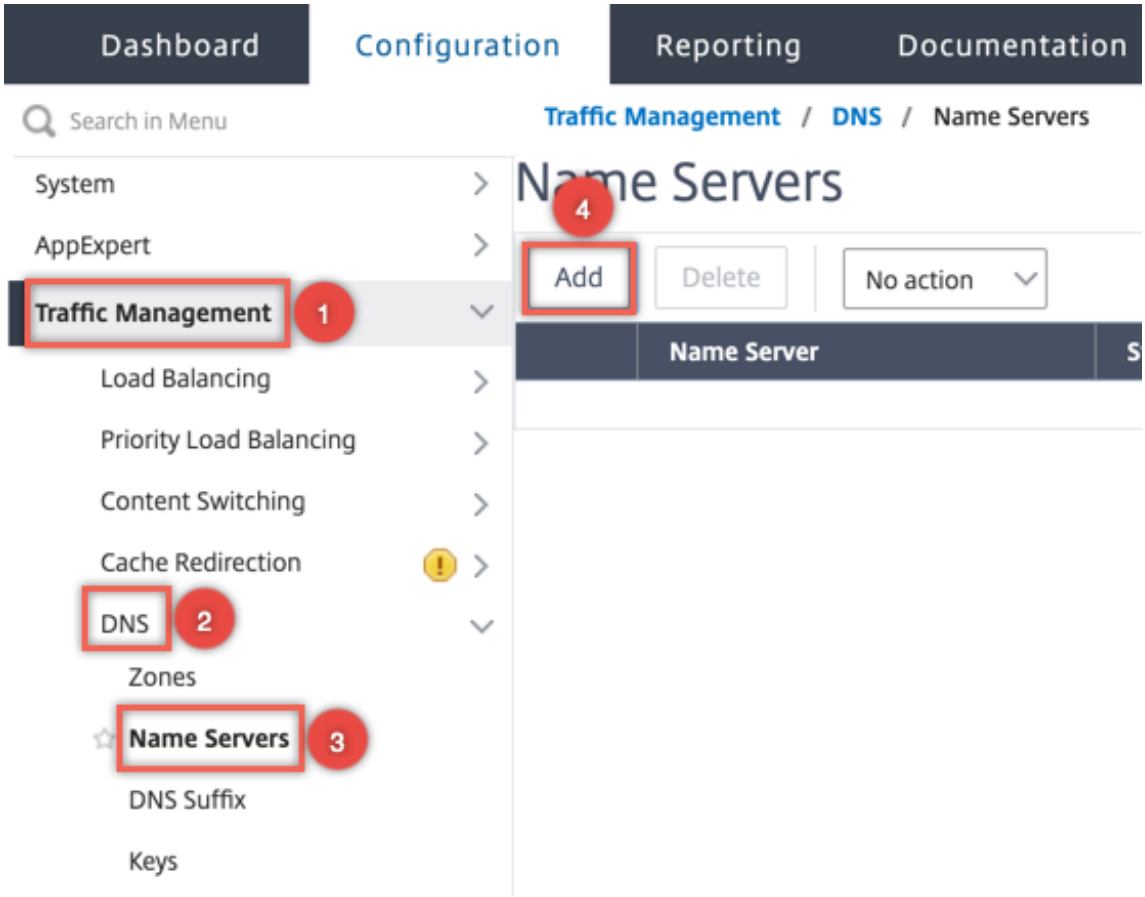
```
1 add dns nameserver <IP address>
2 <!--NeedCopy-->
```

例**:**

```
1 add dns nameserver 192.0.2.150
2 <!--NeedCopy-->
```

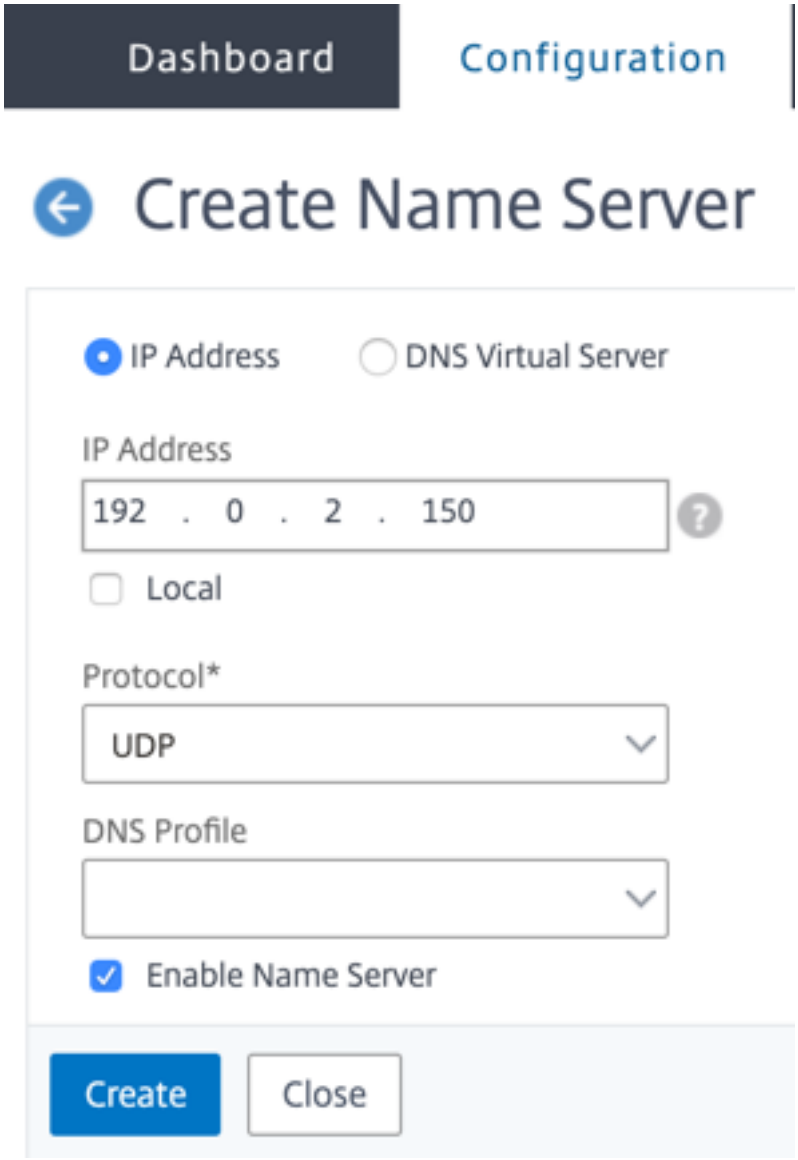
GUI を使用して DNS サーバーを構成するには

1. [トラフィック管理**] > [DNS] > [**ネームサーバー**]** に移動します。[追加] をクリックします。



2. 次のパラメーターの値を入力します。

   - IP アドレス-外部ネームサーバの IP アドレス。Local パラメータが設定されている場合は、ローカル DNS サーバ (LDNS) の IP アドレス。
   - Protocol：ネームサーバーが使用するプロトコル。UDP_TCP は、ネームサーバーがアプライアンスに構成されている DNS 仮想サーバーである場合は無効です。



3. ［作成］をクリックします。

ルート証明書を追加してバインドする

Azure キーボールト`https://<vault_name>.vault.azure.net`と Azure Active Directory (AAD) https://login.microsoftonline.com によって提示された証明書のルート証明書をダウンロードし、ADC アプライアンスにロードします。これらの証明書は、Azure Key Vault と AAD によって提示された証明書を検証するために必要です。1 つ以上の証明書を CA 証明書グループ`ns_callout_certs`にバインドします。

CLI を使用してルート証明書を追加するには

コマンドプロンプトで入力します。


```
add ssl certkey <certkeyname> -cert <certname>
bind ssl caCertGroup <caCertGroupName> <certkeyName>
<!--NeedCopy-->
```

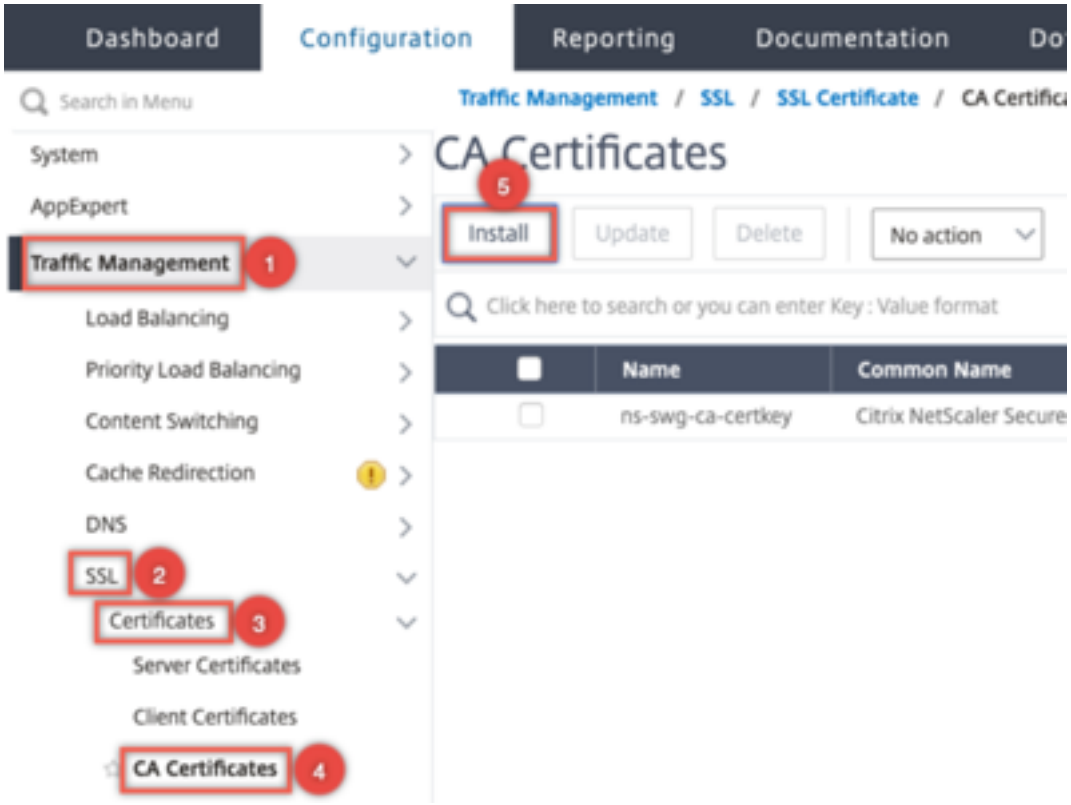
例:

次の例では、Azure Key Vault と AAD によって提示されるルート証明書は同じです。

```
add ssl certKey rootcert -cert RootCyberTrustRoot.crt
bind ssl cacertGroup ns_callout_certs  rootcert
<!--NeedCopy-->
```

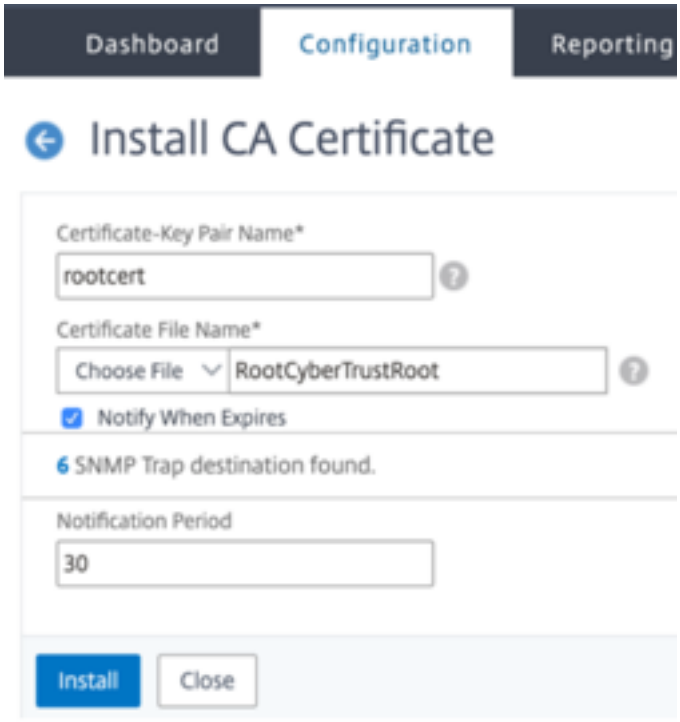
GUI を使用してルート証明書を追加するには

1. トラフィック管理 **> SSL >** 証明書 **> CA** 証明書に移動します。

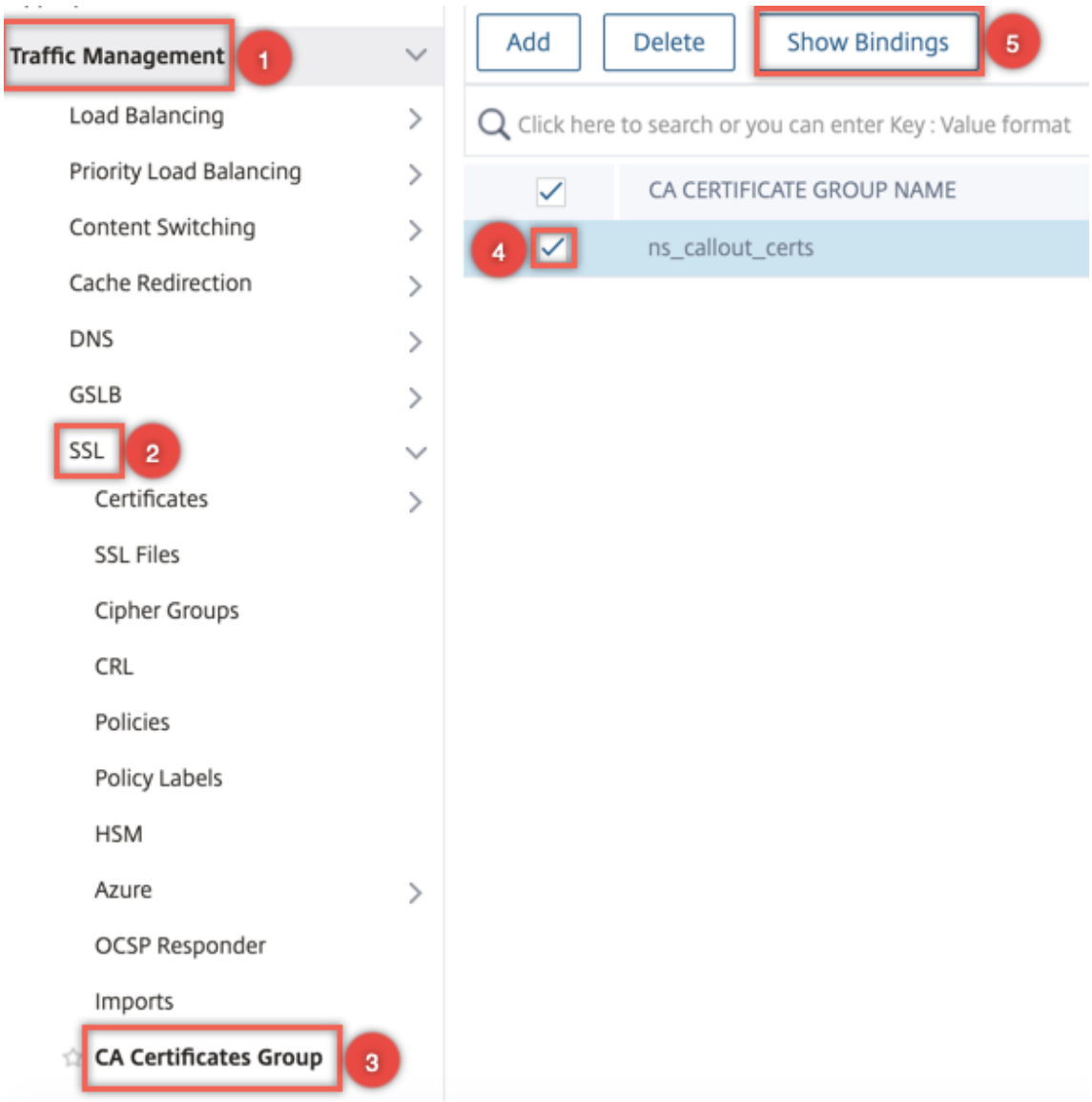


2. 次のパラメーターの値を入力します。

   - 証明書とキーのペア名
   - 証明書ファイル名



3. [インストール] をクリックします。
4. トラフィック管理 **> SSL > CA** 証明書グループに移動します。
5. **ns_callout_certs** を選択し、[バインディングの表示] をクリックします。



6. ［バインド］をクリックします。
7. 前に作成した CA 証明書を選択し、[**Select**] をクリックします。
8. [バインド] をクリックし、[閉じる] をクリックします。

Azure アプリケーションの構成

Azure アプリケーションエンティティには、Azure Active Directory への認証とアクセストークンの取得に必要な資格情報が含まれています。つまり、Key Vault リソースと API への認証アクセスを取得するには、Azure アプリケーション ID、シークレット (パスワード)、テナント ID を ADC アプライアンスに追加します。
CLI を使用して Azure Application エンティティを構成する場合は、パスワードを入力する必要があります。GUI を使用する場合、Azure アプリケーションエンティティには、Azure Active Directory への認証とアクセストークンの取得に必要な資格情報が含まれます。

CLI を使用して Azure アプリケーションを構成するには

リリース 13.0 ～61.x では、アクセストークンがアプリケーションに付与される前にリソースグループのドメインを取得するためのパラメータ VaultResource がadd azure applicationコマンドに追加されました。ドメイン名は地域によって異なる可能性があるため、このパラメータが追加されます。たとえば、ドメインはvault.azure.netまたはvault.usgov.netです。

コマンドプロンプトで入力します。

```
add azure application <name> -clientID <string> -clientSecret -tenantID
    <string> -vaultResource <string> [-tokenEndpoint <URL>]
show azure application
<!--NeedCopy-->
```

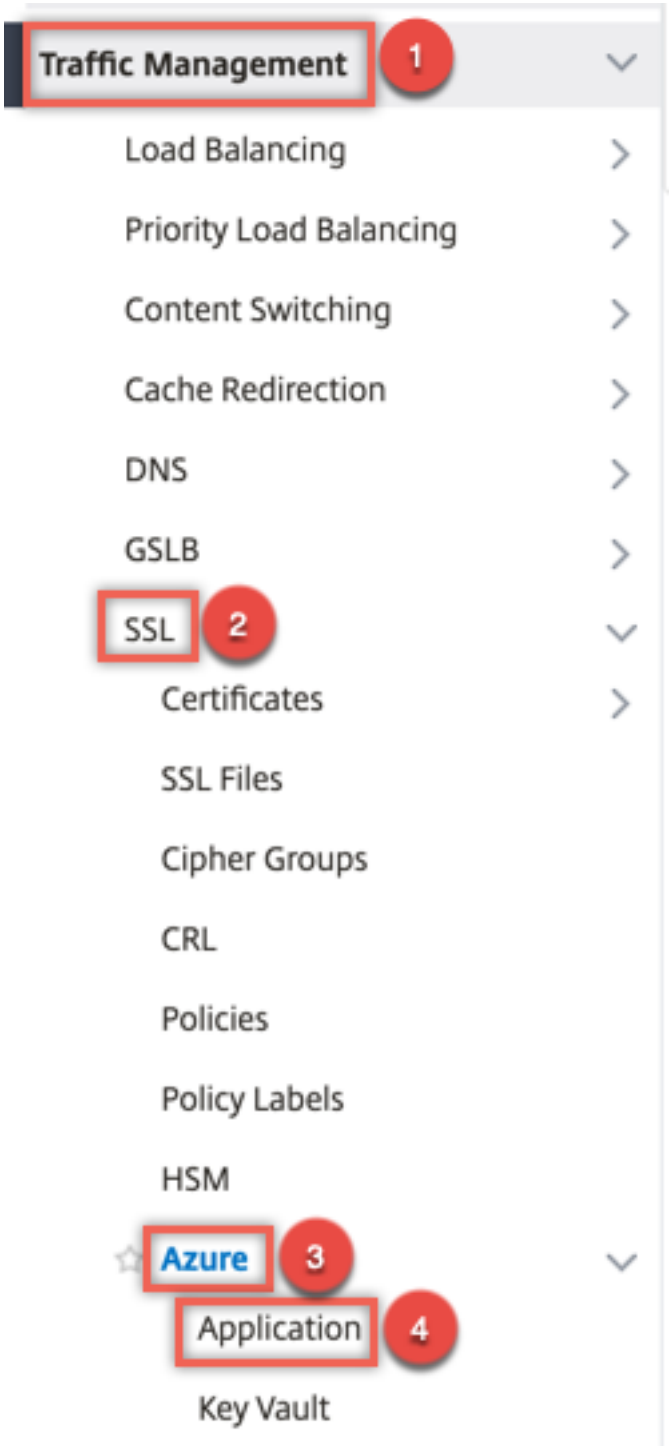
例:

```
add azure application app10 -clientiD 12345t23aaa5 -clientsecret
   csHzOoEzmuY= -vaultResource example.vault.azure.net -tenantID 33583
   ee9ca5b
 Done
> sh azure application app10
1)  Name:  app10            ClientID:  12345t23aaa5
    TokenEndpoint:  ”https://login.microsoftonline.com/33583ee9ca5b/”
    TenantID:  33583ee9ca5b  VaultResource:  example.vault.azure.net
 Done

<!--NeedCopy-->
```

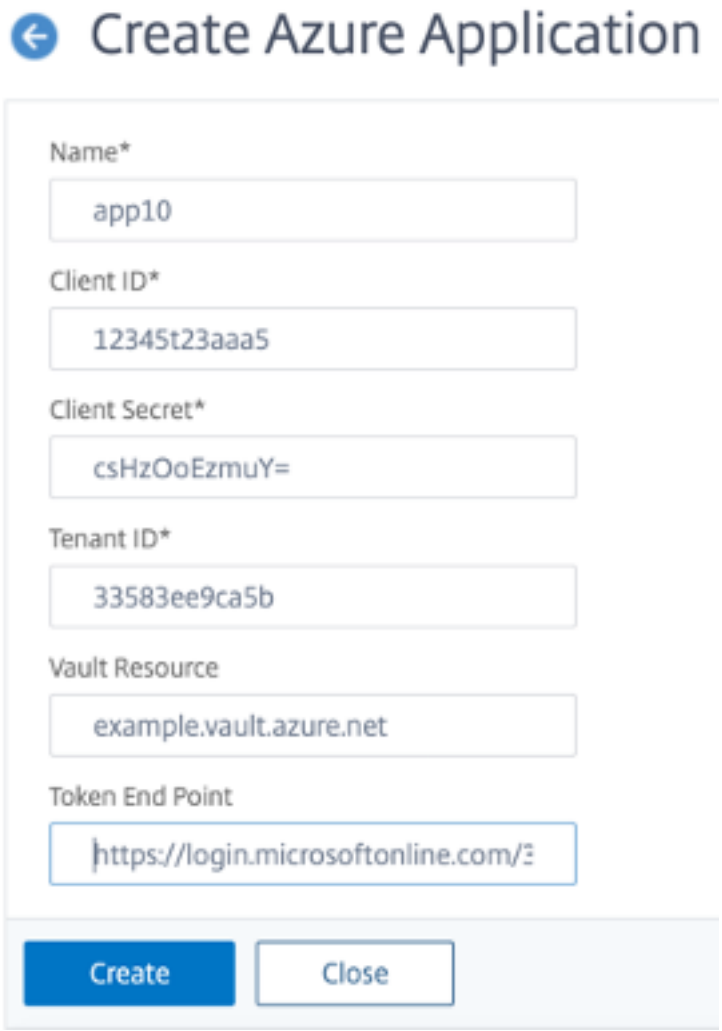
GUI を使用して Azure アプリケーションを構成するには

1. トラフィック管理 **> SSL > Azure >** アプリケーションに移動します。



2. 詳細ペインで、[**Add**] をクリックします。
3. 次のパラメーターの値を入力します。
   - 名前-Citrix ADC アプライアンス上のアプリケーションオブジェクトの名前。
   - クライアント ID — アプリケーションが Azure CLI または Azure ポータル (GUI) を使用して Azure Active Directory に作成されたときに生成されるアプリケーション ID。

- クライアントシークレット — Azure Active Directory で構成されたアプリケーションのパスワード。パスワードは Azure CLI で指定されるか、Azure ポータル (GUI) で生成されます。
- テナント ID — アプリケーションが作成された Azure Active Directory 内のディレクトリの ID。
- Vault リソース-アクセストークンが付与される Vault リソース。例 `vault.azure.net`。
- トークンエンドポイント — アクセストークンを取得できる URL。トークンエンドポイントが指定されていない場合、デフォルト値は`https://login.microsoftonline.com/<tenant id>`です。



Azure キーボールトを構成する

ADC アプライアンスに Azure Key Vault オブジェクトを作成します。

CLI を使用して Azure キーボールトを構成するには

コマンドプロンプトで入力します。


```
add azure keyVault <name> -azureVaultName <string> -azureApplication
       <string>
show azure keyvault
<!--NeedCopy-->
```

例:

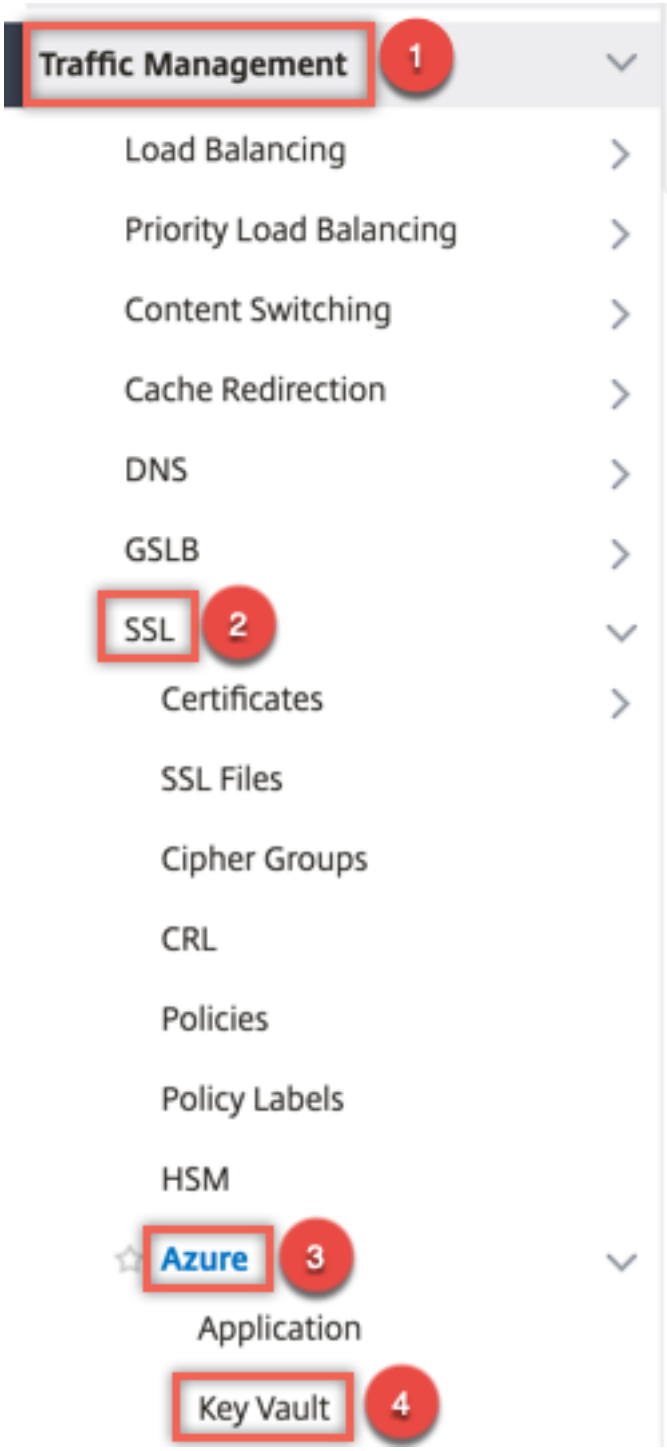
```
add azure keyvault kv1 -azureapplication app10 -azurevaultName pctest.
   vault.azure.net
> sh azure keyVault
1)  Name:  kv1              AzureVaultName:  pctest.vault.azure.net
  AzureApplication:  app10  State:  "Access token obtained"
 Done
<!--NeedCopy-->
```

次の表に、Azure Key Vault の状態で使用できるさまざまな値と、各状態に関する簡単な説明を示します。

State	説明
Created	Key Vault オブジェクトの初期状態。認証は試行されていません。
Could not reach token end point	DNS サーバーが構成されていない、発行者証明書が CA 証明書グループにバインドされていない、またはネットワークの問題のいずれかを示します。
Authorization failed	アプリケーション認証情報が正しくありません。
Token parse error	Azure Active Directory からの応答が期待される形式ではありません。
Access token obtained	Azure Active Directory によって正常に認証されました。

GUI を使用して Azure キーボールトを構成するには

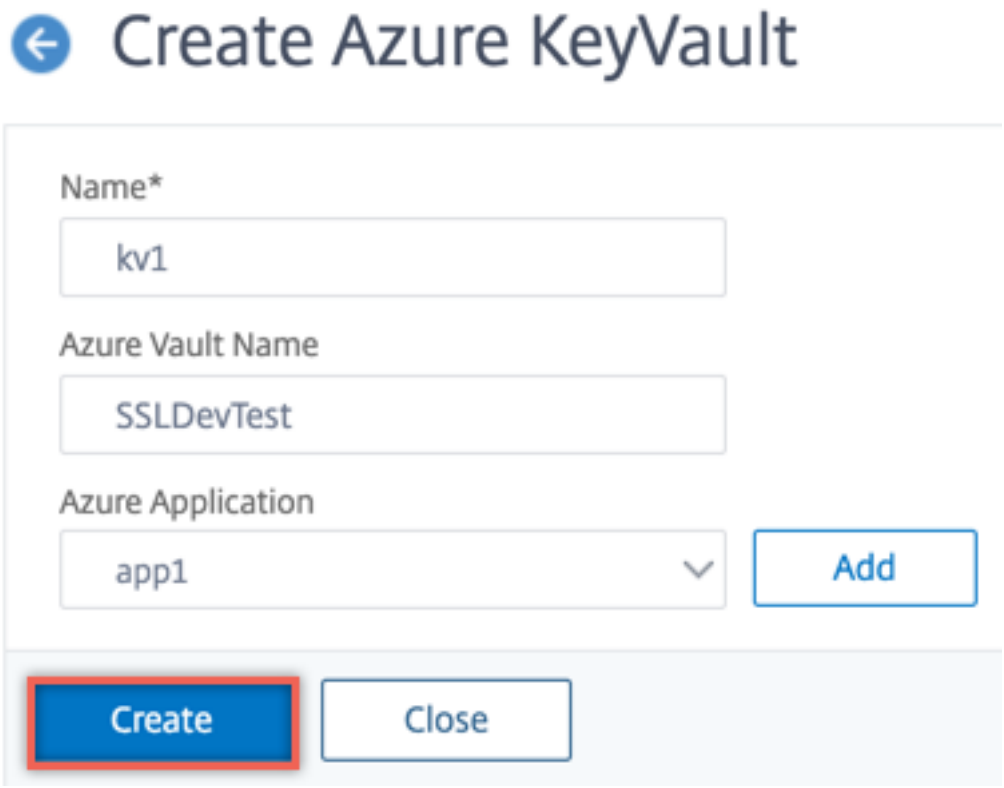
1. トラフィック管理 **> SSL > Azure >** キーボールトに移動します。



2. 次のパラメーターの値を入力します。

   - Name：キーボールトの名前。
   - Azure キーボールト名-Azure CLI または Azure ポータル (GUI) を使用してドメイン名を使用して Azure クラウドで構成されたキーボールトの名前。
   - Azure アプリケーション名-ADC アプライアンスで作成された Azure アプリケーションオブジェクトの

名前。この名前の Azure アプリケーションオブジェクトは、Azure Active Directory での認証に使用されます。



HSM キーの追加

プライベートキーを HSM に保存すると、FIPS 140-2 レベル 2 に準拠できます。

CLI を使用して HSM キーを追加するには

コマンドプロンプトで入力します。

```
add ssl hsmKey <hsmKeyName> [-hsmType <hsmType>] [-key <string> |
       -serialNum <string>] {
 -password  }
  [-keystore <string>]
<!--NeedCopy-->
```

例**:**

```
add ssl hsmKey h1 -keystore kv1 -key san15key -hsmType KEYVAULT
```



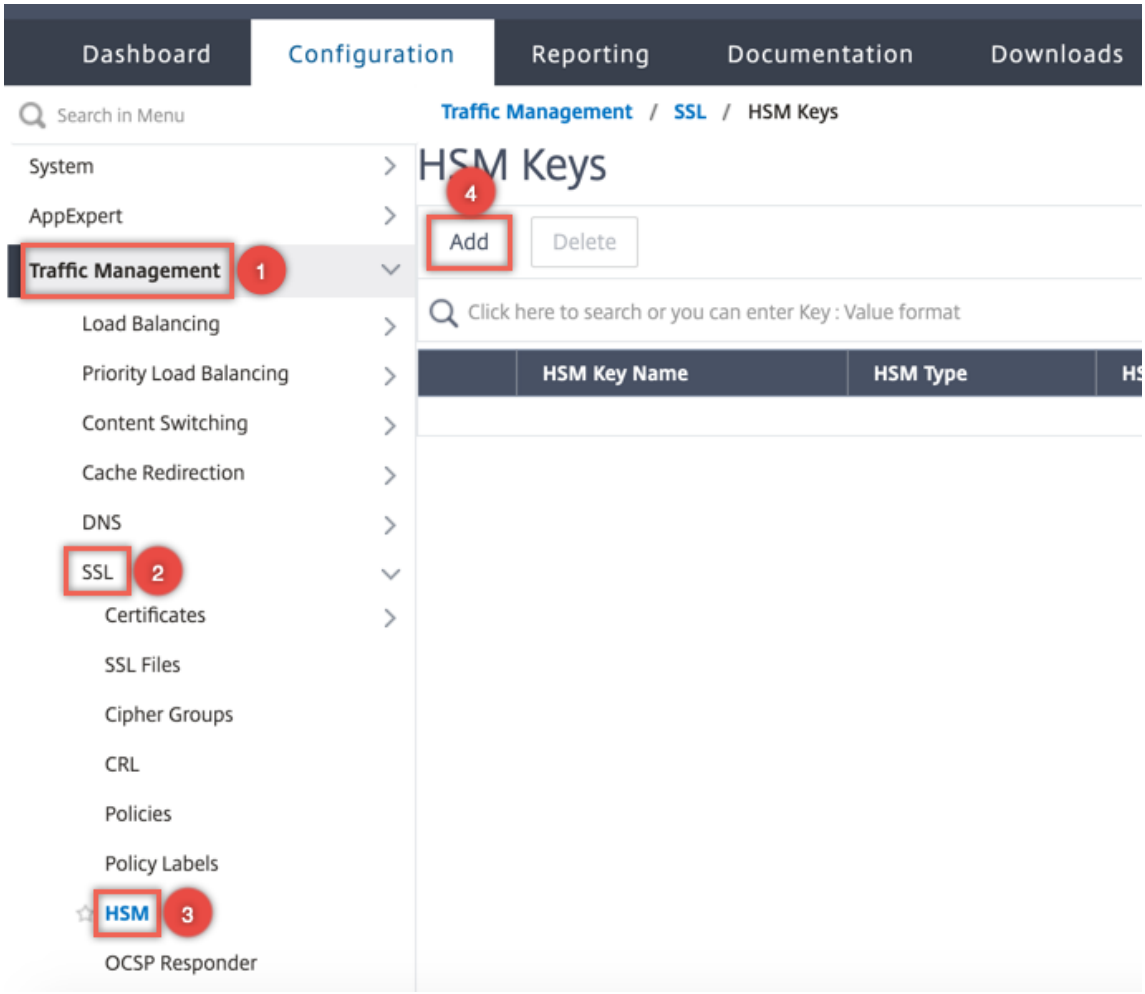
```
2
3
4  > sh ssl hsmKey h1
5      HSM Key Name: h1       Type: KEYVAULT
6      Key: san15key
7      Key store: kv1
8      State:  “Created”
9   Done
10 <!--NeedCopy-->
```

次の表に、HSM キーの状態で使用できるさまざまな値と、各状態に関する簡単な説明を示します。

State	説明
作成日時	HSM キーが ADC アプライアンスに追加されます。キー操作はまだ試行されていません。
アクセストークンが使用	キー操作が試行されたときには、アクセストークンを使用できません。
無許可	構成された Azure アプリケーションには、キー操作を実行する権限がありません。
存在しない	キーは Azure Key ボールトに存在しません。
到達不能	ネットワーク上の Key Vault ホストにアクセスできません。
マークダウン	キーの操作中にしきい値エラーが発生したため、ADC アプライアンスで HSM キーが DOWN とマークされます。
主要なオペレーションが成功しました	キー操作について Key Vault から成功レスポンスが届きました。
キー操作に失敗しました	キー操作のために Key Vault からエラー応答を受信しました。
キー操作が調整された	キー操作のリクエストは、Key Vault によって調整されます。

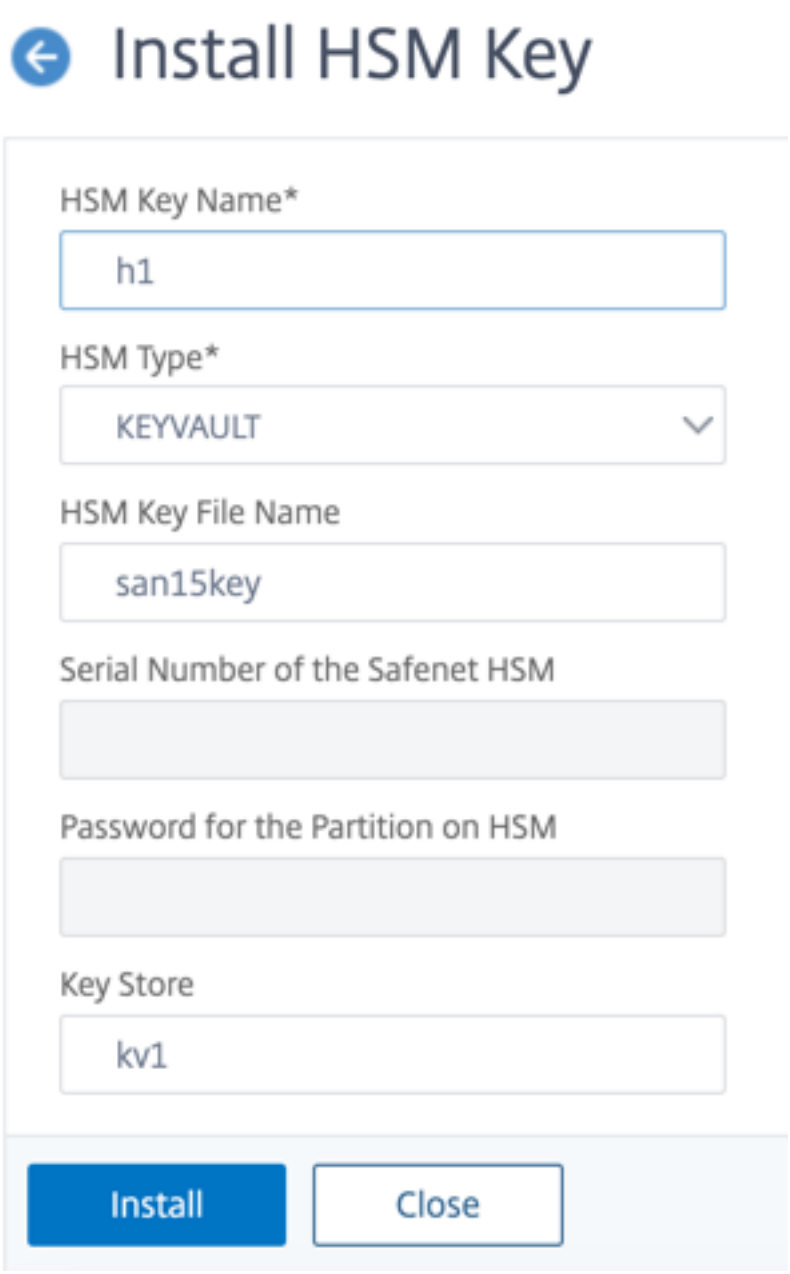
GUI を使用して HSM キーを追加するには

1. トラフィック管理**] > [SSL] > [HSM]** に移動します。



2. 次のパラメータの値を入力します。

   - HSM キー名-キーの名前。
   - HSM タイプ-HSM のタイプ。
   - Key store-キーが格納されている HSM を表すキーストアオブジェクトの名前。たとえば、キー Vault オブジェクトや Azure Key Vault 認証オブジェクトの名前などです。KEYVAULTタイプ HSM にのみ適用されます。



3. ［追加］をクリックします。

証明書とキーのペアを追加します

前に作成した HSM キーを使用して、証明書とキーのペアを追加します。

CLI を使用して証明書とキーのペアを追加するには

コマンドプロンプトで入力します。

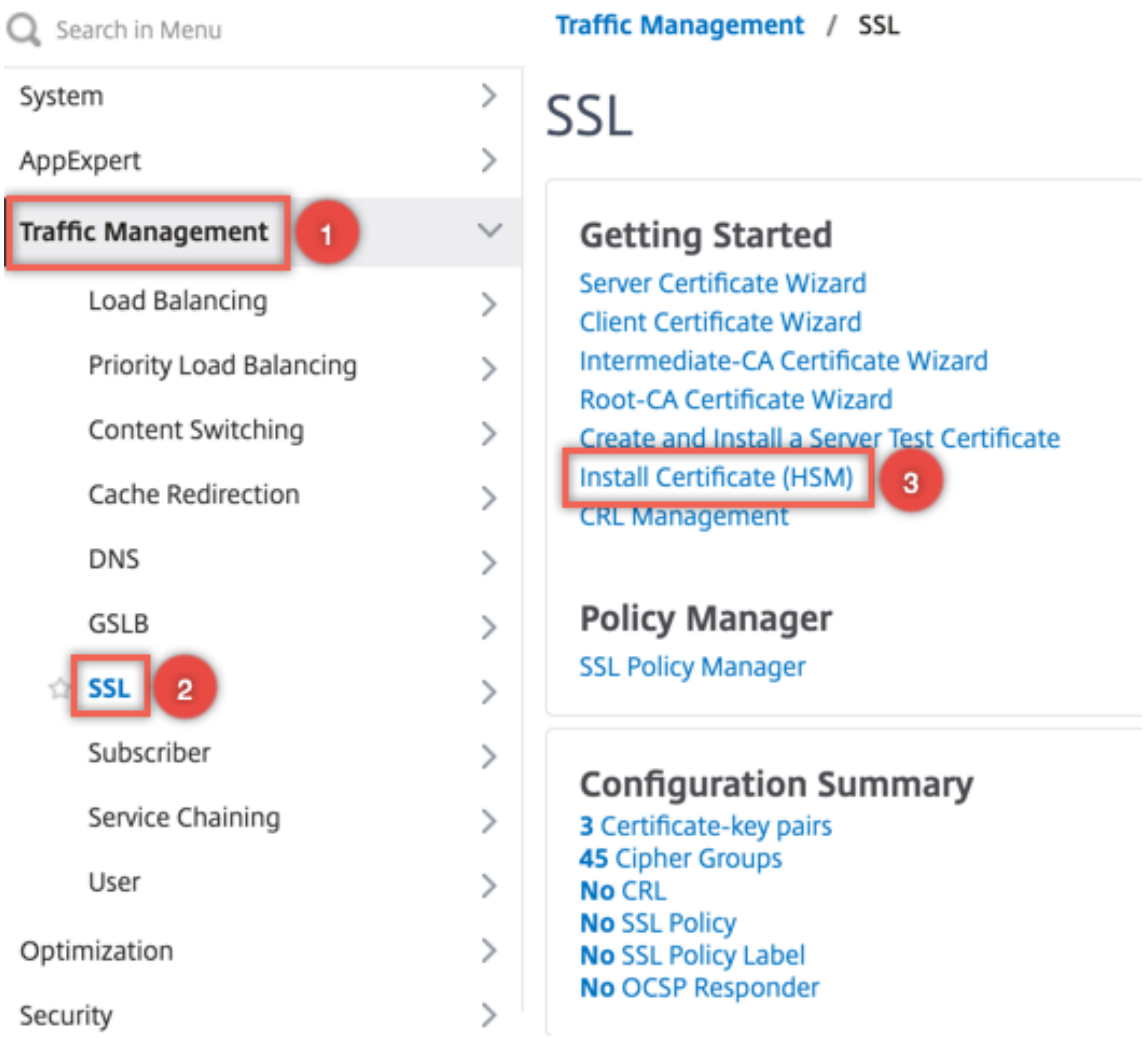

```
add ssl certKey <certkeyName> (-cert <string> [-password]) -hsmKey <
    string>]
show ssl certkey
<!--NeedCopy-->
```

例:

```
add ssl certKey serverrsa_2048 -cert /nsconfig/ssl/san_certs/san15.pem
    -hsmKey h1
> sh ssl certkey serverrsa_2048
    Name: serverrsa_2048        Status: Valid,   Days to expiration
       :9483
    Version: 3
    Serial Number: F5CFF9EF1E246022
    Signature Algorithm: sha256WithRSAEncryption
    Issuer:  C=in,O=citrix,CN=ca
    Validity
        Not Before: Mar 20 05:42:57 2015 GMT
        Not After : Mar 12 05:42:57 2045 GMT
    Certificate Type:   ”Server Certificate”
    Subject:  C=in,O=citrix
    Public Key Algorithm: rsaEncryption
    Public Key size: 2048
    Ocsp Response Status: NONE
 Done
<!--NeedCopy-->
```

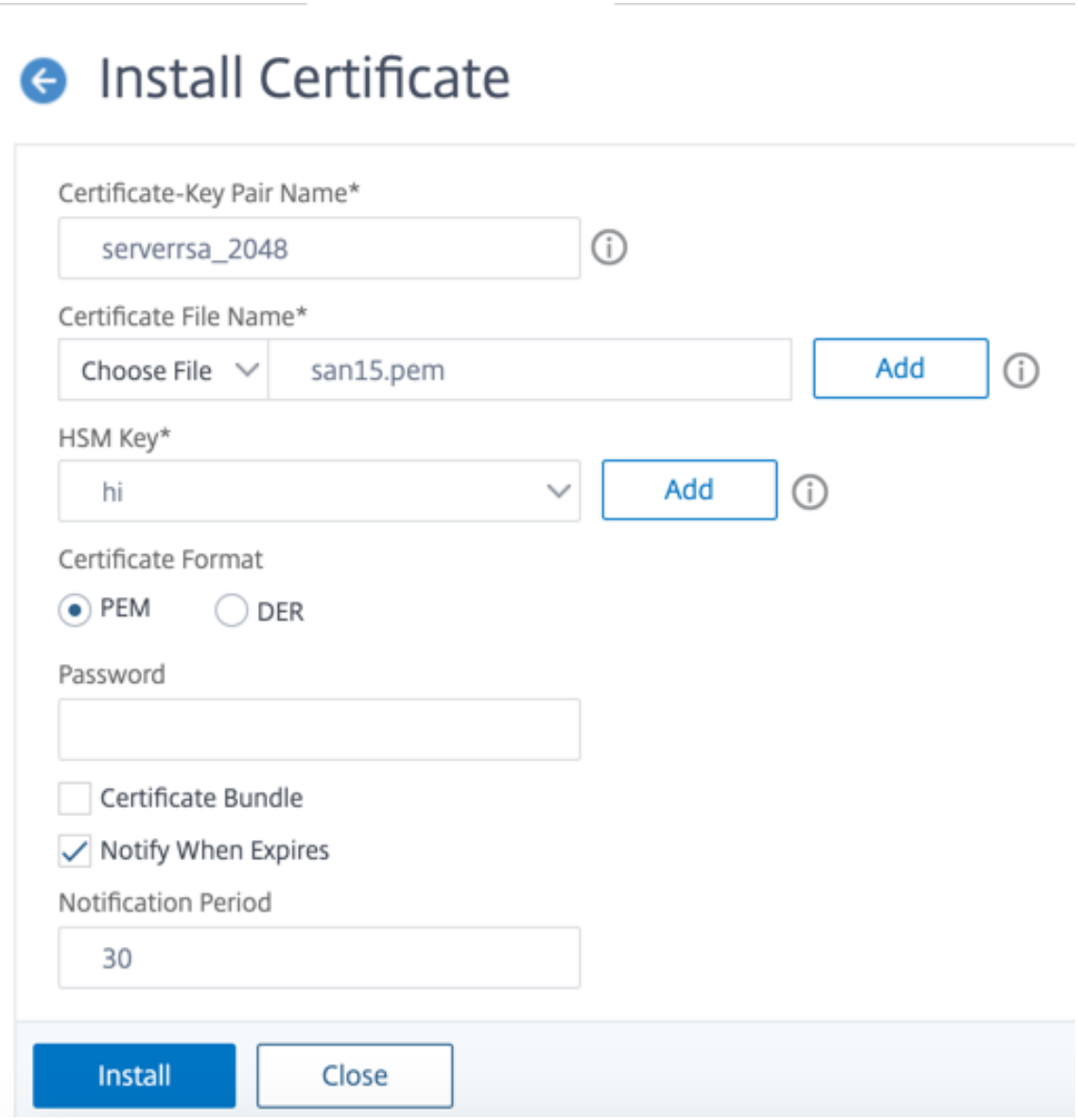
GUI を使用して証明書とキーのペアを追加するには

1. [トラフィック管理] **> [SSL] > [**証明書のインストール **(HSM)**] に移動します。



2. 次のパラメーターの値を入力します。

   - 証明書とキーのペア名
   - 証明書ファイル名
   - HSM キー



3. ［インストール］をクリックします。

証明書とキーのペアを仮想サーバーにバインドする

SSL トランザクションの処理に使用される証明書は、SSL データを受信する仮想サーバーにバインドする必要があります。

CLI を使用して SSL 証明書とキーのペアを仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vserver <vServerName>
```



```
<!--NeedCopy-->
```

例:

```
bind ssl vserver v1 -certkeyName serverrsa_2048

sh ssl vserver v1

    Advanced SSL configuration for VServer v1:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED      Refresh Count: 0
    Session Reuse: ENABLED      Timeout: 120 seconds
    Cipher Redirect: DISABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    OCSP Stapling: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    HSTS Preload: NO
    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1

    ECC Curve: P_256, P_384, P_224, P_521

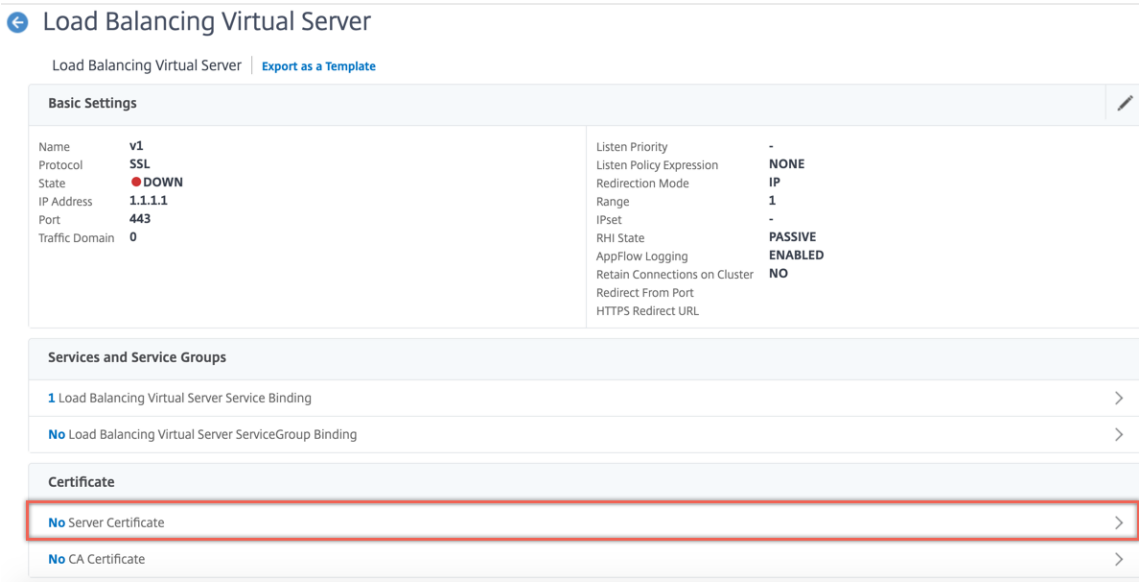
1)  CertKey Name: serverrsa_2048    Server Certificate



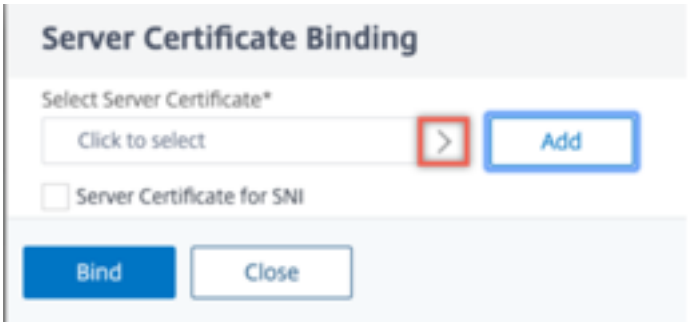
1)  Cipher Name: DEFAULT
    Description: Default cipher list with encryption strength >= 128bit
 Done
<!--NeedCopy-->
```


GUI を使用して SSL 証明書とキーのペアを仮想サーバーにバインドするには

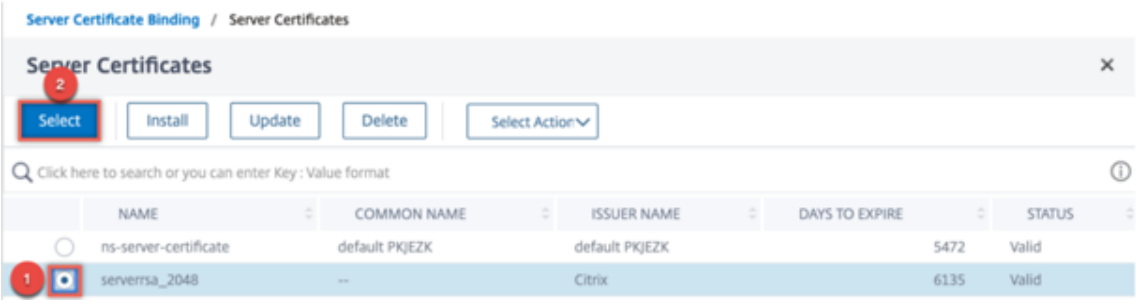
1. [トラフィック管理] **>** [負荷分散] **>** [仮想サーバー] に移動し、SSL 仮想サーバーを開きます。[証明書] セクション内をクリックします。



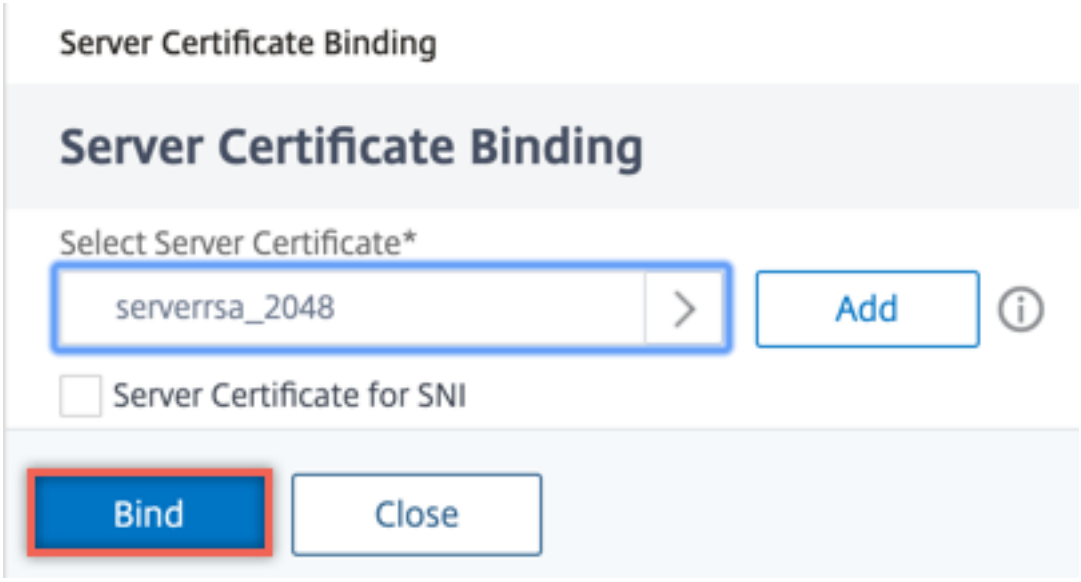
2. 矢印をクリックして、証明書とキーのペアを選択します。



3. リストから証明書とキーのペアを選択します。



4. 証明書とキーのペアを仮想サーバーにバインドします。



制限事項

- キー操作のための Azure Key Vault への同時呼び出しの数には制限があります。ADC アプライアンスのパフォーマンスは、Key Vault の制限によって異なります。詳細については、Microsoft Azure Key Vault のドキュメントを参照してください。
- EC キーはサポートされていません。
- EDT プロトコルと DTLS プロトコルはサポートされていません。
- インテル Coleto SSL チップを搭載した ADC アプライアンスはサポートされていません。
- クラスタリングおよび管理パーティションはサポートされていません。
- Azure アプリケーションエンティティ、Azure Key Vault オブジェクト、および HSM 証明書とキーのペアを ADC アプライアンスに追加した後は、更新できません。
- HSM キーを含む証明書バンドルはサポートされていません。
- HSM キーと証明書が一致しない場合、エラーは表示されません。証明書とキーのペアを追加する際は、HSM キーと証明書が一致していることを確認します。
- HSM キーを DTLS 仮想サーバーにバインドすることはできません。
- HSM キーを使用して作成された証明書とキーのペアを使用して OCSP リクエストに署名することはできません。
- HSM キーを使用して証明書とキーのペアを作成した場合、証明書とキーのペアを SSL サービスにバインドすることはできません。

よくある質問

Azure Key Vault と統合した場合、秘密キーは **ADC** アプライアンスのメモリに保存されますか

いいえ。秘密鍵は ADC アプライアンスのメモリには保存されません。SSL トランザクションごとに、アプライアンスは Key Vault にリクエストを送信します。

統合 **FIPS 140-2** レベル **2** は準拠していますか

はい。統合ソリューションは FIPS 140-2 レベル 2 のサポートを提供します。

どのキータイプがサポートされていますか

RSA キータイプのみがサポートされています。

どのキーサイズがサポートされていますか

1024 ビット、2048 ビット、および 4096 ビットの RSA キーがサポートされています。

どの暗号がサポートされていますか

ECDHE と SHA256 を使用した TLSv1.3 暗号を含む、ADC アプライアンスでサポートされるすべての暗号がサポートされます。

トランザクションはログに記録されますか

ADC アプライアンスは、Key Vault との各トランザクションをログに記録します。時間、Vault IP アドレス、ポート、接続の成功または失敗、エラーなどの詳細がログに記録されます。
次に、SSL ログ出力の例を示します。

```
1 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 896 0 :
    Backend SPCBId 30894 - ServerIP 104.211.224.186 - ServerPort 443
     - ProtocolVersion TLSv1.2 - CipherSuite "ECDHE-RSA-AES256-GCM-
    SHA384 TLSv1.2 Non-Export 256-bit" - Session New -
    SERVER_AUTHENTICATED -SerialNumber "200005
    A75B04365827852D630000000005A75B" - SignatureAlgorithm "
    sha256WithRSAEncryption" - ValidFrom "Mar 17 03:28:42 2019 GMT"
    - ValidTo "Mar 17 03:28:42 2021 GMT" - HandshakeTime 40 ms
2 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 897 0 :
    SPCBId 30894 - IssuerName " C=US,ST=Washington,L=Redmond,O=
    Microsoft Corporation,OU=Microsoft IT,CN=Microsoft IT TLS CA 2"
3 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 898 0 :
    SPCBId 30894 - SubjectName " CN=vault.azure.net"
4 <!--NeedCopy-->
```


トラブルシューティング

October 7, 2021

構成後に SSL 機能が期待どおりに動作しない場合は、一般的なツールを使用して Citrix ADC リソースにアクセスし、問題を診断できます。

トラブルシューティングのリソース

最良の結果を得るには、次のリソースを使用して、Citrix ADC アプライアンスでの SSL 問題のトラブルシューティングを行います。

- 関連する ns.log ファイル
- 最新の ns.conf ファイル
- メッセージファイル
- 関連`newnslog` ファイル
- トレースファイル
- 可能であれば、証明書ファイルのコピー
- 可能であれば、キーファイルのコピー
- エラーメッセージ（存在する場合）

これらのリソースに加えて、Citrix ADC トレースファイル用にカスタマイズした Wireshark アプリケーションを使用して、トラブルシューティングを迅速に行うことができます。

SSL の問題のトラブルシューティング

SSL の問題のトラブルシューティングを行うには、次の手順を実行します。

- Citrix ADC アプライアンスに SSL オフロードと負荷分散のライセンスが付与されていることを確認します。
- アプライアンスで SSL オフロードおよび負荷分散機能が有効になっていることを確認します。
- SSL 仮想サーバーのステータスが DOWN として表示されないことを確認します。
- 仮想サーバにバインドされているサービスのステータスが [DOWN] と表示されないことを確認します。
- 有効な証明書が仮想サーバーにバインドされていることを確認します。
- サービスが適切なポート（ポート 443 が望ましい）を使用していることを確認します。

パケットトレースからの **TLS1.3** トラフィックの復号化

TLS1.3 で実行されるプロトコルのトラブルシューティングを行うには、まず TLS1.3 トラフィックを復号化する必要があります。Wireshark で TLS 1.3 を復号化するには、秘密をNSS キーログ形式でエクスポートする必要があります。キーログ形式の詳細については、「NSS キーログ形式」を参照してください。

パケットトレースのキャプチャ方法については、トレース中の SSL セッションキーのキャプチャを参照してください。

注：Citrix ADC は、各接続の秘密を、使用中の TLS/SSL プロトコルのバージョンに適した形式で自動的に記録します。

HA セットアップでセカンダリノードで **CRL** の更新が行われない

CRL サーバはプライベートネットワークを介してプライマリノードにのみアクセスできるため、更新は行われません。

回避策： CRL サーバの IP アドレスを使用して、プライマリノードでサービスを追加します。このサービスは、CRL サーバーのプロキシとして機能します。ノード間で構成が同期されると、CRL の更新は、プライマリノードで構成されたサービスを通じて、プライマリノードとセカンダリノードの両方に対して機能します。

SSL に関するよくある質問

October 7, 2021

基本的な質問

VPX インスタンスで **GUI** への **HTTPS** アクセスが失敗する。アクセスするにはどうしたらいいですか

GUI への HTTPS アクセスには、証明書とキーのペアが必要です。Citrix ADC アプライアンスでは、証明書とキーのペアが自動的に内部サービスにバインドされます。デフォルトのキーサイズは、MPX または SDX アプライアンスで 1024 バイト、VPX インスタンスで 512 バイトです。ただし、最新の Web ブラウザーの多くは 1024 バイト未満のキーを受け入れません。このため、VPX 構成ユーティリティへの HTTPS アクセスがブロックされてしまいます。

構成ユーティリティへの HTTPS アクセスのために、少なくとも 1024 バイトの証明書とキーのペアをインストールし、内部サービスにバインドすることをお勧めします。または、`ns-server-certificate` を 1024 バイトに更新します。設定ユーティリティまたは CLI への HTTP アクセスを使用して、証明書をインストールできます。

MPX アプライアンスにライセンスを追加すると、証明書とキーのペアバインディングが失われます。この問題を解決するにはどうしたらいいですか

MPX アプライアンスの起動時にライセンスが存在せず、後でライセンスを追加してアプライアンスを再起動すると、証明書のバインドが失われる可能性があります。証明書を再インストールし、内部サービスにバインドします。

アプライアンスを起動する前に、適切なライセンスをインストールすることをお勧めします。

SSL トランザクションのセキュリティで保護されたチャネルの設定に関連するさまざまな手順は何ですか

SSL トランザクション用のセキュアチャネルを設定するには、次の手順を実行します。

1. クライアントは、セキュアチャネルに対する HTTPS 要求をサーバーに送信します。
2. プロトコルと暗号を選択すると、サーバーは証明書をクライアントに送信します。
3. クライアントはサーバー証明書の信頼性をチェックします。
4. いずれかのチェックが失敗した場合、クライアントは対応するフィードバックを表示します。
5. チェックが合格した場合、またはチェックが失敗してもクライアントが継続することを決定した場合、クライアントは一時的な使い捨てキーを作成します。このキーは事前マスターシークレットと呼ばれ、クライアントはサーバー証明書の公開キーを使用してこのキーを暗号化します。
6. サーバーは、プリマスターシークレットを受信すると、サーバーの秘密キーを使用してそれを復号化し、セッションキーを生成します。クライアントは、プリマスターシークレットからセッションキーも生成します。したがって、クライアントとサーバーの両方に、アプリケーションデータの暗号化と復号化に使用される共通のセッションキーがあります。

SSL は **CPU** 集約型のプロセスであることを理解しています。**SSL** プロセスに関連付けられた **CPU** コストはいくらですか

SSL プロセスには、次の 2 つのステージが関連付けられています。

- 公開鍵と秘密鍵技術を使用した最初のハンドシェイクとセキュアチャネル設定。
- 対称キー技術を使用した一括データ暗号化。

前述のステージはどちらもサーバーのパフォーマンスに影響を与える可能性があり、次の理由から集中的な CPU 処理が必要になります。

1. 最初のハンドシェイクには、公開秘密キーの暗号化が含まれます。これは、キーサイズ（1024 ビット、2048 ビット、4096 ビット）が大きいため、CPU が非常に集中します。
2. データの暗号化/復号化は、暗号化または復号化する必要があるデータの量に応じて、計算コストも高くなります。

SSL 設定のさまざまなエンティティは何ですか

SSL 設定には、次のエンティティがあります。

- サーバー証明書
- 認証局 (CA) 証明書
- 次のタスクのプロトコルを指定する暗号スイート。
  - 初期キー交換
  - サーバーとクライアントの認証
  - 一括暗号化アルゴリズム
  - メッセージ認証
- クライアント認証

- CRL
- SSL 証明書キー生成ツールを使用すると、次のファイルを作成できます。
  - 証明書リクエスト
  - 自己署名証明書
  - RSA キー
  - DH パラメータ

Citrix ADC アプライアンスの **SSL** オフロード機能を使いたい。**SSL** 証明書を受信するためのさまざまなオプションは何ですか

Citrix ADC アプライアンスで SSL セットアップを構成する前に、SSL 証明書を受け取る必要があります。SSL 証明書を受信するには、次のいずれかの方法を使用できます。

- 承認された認証局 (CA) に証明書を要求します。
- 既存のサーバー証明書を使用します。
- Citrix ADC アプライアンスで証明書とキーのペアを作成します。

注：この証明書は、Citrix ADC アプライアンスによって生成されたテストルート CA によって署名されたテスト証明書です。テスト root-CA によって署名されたテスト証明書は、ブラウザでは受け入れられません。ブラウザは、サーバーの証明書を認証できないことを示す警告メッセージがスローされます。

- テスト目的以外には、サーバー証明書に署名するための有効な CA 証明書と CA キーを指定する必要があります。

SSL セットアップの最小要件は何ですか

SSL セットアップを構成するための最小要件は次のとおりです。

- 証明書とキーを取得します。
- 負荷分散 SSL 仮想サーバーを作成します。
- HTTP または SSL サービスを SSL 仮想サーバーにバインドします。
- 証明書とキーのペアを SSL 仮想サーバーにバインドします。

SSL のさまざまなコンポーネントの制限は何ですか

SSL コンポーネントには、次の制限があります。

- SSL 証明書のビットサイズ:4096。
- SSL 証明書の数：アプライアンスの使用可能なメモリによって異なります。
- リンクされた中間 CA SSL 証明書の最大数: チェーンあたり 9
- CRL 失効：アプライアンスの使用可能なメモリによって異なります。

Citrix ADC アプライアンスのエンドツーエンドのデータ暗号化に関連するさまざまな手順は何ですか

Citrix ADC アプライアンスのサーバー側の暗号化プロセスに関連する手順は次のとおりです。

1. クライアントは、セキュリティで保護されたサイトの Citrix ADC アプライアンスで構成された SSL VIP に接続します。
2. セキュアな要求を受信すると、アプライアンスは要求を復号化し、レイヤ 4 ～7 のコンテンツスイッチング技術とロードバランシングポリシーを適用します。次に、リクエストに使用可能な最適なバックエンド Web サーバーを選択します。
3. Citrix ADC アプライアンスは、選択したサーバーとの SSL セッションを作成します。
4. SSL セッションを確立した後、アプライアンスはクライアント要求を暗号化し、セキュア SSL セッションを使用して Web サーバに送信します。
5. アプライアンスは、サーバから暗号化された応答を受信すると、データを復号化して再暗号化します。次に、クライアント側の SSL セッションを使用してデータをクライアントに送信します。

Citrix ADC アプライアンスの多重化技術により、アプライアンスは Web サーバーで確立された SSL セッションを再利用できます。したがって、アプライアンスは、フルハンドシェイクと呼ばれる CPU を大量に消費するキー交換を回避します。このプロセスにより、サーバー上の SSL セッションの総数が削減され、エンドツーエンドのセキュリティが維持されます。

証明書とキー

証明書とキーファイルは任意の場所に配置できますか。これらのファイルを保存する推奨場所はありますか

証明書とキーファイルは、Citrix ADC アプライアンスまたはローカルコンピュータに格納できます。ただし、証明書とキーファイルは Citrix ADC アプライアンスの`/nsconfig/ssl`ディレクトリに保存することをお勧めします。`/etc` ディレクトリは、Citrix ADC アプライアンスのフラッシュメモリに存在します。この操作により、移植性が提供され、アプライアンス上の証明書ファイルのバックアップと復元が容易になります。

注**:** 証明書とキーファイルが同じディレクトリに保存されていることを確認してください。

Citrix ADC アプライアンスでサポートされる証明書キーの最大サイズはどれくらいですか

リリース 9.0 より前のソフトウェアリリースを実行している Citrix ADC アプライアンスは、2048 ビットの最大証明書キーサイズをサポートします。リリース 9.0 以降では、4096 ビットの最大証明書キーサイズがサポートされています。この制限は RSA 証明書に適用されます。

MPX アプライアンスは、512 ビットから次のサイズまでの証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）

- バックエンドサーバー上の 4096 ビット証明書
- 4096 ビットのクライアント証明書 (仮想サーバーでクライアント認証が有効になっている場合)

仮想アプライアンスは、512 ビットから次のサイズまでの証明書をサポートします。

- 仮想サーバー上の 4096 ビットサーバー証明書
- サービス上の 4096 ビットのクライアント証明書
- 4096 ビット CA 証明書（中間証明書とルート証明書を含む）
- リリース 12.0-56.x のバックエンドサーバー上の 4096 ビット証明書。古いリリースでは 2048 ビット証明書がサポートされています。
- リリース 12.0-56.x からの 2048 ビットのクライアント証明書（仮想サーバでクライアント認証が有効になっている場合）。

Citrix ADC アプライアンスでサポートされる **DH** パラメータの最大サイズはどれくらいですか

Citrix ADC アプライアンスは、最大 2048 ビットの DH パラメータをサポートしています。

Citrix ADC アプライアンスでサポートされる証明書チェーンの最大長、つまりチェーン内の最大証明書数はどれくらいですか

Citrix ADC アプライアンスは、サーバー証明書メッセージを送信するときに、チェーン内で最大 10 個の証明書を送信できます。最大長のチェーンには、サーバ証明書と 9 つの中間 CA 証明書が含まれます。

Citrix ADC アプライアンスでサポートされるさまざまな証明書とキーの形式は何ですか

Citrix ADC アプライアンスは、次の証明書とキーの形式をサポートしています。

- プライバシー強化メール (PEM)
- 識別エンコーディングルール (DER)

Citrix ADC アプライアンスにインストールできる証明書とキーの数に制限はありますか

なしインストールできる証明書とキーの数は、Citrix ADC アプライアンスの使用可能なメモリによってのみ制限されます。

証明書とキーファイルをローカルコンピューターに保存しました。**FTP** プロトコルを使用してこれらのファイルを **Citrix ADC** アプライアンスに転送したい。これらのファイルを **Citrix ADC** アプライアンスに転送するための優先モードはありますか

はい。FTP プロトコルを使用している場合は、バイナリモードを使用して証明書とキーファイルを Citrix ADC アプライアンスに転送する必要があります。

注**:** デフォルトでは、FTP は無効になっています。証明書およびキーファイルを転送するには、SCP プロトコルをお勧めします。構成ユーティリティは、暗黙的に SCP を使用してアプライアンスに接続します。

証明書とキーのデフォルトのディレクトリパスは何ですか

証明書とキーのデフォルトのディレクトリパスは '/nsconfig/ssl' です。

証明書とキーペアを追加するときに、証明書とキーファイルへの絶対パスを指定しないとどうなりますか

証明書とキーのペアを追加するときは、証明書とキーファイルへの絶対パスを指定します。指定しない場合、ADC アプライアンスはこれらのファイルをデフォルトディレクトリで検索し、カーネルにロードしようとします。デフォルトのディレクトリは/nsconfig/sslです。たとえば、アプライアンスの/nsconfig/sslディレクトリで cert1024.pem ファイルと rsa1024.pem ファイルが使用可能な場合、次のコマンドは両方とも正常に実行できます。

```
1 add ssl certKey cert1 -cert cert1204.pem -key rsa1024.pem
2 <!--NeedCopy-->
```

```
1 add ssl certKey cert1 -cert /nsconfig/ssl/cert1204.pem -key /nsconfig/
     ssl/rsa1024.pem
2 <!--NeedCopy-->
```

高可用性セットアップを構成しました。セットアップに **SSL** 機能を実装したい。高可用性セットアップで証明書とキーファイルをどのように処理する必要がありますか

高可用性設定では、プライマリとセカンダリの Citrix ADC アプライアンスの両方に証明書とキーファイルを保存する必要があります。プライマリアプライアンスに SSL 証明書とキーのペアを追加する前に、証明書とキーファイルのディレクトリパスが両方のアプライアンスで同じである必要があります。

nCipher nShield® HSM

nCipher nShield® HSM と統合する場合、**Citrix ADC** アプライアンスを **HA** に追加する際に特定の構成に留意する必要がありますか

HA の両方のノードで同じ nCipher デバイスを構成します。nCipher 構成コマンドは HA で同期しません。nCipher nShield® HSM の前提条件については、「前提条件」を参照してください。

nCipher nShield® HSM と **RFS** の両方を個別に統合する必要がありますか**? HA** セットアップの前後にこのアクションを完了する必要がありますか

統合は、HA セットアップの前後に完了できます。HA セットアップ後に統合が行われると、セカンダリノードを設定する前にプライマリノードにインポートされたキーは、セカンダリノードに同期されません。したがって、高可用性セットアップの前に nCipher 統合をお勧めします。

プライマリとセカンダリの両方の **Citrix ADC** アプライアンスにキーをインポートする必要がありますか、それともプライマリノードからセカンダリノードにキーが同期されていますか

nCipher が HA を形成する前に両方のデバイスに統合されている場合、キーは統合プロセスで RFS から自動的に同期されます。

HSM が **Citrix ADC** アプライアンスではなく **nCipher** 上にある場合、ノードに障害が発生して交換されたときのキーと証明書はどうなりますか

ノードに障害が発生した場合、新しいノードに nCipher を統合することで、キーと証明書を新しいノードに同期できます。次に、次のコマンドを実行します。

```
sync ha files ssl
force ha sync
<!--NeedCopy-->
```

nCipher の統合プロセスでキーが同期されている場合、証明書は同期され、追加されます。

暗号

ヌル暗号って何ですか

暗号化のない暗号は、NULL-Ciphers と呼ばれます。たとえば、NULL-MD5 はヌル暗号です。

SSL VIP または **SSL** サービスの場合、**NULL** 暗号はデフォルトで有効になっていますか

なし SSL VIP または SSL サービスでは、NULL 暗号はデフォルトで有効になりません。

ヌル暗号を削除する手順は何ですか

SSL VIP から NULL 暗号を削除するには、次のコマンドを実行します。


```
bind ssl cipher <SSL_VIP> REM NULL
<!--NeedCopy-->
```

SSL サービスから NULL 暗号を削除するには、次のコマンドを実行します。

```
bind ssl cipher <SSL_Service> REM NULL -service
<!--NeedCopy-->
```

Citrix ADC アプライアンスでサポートされるさまざまな暗号エイリアスは何ですか

アプライアンスでサポートされている暗号エイリアスを一覧表示するには、コマンドプロンプトで次のように入力します。

```
sh cipher
<!--NeedCopy-->
```

Citrix ADC アプライアンスの事前定義された暗号をすべて表示するコマンドは何ですか

Citrix ADC アプライアンスの事前定義された暗号をすべて表示するには、CLI で次のように入力します。

```
show ssl cipher
<!--NeedCopy-->
```

Citrix ADC アプライアンスの個々の暗号の詳細を表示するコマンドは何ですか

Citrix ADC アプライアンスの個々の暗号の詳細を表示するには、CLI で次のように入力します。

```
show ssl cipher <Cipher_Name/Cipher_Alias_Name/Cipher_Group_Name>
<!--NeedCopy-->
```

例:

```
show cipher SSL3-RC4-SHA
      1) Cipher Name: SSL3-RC4-SHA
```



```
3       Description: SSLv3 Kx=RSA Au=RSA Enc=RC4(128)
4      Mac=SHA1
5       Done
6  <!--NeedCopy-->
```

Citrix ADC アプライアンスの事前定義された暗号を追加することの意義は何ですか

Citrix ADC アプライアンスの事前定義された暗号を追加すると、NULL 暗号が SSL VIP または SSL サービスに追加されます。

Citrix ADC アプライアンスの暗号グループからバインド解除せずに暗号の順序を変更することは可能ですか

はい。カスタム暗号グループから暗号をバインド解除することなく、暗号の順序を変更できます。ただし、組み込みの暗号グループの優先順位を変更することはできません。SSL エンティティにバインドされた暗号の優先順位を変更するには、まず仮想サーバー、サービス、またはサービスグループから暗号をバインド解除します。

注**:** SSL エンティティにバインドされた暗号グループが空の場合、ネゴシエートされた暗号がないため、SSL ハンドシェイクは失敗します。暗号グループには、少なくとも 1 つの暗号が含まれている必要があります。

ECDSA は **Citrix ADC** アプライアンスでサポートされていますか

ECDSA は、次の Citrix ADC プラットフォームでサポートされています。サポートされているビルドの詳細については、Citrix ADC アプライアンスで使用可能な暗号の表 1 と表 2 を参照してください。

- N3 チップを搭載した Citrix ADC MPX および SDX アプライアンス
- Citrix ADC MPX 5900/8900/15000/26000
- Citrix ADC SDX 8900/15000
- Citrix ADC VPX アプライアンス

Citrix ADC VPX アプライアンスは、フロントエンドで **AES-GCM/SHA2** 暗号をサポートしていますか

はい、AES-GCM/SHA2 暗号は Citrix ADC VPX アプライアンスでサポートされています。サポートされているビルドの詳細については、Citrix ADC アプライアンスで使用可能な暗号を参照してください。

証明書

クライアント証明書の識別名は、ユーザーセッションの長さで使用できますか

はい。ユーザーセッションの間は、後続のリクエストでクライアント証明書の識別名にアクセスできます。つまり、SSL ハンドシェイクが完了し、証明書がブラウザによって再度送信されない後でも同様です。次の設定例で説明されているように、変数と代入を使用します。

例:

```
add ns variable v2 -type ”text(100)”

add ns assignment a1 -variable ”$v2” -set       ”CLIENT.SSL.CLIENT_CERT
    .SUBJECT.TYPECAST_NVLIST_T('=','/').VALUE(”CN”)”

add rewrite action act1 insert_http_header subject ”$v2”  // example:
    to insert the distinguished name in the header

add rewrite policy pol1 true a1

add rewrite policy pol2 true act1

bind rewrite global pol1 1 next -type RES_DEFAULT

bind rewrite global pol2 2 next -type RES_DEFAULT

set rewrite param -undefAction RESET
<!--NeedCopy-->
```

サーバー証明書をバインドする必要があるのはなぜですか

サーバー証明書のバインドは、SSL 設定で SSL トランザクションを処理するための基本的な要件です。

サーバー証明書を SSL VIP にバインドするには、CLI で次のように入力します。

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```

サーバー証明書を SSL サービスにバインドするには、CLI で次のように入力します。

```
bind ssl service <serviceName> -certkeyName <cert_name>
<!--NeedCopy-->
```

SSL VIP または **SSL** サービスにバインドできる証明書はいくつありますか

Citrix ADC VPX、MPX/SDX（N3）、および MPX/SDX 14000 FIPS アプライアンスでは、2 つの証明書を SSL 仮想サーバーまたは SSL サービスにバインドできます（SNI が無効の場合）。証明書は RSA および ECDSA の各タイプの

1 つである必要があります。SNI が有効の場合、RSA または ECDSA タイプの複数のサーバ証明書をバインドできます。Citrix ADC MPX（N2）または MPX 9700 FIPS アプライアンスで、SNI が無効になっている場合は、RSA タイプの証明書を 1 つだけバインドできます。SNI が有効な場合、RSA タイプの複数のサーバ証明書のみをバインドできます。

サーバー証明書のバインドを解除または上書きするとどうなりますか

サーバー証明書のバインドを解除または上書きすると、既存の証明書を使用して作成されたすべての接続と SSL セッションが終了します。既存の証明書を上書きすると、次のメッセージが表示されます。

```
ERROR:

Warning: Current certificate replaces the previous binding.
<!--NeedCopy-->
```

Citrix ADC アプライアンスに中間証明書をインストールし、サーバー証明書にリンクするにはどうすればよいですか

中間証明書のインストールについては、「http://support.citrix.com/article/ctx114146」の記事を参照してください。

Citrix ADC に証明書をインストールしようとすると、「リソースはすでに存在します」というエラーが発生するのはなぜですか

「リソースはすでに存在します」エラーの解決方法については、「http://support.citrix.com/article/CTX117284」の記事を参照してください。

Citrix ADC アプライアンスでサーバー証明書を作成して、製品をテストおよび評価したい。サーバー証明書を作成する手順は何ですか

次の手順を実行して、テスト証明書を作成します。

注: この手順で作成された証明書を使用して、すべてのユーザおよびブラウザを認証することはできません。テストに証明書を使用した後は、承認されたルート認証局によって署名されたサーバー証明書を取得する必要があります。

自己署名サーバー証明書を作成するには、次の手順を実行します。

1. ルート CA 証明書を作成するには、CLI で次のように入力します。


```
create ssl rsakey /nsconfig/ssl/test-ca.key 1024

create ssl certreq /nsconfig/ssl/test-ca.csr -keyfile /nsconfig/
   ssl/test-ca.key

Enter the required information when prompted, and then type the
   following command:

create ssl cert /nsconfig/ssl/test-ca.cer /nsconfig/ssl/test-ca.
   csr ROOT_CERT -keyfile /nsconfig/ssl/test-ca.key
<!--NeedCopy-->
```

2. 次の手順を実行して、サーバー証明書を作成し、作成したルート CA 証明書で署名します。

   a) リクエストとキーを作成するには、CLI で次のように入力します。

```
create ssl rsakey /nsconfig/ssl/test-server.key 1024

    create ssl certreq /nsconfig/ssl/test-server.csr -keyfile
       /nsconfig/ssl/test-server.key
<!--NeedCopy-->
```

   b) プロンプトが表示されたら、必要な情報を入力します。

   c) シリアル番号ファイルを作成するには、CLI で次のように入力します。

```
shell
 # echo '01' >
/nsconfig/ssl/serial.txt
 # exit
<!--NeedCopy-->
```

   d) ステップ 1 で作成したルート CA 証明書によって署名されたサーバ証明書を作成するには、CLI で次のように入力します。

```
create ssl cert /nsconfig/ssl/test-server.cer /nsconfig/ssl/
   test-server.csr SRVR_CERT -CAcert /nsconfig/ssl/test-ca.cer
    -CAkey /nsconfig/ssl/test-ca.key -CAserial /nsconfig/ssl/
   serial.txt
<!--NeedCopy-->
```


e) SSL ハンドシェイクとバルク暗号化のサーバー証明書情報を保持するメモリ内オブジェクトである Citrix ADC 証明書とキーのペアを作成するには、CLI で次のように入力します。

```
add ssl certkey test-certkey -cert /nsconfig/ssl/test-server.
    cer -key /nsconfig/ssl/test-server.key
<!--NeedCopy-->
```

f) 証明書とキーのペアを SSL 仮想サーバーにバインドするには、CLI で次のように入力します。

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```

NetScaler ソフトウェアリリース **9.0** がインストールされている **Citrix ADC** アプライアンスを受け取りました。アプライアンスに追加のライセンスファイルがあることに気付きました。**NetScaler** ソフトウェアリリース **9.0** 以降のライセンスポリシーに変更はありますか

はい。Citrix NetScaler ソフトウェアリリース 9.0 以降、アプライアンスに単一のライセンスファイルがない可能性があります。ライセンスファイルの数は、Citrix ADC ソフトウェアリリースエディションによって異なります。たとえば、Advanced Edition をインストールしている場合、さまざまな機能の完全な機能のために追加のライセンスファイルが必要になる場合があります。ただし、Premium エディションをインストールした場合、アプライアンスにはライセンスファイルが 1 つしかありません。

インターネットインフォメーションサービス **(IIS)** から証明書をエクスポートするにはどうすればよいですか

さまざまな方法がありますが、次の方法を使用して、Web サイトの適切な証明書と秘密キーがエクスポートされます。この手順は、実際の IIS サーバーで実行する必要があります。

1. インターネットインフォメーションサービス (IIS) マネージャー管理ツールを開きます。
2. Web サイトノードを展開し、Citrix ADC アプライアンスを介して提供する SSL 対応の Web サイトを見つけます。
3. この Web サイトを右クリックし、[プロパティ] をクリックします。
4. [ディレクトリセキュリティ] タブをクリックし、ウィンドウの [セキュリティで保護された通信] セクションで、[証明書の表示] ボックスを選択します。
5. [詳細] タブをクリックし、[ファイルにコピー] をクリックします。
6. [証明書のエクスポートウィザードへようこそ] ページで、[次へ] をクリックします。
7. [はい、秘密キーをエクスポートする] を選択し、[次へ] をクリックします。

   注： SSL オフロードが Citrix ADC で動作するには、秘密キーをエクスポートする必要があります。

8. *[個人情報交換-PKCS #12]* ラジオボタンが選択されていることを確認し、*[*可能な場合はすべての証明書を証明パスに含める*]* チェックボックスをオンにします。［次へ］をクリックします。
9. パスワードを入力し、[次へ] をクリックします。
10. ファイル名と場所を入力し、[次へ] をクリックします。ファイルの拡張子を.PFX にします。
11. ［完了］をクリックします。

PKCS #12 証明書を変換して **Citrix ADC** にインストールするにはどうすればよいですか

1. エクスポートされた.PFX 証明書ファイルを、Citrix ADC アプライアンスにコピーできる場所に移動します。つまり、Citrix ADC アプライアンスの管理インターフェイスへの SSH アクセスを許可するマシンです。SCP などのセキュアコピーユーティリティを使用して、証明書をアプライアンスにコピーします。
2. BSD シェルにアクセスし、証明書（Cert.pfx など）を.PEM 形式に変換します。

```
root@ns# openssl pkcs12 -in cert.PFX -out cert.PEM
<!--NeedCopy-->
```

3. 変換された証明書が正しい x509 形式であることを確認するには、次のコマンドでエラーが生成されないことを確認します。

```
root@ns# openssl x509 -in cert.PEM -text
<!--NeedCopy-->
```

4. 証明書ファイルに秘密キーが含まれていることを確認します。次のコマンドを発行することから始めます。

```
root@ns# cat cert.PEM

Verify that the output file includes an RSA PRIVATE KEY section.

-----BEGIN RSA PRIVATE KEY-----
Mkm^s9KMs9023pz/s...
-----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

次に、RSA 秘密キーセクションの別の例を示します。


```
    Bag Attributes
    1.3.6.1.4.1.311.17.2: <No Values>
    localKeyID: 01 00 00 00
    Microsoft CSP Name: Microsoft RSA SChannel Cryptographic
    Provider
    friendlyName:
    4b9cef4cc8c9b849ff5c662fd3e0ef7e_76267e3e-6183-4d45-886e-6
       e067297b38f

    Key Attributes
    X509v3 Key Usage: 10
    -----BEGIN RSA PRIVATE KEY-----
    Proc-Type: 4,ENCRYPTED
    DEK-Info: DES-EDE3-CBC,43E7ACA5F4423968
    pZJ2SfsSVqMbRRf6ug37Clua5gY0Wld4frPIxFXyJquUHr31dilW5ta3hbIaQ+
       Rg

    ... (more random characters)
    v8dMugeRplkaH2Uwt/mWBk4t71Yv7GeHmcmjafK8H8iW80ooPO3D/ENV8X4U/
       tlh

    5eU6ky3WYZ1BTy6thxxLlwAullynVXZEflNLxq1oX+ZYl6djgjE3qg==
    -----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

以下は、サーバー証明書のセクションです。

```
    Bag Attributes
    localKeyID: 01 00 00 00
    friendlyName: AG Certificate
    subject=/C=AU/ST=NSW/L=Wanniassa/O=Dave Mother
    Asiapacific/OU=Support/CN=davemother.food.lan
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIFiTCCBHGgAwIBAgIKCGryDgAAAAAHzANBgkqhkiG9w0BAQUFADA8MRMwEQYK


    ... (more random characters) 5
       pLDWYVHhLkA1pSxvFjNJHRSIydWHc5ltGyKqIUcBezVaXyel94pNSUYx07NpPV
       /

    MY2ovQyQZM8gGe3+lGFum0VHbv/y/gB9HhFesog=
    -----END CERTIFICATE-----
```

```
<!--NeedCopy-->
```

以下は、中級 CA 証明書のセクションです。

```
    Bag Attributes: <Empty Attributes>
    subject=/DC=lan/DC=food/CN=hotdog
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIESDCCAzCgAwIBAgIQah20fCRYTY9LRXYMIRaKGjANBgkqhkiG9w0BAQUFADA8


    ... (more random characters)
       Nt0nksawDnbKo86rQcNnY5xUs7c7pj2zxj/IOsgNHUp5W6dDI9pQoqFFaDk
       =

    -----END CERTIFICATE-----
<!--NeedCopy-->
```

エクスポートされた証明書の証明書パスによっては、さらに中間 CA 証明書が続く場合があります。

5. テキストエディタで.PEM ファイルを開きます
6. .PEM ファイルの最初の行と次の行の最初のインスタンスを見つけ、それらの 2 行とその間のすべての行をコピーします。

```
-----END CERTIFICATE-----

Note: Make sure that last copied line is the first
-----END CERTIFICATE----- line in the .PEM file.

<!--NeedCopy-->
```

7. コピーした行を新しいファイルに貼り付けます。cert-key.pem など、新しいファイルを直観的に呼び出します。この証明書とキーのペアは、HTTPS サービスをホストするサーバー用です。このファイルには、RSA PRIVATE KEY というラベルの付いたセクションと、前述の例の SERVER CERTIFICATE というラベルの付いたセクションの両方が含まれている必要があります。

   注**:** 証明書とキーのペアファイルには、秘密キーが含まれているため、安全に保つ必要があります。

8. —BEGIN CERTIFICATE— で始まり —END CERTIFICATE— で終わる後続のセクションを探し、そのようなセクションをそれぞれ別の新しいファイルにコピーします。

   これらのセクションは、証明書パスに含まれている信頼できる CA の証明書に対応しています。これらのセク

ションは、これらの証明書の新しい個別のファイルにコピーして貼り付ける必要があります。たとえば、前の例の中間 CA 証明書セクションをコピーして新しいファイルに貼り付ける必要があります)。

元のファイルに複数の中間 CA 証明書がある場合は、各中間 CA 証明書のファイルを、ファイル内の順番に作成します。後の手順で正しい順序で証明書をリンクする必要があるため、証明書が表示される順序を（適切なファイル名を使用して）追跡します。

9. 証明書キーファイル（cert-key.pem）および追加の CA 証明書ファイルを Citrix ADC アプライアンス上の/nsconfig/ssl ディレクトリにコピーします。
10. BSD シェルを終了し、Citrix ADC プロンプトにアクセスします。
11. 「アプライアンスに証明書キーファイルをインストールする」の手順に従って、デバイスにアップロードされたキー/証明書をインストールします。

PKCS #7 証明書を変換して **Citrix ADC** アプライアンスにインストールするにはどうすればよいですか

OpenSSL を使用して、PKCS #7 証明書を Citrix ADC アプライアンスで認識できる形式に変換できます。この手順は PKCS #12 証明書の手順と同じです。ただし、異なるパラメータで OpenSSL を呼び出す点が異なります。PKCS #7 証明書を変換する手順は次のとおりです。

1. SCP などのセキュアコピーユーティリティを使用して、証明書をアプライアンスにコピーします。
2. 証明書（Cert.p7b など）を PEM 形式に変換します。

```
openssl pkcs7 -inform DER -in cert.p7b -print_certs -text -out
   cert.pem
<!--NeedCopy-->
```

3. PKCS #12 証明書の回答で説明されている手順 3 ～7 に従います。
   注：変換された PKCS #7 証明書をアプライアンスにロードする前に、PKCS #12 手順のステップ 3 で説明したとおりに秘密キーが含まれていることを確認します。PKCS #7 証明書、特に IIS からエクスポートされた証明書には、通常、秘密キーは含まれません。

bind cipher コマンドを使用して暗号を仮想サーバーまたはサービスにバインドすると、「**Command deprecated**?」

暗号を仮想サーバーまたはサービスにバインドするコマンドが変更されました。

SSL 暗号を SSL 仮想サーバにバインドするには、`bind ssl vserver <vsername> -ciphername <ciphername>` コマンドを使用します。

SSL 暗号を SSL サービスにバインドするには、`bind ssl service <serviceName> -ciphername <ciphername>` コマンドを使用します。

注: 新しい暗号および暗号グループは既存のリストに追加され、置き換えられません。

add cipher コマンドを使用して暗号グループを作成し、それに暗号をバインドできないのはなぜですか

add cipher コマンドの機能がリリース 10 で変更されました。このコマンドは、暗号グループのみを作成します。グループに暗号を追加するには、bind cipher コマンドを使用します。

openSSL

OpenSSL を使用して **PEM** と **DER** の間で証明書を変換するにはどうすればよいですか**?**

OpenSSL を使用するには、OpenSSL ソフトウェアを正常にインストールし、コマンドラインから OpenSSL を実行できる必要があります。

x509 証明書と RSA キーは、いくつかの異なる形式で保存できます。

一般的な形式は次のとおりです。

- DER (主に Java および Macintosh プラットフォームで使用されるバイナリ形式)
- PEM (ヘッダーとフッター情報を持つ DER の base64 表現。主に UNIX および Linux プラットフォームで使用されます)。

ルート証明書および中間証明書に加えて、キーと対応する証明書は、単一の PKCS #12 (.P12、.PFX) ファイルに格納することもできます。

手順

OpenSSL コマンドを使用して、次の形式を変換します。

1. 証明書を PEM から DER に変換するには、次の手順を実行します。

```
x509 -in input.crt -inform PEM -out output.crt -outform DER
<!--NeedCopy-->
```

2. 証明書を DER から PEM に変換するには、次の手順を実行します。

```
x509 -in input.crt -inform DER -out output.crt -outform PEM
<!--NeedCopy-->
```

3. キーを PEM から DER に変換するには:

```
rsa -in input.key -inform PEM -out output.key -outform DER
<!--NeedCopy-->
```


4. キーを DER から PEM に変換するには、次の手順を実行します。

```
rsa -in input.key -inform DER -out output.key -outform PEM
<!--NeedCopy-->
```

注: インポートするキーがサポートされている対称暗号で暗号化されている場合は、パスフレーズの入力を求められます。

注: キーを旧式の NET（Netscape サーバ）形式に変換するには、必要に応じて PEM または DER を NET に置き換えます。保存された鍵は弱い無塩 RC4 対称暗号で暗号化されるため、パスフレーズが要求されます。空白のパスフレーズは許容されます。

システム制限

覚えておくべき重要な数字は何ですか

1. 証明書リクエストの作成:
   - リクエストファイル名: 最大 63 文字
   - キーファイル名: 最大 63 文字
   - PEM パスフレーズ (暗号化キー用): 最大 31 文字
   - 一般名: 最大 63 文字
   - 市区町村: 最大 127 文字
   - 組織名: 最大 63 文字
   - 州/州名: 最大 63 文字
   - メールアドレス: 最大 39 文字
   - 組織単位: 最大 63 文字
   - チャレンジパスワード: 最大 20 文字
   - 会社名: 最大 127 文字
2. 証明書の作成:
   - 証明書ファイル名: 最大 63 文字
   - 証明書要求ファイル名: 最大 63 文字
   - キーファイル名: 最大 63 文字
   - PEM パスフレーズ: 最大 31 文字
   - 有効期間: 最大 3650 日
   - CA 証明書ファイル名: 最大 63 文字
   - CA キーファイル名: 最大 63 文字
   - PEM パスフレーズ: 最大 31 文字
   - CA シリアル番号ファイル: 最大 63 文字
3. サーバーテスト証明書を作成してインストールします。

- 証明書ファイル名: 最大 31 文字
- 完全修飾ドメイン名: 最大 63 文字

4. Diffie-Hellman (DH) キーを作成します。

   - DH ファイル名 (パスあり): 最大 63 文字
   - DH パラメータサイズ: 最大 2048 ビット

5. PKCS12 キーをインポート:

   - 出力ファイル名: 最大 63 文字
   - PKCS12 ファイル名: 最大 63 文字
   - パスワードのインポート: 最大 31 文字
   - PEM パスフレーズ: 最大 31 文字
   - PEM パスフレーズの確認: 最大 31 文字

6. PKCS12 をエクスポート

   - PKCS12 ファイル名: 最大 63 文字
   - 証明書ファイル名: 最大 63 文字
   - キーファイル名: 最大 63 文字
   - エクスポートパスワード: 最大 31 文字
   - PEM パスフレーズ: 最大 31 文字

7. CRL 管理:

   - CA 証明書ファイル名: 最大 63 文字
   - CA キーファイル名: 最大 63 文字
   - CA キーファイルパスワード: 最大 31 文字
   - インデックスファイル名: 最大 63 文字
   - 証明書ファイル名: 最大 63 文字

8. RSA キーの作成:

   - キーファイル名: 最大 63 文字
   - キーサイズ: 最大 4096 ビット
   - PEM パスフレーズ: 最大 31 文字
   - パスフレーズの確認: 最大 31 文字

9. SSL の詳細設定を変更します。

   - 最大 CRL メモリサイズ: 最大 1024 M バイト
   - 暗号化トリガタイムアウト (10 mS ティック): 最大 200
   - 暗号化トリガーパケット数: 最大 50
   - OCSP キャッシュサイズ: 最大 512 MB

10. 証明書のインストール:

    - 証明書とキーのペア名: 最大 31 文字

- 証明書ファイル名: 最大 63 文字
- 秘密キーファイル名: 最大 63 文字
- パスワード: 最大 31 文字
- 通知期間: 最大 100

11. 暗号グループの作成:

    - 暗号グループ名: 最大 39 文字

12. CRL の作成:

    - CRL 名: 最大 31 文字
    - CRL ファイル: 最大 63 文字
    - URL: 最大 127 文字
    - ベース DN: 最大 127 文字
    - バインド DN: 最大 127 文字
    - パスワード: 最大 31 文字
    - 日数: 最大 31

13. SSL ポリシーの作成:

    - 名前: 最大 127 文字

14. SSL アクションの作成:

    - 名前: 最大 127 文字

15. OCSP レスポンダーの作成:

    - 名前: 最大 32 文字
    - URL: 最大 128 文字
    - バッチの深さ: 最大 8
    - バッチ処理遅延: 最大 10000
    - タイムスキューで生産: 最大 86400
    - リクエストタイムアウト: 最大 120000

16. 仮想サーバーの作成:

    - 名前: 最大 127 文字
    - リダイレクト URL: 最大 127 文字
    - クライアントのタイムアウト: 最大 31536000 秒

17. サービスの作成:

    - 名前: 最大 127 文字
    - アイドルタイムアウト (秒):
      クライアント: 最大 31536000
      サーバー: 最大 31536000

18. サービスグループの作成:

    - サービスグループ名: 最大 127 文字
    - サーバー ID: 最大 4294967295
    - アイドルタイムアウト (秒):
      クライアント: 最大値 31536000
      サーバー: 最大 31536000

19. モニターの作成:

    - 名前: 最大 31 文字

20. サーバーを作成:

    - サーバー名: 最大 127 文字
    - ドメイン名: 最大 255 文字
    - 解決再試行: 最大 20939 秒

コンテンツ検査

October 7, 2021

最近では、さまざまなマルチメディアコンテンツを表示するためのデバイスタイプが拡張されています。デバイスタイプは、携帯電話からタブレット、およびデスクトップまでです。中間インフラストラクチャプロバイダーは、元のコンテンツを Web サーバーから、コンテンツを要求するデバイスに適した形式に変換する必要があります。外部デバイスは、トランスコードするコンテンツを検査し、それをクライアントに送り返します。これを達成するために一般的に使用されるプロトコルは ICAP です。ICAP を使用すると、Citrix ADC アプライアンスをさまざまな展開に配置できます。ICAP は、マルウェアやセキュリティの問題についてデータを検査するコンテンツ検査技術を使用します。

> 注
>
> HTTP/2 コンテンツ検査とは互換性がありません。を使用するアプリケーション HTTP/2 トラフィックがコンテンツ検査を介して送信される場合、正しく機能しない可能性があります。

リモートコンテンツ検査用の **ICAP**

October 7, 2021

インターネットコンテンツ適応プロトコル（ICAP）は、HTTP メッセージで付加価値変換サービスを実行するためのシンプルで軽量なプロトコルです。典型的なシナリオでは、ICAP クライアントは HTTP 要求と応答を 1 つ以上の ICAP サーバーに転送して処理します。ICAP サーバーは、要求に対してコンテンツ変換を実行し、要求または応答に対して実行する適切なアクションで応答を返します。

Citrix ADC アプライアンスの ICAP

Citrix ADC セットアップでは、アプライアンスはサードパーティの ICAP サーバー（マルウェア対策やデータ損失保護（DLP）など）と相互運用する ICAP クライアントとして機能します。アプライアンスが着信 Web トラフィックを受信すると、アプライアンスはトラフィックを傍受し、コンテンツ検査ポリシーを使用して HTTP 要求に ICAP 処理が必要かどうかを評価します。「はい」の場合、アプライアンスはメッセージを復号化し、プレーンテキストとして ICAP サーバーに送信します。ICAP サーバーは、要求メッセージに対してコンテンツ変換サービスを実行し、アプライアンスに応答を送り返します。適応されたメッセージは、HTTP 要求または HTTP 応答のいずれかになります。アプライアンスが複数の ICAP サーバーと相互運用している場合、アプライアンスは ICAP サーバーの負荷分散を実行します。このシナリオは、1 つの ICAP サーバですべてのトラフィック負荷を処理するのに十分でない場合に発生します。ICAP サーバーが変更されたメッセージを返すと、アプライアンスは変更されたメッセージをバックエンドのオリジンサーバーに転送します。

Citrix ADC アプライアンスは、着信トラフィックが HTTPS タイプの場合、セキュアな ICAP サービスを提供します。アプライアンスは SSL ベースの TCP サービスを使用して、アプライアンスと ICAP サーバー間のセキュアな接続を確立します。

ICAP リクエストの変更（REQMOD）の仕組み

要求変更（REQMOD）モードでは、Citrix ADC アプライアンスはクライアントから受信した HTTP 要求を ICAP サーバーに転送します。ICAP サーバーは、次のいずれかを実行します。

1. 変更されたバージョンの要求を返送し、アプライアンスは変更された要求をバックエンドのオリジンサーバーに送信するか、変更された要求を別の ICAP サーバーにパイプライン処理します。
2. 適応が必要ないことを示すメッセージで応答します。
3. エラーを返し、アプライアンスはエラーメッセージをユーザーに送信します。

ICAP レスポンス修正（RESPMOD）の仕組み

応答変更（RESPMOD）モードでは、Citrix ADC アプライアンスは ICAP サーバーに HTTP 応答を送信します（アプライアンスによって送信される応答は通常、オリジンサーバーによって送信される応答です）。ICAP サーバーは、次のいずれかを実行します。

1. 応答の変更バージョンを送信し、アプライアンスは応答をユーザーに送信するか、応答を別の ICAP サーバーにパイプラインします。
2. 適応が必要ないことを示すメッセージで応答します。
3. エラーを返し、アプライアンスは次にエラーメッセージをユーザーに送信します。

ICAP ライセンス

ICAP 機能は、Citrix ADC スタンドアロンまたは Citrix ADC Premium ライセンスエディションまたは Advanced ライセンスエディションを使用した高可用性セットアップで動作します。

コンテンツ変換サービス用の **ICAP** の構成

コンテンツ変換サービスに ICAP を使用するには、まずコンテンツ検査機能と負荷分散機能を有効にする必要があります。機能を有効にしたら、次のタスクを実行できます。

コンテンツ検査を有効にするには

Citrix ADC アプライアンスを ICAP クライアントとして動作させる場合は、まずコンテンツ検査および負荷分散機能を有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns feature contentInspection LoadBalancing
<!--NeedCopy-->
```

ICAP プロファイルの追加

Citrix ADC アプライアンスの ICAP 構成は、ICAP プロファイルと呼ばれるエンティティで指定されます。プロファイルには、ICAP 設定のコレクションがあります。設定には、ICAP 要求を動的に生成し、ICAP 応答を受信し、コンテンツ検査データをログに記録するためのパラメーターが含まれます。

ICAP サーバーへの ICAP 要求を動的に生成するために、新しいパラメーター「insertHTTPRequest」が ICAP プロファイルに追加されます。このパラメータが設定されている場合、アプライアンスは設定された値をポリシー式として受け取り、式を評価し、結果をカプセル化された HTTP 要求または応答として含め、ICAP サーバーに送信します。また、新しいパラメータ「insertICAPHeaders」は、ICAP ヘッダーを動的に評価して含めるように構成できます。

アプライアンスが ICAP 要求を送信し、ICAP サーバーに応答を受信しない場合、接続は応答しなくなります。これは、ICAP サーバーが応答を送信するか、セッションが解放されるまで発生します。この動作は、ICAP 応答タイムアウトオプションを構成することで処理できます。ICAP 応答が遅延している場合は、アクションの要求タイムアウトパラメータを設定できます。構成済みの要求タイムアウト内に Citrix ADC アプライアンスが応答を受信しない場合、要求タイムアウトアクションが実行されます。

ReqTimeout アクション：可能な値は、バイパス、リセット、ドロップです。
BYPASS：これは、リモート ICAP サーバーの応答を無視し、クライアント/サーバーに要求/応答を送信します。
RESET（デフォルト）：クライアント接続を閉じてリセットします。
DROP: ユーザーに応答を送信せずにリクエストを削除する

ICAP 応答を評価するために、新しいポリシー式 `ICAP.RES` がコンテンツインスペクションコールアウト戻り式で使用されます。この式は`HTTP_CALLOUT`の`HTTP.RES`式と同様 ICAP 応答を評価します。

たとえば、Citrix ADC アプライアンスが、Citrix ADC 仮想 IP アドレスの背後にホストされるサービスに対する HTTP リクエストを受信した場合、アプライアンスは外部サーバーとのクライアントの認証をチェックし、アクションを実行しなければならない場合があります。

コマンドプロンプトで入力します。

```
add ns icapProfile <name> [-preview ( ENABLED | DISABLED )][-previewLength
 <positive_integer>] -uri <string> [-hostHeader <string>] [-userAgent <
string>] -Mode ( REQMOD | RESPMOD )[-queryParams <string>] [-connectionKeepAlive
 ( ENABLED | DISABLED )][-allow204 ( ENABLED | DISABLED )] [-insertICAPHeaders
 <string>][-insertHTTPRequest <string>] [-reqTimeout <positive_integer>][-
reqTimeoutAction <reqTimeoutAction>] [-logAction <string>]
```

例:

```
add icapprofile reqmod-profile -mode RESPMOD -uri “/req_scan” -hostHeader
“Webroot.reqsca” -useragent “NS_SWG-Proxy”
```

```
add ns icapProfile icap_prof1 -uri ”/example”-Mode REQMOD -reqtimeout 4 -
reqtimeoutaction BYPASS
```

```
> add icapProfile reqmode-profile -uri '/example'-mode REQMOD -insertHTTPRequest
 q{ HTTP.REQ.METHOD + ””+ HTTP.REQ.URL + ”HTTP/1.1\r\n”+ ”Host: ”+ HTTP.REQ
.HOSTNAME + ”\r\n\r\n”}
```

ICAP コンテンツ検査アクションの記録

コンテンツ検査ログストリームレコードまたは SYSLOG ログを動的に生成するには、ICAP 応答で ICAP.RES ベースのポリシー式を使用できます。このパラメータは、ICAP プロファイルで構成して、動的ログレコードを生成するポリシー式を構成できます。

コマンドプロンプトで入力します。

```
add audit messageaction icap_log_expr INFORMATIONAL icap.res.full_header
```

```
set icapProfile reqmode-profile -logAction messageaction
```

ICAP サービスを **TCP** サービスまたは **SSL_TCP** サービスとして追加する

コンテンツ検査機能を有効にした後、負荷分散セットアップの一部となる ICAP サーバーに ICAP サービスを追加する必要があります。追加するサービスによって、Citrix ADC アプライアンスと負荷分散仮想サーバー間の ICAP 接続が提供されます。

注: 管理者は、ICAP サービスを追加し、コンテンツ検査アクションで ICAP サーバーの IP アドレスを直接構成できます。

コマンドプロンプトで、次のように入力します。

```
1 add service <name> <IP> <serviceType> <port>
2 <!--NeedCopy-->
```


例:

```
add service icapsv1 10.10.10.10 SSL_TCP 1345

add service icapsv2 10.10.10.11 SSL_TCP 1345
```

TCP または **SSL_TCP** ベースの負荷分散仮想サーバーの追加

ICAP サービスを作成したら、ICAP トラフィックを受け入れ、ICAP サーバーの負荷分散を行う仮想サーバーを作成する必要があります。

> 注:
>
> また、SSL ベースの TCP サービスを、セキュリティで保護されたチャネル経由で使用することもできます。あなたは SSL_TCP サービスを提供し、コンテンツ検査アクションにバインドします。

コマンドプロンプトで、次のように入力します。

```
1 add lb vserver <name> <serviceType> <port>
2 <!--NeedCopy-->
```

例:

```
1 add lb vserver vicap TCP 0.0.0.0.0  - persistenceType NONE -cltTimeout
     9000
2
3 add lb vserver vicap SSL_TCP 0.0.0.0 0  - persistenceType NONE -
     cltTimeout 9000
4 <!--NeedCopy-->
```

負荷分散仮想サーバーに **ICAP** サービスをバインドする

ICAP サービスと仮想サーバーを作成したら、ICAP サービスを仮想サーバーにバインドする必要があります。

コマンドプロンプトで、次のように入力します。

```
1 bind lb vserver <name> <serviceName>
2 <!--NeedCopy-->
```

例:


```
bind lb vserver vicap icapsv1
<!--NeedCopy-->
```

コンテンツ検査アクションの追加

コンテンツ検査機能を有効にしたら、ICAP リクエスト情報を処理するための ICAP アクションを追加する必要があります。作成された ICAP プロファイルとサービス、または負荷分散仮想サーバーは、ICAP アクションにバインドされます。ICAP サーバーがダウンしている場合は、アプライアンスの `ifserverdown` パラメーターを構成して、次のいずれかのアクションを実行できます。

CONTINUE：リモートサーバーがダウンしているときにユーザーがコンテンツ検査をバイパスしたい場合は、デフォルトで「CONTINUE」アクションを選択できます。
RESET（デフォルト）：このアクションは、RST との接続を閉じることによってクライアントに応答します。
DROP：このアクションは、ユーザーに応答を送信せずにパケットをサイレントにドロップします。

コマンドプロンプトで、次のように入力します。

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>

add ContentInspection action <name> -type ICAP -serverip <ip> -
    serverport <port> -icapProfileName <string>
<!--NeedCopy-->
```

> 注:
>
> 負荷分散仮想サーバーの代わりに ICAP サービスを構成できる場合は、`\<-serverip>` オプションでサービス名を指定できます。コンテンツ検査アクションを追加すると、TCP サービスはポート 1344 の指定された IP アドレスに対して自動的に作成され、ICAP 通信に使用されます。

例:

```
add ContentInspection action ci_act_lb -type ICAP -serverName vicap -
    icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName icapsv1
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverip 1.1.1.1 -
    serverport 1344 -icapProfileName icap_reqmod
<!--NeedCopy-->
```


コンテンツ検査ポリシーの追加

コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを作成して、ICAP 処理と監査ロギングの要求を評価する必要があります。ポリシーは、1 つ以上の式で構成される規則に基づいています。ルールは、要求がルールに一致した場合に関連付けられるコンテンツインスペクションアクションに関連付けられます。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

例:

```
add ContentInspection policy ci_pol_basic  - rule true  - action
    ci_act_svc

add ContentInspection policy ci_pol_HTTP  - rule HTTP.REQ.URL.CONTAINS(
    “html” )  - action ci_act_svc
<!--NeedCopy-->
```

コンテンツ検査ポリシーをコンテンツスイッチングまたは負荷分散仮想サーバーにバインドする

ICAP ポリシーを有効にするには、ICAP ポリシーをグローバルにバインドするか、アプリケーションのフロントエンドである Content Switching または負荷分散仮想サーバーにバインドする必要があります。ポリシーをバインドするときは、ポリシーに優先順位を割り当てる必要があります。優先順位によって、定義したポリシーが評価される順序が決まります。

> 注:
>
> アプリケーション仮想サーバーのタイプは、HTTP/SSL/CS-PROXY である必要があります。

コンテンツ変換後にトラフィックをバックエンドオリジンサーバーに転送するための負荷分散設定の構成については、「負荷分散」を参照してください。

セキュリティで保護された **ICAP** サービスの構成

Citrix ADC アプライアンスと ICAP Web サーバー間のセキュアな接続を確立するために、アプライアンスは SSL ベースの TCP サービスまたは ICAP アクションにバインドされた負荷分散仮想サーバーを使用します。

セキュリティで保護された ICAP 接続を確立するには、次のタスクを実行します。

1. SSL ベースの TCP サービスを追加します。
2. SSL ベースの TCP サービスを、TCP または SSL_TCP タイプの仮想サーバの負荷分散にバインドします。

3. SSL ベースの TCP サービスまたは負荷分散仮想サーバーをコンテンツ検査アクションにバインドします。

SSL ベースの **TCP** サービスを仮想サーバーの負荷分散に追加する

Citrix ADC アプライアンスと ICAP Web サーバー間のセキュアな接続を確立するために、アプライアンスは SSL ベースの TCP サービスまたは ICAP アクションにバインドされた負荷分散仮想サーバーを使用します。

セキュリティで保護された ICAP 接続を確立するには、次のタスクを実行します。

1. SSL ベースの TCP サービスを追加します。
2. SSL ベースの TCP サービスを、TCP または SSL_TCP タイプの仮想サーバの負荷分散にバインドします。

SSL ベースの TCP サービスまたは負荷分散仮想サーバーをコンテンツ検査アクションにバインド

SSL ベースの **TCP** サービスを仮想サーバーの負荷分散に追加する

コンテンツ検査機能を有効にした後、負荷分散セットアップの一部となるセキュリティで保護された ICAP サービスを追加する必要があります。追加するサービスにより、Citrix ADC アプライアンスと負荷分散仮想サーバー間のセキュアな ICAP 接続が提供されます。

コマンドプロンプトで、次のように入力します。

```
add service <name> <IP> <serviceType> <port>
<!--NeedCopy-->
```

例**:**

```
add service icapsv2 10.102.29.200 SSL_TCP 1344  -gslb NONE  -maxclient
    0  -maxReq 0  -cip DISABLED  -usip NO  -useproxport YES  -sp ON  -
    cltTimeout 9000  -svrTimeout 9000  -CKA NO  -TCPB  NO  -CMP NO
<!--NeedCopy-->
```

SSL_TCP または **TCP** 負荷分散仮想サーバーに **SSL_TCP** ベースの **TCP** サービスをバインドします

保護された ICAP サービスを作成したら、サービスを負荷分散仮想サーバーにバインドする必要があります。負荷分散仮想サーバーを使用して ICAP サーバーの負荷を分散する場合に必要です。

コマンドプロンプトで、次のように入力します。

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```


例:

```
bind lb vserver vicap icapsv2
<!--NeedCopy-->
```

SSL ベースの **TCP** サービスまたは負荷分散仮想サーバーをコンテンツ検査アクションにバインドする

ICAP 要求情報を処理するための ICAP アクションを追加し、SSL ベースの TCP サービスをアクションにバインドします。

コマンドプロンプトで、次のように入力します。

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>
<!--NeedCopy-->
```

例:

```
add ContentInspection action ci_act_svc -type ICAP -serverName icapsv2
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName vicap -
    icapProfileName icap_reqmod
<!--NeedCopy-->
```

GUI を使用して **ICAP** プロトコルを構成する

1. [負荷分散] > [サービス] に移動し、[追加] をクリックします。
2. [サービス] ページで、サービスの詳細を入力します。
3. [負荷分散] > [仮想サーバー] に移動します。HTTP/SSL タイプの負荷分散仮想サーバーを追加します。または、仮想サーバーを選択して [編集] をクリックすることもできます。
4. サーバーの基本情報を入力したら、[**Continue**] をクリックします。
5. [詳細設定] セクションで、[ポリシー] をクリックします。
6. [ポリシー] セクションに移動し、鉛筆アイコンをクリックして、コンテンツ検査ポリシーを構成します。
7. **[**ポリシーの選択**]** ページで、[コンテンツ検査] を選択します。[続行] をクリックします。
8. [ポリシーのバインド] セクションで、[**+**] をクリックしてコンテンツ検査ポリシーを追加します。
9. **ICAP** ポリシーの作成ページで、ポリシーの名前を入力します。
10. [アクション] フィールドで、[+] 記号をクリックして ICAP アクションを追加します。

11. 「**ICAP** アクションの作成」ページで、アクションの名前を入力します。
12. アクションの名前を入力します。
13. [**Server Name**] フィールドに、すでに作成されている TCP サービスの名前を入力します。
14. 「**ICAP** プロファイル」フィールドで、「+」記号をクリックして ICAP プロファイルを追加します。
15. 「**ICAP** プロファイルの作成」ページで、プロファイル名、URI、およびモードを入力します。
16. ［作成］をクリックします。
17. 「**ICAP** アクションの作成」ページで、「作成」をクリックします。
18. **ICAP** ポリシーの作成ページで、式エディターに「true」と入力し、「作成」をクリックします。
19. ［バインド］をクリックします。
20. コンテンツ検査機能を有効にするかどうかを確認するメッセージが表示されたら、[はい] をクリックします。
21. ［完了］をクリックします。

コンテンツ変換後にトラフィックをバックエンドオリジンサーバーに負荷分散および転送するための Citrix ADC GUI 構成の詳細については、「負荷分散」を参照してください。

GUI を使用してセキュリティで保護された **ICAP** プロトコルを構成する

1. [負荷分散] > [サービス] に移動し、[追加] をクリックします。
2. [サービス] ページで、サービスの詳細を入力します。
3. [負荷分散] > [仮想サーバー] に移動します。HTTP/SSL タイプの仮想サーバーを追加します。または、仮想サーバーを選択して [編集] をクリックすることもできます。
4. サーバーの基本情報を入力したら、[**Continue**] をクリックします。
5. [詳細設定**]** セクションで、[ポリシー] をクリックします。
6. [ポリシー] セクションに移動し、鉛筆アイコンをクリックして、コンテンツ検査ポリシーを構成します。
7. **[**ポリシーの選択**]** ページで、[コンテンツ検査] を選択します。［続行］をクリックします。
8. [ポリシーのバインド] セクションで、[**+**] をクリックしてコンテンツ検査ポリシーを追加します。
9. **ICAP** ポリシーの作成ページで、ポリシーの名前を入力します。
10. [アクション] フィールドで、[+] 記号をクリックして ICAP アクションを追加します。
11. 「**ICAP** アクションの作成」ページで、アクションの名前を入力します。
12. アクションの名前を入力します。
13. 「サーバー名」フィールドに、すでに作成されている TCP_SSL サービスの名前を入力します。
14. 「**ICAP** プロファイル」フィールドで、「+」記号をクリックして ICAP プロファイルを追加します。
15. 「**ICAP** プロファイルの作成」ページで、プロファイル名、URI、およびモードを入力します。
16. ［作成］をクリックします。
17. 「**ICAP** アクションの作成」ページで、「作成」をクリックします。
18. **ICAP** ポリシーの作成ページで、式エディターに「true」と入力し、「作成」をクリックします。
19. ［バインド］をクリックします。
20. コンテンツ検査機能を有効にするかどうかを確認するメッセージが表示されたら、[はい] をクリックします。
21. ［完了］をクリックします。

リモートコンテンツ検査の監査ログサポート

着信要求または発信応答がコンテンツ検査された場合、Citrix ADC アプライアンスは ICAP の詳細を記録します。アプライアンスは、詳細をログ・メッセージとして ns.log ファイルに保存します。

各ログメッセージには、通常、次の詳細が含まれます。

```
<Source IP> <Destination IP> <Domain> <ICAP server IP><ICAP Mode> <
    Service URI> <ICAP response> <Policy action>
<!--NeedCopy-->
```

コンテンツ検査要求ログメッセージの例:

```
Apr 18 14:45:41 <local0.info> 10.106.97.104 04/18/2018:14:45:41 GMT 0-
    PPE-0 : default CI ICAP_LOG 788 0 : Source 10.102.1.98:39048 -
    Destination 10.106.97.89:8011 - Domain 10.106.97.89 - Content-Type
    application/x-www-form-urlencoded - ICAP Server 10.106.97.99:1344 -
    Mode REQMOD - Service /example - Response 204 - Action FORWARD
<!--NeedCopy-->
```

コンテンツ検査された応答ログメッセージの例:

```
Apr 18 12:34:08 <local0.info> 10.106.97.104 04/18/2018:12:34:08 GMT 0-
    PPE-0 : default CI ICAP_LOG 71 0 : Source 10.106.97.105:18552 -
    Destination 10.106.97.99:80 - Domain NA - Content-Type NA - ICAP
    Server 10.106.97.99:1344 - Mode RESPMOD - Service /example -
    Response 400 - Action Internal Error
<!--NeedCopy-->
```

Citrix ADC とのインラインデバイス統合

October 7, 2021

侵入防止システム（IPS）や次世代ファイアウォール（NGFW）などのセキュリティデバイスは、ネットワーク攻撃からサーバーを保護します。これらのデバイスはレイヤ 2 インラインモードで展開され、その主な機能は、ネットワーク攻撃からサーバを保護し、ネットワーク上のセキュリティ上の脅威を報告することです。

脆弱な脅威を防ぎ、高度なセキュリティ保護を提供するために、Citrix ADC アプライアンスは 1 つ以上のインラインデバイスと統合されています。インラインデバイスには、IPS、NGFW などの任意のセキュリティデバイスを使用で

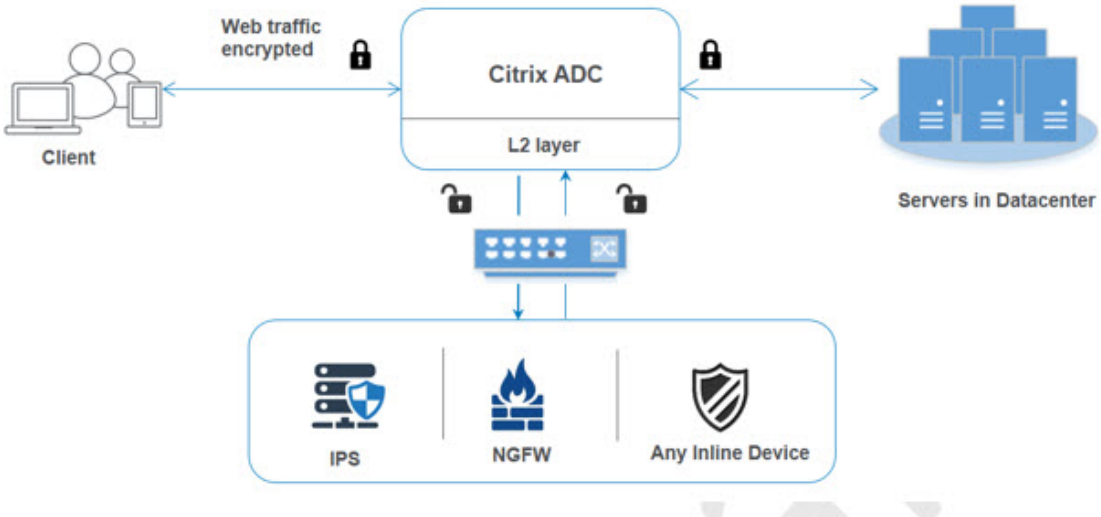
きます。

以下は、Citrix ADC アプライアンスとのインラインデバイス統合を使用することでメリットが得られるいくつかの使用例です。

- 暗号化されたトラフィックの検査。ほとんどの IPS および NGFW アプライアンスは、暗号化されたトラフィックをバイパスするため、サーバは攻撃に対して脆弱になります。Citrix ADC アプライアンスは、トラフィックを復号化し、検査のためにインラインデバイスに送信できます。これにより、お客様のネットワークセキュリティが強化されます。
- **TLS/SSL** 処理からのインラインデバイスのオフロード。TLS/SSL 処理にはコストがかかり、トラフィックを復号化すると、IPS または NGFW アプライアンスのシステム CPU が高くなる可能性があります。暗号化されたトラフィックが急速に増加するにつれて、これらのシステムは暗号化されたトラフィックの復号化と検査に失敗します。Citrix ADC は、インラインデバイスを TLS/SSL 処理からオフロードするのに役立ちます。その結果、大量のトラフィック検査をサポートするインラインデバイスが実現します。
- 負荷分散インラインデバイス。Citrix ADC アプライアンスは、大量のトラフィックがある場合、複数のインラインデバイスの負荷分散を行います。
- トラフィックのスマートな選択。アプライアンスに流れるすべてのパケットは、テキストファイルのダウンロードなど、コンテンツ検査が行われる場合があります。Citrix ADC アプライアンスを構成して、検査用に特定のトラフィック（.exe ファイルなど）を選択し、データを処理するためにトラフィックをインラインデバイスに送信できます。

Citrix ADC をインラインデバイスと統合する方法

次の図は、Citrix ADC がインラインセキュリティデバイスとどのように統合されているかを示しています。



インラインデバイスを Citrix ADC アプライアンスと統合すると、コンポーネントは次のように相互作用します。

1. クライアントが Citrix ADC アプライアンスに要求を送信します。
2. アプライアンスは要求を受信し、ポリシー評価に基づいてインラインデバイスに送信します。

注：複数のインラインデバイスがある場合、アプライアンスはデバイスの負荷分散を行い、トラフィックを送信します。

着信トラフィックが暗号化されたトラフィックの場合、アプライアンスはデータを復号化し、コンテンツ検査のためにプレーンテキストとしてインラインデバイスに送信します。

3. インラインデバイスは、データの脅威を検査し、データをドロップ、リセット、またはアプライアンスに戻すかどうかを決定します。
4. セキュリティ上の脅威がある場合、デバイスはデータを修正してアプライアンスに送信します。
5. Citrix ADC はデータを再暗号化し、要求をバックエンドサーバーに転送します。
6. バックエンドサーバーは、Citrix ADC アプライアンスにレスポンスを送信します。
7. アプライアンスは再びデータを復号化し、検査のためにインラインデバイスに送信します。
8. アプライアンスがデータを再暗号化し、応答をクライアントに送信する

ソフトウェアライセンス

インラインデバイス統合を展開するには、Citrix ADC アプライアンスに次のいずれかのライセンスをプロビジョニングする必要があります。

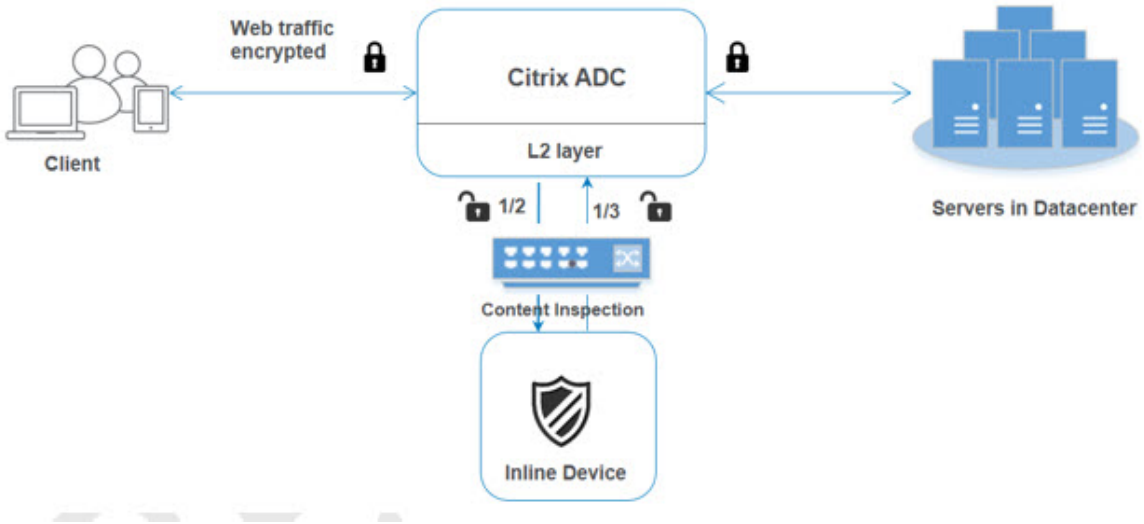
1. ADC Premium
2. ADC Advanced
3. Telco Advanced
4. Telco Premium
5. SWG license

インラインデバイス統合の設定

インラインデバイスを使用して Citrix ADC アプライアンスを構成するには、3 つの方法があります。構成シナリオは次のとおりです。

単一のインラインデバイスを使用する場合のシナリオ **1**

セキュリティデバイス（IPS または NGFW）をインラインモードで統合する場合は、最初にコンテンツ検査機能を有効にし、グローバルモードで MBF（MAC ベースの転送）で Citrix ADC を有効にすることから始める必要があります。機能を有効にしたら、コンテンツ検査プロファイルを追加し、インラインデバイスのコンテンツ検査アクションを追加して、検査に基づいてトラフィックをリセット、ブロック、またはドロップする必要があります。次に、アプライアンスのコンテンツ検査ポリシーを追加して、インラインデバイスに送信するトラフィックのサブセットを決定します。次に、サーバー上でレイヤ 2 接続を有効にして負荷分散仮想サーバーを構成します。最後に、コンテンツ検査ポリシーを負荷分散仮想サーバにバインドします。



MBF（**MAC** ベースフォワーディング）モードを有効にする

Citrix ADC アプライアンスを IPS やファイアウォールなどのインラインデバイスに統合する場合は、このモードを有効にする必要があります。MBF の詳細については、「MAC ベースフォワーディングを設定する」を参照してください。

コマンドプロンプトで入力します。

```
enable ns mode mbf
```

コンテンツ検査を有効にする

Citrix ADC アプライアンスでコンテンツを復号化してから、検査用のコンテンツをインラインデバイスに送信する場合は、コンテンツ検査と負荷分散機能を有効にする必要があります。

```
enable ns feature contentInspection LoadBalancing
```

レイヤ **2** 接続方法の追加

インラインデバイスによって生成された応答を処理するために、アプライアンスは VLAN チャネルをインラインデバイスとの通信のレイヤー 2 メソッド（L2ConnMethod）として使用します。

コマンドプロンプトで入力します。

```
set l4param -l2ConnMethod <l2ConnMethod>
```

例

```
set l4param -l2ConnMethod VlanChannel
```


サービスのコンテンツ検査プロファイルを追加

Citrix ADC アプライアンスのインラインデバイス構成は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルには、インラインデバイスとの統合方法を説明する設定のコレクションがあります。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3”
```

IPS-TCP モニタの追加

モニターを構成する場合は、ユーザー定義モニターを追加します。
注: モニターを構成する場合は、カスタムモニターを使用する必要があります。モニタを追加するときは、transparent パラメータを有効にする必要があります。

コマンドプロンプトで入力します。

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr>] [-destPort
 <port>] [-transparent ( YES | NO )]
```

例:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

サービスを追加する

サービスを追加します。インラインデバイスを含むどのデバイスにも所有されていないダミー IP アドレスを指定します。use source IP address (USIP) を YES に設定します。useproxyportを NO に設定します。デフォルトでは、ヘルスモニタリングは ON で、サービスをヘルスモニタにバインドし、モニタの [トランスペアレント] オプションも設定します。コマンドプロンプトで入力します。

```
add service <Service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor YES -usip ON -useproxyport OFF
```

例:

```
add service ips_service 192.168.10.2 TCP * -healthMonitor YES -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof
```


ヘルスモニターを追加する

デフォルトでは、ヘルスモニターはオンになっており、必要に応じて無効にするオプションもあります。コマンドプロンプトで入力します。

```
add lb monitor <name> TCP -destIP <ip address> -destPort 80 -transparent <
YES, NO>
```

例:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

サービスをヘルスモニターにバインドする

ヘルスモニターを構成したら、サービスをヘルスモニターにバインドする必要があります。コマンドプロンプトで入力します。

```
bind service <name> -monitorName <name>
```

例:

```
bind service ips_svc -monitorName ips_tcp
```

サービスの内容検査アクションを追加

コンテンツ検査機能を有効にした後、インラインプロファイルとサービスを追加した後、要求を処理するためのコンテンツ検査アクションを追加する必要があります。コンテンツ検査アクションに基づいて、インラインデバイスは、データを検査した後、アクションをドロップ、リセット、またはブロックできます。

インラインサーバーまたはサービスがダウンしている場合は、アプライアンスの ifserverdown パラメーターを構成して、次のいずれかのアクションを実行できます。

CONTINUE：リモートサーバーがダウンしているときにユーザーがコンテンツ検査をバイパスしたい場合は、デフォルトで「CONTINUE」アクションを選択できます。
RESET（デフォルト）：このアクションは、RST との接続を閉じることによってクライアントに応答します。
DROP：このアクションは、ユーザーに応答を送信せずにパケットをサイレントにドロップします。

コマンドプロンプトで入力します。

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-reqTimeoutAction
 <reqTimeoutAction>]
```

```
add ContentInspection action <action_name> -type InlineINSPECTION -serverName
 Service_name/Vserver_name>
```

例:


```
add ContentInspection action <Inline_action> -type InlineSPECTION -serverName
 Inline_service1
```

検査用のコンテンツ検査ポリシーの追加

コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを追加して、検査要求を評価する必要があります。ポリシーは、1 つ以上の式で構成される規則に基づいています。ポリシーは、ルールに基づいて検査対象のトラフィックを評価し、選択します。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

例

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーの追加

Web トラフィックを受信するには、負荷分散仮想サーバーを追加する必要があります。また、仮想サーバ上で layer2 接続を有効にする必要があります。

コマンドプロンプトで入力します。

```
add lb vserver <name> <vserver name> -l2Conn ON
```

例:

```
add lb vserver HTTP_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```

コンテンツ検査ポリシーをコンテンツスイッチング仮想サーバーまたは **HTTP/SSL** タイプの負荷分散仮想サーバーにバインド

タイプの負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーをバインドします HTTP/SSL コンテンツ検査ポリシーに。
コマンドプロンプトで、次のように入力します。

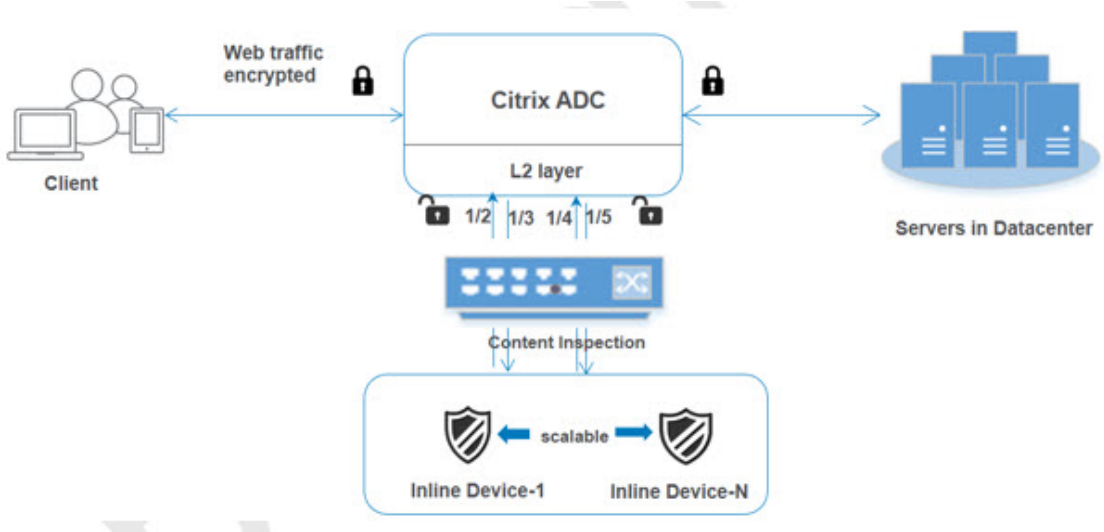
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

例:

```
bind lb vserver HTTP_vserver -policyName Inline_pol1 -priority 100 -type
REQUEST
```


シナリオ **2**: 専用インターフェイスを使用した複数のインラインデバイスの負荷分散

2 つ以上のインラインデバイスを使用している場合は、専用 VLAN 設定で、異なるコンテンツインスペクションサービスを使用してデバイスを負荷分散する必要があります。この場合、Citrix ADC アプライアンスは、専用のインターフェイスを介して各デバイスにトラフィックのサブセットを送信する上でデバイスの負荷分散を行います。
基本的な設定手順については、シナリオ 1 を参照してください。



サービス **1** のコンテンツ検査プロファイル **1** の追加

Citrix ADC アプライアンスのインライン構成は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルには、デバイス設定のコレクションがあります。コンテンツ検査プロファイル 1 はインラインサービス 1 用に作成され、通信は 1/2 および 1/3 専用インターフェイスを介して行われます。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3”
```

サービス **2** のコンテンツ検査プロファイル **2** の追加

コンテンツインスペクションプロファイル 2 が service2 に追加され、インラインデバイスは 1/4 および 1/5 専用インターフェイスを介してアプライアンスと通信します。
コマンドプロンプトで入力します。


```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile2 -type InlineInspection -
ingressinterface “1/4” -egressInterface “1/5”
```

インラインデバイス **1** のサービス **1** の追加

コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 1 のインラインサービス 1 を負荷分散設定の一部として追加する必要があります。追加するサービスは、すべてのインライン構成の詳細を提供します。

コマンドプロンプトで入力します。

```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
<Inline_Profile_1> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service Inline_service1 10.102.29.200 TCP 80 -contentInspectionProfileName
 Inline_profile1 -healthmonitor OFF -usip ON -useproxyport OFF
```

インラインデバイス **2** のサービス **2** の追加

コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 2 のインラインサービス 2 を追加する必要があります。追加するサービスは、すべてのインライン構成の詳細を提供します。

コマンドプロンプトで入力します。

```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
<Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service Inline_service1 10.29.20.205 TCP 80 -contentInspectionProfileName
 Inline_profile2 -healthmonitor OFF -usip ON -useproxyport OFF
```

負荷分散仮想サーバの追加

インラインプロファイルおよびサービスを追加したら、サービスの負荷分散用の負荷分散仮想サーバーを追加する必要があります。

コマンドプロンプトで入力します。

```
add lb vserver <vserver_name> TCP <Pvt_IP3> <port>
```


例:

```
add lb vserver lb-Inline_vserver TCP *
```

サービス **1** を負荷分散仮想サーバにバインド

負荷分散仮想サーバーを追加した後、負荷分散仮想サーバーを最初のサービスにバインドします。

コマンドプロンプトで入力します。

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-Inline_vserver Inline_service1
```

サービス **2** を負荷分散仮想サーバーにバインドする

負荷分散仮想サーバーを追加した後、サーバーを 2 番目のサービスにバインドします。

コマンドプロンプトで入力します。

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-Inline_vserver Inline_service2
```

サービスのコンテンツ検査アクションを追加する

コンテンツ検査機能を有効にしたら、インライン要求情報を処理するための Content Inspection アクションを追加する必要があります。選択されたアクションに基づいて、インラインデバイスは、指定されたトラフィックのサブセットを検査した後、ドロップ、リセット、またはブロックします。

コマンドプロンプトで入力します。

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-reqTimeoutAction
 <reqTimeoutAction>]
```

```
add ContentInspection action < action_name > -type InlineINSPECTION -
serverName Service_name/Vserver_name>
```

例:

```
add ContentInspection action Inline_action -type InlineINSPECTION -serverName
 lb-Inline_vserver
```


検査用のコンテンツ検査ポリシーの追加

コンテンツ検査アクションを作成した後、サービスの要求を評価するためにコンテンツ検査ポリシーを追加する必要があります。ポリシーは、1 つ以上の式で構成される規則に基づいています。ルールは、リクエストがルールに一致した場合に関連付けられるコンテンツ検査アクションに関連付けられます。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

例:

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーの追加

Web トラフィックを受け入れるために、コンテンツスイッチングまたは負荷分散仮想サーバーを追加します。また、仮想サーバ上で layer2 接続を有効にする必要があります。
ロードバランシングの詳細については、「負荷分散の仕組み」トピックを参照してください。

コマンドプロンプトで入力します。

```
add lb vserver <name> <vserver name> -l2Conn ON
```

例:

```
add lb vserver http_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプの仮想サーバーの負荷分散にバインドする

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを、コンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します。

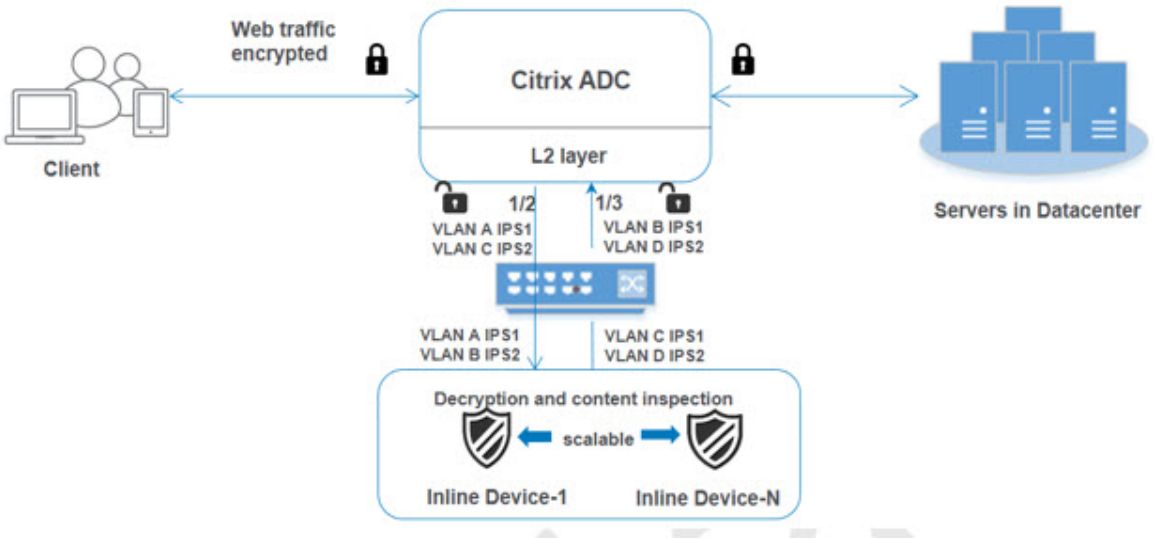
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <L7InlineREQUEST | L4Inline-REQUEST>
```

例:

```
bind lb vserver http_vserver -policyName Inline_pol1 -priority 100 -type
REQUEST
```

シナリオ **3**: 共有インターフェイスを使用した複数のインラインデバイスの負荷分散

複数のインラインデバイスを使用していて、共有 VLAN インターフェイスで異なるサービスを使用してデバイスを負荷分散する場合は、この設定を参照できます。共有 VLAN インターフェイスを使用したこの設定は、ユースケース 2 に似ています。基本的な設定については、シナリオ 2 を参照してください。



共有オプションが有効になっている状態で **VLAN A** をバインド

コマンドプロンプトで、次のように入力します。

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 100 -ifnum 1/2 tagged
```

共有オプションが有効になっている状態で **VLAN B** をバインド

コマンドプロンプトで、次のように入力します。

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 200 -ifnum 1/3 tagged
```

共有オプションを有効にして **VLAN C** をバインド

コマンドプロンプトで、次のように入力します。

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 300 -ifnum 1/2 tagged
```


共有オプションが有効になっている状態で **VLAN D** をバインド

コマンドプロンプトで、次のように入力します。

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 400 -ifnum 1/3 tagged
```

サービス **1** のコンテンツ検査プロファイル **1** の追加

Citrix ADC アプライアンスのインライン構成は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルには、デバイス設定のコレクションがあります。コンテンツ検査プロファイルはインラインサービス 1 用に作成され、通信は 1/2 そして 1/3 専用インターフェース。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3” -egressVlan 100 -ingressVlan
 300
```

サービス **2** のコンテンツ検査プロファイル **2** の追加

コンテンツインスペクションプロファイル 2 が service2 に追加され、インラインデバイスは 1/2 および 1/3 専用インターフェイスを介してアプライアンスと通信します。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

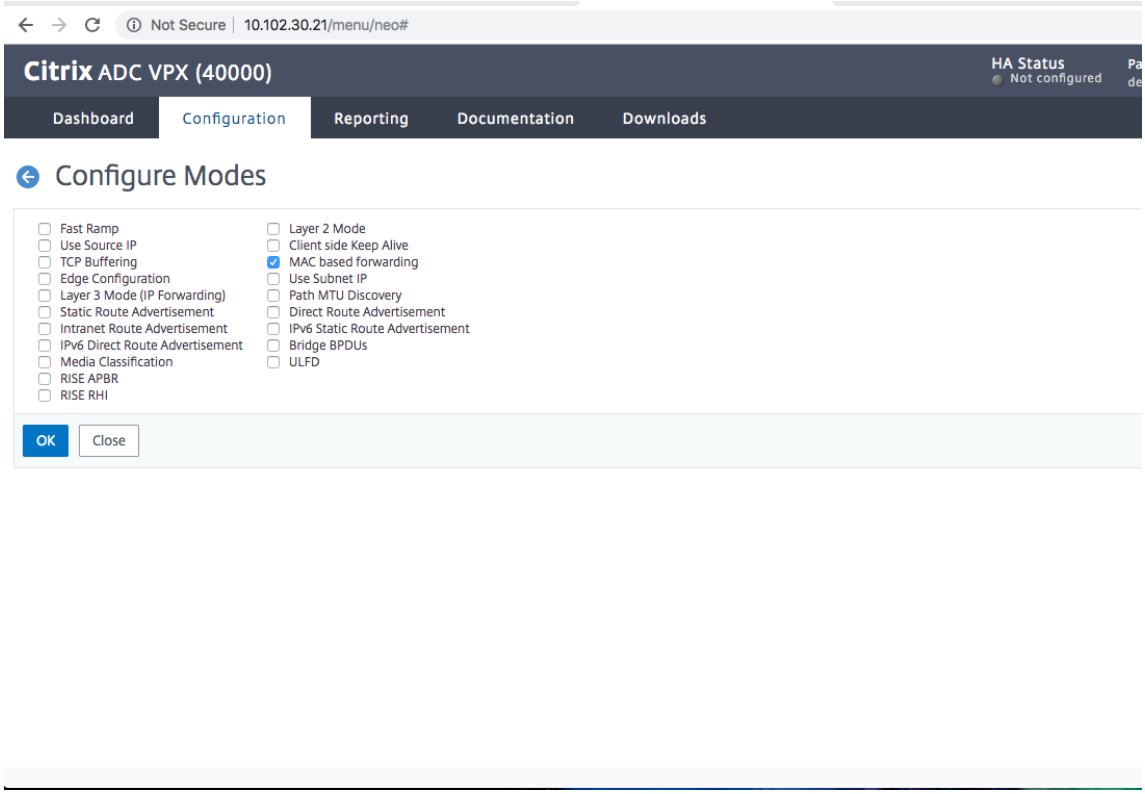
例:

```
add contentInspection profile Inline_profile2 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3” -egressVlan 200 -ingressVlan
 400
```

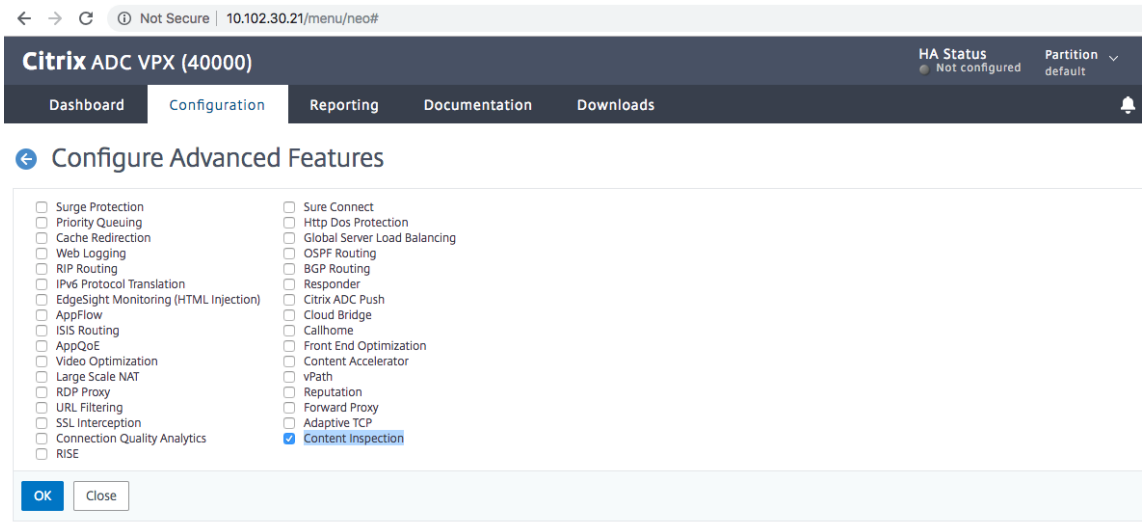
Citrix ADC GUI を使用したインラインサービス統合の構成

1. Citrix ADC アプライアンスにログオンし、［構成］タブページに移動します。

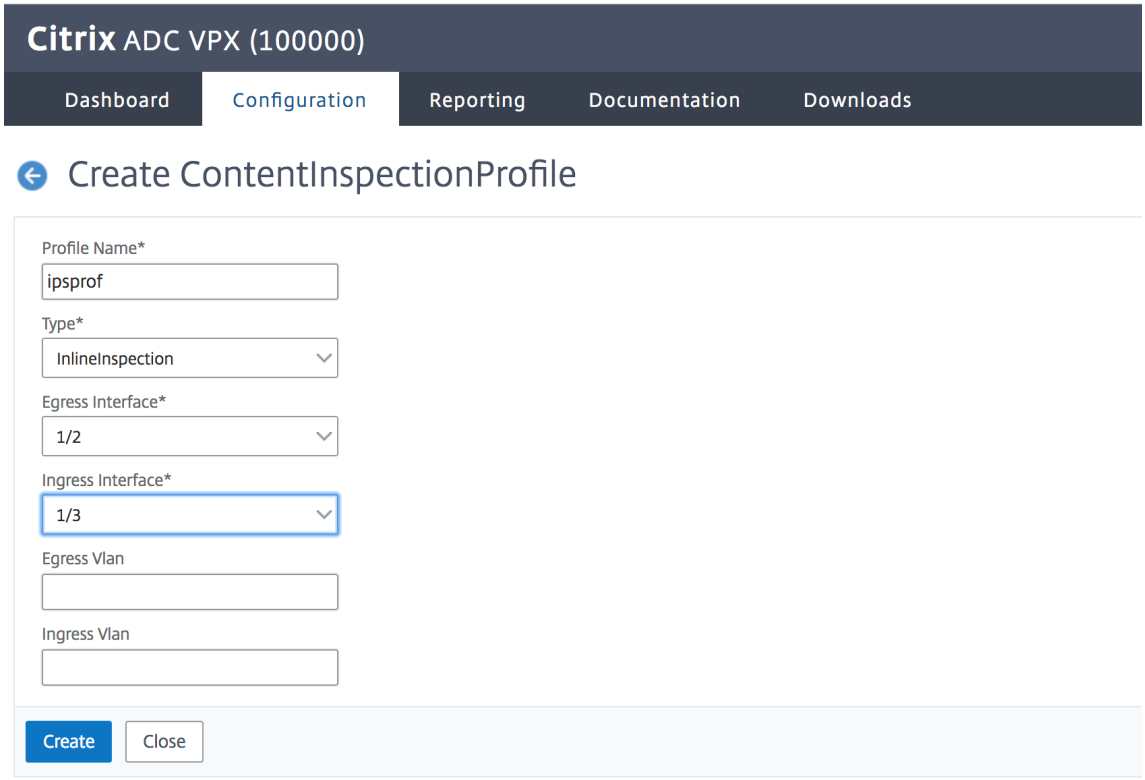
2. [システム] > **[**設定**] > [**モードの設定**]** に移動します。
3. [モードの設定] ページで、[**Mac** ベースの転送] を選択します。
4. [**OK]** をクリックして [閉じる] をクリックします。



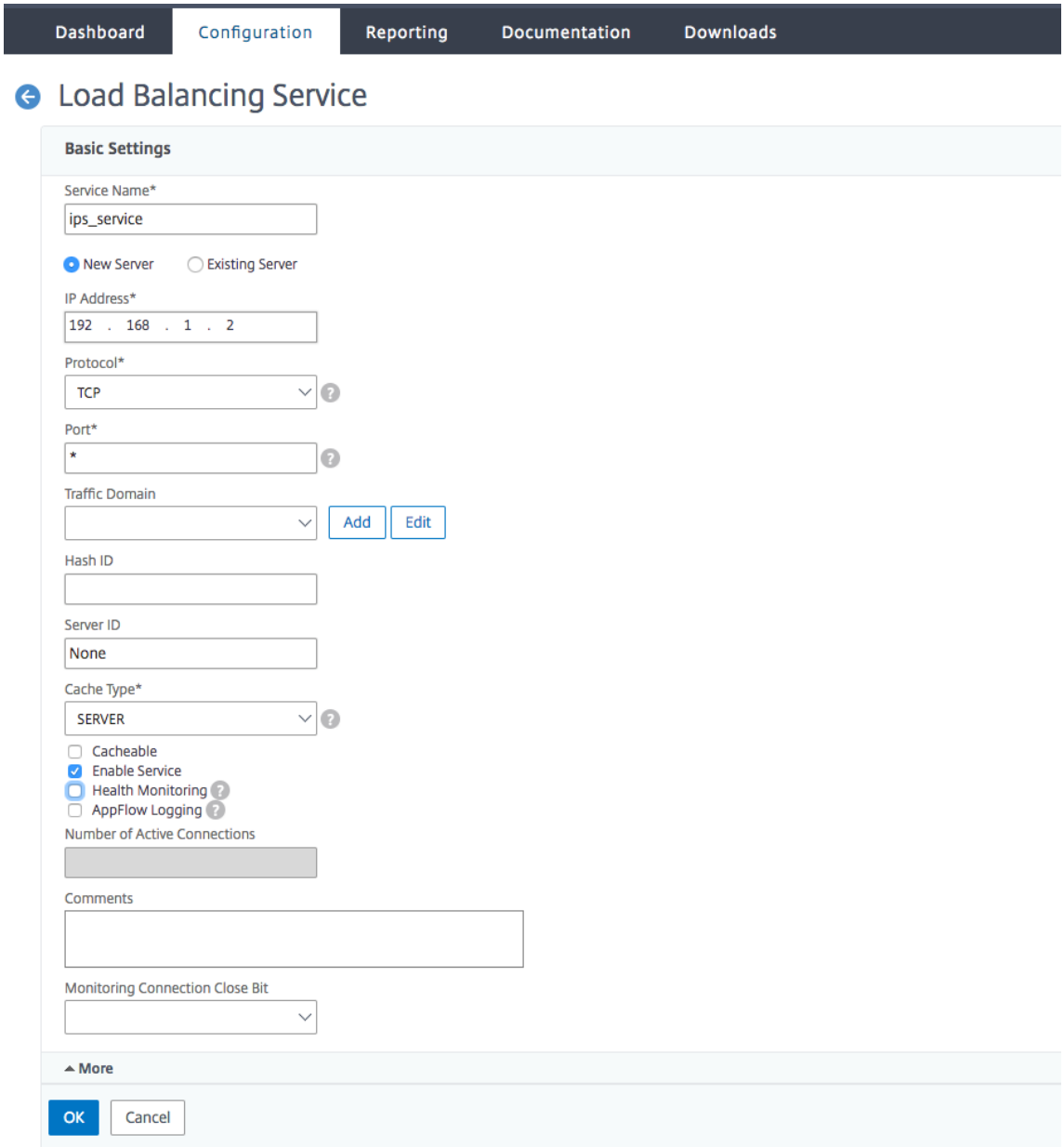
5. [システム] > **[**設定**] > [**拡張機能の設定] に移動します。
6. [拡張機能の設定] ページで、[コンテンツ検査] を選択します。
7. [**OK]** をクリックして [閉じる] をクリックします。



8. [セキュリティ] > [コンテンツ検査] > [コンテンツ検査プロファイル] に移動します。
9. 「コンテンツ検査プロファイル」ページで、「追加」をクリックします。
10. 「コンテンツ検査プロファイルの作成」ページで、次のパラメータを設定します。
    a) プロファイル名。コンテンツ検査プロファイルの名前。
    b) タイプ。プロファイルタイプを [インライン検査] として選択します。
    c) 出力インターフェイス。アプライアンスが Citrix ADC からインラインデバイスにトラフィックを送信するためのインターフェイス。
    d) 入力インターフェイス。アプライアンスがインラインデバイスから Citrix ADC へのトラフィックを受信するためのインターフェイス。
    e) 出力 VLAN。トラフィックがインラインデバイスに送信されるインターフェイス VLAN ID。
    f) 入力 VLAN。アプライアンスがインラインから Citrix ADC へのトラフィックを受信するインターフェイス VLAN ID（設定されている場合）。



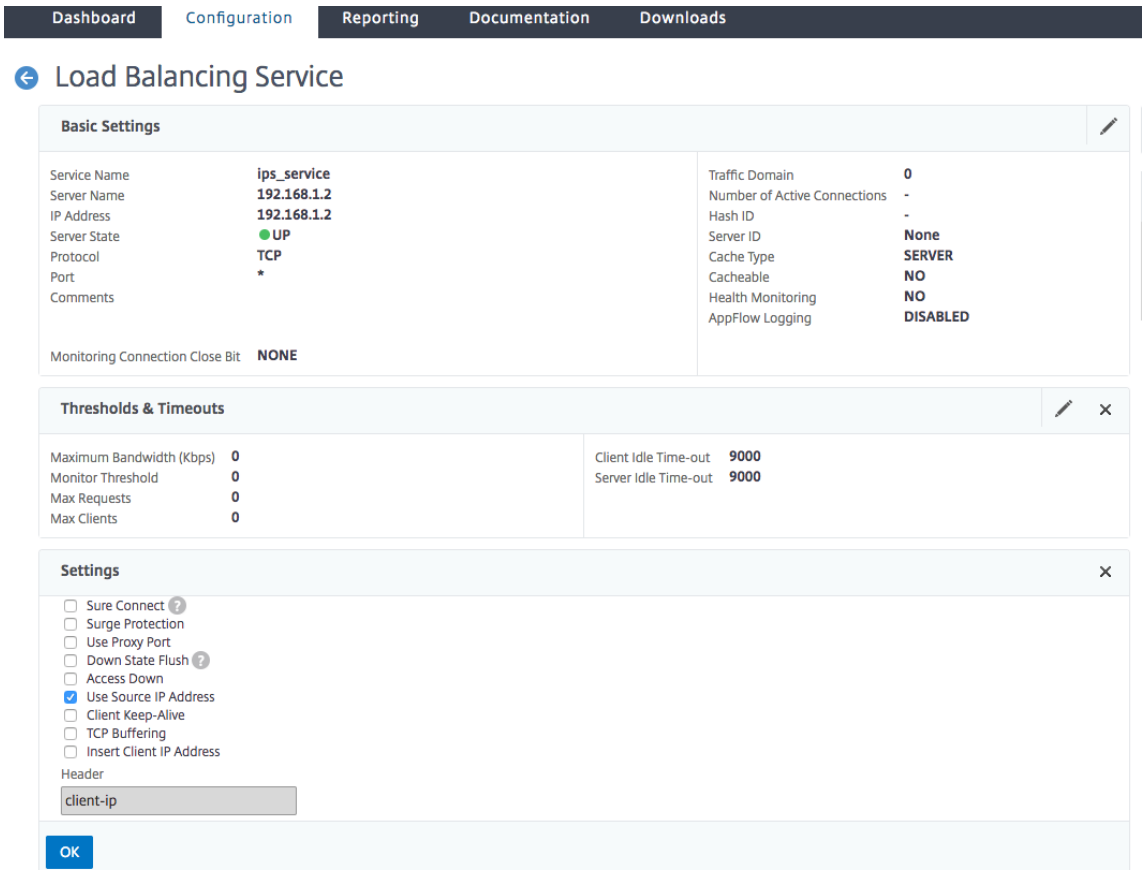
11. **[作成]** して **[閉じる]** をクリックします。
12. [トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックします。
13. [サービス] ページで、次のパラメータを設定します。
    a) サービス名。負荷分散サービスの名前。
    b) IP アドレス。ダミーの IP アドレスを使用します。注：どのデバイスも IP アドレスを所有してはなりません。
    c) プロトコル。プロトコルの種類を TCP として選択します。
    d) ポート。* と入力します。
    e) ヘルスモニタリング。このオプションをクリアし、サービスを TCP タイプモニターにバインドする場合にのみ有効にします。モニターをサービスにバインドする場合は、モニターの TRANSPARENT オプションをオンにする必要があります。モニターを追加する方法とサービスにバインドする方法については、手順 14 を参照してください。
    f) [**OK**] をクリックします。



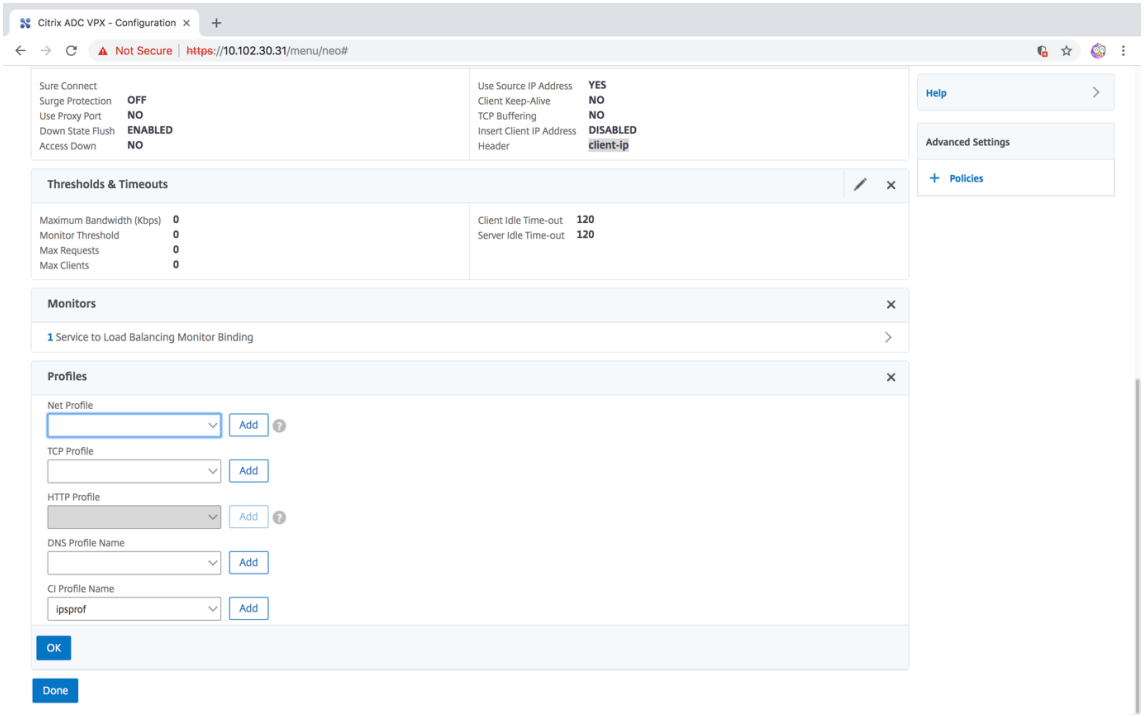
14. [設定] セクションで、次の項目を編集し、[**OK**] をクリックします。

    a) プロキシポートを使用: オフ

    b) 送信元 IP アドレスを使用: オン



15. [詳細設定] セクションで、[プロファイル] をクリックします。
16. [プロファイル] セクションに移動し、インラインコンテンツ検査プロファイルを追加して [**OK**] をクリックします。

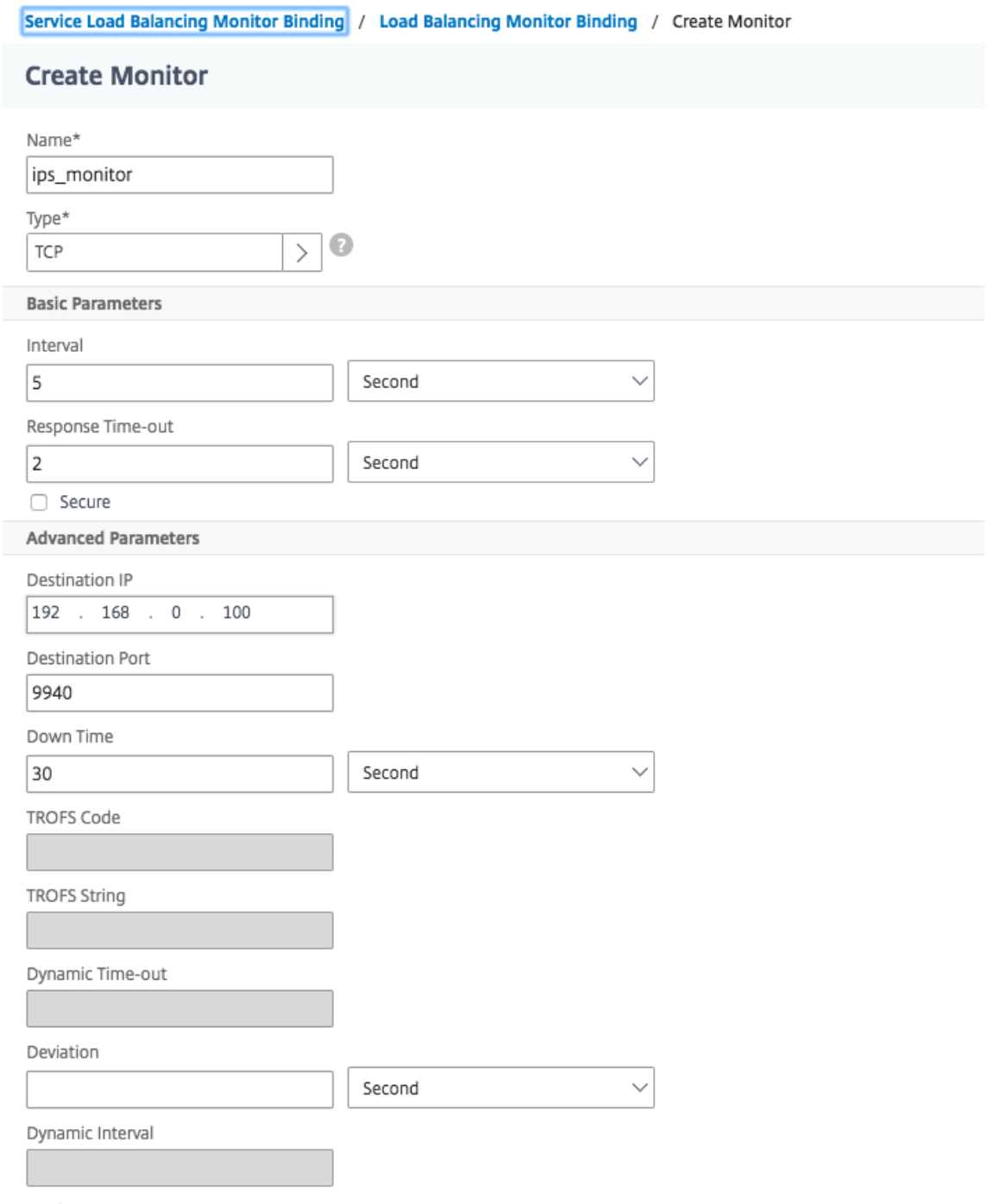


17. [モニター] セクション、**[**バインドの追加**]** > **[**モニター**]** > [追加] の順に選択します。

    a) 名前: モニタの名前
    b) タイプ:TCP タイプの選択
    c) 宛先 IP、ポート：宛先 IP アドレスおよびポート。
    d) 透明: オン

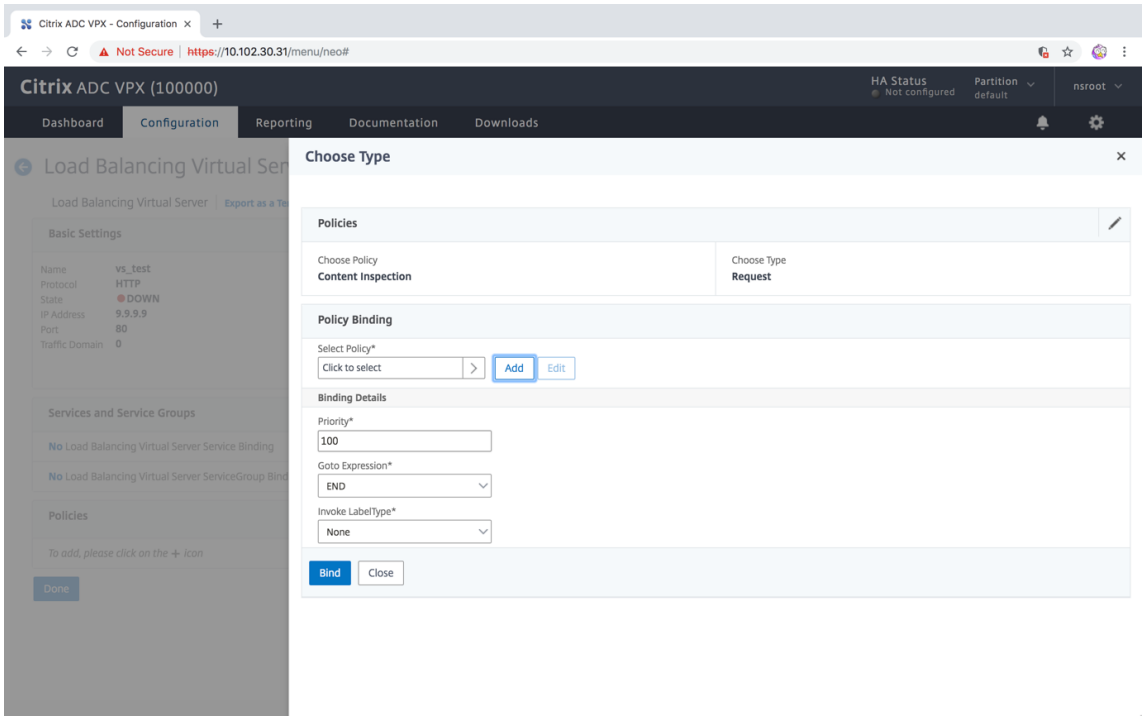
    注意: インラインデバイスのステータスを監視するには、監視パケットがインラインデバイスを通過する必要があります。

18. ［作成］をクリックします。

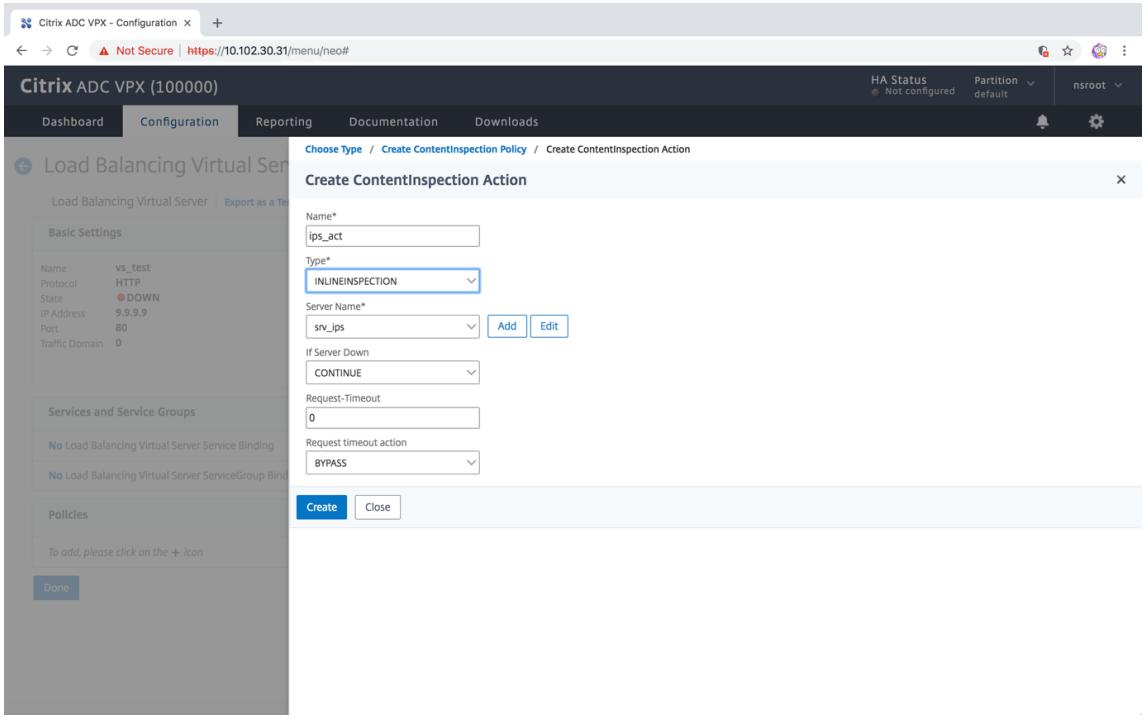


19. [完了] をクリックします。
20. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。HTTP または SSL タイプの仮想サーバーを追加します。
21. サーバの詳細を入力したら、[**OK**] をクリックし、もう一度 [**OK**] をクリックします。
22. [負荷分散仮想サーバー] の **[**トラフィック設定**]** セクションで、[レイヤ 2 パラメータ] をオンにします。
23. [詳細設定**]** セクションで、[ポリシー] をクリックします。

24. [ポリシー] セクションに移動し、[“+” アイコンを使用して、コンテンツ検査ポリシーを構成します。
25. [ポリシーの選択] ページで、[コンテンツ検査] を選択します。[続行] をクリックします。
26. [ポリシーのバインド] セクションで、[追加] をクリックしてコンテンツ検査ポリシーを追加します。



27. [コンテンツ検査ポリシーの作成] ページで、インラインコンテンツ検査ポリシーの名前を入力します。
28. [アクション] フィールドで、[追加] をクリックして、インラインコンテンツインスペクションアクションを作成します。
29. [**CI** アクションの作成] ページで、次のパラメータを設定します。
    a) Name：コンテンツ検査インラインポリシーの名前。
    b) タイプ。タイプを「インライン検査」として選択します。
    c) サーバー。サーバー/サービスをインラインデバイスとして選択します。
    d) サーバがダウンしている場合。サーバーがダウンした場合の操作を選択します。
    e) 要求タイムアウト。タイムアウト値を選択します。デフォルト値を使用できます。
    f) 要求タイムアウトアクション。タイムアウト処理を選択します。デフォルト値を使用できます。
30. [作成] をクリックします。



31. ［作成］をクリックします。
32. [**CI** ポリシーの作成] ページで、その他の詳細を入力します。
33. [**OK**] をクリックして [閉じる] をクリックします。

SSL フォワードプロキシを使用したインラインデバイスとしての **IPS** または **NGFW** との統合

October 7, 2021

侵入防止システム（IPS）や次世代ファイアウォール（NGFW）などのセキュリティデバイスは、ネットワーク攻撃からサーバーを保護します。これらのデバイスはライブトラフィックを検査でき、通常はレイヤ 2 インラインモードで展開されます。SSL 転送プロキシアプライアンスは、インターネット上のリソースにアクセスするときに、ユーザーと企業ネットワークのセキュリティを提供します。

SSL フォワードプロキシアプライアンスは、1 つ以上のインラインデバイスと統合して、脅威を防ぎ、高度なセキュリティ保護を提供できます。インラインデバイスには、IPS や NGFW などの任意のセキュリティデバイスを使用できます。

SSL フォワードプロキシアプライアンスおよびインラインデバイス統合を使用してメリットが得られるユースケースには、次のようなものがあります。

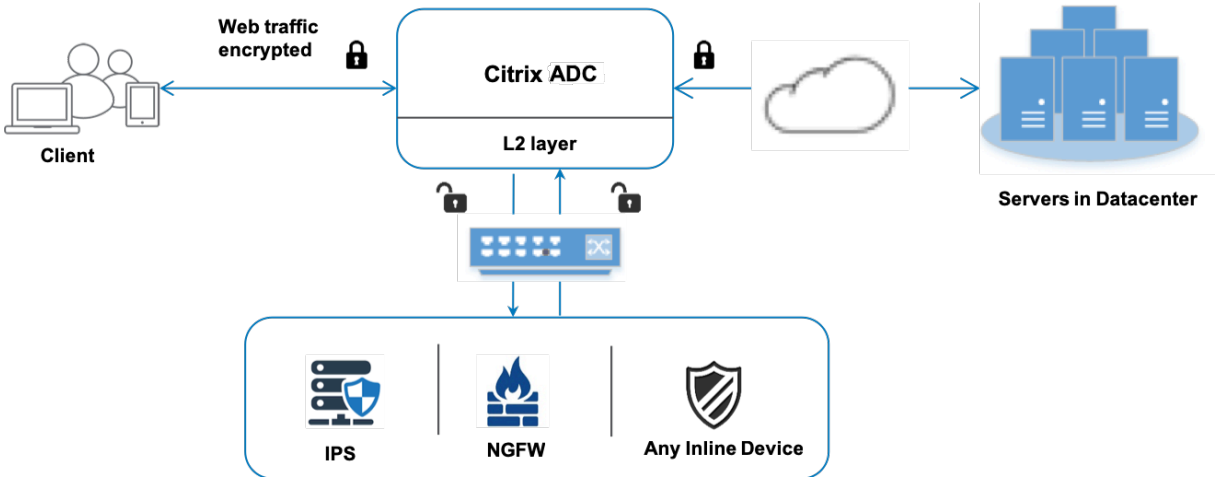
- 暗号化されたトラフィックの検査**:** ほとんどの IPS および NGFW アプライアンスは暗号化されたトラフィックをバイパスするため、サーバーが攻撃に対して脆弱になる可能性があります。SSL 転送プロキシアプライア

ンスは、トラフィックを復号化し、検査のためにインラインデバイスに送信できます。この統合により、お客様のネットワークセキュリティが強化されます。

- **TLS/SSL** 処理からのインラインデバイスのオフロード：TLS/SSL 処理はコストがかかり、IPS または NGFW アプライアンスがトラフィックを復号化すると CPU 使用率が高くなる可能性があります。SSL フォワードプロキシアプライアンスは、インラインデバイスからの TLS/SSL 処理のオフロードに役立ちます。その結果、インラインデバイスは大量のトラフィックを検査できます。
- インラインデバイスの負荷分散：大量のトラフィックを管理するように複数のインラインデバイスを設定している場合、SSL 転送プロキシアプライアンスは負荷分散を行い、これらのデバイスにトラフィックを均等に分散できます。
- トラフィックのスマート選択：アプライアンスは、検査のためにすべてのトラフィックをインラインデバイスに送信する代わりに、トラフィックのスマートな選択を行います。たとえば、インラインデバイスへの検査用のテキストファイルの送信はスキップされます。

インラインデバイスと **SSL** フォワードプロキシの統合

次の図は、SSL フォワードプロキシとインラインセキュリティデバイスとの統合を示しています。



インラインデバイスを SSL フォワードプロキシアプライアンスと統合すると、コンポーネントは次のように相互作用します。

1. クライアントが SSL 転送プロキシアプライアンスに要求を送信します。
2. アプライアンスは、ポリシー評価に基づいてコンテンツ検査のためにデータをインラインデバイスに送信します。HTTPS トラフィックの場合、アプライアンスはデータを復号化し、コンテンツ検査のためにプレーンテキストでインラインデバイスに送信します。

> 注
>
> 2 つ以上のインラインデバイスがある場合、アプライアンスはデバイスの負荷分散を行い、トラフィックを送信します。

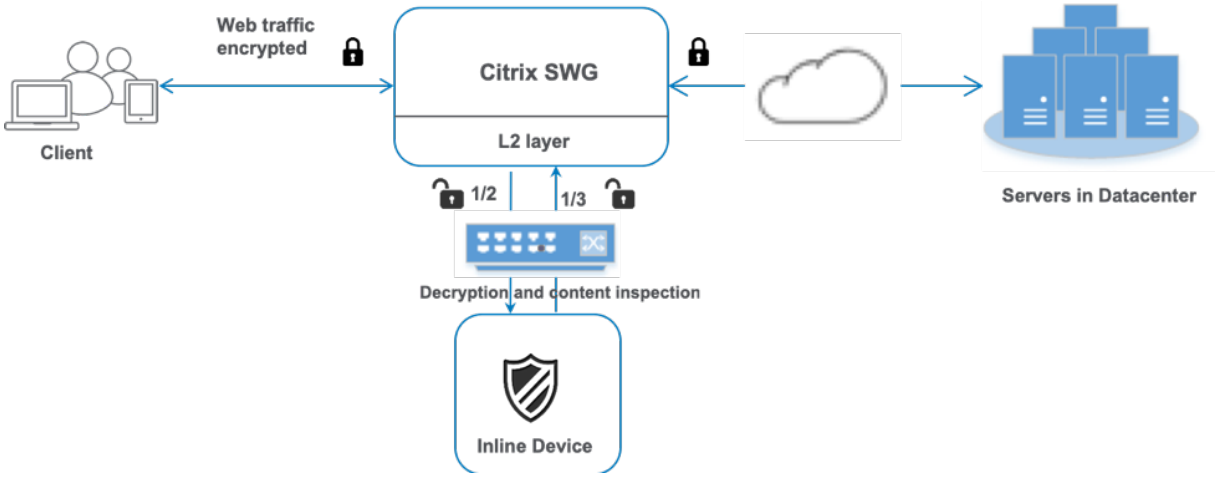
3. コンテンツスイッチングまたは HTTP/HTTPS 負荷分散仮想サーバーを追加します。
4. インラインデバイスは、データの脅威を検査し、データをドロップ、リセット、またはアプライアンスに戻すかどうかを決定します。
5. セキュリティ上の脅威がある場合、デバイスはデータを修正してアプライアンスに送信します。
6. HTTPS トラフィックの場合、アプライアンスはデータを再暗号化し、要求をバックエンドサーバーに転送します。
7. バックエンドサーバーは、アプライアンスに応答を送信します。
8. アプライアンスは再びデータを復号化し、検査のためにインラインデバイスに送信します。
9. インラインデバイスがデータを検査します。セキュリティ上の脅威がある場合、デバイスはデータを修正してアプライアンスに送信します。
10. アプライアンスはデータを再暗号化し、応答をクライアントに送信します。

インラインデバイス統合の設定

インラインデバイスを使用して SSL フォワードプロキシアプライアンスを構成するには、次の 3 つの方法があります。

シナリオ **1**: 単一のインラインデバイスを使用する

セキュリティデバイス（IPS または NGFW）をインラインモードで統合するには、SSL 転送プロキシアプライアンスでグローバルモードでコンテンツ検査と MAC ベース転送（MBF）を有効にする必要があります。次に、コンテンツインスペクションプロファイル、TCP サービス、インラインデバイスのコンテンツインスペクションアクションを追加して、インスペクションに基づいてトラフィックをリセット、ブロック、またはドロップします。また、インラインデバイスに送信するトラフィックのサブセットを決定するためにアプライアンスが使用するコンテンツ検査ポリシーを追加します。最後に、サーバ上でレイヤ 2 接続を有効にしてプロキシ仮想サーバを設定し、コンテンツ検査ポリシーをこのプロキシ仮想サーバにバインドします。



次の手順を実行します。

1. MAC ベース転送（MPF）モードを有効にします。
2. コンテンツ検査機能を有効にします。
3. サービスのコンテンツ検査プロファイルを追加します。コンテンツ検査プロファイルには、SSL 転送プロキシアプライアンスとインラインデバイスを統合するインラインデバイス設定が含まれています。
4. （任意）TCP モニタを追加します。

   > 注:
   >
   > トランスペアレントデバイスには IP アドレスがありません。したがって、ヘルスチェックを実行するには、モニターを明示的にバインドする必要があります。

5. サービスを追加します。サービスは、インラインデバイスを表します。
6. （任意）サービスを TCP モニタにバインドします。
7. サービスのコンテンツインスペクションアクションを追加します。
8. コンテンツ検査ポリシーを追加し、アクションを指定します。
9. HTTP または HTTPS プロキシ (コンテンツスイッチング) 仮想サーバーを追加します。
10. コンテンツ検査ポリシーを仮想サーバにバインドします。

CLI を使用して構成する

コマンドプロンプトで次のコマンドを入力します。例はほとんどのコマンドの後に示されています。

1. MBF を有効にします。

```
enable ns mode mbf
```

1. 本機能を有効にします。

```
enable ns feature contentInspection
```

1. コンテンツ検査プロファイルを追加します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例**:**

```
add contentInspection profile ipsprof -type InlineInspection -ingressinterface
 “1/2” -egressInterface “1/3”
```

1. サービスを追加します。インラインデバイスを含むどのデバイスにも所有されていないダミー IP アドレスを指定します。`use source IP address` (USIP) を YES に設定します。`useproxyport`を NO に設定

します。デフォルトでは、ヘルスモニタリングは ON で、サービスをヘルスモニタにバインドし、モニタの [トランスペアレント] オプションも設定します。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor YES -usip YES -useproxyport NO
```

例:

```
add service ips_service 198.51.100.2 TCP * -healthMonitor YES -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof
```

1. ヘルスモニタを追加します。デフォルトでは、ヘルスモニターはオンになっており、必要に応じて無効にするオプションもあります。コマンドプロンプトで入力します。

   ```
   add lb monitor <name> TCP -destIP <ip address> -destPort 80 -transparent
    <YES, NO>
   ```

例:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

1. サービスをヘルスモニターにバインドする

ヘルスモニターを構成したら、サービスをヘルスモニターにバインドする必要があります。コマンドプロンプトで入力します。

```
bind service <name> -monitorName <name>
```

例:

```
bind service ips_svc -monitorName ips_tcp
```

1. コンテンツインスペクションアクションを追加します。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

例:

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
ips_service
```

1. コンテンツ検査ポリシーを追加します。

```
add contentInspection policy <name> -rule <expression> -action <string>
```

例:

```
add contentInspection policy ips_pol -rule ”HTTP.REQ.METHOD.NE(\”CONNECT\”)
”-action ips_action
```

1. プロキシ仮想サーバーを追加します。


```
add cs vserver <name> PROXY <IPAddress> <port> -cltTimeout <secs> -Listenpolicy
 <expression> -authn401 ( ON | OFF )-authnVsName <string> -l2Conn ON
```

> 注:
>
> HTTP/SSL タイプの負荷分散仮想サーバーもサポートされています。

例:

```
add cs vserver transparentcs PROXY * * -cltTimeout 180 -Listenpolicy exp1 -
authn401 on -authnVsName swg-auth-vs-trans-http -l2Conn ON
```

1. ポリシーを仮想サーバにバインドします。

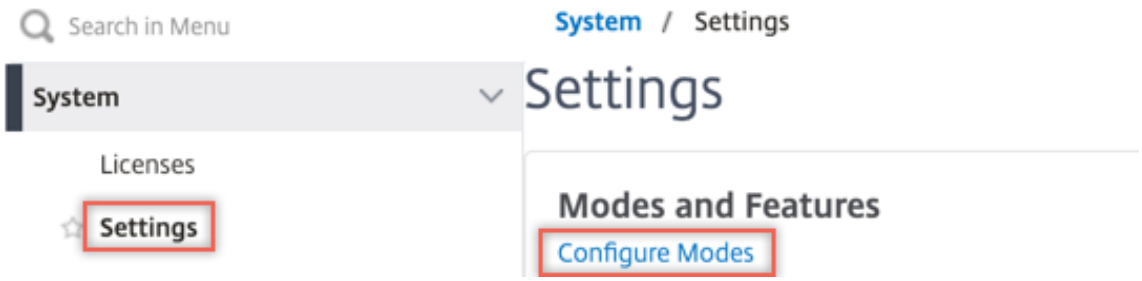
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

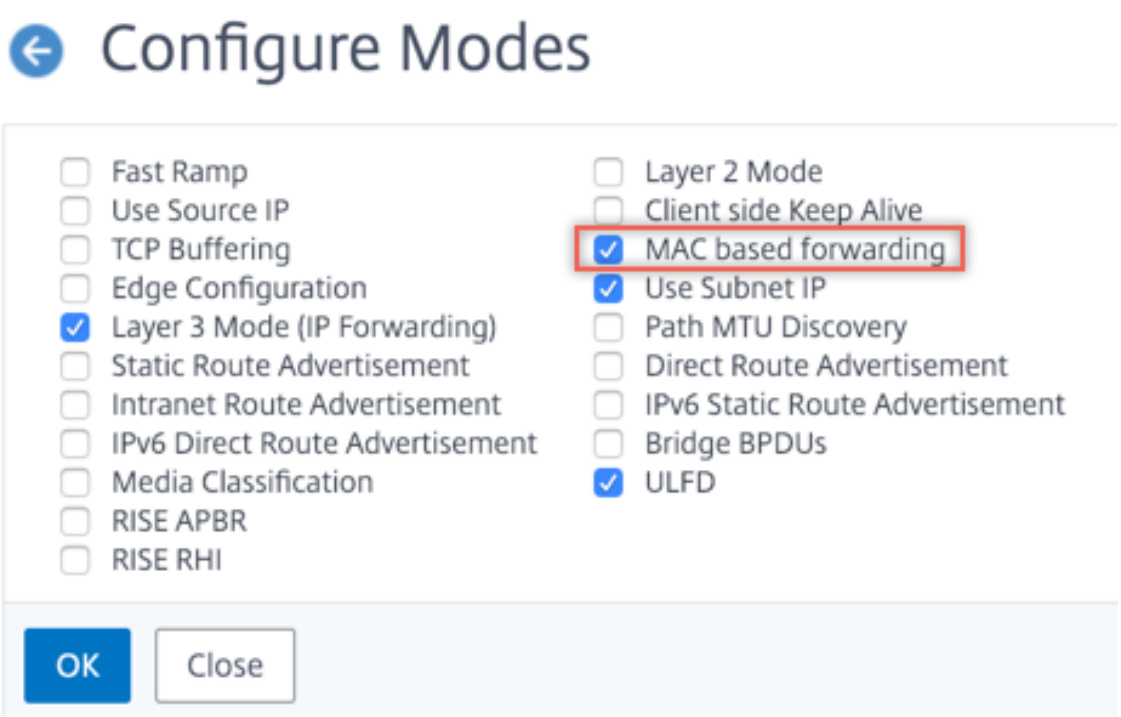
例:

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

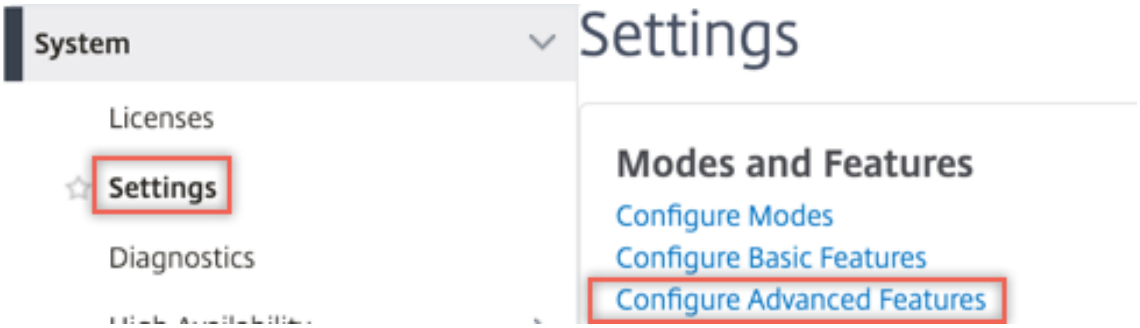
GUI を使用して構成する

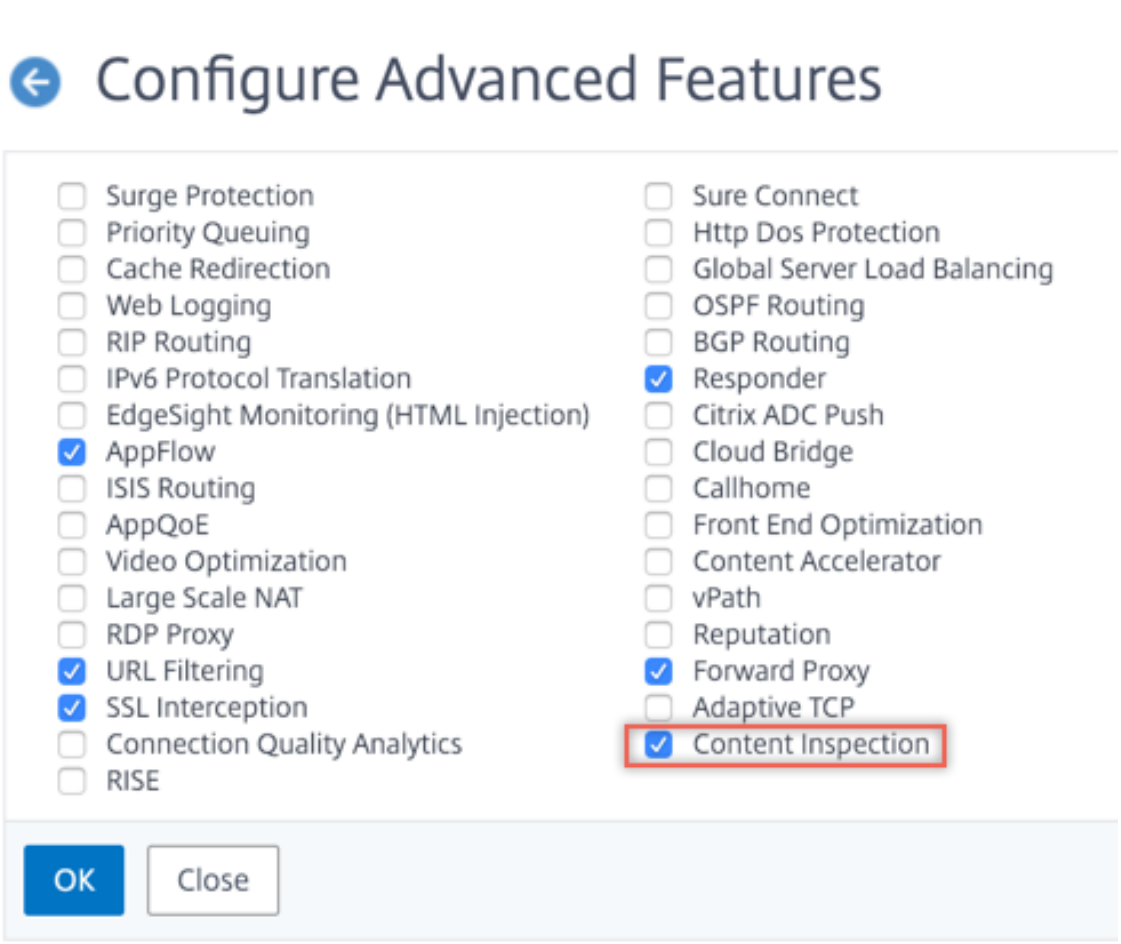
1. **System > Settings** に移動します。[モードと機能] で、[モードの構成] をクリックします。



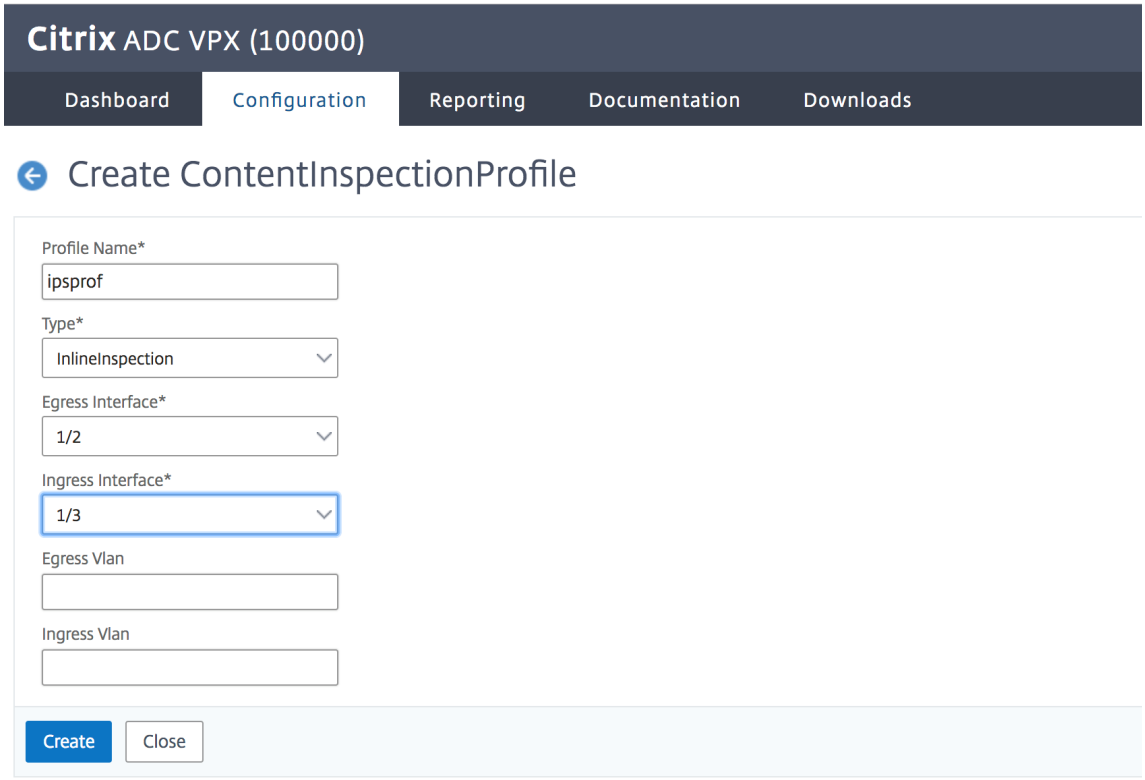


2. **System > Settings** に移動します。[モードと機能] で、[高度な機能の構成] をクリックします。

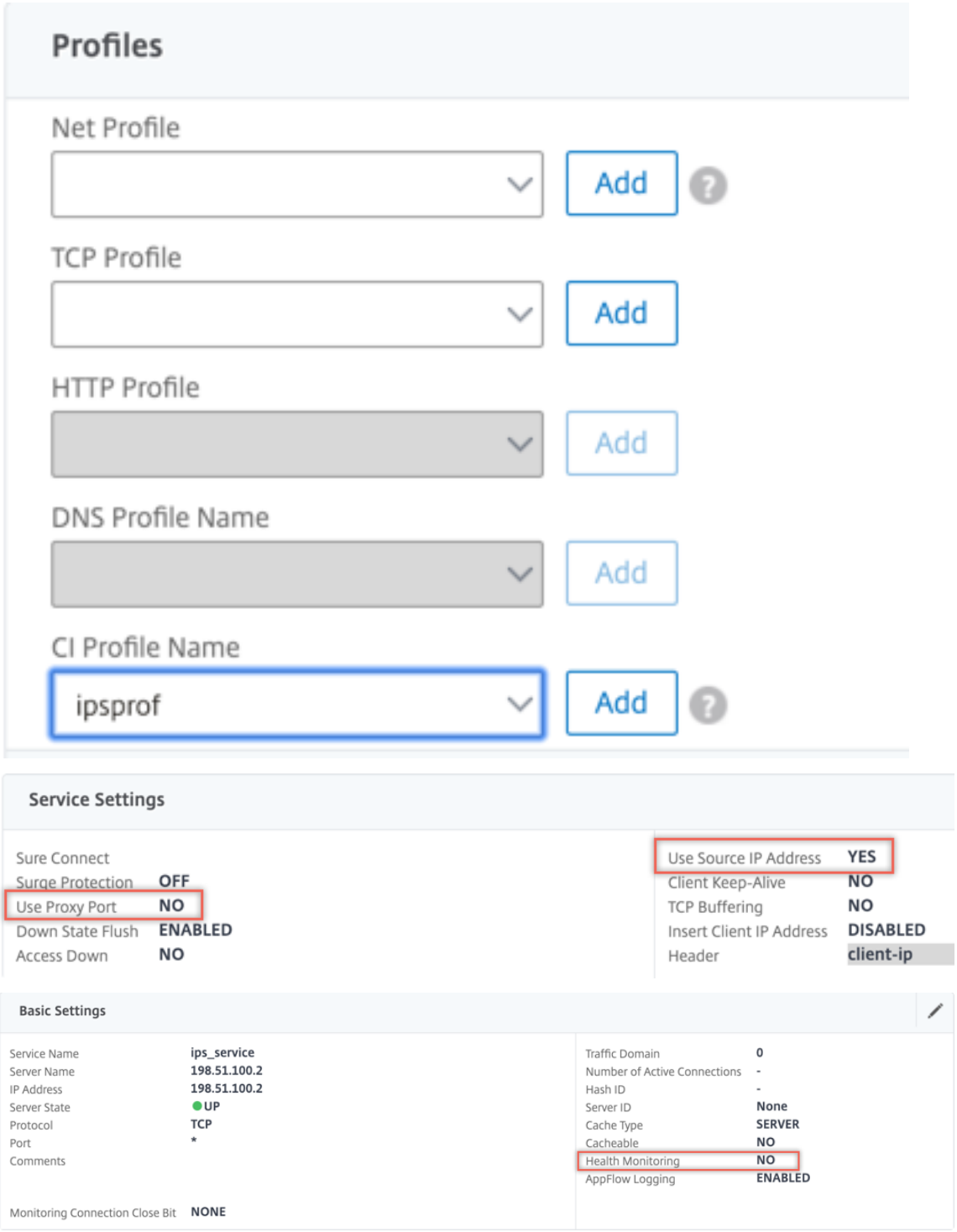




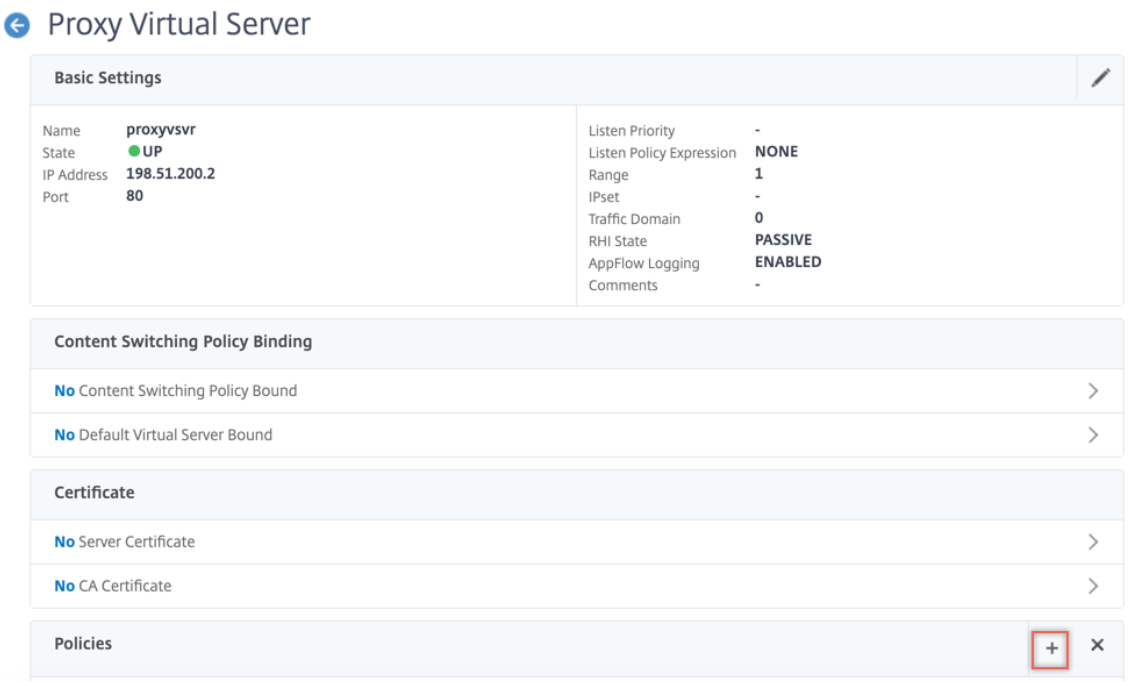
3. [**Secure Web Gateway] > [**コンテンツ検査**] > [**コンテンツ検査プロファイル] に移動します。[追加] をクリックします。



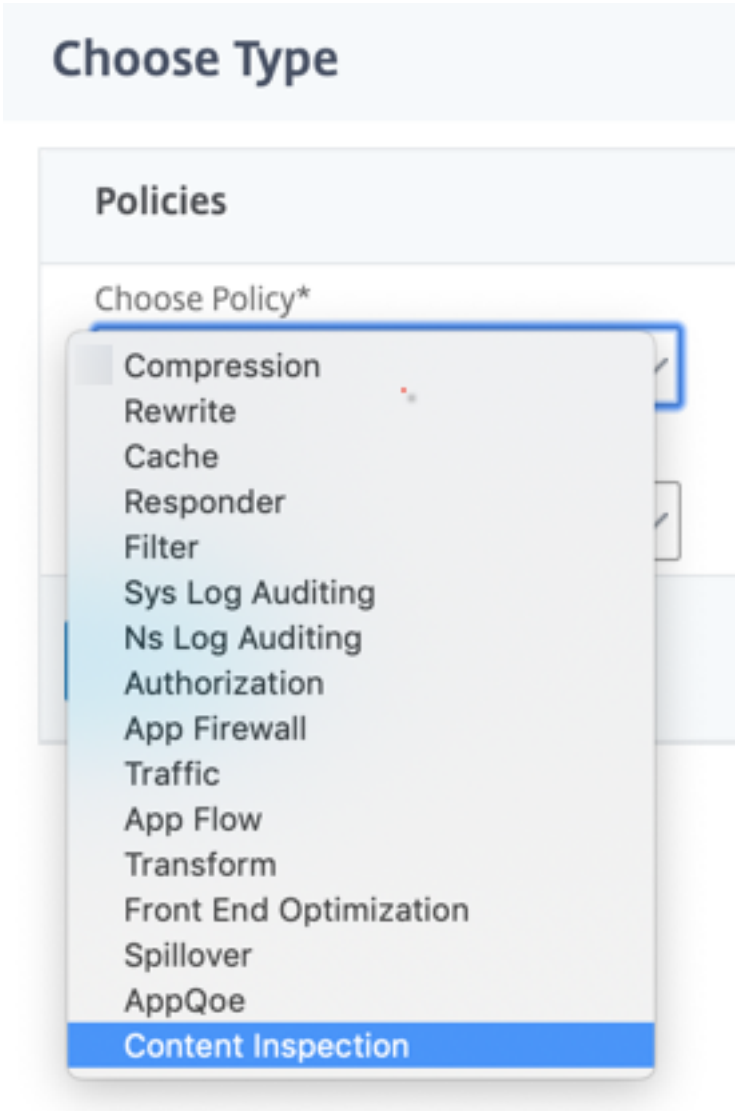
4. [負荷分散] **>** [サービス] **>** [サービスの追加と追加] に移動します。[詳細設定] で、[プロファイル] をクリックします。[**CI** プロファイル名] リストで、以前に作成したコンテンツ検査プロファイルを選択します。[サービス設定] で、[ソース **IP** アドレスを使用] を [はい]、[プロキシポートを使用] を [いいえ] に設定します。[基本設定] で、[ヘルスモニタリング] を [いいえ] に設定します。このサービスを TCP モニターにバインドする場合のみ、ヘルスモニタリングをオンにします。モニタをサービスにバインドする場合は、モニタの TRANSPARENT オプションを ON に設定します。



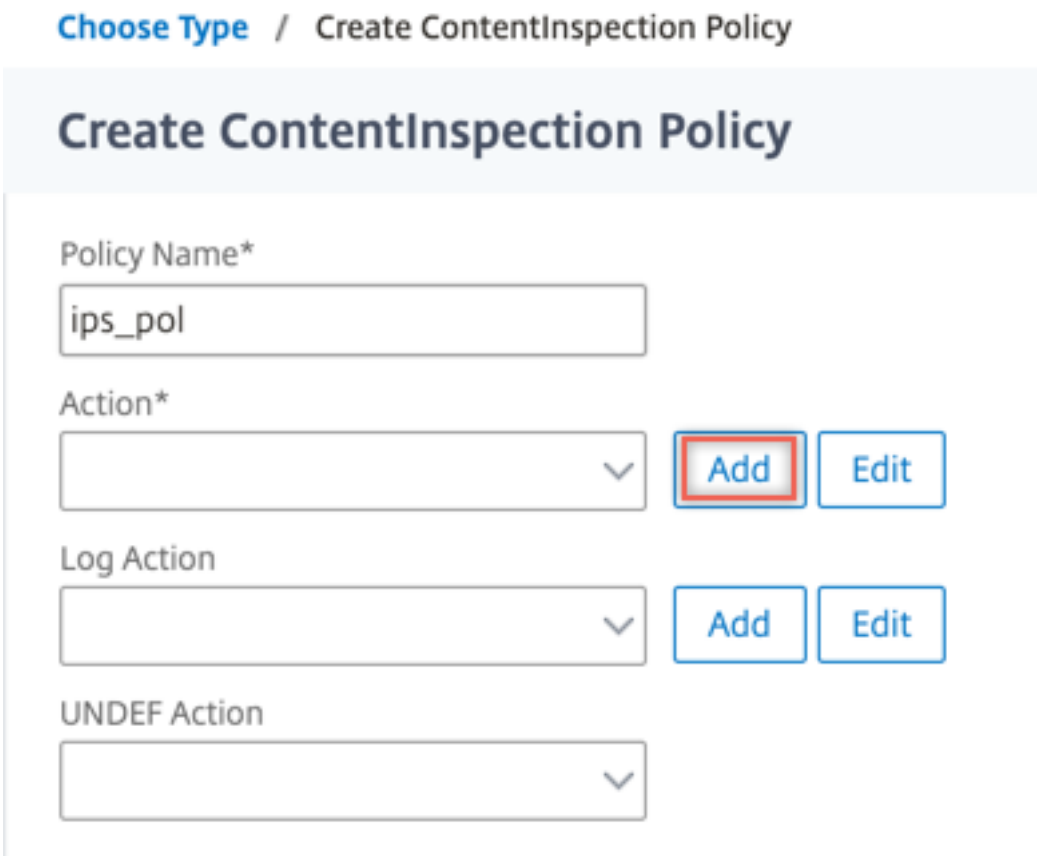
5. [**Secure Web Gateway**] > [プロキシ仮想サーバー] > [追加] に移動します。名前、IP アドレス、およびポートを指定します。[詳細設定] で、[ポリシー] を選択します。「+」記号をクリックします。



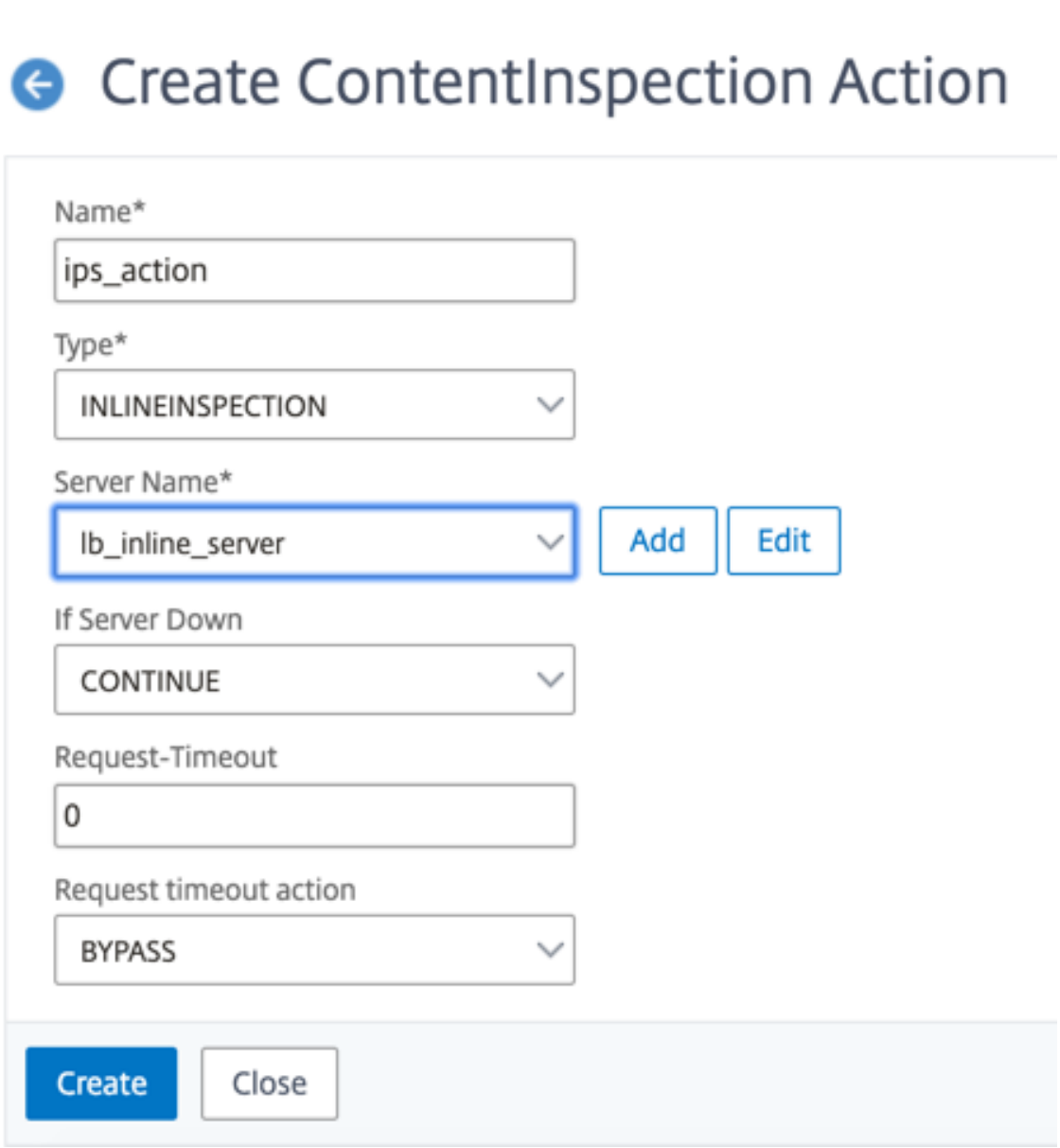
6. [ポリシーの選択] で [コンテンツ検査] を選択します。[続行] をクリックします。



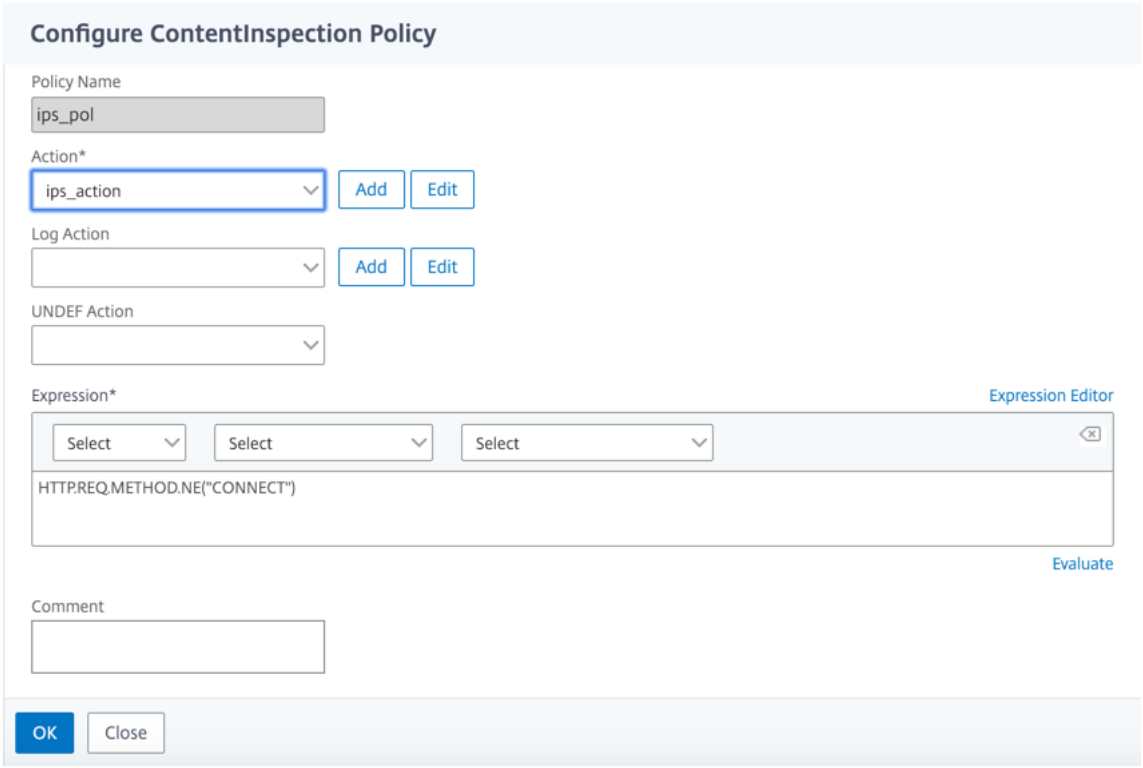
7. ［追加］をクリックします。名前を指定します。「アクション」で、「追加」をクリックします。



8. 名前を指定します。「タイプ」で「**INLINEINSPECTION**」を選択します。「サーバー名」で、以前に作成した TCP サービスを選択します。



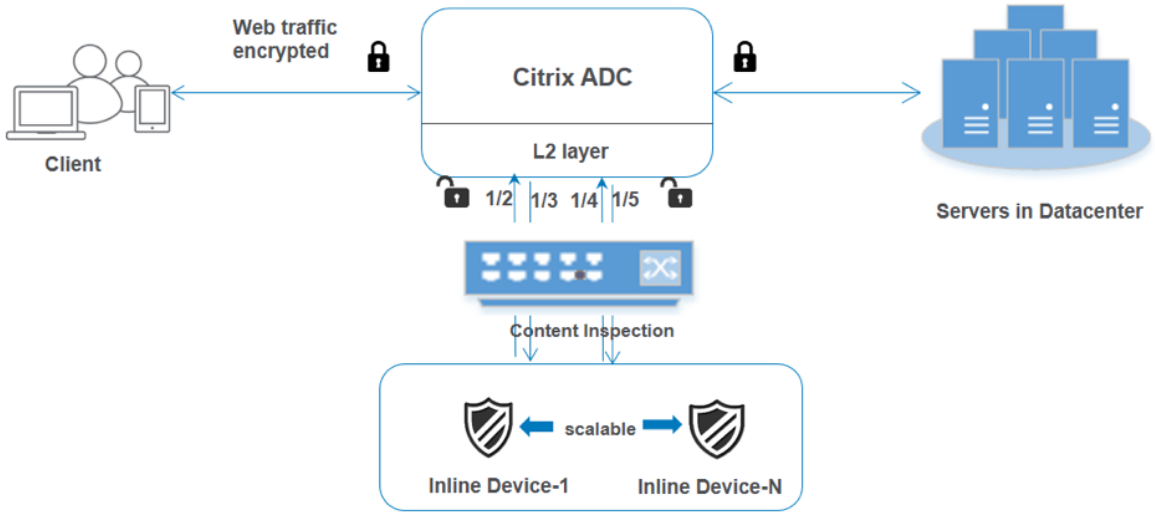
9. [作成] をクリックします。ルールを指定し、[**Create**] をクリックします。



10. ［バインド］をクリックします。
11. ［完了］をクリックします。

シナリオ **2**：専用インターフェイスを持つ複数のインラインデバイスの負荷分散

2 つ以上のインラインデバイスを使用している場合は、専用のインターフェイスで異なるコンテンツインスペクションサービスを使用して、デバイスを負荷分散できます。この場合、SSL 転送プロキシアプライアンスは、専用インターフェイスを介して各デバイスに送信されるトラフィックのサブセットを負荷分散します。サブセットは、設定されたポリシーに基づいて決定されます。たとえば、TXT ファイルやイメージファイルは、検査のためにインラインデバイスに送信されない場合があります。



基本設定は、シナリオ 1 と同じままです。ただし、インラインデバイスごとにコンテンツ検査プロファイルを作成し、各プロファイルで入力および出力インターフェイスを指定する必要があります。インラインデバイスごとにサービスを追加します。負荷分散仮想サーバを追加し、コンテンツインスペクションアクションで指定します。次の追加手順を実行します。

1. サービスごとにコンテンツ検査プロファイルを追加します。
2. デバイスごとにサービスを追加します。
3. 負荷分散仮想サーバーを追加します。
4. コンテンツインスペクションアクションで負荷分散仮想サーバを指定します。

CLI を使用して構成する

コマンドプロンプトで次のコマンドを入力します。各コマンドの後に例が示されています。

1. MBF を有効にします。

```
enable ns mode mbf
```

1. 本機能を有効にします。

```
enable ns feature contentInspection
```

1. サービス 1 のプロファイル 1 を追加します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例**:**

```
add contentInspection profile ipsprof1 -type InlineInspection -ingressInterface
 ”1/2”-egressInterface ”1/3”
```


1. サービス 2 のプロファイル 2 を追加します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof2 -type InlineInspection -ingressInterface
 "1/4"-egressInterface "1/5"
```

1. サービス 1 を追加します。インラインデバイスを含むどのデバイスにも所有されていないダミー IP アドレスを指定します。use source IP address (USIP) を YES に設定します。useproxyportを NO に設定します。TRANSPARENT オプションがオンに設定された TCP モニターでヘルスモニターをオンにします。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof1
```

1. サービス 2 を追加します。インラインデバイスを含むどのデバイスにも所有されていないダミー IP アドレスを指定します。use source IP address (USIP) を YES に設定します。useproxyportを NO に設定します。TRANSPARENT オプションをオンにしてヘルスモニタリングをオンにします。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof2
```

1. 負荷分散仮想サーバーを追加します。

```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

例:

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. サービスを負荷分散仮想サーバーにバインドします。

```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```

例:

```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```


1. コンテンツインスペクションアクションで負荷分散仮想サーバを指定します。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

例:

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
lb_inline_vserver
```

1. コンテンツ検査ポリシーを追加します。ポリシーでコンテンツインスペクションアクションを指定します。

```
add contentInspection policy <name> -rule <expression> -action <string>
```

例:

```
add contentInspection policy ips_pol -rule ”HTTP.REQ.METHOD.NE(\”CONNECT\”)
”-action ips_action
```

1. プロキシ仮想サーバーを追加します。

```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

例:

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. コンテンツ検査ポリシーを仮想サーバにバインドします。

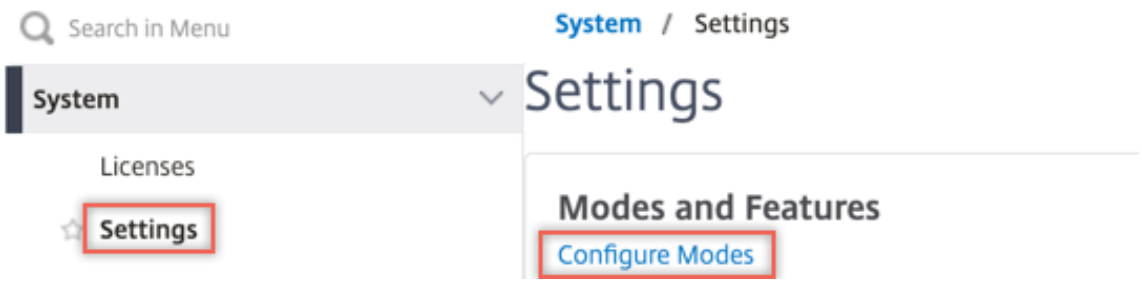
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

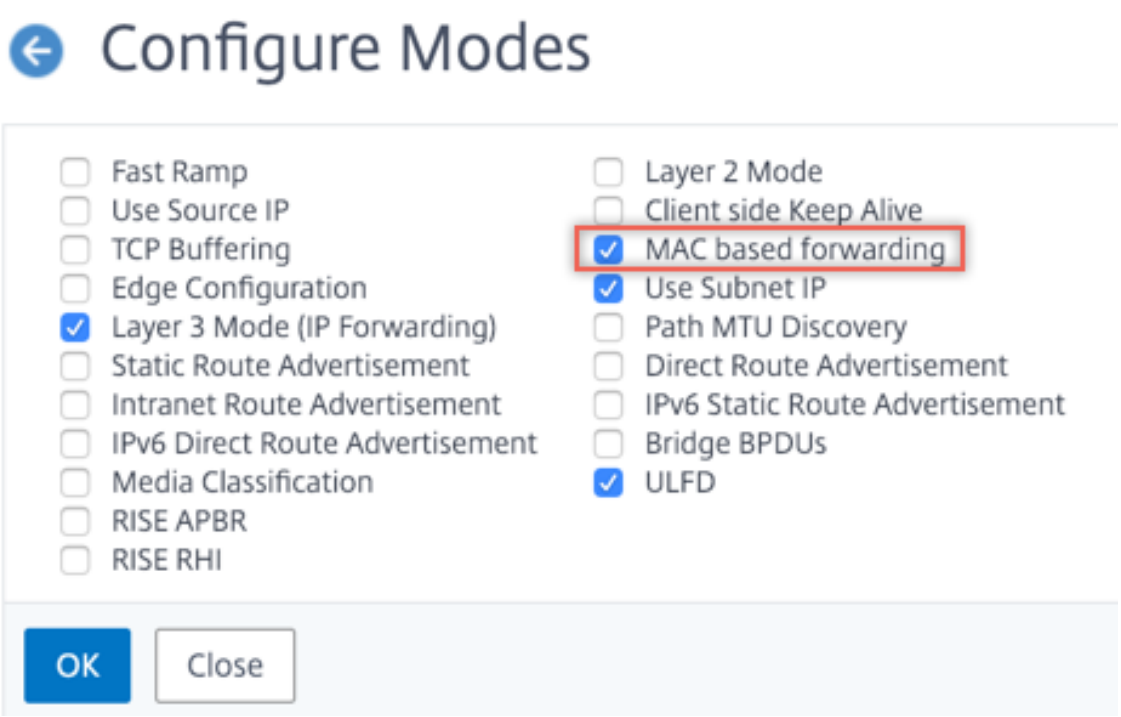
例:

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

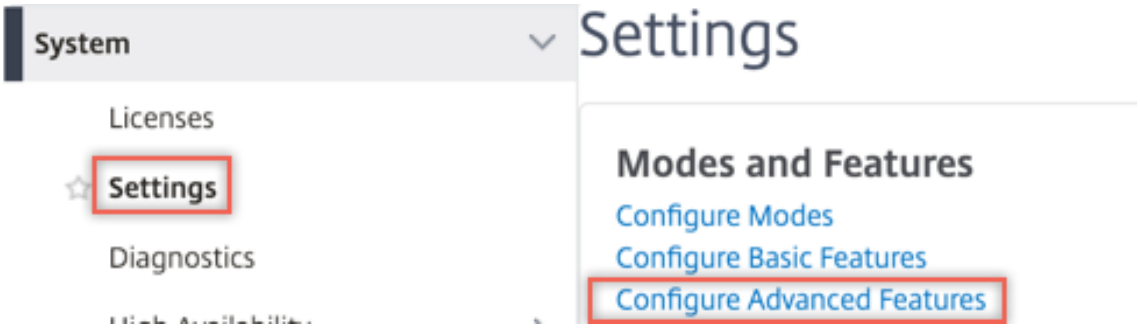
GUI を使用した設定

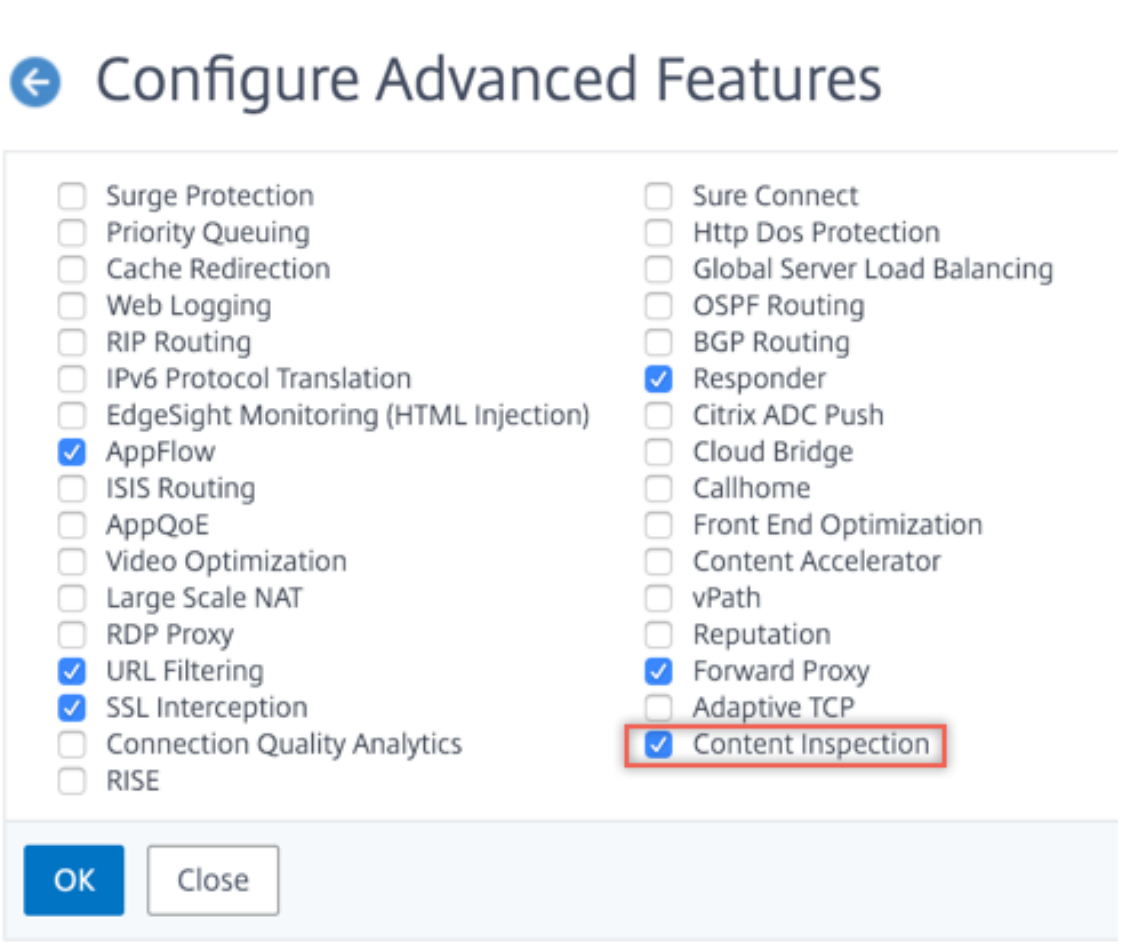
1. **System > Settings** に移動します。[モードと機能] で、[モードの構成] をクリックします。



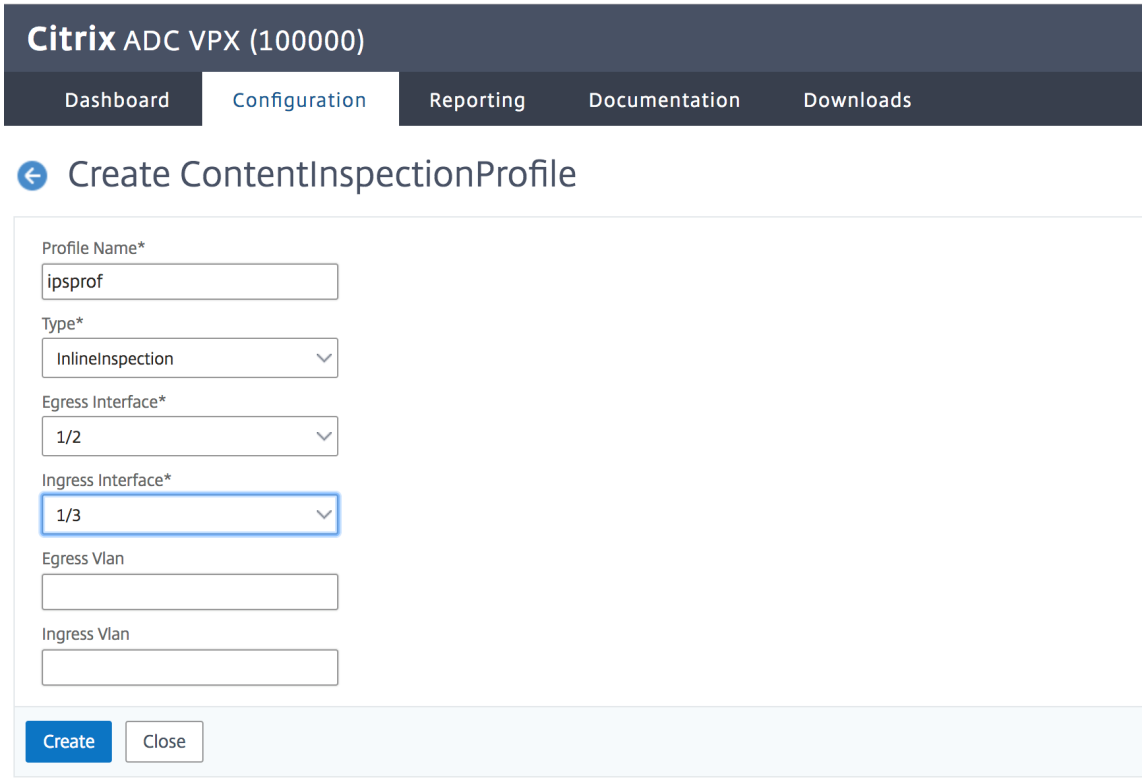


2. **System > Settings** に移動します。[モードと機能] で、[高度な機能の構成] をクリックします。

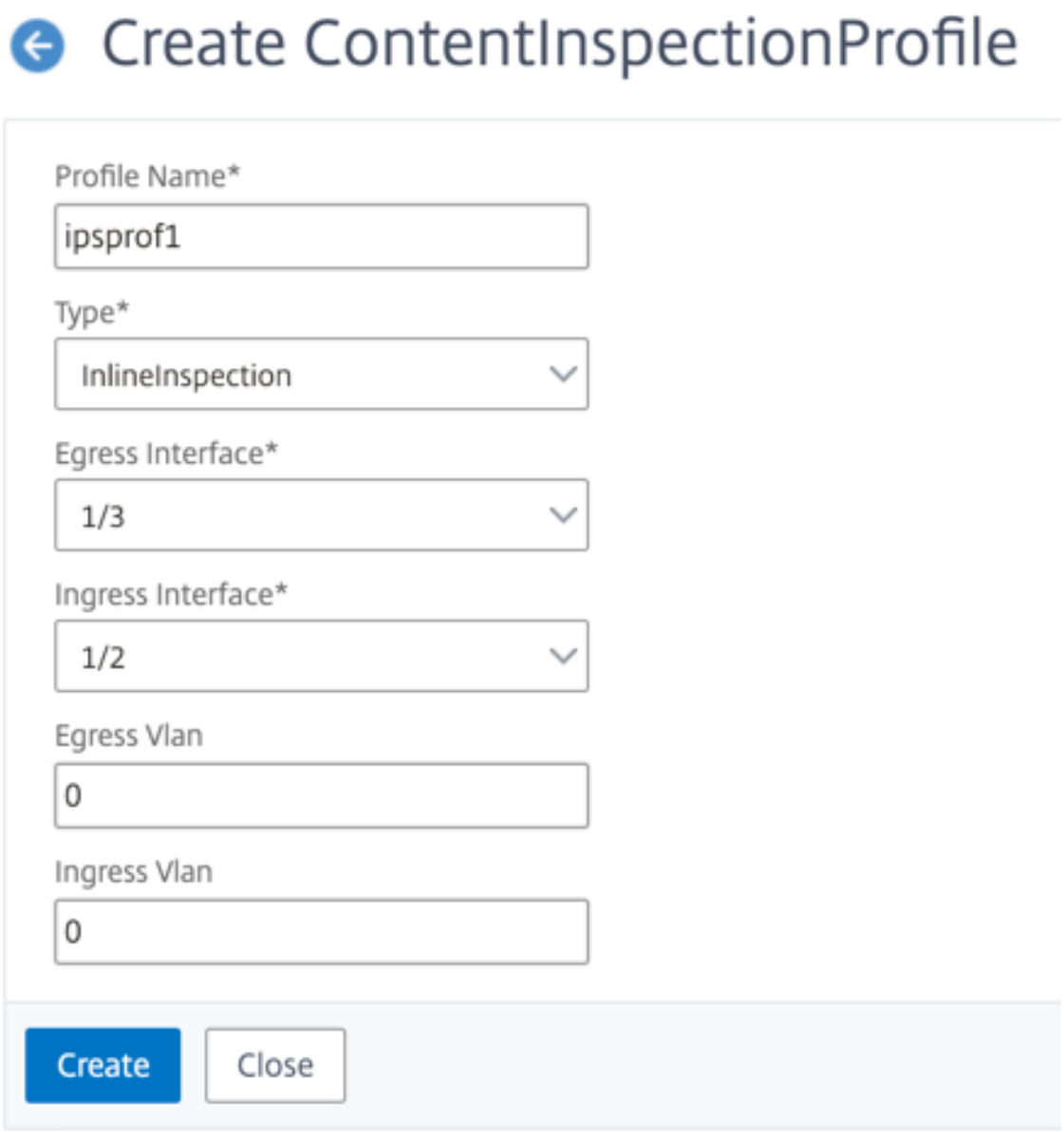




3. [**Secure Web Gateway] > [**コンテンツ検査**] > [**コンテンツ検査プロファイル] に移動します。[追加] をクリックします。

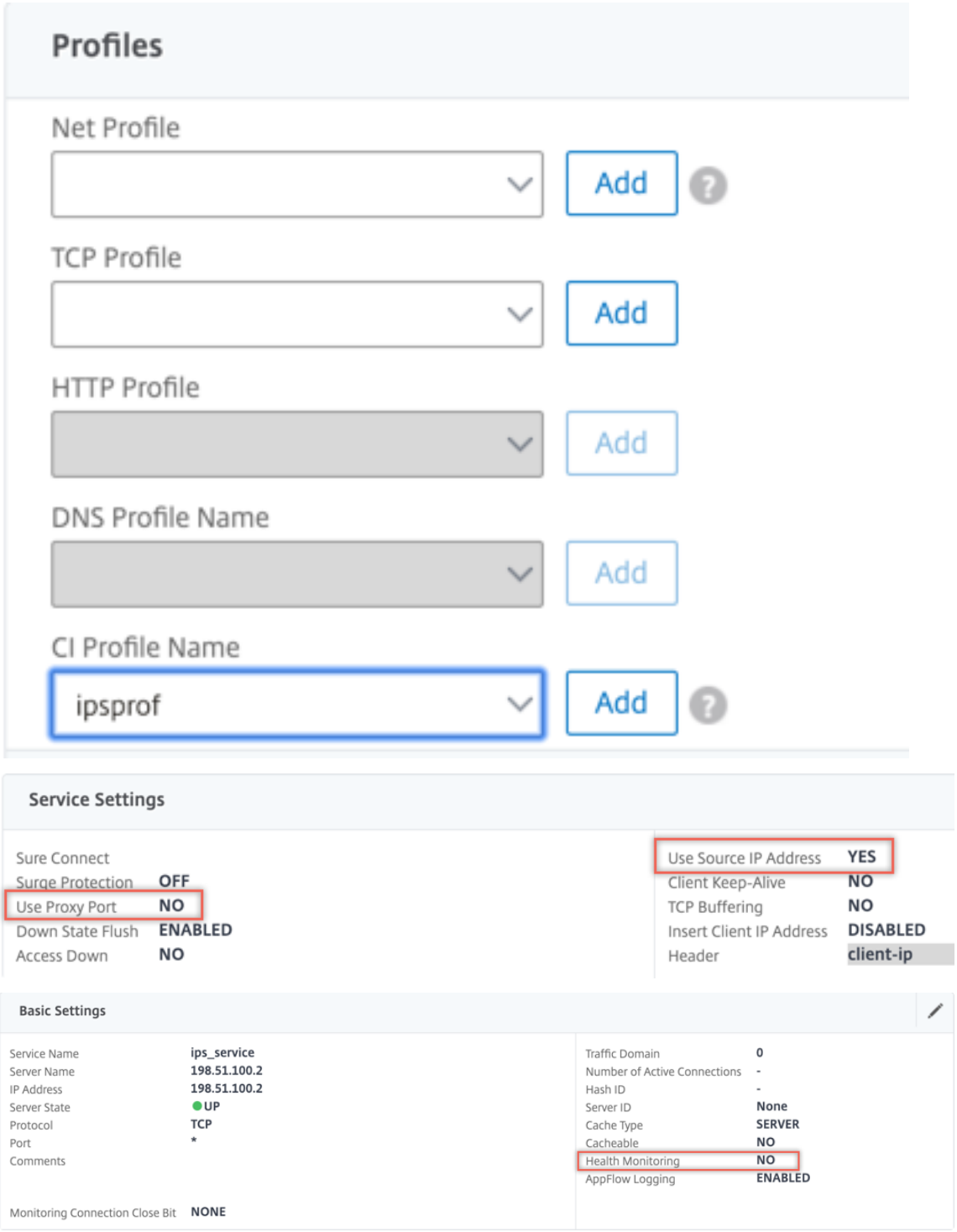


入力および出力インターフェイスを指定します。



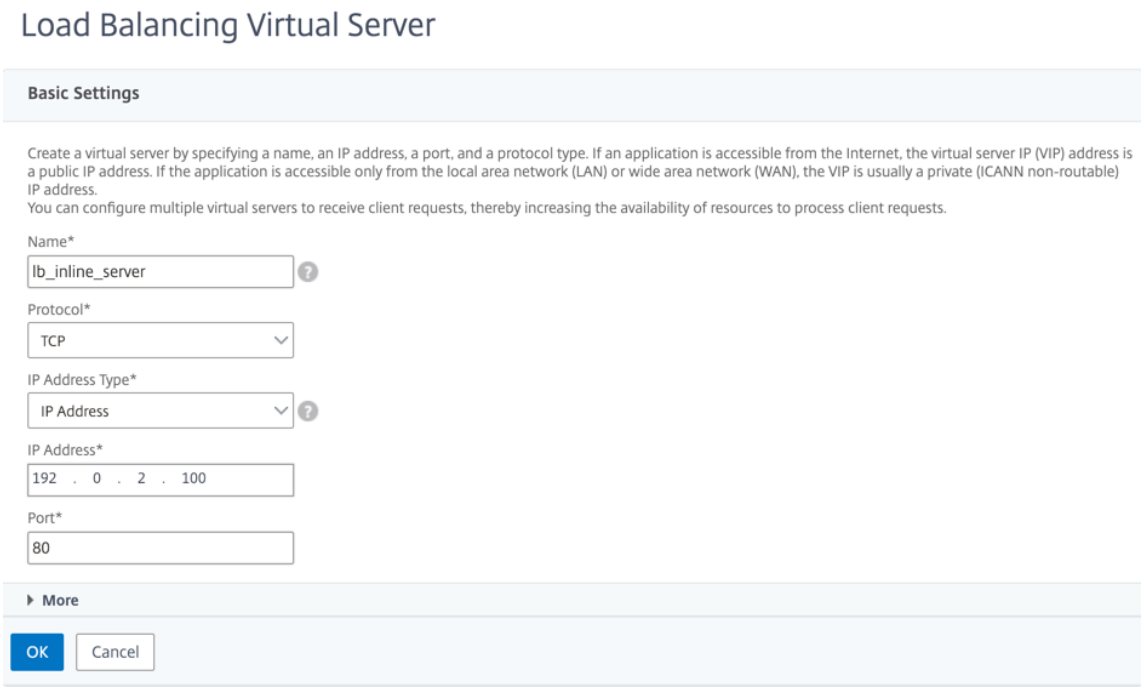
2 つのプロファイルを作成します。2 番目のプロファイルで、異なる入力および出力インターフェイスを指定します。

4. [負荷分散] **>** [サービス] **>** [サービスの追加と追加] に移動します。[詳細設定] で、[プロファイル] をクリックします。[**CI** プロファイル名] リストで、以前に作成したコンテンツ検査プロファイルを選択します。[サービス設定] で、[ソース **IP** アドレスを使用] を [はい]、[プロキシポートを使用] を [いいえ] に設定します。[基本設定] で、[ヘルスモニタリング] を [いいえ] に設定します。このサービスを TCP モニターにバインドする場合のみ、ヘルスモニタリングをオンにします。モニタをサービスにバインドする場合は、モニタの TRANSPARENT オプションを ON に設定します。



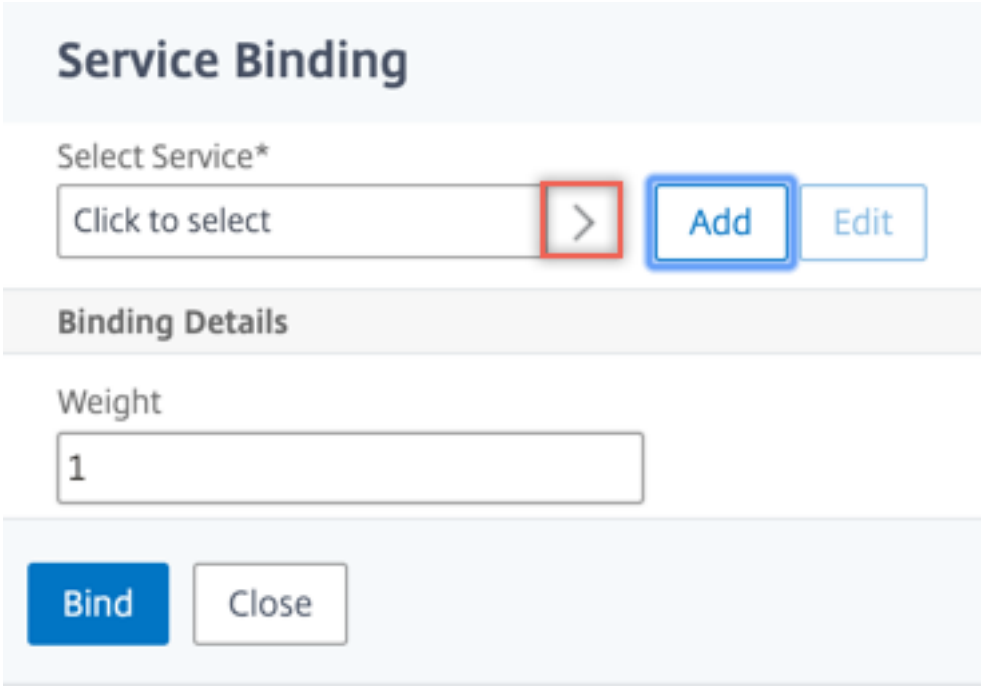
2 つのサービスを作成します。インラインデバイスを含むどのデバイスにも所有されていないダミー IP アドレスを指定します。

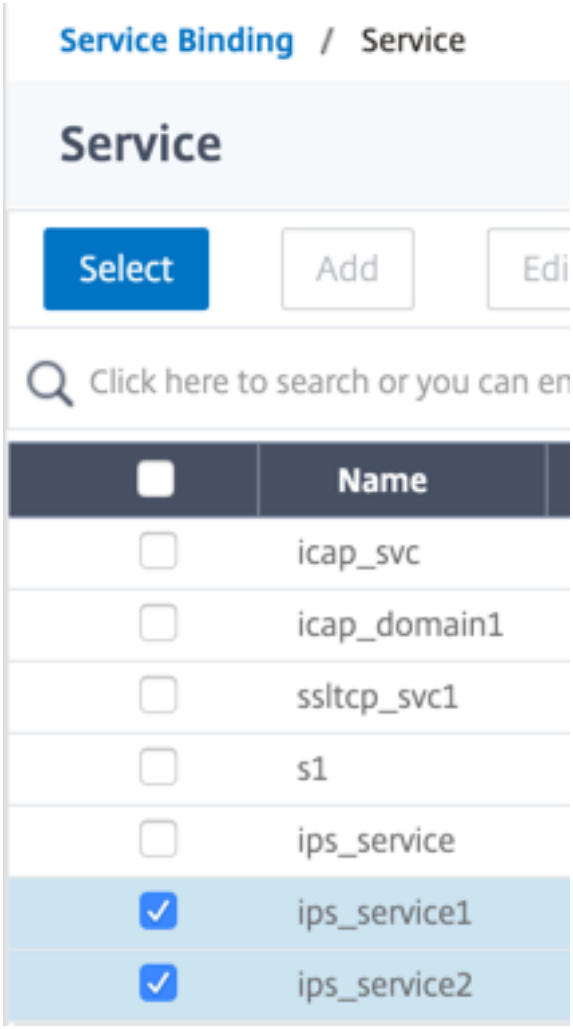
5. [負荷分散] **>** [仮想サーバー] **>** [追加] に移動します。TCP 負荷分散仮想サーバーを作成します。

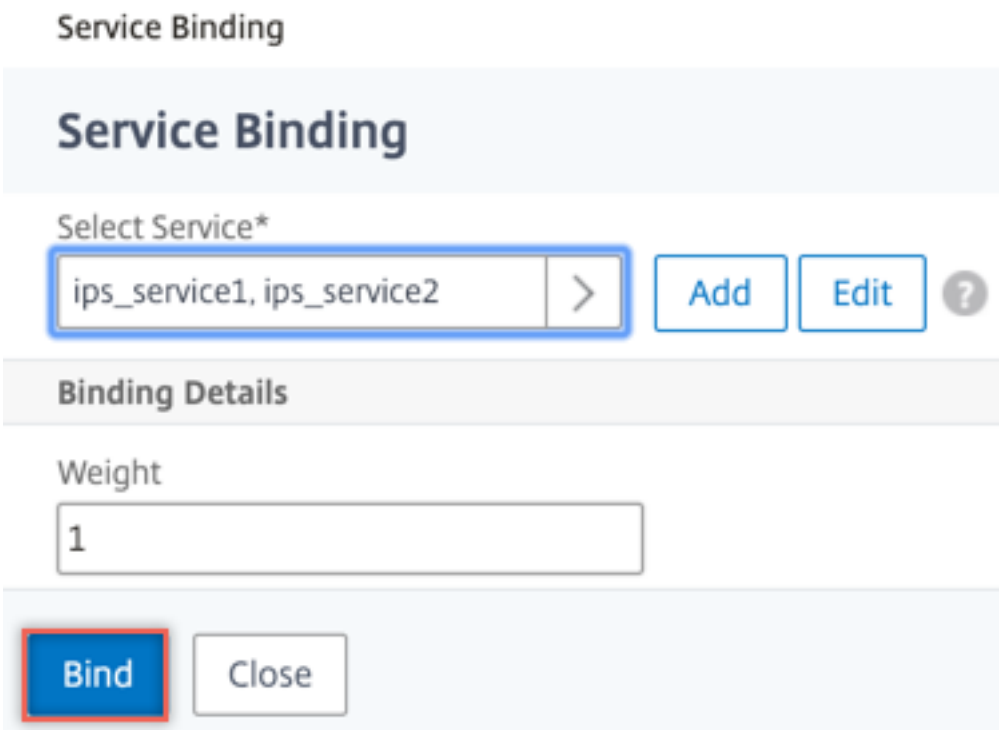


［**OK**］をクリックします。

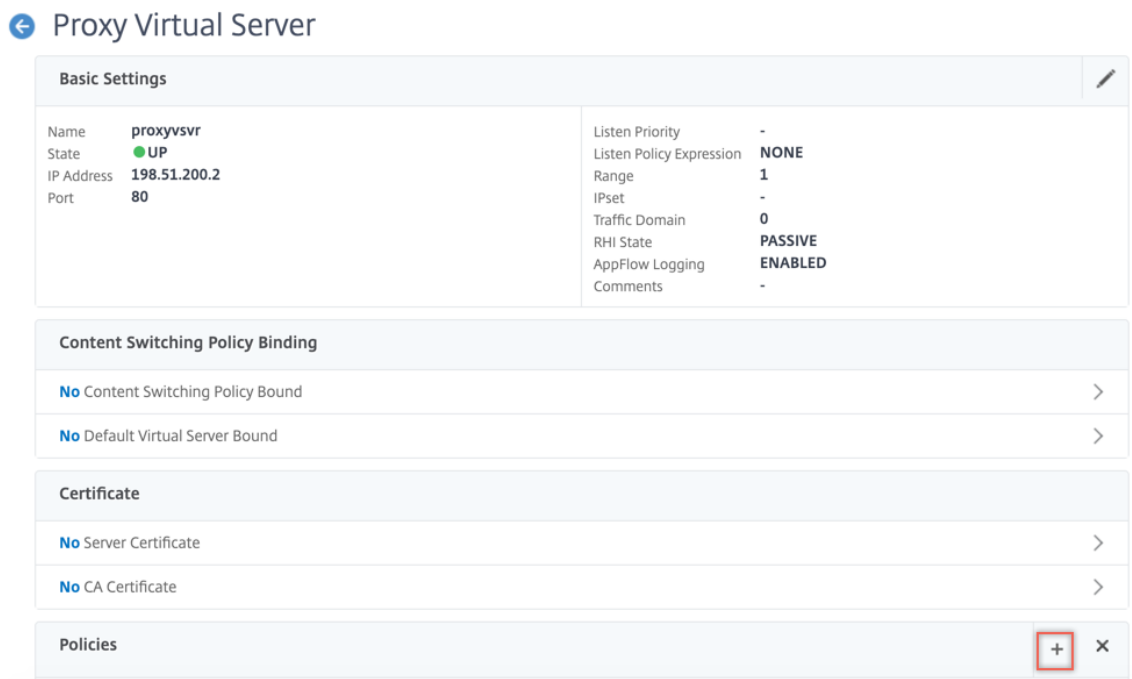
6. [負荷分散仮想サーバーサービスのバインド] セクション内をクリックします。「サービス・バインド」で、「サービスの選択」の矢印をクリックします。前に作成した 2 つのサービスを選択し、[**Select**] をクリックします。［バインド］をクリックします。



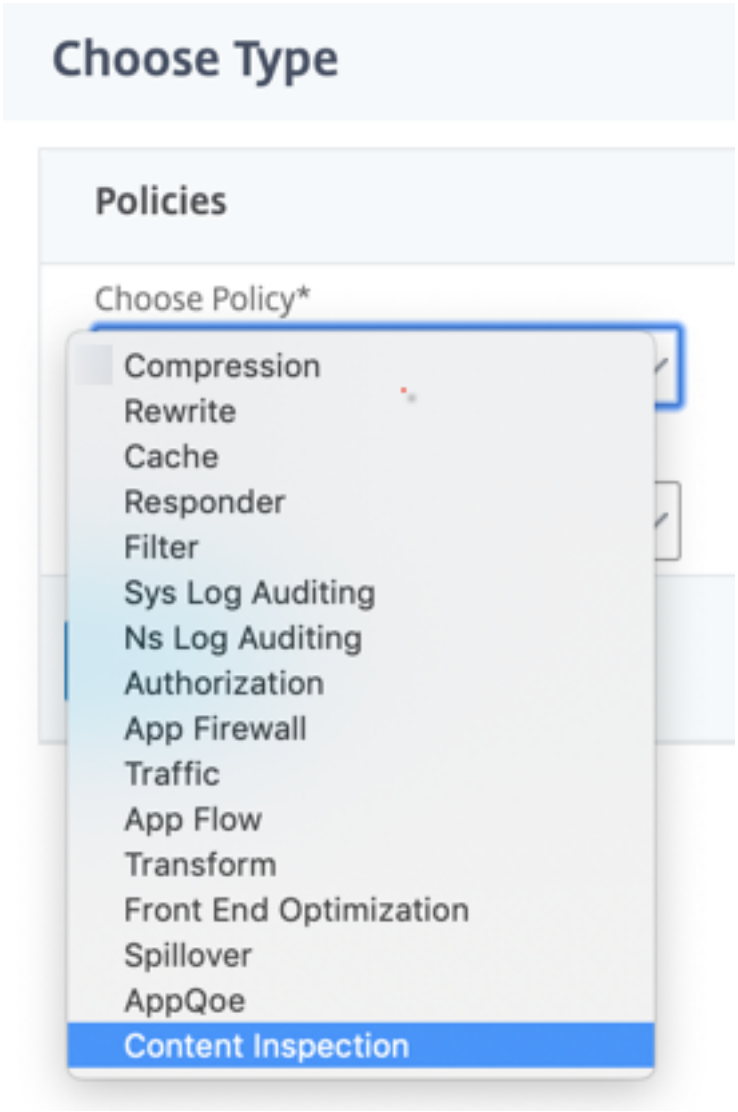




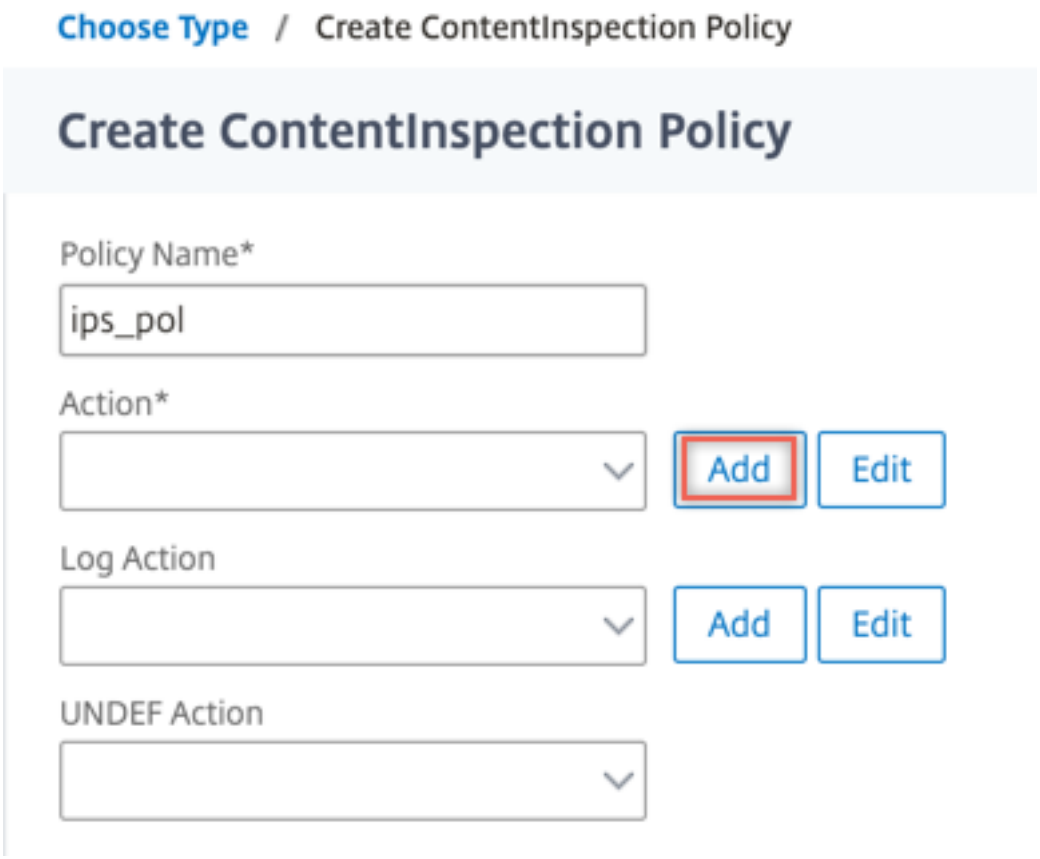
7. [**Secure Web Gateway**] > [プロキシ仮想サーバー] > [追加] に移動します。名前、IP アドレス、およびポートを指定します。[詳細設定] で、[ポリシー] を選択します。「+」記号をクリックします。



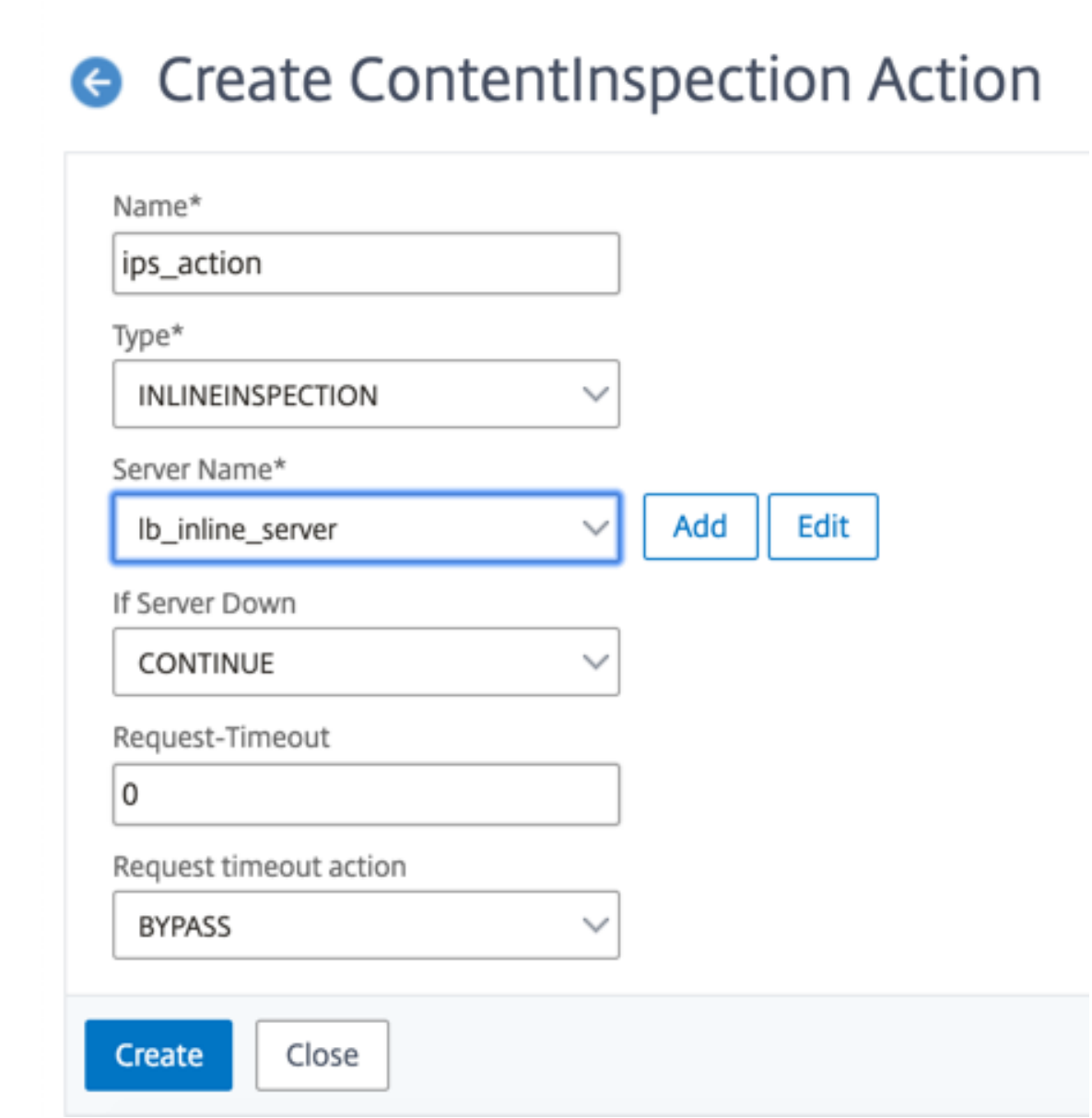
8. [ポリシーの選択] で [コンテンツ検査] を選択します。[続行] をクリックします。



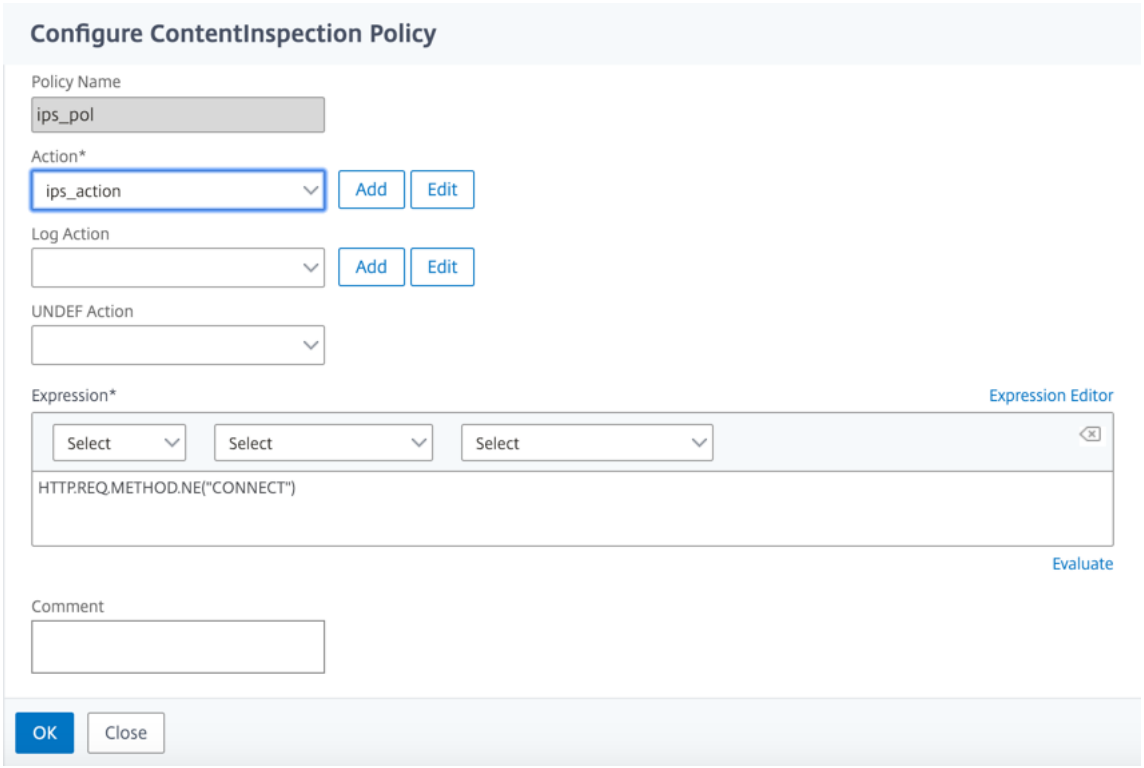
9.［追加］をクリックします。名前を指定します。「アクション」で、「追加」をクリックします。



10. 名前を指定します。「タイプ」で「**INLINEINSPECTION**」を選択します。「サーバー名」で、以前に作成した負荷分散仮想サーバーを選択します。



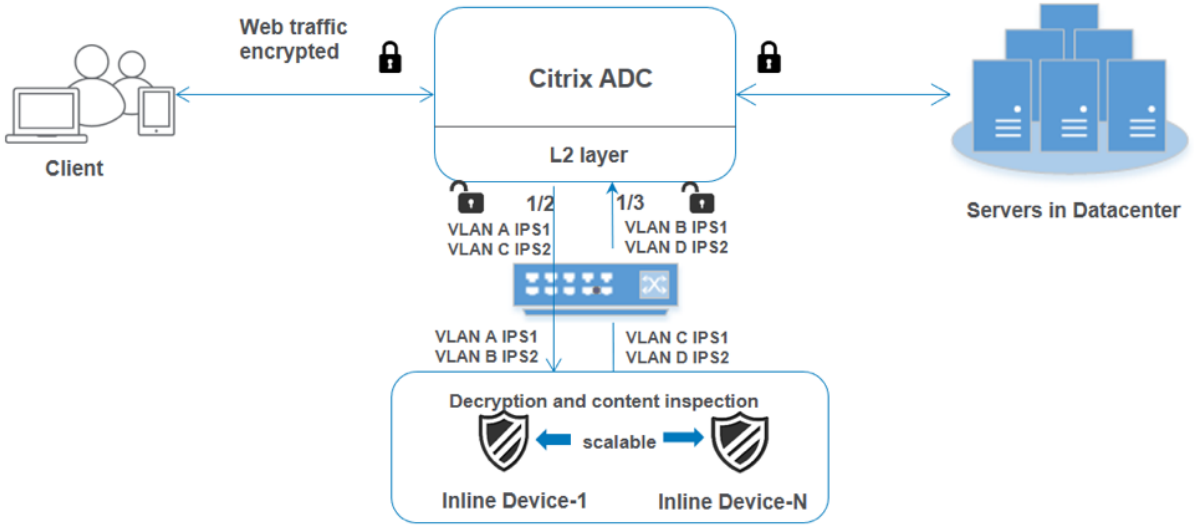
11. ［作成］をクリックします。ルールを指定し、［**Create**］をクリックします。



12. ［バインド］をクリックします。
13. ［完了］をクリックします。

シナリオ **3**: 共有インターフェイスを持つ複数のインラインデバイスの負荷分散

2 つ以上のインラインデバイスを使用している場合は、共有インターフェイスで異なるコンテンツインスペクションサービスを使用して、デバイスを負荷分散できます。この場合、SSL 転送プロキシアプライアンスは、共有インターフェイスを介して各デバイスに送信されるトラフィックのサブセットを負荷分散します。サブセットは、設定されたポリシーに基づいて決定されます。たとえば、TXT ファイルやイメージファイルは、検査のためにインラインデバイスに送信されない場合があります。



基本設定は、シナリオ 2 と同じままです。このシナリオでは、インターフェイスを異なる VLAN にバインドして、各インラインデバイスのトラフィックを分離します。コンテンツインスペクションプロファイルで VLAN を指定します。次の追加手順を実行します。

1. 共有インターフェイスを異なる VLAN にバインドします。
2. コンテンツ検査プロファイルで入力 VLAN と出力 VLAN を指定します。

CLI を使用した設定

コマンドプロンプトで次のコマンドを入力します。各コマンドの後に例が示されています。

1. MBF を有効にします。

```
enable ns mode mbf
```

1. 本機能を有効にします。

```
enable ns feature contentInspection
```

1. 共有インターフェイスを異なる VLAN にバインドします。

```
bind vlan <id> -ifnum <interface> -tagged
```

例:

```
bind vlan 100  - ifnum 1/2 tagged
bind vlan 200  - ifnum 1/3 tagged
bind vlan 300  - ifnum 1/2 tagged
bind vlan 400  - ifnum 1/3 tagged
<!--NeedCopy-->
```


1. サービス 1 のプロファイル 1 を追加します。プロファイルで入力 VLAN と出力 VLAN を指定します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof1 -type InlineInspection -egressInterface
 “1/3” -ingressinterface “1/2” -egressVlan 100 -ingressVlan 300
```

1. サービス 2 のプロファイル 2 を追加します。プロファイルで入力 VLAN と出力 VLAN を指定します。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

例:

```
add contentInspection profile ipsprof2 -type InlineInspection -egressInterface
 “1/3” -ingressinterface “1/2” -egressVlan 200 -ingressVlan 400
```

1. サービス 1 を追加します。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof1
```

1. サービス 2 を追加します。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

例:

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof2
```

1. 負荷分散仮想サーバーを追加します。

```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

例:

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. サービスを負荷分散仮想サーバーにバインドします。


```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```

例:

```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```

1. コンテンツインスペクションアクションで負荷分散仮想サーバを指定します。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

例:

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
lb_inline_vserver
```

1. コンテンツ検査ポリシーを追加します。ポリシーでコンテンツインスペクションアクションを指定します。

```
add contentInspection policy <name> -rule <expression> -action <string>
```

例:

```
add contentInspection policy ips_pol -rule ”HTTP.REQ.METHOD.NE(\”CONNECT\”)
”-action ips_action
```

1. プロキシ仮想サーバーを追加します。

```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

例:

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. コンテンツ検査ポリシーを仮想サーバにバインドします。

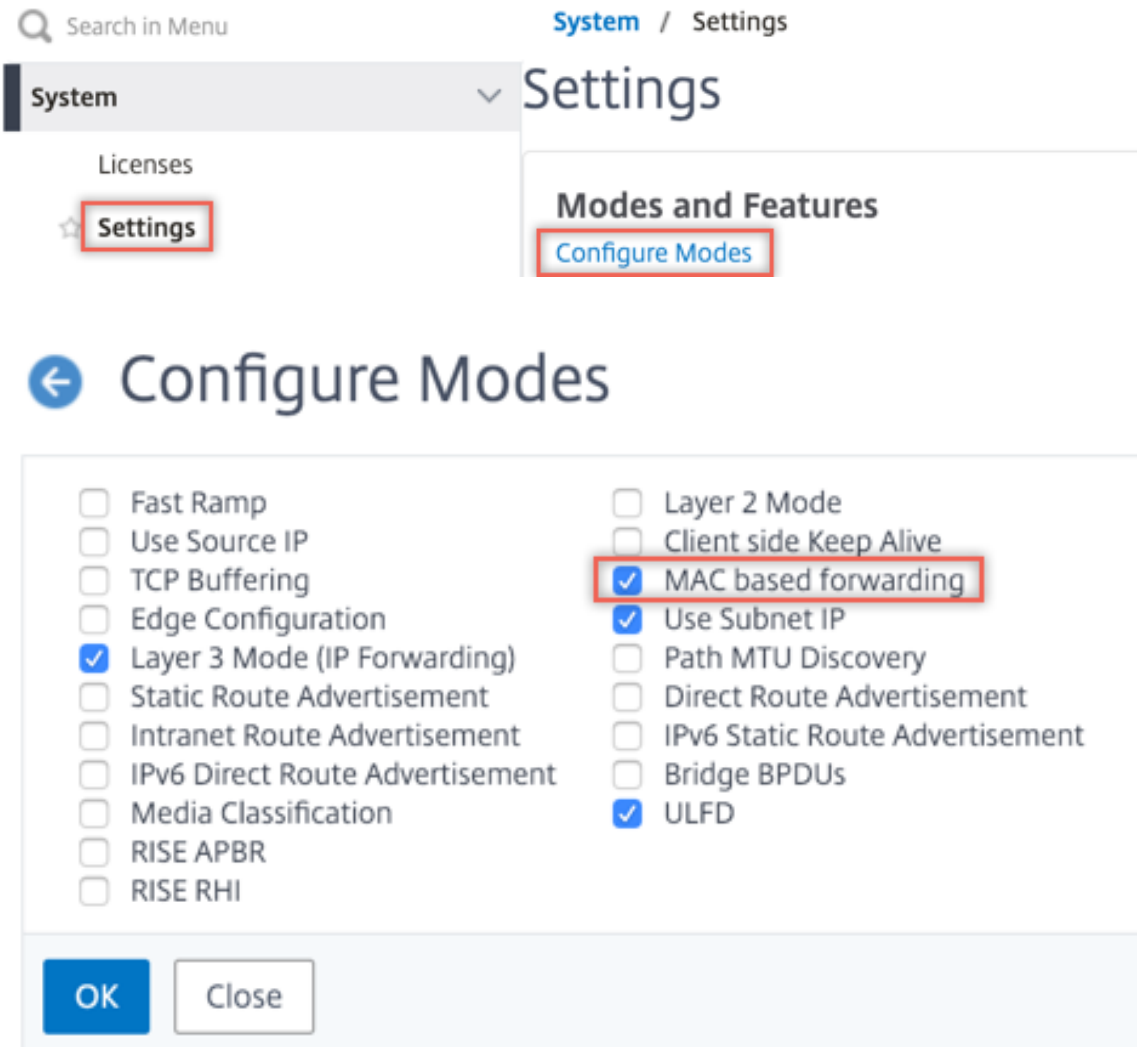
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

例:

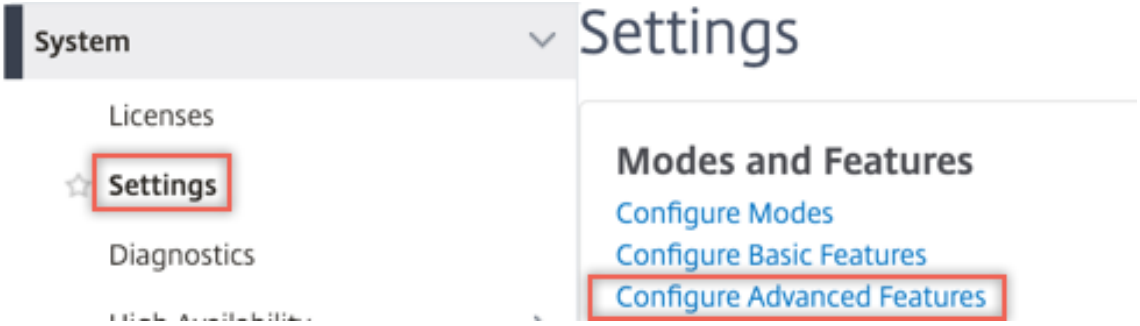
```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

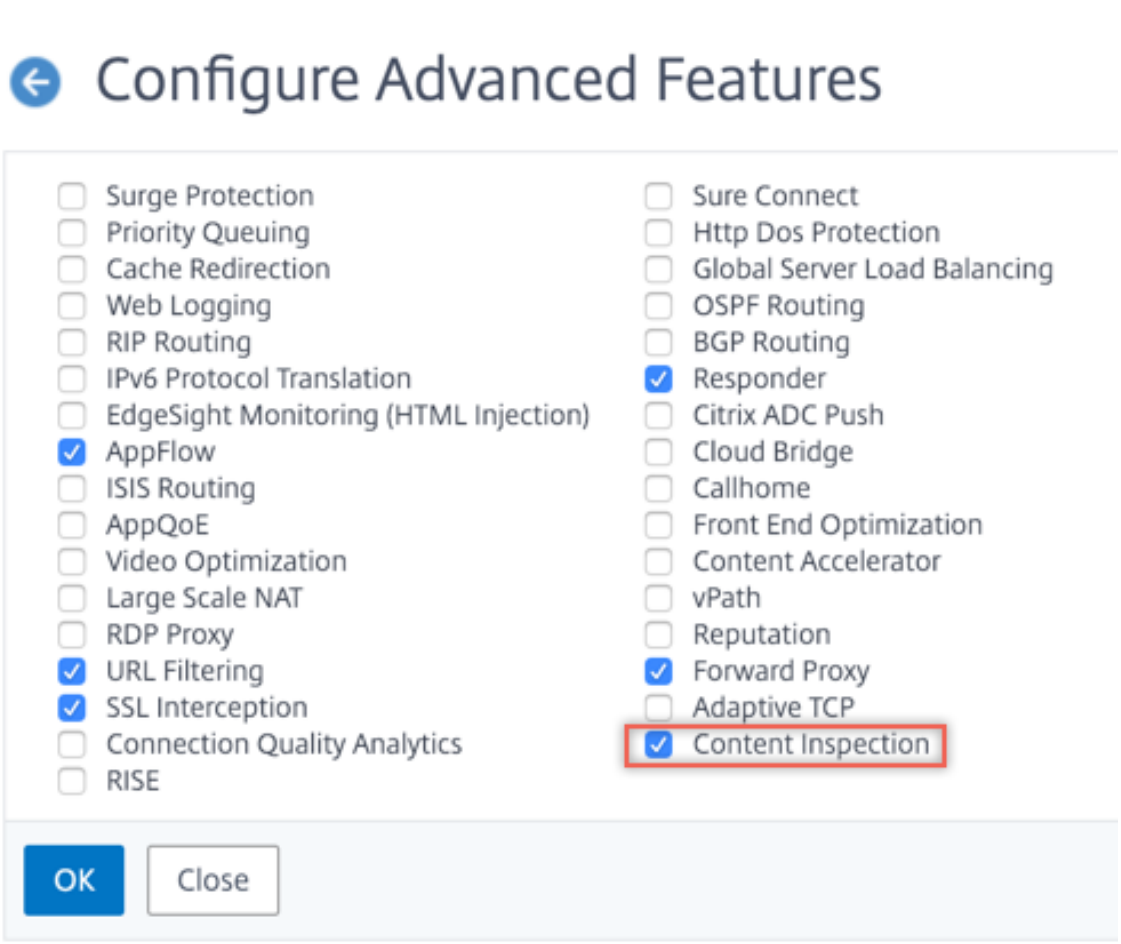
GUI を使用した設定

1. **System > Settings** に移動します。[モードと機能] で、[モードの構成] をクリックします。

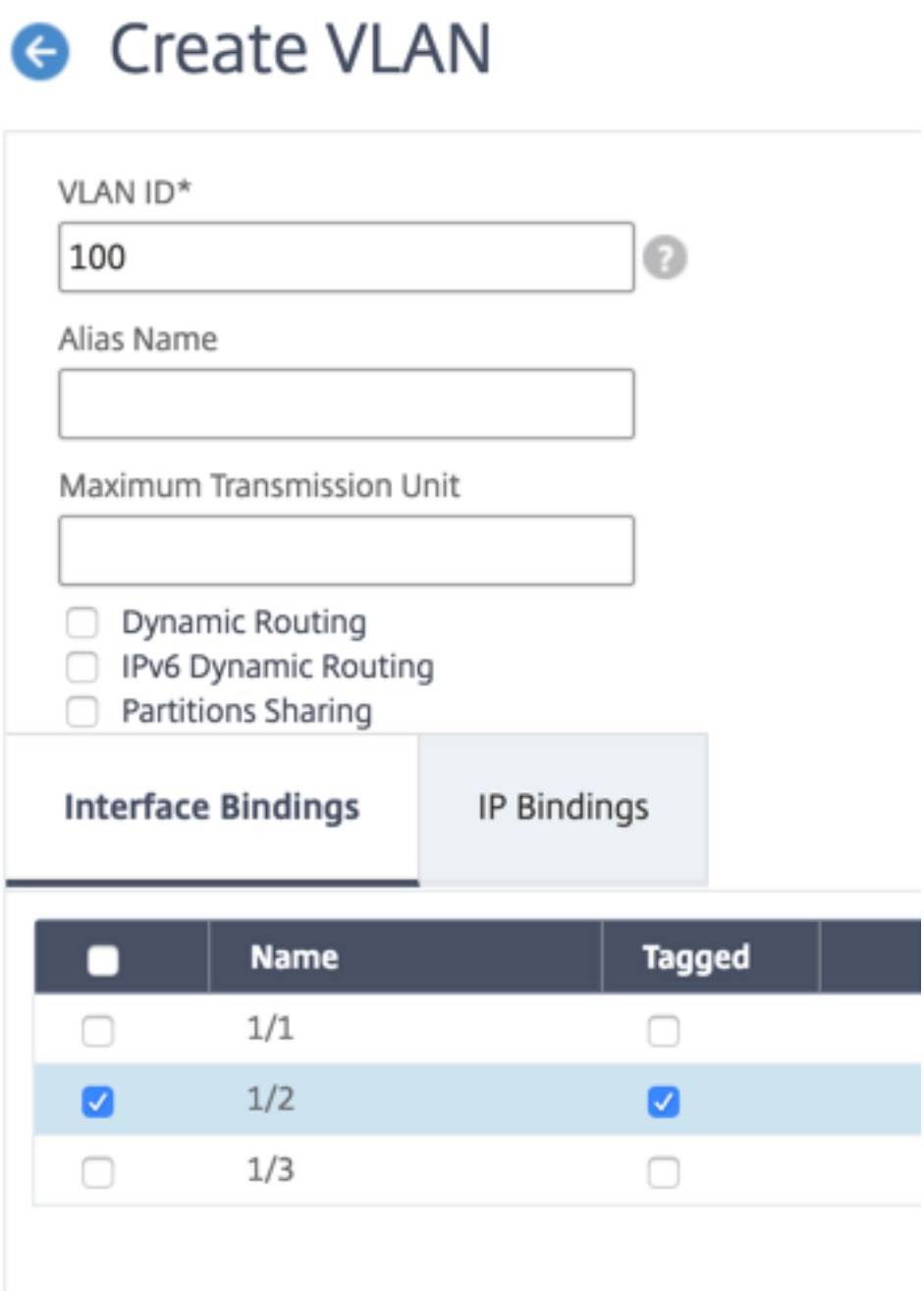


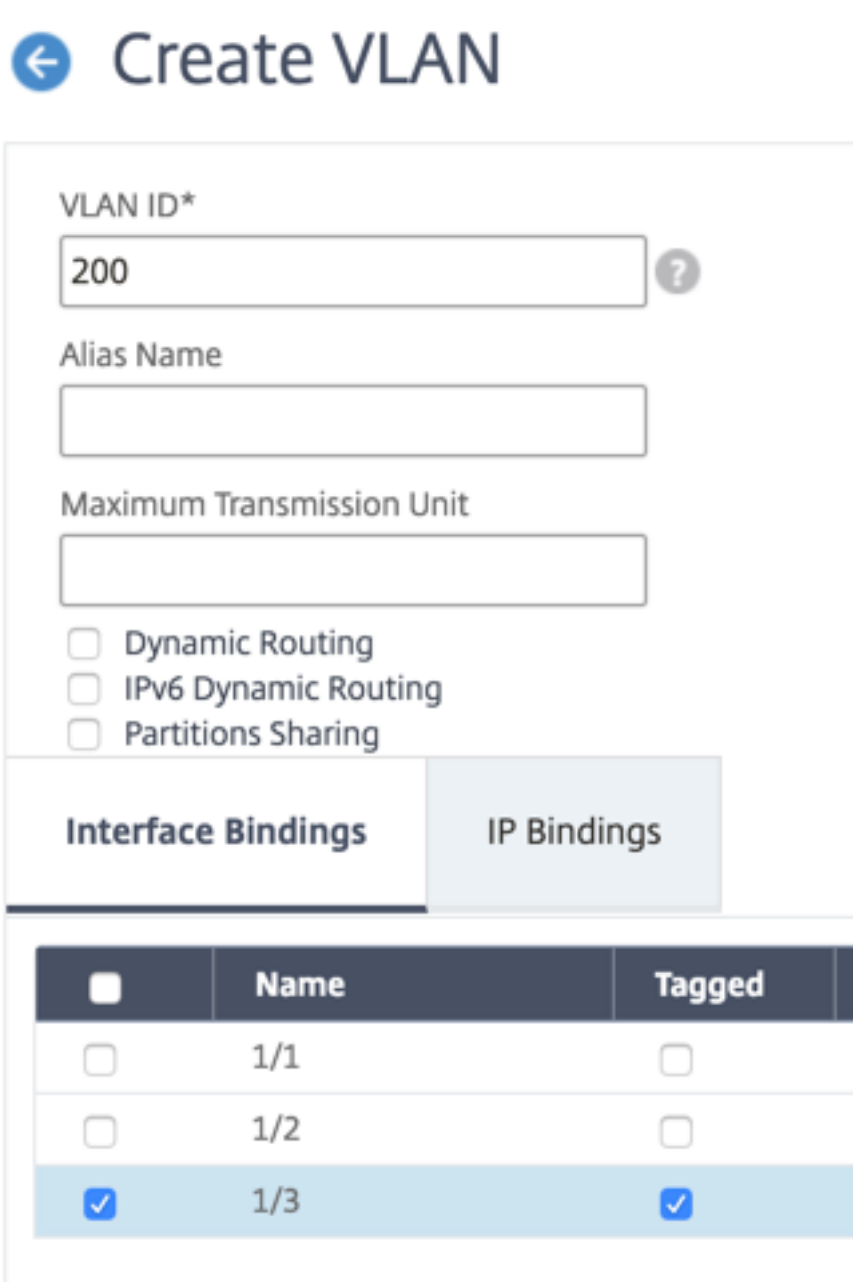
2. **System > Settings** に移動します。[モードと機能] で、[高度な機能の構成] をクリックします。

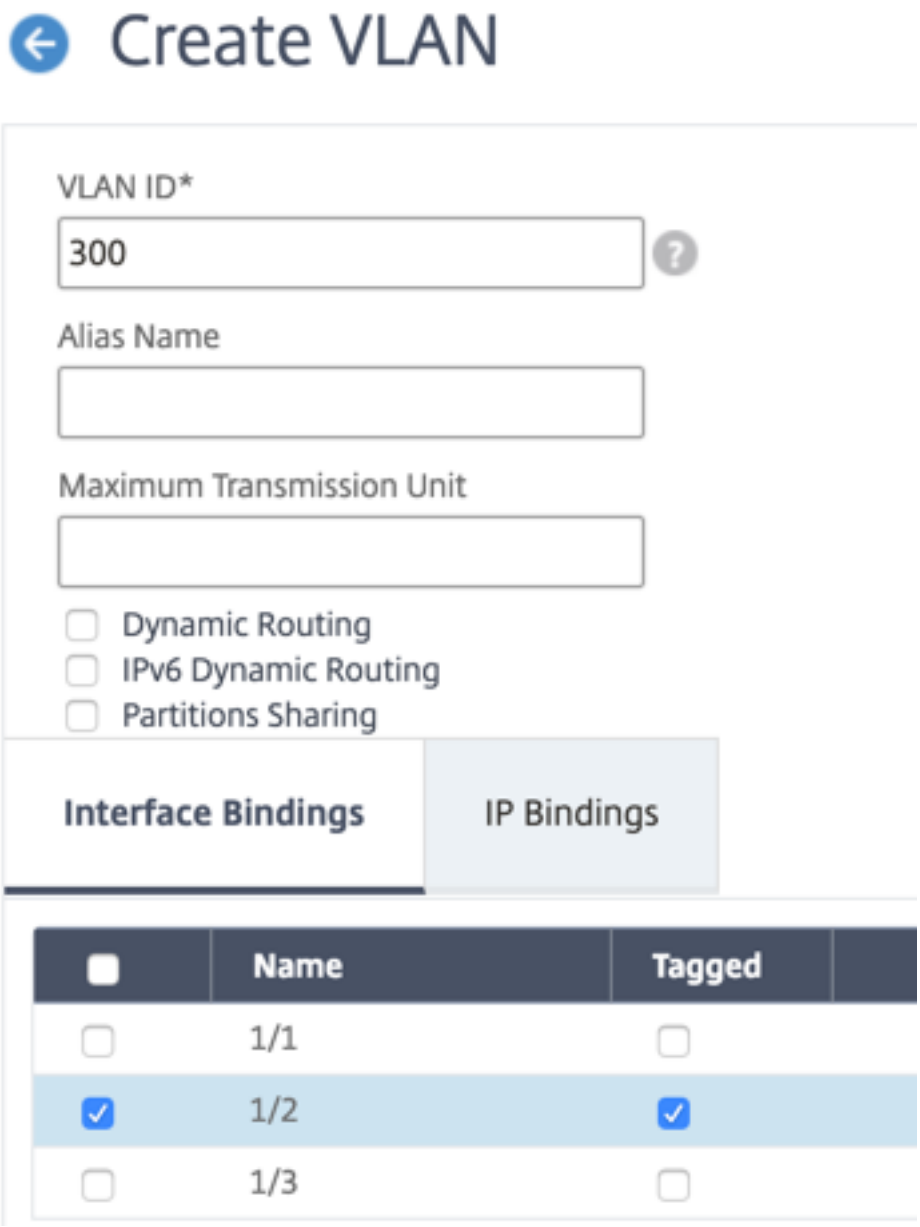


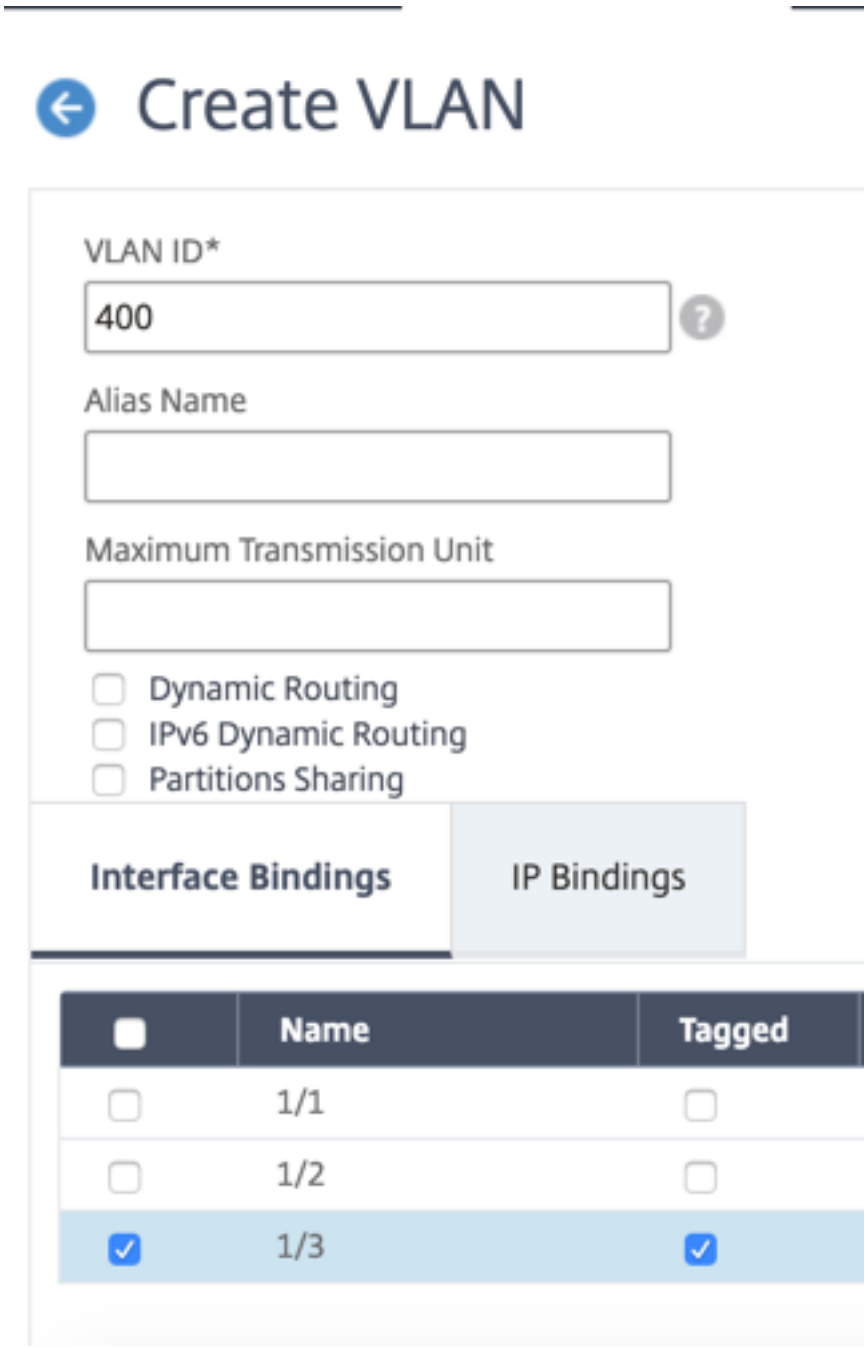


3. [システム] **>** [ネットワーク] **>** **[VLAN] >** [追加] に移動します。4 つの VLAN を追加し、インターフェイスにタグを付けます。

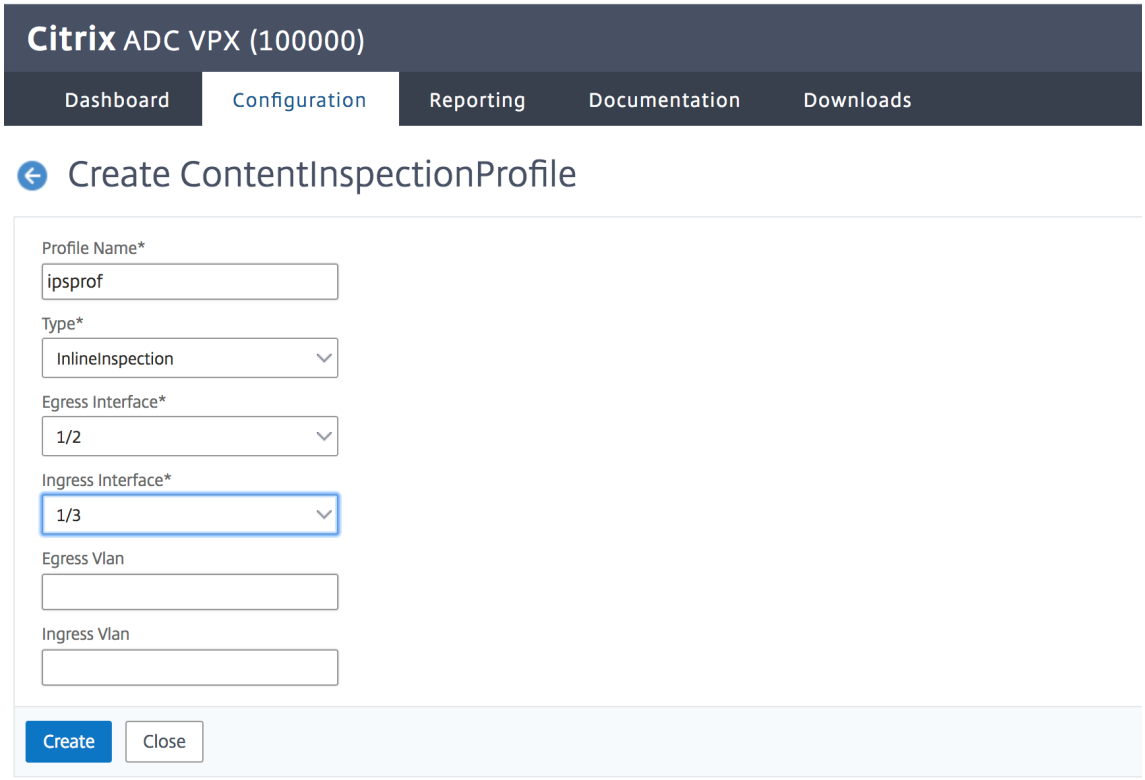




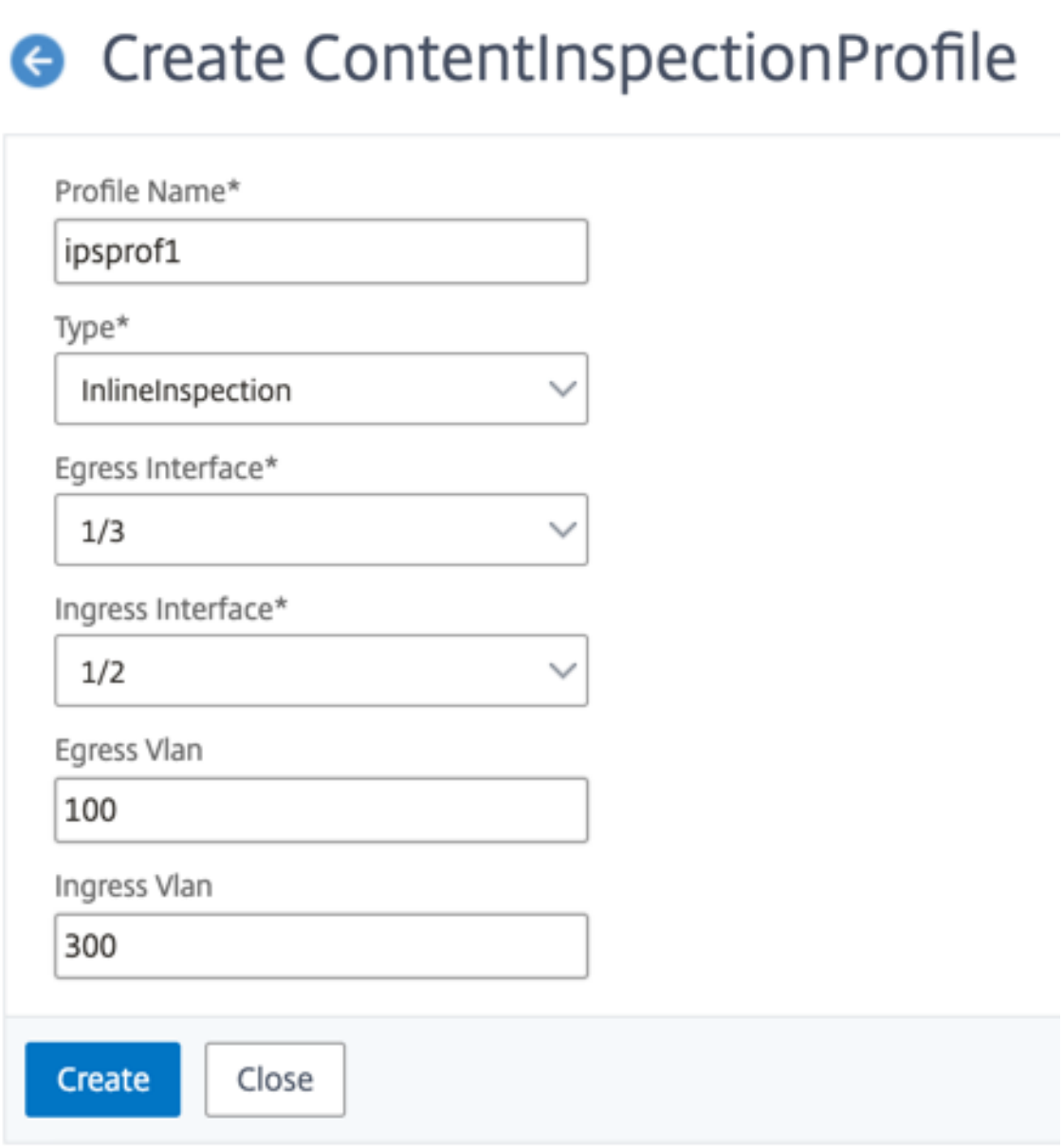




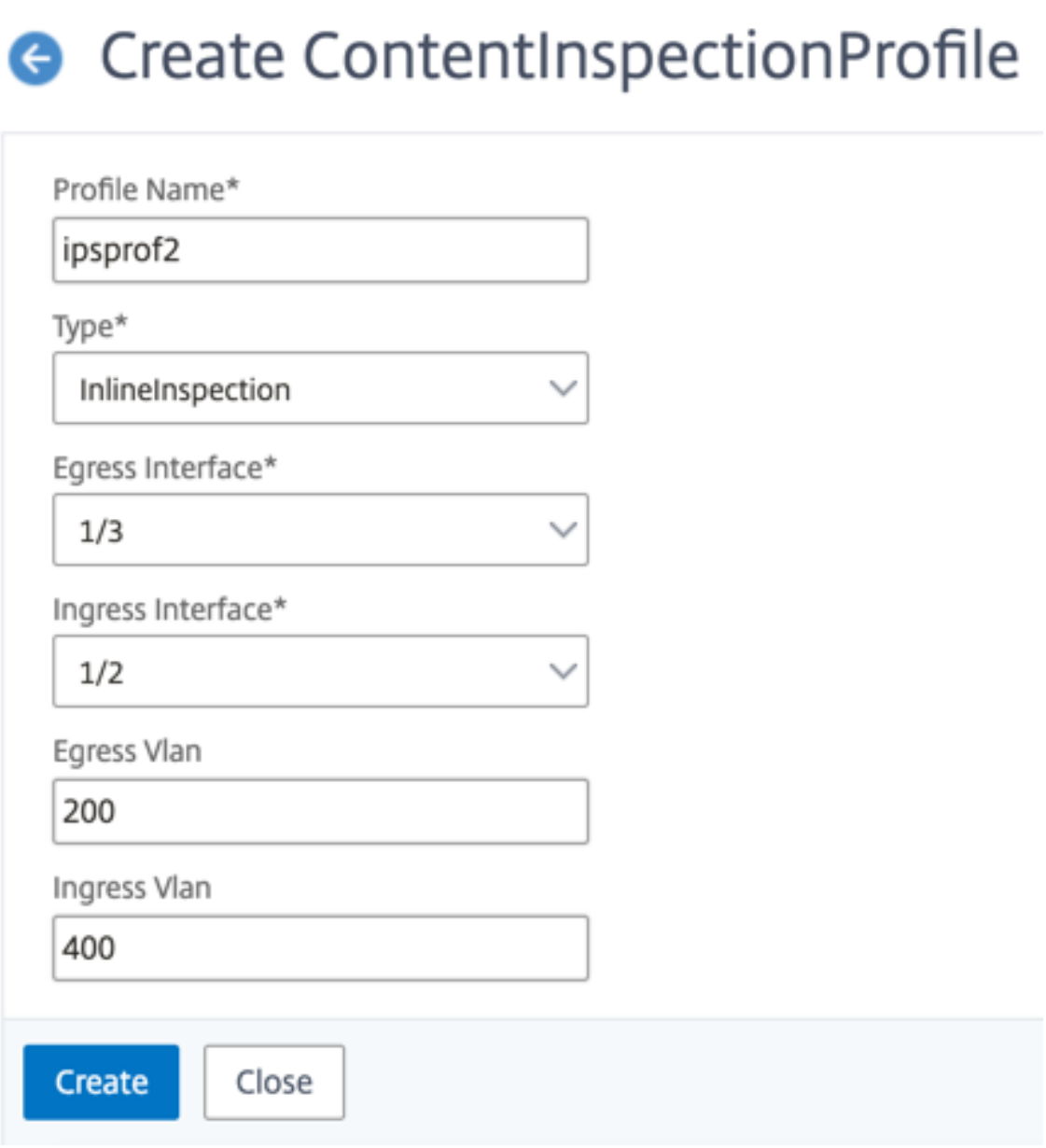
4. [**Secure Web Gateway] > [**コンテンツ検査**] > [**コンテンツ検査プロファイル] に移動します。[追加] をクリックします。



入力 VLAN と出力 VLAN を指定します。

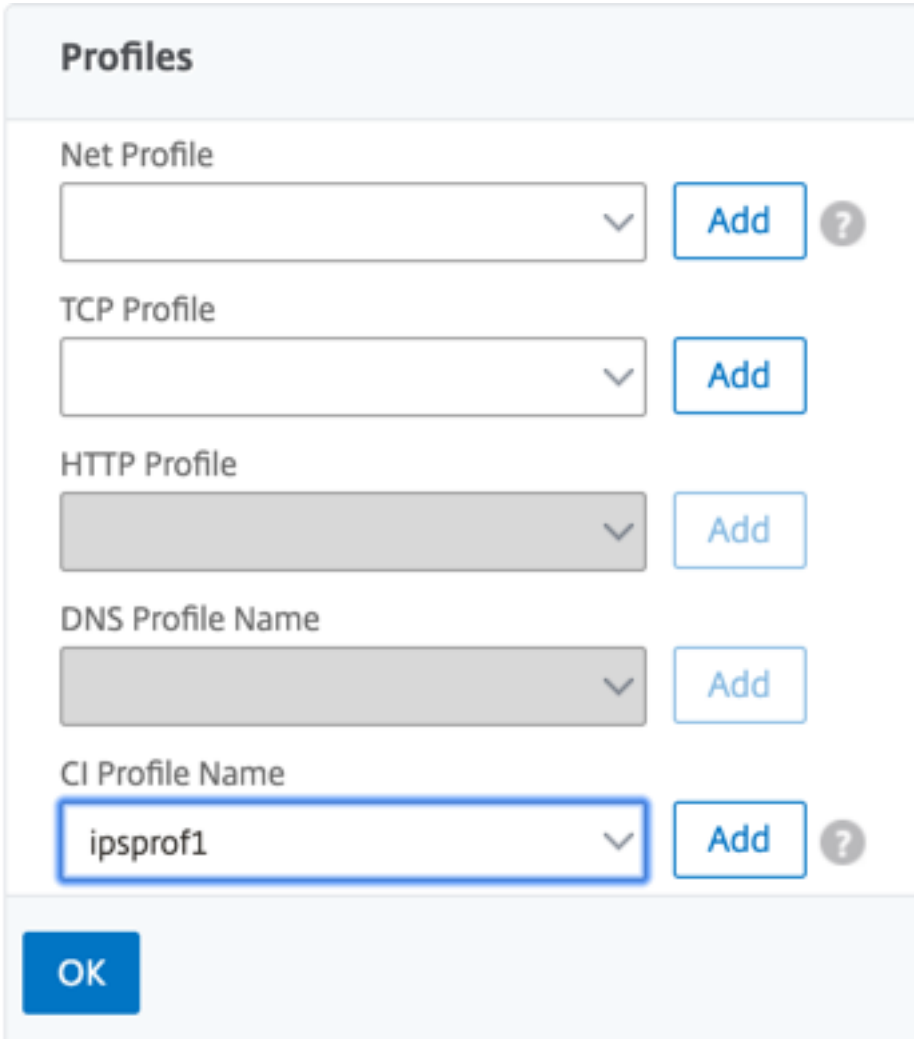


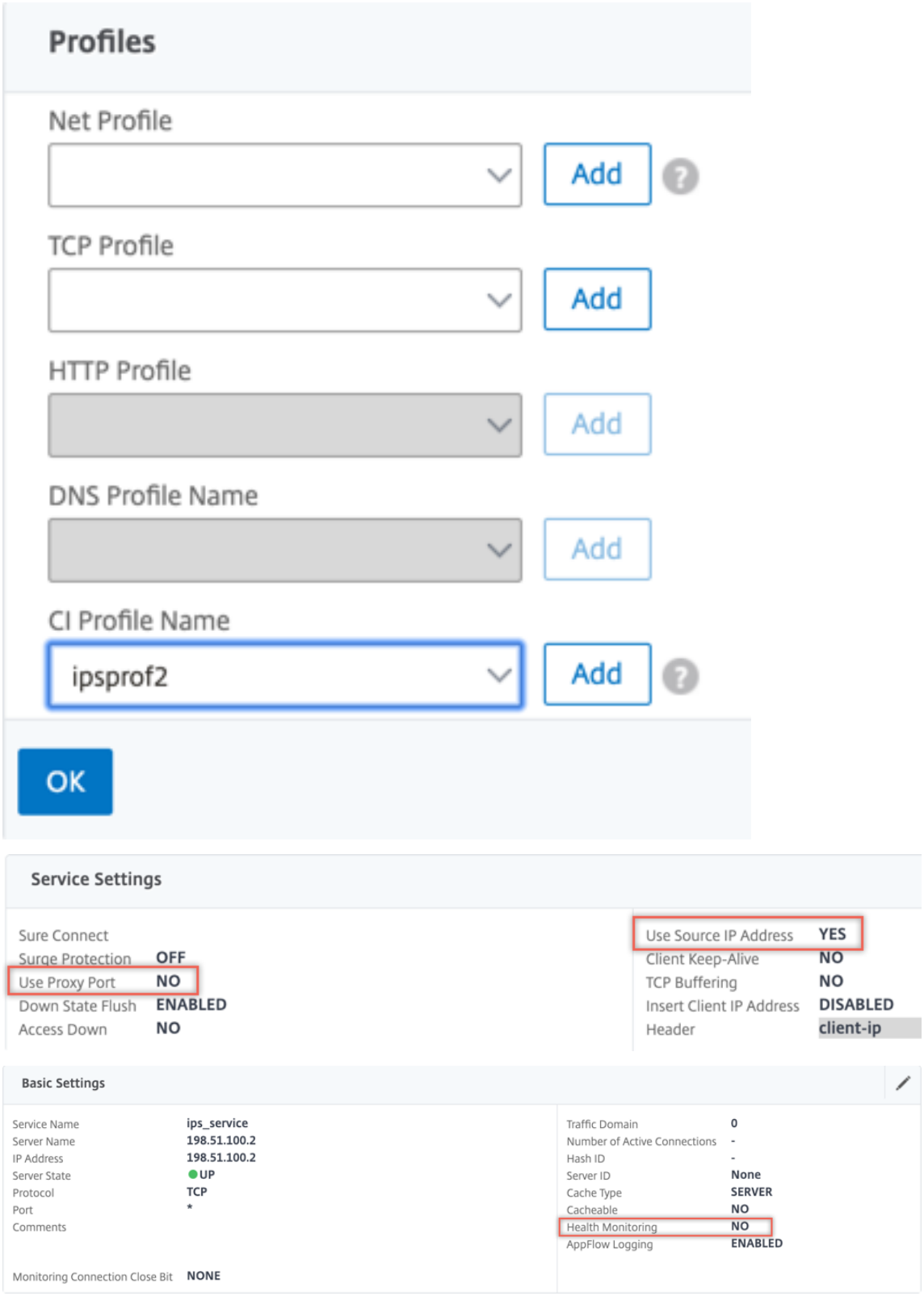
別のプロファイルを作成します。2 番目のプロファイルで異なる入力 VLAN と出力 VLAN を指定します。



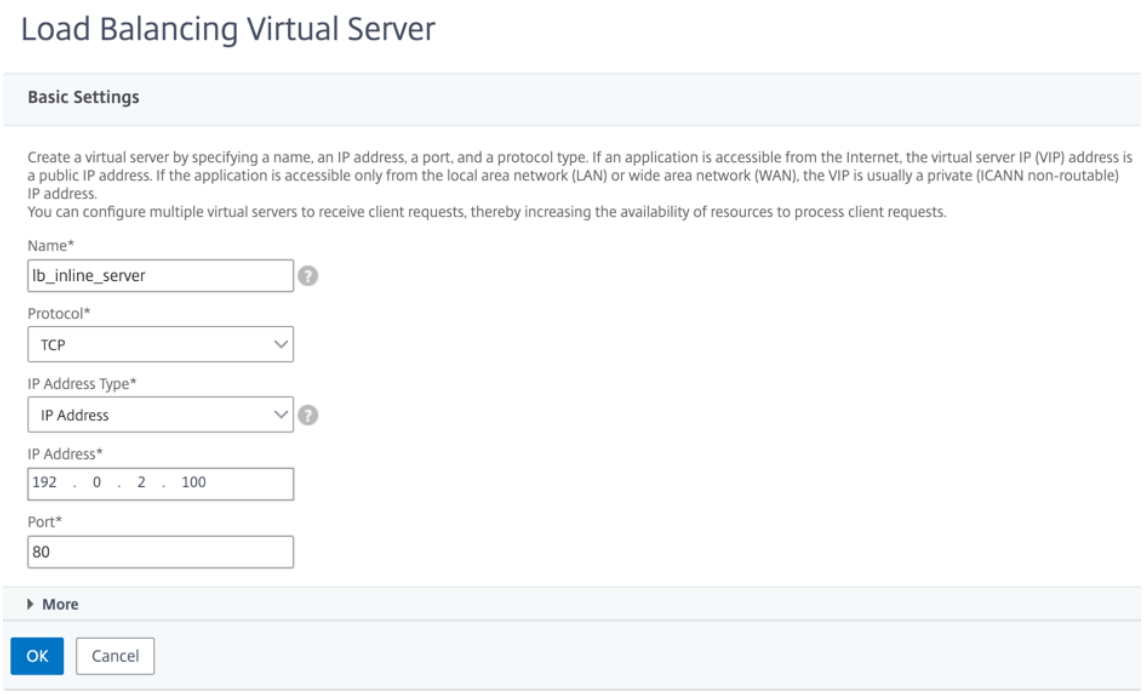
5. [負荷分散] **>** [サービス] **>** [サービスの追加と追加] に移動します。[詳細設定] で、[プロファイル] をクリックします。[**CI** プロファイル名] リストで、以前に作成したコンテンツ検査プロファイルを選択します。[サービス設定] で、[ソース **IP** アドレスを使用] を [はい]、[プロキシポートを使用] を [いいえ] に設定します。[基本設定] で、[ヘルスモニタリング] を [いいえ] に設定します。

   2 つのサービスを作成します。インラインデバイスを含むどのデバイスにも所有されていないダミー IP アドレスを指定します。サービス 1 でプロファイル 1 を指定し、サービス 2 でプロファイル 2 を指定します。

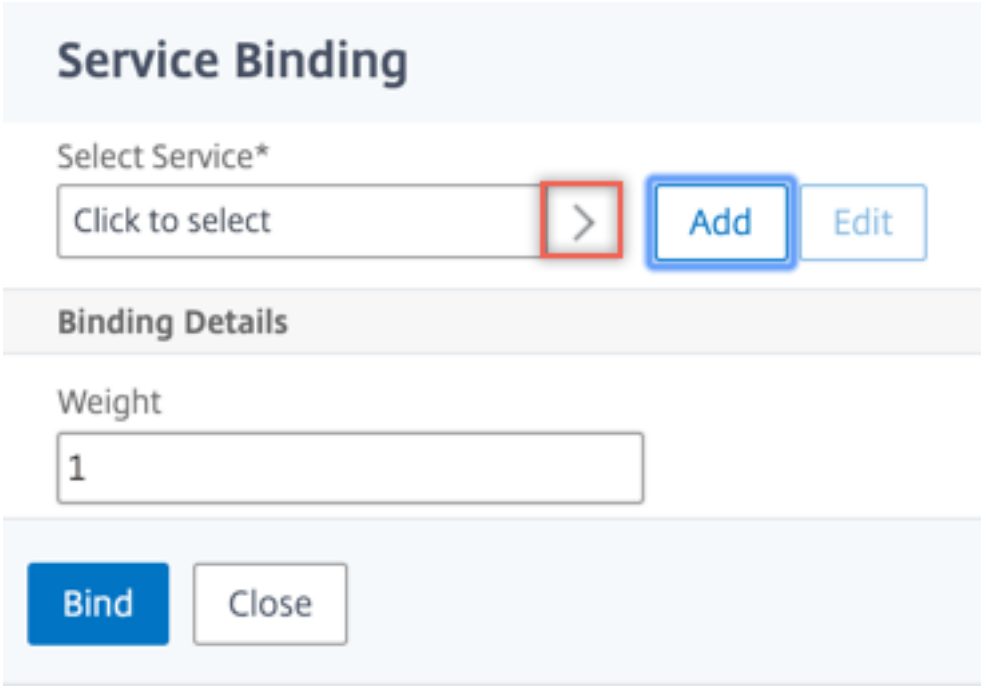


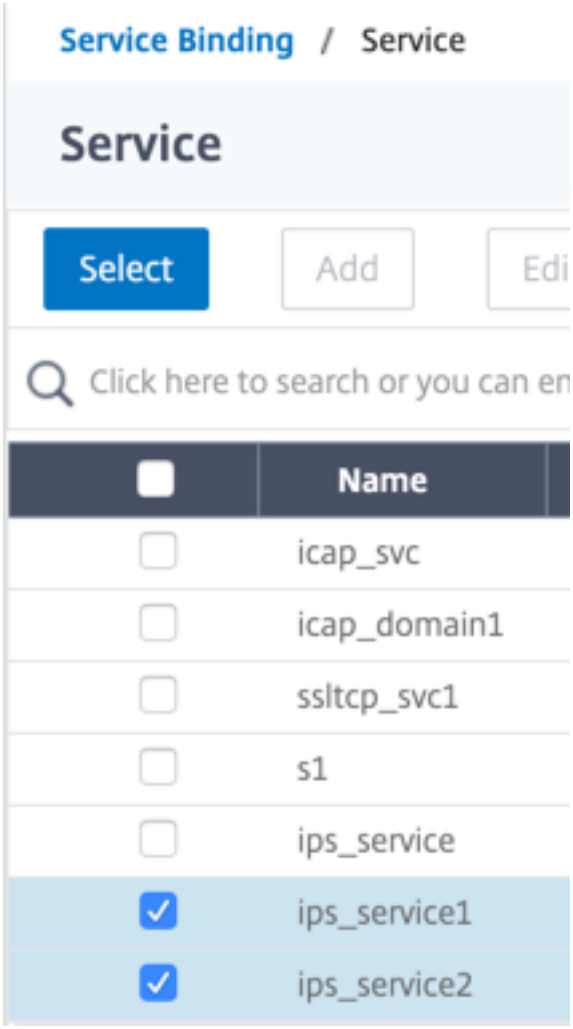


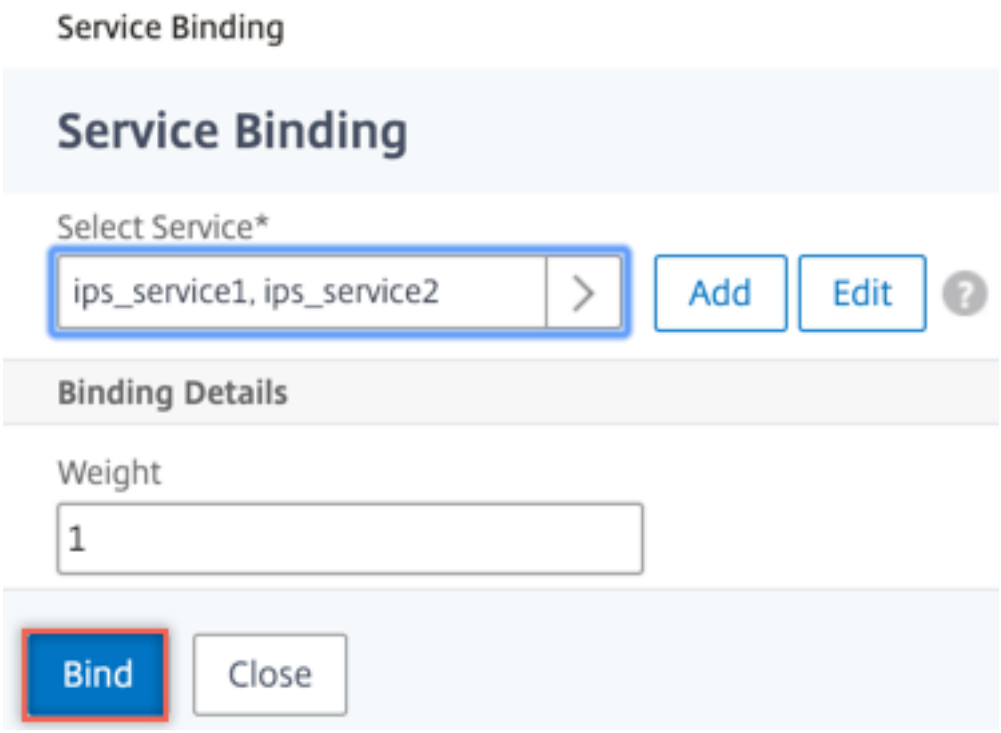
6. [負荷分散] > [仮想サーバー] > [追加] に移動します。TCP 負荷分散仮想サーバーを作成します。



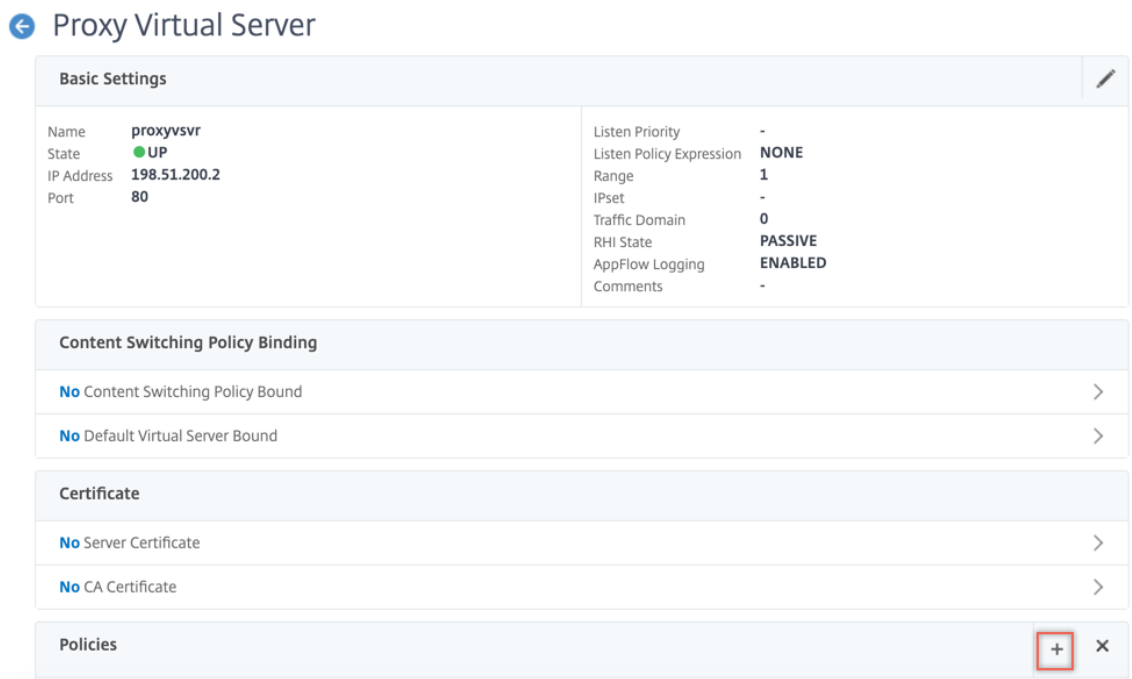
7. [**OK**] をクリックします。
8. [負荷分散仮想サーバーサービスのバインド] セクション内をクリックします。「サービス・バインド」で、「サービスの選択」の矢印をクリックします。前に作成した 2 つのサービスを選択し、[**Select**] をクリックします。[バインド] をクリックします。



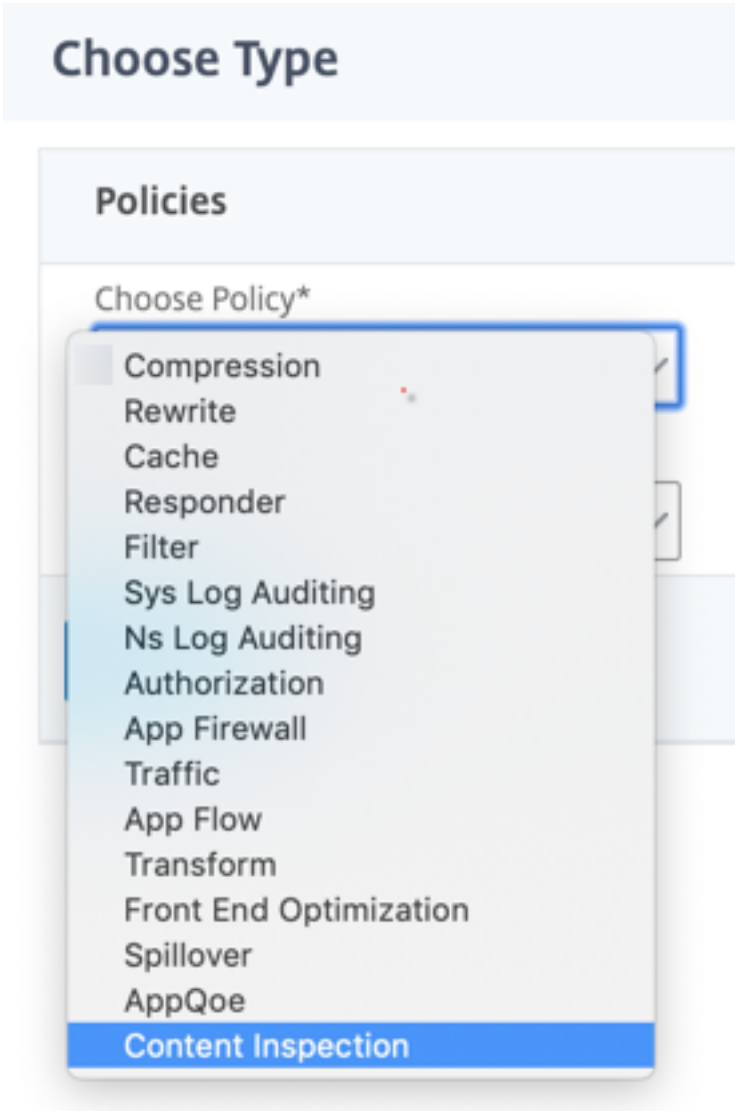




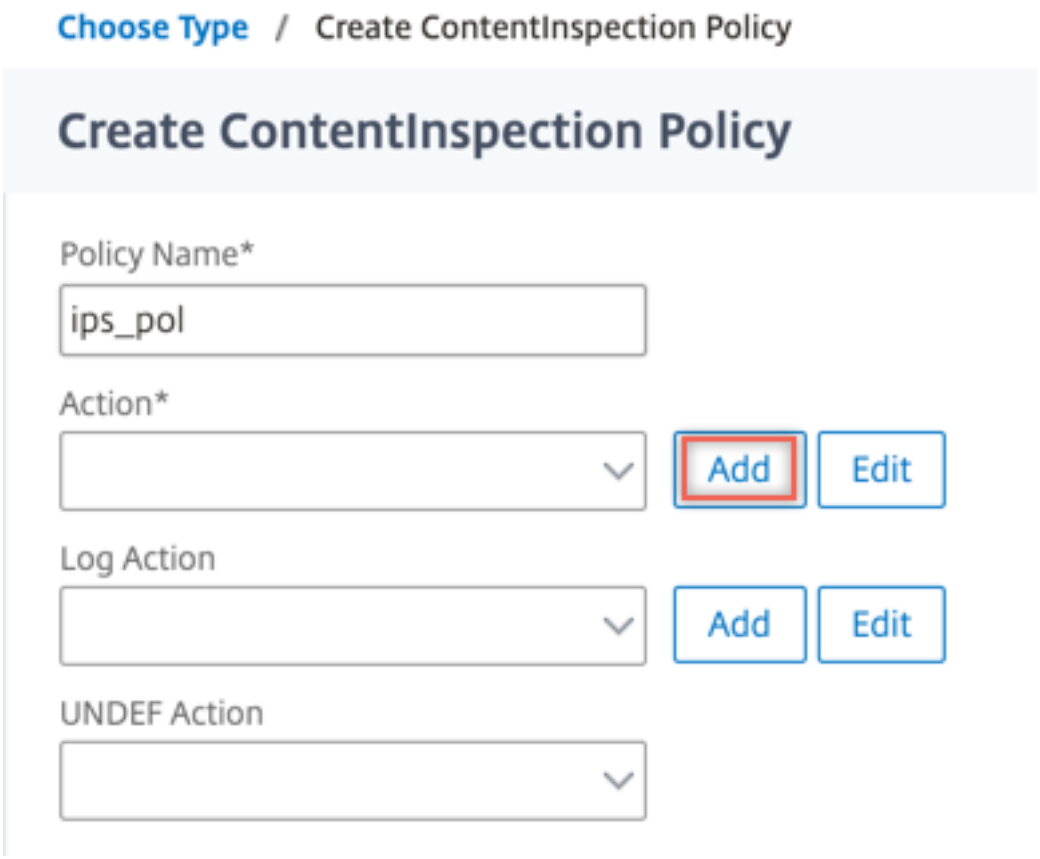
9. [**Secure Web Gateway**] > [プロキシ仮想サーバー] > [追加] に移動します。名前、IP アドレス、およびポートを指定します。[詳細設定] で、[ポリシー] を選択します。「+」記号をクリックします。



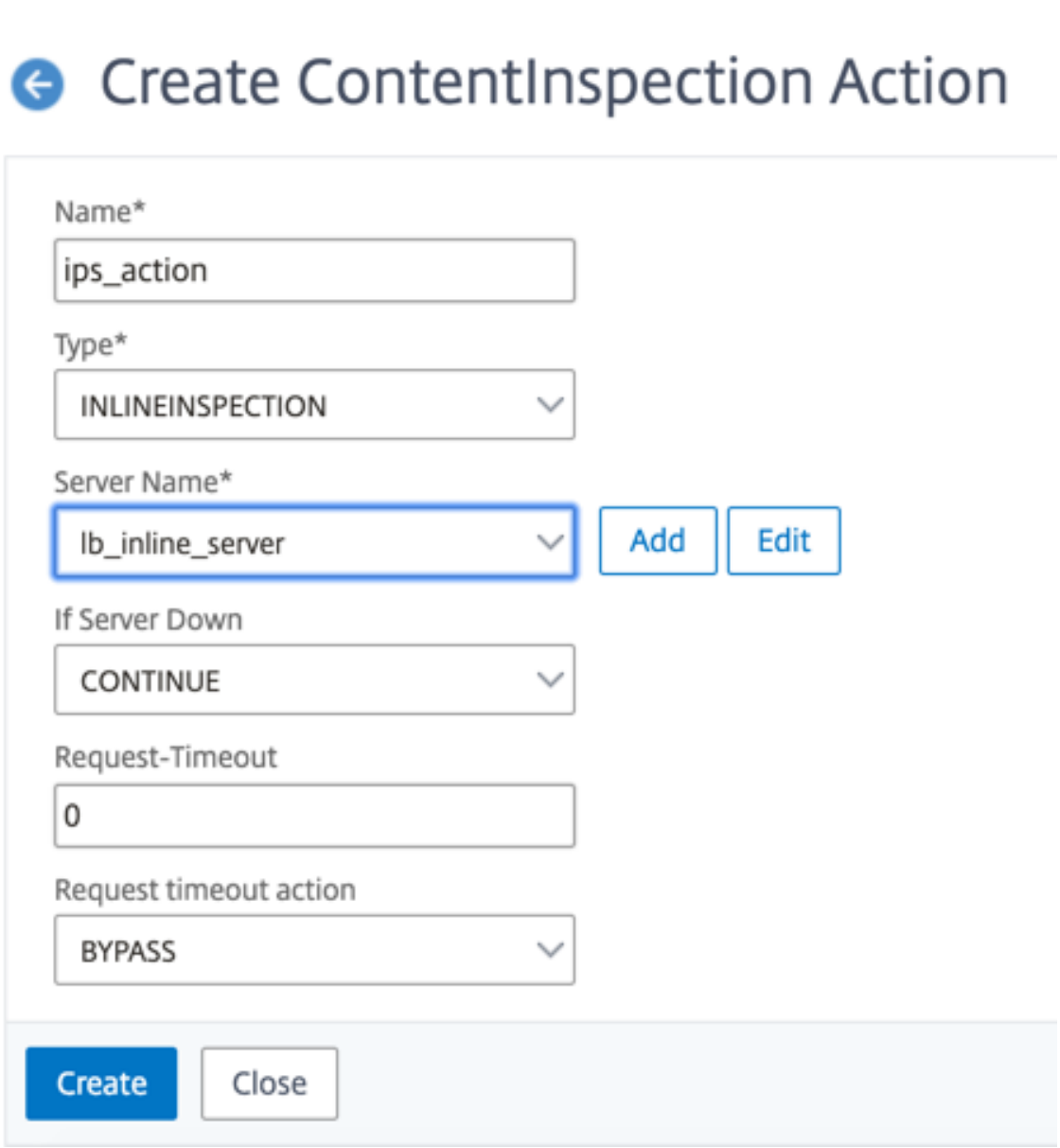
10. [ポリシーの選択] で [コンテンツ検査] を選択します。[続行] をクリックします。



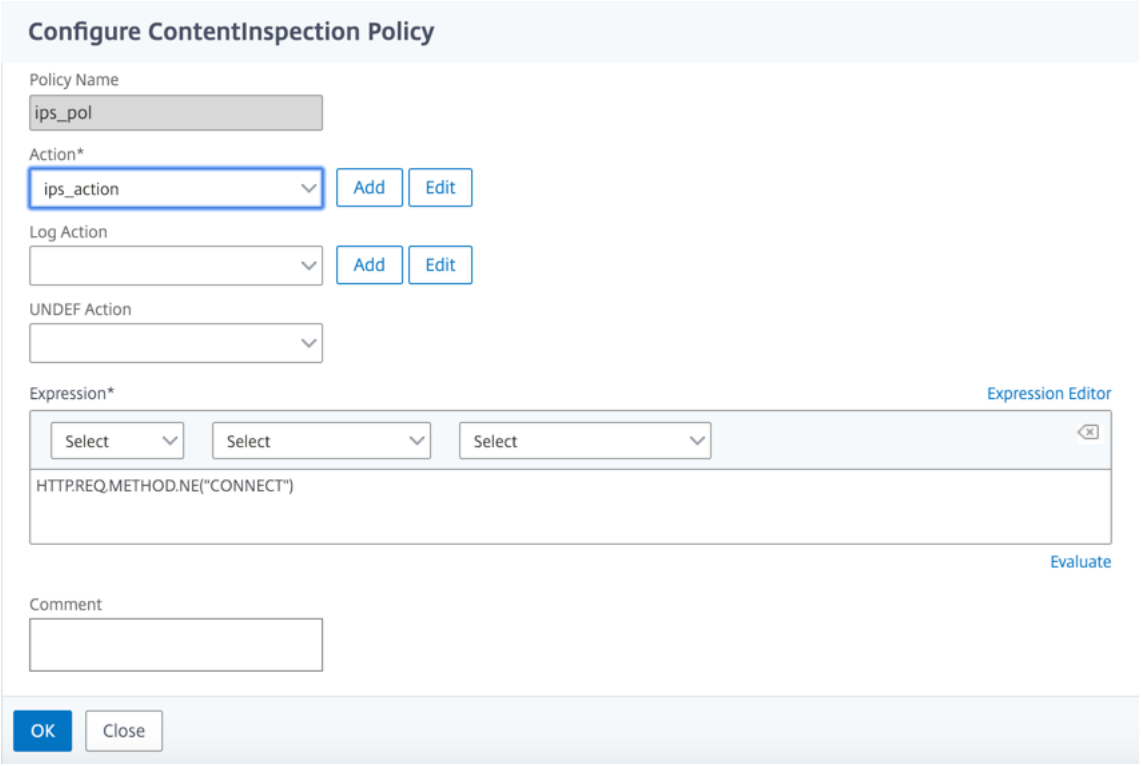
11. ［追加］をクリックします。名前を指定します。「アクション」で、「追加」をクリックします。



12. 名前を指定します。「タイプ」で「**INLINEINSPECTION**」を選択します。「サーバー名」で、以前に作成した負荷分散仮想サーバーを選択します。



13. ［作成］をクリックします。ルールを指定し、［**Create**］をクリックします。



14. ［バインド］をクリックします。
15. ［完了］をクリックします。

Citrix ADC とパッシブセキュリティデバイスの統合（侵入検知システム）

January 25, 2022

Citrix ADC アプライアンスは、侵入検知システム（IDS）などのパッシブセキュリティデバイスと統合されました。これらのパッシブデバイスはログを保存し、不良または非準拠のトラフィックを検出するとアラートをトリガーします。また、コンプライアンスのためのレポートも生成します。Citrix ADC アプライアンスが 2 つ以上の IDS デバイスに統合されており、トラフィックが多い場合、アプライアンスは仮想サーバーレベルでトラフィックのクローンを作成することでデバイスの負荷を分散できます。

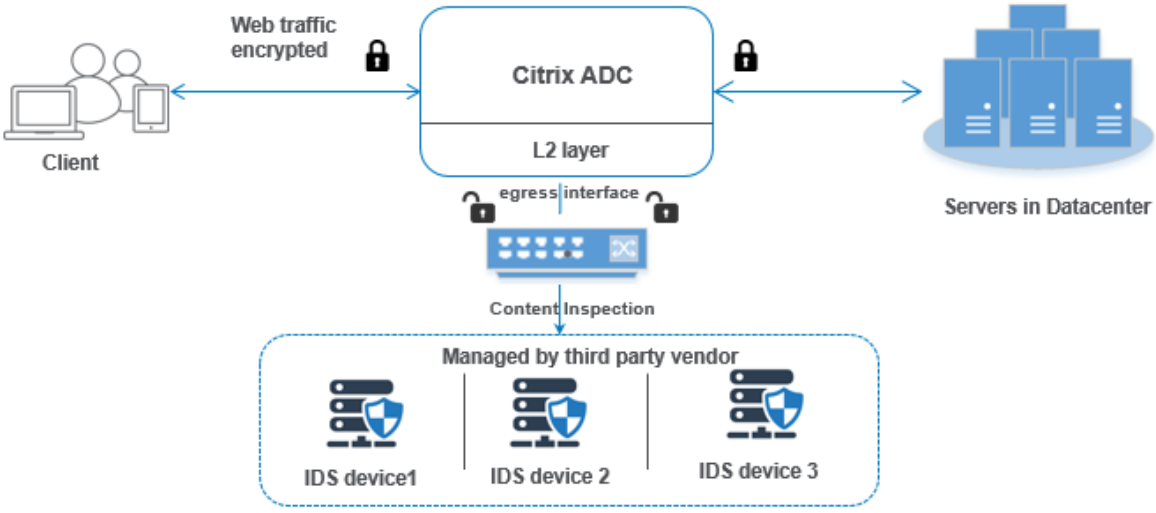
高度なセキュリティ保護のために、Citrix ADC アプライアンスは、検出専用モードで展開された IDS などのパッシブセキュリティデバイスと統合されています。これらのデバイスはログを保存し、不良または非準拠のトラフィックを検出するとアラートをトリガーします。また、コンプライアンスのためのレポートも生成します。Citrix ADC を IDS デバイスと統合する利点の一部を次に示します。

- 暗号化されたトラフィックを検査する。ほとんどのセキュリティデバイスは暗号化されたトラフィックをバイパスするため、サーバは攻撃に対して脆弱になります。Citrix ADC アプライアンスは、トラフィックを復号化して IDS デバイスに送信し、顧客のネットワークセキュリティを強化できます。

- インラインデバイスを **TLS/SSL** 処理からオフロードする。TLS/SSL の処理にはコストがかかり、侵入検知デバイスがトラフィックを復号化すると、システム CPU が高くなります。暗号化されたトラフィックが急速に増加するにつれて、これらのシステムは暗号化されたトラフィックの復号化と検査に失敗します。Citrix ADC は、TLS/SSL 処理から IDS デバイスへのトラフィックをオフロードするのに役立ちます。この方法でデータをオフロードすると、IDS デバイスは大量のトラフィックインスペクションをサポートすることになります。
- **IDS** デバイスの負荷分散 Citrix ADC アプライアンスは、トラフィックが多い場合、仮想サーバーレベルでトラフィックのクローンを作成することにより、複数の IDS デバイスの負荷を分散します。
- トラフィックをパッシブデバイスに複製する。アプライアンスに流入するトラフィックは、コンプライアンスレポートを生成するために、他のパッシブデバイスに複製できます。たとえば、一部のパッシブデバイスにすべてのトランザクションを記録するよう義務付けている政府機関はほとんどありません。
- トラフィックを複数のパッシブデバイスにファンニングする。一部のお客様は、着信トラフィックを複数のパッシブデバイスにファンアウトまたは複製することを好みます。
- トラフィックのスマートな選択。アプライアンスに流入するすべてのパケットは、テキストファイルのダウンロードなど、内容を検査する必要がない場合があります。ユーザーは、特定のトラフィック（.exe ファイルなど）を検査用に選択し、そのトラフィックを IDS デバイスに送信してデータを処理するように Citrix ADC アプライアンスを構成できます。

Citrix ADC が **L2** 接続を備えた **IDS** デバイスとどのように統合されるか

次の図は、IDS が Citrix ADC アプライアンスとどのように統合されるかを示しています。



コンポーネントの相互作用は次のように与えられます。

1. クライアントは、HTTP/HTTPS 要求を Citrix ADC アプライアンスに送信します。
2. アプライアンスはトラフィックをインターセプトし、コンテンツインスペクションポリシーの評価に基づいて IDS デバイスに複製します。

3. トラフィックが暗号化されたものである場合、アプライアンスはデータを復号化し、プレーンテキストとして送信します。
4. ポリシー評価に基づいて、アプライアンスは「MIRROR」タイプのコンテンツ検査アクションを適用します。
5. アクションには、IDS サービスまたは負荷分散サービス (複数の IDS デバイス統合用) が設定されています。
6. IDS デバイスは、アプライアンス上でコンテンツ検査サービスタイプ「Any」として設定されています。コンテンツ検査サービスは、IDS デバイスにデータを転送する必要がある出力インターフェイスを指定する「MIRROR」タイプのコンテンツ検査プロファイルに関連付けられます。オプションで、コンテンツ検査プロファイルに VLAN タグを設定することもできます。

   > 注:
   >
   > - IDS サービスまたはサーバに使用される IP アドレスはダミーアドレスです。
   > - Citrix ADC アプライアンスは、出力インターフェイスで LA チャネルをサポートしていません。

7. その後、アプライアンスは出力インターフェイスを介して 1 つ以上の IDS デバイスにデータを複製します。
8. 同様に、バックエンドサーバーが Citrix ADC に応答を送信すると、アプライアンスはデータを複製して IDS デバイスに転送します。
9. アプライアンスが 1 つ以上の IDS デバイスに統合されていて、デバイスの負荷分散を希望する場合は、負荷分散仮想サーバを使用できます。

ソフトウェアライセンス

インラインデバイス統合を展開するには、Citrix ADC アプライアンスに次のいずれかのライセンスをプロビジョニングする必要があります。

1. ADC Premium
2. ADC Advanced
3. Telco Advanced
4. Telco プレミアム

侵入検知システム統合の設定

IDS デバイスと Citrix ADC は、2 つの異なる方法で統合できます。

シナリオ **1:** 単一の **IDS** デバイスとの統合

コマンドラインインターフェイスを使用して設定する必要がある手順を次に示します。

1. コンテンツ検査を有効にする
2. IDS デバイスを表すサービス用に、MIRROR タイプのコンテンツ検査プロファイルを追加します。
3. タイプ「ANY」の IDS サービスを追加する

4. タイプ「MIRROR」のコンテンツ検査アクションを追加する
5. IDS 検査のコンテンツ検査ポリシーを追加する
6. コンテンツ検査ポリシーを HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする

コンテンツ検査を有効にする

Citrix ADC アプライアンスが検査のためにコンテンツを IDS デバイスに送信するようにするには、復号化の実行に関係なく、コンテンツ検査と負荷分散機能を有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns feature contentInspection LoadBalancing
```

タイプ「**MIRROR**」のコンテンツ検査プロファイルの追加

「MIRROR」タイプのコンテンツ検査プロファイルは、IDS デバイスへの接続方法を説明しています。
コマンドプロンプトで、次のように入力します。

```
add contentInspection profile <name> -type MIRROR -egressInterface <interface_name
> [-egressVlan <positive_integer>]
```

例**:**

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 10
```

IDS サービスの追加

アプライアンスと統合されている IDS デバイスごとに、「ANY」タイプのサービスを設定する必要があります。このサービスには IDS デバイス設定の詳細が含まれています。このサービスは IDS デバイスを表します。

コマンドプロンプトで入力します。

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName <
Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
IDS_profile1 -healthMonitor OFF
```

IDS サービスの **MIRROR** タイプのコンテンツ検査アクションを追加する

コンテンツ検査機能を有効にして IDS プロファイルとサービスを追加したら、要求を処理するための Content Inspection アクションを追加する必要があります。コンテンツ検査アクションに基づいて、アプライアンスは IDS デバイスにデータをドロップ、リセット、ブロック、または送信できます。

コマンドプロンプトで入力します。

```
add ContentInspection action < action_name > -type MIRROR -serverName
Service_name/Vserver_name>
```

例:

```
add ContentInspection action IDS_action -type MIRROR -serverName IDS_service
```

IDS 検査のコンテンツ検査ポリシーを追加する

コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを追加して検査の要求を評価する必要があります。このポリシーは、1 つ以上の式で構成されるルールに基づいています。ポリシーは、ルールに基づいてインスペクション対象のトラフィックを評価し、選択します。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy < policy_name > -rule <Rule> -action <action_name
>
```

例:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする

Web トラフィックを受信するには、負荷分散仮想サーバーを追加する必要があります。
コマンドプロンプトで入力します。

```
add lb vserver <name> <vserver name>
```

例:

```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーにバインドする

HTTP/SSL タイプの負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーをコンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します。

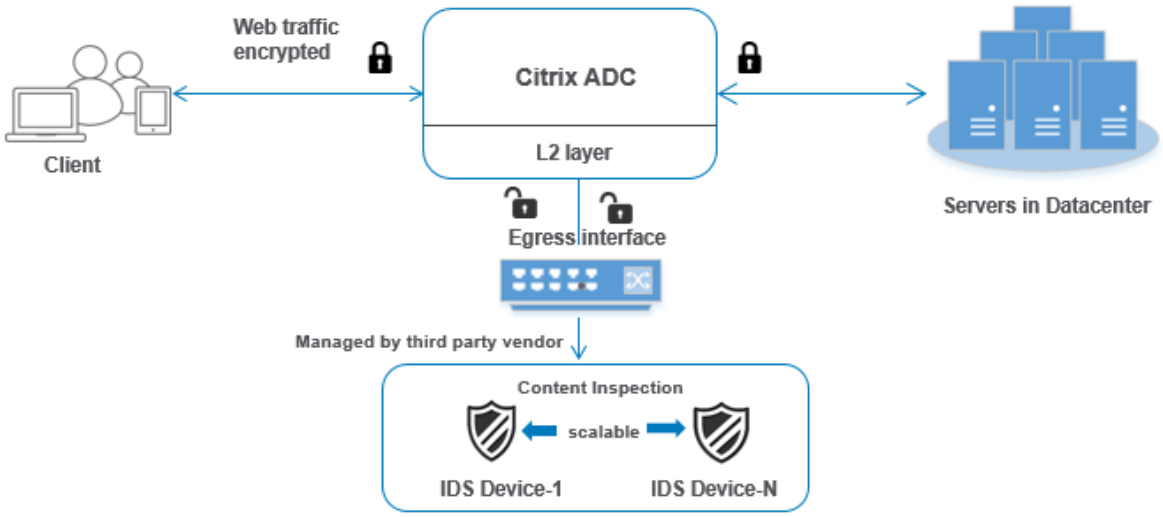
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

例:


```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

シナリオ **2:** 複数の **IDS** デバイスの負荷分散

2 つ以上の IDS デバイスを使用している場合は、異なるコンテンツ検査サービスを使用してデバイスの負荷分散を行う必要があります。この場合、Citrix ADC アプライアンスは、トラフィックのサブセットを各デバイスに送信するだけでなく、デバイスの負荷を分散します。
基本的な設定手順については、シナリオ 1 を参照してください。



コマンドラインインターフェイスを使用して設定する必要がある手順を次に示します。

1. IDS サービス 1 の MIRROR タイプのコンテンツ検査プロファイル 1 を追加します。
2. IDS サービス 2 の MIRROR タイプのコンテンツ検査プロファイル 2 を追加する
3. IDS デバイス 1 にタイプ ANY の IDS サービス 1 を追加する
4. IDS デバイス 2 にタイプ ANY の IDS サービス 2 を追加する
5. ANY タイプの負荷分散仮想サーバを追加する
6. IDS サービス 1 を負荷分散仮想サーバーにバインドする
7. IDS サービス 2 を負荷分散仮想サーバーにバインドする
8. IDS デバイスの負荷分散のためのコンテンツ検査アクションを追加します。
9. 検査用のコンテンツ検査ポリシーを追加する
10. HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する
11. コンテンツ検査ポリシーを HTTP/SSL タイプの負荷分散仮想サーバーにバインドする

IDS サービス **1** の **MIRROR** タイプのコンテンツ検査プロファイル **1** を追加します

IDS 設定は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルにはデバイス設定のコレクションがあります。コンテンツ検査プロファイル 1 が IDS サービス 1 用に作成されます。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type ANY -egressInterface <interface_name
> [-egressVlan <positive_integer>]
```

例:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 1
```

IDS サービス **2** のタイプ **MIRROR** のコンテンツ検査プロファイル **2** を追加する

サービス 2 にはコンテンツ検査プロファイル 2 が追加され、インラインデバイスは出力 1/1 インターフェイスを介してアプライアンスと通信します。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type MIRROR -egressInterface -egressVlan
 <positive_integer>]
```

例:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 1
```

IDS デバイス **1** にタイプ **ANY** の **IDS** サービス **1** を追加する

コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 1 のインラインサービス 1 を負荷分散設定の一部として追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します。

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileName
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

例:

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
IDS_profile1 -usip ON -useproxyport OFF
```

> 注
>
> この例で示されている IP アドレスはダミーの IP アドレスです。

IDS デバイス **2** にタイプ **ANY** の **IDS** サービス **2** を追加する

コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 2 にインラインサービス 2 を追加する必要があります。追加したサービスによって、インライン構成の詳細がすべて提供されます。

コマンドプロンプトで入力します。

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName <
Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```

> 注
>
> この例で示されている IP アドレスはダミーの IP アドレスです。

負荷分散仮想サーバの追加

インラインプロファイルとサービスを追加したら、サービスの負荷分散用の負荷分散仮想サーバを追加する必要があります。

コマンドプロンプトで入力します。

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

例:

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

IDS サービス **1** を負荷分散仮想サーバーにバインドする

負荷分散仮想サーバーを追加したら、負荷分散仮想サーバーを最初のサービスにバインドします。

コマンドプロンプトで入力します。

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service1
```

IDS サービス **2** を負荷分散仮想サーバーにバインドする

負荷分散仮想サーバーを追加したら、そのサーバーを 2 番目のサービスにバインドします。

コマンドプロンプトで入力します。

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service2
```


IDS サービスのコンテンツ検査アクションを追加する

コンテンツ検査機能を有効にしたら、インラインリクエスト情報を処理するための「コンテンツ検査」アクションを追加する必要があります。選択したアクションに基づいて、アプライアンスは IDS デバイスへのトラフィックをドロップ、リセット、ブロック、または送信します。

コマンドプロンプトで入力します。

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>]
```

例**:**

```
add ContentInspection action IDS_action -type MIRROR -serverName lb-IDS_vserver
```

検査用のコンテンツ検査ポリシーを追加する

コンテンツ検査アクションを作成したら、サービス要求を評価するコンテンツ検査ポリシーを追加する必要があります。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

例**:**

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを追加する

Web トラフィックを受け入れるために、コンテンツスイッチングまたは負荷分散仮想サーバーを追加します。また、仮想サーバ上で layer2 接続を有効にする必要があります。

負荷分散の詳細については、「負荷分散の仕組み」トピックを参照してください。

コマンドプロンプトで入力します。

```
add lb vserver <name> <vserver name>
```

例**:**

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプの負荷分散仮想サーバーにバインドする

HTTP/SSL タイプのコンテンツスイッチング仮想サーバーまたは負荷分散仮想サーバーをコンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します。

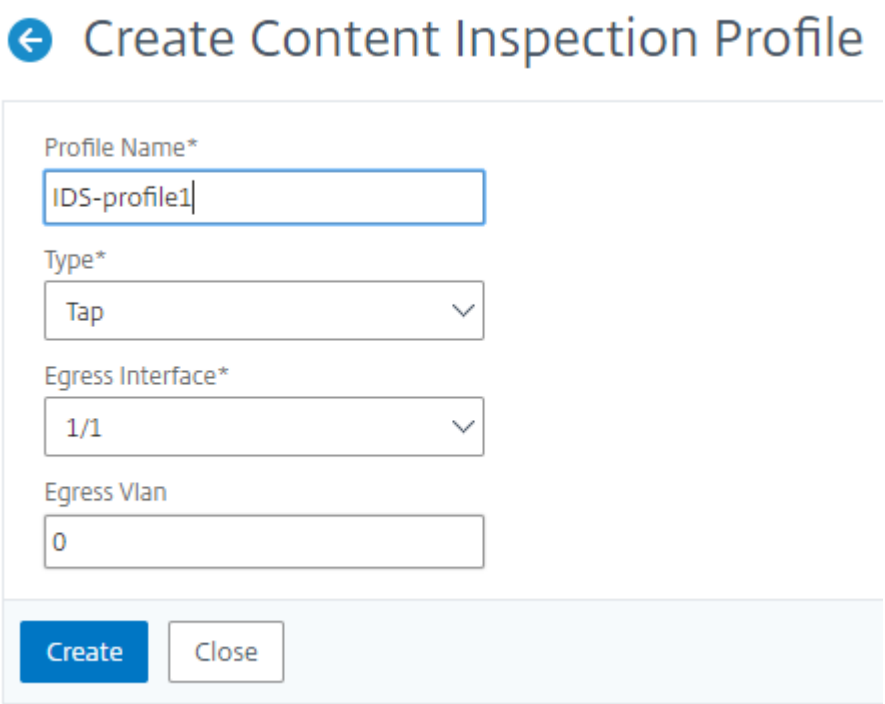
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <REQUEST>
```

例:

```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

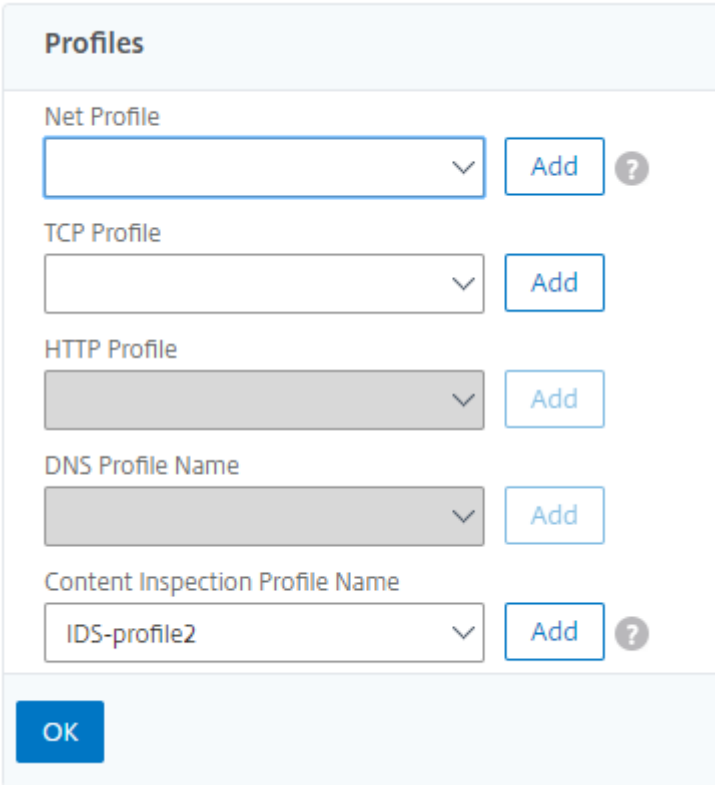
Citrix ADC GUI を使用してインラインサービス統合を構成する

1. [セキュリティ] > [コンテンツ検査] > [コンテンツ検査プロファイル] に移動します。
2. [コンテンツ検査プロファイル] ページで、[追加] をクリックします。
3. **[**コンテンツ検査プロファイルの作成] ページで、次のパラメータを設定します。
   a) プロファイル名。IDS のコンテンツ検査プロファイルの名前。
   b) [タイプ]。プロファイルタイプを MIRROR として選択します。
   c) 出力インターフェイス。Citrix ADC から IDS デバイスにトラフィックが送信されるインターフェイス。
   d) 出力 VLAN（任意）トラフィックが IDS デバイスに送信される際のインターフェイス VLAN ID。
4. ［作成］をクリックします。



5. トラフィック管理 > 負荷分散 > サービスに移動し、追加をクリックします。

6. [負荷分散サービス] ページで、コンテンツ検査サービスの詳細を入力します。
7. **[**詳細設定**]** セクションで、[プロファイル] をクリックします。
8. [プロファイル] セクションに移動し、[鉛筆] アイコンをクリックしてコンテンツ検査プロファイルを追加します。
9. [**OK**] をクリックします。



10. [負荷分散] > [サーバー] に移動します。HTTP または SSL タイプの仮想サーバを追加します。
11. サーバーの詳細を入力したら、「**OK**」をクリックし、もう一度「**OK**」をクリックします。
12. **[**詳細設定**]** セクションで、[ポリシー] をクリックします。
13. [ポリシー] セクションに移動し、[鉛筆] アイコンをクリックしてコンテンツ検査ポリシーを設定します。
14. **[**ポリシーの選択**]** ページで、[コンテンツ検査] を選択します。[続行] をクリックします。
15. [ポリシーバインド] セクションで、[+] をクリックしてコンテンツ検査ポリシーを追加します。
16. [**CI** ポリシーの作成] ページで、インラインコンテンツ検査ポリシーの名前を入力します。
17. [**Action**] フィールドで [+] 記号をクリックし、MIRROR タイプの IDS コンテンツ検査アクションを作成します。
18. **[CI** アクションの作成] ページで、次のパラメータを設定します。

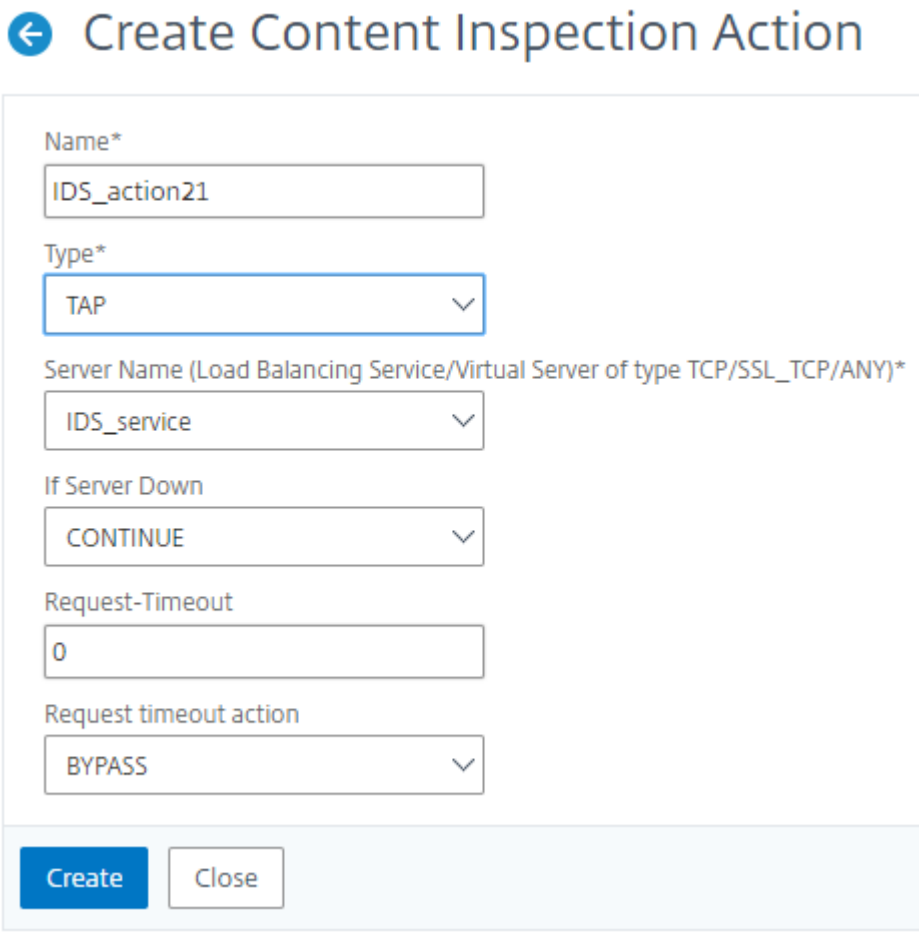
    a. Name：コンテンツ検査インラインポリシーの名前。

b. タイプ。タイプとして MIRROR を選択します。
c. サーバー名：サーバ/サービス名を [インラインデバイス] として選択します。

d. サーバーがダウンした場合サーバがダウンした場合のオペレーションを選択します。
e. リクエストのタイムアウト。タイムアウト値を選択します。デフォルト値を使用できます。

f. タイムアウトアクションの要求。タイムアウトアクションを選択します。デフォルト値を使用できます。

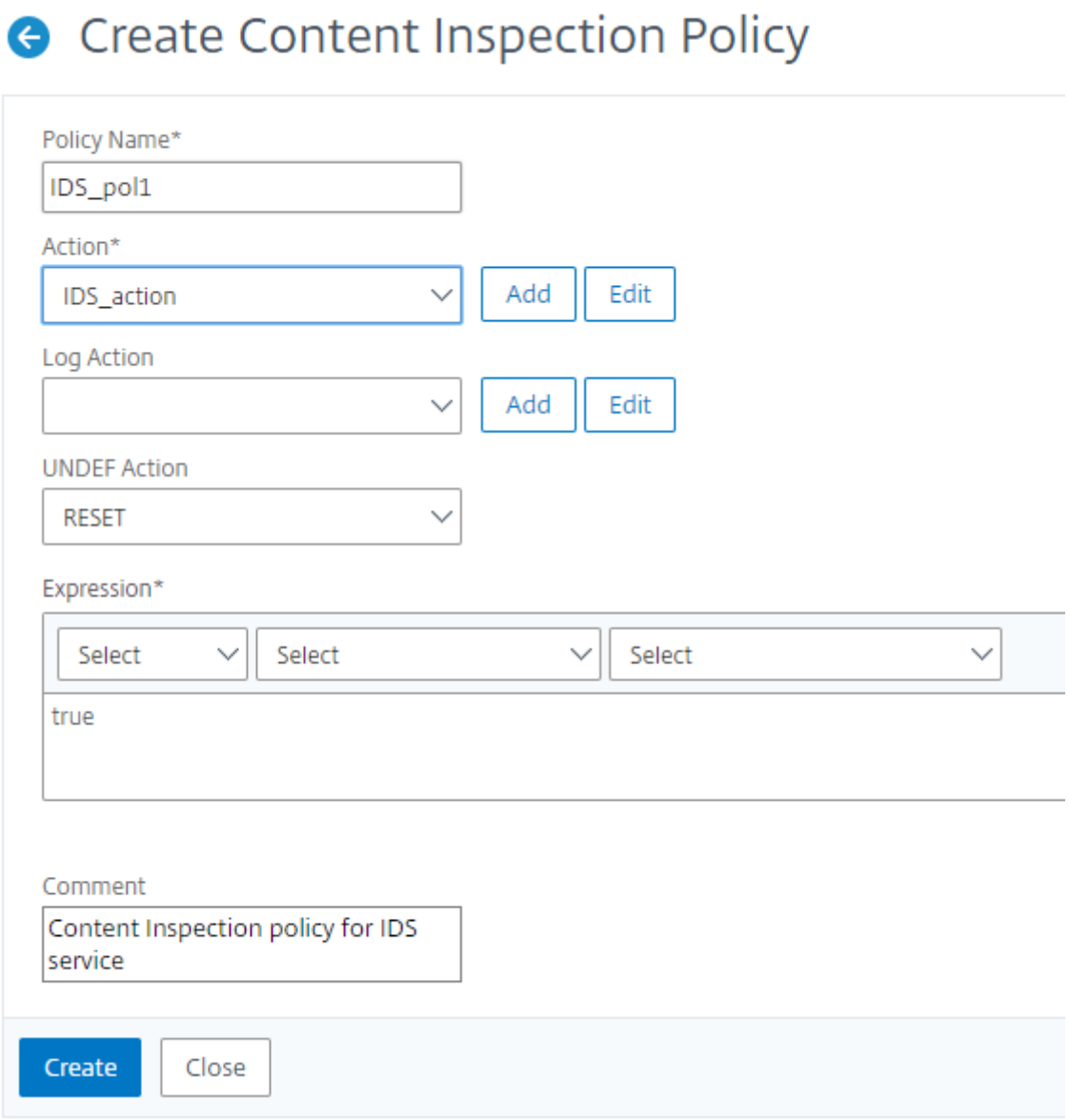
19. ［作成］をクリックします。



20. **[CI** ポリシーの作成] ページで、その他の詳細を入力します。

21. ［**OK**］をクリックして閉じます。

負荷分散および IDS デバイスへのトラフィックのレプリケーションに関する Citrix ADC GUI 構成の詳細については、「負荷分散」を参照してください。



コンテンツ変換後にトラフィックをバックエンドオリジンサーバーに負荷分散および転送するための Citrix ADC GUI 構成の詳細については、「負荷分散 」を参照してください。

Citrix ADC レイヤ 3 とパッシブセキュリティデバイス（侵入検知システム）の統合

October 7, 2021

Citrix ADC アプライアンスは、侵入検知システム（IDS）などのパッシブセキュリティデバイスと統合されました。この設定では、アプライアンスは元のトラフィックのコピーをリモート IDS デバイスに安全に送信します。これらのパッシブデバイスはログを保存し、不良または非準拠のトラフィックを検出するとアラートをトリガーします。また、コンプライアンスの目的でレポートを生成します。Citrix ADC アプライアンスが 2 つ以上の IDS デバイスと統合さ

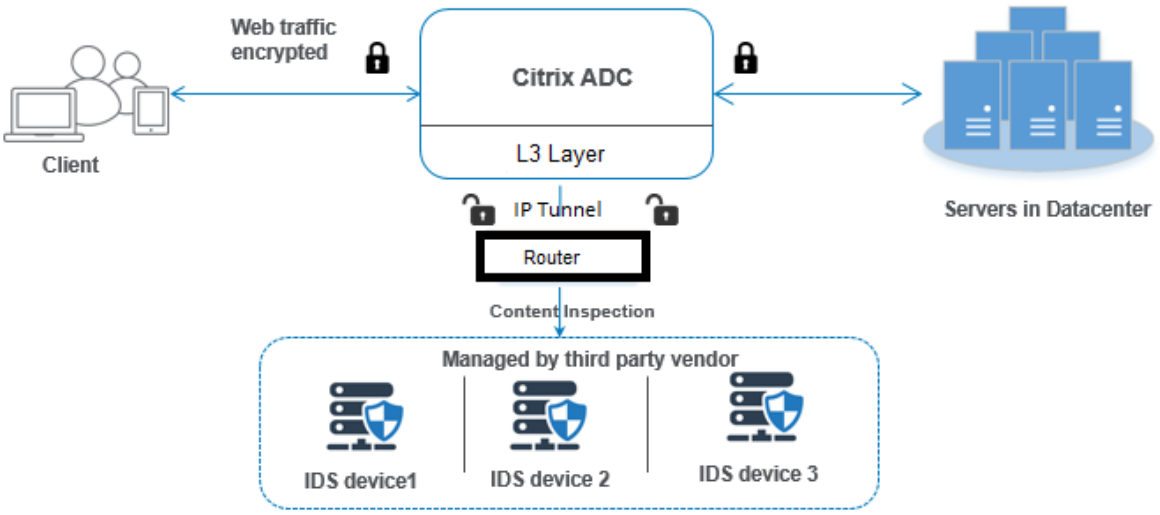
れており、トラフィックが大量にある場合、アプライアンスは仮想サーバーレベルでトラフィックを複製することにより、デバイスの負荷を分散できます。

高度なセキュリティ保護のために、Citrix ADC アプライアンスは、検出専用モードで展開された IDS などのパッシブセキュリティデバイスと統合されています。これらのデバイスは、ログを保存し、不良または非準拠のトラフィックを検出したときにアラートをトリガーします。また、コンプライアンスの目的でレポートを生成します。以下は、Citrix ADC を IDS デバイスと統合することの利点の一部です。

- 暗号化されたトラフィックの検査。ほとんどのセキュリティデバイスは暗号化されたトラフィックをバイパスするため、サーバは攻撃に対して脆弱になります。Citrix ADC アプライアンスは、トラフィックを復号化して IDS デバイスに送信し、顧客のネットワークセキュリティを強化できます。
- **TLS/SSL** 処理からのインラインデバイスのオフロード。TLS/SSL 処理はコストが高く、侵入検知デバイスでトラフィックを復号化すると、システム CPU が高くなります。暗号化されたトラフィックが急速に増加するにつれて、これらのシステムは暗号化されたトラフィックの復号化と検査に失敗します。Citrix ADC は、TLS/SSL 処理から IDS デバイスへのトラフィックのオフロードに役立ちます。このようにデータのオフロードを行うと、IDS デバイスは大量のトラフィックインスペクションをサポートします。
- **IDS** デバイスの負荷分散。Citrix ADC アプライアンスは、トラフィック量が多い場合に、仮想サーバーレベルでトラフィックのクローンを作成することで、複数の IDS デバイスの負荷分散を行います。
- パッシブデバイスへのトラフィックの複製。アプライアンスに流れるトラフィックは、コンプライアンスレポートを生成するために、他のパッシブデバイスに複製できます。たとえば、一部のパッシブデバイスにすべてのトランザクションをログに記録することを義務付けている政府機関はほとんどありません。
- 複数のパッシブデバイスへのトラフィックのファン。一部のお客様は、着信トラフィックを複数のパッシブデバイスにファンアウトまたは複製することを好みます。
- トラフィックのスマートな選択。アプライアンスに流入するすべてのパケットは、テキストファイルのダウンロードなど、コンテンツを検査する必要がない場合があります。ユーザーは、検査対象の特定のトラフィック（たとえば、.exe ファイル）を選択し、データを処理するために IDS デバイスにトラフィックを送信するように Citrix ADC アプライアンスを設定できます。

Citrix ADC と **L3** 接続を備えた **IDS** デバイスとの統合方法

次の図は、IDS が Citrix ADC アプライアンスとどのように統合されているかを示しています。



コンポーネントの相互作用は、次のように与えられます。

1. クライアントは HTTP/HTTPS Citrix ADC アプライアンスへのリクエスト。
2. アプライアンスはトラフィックを傍受し、異なるデータセンターまたはクラウド内のリモート IDS デバイスにデータを送信します。この統合は、IP トンネリングされたレイヤ 3 を介して行われます。Citrix ADC アプライアンスでの IP トンネリングの詳細については、「IP トンネルのトピック」を参照してください。
3. トラフィックが暗号化されたトラフィックの場合、アプライアンスはデータを復号化し、プレーンテキストとして送信します。
4. ポリシー評価に基づいて、アプライアンスは「MIRROR」タイプのコンテンツ検査アクションを適用します。
5. アクションには、IDS サービスまたは負荷分散サービス（複数の IDS デバイス統合用）が構成されています。
6. IDS デバイスは、アプライアンス上でコンテンツ検査サービスタイプ「Any」として設定されます。次に、コンテンツ検査サービスは、タイプ「MIRROR」のコンテンツ検査プロファイルと、データが IDS デバイスに転送される IP トンネルレイヤー 3 インターフェイスを指定するトンネルパラメーターに関連付けられます。

> （注）オプションで、コンテンツ検査プロファイルで VLAN タグを設定することもできます。

1. 同様に、バックエンドサーバーが Citrix ADC に応答を送信すると、アプライアンスはデータを複製して IDS デバイスに転送します。
2. アプライアンスが 1 つ以上の IDS デバイスに統合されており、デバイスの負荷分散を希望する場合は、負荷分散仮想サーバを使用できます。

ソフトウェアライセンス

IDS 統合を展開するには、Citrix ADC アプライアンスに次のいずれかのライセンスをプロビジョニングする必要があります。

1. ADC Premium
2. ADC Advanced

侵入検知システム統合の設定

IDS デバイスを Citrix ADC と統合するには、2 つの方法があります。

シナリオ **1**: 単一の **IDS** デバイスとの統合

コマンドラインインターフェイスを使用して設定する必要がある手順は次のとおりです。

1. コンテンツ検査を有効にする
2. IDS デバイスを表すサービスに、タイプ MIRROR のコンテンツ検査プロファイルを追加します。
3. タイプ「ANY」の IDS サービスを追加する
4. タイプ「MIRROR」のコンテンツ検査アクションを追加
5. IDS 検査のコンテンツ検査ポリシーの追加
6. コンテンツ検査ポリシーを HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする

コンテンツ検査を有効にする

Citrix ADC アプライアンスが検査用のコンテンツを IDS デバイスに送信するようにする場合は、復号化の実行に関係なく、コンテンツ検査および負荷分散機能を有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns feature contentInspection LoadBalancing
```

タイプ「**MIRROR**」のコンテンツ検査プロファイルを追加

タイプ「MIRROR」のコンテンツ検査プロファイルでは、IDS デバイスへの接続方法が説明されています。
コマンドプロンプトで、次のように入力します。

> (注
>) IP トンネルパラメータは、レイヤ 3IDS トポロジにのみ使用する必要があります。それ以外の場合は、出力 VLAN オプションを指定して出力インターフェイスを使用する必要があります。

```
add contentInspection profile <name> -type MIRROR -ipTunnel <iptunnel_name>
```

例**:**

```
add contentInspection profile IDS_profile1 -type MIRROR -ipTunnel ipsect-
tunnel1
```

IDS サービスの追加

アプライアンスと統合された IDS デバイスごとに、タイプ「ANY」のサービスを設定する必要があります。サービスには IDS デバイス設定の詳細が含まれています。サービスは IDS デバイスを表します。

コマンドプロンプトで入力します。

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName <
Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
IDS_profile1 -healthMonitor OFF
```

IDS サービスのタイプ **MIRROR** のコンテンツ検査アクションを追加

コンテンツ検査機能を有効にしてから IDS プロファイルとサービスを追加したら、要求を処理するための Content Inspection アクションを追加する必要があります。コンテンツ検査アクションに基づいて、アプライアンスはデータをドロップ、リセット、ブロック、または IDS デバイスに送信できます。

コマンドプロンプトで入力します。

```
add ContentInspection action < action_name > -type MIRROR -serverName
Service_name/Vserver_name>
```

例:

```
add ContentInspection action IDS_action -type MIRROR -serverName IDS_service
```

IDS 検査のコンテンツ検査ポリシーの追加

コンテンツ検査アクションを作成したら、コンテンツ検査ポリシーを追加して、検査要求を評価する必要があります。ポリシーは、1 つ以上の式で構成される規則に基づいています。ポリシーは、ルールに基づいて検査対象のトラフィックを評価し、選択します。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy < policy_name > -rule <Rule> -action <action_name
>
```

例:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプのコンテンツスイッチングまたは負荷分散仮想サービスにバインドする

Web トラフィックを受信するには、負荷分散仮想サーバーを追加する必要があります。
コマンドプロンプトで入力します。

```
add lb vserver <name> <vserver name>
```

例:


```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

コンテンツ検査ポリシーをコンテンツスイッチング仮想サーバーまたは **HTTP/SSL** タイプの負荷分散仮想サーバーにバインド

負荷分散仮想サーバーまたは HTTP/SSL タイプのコンテンツスイッチング仮想サーバーを、コンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します。

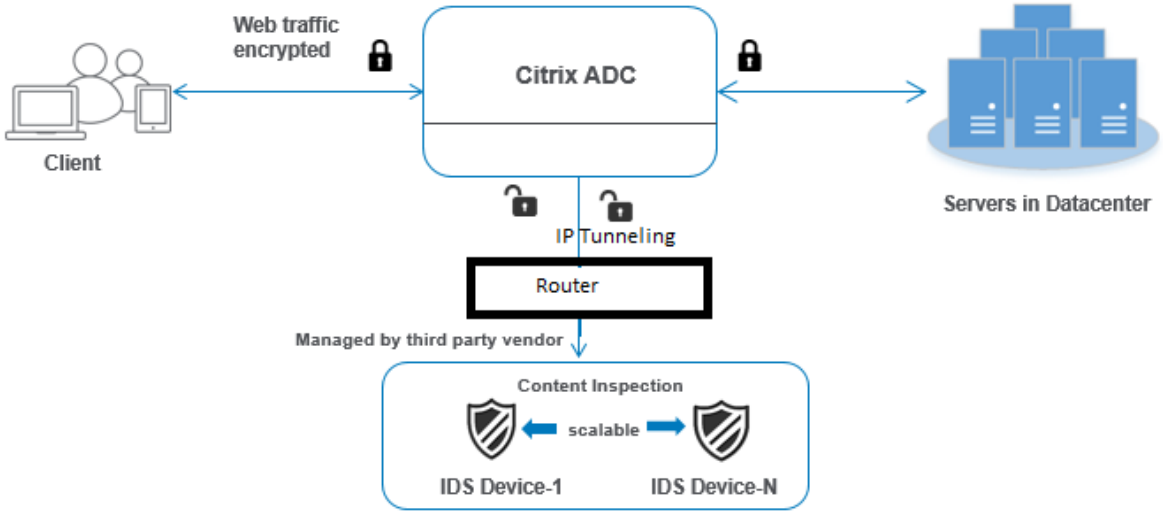
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

例:

```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

シナリオ **2**: 複数の **IDS** デバイスの負荷分散

2 つ以上の IDS デバイスを使用している場合は、異なるコンテンツ検査サービスを使用して IDS デバイスの負荷分散を行う必要があります。この場合、Citrix ADC アプライアンスは、各デバイスにトラフィックのサブセットを送信する上でデバイスの負荷分散を行います。
基本的な設定手順については、シナリオ 1 を参照してください。



コマンドラインインターフェイスを使用して設定する必要がある手順は次のとおりです。

1. IDS サービス 1 のタイプ MIRROR のコンテンツ検査プロファイル 1 を追加します
2. IDS サービス 2 のタイプ MIRROR のコンテンツ検査プロファイル 2 を追加します
3. IDS デバイス 1 に ANY タイプの IDS サービス 1 を追加します

4. IDS デバイス 2 に ANY タイプの IDS サービス 2 を追加します
5. タイプ ANY の負荷分散仮想サーバーを追加する
6. IDS サービス 1 を負荷分散仮想サーバにバインド
7. IDS サービス 2 を負荷分散仮想サーバにバインド
8. IDS デバイスの負荷分散にコンテンツインスペクションアクションを追加します。
9. 検査用のコンテンツ検査ポリシーの追加
10. HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーの追加
11. コンテンツ検査ポリシーを HTTP/SSL タイプの仮想サーバーの負荷分散にバインド

IDS サービス **1** のタイプ **MIRROR** のコンテンツ検査プロファイル **1** を追加します

IDS 構成は、コンテンツ検査プロファイルと呼ばれるエンティティで指定できます。プロファイルには、デバイス設定のコレクションがあります。IDS サービス 1 用にコンテンツ検査プロファイル 1 が作成されます。

> 注:
> IP トンネルパラメータは、レイヤ 3 IDS トポロジにだけ使用する必要があります。それ以外の場合は、出力 VLAN オプションを指定して出力インターフェイスを使用する必要があります。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name>
```

例**:**

```
add contentInspection profile IDS_profile1 -type MIRROR - ipTunnel ipsect_tunnel1
```

IDS サービス **2** のタイプ **MIRROR** にコンテンツ検査プロファイル **2** を追加します

コンテンツ検査プロファイル 2 がサービス 2 に追加され、インラインデバイスが出力を介してアプライアンスと通信します。1/1 インターフェース。

コマンドプロンプトで入力します。

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name>
```

例**:**

```
add contentInspection profile IDS_profile2 -type ANY - ipTunnel ipsect_tunnel2
```

IDS デバイス **1** に **ANY** タイプの **IDS** サービス **1** を追加します

コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 1 のインラインサービス 1 を負荷分散設定の一部として追加する必要があります。追加するサービスは、すべてのインライン構成の詳細を提供します。

コマンドプロンプトで入力します。

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileName
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

例:

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
IDS_profile1 -usip ON -useproxyport OFF
```

> 注:
>
> 例に示されている IP アドレスは、ダミーのアドレスです。

IDS デバイス **2** に **ANY** タイプの **IDS** サービス **2** を追加します

コンテンツ検査機能を有効にしてインラインプロファイルを追加したら、インラインデバイス 2 のインラインサービス 2 を追加する必要があります。追加するサービスは、すべてのインライン構成の詳細を提供します。

コマンドプロンプトで入力します。

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName <
Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

例:

```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```

> 注:
>
> 例に示されている IP アドレスは、ダミーのアドレスです。

負荷分散仮想サーバの追加

インラインプロファイルおよびサービスを追加したら、サービスの負荷分散用の負荷分散仮想サーバーを追加する必要があります。

コマンドプロンプトで入力します。

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

例:

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

IDS サービス **1** を負荷分散仮想サーバにバインド

負荷分散仮想サーバーを追加した後、負荷分散仮想サーバーを最初のサービスにバインドします。

コマンドプロンプトで入力します。

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service1
```

IDS サービス **2** を負荷分散仮想サーバにバインド

負荷分散仮想サーバーを追加した後、サーバーを 2 番目のサービスにバインドします。

コマンドプロンプトで入力します。

```
bind lb vserver <Vserver_name> <Service_name_1>
```

例:

```
bind lb vserver lb-IDS_vserver IDS_service2
```

IDS サービスのコンテンツ検査アクションを追加する

コンテンツ検査機能を有効にしたら、インライン要求情報を処理するための Content Inspection アクションを追加する必要があります。選択したアクションに基づいて、アプライアンスはトラフィックをドロップ、リセット、ブロック、または IDS デバイスに送信します。

コマンドプロンプトで入力します。

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>]
```

例:

```
add ContentInspection action IDS_action -type MIRROR -serverName lb-IDS_vserver
```

検査用のコンテンツ検査ポリシーの追加

コンテンツ検査アクションを作成した後、サービスの要求を評価するためにコンテンツ検査ポリシーを追加する必要があります。

コマンドプロンプトで、次のように入力します。

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

例:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```


HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーの追加

Web トラフィックを受け入れるために、コンテンツスイッチングまたは負荷分散仮想サーバーを追加します。また、仮想サーバ上で layer2 接続を有効にする必要があります。

ロードバランシングの詳細については、「ロードバランシングの仕組み」トピックを参照してください。

コマンドプロンプトで入力します。

```
add lb vserver <name> <vserver name>
```

例:

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

コンテンツ検査ポリシーを **HTTP/SSL** タイプの仮想サーバーの負荷分散にバインドする

HTTP/SSL タイプのコンテンツスイッチングまたは負荷分散仮想サーバーを、コンテンツ検査ポリシーにバインドする必要があります。

コマンドプロンプトで、次のように入力します。

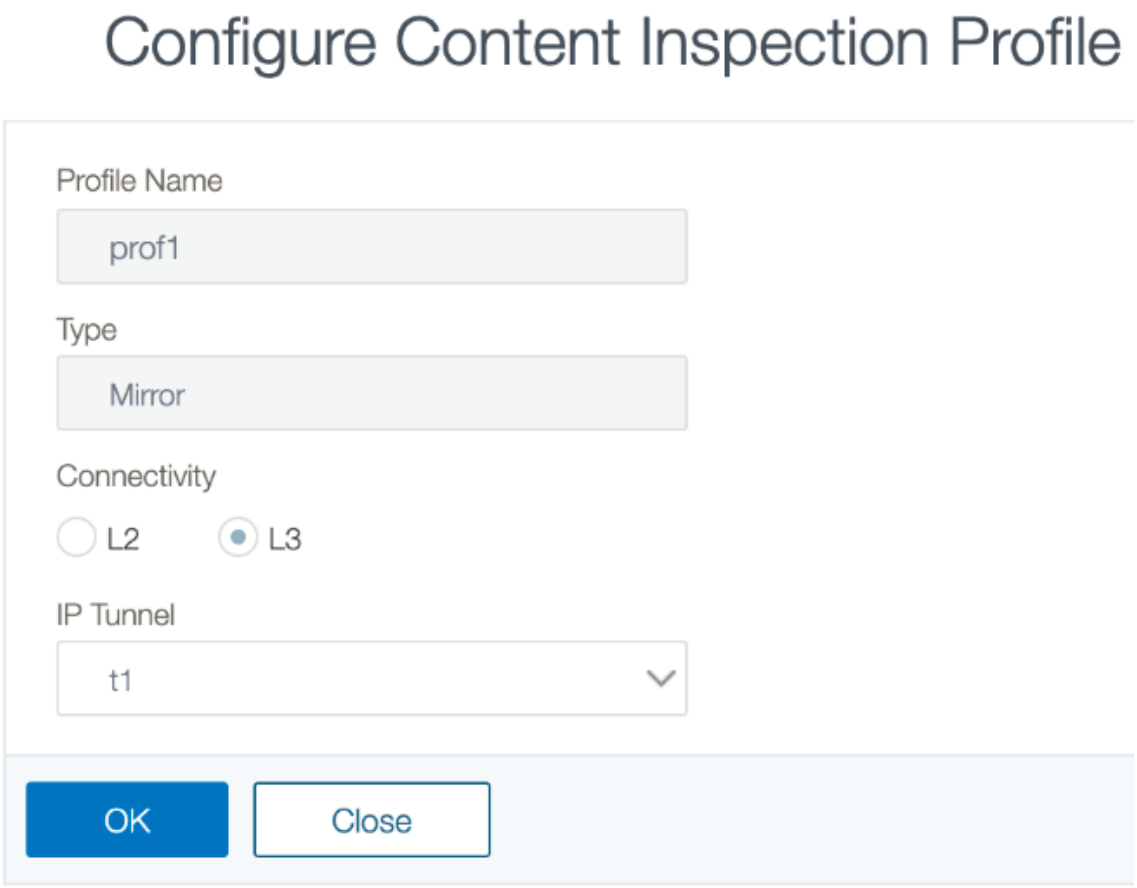
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <REQUEST>
```

例:

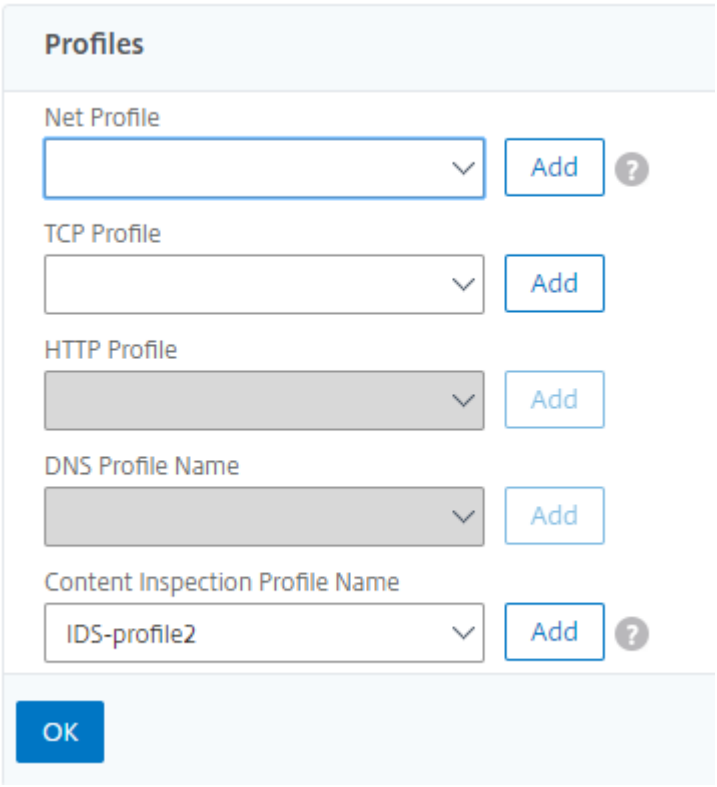
```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

Citrix ADC GUI を使用したインラインサービス統合の構成

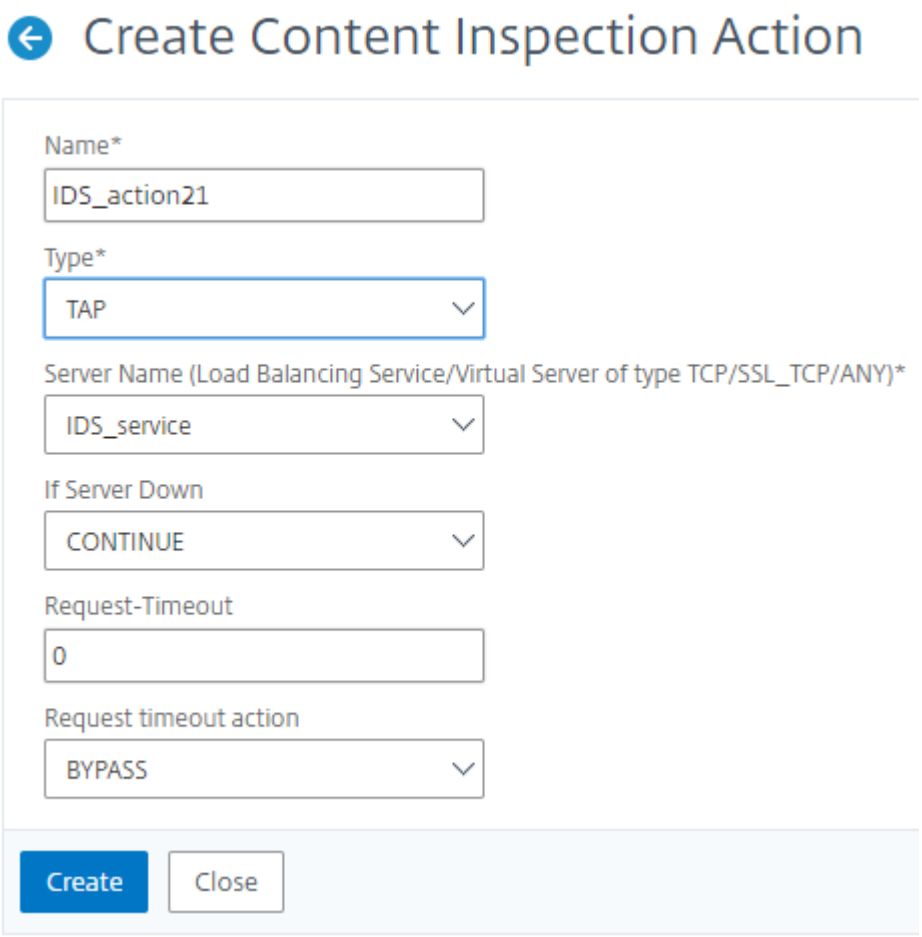
1. [セキュリティ] > [コンテンツ検査] > [コンテンツ検査プロファイル] に移動します。
2. [コンテンツ検査プロファイル] ページで、[追加] をクリックします。
3. 「コンテンツ検査プロファイルの作成」ページで、次のパラメータを設定します。
   a) プロファイル名。IDS のコンテンツ検査プロファイルの名前。
   b) タイプ。プロファイルタイプを MIRROR として選択します。
   c) 接続性。レイヤ 2 またはレイヤ 3 インターフェイス。
   d) IP トンネル。2 つのネットワーク間のネットワーク通信チャネルを選択します。
4. ［作成］をクリックします。



5. [トラフィック管理] > [負荷分散] > [サービス] に移動し、[追加] をクリックします。
6. [負荷分散サービス] ページで、コンテンツ検査サービスの詳細を入力します。
7. [詳細設定] セクションで、[プロファイル] をクリックします。
8. [プロファイル] セクションに移動し、[鉛筆] アイコンをクリックしてコンテンツ検査プロファイルを追加します。
9. [**OK**] をクリックします。

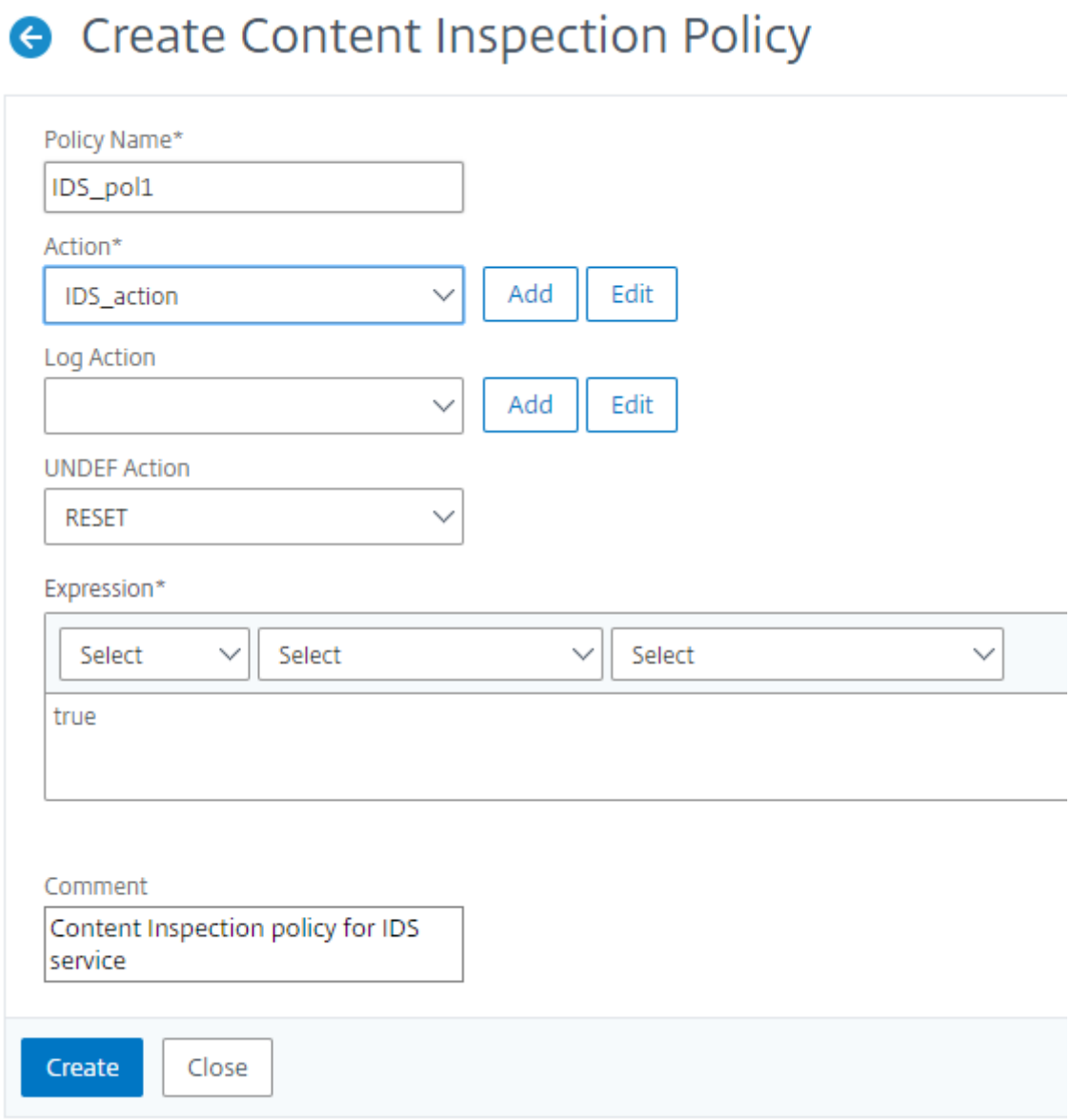


10. [負荷分散] > [サーバー] に移動します。HTTP または SSL タイプの仮想サーバーを追加します。
11. サーバの詳細を入力したら、[**OK**] をクリックし、もう一度 [**OK**] をクリックします。
12. [詳細設定] セクションで、[ポリシー] をクリックします。
13. [ポリシー] セクションに移動し、鉛筆アイコンをクリックして、コンテンツ検査ポリシーを構成します。
14. **[**ポリシーの選択**]** ページで、[コンテンツ検査] を選択します。［続行］ をクリックします。
15. [ポリシーのバインド] セクションで、[+] をクリックしてコンテンツ検査ポリシーを追加します。
16. [**CI** ポリシーの作成**]** ページで、インラインコンテンツ検査ポリシーの名前を入力します。
17. [**Action**] フィールドで、「+」記号をクリックして、タイプ MIRROR の IDS コンテンツインスペクションアクションを作成します。
18. [**CI** アクションの作成**]** ページで、次のパラメータを設定します。
    a) Name：コンテンツ検査インラインポリシーの名前。
    b) タイプ。タイプを MIRROR として選択します。
    c) サーバー名：サーバー/サービス名を [インラインデバイス] として選択します。
    d) サーバがダウンしている場合。サーバーがダウンした場合の操作を選択します。
    e) 要求タイムアウト。タイムアウト値を選択します。デフォルト値を使用できます。
    f) 要求タイムアウトアクション。タイムアウト処理を選択します。デフォルト値を使用できます。
19. ［作成］ をクリックします。



20. [**CI** ポリシーの作成] ページで、その他の詳細を入力します。
21. [**OK**] をクリックして [閉じる] をクリックします。

負荷分散および IDS デバイスへのトラフィックのレプリケーションに関する Citrix ADC GUI 構成の詳細については、「負荷分散」を参照してください。



コンテンツ変換後にトラフィックを負荷分散してバックエンドオリジンサーバーに転送するための Citrix ADC GUI 構成については、負荷分散を参照してください。

ICAP、IPS、および IDS のコンテンツ検査統計

October 7, 2021

ICAP、インラインデバイス統合（IDS）、および侵入防止システム（IPS）デバイスのコンテンツ検査統計は、要求、応答、およびサーバーアクションの詳細の詳細な出力（要約）です。

コンテンツ検査統計は、以下を含む統計データのコレクションです。HTTP/HTTPS コンテンツ検査のために送信さ

れたリクエスト。HTTP/HTTPS IPS、IDS、および ICAP デバイスから受信した応答とバックエンドサーバーのアクション。

CLI を使用してコンテンツ検査統計を表示するには:

コマンドプロンプトで入力します。

```
stat contentInspection
```

```
ContentInspection Stats

Inline Statistics
                                    Total
Requests                             10
Responses                             6
Request Bytes Sent                   3235
Request Bytes Received               2977
Response Bytes Sent                  17302
Response Bytes Received              19681
Serverdown Reset Action taken         1
Serverdown Drop Action taken          0
Serverdown BYPASS Action taken        0
Inline device Generated Response      3

Mirror Statistics
                                    Total
Requests                             4
Responses                            4
Requests Bytes Sent                 2763
Responses Bytes Sent                16732
Serverdown Reset Action taken        0
Serverdown Drop Action taken         0
Serverdown BYPASS Action taken       1

ICAP Statistics
                                    Total
REQMOD requests Sent                   6
RESPMOD requests Sent                  4
Preview requests                       1
204 Responses Received                 6
100 Continue Responses Received        1
204 NO content Received                5
Adaptive Requests                      0
Adaptive Responses                     4
Callout requests Initiated             1
```



```
Callout requests completed          1
ICAP Req/Resp Errors handled        1
Serverdown Reset Action taken       1
Serverdown Drop Action taken        0
Serverdown BYPASS Action taken      1

Done
<!--NeedCopy-->
```

SSL フォワードプロキシ

October 7, 2021

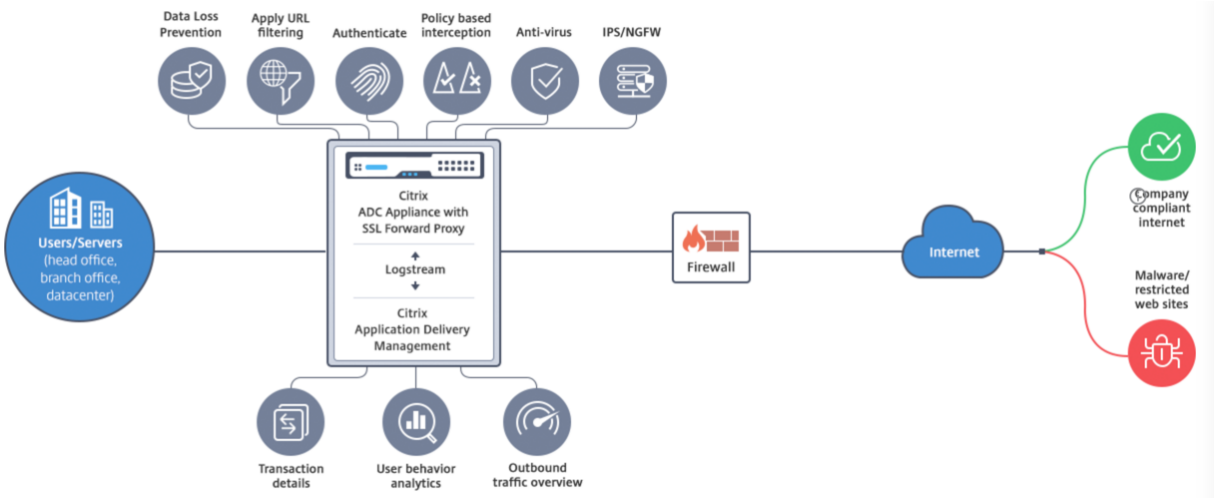
注**:** SSL 転送プロキシ機能は ADC Premium ライセンスで利用できます。

近年、ウェブトラフィックは指数関数的に増加しており、企業は日常業務にインターネットに依存する傾向が高まっています。これは、より多様なエンドポイント、モビリティ、および BYOD の出現に加え、攻撃者基盤の拡大に伴い、ユーザーは最新のマルウェアの標的を容易にするようになっています。ID の盗難やデータの漏洩に対する脆弱性がますます高まっています。従来、企業はマルウェアやウイルスの HTTP トラフィックを検査してきました。彼らはそれほど顕著ではなかったので、HTTPS/TLS トラフィックをバイパスしました。機密性が高く信頼できるコンテンツには控えめに使用されていました。しかし、ほとんどのパブリックインターネット Web サイトがユーザーのプライバシーを保護するために HTTPS を使用することを好むため、これは急速に変化しました。その結果、暗号化されたパケットを検査できないため、マルウェアや企業ネットワークへの侵入が許されます。SSL フォワードプロキシソリューションは、企業がインターネットの脅威から保護するために使用できるツールを提供します。

プロキシは、ユーザーとインターネットまたは SaaS アプリケーション間のすべてのトラフィックを制御するサーバーです。すべてのトラフィックはこのプロキシを通過するため、ユーザー認証や URL 分類などのセキュリティ関連の機能を実行します。

次の図は、SSL フォワードプロキシの実装の概要です。トラフィックは、本社、支社、データセンター、およびリモート従業員から企業ネットワークを経由して流れます。ネットワークのエッジにある Citrix ADC アプライアンスは、プロキシとして機能します。このアプライアンスは透過プロキシモードまたは明示的プロキシモードで動作し、HTTPS を含むインターネットトラフィックの傍受を制御できるようにします。アプライアンスで設定されたポリシーによって、特定の要求を代行受信するか、バイパスするか、ブロックするかが決まります。制限されたサイトへのアクセスは、URL フィルタリングを使用してブロックできます。ユーザーは社内ネットワークにログオンする前に認証されます。すべての要求と応答は、ユーザーを識別するためにタグ付けされ、インターネットサイトへのアクセスは分類されます。ユーザー・アクティビティはログに記録され、レポートの生成に使用されます。侵害が発生した場合、管理者は感染したシステムを隔離し、その Web サイトにアクセスした他のユーザーのデバイスが侵害されていないかどうかを判断し、適切な措置を講じることができます。Citrix Application Delivery Management（ADM）を SSL フォワードプロキシと統合すると、ログに記録されたユーザーアクティビティとアプライアンス内の後続のレコードが `logstream`を使用して Citrix ADM にエクスポートされます。Citrix ADM は、訪問したウェブサイトか

らオンラインで過ごした時間まで、ユーザーのアクティビティに関する情報を照合して表示します。また、帯域幅の使用やマルウェアやフィッシングサイトなどの検出された脅威に関する情報も提供します。これらの主要なメトリックを使用してネットワークを監視し、SSL 転送プロキシ機能を使用して修正措置を講じることができます。



SSL フォワード・プロキシにより、IT ダイレクタは次のことを実行できます。

- バイパスされたセキュアトラフィックを可視化します。
- 悪意のあるサイトや未知のサイトへのアクセスをブロックし、企業内のユーザーの感染を防ぎます。
- 個人用メール、ソーシャルネットワーキング、求人検索 Web サイトなど、一部の Web サイトへのアクセスを企業ネットワークから制御します。
- インテリジェントなコンテンツ制御ポリシーを適用して、ユーザーの生産性を最大限に高めます。

SSL フォワードプロキシ機能の使用を開始する

October 7, 2021

重要:

- OCSP チェックでは、証明書の有効性をチェックするためにインターネット接続が必要です。NSIP アドレスを使用してインターネットからアプライアンスにアクセスできない場合は、アクセス制御リスト（ACL）を追加して、NSIP アドレスからサブネット IP（SNIP）アドレスに NAT を実行します。SNIP はインターネットにアクセスできる必要があります。例:

```
1  add ns acl a1 ALLOW -srcIP = <NSIP> -destIP  "!="
       10.0.0.0-10.255.255.255
2
3  add rnat RNAT-1 a1
4
```



```
bind rnat RNAT-1 -<SNIP>

apply acls
<!--NeedCopy-->
```

- ドメイン名を解決する DNS ネームサーバーを指定します。
- アプライアンスの日付が NTP サーバと同期していることを確認します。日付が同期されていない場合、アプライアンスはオリジンサーバー証明書の有効期限が切れているかどうかを効果的に検証できません。

SSL 転送プロキシ機能を使用するには、次のタスクを実行する必要があります。

- エクスプリシットモードまたはトランスペアレントモードでプロキシサーバーを追加します。
- SSL インターセプションを有効にします。
  - SSL プロファイルを設定します。
  - SSL ポリシーをプロキシサーバーに追加してバインドします。
  - SSL インターセプション用の CA 証明書とキーのペアを追加およびバインドします。

注:

トランスペアレントプロキシモードに設定された ADC アプライアンスは、HTTP および HTTPS プロトコルのみを代行受信できます。telnet などの他のプロトコルをバイパスするには、プロキシ仮想サーバーに次のリッスンポリシーを追加する必要があります。

仮想サーバは、HTTP および HTTPS の着信トラフィックのみを受け入れるようになりました。

```
set cs vserver transparent-pxy1 PROXY * * -cltTimeout 180 -Listenpolicy
    ”CLIENT.TCP.DSTPORT.EQ(80) || CLIENT.TCP.DSTPORT.EQ(443)”`
<!--NeedCopy-->
```

配置によっては、次の機能を設定する必要がある場合があります。

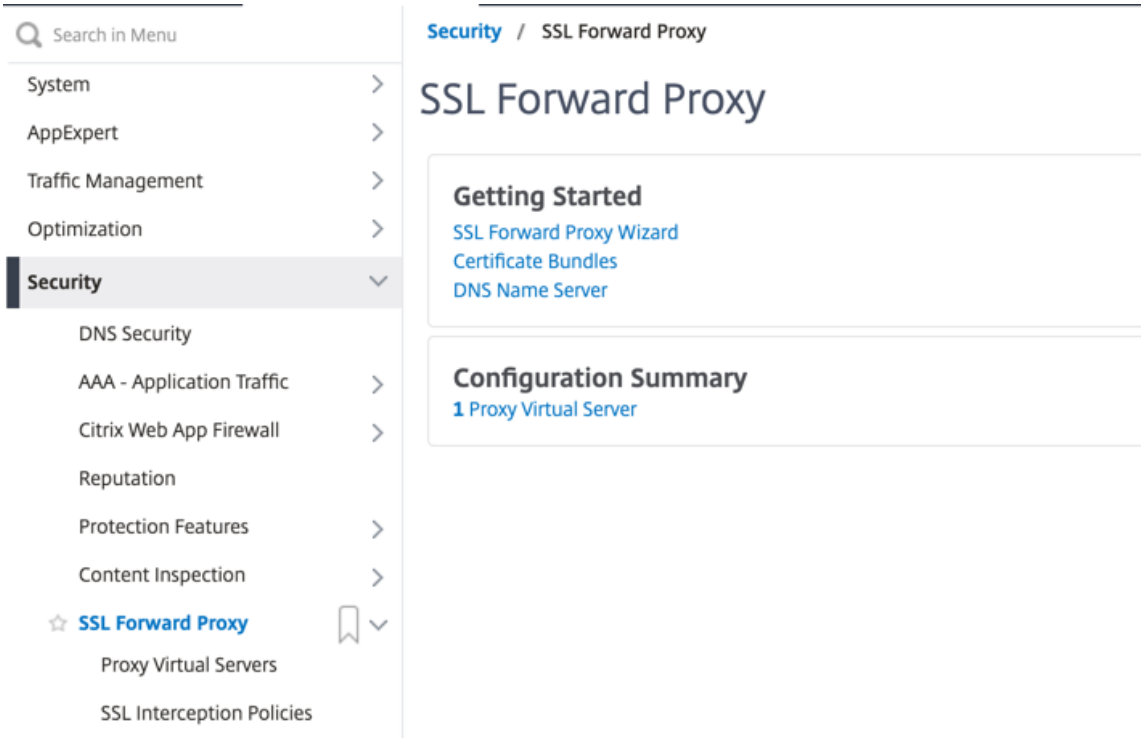
- 認証サービス（推奨）— ユーザーを認証します。認証サービスを使用しない場合、ユーザーアクティビティはクライアント IP アドレスに基づきます。
- URL フィルタリング — カテゴリ、レピュテーションスコア、URL リストによって URL をフィルタリングします。
- 分析-Citrix Application Delivery Management（ADM）でのユーザーアクティビティ、ユーザーリスクインジケータ、帯域幅消費、トランザクションの分類を表示します。

注： SSL フォワードプロキシは、一般的な HTTP および HTTPS 標準に続いて同様の製品を実装しています。この実装は、特定のブラウザを念頭に置いて行われ、最も一般的なブラウザと互換性があります。SSL フォワードプロキシは、一般的なブラウザと Google Chrome、Internet Explorer、および Mozilla の Firefox の最近のバージョンでテストされています。

SSL 転送プロキシウィザード

SSL 転送プロキシウィザードは、Web ブラウザーを使用して SSL 転送プロキシの展開全体を管理するためのツールを提供します。これは、迅速に SSL 転送プロキシサービスを立ち上げるために顧客をガイドし、明確に定義された一連の手順に従うことによって構成を簡素化するのに役立ちます。

1. **Security > SSL Forward Proxy** に移動します。[はじめに] で、[**SSL** 転送プロキシウィザード] をクリックします。



2. ウィザードの手順に従って、配置を設定します。

透過プロキシサーバーへのリッスンポリシーの追加

1. [セキュリティ] **> [SSL** フォワードプロキシ] > [プロキシ仮想サーバー] に移動します。透過プロキシサーバを選択し、[**Edit**] をクリックします。
2. [基本設定] を編集し、[その他] をクリックします。
3. [リッスンプライオリティ] に 1 を入力します。
4. [リッスンポリシー式] に、次の式を入力します。

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```


この式は、HTTP および HTTPS トラフィックの標準ポートを想定しています。HTTP の場合は 8080、HTTPS の場合は 8443 など、異なるポートを設定した場合は、これらのポートを反映するように式を変更します。

制限事項

SSL 転送プロキシは、クラスタ設定、管理パーティション、および Citrix ADC FIPS アプライアンスではサポートされません。

プロキシモード

October 7, 2021

Citrix ADC アプライアンスは、インターネットおよび SaaS アプリケーションに接続するためのクライアントのプロキシとして機能します。プロキシとして、すべてのトラフィックを受け入れ、トラフィックのプロトコルを決定します。トラフィックが HTTP または SSL でない限り、そのまま宛先に転送されます。アプライアンスはクライアントから要求を受信すると、要求を傍受し、ユーザー認証、サイトの分類、リダイレクトなどのアクションを実行します。ポリシーを使用して、許可するトラフィックとブロックするトラフィックを決定します。

アプライアンスは、クライアントとプロキシの間で、もう一方はプロキシとオリジンサーバー間の 2 つの異なるセッションを維持します。プロキシは、ユーザーが定義したポリシーを使用して、HTTP および HTTPS トラフィックを許可またはブロックします。したがって、財務情報などの機密データをバイパスするポリシーを定義することが重要です。アプライアンスは、トラフィック管理ポリシーを作成するために、レイヤ 4 からレイヤ 7 へのトラフィック属性とユーザ ID 属性の豊富なセットを提供します。

SSL トラフィックの場合、プロキシはオリジンサーバーの証明書を確認し、サーバーとの正当な接続を確立します。次に、サーバー証明書をエミュレートし、Citrix ADC にインストールされた CA 証明書を使用して署名し、作成したサーバー証明書をクライアントに提示します。SSL セッションを正常に確立するには、CA 証明書を信頼できる証明書としてクライアントのブラウザに追加する必要があります。

アプライアンスは、透過および明示的なプロキシモードをサポートします。明示的なプロキシモードでは、組織が設定をクライアントのデバイスにプッシュしない限り、クライアントはブラウザで IP アドレスを指定する必要があります。このアドレスは、ADC アプライアンス上で構成されたプロキシ・サーバの IP アドレスです。すべてのクライアント要求は、この IP アドレスに送信されます。明示的なプロキシの場合は、タイプ PROXY のコンテンツスイッチング仮想サーバを設定し、IP アドレスと有効なポート番号を指定する必要があります。

透過プロキシは、名前が示すように、クライアントに対して透過的です。つまり、クライアントは、プロキシサーバーが要求を仲介していることを認識していない可能性があります。ADC アプライアンスはインライン展開で構成され、すべての HTTP および HTTPS トラフィックを透過的に受け入れます。トランスペアレントプロキシの場合、IP アドレスおよびポートとしてアスタリスク（*）を使用して、タイプ PROXY のコンテンツスイッチング仮想サーバを設定する必要があります。GUI で **SSL** 転送プロキシウィザードを使用する場合、IP アドレスとポートを指定する必要はありません。

> 注
>
> 透過プロキシモードで HTTP および HTTPS 以外のプロトコルを代行受信するには、リッスンポリシーを追加し、プロキシサーバにバインドする必要があります。

CLI を使用して SSL 転送プロキシを構成する

コマンドプロンプトで入力します。

```
add cs vserver <name> PROXY <ipaddress> <port>
<!--NeedCopy-->
```

引数:

名前:

プロキシサーバーの名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。CS 仮想サーバの作成後は変更できません。

次の要件は、CLI だけに適用されます。

名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「my server」や「my server」など)。

この引数は必須です。最大長: 127

IPAddress:

プロキシサーバの IP アドレス。

ポート:

プロキシサーバーのポート番号。最小値:1

明示的なプロキシの例:

```
add cs vserver swgVS PROXY 192.0.2.100 80
<!--NeedCopy-->
```

透過プロキシの例:

```
add cs vserver swgVS PROXY * *
<!--NeedCopy-->
```


GUI を使用してリッスンポリシーをトランスペアレントプロキシサーバーに追加する

1. [セキュリティ] > [**SSL** 転送プロキシ] > [プロキシ仮想サーバー] に移動します。透過プロキシサーバを選択し、[**Edit**] をクリックします。
2. [基本設定] を編集し、[その他] をクリックします。
3. [リッスンプライオリティ] に 1 を入力します。
4. [リッスンポリシー式] に、次の式を入力します。

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```

> 注
>
> この式は、HTTP および HTTPS トラフィックの標準ポートを想定しています。異なるポートを構成した場合（たとえば、HTTP の場合は 8080、HTTPS の場合は 8443）、上記の式を変更してそれらのポートを指定します。

SSL インターセプション

October 7, 2021

SSL インターセプション用に構成された Citrix ADC アプライアンスは、プロキシとして機能します。SSL/TLS トラフィックを傍受および復号化し、暗号化されていない要求を検査し、管理者がコンプライアンスルールとセキュリティチェックを適用できるようにします。SSL インターセプションでは、代行受信、ブロック、または許可するトラフィックを指定するポリシーを使用します。たとえば、銀行などの金融ウェブサイトとの間のトラフィックは傍受されてはいけませんが、他のトラフィックは傍受され、ブラックリストに登録されたサイトは識別され、ブロックされます。トラフィックを傍受する一般的なポリシーを 1 つ構成し、一部のトラフィックをバイパスするより具体的なポリシーを構成することをお勧めします。

クライアントとプロキシは、HTTPS/TLS ハンドシェイクを確立します。プロキシは、サーバーと別の HTTPS/TLS ハンドシェイクを確立し、サーバー証明書を受信します。プロキシは、クライアントに代わってサーバー証明書を検証し、オンライン証明書ステータスプロトコル（OCSP）を使用してサーバー証明書の有効性もチェックします。サーバ証明書を再生成し、アプライアンスにインストールされている CA 証明書のキーを使用して署名し、クライアントに提示します。したがって、クライアントと Citrix ADC アプライアンスの間で 1 つの証明書が使用され、アプライアンスとバックエンドサーバー間で別の証明書が使用されます。

> 重要

> 再生成されたサーバー証明書がクライアントによって信頼されるように、サーバー証明書の署名に使用される CA 証明書は、すべてのクライアントデバイスにプレインストールする必要があります。

代行受信された HTTPS トラフィックの場合、プロキシサーバは発信トラフィックを復号化し、クリアテキストの HTTP 要求にアクセスします。また、プレーンテキストの URL を調べて、企業ポリシーと URL レピュテーションに基づいてアクセスを許可またはブロックするなど、レイヤ 7 アプリケーションを使用してトラフィックを処理できます。ポリシーがオリジナル・サーバーへのアクセスを許可する場合、プロキシ・サーバーは再暗号化された要求を（オリジナル・サーバー上の）宛先サービスに転送します。プロキシは、オリジンサーバーからの応答を復号化し、クリアテキストの HTTP 応答にアクセスし、オプションで任意のポリシーを応答に適用します。プロキシは応答を再暗号化し、クライアントに転送します。ポリシーがオリジンサーバーへの要求をブロックする場合、プロキシは HTTP 403 などのエラー応答をクライアントに送信できます。

SSL インターセプションを実行するには、先ほど設定したプロキシサーバに加えて、ADC アプライアンスで次の設定を行う必要があります。

- SSL プロファイル
- SSL ポリシー
- CA 証明書ストア
- SSL エラーの自動学習とキャッシュ

> 注:
>
> HTTP/2 トラフィックは SSL インターセプト機能によってインターセプトされません。

SSL インターセプション証明書ストア

SSL トランザクションの一部である SSL 証明書は、会社（ドメイン）または個人を識別するデジタルデータフォーム（X509）です。SSL 証明書は、認証局 (CA) によって発行されます。CA は、プライベートまたはパブリックにできます。Verisign などのパブリック CA によって発行された証明書は、SSL トランザクションを実行するアプリケーションによって信頼されます。これらのアプリケーションは、信頼する CA の一覧を維持します。

ADC アプライアンスは、フォワード・プロキシとして、クライアントとサーバー間のトラフィックの暗号化と復号化を実行します。これは、クライアント（ユーザー）のサーバーとして、およびサーバーのクライアントとして機能します。アプライアンスは HTTPS トラフィックを処理する前に、不正なトランザクションを防ぐために、サーバーの ID を検証する必要があります。したがって、オリジンサーバーのクライアントとして、アプライアンスはオリジンサーバー証明書を受け入れる前にオリジンサーバー証明書を確認する必要があります。サーバ証明書を検証するには、サーバ証明書の署名および発行に使用されるすべての証明書（ルート証明書および中間証明書など）がアプライアンス上に存在する必要があります。デフォルトの CA 証明書セットは、アプライアンス上にプレインストールされています。アプライアンスは、これらの証明書を使用して、ほぼすべての一般的なオリジンサーバ証明書を検証できます。この既定のセットは変更できません。ただし、展開でさらに多くの CA 証明書が必要な場合は、そのような証明書のバンドルを作成し、そのバンドルをアプライアンスにインポートできます。バンドルには、単一の証明書を含めることもできます。

証明書バンドルをアプライアンスにインポートすると、アプライアンスはリモートの場所からバンドルをダウンロードし、バンドルに証明書のみが含まれていることを確認した後、アプライアンスにインストールします。証明書バン

ドルを使用してサーバ証明書を検証するには、事前に証明書バンドルを適用する必要があります。また、証明書バンドルをエクスポートして編集したり、オフラインの場所にバックアップとして保存したりすることもできます。

CLI を使用して **CA** 証明書バンドルをアプライアンスにインポートして適用する

コマンドプロンプトで入力します。

```
import ssl certBundle <name> <src>
apply ssl certBundle <name>
<!--NeedCopy-->
```

```
show ssl certBundle
<!--NeedCopy-->
```

要点:

名前:

インポートした証明書バンドルに割り当てる名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。次の要件は、CLI だけに適用されます。

名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「my file」や「my file」など)。

最大長：31

src:

インポートまたはエクスポートする証明書バンドルへのプロトコル、ホスト、およびパス（ファイル名を含む）を指定する URL。たとえば、`http://www.example.com/cert_bundle_file`などです。

注: インポートするオブジェクトがアクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。

最大長：2047

例:

```
import ssl certbundle swg-certbundle http://www.example.com/cert_bundle
apply ssl certBundle swg-certbundle
<!--NeedCopy-->
```



```
show ssl certbundle

            Name : swg-certbundle(Inuse)

            URL : http://www.example.com/cert_bundle

    Done
<!--NeedCopy-->
```

GUI を使用して **CA** 証明書バンドルをアプライアンスにインポートして適用する

1. セキュリティ **>** **SSL** 転送プロキシ **>** はじめに **>** 証明書バンドルに移動します。
2. 次のいずれかを行います:
   - リストから証明書バンドルを選択します。
   - 証明書バンドルを追加するには、「+」をクリックし、名前とソース URL を指定します。[**OK**] をクリックします。
3. [**OK**] をクリックします。

CLI を使用して **CA** 証明書バンドルをアプライアンスから削除する

コマンドプロンプトで入力します。

```
remove certBundle <cert bundle name>
<!--NeedCopy-->
```

例:

```
remove certBundle mytest-cacert
<!--NeedCopy-->
```

CLI を使用してアプライアンスから **CA** 証明書バンドルをエクスポートする

コマンドプロンプトで入力します。

```
export certBundle <cert bundle name> <Path to export>
<!--NeedCopy-->
```


要点:

名前:

インポートした証明書バンドルに割り当てる名前。ASCII 英数字またはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。次の要件は、CLI だけに適用されます。

名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「my file」や「my file」など)。

最大長: 31

src:

インポートまたはエクスポートする証明書バンドルへのプロトコル、ホスト、およびパス（ファイル名を含む）を指定する URL。たとえば、`http://www.example.com/cert_bundle_file`などです。

注: インポートするオブジェクトがアクセスにクライアント証明書認証を必要とする HTTPS サーバー上にある場合、インポートは失敗します。

最大長: 2047

例:

```
export certBundle mytest-cacert http://192.0.2.20/
<!--NeedCopy-->
```

Mozilla CA 証明書ストアから **CA** 証明書バンドルをインポート、適用、検証する

コマンドプロンプトで入力します。

```
> import certbundle mozilla_public_ca https://curl.haxx.se/ca/cacert.
    pem
Done
<!--NeedCopy-->
```

バンドルを適用するには、次のように入力します。

```
> apply certbundle mozilla_public_ca
Done
<!--NeedCopy-->
```


使用中の証明書バンドルを確認するには、次のように入力します。

```
> sh certbundle | grep mozilla
                Name : mozilla_public_ca (Inuse)
<!--NeedCopy-->
```

制限事項

- 証明書バンドルは、クラスタ設定またはパーティション化されたアプライアンスではサポートされません。
- TLSv1.3 プロトコルは、SSL フォワードプロキシではサポートされていません。

SSL 代行受信のための SSL ポリシーインフラストラクチャ

ポリシーは、着信トラフィックに対するフィルタのように動作します。ADC アプライアンスのポリシーは、プロキシ接続と要求の管理方法を定義するのに役立ちます。処理は、そのポリシーに対して設定されているアクションに基づきます。つまり、接続要求のデータはポリシーで指定された規則と比較され、規則に一致する接続にアクションが適用されます (式)。ポリシーに割り当ててポリシーを作成するアクションを定義したら、そのアクションをプロキシサーバにバインドして、そのプロキシサーバを通過するトラフィックに適用する必要があります。

SSL インターセプションの SSL ポリシーは、着信トラフィックを評価し、ルール（式）に一致する要求に事前定義されたアクションを適用します。接続の代行受信、バイパス、またはリセットは、定義された SSL ポリシーに基づいて決定されます。ポリシーに対して、INTERCEPT、BYPASS、または RESET の 3 つのアクションのいずれかを設定できます。ポリシーを作成するときは、アクションを指定する必要があります。ポリシーを有効にするには、アプライアンスのプロキシサーバーにポリシーをバインドする必要があります。ポリシーが SSL インターセプションを対象とするように指定するには、プロキシサーバーにポリシーをバインドするときに、タイプ（バインドポイント）を INTERCEPT_REQ として指定する必要があります。ポリシーのバインドを解除するときは、タイプを INTERCEPT_REQ として指定する必要があります。

> 注:
>
> プロキシサーバは、ポリシーを指定しない限り、傍受を決定できません。

トラフィックインターセプションは、任意の SSL ハンドシェイク属性に基づいて行うことができます。最も一般的に使用されるのは SSL ドメインです。SSL ドメインは通常、SSL ハンドシェイクの属性によって示されます。これは、SSL Client Hello メッセージから抽出されたサーバー名インジケータ値（存在する場合）、または元のサーバー証明書から抽出されたサーバー別名（SAN）値になります。SSL 代行受信ポリシーは、特別な属性 DETECTED_DOMAIN を提示します。この属性により、顧客はオリジンサーバー証明書からの SSL ドメインに基づいて代行受信ポリシーを簡単に作成できるようになります。顧客は、ドメイン名を文字列、URL リスト（URL セットまたは`patset`）、またはドメインから派生した URL カテゴリと照合できます。

CLI を使用して **SSL** ポリシーを作成します

コマンドプロンプトで入力します。

```
add ssl policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

例:

次の例は、detected_domain 属性を使用してドメイン名をチェックする式を持つポリシーの例です。

XYZBANK などの金融機関へのトラフィックを傍受しない

```
add ssl policy pol1 -rule client.ssl.detected_domain.contains("XYZBANK")
    ) -action BYPASS
<!--NeedCopy-->
```

ユーザーが企業ネットワークから YouTube に接続することを許可しない

```
add ssl policy pol2 -rule client.ssl.client.ssl.detected_domain.
    url_categorize(0,0).category.eq ("YouTube") -action RESET
<!--NeedCopy-->
```

すべてのユーザトラフィックを代行受信する

```
add ssl policy pol3  - rule true  - action INTERCEPT
<!--NeedCopy-->
```

お客様が detected_domain を使用したくない場合は、任意の SSL ハンドシェイク属性を使用してドメインを抽出および推測できます。

たとえば、ドメイン名が、クライアントの hello メッセージの SNI 拡張に見つかりません。ドメイン名は、オリジンサーバー証明書から取得する必要があります。次の例は、オリジンサーバー証明書のサブジェクト名でドメイン名をチェックする式を持つポリシーの例です。

任意の Yahoo ドメインへのすべてのユーザートラフィックを傍受する

```
add ssl policy pol4 -rule client.ssl.origin_server_cert.subject.
    contains("yahoo")  - action INTERCEPT
<!--NeedCopy-->
```


「ショッピング/小売」カテゴリのすべてのユーザートラフィックを傍受する

```
1 add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
    subject.URL_CATEGORIZE(0,0).CATEGORY.eq(”Shopping/Retail”) -action
    INTERCEPT
2 <!--NeedCopy-->
```

未分類の URL へのすべてのユーザトラフィックを代行受信する

```
1 add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
    subject.url_categorize(0,0).category.eq(”Uncategorized”) -action
    INTERCEPT
2 <!--NeedCopy-->
```

次の例は、URL セットのエントリに対してドメインを照合するポリシーの例です。

SNI のドメイン名が URL セット「top100」のエントリと一致する場合、すべてのユーザートラフィックを代行受信します。

```
1 add ssl policy pol_url_set  -rule client.ssl.client_hello.SNI.
    URLSET_MATCHES_ANY(”top100”) -action INTERCEPT
2 <!--NeedCopy-->
```

オリジンサーバー証明書が URL セット「top100」のエントリと一致する場合、ドメイン名のすべてのユーザートラフィックを代行受信します。

```
1 add ssl policy pol_url_set  -rule client.ssl.origin_server_cert.subject
    .URLSET_MATCHES_ANY(”top100”) -action INTERCEPT
2 <!--NeedCopy-->
```

GUI を使用したプロキシサーバーへの **SSL** ポリシーの作成

1. [**Traffic Management**] > [**SSL**] > [**Policies**] に移動します。
2. [**SSL** ポリシー] タブで、[追加] をクリックし、次のパラメータを指定します。
   - ポリシー名
   - ポリシーアクション：代行受信、バイパス、またはリセットから選択します。
   - 式
3. [作成] をクリックします。

CLI を使用して **SSL** ポリシーをプロキシサーバーにバインドする

コマンドプロンプトで入力します。

```
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type  INTERCEPT_REQ
<!--NeedCopy-->
```

例:

```
bind ssl vserver <name> -policyName pol1 -priority 10 -type
   INTERCEPT_REQ
<!--NeedCopy-->
```

GUI を使用して **SSL** ポリシーをプロキシサーバーにバインドする

1. **[**セキュリティ**] > [SSL** フォワードプロキシ**] > [**プロキシ仮想サーバー**]** に移動します。
2. 仮想サーバを選択し、[**Edit**] をクリックします。
3. [詳細設定**]** で、[**SSL** ポリシー] をクリックします。
4. [**SSL** ポリシー] ボックスの内側をクリックします。
5. 「ポリシーの選択」で、バインドするポリシーを選択します。
6. 「タイプ」で「**INTERCEPT_REQ**」を選択します。
7. [バインド] をクリックし、[**OK**] をクリックします。

CLI を使用して **SSL** ポリシーをプロキシサーバーにバインド解除します

コマンドプロンプトで入力します。

```
unbind ssl vserver <vServerName> -policyName <string> -type
   INTERCEPT_REQ
<!--NeedCopy-->
```

SSL ポリシーで使用される **SSL** 式

式	説明
CLIENT.SSL.CLIENT_HELLO.SNI.*	SNI 拡張を文字列形式で返します。文字列を評価して、指定したテキストが含まれているかどうかを確認します。例：client.ssl.client_hello.sni.contains(“xyz.com”)
CLIENT.SSL.ORIGIN_SERVER_CERT.*	バックエンドサーバーから受け取った証明書を、文字列形式で返します。文字列を評価して、指定したテキストが含まれているかどうかを確認します。例：client.ssl.origin_server_cert.subject.contains(“xyz.com”)
CLIENT.SSL.DETECTED_DOMAIN.*	SNI 拡張またはオリジンサーバー証明書からドメインを文字列形式で返します。文字列を評価して、指定したテキストが含まれているかどうかを確認します。例：client.ssl.detected_domain.contains(“xyz.com”)

SSL エラーの自動学習中

学習モードがオンの場合、アプライアンスは SSL バイパスリストにドメインを追加します。ラーニングモードは、クライアントまたはオリジナルサーバから受信した SSL アラートメッセージに基づいています。つまり、学習は、アラートメッセージを送信するクライアントまたはサーバーによって異なります。アラートメッセージが送信されない場合、ラーニングは行われません。アプライアンスは、次のいずれかの条件が満たされているかどうかを学習します。

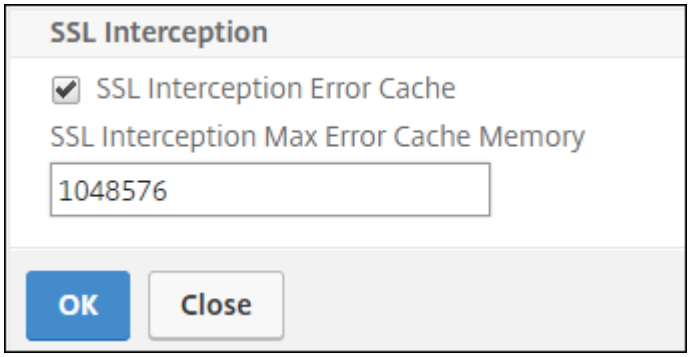
1. クライアント証明書の要求がサーバーから受信されます。
2. ハンドシェイクの一部として、次のアラートのいずれかが受信されます。
    - BAD_CERTIFICATE
    - UNSUPPORTED_CERTIFICATE
    - CERTIFICATE_REVOKED
    - CERTIFICATE_EXPIRED
    - CERTIFICATE_UNKNOWN
    - UNKNOWN_CA（クライアントがピン接続を使用している場合、サーバ証明書を受信すると、この警告メッセージを送信します）。
    - HANDSHAKE_FAILURE

学習を有効にするには、エラーキャッシュを有効にし、学習用に予約されているメモリを指定する必要があります。

GUI を使用した学習の有効化

1. **Traffic Management > SSL** に移動します。

2. **[設定]** で、[**SSL** の詳細設定の変更] をクリックします。
3. 「**SSL** インターセプション」で、「**SSL** インターセプションエラー・キャッシュ」を選択します。
4. 「**SSL** インターセプション最大エラーキャッシュメモリ」で、予約するメモリ（バイト単位）を指定します。



5. [**OK**] をクリックします。

CLI を使用した学習の有効化

コマンドプロンプトで次のように入力します。

```
set ssl parameter -ssliErrorCache ( ENABLED | DISABLED ) -
    ssliMaxErrorCacheMem <positive_integer>
<!--NeedCopy-->
```

引数:

ssliErrorCache:

ダイナミックラーニングを有効または無効にし、学習した情報をキャッシュして、要求の代行受信またはバイパスに関するその後の決定を行います。有効にすると、アプライアンスはキャッシュ検索を実行して、要求をバイパスするかどうかを決定します。

設定可能な値：ENABLED, DISABLED

デフォルト値：無効

ssliMaxErrorCacheMem:

学習したデータのキャッシュに使用できる最大メモリをバイト単位で指定します。このメモリは LRU キャッシュとして使用されるため、設定されたメモリ制限が枯渇した後、古いエントリが新しいエントリに置き換えられます。値 0 を指定すると、自動的に制限が決定されます。

デフォルト値:0

最小値:0

最大値:4294967294

SSL プロファイル

SSL プロファイルは、暗号やプロトコルなどの SSL 設定の集まりです。プロファイルは、異なるサーバーに共通の設定がある場合に役立ちます。各サーバーに同じ設定を指定する代わりに、プロファイルを作成し、プロファイルに設定を指定し、プロファイルを別のサーバーにバインドできます。カスタムフロントエンド SSL プロファイルが作成されない場合、既定のフロントエンドプロファイルは、クライアント側のエンティティにバインドされます。このプロファイルを使用すると、クライアント側の接続を管理するための設定を構成できます。

SSL インターセプトの場合、SSL プロファイルを作成し、プロファイルで SSL インターセプトを有効にする必要があります。デフォルトの暗号グループはこのプロファイルにバインドされますが、展開に合わせてさらに多くの暗号を設定できます。SSL インターセプト CA 証明書をこのプロファイルにバインドしてから、プロファイルをプロキシサーバーにバインドします。SSL インターセプトの場合、プロファイルの重要なパラメーターは、次のアクションに使用されるパラメーターです。

- オリジンサーバー証明書の OCSP ステータスを確認します。
- オリジンサーバーが再ネゴシエーションを要求した場合、クライアントの再ネゴシエーションをトリガーします。
- フロントエンド SSL セッションを再利用する前に、オリジンサーバー証明書を確認してください。

オリジンサーバーと通信するときは、デフォルトのバックエンドプロファイルを使用します。デフォルトのバックエンドプロファイルで、暗号スイートなどのサーバー側のパラメータを設定します。カスタムバックエンドプロファイルはサポートされません。

最も一般的に使用される SSL 設定の例については、このセクションの最後の「サンプルプロファイル」を参照してください。

暗号/プロトコルのサポートは、内部ネットワークと外部ネットワークによって異なります。次の表では、ユーザーと ADC アプライアンスの間の接続は内部ネットワークです。外部ネットワークは、アプライアンスとインターネットの間にあります。

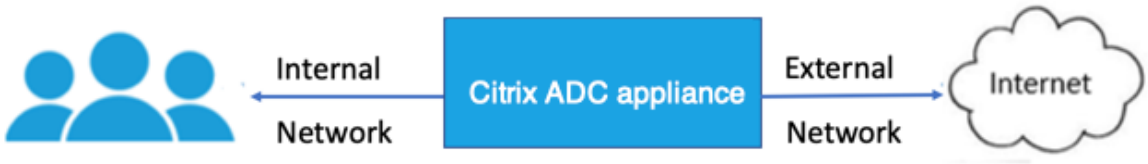


表 1: 内部ネットワークの暗号/プロトコルサポートマトリックス

表 1-Citrix ADC アプライアンスで使用可能な Ciphers での仮想サーバー/フロントエンドサービス/内部サービスのサポートを参照してください。

表 2: 外部ネットワークの暗号/プロトコルサポートマトリックス

表 2-Citrix ADC アプライアンスで使用可能な Ciphers でのバックエンドサービスのサポートを参照してください。

CLI を使用して **SSL** プロファイルを追加し、**SSL** インターセプションを有効にする

コマンドプロンプトで入力します。

```
add ssl profile <name> -sslinterception ENABLED -ssliReneg ( ENABLED |
 DISABLED )-ssliOCSPCheck ( ENABLED | DISABLED )-ssliMaxSessPerServer <
positive_integer>
```

引数:

sslInterception:

SSL セッションのインターセプションを有効または無効にします。

設定可能な値: ENABLED, DISABLED

デフォルト値: 無効

ssliReneg:

オリジンサーバーから再ネゴシエーション要求を受信したときのクライアント再ネゴシエーションのトリガーを有効または無効にします。

設定可能な値: ENABLED, DISABLED

デフォルト値:ENABLED

ssliOCSPCheck:

オリジンサーバー証明書の OCSP チェックを有効または無効にします。

設定可能な値: ENABLED, DISABLED

デフォルト値:ENABLED

サーバごとの最大サイズ:

動的オリジンサーバーごとにキャッシュされる SSL セッションの最大数。クライアント hello メッセージでクライアントから受信した SNI 拡張ごとに、一意の SSL セッションが作成されます。一致するセッションは、サーバーセッションの再利用に使用されます。

デフォルト値:10

最小値:1

最大値:1000

例:

```
1 add ssl profile swg_ssl_profile  -sslinterception ENABLED
2
3 Done
```



```
4
5  sh ssl profile swg_ssl_profile
6
7  1)    Name: swg_ssl_profile (Front-End)
8
9                 SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                      .1: ENABLED  TLSv1.2: ENABLED
10
11                Client Auth: DISABLED
12
13                Use only bound CA certificates: DISABLED
14
15                Strict CA checks:                                 NO
16
17                Session Reuse: ENABLED
                      Timeout: 120 seconds
18
19                DH: DISABLED
20
21                DH Private-Key Exponent Size Limit: DISABLED
                      Ephemeral RSA: ENABLED
                      Refresh Count: 0
22
23                Deny SSL Renegotiation
                      ALL
24
25                Non FIPS Ciphers: DISABLED
26
27                Cipher Redirect: DISABLED
28
29                SSL Redirect: DISABLED
30
31                Send Close-Notify: YES
32
33                Strict Sig-Digest Check: DISABLED
34
35                Push Encryption Trigger: Always
36
37                PUSH encryption trigger timeout:              1 ms
38
39                SNI: DISABLED
40
41                OCSP Stapling: DISABLED
42
43                Strict Host Header check for SNI enabled SSL sessions:
```



```
                                        NO
44
45                  Push flag:            0x0 (Auto)
46
47                  SSL quantum size:                              8 kB
48
49                  Encryption trigger timeout           100 mS
50
51                  Encryption trigger packet count:                45
52
53                  Subject/Issuer Name Insertion Format: Unicode
54
55                  SSL Interception: ENABLED
56
57                  SSL Interception OCSP Check: ENABLED
58
59                  SSL Interception End to End Renegotiation: ENABLED
60
61                  SSL Interception Server Cert Verification for Client
                        Reuse: ENABLED
62
63                  SSL Interception Maximum Reuse Sessions per Server:  10
64
65                  Session Ticket: DISABLED              Session Ticket
                        Lifetime: 300 (secs)
66
67                  HSTS: DISABLED
68
69                  HSTS IncludeSubDomains: NO
70
71                  HSTS Max-Age: 0
72
73                  ECC Curve: P_256, P_384, P_224, P_521
74
75  1)            Cipher Name: DEFAULT Priority :1
76
77                  Description: Predefined Cipher Alias
78
79  Done
80  <!--NeedCopy-->
```

CLI を使用して **SSL** インターセプション **CA** 証明書を **SSL** プロファイルにバインドする

コマンドプロンプトで入力します。


```
bind ssl profile <name> -ssliCACertkey <ssli-ca-cert>
```

例:

```
bind ssl profile swg_ssl_profile -ssliCACertkey swg_ca_cert

Done

sh ssl profile swg_ssl_profile

1)            Name: swg_ssl_profile (Front-End)

                SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                   .1: ENABLED  TLSv1.2: ENABLED

                Client Auth: DISABLED

                Use only bound CA certificates: DISABLED

                Strict CA checks:                                NO

                Session Reuse: ENABLED
                   Timeout: 120 seconds

                DH: DISABLED

                DH Private-Key Exponent Size Limit: DISABLED
                   Ephemeral RSA: ENABLED
                   Refresh Count: 0

                Deny SSL Renegotiation
                   ALL

                Non FIPS Ciphers: DISABLED

                Cipher Redirect: DISABLED

                SSL Redirect: DISABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED

                Push Encryption Trigger: Always
```



```
36
37                  PUSH encryption trigger timeout:              1 ms
38
39                  SNI: DISABLED
40
41                  OCSP Stapling: DISABLED
42
43                  Strict Host Header check for SNI enabled SSL sessions:
                                NO
44
45                  Push flag:             0x0 (Auto)
46
47                  SSL quantum size:                             8 kB
48
49                  Encryption trigger timeout           100 mS
50
51                  Encryption trigger packet count:               45
52
53                  Subject/Issuer Name Insertion Format: Unicode
54
55                  SSL Interception: ENABLED
56
57                  SSL Interception OCSP Check: ENABLED
58
59                  SSL Interception End to End Renegotiation: ENABLED
60
61                  SSL Interception Server Cert Verification for Client
                       Reuse: ENABLED
62
63                  SSL Interception Maximum Reuse Sessions per Server:  10
64
65                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
66
67                  HSTS: DISABLED
68
69                  HSTS IncludeSubDomains: NO
70
71                  HSTS Max-Age: 0
72
73                  ECC Curve: P_256, P_384, P_224, P_521
74
75  1)            Cipher Name: DEFAULT Priority :1
76
77                  Description: Predefined Cipher Alias
```



```
78
79 1)             SSL Interception CA CertKey Name: swg_ca_cert
80
81 Done
82 <!--NeedCopy-->
```

GUI を使用して **SSL** インターセプション **CA** 証明書を **SSL** プロファイルにバインドする

1. [システム] > [プロファイル] > [**SSL** プロファイル] に移動します。
2. [追加] をクリックします。
3. プロファイルの名前を指定します。
4. **SSL** セッションインターセプションを有効にします。
5. [**OK**] をクリックします。
6. [詳細設定**]** で、[証明書キー] をクリックします。
7. プロファイルにバインドする SSL インターセプト CA 証明書キーを指定します。
8. [選択] をクリックし、[バインド] をクリックします。
9. オプションで、展開に合わせて暗号を設定します。
   - 編集アイコンをクリックし、[追加] をクリックします。
   - 1 つ以上の暗号グループを選択し、右矢印をクリックします。
   - [**OK**] をクリックします。
10. [完了] をクリックします。

GUI を使用して **SSL** プロファイルをプロキシサーバーにバインドする

1. [セキュリティ] > [**SSL** フォワードプロキシ**] > [**プロキシ仮想サーバー] に移動し、サーバーを追加するか、変更するサーバーを選択します。
2. [**SSL** プロファイル] で、[編集] アイコンをクリックします。
3. [**SSL** プロファイル] リストで、前に作成した SSL プロファイルを選択します。
4. [**OK**] をクリックします。
5. [完了] をクリックします。

サンプルプロファイル:

```
1 Name: swg_ssl_profile (Front-End)
2
```



```
3                   SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                        .1: ENABLED  TLSv1.2: ENABLED
4
5                   Client Auth: DISABLED
6
7                   Use only bound CA certificates: DISABLED
8
9                   Strict CA checks:                                 NO
10
11                  Session Reuse: ENABLED
                        Timeout: 120 seconds
12
13                  DH: DISABLED
14
15                  DH Private-Key Exponent Size Limit: DISABLED
                        Ephemeral RSA: ENABLED
                        Refresh Count: 0
16
17                  Deny SSL Renegotiation
                        ALL
18
19                  Non FIPS Ciphers: DISABLED
20
21                  Cipher Redirect: DISABLED
22
23                  SSL Redirect: DISABLED
24
25                  Send Close-Notify: YES
26
27                  Strict Sig-Digest Check: DISABLED
28
29                  Push Encryption Trigger: Always
30
31                  PUSH encryption trigger timeout:              1 ms
32
33                  SNI: DISABLED
34
35                  OCSP Stapling: DISABLED
36
37                  Strict Host Header check for SNI enabled SSL sessions:
                                          NO
38
39                  Push flag:             0x0 (Auto)
40
41                  SSL quantum size:                             8 kB
```



```
42
43                  Encryption trigger timeout           100 mS
44
45                  Encryption trigger packet count:               45
46
47                  Subject/Issuer Name Insertion Format: Unicode
48
49                  SSL Interception: ENABLED
50
51                  SSL Interception OCSP Check: ENABLED
52
53                  SSL Interception End to End Renegotiation: ENABLED
54
55                  SSL Interception Maximum Reuse Sessions per Server:  10
56
57                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
58
59                  HSTS: DISABLED
60
61                  HSTS IncludeSubDomains: NO
62
63                  HSTS Max-Age: 0
64
65                  ECC Curve: P_256, P_384, P_224, P_521
66
67 1)             Cipher Name: DEFAULT Priority :1
68
69                  Description: Predefined Cipher Alias
70
71 1)             SSL Interception CA CertKey Name: swg_ca_cert
72 <!--NeedCopy-->
```

ユーザー ID 管理

January 31, 2022

セキュリティ違反の増加とモバイルデバイスの人気の高まりにより、外部インターネットの使用が企業ポリシーに準拠していることを確認する必要性が強調されています。許可されたユーザーのみが、企業の担当者によってプロビジョニングされた外部リソースへのアクセスを許可される必要があります。ID 管理は、個人またはデバイスの ID を確認することで可能になります。これは、個人が取ることができるタスクや個人が参照できるファイルを決定するものではありません。

SSL 転送プロキシ展開では、インターネットへのアクセスを許可する前にユーザーを識別します。ユーザーからのすべての要求と応答が検査されます。ユーザーアクティビティがログに記録され、レポート用に Citrix Application Delivery Management（ADM）にレコードがエクスポートされます。Citrix ADM では、ユーザーのアクティビティ、トランザクション、帯域幅消費に関する統計を表示できます。

デフォルトでは、ユーザーの IP アドレスのみが保存されますが、ユーザーに関する詳細を記録するように機能を構成できます。この ID 情報を使用して、特定のユーザー向けのより豊富なインターネット使用ポリシーを作成できます。

Citrix ADC アプライアンスは、明示的なプロキシ構成に対して次の認証モードをサポートしています。

- ライトウェイトディレクトリアクセスプロトコル **(LDAP)**。外部 LDAP 認証サーバーを介してユーザーを認証します。詳細については、「LDAP 認証ポリシー」を参照してください。
- **RADIUS**。外部 RADIUS サーバーを介してユーザを認証します。詳細については、「RADIUS 認証ポリシー」を参照してください。
- **TACACS+**. 外部ターミナルアクセスコントローラアクセスコントロールシステム (TACACS) 認証サーバを介してユーザを認証します。詳細については、「認証ポリシー」を参照してください。
- **Negotiate**. Kerberos 認証サーバを使用してユーザを認証します。Kerberos 認証でエラーが発生した場合、アプライアンスは NTLM 認証を使用します。詳細については、「認証ポリシーのネゴシエート」を参照してください。

透過プロキシの場合、IP ベースの LDAP 認証のみがサポートされます。クライアント要求を受信すると、プロキシは ActiveDirectory 内のクライアント IP アドレスのエントリをチェックすることによってユーザーを認証します。次に、ユーザーの IP アドレスに基づいてセッションを作成します。ただし、LDAP アクションで ssoNameAttribute を構成すると、IP アドレスの代わりにユーザー名を使用してセッションが作成されます。トランスペアレントプロキシ設定では、従来のポリシーは認証ではサポートされません。

> 注
>
> 明示的なプロキシの場合は、LDAP ログイン名を **sAMAccountName** に設定する必要があります。透過プロキシの場合、LDAP ログイン名を ネットワークアドレス、属性 1 を **sAMAccountName** に設定する必要があります。

明示的なプロキシの例:

```
1 add authentication ldapAction swg-auth-action-explicit -serverIP
      10.105.157.116 -ldapBase ”CN=Users,DC=CTXNSSFB,DC=COM” -ldapBindDn ”
      CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM” -ldapBindDnPassword
      freebsd123$ -ldapLoginName sAMAccountName
2 <!--NeedCopy-->
```

透過プロキシの例:


```
add authentication ldapAction swg-auth-action-explicit -serverIP
    10.105.157.116 -ldapBase ”CN=Users,DC=CTXNSSFB,DC=COM” -ldapBindDn ”
    CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM” -ldapBindDnPassword
    freebsd123$ -ldapLoginName networkAddress -authentication disable -
    Attribute1 sAMAccountName
<!--NeedCopy-->
```

CLI を使用したユーザー認証の設定

コマンドプロンプトで次のように入力します。

```
add authentication vserver <vserver name> SSL

bind ssl vserver <vserver name> -certkeyName <certkey name>

add authentication ldapAction <action name> -serverIP <ip_addr> -
    ldapBase <string> -ldapBindDn <string> -ldapBindDnPassword -
    ldapLoginName <string>

add authentication Policy <policy name> -rule <expression> -action <
    string>

bind authentication vserver <vserver name> -policy <string> -priority <
    positive_integer>

set cs vserver <name> -authn401 ON -authnVsName <string>
<!--NeedCopy-->
```

引数:

仮想サーバ名:

ポリシーをバインドする認証仮想サーバの名前。

最大長：127

serviceType:

認証仮想サーバのプロトコルタイプ。Always SSL.

指定可能な値:SSL

デフォルト値：SSL

アクション名:

新しい LDAP アクションの名前。文字、数字、またはアンダースコア文字 (_) で始まり、文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等しい (=)、コロン (:)、およびアンダースコア文字のみを含める必要があります。LDAP アクションが追加された後は変更できません。次の要件は、CLI だけに適用されます。

名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「認証アクション」や「認証アクション」など)。

最大長：127

serverIP:

LDAP サーバに割り当てられた IP アドレス。

ldapBase:

LDAP 検索を開始するベース（ノード）。LDAP サーバーがローカルで実行されている場合、base のデフォルト値は dc =netscaler、dc=com. 最大長：127

ldapBindDn:

LDAP サーバーへのバインドに使用される完全識別名 (DN)。

デフォルト：cn=Manager,dc=netscaler,dc=com

最大長：127

ldapBindDnPassword:

LDAP サーバへのバインドに使用するパスワード。

最大長：127

ldapLoginName:

LDAP ログイン名属性。Citrix ADC アプライアンスは、LDAP ログイン名を使用して、外部 LDAP サーバーまたは Active Directory のクエリを実行します。最大長：127

ポリシー名:

事前認証ポリシーの名前。文字、数字、またはアンダースコア文字 (_) で始まり、文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等しい (=)、コロン (:)、およびアンダースコア文字のみを含める必要があります。AUTHENTICATION ポリシーの作成後に変更することはできません。次の要件は、CLI だけに適用されます。

名前に 1 つ以上のスペースが含まれる場合は、名前を二重引用符または一重引用符で囲みます (「認証ポリシー」や「認証ポリシー」など)。

最大長：127

rule を次のように設定します。

AUTHENTICATION サーバーでユーザーを認証するかどうかを決定するためにポリシーが使用する規則の名前、またはデフォルトの構文式。

最大長：1499

action。

ポリシーが一致した場合に実行される認証アクションの名前。

最大長：127

priority:

ポリシーのプライオリティを指定する正の整数。数値が小さいほど、プライオリティが高くなります。ポリシーは優先度の順に評価され、要求に一致する最初のポリシーが適用されます。認証仮想サーバにバインドされたポリシーのリスト内で一意である必要があります。

最小値:0

最大値:4294967295

例:

```
add authentication vserver swg-auth-vs SSL

Done

bind ssl vserver explicit-auth-vs -certkeyName ns-swg-ca-certkey

Done

add authentication ldapAction swg-auth-action-explicit -serverIP
    192.0.2.116 -ldapBase ”CN=Users,DC=CTXNSSFB,DC=COM” -ldapBindDn ”CN=
    Administrator,CN=Users,DC=CTXNSSFB,DC=COM” -ldapBindDnPassword zzzzz
     -ldapLoginName sAMAccountName

Done

add authenticationpolicy swg-auth-policy -rule true -action swg-auth-
    action-explicit
Done

bind authentication vserver swg-auth-vs -policy swg-auth-policy -
    priority 1

Done

set cs vserver testswg -authn401 ON -authnVsName swg-auth-vs

Done
```



```
23 <!--NeedCopy-->
```

CLI を使用したユーザー名のロギングの有効化

コマンドプロンプトで入力します。

```
1 set appflow param -AAAUserName ENABLED
2 <!--NeedCopy-->
```

引数:

AAAUserName

AppFlow の認証、承認、およびユーザー名のログ記録の監査を有効にします。

設定可能な値：ENABLED, DISABLED

デフォルト値：無効

例:

```
1 set appflow param -AAAUserName ENABLED
2 <!--NeedCopy-->
```

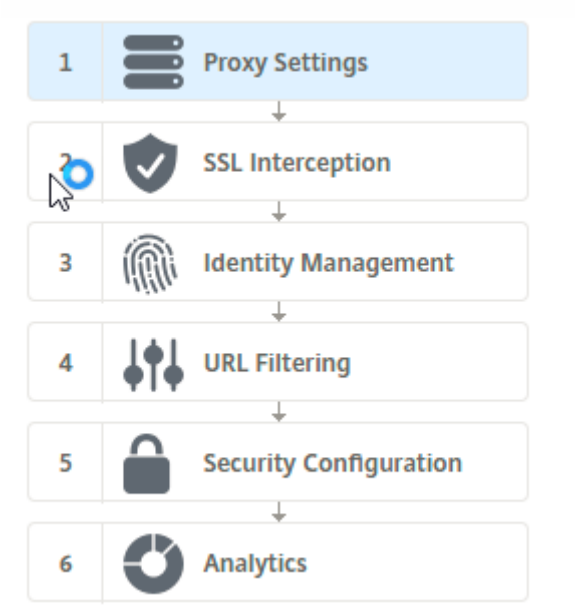
URL フィルタリング

October 7, 2021

URL フィルタリングは、URL に含まれる情報を使用して、Web サイトのポリシーベースの制御を提供します。この機能は、ネットワーク管理者がネットワーク上の悪意のある Web サイトへのユーザーアクセスを監視および制御するのに役立ちます。

はじめに

新規ユーザで URL フィルタリングを設定する場合は、SSL 転送プロキシの初期設定を完了する必要があります。URL フィルタリングを開始するには、まず SSL 転送プロキシウィザードにログオンする必要があります。ウィザードでは、URL フィルタリングポリシーを適用する前に、一連の構成手順を実行します。



> 注
>
> 開始する前に、有効な URL 脅威インテリジェンス機能ライセンスがアプライアンスにインストールされていることを確認してください。試用版を使用している場合は、ADC アプライアンスでこの機能を引き続き使用するには、有効なライセンスを購入してください。

SSL 転送プロキシウィザードにログオンします

SSL フォワードプロキシウィザードは、一連の簡略化された構成タスクをガイドし、右側のペインに対応するフローシーケンスが表示されます。このウィザードを使用して、URL フィルタリングポリシーを URL リストまたは定義済みのカテゴリの一覧に適用できます。

手順 **1:** プロキシ設定を構成する

まず、クライアントがゲートウェイにアクセスするためのプロキシサーバーを構成します。このサーバは SSL タイプであり、明示的または透過的モードで動作します。プロキシサーバー構成の詳細については、「プロキシモード」を参照してください。

ステップ **2: SSL** インターセプションを設定する

プロキシサーバーを構成したら、Citrix ADC アプライアンスで暗号化されたトラフィックを代行受信するように SSL インターセプションプロキシを構成する必要があります。URL フィルタリングの場合、SSL プロキシはトラフィックをインターセプトし、ブロックされた URL を許可しませんが、他のすべてのトラフィックはバイパスできます。SSL 代行受信の設定の詳細については、「SSL インターセプション」を参照してください。

ステップ **3: ID** 管理を構成する

ユーザーは、エンタープライズネットワークへのログオンを許可される前に認証されます。認証では、役割に基づいて、ユーザーまたはユーザーのグループに対して特定のポリシーを柔軟に定義できます。ユーザー認証の詳細については、「ユーザー識別管理」を参照してください。

手順 **4: URL** フィルタリングを構成する

管理者は、URL 分類機能または URL リスト機能を使用して、URL フィルタリングポリシーを適用できます。

URL 分類。事前定義されたカテゴリのリストに基づいてトラフィックをフィルタリングすることにより、Web サイトおよび Web ページへのアクセスを制御します。

URL リスト。アプライアンスにインポートされた URL セットに含まれる URL へのアクセスを拒否することにより、ブラックリストに登録された Web サイトおよび Web ページへのアクセスを制御します。

手順 **5:** セキュリティ構成を構成する

この手順では、レピュテーションスコアを設定し、スコアが低すぎる場合にアクセスを拒否することで、ユーザーがウェブサイトへのアクセスを制御できるようにします。レピュテーションスコアは 1 ～4 の範囲で、スコアが許容できないしきい値を設定できます。しきい値を超えるスコアの場合は、トラフィックを許可する、ブロック、またはリダイレクトするポリシーアクションを選択できます。詳細については、URL レピュテーションスコアを参照してください。

ステップ **6: SSL** 転送プロキシ分析を構成する

この手順では、Web トラフィックの分類、ユーザトランザクションログの URL カテゴリのロギング、トラフィック分析の表示のために SSL 転送プロキシ分析を有効にできます。SSL フォワードプロキシ分析の詳細については、「Analytics」を参照してください。

ステップ **7: [**完了**]** をクリックして初期構成を完了し、**URL** フィルタリング構成の管理を続行します

URL リスト

October 7, 2021

URL リスト機能を使用すると、企業のお客様は、特定の Web サイトおよび Web サイトのカテゴリへのアクセスを制御できます。この機能は、URL 照合アルゴリズムにバインドされたレスポンダーポリシーを適用して Web サイトをフィルタリングします。このアルゴリズムは、着信 URL を、最大 100 万（1,000,000）のエントリで構成される URL セットと照合します。着信 URL 要求がセット内のエントリと一致する場合、アプライアンスは応答側ポリシーを使用して要求（HTTP/HTTPS）を評価し、その要求へのアクセスを制御します。

URL セットのタイプ

URL セット内の各エントリには、URL と、必要に応じてそのメタデータ (URL カテゴリ、カテゴリグループ、またはその他の関連データ) を含めることができます。メタデータを含む URL の場合、アプライアンスはメタデータを評価するポリシー式を使用します。詳細については、「URL セット」を参照してください。

SSL フォワードプロキシはカスタム URL セットをサポートします。パターンセットを使用して URL をフィルタリングすることもできます。

カスタム **URL** セット。最大 1,000,000 個の URL エントリを含むカスタマイズされた URL セットを作成し、それをテキストファイルとしてアプライアンスにインポートできます。

パターンセット。ADC アプライアンスは、Web サイトへのアクセスを許可する前に、パターンセットを使用して URL をフィルタリングできます。パターンセットは、着信 URL と最大 5000 エントリの間で完全に一致する文字列を検索する文字列マッチングアルゴリズムです。詳細については、パターンセットを参照してください。

読み込んだ URL セットの各 URL には、URL メタデータの形式でカスタムカテゴリを設定できます。組織では、セットをホストし、ADC アプライアンスを設定して、手動で介入しなくてもセットを定期的に更新できます。

セットが更新されると、Citrix ADC アプライアンスは自動的にメタデータを検出します。このカテゴリは、URL を評価し、許可、ブロック、リダイレクト、ユーザーへの通知などのアクションを適用するためのポリシー式として使用できるようになりました。

URL セットで使用される高度なポリシー式

次の表に、着信トラフィックの評価に使用できる基本的な式を示します。

1. .URLSET_MATCHES_ANY-URL が URL セット内のエントリと完全に一致する場合に TRUE と評価されます。
2. .GET_URLSET_METADATA ()-GET_URLSET_METADATA () 式は、URL セット内の任意のパターンに正確に一致する場合に、関連するメタデータを返します。一致しない場合は、空の文字列が返されます。
3. `.GET_ URLSET_METADATA().EQ(<METADATA)- .GET_ URLSET_METADATA().EQ(<METADATA)`
4. .GET_URLSET_METADATA () .TYPECAST_LIST_T (‘,’) .GET (0) .EQ ()-一致するメタデータがカテゴリの先頭にある場合は TRUE と評価されます。このパターンは、メタデータ内の個別のフィールドをエンコードするために使用できますが、最初のフィールドのみに一致します。
5. HTTP.REQ.HOSTNAME.APPEND（HTTP.REQ.URL）-ホストと URL のパラメーターを結合します。これらのパラメーターは、照合に使用できます。

レスポンダーのアクションの種類

> 注：この表では、HTTP.REQ.URL は`<URL expression>`として一般化されています。

次の表に、着信インターネットトラフィックに適用できるアクションを示します。

レスポンダのアクション	説明
許可	リクエストにターゲット URL へのアクセスを許可します。
リダイレクト	ターゲットとして指定された URL にリクエストをリダイレクトします。
ブロック	要求を拒否します。

前提条件

ホスト名 URL から URL セットをインポートする場合は、DNS サーバーを構成します。IP アドレスを使用する場合、この構成は必要ありません。

コマンドプロンプトで入力します。

```
add dns nameServer ((<IP> [-local])| <dnsVserverName>)[-state (ENABLED |
DISABLED )] [-type <type>] [-dnsProfileName <string>]
```

例:

```
add dns nameServer 10.140.50.5
```

URL リストを構成する

URL リストを構成するには、Citrix SSL 転送プロキシウィザードまたは Citrix ADC コマンドラインインターフェイス（CLI）を使用します。Citrix ADC アプライアンスでは、まずレスポンダーポリシーを構成してから、URL セットにポリシーをバインドする必要があります。

URL リストを構成するには、Citrix SSL 転送プロキシウィザードを優先オプションとして使用することをお勧めします。ウィザードを使用して、レスポンダーポリシーを URL セットにバインドします。または、ポリシーをパターンセットにバインドすることもできます。

SSL 転送プロキシウィザードを使用して URL 一覧を構成する

GUI を使用して HTTPS トラフィックの URL リストを設定するには、次の手順を実行します。

1. [セキュリティ] > [**SSL** 転送プロキシ] ページに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   a) [**SSL** 転送プロキシウィザード**]** をクリックします。
   b) 既存の構成を選択し、[**Edit**] をクリックします。
3. [**URL** フィルタリング] セクションで、[編集] をクリックします。
4. 機能を有効にするには、[**URL** リスト] チェックボックスをオンにします。

5. **URL** リストポリシーを選択し、[バインド] をクリックします。
6. [続行] をクリックし、[完了] をクリックします。

詳細については、「URL リストポリシーを作成する方法」を参照してください。

CLI を使用した **URL** リストの設定

URL リストを設定するには、次の手順を実行します。

1. HTTP および HTTPS トラフィック用のプロキシ仮想サーバーを構成します。
2. HTTPS トラフィックを代行受信するための SSL インターセプションを設定します。
3. HTTP トラフィック用の URL セットを含む URL リストを設定します。
4. HTTPS トラフィックに設定された URL を含む URL リストを設定します。
5. プライベート URL セットを設定します。

> 注
>
> すでに ADC アプライアンスを設定している場合は、ステップ 1 と 2 をスキップして、ステップ 3 で設定できます。

インターネットトラフィック用のプロキシ仮想サーバーの構成

Citrix ADC アプライアンスは、透過的および明示的なプロキシ仮想サーバーをサポートします。エクスプリシットモードでインターネットトラフィック用のプロキシ仮想サーバーを構成するには、次の手順を実行します。

1. プロキシ SSL 仮想サーバーを追加します。
2. レスポンダーポリシーをプロキシ仮想サーバーにバインドします。

CLI を使用してプロキシ仮想サーバーを追加するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
add cs vserver <name> <serviceType> <IPAddress> <port>
<!--NeedCopy-->
```

例:

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
<!--NeedCopy-->
```

CLI を使用してレスポンダポリシーをプロキシ仮想サーバにバインドするには、次の手順を実行します。


```
bind ssl vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

> 注
>
> Citrix ADC 構成の一部として SSL インターセプターを既に構成している場合は、次の手順をスキップできます。

HTTPS トラフィックの **SSL** インターセプションの設定

HTTPS トラフィックの SSL インターセプションを設定するには、次の手順を実行します。

1. CA 証明書とキーのペアをプロキシ仮想サーバにバインドします。
2. デフォルトの SSL プロファイルを有効にします。
3. フロントエンド SSL プロファイルを作成し、プロキシ仮想サーバーにバインドし、フロントエンド SSL プロファイルで SSL インターセプションを有効にします。

CLI を使用して CA 証明書とキーのペアをプロキシ仮想サーバにバインドするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
<!--NeedCopy-->
```

CLI を使用してフロントエンド SSL プロファイルを構成するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
set ssl parameter -defaultProfile ENABLED

add ssl profile <name> -sslInterception ENABLED -ssliMaxSessPerServer <
    positive_integer>
<!--NeedCopy-->
```

CLI を使用してフロントエンド SSL プロファイルをプロキシ仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
set ssl vserver <vServer name>  -sslProfile <name>
<!--NeedCopy-->
```


HTTP トラフィックの **URL** セットをインポートして **URL** リストを構成する

HTTP トラフィックの URL セットを設定する方法については、「URL セット」を参照してください。

明示的なサブドメイン一致の実行

インポートされた URL セットに対して明示的なサブドメイン照合を実行できるようになりました。新しいパラメータ「subdomainExactMatch」が **import** policy URLset コマンドに追加されました。

パラメータを有効にすると、URL フィルタリングアルゴリズムは明示的なサブドメイン一致を実行します。たとえば、着信 URL がnews.example.comで、URL セットのエントリがexample.comの場合、アルゴリズムは URL と一致しません。

コマンドプロンプトで入力します。

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-rowSeparator
 <character>] -url [-interval <secs>] [-privateSet][-subdomainExactMatch]
[-canaryUrl <URL>]
```

例

```
import policy urlset test -url http://10.78.79.80/top-1k.csv -privateSet -
subdomainExactMatch -interval 900
```

HTTPS トラフィック用の **URL** セットを構成する

CLI を使用して HTTPS トラフィックの URL セットを構成するには

コマンドプロンプトで次のように入力します。

```
1 add ssl policy <name> -rule <expression> -action <string> [-undefAction
      <string>] [-comment <string>]
2 <!--NeedCopy-->
```

例:

```
1 add ssl policy pol1 -rule ”client.ssl.client_hello.SNI.
     URLSET_MATCHES_ANY(”top1m”) -action INTERCEPT
2 <!--NeedCopy-->
```

SSL 転送プロキシウィザードを使用して **HTTPS** トラフィックの **URL** セットを構成するには

URL リストを構成するには、SSL 転送プロキシウィザードを優先オプションとして使用することをお勧めします。ウィザードを使用して、カスタム URL セットをインポートし、レスポンダーポリシーにバインドします。

1. [セキュリティ] **>** **[SSL** 転送プロキシ**]** **>** **[URL** フィルタリング**]** **>** **[URL** リスト] に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**URL** リストポリシー] ページで、ポリシー名を指定します。
4. URL セットをインポートするオプションを選択します。
5. [**URL** リストポリシー] タブページで、[**URL** セットのインポート] チェックボックスをオンにし、次の URL セットパラメータを指定します。
   a) URL セット名-カスタム URL セットの名前。
   b) URL：URL セットにアクセスする場所の Web アドレス。
   c)「上書き」(Overwrite)-以前にインポートした URL セットを上書きします。
   d) Delimiter：CSV ファイルレコードを区切る文字シーケンス。
   e) 行区切り文字—CSV ファイルで使用される行区切り文字。
   f) [Interval]：URL セットが更新される 15 分に最も近い秒数に切り捨てられた間隔（秒単位）。
   g) プライベートセット—URL セットのエクスポートを防止するオプション。
   h) カナリア URL-URL セットのコンテンツが機密扱いかどうかをテストするための内部 URL。URL の最大長は 2047 文字です。
6. ドロップダウンリストから応答者のアクションを選択します。
7. **[**作成**]** して **[**閉じる**]** をクリックします。

プライベート **URL** セットの構成

プライベート URL セットを設定し、その内容を秘密にしておくと、ネットワーク管理者はセット内のブラックリスト URL を知らないことがあります。そのような場合は、カナリアの URL を設定し、URL セットに追加できます。管理者は Canary URL を使用して、すべてのルックアップ要求で使用するプライベート URL セットをリクエストできます。各パラメータの説明については、ウィザードのセクションを参照してください。

CLI を使用して URL セットをインポートするには、次の手順を実行します。

コマンドプロンプトで入力します。

```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
    rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
    ] [-canaryUrl <URL>]
<!--NeedCopy-->
```

例:

```
import policy urlset test1  - url http://10.78.79.80/alytra/top-1k.csv -
    private -canaryUrl http://www.in.gr
<!--NeedCopy-->
```


インポートされた **URL** セットを表示

追加された URL セットに加えて、読み込んだ URL セットも表示できるようになりました。`show urlset` コマンドに新しいパラメータ「imported」が追加されました。このオプションを有効にすると、アプライアンスはインポートされたすべての URL セットを表示し、インポートされた URL セットと追加された URL セットを区別します。

コマンドプロンプトで入力します。

```
show policy urlset [<name>] [-imported]
```

例

```
show policy urlset -imported
```

監査ログメッセージの構成

監査ログを使用すると、URL リストプロセスの任意のフェーズで状態または状況を確認できます。Citrix ADC アプライアンスが着信 URL を受信したときに、レスポンダーポリシーに URL セットの高度なポリシー式がある場合、監査ログ機能は URL セット情報を URL に収集します。詳細は、監査ログで許可されているすべてのターゲットのログメッセージとして保存されます。

ログメッセージには、次の情報が含まれています。

1. タイムスタンプ。
2. ログメッセージのタイプ。
3. 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)。
4. URL セット名、ポリシーアクション、URL などのログメッセージ情報。

URL リスト機能の監査ログを設定するには、次の作業を完了する必要があります。

1. 監査ログを有効化。
2. 監査ログメッセージの作成アクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ 」トピックを参照してください。

URL パターンセマンティクス

October 7, 2021

次の表に、フィルタリングするページのリストを指定するために使用する URL パターンを示します。たとえば、www.example.com/bar というパターンは、www.example.com/bar の 1 ページだけに一致します。URL が ‘www.example.com/bar’ で始まるすべてのページを一致させるには、URL の末尾にアスタリスク (*) を追加します。

メタデータマッピングにマッチする **URL** パターンのセマンティクス

パターンマッチングセマンティクスは、テーブル形式で使用できます。詳細については、パターンセマンティクス PDF ページを参照してください。

URL カテゴリのマッピング

October 7, 2021

サードパーティのカテゴリとカテゴリグループのリスト。詳細については、「URL カテゴリマッピング 」ページを参照してください。

ユースケース**:** カスタム **URL** セットを使用した **URL** フィルタリング

April 25, 2022

特定の Web サイトや Web サイトカテゴリへのアクセスを制御したいと考えている企業のお客様は、レスポンダーポリシーにバインドされたカスタム URL セットを使用します。組織のネットワークインフラストラクチャでは、URL フィルタを使用して、悪意のある Web サイトや危険な Web サイトへのアクセスをブロックできます。たとえば、成人向け、暴力、ゲーム、麻薬、政治、求人ポータルを取り上げた Web サイトなどです。URL のフィルタリングに加えて、カスタマイズされた URL のリストを作成して ADC アプライアンスにインポートできます。たとえば、組織のポリシーで、ソーシャルネットワーキング、ショッピングポータル、求人ポータルなどの特定の Web サイトへのアクセスをブロックするよう求められる場合があります。

リスト内の各 URL は、メタデータの形式でカスタムカテゴリを持つことができます。組織は、Citrix ADC アプライアンス上で URL セットとして URL のリストをホストできます。手動で操作しなくても、セットを定期的に更新するようにアプライアンスを設定します。

セットが更新されると、Citrix ADC アプライアンスはメタデータを自動的に検出します。レスポンダーポリシーは URL メタデータ (カテゴリの詳細) を使用して受信 URL を評価し、許可、ブロック、リダイレクト、ユーザーへの通知などのアクションを適用します。

そのためには、ネットワークで設定し、次のタスクを実行できます。

1. カスタム URL セットをインポートする
2. カスタム URL セットの追加
3. SSL Forward Proxy ウィザードでカスタム URL リストを設定します。

CLI を使用してカスタム **URL** セットをインポートする

コマンドプロンプトで入力します。


```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
    rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
    ] [-canaryUrl <URL>]

import policy urlset test1  - url http://10.78.79.80/alytra/top-1k.csv
<!--NeedCopy-->
```

CLI を使用してカスタム URL セットを追加する

コマンドプロンプトで入力します。

```
add urlset <urlset_name>
```

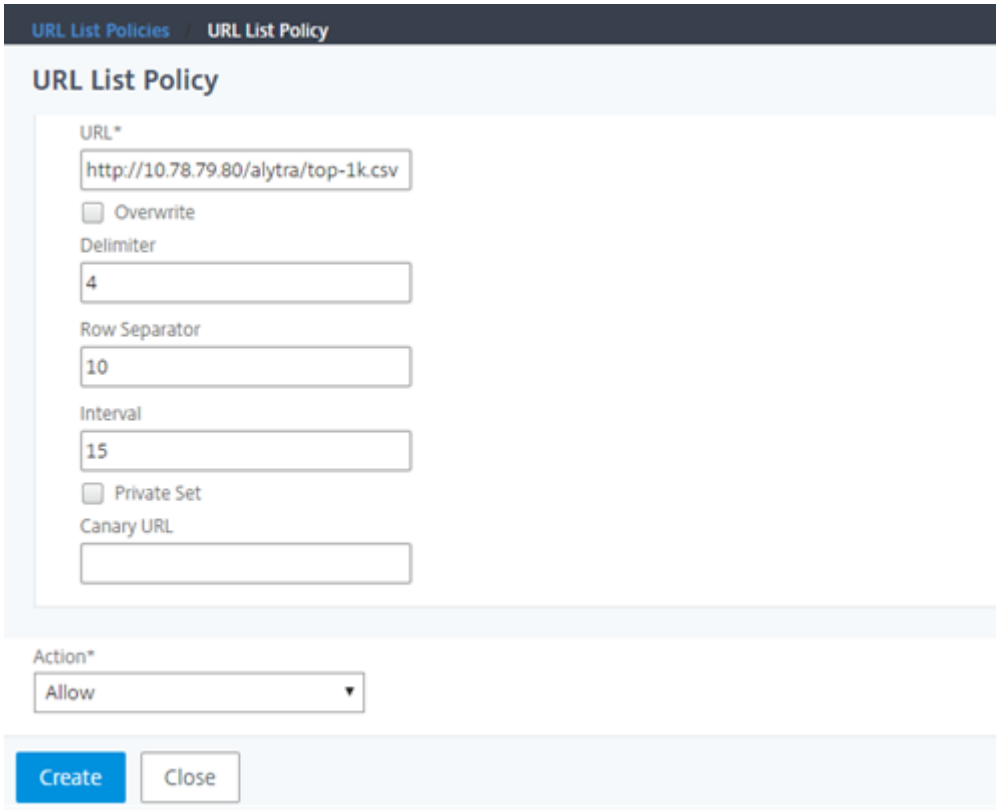
例:

```
add urlset test1
```

SSL Forward Proxy ウィザードを使用して URL リストを構成する

URL リストを構成するには、SSL 転送プロキシウィザードを優先オプションとして使用することをお勧めします。ウィザードを使用して、カスタム URL セットをインポートし、レスポンダーポリシーにバインドします。

1. セキュリティ **>** **SSL** 転送プロキシ **>** **URL** フィルタリング **>** **URL** リストに移動します。
2. 詳細ウィンドウで、[追加] をクリックします。
3. [**URL** リストポリシー] ページで、ポリシー名を指定します。
4. URL セットをインポートするオプションを選択します。
5. [**URL** リストポリシー] タブページで、[**URL** セットのインポート] チェックボックスをオンにし、次の URL セットパラメータを指定します。
   a) URL セット名-カスタム URL セットの名前。
   b) URL：URL セットにアクセスするロケーションの Web アドレス。
   c) [上書き]-以前にインポートした URL セットを上書きします。
   d) 区切り記号-CSV ファイルレコードを区切る文字シーケンス。
   e) 行セパレータ-CSV ファイルで使用される行セパレータ。
   f) [Interval]：URL セットが更新される間隔（秒）。15 分単位で四捨五入されます。
   g) [Private Set]：URL セットのエクスポートを禁止するオプション。
   h) Canary URL：URL セットのコンテンツの機密を保持するかどうかをテストするための内部 URL。URL の最大長は 2047 文字です。
6. ドロップダウンリストからレスポンダーアクションを選択します。
7. **[**作成**]** して **[**閉じる**]** をクリックします。



カスタム **URL** セットのメタデータセマンティクス

カスタム URL セットをインポートするには、URL をテキストファイルに追加し、ソーシャルネットワーキング URL をブロックするレスポンダーポリシーにバインドします。

テキストファイルに追加できる URL の例を次に示します。

cnn.com, ニュース

bbc.com, ニュース

google.com、検索エンジン

yahoo.com, 検索エンジン

facebook.com, ソーシャルメディア

twitter.com, ソーシャルメディア

CLI を使用してソーシャルメディアの **URL** をブロックするレスポンダーポリシーを設定する

```
1 add responder action act_url_unauthorized respondwith '"HTTP/1.1 451
     Unavailable For Legal Reasons\r\n\r\nURL is NOT authorized\n"'
2
```



```
add responder policy pol_url_meta_match 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).GET_URLSET_METADATA(”u1”).EQ(”Social Media”)' act_url_unauthorized
<!--NeedCopy-->
```

URL の分類

May 9, 2022

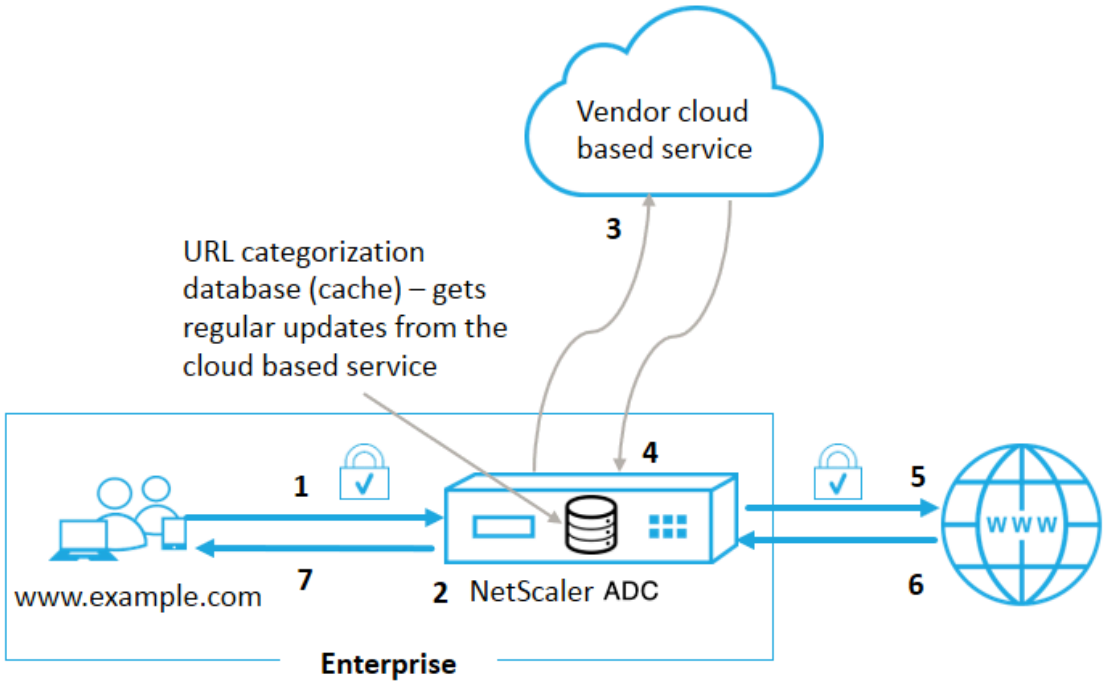
URL の分類は、特定の Web サイトおよび Web サイトのカテゴリへのユーザーアクセスを制限します。NetSTARと連携したサブスクライブサービスであるこの機能により、企業のお客様は、市販の分類データベースを使用して Web トラフィックをフィルタリングできます。NetSTARデータベースには、ソーシャルネットワーキング、ギャンブル、アダルトコンテンツ、ニューメディア、ショッピングなど、さまざまなカテゴリに分類された膨大な数（数十億）の URL があります。分類に加えて、各 URL には、サイトの履歴リスクプロファイルに基づいて最新のレピュテーションスコアが保持されます。カテゴリ、カテゴリグループ（テロ、違法薬物など）、またはサイトレピュテーションスコアに基づいて高度なポリシーを設定することで、NetSTARデータを使用してトラフィックをフィルタリングできます。

たとえば、マルウェアに感染していることがわかっているサイトなど、危険なサイトへのアクセスをブロックできます。また、エンタープライズユーザーに対して、アダルトコンテンツやエンターテイメントストリーミングメディアなどのコンテンツへのアクセスを選択的に制限することもできます。また、ユーザーのトランザクションの詳細と送信トラフィックの詳細をキャプチャして、Citrix ADM サーバー上の Web トラフィック分析を監視することもできます。

Citrix ADC は、事前構成された NetSTAR デバイス nsv10.netstar-inc.com および incompasshybridpc.netstar-inc.com からデータをアップロードまたはダウンロードし、クラウド分類要求のデフォルトでクラウドホストとして使用されます。URL フィルタリングが正しく機能するには、これらの URL にファイアウォールを介してアクセス可能である必要があります。アプライアンスは、NSIP アドレスを送信元 IP アドレスとして使用し、443 を通信の宛先ポートとして使用します。

URL 分類の仕組み

次の図は、Citrix ADC URL 分類サービスが商用 URL 分類データベースおよびクラウドサービスと統合され、頻繁に更新される様子を示しています。



コンポーネントは次のように相互作用します。

1. クライアントは、インターネットにバインドされた URL 要求を送信します。
2. SSL フォワードプロキシは、カテゴリ、カテゴリグループ、サイトレピュテーションスコアなどのカテゴリの詳細に基づいて、要求にポリシー適用を適用します。カテゴリの詳細は URL 分類データベースから取得されます。データベースがカテゴリの詳細を返す場合、プロセスは手順 5 にジャンプします。
3. データベースで分類の詳細が見つからない場合、URL 分類ベンダーが管理するクラウドベースの検索サービスにリクエストが送信されます。ただし、アプライアンスは応答を待たずに、URL が未分類としてマークされ、ポリシーの強制が実行されます（ステップ 5 にジャンプします）。アプライアンスは、クラウドクエリーフィードバックを監視し続け、キャッシュを更新して、今後の要求がクラウドルックアップの恩恵を受けることができるようにします。
4. ADC アプライアンスは、クラウドベースのサービスから URL カテゴリの詳細（カテゴリ、カテゴリグループ、レピュテーションスコア）を受信し、分類データベースに保存します。
5. ポリシーでは URL が許可され、リクエストはオリジンサーバーに送信されます。それ以外の場合、アプライアンスはカスタム HTML ページを使用してドロップ、リダイレクト、または応答します。
6. オリジンサーバーは、要求されたデータで ADC アプライアンスに応答します。
7. アプライアンスは応答をクライアントに送信します。

ユースケース**:** 企業コンプライアンス下での企業向けインターネット利用

URL フィルタリング機能を使用して、コンプライアンスポリシーを検出して実装し、企業のコンプライアンスに違反するサイトをブロックできます。たとえば、アダルトサイト、ストリーミングメディア、ソーシャルネットワーキングなど、非生産的と見なされるサイトや、企業ネットワークで過剰なインターネット帯域幅を消費するサイト。これらの Web サイトへのアクセスをブロックすると、従業員の生産性が向上し、帯域幅使用の運用コストが削減され、ネットワーク消費のオーバーヘッドが削減されます。

前提条件

URL 分類機能は、URL フィルタリング機能と SSL フォワードプロキシの脅威インテリジェンスを備えたオプションのサブスクリプションサービスがある場合にのみ、Citrix ADC プラットフォームで機能します。サブスクリプションにより、顧客は Web サイトの最新の脅威分類をダウンロードし、そのカテゴリを SSL フォワードプロキシに適用できます。この機能を有効にして設定する前に、次のライセンスをインストールする必要があります。

- CNS_WEBF_SSERVER_Retail.lic
- CNS_XXXX_SERVER_PLT_Retail.lic

ここで、XXXXX はプラットフォーム・タイプです（例：V25000）。

レスポンダーポリシー式

次の表に、着信 URL を許可、リダイレクト、またはブロックする必要があるかどうかを確認するために使用できるさまざまなポリシー式を示します。

1. <text>. URL_CATEGORIZE (<min_reputation>, <max_reputation>) -URL_CATEGORY オブジェクトを返します。<min_reputation>が 0 より大きい場合、返されるオブジェクトには<min_reputation>以下の評価を持つカテゴリは含まれません。<max_reputation>が 0 より大きい場合、返されるオブジェクトには、<max_reputation>より高い評価を持つカテゴリは含まれません。カテゴリがタイムリーに解決に失敗した場合は、undef 値が返されます。
2. <url_category>. CATEGORY() -このオブジェクトのカテゴリ文字列を返します。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「Unknown」になります。
3. <url_category>. CATEGORY_GROUP() -オブジェクトのカテゴリグループを識別する文字列を返します。このグループ化は、カテゴリの上位レベルのグループ化であり、URL カテゴリに関する詳細情報が少ない操作で役立ちます。URL にカテゴリがない場合、または URL の形式が正しくない場合、戻り値は「Unknown」になります。
4. <url_category>. REPUTATION() -レピュテーションスコアを 0 ～5 の数値として返します。5 は最もリスクの高いレピュテーションを示します。カテゴリ「不明」がある場合、評価値は 1 です。

ポリシータイプ**:**

1. 検索エンジンのカテゴリに含まれる URL のリクエストを選択するポリシー- `add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).`

CATEGORY().EQ(“Search Engine”)

2. Adult カテゴリグループに含まれる URL のリクエストを選択するポリシー- add responder policy p1 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY_GROUP().EQ(“Adult”)’
3. レピュテーションスコアが 4 未満の検索エンジン URL のリクエストを選択するポリシー- add responder policy p2 ‘HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL).URL_CATEGORIZE (4,0).HAS_CATEGORY(“Search Engine”)
4. 検索エンジンとショッピング URL のリクエストを選択するポリシー- add responder policy p3 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).CATEGORY(). EQ (“good_categories”)
5. レピュテーションスコアが 4 以上の検索エンジン URL のリクエストを選択するポリシー- add responder policy p5 ‘CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(4,0). CATEGORY() .EQ(“Search Engines”)
6. 検索エンジンのカテゴリにある URL のリクエストを選択し、それらを URL セットと比較するポリシー- ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY().EQ(“Search Engine”)&& HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URLSET_MATCHES_ANY(“u1”)’

レスポンダーポリシーの種類

URL 分類機能で使用されるポリシーには 2 種類あり、次の表ではこれらのポリシータイプについて説明します。

ポリシーの種類	説明
URL カテゴリ	Web トラフィックを分類し、評価結果に基づいてトラフィックをブロック、許可、またはリダイレクトします。
URL レピュテーションスコア	Web サイトのレピュテーションスコアを決定し、管理者が設定したレピュテーションスコアのしきい値レベルに基づいてアクセスを制御できるようにします。

URL 分類の構成

Citrix ADC アプライアンスで URL 分類を構成するには、次の手順を実行します。

1. URL フィルタリングを有効にします。
2. Web トラフィック用にプロキシサーバーを構成します。
3. 明示モードで Web トラフィックの SSL インターセプトを設定します。
4. 共有メモリを設定してキャッシュメモリを制限します。
5. URL 分類パラメータを設定します。

6. Citrix SSL フォワードプロキシウィザードを使用して URL 分類を構成します。
7. SSL フォワードプロキシウィザードを使用して URL 分類パラメータを設定します。
8. シードデータベースパスとクラウドサーバー名の設定

ステップ **1: URL** フィルタリングの有効化

URL 分類を有効にするには、URL フィルタリング機能を有効にし、URL 分類のモードを有効にします。

CLI を使用して URL 分類を有効にするには

コマンドプロンプトで入力します。

```
enable ns feature URLFiltering
```

```
disable ns feature URLFiltering
```

手順 **2:** 明示モードで **Web** トラフィック用のプロキシサーバーを構成する

Citrix ADC アプライアンスは、透過的で明示的なプロキシ仮想サーバーをサポートします。明示モードで SSL トラフィック用のプロキシ仮想サーバを設定するには、次の手順を実行します。

1. プロキシサーバを追加します。
2. SSL ポリシーをプロキシサーバーにバインドします。

CLI を使用してプロキシサーバを追加するには

コマンドプロンプトで入力します。

```
add cs vserver <name> [-td <positive_integer>] <serviceType> [-cltTimeout <
secs>]
```

例**:**

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
```

CLI を使用して **SSL** ポリシーをプロキシ仮想サーバーにバインドする

```
bind ssl vserver <vServerName> -policyName <string> [-priority <positive_integer
>]
```

ステップ **3: HTTPS** トラフィックの **SSL** インターセプトを構成する

HTTPS トラフィックの SSL インターセプトを設定するには、次の手順を実行します。

1. CA 証明書とキーのペアをプロキシ仮想サーバにバインドします。
2. SSL パラメータを使用して、デフォルトの SSL プロファイルを設定します。

3. フロントエンド SSL プロファイルをプロキシ仮想サーバーにバインドし、フロントエンド SSL プロファイルで SSL インターセプトを有効にします。

CLI を使用して CA 証明書とキーのペアをプロキシ仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -CA -skipCAName
```

CLI を使用してデフォルト SSL プロファイルを設定するには

コマンドプロンプトで入力します。

```
set ssl profile <name> -denySSLReneg <denySSLReneg> -sslInterception (ENABLED | DISABLED)-ssliMaxSessPerServer positive_integer>
```

CLI を使用してフロントエンド **SSL** プロファイルをプロキシ仮想サーバーにバインドする

コマンドプロンプトで入力します。

```
set ssl vserver <vServer name> -sslProfile ssl_profile_interception
```

手順 **4:** キャッシュメモリを制限するように共有メモリを構成する

CLI を使用してキャッシュメモリを制限するように共有メモリを設定するには

コマンドプロンプトで入力します。

```
set cache parameter [-memLimit <megaBytes>]
```

ここで、キャッシュ用に構成されたメモリ制限は 10 MB に設定されます。

ステップ **5: URL** 分類パラメーターの設定

CLI を使用して URL 分類パラメータを設定するには

コマンドプロンプトで入力します。

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>] [-TimeOfDayToUpdateDB <HH:MM>]
```

例**:**

```
set urlfiltering parameter -urlfilt_hours_betweenDB_updates 20
```


手順 **6**：**Citrix SSL** フォワードプロキシウィザードを使用して **URL** 分類を構成する

1. Citrix ADC アプライアンスにログオンし、[セキュリティ] > [**SSL** フォワードプロキシ] ページに移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   a) [**SSL** フォワードプロキシウィザード] をクリックして、新しい構成を作成します。
   b) 既存の設定を選択し、[**Edit**] をクリックします。
3. [**URL** フィルタ] セクションで、[編集] をクリックします。
4. この機能を有効にするには、[**URL** 分類] チェックボックスをオンにします。
5. **URL** 分類ポリシーを選択し、[バインド] をクリックします。
6. [続行] をクリックし、[完了] をクリックします。

URL 分類ポリシーの詳細については、「URL 分類ポリシーを作成する方法」を参照してください。

ステップ **7: SSL** フォワードプロキシウィザードを使用した **URL** 分類パラメータの設定

1. **Citrix ADC** アプライアンスにログオンし、[セキュリティ **] > [URL** フィルタリング] に移動します。
2. [**URL** フィルタリング] ページで、[**URL** フィルタリング設定の変更] リンクをクリックします。
3. [**URL** フィルタリングパラメータの設定] ページで、次のパラメータを指定します。
   a) DB 更新間隔の時間。URL データベース更新の間隔をフィルタリングする時間。最小値:0、最大値:720。
   b) DB を更新する時刻。URL フィルタリングでデータベースを更新する時刻。
   c) クラウドホスト。クラウドサーバーの URL パス。
   d) シード DB パス。シードデータベース検索サーバーの URL パス。
4. [**OK**] をクリックして閉じます。

サンプル構成**:**

```
enable ns feature LB CS SSL IC RESPONDER AppFlow URLFiltering

enable ns mode FR L3 Edge USNIP PMTUD

set ssl profile ns_default_ssl_profile_frontend -denySSLReneg NONSECURE
    -sslInterception ENABLED -ssliMaxSessPerServer 100

add ssl certKey swg_ca_cert -cert ns_swg_ca.crt -key ns_swg_ca.key

set cache parameter -memLimit 100

add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180

add responder action act1 respondwith ""HTTP/1.1 200 OK\r\n\r\n" + http
    .req.url.url_categorize(0,0).reputation + "\n""

```



```
15 add responder policy p1 ”HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.eq(”
   Shopping/Retail”) || HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.eq(”
   Search Engines & Portals
16
17 ”)” act1
18
19 bind cs vserver starcs_PROXY -policyName p1 -priority 10 -
   gotoPriorityExpression END -type REQUEST
20
21 add dns nameServer 10.140.50.5
22
23 set ssl parameter -denySSLReneg NONSECURE -defaultProfile ENABLED -
   sigDigestType RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-
   SHA512 -ssliErrorCache ENABLED
24
25 -ssliMaxErrorCacheMem 100000000
26
27 add ssl policy pol1 -rule ”client.ssl.origin_server_cert.subject.
   URL_CATEGORIZE(0,0).CATEGORY.eq(”Search Engines & Portals”)”” -
   action INTERCEPT
28
29 add ssl policy pol3 -rule ”client.ssl.origin_server_cert.subject.ne(”
   citrix”)” -action INTERCEPT
30
31 add ssl policy swg_pol -rule ”client.ssl.client_hello.SNI.
   URL_CATEGORIZE(0,0).CATEGORY.ne(”Uncategorized”)” -action INTERCEPT
32
33 set urlfiltering parameter -HoursBetweenDBUpdates 3 -
   TimeOfDayToUpdateDB 03:00
34 <!--NeedCopy-->
```

シードデータベースパスとクラウドサーバー名の設定

クラウド検索サーバー名とシードデータベースパスを手動で設定するために、シードデータベースパスとクラウドルックアップサーバー名を構成できるようになりました。そのために、「CloudHost」と「seedDBPath」という 2 つの新しいパラメータが URL フィルタリングパラメータに追加されます。

コマンドプロンプトで入力します。

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>] [-
TimeOfDayToUpdateDB <HH:MM>] [-LocalDatabaseThreads <positive_integer>] [-
CloudHost <string>] [-SeedDBPath <string>]
```

例:


```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
03:00 -CloudHost localhost -SeedDBPath /mypath
```

Citrix ADC アプライアンスと NetSTAR の間の通信には、ドメインネームサーバーが必要な場合があります。アプライアンスからの単純なコンソール接続または Telnet 接続を使用してテストできます。

例:

```
root@ns# telnet nsv10.netstar-inc.com 443
Trying 1.1.1.1...
Connected to nsv10.netstar-inc.com.
Escape character is '^]'.

root@ns# telnet incompasshybridpc.netstar-inc.com 443
Trying 10.10.10.10...
Connected to incompasshybridpc.netstar-inc.com.
Escape character is '^]'.
<!--NeedCopy-->
```

監査ログメッセージを設定する

監査ログを使用すると、URL 分類プロセスのどの段階でも条件や状況を確認できます。Citrix ADC アプライアンスが着信 URL を受信すると、レスポンダーポリシーに URL フィルタリング式がある場合、監査ログ機能は URL に URL セット情報を収集します。この情報は、監査ログで許可されるすべてのターゲットのログメッセージとして格納されます。

- 送信元 IP アドレス (要求を行ったクライアントの IP アドレス)。
- 宛先 IP アドレス（要求されたサーバの IP アドレス)。
- スキーマ、ホスト、およびドメイン名 (http://www.example.com) を含むリクエストされた URL。
- URL フィルタリングフレームワークが返す URL カテゴリ。
- URL フィルタリングフレームワークが返した URL カテゴリグループ。
- URL フィルタリングフレームワークが返した URL レピュテーション番号。
- ポリシーによって実行された監査ログアクション。

URL リスト機能の監査ロギングを設定するには、次のタスクを完了する必要があります。

1. 監査ログを有効化。
2. 「監査ログの作成」メッセージアクション。
3. [監査ログメッセージ] アクションで URL リストレスポンダポリシーを設定します。

詳細については、「監査ログ 」トピックを参照してください。

SYSLOG メッセージングを使用した障害エラーの保存

URL フィルタリングプロセスのどの段階でも、システムレベルの障害が発生した場合、ADC アプライアンスは監査ログメカニズムを使用して ns.log ファイルにログを保存します。エラーは SYSLOG 形式でテキストメッセージとして保存されるため、管理者は後でイベント発生の時系列順に表示できます。これらのログは、アーカイブのために外部 SYSLOG サーバにも送信されます。詳細については、CTX229399 を参照してください。

たとえば、URL フィルタリング SDK を初期化するときにエラーが発生した場合、エラーメッセージは次のメッセージ形式で格納されます。

```
Oct 3 15:43:40 <local0.err> ns URLFiltering[1349]: Error initializing
NetStar SDK (SDK error=-1). (status=1).
```

Citrix ADC アプライアンスは、4 つの異なる障害カテゴリにエラーメッセージを格納します。

- ダウンロードに失敗しました。分類データベースをダウンロードしようとしたときにエラーが発生した場合。
- 統合に失敗しました。更新を既存の分類データベースに統合するときにエラーが発生した場合。
- 初期化に失敗しました。URL 分類機能の初期化時にエラーが発生した場合は、分類パラメータを設定するか、分類サービスを終了してください。
- 検索に失敗しました。アプライアンスがリクエストの分類の詳細を取得するときにエラーが発生した場合。

NetStar イベント用の SNMP トラップの設定

URL フィルタリング機能では、次の条件が発生すると SNMP トラップが生成されます。

- NetStar データベースの更新が失敗するか成功する。
- NetStar SDK の初期化が失敗するか成功する。

アプライアンスには、SNMP アラームと呼ばれる条件付きエンティティのセットがあります。SNMP アラームの条件が満たされると、アプライアンスはトラップを生成し、指定されたトラップ宛先に送信します。たとえば、NetStar SDK の初期化に失敗すると、SNMP OID 1.3.6.1.4.1.5951.1.1.0.183 が生成され、トラップの送信先に送信されます。

アプライアンスがトラップを生成するには、まず SNMP アラームを有効にして設定する必要があります。次に、生成されたトラップメッセージをアプライアンスが送信するトラップ宛先を指定します。

SNMP アラームを有効にする

Citrix ADC アプライアンスは、有効になっている SNMP アラームに対してのみトラップを生成します。一部のアラームはデフォルトで有効になっていますが、無効にすることもできます。

SNMP アラームを有効にすると、成功または失敗のイベントが発生すると、URL フィルタリング機能によってトラップメッセージが生成されます。一部のアラームは、デフォルトで有効になっています。

コマンドラインインターフェイスを使用して SNMP アラームを有効にするには、次の手順を実行します。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。


```
enable snmp alarm <trapName>
show snmp alarm <trapName>
```

Citrix ADC GUI を使用して SNMP アラームを有効にするには

1. [システム] > [**SNMP**] > [アラーム] に移動し、アラームを選択します。
2. [アクション] をクリックし、[有効] を選択します。

CLI を使用して SNMP アラームを設定する

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue
 <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

例**:**

```
set snmp alarm URL-FIL-DB-UPDATE-STATUS -state ENABLED
set snmp alarm URL-FIL-INIT-SDK -state ENABLED
```

GUI を使用して SNMP アラームを設定する

[システム] > [**SNMP**] > [アラーム] に移動し、アラームを選択し、アラームパラメータを設定します。

SNMP トラップの詳細については、SNMP のトピックを参照してください。

URL レピュテーションスコア

October 7, 2021

URL 分類機能は、ポリシーベースの制御を提供し、ブラックリスト URL を制限します。URL カテゴリ、レピュテーションスコア、URL カテゴリとレピュテーションスコアに基づいて Web サイトへのアクセスを制御できます。ネットワーク管理者は、リスクの高い Web サイトにアクセスするユーザーを監視する場合、URL レピュテーションスコアにバインドされたレスポンダーポリシーを使用して、そのようなリスクの高い Web サイトをブロックできます。

着信 URL 要求を受信すると、アプライアンスは URL 分類データベースからカテゴリとレピュテーションスコアを取得します。データベースから返されたレピュテーションスコアに基づいて、アプライアンスはウェブサイトにレピュテーションレーティングを割り当てます。値の範囲は 1 ～4 です。4 は、次の表に示すように、最もリスクのある Web サイトのタイプです。

URL レピュテーション評価	評価コメント
1	クリーンサイト
2	不明なサイト
3	潜在的に危険な、または危険なサイトに所属している

URL レピュテーション評価	評価コメント
4	悪質なサイト

ユースケース**:URL** レピュテーションスコアによるフィルタリング

ユーザーのトランザクションとネットワーク帯域幅の消費を監視するネットワーク管理者を持つ企業組織を考えてみましょう。マルウェアがネットワークに入る可能性がある場合、管理者はデータのセキュリティを強化し、ネットワークにアクセスする悪意のある危険なウェブサイトへのアクセスを制御する必要があります。このような脅威からネットワークを保護するために、管理者は URL レピュテーションスコアによるアクセスを許可または拒否するように URL フィルタリング機能を設定できます。

ネットワーク上のアウトバウンドトラフィックとユーザーアクティビティの監視の詳細については、「Analytics」を参照してください。

組織の従業員がソーシャルネットワーキング Web サイトにアクセスしようとすると、ADC アプライアンスは URL 要求を受け取ります。URL 分類データベースにクエリを実行して、URL カテゴリをソーシャルネットワーキングおよびレピュテーションスコア 3 として取得します。これは、潜在的に危険な Web サイトを示します。次に、アプライアンスは、レピュテーション評価が 3 以上のサイトへのアクセスをブロックするなど、管理者によって構成されたセキュリティポリシーを確認します。次に、ポリシーアクションを適用して、Web サイトへのアクセスを制御します。

この機能を実装するには、SSL 転送プロキシウィザードを使用して URL レピュテーションスコアとセキュリティしきい値レベルを構成する必要があります。

GUI を使用したレピュテーションスコアの設定

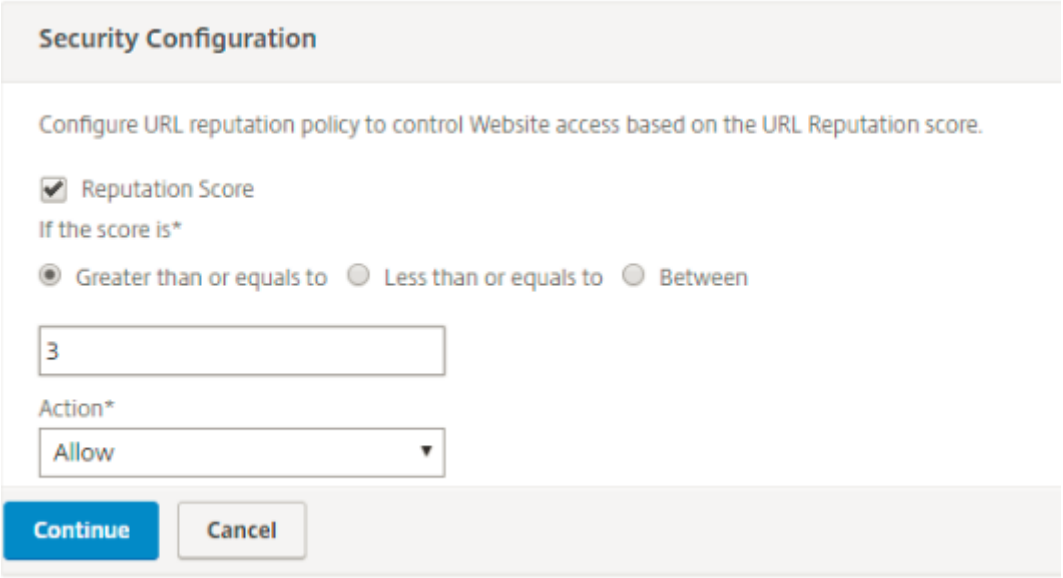
SSL 転送プロキシウィザードを使用して、レピュテーションスコアとセキュリティレベルを構成することをお勧めします。設定されたしきい値に基づいて、トラフィックを許可、ブロック、またはリダイレクトするポリシーアクションを選択できます。

1. [セキュリティ] > [**SSL** 転送プロキシ] に移動します。
2. 詳細ウィンドウで、[**SSL** 転送プロキシウィザード] をクリックします。
3. 詳細ページで、プロキシサーバーの設定を指定します。
4. [**Continue**] をクリックして、SSL インターセプションなどの他の設定を指定し、管理を識別します。
5. [続行] をクリックして、[セキュリティ構成] セクションにアクセスします。
6. [セキュリティの構成] セクションで、[レピュテーションスコア] チェックボックスをオンにして、URL レピュテーションスコアに基づいてアクセスを制御します。
7. セキュリティレベルを選択し、レピュテーションスコアのしきい値を指定します。
   a) より大きい、または等しい：しきい値が N 以上の場合、Web サイトを許可またはブロックします。N の範囲は 1 ～4 です。
   b) 以下-しきい値が N 以下の場合、Web サイトを許可またはブロックします。N の範囲は 1 ～4 です。

c) [In from]：しきい値が N1 ～N2 の間で、1 ～4 の範囲の場合に、Web サイトを許可またはブロックします。

8. ドロップダウンリストから応答者のアクションを選択します。
9. [続行して 閉じる] をクリックします。

次の画像は、SSL 転送プロキシウィザードの [セキュリティ構成] セクションを示しています。[URL レピュテーションスコア] オプションを有効にして、ポリシー設定を構成します。



Analytics

October 7, 2021

Citrix ADC アプライアンスでは、すべてのユーザーレコードとそれ以降のレコードがログに記録されます。Citrix Application Delivery Management（ADM）を Citrix ADC アプライアンスと統合すると、ログに記録されたユーザーアクティビティとアプライアンス内の後続のレコードが、`logstream` 機能を使用して Citrix ADM にエクスポートされます。

Citrix ADM は、訪問した Web サイトや消費された帯域幅など、ユーザーのアクティビティに関する情報を照合して表示します。また、帯域幅の使用量と検出された脅威（マルウェアやフィッシングサイトなど）をレポートします。これらの主要なメトリックを使用して、ネットワークを監視し、Citrix SWG アプライアンスで修正アクションを実行できます。詳細については、「Citrix SSL フォワードプロキシ分析」を参照してください。

Citrix ADC アプライアンスを Citrix ADM と統合するには：

1. Citrix ADC アプライアンスで、SSL フォワードプロキシ機能を構成するときに、分析を有効にし、分析に使用する Citrix ADM インスタンスの詳細を指定します。

2. Citrix ADM で、Citrix ADC アプライアンスをインスタンスとして Citrix ADM に追加します。詳細については、「Citrix ADM へのインスタンスの追加」を参照してください。

ユースケース：リモートマルウェア検査に **ICAP** を使用してエンタープライズネットワークを安全にする

October 7, 2021

Citrix ADC アプライアンスはプロキシとして機能し、すべてのクライアントトラフィックを代行受信します。アプライアンスは、ポリシーを使用してトラフィックを評価し、リソースが存在するオリジンサーバーにクライアント要求を転送します。アプライアンスはオリジンサーバーからの応答を復号化し、プレーンテキストのコンテンツを ICAP サーバーに転送してマルウェア対策チェックを行います。ICAP サーバーは、「適応不要」、エラー、または変更要求を示すメッセージで応答します。ICAP サーバーからの応答に応じて、要求されたコンテンツがクライアントに転送されるか、適切なメッセージが送信されます。

このユースケースでは、Citrix ADC アプライアンスで一般的な構成、プロキシと SSL インターセプションに関連する構成、および ICAP 構成を実行する必要があります。

一般的な構成

次のエンティティを構成します。

- NSIP アドレス
- サブネット IP（SNIP）アドレス
- DNS ネームサーバー
- SSL インターセプションのためにサーバ証明書に署名するための CA 証明書とキーのペア

プロキシサーバと **SSL** インターセプションの設定

次のエンティティを構成します。

- エクスプリシットモードのプロキシサーバで、すべてのアウトバウンド HTTP および HTTPS トラフィックを代行受信します。
- SSL プロファイルを使用して、接続の SSL 設定（暗号やパラメータなど）を定義します。
- SSL ポリシーを使用して、トラフィックを代行受信するためのルールを定義します。すべてのクライアント要求をインターセプトするには、true に設定します。

詳細については、次のトピックを参照してください。

- プロキシモード
- SSL インターセプション

次の構成例では、マルウェア対策検出サービスがwww.example.comにあります。

一般的な設定例:

```
add dns nameServer 203.0.113.2

add ssl certKey ns-swg-ca-certkey -cert ns_swg_ca.crt -key ns_swg_ca.
    key
<!--NeedCopy-->
```

プロキシサーバーと **SSL** インターセプション設定の例:

```
add cs vserver explicitswg PROXY 192.0.2.100 80 -Authn401 ENABLED -
    authnVsName explicit-auth-vs

set ssl parameter -defaultProfile ENABLED

add ssl profile swg_profile -sslInterception ENABLED

bind ssl profile swg_profile -ssliCACertkey ns-swg-ca-certkey

set ssl vserver explicitswg -sslProfile swg_profile

add ssl policy ssli-pol_ssli -rule true -action INTERCEPT

bind ssl vserver explicitswg -policyName ssli-pol_ssli -priority 100 -
    type INTERCEPT_REQ
<!--NeedCopy-->
```

ICAP 設定の例:

```
add service icap_svc 203.0.113.225 TCP 1344

enable ns feature contentinspection

add icapprofile icapprofile1 -uri /example.com -Mode RESMOD

add contentInspection action CiRemoteAction -type ICAP -serverName
    icap_svc -icapProfileName icapprofile1

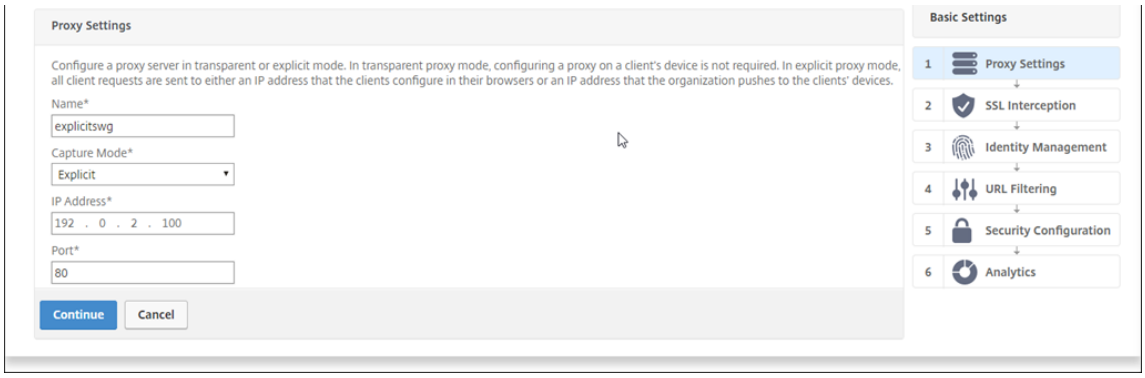
add contentInspection policy CiPolicy -rule "HTTP.REQ.METHOD.NE("
    CONNECT")" -action CiRemoteAction
```



```
10
11 bind cs vserver explicitswg -policyName  CiPolicy -priority 200 -type
       response
12 <!--NeedCopy-->
```

プロキシ設定を構成する

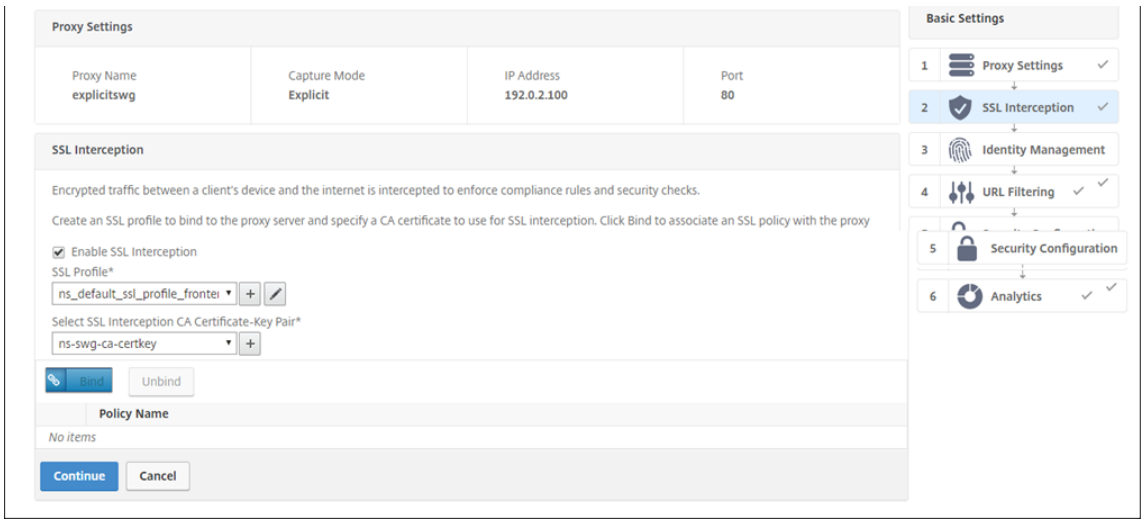
1. [セキュリティ] > [**SSL** 転送プロキシ] > [**SSL** 転送プロキシウィザード] に移動します。
2. [は じめに] をクリックし、[続行] をクリックします。
3. [プロキシ設定] ダイアログボックスで、明示的なプロキシサーバーの名前を入力します。
4. [キャプチャモード] で、[明示的] を選択します。
5. IP アドレスとポート番号を入力します。



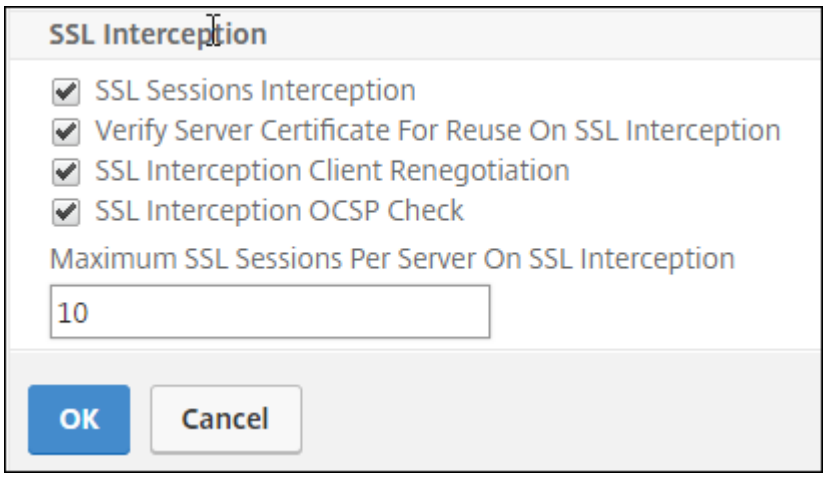
6. [続行] をクリックします。

SSL インターセプション設定の構成

1. 「**SSL** インターセプションを有効にする」を選択します。

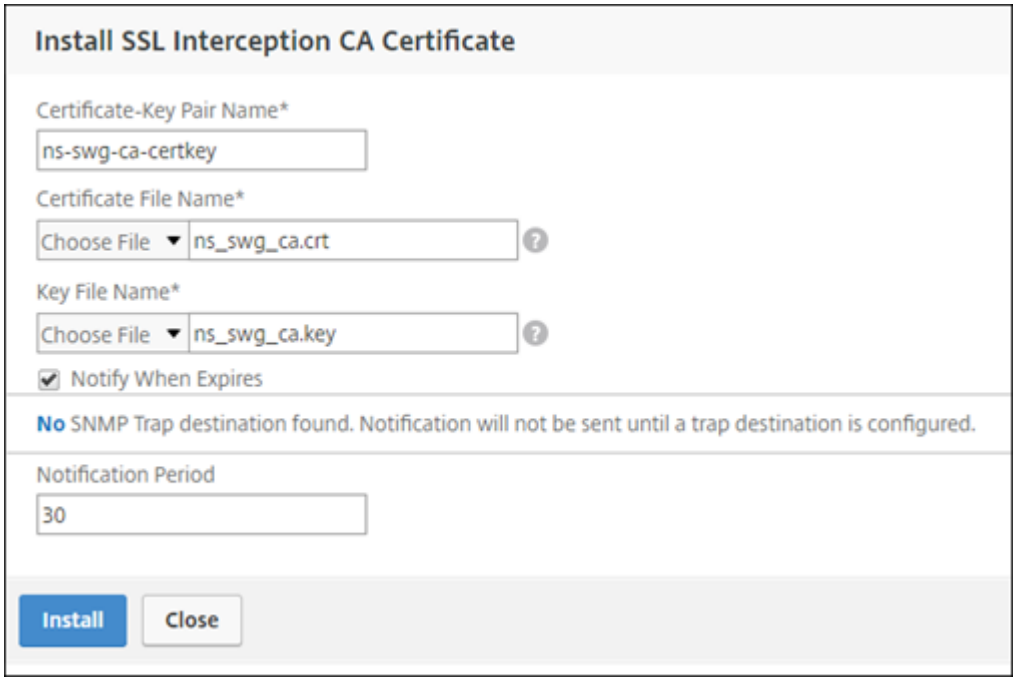


2. [**SSL** プロファイル] で、既存のプロファイルを選択するか、[+] をクリックして新しいフロントエンド SSL プロファイルを追加します。このプロファイルで **SSL** セッションインターセプトを有効にします。既存のプロファイルを選択する場合は、次の手順をスキップします。



3. [**OK]** をクリックし、[完了] をクリックします。

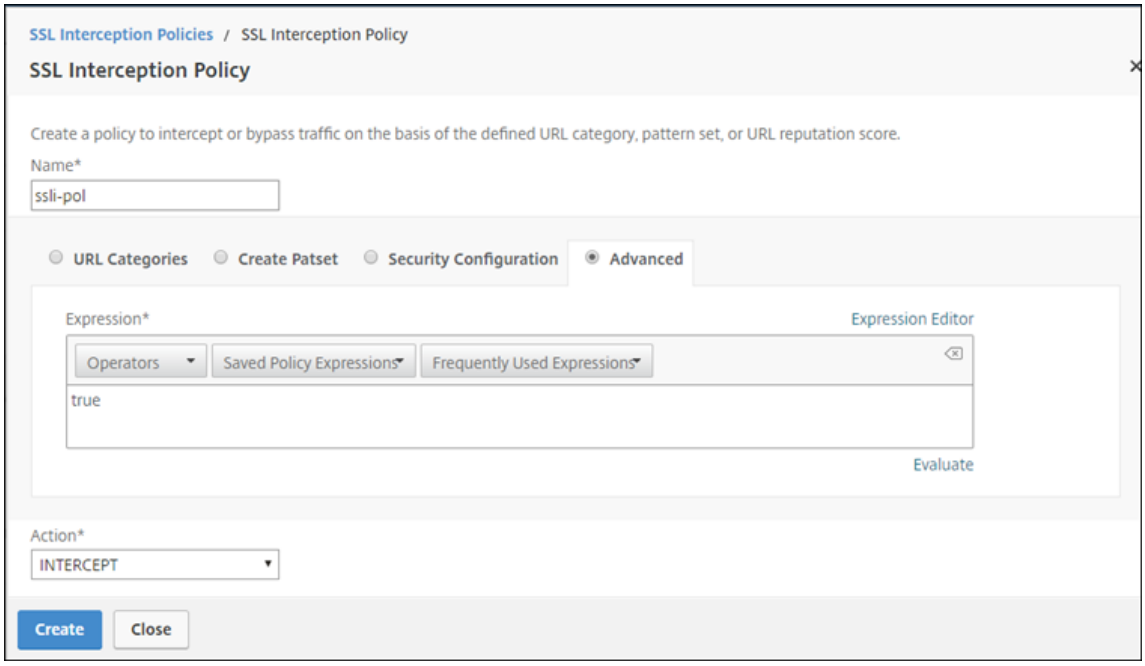
4. 「**SSL** インターセプション **CA** 証明書とキーペアの選択」で、既存の証明書を選択するか、「+」をクリックして SSL インターセプション用の CA 証明書とキーペアをインストールします。既存の証明書を選択した場合は、次の手順をスキップします。



5. [インストール] をクリックし、[閉じる] をクリックします。

6. すべてのトラフィックを代行受信するポリシーを追加します。［バインド］をクリックします。[**Add**] をクリックして新しいポリシーを追加するか、既存のポリシーを選択します。既存のポリシーを選択した場合は、[**Insert**] をクリックし、次の 3 つの手順をスキップします。

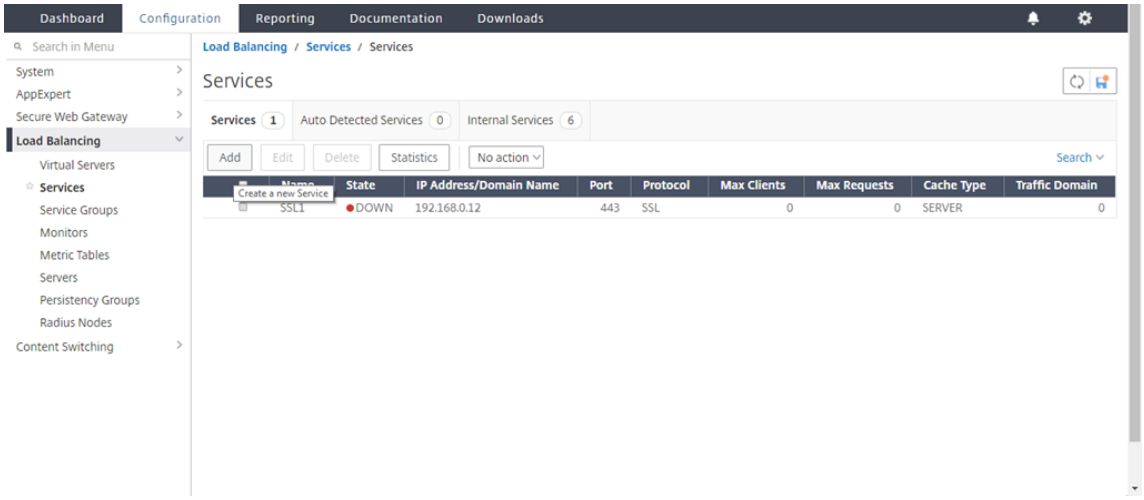
7. ポリシーの名前を入力し、[詳細設定] を選択します。式エディタで true と入力します。
8. [アクション] で、[インターセプト] を選択します。



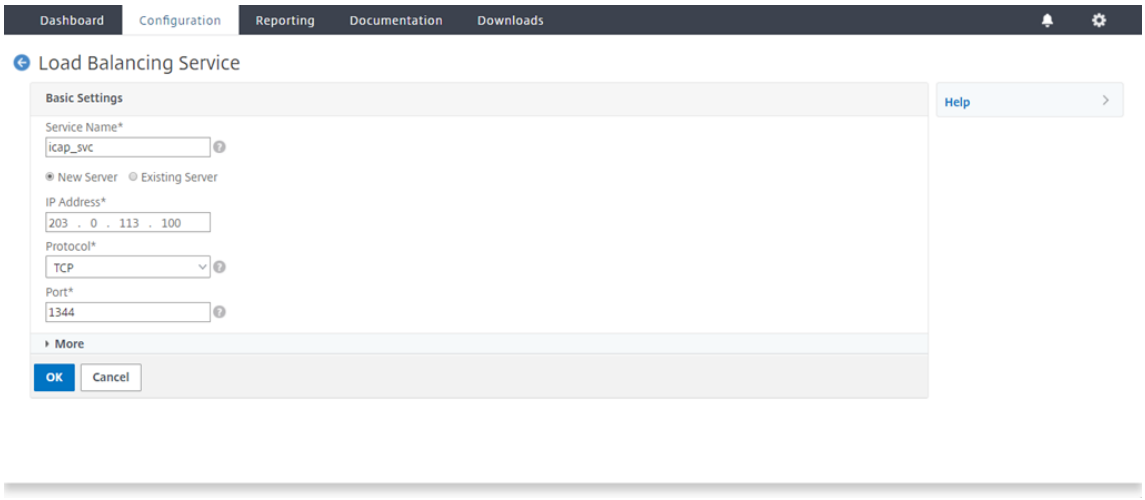
9. [作成] をクリックします。
10. [続行] を 4 回クリックし、[完了] をクリックします。

ICAP の設定を構成する

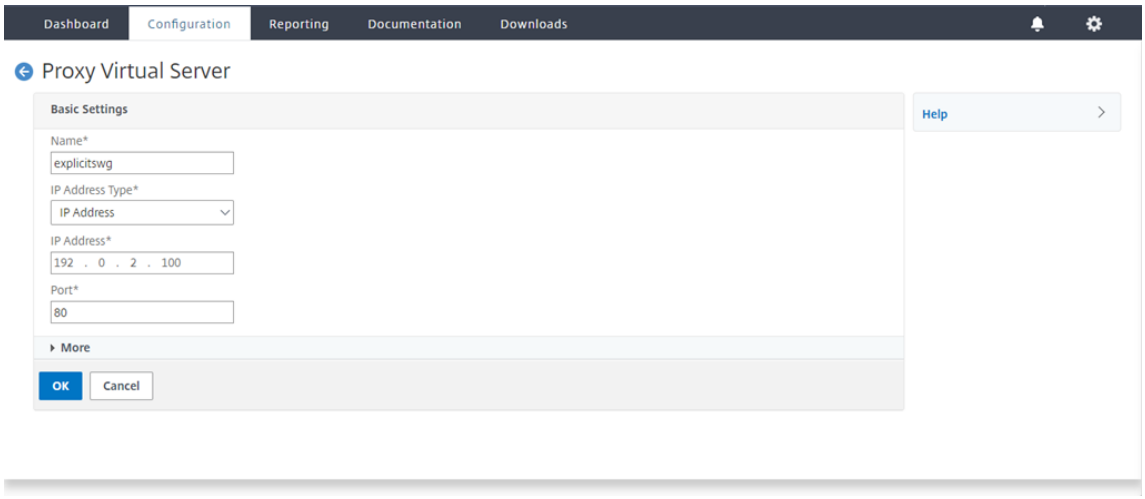
1. [負荷分散] > [サービス] に移動し、[追加] をクリックします。



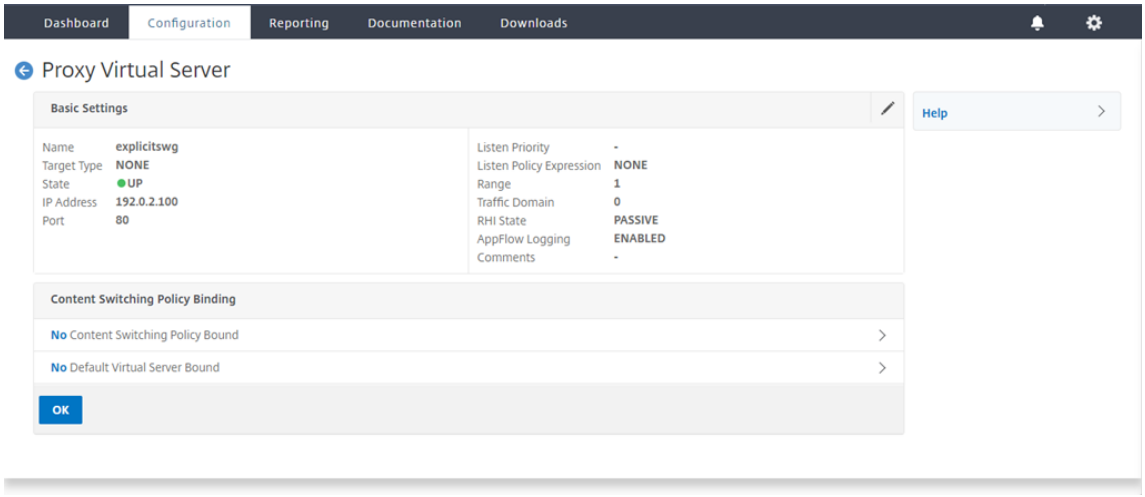
2. 名前と IP アドレスを入力します。「プロトコル」で、「**TCP**」を選択します。[ポート] に **1344** と入力します。[**OK**] をクリックします。



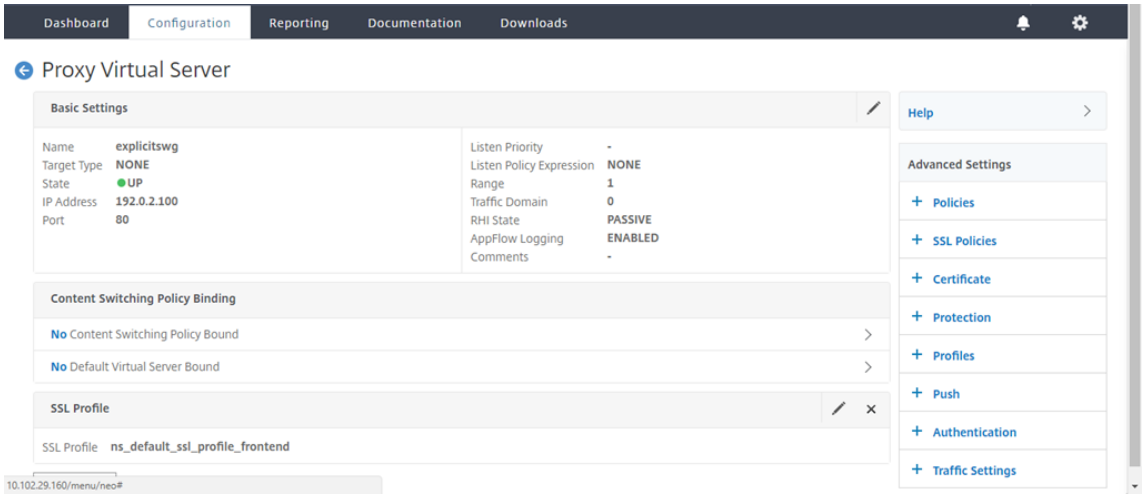
3. [**SSL** フォワードプロキシ] > [プロキシ仮想サーバー] に移動します。プロキシ仮想サーバーを追加するか、仮想サーバーを選択して「編集」をクリックします。詳細を入力したら、[**OK**] をクリックします。



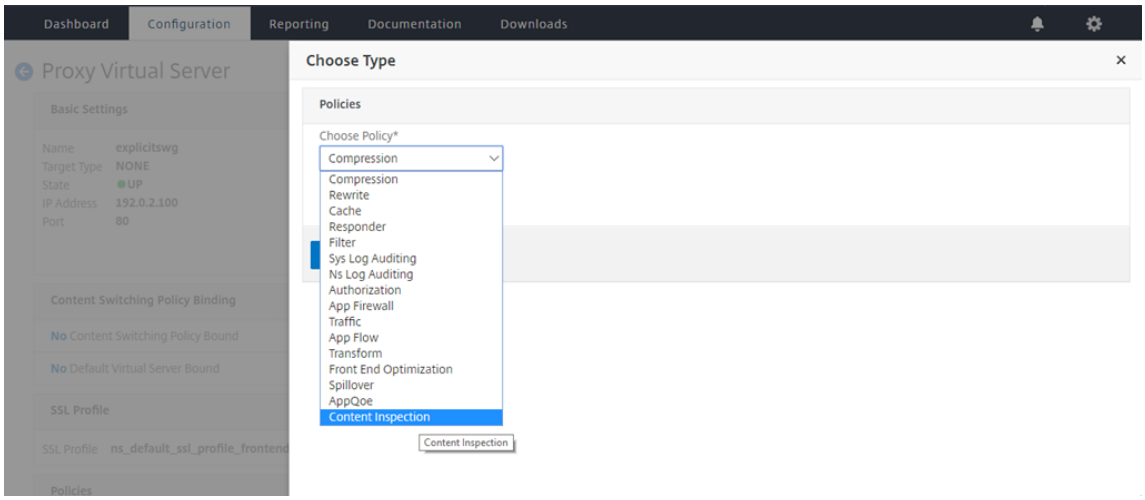
もう一度 [**OK**] をクリックします。



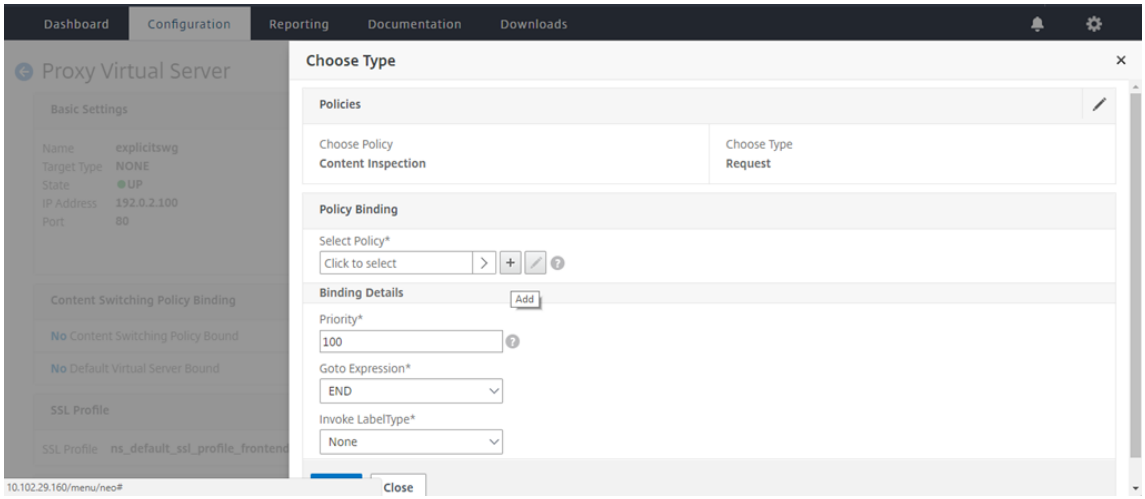
4. [詳細設定] で、[ポリシー] をクリックします。



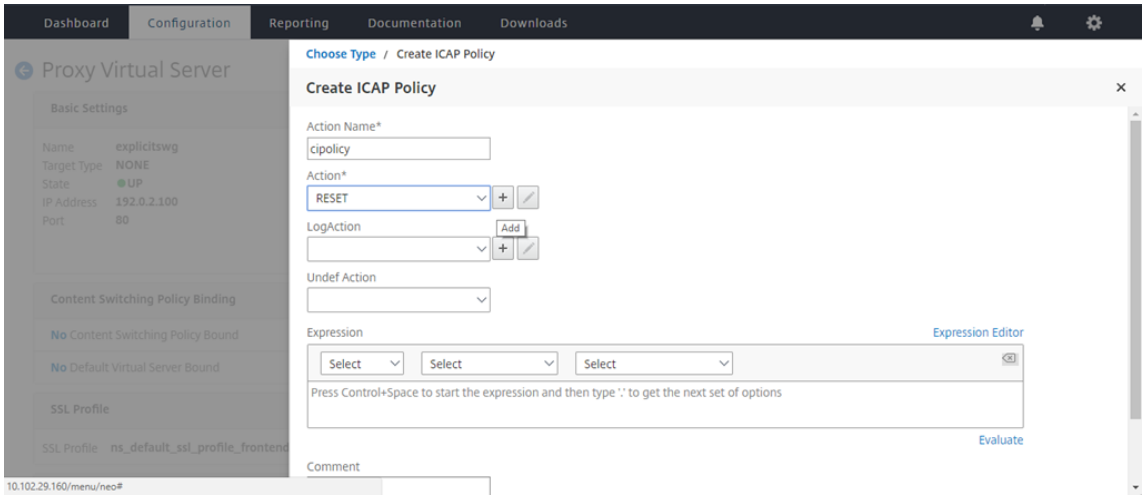
5.「ポリシーの選択」で、「コンテンツ検査」を選択します。[続行] をクリックします。



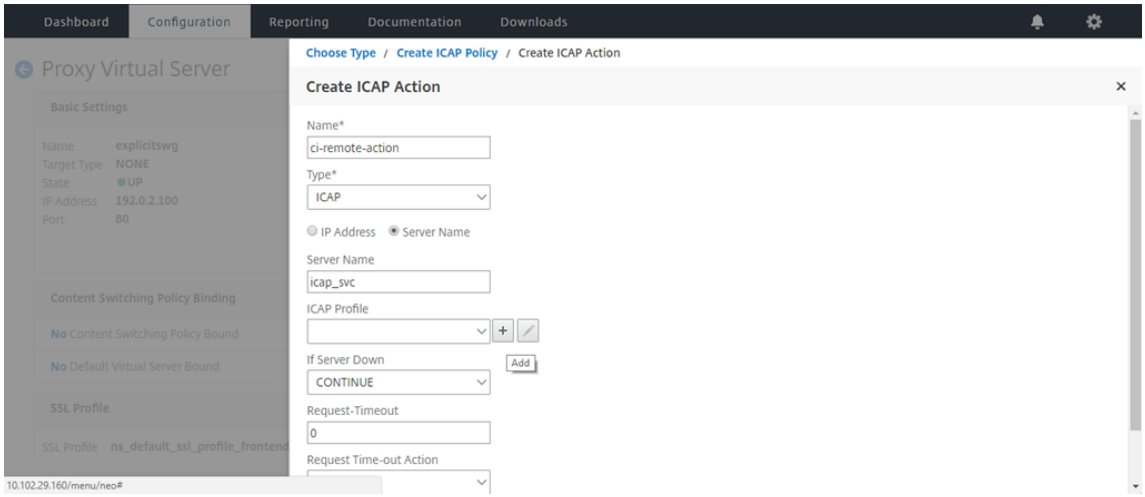
6. **[**ポリシーの選択] で、[+] 記号をクリックしてポリシーを追加します。



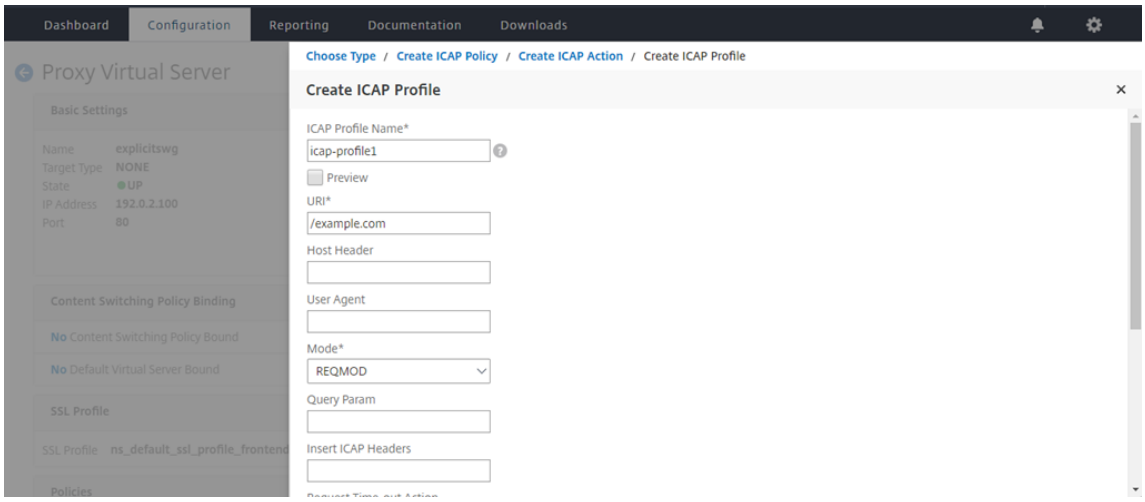
7. ポリシーの名前を入力します。[アクション] で、[+] 記号をクリックしてアクションを追加します。



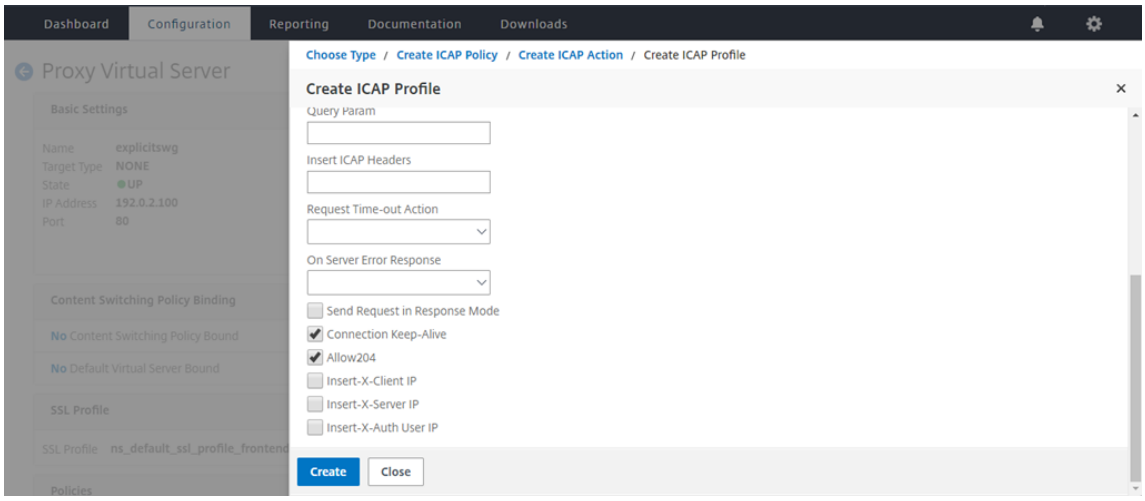
8. アクションの名前を入力します。[サーバー名] に、以前に作成した TCP サービスの名前を入力します。**ICAP** プロファイルで、「+」記号をクリックして ICAP プロファイルを追加します。



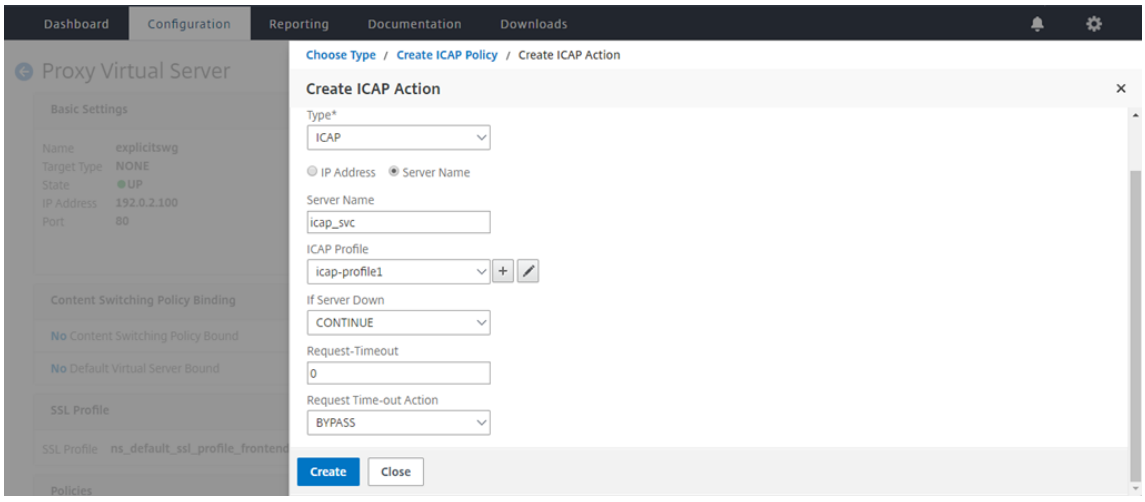
9. プロファイル名「URI」を入力します。「モード」で「**REQMOD**」を選択します。



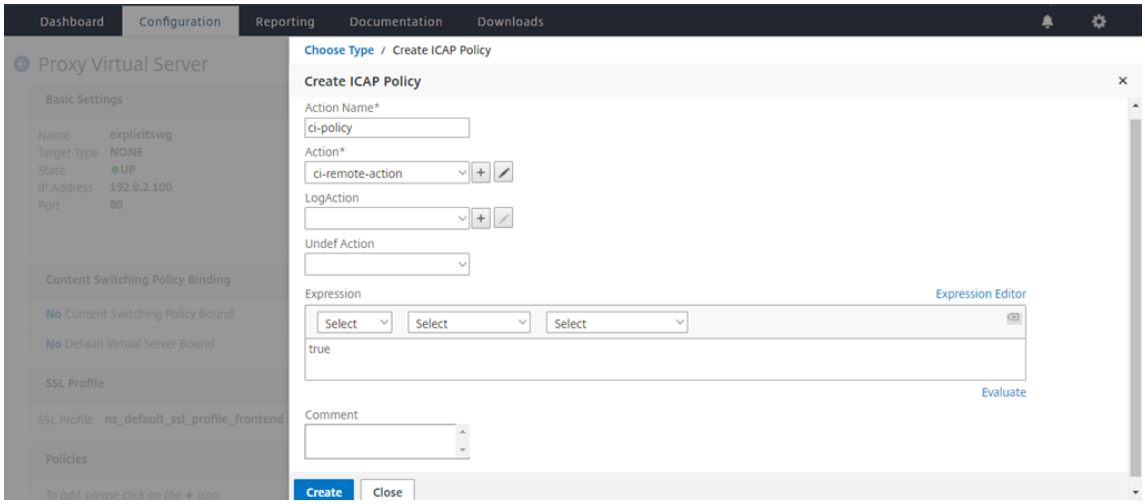
10. ［作成］をクリックします。



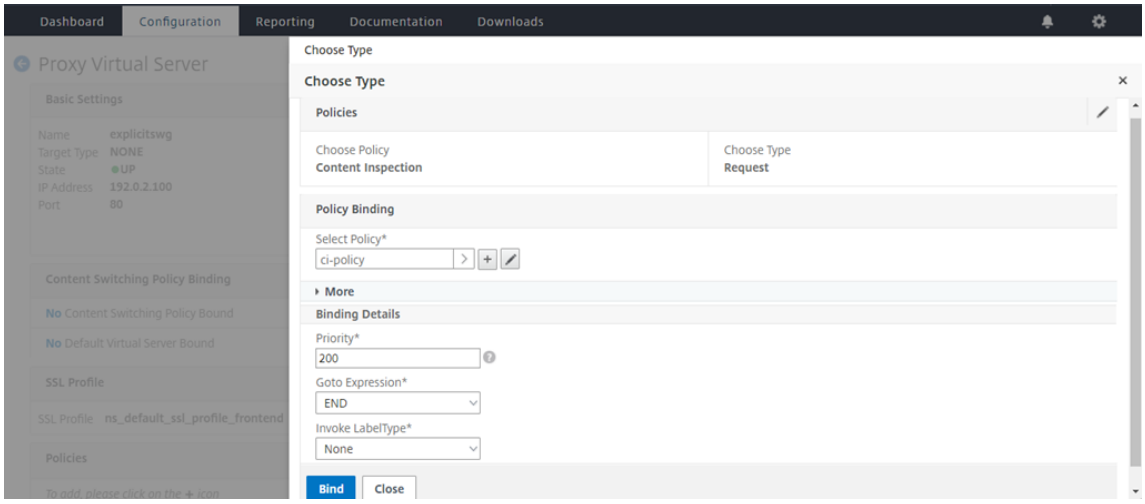
11. 「**ICAP** アクションの作成」ページで、「作成」をクリックします。



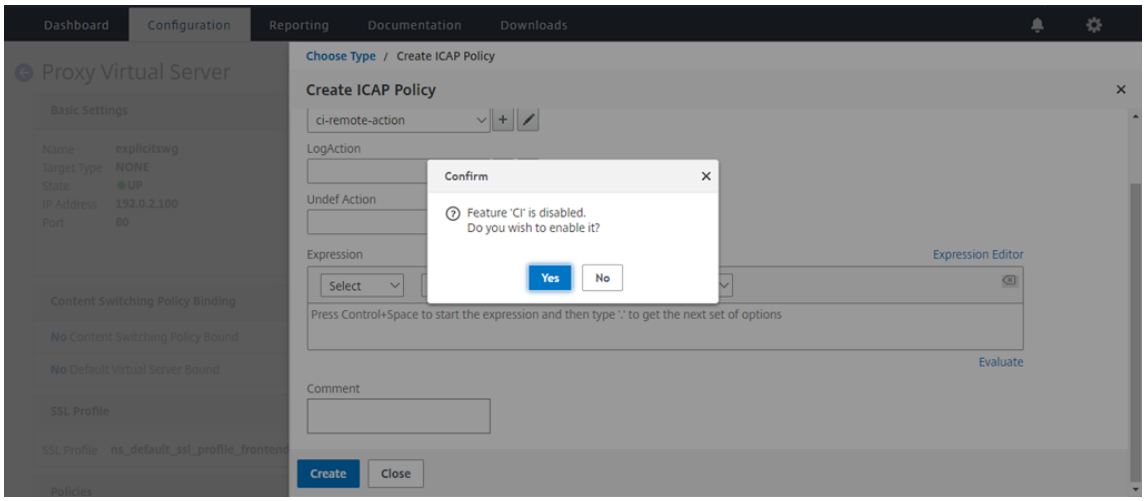
12. **ICAP** ポリシーの作成ページで、式エディターに true と入力します。次に、[作成] をクリックします。



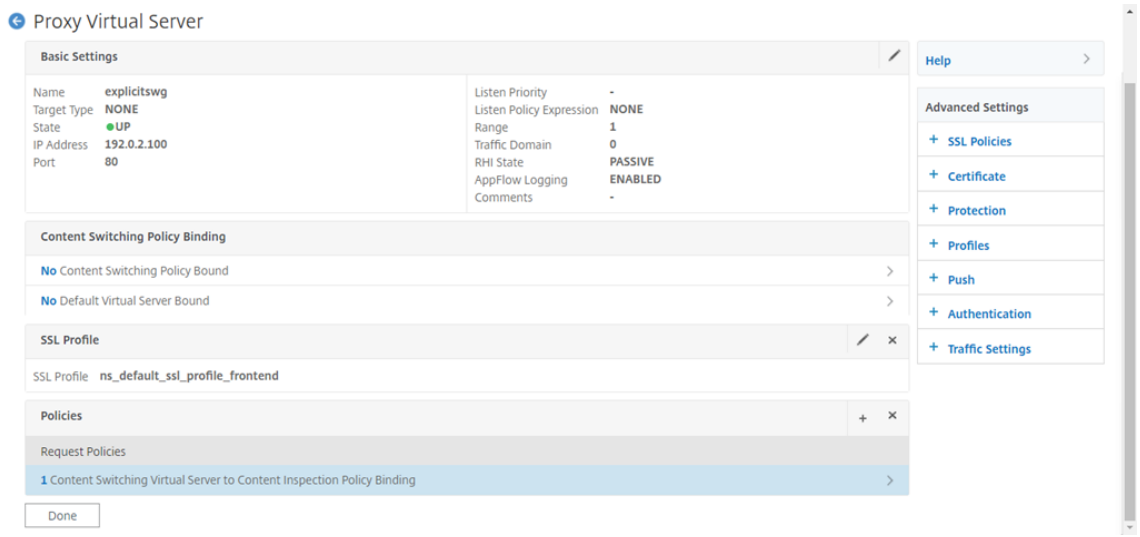
13. ［バインド］をクリックします。



14. コンテンツ検査機能を有効にするように求められたら、[はい] を選択します。



15. ［完了］をクリックします。



Citrix ADC アプライアンスと **RESPMOD** の **ICAP** サーバーとの間のサンプル **ICAP** トランザクション

Citrix ADC アプライアンスから **ICAP** サーバーへの要求:

```
RESPMOD icap://10.106.137.15:1344/resp ICAP/1.0

Host: 10.106.137.15

Connection: Keep-Alive

Encapsulated: res-hdr=0, res-body=282

HTTP/1.1 200 OK

Date: Fri, 01 Dec 2017 11:55:18 GMT

Server: Apache/2.2.21 (Fedora)

Last-Modified: Fri, 01 Dec 2017 11:16:16 GMT

ETag: ”20169-45-55f457f42aee4”

Accept-Ranges: bytes

Content-Length: 69

Keep-Alive: timeout=15, max=100
```



```
24
25 Content-Type: text/plain; charset=UTF-8
26
27 X5O!P%@AP[4PZX54(P^)7CC)7 }
28  $EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
29 <!--NeedCopy-->
```

ICAP サーバーから **Citrix ADC** アプライアンスへの応答:

```
1  ICAP/1.0 200 OK
2
3  Connection: keep-alive
4
5  Date: Fri, 01 Dec, 2017 11:40:42 GMT
6
7  Encapsulated: res-hdr=0, res-body=224
8
9  Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$
10
11 ISTag: ”9.8-13.815.00-3.100.1027-1.0”
12
13 X-Virus-ID: Eicar_test_file
14
15 X-Infection-Found: Type=0; Resolution=2; Threat=Eicar_test_file;
16
17 HTTP/1.1 403 Forbidden
18
19 Date: Fri, 01 Dec, 2017 11:40:42 GMT
20
21 Cache-Control: no-cache
22
23 Content-Type: text/html; charset=UTF-8
24
25 Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$
26
27 Content-Length: 5688
28
29 <html><head><META HTTP-EQUIV=”Content-Type” CONTENT=”text/html; charset
      =UTF-8”/>
30
31 …
32
33 …
```



```
34
35 </body></html>
36 <!--NeedCopy-->
```

ハウツー記事

October 7, 2021

次に、SSL フォワードプロキシの展開の管理に役立つ「方法」の記事に記載されている構成手順または機能的な使用例をいくつか示します。

URL フィルタリング

URL 分類ポリシーの作成方法

URL リストポリシーの作成方法

例外的な URL を許可する方法

アダルトカテゴリのウェブサイトをブロックする方法

セキュリティ

October 7, 2021

以下のトピックでは、Citrix ADC セキュリティ機能の構成とインストールについて説明します。これらの機能のほとんどはポリシーベースです。

コンテンツフィルタリング	不適切な HTML 要求をブロックし、要求が Web サーバーに届かないようにします。
サージ保護	接続試行の急激な上昇を検出し、接続がサーバに進む速度を調整して、サーバの過負荷を防ぎます。
DNS セキュリティオプション	DNS 攻撃から保護するためのポリシーを作成するための簡略化された UI ウィザード。

コンテンツフィルタリング

October 7, 2021

コンテンツフィルタリングは、Citrix Web App Firewall と同じタスクの一部を実行でき、CPU 負荷が低いツールです。ただし、HTTP 要求または応答のヘッダー部分を検査し、一致する接続に対していくつかの簡単なアクションを実行することに限られています。スクリプトを広範囲に使用し、バックエンドデータベースにアクセスする複雑な Web サイトがある場合は、アプリケーションファイアウォールがその Web サイトを保護するための優れたツールである可能性があります。Citrix Web アプリケーションファイアウォールの詳細については、「アプリケーションファイアウォール」を参照してください。

コンテンツフィルタリングは、HTTP リクエストまたは HTTP レスポンスのいずれかに適用できる正規表現に基づいています。たとえば、特定のサイトからのリクエストをブロックするには、各リクエストの URL と式で指定された URL を比較する式を使用できます。式はポリシーの一部であり、式に一致する要求または応答に対して実行されるアクションも指定します。たとえば、アクションがリクエストをドロップしたり、接続をリセットしたりする場合があります。

次に、コンテンツフィルタポリシーでできる操作の例をいくつか示します。

- 許可された場所から接続していない限り、ユーザーがウェブサイトの特定の部分にアクセスできないようにします。
- 不適切な HTTP ヘッダーが Web サーバーに送信されるのを防ぎ、セキュリティを侵害する可能性があります。
- 指定された要求を別のサーバーまたはサービスにリダイレクトします。

コンテンツフィルタを構成するには、機能が有効になっていることを確認したら、選択した接続で実行するフィルタ処理アクションを構成します (定義済みのアクションが目的に適している場合を除く)。次に、選択した接続にアクションを適用するポリシーを構成できます。ポリシーでは、定義済みの式を使用することも、独自の式を作成することもできます。設定したポリシーをアクティブにするには、ポリシーをグローバルにバインドするか、特定の仮想サーバにバインドします。

コンテンツフィルタの有効化

October 7, 2021

デフォルトでは、コンテンツフィルタリングは、Citrix ADC オペレーティングシステム 8.0 以降を実行している Citrix ADC アプライアンスで有効になっています。8.0 より前のオペレーティングシステムバージョンから既存のアプライアンスをアップグレードする場合は、コンテンツフィルタリングを使用する前にライセンスを更新する必要があります。また、コンテンツフィルタリング機能自体を手動で有効にする必要がある場合があります。

CLI を使用したコンテンツのフィルタリングの有効化

コマンドプロンプトで次のコマンドを入力して、コンテンツフィルタを有効にし、構成を確認します。


```
-  enable ns feature ContentFiltering
-  show ns feature
<!--NeedCopy-->
```

例:

```
> enable ns feature ContentFiltering
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   OFF
 .
 .
 .
 .
 11)    Http DoS Protection             HDOSP                OFF
 12)    Content Filtering               CF                   ON
 .
 .
 23)    HTML Injection                  HTMLInjection        ON
 24)    Citrix ADC Push                  push                 OFF
 Done
<!--NeedCopy-->
```

GUI を使用したフィルタリングによるコンテンツの有効化

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] を選択します。
2. 詳細ウィンドウで、[基本機能の構成] をクリックします。
3. [基本機能の構成] ウィンドウで、[コンテンツフィルタ] チェックボックスをオンにし、[OK] をクリックします。

コンテンツフィルタ処理の構成

October 7, 2021

コンテンツフィルタリング機能を有効にしたら、Citrix ADC アプライアンスに受信する接続の処理方法を指示するアクションを 1 つ以上作成します。

コンテンツフィルタリングは、HTTP リクエストに対して次のアクションをサポートします。

- **[追加]:** Web サーバーに要求を送信する前に、指定された HTTP ヘッダーを追加します。
- リセット**:** 接続を終了し、適切な終了通知をユーザーのブラウザに送信します。
- **[転送]:** 要求を指定されたサービスにリダイレクトします。
- **Drop**: ユーザーのブラウザに応答を送信せずに、要求を静かに削除します。
- [破損]**:** 指定された HTTP ヘッダーを変更して、実行しようとしている機能を実行できないようにし、要求をサーバーに送信します。

コンテンツフィルタリングは、HTTP 応答に対して次のアクションをサポートします。

- **Add**: ユーザーのブラウザにレスポンスを送信する前に、指定された HTTP ヘッダーを追加します。
- **ErrorCode**: 指定された HTTP エラーコードをユーザーのブラウザに返します。
- [破損]**:** 指定された HTTP ヘッダーを変更して、そのヘッダーを意図した機能を実行できないようにし、応答をユーザーのブラウザに送信します。

CLI を使用したコンテンツフィルタ処理の設定

コマンドプロンプトで次のコマンドを入力して、コンテンツフィルタ処理アクションを構成し、構成を確認します。

```
1 -  add filter action <name> <qualifier> [<serviceName>] [<value>] [<
     respCode>] [<page>]
2 -  show filter action <name>
3 <!--NeedCopy-->
```

例**:**

```
1 > add filter action act_drop Drop
2  Done
3 > show filter action act_drop
4 1)      Name: act_drop   Filter Type: drop
5  Done
6 <!--NeedCopy-->
```


GUI を使用したコンテンツフィルタアクションの構成

1. [セキュリティ] **>** [保護機能] **>** [フィルタ] に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
   - 新しいアクションを作成するには、[追加] をクリックします。
   - 既存のアクションを変更するには、アクションを選択して、[開く] をクリックします。
3. [フィルタアクションを追加] または [フィルタアクションを設定] ダイアログボックスで、パラメータの値を指定します。
   - アクション名 *—name
   - qualifier*-qualifier（次のパラメータのうちどのパラメータを設定できるかを決定します）
   - サービス名：サービス名
   - ヘッダー名: 値-値
   - レスポンスコード-`respcode`
   - 応答ページ：ページ
4. その他の必要な情報を入力します。たとえば、HTTP エラーコードを送信するようにアクションを設定する場合は、ドロップダウンリストから適切なエラーコードを選択する必要があります。必要に応じて、ドロップダウンリストの下に表示されるエラーメッセージのテキストを変更できます。
5. [作成] または [OK] をクリックし、[閉じる] をクリックします。[アクション] リストには、設定したアクションが表示され、ステータスバーにアクションが作成されたことを示すメッセージが表示されます。

コンテンツフィルタポリシーを構成する

October 7, 2021

コンテンツフィルタリングを実装するには、少なくとも 1 つのポリシーを構成して、フィルタリングする接続を区別する方法を Citrix ADC アプライアンスに指示する必要があります。ポリシーを設定するときに、そのポリシーをアクションに関連付けるため、少なくとも 1 つのフィルタリングアクションを最初に設定しておく必要があります。

コンテンツフィルタポリシーでは、次の 1 つ以上の要素の組み合わせを調べて、フィルタリングする要求または応答を選択します。

- **URL:** HTTP リクエスト内の URL。
- **URL** クエリ**:** URL のクエリ部分のみ。クエリの後の部分 (?) 記号で指定します。
- **URL** トークン**:** URL 内のトークンのみ (ある場合)。アンパサンド (&) で始まり、トークン名、等号 (=)、トークン値が続く部分です。
- **HTTP** メソッド**:** リクエストで使用される HTTP メソッド。通常は GET または POST ですが、定義された 8 つの HTTP メソッドのいずれかを指定できます。
- **HTTP** バージョン**:** リクエスト内の HTTP バージョン。通常は HTTP 1.1 です。
- 標準 **HTTP** ヘッダー： HTTP 1.1 仕様で定義された標準 HTTP ヘッダーのいずれか。

- 標準 **HTTP** ヘッダー値**:** HTTP ヘッダーの値部分。コロンとスペース (:) の後の部分です。
- カスタム **HTTP** ヘッダー**:** ウェブサイトによって発行された、またはユーザーリクエストに表示される非標準 HTTP ヘッダー。
- カスタムヘッダー値：カスタム HTTP ヘッダーの値部分。（標準 HTTP ヘッダーと同様に）コロンとスペース（:）の後の部分です。
- クライアント送信元 **IP**：クライアント要求の送信元 IP。

コンテンツフィルタポリシーでは、従来の式と呼ばれる 2 つの Citrix ADC 式言語のうち、より単純なものが使用されます。従来の式、それらの動作方法、および手動設定の詳細については、「ポリシーと式」を参照してください。

注：Citrix ADC コマンドラインでポリシーを構成した経験がないユーザーは、通常、構成ユーティリティーの使用がかなり簡単になります。

CLI を使用したコンテンツフィルタポリシーの設定

コマンドプロンプトで次のコマンドを入力して、コンテンツフィルタポリシーを構成し、構成を確認します。

```
1 -  add filter policy <name> -rule <expression> (-reqAction <action> | -
     resAction <string>
2 -  show filter policy <name>
3 <!--NeedCopy-->
```

例**:**

```
1 > add filter policy cf-pol -rule ”REQ.HTTP.URL CONTAINS http://abc.com”
      -reqaction DROP
2  Done
3 > show filter policy cf-pol
4 1)      Name: cf-pol    Rule: REQ.HTTP.URL CONTAINS http://abc.com
5         Request action: DROP
6         Response action:
7         Hits: 0
8  Done
9 <!--NeedCopy-->
```

GUI を使用したコンテンツフィルタポリシーの構成

1. [セキュリティ] > [保護機能] > [フィルタ] に移動します。
2. [保護機能] > [フィルタ] に移動します。

3. 詳細ペインで、新しいポリシーを作成するには、[追加] をクリックします。
4. 新しいポリシーを作成する場合は、[フィルタポリシーの作成] ダイアログボックスの [フィルタ名] テキストボックスに、新しいポリシーの名前を入力します。
5.「アクションのリクエスト」または「レスポンスアクション」を選択して、そのアイテムの右側にあるドロップダウンリストを有効にします。
6. ドロップダウンリストの右側にある下向き矢印をクリックし、リクエストまたはレスポンスに対して実行するアクションを選択します。デフォルトの選択肢は、[リセット] と [ドロップ] です。作成したその他のアクションも、このリストに表示されます。
   注： [新規] をクリックして新しいコンテンツフィルタリングアクションを作成するか、[変更] をクリックして既存のコンテンツフィルタリングアクションを変更することもできます。変更できるのは、作成したアクションだけです。デフォルトのアクションは読み取り専用です。
7. 定義済みの式（または名前付き式）を使用してポリシーを定義する場合は、[Named Expressions] リストからいずれかを選択します。
   a) 最初の [名前付き式] ドロップダウンリストの右側にある下向き矢印をクリックし、使用する名前付き式を含む名前付き式のカテゴリを選択します。
   b) 2 番目の [名前付き式] ドロップダウンリストの右側にある下向き矢印をクリックし、名前付きの式を選択します。名前付きエクスプレッションを選択すると、その名前付きエクスプレッションの正規表現定義が [プレビューエクスプレッション] ペインの [名前付きエクスプレッション] リストボックスの下に表示されます。
   c) [式を追加] をクリックして、その名前付き式を [式] リストに追加します。
      注: この手順またはステップ 7 のいずれかを実行する必要がありますが、両方を実行する必要はありません。
8. ポリシーを定義する新しい式を作成する場合は、式エディタを使用します。
   a)［追加］をクリックします。[式の追加] ダイアログボックスが表示されます。
   b) [式の追加] ダイアログボックスで、フィルタする接続の種類を選択します。[Flow Type] はデフォルトで REQ に設定されており、Citrix ADC アプライアンスは着信接続または要求を参照するように指示します。発信接続（応答）をフィルタリングする場合は、ドロップダウンリストの横にある右矢印をクリックして、RES を選択します。
   c) プロトコルが HTTP に設定されていない場合は、[Protocol] ドロップダウンリストの右にある下矢印をクリックして、[HTTP] を選択します。
      注：Citrix ADC 従来の式言語では、「HTTP」には HTTPS リクエストも含まれます。
   d) [修飾子] ドロップダウンリストの右側にある下向き矢印をクリックし、式の修飾子を選択します。選択肢は次のとおりです：
      - 方法**:** リクエストで使用される HTTP メソッド。
      - **URL: URL** ヘッダーの内容。
      - **URLTOKENS:** HTTP ヘッダー内の URL トークン。
      - バージョン**:** 接続の HTTP バージョン。
      - **HEADER**： HTTP リクエストのヘッダー部分。
      - **URLLEN:** URL ヘッダーの内容の長さ。

- **URLQUERY:** URL ヘッダーの内容のクエリ部分。
- **URLQUERYLEN:** URL ヘッダーのクエリ部分の長さ。

残りのリスト・ボックスの内容が、選択した修飾子に適した選択肢に変わります。たとえば、「HEADER」を選択すると、「フロータイプ」リストボックスの下に「ヘッダー名 *」というテキストフィールドが表示されます。

e) [演算子] ドロップダウンリストの右側にある下向き矢印をクリックし、式の演算子を選択します。選択内容は、前の手順で選択したプロトコルによって異なります。次のリストには、すべての演算子が含まれています。

- **==:** 次のテキスト文字列と完全に一致します。
- ！ **=:** 次の文字列と正確に一致しません。
- **>:** 次の整数よりも大きい。
- **CONTAINS:** 次のテキスト文字列が含まれます。
- **CONTENTS:** 指定されたヘッダー、URL、または URL クエリの内容。
- **EXISTS**: 指定されたヘッダーまたはクエリが存在します。
- **NOTCONTAINS:** 次のテキスト文字列は含まれません。
- **NOTEXISTS:** 指定されたヘッダーまたはクエリは存在しません。

f)「値」(Value) テキストボックスが表示されている場合は、適切な文字列または数値を入力します。文字列を何らかの方法でテストする場合は、「値」テキストボックスに文字列を入力します。整数をテストする場合は、「値」(Value) テキストボックスに整数を入力します。

g) [プロトコル] として [HEADER] を選択した場合は、[ヘッダー名 *] テキストボックスに目的のヘッダーを入力します。

h) [OK] をクリックして、式を [式] リストに追加します。

i) 手順 B ～H を繰り返して、プロファイルに必要な式を追加します。

j) [閉じる] をクリックして [式エディタ] を閉じます。

9. 新しいエクスプレッションを作成した場合は、エクスプレッションフレームで、[任意のエクスプレッションと一致] ドロップダウンリストからオプションを選択します。選択肢は次のとおりです:

- 任意の式に一致します。リクエストが [Expressions] リスト内のいずれかの式と一致する場合、リクエストはこのポリシーと一致します。
- [すべての式に一致する] リクエストが [式] リストのすべての式に一致する場合、リクエストはこのポリシーに一致します。すべてと一致しない場合、このポリシーには一致しません。
- [表式] [式] リストを 3 つの列で構成された表形式に切り替えます。最初の列では、BEGIN [(] 演算子を配置することができます。2 番目の列には、選択または作成した式が表示されます。3 番目の列では、次のリストに他の演算子を配置して、複雑なポリシーグループを作成できます。このポリシーグループでは、各グループを match any 式または match all 式に設定できます。
- AND [&&] 演算子は、リクエストが現在の式と次の式の両方に一致することを要求するようにアプライアンスに指示します。

OR [\	\	] 演算子は、要求が現在の式または次の式、またはその両方に一致することを要求するようにアプライアンスに指示します。リクエストがどちらの式にも一致しない場合にのみ、ポリシーと一致しません。

-
- END [）] オペレータは、これがこのエクスプレッショングループまたはポリシーの最後の式であることをアプライアンスに伝えます。
  注意: 表形式では、式ごとに「任意の式に一致」と「すべての式に一致」の両方を含む複雑なポリシーを作成できます。あなたはどちらか一方に限定されるものではありません。
- [高度なフリーフォーム] エクスプレッションエディタを完全にオフにし、エクスプレッションリストをテキスト領域に変更します。テキスト領域には、任意の PCRE-形式の正規表現を入力して、このポリシーを定義できます。これは、ポリシーを作成するための最も強力で最も難しい方法であり、Citrix ADC アプライアンスと PCRE 形式の正規表現に精通している場合にのみお勧めします。
  注意：高度なフリーフォーム式編集モードに切り替えると、他のモードに戻すことはできません。このエクスプレッション編集モードは、それが必要なものであることが確かでない限り、選択しないでください。

10. 手順 6 ～8 を繰り返して、必要な式を [式] リストに追加します。名前付きエクスプレッションとエクスプレッションエディタ (Expressions Editor) で作成したエクスプレッションを混在させることができます。Citrix ADC アプライアンスでは、これらはすべて同じです。
11. [作成] をクリックして、新しいポリシーを作成します。新しいポリシーが [Policies] ペインのリストに表示されます。
12. ［閉じる］ をクリックします。追加のコンテンツフィルタポリシーを作成するには、前の手順を繰り返します。コンテンツフィルタポリシーを削除するには、[ポリシー] タブでポリシーを選択し、[削除] をクリックします。

コンテンツフィルタポリシーのバインド

October 7, 2021

各コンテンツフィルタポリシーをバインドして有効にする必要があります。ポリシーは、グローバルにバインドすることも、特定の仮想サーバにバインドすることもできます。グローバルにバインドされたポリシーは、仮想サーバに送信されるトラフィックがポリシーと一致するたびに評価されます。特定の仮想サーバにバインドされたポリシーは、その仮想サーバがポリシーと一致するトラフィックを受信したときにのみ評価されます。

CLI を使用してポリシーを仮想サーバーにバインドする

コマンドプロンプトで次のコマンドを入力して、ポリシーを仮想サーバーにバインドし、構成を確認します。

```
- bind lb vserver <name>@ -policyName <string> -priority <
   positive_integer>
- show lb vserver <name>
<!--NeedCopy-->
```

例:

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver vs-loadbal
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

 Done
<!--NeedCopy-->
```

CLI を使用してポリシーをグローバルにバインドする

コマンドプロンプトで次のコマンドを入力して、ポリシーをグローバルにバインドし、構成を確認します。


```
- bind filter global (<policyName> [-priority <positive_integer>]) [-
    state ( ENABLED | DISABLED )]
- show filter global
<!--NeedCopy-->
```

例:

```
 bind filter global cf-pol -priority 1
 Done show filter global
1)      Policy Name: cf-pol     Priority: 1
 Done
<!--NeedCopy-->
```

GUI を使用してポリシーを仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. 詳細ペインで、リストからコンテンツフィルタポリシーをバインドする仮想サーバーを選択し、[開く] をクリックします。
3. [仮想サーバーの構成 (負荷分散)] ダイアログボックスで、[ポリシー] タブを選択し、仮想サーバーにバインドするフィルタポリシーの [アクティブ] 列のチェックボックスをオンにします。
4. [**OK**] をクリックします。バインドしたポリシーには、[ポリシー] タブの [ポリシーバインド] 列に、チェックマークと [はい] が表示されます。

GUI を使用してポリシーをグローバルにバインドする

1. [セキュリティ **] > [** 保護機能 **] > [** フィルタ **]** に移動します。
2. 詳細ウィンドウの [ポリシー] タブで、バインドするポリシーを選択し、[グローバルバインド] をクリックします。
3. [フィルタポリシーのバインド/バインド解除] ダイアログボックスの [ポリシー名] ドロップダウンリストでポリシーを選択し、[追加] をクリックします。ポリシーが [構成済み] リストに追加されます。

   > 注
   >
   > リストから複数のポリシーを選択するには、Ctrl キーを押しながら目的のポリシーをクリックします。

4. [**OK**] をクリックし、[**Close**] をクリックします。バインドしたポリシーには、[ポリシー] タブの [グローバルバインド] 列に、チェックマークと [はい] が表示されます。

よく使用される展開シナリオのコンテンツフィルターの構成

October 7, 2021

この例では、構成ユーティリティを使用して、要求された URL に root.exe または cmd.exe が含まれている場合、コンテンツフィルターポリシー`filter-CF-nimda`が評価され、接続がリセットされるコンテンツフィルターポリシーを実装する手順を示します。

このコンテンツフィルタポリシーを構成するには、次の操作を行う必要があります。

- コンテンツフィルタを有効にする
- コンテンツフィルタポリシーを構成する
- コンテンツフィルタポリシーをグローバルにバインドするか、仮想サーバにバインドする
- 構成を確認します

注: この例ではデフォルトのコンテンツフィルタアクションを使用しているため、個別のコンテンツフィルタアクションを作成する必要はありません。

コンテンツフィルタを有効にする

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] をクリックします。
2. 詳細ウィンドウの [モードと機能] で、[基本機能の変更] をクリックします。
3. [基本機能の構成] ダイアログボックスで、[コンテンツフィルタ] チェックボックスをオンにし、[OK] をクリックします。
4. [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。ステータスバーに、選択した機能が有効であることを示すメッセージが表示されます。

コンテンツフィルタリングポリシー **`filter-CF-nimda`** を構成する

1. [セキュリティ] > [保護機能] > [フィルタ] に移動します。
2. 詳細ペインで、[Add] をクリックします。[フィルタポリシーの作成] ダイアログボックスが表示されます。
3. [フィルタポリシーの作成] ダイアログボックスの [フィルタ名] テキストボックスに名前`filter-CF-nimda`を入力します。
4. 「アクションのリクエスト」オプションを選択し、ドロップダウンリストで「リセット」を選択します。
5. [式] フレームで、ドロップダウンリストから [任意の式に一致] を選択し、[追加] をクリックします。
6. [式の追加] ダイアログボックスの [式の種類] ドロップダウンリストで、[一般] を選択します。
7. [フロータイプ] ドロップダウンリストで、[REQ] を選択します。
8. [プロトコル] ドロップダウンリストで、[HTTP] を選択します。
9. [修飾子] ドロップダウンリストで、[URL] を選択します。
10. [演算子] ドロップダウンリストで、[続く] を選択します。
11. [値] ボックスに「cmd.exe」と入力し、[OK] をクリックします。式が [式] テキストボックスに追加されます。

12. 別の式を作成するには、手順 7 ～11 を繰り返しますが、[値] テキストボックスに「root.exe」と入力します。[OK] をクリックし、最後に [閉じる] をクリックします。
13. [フィルタポリシーの作成] ダイアログボックスの [作成] をクリックします。フィルタポリシー`filter-CF-nimda`が [Filter] リストに表示されます。
14. [閉じる] をクリックします。

コンテンツフィルタポリシーをグローバルにバインドする

1. [セキュリティ] > [保護機能] > [フィルタ] に移動します。右側のペインに [Filter] ページが表示されます。
2. 詳細ペインの [ポリシー] タブで、バインドするポリシーを選択し、[グローバルバインド] をクリックします。[フィルタポリシーのバインド/バインド解除] ダイアログボックスが表示されます。
3. [フィルタポリシーのバインド/バインド解除] ダイアログボックスの [ポリシー名] ドロップダウンリストでポリシー`filter-CF-nimda`を選択し、[追加] をクリックします。ポリシーが [構成済み] リストに追加されます。
4. [OK] をクリックし、[Close] をクリックします。バインドしたポリシーでは、[ポリシー] タブの [グローバルバインド] 列にチェックマークと [はい] が表示されます。

コンテンツフィルタポリシーを仮想サーバーにバインドする

1. [Traffic Management] > [Load Balancing] > [Virtual Servers] の順に選択します。
2. 詳細ペインの [仮想サーバー] リストで、コンテンツフィルタポリシーをバインドする vserver-CF-1 を選択し、[開く] をクリックします。
3. [仮想サーバーの構成（負荷分散）] ダイアログボックスで、[ポリシー] タブを選択します。
4. [Active] 列で、ポリシー`filter-CF-nimda`のチェックボックスをオンにし、[OK] をクリックします。コンテンツフィルタポリシーがアクティブになり、フィルタ要求である必要があります。正しく機能している場合、root.exe または cmd.exe を含む URL のリクエストがあるたびに select カウンタが増加します。これにより、コンテンツフィルタポリシーが機能していることを確認できます。コンテンツフィルタポリシーは、仮想サーバーにバインドされます。

コマンドラインインターフェイスを使用してコンテンツフィルタの設定を確認する

コマンドプロンプトで次のコマンドを入力して、コンテンツフィルタの構成を確認します。

```
show filter policy filter-CF-nimda
```

例:

```
1 sh filter policy filter-CF-nimda
2         Name: filter-CF-nimda   Rule: REQ.HTTP.URL CONTAINS cmd.exe ||
              REQ.HTTP.URL CONTAINS root.exe
3         Request action: RESET
```



```
        Response action:
        Hits: 0
Done
<!--NeedCopy-->
```

> 注
>
> Select カウンタには、filter-CF-nimdaポリシーが評価される回数を示す整数が表示されます。前述の手順では、cmd.exe または root.exe を含む URL に対するリクエストがまだ行われていないため、select カウンタは 0 に設定されます。カウンタの増分をリアルタイムで確認する場合は、これらの文字列のいずれかを含む URL を要求するだけです。

GUI を使用してコンテンツフィルタ設定を確認する

1. [セキュリティ] > [保護機能] > [フィルタ] に移動します。
2. 詳細ペインで、フィルタポリシーfilter-CF-nimdaを選択します。ペインの下部に次のものが表示されている必要があります。

```
    \*\*Request Action:\*\*

    RESET

    \*\*Rule:\*\*

    REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe

    \*\*Hits:\*\*

    0
<!--NeedCopy-->
```

トラブルシューティング

October 7, 2021

構成後にコンテンツフィルタリング機能が期待どおりに機能しない場合は、一般的なツールを使用して Citrix ADC リソースにアクセスし、問題を診断できます。

トラブルシューティングのリソース

以下のツールとリソースを使用して、Citrix ADC アプライアンスのコンテンツフィルタリングの問題のほとんどをトラブルシューティングできます。

- Citrix ADC トレースファイル用にカスタマイズされた Wireshark アプリケーション
- リソースへのアクセス時に記録されたトレースファイル
- 設定ファイル
- ns.log ファイル
- `iehttpheaders`、または Fiddler トレースまたは同様のユーティリティ

コンテンツフィルタに関する問題のトラブルシューティング

コンテンツフィルタに関する問題のトラブルシューティングを行うには、次の手順を実行します。

- 機能が有効になっていることを確認します。
- コンテンツフィルタポリシーが正しく構成されていることを確認します。着信要求を評価する式には特に注意してください。

  > 注
  >
  > ほとんどのコンテンツフィルタの問題は、誤った構成が原因で発生し、エラーはポリシー構成で最も頻繁に発生します。

- ポリシーの Select カウンタをチェックして、増加していることを確認します。そうでない場合、ポリシーは評価されません。
- ポリシーが評価され、必要なフィルタリングがまだ実行されない場合は、ポリシーの式とアクションを調べる必要があります。
- ポリシーの式が有効と思われる場合は、単純な NSTRUE 値を割り当ててテストし、式の評価によって問題が発生していないかどうかを確認します。
- フィルタリングがリクエストとレスポンスのどちらに基づいている必要があるかを再評価します。
- アクションが正しく構成されていることを確認します。たとえば、カスタムアクションを使用してリクエストのヘッダーを破損する場合は、アクションのヘッダー名が正しいことを確認します。ヘッダー名がわからない場合は、`iehttpheaders`または同様のユーティリティを使用してブラウザを起動し、リクエスト内のヘッダーを確認します。この機能を使用すると、パケットが Citrix ADC アプライアンスの外に出るときに適切なアクションが実行されるかどうかを調べることができます。
- `iehttpheaders`または Fiddler トレースは、クライアントに記録されたヘッダーオプションと名前、クライアント側のリクエストヘッダー、および応答ヘッダーを見つけるのに役立ちます。
- 要求ヘッダーに加えられた変更を確認するには、Citrix ADC アプライアンスに ns トレースを記録するか、サーバー上で Wireshark トレースを記録します。

- 上記のいずれの方法でも問題が解決しない場合は、接続が追跡不能になっていないことを確認します。これは、特定の状況で発生する可能性があります。接続が追跡不能になった場合、アプライアンスはリクエストのアプリケーションレベルの処理を実行しません。その場合は、Citrix テクニカルサポートにお問い合わせください。

サージ保護

January 25, 2022

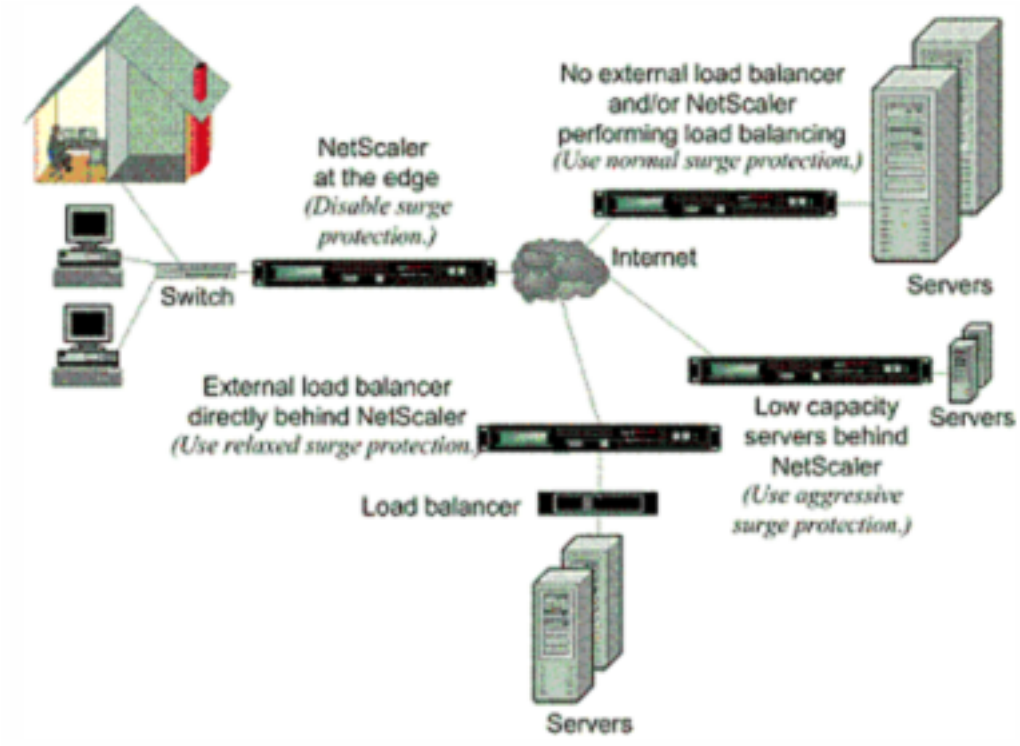
クライアント要求の急増によってサーバが過負荷になると、サーバの応答が遅くなり、サーバは新しい要求に応答できなくなります。サージ保護機能により、サーバが処理できる速度でサーバへの接続が確実に行われます。応答率は、サージ保護の設定方法によって異なります。Citrix ADC アプライアンスはサーバーへの接続数も追跡し、その情報を使用して新しいサーバー接続を開く速度を調整します。

サージ保護はデフォルトで有効になっています。いくつかの特別な構成の場合のように、サージ保護を使用しない場合は、それを無効にする必要があります。

ほとんどの用途にはデフォルトのサージ保護設定で十分ですが、サージ保護を構成して必要に応じて調整できます。まず、スロットル値を設定して、接続試行をどれだけ積極的に管理するかを知ることができます。次に、基本しきい値を設定して、Citrix ADC アプライアンスがサージ保護をトリガーする前に許可する同時接続の最大数を制御できます。(デフォルトのベースしきい値はスロットル値によって設定されますが、スロットル値を設定した後は任意の数値に変更できます)。

次の図は、Web サイトへのトラフィックを処理するようにサージ保護がどのように構成されているかを示しています。

図 1: Citrix ADC サージ保護の機能図



> 注
>
> Citrix ADC アプライアンスがインターネットのクライアント側のネットワークデバイスとやり取りするネットワークのエッジにインストールされている場合は、サージ保護機能を無効にする必要があります。アプライアンスで USIP (ソース IP を使用) モードを有効にする場合は、サージ保護も無効にする必要があります。

次の例と図は、2 つのケースのリクエスト率と応答率を示しています。一方ではサージ保護が無効になり、もう一方ではサージ保護が有効になります。

サージ保護を無効にして要求が急増すると、サーバーは同時に処理できる限り多くの要求を受け入れ、要求の破棄を開始します。サーバの過負荷が増えるにつれて、サーバはダウンし、応答速度はゼロに低下します。数分後にサーバーがクラッシュから回復すると、異常な動作であるすべての保留中の要求のリセットが送信され、リセットで新しい要求にも応答します。このプロセスは、要求の急増ごとに繰り返されます。したがって、DDoS 攻撃を受けて複数の要求が急増するサーバーは、正当なユーザーが使用できなくなる可能性があります。

サージ保護が有効になっていて要求の急増が発生した場合、サージ保護はサーバーへの要求のレートを管理し、サーバーが要求を処理できる速さでサーバーに要求を送信します。これにより、サーバーは各要求に受信順に正しく応答できます。急増が終わると、バックログされたリクエストは、リクエストレートがレスポンスレートと一致するまで、サーバーが処理できる限り早くクリアされます。

サージ保護の無効化と再有効化

October 7, 2021

サージ保護機能はデフォルトで有効になっています。サージ保護を有効にすると、追加したサービスに対してアクティブになります。

CLI を使用したサージ保護の無効化または再有効化

コマンドプロンプトで、次のいずれかのコマンドセットを入力して、サージ保護を無効または再度有効にし、構成を確認します。

```
- disable ns feature SurgeProtection
- show ns feature
- enable ns feature SurgeProtection
- show ns feature
<!--NeedCopy-->
```

例:

```
disable ns feature SurgeProtection
 Done show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  OFF
 .
 .
 .
 23)    HTML Injection                  HTMLInjection       ON
 24)    Citrix ADC Push                  push                OFF
 Done
<!--NeedCopy-->
```

```
 enable ns feature SurgeProtection
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  ON
 .
```



```
 .
 .

 23)    HTML Injection                  HTMLInjection        ON
 24)    Citrix ADC Push                  push                 OFF
 Done
>
<!--NeedCopy-->
```

GUI を使用したサージ保護の無効化または再有効化

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] を選択します。
2. 詳細ウィンドウで、[高度な機能の変更] をクリックします。
3. [高度な機能の構成] ダイアログボックスで、[サージ保護] チェックボックスをオフにしてサージ保護機能を無効にするか、チェックボックスをオンにして機能を有効にします。
4. [**OK**] をクリックします。
5. [機能の有効化/無効化] ダイアログボックスで、[はい] をクリックします。ステータスバーに、機能が有効または無効になったことを示すメッセージが表示されます。

GUI を使用して、特定のサービスのサージ保護を無効または再度有効にします

1. [Traffic Management] > [Load Balancing] > [Services] の順に移動します。構成済みサービスのリストが詳細ペインに表示されます。
2. 詳細ウィンドウで、サージ保護機能を無効にするか、再度有効にするサービスを選択し、[開く] をクリックします。
3. [サービスの構成] ダイアログボックスで、[詳細設定] タブをクリックし、下にスクロールします。
4. [その他] フレームで、[サージ保護] チェックボックスの選択を解除してサージ保護機能を無効にするか、チェックボックスをオンにして機能を有効にします。
5. [OK] をクリックします。ステータスバーに、機能が有効または無効になったことを示すメッセージが表示されます。
   注：サージ保護は、機能とサービス設定の両方が有効になっている場合にのみ機能します。

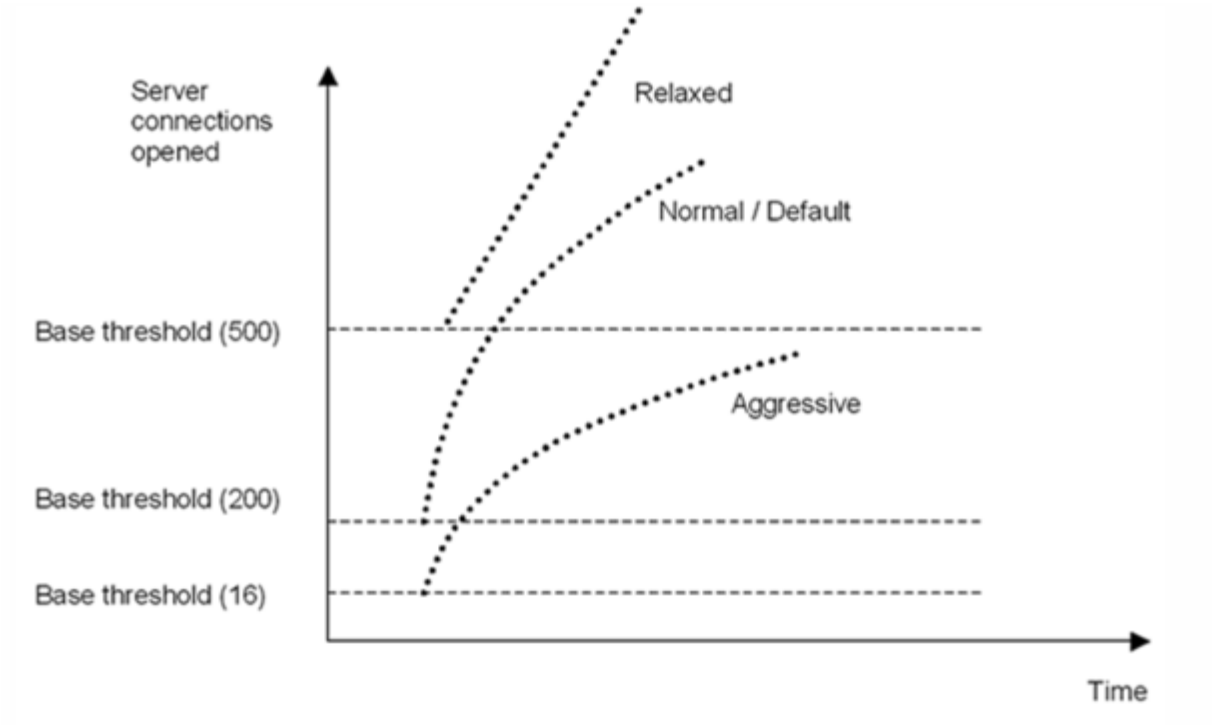
サージ保護のしきい値を設定

October 7, 2021

Citrix ADC アプライアンスがサーバーへの接続を開く頻度を設定するには、サージ保護のしきい値およびスロットル値を設定する必要があります。

次の図は、スロットルレートをリラックス、ノーマル、またはアグレッシブに設定した結果、サージ保護曲線を示しています。サーバ容量の設定に応じて、基準しきい値を設定して、適切なサージ保護曲線を生成できます。

図 1：サージ保護曲線

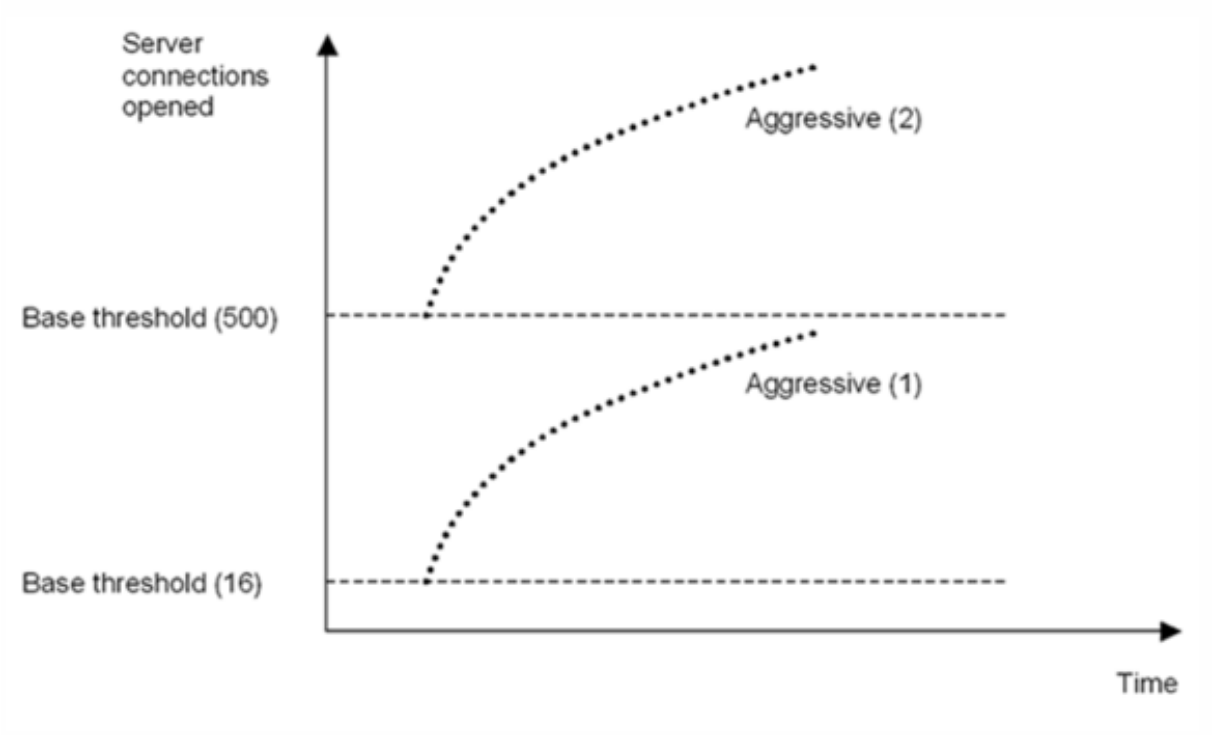


構成設定は、次の方法でサージ保護の動作に影響します。

- スロットルレートを指定しない場合、スロットルレートは標準（デフォルト値）に設定され、ベースしきい値は 200 に設定されます（前の図を参照）。
- ベースしきい値を指定せずにスロットルレート（アグレッシブ、ノーマル、リラックス）を指定すると、そのスロットルレートのベースしきい値のデフォルト値がカーブに反映されます。たとえば、スロットルレートを緩和に設定すると、作成されるカーブのベースしきい値は 500 になります。
- 基本しきい値だけを指定すると、次の図に示すように、指定した値に応じて、サージ保護曲線全体が上下にシフトします。
- 基本しきい値とスロットルレートの両方を指定した場合、サージ保護曲線は設定されたスロットルレートに基づいて生成され、基本しきい値に設定された値に従って調整されます。

次の図では、スロットルレートがアグレッシブに設定されているが、基本しきい値が設定されていない場合に、低いカーブ（アグレッシブ 1）の結果を示します。上限カーブ（アグレッシブ 2）は、基本しきい値が 500 に設定されているが、スロットルレートが設定されていない場合に発生します。2 番目の上側のカーブ（アグレッシブ 2）は、基本しきい値が 500 に設定され、スロットルレートがアグレッシブに設定されている場合にも発生します。

図 2：デフォルトまたは設定されたベースしきい値を使用したアグレッシブレート



GUI を使用してサージ保護のしきい値を設定

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] を選択します。
2. 詳細ウィンドウで、[グローバルシステム設定] をクリックします。
3. スロットル・レートのデフォルトとは異なる基本しきい値を設定する場合は、［グローバル設定の構成］ダイアログ・ボックスの［基本しきい値］テキスト・ボックスに、サージ保護がトリガーされるまでに許可される同時サーバ接続の最大数を入力します。基本しきい値は、サージ保護が有効になる前に開くことができるサーバー接続の最大数です。この設定の最大値は 32,767 サーバー接続です。この値のデフォルト設定は、次のステップで選択したスロットルレートによって制御されます。

   注：ここで明示的な値を設定しない場合、デフォルト値が使用されます。
4. [スロットル] ドロップダウンリストで、スロットルレートを選択します。スロットルは、Citrix ADC アプライアンスがサーバーへの接続を許可する速度です。スロットルは次の値に設定できます。

   - **Aggressive:** サーバーの接続処理能力とサージ処理能力が低く、接続を慎重に管理する必要がある場合に、このオプションを選択します。スロットルをアグレッシブに設定すると、基本しきい値はデフォルト値の 16 に設定されます。つまり、サーバへの同時接続数が 17 個以上ある場合は常にサージ保護がトリガーされます。
   - 通常： Citrix ADC アプライアンスまたはダウンストリームの背後に外部ロードバランサーがない場合、このオプションを選択します。基本しきい値は 200 に設定されています。これは、サーバーへの同時接続が 201 以上ある場合にサージ保護がトリガーされることを意味します。「標準」は、デフォルトのスロットル・オプションです。

- **Relaxed**： Citrix ADC アプライアンスが多数の Web サーバー間で負荷分散を実行し、多数の同時接続を処理できる場合に、このオプションを選択します。基本しきい値は 500 に設定されています。つまり、サーバへの同時接続が 501 以上ある場合にのみサージ保護がトリガーされます。

5. ［OK］をクリックします。ステータスバーに、グローバル設定が構成されていることを示すメッセージが表示されます。

サージキューをフラッシュする

October 7, 2021

物理サーバが要求の急増を受信すると、現在接続しているクライアントへの応答が遅くなり、ユーザが不満になり、不満や不満が残ります。多くの場合、オーバーロードにより、クライアントはエラーページを受信します。このような過負荷を回避するために、Citrix ADC アプライアンスは、サービスへの新しい接続を確立する速度を制御するサージ保護などの機能を提供します。

アプライアンスは、クライアントと物理サーバ間の接続の多重化を行います。サーバ上のサービスにアクセスするためのクライアント要求を受信すると、アプライアンスはすでに確立されているサーバへの接続を空いているか探します。空き接続が見つかった場合、その接続を使用してクライアントとサーバー間の仮想リンクを確立します。既存の空き接続が見つからない場合、アプライアンスはサーバとの新しい接続を確立し、クライアントとサーバ間の仮想リンクを確立します。ただし、アプライアンスがサーバーとの新しい接続を確立できない場合、クライアント要求をサージキューに送信します。負荷分散またはコンテンツスイッチング仮想サーバーにバインドされているすべての物理サーバーがクライアント接続の上限（最大クライアント値、サージ保護しきい値、またはサービスの最大容量）に達した場合、アプライアンスはどのサーバーとも接続を確立できません。サージ保護機能は、サージキューを使用して、物理サーバとの接続が開かれる速度を調整します。アプライアンスは、仮想サーバにバインドされたサービスごとに異なるサージキューを維持します。

サージキューの長さは、アプライアンスが接続を確立できない要求が来ると増加し、キュー内の要求がサーバーに送信されるか、要求がタイムアウトしてキューから削除されるたびに減少します。

サービスまたはサービスグループのサージキューが長すぎる場合は、フラッシュする必要があります。特定のサービスまたはサービスグループ、または負荷分散仮想サーバーにバインドされているすべてのサービスとサービスグループのサージキューをフラッシュできます。サージキューをフラッシュしても、既存の接続には影響しません。サージキューに存在する要求だけが削除されます。これらの要求の場合、クライアントは新しい要求を行う必要があります。

コンテンツスイッチング仮想サーバーのサージキューをフラッシュすることもできます。コンテンツスイッチング仮想サーバーが特定の負荷分散仮想サーバーにいくつかの要求を転送し、負荷分散仮想サーバーが他のいくつかの要求も受信する場合、コンテンツスイッチング仮想サーバーのサージキューをフラッシュすると、このコンテンツスイッチングから受信した要求のみが仮想サーバーがフラッシュされます。負荷分散仮想サーバーのサージキュー内の他の要求はフラッシュされません。

注：

> - キャッシュリダイレクト、認証、VPN、または GSLB 仮想サーバーまたは GSLB サービスのサージキューをフラッシュすることはできません。
> - [ソース IP の使用（USIP）] が有効になっている場合は、サージ保護機能を使用しないでください。

CLI を使用したサージキューのフラッシュ

flush ns surgeQ コマンドは、次のように動作します。

- サージキューをフラッシュする必要があるサービス、サービスグループ、または仮想サーバの名前を指定できます。
- コマンドの実行中に名前を指定すると、指定したエンティティのサージキューがフラッシュされます。複数のエンティティが同じ名前を持つ場合、アプライアンスはこれらすべてのエンティティのサージキューをフラッシュします。
- コマンドの実行中にサービスグループの名前、サーバー名、およびポートを指定すると、アプライアンスは指定されたサービスグループメンバーのみのサージキューをフラッシュします。
- サービスグループ<name>の名前を指定せずにサービスグループメンバー<serverName> and <port>を直接指定することはできません。また、<serverName>なしで<port>を指定することはできません。特定のサービスグループメンバーのサージキューをフラッシュする場合は、<serverName>および<port>を指定します。
- 名前を指定せずにコマンドを実行すると、アプライアンスはアプライアンスに存在するすべてのエンティティのサージキューをフラッシュします。
- サービスグループメンバーがサーバ名で識別される場合は、このコマンドでサーバ名を指定する必要があります。その IP アドレスを指定することはできません。

コマンドプロンプトで入力します。

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
```

例

1. `flush ns surgeQ -name SVC1ANZGB -serverName 10.10.10.1 80`

   前述のコマンドは、SVC1ANZGB という名前で IP アドレスが 10.10.10 のサービスまたは仮想サーバのサージキューをフラッシュします。

2. `flush ns surgeQ`

   前述のコマンドは、アプライアンス上のすべてのサージキューをフラッシュします。

GUI を使用したサージキューのフラッシュ

Traffic Management > Content Switching > Virtual Servers に移動して仮想サーバーを選択し、アクションリストで Flush Surge Queue を選択します。

DNS セキュリティオプション

October 7, 2021

Citrix ADC GUI の［DNS セキュリティプロファイルの追加］ページで DNS セキュリティオプションを構成できるようになりました。Citrix ADC CLI または NITRO API から DNS セキュリティオプションを構成するには、AppExpert コンポーネントを使用します。手順については、NITRO API のドキュメントと Citrix ADC コマンドリファレンスガイドを参照してください。

キャッシュポイズニング保護というオプションの 1 つはデフォルトで有効になっており、無効にすることはできません。次の表に示すように、他のオプションを、展開内のすべての DNS エンドポイントまたは特定の DNS 仮想サーバーに適用できます。

セキュリティオプション	すべての DNS エンドポイントに適用できますか。	特定の DNS 仮想サーバーに適用できますか。
DNS DDoS 保護	はい	はい
例外の管理 — ホワイトリスト/ブラックリストサーバ	はい	はい
ランダムなサブドメイン攻撃を防ぐ	はい	はい
キャッシュをバイパスする	はい	いいえ
TCP 経由で DNS トランザクションを強制する	はい	はい
DNS 応答にルートの詳細を提供する	はい	いいえ

キャッシュポイズニング保護

キャッシュポイズニング攻撃は、正当なサイトから悪意のある Web サイトにユーザーをリダイレクトします。

たとえば、攻撃者は DNS キャッシュ内の正規の IP アドレスを制御する偽の IP アドレスに置き換えます。サーバーがこれらの IP アドレスからの要求に応答すると、キャッシュはポイズニングされます。ドメインのアドレスに対する後続のリクエストは、攻撃者のサイトにリダイレクトされます。

[キャッシュポイズニング保護] オプションを使用すると、DNS サーバーの要求と応答をキャッシュするデータベースに破損したデータが挿入されるのを防ぐことができます。この機能は Citrix ADC アプライアンスに組み込まれており、常に有効になっています。

DNS DDoS 保護

DDoS 攻撃で使用される可能性があると思われる要求の種類ごとに、DNS DDoS 保護オプションを設定できます。アプライアンスは、タイプごとに、指定した期間（タイムスライス）で受信した要求数のしきい値を超えた後に受信したすべての要求をドロップします。このオプションを設定して、SYSLOG サーバに警告を記録することもできます。次に例を示します：

- **DROP:** -ログなしでリクエストをドロップするには、このオプションを選択します。しきい値 15、タイムスライス 1 秒で A レコード保護を有効にし、[DROP] を選択したと仮定します。着信要求が 1 秒で 15 クエリを超えると、パケットがドロップし始めます。
- 警告**:** -リクエストをログに記録およびドロップするには、このオプションを選択します。しきい値 15、タイムスライス 1 秒で A レコード保護を有効にし、[WARN] を選択したと仮定します。着信要求が 1 秒で 15 クエリを超えると、脅威を示す警告メッセージがログに記録され、パケットがドロップされます。WARN のしきい値を、レコードタイプの DROP のしきい値よりも小さく設定することをお勧めします。このような設定は、実際の攻撃が発生して Citrix ADC が着信要求のドロップを開始する前に警告メッセージを記録することにより、管理者が攻撃を識別するのに役立ちます。

GUI を使用して着信トラフィックのしきい値を設定する

1. [設定] > [セキュリティ] > [**DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3. [**DNS** セキュリティプロファイルの追加**]** ページで、次の操作を行います。
4. [**DNS DDoS** 保護] を展開します。
   a) レコードタイプを選択し、しきい値制限とタイムスライス値を入力します。
   b) [ドロップ] または [警告] を選択します。
   c) 保護するその他のレコードの種類ごとに、手順 a と b を繰り返します。
5. [**Submit**] をクリックします。

例外の管理 **—** 許可リスト**/**ブロックリストサーバー

例外の管理を使用すると、ブロックリストまたは許可リストのドメイン名と IP アドレスに例外を追加できます。次に例を示します：

- 攻撃をポストする特定の IP アドレスが特定されると、そのような IP アドレスをブロックリストに追加できます。
- 管理者が特定のドメイン名に対する要求数が予期せず多いことがわかった場合、そのドメイン名をブロックリストに追加できます。
- サーバーリソースを消費する可能性がある`NXDomains` および既存のドメインの一部はブラックリストに登録できます。
- 管理者がリストドメイン名または IP アドレスを許可すると、これらのドメインまたは IP アドレスからのクエリまたは要求のみが応答され、その他はすべて削除されます

GUI を使用して許可リストまたはブロックリストを作成する

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3. [**DNS** セキュリティプロファイルの追加**]** ページで、次の操作を行います。
   a) [例外の管理**] — [**サーバーのホワイトリスト**/**ブラックリスト**]** を展開します。
   b) ブラックリストに登録されたドメイン/アドレスからのクエリをブロックするには [ブロック] を選択し、ホワイトリストに登録されたドメイン/アドレスからのクエリを許可するには **[**のみ許可] を選択します。
   c) [ドメイン名**/ IP** アドレス] ボックスに、ドメイン名、IP アドレス、または IP アドレスの範囲を入力します。エントリを区切るには、カンマを使用します。
   注記:「詳細オプション」**(Advanced Options**) を選択した場合は、「次で始まる」、「次を含む」(contains)、および「次で終わる」(ends with) オプションを使用して条件を設定できます。
   たとえば、「image」で始まる、または「.co.ru」で終わる、または「モバイルサイト」を含む DNS クエリをブロックする条件を設定できます。
4. [**Submit**] をクリックします。

ランダムなサブドメイン攻撃を防ぐ

ランダムなサブドメイン攻撃では、正当なドメインのランダムで存在しないサブドメインにクエリが送信されます。この操作により、DNS リゾルバとサーバーの負荷が増加します。その結果、過負荷になり、減速する可能性があります。

[ランダムサブドメイン攻撃を防止] オプションは、指定した長さを超える DNS クエリをドロップするように DNS レスポンダに指示します。

example.com が所有するドメイン名であるため、解決要求が DNS サーバーに送信されるとします。攻撃者は example.com にランダムなサブドメインを追加して、リクエストを送信することができます。指定したクエリの長さと FQDN に基づいて、ランダムクエリがドロップされます。

たとえば、クエリが www.image987trending.example.com の場合、クエリの長さが 20 に設定されている場合、クエリは削除されます。

GUI を使用した **DNS** クエリの長さの指定

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3. [**DNS** セキュリティプロファイルの追加**]** ページで、次の操作を行います。
   a) [ランダムなサブドメイン攻撃を防止] を展開します。
   b) クエリの長さの数値を入力します。
4. [**Submit**] をクリックします。

キャッシュのバイパス

攻撃中は、すでにキャッシュされているデータを保護する必要があります。キャッシュを保護するために、特定のドメイン、レコードタイプ、またはレスポンスコードに対する新しいリクエストをキャッシュではなくオリジンサーバーに送信できます。

[キャッシュのバイパス］オプションを選択すると、Citrix ADC アプライアンスは、攻撃が検出されたときに、指定されたドメイン、レコードタイプ、または応答コードのキャッシュをバイパスするように指示します。

GUI を使用して、指定したドメイン、レコードタイプ、または応答タイプのキャッシュをバイパスする

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3. **[DNS** セキュリティプロファイルの追加] ページで、[キャッシュのバイパス] を展開し、ドメイン名を入力します。必要に応じて、キャッシュをバイパスする必要があるレコードタイプまたはレスポンスタイプを選択します。
   - [ドメイン] をクリックし、ドメイン名を入力します。エントリを区切るには、カンマを使用します。
   - [レコードタイプ**]** をクリックして、レコードタイプを選択します。
   - 「応答タイプ」をクリックし、応答タイプを選択します。
4. [**Submit**］をクリックします。

TCP 経由で **DNS** トランザクションを強制する

トランザクションが UDP ではなく TCP を使用するように強制される場合、一部の DNS 攻撃を防ぐことができます。たとえば、ボット攻撃の間、クライアントは大量のクエリを送信しますが、応答を処理できません。これらのトランザクションに TCP の使用が強制される場合、ボットは応答を理解できないため、TCP 経由で要求を送信できません。

GUI を使用して **TCP** レベルでドメインまたはレコードタイプを強制的に動作させる

1. [構成**] > [**セキュリティ**] > [DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3. [**DNS** セキュリティプロファイルの追加**]** ページで、[**TCP** 経由の **DNS** トランザクションの強制] を展開し、ドメイン名と/を入力するか、TCP 経由で DNS トランザクションを強制する必要があるレコードタイプを選択します。
   - [ドメイン] をクリックし、ドメイン名を入力します。エントリを区切るには、カンマを使用します。
   - [****** レコードタイプ **] をクリックし、レコードタイプを選択します。
4. [**Submit**］をクリックします。

DNS 応答にルートの詳細を提供する

一部の攻撃では、Citrix ADC アプライアンスに構成またはキャッシュされていない無関係なドメインに対して、攻撃者が大量のクエリを送信します。`dnsRootReferral`パラメータが ENABLED の場合、すべてのルートサーバーを公開します。

[DNS レスポンスでルート詳細を提供する] オプションを選択すると、Citrix ADC アプライアンスは、構成またはキャッシュされていないクエリのルート参照へのアクセスを制限するように指示します。アプライアンスは空白の応答を送信します。

[DNS 応答にルートの詳細を提供する] オプションでは、増幅攻撃を軽減またはブロックすることもできます。dnsrootReferral パラメータが無効の場合、Citrix ADC 応答にルート紹介が存在しないため、それらは増幅されません。

GUI を使用してルートサーバーへのアクセスを有効または無効にする

1. [構成] **>** [セキュリティ] **>** [**DNS** セキュリティ] に移動します。
2. [**DNS** セキュリティプロファイル] ページで、[追加] をクリックします。
3. [**DNS** セキュリティプロファイルの追加] ページで、次の操作を行います。
   a) **DNS** 応答で [ルートの詳細を提供する] を展開します。
   b) [**ON**] または [**OFF**] をクリックして、ルートサーバへのアクセスを許可または制限します。
4. [**Submit**] をクリックします。

システム

October 7, 2021

このセクションでは、Citrix ADC システムレベルの情報を提供します。これには、システムレベルの機能の詳細な説明、機能を使用できるシナリオ、設定手順、および機能をよりよく理解するための例が含まれます。

- 基本操作
- 認証と承認
- TCP 構成
- HTTP 構成
- SNMP
- 監査ログ
- Web サーバーのロギング
- Call Home
- レポート作成ツール
- CloudBridge Connector
- 高可用性

- TCP 最適化

システムベースのオペレーション

April 7, 2022

次の構成では、Citrix ADC アプライアンスでシステムベースの操作を実行できます。

Citrix ADC 構成を表示、保存、およびクリアする方法

Citrix ADC 構成は、/nsconfig/ns.conf directoryに格納されます。セッション間で構成を使用できるようにするには、構成を変更するたびに構成を保存する必要があります。

コマンドインターフェイスを使用して実行構成を表示する

コマンドプロンプトで入力します。

```
show ns runningConfig
```

GUI を使用した実行構成の表示

1. [システム] > [診断] に移動し、[構成の表示] グループで [実行構成] をクリックします。

コマンドインターフェイスを使用して、**2** つの構成ファイルの違いを表示します

コマンドプロンプトで入力します。

```
diff ns config <configfile> <configfile2>
```

GUI を使用して **2** つの設定ファイルの違いを表示します

1. [システム] > [診断] に移動し、[構成の表示] グループで [構成の違い] をクリックします。

コマンドインターフェイスを使用して **Citrix ADC** 構成を保存します

コマンドプロンプトで入力します。

```
save ns config
```

GUI を使用して **Citrix ADC** 構成を保存する

1. [構成] タブの右上隅にある [保存] アイコンをクリックします。

コマンドインターフェイスを使用した保存済み設定の表示

コマンドプロンプトで入力します。

```
show ns ns.conf
```

GUI を使用した保存済み設定の表示

[システム] > [診断] に移動し、[構成の表示] グループで [保存された構成] をクリックします。

コマンドインターフェイスを使用して **Citrix ADC** 構成をクリアします

Citrix ADC 構成をクリアするには、次の 3 つのオプションがあります。

基本レベル。基本レベルで設定をクリアすると、次の設定を除くすべての設定がクリアされます。

- Nsroot: パスワード
- タイムゾーン
- NTP サーバー
- ADM サーバー接続
- ライセンスファイル情報
- NSIP、MIP、および切り取り
- ネットワーク設定（デフォルトゲートウェイ、VLAN、RHI、NTP、および DNS 設定）
- HA ノード定義
- 機能およびモードの設定
- デフォルトの管理者パスワード (nsroot)

拡張レベル。拡張レベルで設定をクリアすると、次の設定を除くすべての設定がクリアされます。

- NSIP, MIP(s), and SNIP(s)
- Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)
- HA node definitions

機能およびモードの設定は、デフォルト値に戻ります。

フルレベル。設定をフルレベルでクリアすると、すべての設定が工場出荷時のデフォルト値に戻ります。ただし、NSIP とデフォルトゲートウェイを変更すると、アプライアンスのネットワーク接続が失われる可能性があるため、変更されません。

コマンドプロンプトで入力します。

```
clear ns config -force
```

例: アプライアンスの基本設定を強制的にクリアする場合。

```
clear ns config -force basic
```


GUI を使用して **Citrix ADC** 構成をクリアします

[システム] > [診断] に移動し、[メンテナンス] グループで [設定のクリア] をクリックし、アプライアンスからクリアする設定レベルを選択します。

保存されていない **Citrix ADC** 構成のアプライアンスを再起動またはシャットダウンする方法

Citrix ADC アプライアンスは、使用可能なユーザーインターフェイスからリモートで再起動またはシャットダウンできます。スタンドアロンの Citrix ADC アプライアンスを再起動またはシャットダウンすると、保存されていない構成（
最後の save ns config コマンドが発行されてから実行された構成）は失われます。

高可用性設定では、プライマリアプライアンスがリブートまたはシャットダウンされると、セカンダリアプライアンスが引き継ぎ、プライマリになります。古いプライマリの未保存の設定は、新しいプライマリアプライアンスで使用できます。

また、Citrix ADC ソフトウェアを再起動するだけで、基盤となるオペレーティングシステムを再起動しないで、アプライアンスを再起動することもできます。これはウォームリブートと呼ばれます。たとえば、新しいライセンスを追加したり、IP アドレスを変更したりすると、Citrix ADC アプライアンスをウォームリブートしてこれらの変更を行うことができます。

> 注:
>
> ウォームリブートは、スタンドアロンの Citrix ADC アプライアンスでのみ実行できます。

コマンドインターフェイスを使用してアプライアンスを再起動します

コマンドプロンプトで入力します。

```
reboot [-warm]
```

GUI を使用して **Citrix ADC** アプライアンスを再起動します

1. 設定ページで、[**Reboot**] をクリックします。
2. 再起動を促すメッセージが表示されたら、[設定の保存（**Save configuration** ）] を選択して、設定が失われないようにします。

> 注:
>
> ウォームリブートを選択すると、ウォームリブートを実行できます。

コマンドインターフェイスを使用してアプライアンスをシャットダウンする

シェルプロンプトで、次のように入力します。

- `shutdown -p now`: ソフトウェアをシャットダウンし、Citrix ADC をオフにします。Citrix ADC MPX を再起動するには、AC 電源スイッチを押します。Citrix ADC VPX を再起動するには、VPX インスタンスを再起動します。
- `shutdown -h now`: ソフトウェアをシャットダウンし、Citrix ADC スイッチを入れたままにします。任意のキーを押して Citrix ADC を再起動します。このコマンドは、Citrix ADC をオフにしません。したがって、AC 電源をオフにしたり、AC 電源ケーブルを取り外したりしないでください。

> 注:
>
> Citrix ADC GUI を使用してアプライアンスをシャットダウンすることはできません。

システムクロックをネットワーク上のサーバーと同期させる方法

Citrix ADC アプライアンスを設定して、ローカルの時刻を、NTP（Network Time Protocol：ネットワークタイムプロトコル）サーバーの時刻と同期することができます。これにより、NetScaler のクロックの設定は、ネットワーク上のほかのサーバーと同じ日付と時刻になります。

アプライアンスでクロック同期を設定するには、GUI またはコマンドラインインターフェイスから NTP サーバエントリを ntp.conf ファイルに追加するか、または ntp.conf ファイルを手動で変更してから NTP デーモン（NTPD）を起動します。アプライアンスが再起動、アップグレード、またはダウングレードされても、クロック同期の設定は変更されません。ただし、高可用性セットアップでは、構成はセカンダリ Citrix ADC に伝播されません。

Citrix ADC GUI を使用すると、初回ユーザー（FTU）画面でクロック同期に必要なタイムゾーンと NTP サーバーの IP アドレスを構成できます。

> 注:
>
> ローカル NTP サーバを持っていない場合は、公式 NTP サイト<`http://www.ntp.org`>の Public Time Servers List の下に、パブリック、オープンアクセス、NTP サーバのリストがあります。パブリック NTP サーバーを使用するように Citrix ADC を構成する前に、「エンゲージメントのルール」ページ（すべてのパブリックタイムサーバーページにリンクが含まれています）を必ずお読みください。
>
> Citrix ADC リリース 11 では、NTP バージョンが 4.2.6p3 から 4.2.8p2 に更新されました。

前提要件

クロック同期を設定するには、次のエンティティを設定する必要があります。

1. NTP サーバ
2. NTP 同期。

コマンドインターフェイスを使用して **NTP** サーバを追加する

コマンドプロンプトで次のコマンドを入力して、NTP サーバを追加し、構成を確認します。

- add ntp server (<serverIP> | <serverName>)[-minpoll <positive_integer>] [-maxpoll <positive_integer>]
- show ntp server

例:

```
add ntp server 10.102.29.30 -minpoll 6 -maxpoll 11
```

GUI を使用して **NTP** サーバを追加する

[システム] > [**NTP** サーバ] に移動し、NTP サーバを作成します。

コマンドインターフェイスを使用した **NTP** 同期の有効化

NTP 同期を有効にすると、Citrix ADC は NTP デーモンを起動し、ntp.conf ファイルの NTP サーバーエントリを使用してローカル時刻設定を同期します。アプライアンスの時刻をネットワーク内の他のサーバと同期させたくない場合は、NTP 同期を無効にして、NTP デーモン（NTPD）を停止できます。

コマンドプロンプトで、次のコマンドのいずれかを入力します。

```
enable ntp sync
```

GUI を使用した **NTP** 同期の有効化

[システム] > [**NTP** サーバ] に移動し、[アクション] をクリックして [**NTP** 同期] を選択します。

GUI を使用して **ntp.conf** ファイルを編集するようにクロック同期を構成する

1. コマンドラインインターフェイスにログオンします。
2. シェルプロンプトに切り替えます。
3. /nsconfig directoryにすでにntp.confファイルが含まれている場合を除き、/etc/ntp.confファイルを/nsconfig/ntp.confにコピーします。
4. 追加する NTP サーバごとに、次の 2 行を/nsconfig/ntp.conf ファイルに追加する必要があります。

```
server <IP address for NTP server> iburst
```

```
restrict <IP address for NTP server> mask <netmask> nomodify notrap nopeer noquery
```

```
1 > Note:
2 >
3 > For security reasons, there should be a corresponding restrict entry
     for each server entry.
```



```
4
5  Example
6
7  In the following example, an administrator has inserted # characters to
       “comment out”  an existing NTP entry, and then added an entry:
8
9  `#server 1.2.3.4 iburst`
10
11 `#restrict 1.2.3.4 mask 55.255.255.255 nomodify notrap nopeer noquery`
12
13 `server 10.102.29.160 iburst`
14
15 `restrict 10.102.29.160 mask 255.255.255.255 nomodify notrap nopeer
      noquery`
```

1. /nsconfigディレクトリにrc.netscalerという名前のファイルがない場合は、ファイルを作成します。
2. 次のエントリを/nsconfig/rc.netscaler: /bin/sh /etc/ntpd_ctl full_startに追加

   このエントリは、ntpd サービスを開始し、ntp.conf ファイルをチェックし、メッセージを /var/log ディレクトリに記録します。

   このプロセスは、Citrix ADC が再起動されるたびに実行されます。
3. Citrix ADC アプライアンスを再起動して、クロック同期を有効にします。または、アプライアンスを再起動せずに時刻同期プロセスを開始するには、シェルプロンプトで次のコマンドを入力します。

- rm /etc/ntp.conf
- ln -s /nsconfig/ntp.conf /etc/ntp.conf
- /bin/sh /etc/ntpd_ctl full_start

アイドル状態のクライアント接続のセッションタイムアウトを構成する方法

セッションタイムアウト間隔は、セッション（GUI、CLI、または API）が使用されていないときにアクティブのままになる時間を制限するために提供されます。Citrix ADC の場合、システムセッションタイムアウトは次のレベルで構成できます。

- ユーザーレベルのタイムアウト。特定のユーザーに適用されます。

インターフェースタイプ	タイムアウト設定
GUI	[システム] > [ユーザ管理] > [ユーザ] に移動し、ユーザを選択して、ユーザのタイムアウト設定を編集します。

インターフェースタイプ	タイムアウト設定
CLI	コマンドプロンプトで、次のコマンドを入力します。 `set system user <name> -timeout <secs>`

- ユーザーグループレベルのタイムアウト。グループ内のすべてのユーザーに適用されます。

インターフェースタイプ	タイムアウト設定
GUI	[システム] > [ユーザー管理] > [グループ] に移動し、グループを選択して、グループのタイムアウト設定を編集します。
CLI	コマンドプロンプトで、次のコマンドを入力します。 `set system group <groupName> -timeout <secs>`

- グローバルシステムタイムアウト。タイムアウトが設定されていないグループのすべてのユーザおよびユーザに適用されます。

インターフェースタイプ	タイムアウト設定
GUI	[システム] > [設定] に移動し、[グローバルシステム設定の変更] をクリックし、必要に応じてタイムアウト値を更新します。
CLI	コマンドプロンプトで、次のコマンドを入力します。 `set system parameter -timeout <secs>`

```
The timeout value specified for a user has the highest priority. If
    timeout is not configured for the user, the timeout configured for a
     member group is considered. If timeout is not specified for a group
     (or the user does not belong to a group), the globally configured
    timeout value is considered. If timeout is not configured at any
    level, the default value of 900 seconds is set as the system session
     timeout.

Additionally, you can specify timeout durations for each of the
```



```
      interfaces you are accessing. However, the timeout value specified
      for a specific interface is restricted to the timeout value
      configured for the user that is accessing the interface. For example
      , let us consider an user "publicadmin" who has a timeout value of
      20 minutes. Now, when accessing an interface, the user must specify
      a timeout value that is within 20 minutes.
4
5 > **Note:**
6 >
7 > You can choose to keep a check on the minimum and maximum timeout
      values by specifying the timeout as restricted (in CLI by specifying
       the *restrictedTimeout* parameter). This parameter is provided to
      account for previous Citrix ADC versions where the timeout value was
       not restricted.
```

- 有効の場合、設定可能な最小タイムアウト値は 5 分（300 秒）で、最大値は 1 日（86400 秒）です。タイムアウト値がすでに 1 日より大きい値に設定されている場合、このパラメータを有効にすると、変更を求めるプロンプトが表示されます。値を変更しない場合、タイムアウト値は次回の再起動時にデフォルトのタイムアウト期間である 15 分（900 秒）に自動的に再構成されます。設定したタイムアウト値が 5 分未満の場合も同じことが起こります。
- 無効にすると、設定されたタイムアウト期間が考慮されます。
- 各インターフェイスのタイムアウト時間**:**

インターフェースタイプ	タイムアウト設定
CLI	次のコマンドを使用して、コマンドプロンプトでタイムアウト値を指定します。 `set cli mode -timeout <secs>`
API	ログインペイロードにタイムアウト値を指定します。

システムの日付と時刻を設定して時計をタイムサーバーと同期させる方法

システムの日付と時刻を変更するには、基盤となる FreeBSD OS へのシェルインターフェースを使用する必要があります。ただし、システムの日付と時刻を表示するには、コマンドラインインターフェイスまたは GUI を使用できます。

コマンドインターフェイスを使用してシステムの日付と時刻を表示する

コマンドプロンプトで入力します。

```
show ns config
```


GUI を使用してシステムの日付と時刻を表示する

[システム] に移動し、[システム情報] タブを選択してシステム日付を表示します。

内部サービス用の **HTTP** および **HTTPS** 管理ポートの設定方法

Citrix ADC アプライアンスのシングル IP モードの展開では、単一の IP アドレスが NSIP、SNIP、および VIP アドレスとして使用されます。この単一の IP アドレスは、異なるポート番号を使用して、NSIP、SNIP、および VIP アドレスとして機能します。

ポート番号 80 と 443 は、HTTP および HTTPS サービスの既知のポートです。以前は、Citrix ADC IP アドレス（NSIP）のポート 80 および 443 は、内部 HTTP および HTTPS 管理サービスの専用ポートでした。これらのポートは内部サービス用に予約されているため、VIP アドレスから HTTP および HTTPS データサービスを提供するために、これらの既知のポートを使用することはできません。VIP アドレスは、シングル IP モード展開の NSIP アドレスと同じアドレスを持っています。

この要件に対処するために、ポート 80 と 443 以外の（NSIP アドレスの）内部 HTTP および HTTPS 管理サービスのポートを構成できるようになりました。

Citrix ADC MPX、VPX、および CPX アプライアンスの内部 HTTP および HTTPS 管理サービスのデフォルトのポート番号を以下に示します。

- Citrix ADC MPX および VPX アプライアンス：80（HTTP）および 443（HTTPS）アプライアンス
- Citrix ADC CPX アプライアンス：9080（HTTP）および 9443（HTTPS）

コマンドインターフェイスを使用して **HTTP** および **HTTPS** 管理ポートを設定します

HTTP および HTTPS 管理サービスをサポートするために、Citrix ADC アプライアンスで HTTP および HTTPS ポートを任意の値に構成できます。ただし、デフォルトでは、Citrix ADC アプライアンスは HTTP および HTTPS 接続に 80 ポートと 443 ポートを使用します。

コマンドプロンプトで入力します。

```
set ns param -mgmtHttpPort<port>
```

例:

```
set ns param -mgmtHttpPort 2000
```

コマンドインターフェイスを使用して HTTPS ポートを設定するには

コマンドプロンプトで入力します。

```
set ns param -mgmtHttpsPort<port>
```

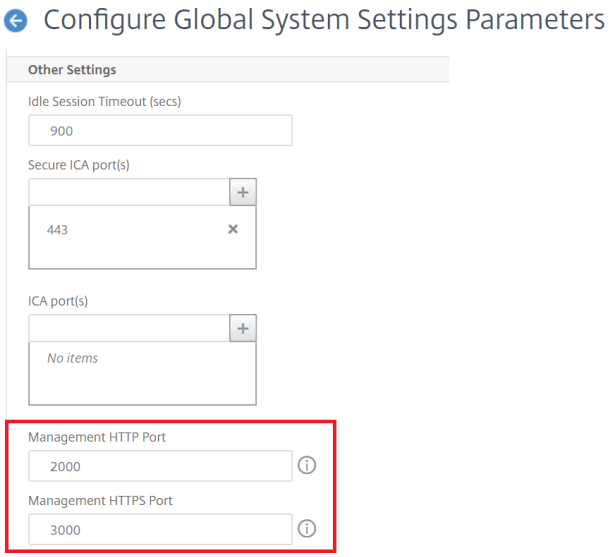
例:

```
set ns param -mgmtHttpsPort 3000
```


GUI を使用して **HTTP** および **HTTPS** 管理ポートを設定します

HTTP および HTTPS ポート値を設定するには、以下の手順に従います。

1. [システム] > [設定] **> [**グローバルシステム設定の変更] に移動します。
2. グローバルシステム設定の構成パラメータページの [その他の設定] セクションで、次のパラメータを設定します。
   a) 管理 HTTP ポート。ポート値を 2000 に設定します。デフォルト = 80、最小 = 1、最大 = 65534。
   b) 管理 HTTPS ポート。ポート値を 3000 に設定します。デフォルト = 443、最小 = 1、最大 = 65534。



データ処理と監視のために追加の管理 **CPU** を割り当てる方法

Citrix ADC MPX アプライアンスの構成と監視のパフォーマンスを向上させる必要がある場合は、アプライアンスのパケットエンジンプールから追加の管理 CPU を割り当てることができます。この機能は、特定の Citrix ADC MPX モデルと、Citrix ADC SDX アプライアンスで実行される VPX インスタンスを除くすべての VPX モデルでサポートされています。これは、統計システム CPU および stat システムコマンドの出力に影響します。

サポートされている Citrix ADC MPX モデル:

- 25xxx
- 22xxx
- 14xxx
- 115xx
- 15xxx
- 26xxx

> 注:
>
> 20 コアを超える Citrix ADC MPX 26xxx モデルの場合、必須の追加管理 CPU 機能がデフォルトで有効になっ

> ています。Citrix ADC VPX モデルの場合、この機能を有効にするには、少なくとも 12 の vCPU をサポートするライセンスが必要です。

コマンドインターフェイスを使用して追加の管理 **CPU** を割り当てる

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `enable extramgmtcpu`
- `disable extramgmtcpu`

> 注:
>
> この機能を有効または無効にすると、Citrix ADC アプライアンスは、変更を有効にするためにアプライアンスを再起動するように警告を表示します。

追加の管理 CPU の設定済みおよび有効な状態を表示します。

コマンドプロンプトで入力します。

```
1 `show extramgmtcpu`
```

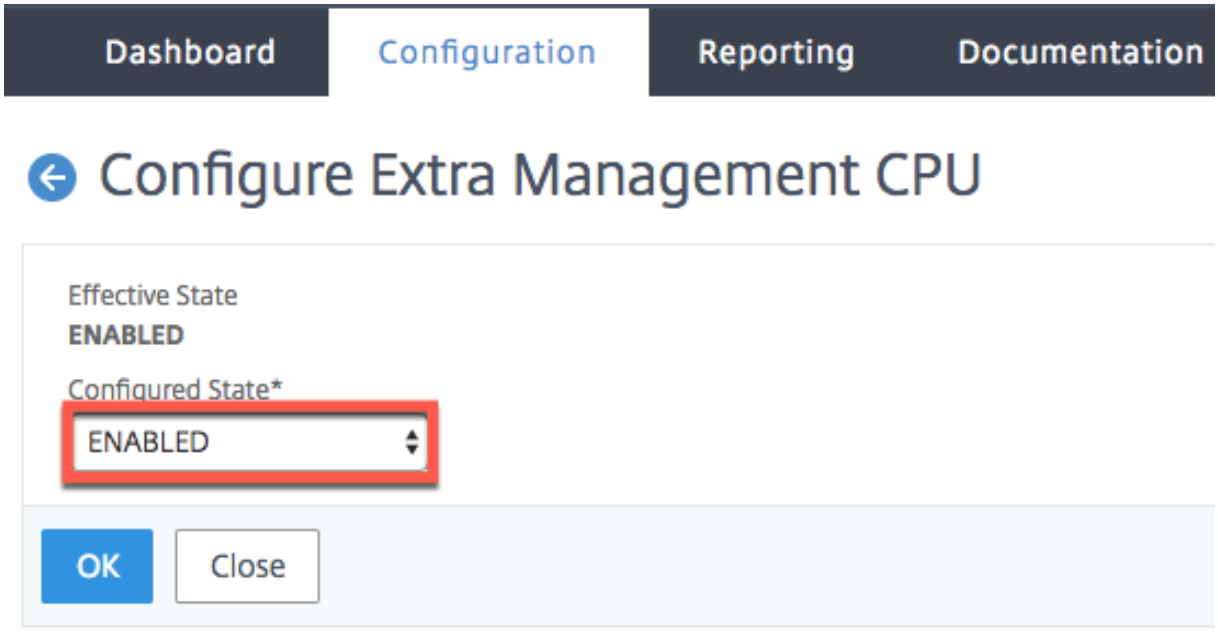
例**:**

```
> show extramgmtcpu ConfiguredState: ENABLED EffectiveState: ENABLED
```

> 注:
>
> この例では、アプライアンスを再起動する前に show コマンドを入力します。

GUI を使用して追加の管理 **CPU** を割り当てる

GUI を使用して追加の管理 CPU を割り当てるには、[システム] > [設定] に移動し、[追加管理 **CPU** の設定] をクリックします。[構成済みの状態] ドロップダウンメニューから [有効] を選択し、[**OK**] を選択します。



CPU 使用率を確認するには、[システム] **>** [設定] **>** [ダッシュボード] に移動します。

NITRO API を使用して追加の管理 **CPU** を構成する

次の NITRO 方式と形式を使用して、追加の管理 CPU を有効化、無効化、および表示します。

追加の管理 **CPU** を有効にするには、次の手順を実行します。

```
HTTP Method: POST

URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=enable

Payload: { ”systemextramgmtcpu”:{ } }

curl -v -X POST -H ”Content-Type: application/json”-u nsroot:nsroot http:
//10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=enable -d '{ ”
systemextramgmtcpu”:{ } } '
```

追加の管理 CPU を無効にするには

```
HTTP Method: POST

URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=disable

Payload: { ”systemextramgmtcpu”:{ } }

curl -v -X POST -H ”Content-Type: application/json”-u nsroot:nsroot http:
//10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=disable -d '{ ”
systemextramgmtcpu”:{ } } '
```

追加の管理 CPU を表示するには


```
HTTP Method: GET

URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu
```

例:

```
curl -v -X GET -H "Content-Type: application/json"-u nsroot:nsroot http://
10.102.201.92/nitro/v1/config/systemextramgmtcpu
```

管理 **CPU** の追加前と後の統計とモニタリング

次の例は、追加の管理 CPU を追加する前と後の stat system CPU コマンドと stat system コマンドの出力の違いを示しています。

```
stat system cpu
```

このコマンドは、CPU の統計情報を表示します。

次に、サポートされているモデルのいずれかで追加の管理 CPU を追加する前の出力例を示します。

例

```
```
> stat system cpu

CPU statistics

ID   Usage

8   1

7   1

11   2

1   1

6   1

9   1

3   1

5   1

4   1
```

 
```

```
25
26 10              1
27
28 2               1
29 <!--NeedCopy--> ```
```

次に、同じ MPX アプライアンスで管理用 CPU を追加した後の出力を示します。

```
1  ```
2  > stat system cpu
3
4  CPU statistics
5
6  ID          Usage
7
8  9               1
9
10 7               1
11
12 5               1
13
14 8               1
15
16 11              2
17
18 10              1
19
20 6               1
21
22 4               1
23
24 3               1
25
26 2               1
27 <!--NeedCopy--> ```
```

`stat system`

このコマンドは CPU 使用率を表示します。次の例では、サポートされているモデルのいずれかで追加の管理 CPU を追加する前の出力は次のようになります。

Mgmt Additional-CPU usage (%) 0.00

例


```
```
> stat system

Citrix ADC Executive View

System Information:

Up since   Wed Oct 11 11:17:54 2017

/flash Used (%)   0

Packet CPU usage (%)   1.30

Management CPU usage (%)   4.00

Mgmt CPU0 usage (%)   4.00

Mgmt Additional-CPU usage (%)   0.00

Memory usage (MB)   2167

InUse Memory (%)   5.76

/var Used (%)   0
<!--NeedCopy--> ```
```

次の例では、同じ MPX アプライアンスで管理用 CPU を追加した後の出力は次のようになります。

Mgmt Additional-CPU usage (%) 0.80

```
```    > stat system

Citrix ADC Executive View

System Information:

Up since        Wed Oct 11 11:55:56 2017

/flash Used (%)                          0

Packet CPU usage (%)                 1.20

```



```
Management CPU usage (%)             5.70

Mgmt CPU0 usage (%)                 10.60

Mgmt Additional-CPU  usage (%)       0.80

Memory usage (MB)                    1970

InUse Memory (%)                     5.75

/var Used (%)                           0

<!--NeedCopy--> ```
```

失われた構成を回復するためにアプライアンスをバックアップおよび復元する方法

アプライアンスが破損した場合やアップグレードが必要な場合は、システム設定をバックアップできます。バックアップ手順は、Citrix CLI または GUI インターフェイスのいずれかを使用して実行されます。アプライアンスでは、外部ソースからバックアップファイルをインポートすることもできます。ただし、これは GUI インターフェイスを介してのみ実行でき、CLI インターフェイスによるサポートはありません。

確認事項

アプライアンスをバックアップおよび復元するときは、次の点を覚えておく必要があります。

- 新しいプラットフォームでのネットワーク構成のサポートが必要です。
- 新しいプラットフォームビルドは、バックアップファイルまたはそれ以降のバージョンと同じである必要があります。

Citrix ADC アプライアンスをバックアップする

データとバックアップの要件に応じて、「基本」バックアップまたは「フル」バックアップを作成できます。

- 基本バックアップ。この種類のバックアップは、常に変化するファイルをバックアップする場合に行うことができます。バックアップできるファイルを次の表に示します。

基本的なバックアップの詳細については、「表 」のトピックを参照してください。

- フルバックアップ基本バックアップでバックアップされるファイルに加えて、完全バックアップではファイルの更新頻度は低くなります。「フル」バックアップオプションを使用したときにバックアップされるファイルは次のとおりです。

ディレクトリ	サブディレクトリまたはファイル
nsconfig	ssl*, license*, fips*
/var/	netscaler/ssl/*、 wi/java_home/jre/lib/security/cacerts/*、 wi/java_home/lib/security/cacerts/*

バックアップされたデータは、圧縮された TAR ファイルとして/var/ns_sys_backup/ディレクトリに格納されます。ディスク領域が利用できないことによる問題を回避するために、このディレクトリには最大 50 個のバックアップファイルを保存できます。rm system backupコマンドを使用して、既存のバックアップファイルを削除し、さらにバックアップを作成できます。

> 注:
>
> バックアップ操作が進行中の場合は、設定に影響を与えるコマンドを実行しないでください。
>
> バックアップが必要なファイルが使用できない場合、そのファイルはスキップされます。

コマンドインターフェイスを使用して **Citrix ADC** アプライアンスをバックアップする

Citrix ADC コマンドインターフェイスを使用して Citrix ADC アプライアンスをバックアップするには、以下の手順に従います。

コマンドプロンプトで、次の操作を行います。

1. Citrix ADC 構成を保存します。

```
save ns config
```

1. バックアップファイルを作成します。

```
create system backup [<fileName>] -level <basic ¦ full> -comment <string>
```

> 注:
>
> ファイル名が指定されていない場合、アプライアンスは次の命名規則で TAR ファイルを作成します。backup_<level>_<nsip_address>_<date-timestamp>.tgz。

例**:** バックアップファイルのデフォルトの命名規則を使用して、完全なアプライアンスをバックアップする場合。

```
> create system backup -level full
```

1. バックアップファイルが作成されたことを確認します。

```
show system backup
```

fileName パラメータを使用すると、特定のバックアップファイルのプロパティを表示できます。

コマンドインターフェイスを使用して **Citrix ADC** アプライアンスを復元する

> 重要:
>
> バックアップファイルの名前を変更または変更すると、アプライアンスを正常に復元できません。

アプライアンスを復元すると、復元操作によって/var/ns_sys_backup/ディレクトリからバックアップファイルが解凍されます。ファイルが解凍されると、ファイルはそれぞれのディレクトリにコピーされます。

コマンドインターフェイスを使用して、ローカルバックアップファイルから **Citrix ADC** を復元します

> 注:
>
> 以前の構成を復元する前に、現在の構成をバックアップ Citrix。ただし、restore コマンドで現在の設定のバックアップを自動的に作成しない場合は、 - skipBackup パラメータを使用します。

コマンドプロンプトで、次の操作を行います。

1. アプライアンスで使用可能なバックアップファイルのリストを取得します。

   ```
   show system backup
   ```

2. バックアップファイルの 1 つを指定して、アプライアンスを復元します。

   ```
   restore system backup <filename> [-skipBackup]
   ```

   例**:** アプライアンスの完全バックアップを使用して復元するには

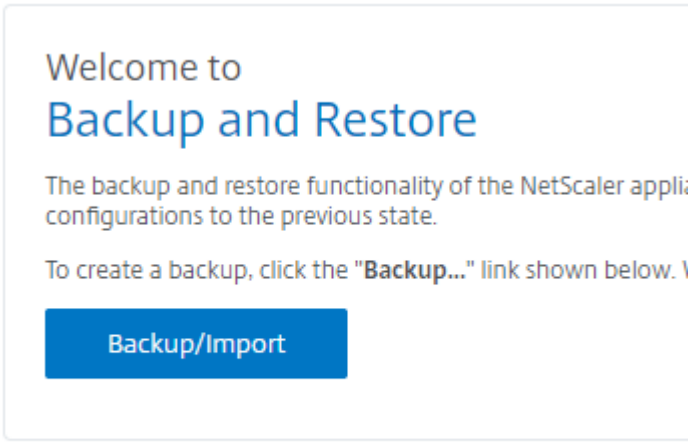
   ```
   > restore system backup backup_full_<nsip_address>_<date-timestamp>.tgz
   ```

3. アプライアンスを再起動します。

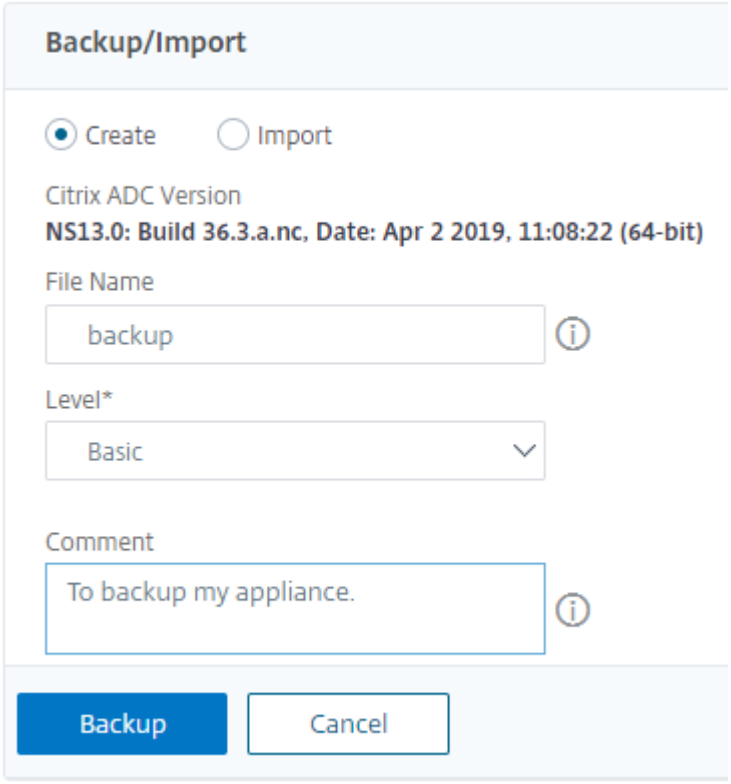
   ```
   reboot
   ```

GUI を使用して **Citrix ADC** アプライアンスをバックアップおよび復元する

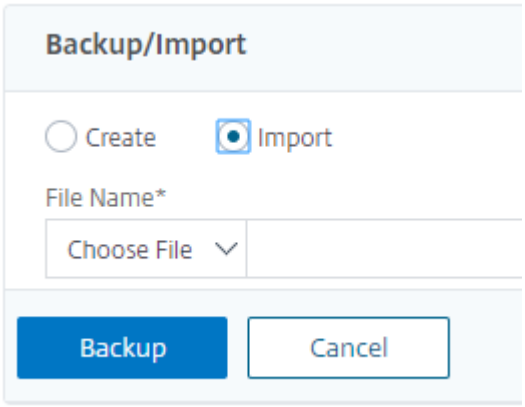
1. [システム] > [バックアップと復元] に移動します。



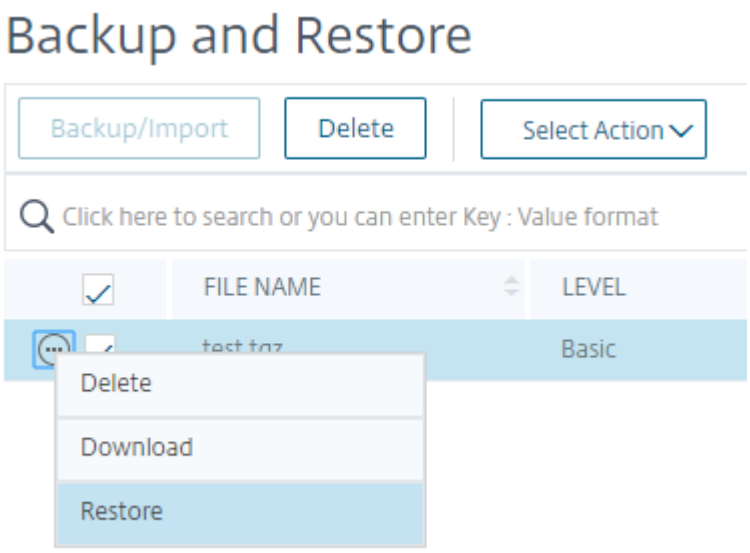
2. [バックアップ/インポート] をクリックして、プロセスを開始します。
3. [バックアップ/インポート] ページで、[作成] を選択し、次のパラメータを設定します。
   a) ファイル名。アプライアンスのバックアップファイルの名前。
   b) [レベル]。バックアップレベルとして「基本」または「フル」を選択します。
   c) [コメント]。バックアップの簡単な説明を入力します。
4. [バックアップ] をクリックします。



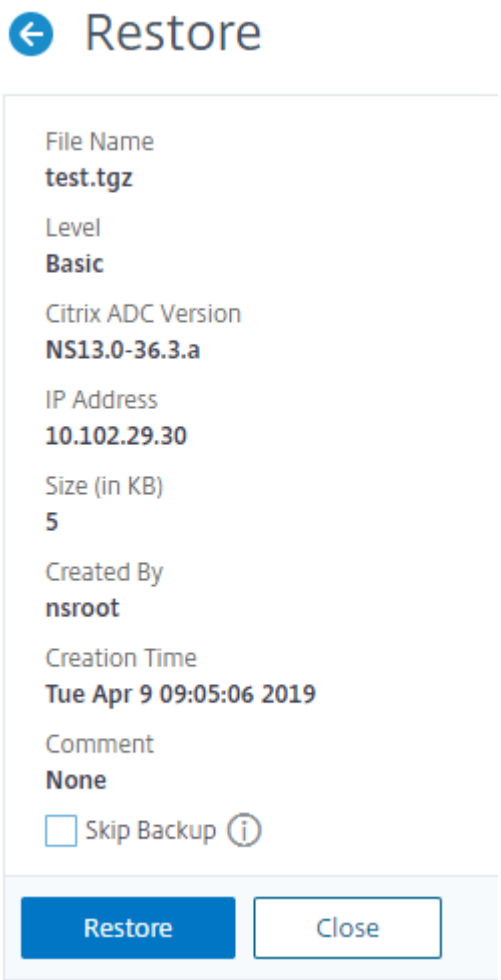
5. バックアップをインポートする場合は、[インポート] を選択する必要があります。



6. バックアップが完了したら、ファイルを選択して [ダウンロード] をクリックします。
7. 復元するには、バックアップファイルを選択して [復元] をクリックします。



8. [復元] ページで、バックアップファイルの詳細を確認し、[復元] をクリックします。



9. 復元した後、アプライアンスを再起動する必要があります。

Citrix ADC インスタンスのバックアップと復元方法の詳細については、「Citrix ADM を使用したバックアップと復元」トピックを参照してください。

SDX アプライアンスのバックアップと復元方法の詳細については、SDX アプライアンスのバックアップと復元を参照してください。

システムバックアップで実行される操作の詳細については、「システムバックアップ」トピックを参照してください。

アプライアンスの問題を解決するためのテクニカルサポートバンドルを生成する方法

Citrix ADC アプライアンスに関する問題の分析と解決については、アプライアンスでテクニカルサポートバンドルを生成し、そのバンドルを Citrix テクニカルサポートに送信できます。Citrix ADC テクニカルサポートバンドルは、システム構成データと統計情報の圧縮された tar アーカイブです。バンドルを生成する Citrix ADC アプライアンスから次のデータを収集します。

- 設定ファイル。/flash/nsconfig ディレクトリ内のすべてのファイル。
- **Newnslog** ファイル。現在実行中の newnslog といくつかの以前のファイル。アーカイブファイルのサイズを最小化するために、`newnslog`コレクションは 500 MB、6 ファイル、7 日のいずれか早い方に制限されます。古いデータが必要な場合は、手動で収集する必要があります。
- ログファイル。/var/log/messages 、*/var/log/ns.log* 内のファイル、および /var/log および /var/nslog の下のその他のファイル。
- アプリケーションのコアファイル。先週以内に /var/core ディレクトリに作成されたファイル (存在する場合)。
- いくつかの **CLI** の **show** コマンドの出力。
- いくつかの **CLI** 統計コマンドの出力。
- **BSD** シェルコマンドの出力。

1 つのコマンドを使用してテクニカルサポートバンドルを生成し、Citrix テクニカルサポートサーバーに安全にアップロードできます。アップロードするには、Citrix 資格情報を指定する必要があります。バンドルを生成するときに、Citrix テクニカルサポートによって割り当てられたケースまたはサービスリクエスト番号を指定できます。テクニカルサポートバンドルをすでに生成している場合は、フルパスでファイル名を指定することで、既存のアーカイブファイルを Citrix テクニカルサポートサーバーにアップロードできます。

テクニカルサポートバンドルは、Citrix ADC アプライアンスの次の場所にアーカイブに保存されます。

`/var/tmp/support/support.tgz`

このパスは、簡単にアクセスできるように最新のコレクターへのシンボリックリンクです。完全なファイル名は、デプロイメントトポロジによって異なりますが、一般的に次のような形式になります。

`collector_<P/S>_<NS IP>_<DateTime>.tgz.`

Citrix ADC アプライアンスに直接インターネット接続がない場合は、プロキシサーバーを使用して、テクニカルサポートバンドルを Citrix テクニカルサポートサーバーに直接アップロードできます。プロキシ文字列の基本的な形式は次のとおりです。


```
proxy_IP:<proxy_port>
```

プロキシサーバーが認証を必要とする場合、形式は次のとおりです。

```
username:password@proxsy_IP:<proxy_port>
```

> 注:
>
> 高可用性ペアの Citrix ADC アプライアンスの場合は、2 つのノードのそれぞれでテクニカルサポートバンドルを生成する必要があります。
>
> クラスタセットアップの Citrix ADC アプライアンスの場合、各ノードで個別にテクニカルサポートバンドルを生成することも、クラスタ IP アドレスを使用してすべてのノードに対してより小さな省略アーカイブを生成することもできます。
>
> Citrix ADC 管理パーティションの場合は、デフォルトの管理パーティションからテクニカルサポートバンドルを生成する必要があります。特定のパーティションのテクニカルサポートバンドルを取得するには、テクニカルサポートバンドルを生成するパーティションの名前を指定する必要があります。パーティションの名前を指定しない場合、データはすべての管理パーティションから収集されます。

コマンドインターフェイスを使用して **Citrix ADC** テクニカルサポートバンドルを生成する

コマンドプロンプトで入力します。

```
show techsupport [-scope <scope> <partitionName>] [-upload [-proxy <string
>] [-casenumber <string>] [-file <string>] [-description <string>] [-
userName <string> -password ]]
```

シニアいいえ	タスク	コマンド
1	テクニカルサポートバンドルを生成し、Citrix テクニカルサポートサーバーにアップロードします。	show techsupport –upload –userName account1 –password xxxxxxx
2	テクニカルサポートバンドルを生成し、プロキシサーバーを介して Citrix テクニカルサポートサーバーにアップロードします。	show techsupport –upload –proxy 1.1.1.1:80 –userName account1 –password xxxxxxx
3	既存のテクニカルサポートバンドルを Citrix テクニカルサポートサーバーにアップロードします。	show techsupport –upload -file,/var/tmp/support/collector_P_10.102.29.1 –userName account1 –password xxxxxxx

シニアいいえ	タスク	コマンド
4	クラスタ設定のすべてのノードに対して、小さくて省略されたアーカイブを生成します。クラスター IP アドレスを使用してこのコマンドを実行します。	show techsupport –scope CLUSTER
5	管理者パーティションに固有のテクニカルサポートバンドルを生成します。このコマンドをデフォルトの admin パーティションで実行します。	show techsupport –scope PARTITION partition1

洞察分析のために **SDX** および **VPX** アプライアンスからテクニカルサポートバンドルを収集する方法

Citrix ADC アプライアンスには、ログファイルを収集するメカニズムが組み込まれています。ログファイルは、分析のために Citrix Insight Services に送信されます。

> 注:
>
> すべての手順は、ソフトウェアリリース 9.2 以降に適用されます。

Citrix ADC MPX および **VPX** アプライアンスからテクニカルサポートバンドルをダウンロードする

Citrix ADC GUI を使用してコレクタファイルを実行するには、次の手順を完了する必要があります。

> 注**:**
>
> この手順は、ソフトウェアリリース 9.2 以降に適用されます。

1. [システム] > [診断] に移動します。
2. [テクニカルサポートツール] セクションで、[サポートファイルを生成] リンクをクリックします。
3. [テクニカルサポート（**Tech** Support）] ページで、次のパラメータを設定します。
    a) スコープ。1 つ以上のノードからデータを収集します。
    b) パーティション。パーティションの名前。
    c) Citrix テクニカルサポートのロードオプション。プロキシサーバ、サービスケース番号、コレクタアーカイブファイル名、テクニカルサポートバンドルをアップロードするためのアーカイブファイルの簡単な説明など、すべてのオプションを設定します。
    d) Citrix アカウント。Citrix 資格情報を入力します。
4. ［実行］をクリックします。

5. テクニカルサポートバンドルが生成されます。
6. [はい] をクリックして、テクニカルサポートバンドルをローカルデスクトップにダウンロードします。

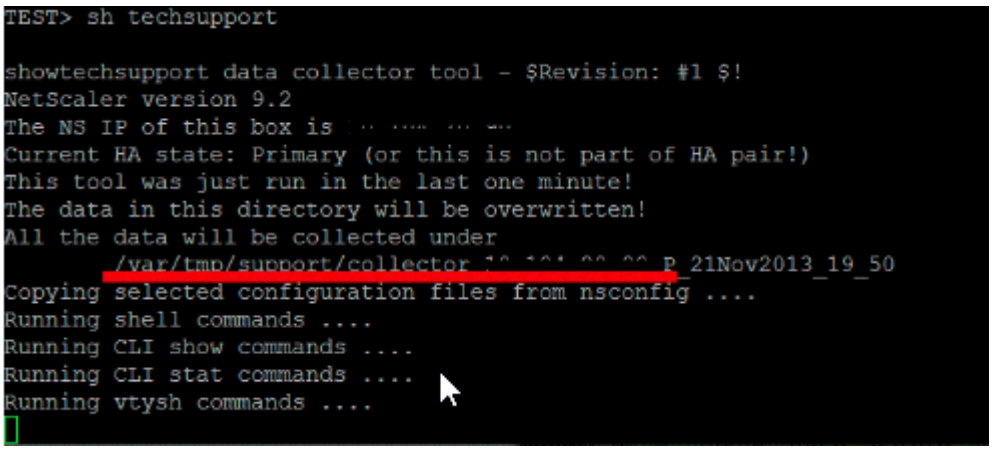
コマンドインターフェイスを使用してテクニカルサポートバンドルを入手する

1. WinSCPなどのセキュア FTP（SFTP）またはセキュアコピー（SCP）ユーティリティを使用してアプライアンスからファイルをダウンロードし、分析のために Citrix Insight Services にアップロードします。

> 注:
>
> 9.0 より前の Citrix ADC ソフトウェアリリースでは、コレクタースクリプトを個別にダウンロードして実行する必要があります。

```
> show techsupport -scope CLUSTER
```

1. これにより、クラスター内のすべてのノードから表示テクニカルサポート情報が収集され、ファイルが 1 つのアーカイブに圧縮されます。
2. アプライアンスがコレクタアーカイブを生成すると、次のスクリーンショットのようにファイルの場所が表示されます。



ファイルは/var/tmp/supportに保存され、Citrix ADC アプライアンスにログインし、シェルプロンプトから次のコマンドを実行することで確認できます。

```
root@NS## cd /var/tmp/support/
root@NS## ls -l
```

GUI を使用して **Citrix ADCSDX** から診断バンドルを取得します

1. Citrix SDX GUI を開きます。
2. [診断] ノードを展開します。
3. [テクニカルサポート] ノードを選択します。
4. [テクニカルサポートファイルを生成] をクリックします。

5. ドロップダウンメニューから [アプライアンス (インスタンスを含む)] を選択します。
6. [追加] をクリックします。
7. 追加するインスタンスを 1 つ以上選択します。
8. [**OK**] をクリックします。プロセスが完了するのを待ちます。
9. 生成されたバンドル名を選択し、[ダウンロード] をクリックします。
10. バンドルファイルを Citrix Insight Servicesアップロードします。

その他のリソース

ビデオを見る

別のトピックを読む

コマンドリファレンスドキュメント

システムユーザー認証と承認

October 7, 2021

Citrix ADC ユーザーの認証と承認を構成するには、最初に Citrix ADC アプライアンスにアクセスできるユーザーを定義してから、これらのユーザーをグループに編成する必要があります。ユーザーとグループを構成した後、コマンドポリシーを構成してアクセスの種類を定義し、ポリシーをユーザーに割り当てる必要があります。and/or グループ。

ユーザー、グループ、およびコマンドポリシーを構成するには、管理者としてログオンする必要があります。デフォルトの Citrix ADC 管理者ユーザー名は *nsroot* です。デフォルトの管理者としてログオンした後、nsroot アカウントのパスワードを変更する必要があります。パスワードを変更すると、そのユーザーのアカウントを作成するまで、ユーザーは Citrix ADC アプライアンスにアクセスできなくなります。デフォルトから変更した後で管理者パスワードを忘れた場合は、nsroot にリセットできます。

> 注:
>
> - ローカルユーザーは、外部認証サーバーが構成されている場合でも、Citrix ADC に対して認証できます。これを制限するには、set システムパラメータコマンドの localAuth パラメータを無効にします。
> - セキュリティを強化するには、nsroot パスワードを変更することをお勧めします。パスワードを頻繁に変更することをお勧めします。nsroot パスワードを変更する方法については、「デフォルトの管理者 (nsroot) パスワードのリセット 」トピックを参照してください。

ユーザー、ユーザーグループ、およびコマンドポリシー

October 13, 2021

最初にアカウントを持つユーザーを定義してから、すべてのユーザーをグループに編成する必要があります。コマンドポリシーを作成するか、組み込みのコマンドポリシーを使用して、コマンドへのユーザーアクセスを規制できます。

> 注:
>
> トラフィック管理の Citrix ADC 認証および承認セットアップの一部としてユーザーおよびユーザーグループを構成する方法の詳細については、「ユーザーとグループの構成」トピックを参照してください。

また、ユーザーのコマンドラインプロンプトをカスタマイズすることもできます。プロンプトは、ユーザーの構成、ユーザーグループ構成、およびグローバルシステム構成設定で定義できます。ユーザに対して表示されるプロンプトは、次の優先順位です。

1. ユーザーの構成で定義されているプロンプトを表示します。
2. ユーザーのグループのグループ構成で定義されているプロンプトを表示します。
3. システムグローバル設定に定義されているプロンプトを表示します。

システムユーザの非アクティブ CLI セッションのタイムアウト値を指定できるようになりました。ユーザーの CLI セッションがタイムアウト値を超えた時間アイドル状態である場合、Citrix ADC アプライアンスは接続を終了します。タイムアウトは、ユーザーの構成、ユーザーグループ構成、またはグローバルシステム構成設定で定義できます。ユーザの非アクティブ CLI セッションのタイムアウトは、次の優先順位によって決まります。

1. [ユーザー設定]:
2. ユーザーのグループのグループ構成
3. グローバルシステム構成設定。

Citrix ADC ルート管理者は、システムユーザーの最大同時セッション制限を設定できます。制限を制限することで、開いている接続の数を減らし、サーバーのパフォーマンスを向上させることができます。CLI カウントが設定された制限内であれば、同時ユーザは GUI に何回でもログインできます。ただし、CLI セッションの数が設定された制限に達すると、ユーザは GUI にログオンできなくなります。たとえば、同時セッションの数が 20 に設定されている場合、同時ユーザーは 19 の CLI セッションにログオンできます。ただし、ユーザーが 20th CLI セッションにログオンしている場合、GUI、CLI、または NITRO にログオンしようとすると、エラーメッセージが表示されます（(エラー：CFE への接続制限を超えました)。

> 注:
>
> デフォルトでは、同時セッション数は 20 に設定され、同時セッションの最大数は 40 に設定されています。

ユーザーアカウントの構成

ユーザーアカウントを構成するには、ユーザー名とパスワードを指定するだけです。パスワードの変更やユーザーアカウントの削除はいつでも行えます。

> 注:
>
> パスワードのすべての文字が受け入れられるわけではありません。ただし、引用符で囲んだ文字を入力すると機能します。

> また、文字列は最大長の 127 文字を超えてはなりません。

コマンドラインインターフェイスを使用してユーザーアカウントを作成するには

コマンドプロンプトで次のコマンドを入力して、ユーザーアカウントを作成し、構成を確認します。

- `add system user <username> [-externalAuth ( ENABLED | DISABLED )] [-promptString <string>] [-timeout \<secs>] [-logging ( ENABLED | DISABLED )] [-maxsession <positive_integer>]`
- `show system user <userName>`

外部ユーザーは、「logging」パラメーターを構成して、Web ロギングまたは監査ロギングメカニズムを使用して外部ログを収集できます。パラメータが有効になっている場合、監査クライアントは Citrix ADC アプライアンスで自身を認証してログを収集します。

例**:**

```
> add system user johnd -promptString user-%u-at-%T
```

```
1 Enter password:
2 Confirm password:
3 > show system user johnd
4 user name: john
5       Timeout:900 Timeout Inherited From: Global
6       External Authentication: ENABLED
7       Logging: DISABLED
8       Maximum Client Sessions: 20
9 <!--NeedCopy-->
```

パラメータの説明については、「認証および認可ユーザコマンドリファレンス」を参照してください。

Citrix ADC GUI を使用してユーザーアカウントを構成する

1. **System** > **User Administration** > **Users** に移動してユーザーを作成します。
2. 詳細ウィンドウで、**[**追加**]** をクリックしてシステムユーザーを作成します。
3. [**Create System Group**] ページで、次のパラメーターを設定します。
   a) ユーザー名：ユーザーグループの名前。
   b) CLI プロンプト。CLI インターフェイスアクセス用に設定するプロンプト。
   c) アイドルセッションのタイムアウト（秒）。セッションがタイムアウトして閉じる前に、ユーザーが非アクティブにできる時間を設定します。
   d) 最大セッション数。ユーザーが試行できるセッションの最大数を設定します。
   e) ロギング特権を有効にします。ユーザーのロギング特権を有効にします。

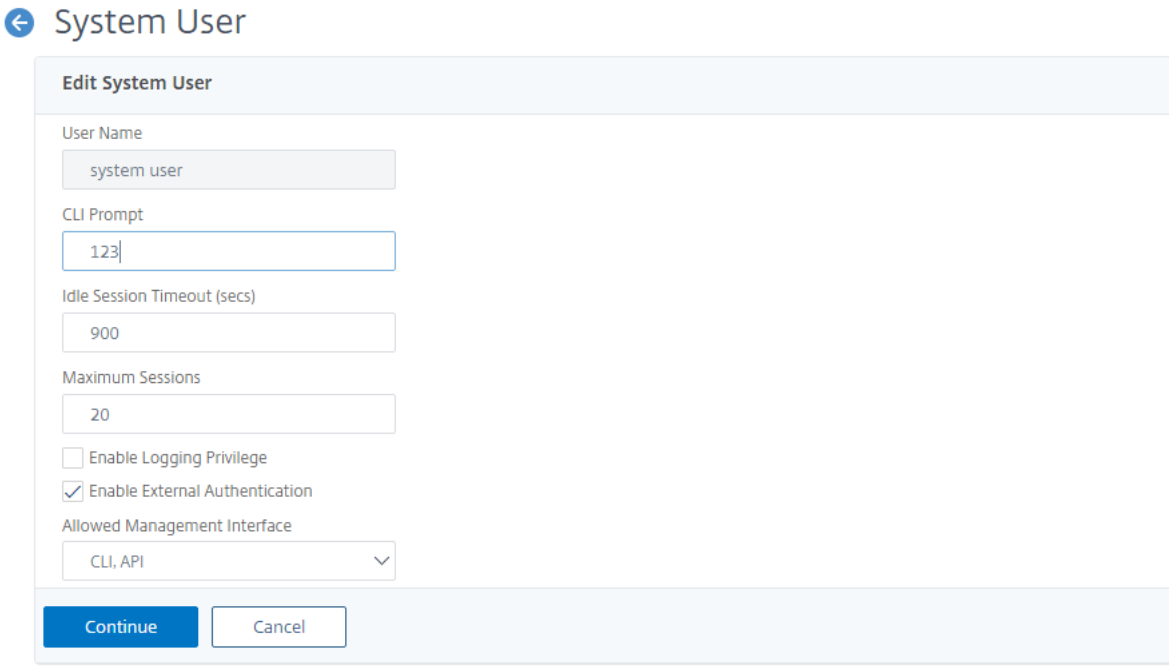
f) 外部認証を有効にする。ユーザーを認証するために外部認証サーバーを使用する場合は、オプションを選択します。

g) 許可された管理インターフェイス。ユーザーグループにアクセス許可が付与されている Citrix ADC インターフェイスを選択します。

h) コマンドポリシー。コマンドポリシーをユーザーグループにバインドします。

i) パーティション。パーティションをユーザーグループにバインドします。

4. **[作成]** して **[閉じる]** をクリックします。



ユーザーグループを構成する

ユーザーグループを構成したら、グループ内のすべてのユーザーに同じアクセス権を簡単に付与できます。グループを構成するには、グループを作成し、ユーザーをグループにバインドします。各ユーザーアカウントを複数のグループにバインドできます。ユーザーアカウントを複数のグループにバインドすると、コマンドポリシーをより柔軟に適用できます。

コマンドラインインターフェイスを使用してユーザーグループを作成するには

コマンド・プロンプトで次のコマンドを入力して、ユーザー・グループを作成し、構成を確認します。

- `add system group <groupName> [-promptString <string>] [-timeout <secs>]`
- `show system group <groupName>`

例:

```
> add system group Managers -promptString Group-Managers-at-%h
```


CLI を使用してユーザーアカウントをグループにバインドする

コマンドプロンプトで次のコマンドを入力して、ユーザーアカウントをグループにバインドし、構成を確認します。

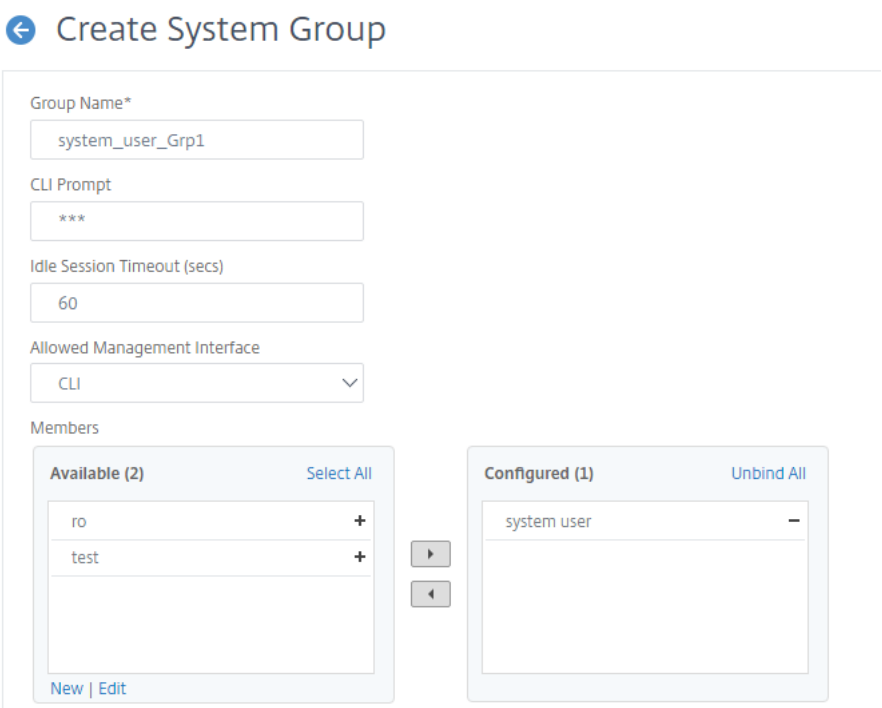
- `bind system group <groupName> -userName <userName>`
- `show system group <groupName>`

例**:**

```
> bind system group Managers -userName user1
```

Citrix ADC GUI を使用してユーザグループを構成する

1. **System** > **User Administration** > **Groups** に移動してユーザーグループを作成します。
2. 詳細ウィンドウで、システムユーザーグループを作成するには、**[**追加**]** をクリックします。
3. [**Create System Group**] ページで、次のパラメーターを設定します。
   a) グループ名。ユーザーグループの名前。
   b) CLI プロンプト。CLI インターフェイスアクセス用に設定するプロンプト。
   c) アイドルセッションのタイムアウト（秒）。セッションがタイムアウトして閉じる前に、ユーザーが非アクティブにできる時間を設定します。
   d) 許可された管理インターフェイス。ユーザーグループにアクセス許可が付与されている Citrix ADC インターフェイスを選択します。
   e) メンバー。グループにユーザーアカウントを追加します。
   f) コマンドポリシー。コマンドポリシーをユーザーグループにバインドします。
   g) パーティション。パーティションをユーザーグループにバインドします。
4. **[**作成**]** して **[**閉じる**]** をクリックします。



> 注:
>
> グループにメンバーを追加するには、[メンバー] セクションで [追加] をクリックします。「使用可能」リストからユーザーを選択し、「構成済み」リストに追加します。

コマンドポリシーの設定

コマンドポリシーは、ユーザーおよびユーザーグループの使用を許可するコマンド、コマンドグループ、仮想サーバー、およびその他のエンティティを規制します。

アプライアンスには一連の組み込みコマンドポリシーが用意されており、カスタムポリシーを設定できます。ポリシーを適用するには、ポリシーをユーザーまたはグループにバインドします。

ここでは、コマンドポリシーを定義および適用する際に留意すべき重要なポイントを示します。

- グローバルコマンドポリシーは作成できません。コマンドポリシーは、アプライアンス上のユーザーとグループに直接バインドする必要があります。
- コマンドポリシーが関連付けられていないユーザまたはグループは、デフォルト（DENY-ALL）コマンドポリシーに従うため、適切なコマンドポリシーがアカウントにバインドされるまで、構成コマンドを実行できません。
- すべてのユーザーは、自分が属するグループのポリシーを継承します。
- コマンドポリシーをユーザアカウントまたはグループアカウントにバインドするときは、コマンドポリシーに優先順位を割り当てる必要があります。これにより、複数の競合するポリシーが同じユーザーまたはグループに適用される場合に、アプライアンスが優先度を持つポリシーを判断できるようになります。
- 次のコマンドは、デフォルトですべてのユーザが使用でき、指定したコマンドの影響を受けません。

- ヘルプ、CLI 属性の表示、CLI プロンプトの設定、CLI プロンプトのクリア、CLI プロンプトの表示、エイリアス、エイリアスの解除、履歴、終了、whoami、設定、CLI モードの設定、CLI モードの設定解除、CLI モードの表示

次の表は、組み込みポリシーの説明です。

ポリシー名	許可内容
read-only	show ns runningConfig、show ns ns.conf、および Citrix ADC コマンドグループの show コマンドを除くすべての show コマンドへの読み取り専用アクセス。
operator	読み取り専用のアクセスと、サービスおよびサーバーを有効または無効にするコマンドへのアクセス。
network	フルアクセス。ただし set and unset SSL コマンドの設定と解除、show ns ns.conf、show ns runningConfig、show gslb runningConfig コマンドを除きます。
sysadmin	[Citrix ADC 12.0 以降に含まれる] Sysadmin はスーパーユーザーよりも低いが、アプライアンスで許可されるアクセス条件です。sysadmin ユーザーは、Citrix ADC シェルにアクセスできない、ユーザー構成を実行できない、パーティション構成を実行できない、sysadmin コマンドポリシーに記載されているその他の構成を除いて、すべての Citrix ADC 操作を実行できます。
superuser	フルアクセス。nsroot ユーザーと同じ権限。

カスタムコマンドポリシーの作成

よりカスタマイズされた式を維持するためのリソースを持つユーザーや、正規表現が提供する柔軟性を必要とするデプロイメントには、正規表現のサポートが提供されます。ほとんどのユーザーにとって、組み込みのコマンドポリシーで十分です。より多くのレベルの制御が必要であるが正規表現に慣れていないユーザーは、ポリシーの可読性を維持するために、このセクションで提供される例のような単純な式のみを使用することをお勧めします。

正規表現を使用してコマンドポリシーを作成する場合は、次の点に注意してください。

- コマンドポリシーの影響を受けるコマンドを定義するために正規表現を使用する場合は、コマンドを二重引用符で囲む必要があります。たとえば、show で始まるすべてのコマンドを含むコマンドポリシーを作成するには、次のように入力します。
- “^show .*$”

- rm で始まるすべてのコマンドを含むコマンドポリシーを作成するには、次のように入力します。
- “^rm .*$”
- コマンドポリシーで使用される正規表現では、大文字と小文字が区別されません。

次の表に、コマンドポリシーの正規表現の例を示します。

コマンド仕様	これらのコマンドに一致します。
“^rm\s+.*$”	すべての削除アクションは rm 文字列で始まり、その後にスペースと、コマンドグループ、コマンドオブジェクトタイプ、引数などのパラメーターが続くため、すべての削除アクション。
“^show\s+.*$”	すべての show コマンドは、すべての show アクションが show 文字列で始まり、その後にスペースと、コマンドグループ、コマンドオブジェクトタイプ、引数などのパラメーターが続くためです。
“^shell$”	シェルコマンドだけですが、コマンドグループ、コマンドオブジェクトタイプ、引数などの追加パラメータと組み合わせることはできません。
“^add\s+vserver\s+.*$”	すべてが仮想サーバーアクションを作成します。これは、仮想サーバーコマンドの追加と、それに続くスペース、およびコマンドグループ、コマンドオブジェクトタイプ、引数などのパラメーターで構成されます。
“^add\s+(lb\s+vserver)\s+.*”	すべてが lb 仮想サーバーアクションを作成します。これは、add lb 仮想サーバーコマンドとそれに続くスペース、およびコマンドグループ、コマンドオブジェクトタイプ、引数などのパラメーターで構成されます。

組み込みコマンドポリシーの詳細については、「組み込みコマンドポリシーテーブル」を参照してください。

コマンドラインインターフェイスを使用してコマンドポリシーを作成するには

コマンドプロンプトで次のコマンドを入力して、コマンドポリシーを作成し、構成を確認します。

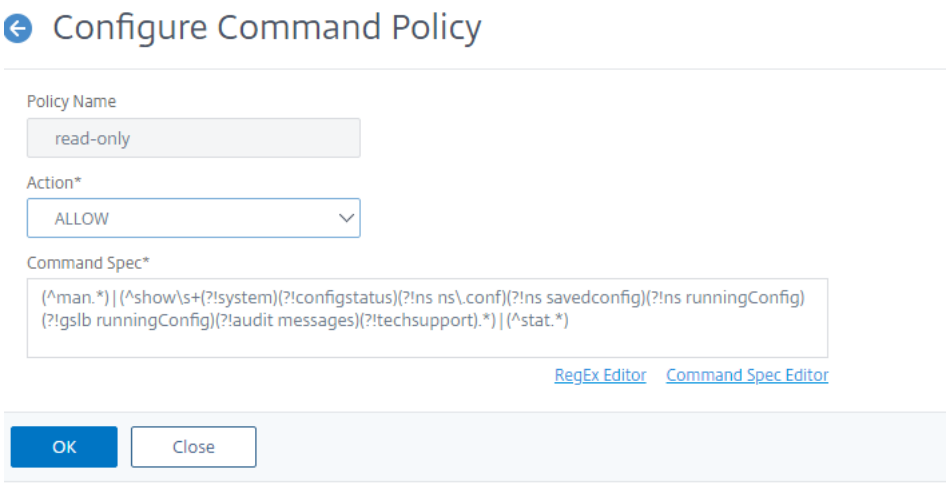
- `add system cmdPolicy <policyname> <action> <cmdspec>`
- `show system cmdPolicy <policyName>`

例:

```
add system cmdPolicy USER-POLICY ALLOW (\ server\ )|(\ service(Group)*\ )
|(\ vserver\ )|(\ policy\ )|(\ policylabel\ )|(\ limitIdentifier\ )|(^show\
 (?!(system|ns\ (ns.conf|runningConfig))))|(save)|(stat\ .*serv)
```


Citrix ADC GUI を使用してコマンドポリシーを設定する

1. **System** > **User Administration** > **Command Policies** に移動します。
2. 詳細ウィンドウで **Add** をクリックして新しいコマンドポリシーを作成します。
3. **[**コマンドポリシーの構成**]** ページで、次のパラメータを設定します。
   a) ポリシー名
   b) 操作（アクション）
   c) コマンド仕様
4. [**OK**] をクリックします。



コマンドポリシーをユーザーアカウントとユーザーグループにバインドする

コマンドポリシーを定義したら、それらを適切なユーザアカウントおよびグループにバインドする必要があります。ポリシーをバインドする場合は、2 つ以上の適用可能なコマンドポリシーが競合している場合に、アプライアンスが従うコマンドポリシーを決定できるように、ポリシーに優先順位を割り当てる必要があります。

コマンドポリシーは、次の順序で評価されます。

- ユーザと対応するグループに直接バインドされたコマンドポリシーは、プライオリティ番号に従って評価されます。プライオリティ番号が小さいコマンドポリシーは、プライオリティ番号が高いポリシーポリシーよりも先に評価されます。したがって、小さい番号のコマンドポリシーが明示的に許可または拒否する特権は、大きい番号のコマンドポリシーによって上書きされません。
- 2 つのコマンドポリシー（1 つはユーザアカウントにバインドされ、もう 1 つはグループにバインドされたコマンドポリシー）が同じプライオリティ番号を持つ場合、ユーザアカウントに直接バインドされたコマンドポリシーが最初に評価されます。

コマンドラインインターフェイスを使用してコマンドポリシーをユーザーにバインドするには

コマンドプロンプトで次のコマンドを入力して、コマンドポリシーをユーザーにバインドし、構成を確認します。

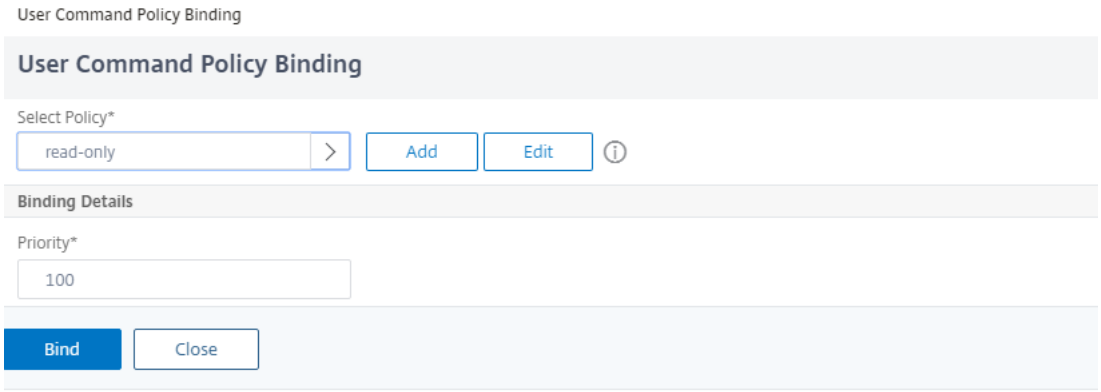
- `bind system user <userName> -policyName <policyName> <priority>`
- `show system user <userName>`

例:

```
> bind system user user1 -policyName read_all 1
```

Citrix ADC GUI を使用してコマンドポリシーをユーザーにバインドする

System > **User Administration** > **Users** に移動してユーザーを選択し、コマンドポリシーをバインドします。



オプションで、デフォルトのプライオリティを変更して、ポリシーが正しい順序で評価されるようにできます。

コマンドラインインターフェイスを使用してコマンドポリシーをグループにバインドするには

コマンドプロンプトで次のコマンドを入力して、コマンドポリシーをユーザーグループにバインドし、構成を確認します。

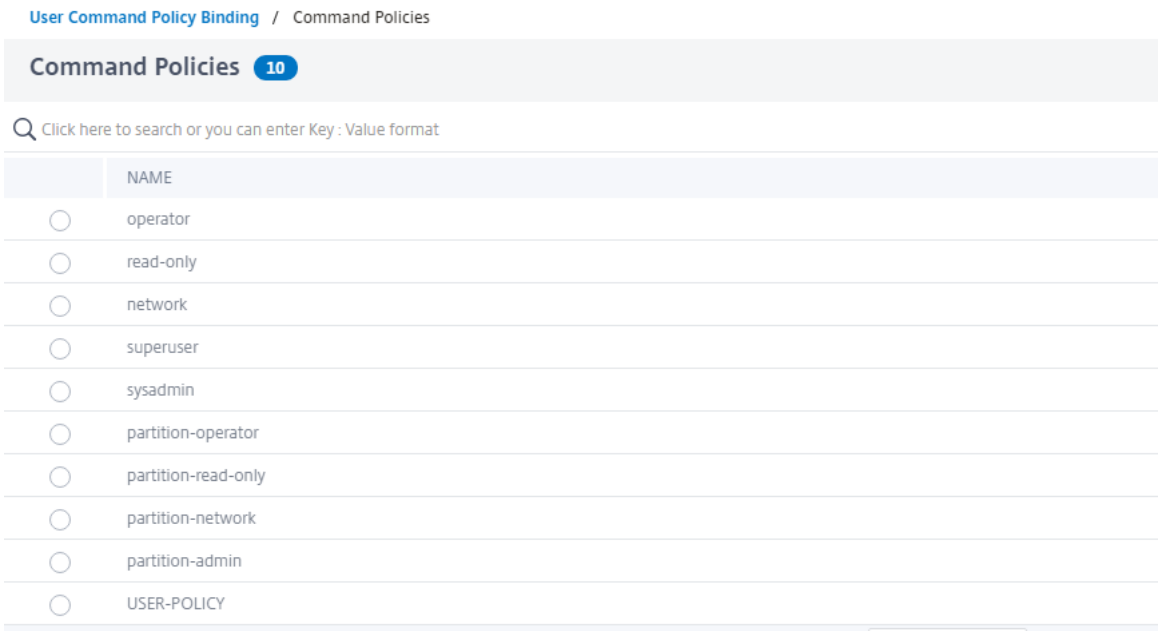
- `bind system group <groupName> -policyName <policyName> <priority>`
- `show system group <groupName>`

例:

```
> bind system group Managers -policyName read_all 1
```

Citrix ADC GUI を使用してコマンドポリシーをグループにバインドする

[システム] > [ユーザー管理] > [グループ] に移動し、グループを選択し、コマンドポリシーをバインドします。



オプションで、デフォルトのプライオリティを変更して、ポリシーが正しい順序で評価されるようにできます。

ユースケースの例：製造組織のユーザーアカウント、ユーザーグループ、およびコマンドポリシーを管理する

次に、ユーザアカウント、グループ、およびコマンドポリシーの完全なセットを作成し、各ポリシーを適切なグループおよびユーザにバインドする例を示します。サンプルマニュファクチャリング社には、Citrix ADC アプライアンスにアクセスできる 3 人のユーザーがいます。

- ジョン・ドー IT マネージャ。John は、Citrix ADC 構成のすべての部分を見ることができる必要がありますが、何も変更する必要はありません。
- マリア・ラミエス主任の IT 管理者。Maria は、Citrix ADC コマンドを除き、Citrix ADC 構成のすべての部分を表示および変更できる必要があります（nsroot としてログオンしている間、ローカル・ポリシーによって指示される必要があります）。
- マイケル・バルドロック負荷分散を担当する IT 管理者。Michael は、Citrix ADC 構成のすべての部分を確認できる必要がありますが、変更する必要があるのは負荷分散機能のみです。

次の表に、サンプル会社のネットワーク情報、ユーザーアカウント名、グループ名、およびコマンドポリシーの内訳を示します。

フィールド	値	注
Citrix ADC のホスト名	ns01.example.net	-
ユーザーアカウント	ジョーンズ、マリアー、マイケルブ	ジョン・ドゥー、IT マネージャー、マリア・ラミレス、IT 管理者、マイケル・バルドロック、IT 管理者。

フィールド	値	注
グループ	マネージャとシステムオペレーション	すべてのマネージャとすべての IT 管理者。
コマンドポリシー	すべて読み取り、変更 lb、およびすべて変更	[完全な読み取り専用アクセスを許可する]、[ロードバランシングへの変更アクセスを許可する]、および[完全な変更アクセスを許可する]。

以下の説明では、ns01.example.net という名前の Citrix ADC アプライアンスにユーザーアカウント、グループ、コマンドポリシーの完全なセットを作成する手順について説明します。

この説明には、適切なユーザアカウントとグループを相互にバインドする手順、および適切なコマンドポリシーをユーザアカウントとグループにバインドする手順が含まれています。

この例では、優先順位付けを使用して、IT 部門の各ユーザーに正確なアクセスと権限を付与する方法を示します。

この例では、Citrix ADC で初期インストールと構成がすでに実行されていることを前提としています。

サンプル組織のユーザーアカウント、グループ、およびコマンドポリシーを構成します

1. 「ユーザーアカウントの構成」セクションで説明されている手順を使用して、ユーザーアカウント **johnd**、**mariar**、および **michaelb** を作成します。
2. 「ユーザグループの設定」で説明されている手順を使用して、ユーザグループ マネージャと **SysOps** を作成し、ユーザ **mariar** と **michaelb** を **SysOps** グループにバインドし、ユーザ **johnd** を マネージャ・グループ。
3. カスタムコマンドポリシーの作成で説明されている手順を使用して、次のコマンドポリシーを作成します。
   - **read_all** とアクション **Allow** およびコマンド仕様”(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)”
   - **modify_lb** とアクション **Allow** およびコマンド仕様”^set\s+lb\s+.*$”
   - modify_all とアクション Allow およびコマンド仕様 ”^\S+\s+(?!system).*”
4. **read_all**コマンドポリシーをプライオリティ値 **1** で **SysOps** グループにバインドするには、「コマンドポリシーをユーザおよびグループへのバインド」で説明する手順を使用します。
5. 「ユーザーおよびグループへのコマンドポリシーのバインド」で説明されている手順を使用して、**modify_lb** コマンドポリシーをユーザー **michaelb** にバインドし、優先度値 **5** を指定します。

作成した構成は、次のようになります。

- IT マネージャである John Doe は、Citrix ADC 構成全体に対して読み取り専用でアクセスできますが、変更を加えることはできません。
- IT 責任者であるマリア・ラミレスは、Citrix ADC 構成のすべての領域にほぼ完全にアクセスでき、Citrix ADC レベルのコマンドを実行するためだけにログオンする必要があります。

- 負荷分散を担当する IT 管理者 Michael Baldrock は、Citrix ADC 構成に読み取り専用でアクセスでき、負荷分散の構成オプションを変更できます。

特定のユーザーに適用されるコマンドポリシーのセットは、ユーザーのアカウントに直接適用されるコマンドポリシーと、ユーザーがメンバーである 1 つ以上のグループに適用されるコマンドポリシーの組み合わせです。

ユーザーがコマンドを入力するたびに、オペレーティングシステムは、コマンドに一致する許可または拒否アクションを持つポリシーが見つかるまで、そのユーザーのコマンドポリシーを検索します。一致するものが見つかると、オペレーティングシステムはコマンドポリシーの検索を停止し、コマンドへのアクセスを許可または拒否します。

オペレーティングシステムが一致するコマンドポリシーが見つからない場合、Citrix ADC アプライアンスのデフォルトの拒否ポリシーに従って、コマンドへのユーザーアクセスを拒否します。

> 注:
>
> ユーザを複数のグループに配置する場合は、意図しないユーザコマンドの制限や権限が発生しないように注意してください。これらの競合を回避するには、ユーザーをグループに編成する場合は、Citrix ADC コマンドのポリシー検索手順とポリシーの順序付け規則に注意してください。

ユーザーアカウントとパスワードの管理

October 7, 2021

Citrix ADC を使用すると、ユーザーアカウントとパスワードの構成を管理できます。以下は、アプライアンスのシステムユーザーアカウントまたは `nsroot` 管理ユーザーアカウントに対して実行できるアクティビティの一部です。
Google NMT

- システムユーザーアカウントのロックアウト
- 管理アクセスのためにシステムユーザーアカウントをロックする
- 管理アクセスのためにシステムユーザーアカウントをロックする
- システムユーザーアカウントの管理アクセスを無効にする
- `nsroot` 管理ユーザーのパスワード変更を強制する
- システムユーザーアカウントの機密ファイルを削除する
- システムユーザーのための強いパスワード構成

システムユーザーアカウントのロックアウト

ブルートフォースセキュリティ攻撃を防ぐために、ユーザーロックアウト構成を構成できます。この構成により、ネットワーク管理者は、システムユーザーが Citrix ADC アプライアンスにログオンできないようにすることができます。また、ロック期間が終了する前にユーザーアカウントのロックを解除します。

コマンドプロンプトで入力します。


```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <value> -
persistentLoginAttempts (ENABLED | DISABLED)
```

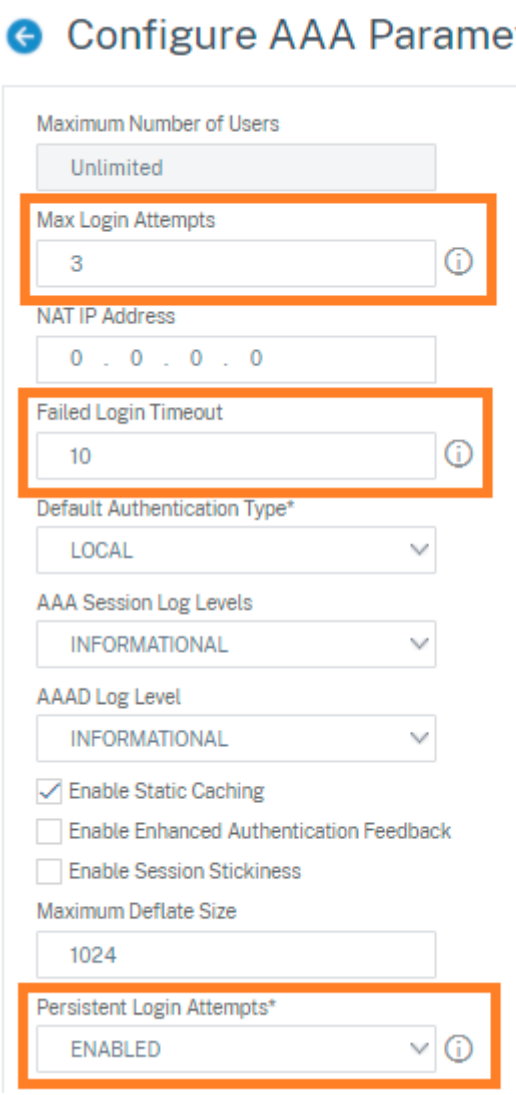
> 注
>
> 失敗したユーザーログイン試行の永続ストレージの詳細を取得するには、「persistentLoginAttempts」パラメーターを有効にする必要があります。

例:

```
set aaa parameter -maxloginAttempts 3 -failedLoginTimeout 10 -persistentLoginAttempts
 ENABLED
```

GUI を使用してシステムユーザーアカウントのロックアウトを構成する

1. **Configuration > Security > AAA-Application Traffic > Authentication Settings > Change authentication AAA Settings** に移動します。
2. [**AAA** パラメータの設定] ページで、次のパラメータを設定します。
   a) 最大ログイン試行回数。ユーザーが試行できるログオン試行の最大数。
   b) ログインタイムアウトに失敗しました。ユーザーによる無効なログオン試行の最大数。
   c) 永続的なログイン試行。失敗したユーザーログイン試行の永続的なストレージ。
3. [**OK**] をクリックします。



パラメータを設定すると、3 回以上の無効なログイン試行でユーザーアカウントが 10 分間ロックされます。また、ユーザーは有効な資格情報を使用しても 10 分間ログオンできません。

> 注
>
> ロックされたユーザーがアプライアンスにログオンしようとすると、エラーメッセージ `RBA Authentication Failure: maxlogin attempt reached for test.` が表示されます。

管理アクセスのためにシステムユーザーアカウントをロックする

Citrix ADC アプライアンスを使用すると、システムユーザーを 24 時間ロックし、ユーザーへのアクセスを拒否できます。

Citrix ADC アプライアンスは、システムユーザーと外部ユーザーの両方の構成をサポートします。

> 注
>
> この機能は、aaa パラメーターで persistentLoginAttempts オプションを無効にした場合にのみサポートされます。

コマンドプロンプトで次のように入力します。

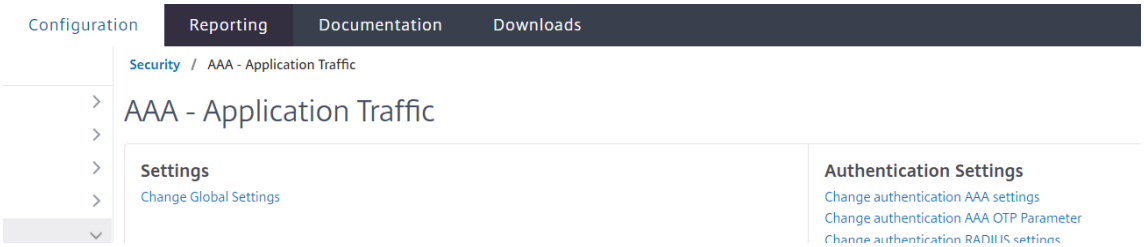
```
set aaa parameter -persistentLoginAttempts DISABLED
```

ここで、ユーザーアカウントをロックするには、コマンドプロンプトで次のように入力します。

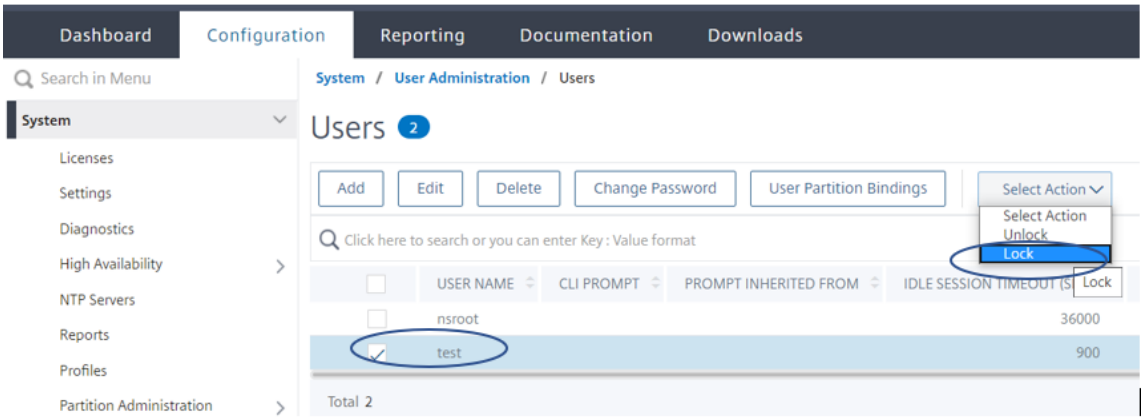
```
lock aaa user test
```

GUI を使用してシステムユーザーアカウントをロックする

1. **Configuration > Security > AAA-Application Traffic > Authentication Settings > Change authentication AAA Settings** に移動します。



2. [**Configure AAA Parameter]** の [**PersistentLogin Attempts]** リストで、[**DISABLED]** を選択します。
3. [**System**] > [**User Administration**] > [**Users**] の順に選択します。
4. ユーザーを選択してください。
5. [アクションの選択] リストで、[ロック] を選択します。



> 注
>
> Citrix ADC GUI には、外部ユーザーをロックするオプションがありません。外部ユーザーをロックするには、

> ADC 管理者は CLI を使用する必要があります。
> ロックされたシステムユーザー（ロック認証、承認、および監査ユーザーコマンドでロックされている）が Citrix ADC にログインしようとすると、アプライアンスは「RBA 認証の失敗：ユーザーテストが 24 時間ロックダウンされています」というエラーメッセージを表示します。
>
> ユーザーが管理アクセスにログオンするためにロックされている場合、コンソールアクセスは免除されます。ロックされたユーザーはコンソールにログオンできます。

管理アクセスのためにシステムユーザーアカウントをロックする

システムユーザーと外部ユーザーは、lock authentication、authorization、and auditinguser コマンドを使用して 24 時間ロックできます。

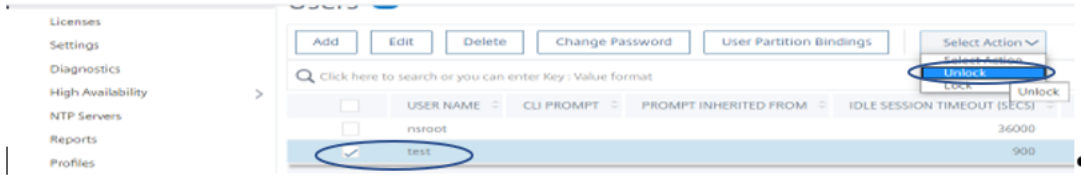
> 注
>
> ADC アプライアンスを使用すると、管理者はロックされたユーザーのロックを解除でき、この機能では「persistentloginAttempts」コマンドの設定は必要ありません。

コマンドプロンプトで入力します。

```
unlock aaa user test
```

GUI を使用してシステムユーザーのロック解除を構成する

1. ［**System**］**>** ［**User Administration**］**>** ［**Users**］の順に選択します。
2. ユーザーを選択してください。
3. ［ロック解除］をクリックします。



Citrix ADC GUI は、ADC で作成されたシステムユーザーのみを一覧表示するため、GUI には外部ユーザーのロックを解除するオプションはありません。外部ユーザーのロック解除をするには、`nsroot`管理者は CLI を使用する必要があります。

システムユーザーアカウントの管理アクセスを無効にする

アプライアンスで外部認証が構成されていて、管理者としてシステムユーザーへのアクセスを拒否して管理アクセスにログオンする場合は、システムパラメーターの localAuth オプションを無効にする必要があります。

コマンドプロンプトで、次のように入力します。

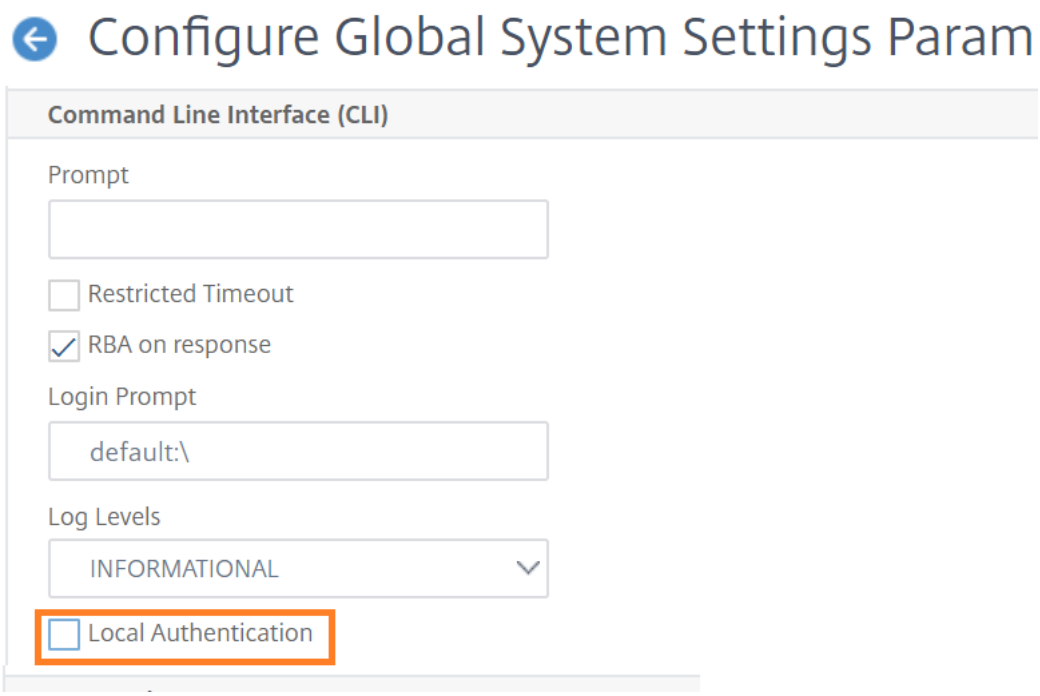

```
set system parameter localAuth <ENABLED|DISABLED>
```

例:

```
set system parameter localAuth DISABLED
```

GUI を使用して、システムユーザーへの管理アクセスを無効にします

1. **Configuration > System > Settings > Change Global System Settings** に移動します。
2. **Command Line Interface (CLI)** セクションで、**Local Authentication** チェックボックスをオフにします。



このオプションを無効にすると、ローカルシステムユーザーは ADC 管理アクセスにログオンできなくなります。

> 注
>
> システムパラメータでローカルシステムユーザー認証を許可しないように、外部認証サーバーを構成して到達可能にする必要があります。管理アクセス用に ADC で構成された外部サーバーに到達できない場合、ローカルシステムユーザーはアプライアンスにログオンできます。この動作は、回復を目的として設定されています。

管理ユーザーのパスワード変更を強制する

nsrootセキュア認証の場合、Citrix ADC アプライアンスはforcePasswordChangeオプションがシステムパラメーターで有効になっている場合、デフォルトのパスワードを新しいパスワードに変更するようメッセージを表示します。nsroot パスワードは、デフォルトの資格情報を使用した最初のログイン時に、CLI または GUI から変更できます。

コマンドプロンプトで、次のように入力します。

```
set system parameter -forcePasswordChange ( ENABLED | DISABLED )
```

NSIP の **SSH** セッション例:

```
ssh nsroot@1.1.1.1
Connecting to 1.1.1.1:22...
Connection established.
To escape to local shell, press Ctrl+Alt+].
###############################################################################

WARNING: Access to this system is for authorized users only #
Disconnect IMMEDIATELY if you are not an authorized user! #

###############################################################################

Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
Done
<!--NeedCopy-->
```

システムユーザーアカウントの機密ファイルを削除する

システムユーザーアカウントで認証キーやパブリックキーのような機密データを管理するには、removeSensitiveFilesオプションを有効にする必要があります。システムパラメータが有効になっているときに機密ファイルを削除するコマンドは次のとおりです。

- rm cluster instance
- rm cluster node
- rm high availability node
- clear config full
- join cluster
- add cluster instance

コマンドプロンプトで、次のように入力します。

```
set system parameter removeSensitiveFiles ( ENABLED | DISABLED )
```

例:

```
set system parameter -removeSensitiveFiles ENABLED
```


システムユーザーのための強いパスワード構成

安全な認証のために、Citrix ADC アプライアンスは、システムユーザーと管理者に、アプライアンスにログオンするための強力なパスワードを設定するように求めます。パスワードは長く、次の組み合わせである必要があります。

- 1 つの小文字
- 1 つの大文字
- 1 つの数字
- 1 つの特殊文字

コマンドプロンプトで、次のように入力します。

```
set system parameter -strongpassword <value> -minpasswordlen <value>
```

各項目の意味は次のとおりです。

Strongpassword。強力なパスワードを有効にした後（enable all / enablelocal）すべてのパスワードまたは機密情報には次のものが必要です。

- 少なくとも 1 つの小文字
- 少なくとも 1 つの大文字
- 少なくとも 1 つの数字
- 少なくとも 1 つの特殊文字

enablelocalで除外する一覧は-NS_FIPS、NS_CRL、NS_RSAKEY、NS_PKCS12、NS_PKCS8、NS_LDAP、NS_TACACS、NS_TACACSACTION、NS_RADIUS、NS_RADIUSACTION、NS_ENCRYPTION_PARAMS。したがって、システムユーザーのこれらの ObjectType コマンドに対して強力なパスワードチェックは実行されません。

可能な値：enableall、enablelocal、無効
デフォルト値：無効

minpasswordlen。システムユーザーパスワードの最小の長さ。強力なパスワードがデフォルトで有効になっている場合、最小の長さは 4 です。4 以上の値が入力されている可能性があります。強力なパスワードが無効な場合、デフォルトの最低値は 1 です。どちらの場合でも最大値は 127 です。

最小値:1,
最大値:127

例:

```
set system parameter -strongpassword enablelocal -minpasswordlen 6
```

デフォルトのユーザーアカウント

管理者は、nsrecover ユーザーアカウントを使用して Citrix ADC アプライアンスを回復できます。予期しない問題が原因でデフォルトのシステムユーザー（nsroot）がログインできない場合は、nsrecover を使用して ADC

アプライアンスにログインできます。nsrecover ログインはユーザー構成に依存せず、シェルプロンプトに直接アクセスできます。構成の最大制限に達しているかどうかに関係なく、nsrecover を介して常にログインできます。

ルート管理者 **(nsroot)** パスワードをリセットする方法

June 1, 2022

Citrix ADC ルート管理者（nsroot）アカウントは、すべての ADC 機能への完全なアクセスを提供します。したがって、セキュリティを維持するために、管理者アカウントは必要な場合にのみ使用する必要があります。

管理者として、パスワードを変更することをお勧めします。パスワードを忘れた場合は、最初にデフォルトのパスワードにリセットしてから、新しいパスワードに変更する必要があります。

nsroot 管理者は、パスワードをリセットするには、アプライアンスにログオンしてパスワードを変更する必要があります。ただし、パスワードを覚えていない場合は、アプライアンスをシングルユーザーモードで再起動できます。ファイルシステムを読み取り/書き込みモードでマウントし、**ns.conf** ファイルから **Citrix ADC** エントリを削除します。最後の手順として、再起動してデフォルトのパスワードでアプライアンスにログオンし、新しいパスワードを設定します。

root 管理者パスワードをリセットするには、以下のステップを実行します。

1. コンピューターを Citrix ADC コンソールポートに接続してログオンします。

   > 注
   >
   > この手順を実行するために SSH を使用してログオンすることはできません。アプライアンスに直接接続する必要があります。

2. Citrix ADC を再起動します。
3. 次のメッセージが表示されたら、Ctrl キーを押しながら C キーを押します。

   ```
   Press [Ctrl-C] for command prompt, or any other key to boot immediately
   .
   Booting [kernel] in ## seconds.
   ```

   > 注
   >
   > Azure シリアルコンソールでは、ADC アプライアンスが起動されるまで、Citrix ADC アプライアンスはシングルブートをサポートしません。

4. シングルユーザーモードで Citrix ADC を起動するには、次のコマンドを実行します。

   ```
   boot -s
   ```

   アプライアンスの起動後、次のメッセージが表示されます。

   シェルのフルパス名またはRETURN for /bin/sh:を入力する

5. Enter キーを押して # プロンプトを表示し、次のコマンドを入力してファイルシステムをマウントします。

   a) 次のコマンドを実行して、ディスクの整合性を確認します。

   ```
   fsck_ufs /dev/ad0s1a
   ```

   > 注
   >
   > フラッシュドライブには、Citrix ADC に応じて特定のデバイス名があるため、前述のコマンドの ad0s1a を適切なデバイス名に置き換える必要があります。

   b) dev ディレクトリにアクセスし、'ls' と入力してドライブの詳細を確認します。

   c) 次のコマンドを実行して、マウントされたパーティションを表示します。

   ```
   df
   ```

   > (注
   >
   >) フラッシュパーティションがリストにない場合は、手動でマウントする必要があります。

   d) 次のコマンドを実行して、フラッシュドライブをマウントします。

   ```
   mount dev/ad0s1a/flash
   ```

6. 以下のコマンドを実行して、nsconfig ディレクトリに移動します。

   ```
   cd /flash/nsconfig
   ```

7. 次のコマンドを実行して ns.conf ファイルを書き換え、デフォルトで admin に設定されている一連のシステムコマンドを削除します。

   a) 次のコマンドを実行して、管理者にデフォルト設定されるコマンドを含まない設定ファイルを作成します。

   ```
   grep -v “set system user nsroot” ns.conf > new.conf
   ```

   b) 次のコマンドを実行して、既存の設定ファイルのバックアップを作成します。

   ```
   mv ns.conf old.ns.conf
   ```

   c) 次のコマンドを実行して、新しい.conf ファイルの名前を ns.conf に変更します。

   ```
   mv new.conf ns.conf
   ```

8. 次のコマンドを実行して、Citrix ADC を再起動します。

   ```
   reboot
   ```

9. デフォルトの管理者認証情報を使用してログオンします。

10. 以下のコマンドを実行して、管理者パスワードをリセットします。

    ```
    set system user nsroot <New_Password>
    ```


> 注
>
> “? ” パスワード文字列内の文字。この文字の前に\文字を付けます。

たとえば、次の操作を実行した後、yourexamplepasswd\?が管理者アカウントに設定されています。

```
> set system user nsroot yourexamplepasswd\?
```

> 注
>
> 高可用性設定で忘れた（nsroot）パスワードをリセットするには、ピアノードをシャットダウンすることをお勧めします。ピアノードがアクティブな場合、再起動後にノードが起動すると config sync がトリガーされるため、パスワードは上書きされます。

また、Citrix ADC アプライアンスへのセキュアな SSH アクセスの仕組みについては、Citrix の記事 CTX224027 を読んでください 。

外部ユーザー認証

January 31, 2022

Citrix ADC アプライアンスの認証サービスは、ローカルでも外部でもかまいません。外部ユーザー認証では、アプライアンスは LDAP、RADIUS、TACACS + などの外部サーバーを使用してユーザーを認証します。外部ユーザーを認証し、アプライアンスへのアクセスをユーザーに許可するには、認証ポリシーを適用する必要があります。Citrix ADC システム認証は、高度なポリシー式を備えた高度な認証ポリシーを使用します。高度な認証ポリシーは、パーティション化された Citrix ADC アプライアンスのシステムユーザー管理にも使用されます。

> 注
>
> アプライアンスがまだクラシックポリシーとその式を使用している場合は、アプライアンスの使用を停止し、クラシックポリシーの使用を高度なポリシーインフラストラクチャに移行する必要があります。

認証ポリシーを作成したら、それをシステムグローバルエンティティにバインドする必要があります。単一の認証ポリシーをシステムグローバルエンティティにバインドすることで、外部認証サーバ（TACACS など）を設定できます。また、複数のポリシーをシステムグローバルエンティティにバインドすることによって、認証サーバのカスケードを構成することもできます。

> 注
>
> 外部ユーザーがアプライアンスにログインすると、システムは ns.log ファイルに「ユーザーが存在しません」というエラーメッセージを生成します。発生は、システムが実行しているためです systemuser_systemcmdpolicy_binding ユーザーの GUI を初期化するコマンド。

LDAP 認証（外部 **LDAP** サーバーを使用）

Citrix ADC アプライアンスは、1 つ以上の LDAP サーバーによるユーザーアクセスを認証するように構成できます。LDAP 許可には、Active Directory、LDAP サーバー、およびアプライアンスで同一のグループ名が必要です。文字と大文字小文字も同じである必要があります。

LDAP 認証ポリシーの詳細については、「LDAP 認証ポリシー 」トピックを参照してください。

デフォルトでは、LDAP 認証は SSL/TLS プロトコルを使用して保護されています。セキュア LDAP 接続には 2 つのタイプがあります。最初のタイプでは、LDAP サーバーは、クリア LDAP 接続を受け入れるために使用されるポートとは別のポートで SSL/TLS 接続を受け入れます。ユーザーが SSL/TLS 接続を確立すると、LDAP トラフィックは接続を介して送信できます。2 番目のタイプでは、非セキュアな LDAP 接続とセキュアな LDAP 接続の両方を許可し、単一のポートがサーバ上でそれを処理します。このシナリオでは、セキュリティで保護された接続を作成するには、クライアントはまず、クリア LDAP 接続を確立します。次に、**LDAP** コマンド StartTLS が接続を介してサーバーに送信されます。LDAP サーバーが StartTLS をサポートしている場合、接続は TLS を使用してセキュリティで保護された LDAP 接続に変換されます。

LDAP 接続のポート番号は次のとおりです。

- セキュリティで保護されていない LDAP 接続の場合は 389
- 安全な LDAP 接続用の 636
- Microsoft のセキュリティで保護されていない LDAP 接続の場合 3268
- Microsoft の LDAP 接続のセキュリティで保護された 3269

StartTLS コマンドを使用する LDAP 接続では、ポート番号 389 が使用されます。アプライアンスでポート番号 389 または 3268 が構成されている場合、StartTLS を使用して接続を試みます。他のポート番号が使用されている場合、接続試行は SSL/TLS を使用します。StartTLS または SSL/TLS を使用できない場合、接続は失敗します。

LDAP サーバを設定する場合、英字の大文字と小文字は、サーバとアプライアンスの大文字と小文字と一致する必要があります。LDAP サーバーのルートディレクトリが指定されている場合、すべてのサブディレクトリも検索され、ユーザー属性が検索されます。大きなディレクトリでは、パフォーマンスに影響する可能性があります。このため、特定の組織単位（OU）を使用することをお勧めします。

次の表に、ベース識別名 (DN) の例を示します。

LDAP サーバ	ベース **DN**
Microsoft Active Directory	DC=Citrix, DC=local
Novell eDirectory	dc=Citrix, dc=net
IBM Directory Server	cn=users
Lotus Domino	OU=City, O=Citrix, C=US
Sun ONE Directory（旧 iPlanet）	ou=People, dc=Citrix, dc=com

次の表に、バインド識別名 (DN) の例を示します。

LDAP サーバ	バインド **DN**
Microsoft Active Directory	CN=Administrator, CN=Users, DC=Citrix, DC=local
Novell eDirectory	cn=admin, dc=Citrix, dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE Directory（旧 iPlanet）	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

LDAP サーバ	バインド **DN**
Microsoft Active Directory	CN=Administrator, CN=Users, DC=Citrix, DC=local
Novell eDirectory	cn=admin, dc=Citrix, dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE Directory（旧 iPlanet）	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

CLI を使用して **LDAP** ユーザー認証を構成します

外部ユーザーの LDAP 認証を構成するには、次の手順を実行します

LDAP ポリシーを構成する

コマンドプロンプトで、次の操作を行います。

ステップ 1: LDAP アクションを作成します。

```
add authentication ldapAction <name> { -serverIP <ip_addr|ipv6_addr|*> | {
 -serverName <string> } } >] [-authTimeout <positive_integer>] [-ldapBase
<string>] [-ldapBindDn <string>] { -ldapBindDnPassword } [-ldapLoginName <
string>] [-groupAttrName <string>] [-subAttributeName <string>]
```

例**:**


```
add authentication ldapAction ldap_act -serverIP <IP> -authTimeout 30 -
ldapBase ”CN=xxxxx,DC=xxxx,DC=xxx”-ldapBindDn ”CN=xxxxx,CN=xxxxx,DC=xxxx,DC
=xxx”-ldapBindDnPassword abcd -ldapLoginName sAMAccountName -groupattrName
memberOf -subAttributeName CN
```

パラメータの説明については、「認証および認可コマンドのリファレンス」を参照してください。

ステップ 2: 従来の LDAP ポリシーを作成します。

```
add authentication ldapPolicy <name> <rule> [<reqAction>]
```

例:

```
add authentication ldappolicy ldap_pol_classic ns_true ldap_act
```

> 注
>
> クラシックまたは高度な LDAP ポリシーを使用して構成できますが、クラシックポリシーは Citrix ADC 13.0 リリース以降から廃止されるため、高度な認証ポリシーを使用することをお勧めします。

ステップ 3: 高度な LDAP ポリシーを作成する

```
add authentication Policy <name> <rule> [<reqAction>]
```

例:

```
add authentication policy ldap_pol_advance -rule true -action ldap_act
```

ステップ 4: LDAP ポリシーをシステムグローバルにバインドする

コマンドラインプロンプトで、次の操作を実行します。

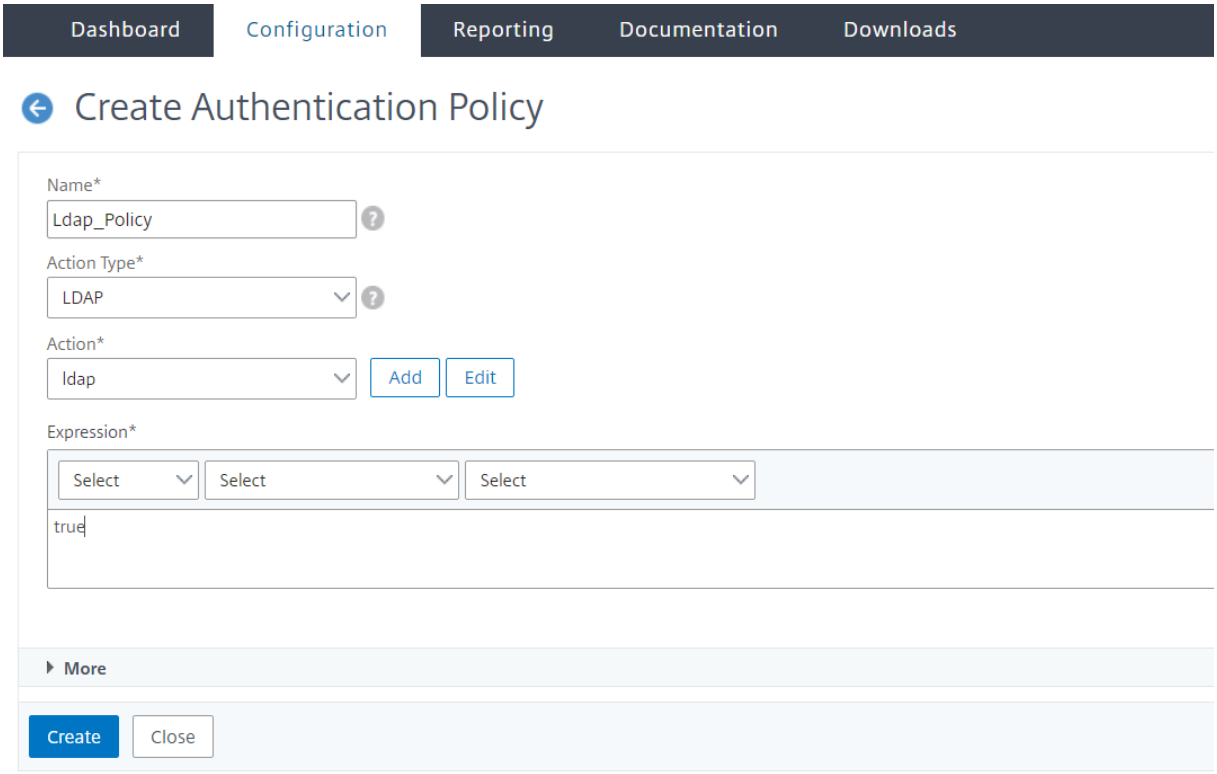
```
bind system global <policyName> [-priority <positive_integer]
```

例:

```
bind system global ldap_pol_advanced -priority 10
```

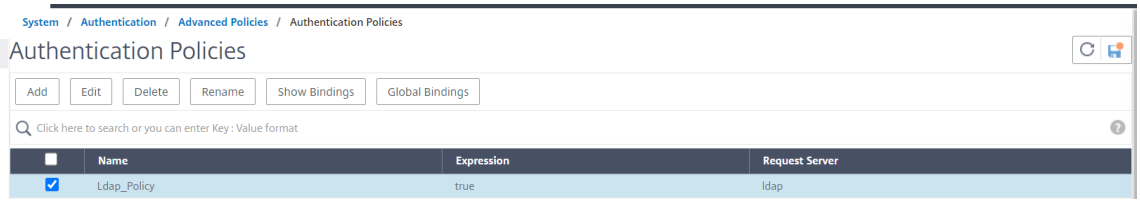
Citrix ADCGUI を使用して LDAP ユーザー認証を構成する

1. **System** > **Authentication** > **Advanced Policies** > **Policy** に移動します。
2. [追加] をクリックして、LDAP タイプの認証ポリシーを作成します。
3. **[**作成**]** して **[**閉じる**]** をクリックします。

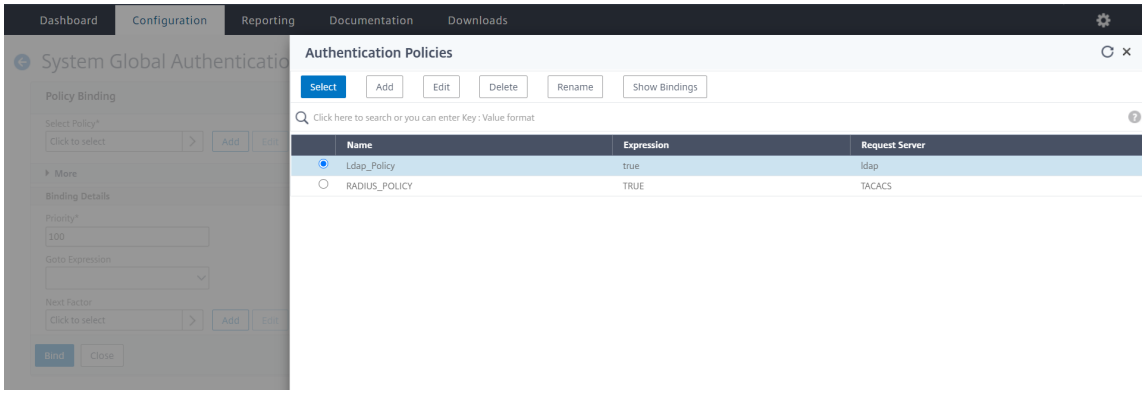


Citrix ADC GUI を使用して、**LDAP** 認証用に認証ポリシーをシステムグローバルにバインドします

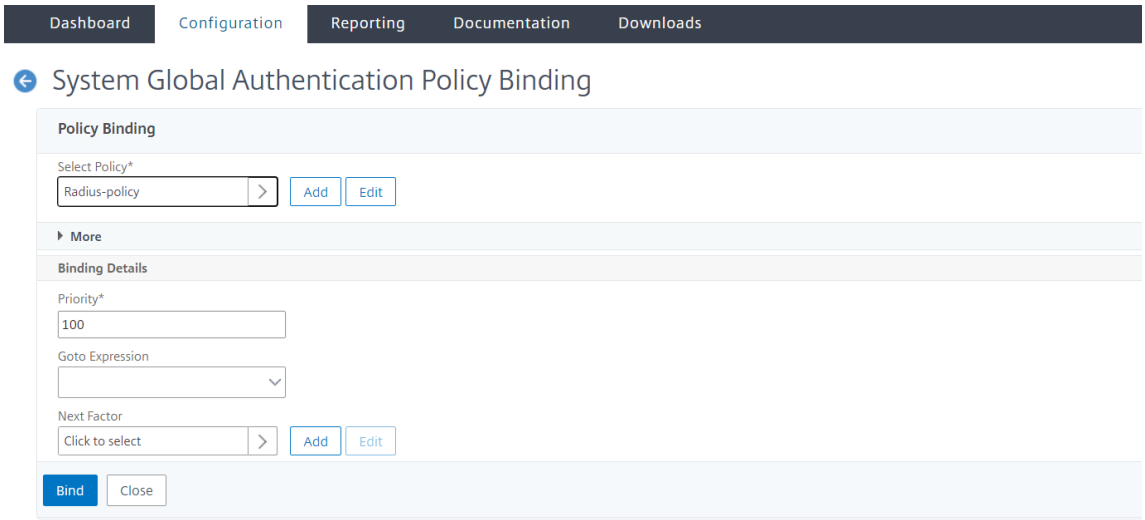
1. システムに移動 > 認証 > 高度なポリシー > 認証ポリシーポリシー。
2. 詳細ペインで、[グローバルバインディング] をクリックして、システムグローバル認証ポリシーバインディングを作成します。
3. **Global Bindings** をクリックします。



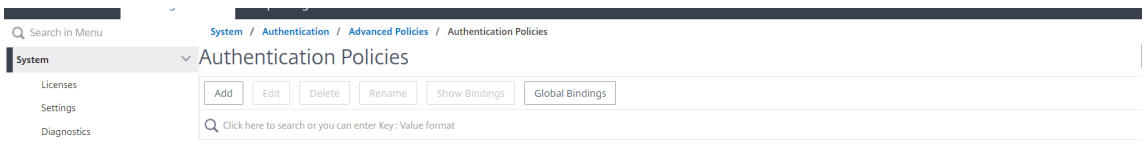
4. 認証プロファイルを選択します。



5. LDAP のポリシーを選択します。
6. [システムグローバル認証ポリシーのバインド] ページで、次のパラメータを設定します。
   a) ポリシーを選択します。
   b) バインディングの詳細



7. [バインド] と [完了] をクリックします。
8. [グローバルバインディング] をクリックして、システムグローバルにバインドされているポリシーを確認します。



LDAP ディレクトリ内の属性の決定

LDAP ディレクトリ属性を決定する助けが必要な場合は、Softerra の無料の LDAP ブラウザーで簡単に検索できます。

LDAP ブラウザは、Softerra LDAP 管理者 Web サイト<http://www.ldapbrowser.com>からダウンロードできます。ブラウザをインストールしたら、次の属性を設定します。

- LDAP サーバーのホスト名または IP アドレス。
- LDAP サーバーのポート。デフォルトは 389 です。
- ベース DN フィールドは空白のままにできます。
- LDAP ブラウザから提供される情報は、[認証] タブに必要なベース DN を決定するのに役立ちます。
- 匿名バインドチェックは、LDAP サーバーが接続するためにブラウザーのユーザークレデンシャルを必要とするかどうかを決定します。LDAP サーバーでクレデンシャルが必要な場合は、チェックボックスをオフのままにします。

設定が完了すると、LDAP ブラウザは左ペインにプロファイル名を表示し、LDAP サーバに接続します。

詳細については、LDAP のトピックを参照してください。

LDAP ユーザーに対するキーベース認証のサポート

キーベース認証では、SSH を使用して LDAP サーバー内のユーザーオブジェクトに格納されている公開キーのリストを取得できるようになりました。ロールベース認証（RBA）プロセス中の Citrix ADC アプライアンスは、LDAP サーバーから公開 SSH キーを抽出する必要があります。取得した公開キーは SSH と互換性があり、RBA メソッドを使用してログインできる必要があります。

「認証の追加 ldapAction」コマンドと「認証の設定 ldapAction」コマンドに新しい属性「sshPublicKey」が導入されました。この属性を使用すると、次の利点があります。

- 取得した公開キーを格納できます。LDAP アクションは、この属性を使用して LDAP サーバから SSH キー情報を取得します。
- 最大 24 KB の属性名を抽出できます。

> 注
>
> LDAP などの外部認証サーバは、SSH キー情報の取得にのみ使用されます。認証目的には使用されません。

次に、SSH を介したイベントのフローの例を示します。

- SSH デーモンは、パスワードフィールドが空の AAA_AUTHENTICATE 要求を認証、認可、および監査デーモンポートに送信します。
- LDAP が SSH 公開鍵を保存するように設定されている場合、認証、承認、および監査は「sshPublicKey」属性と他の属性とともに応答します。
- SSH デーモンは、クライアントキーでこれらのキーを検証します。
- SSH デーモンは、要求ペイロードでユーザー名を渡し、認証、承認、および監査は、汎用キーとともにこのユーザーに固有のキーを返します。

sshPublicKey 属性を構成するには、コマンドプロンプトで次のコマンドを入力します。

- 追加操作では、ldapAction コマンドの設定中に「sshPublicKey」属性を追加できます。


```
add authentication ldapAction <name> { -serverIP <ip_addr|ipv6_addr
|*> | { -serverName <string> } } [-serverPort <port>] … [-Attribute1 <
string>] … [-Attribute16 <string>][-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```

- 設定操作では、すでに追加されている ldapAction コマンドに「sshPublicKey」属性を設定できます。

```
set authentication ldapAction <name> [-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```

RADIUS 認証（外部 RADIUS サーバーを使用）

Citrix ADC アプライアンスは、1 つ以上の RADIUS サーバーによるユーザーアクセスを認証するように構成できます。RSA SecurID、SafeWord、または Gemalto Protiva 製品を使用している場合は、RADIUS サーバを使用してください。

RADIUS 認証ポリシーの詳細については、「RADIUS 認証ポリシー」トピックを参照してください。

構成によっては、ネットワークアクセスサーバの IP アドレス (NAS IP) またはネットワークアクセスサーバ識別子 (NAS ID) の使用が必要になる場合があります。RADIUS 認証サーバを使用するようにアプライアンスを設定する場合は、次のガイドラインに従ってください。

- NAS IP の使用を有効にした場合、アプライアンスは、RADIUS 接続の確立に使用される送信元 IP アドレスではなく、構成済みの IP アドレスを RADIUS サーバに送信します。
- NAS ID を構成すると、アプライアンスは RADIUS サーバーにこの識別子を送信します。NAS ID を構成しないと、アプライアンスは RADIUS サーバーにホスト名を送信します。
- NAS IP アドレスが有効になっている場合、アプライアンスは RADIUS サーバーとの通信に使用した NASID をすべて無視します。

GUI を使用した RADIUS 認証の構成

コマンドプロンプトで、次の操作を行います。

ステップ 1: RADIUS アクションを作成します。

```
add authentication radiusaction <name> -serverip <ip> -radkey <key> -
radVendorID <id> -radattributetype <value>
```

ここで、
radVendorID RADIUS ベンダー ID 属性。RADIUS グループの抽出に使用されます。
radAttributeType RADIUS 属性タイプ。RADIUS グループの抽出に使用されます。

例:

```
add authentication radiusaction RADserver531 rad_action -serverip 1.1.1.1 -
radkey key123 -radVendorID 66 -radattributetype 6
```


ステップ 2: 従来の RADIUS ポリシーを作成します。

```
add authentication radiusPolicy <name> <rule> [<reqAction>]
```

例:

```
add authentication radiuspolicy radius_pol_classic ns_true radius_act
```

> 注
>
> クラシック RADIUS ポリシーまたは高度な RADIUS ポリシーを使用して設定できます。従来のポリシーは Citrix ADC 13.0 リリースから廃止されるため、高度な認証ポリシーを使用することをお勧めします。

ステップ 3: 高度な RADIUS ポリシーを作成する

```
add authentication policy <policyname> -rule true -action <radius action
name>
```

例:

```
add authentication policy rad_pol_advanced -rule true -action radserver531rad_action
```

ステップ 4: RADIUS ポリシーをシステムグローバルにバインドします。

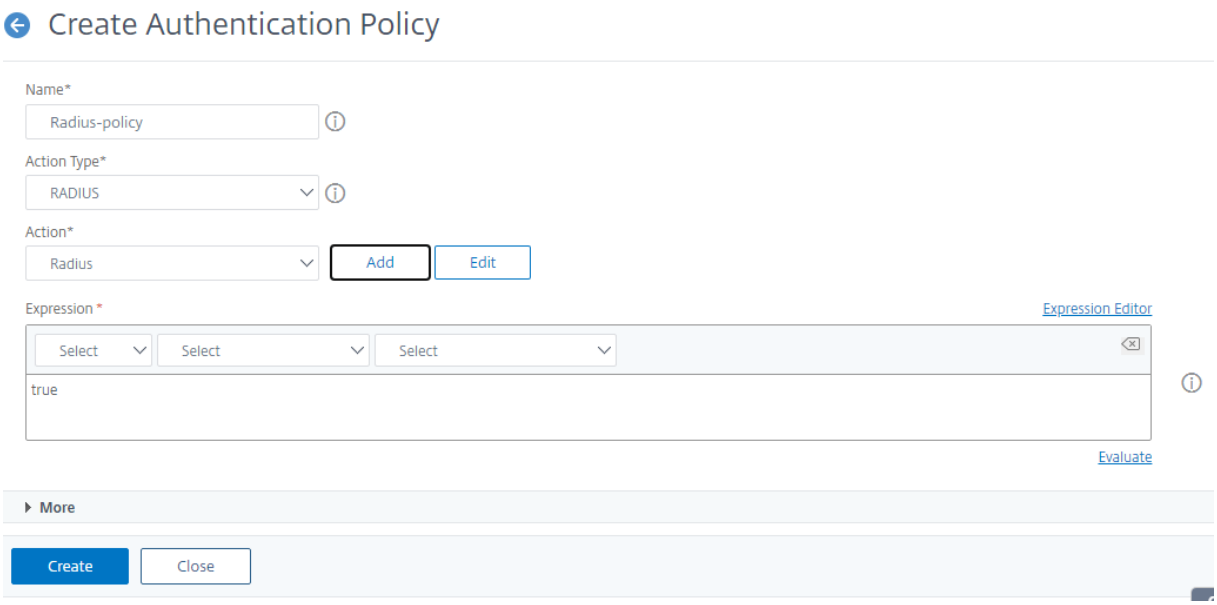
```
bind system global <policyName> -priority <positive_integer
```

例:

```
bind system global radius_pol_advanced -priority 10
```

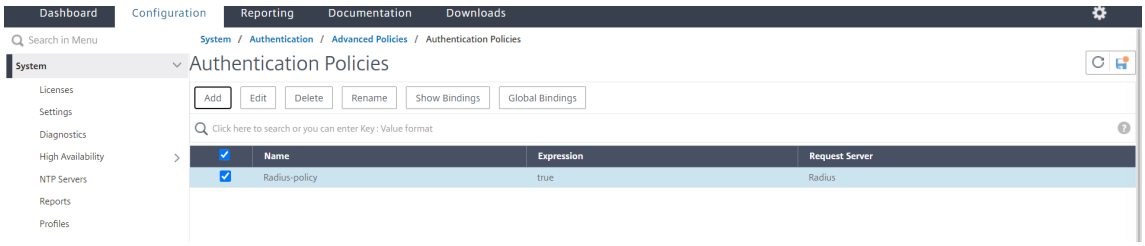
GUI を使用した **RADIUS** 認証の構成

1. **System** > **Authentication** > **Advanced Policies** > **Policy** に移動します。
2. [追加] をクリックして、RADIUS タイプの認証ポリシーを作成します。
3. **[**作成**]** して **[**閉じる**]** をクリックします。

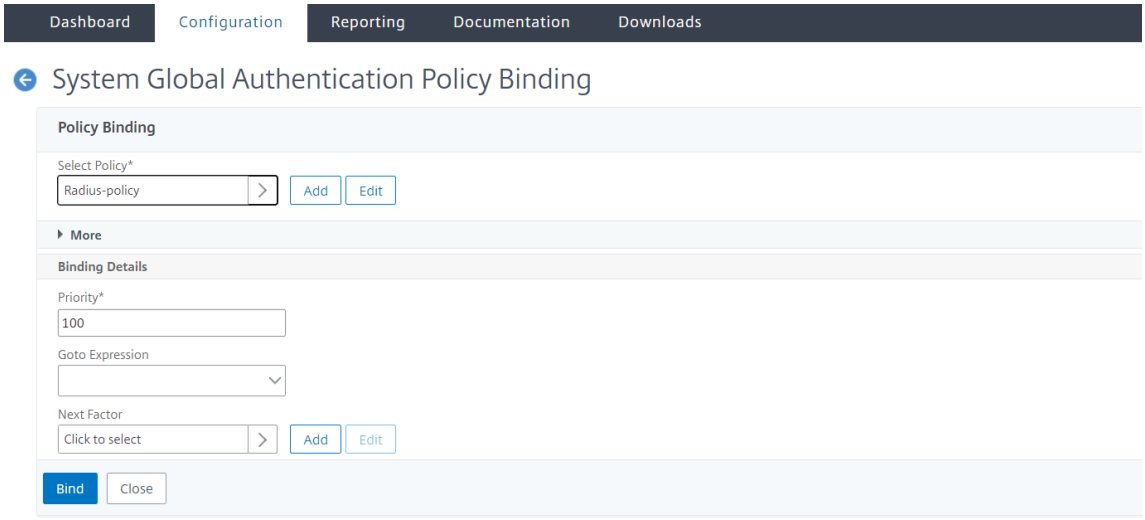


GUI を使用して、**RADIUS** 認証用に認証ポリシーをシステムグローバルにバインドします

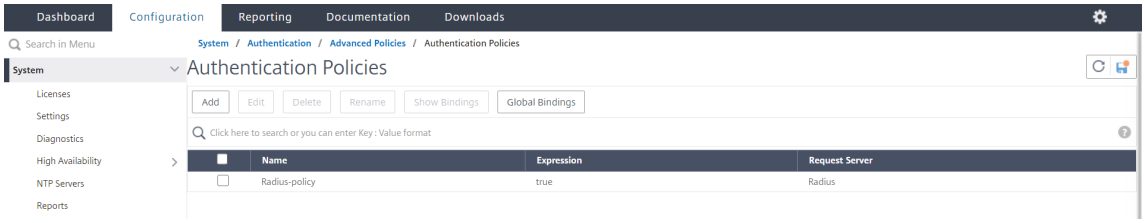
1. **System** > **Authentication** > **Advanced Policies** > **Policy** に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックして、システムグローバル認証ポリシーバインディングを作成します。
3. **Global Bindings** をクリックします。



4. RADIUS を選択します。
5. [システムグローバル認証ポリシーのバインド] ページで、次のパラメータを設定します。
   a) ポリシーを選択します。
   b) 詳細をバインドします。



6. [バインドして 閉じる] をクリックします。
7. [グローバルバインディング] をクリックして、システムグローバルにバインドされているポリシーを確認します。



RADIUS ユーザー認証プロトコルを選択します

Citrix ADC アプライアンスは、ユーザー認証に次のようないくつかのプロトコルのいずれかを使用するように構成された RADIUS の実装をサポートします。

- パスワード認証プロトコル
- チャレンジハンドシェイク認証プロトコル (CHAP)
- Microsoft のチャレンジハンドシェイク認証プロトコル (MS-CHAP バージョン 1 およびバージョン 2)

展開が RADIUS 認証を使用するように構成され、RADIUS サーバにパスワード認証プロトコルが設定されている場合。RADIUS サーバに強力な共有秘密を割り当てることで、ユーザ認証を強化できます。強力な RADIUS 共有シークレットは、大文字と小文字、数字、句読点のランダムなシーケンスで構成され、長さは 22 文字以上です。可能であれば、ランダムな文字生成プログラムを使用して RADIUS 共有秘密を特定します。

RADIUS トラフィックをさらに保護するには、各アプライアンスまたは仮想サーバに異なる共有シークレットを割り当てます。RADIUS サーバでクライアントを定義する場合、各クライアントに個別の共有シークレットを割り当てることもできます。また、RADIUS 認証を使用する各ポリシーを個別に構成する必要があります。

IP アドレス抽出を構成する

RADIUS サーバから IP アドレスを抽出するようにアプライアンスを構成できます。ユーザが RADIUS サーバで認証されると、サーバはユーザに割り当てられたフレーム IP アドレスを返します。IP アドレス抽出の属性を次に示します。

- リモート RADIUS サーバが、アプライアンスにログオンしたユーザーの内部ネットワークからの IP アドレスを提供できるようにします。
- IP アドレスタイプを使用する任意の RADIUS 属性の設定（ベンダーエンコードを含む）を許可します。

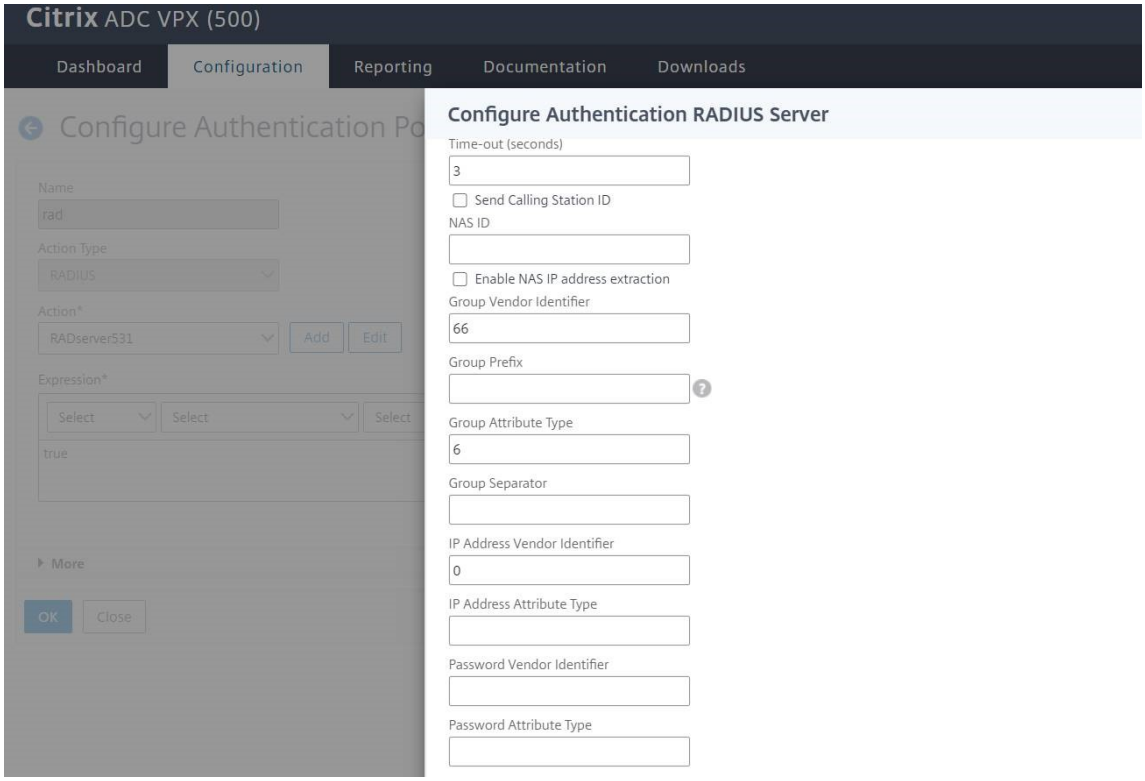
IP アドレス抽出用に RADIUS サーバを設定する場合は、ベンダー ID と属性タイプを設定します。

ベンダー識別子により、RADIUS サーバは、RADIUS サーバ上で設定されている IP アドレスのプールから IP アドレスをクライアントに割り当てることができます。ベンダー ID と属性は、RADIUS クライアントと RADIUS サーバ間のアソシエーションを作成するために使用されます。ベンダー ID は、内部ネットワークの IP アドレスを提供する RADIUS 応答の属性です。値 0 は、属性がベンダーエンコードされていないことを示します。属性タイプは、RADIUS 応答のリモート IP アドレス属性です。最小値は 1 で、最大値は 255 です。

一般的な設定では、**RADIUS** 属性 フレーム付き *IP* アドレスを抽出します。ベンダー ID がゼロに設定されているか、指定されていません。属性タイプは 8 に設定されています。

GUI を使用した **RADIUS** のグループ抽出

1. システムに移動 > 認証 > 高度なポリシー > 半径、およびポリシーを選択します。
2. RADIUS ポリシーを選択または作成します。
3. [認証 **RADIUS** サーバーの構成] ページで、次のパラメーターを設定します。
   a) グループのベンダー識別子
   b) グループの属性の種類
4. [**OK]** をクリックして [閉じる] をクリックします。



TACACS + 認証（外部 **TACACS +** サーバーを使用）

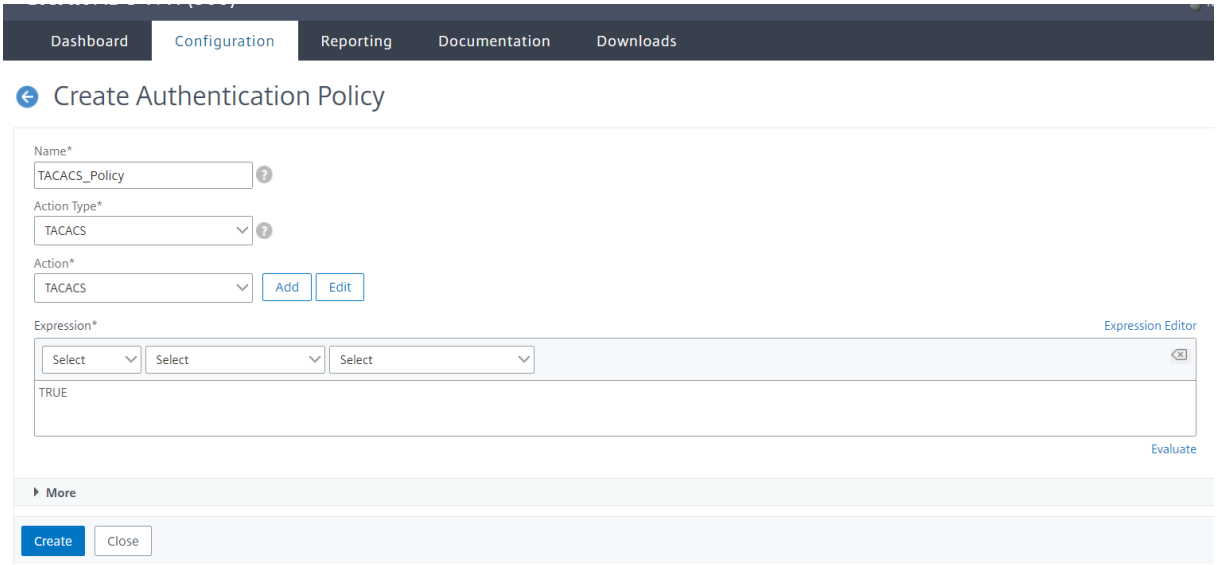
> 重要
>
> - 「clear ns config」コマンドを実行するときは、TACACS 関連の設定を変更しないことをお勧めします。
> - 詳細ポリシーに関する TACACS 関連の設定は、詳細ポリシーの「clear ns config」コマンドで`RBAconfig`パラメータが NO に設定されると、クリアされ、再適用されます。

TACACS+ サーバを認証用に設定できます。RADIUS 認証と同様に、TACACS+ は秘密キー、IP アドレス、およびポート番号を使用します。デフォルトのポート番号は 49 です。TACACS+ サーバを使用するようにアプライアンスを設定するには、サーバの IP アドレスと TACACS+ シークレットを指定します。使用中のサーバーのポート番号がデフォルトのポート番号 49 以外の場合にのみ、ポートを指定する必要があります。

詳細については、TACACS 認証を参照してください。

GUI を使用して **TACACS +** 認証を構成します

1. **System** > **Authentication** > **Advanced Policies** > **Policy** に移動します。
2. [追加] をクリックして、タイプ TACACS の認証ポリシーを作成します。
3. **[**作成**]** して **[**閉じる**]** をクリックします。



アプライアンスで TACACS+ サーバ設定を構成したら、ポリシーをシステムグローバルエンティティにバインドします。

CLI を使用して、認証ポリシーをシステムグローバルエンティティにバインドします

認証ポリシーが構成されたら、ポリシーをシステムグローバルエンティティにバインドします。

コマンドラインプロンプトで、次の操作を実行します。

```
bind system global <policyName> [-priority <positive_integer>]
```

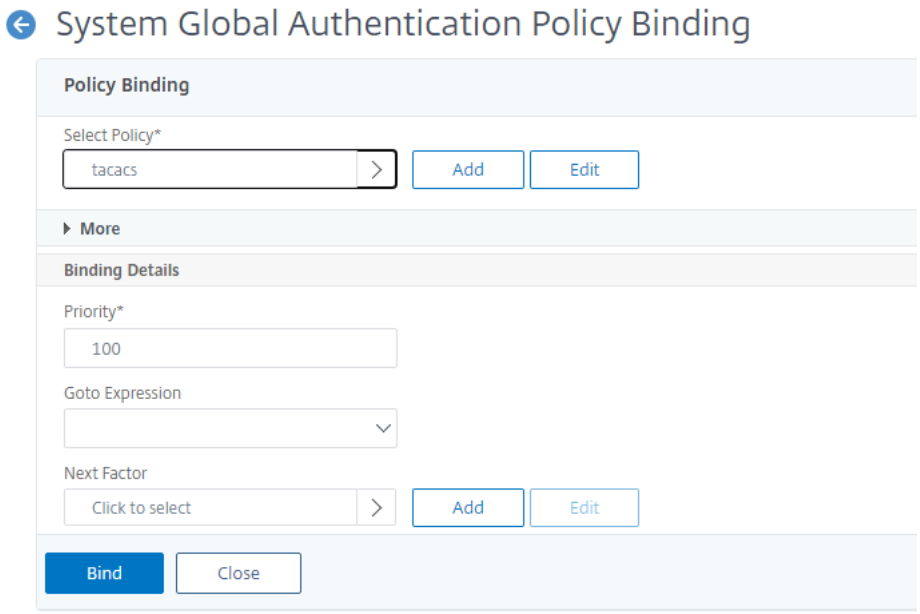
例:

```
bind system global pol_classic -priority 10
```

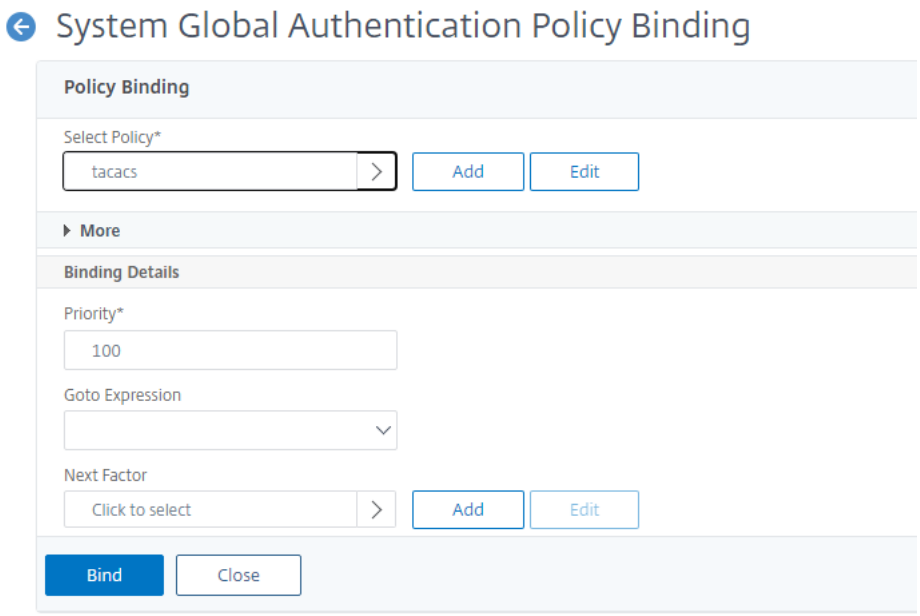
また、TACACS を使用した外部認証については、Citrix x の記事 CTX113820 を参照してください。

GUI を使用して、RADIUS 認証用に認証ポリシーをシステムグローバルにバインドします

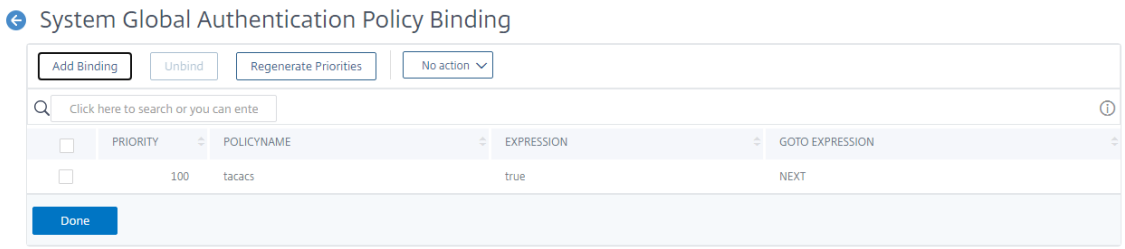
1. **System** > **Authentication** > **Advanced Policies** > **Authentication Policies > Policy** に移動します。
2. 詳細ペインで、[グローバルバインディング] をクリックして、システムグローバル認証ポリシーバインディングを作成します。
3. **Global Bindings** をクリックします。



4. TACACS ポリシーを選択します。
5. System Global Authentication Policy Binding ページで次のパラメーターを設定します。
   a) ポリシーを選択します。
   b) バインディングの詳細



6. [バインドして 閉じる] をクリックします。
7. [グローバルバインディング] をクリックして、システムグローバルにバインドされているポリシーを確認します。



TACACS グループの抽出の詳細については、Citrix x の記事 CTX220024を参照してください。

外部ユーザーのログオン試行の失敗回数を表示する

Citrix ADC アプライアンスは、Citrix ADC 管理コンソールに正常にログオンする前に、少なくとも 1 回失敗したログインを試行すると、外部ユーザーに無効なログイン試行の数を表示します。

> 注
>
> 現在、Citrix は、外部ユーザーに対してシステムパラメータで”persistentLoginAttempts” パラメータを有効にしたキーボードインタラクティブ認証のみをサポートしています。

コマンドプロンプトで入力します。

```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <value> -
persistentLoginAttempts (ENABLED | DISABLED )]
```

例:

```
set aaa parameter -maxloginAttempts 5 -failedLoginTimeout 4 -persistentLoginAttempts
 ENABLED
```

```
1  Following msg will be seen to external user when he tries 1 invalid
       login attempt before successfully login to the ADC management access
       .
2
3  Connection established.
4  To escape to local shell, press 'Ctrl+Alt+]'.
5  ##############################################################################

6  #

       #
7  #         WARNING: Access to this system is for authorized users only
              #
8  #          Disconnect IMMEDIATELY if you are not an authorized user!
               #
```



```
9  #
       #
10 ################################################################################
11
12
13 WARNING! The remote SSH server rejected X11 forwarding request.
14 Last login: Mon Aug 24 17:09:00 2020 from 10.10.10.10
15
16 The number of unsuccessful login attempts since the last successful
       login : 1
17 Done
18 >
19 The number of unsuccessful login attempts since the last successful
       login : 1
20 Done
21 >
22 <!--NeedCopy-->
```

ローカルシステムユーザの **SSH** キーベース認証

October 7, 2021

Citrix ADC アプライアンスの安全なユーザーアクセスを取得するには、SSH サーバーの公開鍵認証を使用できます。SSH キーベース認証は、次の理由から、従来のユーザ名/パスワードタイプの認証よりも優先されます。

- ユーザーパスワードよりも優れた暗号強度を提供します。
- 複雑なパスワードを覚えておく必要がなくなり、パスワードを使用した場合に発生する可能性のあるショルダーサーフィン攻撃を防ぎます。
- 自動化シナリオをより安全にするためのパスワードなしのログインを提供します。

Citrix ADC は、公開鍵と秘密鍵の概念を適用することにより、SSH 鍵ベースの認証をサポートします。Citrix ADC の SSH キーベースの認証は、特定のユーザーまたはすべてのローカルユーザーに対して有効にできます。

> 注
>
> この機能は、Citrix ADC ローカルユーザーに対してのみサポートされており、外部ユーザーに対してはサポートされていません。

ローカルシステムユーザの **SSH** キーベース認証

Citrix ADC アプライアンスでは、管理者はセキュアなシステムアクセスのために SSH キーベース認証を設定できます。ユーザーが秘密鍵を使用して Citrix ADC にログインすると、システムはアプライアンスで構成された公開鍵を使用してユーザーを認証します。

ローカルシステムユーザの **SSH** キーベース認証の設定

次の構成は、Citrix ADC ローカルシステムユーザーのキーベースの認証を構成するのに役立ちます。

1. 管理者の資格情報を使用して Citrix ADC アプライアンスにログオンします。
2. デフォルトでは、sshd_config ファイルは次のパスにアクセスします: **AuthorizedKeysFile**/nsconfig/ssh/authorized_keys 。
3. 公開鍵を authorized_keys ファイルに追加します: **/nsconfig/ssh/authorized_keys **。sshd_config のファイルパスは /etc/sshd_configです。
4. sshd_config ファイルを /nsconfig にコピーして、アプライアンスを再起動した後も変更が保持されるようにします。
5. 次のコマンドを使用して、sshd プロセスを再開できます。

```
    kill -HUP `cat /var/run/sshd.pid`
<!--NeedCopy-->
```

> 注
>
> authorized_keys ファイルが利用できない場合は、最初に作成してから公開鍵を追加する必要があります。ファイルに **authorized_keys** に対する次の権限があることを確認してください。
>
> ```
> root@Citrix ADC## chmod 0644 authorized_keys
> ```

```
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
    1994
The Regents of the University of California. All rights reserved.
root@ns# cd /nsconfig/ssh
root@ns# vi authorized_keys
### Add public keys in authorized_keys file
<!--NeedCopy-->
```


ローカルシステムユーザー向けのユーザー固有の **SSH** キーベースの認証

Citrix ADC アプライアンスでは、管理者は、セキュリティで保護されたシステムアクセス用にユーザー固有の SSH キーベースの認証を設定できるようになりました。管理者は、最初に sshd_config ファイルで Authorizedkeysfile オプションを構成してから、システムユーザーの authorized_keys ファイルに公開鍵を追加する必要があります。

> 注
>
> authorized_keys ファイルがユーザーに使用できない場合、管理者はまずファイルを作成してから、そのファイルに公開キーを追加する必要があります。

CLI を使用して、ユーザー固有の **SSH** キーベースの認証を構成します

次の手順は、Citrix ADC ローカルシステムユーザーのユーザー固有の SSH キーベースの認証を構成するのに役立ちます。

1. 管理者の資格情報を使用して Citrix ADC アプライアンスにログオンします。
2. シェルプロンプトで、sshd_config ファイルにアクセスし、次の構成行を追加します。

   ```
   AuthorizedKeysFile ~/.ssh/authorized_keys
   ```

   > 注
   >
   > ~ はホームディレクトリで、ユーザーによって異なります。別のホームディレクトリに展開されます。

3. ディレクトリをシステムユーザーフォルダに変更し、authorized_keys ファイルに公開鍵を追加します。

   ```
   /var/pubkey/<username>/.ssh/authorized_keys
   ```

前の手順を完了したら、次のコマンドを使用して、アプライアンスで sshd プロセスを再起動します。

```
1     kill -HUP `cat /var/run/sshd.pid`
2
3 <!--NeedCopy-->
```

> 注
>
> authorized_keys ファイルが利用できない場合は、まず作成してから公開鍵を追加する必要があります。

```
1 > shell
2 Copyright (c) 1992-2013 The FreeBSD Project.
3 Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
      1994
4 The Regents of the University of California. All rights reserved.
```



```
root@ns# cd /var/pubkey/<username>/
root@ns# ls
.ssh
root@ns# cd .ssh
root@ns# vi authorized_keys
### Add public keys in authorized_keys file

<!--NeedCopy-->
```

また、Citrix ADC アプライアンスへのセキュアな SSH アクセスの仕組みについては、Citrix の記事 CTX109011 を読んでください 。

システムユーザーと外部ユーザーの 2 要素認証

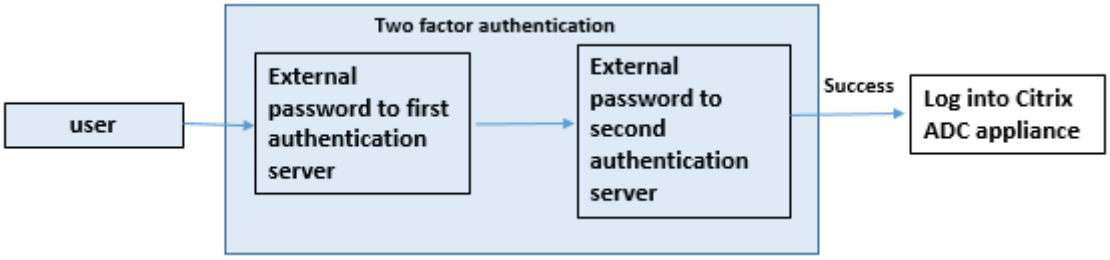
January 31, 2022

二要素認証は、Citrix ADC アプライアンスが 2 つのオーセンティケーターレベルでシステムユーザーを認証するセキュリティメカニズムです。アプライアンスは、両方の認証レベルによるパスワードの検証に成功した後にのみ、ユーザーにアクセスを許可します。ユーザーがローカルで認証される場合、ユーザープロファイルは Citrix ADC データベースに作成する必要があります。ユーザが外部で認証される場合、ユーザ名とパスワードは外部認証サーバに登録されているユーザ ID と一致する必要があります。

> 注:
>
> 2 要素認証機能は、Citrix ADC 12.1 ビルド 51.16 以降で動作します。

二要素認証のしくみ

ユーザーが Citrix ADC アプライアンスにログオンしようとしているとします。要求されたアプリケーションサーバーは、最初の外部認証サーバー（RADIUS、TACACS、LDAP、または AD）にユーザー名とパスワードを送信します。ユーザ名とパスワードが検証されると、ユーザは第 2 レベルの認証を要求されます。これで、ユーザーは 2 番目のパスワードを入力できます。両方のパスワードが正しい場合にのみ、ユーザーは Citrix ADC アプライアンスにアクセスできます。次の図は、Citrix ADC アプライアンスで 2 要素認証がどのように機能するかを示しています。



外部ユーザとシステムユーザの 2 要素認証を設定する場合のさまざまな使用例を次に示します。

Citrix ADC アプライアンスで 2 要素認証をさまざまな方法で構成できます。以下は、Citrix ADC アプライアンスでの 2 要素認証のさまざまな構成シナリオです。

1. Citrix ADC、GUI、CLI、API、SSH にわたる 2 要素認証（2FA）。
2. システムユーザに対して外部認証が有効になり、ローカル認証が無効になります。
3. 外部認証は、システムユーザのポリシーベースのローカル認証で有効になっています。
4. ローカル認証が有効になっているシステムユーザに対して、外部認証が無効になっています。
5. 外部認証を有効にし、システムユーザに対してローカル認証を有効にします。
6. 選択した LDAP ユーザーに対して有効な外部認証

使用事例 **1: Citrix ADC**、**GUI**、**CLI**、**API**、および **SSH** インターフェイス全体での **2** 要素認証（**2FA**）

2 要素認証が有効になっており、GUI、API、および SSH のすべての Citrix ADC 管理アクセスで使用できます。

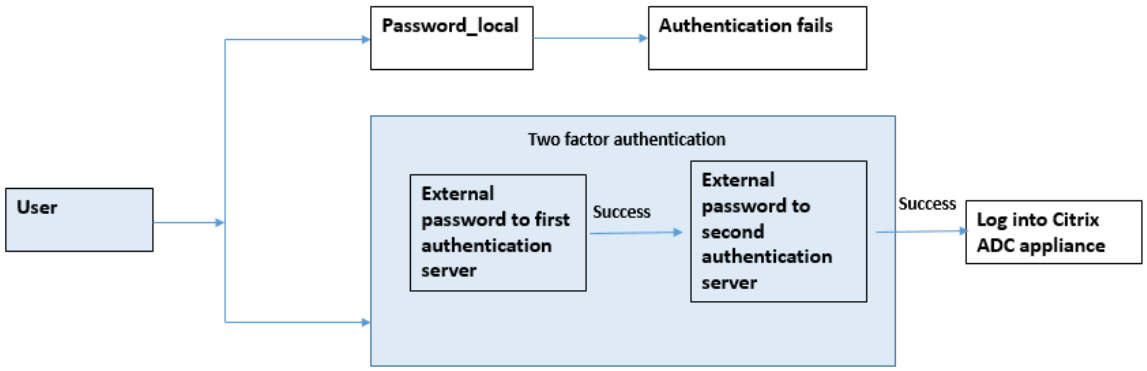
使用事例 **2: LDAP**、**RADIUS**、**Active Directory**、**TACACS** などの外部認証サーバーでサポートされる **2** 要素認証

次の外部認証サーバで、第 1 レベルおよび第 2 レベルのユーザ認証用に 2 要素認証を設定できます。

- RADIUS
- LDAP
- Active Directory
- TACACS

使用事例 **3:** システムユーザーに対して外部認証を有効にし、ローカル認証を無効にする

認証プロセスを開始するには、外部認証オプションを有効にし、システムユーザのローカル認証を無効にします。



コマンドラインインターフェイスを使用して、次の手順を実行します。

1. LDAP ポリシーの認証アクションを追加する
2. LDAP ポリシーの認証ポリシーの追加
3. RADIUS ポリシーの認証アクションを追加する

4. RADIUS ポリシーの認証ポリシーの追加
5. 認証ログインスキーマの追加
6. RADIUS サーバへの認証ポリシー・ラベルの追加とバインド
7. LDAP ポリシーのバインドシステムグローバル認証
8. システムパラメータでローカル認証を無効にする

LDAP サーバーの認証アクションを追加 **(**第 **1** レベル認証**)**

コマンドプロンプトで入力します。

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname <
loginname> -groupattrname <grp attribute name> -subAttributename <string>-
ssoNameAttribute <string>
```

例**:**

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

LDAP サーバの認証ポリシーの追加（第 **1** レベル認証）

コマンドプロンプトで入力します。

```
add authentication policy <ldap policy name> -rule true -action <ldap
action name>
```

例**:**

```
add authentication policy pol1 -rule true -action ldapact1
```

RADIUS サーバの認証アクションを追加する（第 **2** レベルの認証）

コマンドプロンプトで、次のように入力します。

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

例**:**

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

RADIUS サーバの認証ポリシーの追加（第 **2** レベルの認証）

コマンドプロンプトで入力します。


```
add authentication policy <radius policy name> -rule true -action <rad
action name>
```

例:

```
add authentication policy radpol11 -rule true -action radact1
```

認証ログインスキーマの追加

システムユーザーに「SingleAuth.xml」ログインスキーマを使用して、Citrix ADC アプライアンスの 2 番目のパスワードを入力できます。コマンドプロンプトで入力します。

```
add authentication loginSchema <login schema name> -authenticationSchema
LoginSchema/SingleAuth.xml
```

例:

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

RADIUS サーバへの認証ポリシー・ラベルの追加とバインド

コマンドプロンプトで入力します。

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```

```
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

例:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
```

```
bind authentication policylabel label1 -policyName radpol11 -priority 1
```

LDAP ポリシーのグローバルバインド認証システム

コマンドプロンプトで入力します。

```
bind system global ldappolicy -priority <priority> -nextFactor <policy
label name>
```

例:

```
bind system global pol11 -priority 1 -nextFactor label1
```

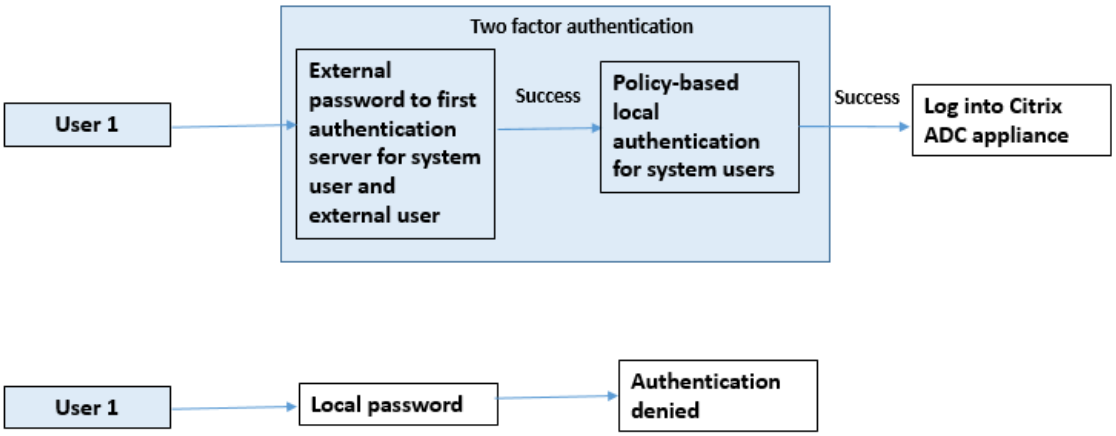

システムパラメータでローカル認証を無効にする

コマンドプロンプトで入力します。

```
set system parameter -localauth disabled
```

使用事例 **4:** ローカル認証ポリシーが添付されたシステムユーザーに対して外部認証が有効になっている

このシナリオでは、ユーザー識別の 2 番目のレベルで、ローカル認証ポリシーの評価による 2 要素認証を使用してアプライアンスにログオンできます。



コマンドラインインターフェイスを使用して、次の手順を実行します。

1. LDAP サーバーの認証アクションを追加する
2. LDAP ポリシーの認証ポリシーの追加
3. ローカル認証ポリシーの追加
4. 認証ポリシーラベルの追加
5. LDAP ポリシーをシステムグローバルとしてバインド
6. システムパラメータでローカル認証を無効にする

LDAP サーバーの認証アクションを追加 **(**第 **1** レベル認証**)**

コマンドプロンプトで入力します。

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname <
loginname> -groupattrname <grp attribute name> -subAttributename <string>-
ssoNameAttribute <string>
```

例**:**


```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name -ssoNameAttribute name
```

LDAP サーバの認証ポリシーの追加（第 **1** レベル認証）

コマンドプロンプトで入力します。

```
add authentication policy <ldap policy name> -rule true -action <ldap
action name>
```

例**:**

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

システムユーザーのローカル認証ポリシーの追加 **(**第 **2** レベルの認証**)**

コマンドプロンプトで入力します。

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type
```

例**:**

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

認証ポリシーラベルの追加とバインド

コマンドプロンプトで入力します。

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

例**:**

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel label1 -policyName radpol11 -priority 1 -
gotoPriorityExpression NEXT
```

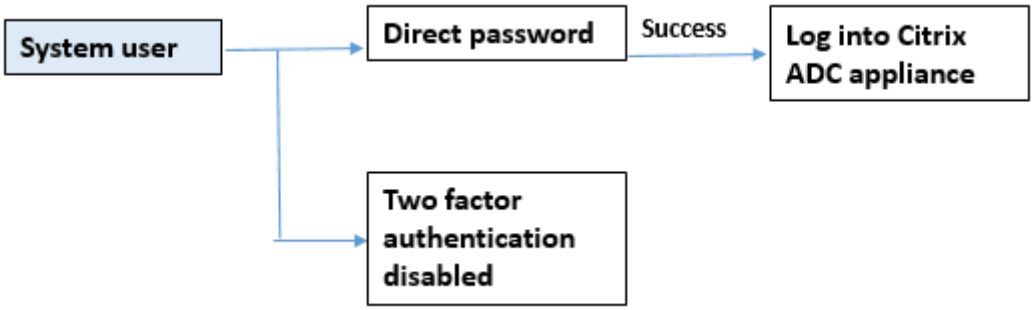

システムパラメータでローカル認証を無効にする

コマンドプロンプトで入力します。

```
set system parameter -localauth disabled
```

使用事例 **5:** システムユーザーに対して外部認証を無効にし、ローカル認証を有効にする

ユーザーが「externalAuth」を無効にしている場合は、ユーザーが認証サーバーに存在しないことを示しています。同じユーザー名のユーザーが外部認証サーバーに存在する場合でも、ユーザーは外部認証サーバーで認証されません。ユーザーはローカルで認証されます。



システムユーザーのパスワードを有効にし、外部認証を無効にするには

コマンドプロンプトで、次のように入力します。

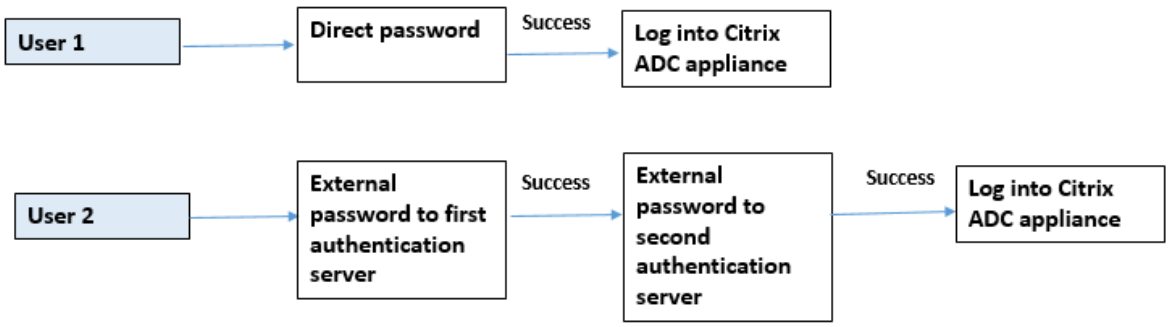
```
add system user <name> <password> -externalAuth DISABLED
```

例**:**

```
add system user user1 password1 -externalAuth DISABLED
```

使用事例 **6:** システムユーザーに対して外部認証が有効で、ローカル認証が有効

ローカルパスワードを使用してシステムユーザーを認証するようにアプライアンスを構成します。この認証が失敗した場合、ユーザーは 2 つのレベルの外部認証サーバーで外部認証パスワードを使用して認証されます。



CLI を使用して、次の手順を構成します。

1. LDAP サーバーの認証アクションを追加する
2. LDAP ポリシーの認証ポリシーの追加
3. RADIUS ポリシーの認証アクションを追加する
4. RADIUS ポリシーの認証ポリシーの追加
5. 認証ログインスキーマの追加
6. 認証ポリシーラベルの追加
7. ログインスキーマの認証ポリシーラベルのバインド
8. RADIUS ポリシーのグローバルバインド認証システム
9. LDAP ポリシーのグローバルバインド認証システム

LDAP サーバーの認証アクションを追加する

コマンドプロンプトで入力します。

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
ssoNameAttribute <>
```

例**:**

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

LDAP ポリシーの認証ポリシーの追加

コマンドプロンプトで入力します。

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```


例:

```
add authentication policy pol1 -rule true -action ldapact1
```

RADIUS サーバーの認証アクションを追加する

コマンドプロンプトで入力します。

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

例:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

RADIUS サーバーの高度な認証ポリシーを追加する

コマンドプロンプトで入力します。

```
add authentication policy <policy name> -rule true -action <rad action name
>
```

例:

```
add authentication policy radpol11 -rule true -action radact1
```

認証ログインスキーマの追加

SingleAuth.xml ログインスキーマを使用して、ログインページを表示し、第 2 レベルの認証でシステムユーザーを認証できます。

コマンドプロンプトで入力します。

```
add authentication loginSchema <name> -authenticationSchema <string>
```

例:

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

ユーザーログイン用の **RADIUS** 認証ポリシーに認証ポリシーラベルを追加してバインドします

コマンドプロンプトで入力します。

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```


例:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

例:

```
bind authentication policylabel label1 -policyName rad pol11 -priority 1
```

認証ポリシーをグローバルにバインド

コマンドプロンプトで入力します。

```
bind system global [<policyName> [-priority <positive_integer>] [-nextFactor
 <string>] [-gotoPriorityExpression <expression>]]
```

例:

```
bind system global radpol11 -priority 1 -nextFactor label11
```

使用事例 **7:** 選択した外部ユーザーに対してのみ外部認証を有効にする

LDAP アクションで構成された検索フィルターに従って、選択的な外部ユーザーを 2 要素認証で構成し、他のシステムユーザーは単一要素認証を使用して認証されます。

CLI を使用して、次の手順を構成します。

1. LDAP サーバーの認証アクションを追加する
2. LDAP ポリシーの認証ポリシーの追加
3. RADIUS ポリシーの認証アクションを追加する
4. RADIUS ポリシーの認証ポリシーの追加
5. 認証ログインスキーマの追加
6. 認証ポリシーラベルの追加
7. ログインスキーマの認証ポリシーラベルのバインド
8. RADIUS ポリシーのグローバルバインド認証システム

LDAP サーバーの認証アクションを追加する

コマンドプロンプトで入力します。

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
ssoNameAttribute <>
```


例:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

LDAP ポリシーの認証ポリシーの追加

コマンドプロンプトで入力します。

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```

例:

```
add authentication policy pol1 -rule true -action ldapact1
```

RADIUS サーバーの認証アクションを追加する

コマンドプロンプトで入力します。

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

例:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

RADIUS サーバーの高度な認証ポリシーを追加する

コマンドプロンプトで入力します。

```
add authentication policy <policy name> -rule true -action <rad action name
>
```

例:

```
add authentication policy radpol11 -rule true -action radact1
```

認証ログインスキーマの追加

SingleAuth.xml ログインスキーマを使用して、アプライアンスのログインページを提供し、第 2 レベルの認証でシステムユーザーを認証できます。

コマンドプロンプトで入力します。

```
add authentication loginSchema <name> -authenticationSchema <string>
```


例:

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

ユーザーログイン用の **RADIUS** 認証ポリシーに認証ポリシーラベルを追加してバインドします

コマンドプロンプトで入力します。

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```

例:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

例:

```
bind authentication policylabel label1 -policyName radpol11 -priority
```

認証ポリシーをグローバルにバインド

コマンドプロンプトで入力します。

```
bind system global [<policyName> [-priority <positive_integer>] [-nextFactor
 <string>] [-gotoPriorityExpression <expression>]]
```

例:

```
bind system global radpol11 -priority 1 -nextFactor label11
```

検索フィルターを使用してグループユーザーの 2 要素認証なしで構成するには:

1. LDAP サーバーの認証アクションを追加する
2. LDAP サーバーの認証ポリシーを追加する
3. LDAP サーバーのグローバルなバインド認証システム

LDAP サーバーの認証アクションを追加する

コマンドプロンプトで入力します。

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
searchFilter<>
```


例:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name - searchFilter ”memberOf=CN=grp4,CN=Users,DC=
aaatm-test,DC=com”
```

LDAP サーバーの認証ポリシーを追加する

コマンドプロンプトで入力します。

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```

例:

```
add authentication policy pol1 -rule true -action ldapact1
```

LDAP ポリシーのグローバルバインド認証システム

コマンドプロンプトで入力します。

```
bind system global ldappolicy -priority <priority> -nextFactor <policy
label name>
```

例:

```
bind system global pol11 -priority 1 -nextFactor label11
```

二要素認証用にカスタマイズされたプロンプトメッセージを表示する

/flash/nsconfig/loginschema/LoginSchemaで SingleAuth.xml ファイルで 2 段階のパスワードフィールドを構成する場合

以下は、SingleAuth.xml ファイルのスニペットです。‘SecondPassword:’ 2 番目のパスワードを入力するようにユーザーに求められる 2 番目のパスワードフィールド名です。

```
1 <?xml version=”1.0” encoding=”UTF-8”?>
2 <AuthenticateResponse xmlns=”http://citrix.com/authentication/response
     /1”>
3 <Status>success</Status>
4 <Result>more-info</Result>
5 <StateContext/>
6 <AuthenticationRequirements>
7 <PostBack>/nf/auth/doAuthentication.do</PostBack>
```



```
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</
    SaveID><Type>username</Type></Credential><Label><Text>
    singleauth_user_name</Text><Type>nsg-login-label</Type></Label><
    Input><AssistiveText>singleauth_please_supply_either_domain\
    username_or_user@fully.qualified.domain</AssistiveText><Text><Secret
    >false</Secret><ReadOnly>false</ReadOnly><InitialValue/><Constraint
    >.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password
    </SaveID><Type>password</Type></Credential><Label><Text>
    SecondPassword:</Text><Type>nsg-login-label</Type></Label><Input><
    Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue/><
    Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>
    singleauth_first_factor</Text><Type>nsg_confirmation</Type></Label><
    Input/></Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
    </Type></Credential><Label><Text>singleauth_remember_my_password</
    Text><Type>nsg-login-label</Type></Label><Input><CheckBox><
    InitialValue>false</InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
    ><Label><Type>none</Type></Label><Input><Button>singleauth_log_on</
    Button></Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

Citrix ADC GUI を使用した 2 要素認証の構成

1. Citrix ADC アプライアンスにログオンします。
2. [システム] > [認証] > [詳細ポリシー] > [ポリシー **] に移動します。
3. [追加] をクリックして、第 1 レベルの認証ポリシーを作成します。
4. [認証ポリシーの作成] ページで、次のパラメータを設定します。
    a) Name：ポリシーの名前
    b) アクションタイプ。アクションタイプを LDAP、Active Directory、RADIUS、TACACS などとして選択します
    c) 操作。ポリシーに関連付ける認証アクション（プロファイル）。既存の認証アクションを選択するか、プラスをクリックして適切なタイプのアクションを作成できます。

d) 式。高度なポリシー式を提供。

5. [作成] をクリックし、[閉じる] をクリックします。

a) 式。高度なポリシー式を提供。

6. ［作成］をクリックします。

7. [**Add**] をクリックして、第 2 レベルの認証ポリシーを作成します。

8. [認証ポリシーの作成**]** ページで、次のパラメータを設定します。

a) Name：ポリシーの名前
b) アクションタイプ。アクションタイプを LDAP、Active Directory、RADIUS、TACACS などとして選択します
c) 操作。ポリシーに関連付ける認証アクション（プロファイル）。既存の認証アクションを選択するか、[+ アイコンを使用して、適切なタイプのアクションを作成します。
d) 式。高度なポリシー表現を提供する

9. [作成] をクリックし、[閉じる] をクリックします。

a) 式。高度なポリシー式を提供。

10. ［作成］をクリックします。

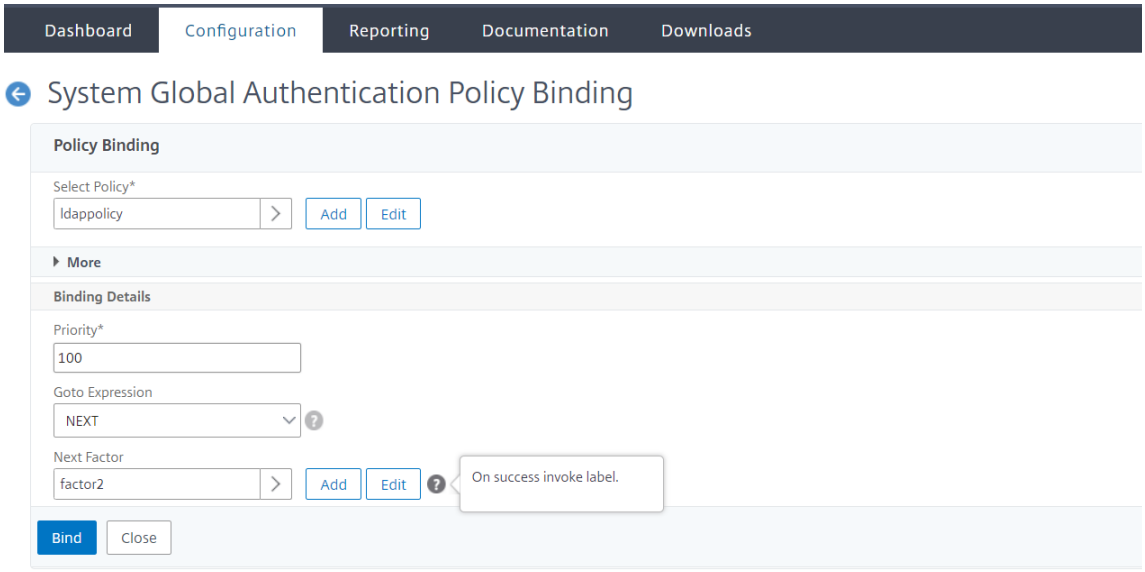
11. [認証ポリシー] ページで、[グローバルバインド] をクリックします。

12. [グローバル認証ポリシーバインディングの作成**]** ページで、第 1 レベルの認証ポリシーを選択し、[バインディングの追加] をクリックします。

13. [ポリシーバインディング] ページで、認証ポリシーを選択し、次のポリシーバインディングパラメータを設定します。

a) 次の要因。第 2 レベルの認証ポリシーラベルを選択します。

14. [バインドして 閉じる] をクリックします。



15. ［完了］をクリックします。
16. 第 2 レベルの認証のために Citrix ADC アプライアンスにログオンします。これで、ユーザーは 2 番目のパスワードを入力できます。両方のパスワードが正しい場合にのみ、ユーザーは Citrix ADC アプライアンスにアクセスできます。

> 注
>
> 2 要素認証用に設定された TACACS は、で有効にした場合でも、許可とアカウンティングをサポートしません。“ tacacsAction “ コマンド。2 番目の要素は、認証目的でのみ使用されます。

また、Citrix ADC nFactor 認証のトピックの「二要素認証」を参照してください。

Citrix ADC 管理インターフェースへの制限されたシステムユーザー認証

October 7, 2021

CLI や API などの特定の Citrix ADC 管理インターフェイスへのシステムユーザーアクセスを制限できます。`allowedManagementInterface`パラメータは、許可される管理インターフェイスのリストを定義します。たとえば、ユーザーまたはグループの管理インターフェイスが API に設定されている場合、グループ内のすべてのユーザーは CLI 経由ではなく、API 経由で Citrix ADC にアクセスできます。ただし、Citrix ADC GUI は API インターフェースの一部であり、API 権限を持つユーザーも GUI インターフェースにアクセスできます。

> 注:
>
> デフォルトでは、ユーザーとグループはすべてのインターフェース（CLI、API、および GUI）にアクセスできます。

パラメータは、ユーザレベルまたはユーザグループレベルのいずれかで設定できます。グループレベルで設定する場合、設定はグループ内のすべてのユーザアカウントに適用されます。ユーザーが複数のグループにバインドされている場合、アプライアンスは管理インターフェイスの集約セットへのアクセスを許可します。ユーザーレベルでパラメーターを構成することにより、グループ内のユーザーの設定を指定できます。この場合–ユーザーレベルの設定はグループに対して構成されます。
特定のシナリオでは、ユーザーが外部認証サーバーを使用してユーザー・アカウントを管理している場合、サーバーの詳細がアプライアンス上で構成されます。この場合、管理者は Citrix ADC アプライアンスにユーザーグループを作成し、すべてのユーザー（外部サーバーでグループ化）をグループに追加できます。たとえば、外部サーバーで管理されているすべてのユーザーが API_users グループに追加され、管理者はアプライアンス上でローカルにグループを設定できます。

> 注:
>
> Citrix ADC アプライアンスでは、`nsroot` 管理者（スーパーユーザー）のみがパラメーターを構成でき、システムユーザーがパラメーター設定を変更することはできません。

CLI を使用して、Citrix ADC 管理インターフェイスへのユーザーアクセスを構成します

特定の管理インターフェイスへのユーザーアクセスを許可するには、許可された管理インターフェイスパラメータを設定する必要があります。コマンドプロンプトで入力します。

```
set system group <groupName> [-allowedManagementInterface ( CLI | API )]
```

例:

```
set system group network_usergroup -allowedManagementInterface CLI
```

パラメータの説明については、「認証および認可コマンドのリファレンス」を参照してください。

Citrix GUI および CLI インターフェイスについては、Citrix ADC にアクセスするトピックを参照してください。

TCP 構成

June 24, 2022

Citrix ADC アプライアンスの TCP 構成は、TCP 設定の集合である TCP プロファイルと呼ばれるエンティティで指定できます。TCP プロファイルは、これらの TCP 設定を使用するサービスまたは仮想サーバに関連付けることができます。

デフォルトの TCP プロファイルを設定して、すべてのサービスおよび仮想サーバにグローバルに適用される TCP 設定を設定できます。

> 注:
>
> TCP パラメータがサービス、仮想サーバ、およびグローバルに異なる値を持つ場合、最も具体的なエンティテ

> ィ（サービス）の値が最も優先されます。Citrix ADC アプライアンスは、TCP を構成するための他のアプローチも提供します。詳細については読んでください。

サポートされる **TCP** 設定

Citrix ADC アプライアンスは、次の TCP 機能をサポートしています。

なりすまし攻撃に対する **TCP** の防御

Citrix ADC によるウィンドウ減衰の実装は、RFC 4953 に準拠しています。

明示的な輻輳通知（**ECN**）

アプライアンスは、ネットワークの輻輳ステータスの通知をデータの送信者に送信し、データの輻輳またはデータ破損に対する是正措置を講じます。ECN の Citrix ADC の実装は、RFC 3168 に準拠しています。

タイムスタンプオプションを使用したラウンドトリップ時間測定 **(RTTM)**

TimeStamp オプションが機能するためには、接続の少なくとも片側（クライアントまたはサーバ）がそれをサポートしている必要があります。`TimeStamp`オプションの Citrix ADC の実装は、RFC 1323 に準拠しています。

不適切な再送信の検出

これは、TCP 重複選択確認応答 (D-SACK) およびフォワード RTO 回復 (F-RTO) を使用して実行できます。スプリアス再送信がある場合、輻輳制御設定は元の状態に戻ります。D-SACK の Citrix ADC 実装は RFC 2883 に準拠しており、F-RTO は RFC 5682 に準拠しています。

輻輳制御

この機能は、New-Reno、BIC、CUBIC、Nile、TCP ウエストウッドのアルゴリズムを使用します。

ウィンドウスケーリング

これにより、**TCP** 受信ウィンドウのサイズが最大値 65,535 バイトを超えて増加します。

ウィンドウスケーリングを構成する前に考慮すべきポイント

- スケールファクターには高い値を設定しないでください。これは、アプライアンスとネットワークに悪影響を及ぼす可能性があるためです。
- ウィンドウサイズを変更する理由が明確にわかっていない限り、ウィンドウのスケーリングは構成しません。
- TCP 接続内の両方のホストは、接続の確立時にウィンドウスケールオプションを送信します。接続の片側のみでこのオプションが設定されている場合、ウィンドウの拡大/縮小は接続に使用されません。

- 同じセッションの各接続は、独立したウィンドウスケーリングセッションです。たとえば、クライアントの要求とサーバーの応答がアプライアンスを通過する場合、アプライアンスとサーバーの間でウィンドウスケーリングを行わずに、クライアントとアプライアンスの間でウィンドウスケーリングを行うことができます。

TCP 最大輻輳ウィンドウ

ウィンドウサイズは、ユーザーが設定可能なサイズです。デフォルト値は 8190 バイトです。

選択的確認応答 **(SACK)**

これは、データレシーバー（Citrix ADC アプライアンスまたはクライアント）を使用して、正常に受信されたすべてのセグメントについて送信者に通知します。

転送確認 **(FACK)**

この機能により、ネットワーク内の未処理のデータバイトの総数を明示的に測定し、送信者（Citrix ADC またはクライアント）が再送信タイムアウト時にネットワークに注入されるデータ量を制御できるようにすることで、TCP の輻輳を回避できます。

TCP 接続多重化

この機能により、既存の TCP 接続の再利用が可能になります。Citrix ADC アプライアンスは、確立された TCP 接続を再利用プールに保存します。クライアント要求が受信されるたびに、アプライアンスは再使用プール内の使用可能な接続をチェックし、接続が使用可能な場合は新しいクライアントに処理します。使用できない場合、アプライアンスはクライアント要求の接続を作成し、その接続を再利用プールに保存します。Citrix ADC は、HTTP、SSL、および DataStream 接続タイプの接続多重化をサポートしています。

動的受信バッファリング

これにより、メモリとネットワークの状態に基づいて受信バッファを動的に調整できます。

マルチパス **TCP** 接続

クライアントと Citrix ADC アプライアンス間のマルチパス TCP（MPTCP）接続。Citrix ADC アプライアンスとバックエンドサーバー間では、MPTCP 接続はサポートされていません。MPTCP の Citrix ADC 実装は、RFC 6824 および RFC 8684 に準拠しており、MPTCP バージョン 0 と 1 の両方をサポートしています。

コマンドラインインターフェイスを使用して、アクティブ MPTCP 接続やアクティブサブフロー接続などの MPTCP 統計情報を表示できます。

コマンドプロンプトで、次のコマンドのいずれかを入力して、MPTCP 統計情報の概要または詳細な概要を表示するか、統計情報の表示をクリアします。

1. `Stat MPTCP`
2. `Stat mptcp -detail`
3. `Clearstats basic`

> 注:
>
> MPTCP 接続を確立するには、クライアントと Citrix ADC アプライアンスの両方が同じ MPTCP バージョンをサポートしている必要があります。Citrix ADC アプライアンスをサーバーの MPTCP ゲートウェイとして使用する場合、サーバーは MPTCP をサポートする必要はありません。クライアントが新しい MPTCP 接続を開始すると、アプライアンスは SYN パケットの MP_CAPABALE オプションからクライアントの MPTPC バージョンを識別します。クライアントのバージョンがアプライアンスでサポートされているバージョンよりも高い場合、アプライアンスは SYN-ACK パケットの MP_CAPABALE オプションで最上位のバージョンを示します。その後、クライアントは下位バージョンにフォールバックし、ACK パケットの MP_CAPABALE オプションでバージョン番号を送信します。そのバージョンがサポート可能な場合、アプライアンスは MPTCP 接続を続行します。それ以外の場合、アプライアンスは通常の TCP にフォールバックします。Citrix ADC アプライアンスはサブフロー（MP_JOIN）を開始しません。アプライアンスは、クライアントがサブフローを開始することを想定しています。

MPTCP での追加アドレスアドバタイズ **(ADD_ADDR)** のサポート

MPTCP 展開では、追加の仮想サーバの IP アドレスを持つ IP セットにバインドされた仮想サーバがある場合、追加のアドレスアドバタイズメント（ADD_ADDR）機能によって、IP セットにバインドされた仮想サーバの IP アドレスがアドバタイズされます。クライアントは、アドバタイズされた IP アドレスに対してさらに多くの`MP-JOIN`サブフローを開始できます。

MPTCP ADD_ADDR 機能について覚えておくべきポイント

- `ADD_ADDR`オプションの一部として最大 10 個の IP アドレスを送信できます。`mptcpAdvertise` パラメータが有効で 10 個以上の IP アドレスがある場合、10 の IP アドレスをアドバタイズした後、アプライアンスは残りの IP アドレスを無視します。
- MP-CAPABLE サブフローがプライマリ仮想サーバの IP アドレスではなく、IP セット内のいずれかの IP アドレスに対して行われる場合、仮想サーバの IP アドレスに対して`mptcpAdvertise`パラメータが有効になっている場合、仮想サーバの IP アドレスがアドバタイズされます。

CLI を使用して、追加アドレスアドバタイズメント（**ADD_ADDR**）機能を設定し、より多くの **VIP** アドレスをアドバタイズする

IPv4 と IPv6 の両方のアドレスタイプに対して`MPTCP ADD_ADDR`機能を設定できます。一般に、複数の IPv4 および IPv6 IP を単一の IP セットに接続でき、このパラメータは任意の IP アドレスのサブセットで有効にできます。ADD_ADDR 機能では、「mptcapAdvertise」オプションが有効になっている IP アドレスのみがアドバタイズされ、IP セットの残りの IP アドレスは無視されます。
`ADD_ADDR`機能を設定するには、次の手順を実行します。

1. IP セットを追加します。
2. MPTCP アドバタイズを有効にして、タイプ仮想サーバ IP（VIP）の IP アドレスを追加します。
3. IP アドレスを IP セットにバインドします。
4. 負荷分散仮想サーバーで IP セットを構成します。

IP セットの追加

コマンドプロンプトで入力します。

```
add ipset <name> [-td <positive_integer>]
<!--NeedCopy-->
```

例**:**

```
add ipset ipset_1
<!--NeedCopy-->
```

MPTCP アドバタイズが有効になっているタイプの仮想サーバ **IP**（**VIP**）の **IP** アドレスを追加します

コマンドで次のように入力します。

```
add ns ip <IPAddress>@ <netmask> [-mptcpAdvertise ( YES | NO )] -type <
    type>
<!--NeedCopy-->
```

例**:**

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
```

IP アドレスを **IP** セットにバインドする

コマンドプロンプトで入力します。

```
bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

例**:**

```
bind ipset ipset_1 10.10.10.10
```


IP セットを負荷分散仮想サーバーに設定する

コマンドプロンプトで入力します。

```
set lb vserver <name> [-ipset  <string>]
<!--NeedCopy-->
```

例**:**

```
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

サンプル構成**:**

```
Add ipset ipset_1
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
bind ipset ipset_1 10.10.10.10
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

ADD_ADDR 機能を使用してアドバタイズする外部 **IP** アドレスを構成する

アドバタイズされた IP アドレスが外部エンティティによって所有されており、Citrix ADC アプライアンスが IP アドレスをアドバタイズする必要がある場合は、「mptcpAdvertise」パラメータを有効にして、状態パラメータと ARP パラメータを無効にする必要があります。

外部 IP アドレスをアドバタイズするようにADD_ADDRを設定するには、次の手順を実行します。

1. MPTCP アドバタイズを有効にして、タイプ仮想サーバ IP（VIP）の IP アドレスを追加します。
2. IP アドレスを IP セットにバインドします。
3. IP セットを負荷分散仮想サーバーとバインドする

MPTCP アドバタイズを有効にした仮想サーバ **IP**（**VIP**）タイプの外部 **IP** アドレスを追加する

コマンドプロンプトで入力します。

```
add ns ip <IPAddress>@ <External-IP-mask -type VIP> [-mptcpAdvertise (
    YES | NO )] -type <type> -state DISABLED -arp DISABLED
<!--NeedCopy-->
```


例:

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP -state
DISABLED -arp DISABLED
```

IP アドレスを **IP** セットにバインドする

コマンドプロンプトで入力します。

```
1 bind ipset <name> <IPAddress>
2 <!--NeedCopy-->
```

例:

```
bind ipset ipset_1 10.10.10.10
```

IP セットを負荷分散仮想サーバーに設定する

コマンドプロンプトで入力します。

```
1 set lb vserver <name> [-ipset  <string>]
2 <!--NeedCopy-->
```

例:

```
set lb vserver lb1 -ipset ipset_1
```

サンプル構成:

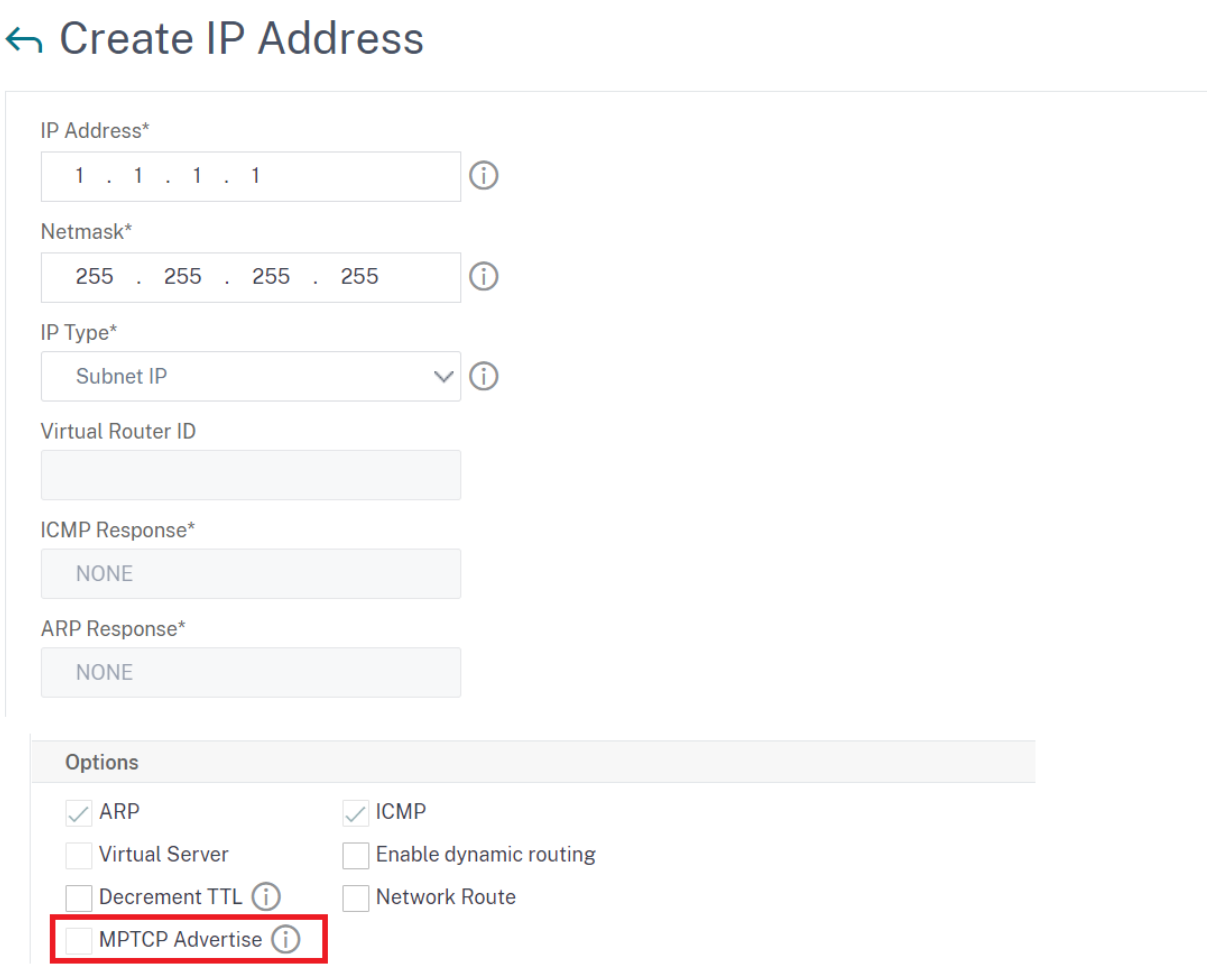
```
1 add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
     state DISABLED -arp DISABLED
2 bind ipset ipset_1 10.10.10.10
3 set lb vserver lb1 -ipset ipset_1
4 <!--NeedCopy-->
```

Citrix ADC GUI を使用して、**MPTCP** 対応クライアントに **IP** アドレスをアドバタイズする

MPTCP 対応クライアントに IP アドレスをアドバタイズするには、次の手順を実行します。

1. [システム] **>** [ネットワーク] **>** [**IP**] に移動します。
2. 詳細ウィンドウで、[追加] をクリックします。

3. [**IP** アドレスの作成] ページで、[**MPTCP** アドバタイズ] チェックボックスをオンにしてパラメータを設定します。デフォルトでは、無効になっています。



TCP/IP パスオーバーレイオプションの抽出とクライアント **IP HTTP** ヘッダーの挿入

TCP/IP パスオーバーレイを抽出し、クライアント IP HTTP ヘッダーを挿入します。オーバーレイネットワークを介したデータ転送では、接続終了または送信元クライアントの IP アドレスが失われるネットワークアドレス変換 (NAT) を使用することがよくあります。これを回避するために、Citrix ADC アプライアンスは TCP/IP パスオーバーレイオプションを抽出し、ソースクライアントの IP アドレスを HTTP ヘッダーに挿入します。ヘッダーに IP アドレスが含まれていると、Web サーバーは接続を確立したソースクライアントを識別できます。抽出されたデータは TCP 接続の存続期間にわたって有効であるため、ネクストホップホストがオプションを再度解釈する必要がなくなります。このオプションは、client-IP 挿入オプションが有効になっている Web サービスに対してのみ適用できます。

TCP セグメンテーションオフロード

TCP セグメンテーションを NIC にオフロードします。オプションを「自動」に設定すると、NIC がサポートされている場合、TCP セグメンテーションは NIC にオフロードされます。

TCP ハンドシェイクの **Cookie** をクライアントと同期させる

これは、SYN フラッド攻撃に抵抗するために使用されます。クライアントとの TCP `SYNCOOKIE` ハンドシェイクのメカニズムを有効または無効にできます。`SYNCOOKIE`を無効にすると、`SYN`が Citrix ADC アプライアンスでの攻撃保護を防止します。

MSS の学習により、アプライアンスに設定されているすべての仮想サーバーで **MSS** ラーニングが有効になります

サポート可能な **TCP** パラメータ

次の表に、Citrix ADC アプライアンスで構成された TCP パラメータとそのデフォルト値のリストを示します。

パラメーター	デフォルト値	説明
ウィンドウ管理		
TCP 遅延-ACK タイマー	100 ミリ秒	TCP 遅延 ACK のタイムアウト（ミリ秒単位）。
TCP 最小再送信タイムアウト（RTO）（ミリ秒）	1000 ミリ秒	10 ミリ秒単位で指定された最小再送信タイムアウト（ミリ秒単位）（10 で割った場合は整数である必要があります）
キープアライブプローブを開始する前の接続のアイドル時間	900 秒	アイドルタイムアウト時に TCP 確立された接続をサイレントにドロップアイドルタイムアウト時に確立した接続
TCP タイムスタンプオプション	無効	タイムスタンプオプションを使用すると、正確な RTT 測定が可能になります。TCP タイムスタンプオプションを有効または無効にします。

パラメーター	デフォルト値	説明
マルチパス TCP セッションタイムアウト	0 秒	MPTCP セッションタイムアウト(秒単位)。この値が設定されていない場合は、アイドル状態になります。MPTCP セッションは、仮想サーバーのクライアントのアイドルタイムアウト後にフラッシュされます。
アイドルタイムアウト時に半閉じた接続をサイレントにドロップ	0 秒	アイドルタイムアウトで TCP ハーフクローズ接続をサイレントにドロップします。
アイドルタイムアウト時に確立された接続をサイレントにドロップする	無効	アイドルタイムアウト時に TCP 確立した接続をサイレントにドロップする
メモリ管理		
TCP バッファサイズ	131072 バイト	TCP バッファサイズは、Citrix ADC 受信バッファサイズです。このバッファサイズは、Citrix ADC からクライアントとサーバーに通知され、Citrix ADC にデータを送信する機能を制御します。デフォルトのバッファサイズは 8K で、通常、内部サーバファームと通信するときにこの値を増やしても安全です。バッファサイズは、SSL エンドポイントの場合は 40 K に設定され、圧縮の場合は 96 K に設定されるなど、Citrix ADC 実際のアプリケーション層の影響を受けます。注: 動的調整を行うには、バッファサイズ引数を設定する必要があります。
TCP 送信バッファサイズ	8190 バイト	TCP 送信バッファサイズ

パラメーター	デフォルト値	説明
TCP ダイナミック受信バッファリング	無効	ダイナミック受信バッファリングを有効または無効にします。有効にすると、メモリやネットワークの状態に基づいて受信バッファを動的に調整できます。注**:** 動的調整を行うには、buffer size 引数を設定する必要があります。
TCP 最大輻輳ウィンドウ (CWND)	524288 バイト	TCP 最大輻輳ウィンドウ
ウィンドウのスケーリングステータス	有効に	ウィンドウのスケーリングを有効または無効にします。
ウィンドウのスケーリング係数	8	新しいウィンドウサイズの計算に使用される係数。この引数は、ウィンドウのスケーリングが有効になっている場合にのみ必要です。
接続セットアップ		
キープアライブプローブ	無効	定期的な TCP キープアライブ（KA）プローブを送信して、ピアがまだアップしているかどうかを確認します。
キープアライブプローブを開始する前の接続のアイドル時間	900 秒	キープアライブ（KA）プローブを送信する前の接続がアイドル状態になるまでの時間（秒）。
キープアライブプローブ間隔	75 秒	ピアが応答しない場合、次のキープアライブ（KA）プローブの前の時間間隔（秒）。
接続をドロップする前に逃すキープアライブプローブの最大数。	3	ピアがダウンしていると仮定する前に、確認応答がないときに送信されるキープアライブ（KA）プローブの数。
RST ウィンドウ減衰 (なりすまし保護)。	無効	スプーフィングから保護するために、RST ウィンドウ減衰を有効または無効にします。有効にすると、シーケンス番号が無効の場合、応答は是正 ACK となります。

パラメーター	デフォルト値	説明
最後に確認応答されたシーケンス番号で RST を受け入れます。	有効	
データ転送		
PUSH パケットの即時 ACK	有効	PUSH フラグを使用して TCP パケットの受信時に即時肯定応答（ACK）を送信します。
MSS あたりの最大パケット数	0	TCP データセグメントで許可するオクテットの最大数
ナーグルのアルゴリズム	無効	Nagle のアルゴリズムは、TCP 伝送のパケットが小さいという問題に対処します。Telnet などのリアルタイムエンジンなどのアプリケーションは、すべてのキーストロークを反対側に渡す必要があるため、小さなパケットを作成することがよくあります。Nagle のアルゴリズムで Citrix ADC は、このような小さなパケットをバッファリングし、接続効率を高めるために一緒に送信することができます。このアルゴリズムは、Citrix ADC 他の TCP 最適化手法と一緒に機能する必要があります。
バーストで許可される TCP セグメントの最大数	10 MSS	バーストで許可される TCP セグメントの最大数
キューに入れる順序外パケットの最大数	300	順不同パケットキューの最大サイズ。値 0 は制限がないことを意味します。
輻輳制御		
TCP フレーバー	キュービック	
初期輻輳ウィンドウ (cwnd) 設定	4 MSS	サーバへの TCP リンクで未処理になることができる TCP パケット数の初期の上限

パラメーター	デフォルト値	説明
TCP 明示的輻輳通知 (ECN)	無効	明示的な輻輳通知（ECN）は、パケットをドロップすることなく、ネットワーク輻輳のエンドツーエンド通知を提供します。
TCP 最大輻輳ウィンドウ (CWND)	524288 バイト	TCP は輻輳ウィンドウ（CWND）を維持し、エンドツーエンドで送信される可能性のある未確認パケットの総数を制限します。TCP では、輻輳ウィンドウは、いつでも未処理になる可能性のあるバイト数を決定する要素の 1 つです。輻輳ウィンドウは、送信者と受信者の間のリンクが多すぎるトラフィックで過負荷にならないようにする手段です。これは、リンク上に存在する輻輳の量を推定することによって計算されます。
TCP ハイブリッドスタート (HyStart)	8 バイト	
TCP 最小再送信タイムアウト（RTO）（ミリ秒）	1000	最小再送信タイムアウト（ミリ秒単位）。10 ミリ秒単位で指定します（10 で割った場合は、値が整数になる必要があります）。
TCP デュパックのしきい値	無効	
バーストレート制御	3	TCP バーストレート制御無効/固定/ダイナミック。FIXED には TCP レートを設定する必要があります
TCP レート	無効	TCP 接続ペイロード送信レート（KB/秒）
TCP レート最大キュー	0	BurstrateControl が使用されている場合の最大接続キューサイズ（バイト単位）。
MPTCP		

パラメーター	デフォルト値	説明
マルチパス TCP	無効	Multipath TCP（MPTCP）は、マルチパス TCP サービスを提供するための通常の TCP に対する拡張のセットです。これにより、トランスポート接続を複数のパスで同時に動作させることができます。
事前に確立されたサブフロー上のマルチパス TCP ドロップデータ	無効	事前確立されたサブフローにデータをサイレントドロップするのを有効または無効にします。有効の場合、DSS データパケットは、事前に確立されたサブフローでデータが受信されたときに、接続をドロップするのではなく、サイレントにドロップされます。
マルチパス TCP ファストオープン	無効	マルチパス TCP ファストオープンを有効または無効にします。有効の場合、DSS データパケットは SYN ハンドシェイクの 3 番目の ACK を受信する前に受け入れられます。
マルチパス TCP セッションタイムアウト	0 秒	MPTCP セッションタイムアウト(秒単位)。この値が設定されていない場合、仮想サーバーのクライアントのアイドルタイムアウト後にアイドル状態の MPTCP セッションがフラッシュされます。
Security		
SYN スプーフィング保護	無効	スプーフィングから保護するために、無効な SYN パケットのドロップを有効または無効にします。無効にすると、SYN パケットが受信されたときに確立された接続がリセットされます。

パラメーター	デフォルト値	説明
TCP Syncookie	無効	これは、SYN フラッド攻撃に抵抗するために使用されます。クライアントとの TCP ハンドシェイクの SYNCOOKIE メカニズムを有効または無効にします。SYNCOOKIE を無効にすると、Citrix ADC アプライアンスでの SYN 攻撃保護が防止されます。
損失検出とリカバリ		
重複選択的謝辞 (DSACK)	有効	Citrix ADC アプライアンスは、重複選択確認（DSACK）を使用して、再送信がエラーで送信されたかどうかを判断します。
フォワード RTO リカバリ (FRTO)	有効	スプリアス TCP 再送信タイムアウトを検出します。タイムアウトによってトリガーされた最初の未確認のセグメントを再送信した後、TCP 送信者のアルゴリズムは、着信確認を監視して、タイムアウトがスプリアスであるかどうかを判断します。次に、新しいセグメントを送信するか、未確認のセグメントを再送信するかを決定します。このアルゴリズムは、別の不要な再送信を効果的に回避するのに役立ち、スプリアスタイムアウトの場合の TCP パフォーマンスが向上します。
TCP 転送確認応答 (FACK)	有効	FACK（フォワード ACK）を有効または無効にします。

パラメーター	デフォルト値	説明
選択的確認応答 (SACK) ステータス	有効	TCP SACK は、全体的なスループット容量を低下させる複数のパケット損失の問題に対処します。選択的確認応答を使用すると、受信者は正常に受信されたすべてのセグメントについて送信者に通知できるため、送信者は失われたセグメントのみを再送信できるようになります。この手法は、Citrix ADC が全体的なスループットを向上させ、接続待ち時間を短縮するのに役立ちます。
再送信あたりの最大パケット数	1	Citrix ADC が、1 回の試行で再送信するパケット数を制御できるようにします。Citrix ADC が部分的な ACK を受信し、再送信を行う必要がある場合、この設定が考慮されます。これは RTO ベースの再送信には影響しません。
TCP 遅延-ACK タイマー	100 ミリ秒	TCP 遅延 ACK のタイムアウト（ミリ秒単位）
TCO 最適化		
TCP 最適化モード	トランスペアレント	TCP 最適化モードトランスペアレント/エンドポイント
適応型 TCP 最適化の適用	無効	アダプティブ TCP 最適化の適用
TCP セグメンテーションオフロード	自動	TCP セグメンテーションを NIC にオフロードします。AUTOMATIC に設定すると、NIC がサポートしている場合、TCP セグメンテーションは NIC にオフロードされます。
ACK 集約	無効	ACK 集約を有効または無効にする
TCP タイム待機 (または time_Wait)	40 秒	閉じた TCP 接続を解放するまでの経過時間

パラメーター	デフォルト値	説明
RST 上のクライアントとサーバーのリンクを解消する	無効	相手側に送信する未処理のデータがある場合は、クライアントとサーバーの接続をリンク解除します。

グローバル **TCP** パラメータの設定

Citrix ADC アプライアンスでは、すべての Citrix ADC サービスと仮想サーバーに適用される TCP パラメータの値を指定できます。これは、以下を使用して実行できます。

- デフォルトの TCP プロファイル
- グローバル TCP コマンド
- TCP バッファリング機能

> 注意:
>
> set ns tcppParam コマンドの`recvBuffSize`パラメータは、リリース 9.2 以降から廃止されました。以降のリリースでは、set ns tcpProfile コマンドの`bufferSize`パラメータを使用してバッファサイズを設定します。`recvBuffSize`パラメータが廃止されるリリースにアップグレードすると、`bufferSize`パラメータはデフォルト値に設定されます。

デフォルトの **TCP** プロファイル

`nstcp_default_profile`という TCP プロファイルは、サービスレベルまたは仮想サーバレベルで TCP 設定が提供されない場合に使用される TCP 設定を指定するために使用されます。

> メモ:
>
> - すべての TCP パラメータをデフォルトの TCP プロファイルで設定できるわけではありません。一部の設定は、グローバル TCP コマンドを使用して実行する必要があります (以下のセクションを参照)。
> - デフォルトプロファイルは、サービスまたは仮想サーバーに明示的にバインドする必要はありません。

デフォルトの TCP プロファイルを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

```
1 set ns tcpProfile nstcp_default_profile...
2 <!--NeedCopy-->
```

- GUI で、[システム] > [プロファイル] に移動し、[**TCP** プロファイル] をクリックして、nstcp_default_profile を更新します。

グローバル **TCP** コマンド

グローバル TCP パラメータを設定するために使用できるもう 1 つの方法は、global TCP コマンドです。このコマンドでは、一意のパラメータに加えて、TCP プロファイルを使用して設定できるいくつかのパラメータが複製されます。これらの重複パラメータに対する更新は、デフォルトの TCP プロファイルの対応するパラメータに反映されます。

たとえば、この方法を使用して SACK パラメータを更新すると、デフォルトの TCP プロファイル（nstcp_default_profile）の SACK パラメータにその値が反映されます。

> 注:
>
> このアプローチは、デフォルトの TCP プロファイルで使用できない TCP パラメータにのみ使用することをお勧めします。

グローバル TCP コマンドを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

```
1 set ns tcpParam …
2 <!--NeedCopy-->
```

- GUI で、[システム] **>** [設定] に移動します。[**TCP** パラメータの変更] をクリックし、必要な TCP パラメータを更新します。

TCP バッファリング機能

Citrix ADC は、TCP バッファサイズを指定するために使用できる、TCP バッファリングと呼ばれる機能を提供します。この機能は、グローバルに有効にすることも、サービスレベルで有効にすることもできます。

> 注:
>
> バッファサイズは、デフォルトの TCP プロファイルで設定することもできます。TCP バッファリング機能とデフォルトの TCP プロファイルでバッファサイズの値が異なる場合は、大きい値が適用されます。

TCP バッファリング機能をグローバルに設定するには

- コマンドプロンプトで、次のように入力します。

  enable ns mode TCPB

  ```
  set ns tcpbufParam -size <positiveInteger> -memLimit <positiveInteger>
  ```

- GUI で、[システム] > [設定] に移動し、[モードの設定] をクリックし、[**TCP** バッファリング] を選択します。

  そして、[システム] > [設定] に移動し、[**TCP** パラメータの変更] をクリックし、[バッファサイズ] と [メモリ使用制限] の値を指定します。

サービスまたは仮想サーバ固有の **TCP** パラメータの設定

TCP プロファイルを使用して、サービスおよび仮想サーバの TCP パラメータを指定できます。TCP プロファイルを定義し（または組み込みの TCP プロファイルを使用して）、プロファイルを適切なサービスおよび仮想サーバに関連付ける必要があります。

> 注:
>
> 要件に従って、デフォルトプロファイルの TCP パラメータを変更することもできます。

TCP バッファリング機能で指定されたパラメータを使用して、サービスレベルで TCP バッファサイズを指定できます。

コマンドラインインターフェイスを使用してサービスレベルまたは仮想サーバーレベルの TCP 構成を指定するには

コマンドプロンプトで、次の操作を実行します。

1. TCP プロファイルを設定します。

```
set ns tcpProfile <profile-name>...
<!--NeedCopy-->
```

2. TCP プロファイルをサービスまたは仮想サーバーにバインドします。

```
    set service <name> ....
    <!--NeedCopy-->
```

例**:**

```
> set service service1 -tcpProfileName profile1
```

TCP プロファイルを仮想サーバにバインドするには、次の手順を実行します。

```
set lb vserver <name> ....
<!--NeedCopy-->
```

例:

```
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```

GUI を使用してサービスレベルまたは仮想サーバレベルの TCP 構成を指定するには

GUI で、次の手順を実行します。

1. TCP プロファイルを設定します。

   [システム] > [プロファイル] > [**TCP** プロファイル] に移動し、TCP プロファイルを作成します。

2. TCP プロファイルをサービスまたは仮想サーバーにバインドします。

   [トラフィック管理] > [負荷分散] > [サービス**/**仮想サーバー] に移動し、サービスまたは仮想サーバーにバインドする TCP プロファイルを作成します。

組み込み **TCP** プロファイル

構成の便宜上、Citrix ADC には TCP プロファイルがいくつか組み込まれています。次の組み込みプロファイルを確認し、プロファイルを選択してそのまま使用するか、要件に合わせて変更します。これらのプロファイルを、必要なサービスまたは仮想サーバーにバインドできます。

組み込みのプロファイル	説明
nstcp_default_profile	アプライアンスのデフォルトのグローバル TCP 設定を表します。
nstcp_default_tcp_lan	バックエンドサーバー接続で、これらのサーバーがアプライアンスと同じ LAN 上に存在する場合に役立ちます。
nstcp_default_WAN	WAN 展開に便利です。
nstcp_default_tcp_lan_thin_stream	nstcp_default_tcp_lan プロファイルに似ています。ただし、この設定は小さいサイズのパケットフローに合わせて調整されます。
nstcp_default_tcp_interactive_stream	nstcp_default_tcp_lan プロファイルに似ています。ただし、遅延 ACK タイマーと ACK ON **PUSH** パケットの設定が減少します。
nstcp_default_tcp_lfp	クライアント側の長太パイプネットワーク (WAN) に便利です。長いファットパイプネットワークには、遅延が長く、パケットドロップが最小限に抑えられ、帯域幅の回線が長くなります。
nstcp_default_tcp_lfp_thin_stream	nstcp_default_tcp_lfp プロファイルに似ています。ただし、この設定は小さいサイズのパケットフローに合わせて調整されます。
nstcp_default_tcp_lnp	クライアント側の細長いパイプネットワーク (WAN) に便利です。細長いパイプネットワークでは、かなりのパケット損失が発生することがあります。

組み込みのプロファイル	説明
nstcp_default_tcp_lnp_thin_stream	nstcp_default_tcp_lnp プロファイルに似ています。ただし、この設定は小さいサイズのパケットフローに合わせて調整されます。
nstcp_internal_apps	アプライアンス上の内部アプリケーション（GSLB サイトの同期など）に便利です。これには、目的のアプリケーションに合わせて調整されたウィンドウスケーリングと SACK オプションが含まれています。このプロファイルは、内部アプリケーション以外のアプリケーションにバインドしないでください。
nstcp_default_mobile_profile	モバイルデバイスに便利です。
nstcp_default_xd_profile	Citrix Virtual Apps and Desktops の展開に役立ちます。

TCP 設定の例

次の設定に使用するコマンドラインインターフェイスの例の例。

なりすまし攻撃に対する TCP の防御

Citrix ADC がなりすまし攻撃から TCP を防御できるようにします。デフォルトでは、「rstWindowWattenuation」パラメータは無効になっています。このパラメータは、スプーフィングからアプライアンスを保護するために有効になっています。有効にすると、無効なシーケンス番号に対する修正確認応答（ACK）で応答します。有効な値は、[有効]、[無効] です。

ここで、RST window attenuate パラメータは、スプーフィングからアプライアンスを保護します。有効にすると、シーケンス番号が無効な場合に是正 ACK で返信します。

```
1      > set ns tcpProfile profile1 -rstWindowAttenuate ENABLED -
          spoofSynDrop ENABLED
2      Done
3      > set lb vserver lbvserver1 -tcpProfileName profile1
4      Done
5  <!--NeedCopy-->
```

明示的な輻輳通知（ECN）

```
Enable ECN on the required TCP profile
```



```
    > set ns tcpProfile profile1 -ECN ENABLED
    Done
    > set lb vserver lbvserver1 -tcpProfileName profile1
    Done
<!--NeedCopy-->
```

選択的謝辞 **(SACK)**

必要な TCP プロファイルで SACK を有効にします。

```
    > set ns tcpProfile profile1 -SACK ENABLED
    Done
    > set lb vserver lbvserver1 -tcpProfileName profile1
    Done
<!--NeedCopy-->
```

転送確認応答 **(FACK)**

必要な TCP プロファイルで FACK を有効にします。

```
> set ns tcpProfile profile1 -FACK ENABLED
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```

ウィンドウスケーリング **(WS)**

ウィンドウスケーリングを有効にし、必要な TCP プロファイルでウィンドウのスケーリング係数を設定します。

```
set ns tcpProfile profile1  -WS ENABLED  -WSVal 9
Done
set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```


最大セグメントサイズ **(MSS)**

MSS 関連の設定を更新します。

```
> set ns tcpProfile profile1  - mss 1460 - maxPktPerMss 512
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```

Citrix ADC で仮想サーバーの **MSS** を学習する

Citrix ADC が VSS を学習し、その他の関連構成を更新できるようにします。

```
> set ns tcpParam -learnVsvrMSS ENABLED  - mssLearnInterval 180 -
    mssLearnDelay 3600
Done
<!--NeedCopy-->
```

TCP キープアライブ

TCP キープアライブを有効にし、その他の関連設定を更新します。

```
> set ns tcpProfile profile1 -KA ENABLED -KaprobeUpdateLastactivity ENABLED
-KAconnIdleTime 900 -KAmaxProbes 3 -KaprobeInterval 75
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

バッファサイズ**-TCP** プロファイルの使用

バッファサイズを指定します。

```
> set ns tcpProfile profile1 -bufferSize 8190
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```


バッファサイズ-**TCP** バッファリング機能を使用

TCP バッファリング機能（グローバルまたはサービス）を有効にし、バッファサイズとメモリ制限を指定します。

```
> enable ns feature TCPB
Done
> set ns tcpbufParam -size 64 -memLimit 64
Done
```

MPTCP

MPTCP を有効にし、オプションの MPTCP 設定を設定します。

```
> set ns tcpProfile profile1 -mptcp ENABLED
Done
> set ns tcpProfile profile1 -mptcpDropDataOnPreEstSF ENABLED -mptcpFastOpen
 ENABLED -mptcpSessionTimeout 7200
Done
> set ns tcpparam -mptcpConCloseOnPassiveSF ENABLED -mptcpChecksum ENABLED
-mptcpSFtimeout 0 -mptcpSFReplaceTimeout 10
-mptcpMaxSF 4 -mptcpMaxPendingSF 4 -mptcpPendingJoinThreshold 0 -mptcpRTOsToSwitchSF
 2 -mptcpUseBackupOnDSS ENABLED
Done
```

輻輳制御

必要な TCP 輻輳制御アルゴリズムを設定します。

```
set ns tcpProfile profile1 -flavor Westwood
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

動的受信バッファリング

必要な TCP プロファイルでダイナミック受信バッファリングを有効にします。

```
> set ns tcpProfile profile1 -dynamicReceiveBuffering ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```


マルチパス **TCP (MPTCP)** での **TCP** ファストオープン **(TFO)** のサポート

Citrix ADC アプライアンスは、マルチパス TCP（MPTCP）接続を確立し、データ転送を高速化するための TCP 高速オープン（TFO）メカニズムをサポートするようになりました。このメカニズムにより、SYN および SYN-ACK パケットでの初期 MPTCP 接続ハンドシェイク中にサブフローデータを伝送できます。また、MPTCP 接続の確立中に受信ノードによってデータが消費されることも可能になります。

詳細については、「TCP 高速オープン」のトピックを参照してください。

MPTCP の可変 **TFO** クッキーサイズのサポート

Citrix ADC アプライアンスでは、TCP プロファイルで最小サイズが 4 バイト、最大サイズが 16 バイトの可変長 TCP ファストオープン（TFO）Cookie を構成できるようになりました。これにより、アプライアンスは SYN-ACK パケットで設定された TFO cookie サイズでクライアントに応答できます。

コマンドラインインターフェイスを使用して TCP プロファイルで TCP Fast Open（TFO）Cookie を設定するには

コマンドプロンプトで入力します。

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize <positive_integer
>
```

例

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize 8
```

GUI を使用して TCP プロファイルで TCP ファストオープン（TFO）クッキーを設定するには

1. [設定] > [システム] > [プロファイル] に移動します。
2. 詳細ペインで、[**TCP** プロファイル] タブに移動し、TCP プロファイルを選択します。
3. [**TCP** プロファイルの設定] ページで、[**TCP** 高速オープン Cookie サイズ] を設定します。
4. [**OK**] をクリックし、[完了] をクリックします。

SYN-Cookie timeout interval

TCPSyncookieパラメータは、SYN 攻撃に対する堅牢な（RFC 4987）ベースの保護を提供するために、TCP プロファイルでデフォルトで有効になっています。この保護と互換性のないカスタム TCP クライアントに対応する必要があるが、攻撃が発生した場合にフォールバックを確実に実行する場合、synAttackDetectionは、autosyncookietimeout パラメータで指定された時間内に、SYNCookie 動作を内部で自動的にアクティブ化することで、この処理を行います。

コマンドラインインターフェイスを使用して SYN ACK 再送信の最大しきい値を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。


```
    set ns tcpparam [-maxSynAckRetx <positive_integer>]

    Set ns tcpparam [-maxSynAckRetx 150]
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して自動 SYN Cookie タイムアウト間隔を設定するには

コマンドプロンプトで入力します。

```
set ns tcpparam [-autosyncookietimeout <positive_integer>]
```

```
Set ns tcpparam [-autosyncookietimeout 90]
```

クライアントとサーバーの接続をリンク解除する

有効にすると、相手側に送信する未処理のデータがある場合、このパラメータはクライアントとサーバーの接続を切断します。デフォルトでは、パラメータは無効になっています。

```
set ns tcpparam -delinkClientServerOnRST ENABLED
Done

<!--NeedCopy-->
```

HTTP 構成

January 31, 2022

> 重要:
>
> Citrix ADC リリース 13.0 ビルド 71.x 以降、Citrix ADC アプライアンスは L7 アプリケーション要求に対応するために、大きなヘッダーサイズの HTTP 要求を処理できます。ヘッダーサイズは最大 128 KB まで設定できます。

Citrix ADC アプライアンスの HTTP 構成は、HTTP 設定の集合である HTTP プロファイルと呼ばれるエンティティで指定できます。HTTP プロファイルは、これらの HTTP 設定を使用するサービスまたは仮想サーバに関連付けることができます。

デフォルトの HTTP プロファイルは、すべてのサービスおよび仮想サーバにグローバルに適用される HTTP 設定を設定するように設定できます。

> 注:
>
> HTTP パラメータにサービス、仮想サーバー、およびグローバルに異なる値がある場合、最も固有のエンティティ（サービス）の値が最も高い優先順位が与えられます。

Citrix ADC アプライアンスは、HTTP を構成するための他のアプローチも提供します。詳細については読んでください。

Citrix ADC は WebSocket プロトコルをサポートしています。これにより、ブラウザや他のクライアントはサーバーへの双方向全二重 TCP 接続を作成できます。WebSocket の Citrix ADC の実装は、RFC 6455 に準拠しています 。

> 注:
>
> Citrix ADC アプライアンスは、HTTP/1.1 プロトコルと HTTP/2 プロトコルのユーザーソース IP（USIP）アドレス構成をサポートするようになりました。

グローバル **HTTP** パラメータの設定

Citrix ADC アプライアンスでは、すべての Citrix ADC サービスと仮想サーバーに適用される HTTP パラメータの値を指定できます。これは、以下を使用して実行できます。

- デフォルトの HTTP プロファイル
- グローバル HTTP コマンド

デフォルトの **HTTP** プロファイル

nshttp_default_profile という名前の HTTP プロファイルは、サービスレベルまたは仮想サーバレベルで HTTP 構成が提供されない場合に使用される HTTP 構成を指定するために使用されます。

> メモ:
>
> - すべての HTTP パラメータをデフォルトの HTTP プロファイルで設定できるわけではありません。一部の設定は、グローバル HTTP コマンドを使用して実行します (次のセクションを参照)。
> - デフォルトプロファイルは、サービスまたは仮想サーバーに明示的にバインドする必要はありません。

デフォルトの HTTP プロファイルを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

  ```
  set ns httpProfile nshttp_default_profile …
  ```

- GUI で、[システム] > [プロファイル] に移動し、[**HTTP** プロファイル] をクリックし、nshttp_default_profile を更新します。

グローバル **HTTP** コマンド

グローバル HTTP パラメータを設定するために使用できるもう 1 つの方法は、global HTTP コマンドです。このコマンドは一意のパラメータに加えて、HTTP プロファイルを使用して設定できるいくつかのパラメータを複製します。

これらの重複パラメータに対して行われた更新は、デフォルトの HTTP プロファイルの対応するパラメータに反映されます。

たとえば、このアプローチを使用して maxReusePool パラメータを更新すると、値はデフォルトの HTTP プロファイル (nshttp_default_profile) の maxReusePool パラメータに反映されます。

> 注:
>
> このアプローチは、デフォルトの HTTP プロファイルで使用できない HTTP パラメータにのみ使用することをお勧めします。

グローバル HTTP コマンドを設定するには

- コマンド・ライン・インタフェースを使用して、コマンド・プロンプトに次のように入力します。

  ```
  set ns httpParam …
  ```

- GUI で、[システム] **>** [設定] に移動し、[**HTTP** パラメータの変更] をクリックし、必要な HTTP パラメータを更新します。

接続要求に対して無視コーディングスキームを設定するには

HTTP/2 を有効にし、接続リクエストのコーディングスキームを無視するように HTTP/2 パラメータを設定するには、コマンドプロンプトで次のように入力します。

```
set ns httpParam [-ignoreConnectCodingScheme ( ENABLED | DISABLED )]
```

例**:**

```
set ns httpParam -ignoreConnectCodingScheme ENABLED
```

Citrix ADC コマンドラインを使用して HTTP プロファイルを仮想サーバーにバインドするには

HTTP プロファイルを設定して **TRACE** または **TRACK** 無効な要求をドロップする

marktraceReqInval パラメーターを有効にして、TRACK リクエストと TRACK リクエストを無効としてマークできます。仮想 IP アドレスで dropInvalidReqs オプションとともにこのオプションを有効にすると、Citrix ADC アプライアンスに TRACE または TRACK 要求を送信するクライアントをリセットできます。

CLI を使用して HTTP プロファイルを設定するには

コマンドプロンプトで入力します。

```
set ns httpProfile <profile name> [-markTraceReqInval ENABLED | DISABLED ]
```

例**:**

```
set ns httpProfile profile1 -markTraceReqInval ENABLED
```


サービスグループの **HTTP** プロファイルの設定

コマンドプロンプトで入力します。

```
add serviceGroup <serviceGroupName>@ <serviceType> [-cacheType <
    cacheType>] [-td <positive_integer>] [-maxClient <positive_integer>]
     [-maxReq <positive_integer>] [-cacheable ( YES | NO )] [-cip (
    ENABLED | DISABLED ) [<cipHeader>]] [-usip ( YES | NO )] [-
    pathMonitor ( YES | NO )] [-pathMonitorIndv ( YES | NO )] [-
    useproxyport ( YES | NO )] [-healthMonitor ( YES | NO )] [-sp ( ON |
     OFF )] [-rtspSessionidRemap ( ON | OFF )] [-cltTimeout <secs>] [-
    svrTimeout <secs>] [-CKA ( YES | NO )] [-TCPB ( YES | NO )] [-CMP (
    YES | NO )] [-maxBandwidth
<positive_integer>] [-monThreshold <positive_integer>] [-state ENABLED
    DISABLED )][-downStateFlush ( ENABLED | DISABLED )] [-tcpProfileName
     <string>] [-httpProfileName <string>] [-comment <string>] [-
    appflowLog ( ENABLED | DISABLED )] [-netProfile <string>] [-
    autoScale <autoScale> -memberPort <port> [-autoDisablegraceful ( YES
     | NO )] [-autoDisabledelay <secs>] ] [-monConnectionClose ( RESET |
     FIN )]

<!--NeedCopy-->
```

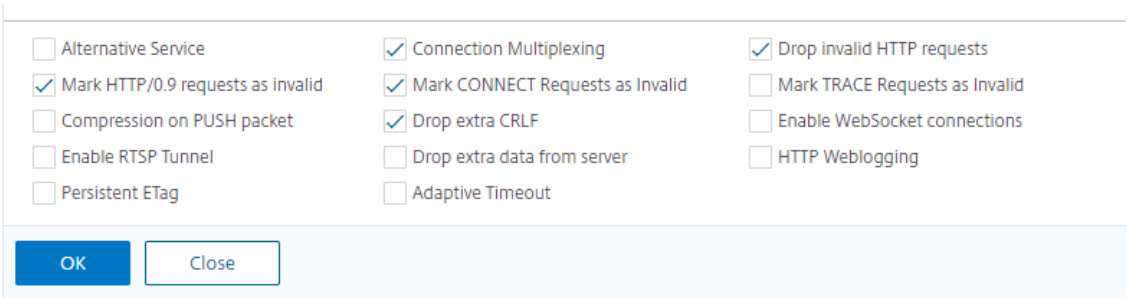
例**:**

```
add serviceGroup Service-Group-1 HTTP -maxClient 0 -maxReq 0 -cip ENABLED -
usip NO -useproxyport YES -cltTimeout 200 -svrTimeout 300 -CKA NO -TCPB NO
-CMP NO -httpProfileName profile1
```

Citrix ADC GUI を使用して **HTTP** プロファイルを構成する

TRACE または TRACK の無効なリクエストをマークするには、次の手順を実行します。

1. Citrix ADC アプライアンスにサインインし、［構成］>［システム］>［プロファイル］に移動します。
2. [**HTTP** プロファイル] タブページで、[追加] をクリックします。
3. [**HTTP** プロファイルの作成] ページで、[トレース要求を無効としてマーク] オプションを選択します。
4. ［作成］をクリックします。



サービスまたは仮想サーバー固有の **HTTP** パラメータの設定

HTTP プロファイルを使用して、サービスおよび仮想サーバの HTTP パラメータを指定できます。HTTP プロファイルを定義し（または組み込みの HTTP プロファイルを使用して）、プロファイルを適切なサービスおよび仮想サーバに関連付ける必要があります。

> 注:
>
> 要件に従って、デフォルトプロファイルの HTTP パラメータを変更することもできます。

コマンドラインインターフェイスを使用してサービスレベルまたは仮想サーバーレベルの **HTTP** 構成を指定するには

コマンドプロンプトで、次の操作を実行します。

1. HTTP プロファイルを設定します。

   ```
   set ns httpProfile <profile-name>...
   ```

2. HTTP プロファイルをサービスまたは仮想サーバーにバインドします。

   HTTP プロファイルをサービスにバインドするには、次の手順を実行します。

```
set service <name> .....
```

例:

```
1 > set service service1 -httpProfileName profile1
2 <!--NeedCopy-->
```

HTTP プロファイルを仮想サーバーにバインドするには、次の手順を実行します。

```
set lb vserver <name> .....
```

例:


```
> set lb vserver lbvserver1 -httpProfileName profile1
<!--NeedCopy-->
```

GUI を使用してサービスレベルまたは仮想サーバレベルの **HTTP** 構成を指定するには

GUI で、次の手順を実行します。

1. HTTP プロファイルを設定します。

   [システム] > [プロファイル] > [**HTTP** プロファイル] に移動し、HTTP プロファイルを作成します。

2. HTTP プロファイルをサービスまたは仮想サーバーにバインドします。

   [トラフィック管理] > [負荷分散] > [サービス/仮想サーバー] に移動し、サービス/仮想サーバーにバインドする必要がある HTTP プロファイルを作成します。

組み込みの **HTTP** プロファイル

構成の便宜のために、Citrix ADC は組み込みの HTTP プロファイルをいくつか提供しています。リストされているプロファイルを確認し、そのまま使用するか、要件に合わせて変更します。これらのプロファイルは、必要なサービスまたは仮想サーバーにバインドできます。

組み込みのプロファイル	説明
nshttp_default_profile	アプライアンスのデフォルトのグローバル HTTP 設定を表します。
nshttp_default_strict_validation	HTTP リクエストとレスポンスの厳密な検証が必要なデプロイ用の設定。

HTTP 設定の例

次の設定に使用するコマンドラインインターフェイスのサンプル例。

- HTTP バンド統計情報
- WebSocket 接続

HTTP バンド統計情報

HTTP リクエストとレスポンスのバンドサイズを指定します。

```
> set protocol httpBand reqBandSize 300 respBandSize 2048
```



```
Done
> show protocol httpband -type REQUEST
<!--NeedCopy-->
```

WebSocket 接続

必要な HTTP プロファイルで WebSocket を有効にします。

```
> set ns httpProfile http_profile1 -webSocket ENABLED
Done
> set lb vserver lbvserver1 -httpProfileName profile1
Done
<!--NeedCopy-->
```

HTTP/2 構成

January 25, 2022

注: HTTP/2 機能は、Citrix ADC MPX、VPX、および SDX モデルでサポートされています。Citrix ADC VPX アプライアンスでは、Citrix ADC バージョン 11.0 以降から HTTP/2 機能がサポートされています。

Web アプリケーションのパフォーマンスに関する問題は、ページサイズと Web ページ上のオブジェクト数の増加傾向に直接関係しています。HTTP/1.1 は、現在一般的なものよりも小さな Web ページ、低速のインターネット接続、および制限されたサーバーハードウェアをサポートするために開発されました。JavaScript やカスケードスタイルシート (CSS) などの新しいテクノロジーや、Flash ビデオやグラフィックが豊富な画像などの新しいメディアタイプには適していません。これは、サーバーへの接続ごとに 1 つのリソースしか要求できないためです。この制限により、ラウンドトリップの数が大幅に増加し、ページレンダリングが長くなり、ネットワークパフォーマンスが低下します。

HTTP/2 プロトコルは、ネットワーク上で送信されるデータを減らして通信を可能にし、1 つの接続で複数の要求と応答を送信する機能を提供することで、これらの制限に対処します。HTTP/2 では、基盤となるネットワーク接続をより効率的に使用することで、HTTP/1.1 の主な制限に対処しています。これは、要求と応答がネットワーク上で移動する方法を変更します。

HTTP/2 はバイナリプロトコルです。HTTP/1.1 のようなテキストプロトコルと比較して、解析がより効率的で、ネットワーク上でよりコンパクトになり、最も重要なこととして、エラーが発生しにくくなります。HTTP/2 プロトコルは、フレームタイプと、クライアントとサーバ間で HTTP メッセージのカプセル化および転送方法を定義するバイナリフレーミング層を使用します。HTTP/2 機能は、CONNECT メソッドを使用して、単一の HTTP/2 ストリームを介してリモートホストへのトンネル接続を確立します。

HTTP/2 プロトコルには、特にモバイルネットワーク経由で接続するクライアントに対して、パフォーマンスを大幅に向上させるパフォーマンス向上の変更が多数含まれています。

次の表に、HTTP/1.1 から HTTP/2 の主な改善点を示します。

HTTP/2 の機能	説明
ヘッダー圧縮	HTTP ヘッダーには多くの繰り返し情報があるため、データ転送中に不要な帯域幅を消費します。HTTP/2 ヘッダーを圧縮し、要求と応答ごとに HTTP ヘッダーを転送する要件を最小限に抑えることで、帯域幅の要件を削減します。
接続多重化	レイテンシーは、ページの読み込み時間とエンドユーザーエクスペリエンスに大きな影響を与える可能性があります。接続の多重化は、単一の接続で複数の要求と応答を送信することにより、この問題を克服します。
サーバープッシュ	サーバープッシュにより、サーバーはクライアントブラウザーにコンテンツをプロアクティブにプッシュできるため、ラウンドトリップの遅延を回避できます。この機能は、クライアントが必要と考える応答をキャッシュし、ラウンドトリップ回数を減らし、ページのレンダリング時間を改善します。重要：Citrix ADC アプライアンスは、サーバープッシュ機能をサポートしていません。
ヘッドオブラインブロッキングなし	HTTP 1.1 では、ブラウザは接続ごとに一度に 1 つのリソースをダウンロードできます。ブラウザが大きなリソースをダウンロードする必要がある場合、最初のダウンロードが完了するまで、他のすべてのリソースのダウンロードをブロックします。HTTP/2 は、多重化アプローチでこの問題を克服します。これにより、クライアントブラウザーは、同じ接続を介して他の Web コンポーネントを並行してダウンロードし、利用可能になったときにそれらを表示できます。

HTTP/2 の機能	説明
リクエストの優先順位付け	ブラウザーが Web ページをレンダリングするとき、すべてのリソースが同等の優先度を持つわけではありません。ロード時間を短縮するために、最新のブラウザーはすべて、アセットのタイプ、ページ上の位置、および以前の訪問から学習した優先度によってリクエストを優先します。HTTP/1.1 では、このプロトコルは多重化をサポートしておらず、サーバーによる要求の優先順位付けを通信する方法がないため、ブラウザは優先度データを使用する能力が制限される。その結果、不要なネットワーク遅延が生じます。HTTP/2 は、ブラウザがすべてのリクエストをディスパッチできるようにすることで、この問題を克服します。ブラウザーは、ストリームの依存関係と重みによってストリームの優先順位付けの優先度を伝えることができ、サーバーが応答配信を最適化できるようにします。重要: Citrix ADC アプライアンスは、要求の優先順位付け機能をサポートしていません。

HTTP/2 のしくみ

Citrix ADC アプライアンスはクライアント側でもサーバー側でも HTTP/2 をサポートします。クライアント側では、Citrix ADC アプライアンスは HTTP/2 の HTTP/HTTPS 仮想サーバーをホストするサーバーとして動作します。バックエンド側では、Citrix ADC は仮想サーバーにバインドされているサーバーのクライアントとして機能します。

したがって、Citrix ADC アプライアンスは、クライアント側とサーバー側で別々の接続を維持します。Citrix ADC アプライアンスには、クライアント側とサーバー側に別々の HTTP/2 構成があります。

HTTPS (SSL) 負荷分散構成の場合は **HTTP/2**

HTTPS 負荷分散構成の場合、Citrix ADC アプライアンスは TLS ALPN 拡張（RFC 7301）を使用して、クライアント/サーバーが HTTP/2 をサポートするかどうかを判断します。その場合、アプライアンスは、クライアント/サーバー側でデータを送信するためのアプリケーション層プロトコルとして HTTP/2 を選択します（RFC 7540-セクション 3.3 を参照）。
アプライアンスは、TLS ALPN 拡張を使用してアプリケーション層プロトコルを選択するときに、次の優先順位を使用します。

- HTTP/2（HTTP プロファイルで有効になっている場合）
- SPDY (HTTP プロファイルで有効になっている場合)

- HTTP/1.1

HTTP 負荷分散設定用の HTTP/2

HTTP 負荷分散構成の場合、Citrix ADC アプライアンスは次のいずれかの方法を使用して、HTTP/2 を使用してクライアント/サーバーとの通信を開始します。

> 注:
>
> 次のメソッドの説明では、クライアントとサーバーは HTTP/2 接続の一般的な用語です。たとえば、を使用した Citrix ADC アプライアンスの負荷分散セットアップの場合 HTTP/2, Citrix ADC アプライアンスは、クライアント側でサーバーとして機能し、サーバー側でクライアントとして機能します。

- **HTTP/2** アップグレード。クライアントは HTTP/1.1 リクエストをサーバーに送信します。リクエストにはアップグレードヘッダーが含まれており、HTTP/2 への接続をサーバーにアップグレードするように要求します。サーバーが HTTP/2 をサポートしている場合、サーバーはアップグレード要求を受け入れ、応答で通知します。クライアントとサーバは、クライアントがアップグレード確認応答を受信した後、HTTP/2 を使用して通信を開始します。
- 直接 **HTTP/2**。クライアントは、HTTP/2 アップグレード方式を使用する代わりに、HTTP/2 でサーバーとの通信を直接開始します。サーバーが HTTP/2 をサポートしていない場合、または HTTP/2 要求を直接受け付けるように構成されていない場合、クライアントからの HTTP/2 パケットはドロップされます。この方法は、クライアントデバイスの管理者が、サーバーが HTTP/2 をサポートしていることをすでに知っている場合に役立ちます。
- 代替サービス **(ALT-SVC)** を使用して直接 **HTTP/2**。サーバーは、HTTP/1.1 応答に代替サービス (ALT-SVC) フィールドを含めることで、HTTP/2 をサポートしていることをクライアントにアドバタイズします。クライアントが ALT-SVC フィールドを認識するように設定されている場合、クライアントとサーバは、クライアントが応答を受信した後、HTTP/2 を使用して直接通信を開始します。

Citrix ADC アプライアンスは、HTTP/2 メソッドの HTTP プロファイルで構成可能なオプションを提供します。これら HTTP/2 オプションは、HTTPS または HTTP 負荷分散セットアップのサーバー側だけでなくクライアント側にも適用できます。HTTP/2 メソッドとオプションの詳細については、HTTP/2 オプション PDF を参照してください。

はじめに

Citrix ADC アプライアンスで HTTP/2 の構成を開始する前に、次の点に注意してください。

- Citrix ADC アプライアンスは、クライアント側とサーバー側で HTTP/2 をサポートします。
- Citrix ADC アプライアンスは、HTTP/2 サーバープッシュ機能をサポートしていません。
- Citrix ADC アプライアンスは、HTTP/2 リクエストの優先順位付け機能をサポートしていません。
- Citrix ADC アプライアンスは、HTTPS 負荷分散セットアップの HTTP/2 SSL 再ネゴシエーションをサポートしていません。
- Citrix ADC アプライアンスは HTTP/2 NTLM 認証をサポートしていません。

- HTTP/2 が有効で、接続多重化が無効で（USIP が有効の場合など）、クライアントとサーバの TCP 接続を 1 対 1 でマッピングすると、FIN、リセット（RST）などのクローズイベントがクライアントまたはサーバ接続からリンクされたピア接続に転送されます。

HTTP/2 を構成する

負荷分散設定（HTTPS または HTTP）の HTTP/2 の設定は、次のタスクで構成されます。

- **HTTP/2** を有効にし、**HTTP** プロファイルでオプションの **HTTP/2** パラメータを設定します。HTTP プロファイルで HTTP/2 を有効にします。HTTP プロファイルで HTTP/2 のみを有効にすると、Citrix ADC アプライアンスは HTTP/2 での通信にアップグレード方法（HTTP の場合）または TLS ALPN 方式（HTTPS）のみを使用します。

  Citrix ADC アプライアンスが直接使用する場合 HTTP/2 メソッド、直接 HTTP/2 オプションは HTTP プロファイルで有効にする必要があります。Citrix ADC アプライアンスが代替サービス方法を使用して直接 HTTP/2 を使用するには、**HTTP** プロファイルで代替サービス（**altsvc**）オプションを有効にする必要があります。

- **HTTP** プロファイルを仮想サーバーまたはサービスにバインドします。HTTP プロファイルを仮想サーバーにバインドして、負荷分散セットアップのクライアント側の HTTP/2 を構成します。HTTP プロファイルをサービスにバインドして、負荷分散設定のサーバー側の HTTP2 を設定します。

> 注:
>
> クライアント側とサーバー側で別々の HTTP プロファイルをバインド Citrix。

- **HTTP/2** サーバー側のサポートのグローバルパラメータを有効にする。**HTTP/2** サービス側（**http2ServerSide**）グローバル HTTP パラメータを有効にして、HTTP/2 が構成されているすべての負荷分散セットアップのサーバー側で HTTP/2 サポートを有効にします。

  HTTP/2 サービス側が無効な場合、関連する負荷分散サービスにバインドされた **HTTP** プロファイルで **HTTP/2** が有効になっている場合でも、HTTP/2 は負荷分散設定のサーバー側では機能しません。

Citrix ADC コマンドライン手順:

Citrix ADC コマンドラインを使用して HTTP/2 を有効にして、HTTP/2 パラメーターを設定するには

- HTTP プロファイルの追加中に HTTP/2 を有効にして HTTP/2 パラメータを設定するには、コマンドプロンプトで次のように入力します。

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )] [-altsvc ( ENABLED | DISABLED )]
show ns httpProfile <name>
```

- HTTP プロファイルの変更中に HTTP/2 を有効にして HTTP/2 パラメータを設定するには、コマンドプロンプトで次のように入力します。

```
set ns httpProfile <name> -http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED)] [-altsvc (ENABLED | DISABLED )]
show ns httpProfile <name>
```


Citrix ADC コマンドラインを使用して HTTP プロファイルを仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
set lb vserver <name> - httpProfileName <string>
show lb vserver <name>
```

Citrix ADC コマンドラインを使用して HTTP プロファイルを負荷分散サービスにバインドするには

コマンドプロンプトで入力します。

```
set service <name> -httpProfileName <string>
show service <name>
```

Citrix ADC コマンドラインを使用してサーバー側で HTTP/2 サポートをグローバルに有効にするには

コマンドプロンプトで入力します。

```
set ns httpParam -HTTP2Serverside( ENABLED | DISABLED )
show ns httpParam
```

Citrix ADC GUI を使用して HTTP/2 を有効にし、HTTP/2 パラメーターを設定するには

1. [システム] > [プロファイル] に移動し、[**HTTP** プロファイル] タブをクリックします。
2. HTTP プロファイルを追加するとき、または既存の HTTP プロファイルを変更するときに、**HTTP/2** を有効にします。

Citrix ADC GUI を使用して HTTP プロファイルを仮想サーバーにバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、仮想サーバーを開きます。
2. [詳細設定] で、[**+ HTTP** プロファイル] をクリックして、作成した HTTP プロファイルを仮想サーバーにバインドします。

Citrix ADC GUI を使用して HTTP プロファイルを負荷分散サービスにバインドするには

1. [トラフィック管理] > [負荷分散] > [サービス] に移動し、サービスを開きます。
2. [詳細設定] で、[**+ HTTP** プロファイル] をクリックして、作成した HTTP プロファイルをサービスにバインドします。

GUI を使用してサーバー側で HTTP/2 サポートをグローバルに有効にするには

[システム] > [設定] に移動し、[**HTTP** パラメーターの変更] をクリックし、[**HTTP/2** サーバー側] を有効にします。

設定例

次の設定例では、HTTP プロファイル HTTP-PROFILE-HTTP2-クライアント側で HTTP/2 およびダイレクト HTTP/2 が有効になっています。プロファイルは仮想サーバー LB-VS-1 にバインドされています。


```
set ns httpProfile HTTP-PROFILE-HTTP2-CLIENT-SIDE -http2 enabled -
    http2Direct enabled
Done

set lb vserver LB-VS-1 -httpProfileName HTTP-PROFILE-HTTP2-CLIENT-SIDE

Done
<!--NeedCopy-->
```

次の設定例では、HTTP プロファイル HTTP-PROFILE-HTTP2-サーバー側で HTTP/2 および代替サービス（ALT-SVC）が有効になっています。プロファイルはサービス LB-SERVICE-1 にバインドされます。

```
set ns httpparam -HTTP2Serverside ENABLED
Done

set ns httpProfile HTTP-PROFILE-HTTP2-SERVER-SIDE -http2 ENABLED -
    altsvc ENABLED
Done

set service LB-SERVICE-1 -httpProfileName HTTP-PROFILE-HTTP2-SERVER-
    SIDE
Done
<!--NeedCopy-->
```

HTTP/2 の初期接続ウィンドウサイズを構成する

RFC 7540 に従って、HTTP2 ストリームおよび接続のフロー制御ウィンドウを 64 K（65535）オクテットに設定する必要があり、この値に加えられた変更はピアに伝達する必要があります。ADC アプライアンスは、フロー制御ウィンドウサイズの変化を次のように伝えます。

- ストリームにSETTINGSフレームを使用する。
- 接続にWINDOW_UPDATEフレームを使用する。

HTTP プロファイルでは､ストリームレベルで初期ウィンドウサイズを設定するようにhttp2InitialWindowSizeパラメータを設定する必要があります。内部システムエラーのため、ADC アプライアンスは接続のフロー制御ウィンドウも初期化します。ストリームに設定されたフロー制御ウィンドウに変更がある場合、ADC アプライアンスは設定フレームを使用してピアと通信します。しかし、ADC アプライアンスは、WINDOW_UPDATEフレームを使用して接続のフロー制御ウィンドウの変更を伝達しません。これにより、接続がフリーズします。

この問題を克服するために、接続のフロー制御ウィンドウを制御するhttp2InitialConnWindowSizeパラメータ（バイト単位）が追加されました。個別の設定可能なパラメータを使用することで、ストリームレベルと接続レベルの両方で、変更されたウィンドウサイズの更新をアプライアンスが送信できるようにできるようになりました。

CLI を使用して **HTTP/2** 初期接続ウィンドウサイズパラメータを設定します

コマンドプロンプトで入力します。

```
set http profile p1 -http2InitialConnWindowSize 8290
Initial window size for stream level flow control, in bytes.
Default value: 65535
Minimum value: 8192
Maximum value: 20971520
<!--NeedCopy-->
```

HTTP/2 DoS の軽減

October 7, 2021

Http/2 サービス拒否（DoS）攻撃は、Citrix ADC アプライアンスに影響を与えなくなりました。アプライアンスが最大制限を超えるフレームを受信した場合、アプライアンスはサイレントに接続を閉じます。

攻撃を軽減するために、HTTP プロファイルを使用すると、HTTP/2 接続で受信したフレームのデフォルト構成を変更できます。

HTTP/2 DoS 緩和の表には 、HTTP/2 DoS 攻撃とその緩和策のリストが示されています。

コマンドラインインターフェイスを使用して **DoS** 攻撃を軽減する **HTTP/2** フレームの上限を設定します

コマンドプロンプトで、次のように入力します。

```
set ns httpprofile <profile_name> - http2MaxEmptyFramesPerMin <positive_integer
> -http2MaxPingFramesPerMin <positive_integer> -http2MaxSettingsFramesPerMin
 <positive_integer> -http2MaxResetFramesPerMin <positive_integer>
```

例**:**

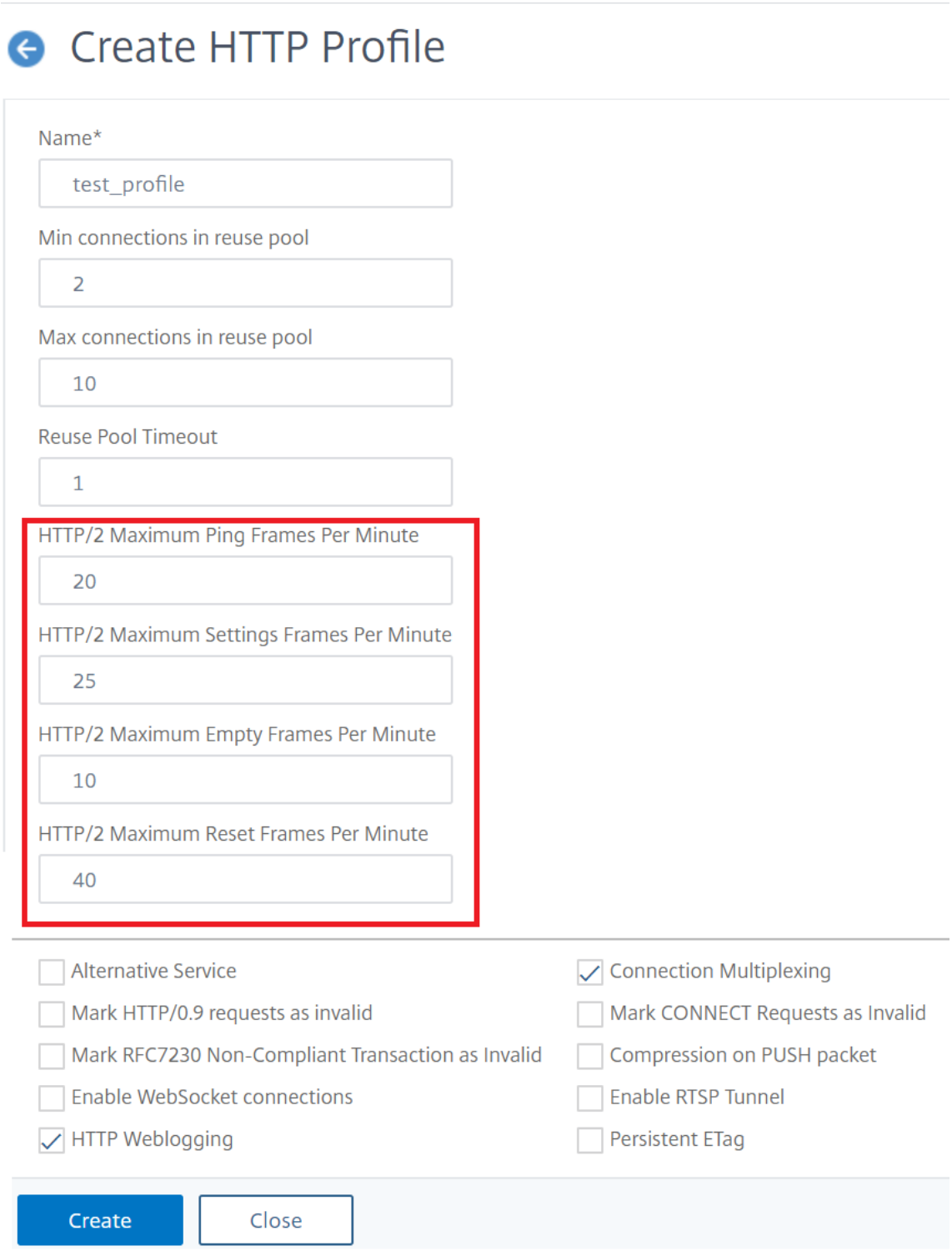
```
set ns httpprofile profile1 -http2MaxEmptyFramesPerMin 20 -http2MaxPingFramesPerMin
 20 -http2MaxSettingsFramesPerMin 20 -http2MaxResetFramesPerMin 20
```

Citrix ADC GUI を使用して、**HTTP/2** 接続で受信するフレームの最大制限を構成する

HTTP/2 接続で受信するフレームの最大制限を設定するには、次の手順に従います。

1. ナビゲーションウィンドウで、[システム] を展開し、[プロファイル] をクリックします。
2. [プロファイル] ページで、[**HTTP** プロファイル] タブを選択します。

3. [**HTTP** プロファイル] タブページで、[追加] をクリックします。
4. [**HTTP** プロファイルの設定**]** ページで、次のパラメータを設定します。
   a) http2MaxPingFramesPerMin. 接続ごとに受信する最大 PING フレームを 1 分単位で設定します。PING フレーム数が構成制限を超えた場合、アプライアンスは接続上のパケットをサイレントにドロップします。
   b) http2MaxSettingsFramesPerMin. 接続ごとに受信する最大 SETTINGS フレームを 1 分単位で設定します。SETTINGS フレーム数が設定制限を超えた場合、ADC は接続上のパケットをサイレントにドロップします。
   c) http2MaxResetFramesPerMin. 接続ごとに送信される最大 RESET フレームを 1 分単位で設定します。RESET フレーム数が設定制限を超えた場合、ADC は接続上のパケットをサイレントにドロップします。
   d) http2MaxEmptyFramesPerMin. 接続ごとに送信される最大空フレームを 1 分単位で設定します。空のフレーム数が設定制限を超えた場合、ADC は接続上のパケットをサイレントにドロップします。
5. [**OK]** をクリックして [閉じる] をクリックします。



HTTP3 over QUIC プロトコル

October 7, 2021

HTTP/2 over TCP は、単一の接続で複数の HTTP リクエストストリームを送信する場合に推奨される標準です。ただし、TCP トランスポートメカニズムでは、Web サイトや Web アプリケーションへのアクセスに一定の制限とレイテンシーの問題があります。同じ接続で複数のリクエストを多重化すると、同じ接続の信頼性の影響を受けます。1 つの要求のパケットが失われた場合、他のすべての多重化要求は、失われたパケットが検出されて再送信されるまで遅延します。これにより、ヘッドオブラインブロッキングの遅延とレイテンシーの問題が発生します。

接続および転送の遅延については、HTTP/3 は TCP プロトコルの代わりに QUIC を使用します。QUIC は、TCP の代わりに UDP を基本トランスポートとして使用する新しいプロトコルです。HTTP-over-quic では、単一の TCP 接続に依存することなく、複数の独立した要求を多重化することができます。QUIC は、複数の HTTP リクエストをストリーミングできる信頼性の高い接続を実装しています。QUIC は、HTTP/1.1 や HTTP/2 のように追加のレイヤーとしてではなく、TLS を統合コンポーネントとして組み込んでいます。

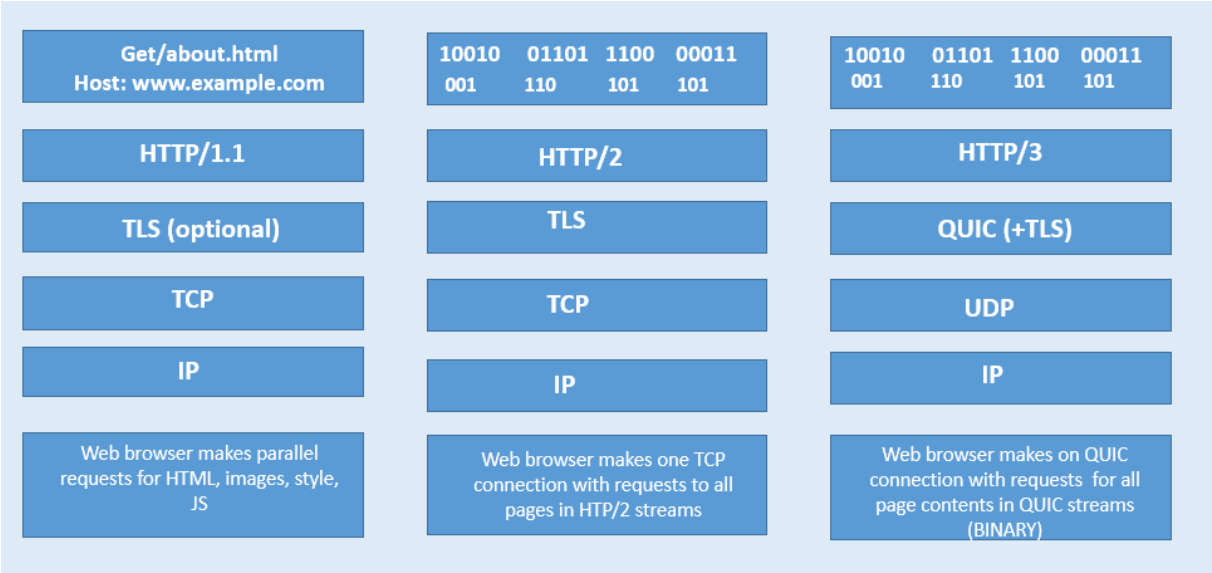
HTTP/3 プロトコルを使用する利点

HTTP/3 データ転送に QUIC プロトコルを使用する重要な利点のいくつかを以下に示します。

- ストリーム多重化
- ストリームおよび接続レベルのフロー制御
- 低レイテンシーの接続確立
- NAT 再バインドへの接続の移行と復元力
- 認証され、暗号化されたヘッダーとペイロード

HTTP プロトコルのトランスポートスタック

以下の図は、HTTP/1.1、HTTP/2、および HTTP/3 プロトコルのトランスポートスタックを示しています。



Citrix ADC での **QUIC** および **HTTP/3** 接続管理の仕組み

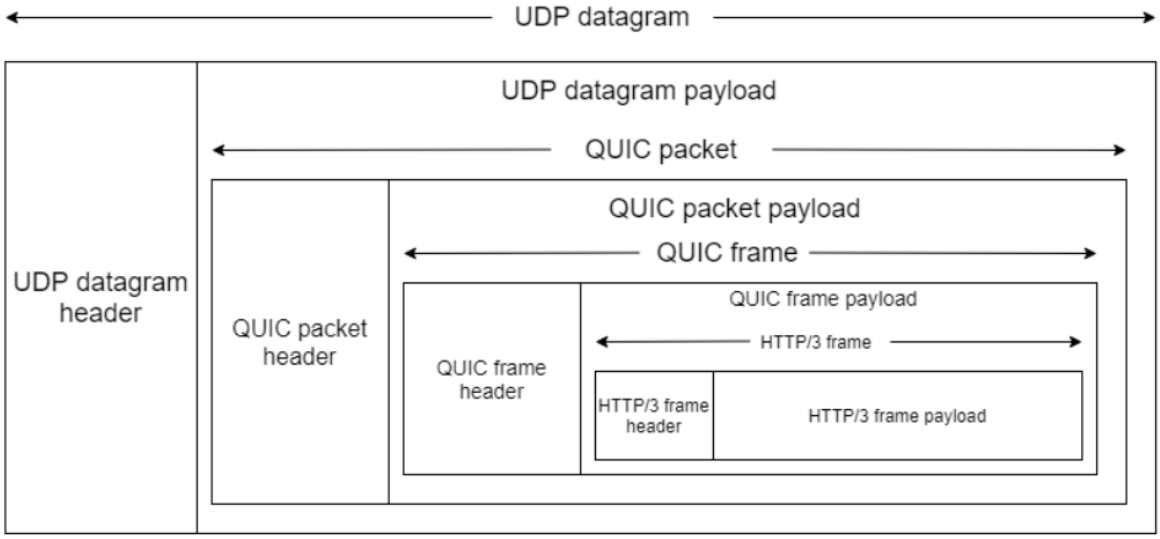
次の図は、Citrix ADC アプライアンスでの QUIC および HTTP/3 接続管理と、コンポーネントがどのように相互作用するかを示しています。

ステップ 1: Citrix ADC アプライアンスへの QUIC プロトコル経由のクライアント側の HTTP/3 リクエスト。
ステップ 2: バックエンドサーバーのサポートに応じて、Citrix ADC AS HTTP/1.1 または HTTP/2 によって転送されるリクエスト。
ステップ 3: バックエンドサーバーから Citrix ADC への HTTP/2 または HTTP/1.1 を介して応答します。
ステップ 4: ADC は HTTP/3 応答として応答をクライアントに転送します。

HTTP/3 プロトコルのしくみ

HTTP/3 では、クライアントが特定のエンドポイントに HTTP/3 サーバーが存在することを認識すると、QUIC 接続を開きます。QUIC プロトコルは、多重化およびフロー制御を提供します。各ストリーム内では、HTTP/3 通信の基本単位はフレームです。フレームタイプごとに異なる目的を果たします。たとえば、HEADERS フレームと DATA フレームは HTTP リクエストとレスポンスの基礎を形成します。

要求の多重化は、QUIC ストリーム抽象化を使用して実行されます。各リクエストとレスポンスのペアは、単一の QUIC ストリームを消費します。ストリームは互いに独立しているため、ブロックされている、またはパケット損失が発生する 1 つのストリームは、他のストリームでの進行を妨げません。サーバープッシュは HTTP/2 で導入されたインタラクションモードで、指定された要求を行うクライアントを予測して、サーバーが要求と応答交換をクライアントにプッシュできるようにします。これにより、潜在的なレイテンシーゲインとネットワーク使用率がトレードオフされます。PUSH_PROMISE、MAX_PUSH_ID、CANCEL_PUSH など、サーバープッシュの管理にはいくつかの HTTP/3 フレームが使用されます。HTTP/2 と同様に、リクエストフィールドとレスポンスフィールドは送信用に圧縮されます。HPACK は圧縮フィールドセクションの順序通りの送信（QUIC によって提供されていない保証）に依存しているため、HTTP/3 は HPACK を QPACK に置き換えます。QPACK は、個別の単方向ストリームを使用してフィールドテーブルの状態を変更および追跡します。一方、エンコードされたフィールドセクションは、テーブルの状態を変更せずに表の状態を参照します。



HTTP/3 の設定と統計の概要

July 15, 2022

QUIC を使用して HTTP/3 データの複数のストリームを送信するための HTTP/3 プロトコルを構成するには、次の手順を実行する必要があります。

1. SSL および負荷分散機能を有効にします。
2. HTTP_QUIC タイプの負荷分散とコンテンツスイッチング（オプション）仮想サーバーを追加します。
3. QUIC プロトコルパラメータを HTTP_QUIC 仮想サーバに関連付けます。
4. HTTP_QUIC 仮想サーバーで HTTP/3 を有効にします。
5. SSL 証明書とキーのペアを HTTP_QUIC 仮想サーバーとバインドします。
6. SSL/TLS プロトコルパラメータを HTTP_QUIC 仮想サーバーに関連付けます。

SSL と負荷分散を有効にする

開始する前に、アプライアンスで SSL 機能と負荷分散機能が有効になっていることを確認します。コマンドプロンプトで次のように入力します。

```
enable ns feature ssl lb
<!--NeedCopy-->
```

HTTP/3 サービスのタイプ HTTP_QUIC の負荷分散とコンテンツスイッチング (オプション) 仮想サーバーを追加します

QUIC 経由で HTTP/3 トラフィックを受け入れるように、負荷分散仮想サーバーを追加します。
注: タイプ HTTP_QUIC の負荷分散仮想サーバーには、QUIC、SSL、および HTTP3 プロファイルが組み込まれています。ユーザー定義プロファイルを作成する場合は、新しいプロファイルを追加し、負荷分散仮想サーバーとバインドできます。

```
add lb vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-
    port>
add cs vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-
    port>
<!--NeedCopy-->
```

例:

```
add lb vserver lb-http3 HTTP_QUIC 1.1.1.1 443
add cs vserver cs-http3 HTTP_QUIC 10.10.10.10 443
```

QUIC プロトコルパラメータを HTTP_QUIC 仮想サーバに関連付ける

QUIC プロファイルを作成し、QUIC サービスの QUIC パラメータを指定し、それを負荷分散仮想サーバに関連付けることができます。ユーザー定義プロファイルを作成するか、組み込みの QUIC プロファイルを使用して、プロファイルを負荷分散仮想サーバーにバインドする必要があります。

ステップ 1: ユーザ定義の QUIC
プロファイルを設定するコマンドプロンプトで、次のように入力します。

```
set quic profile <profile_name> -transport_param <value>
<!--NeedCopy-->
```


例:

```
set quic profile quic_http3 -ackDelayExponent 10 -activeConnectionIDlimit 4
```

異なる QUIC トランスポートパラメータは次のとおりです。

-ackDelayExponent. Citrix ADC がリモート QUIC エンドポイントにアドバタイズする整数値。リモート QUIC エンドポイントが Citrix ADC によって送信される QUIC ACK フレームの ACK 遅延フィールドをデコードするために使用する必要のある指数を示します。

-activeConnectionIDlimit. Citrix ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値です。Citrix ADC が保存するリモート QUIC エンドポイントからの QUIC 接続 ID の最大数を指定します。

-activeConnectionMigration. Citrix ADC がリモート QUIC エンドポイントでアクティブな QUIC 接続移行を実行できるようにする必要があるかどうかを指定します。

-congestionCtrlAlgorithm. QUIC 接続に使用する輻輳制御アルゴリズムを指定します。

-initialMaxData. Citrix ADC がリモート QUIC エンドポイントにアドバタイズする整数値で、QUIC 接続で送信できる最大データ量の初期値をバイト単位で指定します。

-initialMaxStreamDataBidiLocal. Citrix ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。Citrix ADC によって開始される双方向 QUIC ストリームの初期フロー制御制限（バイト単位）を指定します。

-initialMaxStreamDataBidiRemote. Citrix ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントによって開始される双方向 QUIC ストリームの初期フロー制御制限（バイト単位）を指定します。

-initialMaxStreamDataUni. Citrix ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントによって開始される単方向ストリームの初期フロー制御制限（バイト単位）を指定します。

-initialMaxStreamsBidi. Citrix ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントが開始する必要がある双方向ストリームの初期最大数を指定します。

-initialMaxStreamsUni. Citrix ADC によってリモート QUIC エンドポイントにアドバタイズされる整数値。リモート QUIC エンドポイントが開始する必要のある単一方向ストリームの初期最大数を指定します。

-maxAckDelay. Citrix ADC がリモート QUIC エンドポイントにアドバタイズする整数値。Citrix ADC が確認応答の送信を遅延する最大時間をミリ秒単位で指定します。

-maxIdleTimeout. Citrix ADC がリモート QUIC エンドポイントにアドバタイズする整数値。QUIC 接続の最大アイドルタイムアウトを秒単位で指定します。Citrix ADC とリモート QUIC エンドポイントによってアドバタイズされるアイドルタイムアウト値の最小値よりも長くアイドル状態のままであり、現在のプローブタイムアウト（PTO）の 3 倍の QUIC 接続は、Citrix ADC によってサイレントに破棄されます。

-maxUDPPayloadSize. Citrix ADC がリモート QUIC エンドポイントにアドバタイズする整数値。Citrix ADC が QUIC 接続で受信する最大の UDP データグラムペイロードのサイズをバイト単位で指定します。

-newTokenValidityPeriod. Citrix ADC によって送信される QUIC NEW_TOKEN フレームを介して発行されたアドレス検証トークンの有効期間を秒単位で指定する整数値です。
-retryTokenValidityPeriod. Citrix ADC によって送信された QUIC 再試行パケットを介して発行されるアドレス検証トークンの有効期間を秒単位で指定する整数値です。
-statelessAddressValidation. Citrix ADC が QUIC クライアントのステートレスアドレス検証を実行する必要があるかどうかを指定します。QUIC 接続の確立中に QUIC 再試行パケットでトークンを送信し、QUIC 接続の確立後に QUIC NEW_TOKEN フレームでトークンを送信する必要があります。

ステップ 2: ユーザ定義の QUIC プロファイルを http_quic タイプの負荷分散仮想サーバーに関連付けます。

コマンドプロンプトで入力します。

```
set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
    serviceName>@] [-persistenceType <persistenceType>] [-
    quicProfileName <string>]
<!--NeedCopy-->
```

例:

```
set lb vserver lb-http3 -quicProfileName quic_http3
```

HTTP_QUIC 仮想サーバーで HTTP/3 を有効にしてバインドする

HTTP_QUIC 仮想サーバーで HTTP/3 を有効にするには、設定パラメータのセットが HTTP プロファイル設定に追加されます。設定を容易にするため、HTTP_QUIC 仮想サーバーを追加すると、新しいデフォルト/組み込みの HTTP プロファイルがアプライアンスで使用可能になります。プロファイルの HTTP/3 プロトコルサポートパラメータが ENABLED に設定され、HTTP_QUIC 仮想サーバにも制限されます（HTTP_QUIC 仮想サーバをユーザが追加した HTTP プロファイルに関連付けない場合に適用可能）。HTTP プロファイルの HTTP/3 パラメータの値は、QUIC プロトコルハンドシェイク中に TLS ALPN（アプリケーション層プロトコルネゴシエーション）拡張を処理するときに HTTP/3 プロトコルを選択し、アドバタイズするかどうかを決定します。

HTTP/3 プロファイルを作成し、HTTP/3 サービスと負荷分散仮想サーバーの HTTP パラメータを指定できます。ユーザー定義プロファイルを作成するか、組み込みの HTTP/3 プロファイルを使用して、プロファイルを負荷分散仮想サーバーにバインドする必要があります。

ステップ 1: ユーザ定義の HTTP/3
プロファイルを設定するコマンドプロンプトで、次のように入力します。

```
Add ns httpProfile <profile_name> -http3 ENABLED
<!--NeedCopy-->
```

例:


```
add ns httpProfile http3_quic -http3 ENABLED
```

ステップ 2: ユーザー定義の HTTP/3 プロファイルを http_quic
タイプの負荷分散仮想サーバーにバインドするコマンドプロンプトで、次のように入力します。

```
1 set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
     serviceName>@] [-persistenceType <persistenceType>] [-
     httpProfileName <string>]
2 <!--NeedCopy-->
```

例:

```
set lb vserver lb-http3 -httpProfileName http3_quic
```

SSL 証明書とキーのペアを HTTP_QUIC 仮想サーバーでバインドする

暗号化されたトラフィックを処理するには、SSL 証明書とキーのペアを追加し、HTTP_QUIC 仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します。

```
1 bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
2
3 <!--NeedCopy-->
```

例:

```
bind ssl vserver lb-http3 -certkeyName rsa_certkeypair
```

詳細については、「SSL 証明書のバインド」トピックを参照してください。

SSL/TLS プロトコルパラメーターを HTTP_QUIC 仮想サーバーでバインドする

HTTP_QUIC タイプの仮想サーバーには、QUIC プロトコルが必須のセキュリティコンポーネントとして TLS 1.3 を使用するため、組み込みの TLS 1.3 サーバー機能があります。HTTP_QUIC 仮想サーバーを追加する際の設定を容易にするため、タイプ Quic-Frontend の新しいデフォルトまたは組み込みの SSL プロファイルが追加されます。SSL プロファイルでは、TLS 1.3 暗号化スイート（および楕円曲線）が設定された TLS 1.3 バージョンが有効になっています。SSL プロファイルは、新しく追加された HTTP_QUIC 仮想サーバーにバインドする必要があります。
SSL プロファイルを作成し、TLP 1.1 サービスと負荷分散仮想サーバーの SSL 暗号化パラメーターを指定できます。ユーザー定義プロファイルを作成するか、組み込みの SSL プロファイルを使用して、プロファイルを負荷分散仮想サーバーにバインドする必要があります。

ステップ 1: ユーザ定義の SSL
プロファイルを構成するコマンドプロンプトで、次のように入力します。

```
add ssl profile <name> -sslprofileType QUIC-FrontEnd
<!--NeedCopy-->
```

例:

```
add ssl profile ssl_profile1 -sslprofileType QUIC-FrontEnd -tls13 ENABLED -
tls12 DISABLED -tls11 DISABLED -tls1 DISABLED
```

ステップ 2: ユーザー定義の SSL プロファイルを HTTP_QUIC
タイプの負荷分散仮想サーバーにバインドするコマンドプロンプトで、次のように入力します。

```
set ssl vserver <name>@ [-sslProfile <string>]
<!--NeedCopy-->
```

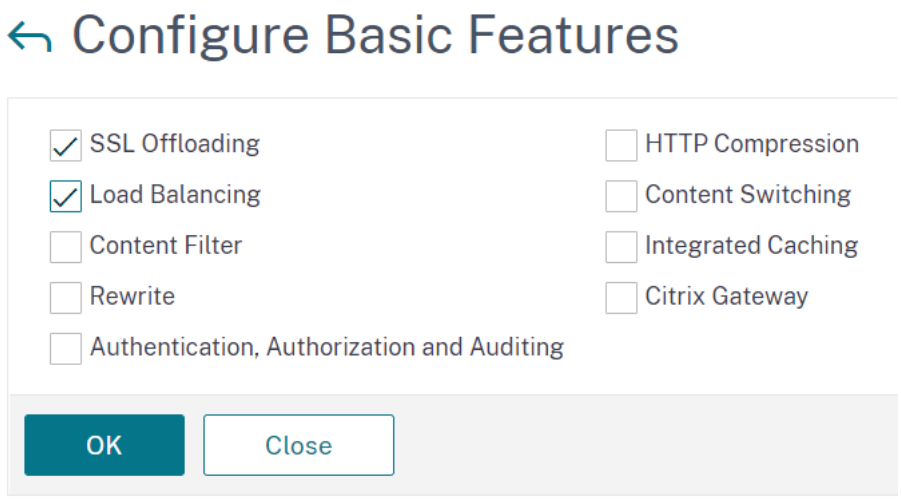
例:

```
set ssl vserver lb-http3 -sslprofile ssl_profile1
```

GUI を使用して SSL および負荷分散機能を有効にする

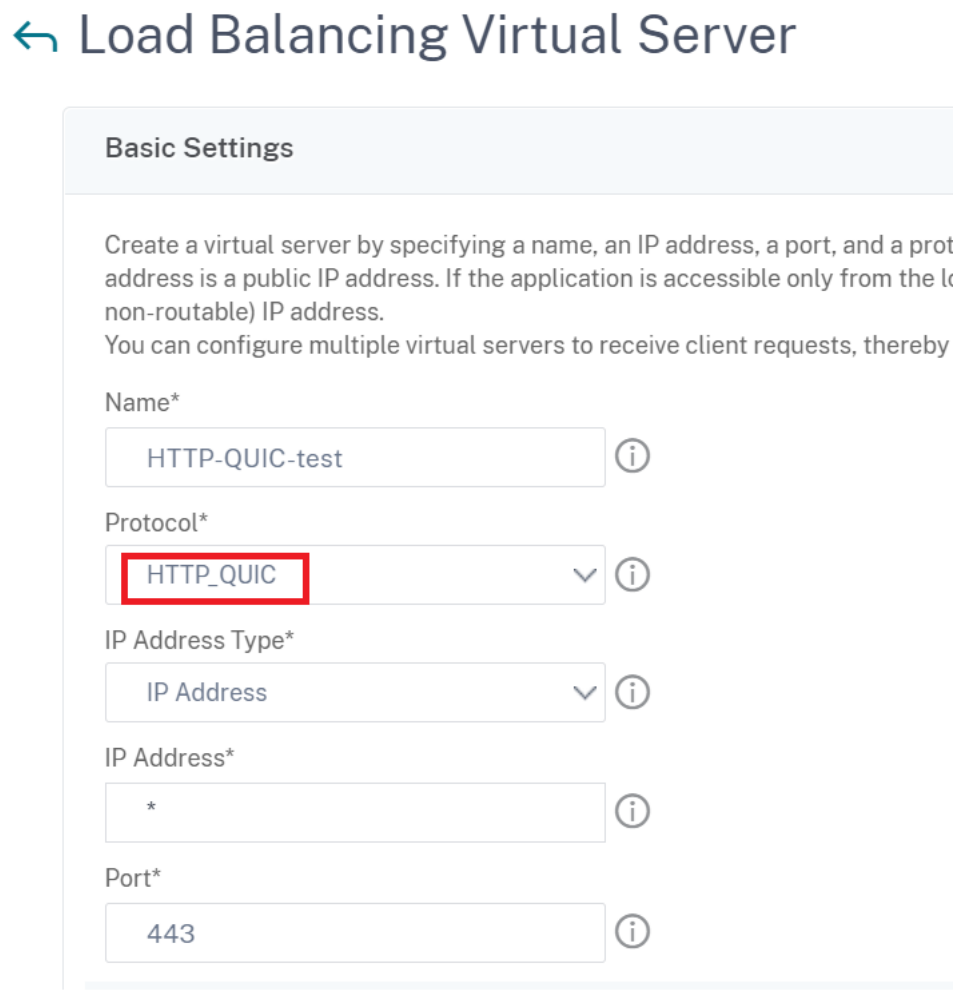
SSL および負荷分散機能を有効にするには、次の手順を実行します。

1. ナビゲーションウィンドウで、[システム] を展開し、[設定] をクリックします。
2. [基本機能の構成] ページで、[**SSL** と負荷分散] を選択します。
3. 「**OK**」をクリックし、「閉じる」をクリックします。



GUI を使用して **HTTP_QUIC** タイプの負荷分散とコンテンツスイッチング（オプション）仮想サーバーを追加します

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. [追加] をクリックして、タイプ HTTP_QUIC の負荷分散仮想サーバーを作成します。
3. **Load Balancing Virtual Server** ページで **Profiles** をクリックします。
4. [プロファイル] セクションで、プロファイルタイプを [QUIC] として選択します。注:QUIC、HTTP/3、および SSL プロファイルは組み込みのプロファイルです。
5. **OK** をクリックしてから、「完了」をクリックします。



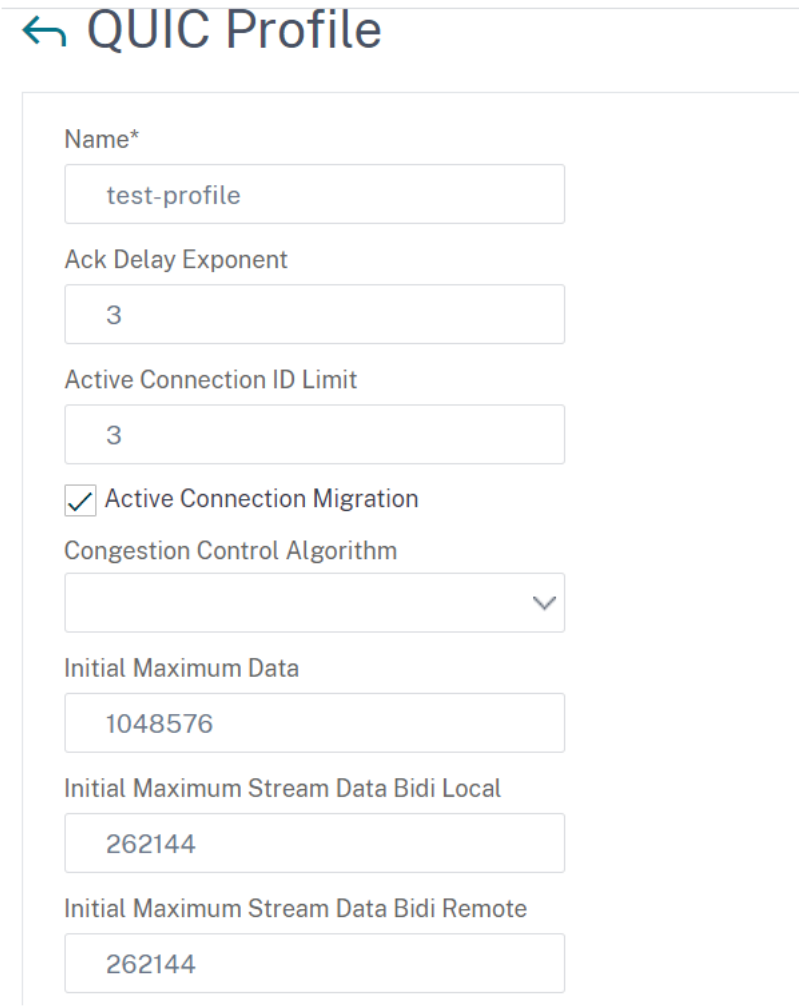
GUI を使用して **QUIC** プロトコルパラメータを **HTTP_QUIC** 仮想サーバに関連付けます

ステップ 1: QUIC プロファイルを追加する

1. [システム] **>** [プロファイル] **>** [**QUIC** プロファイル] に移動します。
2. [追加] をクリックします。

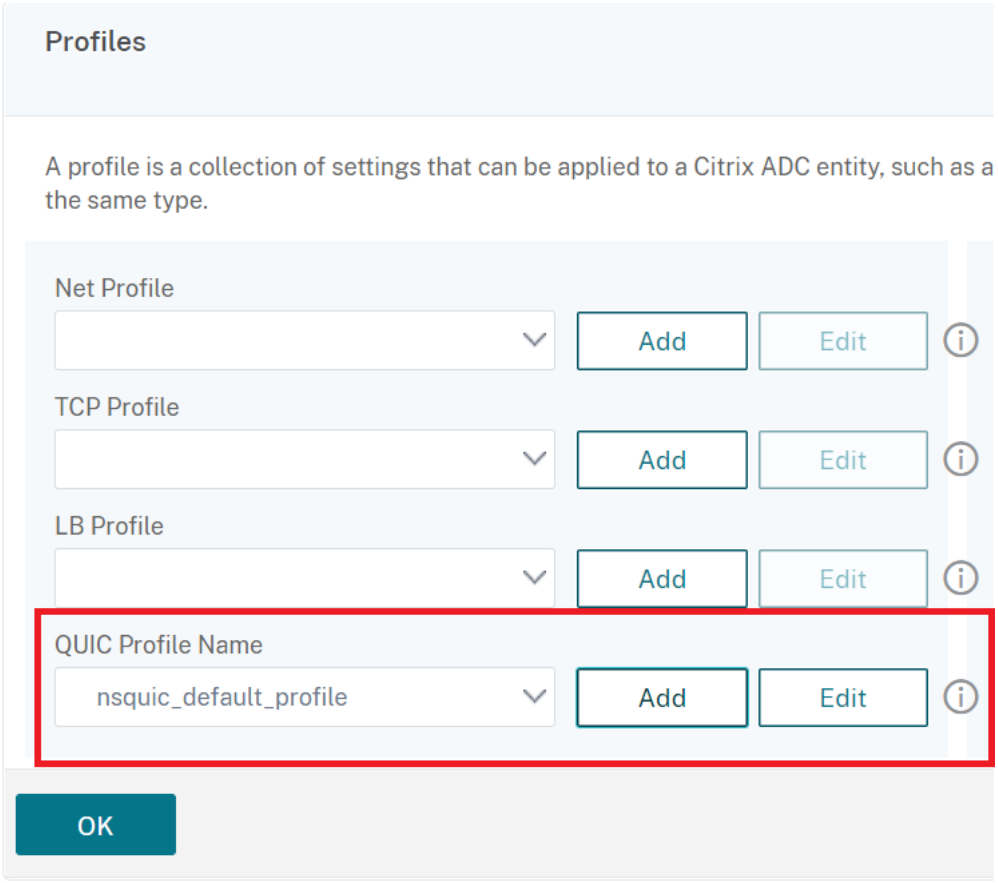
3. [QUIC プロファイル] ページで、次のパラメータを設定します。各パラメータの詳細については、「QUIC プロトコル CLI の関連付け」セクションを参照してください。

   a) `Ack Delay` 指数
   b) アクティブ接続 ID 制限
   c) アクティブな接続の移行
   d) 輻輳制御アルゴリズム
   e) 初期最大データ
   f) 初期最大ストリームデータ Bidi ローカル
   g) 初期最大ストリームデータ Bidi リモート
   h) 初期最大ストリームデータ単位
   i) 初期最大ストリーム bidi
   j) 初期最大ストリーム単位
   k) 最大確認応答遅延
   l) 最大アイドルタイムアウト
   m) バーストあたりの最大 UDP データグラム数
   n) 新しいトークンの有効期間
   o) トークンの有効期間を再試行
   p) ステートレスアドレス検証



ステップ 2: QUIC プロファイルを HTTP_QUIC タイプの負荷分散仮想サーバーに関連付ける

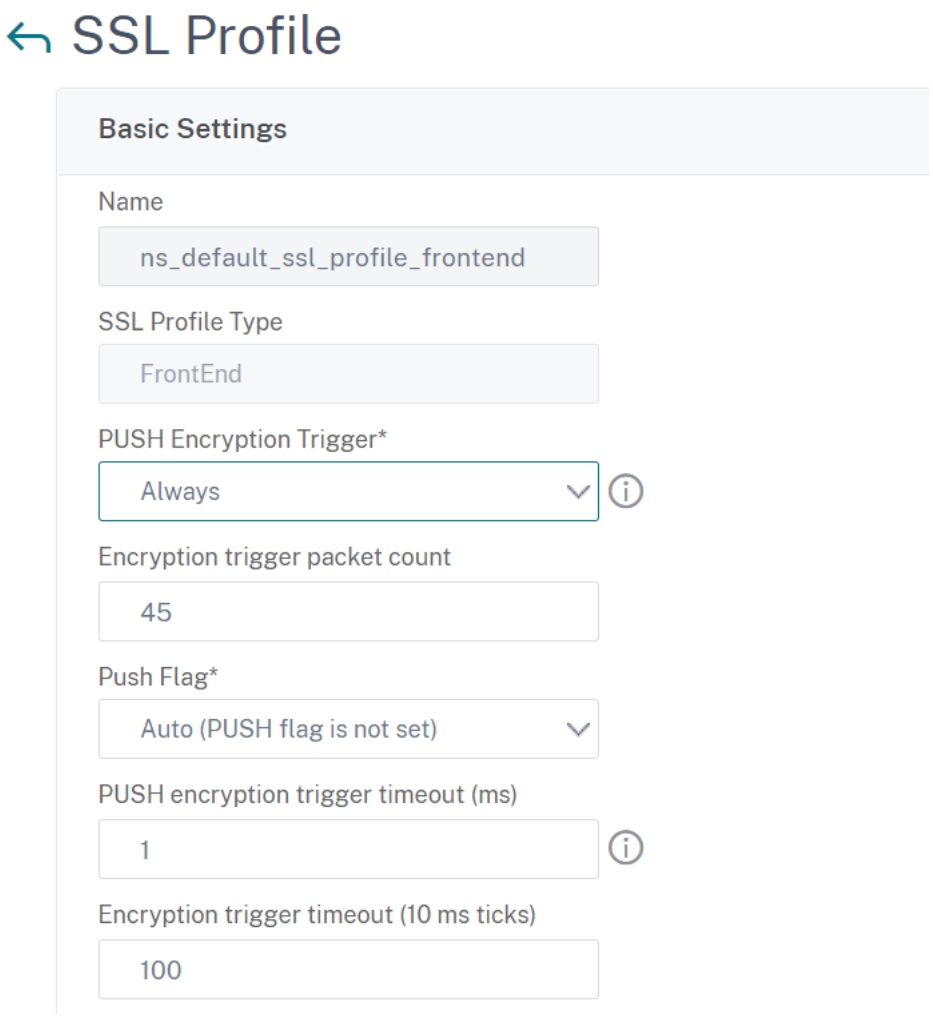
1. [プロファイル] セクションで、QUIC プロファイルを選択します。注:QUIC、HTTP/3、および SSL プロファイルは組み込みのプロファイルです。
2. **OK** をクリックしてから、「完了」をクリックします。



GUI を使用して、**SSL/TLS** プロトコルパラメータを **SSL** タイプの仮想サーバーに関連付けます

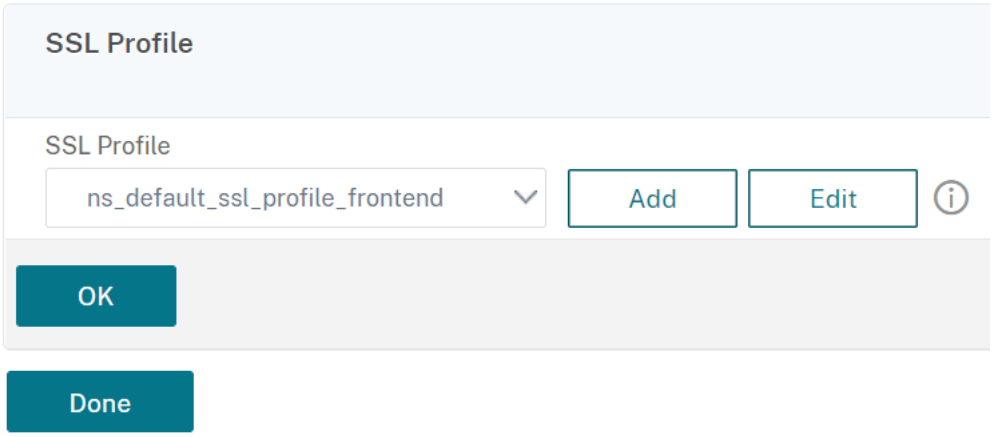
ステップ 1: SSL プロファイルを追加する

1. システム **>** プロファイル **> SSL** プロファイルに移動します。
2. [追加] をクリックします。
3. [**QUIC** プロファイル] ページで、SSL パラメータを設定します。詳細な説明については、SSL プロファイルの構成トピックを参照してください。
4. [**OK**] をクリックして閉じます。



ステップ 2: SSL プロファイルを SSL タイプの負荷分散仮想サーバーに関連付けます。

1. [プロファイル] セクションで、SSL プロファイルを選択します。
2. **OK** をクリックしてから、「完了」をクリックします。



QUIC、および **HTTP/3** の統計情報を表示する

次のコマンドは、QUIC および HTTP3 統計の詳細なサマリーを表示します。コマンドプロンプトで、次のように入力します。

```
> stat quic
> stat quic  - detail
<!--NeedCopy-->
```

統計情報の表示をクリアするには、次のいずれかを入力します。

```
> stat quic -clearstats basic
> stat quic -clearstats full

<!--NeedCopy-->
```

HTTP/3 統計の詳細なサマリーを表示するには、次の手順を実行します。

```
> stat http3
> stat http3  - detail
<!--NeedCopy-->
```

統計情報の表示をクリアするには、次のいずれかを入力します。

```
> stat http3 -clearstats basic
> stat http3 -clearstats full
<!--NeedCopy-->
```

HTTP/3 トラフィックのポリシー設定

October 7, 2021

HTTP/3 は UDP に基づく QUIC トランスポートを使用します。TCP ポリシー式を含む HTTP または SSL 仮想サーバに対して定義されたポリシー式がある場合、HTTP_QUIC 仮想サーバでは使用できなくなります。TCP またはクラシック式を持たない他のすべてのポリシーは、HTTP_QUIC 仮想サーバーでバインドできます。ポリシーを有効にするには、次のように、機能ポリシーが新しく追加されたグローバルバインドポイントにバインドされていることを確認する必要があります。

- HTTPQUIC_REQ_DEFAULT
- HTTPQUIC_REQ_OVERRIDE
- HTTPQUIC_RES_DEFAULT
- HTTPQUIC_RES_OVERRIDE

または、ポリシーを特定の仮想サーバーのバインドポイントにバインドできます。

- REQUEST
- RESPONSE

詳細については、「高度なポリシーインフラストラクチャを使用したポリシーのバインド」トピックを参照してください。

HTTP over QUIC 設定でサポートされているポリシーは次のとおりです。

- レスポンダー
- リライト
- HTTP 圧縮
- 統合キャッシング
- Web アプリケーションファイアウォール
- URL 変換
- SSL
- フロントエンド最適化 (FEO)
- AppQoE

HTTP/3 トラフィックのレスポンダポリシー設定

HTTP over QUIC タイプの仮想サーバには、レスポンダポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。

TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 QUIC 仮想サーバーまたは QUIC グローバルバインドポイント経由の HTTP にバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

URL をリダイレクトするためのレスポンダーアクションを追加する

レスポンダーアクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add responder action <name> <type> (<target> | <htmlpage>) [-comment <
     string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
     expression>] [-headers <name(value)> ...]
2 <!--NeedCopy-->
```


例:

```
add responder action redirectURL redirect ”\”https://www.citrix.com/\””
```

レスポンダーポリシーの追加

レスポンダーポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```

例:

```
add responder policy res-pol ”CLIENT.IP.SRC.IN_SUBNET(10.10.10.10/32)”
redirectURL
```

レスポンダーポリシーベースの **UDP** 式を追加する

レスポンダーポリシーベースの UDP 式を追加するには、コマンドプロンプトで次のように入力します。

```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```

例:

```
add responder policy redirectCitrixUdp ”CLIENT.UDP.DSTPORT.EQ(443)”redirectURL
```

HTTP/3 QUIC ベースの負荷分散仮想サーバーでレスポンダーポリシーベースの **UDP** 式をバインドする

レスポンダーポリシーベースの UDP 式を負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
     ) | <serviceGroupName>@ |          (-policyName <string>@  [-
     priority <positive_integer>]  [-gotoPriorityExpression <expression>]
       [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
     analyticsProfile <string>@)
2 <!--NeedCopy-->
```


例:

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 9 -gotoPriorityExpres
 END -type REQUEST
```

HTTP/3 QUIC ベースの負荷分散仮想サーバーでレスポンダーポリシーをバインドする

レスポンダーポリシーを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
      ) | <serviceGroupName>@ |          (-policyName <string>@  [-
     priority <positive_integer>]  [-gotoPriorityExpression <expression>]
       [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
     analyticsProfile <string>@)
2 <!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 10 -
gotoPriorityExpression END -type REQUEST
```

HTTP/3 グローバルバインドポイントにレスポンダーポリシーをバインドする

HTTP/3 グローバルバインドポイントでレスポンダーポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind responder global <policyName> <priority>  [<gotoPriorityExpression
     >] [-type <type>] [-invoke (<labelType>  <labelName>) ] bind
     responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
2 <!--NeedCopy-->
```

例:

```
bind responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
```

> 注:
>
> 詳細については、レスポンダーポリシーのドキュメントを参照してください。

HTTP/3 トラフィックのポリシー設定を書き換え

HTTP over QUIC タイプの仮想サーバには、書き換えポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。

TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

次に、QUIC 上の HTTP3 の書き換えポリシーを設定する設定手順を示します。

QUIC 上の **HTTP** の書き換えアクションを追加

書き換えアクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
     pattern <expression> | -search <expression>] [-refineSearch <
     expression>] [-comment <string>]
2 <!--NeedCopy-->
```

例:

```
add rewrite action http3-altsvc-action insert_http_header Alt-Svc q/"h3
-29=\":443\"; ma=3600; persist=1"/
```

HTTP over QUIC の書き換えポリシーを追加する

書き込みアクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add rewrite policy <name> <rule> <action> [<undefAction>] [-comment  <
     string>] [-logAction <string>]
2 <!--NeedCopy-->
```

例:

```
add rewrite policy http3-altsvc-policy true http3-altsvc-action
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーに書き換えポリシーをバインドする

書き換えポリシーを負荷分散仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] )
      | <serviceGroupName>@ | (-policyName <string>@ [-priority <
     positive_integer>] [-gotoPriorityExpression <expression>] [-type <
     type>] [-invoke (<labelType> <labelName>) ] ) | -analyticsProfile  <
     string>@)
```



```
<!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName http3-altsvc-policy -priority 10 -type
 RESPONSE
```

HTTP/3 グローバルバインドポイントに書き換えポリシーをバインドする

```
To bind a responder policy with HTTP/3 global bind point, at the
    command prompt, type:
bind rewrite global <policyName> <priority> [<gotoPriorityExpression>]
     [-type <type>] [-invoke (<labelType> <labelName>)]
<!--NeedCopy-->
```

例:

```
bind rewrite global http3-altsvc-policy 3 -type HTTPQUIC_RES_DEFAULT
```

> 注:
>
> 詳細については、「ポリシーの書き換え」を参照してください。

HTTP/3 トラフィックの圧縮ポリシー設定

Citrix ADC はサーバーから HTTP 応答を受信すると、組み込みの圧縮ポリシーとカスタム圧縮ポリシーを評価して、応答を圧縮するかどうか、圧縮する場合は適用する圧縮の種類を決定します。ポリシーに割り当てられた優先順位によって、ポリシーがリクエストに対して照合される順序が決まります。
HTTP over QUIC タイプの仮想サーバには、圧縮ポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

圧縮ポリシーの追加

圧縮ポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
add cmp policy <name> -rule <expression> -resAction <string>
<!--NeedCopy-->
```


例:

```
add cmp policy udp_port_cmp_policy -rule ”CLIENT.UDP.DSTPORT.EQ(443)”-
resAction COMPRESS
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーで圧縮ポリシーをバインドする

HTTP/3_QUIC タイプの負荷分散仮想サーバーで URL 変換ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
     ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
     positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
      REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
     -analyticsProfile <string>@)
2 <!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName udp_port_cmp_policy -priority 10 -type
 RESPONSE
```

HTTP/3 グローバルバインドポイントにグローバル圧縮をバインドする

HTTP/3 グローバルバインドポイントで圧縮ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind compression global <policyName> <priority>  [<
     gotoPriorityExpression>] [-type <type>] [-invoke (<labelType>  <
     labelName>) ] bind responder global redirectCitrixUdp 3 -type
     HTTPQUIC_REQ_DEFAULT
2 <!--NeedCopy-->
```

例:

```
bind cmp global udp_port_cmp_policy -priority 100 -type HTTPQUIC_RES_DEFAULT
 Global built-in compression policies
```

アプライアンスを Citrix ADC リリース 13.0 ビルド 82.x にアップグレードすると、次の圧縮ポリシーが HTTP/3 のデフォルトバインドポイントに自動的にバインドされます。


```
> sho cmp global -type HTTPQUIC_RES_DEFAULT
        Policy Name: ns_adv_nocmp_xml_ie
        Priority: 8700
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_nocmp_mozilla_47
        Priority: 8800
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_mscss
        Priority: 8900
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_msapp
        Priority: 9000
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_content_type
        Priority: 10000
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT
<!--NeedCopy-->
```

バインドされていない場合、次のコマンドはコマンドプロンプトで設定でき、アプライアンスで設定できます。

```
bind cmp global ns_adv_nocmp_xml_ie -priority 8700 -gotoPriorityExpression
END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_nocmp_mozilla_47 -priority 8800 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_mscss -priority 8900 -gotoPriorityExpression END
 -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_msapp -priority 9000 -gotoPriorityExpression END
 -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_content_type -priority 10000 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT
```

詳細については、「圧縮ポリシーの設定」を参照してください。

HTTP/3 トラフィックのキャッシュポリシー設定

統合キャッシュは、Citrix ADC アプライアンスのメモリ内ストレージを提供し、オリジンサーバーへの往復を必要とせずにユーザーに Web コンテンツを提供します。静的コンテンツの場合、統合キャッシュの初期設定はほとんど必要ありません。統合キャッシュ機能を有効にして基本セットアップ（たとえば、キャッシュが使用を許可されている Citrix ADC アプライアンスメモリの量を決定する）を実行すると、統合キャッシュは組み込みポリシーを使用して、次のような特定のタイプの静的コンテンツを保存および提供します。シンプルなウェブページと画像ファイル。統合キャッシュを構成して、Web サーバーおよびアプリケーションサーバーによってキャッシュ不可としてマークされた動的コンテンツ（データベースレコードや株価など）を保存および提供することもできます。
HTTP over QUIC タイプの仮想サーバには、キャッシュポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。

TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

キャッシュコンテンツグループの追加

キャッシュコンテンツグループを追加するには、コマンドプロンプトで次のように入力します。

```
1 add cache contentGroup <name> [-weakPosRelExpiry <secs> | -relExpiry <
      secs> | -relExpiryMilliSec <msecs> | -absExpiry <HH:MM> ... | -
      absExpiryGMT <HH:MM> ...] [-heurExpiryParam <positive_integer>] [-
      weakNegRelExpiry <secs>] [-maxResSize <KBytes>] [-memLimit <MBytes>]
      ...
2 <!--NeedCopy-->
```

例:

```
add cache contentGroup DEFAULT -maxResSize 500
```

キャッシュポリシーの追加

キャッシュポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
1 add cache policy <policyName> -rule <expression> -action <action>  [-
      storeInGroup <string>] [-invalGroups <string> ...] [-invalObjects <
      string> ...] [-undefAction ( NOCACHE | RESET )] add cache policy <
      name> <rule> <profileName> [-comment <string>]  [-logAction <string
      >]
2 <!--NeedCopy-->
```


例:

```
add cache policy ctx_doc_pdf -rule ”HTTP.REQ.URL.ENDSWITH(\”.pdf\”)”-action
 CACHE -storeInGroup DEFAULT
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーでキャッシュポリシーをバインドする

HTTP/3_QUIC タイプの負荷分散仮想サーバーでキャッシュポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
     ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
     positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
      REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
     -analyticsProfile <string>@)
2 <!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName ctx_doc_pdf -priority 100 -type
REQUEST
```

HTTP/3 グローバルバインドポイントにグローバルキャッシュポリシーをバインドする

キャッシュポリシー HTTP/3 グローバルバインドポイントをバインドするには、次の手順を実行します。

```
1 bind cache global <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

例:

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

詳細については、統合キャッシュポリシーの設定を参照してください。

グローバル組み込みキャッシュポリシー

アプライアンスを Citrix ADC リリース 13.0 ビルド 82.x にアップグレードすると、次のキャッシュポリシーが自動的に HTTP/3 のデフォルトバインドポイントにバインドされます。

13.0 82.x リリースにアップグレードすると、次のキャッシュポリシーが HTTP/3 デフォルトのバインドポイントに自動的にバインドされます。

```
> sho cache global -type HTTPQUIC_REQ_DEFAULT
1)      Policy Name: NOPOLICY
        Priority: 185883
        GotoPriorityExpression: USE_INVOCATION_RESULT
        Invoke type: policylabel        Invoke name:
            _httpquicReqBuiltinDefaults
        Global bindpoint: HTTPQUIC_REQ_DEFAULT

 Done
> sho cache global -type HTTPQUIC_RES_DEFAULT
1)      Policy Name: NOPOLICY
        Priority: 185883
        GotoPriorityExpression: USE_INVOCATION_RESULT
        Invoke type: policylabel        Invoke name:
            _httpquicResBuiltinDefaults
        Global bindpoint: HTTPQUIC_RES_DEFAULT

<!--NeedCopy-->
```

アップグレード後、ポリシーがバインドされていない場合は、次のコマンドを使用して、手動で設定をバインドして保存できます。

```
add cache policylabel _httpquicReqBuiltinDefaults -evaluates
    HTTPQUIC_REQ

add cache policylabel _httpquicResBuiltinDefaults -evaluates
    HTTPQUIC_RES

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _nonGetReq -priority 100

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _advancedConditionalReq -priority 200

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _personalizedReq -priority 300

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheableStatusRes -priority 100
```



```
12
13 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _uncacheableVaryRes -priority 200
14
15 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _uncacheableCacheControlRes -priority 300
16
17 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _cacheableCacheControlRes -priority 400
18
19 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _uncacheablePragmaRes -priority 500
20
21 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _cacheableExpiryRes -priority 600
22
23 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _imageRes -priority 700
24
25 bind cache policylabel _httpquicResBuiltinDefaults -policyName
       _personalizedRes -priority 800
26
27 bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
       USE_INVOCATION_RESULT -type HTTPQUIC_REQ_DEFAULT -invoke policylabel
        _httpquicReqBuiltinDefaults
28
29 bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
       USE_INVOCATION_RESULT -type HTTPQUIC_RES_DEFAULT -invoke policylabel
        _httpquicResBuiltinDefaults
30
31 <!--NeedCopy-->
```

> 注:
>
> コマンドリスト内の最初の 2 つのコマンドと、同じリストの最後の 2 つのコマンドは完全性のために含まれています。アプライアンスの再起動時にコマンドがすでに実行されているため、4 つのコマンドの実行時にエラーが発生することがあります。しかし、これらのエラーは無視してもかまいません。

HTTP/3 トラフィックの **URL** 変換ポリシー設定

URL 変換は、外部ユーザーが参照する外部バージョンからの指定されたリクエストに含まれるすべての URL を、Web サーバーおよび管理者だけが参照する内部 URL に変更します。ネットワーク構造をユーザーに公開することなく、ユーザー要求をシームレスにリダイレクトできます。また、ユーザーが覚えにくい複雑な内部 URL を、よりシンプルで覚えやすい外部 URL に変更することもできます。

HTTP over QUIC タイプの仮想サーバには、キャッシュポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

URL 変換プロファイルを追加する

URL 変換プロファイルを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform profile <name> [-type URL]
2 <!--NeedCopy-->
```

例:

```
add transform profile msapps
```

URL 変換アクションを追加

URL 変換アクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform action <name> <profileName> <priority> [-state ( ENABLED
      | DISABLED )]
2 <!--NeedCopy-->
```

例:

```
add transform action docx2doc msapps 2
```

URL 変換アクションを追加

URL を置き換える URL 変換アクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform action <name> <profileName> <priority> [-state ( ENABLED
      | DISABLED )]
2 <!--NeedCopy-->
```

例:

```
add transform action docx2doc msapps 1
```


URL トランスフォームポリシーの追加

URL 変換ポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
1 add transform policy <name> <rule> <profileName> [-comment <string>]
      [-logAction <string>]
2 <!--NeedCopy-->
```

例:

```
add transform policy urltrans_udp ”CLIENT.UDP.DSTPORT.EQ(443)”msapps
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーで **URL** 変換ポリシーをバインドする

HTTP/3_QUIC タイプの負荷分散仮想サーバーで URL 変換ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
      ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
      positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
       REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
      -analyticsProfile <string>@)
2 <!--NeedCopy-->
```

例:

```
bind lb vs lb-http3 -policyName urltrans_udp -type REQUEST -priority 8
```

HTTP/3 QUIC ベースの負荷分散仮想サーバーで **URL** 変換ポリシーをグローバルにバインドする

URL 変換ポリシー HTTP/3 グローバルバインドポイントをバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind transform global <policyName> <priority>  [<gotoPriorityExpression
      >] [-type <type>] [-invoke (<labelType>  <labelName>) ]
2 <!--NeedCopy-->
```

例:

```
bind transform global urltrans_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

詳細については、URL 変換ポリシーの設定を参照してください。

HTTP/3 トラフィックに対するフロントエンド最適化 (FEO) ポリシー設定

Web アプリケーションの基礎となる HTTP プロトコルは、元々、単純な Web ページの送信とレンダリングをサポートするために開発されました。JavaScript やカスケードスタイルシート (CSS) などの新しいテクノロジや、Flash ビデオやグラフィックが豊富な画像などの新しいメディアタイプでは、フロントエンドのパフォーマンス、つまりブラウザーレベルでのパフォーマンスに多大な要求が課せられます。Citrix ADC フロントエンド最適化（FEO）機能は、このような問題に対処し、ウェブページの読み込み時間とレンダリング時間を短縮します。

> 注:
>
> HTTP_QUIC _Override/Default_Request タイプは FEO ポリシーグローバルバインディングではサポートされていません。

フロントエンド最適化 (FEO) アクションを追加

FEO アクションを追加するには、コマンドプロンプトで次のように入力します。

```
1 add feo action <name> [-pageExtendCache] [<cacheMaxage>][-
     imgShrinkToAttrib] [-imgGifToPng] [-imgToWebp] [-imgToJpegXR] [-
     imgInline] [-cssImgInline] [-jpgOptimize] [-imgLazyLoad] [-cssMinify
     ] [-cssInline] [-cssCombine] [-convertImportToLink] [-jsMinify] [-
     jsInline] [-htmlMinify] [-cssMoveToHead] [-jsMoveToEND][-
     domainSharding <string> <dnsShards> ...] [-clientSideMeasurements]
2
3 <!--NeedCopy-->
```

例:

```
add feo action feoact -imgGifToPng -pageExtendCache
```

フロントエンド最適化 (FEO) ポリシーの追加

FEO ポリシーを追加するには、コマンドプロンプトで次のように入力します。

```
add feo policy <name> <rule> <action>
```

例:

```
add feo policy udp_feo_img ”CLIENT.UDP.DSTPORT.EQ(443)”IMG_OPTIMIZE
```

HTTP/3_QUIC タイプの負荷分散仮想サーバーと FEO ポリシーをバインドします

HTTP/3_QUIC タイプの負荷分散仮想サーバーと FEO ポリシーをバインドするには、コマンドプロンプトで次のように入力します。


```
bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
     ) | <serviceGroupName>@ |          (-policyName <string>@  [-
    priority <positive_integer>]  [-gotoPriorityExpression <expression>]
      [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
    analyticsProfile <string>@)
<!--NeedCopy-->
```

例:

```
bind lb vserver lb-http3 -policyName udp_feo_img -priority 4 -gotoPriorityExpression
 END -type REQUEST
```

FEO ポリシーを **HTTP/3** グローバルバインドポイントにバインドする

HTTP/3 グローバルバインドポイントにキャッシュポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
bind cache global <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

例:

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

詳細については、「フロントエンド最適化ポリシーの設定」を参照してください。

HTTP/3 トラフィックの **SSL** ポリシー設定

HTTP over QUIC タイプの仮想サーバには SSL ポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。
TLSv1.3 でサポートされているアクションを持つ SSL ポリシーは、HTTP/3 バインドポイントまたは仮想サーバーにのみ適用されます。

SSL ポリシーの追加

FEO ポリシーを追加するには、コマンドプロンプトで次のように入力します。


```
add ssl policy <name> -rule <expression> [-action <string>] [-
    undefAction <string>] [-comment <string>]
<!--NeedCopy-->
```

例:

```
add ssl policy ssl-pol -rule CLIENT.SSL.IS_SSL -action NOOP
```

SSL ポリシーを HTTP/3 仮想サーバーにバインドする

SSL ポリシーを HTTP/3 仮想サーバーにバインドするには、コマンドプロンプトで次の手順を実行します。

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

例:

```
bind ssl vserver lb-http3 -policyName ssl-pol -priority 4 -type REQUEST
```

SSL ポリシーの UDP 式で SSL ポリシーを追加する

UDP 式で SSL ポリシーを追加するには、コマンドプロンプトで次の操作を行います。

```
add ssl policy <name> -rule <expression> [-action <string>] [-
    undefAction <string>] [-comment <string>]
<!--NeedCopy-->
```

例:

```
add ssl policy ssl_udp_clnt -rule ”CLIENT.UDP.DSTPORT.EQ(443)”-action NOOP
```

SSL ポリシーを UDP 式で HTTP/3 仮想サーバーにバインドする

HTTP/3 仮想サーバーに UDP 式を持つ SSL ポリシーをバインドするには、コマンドプロンプトで次のように入力します。


```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

例:

```
bind ssl vs lb-http3 -policyName ssl_udp_clnt -priority 8 -type REQUEST
```

HTTP/3 トラフィックの **CLIENTHELLO** バインドポイントの **SSL** ポリシーを追加する

HTTP/3 トラフィックの CLIENTHELLO バインドポイントの SSL ポリシーをバインドするには、コマンドプロンプトで次のように入力します。

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

例:

```
add ssl policy ssl-pol-ch -rule ”CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE
(0x1301)”-action RESET
```

SSL ポリシーを **CLIENTHello** バインドポイントにバインドする

SSL ポリシーを CLIENTHELLO バインドポイントにバインドするには、コマンドプロンプトで次のように入力します。

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

例:

```
bind ssl vs lb-http3 -policyName ssl-pol-ch -type CLIENTHELLO_REQ -priority
 100
```

SSL ポリシーを **HTTP/3** グローバルバインドポイントにバインドする

SSL ポリシーを HTTP/3 グローバルバインドポイントにバインドするには、コマンドプロンプトで次のように入力します。


```
bind cache global <policy> -priority <positive_integer> [-gotoPriorityExpression
 <expression>] [-type <type>] [-invoke (<labelType> <labelName>)]
```

例:

HTTP/3 グローバルバインドポイントにバインドされている DATA ポリシーの例を次に示します。

```
Bind ssl global -policyName ssl-pol-ch -priority 7 -type HTTPQUIC_DATA_DEFAULT
```

> 注:
>
> SSL 仮想サーバーの CLIENTHELLO バインドポイントに設定できる転送アクションは、HTTP_QUIC タイプの仮想サーバーでは現在サポートされていません。

HTTP/3 トラフィック用のアプリケーションファイアウォールポリシーの設定

HTTP over QUIC タイプの仮想サーバには、Web アプリケーションファイアウォールポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

UDP 式で Web アプリケーションファイアウォールポリシーを追加する

UDP 式を使用して Web アプリケーションファイアウォールポリシーを追加するには、コマンドプロンプトで次の操作を行います。

```
1 add appfw policy <name> <rule> <profileName> [-comment <string>] [-
     logAction <string>]
2 <!--NeedCopy-->
```

例:

```
add appfw policy appfw_udp ”CLIENT.UDP.DSTPORT.EQ(443)”APPFW_BYPASS
```

Web アプリケーションファイアウォールプロファイルの UDP ベースの式を使用してログ式をバインドする

UDP for Web アプリケーションファイアウォールプロファイルでログ式をバインドするには、コマンドプロンプトで次の操作を行います。

例:


```
bind appfw profile APPFW_BLOCK -logExpression logexp-1 ”CLIENT.UDP.DSTPORT.
EQ(443)”
```

HTTP/3 仮想サーバーでアプリケーションファイアウォールポリシーをバインドする

Web アプリケーションファイアウォールポリシーを HTTP/3 仮想サーバーでバインドするには、コマンドプロンプトで次の手順を実行します。

```
1 bind appfw policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

例:

```
bind lb vs lb-http3 -policyName appfw_udp -priority 3 -type REQUEST
```

Web アプリケーションファイアウォールポリシーを **HTTP/3** グローバルバインドポイントにバインドする

Web アプリケーションファイアウォールポリシーを HTTP/3 グローバルバインドポイントにバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind appfw global <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

例:

```
bind appfw global appfw_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

HTTP/3 トラフィックの **AppQoE** ポリシー設定

HTTP over QUIC タイプの仮想サーバーには AppQoE ポリシーがサポートされています。ただし、QUIC は転送メカニズムとして UDP を使用するため、TCP ベースの式は除外され、UDP ベースの式も含まれます。
TCP 式を使用した新規または既存のポリシー設定は、HTTP/3 仮想サーバーまたは新しく追加された HTTP/3 グローバルバインドポイントにバインドできません。TCP 式の代わりに、HTTP/3 QUIC 仮想サーバーまたは HTTP over QUIC バインドポイントにバインドされているポリシー設定に UDP 式を含めることができます。

UDP ベースの式で **AppQoE** ポリシーを追加する

UDP 式で AppQoE ポリシーを追加するには、コマンドプロンプトで次の操作を行います。

```
1 add AppQoE policy <name> <rule> <profileName> [-comment <string>] [-
     logAction <string>]
2 <!--NeedCopy-->
```

例:

```
add appqoe policy appqoe-pol-udp -rule ”CLIENT.UDP.DSTPORT.EQ(443)”-action
appqoe-act-basic-prhigh
```

AppQoE ポリシーを **HTTP/3** 仮想サーバーでバインドする

AppQoE ポリシーを HTTP/3 仮想サーバーとバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind appqoe policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

例:

```
bind lb vs lb-http3 -policyName appqoe-pol-udp -type REQUEST -priority 3
```

AppQoE ポリシーを **HTTP_QUIC** 仮想サーバーにバインドする

AppQoE HTTP_QUIC ポリシーを仮想サーバーにバインドするには、コマンドプロンプトで次のように入力します。

```
1 bind appqoe <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

例:

```
bind lb vs lb-http3 -policyName appqoe-pol-primd -priority 8 -type REQUEST
```


HTTP/3 サービスの検出

October 7, 2021

HTTP プロトコルは、同等のサービスの可用性をアドバタイズするために、オリジンサーバーに HTTP 代替サービスを使用することに依存しています。HTTP/3 サービスディスカバリも同じ原理を使用します。代替の HTTP/3 エンドポイントは、次のいずれかの方法を使用してアドバタイズできます。

- HTTP Alt-Svc レスポンスヘッダー
- HTTP/2 レスポンス内の Alt-Svc フレーム
- アプリケーション層プロトコルネゴシエーション (ALPN)

代替サービスは、HTTP Alt-Svc 応答ヘッダーと HTTP/2 Alt-Svc フレームの使用を HTTP/3 エンドポイントとしてアドバタイズします。サーバーは任意の UDP ポートで HTTP/3 を提供できます。代替サービスアドバタイズメントには明示的なポートが含まれ、URL にはスキームに関連付けられた明示的なポートまたはデフォルトポートのいずれかが含まれます。

代替サービスヘッダーまたはフレームを受信するクライアントは、それらを使用するようにバインドされません。クライアントは、代替サービスを認識し、代替サービスメカニズムをサポートしている場合は、アドバタイズされた適切な代替サービスを使用する必要があります。つまり、HTTP/1.1 サービスまたは HTTP/2 サービスは、HTTP/3 プロトコルをサポートする同等のエンドポイントをアドバタイズする可能性があります。この代替サービス情報を受信するクライアントは、指定された代替サービスとの QUIC 接続を確立することを選択できます。使用可能になると、この接続を後続の要求に使用できます。選択した代替サービスとの接続の確立に失敗した場合、クライアントは元のエンドポイントにフォールバックできます。クライアントがアドバタイズされた代替サービスの使用を開始すると、は Alt-Used ヘッダーを含めることでこれを示します。

Citrix ADC は、HTTP および SSL タイプの仮想サーバー上の同等の HTTP/3 エンドポイントの広告をサポートしています。

HTTP/3 サービスディスカバリを構成する

HTTP/3 サービスディスカバリを設定するには、次の手順を実行します。

1. HTTP Alt-Svc ヘッダーを使用して HTTP/3 代替サービスエンドポイントを構成する
2. HTTP/2 Alt-Svc フレームを使用した HTTP/3 代替サービスエンドポイントの設定 HTTP Alt-Svc ヘッダーを使用して HTTP/3
   代替サービスエンドポイントを構成する HTTP Alt-Svc ヘッダーを使用して HTTP/3
   エンドポイントをアドバタイズするには、次のコマンドを入力します。

注：代替サービスを宣伝する主な目的は、HTTP/1.1 または.b.cd: 443 の HTTP/2 サービスでも HTTP/3 機能にアクセスできることをユーザーに知らせることです。


```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ]
<!--NeedCopy-->
```

例:

```
add ns httpProfile http-profile -altsvc ENABLED -altSvcValue  ”h3-29=”
    :443”; ma=3600; persist=1”
<!--NeedCopy-->
```

または

```
set ns httpProfile http-custom -altsvc ENABLED -altSvcValue  ”h3-29=”
    :443”; ma=3600; persist=1”
<!--NeedCopy-->
```

HTTP/2 Alt-Svc フレームを使用して **HTTP/3** 代替サービスエンドポイントを構成する

HTTP/2 Alt-svc フレームを使用して HTTP/3 エンドポイントをアドバタイズするには、次のコマンドを入力します。

```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ] -
    http2AltSvcFrame [ ENABLED | DISABLED ]
<!--NeedCopy-->
```

例:

```
add ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -http2AltSvcFrame
 ENABLED -altsvc ENABLED -altSvcValue ”h3-29=\”:443\”; ma=3600; persist=1”
```

または

```
set ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -http2AltSvcFrame
 ENABLED -altsvc ENABLED -altSvcValue ”h3-29=\”:443\”; ma=3600; persist=1”
```

GUI を使用して **HTTP Alt-Svc** ヘッダー値を使用した **HTTP/3** 代替サービスを構成する

1. [システム] **>** [プロファイル] **>** [**HTTP** プロファイル] に移動します。
2. [追加] をクリックします。
3. [**HTTP** プロファイルの作成] ページで、[HTTP/3] セクションに移動し、[代替サービス] チェックボックスをオンにします。

4. http2 セクションに [代替サービス値] テキストボックスが表示されます。
5. 代替サービス値を”h3-29=”: 443”、ma=3600、永続 =1」と入力します。
6. [**OK]** をクリックして [閉じる] をクリックします。

gRPC

October 7, 2021

Citrix ADC アプライアンスの gRPC は、軽量で高性能なオープンソースのユニバーサルリモートプロシージャコール（RPC）フレームワークです。このフレームワークは、任意のオペレーティングシステムで実行されている複数の言語で機能するのに最適です。また、他のプロトコルと比較した場合、gRPC はより優れたパフォーマンスとセキュリティを提供します。

Citrix ADC 用の gRPC は、次の理由で推奨されます。

- データセンター向けの分散アプリケーションを構築し、public/private クラウドインフラストラクチャ。
- モバイル、Web、またはクラウドにクライアントサーバー通信を提供します。
- クラウドサービスとアプリケーションにアクセスする
- マイクロサービスのデプロイ

Why gRPC in Citrix ADC

Citrix ADC の gRPC は HTTP/2 高性能でスケーラブルな API をサポートします。テキストよりもバイナリを使用すると、ペイロードがコンパクトで効率的になります。Citrix ADC では、HTTP/2 要求は単一の TCP 接続を介して多重化されるため、ネットワークリソースの使用量を犠牲にすることなく、複数の同時メッセージを送信できます。また、ヘッダー圧縮を使用して、要求と応答のサイズを削減します。

gRPC は、クライアントがパラメーターをリモートで呼び出して戻り値の型を返すために、次の種類のサービスメソッドをサポートしています。

1. 単項 **RPC**。クライアントは単一のリクエストを gRPC サーバーに送信し、単一のレスポンスを返します。

例:

```
rpc SayHello(HelloRequest)returns (HelloResponse);
```

2. サーバーストリーミング **RPC**。クライアントは単一のリクエストを gRPC サーバーに送信し、ストリーム応答を取得します。

   例:

   ```
   rpc StreamingResponse(HelloRequest)returns (HelloResponse);
   ```

3. クライアントストリーミング **RPC**。クライアントは一連のメッセージを送信し、サーバーがその応答を読み取って返すのを待ちます。

   例:

   ```
   rpc IntroduceYourself(stream HelloRequest)returns (HelloResponse)
   ```

4. 双方向ストリーミング **RPC**。両側のクライアントとサーバーの両方が、読み取り/書き込みストリームを使用してメッセージのストリームを送信します。2 つのストリームは独立して動作します。

   例:

   ```
   rpc ChatSession (stream HelloRequest)returns (stream HelloResponse)
   ```

Citrix ADC は、gRPC エンドポイントを使用したサービスに対して次の機能をサポートしています。

- 負荷分散
- コンテンツの切り替え
- Web アプリケーションファイアウォール、認証などの安全なエンドポイントサービス。
- ポリシー設定
- 統計とログ
- コンテンツの書き換え、コンテンツフィルタリング
- Layer 4 and Layer 7 optimizations, TLS offering
- Gateway solutions for protocol translations

gRPC エンドツーエンド構成

October 7, 2021

gRPC エンドツーエンド構成は、クライアントから gRPC リクエストを送信することで機能します。HTTP/2 プロトコルと、gRPC サーバーによって応答された gRPC メッセージの転送。

エンドツーエンドの gRPC 構成の仕組み

次の図は、Citrix ADC アプライアンスで機能する gRPC 構成を示しています。

1. gRPC 構成をデプロイするには、最初に有効にする必要があります HTTP/2 HTTP プロファイルで、有効にします HTTP/2 サーバー側でグローバルにサポートします。
2. クライアントが gRPC リクエストを送信すると、負荷分散仮想サーバーはポリシーを使用して gRPC トラフィックを評価します。
3. ポリシー評価に基づいて、負荷分散仮想サーバー（gRPC サービスがバインドされている）は要求を終了し、gRPC 要求としてバックエンド gRPC サーバーに転送します。
4. 同様に、gRPC サーバーがクライアントに応答すると、アプライアンスは応答を終了し、gRPC 応答としてクライアントに転送します。

gRPC サーバーに送信される gRPC リクエストの例

リクエストヘッダーは次のように送信されます HTTP/2 のヘッダー HEADERS+CONTINUATION フレーム。

```
```
HEADERS (flags = END_HEADERS)
: method = POST
: scheme = http
: path = /helloworld.citrix-adc/SayHello
: authority = 10.10.10.10.:80
grpc-timeout = 15
content-type = application/grpc+proto
grpc-encoding = gzip
DATA (flags = END_STREAM)
<Length-Prefixed Message>
<!--NeedCopy--> ```
```

### gRPC サーバーから Citrix ADC アプライアンスへの gRPC 応答ヘッダーの例

応答-ヘッダー & トレーラー-単一でのみ配信されます HTTP/2 HEADERS フレームブロック。ほとんどの応答にはヘッダーとトレーラーの両方が含まれていると予想されますが、即時エラーが発生する呼び出しにはトレーラーのみ

 
```

が許可されます。HTTP ステータスコードが OK の場合でも、ステータスは Trailers で送信する必要があります。

```
```
HEADERS (flags = END_HEADERS)
: status = 200
Grpc-encoding= gzip
Content-type = application/grpc+proto
DATA
<Length-Prefixed Message>
HEADERS (flags = END_STREAM, END_HEADERS)
grpc-status = 0 # OK

<!--NeedCopy--> ```
```

### CLI を使用して gRPC を構成する

エンドツーエンドの gRPC デプロイメントを設定するには、以下を完了する必要があります。

- HTTP/2 および HTTP/2 ダイレクトを有効にした HTTP プロファイルを追加
- HTTP パラメーターでグローバルバックエンド HTTP/2 サポートを有効にする
- SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイルを設定します
- gRPC エンドポイントのサービスを追加して HTTP プロファイルを設定する
- gRPC エンドポイントサービスを負荷分散仮想サーバーにバインドする

#### HTTP/2 および HTTP/2 ダイレクトを有効にした HTTP プロファイルを追加

HTTP プロファイルで HTTP/2 および HTTP/2 ダイレクトパラメータを有効にする必要があります。また、gPRC over HTTP/2 クリアテキストが必要な場合、HTTP/2 ダイレクトパラメーターを有効にする必要があります。

コマンドプロンプトで入力します。

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

例:

```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```

HTTP パラメーターでグローバルバックエンド HTTP/2 サポートを有効にする

Citrix ADC コマンドラインを使用してサーバーでグローバルに HTTP/2 サポートを有効にするには。

コマンドプロンプトで入力します。


```

```
set ns httpParam -http2ServerSide(ON | OFF)
```

例:

```
set ns httpParam -http2ServerSide ON
```

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイルを設定します

Citrix ADC コマンドインターフェイスを使用して負荷分散仮想サーバーを追加するには、次の手順に従います。

コマンドプロンプトで入力します。

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

例:

```
add lb vserver lb-grpc HTTP 10.10.10.11 80 -httpProfileName http2gRPC
```

> 注:
>
> タイプ SSL の負荷分散仮想サーバーを使用している場合は、サーバー証明書をバインドする必要があります。詳細については、サーバー証明書のバインドのトピックを参照してください。

gRPC エンドポイントのサービスを追加して **HTTP** プロファイルを設定する

Citrix ADC コマンドインターフェイスを使用して HTTP プロファイルで gRPC サービスを追加するには:
コマンドプロンプトで、次のように入力します。

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

例:

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

gRPC エンドポイントサービスを負荷分散仮想サーバーにバインドする

Citrix ADC コマンドインターフェイスを使用して、gRPC サービスを負荷分散仮想サーバーにバインドするには:

コマンドインターフェイスで、次のように入力します。

```
bind lb vserver <name> <serviceName>
```

例:

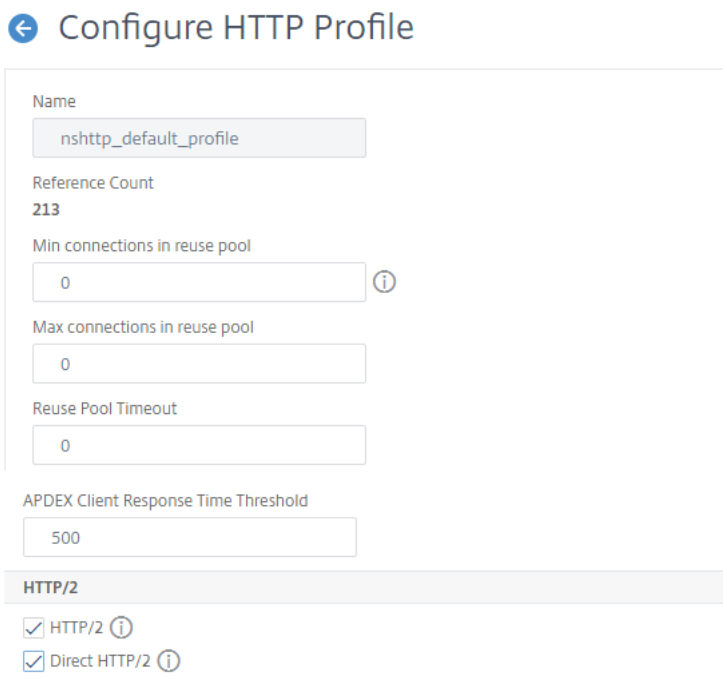
```
bind lb vserver lb-grpc svc-grpc
```

GUI を使用してエンドツーエンドの **gRPC** デプロイメントを構成します

GUI を使用して gRPC を設定するには、次の手順を実行します。

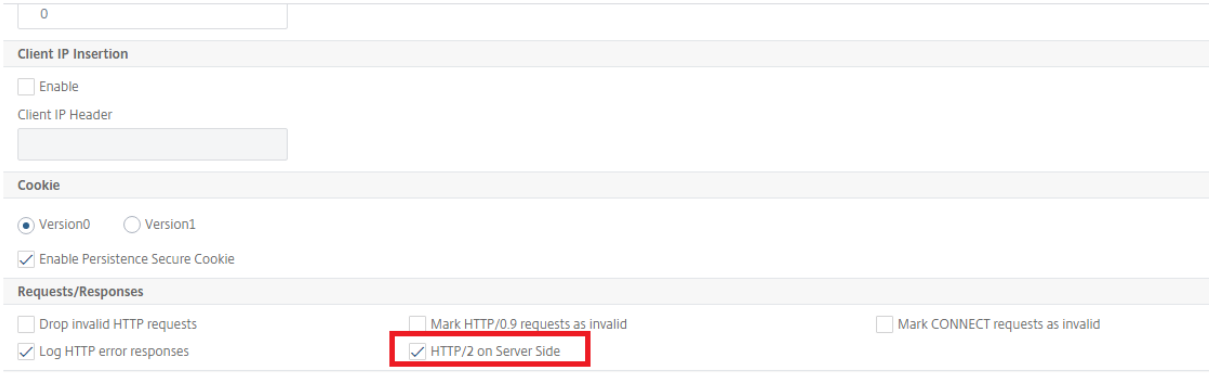
HTTP/2 および **HTTP/2** ダイレクトを有効にした **HTTP** プロファイルを追加

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. 新しい HTTP プロファイルまたは HTTP プロファイルで HTTP/2 オプションを有効にする



HTTP パラメーターでグローバルバックエンド **HTTP/2** サポートを有効にする

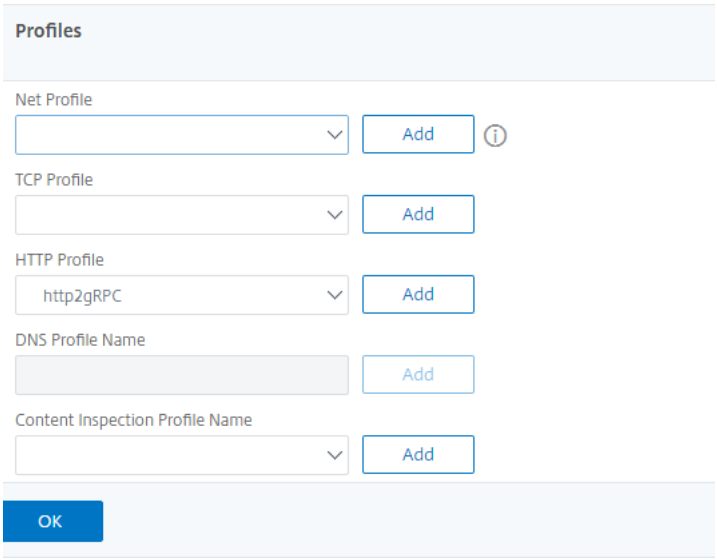
1. **System > Settings > HTTP Parameters** に移動します。
2. [HTTP パラメータの構成] ページで、サーバー側の HTTP/2 を選択します。
3. [**OK**] をクリックします。



SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイルを設定します

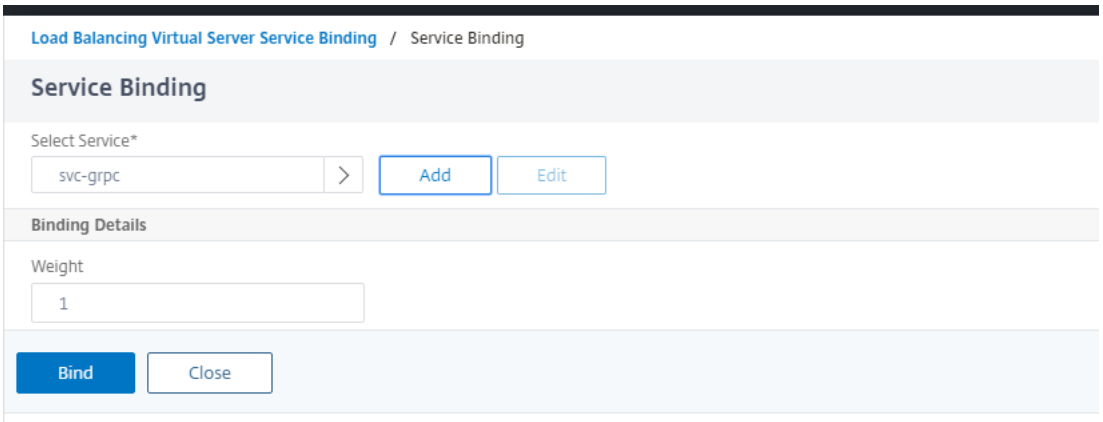
1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。

2. Add をクリックして gRPC トラフィックの負荷分散仮想サーバーを作成します。
3. Load Balancing Virtual Server ページで Profiles をクリックします。
4. [プロファイル] セクションで、プロファイルタイプを HTTP として選択します。
5. [OK] 、[完了] の順にクリックします。



gRPC エンドポイントのサービスを追加して **HTTP** プロファイルを設定する

1. **Traffic Management > Load Balancing > Services** に移動します。
2. Add をクリックして gRPC トラフィックのアプリケーションサーバーを作成します。
3. [負荷分散サービス] ページで、[プロファイル] セクションに移動します。
4. [プロファイル] で、gRPC エンドポイントの HTTP プロファイルを追加します。
5. [OK] 、[完了] の順にクリックします。



ロードバランシングに関連する GUI 手順の詳細については、「負荷分散 」のトピックを参照してください。

gRPC bridging

October 7, 2021

クライアントがリクエストを送信するとき HTTP/1.1 プロトコルでは、Citrix ADC アプライアンスは gRPC リクエストのブリッジングをサポートします HTTP/1.1 gRPC サーバーに準拠しているプロトコル HTTP/2 プロトコル。同様に、リバースブリッジングでは、アプライアンスはクライアントの gRPC 要求を HTTP/2 プロトコルを実行し、の gRPC サーバーに準拠して gRPC リクエストのリバースブリッジを実行します。HTTP/1.1 プロトコル。

How gRPC bridging works

このシナリオでは、Citrix ADC アプライアンスは、受信した gRPC コンテンツをシームレスにブリッジします。HTTP/1.1 接続し、それをバックエンド gRPC サーバーに転送します HTTP/2.

次の図は、gRPC ブリッジ構成でコンポーネントがどのように相互作用するかを示しています。

1. gRPC リクエストが送信されると、Citrix ADC アプライアンスは接続が HTTP/1.1 コンテンツタイプは application/grpc. ザ・HTTP/1.1 リクエストは、次の疑似ヘッダーに変換されます。
2. で gRPC リクエストを受信すると HTTP/1.1. Content-Type ヘッダーで示される接続では、ADC アプライアンスはリクエストを gRPC に変換します。HTTP/2 以下のように:

```
    :method: Method-name in HTTP/1.1 request
    :path: Path is HTTP/1.1 request
    content-type: application/grpc
<!--NeedCopy-->
```

1. ポリシー評価に基づいて、負荷分散仮想サーバー（gRPC サービスがバインドされている）は要求を終了するか、転送します HTTP/2 バックエンド gRPC サーバーへのフレーム。
2. で応答を受信すると HTTP/2 gRPC サーバーからの接続、アプライアンスは受信するまでバッファリングします HTTP/2 トレーラーを開き、gRPC ステータスコードを確認します。ゼロ以外の gRPC エラーステータスの場合、アプライアンスはマッピング HTTP ステータスコードを探し、適切なものを送信します HTTP/1.1 エラー応答。

CLI を使用して gRPC ブリッジを構成する

To configure gRPC reverse bridging, you must complete the following steps:

1. HTTP/2 および HTTP/2 ダイレクトを有効にした HTTP プロファイルを追加
2. グローバルバックエンドを有効にする HTTP/2 HTTP パラメータでのサポート
3. SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイルを設定します
4. gRPC エンドポイントのサービスを追加し、HTTP プロファイルを設定します
5. gRPC エンドポイントサービスを負荷分散仮想サーバーにバインドする
6. gRPC ステータスコードをゼロ以外の gRPC ステータスの HTTP 応答にマップします
7. 時間および/またはサイズによる gRPC バッファリングの構成

HTTP/2 および HTTP/2 ダイレクトを有効にした HTTP プロファイルを追加

設定を開始するには、HTTP プロファイルで HTTP/2 機能を有効にする必要があります。クライアントが HTTP1.1 リクエストを送信すると、アプライアンスはリクエストをブリッジしてバックエンドサーバーに転送します。

コマンドプロンプトで入力します。

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

例:

```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```

グローバルバックエンドを有効にする HTTP/2 HTTP パラメータでのサポート

To enable HTTP/2 support globally on the server side by using the Citrix ADC command line.

コマンドプロンプトで入力します。

```
set ns httpParam -http2ServerSide( ON | OFF )
```

例:

```
set ns httpParam -http2ServerSide ON
```

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイルを設定します

Citrix ADC コマンドインターフェイスを使用して負荷分散仮想サーバーを追加するには

コマンドプロンプトで入力します。

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

例:


```
add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName http2gRPC
```

> 注:
>
> タイプ SSL の負荷分散仮想サーバーを使用している場合は、サーバー証明書をバインドする必要があります。詳細については、「サーバー証明書のバインド」のトピックを参照してください。

gRPC エンドポイントのサービスを追加し、**HTTP** プロファイルを設定します

Citrix ADC コマンドインターフェイスを使用して、HTTP プロファイルを使用して gRPC サービスを追加します。

コマンドプロンプトで入力します。

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

例:

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

gRPC エンドポイントサービスを負荷分散仮想サーバーにバインドする

CLI を使用して、gRPC エンドポイントサービスを負荷分散仮想サーバーにバインドします。

コマンドインターフェイスで、次のように入力します。

```
bind lb vserver <name> <serviceName>
```

例:

```
bind lb vserver lb-grpc svc-grpc
```

gRPC ステータスコードを **HTTP/1.1** 応答の **HTTP** ステータスコードにマップします

gRPC ブリッジングシナリオでは、gRPC サービスは gRPC ステータスコードでリクエストに応答します。アプライアンスは、gRPC ステータスコードを対応する HTTP 応答コードと理由フレーズにマップします。マッピングは、以下の表に基づいて行われます。送信時の Citrix ADC アプライアンス HTTP/1.1 クライアントへの応答は、HTTP ステータスコードと理由フレーズを送信します。

gRPC status-code	HTTP response status-code	HTTP response reason-phrase
OK = 0	200	OK
CANCELLED = 1	499	*
UNKNOWN = 2	500	内部サーバーエラー
INVALID_ARGUMENT = 3	400	不正な要求

gRPC status-code	HTTP response status-code	HTTP response reason-phrase
DEADLINE_EXCEEDED = 4	504	Gateway Timeout
NOT_FOUND = 5	404	*
ALREADY_EXISTS = 6	409	競合
PERMISSION_DENIED = 7	403	禁止
UNAUTHENTICATED = 16	401	未承認
RESOURCE_EXHAUSTED = 8	429	*
FAILED_PRECONDITION = 9	400	不正な要求
ABORTED = 10	409	競合
OUT_OF_RANGE = 11	400	不正な要求
UNIMPLEMENTED = 12	501	Not Implemented
INTERNAL = 13	500	内部サーバーエラー
UNAVAILABLE = 14	503	Service Unavailable
DATA_LOSS = 15	500	内部サーバーエラー

時間および/またはサイズによる **gRPC** バッファリングの構成

Citrix ADC アプライアンスは、応答トレーラーが受信されるまで、バックエンドサーバーからの gRPC 応答をバッファリングします。これにより、双方向の gRPC 呼び出しが中断されます。また、gRPC 応答が大きい場合、応答を完全にバッファリングするために大量のメモリを消費します。この問題を解決するために、gRPC ブリッジ構成が拡張され、時間によるバッファリングが制限されます。and/or サイズ。バッファサイズまたは時間制限がしきい値を超えると、制限のいずれかがトリガーされた場合でも（設定されたバッファサイズ内でトレーラーが受信されない場合、または設定されたタイムアウトが発生した場合）、アプライアンスはバッファリングを停止し、応答をクライアントに転送します。その結果、構成されたポリシーとその式（grpc-status コードに基づく）が期待どおりに機能しません。

時間によって gRPC バッファリングを制限するには and/or CLI によるサイズで、新しい HTTP プロファイルを追加するときに構成したり、既存のプロファイルを変更するときに構成したりできます。

コマンドプロンプトで入力します。

```
add ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout
 <positive_integer>]
```

または

```
set ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout
 <positive_integer>]
```


各項目の意味は次のとおりです。

grpcholdlimit。トレーラーが受信されるまで gRPC パケットをバッファリングできるバイト単位の最大サイズ。パラメータといずれか 1 つを設定できます。

デフォルト値：131072
最小値：0
最大値：33554432

grpcholdtimeout。トレーラーが受信されるまで gRPC パケットをバッファリングできる最大時間（ミリ秒単位）。値は 100 の倍数である必要があります。
デフォルト値：1000
最小値：0
最大値：180000

例**:**

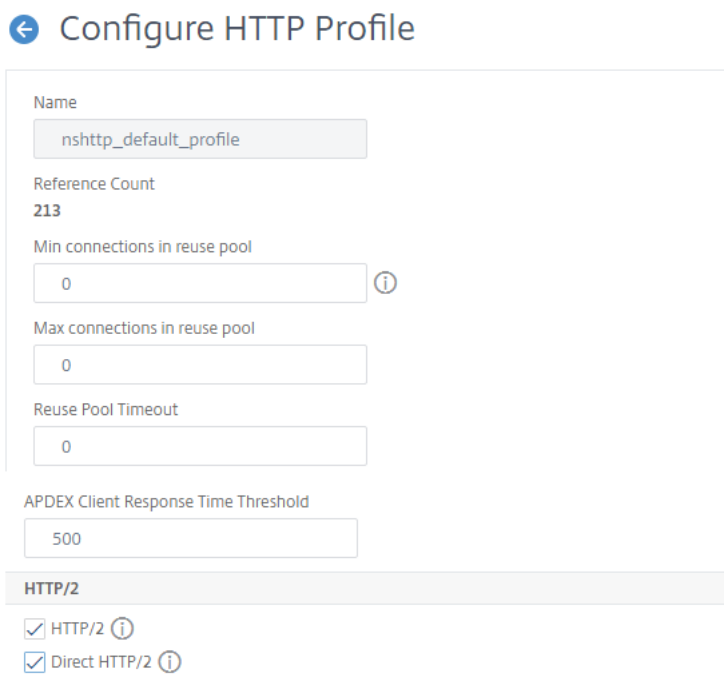
```
add httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
set httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
```

GUI を使用して **gRPC** ブリッジを構成する

Citrix ADC GUI を使用して gRPC ブリッジングを構成するには、次の手順を実行します。

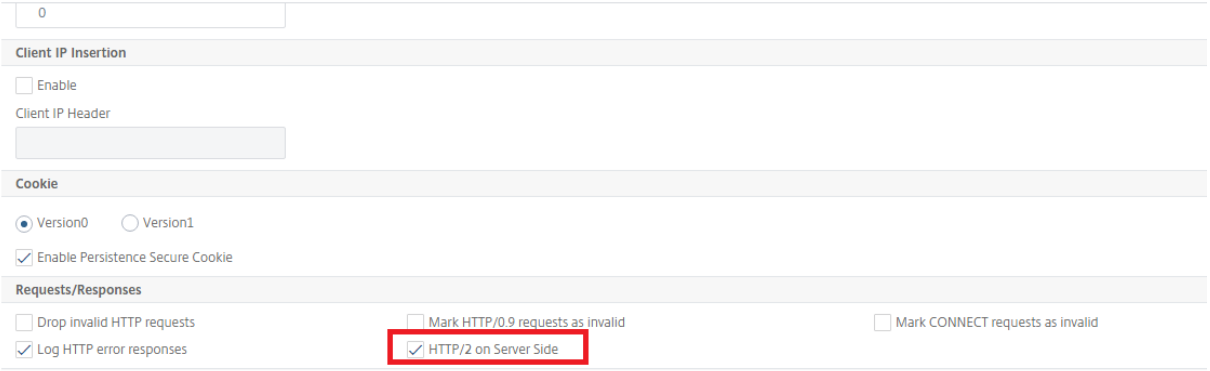
HTTP/2 および **HTTP/2** ダイレクトを有効にした **HTTP** プロファイルを追加

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. HTTP プロファイルで **HTTP/2** を選択します。



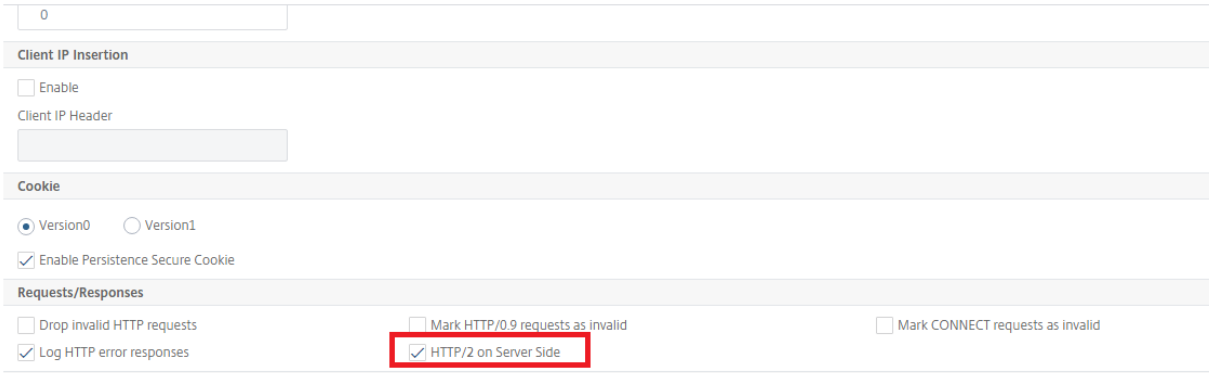
グローバルバックエンドを有効にする **HTTP/2 HTTP** パラメータでのサポート

1. **System > Settings > HTTP Parameters** に移動します。
2. [**HTTP** パラメータの構成] ページで、[HTTP/2 サーバー側オプション。
3. [**OK**] をクリックします。



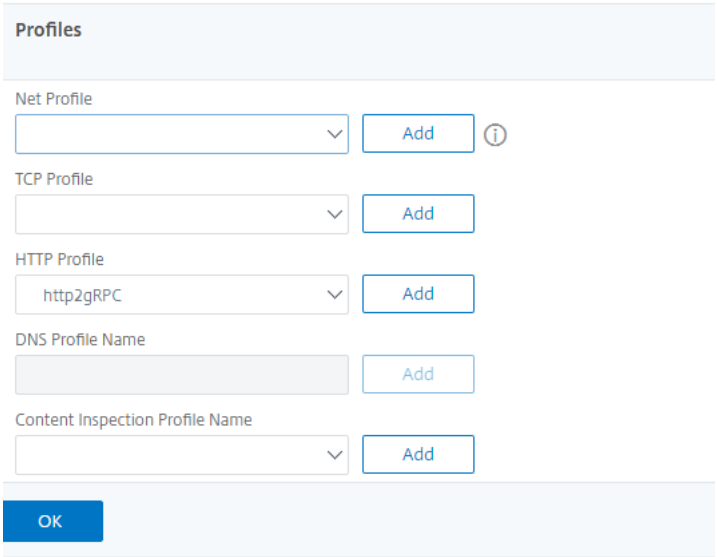
SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイルを設定します

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. **Add** をクリックして gRPC トラフィックの負荷分散仮想サーバーを作成します。
3. **Load Balancing Virtual Server** ページで **Profiles** をクリックします。
4. [プロファイル] セクションで、プロファイルタイプを HTTP として選択します。
5. [**OK**]、[完了] の順にクリックします。



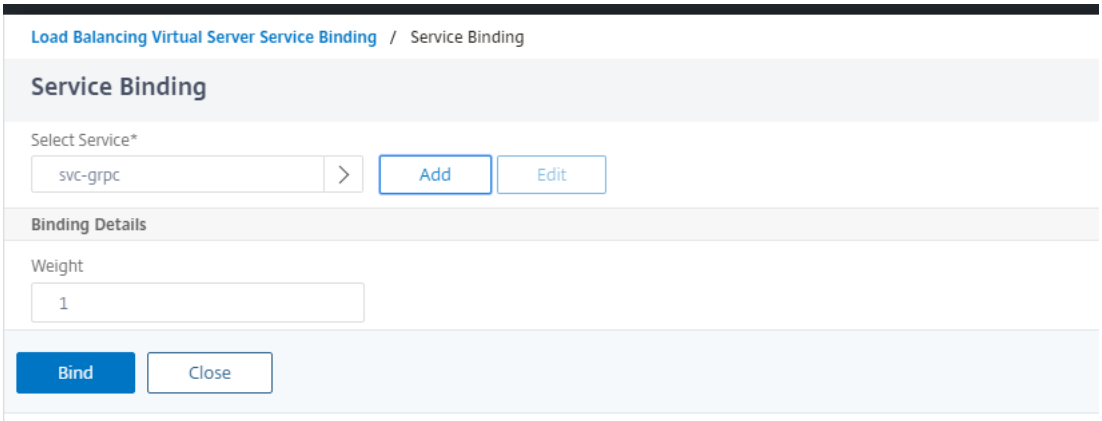
gRPC エンドポイントのサービスを追加して **HTTP** プロファイルを設定する

1. **Traffic Management > Load Balancing > Services** に移動します。
2. **Add** をクリックして gRPC トラフィックのアプリケーションサーバーを作成します。
3. [負荷分散サービス] ページで、[プロファイル] セクションに移動します。
4. [プロファイル] で、gRPC エンドポイントの **HTTP** プロファイルを追加します。
5. [**OK**]、[完了] の順にクリックします。



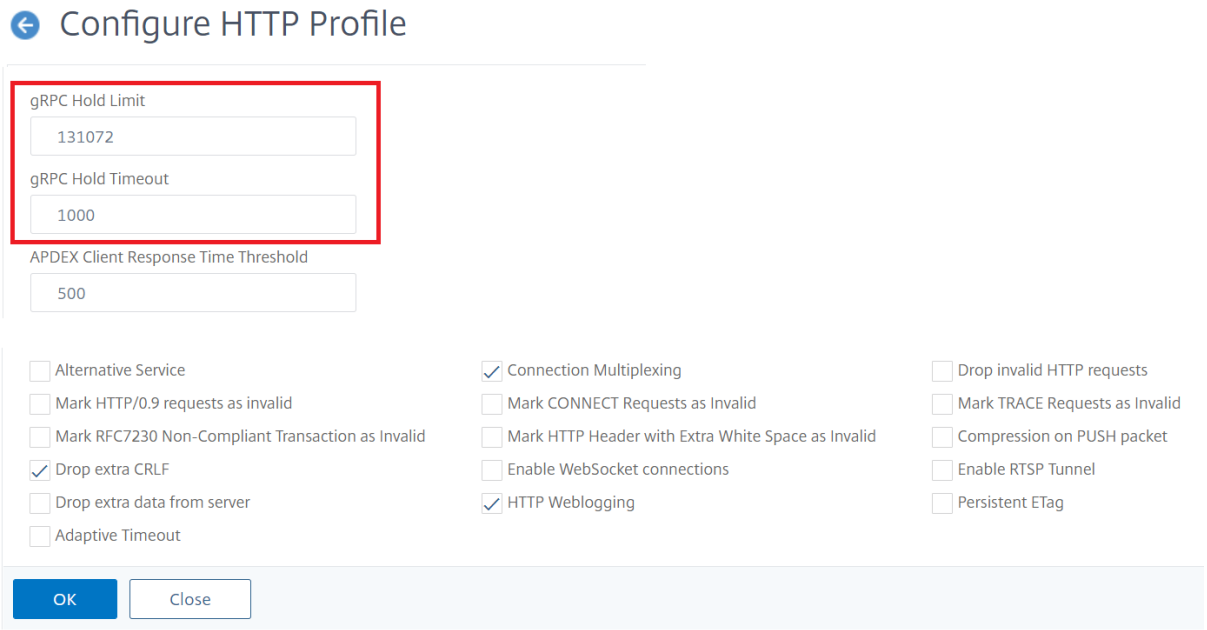
gRPC エンドポイントのサービスを負荷分散仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. **Add** をクリックして gRPC トラフィックの負荷分散仮想サーバーを作成します。
3. [負荷分散仮想サーバー] ページで、[サービスとサービスグループ] セクションをクリックします。
4. **[** 負荷分散仮想サーバーサービスのバインド**]** ページで、バインドする gRPC サービスを選択します。
5. **[**閉じる**]**、**[**完了**]** の順にクリックします。



GUI を使用して、時間とサイズで **gRPC** バッファリングを構成します

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. HTTP プロファイルで **HTTP/2** を選択します。
3. 「**HTTP** プロファイルの構成」ページで、以下のパラメーターを設定します。
   a) grpcHoldTimeout. トレーラーが受信されるまで gRPC パケットをバッファリングする時間をミリ秒単位で入力します。
   b) grpcHoldLimit. トレーラーが受信されるまで gRPC パケットをバッファリングするための最大サイズをバイト単位で入力します。
4. [**OK]** をクリックして [閉じる] をクリックします。



サービスと負荷分散仮想サーバーをバインドするための GUI 手順の詳細については、「負荷分散 」のトピックを参照してください。

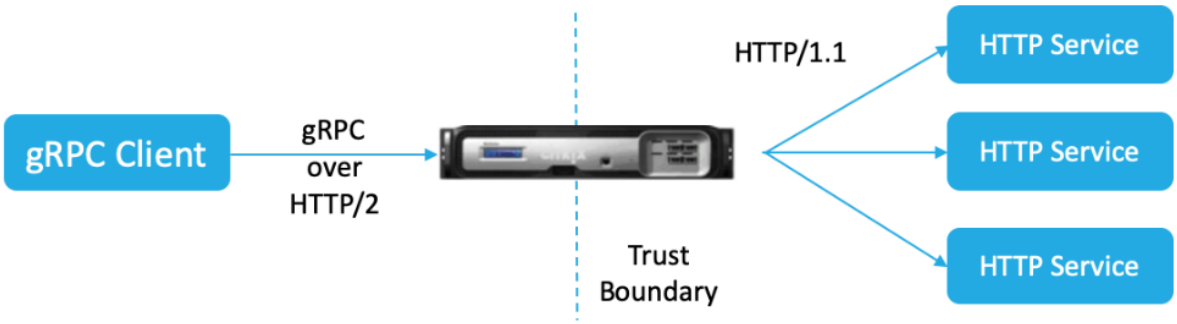
gRPC リバースブリッジング

October 7, 2021

このシナリオでは、Citrix ADC アプライアンスは、受信した gRPC コンテンツをシームレスにブリッジします。HTTP/2 接続し、それをバックエンド gRPC サーバーに転送します HTTP/1.1.

リバースブリッジの仕組み

次の図は、gRPC ブリッジ構成でコンポーネントがどのように相互作用するかを示しています。



1. クライアントは gRPC 要求を HTTP/2 接続で HTTP/2 フレームおよび proto-buf ペイロードで gRPC ヘッダーとともに送信します。
2. ポリシー評価に基づいて、負荷分散仮想サーバー（gRPC サービスがバインドされている）が要求を変換してバックエンドサーバーへの HTTP/1.1 接続で転送します。
3. 受信時に HTTP/1.1 応答、応答に grpc-status コードがない場合、ADC は HTTP 応答コードから grpcstatus-case を導出します。
4. 次に、応答をクライアントに転送する前にアプライアンスは gRPC ヘッダーを HTTP/2 トレーラー挿入します。

CLI を使用して gRPC リバースブリッジを構成する

gRPC リバースブリッジを設定するには、次の手順を実行する必要があります。

- HTTP/2 および HTTP/2 ダイレクトを有効にした HTTP プロファイル 1 を追加し、仮想サーバーの負荷分散を行います
- バックエンドサーバーで HTTP/2 を無効にした HTTP プロファイル 2 を追加する
- SSL/HTTP タイプの負荷分散仮想サーバーを追加し、HTTP プロファイル 1 に設定します
- gRPC エンドポイントでサービスを追加し HTTP プロファイル 2 を設定する
- gRPC エンドポイントのサービスを負荷分散仮想サーバーにバインドする
- 応答に grpc ステータスコードがない場合は、HTTP ステータスコードを gRPC ステータスコードにマップします

HTTP/2 および **HTTP/2** ダイレクトを有効にした **HTTP** プロファイル **1** を追加し、仮想サーバーの負荷分散を行います

リバースブリッジ設定を開始するには、2 つの HTTP プロファイルを追加する必要があります。gRPC クライアント要求の HTTP/2 を有効にする 1 つのプロファイルと非 gRPC サーバー応答の HTTP/2 を無効にする別のプロファイル。

コマンドプロンプトで入力します。

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

例**:**

add ns httpProfile profile1 –http2 ENABLED -http2Direct ENABLED

gRPC エンドポイントでサービスを追加し **HTTP** プロファイル **2** を設定する

Citrix ADC コマンドラインを使用してバックエンドサーバー応答のために HTTP プロファイルで HTTP/2 サポートを無効にするには。

コマンドプロンプトで入力します。

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

例**:**

add ns httpProfile profile2 –http2 DISABLED http2Direct DISABLED

SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイル **1** に設定します

Citrix ADC コマンドインターフェイスを使用して負荷分散仮想サーバーを追加するには、次の手順に従います。

コマンドプロンプトで入力します。

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

例**:**

add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName profile1

> 注:
>
> タイプ SSL の負荷分散仮想サーバーを使用している場合は、サーバー証明書をバインドする必要があります。詳細については、サーバー証明書のバインドのトピックを参照してください。

gRPC エンドポイントでサービスを追加し **HTTP** プロファイル **2** を設定する

gRPC エンドポイントでサービスを追加し、Citrix ADC コマンドインターフェイスを使用して HTTP プロファイル 2 を設定します。

コマンドプロンプトで入力します。

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

例:

```
add service svc-grpc 10.10.10.11 HTTP 80 -httpProfileName profile2
```

gRPC エンドポイントのサービスを負荷分散仮想サーバーにバインドする

Citrix ADC コマンドインターフェイスを使用して、gRPC サービスを負荷分散仮想サーバーにバインドします。

コマンドインターフェイスで、次のように入力します。

```
bind lb vserver <name> <serviceName>
```

例:

```
bind lb vserver lb-grpc svc-grpc
```

HTTP 応答コードを **gRPC** ステータスコードにマップする

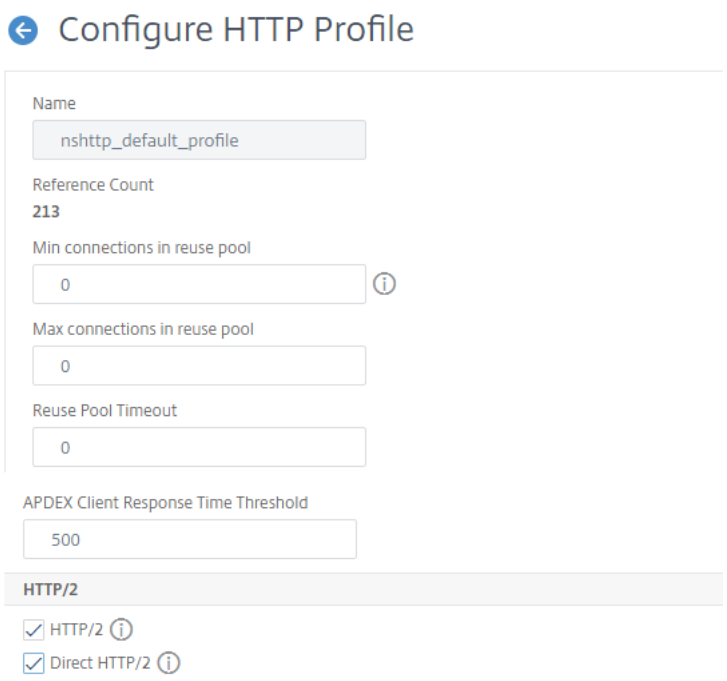
サーバーが gRPC ステータスコードを生成しない場合、Citrix ADC アプライアンスが受信した HTTP 応答を基に適切な gRPC ステータスコードを生成します。ステータスコードは以下のマッピングテーブルに表示されています。

HTTP Response status-code	gRPC status code
200	OK
400	INTERNAL = 13
403	PERMISSION_DENIED = 7
401	UNAUTHENTICATED = 16
429, 502, 503, 504	UNAVAILABLE = 14
404	UNIMPLEMENTED = 12

Configure gRPC reverse bridging by using the GUI

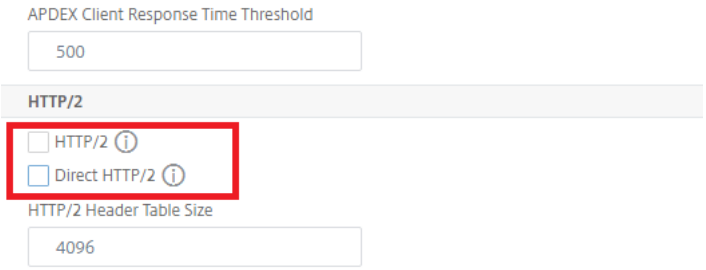
HTTP/2 および **HTTP/2** ダイレクトを有効にした **HTTP** プロファイル **1** を追加し、仮想サーバーの負荷分散を行います

1. System > Profiles に移動して HTTP Profiles をクリックします。
2. HTTP プロファイル 1 で HTTP/2 オプションを有効にします。



gRPC エンドポイントでサービスを追加し **HTTP** プロファイル **2** を設定する

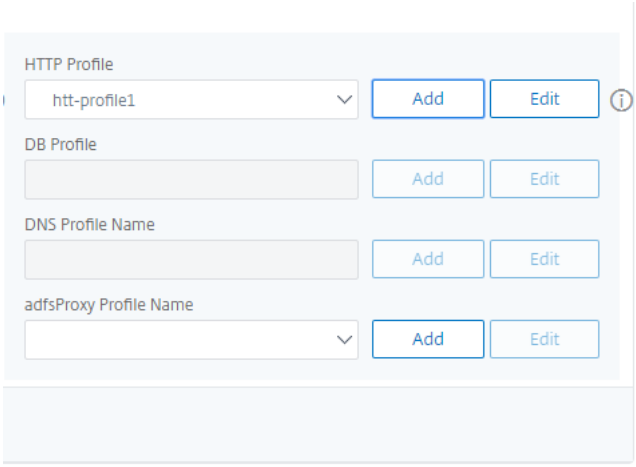
1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. HTTP プロファイル 2 の **HTTP/2** オプションを有効にします。
3. [**OK**] をクリックします。



SSL/HTTP タイプの負荷分散仮想サーバーを追加し、**HTTP** プロファイル **1** に設定します

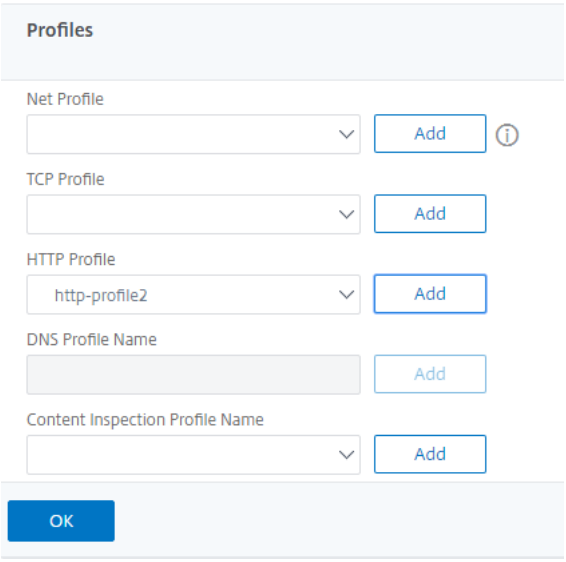
1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。

2. **Add** をクリックして gRPC トラフィックの負荷分散仮想サーバーを作成します。
3. **Load Balancing Virtual Server** ページで **Profiles** をクリックします。
4. [プロファイル] セクションで、プロファイルタイプを HTTP として選択します。
5. [**OK**]、[完了] の順にクリックします。



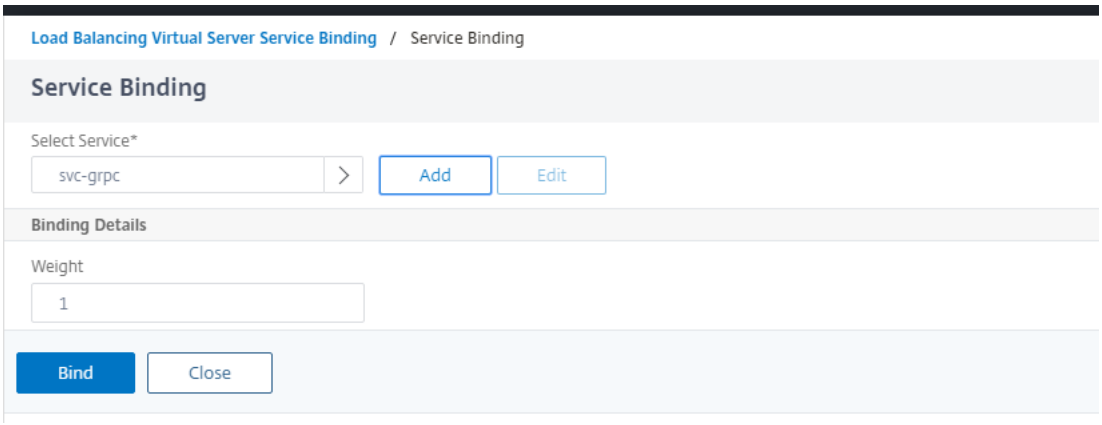
gRPC エンドポイントでサービスを追加し、**HTTP** プロファイル **2** に設定します

1. **Traffic Management > Load Balancing > Services** に移動します。
2. **Add** をクリックして gRPC トラフィックのアプリケーションサーバーを作成します。
3. [負荷分散サービス] ページで、[プロファイル] セクションに移動します。
4. [プロファイル] で、gRPC エンドポイントの **HTTP** プロファイルを追加します。
5. [**OK**]、[完了] の順にクリックします。



gRPC エンドポイントのサービスを負荷分散仮想サーバーにバインドする

1. **Traffic Management > Load Balancing > Virtual Servers** に移動します。
2. **Add** をクリックして gRPC トラフィックの負荷分散仮想サーバーを作成します。
3. [仮想サーバーの負荷分散] ページで、[サービスと サービスグループ] セクションをクリックします。
4. **[** 負荷分散仮想サーバーサービスのバインド] ページで、バインドする gRPC サービスを選択します。
5. **[**閉じる]、**[**完了] の順にクリックします。



GUI の手順の詳細については、負荷分散のトピックを参照してください 。

gRPC call termination

October 7, 2021

Citrix ADC アプライアンスにレート制限、Web App Firewall セキュリティなどのポリシーが設定されている場合、ポリシーが true と評価された場合、アプライアンスは通話を終了し、計算可能な gRPC エラーメッセージでクライアントに応答できます。

書き換えポリシーを使用した **gRPC**

October 7, 2021

gRPC with rewrite policy のユースケースでは、Citrix ADC アプライアンスが gRPC 要求または応答の一部の情報を書き換える際にどのように機能するかを説明しています。次の図は、コンポーネントが相互作用することを示しています。

次の図は、リライトポリシー設定を使用して gRPC でコンポーネントがどのように相互作用するかを示しています。

1. アプライアンスで書き換え機能を有効にします。
2. gRPC ヘッダーを変更、追加、または削除するための書き換えアクションを構成します。
3. アクションを実行する必要がある gRPC リクエスト（トラフィック）を決定するための書き換えポリシーを構成します。
4. 書き換えポリシーを負荷分散仮想サーバーにバインドして、トラフィックがポリシー式と一致するかどうかを調べます。
5. リライトポリシーを使用することで、gRPC ステータスコードに基づいて以下を実行できます。
   a) gRPCWeb サーバーからの応答を変更します。
   b) gRPC ヘッダーを変更、追加、または削除します。
   c) gRRC サーバーへのリクエストの URL を変更します。

書き換えポリシーを使用して **gRPC** 呼び出しの終了を構成する

リライトポリシーを使用して gRPC コールの終了を設定するには、次の手順を実行する必要があります。

1. 書き換え機能の有効化
2. 書き換えポリシーの追加
3. 書き換えポリシーを負荷分散仮想サーバーにバインドする

書き換え機能の有効化

書き換え機能を使用するには、最初にそれを有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns rewrite
```

書き換えポリシーの追加

リライトアクションを構成した後、次にリライトポリシーを構成して、Citrix ADC アプライアンスがリライトする必要のある gRPC リクエストを選択する必要があります。

コマンドプロンプトで入力します。


```
add rewrite policy <name> <expression> <action> [<undefaction>]-appFlowaction
 <actionName>
```

例:

```
add rewrite policy grpc-rewr_pol1 ”http.res.header(\”grpc-status\”).NE
(\”0\”)”RESET
```

書き換えポリシーを負荷分散仮想サーバーにバインドする

ポリシーを有効にするには、gRPC サービスを使用してポリシーを負荷分散仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します。

```
bind rewrite global <policyName> <priority> [<gotoPriorityExpression> [-
type <type>] [-invoke (<labelType> <labelName>)]
```

例:

```
bind lb vserver lb-grpc -policyName grpc-rewr_pol1 -priority 100
```

レスポンダーポリシーを持つ gRPC

October 7, 2021

レスポンダー付きの gRPC ポリシー構成では、Citrix ADC アプライアンスが HTTP/2 プロトコルを介して gRPC 要求に対して異なる応答を提供する方法について説明します。ユーザーが Web サイトのホームページを要求するときは、各ユーザーの場所やユーザーが使用しているブラウザーに応じて、異なるホームページを提供することをお勧めします。

次の図は、相互作用するコンポーネントを示しています。

1. アプライアンスでレスポンダ機能を有効にします。

2. カスタム応答を生成する、要求を別の Web ページにリダイレクトする、または接続をリセットするようにレスポンダーアクションを構成します。
3. アクションを実行する必要がある gRPC 要求（トラフィック）を決定するためのレスポンダーポリシーを構成します。
4. レスポンダーポリシーを負荷分散仮想サーバーにバインドして、トラフィックがポリシー式に一致するかどうかを調べます。
5. レスポンダーポリシーを使用すると、gRPC ステータスコードに基づいて以下を実行できます。

CLI を使用して gRPC コール終了をレスポンダポリシーで設定します

レスポンダポリシーを使用して gRPC コールの終了を設定するには、次の手順を実行する必要があります。

1. レスポンダー機能を有効にする
2. レスポンダーアクションを追加する
3. レスポンダーポリシーを追加し、レスポンダーアクションを関連付ける
4. レスポンダーポリシーを負荷分散仮想サーバーにバインドする

レスポンダー機能を有効にする

レスポンダー機能を使用するには、最初にそれを有効にする必要があります。

コマンドプロンプトで入力します。

```
enable ns responder
```

レスポンダーアクションを追加する

この機能を有効にした後、バックエンドサーバーから返されたステータスコードに基づいて gRPC 応答を処理するためのレスポンダーアクションを構成する必要があります。

コマンドプロンプトで入力します。

```
add responder action <name> <type>
```

例:

```
add responder action grpc-act respondwith "HTTP/1.1 200 OK\r\nServer: NS
-Responder\r\nContent-Type:application/grpc\r\ngrpc-status: 12\r\ngrpc
-message: Not Implemented\r\n\r\n"+ "Method: "+ HTTP.REQ.URL+ "is not
implemented."
```

レスポンダーポリシーの追加

レスポンダーのアクションを構成したら、次にレスポンダーポリシーを構成して、Citrix ADC アプライアンスが応答する gRPC リクエストを選択する必要があります。

コマンドプロンプトで入力します。

```
add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction
 <actionName>
```

例:

```
add responder policy grpc-resp-pol1 HTTP.REQ.URL.NE( “/helloworld.Greeter/
SayHello” )grpc-act
```

レスポンダーポリシーを負荷分散仮想サーバーにバインドする

ポリシーを有効にするには、gRPC サービスを使用してポリシーを負荷分散仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します。

```
bind responder global <policyName> <priority> [<gotoPriorityExpression> [-
type <type>] [-invoke (<labelType> <labelName>)]
```

例:

```
bind lb vserver lb-grpc svc-grpc -policyName grpc-resp-pol1 -priority 100
```

レスポンダーポリシーの詳細については、「レスポンダーポリシー」トピックを参照してください。

gRPC プロトコルバッファフィールドを照合するためのポリシー式

Citrix ADC アプライアンスは、gRPC 構成で次のポリシー式をサポートしています。

- **gRPC** プロトコルバッファフィールドアクセス。任意の gRPC API 呼び出しは、メッセージフィールド番号と新しいポリシー式と一致します。PI 構成では、一致は「フィールド番号」と「API パス」のみを使用して行われます。
- **gRPC** ヘッダーフィルタリング。gRPC の「HttpProfile」パラメータは、gRPC 解析のデフォルトの動作（gRPC ポリシー式を含む）を調整するために使用されます。gRPC ポリシー式には、次のパラメータが適用されます。
  - **gRPCLengthDelimitation**。これはデフォルトで有効になっており、プロトコルバッファに長さ区切りのメッセージが表示されることを想定しています。
  - **grpCholdLimit**。デフォルト値は 131072 です。これは、プロトコルバッファメッセージの最大サイズ（バイト単位）です。また、文字列の最大長と最大 ‘byte’ フィールド長でもあります。

CLI を使用して **gRPC** アドバンスポリシー式を設定します

コマンドプロンプトで入力します。


```
set ns httpProfile <name> -http2 ( ENABLED | DISABLED ) -
    gRPCLengthDelimitation ( ENABLED | DISABLED ) -gRPCHoldLimit <int>
```

例:

```
set ns httpProfile http2gRPC -http2 ENABLED -gRPCLengthDelimitation
    ENABLED -gRPCHoldLimit 131072
```

GUI を使用して **gRPC** ヘッダーフィルタリングパラメータを設定する

1. **System > Profiles** に移動して **HTTP Profiles** をクリックします。
2. [**HTTP** プロファイルの作成] ページで、[**HTTP/3**] セクションまでスクロールダウンし、[**gRPC** の長さの区切り] を選択します。

次のポリシー式の例は、メッセージ 5、サブメッセージ 4、およびフィールド 3 の値を示しています。これは 2 に等しい 32 ビットの int です。

```
http.req.body(1000).grpc.message(5).message(4).int32(3).eq(2)
```

gRPC プロトコルバッファメッセージフィールドを番号で照合するために、次のポリシー式が追加されます。

- message
- ダブル
- フロート
- int32
- int64
- uint32
- uint64
- sint64
- sint32
- fixed32
- fixed64
- sfixed32
- sfixed64
- bool
- string
- 列挙型
- bytes

API パスマッチング

API パスマッチングは、複数の API が使用されている場合に、正しい gRPC API 呼び出しを照合するために使用されます。API パスと一致します。これは HTTP リクエストの ‘: path’ 疑似ヘッダーにあります。

例**:**

```
1 http.req.header(":path").eq("acme.inventory.v1/ListBooks")
```

QUIC

October 7, 2021

Quick UDP Internet Protocol (QUIC) は、UDP に実装されている (TCP+TLS+HTTP/2) プロトコルの組み合わせである。QUIC トランスポートプロトコルは、UDP を使用して 2 つのエンドポイント間の接続を多重化します。また、他のプロトコルと比較すると、QUIC はセキュリティ、トラフィックの迅速な配信、および低レイテンシという点で高性能を提供します。

QUIC ブリッジは、QUIC クライアントと QUIC バックエンドサーバー間の QUIC トラフィックの負荷分散のために、Citrix ADC アプライアンスで構成されます。QUIC ブリッジを使用すると、NAT リバインドまたは接続の移行がある場合に、クライアントとサーバ間で永続的な QUIC 接続を確立できます。ただし、この設定はデータを処理しません。これは、Citrix ADC アプライアンスを介した QUIC トラフィックの負荷分散にのみ使用されます。

QUIC パケットには接続 ID が含まれており、エンドポイントがパケットを異なるアドレスまたは 4 タプルで同じ接続に関連付けることができます。接続 ID には、Citrix ADC アプライアンスおよびバックエンドサーバーと共有されるサーバー ID の詳細が含まれます。Citrix ADC アプライアンスは、サーバー ID の接続 ID の詳細を抽出し、トラフィックをバックエンドサーバーに送信します。接続 ID は保護されたパケットの中にあり、接続の移行時に接続を堅牢にします。

> 重要
>
> バックエンドサーバーは、QUIC 接続 ID でサーバー ID をエンコードするためのサポートが必要です。

QUIC ブリッジのメリット

Citrix ADC アプライアンスの QUIC ブリッジは、次の理由で優先されます。

- 高価な暗号操作はありません。
- ステートレスルーティングが可能です（4 タプルベースのロードバランシングなし）。

QUIC ブリッジ設定

October 7, 2021

QUIC ブリッジを設定するには、次の作業を完了する必要があります。

- QUIC ブリッジプロファイルの追加
- QUIC バックエンドサーバーの追加
- アプライアンスに QUIC サービスを追加する
- QUIC ブリッジタイプの負荷分散仮想サーバーの追加
- QUIC ブリッジタイプの QUIC ブリッジのロードバランシング仮想サーバーにバインドする

> 重要
>
> QUIC ブリッジを設定する前に、まずアプライアンスでロードバランシング機能を有効にしてください。詳細については、「基本的なロードバランシングの設定」を参照してください。

CLI を使用した QUIC ブリッジの設定

次のセクションは、CLI を使用して設定する必要があります。

QUIC ブリッジプロファイルを追加する

QUIC ブリッジプロファイルを追加する必要があります。

コマンドプロンプトで入力します。

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```

例:

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```

> 注
>
> この例で設定されるserveridlenパラメータは、IP および PORT の 16 進数の文字列であるカスタムサーバー ID の長さです。

QUIC バックエンドアプリケーションサーバーの追加

QUIC バックエンドアプリケーションサーバーを追加する必要があります。

コマンドプロンプトで入力します。

```
1 -  add server <name> (<IPAddress>)
2 -  add server <name> (<IPAddress>)
```

例:

```
1 -  add server s1 192.0.2.20
2 -  add server s2 192.0.2.30
```

QUIC ブリッジサービスを追加する

QUIC ブリッジサービスをアプリケーションサーバに追加する必要があります。

コマンドプロンプトで入力します。

```
1 -  add service  <name>  (<IP> | <serverName>) <serviceType> <port> [-
     CustomServerID <string>]
2
3 -  add service  <name>  (<IP> | <serverName>) <serviceType> <port> [-
     CustomServerID <string>]
```

例:

```
1 -  add service src1 s1 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A8026401BB
2
3 -  add service src2 s2 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A802C801BB
```

> 注
>
> 前の例で設定したCustomServerIDパラメータは、対応する IP の 16 進文字列とサーバーの PORT（s1 および s2）です。QUIC ブリッジ機能では、16 進文字列形式でのみCustomServerIDパラメータを構成することをお勧めします。

QUIC ブリッジタイプの負荷分散仮想サーバーを追加する

QUIC ブリッジタイプの負荷分散仮想サーバーを追加する必要があります。

コマンドプロンプトで入力します。

```
add lb vserver <name> [<IPAddress>@ <port> [-persistenceType <
    persistenceType >] [-lbMethod < lbMethod > [-cltTimeout <secs>] [-
    quickBridgeProfilename <name>]
```

例:

```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -cltTimeout 120 -
    quicBridgeProfilename q1
```

> 注
>
> QUIC ブリッジ仮想サーバの設定時に、persistenceTypeパラメータをCUSTOMSERVERIDに設定し、「lbMethod」パラメータをTOKENとして設定する必要があります。

QUIC ブリッジサービスをタイプ **QUIC** ブリッジのロードバランシング仮想サーバーにバインドします

QUIC ブリッジサービスは、QUIC ブリッジタイプのロードバランシング仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します。

```
-  bind lb vserver <name> (<serviceName>)

-  bind lb vserver <name> (<serviceName>)
```

例:

```
-  bind lb vserver quic_bridge_vip src1

-  bind lb vserver quic_bridge_vip src2
```


サービスグループの **QUIC** ブリッジの設定

サービスグループに QUIC ブリッジ機能を設定することもできます。次の手順では、サービスグループに QUIC ブリッジを設定する手順を示します。

サービスグループに QUIC ブリッジを設定するには、次の作業を完了する必要があります。

QUIC ブリッジプロファイルの追加

コマンドプロンプトで入力します。

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```

例**:**

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```

QUIC タイプのサーバーを追加する

コマンドプロンプトで入力します。

```
-  add server <name> (<IPAddress>)
-  add server <name> (<IPAddress>)
```

例**:**

```
-  add server s1 192.0.2.20
-  add server s2 192.0.2.30
```

QUIC ブリッジサービスグループの追加

コマンドプロンプトで入力します。

```
add serviceGroup <serviceGroupName> (<IP> | <serverName>) <serviceType>
```


例:

```
add serviceGroup svg1 QUIC_BRIDGE
```

QUIC サーバをサービスグループにバインドします

コマンドプロンプトで入力します。

```
- bind serviceGroup <serviceGroupName> (<IP>@ | (<serverName>) [-
    CustomServerID <string>]
- bind serviceGroup <serviceGroupName> (<IP>@ | (<serverName>) [-
    CustomServerID <string>]
```

例:

```
- bind serviceGroup svg1 s1 443 -customServerID C0A8026401BB
- bind serviceGroup svg1 s2 443 -customServerID C0A802C801BB
```

QUIC ブリッジタイプの負荷分散仮想サーバーの追加

コマンドプロンプトで入力します。

```
add lb vserver <name> [<IPAddress>@ <port> [-persistenceType <
    persistenceType >] [-lbMethod < lbMethod > [-cltTimeout <secs>] [-
    quickBridgeProfilename <name>]
```

例:

```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -cltTimeout 120 -
    quicBridgeProfilename q1
```

QUIC ブリッジタイプの負荷分散仮想サーバーをサービスグループにバインドします

コマンドプロンプトで入力します。


```
bind lb vserver <name>@ (<serviceName>@ <serviceGroupName>
```

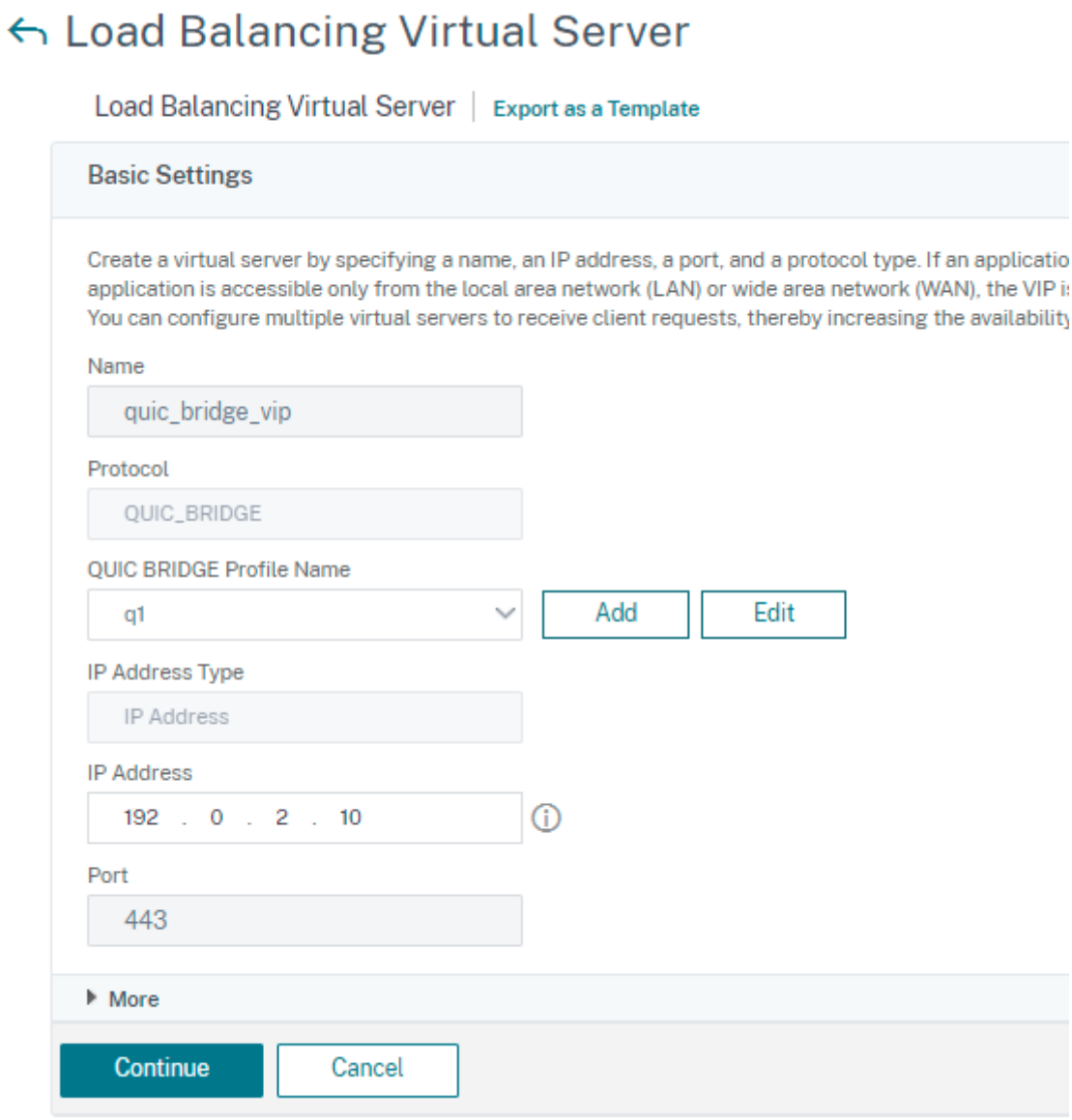
例**:**

```
bind lb vserver quic_bridge_vip svg1
```

GUI を使用した **QUIC** ブリッジの設定

GUI を使用して QUIC ブリッジを設定するには、次の手順を実行します。

1. [トラフィック管理**] > [**負荷分散**] > [**仮想サーバー] に移動します。
2. [仮想サーバー] ページで、[追加] をクリックします。
3. [負荷分散仮想サーバー] ページで、[プロトコル] を QUIC_BRIDGE として選択し、詳細を入力します。[**OK**] をクリックします。

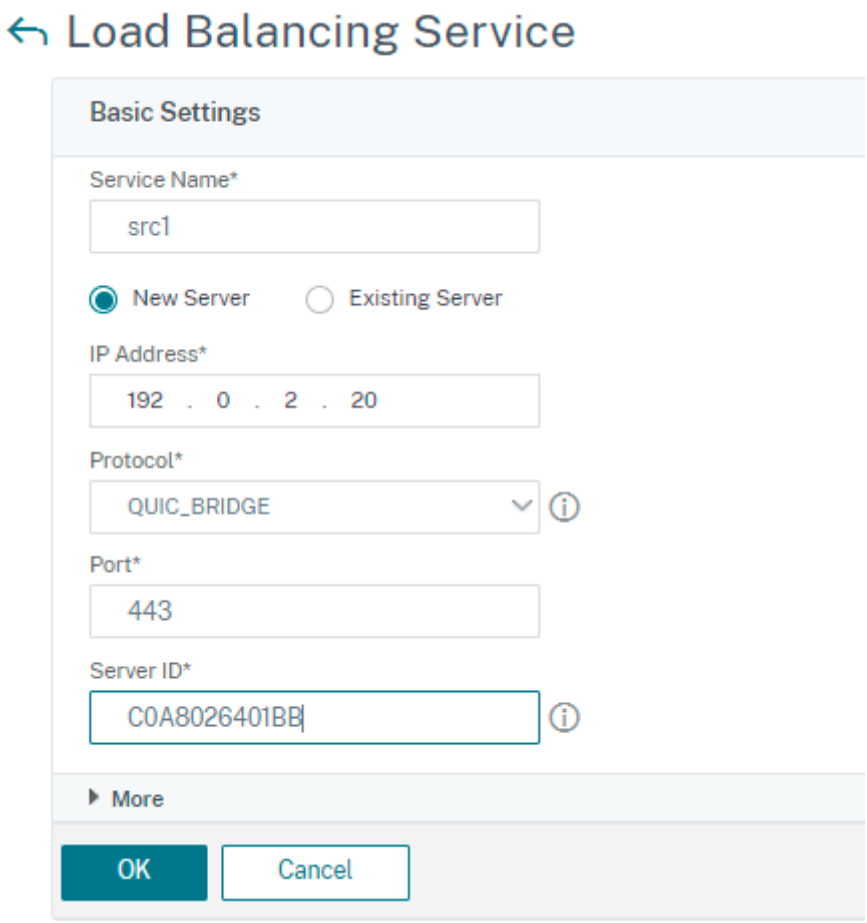


4. [負荷分散仮想サーバー] ページで、[続行] と [完了] をクリックします。

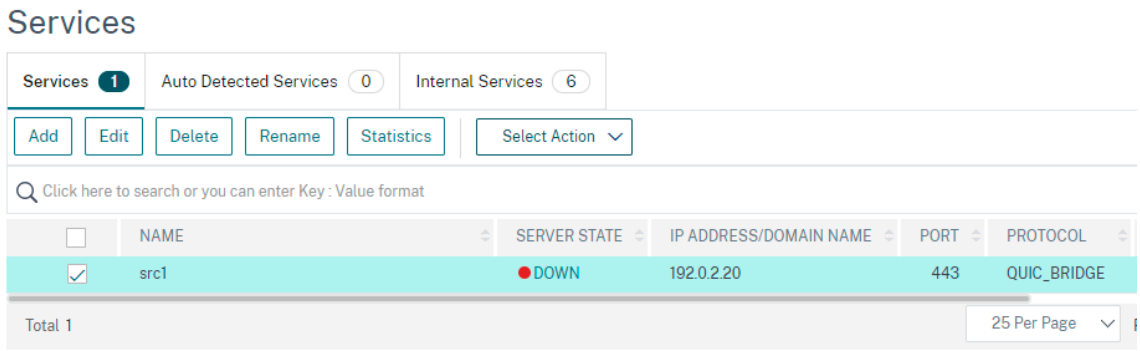
GUI を使用してサービスのロードバランシングを構成します

GUI を使用してサービスのロードバランシングを設定するには、次の手順を実行します。

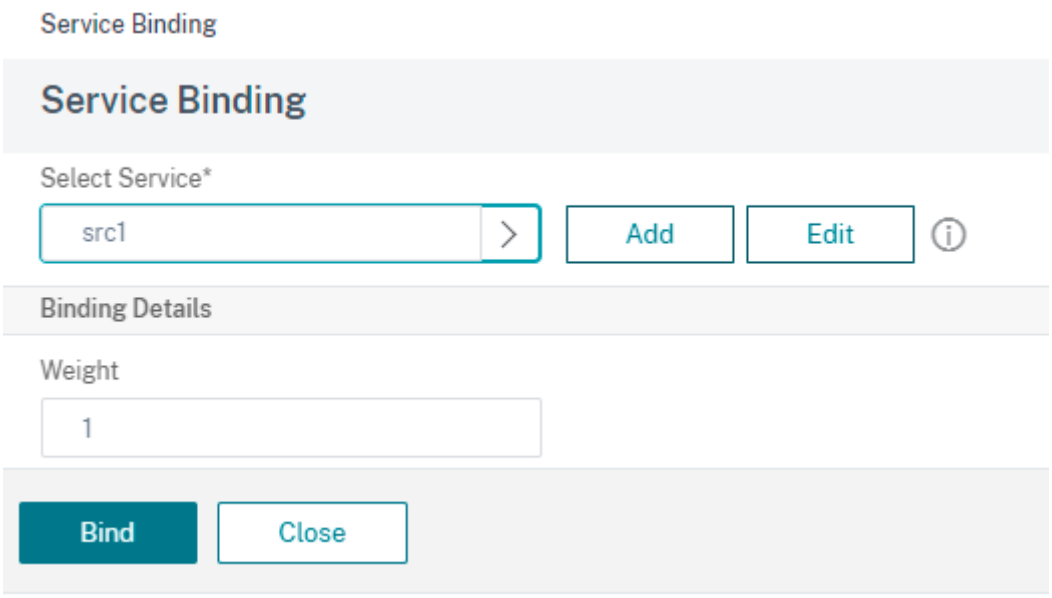
1. **Traffic Management > Load Balancing > Services** に移動します。[サービス] ページで、[追加] をクリックします。
2. ［負荷分散サービス］ページで、詳細を入力し、［**OK**］をクリックします。



3. [仮想サーバー] ページで、サービスをバインドするために作成された仮想サーバーを選択します。
4. [負荷分散仮想サーバー] ページを下にスクロールして、[サービスとサービスグループ] を選択します。
5. [サービスバインド] 画面で、[サービスフィールドの選択] をクリックします。
6. [サービス] 画面で、負荷分散仮想サーバーにバインドするサービスを選択し、[選択] をクリックします。



7. src1 サービスが選択され、[サービスバインド] 画面で [バインド] をクリックします。



8. [負荷分散仮想サーバー] ページで、[完了] をクリックします。

プロキシプロトコル

June 1, 2022

プロキシプロトコルは、Citrix ADC アプライアンスを介してクライアントからサーバーにクライアントの詳細を安全に転送します。アプライアンスは、クライアントの詳細を含むプロキシプロトコルヘッダーを追加し、バックエンドサーバーに転送します。以下は、Citrix ADC アプライアンスでのプロキシプロトコルの使用シナリオの一部です。

- 元のクライアント IP アドレスの学習
- Web サイトの言語を選択する
- 選択した IP アドレスの一覧表示をブロックする
- 統計のロギングと収集。

以下に、3 つの動作モードを示します。

- [挿入]。アプライアンスはクライアントの詳細を挿入し、バックエンドサーバーに送信します。
- フォワード。アプライアンスは、クライアントの詳細をバックエンドサーバに転送します。
- 剥ぎ取られた。アプライアンスは、ロギング目的でクライアントの詳細を保存します。また、プロキシプロトコルがバックエンドサーバーでサポートされていない場合は、書き換えポリシー設定を使用してクライアントの詳細をサーバーに送信します。

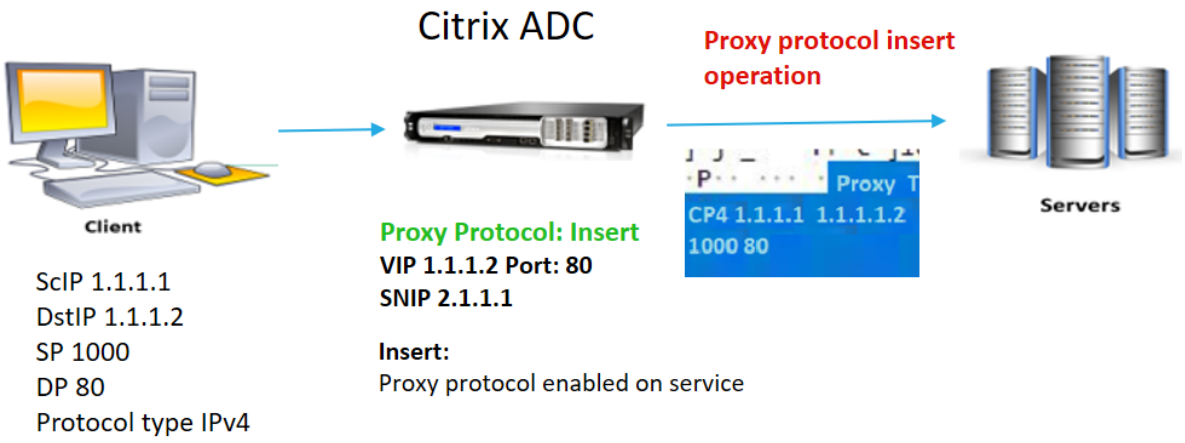
制限事項

プロキシプロトコルは、TCP Fast Open (TFO) およびマルチパス TCP 機能ではサポートされていません。この機能は、Citrix ADC アプライアンスが TCP 接続終了を行うサービスでのみサポートされます。他のサービス（「任意」など）はサポートされていません。

Citrix ADC アプライアンスでのプロキシプロトコルのしくみ

次のフロー図は、挿入、転送、および削除操作のために Citrix ADC アプライアンス間でプロキシプロトコルを構成する方法を示しています。

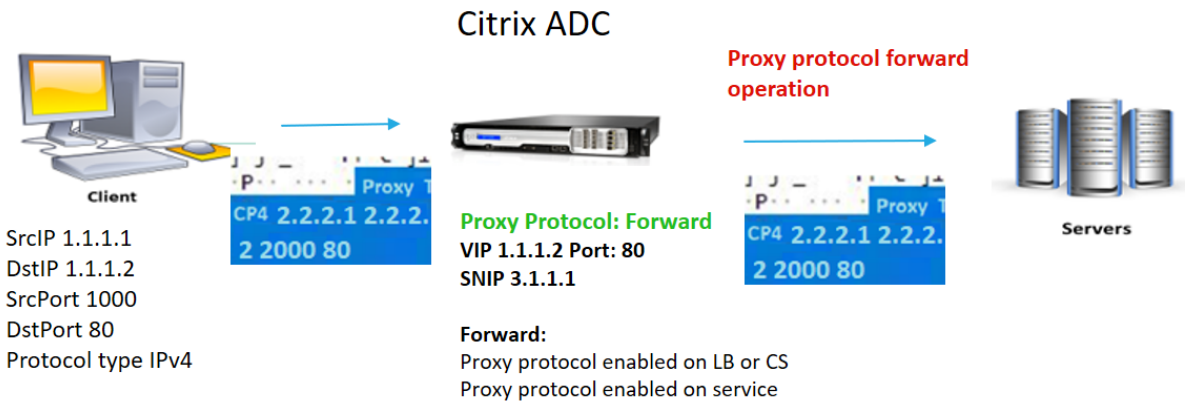
挿入操作



コンポーネントの相互作用は次のとおりです。

- Citrix ADC インスタンスでは、ネットプロファイルでプロキシプロトコルを有効にし、サービスにバインドする必要があります。
- 挿入操作では、Citrix ADC はクライアント接続の詳細を含むプロキシヘッダーを追加し、バックエンドサーバーに転送します。
- 送信側では、アプライアンスは CLI 設定に基づいてプロキシプロトコルのバージョンを決定します。

フォワード操作

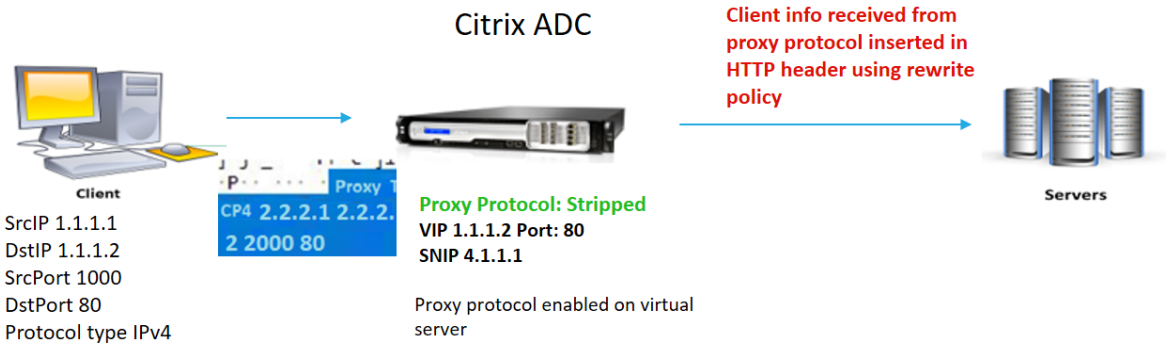


*** The original client details 2.2.2.1, 2.2.2.2, 2000, 80 in the proxy header is forwarded to the back-end server**

コンポーネントの相互作用は次のとおりです。

- クライアントは、プロキシヘッダーとともにリクエストを Citrix ADC に送信します。アプライアンスはバージョンを動的に識別します。
- Citrix ADC アプライアンスでは、転送操作です。プロキシプロトコルは、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーで有効になり、サービスでは有効になります。アプライアンスはプロキシヘッダーを受信し、ヘッダーの詳細をバックエンドサーバーに転送します。
- プロキシヘッダーの詳細が無効な形式の場合、アプライアンスは接続をリセットします。
- 送信側では、アプライアンスは CLI 設定に基づいてプロキシプロトコルのバージョンを決定します。

操作を剥ぎ取った



コンポーネントの相互作用は次のとおりです。

- クライアントは、プロキシヘッダーとともに要求を Citrix ADC アプライアンスに送信します。

- Citrix ADC アプライアンスでは、ストリップ操作の場合、アプライアンスはプロキシプロトコルから取得したクライアント情報を転送し、書き換えポリシー式を使用して HTTP ヘッダーに挿入します。
- 送信元 IP アドレス、宛先 IP アドレス、送信元ポート、宛先ポートなどのクライアントの詳細は、書き換えポリシー式を使用して HTTP ヘッダーに追加されます。書き換えポリシーによって式が評価され、「true」の場合、対応する書き換えポリシーアクションがトリガーされます。クライアントの詳細は HTTP ヘッダーでバックエンドサーバーに転送されます。
- プロキシヘッダーの詳細が無効な形式の場合、アプライアンスは接続をリセットします。

プロキシプロトコルのバージョン形式

プロキシプロトコルバージョンには 2 つの形式があります。アプライアンスは、着信データの長さに基づいてフォーマットを使用することを決定します。詳細については、プロキシプロトコル RFP を参照してください。

1. プロキシプロトコルバージョン 1 形式

   ```
   PROXY TCP4/TCP6/UNKNOWN <SRC IP> <DST IP> <SRC PORT> <DST PORT>
   ```

   - PROXY-> プロキシヘッダーバージョン -1 の一意の文字列形式。
   - プロトコル TCP over IPv4 および TCP over IPv6 をサポートします。残りのプロトコルでは、これは UNKNOWN です。
   - SRC IP：パケットの送信元 IP（元のクライアント IP）アドレス。
   - DST IP：パケットの宛先 IP アドレス。
   - SRC ポート：パケットの送信元ポート。
   - DST ポート：パケットの宛先ポート。

2. プロキシプロトコルバージョン 2 形式

   ```
   0D 0A 0D 0A 00 0D 0A 51 55 49 54 0A <13th byte> <14th byte> <15-16th byte> <17th byte onwards>
   ```

   - D 0A 0D 0A 00 0D 0A 51 55 49 54 0A-> プロキシヘッダバージョン-2 の一意のバイナリ文字列。
   - プロトコル TCP over IPv4 および TCP over IPv6 をサポートします。残りのプロトコルでは、これは UNKNOWN です。
   - 13 バイト：プロトコルのバージョンとコマンド。
   - 14 バイト：アドレスとプロトコルファミリ。
   - 15-16 バイト — ネットワークオーダーのアドレス長。
   - 17 バイト目以降：ネットワーク順に存在するアドレス情報（src IP、DST IP、src ポート、dst ポート）。

Citrix ADC アプライアンスでプロキシプロトコルを構成する

Citrix ADC アプライアンスでプロキシプロトコルを構成するには、次の手順を実行します。

1. プロキシプロトコルをグローバルとして有効にします。
2. 挿入操作用のプロキシプロトコルの設定

3. 転送操作用のプロキシプロトコルの設定
4. ストリップ操作用のプロキシプロトコルの設定
5. 操作がないようにプロキシプロトコルを構成する

プロキシプロトコルをグローバルとして有効にする

コマンドプロンプトで、次のように入力します。

```
set ns param -proxyProtocol ENABLED
```

挿入操作用のプロキシプロトコルの設定

挿入操作のプロキシプロトコルを構成するには、負荷分散仮想サーバーでプロトコルを有効または無効にし、サービスで有効にする必要があります。

負荷分散仮想サーバーのプロキシプロトコルを無効にしたネットプロファイルを追加する

コマンドプロンプトで、次のように入力します。

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
Add netprofile proxyprofile-1 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

> 注:
>
> アプライアンスでプロキシプロトコルを無効にする場合は、プロトコルバージョンパラメータを設定する必要はありません。

サービスに対してプロキシプロトコルが有効になっているネットプロファイルを追加する

コマンドプロンプトで、次のように入力します。

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
add netprofile proxyprofile-2 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```


プロキシレイヤーに **Citrix ADC** アプライアンスの負荷分散仮想サーバーを追加します

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

例:

```
add lb vserver lbvserver-1 http 1.1.1.1 80
```

プロキシレイヤーに **Citrix ADC** アプライアンスの **HTTP** サービスを追加します

コマンドプロンプトで、次のように入力します。

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

例:

```
Add service http-service-1 2.2.2.1 http 80
```

Citrix ADC アプライアンスで負荷分散仮想サーバーを使用してネットプロファイルを設定する

コマンドプロンプトで、次のように入力します。

```
set lb vserver <vserver name> -netprofile <name>
```

例:

```
set lb vserver lbvserver-1 -netprofile proxyProfile-1
```

Citrix ADC アプライアンスで **HTTP** サービスを使用してネットプロファイルを設定する

コマンドプロンプトで、次のように入力します。

```
set service <service name> -netprofile <name>
```

例:

```
set service http-service-1 -netprofile proxyProfile-1
```

転送操作用のプロキシプロトコルの設定

プロキシレイヤー内の次の Citrix ADC インスタンスの転送操作用のプロキシプロトコルを構成するには。プロトコルを有効または無効にし、仮想サーバーまたはサービスにバインドする必要があります。

負荷分散仮想サーバーでプロキシプロトコルが有効になっているネットプロファイルを追加する

コマンドプロンプトで、次のように入力します。

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
add netprofile proxyprofile-3 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

サービスに対してプロキシプロトコルが有効になっているネットプロファイルを追加する

コマンドプロンプトで、次のように入力します。

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

例:

```
add netprofile proxyprofile-4 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

プロキシレイヤーに **Citrix ADC** アプライアンスの負荷分散仮想サーバーを追加します

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

例:

```
add lb vserver lbvserver-2 http 2.2.2.2 80
```

プロキシレイヤーに **Citrix ADC** アプライアンスの **HTTP** サービスを追加します

コマンドプロンプトで、次のように入力します。

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

例:

```
Add service http-service-2 3.3.3.1 http 80
```

Citrix ADC アプライアンスで負荷分散仮想サーバーを使用してネットプロファイルを設定する

コマンドプロンプトで、次のように入力します。

```
set lb vserver <vserver name> -netprofile <name>
```


例:

```
set lb vserver lbvserver-2 -netprofile proxyProfile-3
```

Citrix ADC アプライアンスで **HTTP** サービスを使用してネットプロファイルを設定する

コマンドプロンプトで、次のように入力します。

```
set service <service name> -netprofile <name>
```

例:

```
set service http-service-2 -netprofile proxyProfile-4
```

ストリップ操作用のプロキシプロトコルの設定

ストリップ操作のプロキシプロトコルを構成するには、負荷分散仮想サーバーでプロキシプロトコルを有効にし、サービスのプロキシプロトコルを無効にする必要があります。

仮想サーバーでプロキシプロトコルが有効になっているネットプロファイルを追加する

コマンドプロンプトで、次のように入力します。

```
add netprofile <name> -proxyProtocol ENABLED> -proxyprotocoltxversion <V1/
V2>
```

例:

```
add netprofile proxyprofile-5 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

Citrix ADC アプライアンスの負荷分散またはコンテンツスイッチング仮想サーバーをプロキシレイヤーに追加する

コマンドプロンプトで、次のように入力します。

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

例:

```
add lb vserver lbvserver-3 http 2.2.2.2 80
```

プロキシレイヤーに **Citrix ADC** アプライアンスの **HTTP** サービスを追加します

コマンドプロンプトで、次のように入力します。

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```


例:

```
Add service http-service-3 3.3.3.1 http 80
```

Citrix ADC アプライアンスの負荷分散またはコンテンツスイッチング仮想サーバーを使用したネットプロファイルの設定

コマンドプロンプトで、次のように入力します。

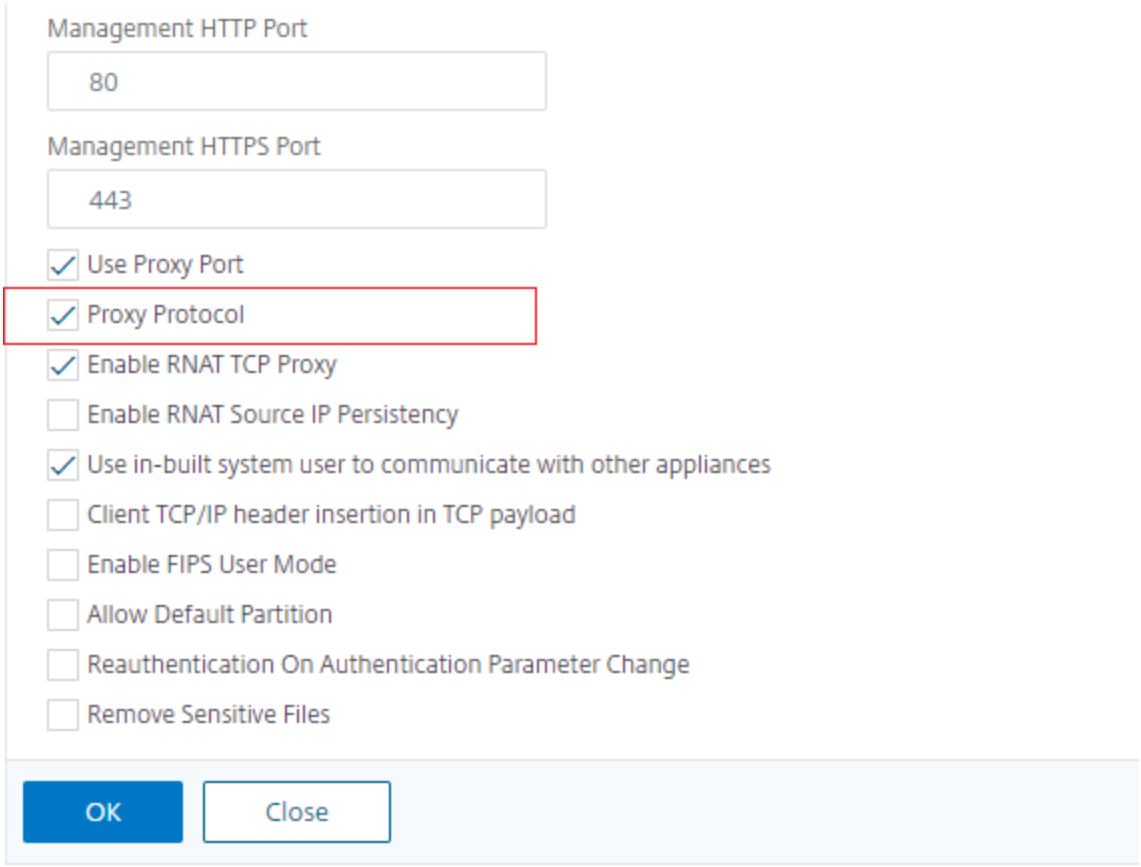
```
set lb vserver <vserver name> -netprofile <name>
```

例:

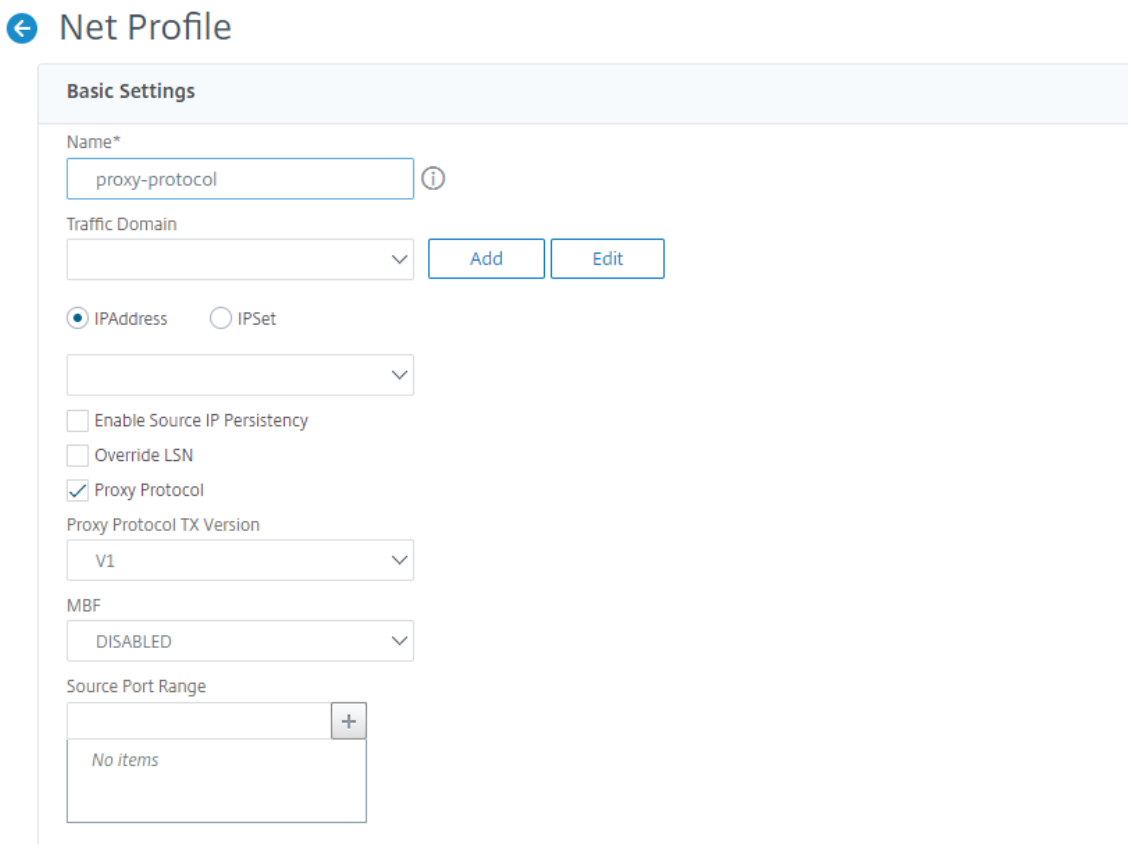
```
set lb vserver lbvserver-3 -netprofile proxyProfile-5
```

Citrix ADC GUI を使用してプロキシプロトコルを構成する

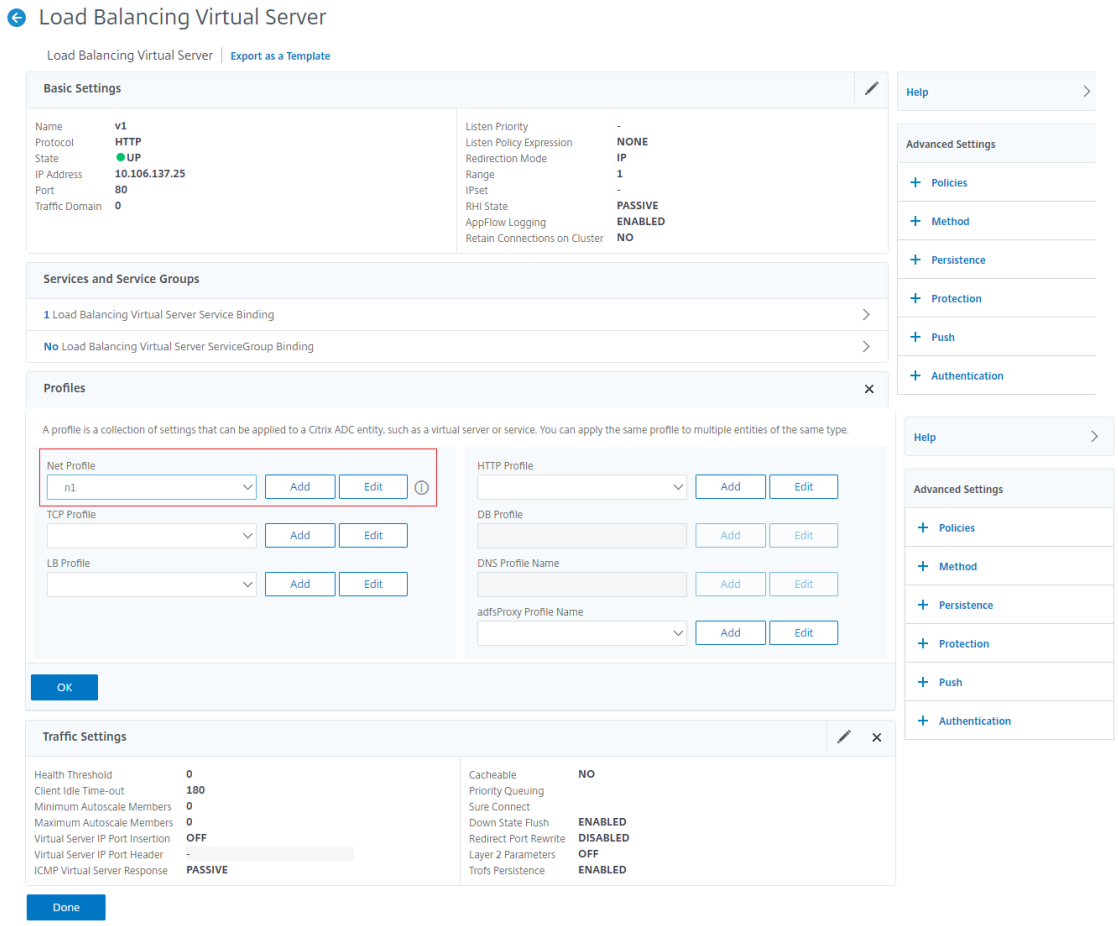
1. [システム] > [設定**] > [**グローバルシステム設定の変更] に移動します。
2. [グローバルシステム設定の構成**]** パラメータページで、[プロキシプロトコル] チェックボックスをオンにします。
3. [**OK**] をクリックして閉じます。



4. [システム] > [ネットワーク] > [ネットプロファイル] に移動します。
5. 詳細ウィンドウで、[追加] をクリックして、負荷分散仮想サーバーのネットプロファイルを作成します。
6. [ネットプロファイル] ページで、次のパラメータを設定します。
   a) Name：ネットプロファイルの名前。
   b) プロキシプロトコル。負荷分散仮想サーバーのプロキシプロトコルを有効または無効にします。
   c) プロキシプロトコル TX バージョン。受信データ形式に基づいて、プロキシプロトコルのバージョンを V1 または V2 に設定します。
7. [**OK**] をクリックします。

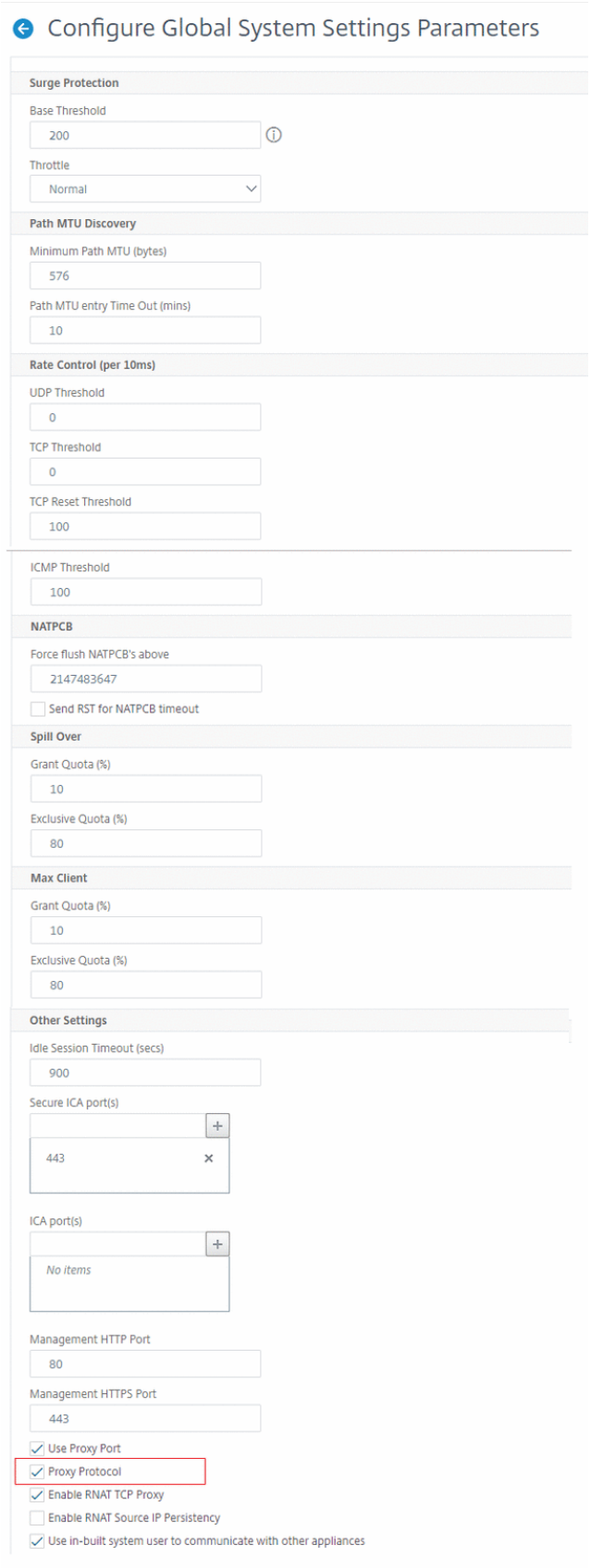


8. トラフィック管理] > [負荷分散] > [仮想サーバー] に移動します。
9. 詳細ウィンドウで、[追加] をクリックします。
10. [負荷分散仮想サーバー] ページで、基本パラメータを設定します。
11. **[**詳細設定**]** セクションで、[プロファイル] を選択します。
12. [プロファイル] セクションで、鉛筆アイコンをクリックします。
13. ネットプロファイルを選択し、「**OK**」をクリックします。
14. [完了] をクリックします。



15. トラフィック管理] > [負荷分散] > [サービス] に移動します。
16. 詳細ウィンドウで、[追加] をクリックします。
17. [負荷分散サービス] ページで、基本パラメータを設定します。
18. **[詳細設定]** セクションで、[プロファイル] を選択します。
19. [プロファイル] セクションで、鉛筆アイコンをクリックします。
20. ネットプロファイルを選択し、「**OK**」をクリックします。
21. [完了] をクリックします。

> 注:
>
> プロキシレイヤーの一部として複数の Citrix ADC アプライアンスがある場合は、転送操作のために各アプライアンスでプロキシプロトコル構成を設定する必要があります。



TCP オプションのクライアントの IP アドレス

October 7, 2021

Citrix ADC アプライアンスは、さまざまな方法でクライアント情報をバックエンドサーバーに送信します。そのような方法の 1 つは、最初のデータパケットの TCP オプションでクライアントの IP アドレスを送信することです。バックエンドサーバーが TCP オプションを使用してクライアントの IP アドレスを読み取る場合、アプライアンスは TCP プロファイルの TCP オプション番号を使用します。IP アドレスは、TCP オプション番号 28（アプライアンスサービスで設定可能）で伝送されます。
TCP オプション方式には、クライアント IP アドレスをバックエンドサーバに伝送する際に、挿入機能と転送機能の両方が含まれます。
TCP オプション構成では、アプライアンスは TCP オプション（28）を追加して、クライアント IP アドレスを挿入してバックエンドサーバーに転送します。次に、Citrix ADC アプライアンスでの TCP オプション構成の使用シナリオをいくつか示します。
TCP プロファイルに着信するトラフィックに対してこの機能が有効になっている場合、多重化は無効になります。また、TCP プロファイルの nsapimgr およびクライアントの tcp オプションが有効になっている場合は、クライアントの tcp オプションが優先されます。

> 注:
>
> ただし、TCP プロファイルに送信されるトラフィックに対して Client IP TCP オプションが有効になっている場合、アプライアンスでは多重化が無効になります。

- 元のクライアント IP アドレスを学習する
- Web サイトの言語の選択
- 選択した IP アドレスの一覧表示をブロックする

操作の 2 つのモードは次のとおりです。

- [挿入]: アプライアンスは、TCP オプション 28（設定可能ですが、推奨値は 28）フィールドにクライアントの詳細を追加し、それをバックエンドサーバーに送信します。
- 進む。アプライアンスは、TCP オプション 28（アプライアンスサービスのフロントエンドで設定可能）でクライアントの詳細を転送します。ただし、バックエンドのオプション番号は、バックエンドで設定された値に基づいて変更できます。

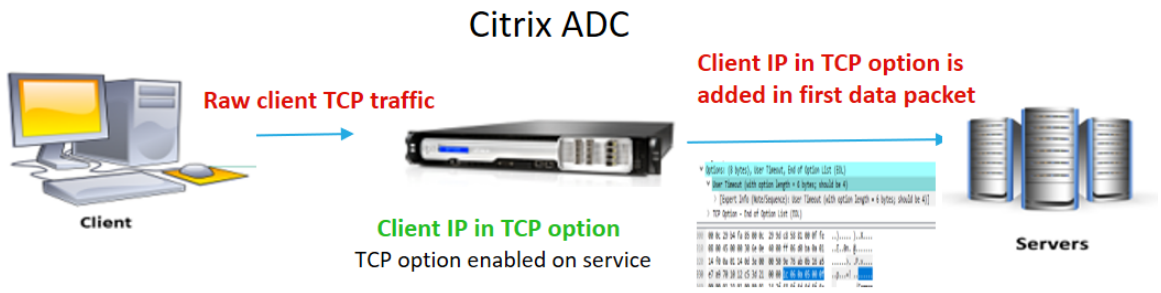
> 注:
>
> TCP または HTTP 仮想サーバの場合、TCP オプション番号は、トランスペアレントモードでこの機能が有効になっているかどうかに関係なく、転送されます。

制限事項

TCP オプション設定機能は、TFO、マルチパス TCP、および HTTP2 機能ではサポートされません。

Citrix ADC アプライアンスの **TCP** オプション構成方法

以下のフロー図は、Citrix ADC アプライアンスで挿入操作と転送操作用の TCP オプションを構成する方法を示しています。



コンポーネントの相互作用は次のとおりです。

- クライアントが Citrix ADC に要求を送信します。
- Citrix ADC アプライアンスでは、TCP プロファイルを作成し、TCP オプション機能を有効にして、TCP オプション番号を指定する必要があります。

  注：TCP プロファイルでは、TCP オプション番号を 28 に設定することをお勧めします。
- ［挿入］操作では、Citrix ADC は、サービスに対してバインドされた TCP オプション 28 にクライアントの詳細を挿入します。クライアントの詳細はバックエンドサーバーに送信されます。着信トラフィックが HTTPS の場合、TCP オプションのクライアント IP アドレスは、TCP レベルの最初のデータパケットである SSL クライアントの hello メッセージで送信されます。

フォワードオペレーション**:**

コンポーネントの相互作用は次のとおりです。

- クライアントは Citrix ADC に HTTP/HTTPS 要求を送信します。
- Citrix ADC アプライアンスで、が転送操作の場合、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーで TCP オプションが有効になり、サービスでも有効になります。アプライアンスは、仮想サーバで指定された TCP オプション番号でクライアント情報を受信し、最初のデータパケットに追加された TCP オプション番号（サービスで設定可能）でバックエンドサーバに転送します。

挿入操作の **TCP** オプションの構成

以下の手順に従って、Citrix ADC アプライアンスで TCP オプションを設定します。

1. TCP プロファイルを追加します。
2. 挿入操作の TCP オプションの構成
3. TCP プロファイルをサービスにバインドする

TCP プロファイルの追加

コマンドプロンプトで入力します。

```
add tcpprofile <name> -clientIpTcpOption (enabled | disabled> -clientIpTcpOptionNumber
 <positive_integer>
```

例:

```
add tcprofile p1
```

挿入操作の **TCP** オプションの構成

コマンドプロンプトで入力します。

```
add tcpprofile <name> -clientIpTcpOption (enabled | disabled> -clientIpTcpOptionNumber
 <positive_integer>
```

例:

```
add tcpprofile p1 -clientIpTcpOption ENABLED -clientIpTcpOptionNumber 28
```

サービスの追加

コマンドプロンプトで入力します。

```
add service <name> <server name> <service type> <port>
```

例:

```
add service service-http1 1.1.1.1 HTTP 80
```

TCP プロファイルをサービスにバインドする

コマンドプロンプトで入力します。

```
set service <name> -tcpprofileName <name>
```

例:

```
set service s1 -tcpprofileName p1
```


> 注:
>
> サービスの基本設定には注意が必要です。

転送操作の **TCP** オプションの構成

転送動作のための TCP プロファイルで TCP オプションを設定するには、以下の手順に従います。

1. TCP オプション番号を使用して TCP プロファイルを追加します
2. TCP プロファイルを仮想サーバーにバインドする
3. TCP プロファイルをサービスにバインドします。

TCP オプション番号を使用して **TCP** プロファイルを追加します

コマンドプロンプトで入力します。

```
add tcpprofile <name> -clientIpTcpOption (enabled | disabled> -clientIpTcpOptionNumber
 <positive_integer>
```

例:

```
add tcpprofile p1 -clientIpTcpOption ENABLED -clientIpTcpOptionNumber 29
```

TCP プロファイルを仮想サーバーにバインドする（負荷分散またはコンテンツスイッチング）

コマンドプロンプトで入力します。

```
set lb vserver <name> -tcpprofileName <name>
```

例:

```
set lb vservice s1 -tcpprofileName p1
```

TCP プロファイルをサービスにバインドする

コマンドプロンプトで入力します。

```
set service <name> -tcpprofileName p1
```

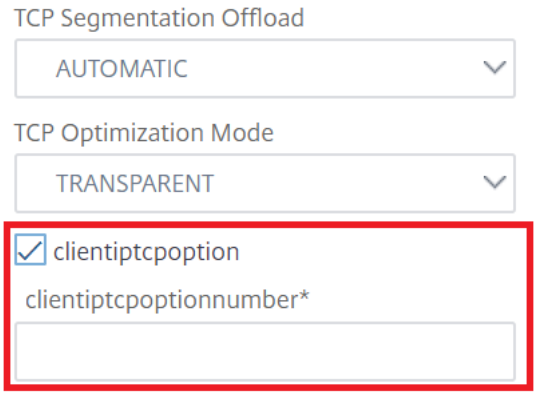
例:

```
set service s1 -tcpprofileName p1
```

Citrix ADC GUI を使用して **TCP** オプションを構成する

1. **System** > **Profiles** に移動します。
2. [**TCP** プロファイル] タブページで、[追加] をクリックします。

3. [**TCP** プロファイルの設定] ページで、次のパラメータを構成します。
   a. clientIptcpoption。クライアントの IP アドレスを送受信する TCP オプション。
   b. clientiptcpoptionnumber。クライアント IP アドレスを受信するための設定可能な TCP オプション番号。



4. [**OK**] をクリックして [閉じる] をクリックします。

SNMP

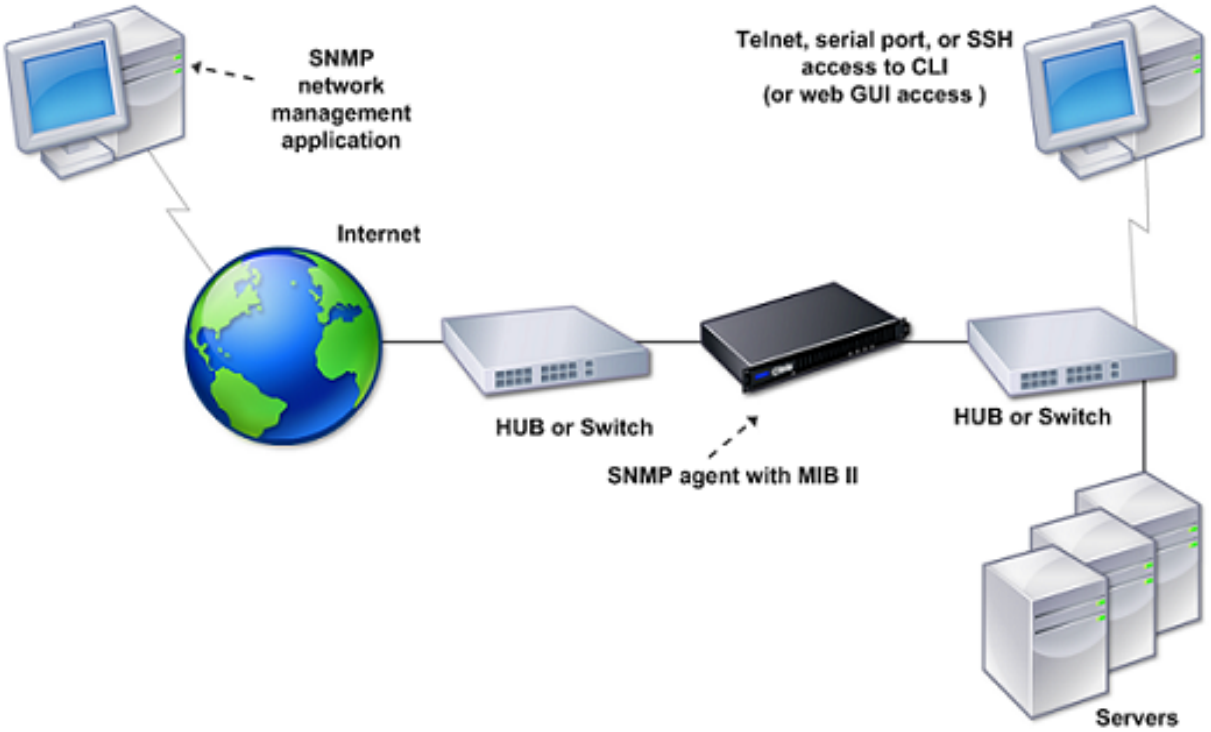
October 7, 2021

簡易ネットワーク管理プロトコル（SNMP）を使用して、Citrix ADC アプライアンス上の SNMP エージェントを構成して、トラップと呼ばれる非同期イベントを生成できます。トラップは、Citrix ADC で異常な状態が発生したときに生成されます。その後、トラップは トラップリスナーと呼ばれるリモートデバイスに送信され、Citrix ADC アプライアンスの異常状態を通知します。または、SNMP マネージャと呼ばれるリモートデバイスからシステム固有の情報を *SNMP* エージェントに照会することもできます。次に、エージェントは管理情報ベース（MIB）で要求されたデータを検索し、そのデータを SNMP マネージャに送信します。

Citrix ADC 上の SNMP エージェントは、SNMPv1、SNMPv2、SNMPv3 に準拠したトラップを生成できます。クエリでは、SNMP エージェントは SNMP バージョン 1（SNMPv1）、SNMP バージョン 2（SNMPv2）、および SNMP バージョン 3（SNMPv3）をサポートします。

SNMP パラメータ、トラップ、およびその説明については、Citrix ADC SNMP OID リファレンスを参照してください。

次の図は、SNMP が有効で構成されている Citrix ADC を使用するネットワークを示しています。この図では、各 SNMP ネットワーク管理アプリケーションは、SNMP を使用して Citrix ADC 上の SNMP エージェントと通信しています。SNMP エージェントは、管理情報ベース（MIB）を検索して、SNMP マネージャによって要求されたデータを収集し、その情報をアプリケーションに提供します。



> 重要
>
> Citrix ADC アプライアンスの SNMP モジュールは、SNMP OID の最大 128 バイト（RFC 3416 に準拠している）をサポートします。オブジェクトの長いインデックス変数名を使用すると、SNMP OID の長さが 128 バイトを超える場合があります。
>
> この問題を解決するために、Citrix ADC SNMP モジュールは、インデックス変数名に対して最大 31 文字をサポートします。インデックス変数名の長さが 31 文字を超える場合、ハッシュアルゴリズムを使用する SNMP モジュールは、名前を 31 文字のハッシュ値に変換します。このハッシュ値は、その変数の SNMP OID で使用されます。
>
> 元のインデックス変数名は、次の名前の形式を持つ別の変数に格納されます。`<variable type>FullName`。たとえば、負荷分散仮想サーバーの名前が 31 文字を超える場合、`vserverName` SNMP OID にはハッシュ値が含まれ、`vsvrFullName` SNMP OID には仮想サーバーの完全な（元の）名前が含まれます。
>
> 同様に、SNMP トラップの場合、インデックス変数にはハッシュ値が表示されます。`<variable type>FullName`は、元のインデックス変数名のフルネームを格納し、トラップメッセージの一部でもあります。

SNMP マネージャおよびトラップリスナーへの MIB ファイルのインポート

Citrix ADC アプライアンスを監視するには、MIB オブジェクト定義ファイルをダウンロードする必要があります。Citrix ADC アプライアンスは、次のエンタープライズ固有の MIB をサポートします。

- 標準 **MIB-2** グループのサブセット。MIB-2 グループの SYSTEM、IF、ICMP、UDP、および SNMP を提供

します。

- システムエンタープライズ **MIB**。システム固有の設定と統計情報を提供します。

MIB オブジェクト定義ファイルは、/netscaler/snmp ディレクトリまたは GUI の [ダウンロード] タブから取得できます。

SNMP トラップを生成するように **Citrix ADC** を構成する

October 7, 2021

Citrix ADC アプライアンスは、トラップと呼ばれる非同期イベントを生成するように設定できます。トラップは、アプライアンスに異常な状態がある場合に生成されます。トラップは、トラップ リスナーと呼ばれるリモートデバイスに送信されます。管理者は、アプライアンスを監視し、問題に対して迅速に対応できます。

Citrix ADC アプライアンスは、*SNMP* アラームと呼ばれる一連の条件エンティティを提供します。SNMP アラームの条件が満たされると、アプライアンスは SNMP トラップメッセージを生成し、設定されたトラップリスナーに送信します。たとえば、LOGIN-FAILURE アラームが有効の場合、アプライアンスでログインに失敗すると、トラップメッセージが生成され、トラップリスナーに送信されます。

トラップを生成するように Citrix ADC アプライアンスを設定するには、アラームを有効にして設定する必要があります。次に、アプライアンスが生成したトラップメッセージを送信するトラップリスナーを指定します。

SNMP アラームの有効化

Citrix ADC アプライアンスは、有効になっている SNMP アラームに対してのみトラップを生成します。一部のアラームはデフォルトで有効になっていますが、無効にすることもできます。

SNMP アラームを有効にすると、一部のイベントが発生すると、アプライアンスは対応するトラップメッセージを生成します。一部のアラームは、デフォルトで有効になっています。

CLI を使用して **SNMP** アラームを有効にするには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `enable snmp alarm <trapName>`
- `show snmp alarm <trapName>`

GUI を使用して **SNMP** アラームを有効にするには

1. [システム**] > [SNMP] > [**アラーム] に移動し、アラームを選択します。
2.「アクション」をクリックし、「有効化」を選択します。

アラームの設定

Citrix ADC アプライアンスは、*SNMP* アラームと呼ばれる一連の条件エンティティを提供します。SNMP アラームに設定された条件が満たされると、アプライアンスは SNMP トラップメッセージを生成し、設定済みのトラップリスナーに送信します。たとえば、LOGIN-FAILURE アラームが有効の場合、アプライアンスでログインに失敗すると、トラップメッセージが生成され、トラップリスナーに送信されます。

重大度レベルの SNMP アラームを割り当てることができます。これを行うと、対応するトラップメッセージにその重大度が割り当てられます。

アプライアンスに定義されている重大度レベルを、重大度の降順に示します。

- 重大
- 重要
- 軽度
- 警告
- 情報

たとえば、LOGIN-FAILURE という名前の SNMP アラームの警告重大度を設定すると、ログイン障害があるときに生成されるトラップメッセージに警告重大度が割り当てられます。

> 注
>
> Citrix ADC は、さまざまな SNMP アラームをサポートしています。詳細については、「SNMP アラーム」を参照してください。

SNMP アラームを設定して、そのアラームの条件が満たされたときに生成される対応するトラップメッセージをログに記録することもできます。

CLI を使用して **SNMP** アラームを設定するには

コマンドプロンプトで次のコマンドを入力して、SNMP アラームを設定し、設定を確認します。

- `set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm <trapName>`

各項目の意味は次のとおりです。

しきい値: 上限しきい値の値。Citrix ADC アプライアンスは、アラームに関連付けられた属性の値が、指定された上限しきい値以上になると、SNMP トラップメッセージを生成します。

ノーマル値: ノーマルしきい値の値。上限しきい値を超えた後に、それぞれの属性の値がこの値以下になると、トラップメッセージが生成されます。

GUI を使用して **SNMP** アラームを設定するには

[システム] **> [SNMP] > [**アラーム] に移動し、アラームを選択し、アラームパラメータを設定します。

SNMPv1 トラップまたは **SNMPv2** トラップの設定

アラームを設定したら、アプライアンスがトラップメッセージを送信するトラップリスナーを指定する必要があります。IP アドレスや IPv6 アドレス、トラップリスナーの宛先ポートなどのパラメーターの指定以外に、トラップのタイプ（汎用または特定）と SNMP バージョンを指定できます。

汎用または専用のトラップを受信するために、最大 20 のトラップリスナーを構成できます。

また、Citrix ADC IP（NSIP または NSIP6）アドレス以外の送信元 IP アドレスを持つ SNMP トラップ・メッセージを特定のトラップ・リスナーに送信するようにアプライアンスを構成することもできます。IPv4 アドレスを持つトラップリスナーの場合、送信元 IP を、アプライアンス上で構成されたマッピング IP（MIP）アドレスまたはサブネット IP（SNIP）アドレスのいずれかに設定できます。IPv6 アドレスを持つトラップリスナーの場合、アプライアンスに構成されたサブネット IPv6（SNIP6）アドレスにソース IP を設定できます。

また、重大度に基づいてトラップリスナーにトラップメッセージを送信するようにアプライアンスを設定することもできます。たとえば、トラップリスナーに対して重大度を Minor に設定すると、重大度レベルが Minor（Minor、Major、Critical）以上のすべてのトラップメッセージがトラップリスナーに送信されます。

トラップリスナーにコミュニティストリングを定義した場合は、リスナーに送信されるトラップごとにコミュニティストリングも指定する必要があります。コミュニティストリングが定義されているトラップリスナーは、トラップリスナーで定義されたコミュニティストリングと一致するコミュニティストリングを含むトラップメッセージだけを受け入れます。その他のトラップメッセージはドロップされます。

CLI を使用して **SNMP** トラップを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 )-destPort <port> -communityName <string> -srcIP <ip_addr> -severity <severity>`
- `show snmp trap`

例:

```
> `add snmp trap specific 192.0.2.10 -version V2 -destPort 80 -communityName com1 -severity Major`
<!--NeedCopy-->
```


GUI を使用して **SNMP** トラップを構成するには

[システム] **> [SNMP] > [**トラップ] に移動し、SNMP トラップを作成します。

SNMPv3 トラップの設定

SNMPv3 は、SNMP ユーザのクレデンシャルを使用した認証や暗号化などのセキュリティ機能を提供します。SNMP マネージャは、SNMP ユーザに割り当てられたパスワードが設定されている場合に限り、SNMPv3 トラップメッセージを受信できます。

トラップ宛先は、SNMPv1、SNMPv2、および SNMPv3 トラップメッセージを受信できるようになりました。

CLI を使用して **SNMPv3** トラップを設定するには

コマンドプロンプトで、次の操作を行います。

1. SNMPv3 トラップを追加します。

   ```
   add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 | V3)
   -destPort <port> -communityName <string> -srcIP <ip_addr> -severity <
   severity>
   ```

   > 注
   >
   > いったん設定すると、SNMP トラップのバージョンは変更できません。

   例

   ```
   1 > add snmp trap specific 192.0.2.10 -version V3 -destPort 80 -
       communityName com1 -severity Major
   2 <!--NeedCopy-->
   ```

2. SNMP ユーザを追加します。

   ```
   add snmp user <name> -group <string> [ -authType ( MD5 | SHA ){ -
   authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]
   ```

   例

   ```
   1 > add snmp user edocs_user -group edocs_group
   2 <!--NeedCopy-->
   ```

3. SNMPv3 トラップを SNMP ユーザにバインドします。


```
bind snmp trap <trapClass> <trapDestination> [-version <version>] (-userName
 <string> [-securityLevel <securityLevel>])
```

例

```
1      > bind snmp trap specific 192.0.2.10 -version V3 -userName
          edocs_user -securityLevel authPriv
2  <!--NeedCopy-->
```

GUI を使用して **SNMPv3** トラップを設定するには

1. SNMPv3 トラップを追加します。

   [システム] **>** **[SNMP]** **>** **[**トラップ] に移動し、SNMP バージョンとして V3 を選択して SNMP トラップを作成します。

2. SNMP ユーザを追加します。

   [システム] **>** **[SNMP]** **>** **[**ユーザ] に移動し、SNMP ユーザを作成します。

3. SNMPv3 トラップを SNMP ユーザにバインドします。

   - [システム] **>** **[SNMP]** **>** **[**トラップ] に移動し、SNMP バージョン 3 トラップを選択します。
   - トラップをバインドするユーザーを選択し、適切なセキュリティ・レベルを定義します。

SNMP トラップロギング

Citrix ADC アプライアンスは、SNMP トラップロギングオプションを有効にし、アプライアンス上で少なくとも 1 つのトラップリスナーが設定されている場合、SNMP トラップメッセージ（ロギング機能が有効になっている SNMP アラーム用）を記録できます。これで、外部ログサーバーに送信されるトラップメッセージの監査ログレベルを指定できるようになりました。デフォルトのログレベルは [情報] です。設定可能な値は、緊急、アラート、クリティカル、エラー、警告、デバッグ、および通知です。

たとえば、ログオンの失敗によって生成される SNMP トラップメッセージの監査ログレベルを [緊急] に設定できます。この情報は、NSLOG サーバまたは SYSLOG サーバでトラブルシューティングに使用できます。

CLI を使用して **SNMP** トラップロギングを有効にし、トラップログレベルを設定するには

コマンドプロンプトで次のコマンドを入力して、SNMP トラップロギングを構成し、構成を確認します。

- `set snmp option [-snmpTrapLogging (ENABLED | DISABLED)][-snmpTrapLoggingLevel <snmpTrapLoggingLevel>]`
- `show snmp option`

GUI を使用して **SNMP** トラップロギングを有効にし、**SNMP** トラップログレベルを構成するには

[システム**] > [SNMP**] に移動し、[SNMP オプションの変更] をクリックし、次のパラメータを設定します。

1. SNMP トラップロギング：アプライアンスに少なくとも 1 つのトラップリスナーが設定されている場合に SNMP トラップロギングを有効にするには、このチェックボックスをオンにします。
2. [SNMP トラップロギングレベル]：SNMP トラップの監査ログレベルを選択します。デフォルトでは、SNMP トラップの監査レベルは「情報」に設定されています。

SNMP v1 および **v2** クエリに対する **Citrix ADC** 構成

October 7, 2021

Citrix ADC SNMP エージェントに対して、SNMP マネージャーと呼ばれるリモートデバイスからシステム固有の情報を問い合わせることができます。次に、エージェントは管理情報ベース（MIB）で要求されたデータを検索し、そのデータを SNMP マネージャに送信します。

SNMP エージェントでは、次のタイプの SNMP v1 および v2 クエリがサポートされています。

- GET
- GET NEXT
- ALL
- GET BULK

コミュニティストリングと呼ばれるストリングを作成し、それぞれをクエリタイプに関連付けることができます。1 つ以上のコミュニティストリングを各クエリタイプに関連付けることができます。コミュニティストリングはパスワードで、SNMP マネージャからの SNMP クエリを認証するために使用されます。

たとえば、**ab** c や **bcd** などの 2 つのコミュニティ文字列をクエリタイプ GET NEXT に関連付けると、Citrix ADC アプライアンス上の SNMP エージェントは、**ab** c または **bcd** を含む GET NEXT SNMP クエリパケットのみをコミュニティ文字列とみなします。

SNMP マネージャの指定

Citrix ADC アプライアンスを構成して、適切な SNMP マネージャーがクエリを実行できるようにする必要があります。また、必要な Citrix ADC 固有の情報を SNMP マネージャーに提供する必要があります。最大 100 個の SNMP マネージャまたはネットワークを追加できます。

IPv4 SNMP マネージャの場合は、マネージャの IP アドレスの代わりにホスト名を指定できます。その場合は、SNMP マネージャのホスト名を IP アドレスに解決する DNS ネームサーバを追加する必要があります。最大 5 つのホスト名ベースの SNMP マネージャを追加できます。

> 注:
> アプライアンスは、IPv6 アドレスを持つ SNMP マネージャーのホスト名の使用をサポートしていません。IPv6 アドレスを指定する必要があります。

少なくとも 1 つの SNMP マネージャを設定しない場合、アプライアンスはネットワーク上のすべての IP アドレスからの SNMP クエリを受け入れ、応答します。1 つ以上の SNMP マネージャーを構成する場合、アプライアンスはこれらの特定の IP アドレスからの SNMP クエリのみを受け入れ、応答します。

設定から SNMP マネージャを削除すると、そのマネージャはアプライアンスを照会できなくなります。

コマンドラインインターフェイスを使用して **IP** アドレスを指定して **SNMP** マネージャーを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp manager <IPAddress> ... [-netmask <netmask>]`
- `show snmp manager`

例

```
> add snmp manager 10.102.29.10 10.102.29.15 10.102.29.30
```

コマンドラインインターフェイスを使用してホスト名を指定して **SNMP** マネージャーを追加するには

重要:IP アドレスの代わりに SNMP マネージャのホスト名を指定する場合は、ホスト名を SNMP マネージャの IP アドレスに解決するように DNS ネームサーバを構成する必要があります。詳細については、「ネームサーバーの追加」を参照してください。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp manager <IPAddress> [-domainResolveRetry ****<integer>]`
- `show snmp manager`

例

```
add nameserver 10.103.128.15
add snmp manager engwiki.eng.example.net -domainResolveRetry 10
```

GUI を使用して **SNMP** マネージャーを追加するには

1. [システム] > [**SNMP**] > [マネージャ] に移動し、SNMP マネージャを作成します。

   > 重要:

> IPv4 アドレスの代わりに SNMP マネージャーのホスト名を指定する場合は、ホスト名を SNMP マネージャの IP アドレスに解決するように DNS ネームサーバーを構成する必要があります。

> 注:
> アプライアンスは、IPv6 アドレスを持つ SNMP マネージャーのホスト名をサポートしていません。

SNMP コミュニティの指定

コミュニティストリングと呼ばれるストリングを作成し、アプライアンスで次の SNMP クエリタイプに関連付けることができます。

- GET
- GET NEXT
- ALL
- GET BULK

1 つ以上のコミュニティストリングを各クエリタイプに関連付けることができます。たとえば、**abc** や **bcd** などの 2 つのコミュニティストリングをクエリタイプ GET NEXT に関連付けると、アプライアンスの SNMP エージェントは、**abc** または **bcd** を含む GET NEXT SNMP クエリパケットだけをコミュニティストリングと見なします。

クエリタイプにコミュニティストリングを関連付けない場合、SNMP エージェントはそのタイプのすべての SNMP クエリに応答します。

コマンドラインインターフェイスを使用して **SNMP** コミュニティを指定するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp community <communityName> <permissions>`
- `show snmp community`

例

```
> add snmp community com all
```

GUI を使用して **SNMP** コミュニティストリングを設定するには

[システム] > [**SNMP**] > [コミュニティ] に移動し、SNMP コミュニティを作成します。

SNMPv3 クエリに対する Citrix ADC の構成

October 7, 2021

簡易ネットワーク管理プロトコルバージョン 3（SNMPv3）は、SNMPv1 および SNMPv2 の基本構造とアーキテクチャに基づいています。ただし、SNMPv3 では、認証、アクセスコントロール、データ整合性チェック、データ発信元検証、メッセージの適時性チェック、データの機密性などの管理機能とセキュリティ機能を組み込むための基本アーキテクチャが強化されています。

メッセージレベルのセキュリティとアクセスコントロールを実装するために、SNMPv3 では、ユーザベースセキュリティモデル（USM）とビューベースアクセスコントロールモデル（VACM）が導入されています。

- ユーザーベースのセキュリティモデル。ユーザベースセキュリティモデル（USM）は、メッセージレベルのセキュリティを提供します。SNMP エージェントと SNMP マネージャのユーザおよびセキュリティパラメータを設定できます。USM には次の機能があります。
  - データの整合性**:** ネットワーク経由の送信中にメッセージが変更されないように保護します。
  - データ発信元検証：メッセージ要求を送信したユーザーを認証します。
  - メッセージの適時性**:** メッセージの遅延や再生から保護します。
  - データの機密性**:** メッセージの内容が、権限のない団体や個人に開示されないように保護します。
- ビューベースのアクセス制御モデル。ビューベースアクセスコントロールモデル（VACM）を使用すると、セキュリティレベル、セキュリティモデル、ユーザ名、ビュータイプなど、さまざまなパラメータに基づいて MIB の特定のサブツリーへのアクセス権を設定できます。これにより、エージェントを設定して、異なるマネージャに MIB へのさまざまなレベルのアクセスを提供できます。

Citrix ADC では、SNMPv3 のセキュリティ機能を実装するための次のエンティティがサポートされています。

- SNMP エンジン
- SNMP ビュー
- SNMP グループ
- SNMP ユーザ

これらのエンティティは連携して機能し、SNMPv3 セキュリティ機能を実装します。ビューは、MIB のサブツリーへのアクセスを許可するために作成されます。次に、必要なセキュリティレベルと定義されたビューへのアクセス権を持つグループが作成されます。最後に、ユーザーが作成され、グループに割り当てられます。

> 注:
>
> ビュー、グループ、およびユーザーの構成が同期され、高可用性（HA）ペアのセカンダリ・ノードに伝播されます。ただし、エンジン ID は各 Citrix ADC アプライアンスに固有であるため、伝播も同期もされません。

メッセージ認証とアクセス制御を実装するには、次の操作を行う必要があります。

エンジン **ID** の設定

SNMP エンジンは、SNMP エージェントに存在するサービスプロバイダーです。メッセージの送信、受信、認証などのサービスを提供します。SNMP エンジンは、エンジン ID を使用して一意に識別されます。

Citrix ADC アプライアンスは、そのインターフェイスの MAC アドレスに基づいて一意のエンジン ID を持ちます。engineID を上書きする必要はありません。ただし、エンジン ID を変更する場合は、リセットできます。

コマンドラインインターフェイスを使用してエンジン **ID** を設定するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `set snmp engineId <engineID>`
- `show snmp engineId`

例

```
> set snmp engineId 8000173f0300c095f80c68
```

GUI を使用してエンジン **ID** を設定するには

[システム] > [**SNMP**] > [ユーザー] に移動し、[エンジン **ID** の設定] をクリックしてエンジン ID を入力します。

ビューを設定する

SNMP ビューは、MIB の特定の部分へのユーザアクセスを制限します。SNMP ビューは、アクセスコントロールを実装するために使用されます。

コマンドラインインターフェイスを使用して **SNMP** ビューを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp view <name> <subtree> -type ( included | excluded )`
- `show snmp view <name>`
- `rm snmp view <name> <subtree>`

各項目の意味は次のとおりです。

Name: SNMPv3 ビューの名前。大文字と小文字、数字、ハイフン（-)、ピリオド（#)、スペース（)、アットマーク(@)、等号（=)、コロン（:)、およびアンダースコア（_）を含む 1～31 文字で構成できます。SNMPv3 ビューの識別に役立つ名前を選択する必要があります。

Subtree。この SNMPv3 ビューに関連付ける MIB ツリーの特定のブランチ（サブツリー)。サブツリーは SNMP OID として指定する必要があります。これは最大の長さの引数です:99.

type。サブツリーパラメータで指定されたサブツリーをこのビューに含めるか、またはこのビューから除外します。この設定は、A などのサブツリーを SNMPv3 ビューに追加し、A の特定のサブツリー（B など）を SNMPv3 ビューから除外する場合に便利です。これは必須の引数です。指定可能な値: 含める、除外する。

例

add snmp view SNMPv3test 1.1.1.1 -type included
sh snmp view SNMPv3test
rm snmp view SNMPv3test 1.1.1.1

GUI を使用して **SNMP** ビューを構成するには

[システム] > [**SNMP] > [**ビュー**]** に移動し、SNMP ビューを作成します。

グループを構成する

SNMP グループは、SNMP ユーザの論理的な集約です。これらは、アクセス制御を実装し、セキュリティレベルを定義するために使用されます。SNMP グループを構成して、そのグループに割り当てられたユーザーのアクセス権を設定し、ユーザーを特定のビューに制限できます。

そのグループに割り当てられたユーザーのアクセス権を設定するには、SNMP グループを構成する必要があります。

コマンドラインインターフェイスを使用して **SNMP** グループを追加するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp group <name> <securityLevel> -readViewName <string>`
- `show snmp group <name> <securityLevel>`

各項目の意味は次のとおりです。

Name: SNMPv3 グループの名前。大文字と小文字、数字、ハイフン (-)、ピリオド (.)、ポンド (#)、スペース ()、アットマーク (@)、等号 (=)、コロン (:)、アンダースコア (_) などの文字を使用できます。SNMPv3 グループの識別に役立つ名前を選択する必要があります。

securityLevel。Citrix ADC アプライアンスとグループに属する SNMPv3 ユーザーの間の通信に必要なセキュリティレベル。次のいずれかのオプションを指定します。

noAuthNoPriv。認証も暗号化も必須ではありません。

authNoPriv。認証は必須ですが、暗号化は不要です。

authPriv。認証と暗号化が必要です。注: 認証を指定する場合は、SNMPv3 ユーザーをグループに割り当てるときに暗号化アルゴリズムを指定する必要があります。暗号化も指定する場合は、各グループメンバーに認証アルゴリズムと暗号化アルゴリズムの両方を割り当てる必要があります。これは必須の引数です。指定可能な値: noAuthNoPriv, authNoPriv, authPriv.

readViewName。この SNMPv3 グループにバインドする、設定済みの SNMPv3 ビューの名前。このグループにバインドされた SNMPv3 ユーザは、タイプ INCLUDED としてこの SNMPv3 ビューにバインドされているサブツリーにアクセスできますが、タイプ EXCLUDED のサブツリーにはアクセスできません。Citrix ADC アプライアンスに同じ名前の SNMPv3 ビュー・エントリーが複数ある場合、そのようなエントリはすべて SNMPv3 グループに関連付けられます。これは必須の引数です。最大長: 31

GUI を使用して **SNMP** グループを構成するには

[システム] > [**SNMP**] > [グループ] に移動し、SNMP グループを作成します。

ユーザーの設定

SNMP ユーザは、エージェントが MIB へのアクセスを許可する SNMP マネージャです。各 SNMP ユーザは、SNMP グループに割り当てられます。

エージェントでユーザを設定し、各ユーザをグループに割り当てる必要があります。

コマンドラインインターフェイスを使用してユーザーを構成するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

- `add snmp user <name> -group <string> [-authType ( MD5 | SHA ){ -authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]`
- `show snmp user <name>`

各項目の意味は次のとおりです。

authType は、ユーザーの構成中に使用できる認証オプションです。MD5 と SHA などの 2 種類の認証タイプがあります。

privType は、ユーザの設定中に使用できる暗号化オプションです。暗号化には、鍵サイズ 128 ビットの DES と鍵サイズ 128 ビットの AES などの 2 種類があります。

例

```
1 > add snmp user edocs_user -group edocs_group
2 <!--NeedCopy-->
```

GUI を使用して **SNMP** ユーザを構成するには

[システム] > [**SNMP**] > [ユーザ] に移動し、SNMP ユーザを作成します。

レート制限のための **SNMP** アラームの設定

October 7, 2021

Citrix ADC MPX 10500、12500、15500 のような Citrix ADC あぷアライアンスはレート制限されます。最大スループット（Mbps）およびパケット/秒（PPS）は、アプライアンス用に購入したライセンスによって決まります。レート制限されたプラットフォームでは、スループットと PPS が制限に近づいたとき、および通常の状態に戻ったときに通知を送信するように SNMP トラップを設定できます。

スループットと PPS は 7 秒ごとに監視されます。上限しきい値および標準しきい値を使用してトラップを設定できます。これらの値は、ライセンス制限に対するパーセンテージで表されます。次に、スループットまたは PPS が上限しきい値を超えると、アプライアンスはトラップを生成し、監視対象パラメータが通常のしきい値に達すると 2 番目のトラップを生成します。Citrix ADC は、設定された宛先デバイスにトラップを送信するだけでなく、/var/log/ns.log ファイルにトラップに関連するイベントを、イベントアラートおよびイベントアラートとして記録します。

スループットの制限を超えると、パケットが失われる可能性があります。パケット損失を報告するように SNMP アラームを設定できます。

SNMP アラームとトラップの詳細については、「SNMP v1 および v2 トラップを生成するように Citrix ADC を構成する」を参照してください。

このドキュメントでは、次の詳細について説明します。

- スループットまたは PPS に対する SNMP アラームの設定
- ドロップされたパケットに対する SNMP アラームの設定

スループットまたは **PPS** に対する **SNMP** アラームの設定

全体と PPS の両方を監視するには、アラームをそれぞれ設定し、しきい値 pps 値を Mbps で設定する必要があります。

コマンドラインインターフェイスを使用してスループットレートの **SNMP** アラームを設定するには

コマンドプロンプトで次のコマンドを入力して、SNMP アラームを設定し、しきい値を Mbps で設定し、設定を確認します。

- `set snmp alarm PF-RL-RATE-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( **ENABLED** | **DISABLED** )] [-severity <severity>] [-logging ( **ENABLED** | **DISABLED** )]`
- `show snmp alarm PF-RL-RATE-THRESHOLD`

例

```
1 > set snmp alarm PF-RL-RATE-THRESHOLD -thresholdValue 70 -normalValue
     50
2 <!--NeedCopy-->
```


コマンドラインインターフェイスを使用して **PPS** の **SNMP** アラームを構成するには

コマンドプロンプトで次のコマンドを入力して、PPS の SNMP アラームを構成し、構成を確認します。

- `set snmp alarm PF-RL-PPS-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( **ENABLED** | **DISABLED** )] [-severity <severity>] [-logging ( **ENABLED** | **DISABLED** )]`
- `show snmp alarm PF-RL-PPS-THRESHOLD`

例

```
> set snmp alarm PF-RL-PPS-THRESHOLD -thresholdValue 70 -normalValue 50
<!--NeedCopy-->
```

GUI を使用してスループットまたは **PPS** の **SNMP** アラームを構成するには

1. [システム] > [**SNMP**] > [アラーム] に移動し、[**PF-RL-RATE-THRESHOLD**]（スループットレートの場合）または [**PF-RL-PPS-THRESHOLD** （秒あたりのパケット数）] を選択します。
2. アラームパラメータを設定し、選択した SNMP アラームを有効にします。

ドロップされたパケットに対する **SNMP** アラームの設定

スループット制限を超えた結果としてドロップされたパケットに対してアラームを設定し、PPS 制限を超えた結果としてドロップされたパケットに対してアラームを設定できます。

過剰なスループットのためにドロップされたパケットに対して **SNMP** アラームを設定するには、コマンドラインインターフェイスを使用します

コマンドプロンプトで入力します。

```
set snmp alarm PF-RL-RATE-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-severity <severity>] [-logging ( ENABLED | DISABLED )]
```

コマンドラインインターフェイスを使用して、過剰な **PPS** が原因でドロップされたパケットに対して **SNMP** アラームを設定するには

コマンドプロンプトで入力します。

```
set snmp alarm PF-RL-PPS-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-severity <severity>] [-logging ( ENABLED | DISABLED )]
```


GUI を使用して、ドロップされたパケットの **SNMP** アラームを設定するには

1. [システム] > [**SNMP**] > [アラーム] に移動し、[**PF-RL-RATE-PKTS-DROPPED**]（過剰なスループットのためにドロップされたパケットの場合）または **[PF-RL-PPS-PKTS-DROPPED**]（過剰な PPS のためにドロップされたパケットの場合）を選択します。
2. アラームパラメータを設定し、選択した SNMP アラームを有効にします。

FIPS モードでの SNMP の設定

October 7, 2021

FIPS モードでは、認証とプライバシー (authPriv) オプションを持つ簡易ネットワーク管理プロトコルバージョン 3 (SNMPv3) が必要です。SNMP バージョン 1 およびバージョン 2 では、コミュニティストリングメカニズムを使用して、管理データへのセキュアなアクセスを提供します。コミュニティストリングは、SNMP マネージャと SNMP エージェントの間でクリアテキストとして送信されます。このタイプの通信は安全ではないため、侵入者はネットワーク上の SNMP 情報にアクセスできます。

SNMPv3 プロトコルは、ユーザベースセキュリティモデル（USM）およびビューベースアクセスコントロールモデル（VACM）を使用して、SNMP メッセージングデータへの管理アクセスを認証および制御します。SNMPv3 には、認証なしプライバシーなし（noAuthNoPriv）、認証とプライバシーなし（authNoPriv）、認証とプライバシーなし（authNoPriv）の 3 つのセキュリティレベルがあります。

FIPS モードを有効にして Citrix ADC アプライアンスを再起動すると、アプライアンスから次の SNMP 構成が削除されます。

1. SNMPv1 および SNMPv2 プロトコルのコミュニティ設定。
2. noAuthNoPriv または authNoPriv セキュリティレベルオプションを使用して設定された SNMPv3 グループ。
3. noAuthNoPriv セキュリティレベルオプションを使用して SNMPv1、SNMPv2、または SNMPv3 に対して設定されたトラップ。

アプライアンスを再起動したら、authPriv オプションを使用して SNMPv3 を設定します。SMNP v3 で AuthPriv オプションを構成する方法の詳細については、SNMPV3 のトピックを参照してください。

> 注:
>
> FIPS モードを有効にしてアプライアンスを再起動すると、次の SNMP トラップおよびグループコマンドの実行がブロックされます。

```
1    1.  add snmp community <communityName> <permissions>
2
3    2.  add snmp trap <trapClass> <trapDestination> ... [-version: v1/
        v2]   [-td <positive_integer>] [-destPort <port>] [-
```



```
        communityName <string>] [-srcIP <ip_addr|ipv6_addr>] [-severity
        <severity>] [-allPartitions ( ENABLED | DISABLED )]
4
5       3.  add snmp group <name> <securityLevel : noAuthNoPriv/ authNoPriv
            > -readViewName <string>
6
7       4.  bind snmp trap specific <TrapIp>-userName <v3 user name> -
        securityLevel <noAuthNoPriv/ authNoPriv>
8   <!--NeedCopy-->
```

監査ログ

October 7, 2021

> 重要
>
> SYSLOG または NSLOG 構成は、メンテナンス中またはダウンタイム中にのみ更新することをお勧めします。セッションの作成後に構成を更新した場合、変更は既存のセッションログに適用されません。

監査は、状態または状況の系統的な検査またはレビューです。監査ログ機能を使用すると、さまざまなモジュールによって収集された Citrix ADC の状態とステータス情報をログに記録できます。ログ情報は、カーネルおよびユーザーレベルのデーモンに含めることができます。監査ログには、SYSLOG プロトコル、ネイティブ NSLOG プロトコル、またはその両方を使用できます。

SYSLOG は、ロギング用の標準プロトコルです。これには 2 つのコンポーネントがあります。

- **SYSLOG** 監査モジュール。 Citrix ADC アプライアンスで実行されます。
- **SYSLOG** サーバー。Citrix ADC アプライアンスの基盤となる FreeBSD オペレーティングシステム（OS）またはリモートシステムで実行されます。

SYSLOG は、データ転送にユーザーデータプロトコル (UDP) を使用します。

同様に、ネイティブ NSLOG プロトコルには 2 つのコンポーネントがあります。

- **NSLOG** 監査モジュール。 Citrix ADC アプライアンスで実行されます。
- **NSLOG** サーバー。 Citrix ADC アプライアンスの基盤となる FreeBSDOS またはリモートシステムで実行されます。

NSLOG はデータ転送に TCP を使用します。

SYSLOG サーバーまたは NSLOG サーバーを実行すると、Citrix ADC アプライアンスに接続されます。次に、Citrix ADC アプライアンスは、すべてのログ情報を SYSLOG または NSLOG サーバーに送信し始めます。また、サーバーはログエントリをフィルタリングしてからログファイルに保存します。NSLOG または SYSLOG サーバーは、複数の Citrix ADC アプライアンスからログ情報を受信します。Citrix ADC アプライアンスは、ログ情報を複数の SYSLOG サーバーまたは NSLOG サーバーに送信します。

複数の SYSLOG サーバーが構成されている場合、Citrix ADC アプライアンスは SYSLOG イベントとメッセージを、構成されたすべての外部ログサーバーに送信します。その結果、冗長なメッセージが保存され、システム管理者の監視が困難になります。この問題に対処するために、Citrix ADC アプライアンスは負荷分散アルゴリズムを提供します。この問題に対処するため、Citrix ADC アプライアンスは、外部ログサーバー間で SYSLOG メッセージを負荷分散できる負荷分散アルゴリズムを提供し、メンテナンスとパフォーマンスを向上させます。サポートされている負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小パケット、監査ログハッシュがあります。

> 注
>
> Citrix ADC アプライアンスは、最大 16 KB の監査ログメッセージを外部 SYSLOG サーバーに送信できます。

SYSLOG または NSLOG サーバーが Citrix ADC アプライアンスから収集するログ情報は、メッセージ形式でログファイルに保存されます。通常、これらのメッセージには次の情報が含まれています。

- ログメッセージを生成した Citrix ADC アプライアンスの IP アドレス。
- タイムスタンプ
- メッセージの種類
- 定義済みのログレベル (重大、エラー、通知、警告、情報、デバッグ、アラート、緊急)
- メッセージの情報

監査ログを構成するには、まず Citrix ADC アプライアンスの監査モジュールを構成します。これには、監査ポリシーの作成と、NSLOG サーバまたは SYSLOG サーバ情報の指定が含まれます。次に、Citrix ADC アプライアンスの基盤となる FreeBSD OS またはリモートシステムに SYSLOG サーバーまたは NSLOG サーバーをインストールして設定します。

> 注
>
> SYSLOG は、プログラムメッセージをログに記録するための業界標準であり、さまざまなベンダーがサポートを提供しています。ドキュメントには、SYSLOG サーバーの構成情報は含まれていません。

NSLOG サーバには独自の構成ファイル（auditlog.conf）があります。構成ファイル（auditlog.conf）に追加の変更を加えることにより、NSLOG サーバーシステムのロギングをカスタマイズできます。

監査ログ用の **Citrix ADC** アプライアンスの構成

January 25, 2022

> 警告:
>
> Citrix ADC 12.0 ビルド 56.20 以降では、Citrix ADC 12.0 ビルド 56.20 以降、従来のポリシー式とその使用法は非推奨（使用は推奨されませんが、引き続きサポートされています）。詳細については、「高度なポリシー」を参照してください。

監査ロギングは、管理者がイベント履歴を時系列で表示できるように、さまざまなモジュールのステータス情報を表示します。監査フレームワークの主なコンポーネントは、「監査アクション」、「監査ポリシー」です。「監査アクショ

ン」は監査サーバーの設定情報を記述しますが、「監査ポリシー」はバインドエンティティを「監査アクション」にリンクします。監査ポリシーは、「クラシックポリシーエンジン」(CPE) フレームワークまたはプログレス統合 (PI) フレームワークを使用して、「監査アクション」を「システムグローバルバインドエンティティ」にリンクします。

ただし、監査ログポリシーをグローバルエンティティにバインドする点で、ポリシーフレームワークは互いに異なります。以前は、監査モジュールはクラシック式のみをサポートしていましたが、クラシックポリシー式と高度なポリシー式の両方をサポートするようになりました。現在、Advanced 式は監査ログポリシーをシステムグローバルエンティティにのみバインドできます。

> 注
>
> ポリシーをグローバルエンティティにバインドする場合は、同じ式のシステムグローバルエンティティにポリシーをバインドする必要があります。たとえば、クラシックポリシーを高度なグローバルエンティティにバインドしたり、高度なポリシーをクラシックグローバルエンティティにバインドしたりすることはできません。
>
> また、従来の監査ログポリシーと高度な監査ログポリシーの両方を負荷分散仮想サーバーにバインドすることはできません。

クラシックポリシー式での監査ログポリシーの設定

クラシックポリシーでの監査ロギングの設定は、次の手順で構成されます。

1. 監査ログアクションの設定。監査アクションは、異なるサーバーおよび異なるログレベルに対して設定できます。「監査アクション」は監査サーバーの設定情報を記述しますが、「監査ポリシー」はバインドエンティティを「監査アクション」にリンクします。デフォルトでは、SYSLOG と NSLOG は TCP のみを使用してログ情報をログサーバに転送します。TCP は、完全なデータを転送するために UDP よりも信頼性が高いです。SYSLOG に TCP を使用する場合、Citrix ADC アプライアンスのバッファ制限を設定してログを保存できます。その後、ログは SYSLOG サーバに送信されます。
2. 監査ログポリシーの設定。メッセージを SYSLOG サーバに記録するように SYSLOG ポリシーを設定するか、NSLOG サーバにメッセージを記録する NSLOG ポリシーを設定できます。各ポリシーには、ログに記録されるメッセージを識別するルールと、SYSLOG または NS LOG アクションが含まれます。
3. 監査ログポリシーをグローバルエンティティにバインドします。監査ログポリシーは、システム、VPN、Citrix ADC AAA などのグローバルエンティティにグローバルにバインドする必要があります。これを実行して、すべての Citrix ADC システムイベントのログを有効にすることができます。優先度レベルを定義すると、監査サーバーロギングの評価順序を設定できます。優先度 0 が最高で、最初に評価されます。プライオリティ番号が大きいほど、評価のプライオリティは低くなります。

これらの各手順については、次のセクションで説明します。

監査ログアクションの設定

コマンドラインインターフェイスを使用して、高度なポリシーインフラストラクチャで SYSLOG アクションを構成するには。

> 注
>
> Citrix ADC アプライアンスでは、SYSLOG サーバーの IP アドレスとポートに対して 1 つの SYSLOG アクションのみを構成できます。アプライアンスでは、同じサーバ IP アドレスおよびポートに対して複数の SYSLOG アクションを設定することはできません。

syslog アクションには、syslog サーバへの参照が含まれます。ログに記録する情報を指定し、その情報を記録する方法を説明します。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]`
2 -  show audit syslogAction [<name>]
3
4 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、高度なポリシーインフラストラクチャで NSLOG アクションを構成するには

ns ログアクションには、nslog サーバへの参照が含まれます。ログに記録する情報を指定し、その情報を記録する方法を説明します。

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
1 -  add audit nslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2 -  show audit nslogAction [<name>]
3 <!--NeedCopy-->
```

監査ログポリシーの設定

コマンドラインインターフェイスを使用してクラシックポリシーインフラストラクチャの監査ログポリシーを構成するには

コマンドプロンプトで入力します。

```
1 -  add audit syslogpolicy <name> <-rule> <action>
2 -  add audit nslogpolicy <name> < rule> <action>rm audit nslogpolicy <
     name>show audit nslogpolicy [<name>]set audit nslogpolicy <name>  [-
     rule <expression>] [-action <name>]
```



```
3 <!--NeedCopy-->
```

監査 **syslog** ポリシーを監査 **syslog** グローバルにバインドする

コマンドラインインターフェイスを使用してクラシックポリシーフレームワークで監査ログポリシーをバインドするには

コマンドプロンプトで入力します。

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType>]
```

高度なポリシー式を使用した監査ログポリシーの設定

詳細ポリシーでの監査ロギングの設定は、次の手順で構成されます。

1. 監査ログアクションの設定。監査アクションは、異なるサーバーおよび異なるログレベルに対して設定できます。「監査アクション」は監査サーバーの設定情報を記述しますが、「監査ポリシー」はバインドエンティティを「監査アクション」にリンクします。デフォルトでは、SYSLOG と NSLOG は TCP のみを使用してログ情報をログサーバに転送します。TCP は、完全なデータを転送するために UDP よりも信頼性が高いです。SYSLOG に TCP を使用する場合、Citrix ADC アプライアンスのバッファ制限を設定してログを保存できます。その後、ログは SYSLOG サーバに送信されます。
2. 監査ログポリシーの設定。メッセージを SYSLOG サーバに記録するように SYSLOG ポリシーを設定するか、NSLOG サーバにメッセージを記録する NSLOG ポリシーを設定できます。各ポリシーには、ログに記録されるメッセージを識別するルールと、SYSLOG または NS LOG アクションが含まれます。
3. 監査ログポリシーをグローバルエンティティにバインドします。すべての Citrix ADC システムイベントのログを有効にするには、監査ログポリシーを SYSTEM グローバルエンティティにグローバルにバインドする必要があります。優先度レベルを定義すると、監査サーバーロギングの評価順序を設定できます。優先度 0 が最高で、最初に評価されます。プライオリティ番号が大きいほど、評価のプライオリティは低くなります。

> 注
>
> Citrix ADC アプライアンスは、true にバインドされているすべてのポリシーを評価します。

監査ログアクションの設定

コマンドラインインターフェイスを使用して高度なポリシーインフラストラクチャで syslog アクションを設定するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。


```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]
2 -  show audit syslogAction [<name>]
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、高度なポリシーインフラストラクチャで NSLOG アクションを構成するには

コマンドプロンプトで次のコマンドを入力して、パラメーターを設定し、構成を確認します。

```
1 -  add audit nslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2 -  show audit nslogAction [<name>]
3 <!--NeedCopy-->
```

監査ログポリシーの設定

コマンドラインインターフェイスを使用して syslog 監査アクションを追加するには

コマンドプロンプトで入力します。

```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
         domainResolveRetry <integer>])
2     | -lbVserverName <string>))[-serverPort <port>] -logLevel <logLevel
         >[-dateFormat <dateFormat>]
3      [-logFacility <logFacility>][-tcp ( NONE | ALL )] [-acl ( ENABLED
          | DISABLED )]
4     [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog ( YES |
         NO )]
5      [-appflowExport ( ENABLED | DISABLED )] [-lsn ( ENABLED | DISABLED
           )][-alg ( ENABLED | DISABLED )]
6     [-subscriberLog ( ENABLED | DISABLED )][-transport ( TCP | UDP )]
         [-tcpProfileName <string>][-maxLogDataSizeToHold
7 <!--NeedCopy-->
```

例

```
1     > add audit syslogaction audit-action1 10.102.1.1 -loglevel
         INFORMATIONAL -dateformat MMDDYYYY
```



```
2      > add audit nslogAction nslog-action1 10.102.1.3 -serverport 520 -
           loglevel INFORMATIONAL -dateFormat MMDDYYYY
3      > add audit syslogpolicy syslog-pol1 TRUE audit-action1
4      > add audit nslogPolicy nslog-pol1 TRUE nslog-action1
5      > bind system global nslog-pol1 -priority 20
6  <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して nslog 監査アクションを追加するには

コマンドプロンプトで入力します。

```
1      add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
           domainResolveRetry <integer>])) [-serverPort <port>]       -
           logLevel <logLevel> ... [-dateFormat <dateFormat>][-logFacility
           <logFacility>] [-tcp ( NONE | ALL )][-acl ( ENABLED | DISABLED )
           ] [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog (
           YES | NO )][-appflowExport ( ENABLED | DISABLED )] [-lsn (
           ENABLED | DISABLED )][-alg ( ENABLED | DISABLED )] [-
           subscriberLog ( ENABLED | DISABLED )]`
2  <!--NeedCopy-->
```

監査ログポリシーをグローバルエンティティにバインドする

コマンドラインインターフェイスを使用して高度なポリシーフレームワークで syslog 監査ログポリシーをバインドするには

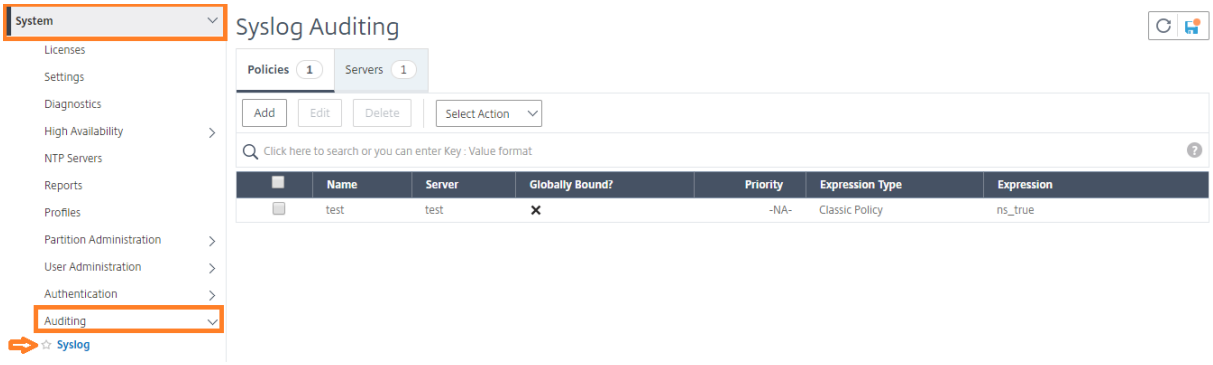
コマンドプロンプトで入力します。

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType
```

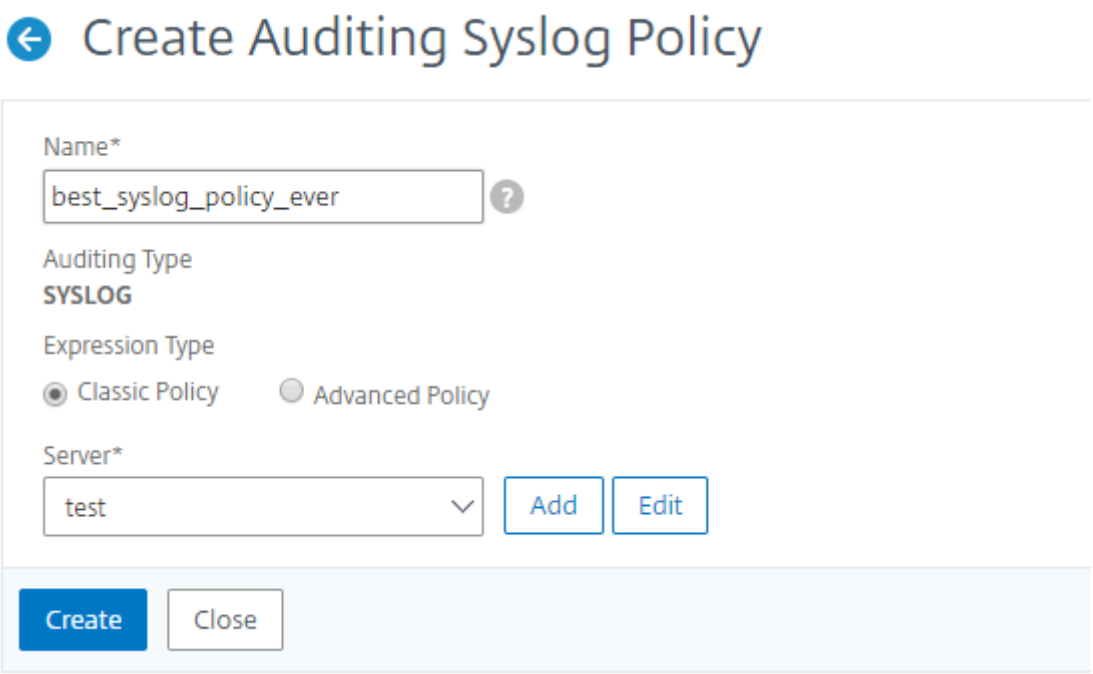
```
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType>]
```

GUI を使用した監査ログポリシーの設定

1. [設定] > [システム] > [監査] > [**Syslog**] に移動します。



1. [サーバ] タブを選択します。
2. [追加] をクリックします。
3. [監査サーバーの作成] ページで、関連するフィールドに入力し、[作成] をクリックします。
4. ポリシーを追加するには、**[**ポリシー**]** タブを選択し、**[**** 追加] をクリックします。**
5. [監査 **Syslog** ポリシーの作成] ページで、関連するフィールドに値を入力し、[作成] をクリックします。



6. ポリシーをグローバルにバインドするには、ドロップダウンリストから [詳細ポリシー**] [**グローバルバインディング] を選択します。**best_syslog_policy_ever** ポリシーを選択します。[**Select**] をクリックします。
7. ドロップダウンリストから、**SYSTEM_GLOBAL** としてバインドポイントを選択し、[バインド] をクリックし、[完了] をクリックします。

ポリシーベースのロギングの設定

書き換えポリシーとレスポンダーポリシーのポリシーベースのロギングを設定できます。監査メッセージは、ポリシーの規則が TRUE と評価されたときに、定義された形式で記録されます。ポリシーベースのロギングを設定するには、デフォルトの構文式を使用して監査メッセージの形式を指定する監査メッセージアクションを設定します。アクションをポリシーに関連付けます。ポリシーは、グローバルにバインドすることも、負荷分散仮想サーバーまたはコンテンツスイッチング仮想サーバーにバインドすることもできます。監査メッセージアクションを使用して、syslog 形式のみ、または syslog 形式と新しい nslog 形式の両方で、さまざまなログレベルでメッセージをログに記録できます。

前提条件

- ユーザー設定可能なログメッセージ (UserDefinedAuditLog) オプションは、定義された形式でログを送信する監査アクションサーバーを設定するときに有効になります。
- 関連する監査ポリシーは、システムグローバルにバインドされます。

監査メッセージアクションの設定

監査メッセージアクションは、syslog 形式のみ、または syslog 形式と新しい ns ログ形式の両方で、さまざまなログレベルでメッセージをログに記録するように設定できます。監査メッセージアクションでは、式を使用して監査メッセージの形式を指定します。

コマンドラインインターフェイスを使用して監査メッセージアクションを作成するには

コマンドプロンプトで入力します。

```
1 add audit messageaction <name> <logLevel> <stringBuilderExpr> [-
     logtoNewnslog (YES|NO)] [-bypassSafetyCheck (YES|NO)]
2 <!--NeedCopy-->
```

```
1 add audit messageaction log-act1 CRITICAL '"Client:"+CLIENT.IP.SRC+"
     accessed "+HTTP.REQ.URL' -bypassSafetyCheck YES
2 <!--NeedCopy-->
```

GUI を使用して監査メッセージアクションを構成するには

[システム] **>** [監査] **>** [メッセージアクション] に移動し、監査メッセージアクションを作成します。

監査メッセージアクションをポリシーにバインドする

監査メッセージアクションを作成したら、それを書き換えポリシーまたはレスポンダーポリシーにバインドする必要があります。ログメッセージアクションをリライトポリシーまたはレスポンダーポリシーにバインドする方法の詳細については、「[書き換えまたはレスポンダー](/ja-jp/citrix-adc/13/appexpert/responder.html)」を参照してください。

NSLOG サーバーのインストールと構成

October 7, 2021

インストール中に、NSLOG サーバーの実行可能ファイル（auditserver）が他のファイルと一緒にインストールされます。監査サーバーの実行可能ファイルには、NSLOG サーバーの実行や停止など、NSLOG サーバーでいくつかのアクションを実行するためのオプションが含まれています。さらに、auditserver 実行可能ファイルを使用して、NSLOG サーバーがログの収集を開始する Citrix ADC アプライアンスの IP アドレスで NSLOG サーバーを構成します。構成設定は、NSLOG サーバー構成ファイル（auditlog.conf）に適用されます。

次に、auditserver 実行可能ファイルを実行して NSLOG サーバーを起動します。NSLOG サーバーの構成は、構成ファイルの設定に基づいています。NSLOG サーバー構成ファイル（auditlog.conf）に追加の変更を加えることにより、NSLOG サーバーシステムのロギングをさらにカスタマイズできます。

> 注意:
>
> NSLOG サーバーパッケージのバージョンは、Citrix ADC のバージョンと同じである必要があります。たとえば、Citrix ADC のバージョンが 10.1 ビルド 125.9 の場合、NSLOG サーバーも同じバージョンである必要があります。

次の表に、NSLOG サーバーがサポートされているオペレーティングシステムを示します。

オペレーティングシステム	ソフトウェア要件	注釈
Windows	Windows XP Professional, Windows Server 2003, Windows 2000/NT, Windows Server 2008, Windows Server 2008 R2	
Linux	Linux 4 以降、SUSE Linux Enterprise バージョン 9.3 以降	
FreeBSD ライセンス	FreeBSD バージョン 6.3 以降	Citrix ADC 10.5 では、FreeBSD 8.4 のみを使用してください。

オペレーティングシステム	ソフトウェア要件	注釈
Mac OS	Mac OS 8.6 以降	Citrix ADC 10.1 以降のリリースではサポートされません。

NSLOG サーバーを実行しているプラットフォームの最小ハードウェア仕様は次のとおりです。

- Processor- Intel x86 ~501 megahertz (MHz)
- RAM-512 メガバイト（MB）
- Controller - SCSI

Linux オペレーティングシステムへの **NSLOG** サーバーのインストール

管理者として Linux システムにログオンします。次の手順を使用して、NSLOG サーバーの実行可能ファイルをシステムにインストールします。

Linux オペレーティングシステムに **NSLOG** サーバーパッケージをインストールするには

1. Linux コマンドプロンプトで、次のコマンドを入力して、NSauditserver.rpm ファイルを一時ディレクトリにコピーします。

   ```
   cp <path_to_cd>/Utilities/auditserver/Linux/NSauditserver.rpm /tmp
   ```

2. 次のコマンドを入力して、NSauditserver.rpm ファイルをインストールします。

   rpm -i NSauditserver.rpm

   このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

Linux オペレーティングシステムで **NSLOG** サーバーパッケージをアンインストールするには

1. コマンドプロンプトで次のコマンドを入力して、監査サーバーのログ機能をアンインストールします。

   ```
   rpm -e NSauditserver
   ```

2. NSauditserver RPM ファイルの詳細については、次のコマンドを使用してください。

   ```
   rpm -qpi \*.rpm
   ```

3. インストールされている監査サーバーファイルを表示するには、次のコマンドを使用します。

   ```
   rpm -qpl *.rpm
   ```

   *.rpm: ファイル名を指定します。

FreeBSD オペレーティングシステムへの NSLOG サーバーのインストール

NSLOG サーバーをインストールする前に、Citrix ADC 製品 CD から NSLOG パッケージをコピーするか、www.citrix.com からダウンロードする必要があります。NSLOG パッケージの名前の形式は次のとおりです。

```
AuditServer_<release number>-<build number>.zip
```

たとえば、次のようになります：`AuditServer_10.5-58.11.zip`

このパッケージには、サポートされているすべてのプラットフォーム（Linux、Windows、および FreeBSD）のファイルが含まれています。FreeBSD オペレーティングシステムで、次の名前形式の NSLOG パッケージをインストールします。

```
audserver_bsd-<release number>-<build number>.tgz
```

たとえば、次のようになります：`audserver_bsd-10.5-58.11.tgz`

www.citrix.com から NSLOG パッケージをダウンロードするには：

1. ウェブブラウザで www.citrix.com にアクセスしてください。
2. メニューバーで、[ログイン] をクリックします。
3. ログイン認証情報を入力し、[ログイン] をクリックします。
4. メニューバーで、[ダウンロード] をクリックします。
5. **[**製品の選択] リストから、[**Citrix ADC]** を選択します。
6. [**Citrix ADC]** ページで、NSLOG パッケージをダウンロードするリリース（たとえば、リリース 10.5）を選択し、[ファームウェア**]** を選択します。
7. [ファームウェア**]** で、NSLOG パッケージをダウンロードするビルド番号の Citrix ADC ファームウェアを選択します。
8. 表示されるページで、下にスクロールし、[**Audit Servers**] を選択し、ダウンロードするパッケージの横にある [**Download File**] をクリックします。

NSLOG サーバーパッケージを FreeBSD オペレーティングシステムにインストールするには

1. NSLOG パッケージ`AuditServer_<release number>-<build number>.zip`をダウンロードしたシステム（`AuditServer_9.3-51.5.zip`など）で、パッケージから`FreeBSD NSLOG server package audserver_bsd-<release number>-<build number>.tgz`（`audserver_bsd-9.3-51.5.tgz`など）を抽出します。
2. FreeBSD NSLOG サーバーパッケージ `audserver_bsd-<release number>-<build number>.tgz` （たとえば、`audserver_bsd-9.3-51.5.tgz`）を FreeBSDOS を実行しているシステム上のディレクトリにコピーします。
3. FreeBSD NSLOG サーバーパッケージがコピーされたディレクトリのコマンドプロンプトで、次のコマンドを実行してパッケージをインストールします。

   ```
   pkg_add audserver_bsd-<release number>-<build number>.tgz
   ```

   例:


```
pkg_add audserver_bsd-9.3-51.5.tgz
<!--NeedCopy-->
```

次のディレクトリが抽出されます。

- <root directory extracted from the FreeBSD NSLOG server package tgz file>Citrix ADCbin（たとえば、/var/auditserver/netscaler/bin）
- <root directory extracted from the FreeBSD NSLOG server package tgz file>netscaler/etc（たとえば、/var/auditserver/netscaler/etc）
- <root directory extracted from the FreeBSD NSLOG server package tgz file>\netscaler\samples（たとえば、/var/auditserver/samples）

4. コマンドプロンプトで次のコマンドを入力して、パッケージがインストールされていることを確認します。

   ```
   pkg_info | grep NSaudserver
   ```

FreeBSD オペレーティングシステムで **NSLOG** サーバーパッケージをアンインストールするには

コマンドプロンプトで、次のコマンドを実行します:

```
pkg_delete NSaudserver
```

Windows オペレーティングシステムへの **NSLOG** サーバーファイルのインストール

NSLOG サーバーをインストールする前に、Citrix ADC 製品 CD から NSLOG パッケージをコピーするか、www.citrix.com からダウンロードする必要があります。NSLOG パッケージの名前の形式は次のとおりです: AuditServer _<release number>-<build number>.zip (AuditServer_9.3-51.5.zip など)。このパッケージには、サポートされているすべてのプラットフォーム用の NSLOG インストールパッケージが含まれています。

www.Citrix.com から **NSLOG** パッケージをダウンロードするには

1. ウェブブラウザで www.citrix.com にアクセスしてください。
2. メニューバーで、[ログイン] をクリックします。
3. ログイン資格情報を入力し、[ログイン] をクリックします。
4. メニューバーで、[ダウンロード] をクリックします。
5. 適切なリリース番号とビルドを提供するページを検索して見つけます。
6. そのページの [Audit Servers] で、[Download] をクリックして、AuditServer_<release number>-<build number>.zip形式の NSLOG パッケージをローカルシステム (AuditServer_9.3-51.5.zipなど) にダウンロードします。

Windows オペレーティングシステムに **NSLOG** サーバーをインストールするには

1. NSLOG パッケージをダウンロードしたシステムAuditServer_<release number>-<build number>.zip（AuditServer_9.3-51.5.zipなど）で、パッケージからaudserver_win-<release number>-<build number>.zip（audserver_win-9.3-51.5.zipなど）を抽出します。
2. 抽出したファイルaudserver_<release number>-<build number>.zip（audserver_win-9.3-51.5.zipなど）を、NSLOG サーバをインストールする Windows システムにコピーします。
3. audserver_<release number>-<build number>.zipファイルを解凍します（例: audserver_win-9.3-51.5.zip)。
4. 次のディレクトリが抽出されます。
   a) <root directory extracted from the Windows NSLOG server **package** zip file>\bin（たとえば、C:\audserver_win-9.3-51.5\bin）
   b) <root directory extracted from the Windows NSLOG server **package** zip file>\etc（たとえば、C:\audserver_win-9.3-51.5\etc）
   c) <root directory extracted from the Windows NSLOG server **package** zip file>\samples（たとえば、C:\audserver_win-9.3-51.5\samples）
5. コマンドプロンプトで、<root directory extracted from the Windows NSLOG server **package** zip file>\bin pathから次のコマンドを実行します

   ```
   audserver -install -f <directorypath>\auditlog.conf
   ```

   <directorypath>: 構成ファイルへのパスを指定します（auditlog.conf）。デフォルトでは、log.conf は \<root directory extracted from Windows NSLOG server **package** zip file\>\samples ディレクトリの下にあります。ただし、auditlog.conf を目的のディレクトリにコピーできます。

Windows オペレーティングシステムで **NSLOG** サーバーをアンインストールするには

コマンドプロンプトで、<root directory extracted from Windows NSLOG server **package** zip file>\bin パスから次を実行します。

```
audserver -remove
```

NSLOG サーバーコマンドオプション

NSLOG サーバコマンドの詳細については、監査サーバオプションを参照してください。

監査サーバーの実行可能ファイルが存在するディレクトリから audserver コマンドを実行します。

- Windows の場合:\ns\bin
- Solaris および Linux の場合: \usr\local\netscaler\bin

監査サーバー構成ファイルは、次のディレクトリーにあります。

- Windows の場合:\ns\etc
- Linux の場合：\usr\local\netscaler\etc

監査サーバーの実行可能ファイルは、Linux および FreeBSD の./auditserverとして起動されます。

NSLOG サーバーへの Citrix ADC アプライアンスの IP アドレスの追加

構成ファイル（
auditlog.conf）に、イベントをログに記録する必要がある Citrix ADC アプライアンスの IP アドレスを追加します。

Citrix ADC アプライアンスの IP アドレスを追加するには

コマンドプロンプトで、次のコマンドを入力します。

```
audserver -addns -f <directorypath\>\auditlog.conf
```

<directorypath>: 構成ファイル（auditlog.conf）へのパスを指定します。

次のパラメータの情報を入力するように求められます。

NSIP：Citrix ADC アプライアンスの IP アドレスを指定します（例：10.102.29.1）。

ユーザー ID：ユーザー名を指定します（例：nsroot）。

パスワード：パスワードを指定します（例：nsroot）。

複数の Citrix ADC IP アドレス（NSIP）を追加し、後ですべての Citrix ADC アプライアンスイベントの詳細をログに記録したくない場合は、auditlog.conf ファイルの最後にある NSIP ステートメントを削除して NSIP を手動で削除できます。高可用性（HA）のセットアップ中に、audserver コマンドを使用して、プライマリとセカンダリの両方の Citrix ADC IP アドレスを uditlog.conf に追加する必要があります。IP アドレスを追加する前に、システムにユーザー名とパスワードが存在することを確認してください。

NSLOG サーバ構成ファイルの確認

構成ファイル（audit log.conf）で構文が正しいかどうかを確認して、ロギングを開始して正しく機能できるようにします。

構成を確認するには、コマンドプロンプトで、次のコマンドを入力します。

```
audserver -verify -f <directorypath>\auditlog.conf
```

<directorypath>: Specifies the path to the configuration file (audit log.conf)。

NSLOG サーバの実行

October 7, 2021

監査サーバーのログを開始するには

コマンドプロンプトで次のコマンドを入力します。

audserver -start -f <directorypath>\auditlog.conf

<directorypath>: 構成ファイル (監査 log.conf) へのパスを指定します。

FreeBSD または Linux でバックグラウンドプロセスとして起動する監査サーバーのロギングを停止するには

次のコマンドを入力します。

```
audserver -stop
```

Windows でサービスとして起動する監査サーバーのログを停止するには

次のコマンドを入力します。

```
audserver -stopservice
```

NSLOG サーバでのロギングのカスタマイズ

October 7, 2021

NSLOG サーバ構成ファイル（log.conf）にさらに変更を加えることで、NSLOG サーバでのロギングをカスタマイズできます。テキスト・エディタを使用して、サーバ・システム上の log.conf 構成ファイルを変更します。

ロギングをカスタマイズするには、設定ファイルを使用してフィルタとログのプロパティを定義します。

- ログフィルタ。Citrix ADC アプライアンスまたは一連の Citrix ADC アプライアンスからのログ情報をフィルタリングします。
- ログのプロパティ。各フィルタには、関連する一連のログプロパティがあります。ログのプロパティは、フィルタリングされたログ情報の格納方法を定義します。

このドキュメントでは、次の詳細について説明します。

- フィルタの作成
- ログプロパティの指定

フィルタの作成

構成ファイル（audit log.conf）にあるデフォルトのフィルター定義を使用するか、フィルターを変更するか、新しいフィルターを作成することができます。複数のログフィルタを作成できます。

> 注:
> 統合ロギングでは、フィルタ定義がないログトランザクションが発生した場合、デフォルトのフィルタが使用されます (有効になっている場合)。すべての Citrix ADC アプライアンスの統合ログを構成できる唯一の方法は、デフォルトのフィルタを定義することです。

フィルタを作成するには

コマンドプロンプトで、構成ファイル（auditlog.conf）に次のコマンドを入力します。

```
filter <filterName> [IP <ip>] [NETMASK <mask>] ON | OFF]
<!--NeedCopy-->
```

filterName：フィルターの名前を指定します（最大 64 文字の英数字）。

ip：IP アドレスを指定します。

マスク：サブネットで使用するサブネットマスクを指定します。

フィルターがトランザクションをログに記録できるようにするには ON を指定し、フィルターを無効にするには OFF を指定します。引数が指定されていない場合、フィルターはオンです。

例:

```
filter F1 IP 192.168.100.151 ON
<!--NeedCopy-->
```

フィルタ F2 を IP アドレス 192.250.100.1～192.250.100.254 に適用するには:

```
filter F2 IP 192.250.100.0 NETMASK 255.255.255.0 ON
<!--NeedCopy-->
```

filterName は、IP アドレス、または IP アドレスとネットマスクの組み合わせなどの他のオプションのパラメーターを使用してフィルターを定義する場合に必須のパラメーターです。

ログ・プロパティの指定

フィルタに関連付けられたログプロパティは、フィルタに存在するすべてのログエントリに適用されます。次の例に示すように、ログプロパティの定義はキーワード BEGIN で始まり、END で終わります。

```
BEGIN <filtername>
    logFilenameFormat ...
    logDirectory ...
    logInterval ...
    logFileSizeLimit ....
END
<!--NeedCopy-->
```

定義内のエントリには、次のものを含めることができます。

- **LogFilenameFormat** は、ログファイルのファイル名形式を指定します。ファイル名には、次のタイプがあります。
  - 静的：絶対パスとファイル名を指定する定数文字列。
  - 動的：次の形式指定子を含む式：
    - Date (%{format}t)
    - NSIP でファイル名を作成します

  例:

  ```
  LogFileNameFormat Ex%`{
  `%m%d%y }
  t.log
  <!--NeedCopy-->
  ```

  これにより、最初のファイル名が Exmmddyy.log として作成されます。新しいファイルの名前は、Exmmddyy.log.0、Exmmddyy.log.1 などです。次の例では、ファイルサイズが 100MB に達すると新しいファイルが作成されます。

  例:

  ```
  LogInterval size
  LogFileSize 100
  LogFileNameFormat Ex%`{
  `%m%d%y }
  t
  <!--NeedCopy-->
  ```


> 注意
>
> LogFilenameFormat パラメーターで指定された日付形式%t は、そのフィルタのログ間隔プロパティを上書きします。指定したログファイルのサイズに達したときではなく、毎日新しいファイルが作成されないようにするには、LogFilenameFormat パラメーターで%t を使用しないでください。

- **logDirectory** は、ログファイルのディレクトリ名の形式を指定します。ファイルの名前は、次のいずれかになります。
  - 静的：絶対パスとファイル名を指定する定数文字列。
  - [動的]: 次の形式を含む式を指定します。
    - Date (%{format}t)
    - NSIP でディレクトリを作成します

  ディレクトリ区切り文字は、オペレーティングシステムによって異なります。Windows では、ディレクトリ区切り文字を使用します。

  例**:**

```
LogDirectory dir1\dir2\dir3
<!--NeedCopy-->
```

  他のオペレーティングシステム（Linux、FreeBsd など）では、ディレクトリ区切り文字を使用します。

- **LogInterval** は、新しいログファイルが作成される間隔を指定します。次のいずれかの値を使用します。
  - 毎時：ファイルは毎時作成されます。デフォルト値です。
  - 毎日：ファイルは真夜中に作成されます。
  - 毎週: 毎週日曜日の深夜にファイルが作成されます。
  - 毎月: ファイルは、月の初日の深夜に作成されます。
  - なし：監査サーバーのログ記録が開始されると、ファイルは 1 回だけ作成されます。
  - サイズ：ファイルは、ログファイルのサイズ制限に達した場合にのみ作成されます。

  例:

```
LogInterval Hourly
<!--NeedCopy-->
```

  - **LogFileSizeLimit** は、ログファイルの最大サイズ（MB 単位）を指定します。制限に達すると、新しいファイルが作成されます。

> 注
>
> サイズを値として割り当てることにより、loginterval プロパティをオーバーライドできます。

デフォルトのログファイルサイズ制限は 10 MB です。

例:

```
1  LogFileSizeLimit 35
2  <!--NeedCopy-->
```

TCP を介したシステムログ

October 7, 2021

Syslog は、イベント通知メッセージを送信するための標準です。これらのメッセージは、ローカルに保存することも、外部ログサーバに保存することもできます。Syslog を使用すると、ネットワーク管理者はログメッセージを統合し、収集したデータから洞察を得ることができます。

Syslog は元々 UDP を介して動作するように設計されており、UDP は、パケット損失を最小限に抑えて同じネットワーク内で大量のデータを送信できます。ただし、通信事業者は、ネットワーク間で信頼性の高い順序付けされたデータ転送を必要とするため、TCP 経由で syslog データを送信することを好みます。たとえば、電話会社はユーザーアクティビティを追跡し、TCP はネットワーク障害が発生した場合に再送信を提供します。

TCP を介した Syslog の仕組み

TCP を介した syslog の動作を理解するには、次の 2 つの仮説的なケースを検討してください。

ネットワーク管理者の Sam は、外部の syslog サーバで重要なイベントをログに記録したいと考えています。

ISP である XYZTelecom は、政府の規制に準拠するために、大量のデータを送信して syslog サーバーに保存する必要があります。

どちらの場合も、ログメッセージは信頼できるチャネル経由で送信され、外部 syslog サーバに安全に保存される必要があります。UDP とは異なり、TCP は接続を確立し、メッセージを安全に送信し、ネットワーク障害により破損または失われたデータを送信者から受信者に再送信します。

Citrix ADC アプライアンスは、ログメッセージを UDP 経由でローカル Syslog デーモンに送信し、ログメッセージを TCP または UDP 経由で外部 Syslog サーバーに送信します。

Syslog に対する SNIP のサポート

監査ログモジュールが syslog メッセージを生成するとき、外部の syslog サーバーにメッセージを送信するための送信元アドレスとして Citrix ADC IP（NSIP）アドレスを使用します。SNIP を送信元アドレスとして設定するには、SNIP を netProfile オプションの一部にして、netProfile を syslog アクションにバインドする必要があります。

> 注
>
> TCP は、SNIP を使用して監視プローブを送信して接続を確認してから、NSIP を介してログを送信します。したがって、syslog サーバーは SNIP 経由で到達可能である必要があります。ネットプロファイルを使用して、SNIP を介してすべての TCPSyslog トラフィックを完全にリダイレクトできます。

SNIP アドレスの使用は、内部ロギングではサポートされていません。

完全修飾ドメイン名監査ログのサポート

以前は、audit-log モジュールは、ログメッセージの送信先となる外部 syslog サーバの宛先 IP アドレスで設定されていました。現在、監査ログサーバーは、宛先 IP アドレスの代わりに完全修飾ドメイン名（FQDN）を使用します。FQDN 設定は、syslog サーバの設定済みドメイン名を、audit-log モジュールからログメッセージを送信するための対応する宛先 IP アドレスに解決します。ネームサーバーは、ドメイン名を解決し、ドメインベースのサービスの問題を回避するように適切に構成する必要があります。

> 注
>
> FQDN を構成する場合、syslog アクションまたは nslog アクションでの同じ Citrix ADC アプライアンスのサーバードメイン名構成はサポートされていません。

コマンドラインインターフェイスを使用した **SyslogoverTCP** の設定

コマンドラインインターフェイスを使用して TCP 経由で Syslog メッセージを送信するように Citrix ADC アプライアンスを構成するには

コマンドプロンプトで入力します。

```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
          domainResolveRetry <integer>]) | -lbVserverName<string>))[-
          serverPort <port>] -logLevel <logLevel>[-dateFormat <dateFormat
          >] [-logFacility <logFacility>]   [-tcp ( NONE | ALL )] [-acl (
          ENABLED | DISABLED )][-timeZone ( GMT_TIME | LOCAL_TIME )][-
          userDefinedAuditlog ( YES | NO )][-appflowExport ( ENABLED |
          DISABLED )] [-lsn ( ENABLED | DISABLED )][-alg ( ENABLED |
          DISABLED )] [-subscriberLog ( ENABLED | DISABLED )][-transport (
           TCP | UDP )] [-tcpProfileName <string>][-maxLogDataSizeToHold <
          positive_integer>][-dns ( ENABLED | DISABLED )] [-netProfile <
          string>]
2 <!--NeedCopy-->
```



```
    add audit syslogaction audit-action1 10.102.1.1 -loglevel
       INFORMATIONAL -dateformat MMDDYYYY -transport TCP
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して、ネットプロファイルオプションに **SNIPIP** アドレスを追加する

コマンドラインインターフェイスを使用してネットプロファイルに SNIPIP アドレスを追加するには

コマンドプロンプトで入力します。

```
    add netProfile <name> [-td <positive_integer>] [-srcIP <string>][-
       srcippersistency ( ENABLED | DISABLED )][-overrideLsn ( ENABLED
       | DISABLED )]add syslogaction <name> <serverIP>  - loglevel all
        - netprofile net1
<!--NeedCopy-->
```

```
    add netprofile net1  - srcip 10.102.147.204`
<!--NeedCopy-->
```

ここで、srcIP は SNIP です。

コマンドラインインターフェイスを使用して **syslog** アクションにネットプロファイルを追加する

コマンドラインインターフェイスを使用して syslog アクションに netProfile オプションを追加するには

コマンドプロンプトで入力します。

```
     add audit syslogaction <name> (<serverIP> | -lbVserverName <string
        >)  -logLevel <logLevel>
    -netProfile <string> …

<!--NeedCopy-->
```

```
    add syslogaction sys_act1 10.102.147.36  - loglevel all  - netprofile
        net1
<!--NeedCopy-->
```


ここで、-netprofile は、構成されたネットプロファイルの名前を指定します。SNIP アドレスは netProfile の一部として設定され、この netProfile オプションは syslog アクションにバインドされます。

> 注
>
> LB 仮想サーバー名が syslog アクションで構成されている場合は、常に netProfile オプションを SYSLOGUDP または SYSLOGTCP 負荷分散仮想サーバーにバインドされている SYSLOGUDP または SYSLOGTCP サービスにバインドする必要があります。

コマンドラインインターフェイスを使用した **FQDN** サポートの構成

コマンドラインインターフェイスを使用して Syslog アクションにサーバードメイン名を追加するには

コマンドプロンプトで入力します。

```
1 add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
     domainResolveRetry <integer>]) | -lbVserverName <string>)) -logLevel
      <logLevel> ...
2     set audit syslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>]-
        serverDomainName <string>] [-lbVserverName <string>]-
        domainResolveRetry <integer>] [-domainResolveNow]
3 <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して Nslog アクションにサーバードメイン名を追加する。

コマンドプロンプトで入力します。

```
1     add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
        domainResolveRetry <integer>]))  -logLevel <logLevel> ...
2     set audit nslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>][-
        serverDomainName <string>] [-domainResolveRetry <integer>][-
        domainResolveNow]
3 <!--NeedCopy-->
```

ここで serverDomainName。ログサーバーのドメイン名。serverIP / lbVserverName と相互に排他的です。

ドメイン解決再試行整数。DNS 解決が失敗した後、DNS クエリを送信してドメイン名を解決するまでの Citrix ADC アプライアンスが待機する時間（秒単位）。

今すぐドメイン解決。サーバーのドメイン名を解決するために DNS クエリをすぐに送信する必要がある場合に含まれます。

GUI を使用した **TCP** 経由の **Syslog** の設定

GUI を使用して TCP 経由で Syslog メッセージを送信するように Citrix ADC アプライアンスを構成するには

1. [システム] > [監査] > [**Syslog**] に移動し、[サーバ] タブを選択します。
2. [追加] をクリックし、[**TCP**] として [トランスポートタイプ] を選択します。

GUI を使用した **SNIP** サポート用のネットプロファイルの構成

GUI を使用して SNIP サポートのネットプロファイルを構成するには

1. **System > Auditing > Syslog** に移動して **Servers** タブを選択します。
2. [追加**]** をクリックして、リストからネットプロファイルを選択します。

GUI を使用した **FQDN** の構成

GUI を使用して FQDN を構成するには

1. [システム] > [監査] > [**Syslog**] に移動し、[サーバ] タブを選択します。
2.「追加」をクリックし、リストから「サーバーの種類」と「サーバーのドメイン名」を選択します。

SYSLOG サーバーの負荷分散

September 2, 2022

Citrix ADC アプライアンスは、構成されたすべての外部ログサーバーに SYSLOG イベントとメッセージを送信します。その結果、冗長なメッセージが保存され、システム管理者の監視が困難になります。この問題に対処するために、Citrix ADC アプライアンスは、メンテナンスとパフォーマンスを向上させるために外部ログサーバー間で SYSLOG メッセージを負荷分散できる負荷分散アルゴリズムを提供します。サポートされる負荷分散アルゴリズムには、ラウンドロビン、最小帯域幅、カスタムロード、最小接続、最小パケット、および AuditLogHash が含まれます。

コマンドラインインターフェイスを使用した SYSLOG サーバーの負荷分散

コマンドプロンプトで入力します。

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   ```
   add service <name>(<IP> | <serverName>)<serviceType (SYSLOGTCP |
   SYSLOGUDP)> <port>
   ```

2. 負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定し、負荷分散方式を AUDITLOGHASH として指定します。

   ```
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
   <AUDITLOGHASH>]
   ```


3. サービスを負荷分散仮想サーバーにバインドします。

   ```
   Bind lb vserver <name> <serviceName>
   ```

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。

   ```
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
   ```

5. ルールとアクションを指定して SYSLOG ポリシーを追加します。

   ```
   add syslogpolicy <name> <rule> <action>
   ```

6. ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

   ```
   bind system global <policyName>
   ```

GUI を使用した SYSLOG サーバーの負荷分散

1. サービスを追加し、サービスタイプを SYSLOGTCP または SYSLOGUDP として指定します。

   [トラフィック管理] > [サービス] に移動し、[追加] をクリックして、プロトコルとして [**SYLOGTCP**] または [**SYSLOGUDP**] を選択します。

2. 負荷分散仮想サーバーを追加し、サービスタイプを SYSLOGTCP または SYSLOGTCP、負荷分散方法を AUDITLOGHASH として指定します。

   [トラフィック管理] > [仮想サーバー] に移動し、[追加] をクリックして、プロトコルとして [**SYLOGTCP**] または [**SYSLOGUDP**] を選択します。

3. サービスを負荷分散仮想サーバーにバインドします。

   [トラフィック管理] > [仮想サーバー] に移動し、仮想サーバーを選択し、[負荷分散方法**]** で **[**AUDIT-LOGHASH**] を選択します。**

4. SYSLOG アクションを追加し、サービスタイプとして SYSLOGTCP または SYSLOGUDP を持つ負荷分散サーバー名を指定します。

   [システム] > [監査] に移動し、[サーバー] をクリックし、[サーバー] で [**LB Vserver**] オプションを選択してサーバーを追加します。

5. ルールとアクションを指定して SYSLOG ポリシーを追加します。

   [システム] > [**Syslog**] に移動し、[ポリシー] をクリックして、SYSLOG ポリシーを追加します。

6. ポリシーを有効にするには、SYSLOG ポリシーをシステムグローバルにバインドします。

   [システム] > [**Syslog**] に移動し、SYSLOG ポリシーを選択して [アクション] をクリックし、[グローバルバインディング] をクリックして、ポリシーをシステムグローバルにバインドします。

例:

次の構成では、AUDITLOGHASH を負荷分散方法として使用して、外部ログサーバー間で SYSLOG メッセージの負荷分散を指定します。AUDITLOGHASH メソッドは、監査エージェントからの入力ハッシュ値に基づいてトラフィックの負荷を分散します。エージェントは、Citrix ADC アプライアンスで監査ログを生成するモジュールです。たとえば、エージェント LSN がクライアント IP アドレスに基づいて監査ログをロードバランシングする場合、LSN モジュールは clientIP に基づいてハッシュ値を生成し、そのハッシュ値を監査ログモジュールに渡します。監査ログモジュールは、同じハッシュ値を持つ監査ログメッセージを外部 syslog サーバーに送信します。

Citrix ADC アプライアンスは、サービス、service1、service2、およびサービス 3 の間で負荷分散される SYSLOG イベントとメッセージを生成します。

```
add service service1 192.0.2.10 SYSLOGUDP 514
add service service2 192.0.2.11 SYSLOGUDP 514
add service service3 192.0.2.11 SYSLOGUDP 514
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
bind lb vserver lbvserver1 service1
bind lb vserver lbvserver1 service2
bind lb vserver lbvserver1 service3
add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
add syslogpolicy syspol1 ns_true sysaction1
bind system global syspol1
<!--NeedCopy-->
```

制限事項:

- Citrix ADC アプライアンスは、ログサーバー間で SYSLOG メッセージを負荷分散する外部負荷分散仮想サーバーをサポートしていません。

ログプロパティのデフォルト設定

October 7, 2021

以下は、ログプロパティのデフォルト設定を使用したデフォルトフィルターの例です。

```
begin default
 logInterval Hourly
 logFileSizeLimit 10
 logFilenameFormat  auditlog%`{
 `%y%m%d }
 t.log
 end default
```



```
<!--NeedCopy-->
```

以下は、デフォルトのフィルターを定義する 2 つの例です。

例 **1**:

```
Filter f1 IP 192.168.10.1
<!--NeedCopy-->
```

これにより、有効なログのデフォルト値を使用して NSI192.168.10.1 のログファイルが作成されます。

例 **2**:

```
Filter f1 IP 192.168.10.1
begin f1
    logFilenameFormat logfiles.log
end f1
<!--NeedCopy-->
```

これにより、NSIP192.168.10.1 のログファイルが作成されます。ログファイル名の形式が指定されているため、他のログプロパティのデフォルト値が有効になります。

Sample configuration file (audit.conf)

October 7, 2021

次に、設定ファイルの例を示します。

```
################################
# This is the Auditserver configuration file
# Only the default filter is active
# Remove leading # to activate other filters
################################
MYIP <NSAuditserverIP>
MYPORT 3023
#   Filter filter_nsip  IP <Specify the Citrix ADC IP address to filter
    on > ON
#   begin filter_nsip
#       logInterval         Hourly
```



```
#       logFileSizeLimit    10
#       logDirectory         logdir\%A\
#       logFilenameFormat   nsip%\\{
 \\%d%m%Y }
 t.log
#   end filter_nsip
Filter default
begin default
    logInterval      Hourly
    logFileSizeLimit    10
    logFilenameFormat   auditlog%\{
 \%y%m%d }
 t.log
end default
<!--NeedCopy-->
```

Web サーバーロギング

October 7, 2021

Web サーバーのログ機能を使用して、HTTP および HTTPS 要求のログをクライアントシステムに送信し、格納および取得できます。この機能には、次の 2 つのコンポーネントがあります。

- Citrix ADC 上で動作する Web ログサーバー。
- クライアントシステムで実行される Citrix ADC Web ログ (NSWL) クライアント。

Citrix ADC Web ログ (NSWL) クライアントを実行すると、次のようになります。

1. これは、Citrix ADC に接続します。
2. Citrix ADC は、クライアントに送信する前に、HTTP リクエストと HTTPS リクエストのログエントリをバッファリングします。
3. クライアントは、エントリを格納する前にエントリをフィルタリングできます。

Web サーバーのログ記録を構成するには、まず Citrix ADC で Web ログ記録機能を有効にし、ログエントリを一時的に保存するためのバッファーのサイズを設定します。次に、クライアントシステムに NSWL をインストールします。次に、Citrix ADC の IP アドレス（NSIP）を NSWL 構成ファイルに追加します。これで、NSWL クライアントを起動してログを開始する準備ができました。NSWL 構成ファイル (log.conf) にさらに変更を加えることで、Web サーバーのログをカスタマイズできます。

Web サーバーのログ記録用に Citrix ADC を構成する

October 7, 2021

Citrix ADC を Web サーバーログ用に構成するには、Web サーバーログ機能のみを有効にする必要があります。オプションで、次の設定を実行できます。

- Citrix ADC Web Logging（NSWL）クライアントに送信する前に、ログに記録された情報を格納するバッファーのサイズ（デフォルトサイズは 16 MB）を変更します。
- NSWL クライアントにエクスポートするカスタム HTTP ヘッダーを指定します。最大 2 つの HTTP 要求と 2 つの HTTP 応答ヘッダー名を設定できます。

コマンドラインインターフェイスを使用して Web サーバーのログを設定するには

コマンドプロンプトで、次の操作を実行します。

- Web サーバロギング機能を有効にします。

  ```
  enable ns feature WL
  ```

- [（オプション）] ログ情報を格納するためのバッファサイズを変更します。

  ```
  set ns weblogparam -bufferSizeMB <size>
  ```

  > 注:
  >
  > 変更をアクティブにするには、Web サーバーのログ機能を無効にしてから再度有効にする必要があります。

- [（オプション）] エクスポートするカスタム HTTP ヘッダー名を指定します。

  ```
  set ns weblogparam [-customReqHdrs <string> ...] [-customRspHdrs <
  string> ...]
  ```

```
1     > enable ns feature WL
2     Done
3     > set ns weblogparam -bufferSizeMB 60
4     Done
5     > show ns weblogparam
6         Web Logging parameters:
7         Log buffer size:  60MB
8         Custom HTTP request headers: (none)
9         Custom HTTP response headers: (none)
10    Done
11    > set ns weblogparam -customReqHdrs req1 req2 -customRspHdrs res1
         res2
```



```
    Done
    > show ns weblogparam
        Web Logging parameters:
        Log buffer size:  60MB
        Custom HTTP request headers: req1, req2
        Custom HTTP response headers: res1, res2
    Done
<!--NeedCopy-->
```

GUI を使用して **Web** サーバーのログを設定するには

1. [システム] **>** [設定] に移動し、次の操作を実行します。
    a) Web サーバーのログ機能を有効にするには、[詳細機能の変更] をクリックし、[**Web** ログ] を選択します。
    b) バッファサイズを変更するには、[グローバルシステム設定の変更] をクリックし、[**Web** ログ] でバッファサイズを入力します。
    c) エクスポートするカスタム HTTP ヘッダーを指定するには、[グローバルシステム設定の変更] をクリックし、[**Web** ログ] でヘッダー値を指定します。

Citrix ADC Web ログ (NSWL) クライアントのインストール

October 7, 2021

NSWL をインストールすると、クライアント実行可能ファイル（NSWL）が他のファイルと一緒にインストールされます。NSWL 実行可能ファイルには、使用できるオプションのリストが表示されます。詳細については、「NSWL クライアントの設定」を参照してください。

> 注意:
>
> NSWL クライアントのバージョンは Citrix ADC と同じである必要があります。たとえば、Citrix ADC のバージョンが 10.1 ビルド 125.9 の場合、NSWL クライアントも同じバージョンである必要があります。また、Web ロギング（NSWL）クライアントは、32 ビットと 64 ビットの両方のサーバーマシンで動作します。ダウンロードページには、32 ビットのウェブログクライアントしかありません。64 ビットのウェブログクライアントはリクエストに応じて利用できます。詳細については、Citrix サポートに問い合わせることをお勧めします。

次の表に、NSWL クライアントをインストールできるオペレーティング・システムを示します。

オペレーティングシステム	バージョン	ハードウェア要件	注釈
Windows	Windows サーバー 2016 または later	Processor - x86/amd64 CPU（1 GHz 以上）、RAM-4 GB（またはそれ以上）	
macOS	macOS 8.6 または later	Not Citrix ADC10.1 以降のリリースでサポートされています。	
Linux	Ubuntu, 2016 年にリリースされた SUSELinux、CentOS、Red Hat EnterpriseLinux または later	Processor - x86/amd64 CPU（1 GHz 以上）、RAM-4 GB（またはそれ以上）	
Solaris	Solaris Sun OS5.6 または later	Processor -UltraSPARC-IIi 400 MHz、RAM-512 MB、コントローラー- SCSI	Not Citrix ADC10.5 以降のリリースでサポートされています。
FreeBSD	FreeBSD 6.3 または later	Processor - x86/amd64 CPU（1 GHz 以上）、RAM-4 GB（または higher)	For Citrix ADC 10.5、FreeBSD8.4 のみを使用してください。
AIX	AIX 6.1	-	Not Citrix ADC10.5 以降のリリースでサポートされています。

CPU の制限が原因で NSWL クライアントシステムがログトランザクションを処理できない場合、Web ログバッファがオーバーランし、ログプロセスが再び開始されます。

> 注意
>
> ロギングを再開すると、ログトランザクションが失われる可能性があります。

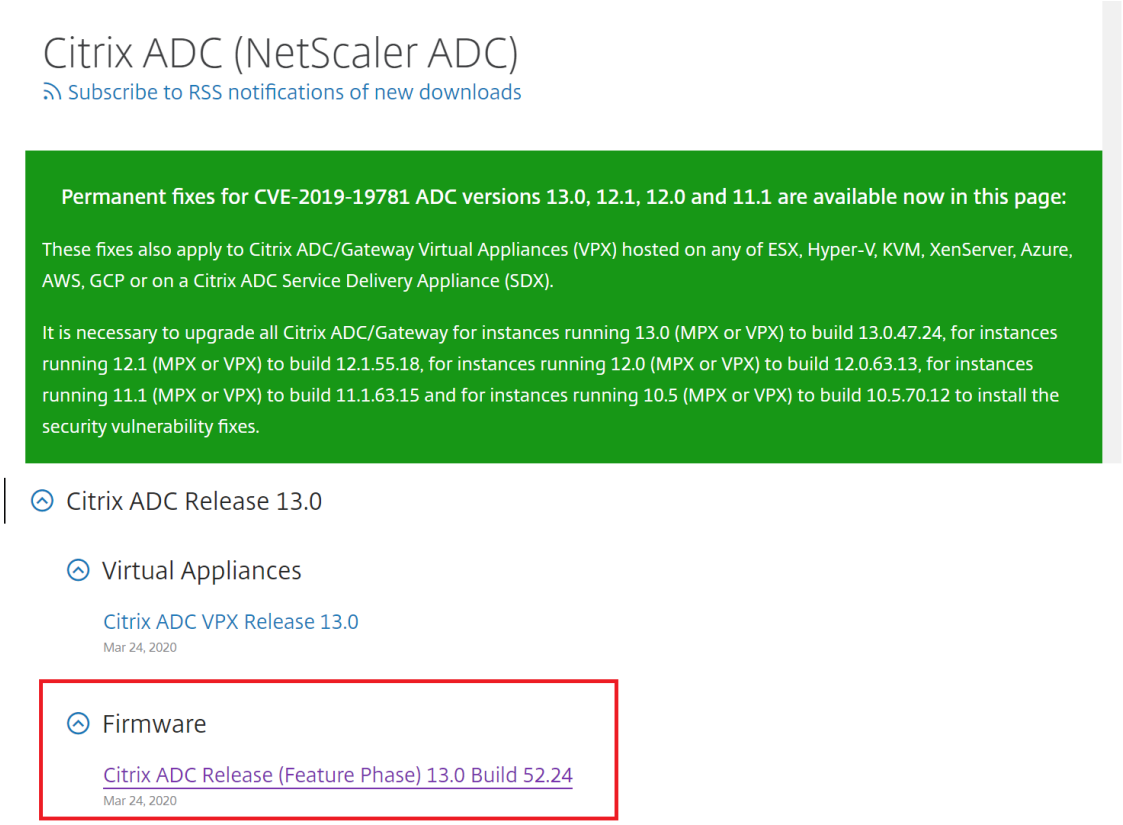
CPU の制限によって生じた NSWL クライアントシステムのボトルネックを一時的に解決するには、Citrix ADC アプライアンスで Web サーバーのログバッファサイズを調整します。この問題を解決するには、サイトのスループットを処理できるクライアントシステムが必要です。

NSWL クライアントをダウンロードする

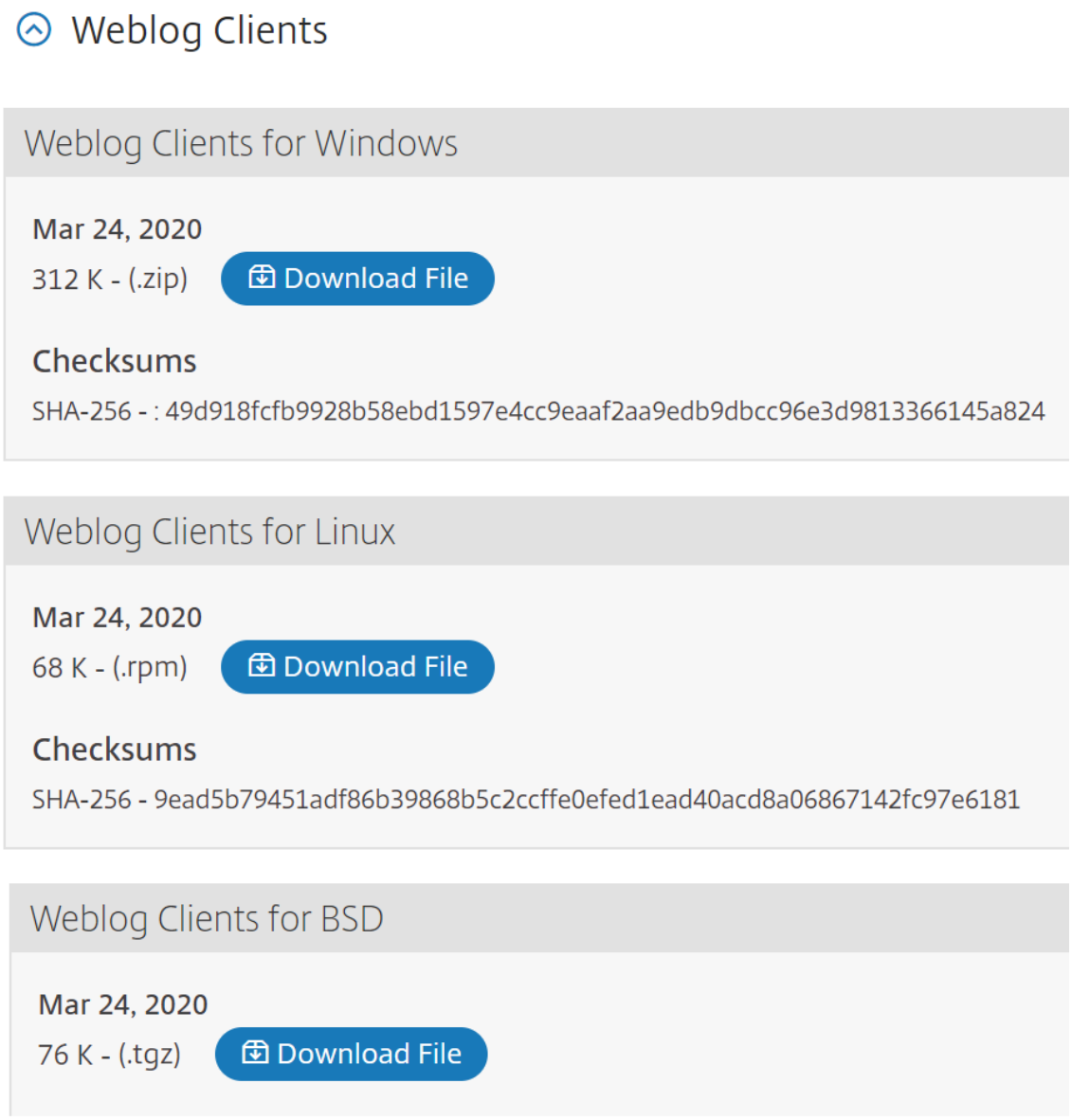
NSWL クライアント・パッケージは、Citrix ADC 製品 CD または Citrix ダウンロード・サイトから入手できます。パッケージ内には、サポートされているプラットフォームごとに個別のインストールパッケージがあります。

Citrix Web サイトから **NSWL** クライアントをダウンロードするには

1. URL`https://www.citrix.com/downloads/citrix-adc/`にアクセスして Citrix にログオンします。
2. 特定の Citrix ADC リリースバージョンに移動し、そのファームウェアを探します。
3. [ファームウェア] をクリックします（たとえば、Citrix ADC リリース（機能フェーズ）13.0 ビルド 52.24）。



4. **Citrix ADC** リリース（機能フェーズ）ビルドページで、[ブログクライアント] セクションに移動します。
5. このセクションでは、Windows、Linux、および BSD 用のウェブログクライアントをダウンロードできます。



Solaris に **NSWL** クライアントをインストールする

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. パッケージから`nswl_solaris-<release number>-<build number>.tar file`を抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする Solaris システムにコピーします。
3. 次のコマンドを使用して、tar ファイルからファイルを抽出します。

   ```
   tar xvf nswl_solaris-9.3-51.5.tar
   ```


一時ディレクトリに Weblog ディレクトリが作成され、ファイルが Weblog ディレクトリに抽出されます。

- 次のコマンドでパッケージをインストールします。

  ```
  pkgadd -d
  ```

- 使用可能なパッケージのリストが表示されます。次の例では、1 つのウェブログパッケージが示されています。

  ```
  1 NSweblog Citrix ADC Weblogging (SunOS,sparc)7.0
  ```

パッケージを選択するように求められます。インストールするウェブログのパッケージ番号を選択します。

パッケージ番号を選択して **Enter** キーを押すと、ファイルが抽出され、次のディレクトリにインストールされます。

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. NSWL パッケージがインストールされているかどうかを確認するには、次のコマンドを実行します。

   ```
   pkginfo | grep NSweblog
   ```

2. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   pkgrm NSweblog
   ```

Linux に NSWL クライアントをインストールする

> 重要
>
> Linux に NSWL クライアントをインストールすると、構成ファイルが置き換えられます。インストールする前にバックアップを取らなければなりません。

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. パッケージからnswl_linux-<release number>-<build number>.rpm ファイルを抽出します。
2. 展開したファイルを、NSWL クライアントをインストールする Linux OS を実行しているシステムにコピーします。
3. NSWL パッケージをインストールするには、次のコマンドを実行します。

   ```
   rpm -i nswl_linux-9.3-51.5.rpm
   ```

このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   rpm -e NSweblog
   ```


2. Weblog RPM ファイルの詳細を取得するには、次のコマンドを実行します。

   ```
   rpm -qpi *.rpm
   ```

3. インストールされている Web サーバーのログファイルを表示するには、次のコマンドを実行します。

   ```
   rpm -qpl *.rpm
   ```

FreeBSD に NSWL クライアントをインストールする

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. パッケージから`nswl_bsd-<release number>-<build number>.tgz` ファイルを抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする FreeBSD OS を実行しているシステムにコピーします。
3. NSWL パッケージをインストールするには、次のコマンドを実行します。

   ```
   pkg_add nswl_bsd-9.3-51.5.tgz
   ```

このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

```
1 -  /usr/local/netscaler/etc
2 -  /usr/local/netscaler/bin
3 -  /usr/local/netscaler/samples
```

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   pkg_delete NSweblog
   ```

2. パッケージがインストールされていることを確認するには、次のコマンドを実行します。

   ```
   pkg_info | grep NSweblog
   ```

Mac に NSWL クライアントをインストールする

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. パッケージから`nswl_macos-<release number>-<build number>.tgz` ファイルを抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする macOS を実行しているシステムにコピーします。
3. NSWL パッケージをインストールするには、次のコマンドを実行します。

   ```
   pkg_add nswl_macos-9.3-51.5.tgz
   ```


このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

   ```
   pkg_delete NSweblog
   ```

2. パッケージがインストールされていることを確認するには、次のコマンドを実行します。

   ```
   pkg_info | grep NSweblog
   ```

Windows に NSWL クライアントをインストールする

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. パッケージからnswl_win-<release number>-<build number>.zip ファイルを抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする Windows システムにコピーします。
3. Windows システムでは、ファイルをディレクトリ（<NSWL-HOME>を参照）に解凍します。次のディレクトリが抽出されます: /bin、および /etc と /samples。
4. コマンドプロンプトで、<NSWL-HOME>\bin directory:から次のコマンドを実行します、

   ```
   nswl -install -f <directorypath>\log.conf
   ```

各項目の意味は次のとおりです。

ディレクトリパスは、構成ファイル (log.conf) のパスを参照します。デフォルトでは、ファイルは <NSWL-HOME> および /etc ディレクトリにあります。構成ファイルを他のディレクトリにコピーできます。

> 注
>
> NSWL クライアントをアンインストールするには、コマンドプロンプトで、<NSWL-HOME>\bin directory:から次のコマンドを実行します。

```
1 > nswl -remove
```

NSWL クライアントを AIX システムにインストールします

NSWL クライアントをインストールするには、パッケージをダウンロードしたシステムで次の操作を実行します。

1. パッケージからnswl_aix-<release number>-<build number>.rpm ファイルを抽出します。
2. 抽出したファイルを、NSWL クライアントをインストールする AIX OS を実行しているシステムにコピーします。

3. NSWL パッケージをインストールするには、次のコマンドを実行します。

```
rpm -i nswl_aix-9.3-51.5.rpm
```

このコマンドは、ファイルを抽出し、次のディレクトリにインストールします。

- /usr/local/netscaler/etc
- /usr/local/netscaler/
- usr/local/netscaler/samples

1. NSWL パッケージをアンインストールするには、次のコマンドを実行します。

```
rpm -e NSweblog
```

2. Weblog RPM ファイルの詳細を取得するには、次のコマンドを実行します。

```
rpm -qpi *.rpm
```

3. インストールされている Web サーバーのログファイルを表示するには、次のコマンドを実行します。

```
rpm -qpl *.rpm
```

NSWL クライアントの構成

July 15, 2022

NSWL クライアントをインストールしたら、nswl 実行可能ファイルを使用して NSWL クライアントを構成できます。これらの構成は、NSWL クライアント構成ファイル (log.conf) に保存されます。

> 注:
>
> NSWL 構成ファイル (log.conf) にさらに変更を加えることで、Web サーバーのログをカスタマイズできます。詳細については、「NSWL クライアントシステムでのロギングのカスタマイズ」を参照してください。

次の表では、NSWL クライアントの構成に使用できるコマンドについて説明します。

NSWL コマンド	**Specifies**
nswl-help	使用可能な NSWL ヘルプオプション。
nswl-addns-f <path-to-configuration-file>	ログトランザクションデータを収集するシステム。Citrix ADC アプライアンスの IP アドレスを入力するように求められます。有効なユーザ名とパスワードを入力します。
nswl -verify -f <path-to-configuration-file>	設定ファイルに構文エラーまたはセマンティックエラーがないかチェックします。

NSWL コマンド	**Specifies**
nswl -start -f <path-to-configuration-file>	構成ファイルの設定に基づいて NSWL クライアントを起動します。注:Solaris および Linux の場合:Web サーバーのロギングをバックグラウンドプロセスとして開始するには、コマンドの最後にアンパサンド記号 (&) を入力します。
nswl-stop (ソラリスおよび Linux のみ)	NSWL クライアントがバックグラウンドプロセスとして起動された場合は停止します。それ以外の場合は、Ctrl+C キーを押して Web サーバのロギングを停止します。
nswl-install-f <path-to-configuration-file> (Windows のみ)	Windows で NSWL クライアントをサービスとしてインストールします。
nswl-サービス開始 (Windows のみ)	nswl インストールオプションで指定されている構成ファイルの設定を使用して、NSWL クライアントを起動します。NSWL クライアントは、[スタート] → [コントロールパネル] → [サービス] からも起動できます。注:NSWL ログファイルは C:\Windows\SysWOW64. に作成されます。
nswl-stopservice (Windows のみ)	NSWL クライアントを停止します。
nswl-削除	NSWL クライアントサービスをレジストリから削除します。

NSWL 実行可能ファイルが配置されているディレクトリから次のコマンドを実行します。

- Windows: \ns\bin
- Solaris and Linux: \usr\local\netscaler\bin

Web サーバーロギング設定ファイルは、次のディレクトリパスにあります。

- Windows: \ns\etc
- Solaris and Linux: \usr\local\netscaler\etc

NSWL 実行可能ファイルはとして起動されます。\ nswl は Linux と Solaris で使えます。

Citrix ADC アプライアンスの **IP** アドレスを追加します

NSWL クライアント構成ファイル（log.conf）に、NSWL クライアントがログの収集を開始する Citrix ADC IP アドレス（NSIP）を追加します。

Citrix ADC アプライアンスの NSIP アドレスを追加するには

1. クライアントシステムのコマンドプロンプトで、次のように入力します。

   ```
   nswl -addns -f < directorypath > \log.conf
   < directorypath >: Specifies the path to the configuration file (log.
   conf)。
   ```

2. 次のプロンプトで、次の情報を入力します。

   - **NSIP**: Citrix ADC アプライアンスの IP アドレスを指定します。
   - ユーザー名とパスワード: Citrix ADC アプライアンスの nsroot ユーザー資格情報を指定します。

     > 注:
     >
     > ロギング権限が有効になっているすべてのシステムユーザがこの機能をサポートします。

> 注意:
>
> 複数の Citrix ADC IP アドレス（NSIP）を追加し、後で Citrix ADC システムログの詳細をすべて記録したくない場合は、log.conf ファイルの最後にある NSIP ステートメントを削除して、NSIP を手動で削除できます。フェイルオーバーのセットアップ中に、コマンドを使用して、プライマリとセカンダリの両方の Citrix ADC IP アドレスを log.conf に追加する必要があります。IP アドレスを追加する前に、Citrix ADC アプライアンスにユーザー名とパスワードが存在することを確認してください。

NSWL 構成ファイルの検証

ロギングが正しく機能することを確認するには、クライアントシステムの NSWL 構成ファイル (log.conf) に構文エラーがないか確認します。

NSWL 構成ファイル内の構成を確認するには

クライアントシステムのコマンドプロンプトで、次のように入力します。

```
nswl -verify -f <directorypath>\log.conf
```

< directorypath>: 設定ファイル (log.conf) へのパスを指定します。

NSWL クライアントの実行

Web サーバーロギングの開始

クライアントシステムのコマンドプロンプトで、次のように入力します。

```
nswl -start -f <directorypath>\log.conf
```

<directorypath>: 設定ファイル (log.conf) へのパスを指定します。

Solaris または Linux オペレーティングシステムでバックグラウンドプロセスとして開始された Web サーバーロギングの停止

コマンドプロンプトで入力します。

```
nswl -stop
```

Windows オペレーティングシステムでサービスとして開始された Web サーバーのロギングを停止するには

コマンドプロンプトで入力します。

```
nswl -stopservice
```

NSWL クライアントシステムでのログ記録のカスタマイズ

May 9, 2022

NSWL クライアント構成ファイル（log.conf）をさらに変更することで、Citrix ADC Web ロギング（NSWL）クライアントシステムでのロギングをカスタマイズできます。テキスト・エディタを使用して、クライアント・システム上の log.conf 構成ファイルを変更します。

ロギングをカスタマイズするには、設定ファイルを使用してフィルタとログプロパティを定義します。

- ログフィルタ。Web サーバーのホスト IP アドレス、ドメイン名、およびホスト名に基づいてログ情報をフィルタリングします。
- ログプロパティ。各フィルタには、関連する一連のログプロパティがあります。ログプロパティでは、フィルタリングされたログ情報の保存方法を定義します。

サンプル設定ファイル

次に、設定ファイルの例を示します。

```
1  ##########
2  # This is the NSWL configuration file
3  # Only the default filter is active
4  # Remove leading # to activate other filters
5  ##########
6  ##########
7  # Default filter (default on)
8  # W3C Format logging, new file is created every hour or on reaching 10
       MB file size,
9  # and the file name is Exyymmdd.log
10 ##########
11 Filter default
12 begin default
13         logFormat               W3C
```



```
        logInterval             Hourly
        logFileSizeLimit        10
        logFilenameFormat       Ex%`{
 `%y%m%d }
 t.log
end default
##########
# Citrix ADC caches example
# CACHE_F filter covers all the transaction with HOST name www.
   netscaler.com and the listed server ip's
##########
#Filter CACHE_F HOST www.netscaler.com IP 192.168.100.89 192.168.100.95
    192.168.100.52 192.168.100.53 ON
##########
# netscaler origin server example
# Not interested in Origin server to Cache traffic transaction logging
##########
#Filter ORIGIN_SERVERS IP 192.168.100.64 192.168.100.65 192.168.100.66
   192.168.100.67 192.168.100.225 192.168.100.226 192.168.
100.227 192.168.100.228 OFF
##########
# netscaler image server example
# all the image server logging.
##########
#Filter IMAGE_SERVER HOST www.netscaler.images.com IP 192.168.100.71
   192.168.100.72 192.168.100.169 192.168.100.170 192.168.10
0.171 ON
##########
# NCSA Format logging, new file is created every day midnight or on
   reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.
   log.
# Exclude objects that ends with .png .jpg .jar.
##########
#begin ORIGIN_SERVERS
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        40
#       logFilenameFormat       /datadisk5/ORGIN/log/%v/NS%`{
 `%m%d%y }
 t.log
#       logExclude              .png .jpg .jar
#end ORIGIN_SERVERS

##########
```



```
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log with log record timestamp as GMT.
##########
#begin CACHE_F
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk5/netscaler/log/%v/NS%`{ `%m%d%y }  t.log
#       logtime                 GMT
#end CACHE_F

##########
# W3C Format logging, new file on reaching 20MB and the log file path name is
# atadisk6/netscaler/log/server's ip/Exmmyydd.log with log record timestamp as LOCAL.
##########
#begin IMAGE_SERVER
#       logFormat               W3C
#       logInterval             Size
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk6/netscaler/log/%AEx%`{ `%m%d%y }  t
#       logtime                 LOCAL
#end IMAGE_SERVER

##########
# Virtual Host by Name firm, can filter out the logging based on the host name by,
##########

#Filter VHOST_F IP 10.101.2.151 NETMASK 255.255.255.0
#begin VHOST_F
#       logFormat               W3C
#       logInterval             Daily
#       logFileSizeLimit        10
logFilenameFormat /ns/prod/vhost/%v/Ex%`{ `%m%d%y }  t
#end VHOST_F
```



```
93
94 ########## END FILTER CONFIGURATION ##########
95 <!--NeedCopy-->
```

フィルタの作成

設定ファイル (log.conf) でデフォルトのフィルタ定義を使用するか、フィルタを変更するか、フィルタを作成することもできます。複数のログフィルタを作成できます。

> 注
>
> フィルタが定義されていないトランザクションをログに記録する統合ロギングは、有効な場合にデフォルトのフィルタを使用します。すべてのサーバーの統合ログは、デフォルトのフィルタのみを定義することによって実行できます。

サーバーが複数の Web サイトをホストし、各 Web サイトに独自のドメイン名があり、各ドメインが仮想サーバーに関連付けられている場合は、Web サーバーログを構成して Web サイトごとに個別のログディレクトリを作成できます。次の表に、フィルタを作成するためのパラメータを示します。

パラメーター	Specifies
filterName	フィルタの名前。フィルタ名には英数字を含めることができ、59 文字を超えることはできません。59 文字を超えるフィルタ名は 59 文字に切り捨てられます。
ホスト名	トランザクションがログに記録されるサーバーのホスト名。
IP `ip`	トランザクションがログに記録されるサーバーの IP アドレス (たとえば、サーバーに 1 つの IP アドレスを持つドメインが複数ある場合)。
IP `ip 2...ip` N:	複数の IP アドレス（たとえば、サーバドメインに複数の IP アドレスがある場合)。
ip6 IP	トランザクションがログに記録されるサーバの IPv6 アドレス。
IP ip NETMASK mask	サブネットで使用される IP アドレスとネットマスクの組み合わせ。
`ON | OFF`	フィルタを有効または無効にしてトランザクションをログに記録します。引数が選択されていない場合、フィルタは有効 (オン) になります。

表 1. フィルタ作成用パラメータ

フィルタを作成するには

フィルタを作成するには、log.conf ファイルに次のコマンドを入力します。

- `filter <filterName> <HOST name> | [IP<ip> ] | [IP<ip 2...ip n> ] | <IP ip NETMASK mask> [ON | OFF]`
- `filter <filterName> <HOST name> | [IP6 ip/<prefix length>] [ON | OFF]`

仮想サーバ用のフィルタを作成するには

仮想サーバ用のフィルタを作成するには、log.conf ファイルに次のコマンドを入力します。

`filter <filterName> <VirtualServer IP address>`

例

次の例では、IP アドレス 192.168.100.0、ネットマスク 255.255.255.0 を指定しています。このフィルタは、192.168.100.1 から 192.168.100.254 までの IP アドレスに適用されます。

```
Filter F1 HOST www.netscaler.com ON
Filter F2 HOST www.netscaler.com IP 192.168.100.151 ON
Filter F3 HOST www.netscaler.com IP 192.168.100.151 192.165.100.152 ON
Filter F4 IP 192.168.100.151
Filter F5 IP 192.168.100.151 HOST www.netscaler.com OFF
Filter F6 HOST www.netscaler.com HOST www.xyz.com HOST www.abcxyz.com
   IP 192.168.100.200 ON
Filter F7 IP 192.250.100.0 NETMASK 255.255.255.0
Filter F8 HOST www.xyz.com IP 192.250.100.0 NETMASK 255.255.255.0 OFF
For creating filters for servers having IPv6 addresses.
Filter F9 2002::8/112 ON
Filter F10 HOST www.abcd.com IP6 2002::8 ON

<!--NeedCopy-->
```

ログプロパティを指定する

ログプロパティは、フィルタに関連付けられているすべてのログエントリに適用されます。ログプロパティの定義は、次の例に示すように、キーワード BEGIN で始まり、END で終わります。

```
BEGIN <filtername>
 logFormat ...
 logFilenameFormat ...
```



```
 logInterval ...
 logFileSize ....
 logExclude ....
 logTime ….
 END
<!--NeedCopy-->
```

定義には、次の項目を含めることができます。

-

LogFormat は、NCSA、W3C 拡張、およびカスタムログファイル形式をサポートする Web サーバーロギング機能を指定します。

```
By default, the `logformat` property is w3c. To override, enter custom
    or NCSA in the configuration file, for example:
```

```
     LogFormat NCSA
<!--NeedCopy-->
```

> 注
>
> NCSA およびカスタムログ形式では、トランザクションのタイムスタンプとファイルのローテーションにローカル時間が使用されます。

- **LogInterval** は、新しいログファイルが作成される間隔を指定します。次のいずれかの値を使用します。
  - Hourly: 1 時間ごとにファイルが作成されます。
  - 毎日: 毎日午前 0 時にファイルが作成されます。デフォルト値です。
  - Weekly: 毎週日曜日の午前 0 時にファイルが作成されます。
  - 毎月: 月の初日の午前 0 時にファイルが作成されます。
  -

例:

```
LogInterval Daily
<!--NeedCopy-->
```

LogFileSizeLimit は、ログファイルの最大サイズを MB 単位で指定します。任意のログ間隔 (毎週、毎月など) で使用できます。ファイルは、最大ファイルサイズ制限に達したとき、または定義されたログ間隔が経過したときに作成されます。

この動作をオーバーライドするには、ログファイルのサイズ制限に達したときにのみファイルが作成されるように、サイズを`loginterval`プロパティとして指定します。

デフォルトのログファイルサイズ制限は 10 MB です。

例:

```
LogFileSizeLimit 35
<!--NeedCopy-->
```

- **LogFileNameFormat** は、ログファイルのファイル名形式を指定します。ファイル名には次の種類があります。
  - Dynamic: 次の形式を含むエクスプレッションを指定します。
    * サーバ IP アドレス
    * 日付 (%{フォーマット}t)
    * URL サフィックス (%x)
    * ホスト名 (%v)

例:

```
LogFileNameFormat Ex%`{
 `%m%d%y }
 t.log
<!--NeedCopy-->
```

このコマンドは、最初のファイル名を Exmmddyy.log として作成し、1 時間ごとにファイル名でファイルを作成します。exmmddyy.log.0、exmmDDYY.log.1、…、exmmDDYY.log.n。

例:

```
      LogInterval size
      LogFileSize 100
      LogFileNameFormat Ex%`{
 `%m%d%y }
 t
<!--NeedCopy-->
```

> 注意
>
> LogFilenameFormat コマンドで指定された日付形式%t は、そのフィルタのログ間隔プロパティを上書きします。指定したログファイルのサイズに達したときではなく、毎日新しいファイルが作成されないようにする

> には、LogFilenameFormat で%t を使用しないでください。

- **LogExclude** は、指定したファイル名拡張子を持つトランザクションのログを防止します。

例**:**

```
1       LogExclude .html
2 <!--NeedCopy-->
```

このコマンドは、*.html ファイルのログトランザクションを除外したログファイルを作成します。

LogTime は、ログ時間を GMT または LOCAL として指定します。

既定値は次のとおりです。

- NCSA ログファイル形式:LOCAL
- W3C ログファイル形式:GMT

NCSA および **W3C** ログ形式について

Citrix ADC は、次の標準ログファイル形式をサポートしています。

- NCSA 共通ログ形式
- W3C 拡張ログ形式

NCSA 共通ログ形式

ログファイル形式が NCSA の場合、ログファイルには次の形式でログ情報が表示されます。

```
1 Client_IP_address -User_Name [Date:Time -TimeZone] ”Method Object
     HTTP_version” HTTP_StatusCode BytesSent
2 <!--NeedCopy-->
```

NCSA 共通ログ形式を使用するには、log.conf ファイルの logFormat 引数に NCSA と入力します。

次の表に、NCSA Common ログの形式を示します。

引数	Specifies
Client_ip_address	クライアントコンピュータの IP アドレス。
ユーザー名	ユーザー名。
日付	トランザクションの日付。
時間	トランザクションが完了した時刻。

引数	Specifies
タイムゾーン	タイムゾーン (グリニッジ標準時または現地時間)。
方法	リクエストメソッド (GET、POST など)。
オブジェクト	URL。
HTTP_version	クライアントが使用する HTTP のバージョン。
HTTP_StatusCode	レスポンスのステータスコード。
送信済みバイト数	サーバから送信されたバイト数。

W3C 拡張ログ形式

拡張ログファイルには、ラインフィード (LF) またはシーケンスのキャリッジリターンラインフィード (CRLF) のいずれかで終了する ASCII 文字を含む一連の行が含まれます。ログファイルジェネレータは、そのジェネレータが実行されるプラットフォームの行終了規則に従う必要があります。

ログアナライザは、LF 形式または CRLF 形式を受け入れる必要があります。各行には、ディレクティブまたはエントリのいずれかを含めることができます。W3C 拡張ログ形式を使用する場合は、log.conf ファイルの Log-Format 引数として W3C と入力します。

既定では、標準の W3C ログ形式は、次のようにカスタムログ形式として内部的に定義されています。

```
%`{
 `%Y-%m-%d%H:%M:%S }
 t %a %u %S %A %p %m %U %q %s %j %J %T %H %+{
 user-agent }
 i %+{
 cookie }
  i%+{
 referer }
 i
<!--NeedCopy-->
```

この W3C ログ形式では、順序を変更したり、一部のフィールドを削除したりすることもできます。例:

```
logFormat W3C %`{
 `%Y-%m-%d%H:%M:%S }
 t %m %U
<!--NeedCopy-->
```


W3C ログエントリは、次の形式で作成されます。

```
#Version: 1.0
#Fields: date time cs-method cs-uri
#Date: 12-Jun-2001 12:34
2001-06-12 12:34:23 GET /sports/football.html 2001-06-12 12:34:30
GET /sports/football.html
<!--NeedCopy-->
```

エントリー

エントリは、単一の HTTP トランザクションに関連する一連のフィールドで構成されます。フィールドは空白で区切られます。タブ文字の使用することをお勧めします。特定のエントリのフィールドが使用されていない場合、省略されたフィールドはダッシュ（-）でマークされます。

ディレクティブ

ロギングプロセスの詳細については、ディレクティブ （Directives）の表を参照してください。シャープ記号 (#) で始まる行には、ディレクティブが含まれます。

例:

次のサンプルログファイルには、W3C Extended ログ形式でログエントリが表示されます。

```
#Version: 1.0
#Fields: time cs-method cs-uri
#Date: 12-Jan-1996 00:00:00
00:34:23 GET /sports/football.html
12:21:16 GET /sports/football.html
12:45:52 GET /sports/football.html
12:57:34 GET /sports/football.html
<!--NeedCopy-->
```

フィールド

Fields ディレクティブは，各エントリに記録される情報を指定する一連の項目識別子をリストします。フィールド識別子には、次のいずれかの形式があります。

- **identifier**：トランザクション全体に関連します。
- **prefix-identifier:** 値プレフィックスによって定義されるパーティ間の情報転送に関連します。

- **prefix (header):** 値プレフィックスで定義されるパーティ間の転送用の HTTP ヘッダーフィールドヘッダーの値を指定します。この方法で指定されたフィールドには、常に型があります。

次の表に、定義済みの接頭辞を示します。

前	Specifies
c	クライアント
s	サーバー
r	リモート
cs	クライアントからサーバへ
sc	サーバからクライアントへ
sr	サーバーからリモートサーバー (プロキシが使用するプレフィックス)
rs	リモートサーバーからサーバー (プロキシが使用するプレフィックス)
x	アプリケーション固有識別子

例:

次の例は、接頭辞を使用する定義済み識別子です。

cs-method：クライアントからサーバーに送信されるリクエストのメソッド。

sc (リファラー**):** 返信の`Referer`フィールド。

c-ip: クライアントの IP アドレス。

[識別子**]**

次の表に、プレフィックスを必要としない W3C 拡張ログ形式 ID を示します。

識別子	説明
日付	トランザクションが行われた日付。
時間	トランザクションが完了した時刻。
時間かかった	トランザクションの完了にかかった時間 (秒単位)。
bytes	転送されたバイト数。
キャッシュされます	キャッシュヒットが発生したかどうかを記録します。ゼロはキャッシュミスであることを示します。

表 5. W3C 拡張ログ形式識別子 (接頭辞不要)

次の表に、プレフィックスが必要な W3C 拡張ログ形式 ID を示します。

識別子	説明
IP	IP アドレスとポート番号。
DNS	DNS 名。
状態	ステータスコード。
comment	コメントはステータスコードとともに返されました。
method	メソッド。
url	URL。
url-stem	URL の語幹部分。
URL クエリ	URL のクエリ部分。

表 6. W3C 拡張ログ形式識別子 (接頭辞が必要)

W3C 拡張ログファイル形式では、ログフィールドを選択できます。これらのフィールドを次の表に示します。

フィールド	説明
日付	トランザクションが完了した日付。
時間	トランザクションが完了した時刻。
クライアント IP	クライアントの IP アドレス。
ユーザー名	ユーザー名。
サービス名	サービス名。常に HTTP です。
サーバー IP	サーバの IP アドレス。
サーバーポート	サーバのポート番号
方法	リクエストメソッド (GET、POST など)。
URL ステム	URL ステム。
URL クエリー	URL のクエリ部分。
HTTP ステータス	レスポンスのステータスコード。
送信済みバイト数	サーバーに送信されたバイト数 (HTTP ヘッダーを含むリクエストサイズ)。
受信バイト数	サーバーから受信したバイト数 (HTTP ヘッダーを含むレスポンスサイズ)。

フィールド	説明
Time Taken	トランザクションが完了するまでの時間（秒）。
Protocol Version	クライアントが使用している HTTP のバージョン番号。
ユーザーエージェント	HTTP プロトコルの [User-Agent] フィールド。
クッキー	HTTP プロトコルの Cookie フィールド。
`Referer`	HTTP プロトコルの`Referer`フィールド。

表 7. W3C 拡張ログファイル形式 (ログフィールドを許可)

カスタムログ形式を作成する

ログファイルデータの表示形式は、手動で、または NSWL ライブラリを使用してカスタマイズできます。カスタムログ形式を使用すると、Apache が現在サポートしているほとんどのログ形式を取得できます。

NSWL ライブラリを使用したカスタムログ形式の作成

NSWL 実行可能ファイルが Windows または Solaris のホスト・コンピュータにインストールされているかどうかに応じて、次の NSWL ライブラリのいずれかを使用します。

- **Windows:** システムマネージャのホストコンピュータ上の\ ns\ bin ディレクトリにある nswl.lib ライブラリ。
- **Solaris:** /usr/local/netscaler/bin 内の libnswl.a ライブラリです。

NSWL ライブラリを使用してカスタムログ形式を作成するには

1. システムで定義された次の 2 つの C 関数を C ソースファイルに追加します。

   ns_userDefFieldName (): この関数は、ログレコードにカスタムフィールド名として追加する必要がある文字列を返します。

   ns_userDefFieldVal (): この関数は、カスタムフィールド値を実装し、それを文字列として返します。この値はログレコードの最後に追加する必要があります。

2. ファイルをオブジェクトファイルにコンパイルします。
3. オブジェクトファイルを NSWL ライブラリ (およびオプションでサードパーティのライブラリ) にリンクして、新しい NSWL 実行可能ファイルを作成します。
4. 設定ファイル (log.conf) の LogFormat 文字列の最後に%d 文字列を追加します。

例:

```
##########
# A new file is created every midnight or on reaching 20MB file size,
# and the file name is
/datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log and create
digital
#signature field for each record.
BEGIN CACHE_F
    logFormat custom "%a - "%{
 user-agent }
 i" [%d/%B/%Y %T -%g] "%x"
%s %b%{
 referrer }
 i "%{
 user-agent }
 i" "%{
 cookie }
 i" %d "
    logInterval Daily
    logFileSizeLimit 20
    logFilenameFormat
/datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log
END CACHE_F
<!--NeedCopy-->
```

カスタムログ形式の手動作成

ログファイルデータの表示形式をカスタマイズするには、LogFormat ログプロパティ定義の引数として文字列を指定します。次に、文字列を使用してログ形式を作成する例を示します。

```
LogFormat Custom ""%a - "%{
 user-agent }
 i" %[%d/%m/%Y]t %U %s %b %T"
<!--NeedCopy-->
```

- 文字列には、新しい行とタブを表す「c」タイプの制御文字\ n と\ t を含めることができます。
- リテラル引用符とバックスラッシュで Esc キーを使用します。

リクエストの特性は、フォーマット文字列に% ディレクティブを配置することで記録されます。このディレクティブは、ログファイルでは値で置き換えられます。

%v (ホスト名) または%x (URL サフィックス) の形式指定子がログファイル名の形式文字列に存在する場合、ファイル名の次の文字がログ設定ファイル名のアンダースコア記号に置き換えられます。

” *. /: < > ? \

ASCII 値が 0 ～31 の範囲にある文字は、次のように置き換えられます。

%<ASCII value of character in hexadecimal>。

たとえば、ASCII 値 22 の文字は%16 に置き換えられます。

> 注意
>
> %v 形式指定子がログファイル名の形式文字列に存在する場合、仮想ホストごとに個別のファイルが開きます。継続的なログ記録を確保するには、プロセスが開くことができるファイルの最大数が十分に大きい必要があります。開くことができるファイル数を変更する手順については、オペレーティングシステムのマニュアルを参照してください。

Apache ログフォーマットの作成

Apache が現在サポートしているほとんどのログ形式をカスタムログから引き出すことができます。Apache ログ形式に一致するカスタムログ形式は次のとおりです。

NCSA/combined: LogFormat custom %h %l %u [%t] “%r” %s %B “%{referer}i” “%{user-agent}i”

NCSA/Common: LogFormat custom %h %l %u [%t] “%r” %s %B

Referer Log: LogFormat custom “%{referer}i” -> %U

User agent: LogFormat custom %{user-agent}i

同様に、カスタム形式から他のサーバーログ形式を派生させることができます。

カスタムログ形式を定義するための引数

カスタムログ形式の定義の詳細については 、カスタムログ形式の PDF 表を参照してください。

> 注
>
> カスタム HTTP ヘッダーをエクスポートする方法については、「Web サーバーロギング用の Citrix ADC 構成」を参照してください。

たとえば、ログ形式を%+ {user-agent} i として定義し、ユーザーエージェントの値が Citrix ADC システムの Web クライアントである場合、情報は Citrix ADC システム +Web+ クライアントとして記録されます。代わりに、二重引用符を使用することもできます。たとえば、「% {user-agent} i」は、それを「Citrix ADC システム Web クライア

ント」として記録します。<Esc\ >%.. .r,%. .i と,%. .o. の文字列には\ キーを使用しないでください。これは、共通ログ形式の要件に準拠しています。クライアントはログに制御文字を挿入できます。したがって、raw ログファイルを操作するときは注意が必要です。

時間形式の定義

カスタムログフォーマット表で説明されている% {format} t 文字列のフォーマット部分については、時間形式の定義表を参照してください 。角括弧 ([]) 内の値は、表示される値の範囲を示します。たとえば、次の表の%d の説明の [1,31] は、1 から 31 までの%d の範囲を示しています。

> 注
>
> 前の表で説明した変換のいずれにも対応しない変換を指定した場合、または次の段落で示した変換指定を変更した場合、動作は未定義になり、0 が返されます。

%U と%W の差 (および修正コンバージョン%OU と%OW) の違いは、週の最初の曜日とみなされる日です。週番号 1 は、1 月の最初の週です (%U は日曜日、%W は月曜日で始まります)。週番号 0 には、%U および%W の 1 月の最初の日曜日または月曜日の前の日が含まれます。

サーバーログの表示

NSWL 機能を構成して、コンソールにサーバーログを表示したり、サーバーログを Citrix ADC アプライアンス上のディレクトリにリダイレクトしたりできます。

コンソールにログを表示する方法は 2 つあります (標準出力)。
オプション 1: コンソールにすべてのログを表示する。
オプション 2: STDOUT としてlogfilenameformatフィルタを使用して、選択したログのみをコンソールに表示します。

Call Home

October 7, 2021

アプライアンスは、ソフトウェアまたはハードウェアの問題により、正常に動作しないことがあります。このような場合、Citrix は、お客様のサイトで潜在的な影響が発生する前に、データを収集して問題解決を行う必要があります。Citrix ADC アプライアンスで Call Home を有効にすると、エラー通知プロセスを自動化できます。サポートチームが問題のトラブルシューティングを行う前に、Citrix サポートへの電話、サービスリクエストの発生、システムデータのアップロードを回避できるだけでなく、サポートは問題が発生する前に特定して対処できます。Call Home はアプライアンスを定期的に監視し、Citrix テクニカルサポートサーバーにデータを自動的にアップロードします。さらに、着信 Call Home データは、Citrix ADC の使用状況に関する洞察を提供します。Citrix 社内の複数のチームがこのデータを使用して、Citrix ADC の設計、サポート、実装を改善できます。

デフォルトでは、Call Home はすべてのプラットフォームと Citrix ADC（MPX、VPX、SDX）のすべてのフレーバーで有効になっています。この機能を有効にすると、Citrix ADC の展開とテレメトリデータを収集して、実装とサポートサービスを向上させることができます。

> 注
>
> また、Call Home に関する詳細については、Call Home FAQ ページを参照してください。

長所

Call Home には、次のような利点があります。

- ハードウェアおよびソフトウェアのエラー状態を監視します。詳細については、「重大なエラー状態を監視する」を参照してください。
- ネットワークに影響を与える重要なイベントを通知します。
- パフォーマンスデータとシステム使用状況の詳細を Citrix に送信します。
  - 製品の品質を分析し、改善します。
  - プロアクティブな問題の特定と迅速な問題解決のために、リアルタイムのトラブルシューティング情報を提供します。

プラットフォームのサポート

Call Home 機能は、すべての Citrix ADC プラットフォームとすべてのアプライアンスモデル（MPX、VPX、SDX）でサポートされています。

- Citrix ADC の MPX: すべての MPX モデル.
- Citrix ADC VPX: 外部または中央のライセンスプールからライセンスを取得する VPX アプライアンスを含む、すべての VPX モデル。
- Citrix ADC SDX: ディスクドライブを監視し、エラーや障害がないか SSL チップを割り当てます。ただし、VPX インスタンスは電源ユニット（PSU）にアクセスできないため、ステータスは監視されません。SDX プラットフォームでは、Call Home を個々のインスタンスで直接設定することも、SVM を介して設定することもできます。

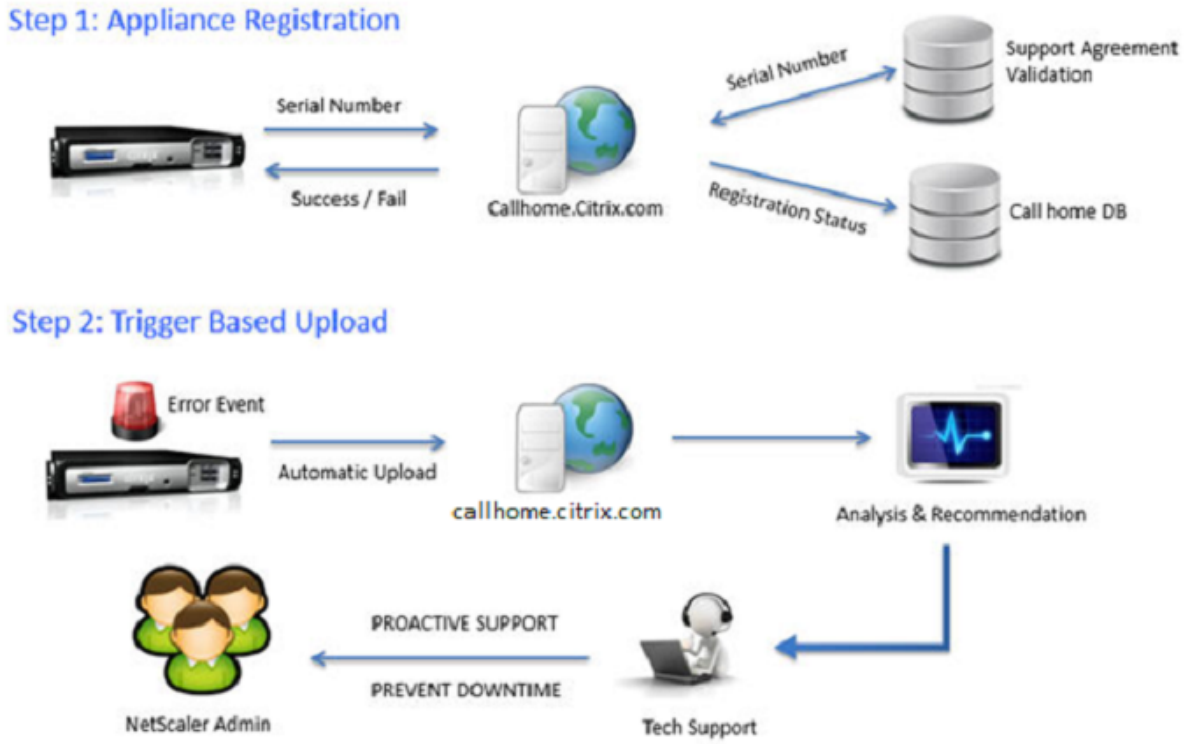
前提条件

Call Home を使用するには、Citrix ADC アプライアンスに次のものが必要です。

- インターネット接続。Call Home では、データアーカイブをアップロードするために Citrix ADC が Citrix サポートサーバーに接続するためのインターネット接続が必要です。

Call Home の仕組み

次の図は、お客様のサイトに展開された Citrix ADC アプライアンスでの CallHome の基本的なワークフローを示しています。



Call Home のワークフローは次のとおりです。

1. インターネット接続をセットアップします。Call Home でシステムデータをアップロードするには、アプライアンスにインターネット接続が必要です。そうでない場合は、インターネット接続を提供するようにプロキシサーバー構成を構成できます。詳細については、「Call Home の設定」を参照してください。

2. Call Home を有効にします。Citrix ADC コマンドインターフェイスまたは GUI を使用してアプライアンスを最新のソフトウェアにアップグレードする場合、Call Home はデフォルトで有効になっており、システムは登録プロセスを 24 時間遅らせます。この期間中、機能を手動で無効にすることもできますが、有効にすることをお勧めします。

> 注
>
> Call Home が明示的に無効になっている古いバージョンからアプライアンスをアップグレードする場合、システムはデフォルトでこの機能を有効にし、初回ログイン時に通知メッセージを表示します。

さらに、インターネット接続の設定を変更する場合は、Call Home を無効にして有効にする必要があります。これにより、Call Home は、障害エラーを発生させずに Citrix Insight Services（CIS）サーバーに登録できます。

3. Citrix ADC アプライアンスを **Citrix** サポートサーバーに登録します。Call Home がアプライアンスを Citrix サポートサーバーに登録すると、サーバーはデータベースでアプライアンスのシリアル番号の妥当性をチェックします。シリアル番号が有効な場合、サーバはアプライアンスを Call Home サービスに登録し、正常な登録応答を送信します。それ以外の場合、サーバは登録失敗メッセージを返送します。基本的なシステム情報は、別のメッセージとして送信されます。データには、メモリと CPU 使用率の詳細と、スループット数が含まれます。デフォルトでは、ハートビートメッセージの一部として定期的に 7 日ごとにデータが送信されます。ただし、頻繁なアップロードは役に立たないため、5 日未満の値はお勧めしません。

4. 重大なエラー状態を監視する。登録が完了すると、Call Home はアプライアンスの監視を開始します。次の表に、Call Home がアプライアンスで監視できる条件を示します。

重大なエラー状態	説明	Call Home モニタリングインターバル	対応する SNMP アラーム名
コンパクトフラッシュドライブのエラー	アプライアンスのコンパクトフラッシュドライブで読み取りまたは書き込みエラーが発生しました。	24 時間	COMPACT-FLASH-ERRORS
ハードディスクドライブのエラー	アプライアンスのハードドライブで読み取りまたは書き込みエラーが発生しました。	24 時間	HARD-DISK-DRIVE-ERRORS
電源装置障害	Citrix ADC アプライアンスの電源装置の 1 つに障害が発生しました。	7 秒	POWER-SUPPLY-FAILURE
SSL カードの障害	Citrix ADC アプライアンスの SSL カードの 1 つに障害が発生しました。	7 秒	SSL-CARD-FAILED
ウォームリスタート	システムプロセスの障害により、アプライアンスがウォームリスタートしました。	Citrix ADC アプライアンスを再起動するたびに。	WARM-RESTART-EVENT
Memory anomaly error	メモリ使用率は、通常の制限を超えて徐々に増加し、しきい値を超えています。	1 日	No SNMP alarm
レート制限パケットドロップ	スループット制限またはパケット/秒（pps）制限に達した。	7 秒	PF-RL-PPS-PKTS-DROPPED, PF-RL-RATE-PKTS-DROPPED

5. Call Home データをアップロードします。アプライアンスで以前の重大な状態のいずれかが特定された場合、CallHome 機能は Citrix サポートに自動的に通知します。サポートアーカイブは Citrix サポートサーバーにアップロードされます。また、Call Home アップロードが発生するたびに SNMP アラートを生成するように、CALLHOME-UPLOAD-EVENTSNMP アラームを設定できます。SNMP アラートは、クリティカルイベントについてローカル管理者に通知します。

> 注
>
> Call Home は、最後の再起動以降に特定のエラー状態が最初に発生した場合にのみ、Call Home の tar ファイルを作成し、Citrix テクニカルサポートサーバーにアップロードします。特定のエラー状態が発生するたびにアプライアンスがアラートを送信するようにするには、エラー条件に対応する SNMP アラームを設定します。

6. サービスリクエストの作成。Call Home は、ハードウェアに関連するすべての重要なイベントのサービスリクエストを自動的に作成します。イベントは次のように分類されます。電源装置の障害、SSL カードの障害、ハードディスクドライブのエラー、およびコンパクトフラッシュのエラー。その他のエラーについては、システムログを確認した後、Citrix サポートチームに連絡して調査のためのサービスリクエストを提出できます。

Call Home の設定

Call Home を設定するには、アプライアンスのインターネット接続を確認し、DNS ネームサーバが設定されていることを確認します。インターネット接続がない場合は、プロキシサーバーまたはサービスを構成します。次に、アプライアンスで Call Home を有効にし、Citrix サポートサーバーへのアプライアンスの登録ステータスを確認します。登録が完了すると、Call Home はデータを監視およびアップロードできます。さらに、SNMP アラームを設定して、カスタマーサイトの管理者に通知することもできます。

Call Home を構成するには、Citrix ADC コマンドインターフェイスまたは GUI のいずれかを使用して、次のタスクを実行できます。

- CallHome を有効にします。
- オプションのプロキシサーバパラメータの Call Home を設定します。
- Call Home 登録ステータスを確認します。
- エラーとタイムスタンプの詳細を表示します。
- SNMP アラームを設定します。

Citrix ADC コマンドインターフェイスを使用して **CallHome** を構成するには

Citrix ADC コマンドインターフェイスでは、次の操作を実行できます。

`Enabling Call Home`

コマンドプロンプトで入力します。

```
enable ns feature callhome
```

オプションのプロキシサーバパラメータの Call Home の設定

Call Home を使用すると、インターネット接続用のオプションのプロキシサーバを設定できます。IP アドレスとポートを使用してプロキシサーバーを構成するか、一方向または双方向認証でプロキシ認証サービスを構成できます。

`To configure optional proxy server with IP address and port`

コマンドプロンプトで入力します。


```
set callhome -proxyMode ( YES | NO )[-IPAddress <ip_addr|ipv6_addr|*>] [-
port <port |*>]
```

```
1 set callhome -proxyMode YES -IPAddress 10.102.167.33 -port 80
2 <!--NeedCopy-->
```

> 注
>
> Call Home は、proxy-mode パラメーターを YES に設定した場合にのみ、プロキシサーバーを使用します。[NO] に設定すると、IP アドレスとポートが設定されていても、プロキシ機能は機能しません。ポート番号は、HTTPS サービスではなく、HTTP サービス用である必要があります。

オプションのプロキシ認証サービスを構成するには

このモードでは、一方向と双方向の 2 種類のセキュリティ認証が提供されます。どちらのタイプも設定するには、SSL サービスを設定する必要があります。詳細については、「SSL サービスの構成」トピックを参照してください。

一方向認証では、Citrix ADC アプライアンスのみがプロキシサーバーを認証します。双方向認証では、Citrix ADC アプライアンスはプロキシサーバーを認証し、プロキシサーバーはアプライアンスを認証します。

プロキシ認証サービスを構成するには

コマンドプロンプトで入力します。

```
set callhome -proxyMode ( YES | NO )[-proxyAuthService <string>]
```

```
1 set callhome -proxyMode YES -proxyAuthService callhome_proxy
2 <!--NeedCopy-->
```

一方向プロキシサーバ認証を構成するには

一方向プロキシサーバ認証を設定するには、次のタスクを実行します。

1. SSL サービスを作成します。
2. サービスに CA 証明書をバインドします。
3. HTTPS モニターをサービスにバインドします。
4. SSL サービスを使用するように Call Home を設定します。

双方向プロキシサーバー認証を構成するには

双方向プロキシサーバ認証を設定するには、次のタスクを実行します。

1. SSL サービスの作成
2. サービスに CA 証明書をバインドします。
3. クライアント証明書をバインドします。
4. HTTPS モニターをサービスにバインドします。

5. SSL サービスを使用するように Call Home を設定します。

Call Home 登録ステータスの確認

コマンドプロンプトで入力します。

```
show callhome

    show callhome

     Registration with Citrix upload server SUCCESSFUL

     Mode: Default

     Contact email address: exampleadmin@example.com

     Heartbeat Custom Interval (days): 7

     Proxy Mode: Yes

        Proxy IP Address:10.102.29.200

        Proxy Authentication Service:

        Proxy Port: 80

    Trigger event                  State   First occurrence
                      Latest occurrence

     -------------                  -----   ----------------
                     -----------------

     1) Warm boot                   Enabled N/A
                                    ..

     2) Compact flash errors        Enabled ..
                                      ..

     3) Hard disk drive errors      Enabled ..
                                      ..

     4) SSL card failure            N/A     N/A
                                    N/A

```



```
33       5) Power supply unit failure   N/A     N/A
                                        N/A
34
35       6) Rate limit packet drops     Enabled ..
                                          ..
36
37       7) Memory anomaly              Enabled ..
                                          ..
38
39       Done
40 <!--NeedCopy-->
```

> 注
>
> Call Home が CIS に登録できない場合、アプライアンスはエラーメッセージを表示します。

SNMP アラームの有効化

Citrix ADC アプライアンスは、*SNMP* アラームと呼ばれる一連のエラー状態エンティティを提供します。SNMP アラームのエラー条件が満たされると、アプライアンスは SNMP トラップメッセージを生成し、設定されたトラップリスナーに送信されます。たとえば、SSL-CARD-FAILED アラームが有効の場合、トラップメッセージが生成され、トラップリスナーに送信されます。アプライアンスで SSL カードに障害が発生すると、トラップメッセージが送信されます。詳細については、SNMPを参照してください。

コマンドプロンプトで入力します。

```
enable snmp alarm <trapName>
```

```
show snmp alarm <trapName>
```

GUI を使用して **Call Home** を設定するには

GUI で Call Home 機能がデフォルトで有効になっているかどうかを確認するには

1. [設定] > [システム] >**[**プロファイル**]** に移動します。
2. 詳細ペインで、[高度な機能の構成] リンクをクリックします。
3. [拡張機能の設定（**Configure Advanced Features**）] ページで、[コールホーム（**Call Home**）] オプションが有効と表示されている必要があります。

GUI を使用して Call Home を有効にするには

1. [設定] > [システム] >**[**プロファイル**]** に移動します。
2. 詳細ペインで、[高度な機能の構成] リンクをクリックし、[**Callhome**] オプションを選択します。

GUI を使用してオプションのプロキシモード認証用に Call Home を設定するには

1. Call Home ページにアクセスするには、次の 2 つの方法のいずれかを使用できます。
   a)「システム」>「システム情報」に移動します。

b) [システム] > [診断] に移動します。
   i. 詳細ペインの **[**テクニカルサポートツール**]** で、**[Call Home]** を選択します。

2. [**Configure Call Home**] ページで、次のパラメータを設定します。
   a) モード。Call Home モードの動作。可能な種類：デフォルトの Citrix Service Provider（CSP）展開。

      注
      このオプションはユーザーが構成することはできません。モードは、Citrix ADC 展開のタイプに基づいて自動的に決定および設定されます。

   b) **E** メールアドレス。顧客サイトの連絡先管理者の電子メールアドレス。
   c) コールホームハートビートインターバル（日数）。Call Home ハートビート間のモニタリング間隔（日数）。最小値 = 1、最大値 = 7。
   d) **CallHome** を有効にします。Call Home 機能を有効または無効にして、Citrix サポートサーバー上のアプライアンス登録のステータスを表示します。
   e) プロキシモード。インターネットに接続できない場合は、プロキシモードを有効にして、オプションのプロキシパラメータを設定します。
   f) プロキシサーバー。プロキシサーバーを使用してプロキシモードを設定する場合は、サーバーの IP アドレスを指定します。
      i. プロキシサービス。プロキシサービスを使用してプロキシモードを設定する場合は、サービス名を指定してください。
      ii. **IP** アドレス。プロキシサーバの IP アドレス。
      iii. ポート。プロキシサーバーのポート番号。
      iv. プロキシ認証 **SSL** サービス。プロキシモード認証を提供するプロキシサービスの名前。
3. [**OK**] をクリックし、[完了] をクリックします。

GUI を使用してプロキシサーバー認証用の SSL サービスを構成するには

GUI を使用した SSL サービスの構成の詳細については、「SSL サービスの構成 」トピックを参照してください。

GUI を使用して CallHome の登録ステータスを確認するには

1. **Call Home** ページにアクセスするには、次の 2 つの方法のいずれかを使用できます。
   a)「システム」>「システム情報」に移動します。
   b) [システム] > [診断] に移動します。
      i. 詳細ペインの **[**テクニカルサポートツール**]** で、**[Call Home]** を選択します。
2. [**Call Home** の構成] ページの [**Citrix** アップロードサーバーへの登録] フィールドに登録ステータスが表示されます。

SNMP アラームを設定するには

1. **[**システム**]** > **[SNMP]** > **[**アラーム**]** に移動します。
2. 詳細ペインで、アラームを選択し、そのパラメータを設定します。
3. [**OK]** をクリックして [閉じる] をクリックします。

Citrix Service Provider（CSP）導入サポート

Citrix ADC サービスが VPX インスタンスに展開されている Citrix Service Provider（CSP）環境では、Call Home はライセンス固有の情報を監視および追跡し、情報を Citrix Insight Services（CIS）に安全に送信できます。次に、CIS は、アカウンティングの目的で、および CSP のお客様がライセンス使用状況を確認するために、License Usage Insights (LUI) ポータルに情報を送信します。現在、CSP 環境では VPX インスタンスでのみ Citrix ADC サービスをサポートしており、MPX または SDX アプライアンスではサポートしていません。VPX インスタンスは、スタンドアロンモードまたは高可用性モードで展開できます。

レポート作成ツール

October 7, 2021

Citrix® Citrix ADC® レポートツールを使用して、Citrix ADC のパフォーマンス統計データをレポートとして表示します。統計データは`nscollect`ユーティリティによって収集され、データベースに保存されます。ある期間にわたって特定のパフォーマンスデータを表示する場合、レポートツールはデータベースから指定されたデータを取り出し、グラフに表示します。

レポートは、グラフのコレクションです。レポートツールには、組み込みのレポートと、カスタムレポートを作成するオプションが用意されています。レポートでは、グラフを変更したり、新しいグラフを追加できます。また、データ収集ユーティリティ`nscollect`の操作を変更し、その操作を停止または開始することもできます。

レポート作成ツールの使用

レポートツールは、Citrix® Citrix ADC® アプライアンスからアクセスする Web ベースのインターフェイスです。レポート作成ツールを使用して、パフォーマンス統計データをグラフを含むレポートとして表示します。組み込みレポートを使用するだけでなく、カスタムレポートを作成することもできます。カスタムレポートはいつでも変更できます。レポートには、1～4 つのグラフがあります。最大 256 個のカスタムレポートを作成できます。任意の数のエンティティに対してカスタムレポートを作成できます。

レポート作成ツールを起動するには

1. 任意の Web ブラウザを使用して、Citrix ADC の IP アドレス（`http://10.102.29.170/`など）に接続します。[Web ログオン] 画面が表示されます。
2. ［ユーザー名］テキストボックスに、Citrix ADC に割り当てられているユーザー名を入力します。
3. 「パスワード」テキスト・ボックスにパスワードを入力します。
4. [開始場所] ドロップダウンリストボックスで、[レポート] を選択します。[ログイン] をクリックします。

次のスクリーンショットは、レポートツールバーとチャートツールバーを示しています。このツールバーは、このドキュメントで頻繁に参照されています。

図 1: レポートツールバー

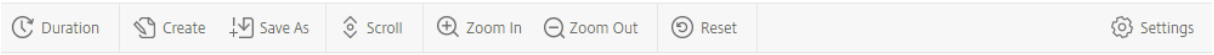
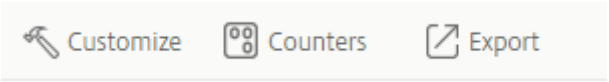


図 2: *[グラフ]* ツールバー



レポートの操作

Citrix ADC 上で構成されたさまざまな機能グループの統計を、指定した時間間隔でプロットおよび監視できます。レポートを使用すると、アプライアンスの動作をトラブルシューティングまたは分析できます。レポートには、組み込みレポートとカスタムレポートの 2 種類があります。組み込みレポートまたはカスタムレポートのレポートコンテンツは、グラフ形式または表形式で表示できます。グラフィカル・ビューは、最大 32 セットのデータ（カウンタ）を表示できる折れ線グラフ、面グラフ、棒グラフで構成されています。表形式ビューでは、列と行でデータが表示されます。このビューは、エラーカウンタのデバッグに役立ちます。

レポートツールに表示される既定のレポートは、CPU とメモリ使用量と HTTP 要求レート。既定のレポートビューを変更するには、既定のビューにするレポートを表示し、[既定のレポート] をクリックします。

レポートは、過去 1 時間、最終日、先週、先月、昨年に対して生成することも、期間をカスタマイズすることもできます。

レポートでは、次の操作を実行できます。

- データの表形式ビューとデータのグラフィカルビューを切り替えます。
- 棒グラフや折れ線グラフなどのグラフィカルな表示タイプを変更します。
- レポートのグラフをカスタマイズします。
- グラフを Excel コンマ区切り値 (CSV) ファイルとしてエクスポートします。
- ズームイン、ズームアウト、またはドラッグ操作 (スクロール) を使用して、グラフの詳細を表示します。
- ログオン時に表示する既定のレポートとしてレポートを設定します。
- カウンタを追加または削除します。
- レポートを印刷します。
- レポートを更新して、最新のパフォーマンス・データを表示します。

組み込みレポートの使用

レポートツールには、頻繁に表示されるデータの組み込みレポートが用意されています。組み込みレポートは、システム、ネットワーク、SSL、圧縮、統合キャッシュ、Citrix ADC ゲートウェイ、および Citrix ADC Application Firewall の機能グループで使用できます。既定では、組み込みレポートは最終日に表示されます。ただし、過去 1 時間、先週、先月、または昨年のレポートを表示できます。

> 注:
>
> 組み込みレポートへの変更を保存することはできませんが、変更した組み込みレポートをユーザー設定レポートとして保存することはできます。

組み込みレポートを表示するには

1. レポートツールの左側のウィンドウの [組み込みレポート] で、グループ (SSL など) を展開します。
2. レポートをクリックします ([**SSL] > [**すべてのバックエンド暗号**]** など**)**。

レポートの作成と削除

独自のカスタムレポートを作成し、ユーザ定義の名前で保存して再利用することができます。要件に基づいて、グループごとに異なるカウンタをプロットできます。最大 256 個のカスタムレポートを作成できます。

新しいレポートを作成することも、組み込みレポートをカスタムレポートとして保存することもできます。既定では、新しく作成されたカスタムレポートには [System Overview] という名前のグラフが 1 つ含まれています。このグラフには、最終日にプロットされた CPU 使用率カウンタが表示されます。レポート・ツールバーから間隔をカスタマイズし、データ・ソースとタイム・ゾーンを設定できます。

カスタムレポートを作成するには

1. レポートツールのレポートツールバーの [作成] をクリックします。既存のレポートに基づいて新しいカスタムレポートを作成する場合は、既存のレポートを開き、[名前を付けて保存] をクリックします。
2. [レポート名] ボックスに、カスタムレポートの名前を入力します。
3. 次のいずれかを行います:
    - レポートを既存のフォルダに追加するには、[作成先] または [保存先] で下向き矢印をクリックして既存のフォルダを選択し、[**OK**] をクリックします。
    - レポートを保存する新しいフォルダを作成するには、[フォルダを追加するにはクリックしてください] アイコンをクリックし、[フォルダ名] にフォルダの名前を入力します。次に、[作成場所] で、階層内の新しいフォルダを配置する場所を指定し、[**OK**] をクリックします。

> 注:
> 最大 128 個のフォルダを作成できます。

カスタムレポートを削除するには

1. レポートツールの左ペインで、[カスタムレポート] の横にある [クリックしてカスタムレポートを管理する] アイコンをクリックします。
2. 削除するレポートに対応するチェックボックスをオンにし、[削除] をクリックします。

> 注:
>
> フォルダを削除すると、そのフォルダのすべてのコンテンツが削除されます。

時間間隔の変更

既定では、組み込みレポートには最終日のデータが表示されます。ただし、組み込みレポートの時間間隔を変更する場合は、レポートをカスタムレポートとして保存できます。新しい間隔は、レポート内のすべてのグラフに適用されます。次の表では、時間間隔のオプションについて説明します。

時間間隔を変更するには

1. レポートツールの左側のウィンドウで、レポートをクリックします。
2. レポートツールバーの [期間] をクリックし、時間間隔をクリックします。

データソースとタイムゾーンの設定

さまざまなデータソースからデータを取得して、レポートに表示することができます。また、レポートのタイムゾーンを定義し、現在表示されているレポートの時間選択を、組み込みレポートを含むすべてのレポートに適用することもできます。

データソースとタイムゾーンを設定するには

1. レポートツールのレポートツールバーの [設定] をクリックします。
2. [設定] ダイアログボックスの [データソース] で、カウンタ情報を取得するデータソースを選択します。
3. 次のいずれかまたは両方の操作を行います。
    - グラフがプロットされる期間をツールに記憶させる場合は、[グラフの 時間選択を保存する] チェックボックスをオンにします。
    - レポートで Citrix ADC アプライアンスの時間設定を使用する場合は、［アプライアンスのタイムゾーンを使用する］チェックボックスをオンにします。

カスタムレポートのエクスポートとインポート

レポートをエクスポートすることで、他の Citrix ADC 管理者とレポートを共有できます。レポートをインポートすることもできます。

カスタムレポートをエクスポートまたはインポートするには

1. レポートツールの左側のウィンドウで、[カスタムレポート] の横にある **[カスタムレポートを管理するにはここをクリック]** アイコンをクリックします。

2. エクスポートまたはインポートするレポートに対応するチェックボックスをオンにし、[エクスポート] または [インポート] をクリックします。

   > 注:
   >
   > ファイルをエクスポートすると、.gz ファイル形式でエクスポートされます。

グラフの操作

チャートを使用して、カウンタまたはカウンタのグループをプロットおよび監視します。1 つのレポートに最大 4 つのグラフを含めることができます。各チャートでは、最大 32 個のカウンタをプロットできます。グラフでは、さまざまなグラフィカルな形式 (たとえば、面と棒グラフ) を使用できます。レポート内でグラフを上下に移動したり、グラフ内の各カウンタの色や表示をカスタマイズしたり、グラフを監視したくない場合は削除したりできます。

すべてのレポートグラフで、横軸は時間を表し、縦軸はカウンタの値を表します。

チャートの追加

レポートにグラフを追加すると、最後の 1 日の CPU 使用率カウンタがプロットされた状態で、[システム概要] グラフが表示されます。

> 注:
>
> 組み込みレポートにグラフを追加し、そのレポートを保持する場合は、レポートをカスタムレポートとして保存する必要があります。

レポートにグラフを追加するには、次の手順に従います。

レポートにグラフを追加するには

1. レポートツールの左側のウィンドウで、レポートをクリックします。
2. 新しいグラフを追加するグラフの下で、[追加] アイコンをクリックします。

チャートの変更

グラフを変更するには、統計情報を表示する機能グループを変更し、別のカウンタを選択します。

グラフを変更するには

1. レポートツールの左側のウィンドウで、レポートをクリックします。
2. 変更するグラフの下の [カウンタ] をクリックします。
3. 表示されるダイアログボックスの [タイトル] ボックスに、グラフの名前を入力します。
4. のプロットチャートの横で、次のいずれかを実行します。

- 統合キャッシュや圧縮などのグローバルカウンタのカウンタをプロットするには、[システムグローバル統計] をクリックします。
- 負荷分散や GSLB などのエンティティタイプのエンティティカウンタをプロットするには、[システムエンティティの統計] をクリックします。

5. [選択] グループで、目的のエンティティをクリックします。
6. [カウンタ] の [使用可能] で、プロットする 1 つ以上のカウンタ名をクリックし、[>] ボタンをクリックします。
7. 手順 4 で [システムエンティティの統計] を選択した場合は、[エンティティ] タブの [使用可能] で、プロットする 1 つまたは複数のエンティティインスタンス名をクリックし、[>] ボタンをクリックします。
8. [OK] をクリックします。

チャートの表示

グラフでプロットされるカウンタのグラフィック形式を指定できます。グラフは、折れ線グラフ、スプライングラフ、ステップ折れ線グラフ、散布図、面グラフ、棒グラフ、積み上げ面グラフ、積み上げ棒グラフとして表示できます。グラフのプロットエリア内では、ズームイン、ズームアウト、またはスクロールすることもできます。1 時間、1 日、1 週間、1 ヶ月、1 年、3 年間のすべてのデータソースを拡大または縮小できます。

グラフのビューをカスタマイズするその他のオプションには、グラフの軸のカスタマイズ、プロットエリアの背景とエッジの色変更、グリッドの色とサイズのカスタマイズ、グラフ内の各データセット (カウンタ) の表示のカスタマイズなどがあります。

データセット 1 などのデータセット番号は、グラフの下部に表示されるカウンタの順序に対応しています。たとえば、CPU 使用率とメモリ使用率がグラフの一番下に 1 番目と 2 番目の順序で表示されている場合、CPU 使用率はデータセット 1 に、メモリ使用率はデータセット 2 に等しくなります。

組み込みレポートを変更するたびに、変更を保持するために、レポートをカスタムレポートとして保存する必要があります。

グラフのグラフの種類を変更するには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、表示するグラフの下の [グラフ] ツールバーの [ユーザー設定] をクリックします。
3. [グラフ] タブの [分類] で、[プロットの種類] をクリックし、グラフに表示するグラフの種類をクリックします。グラフを 3D で表示する場合は、[3D を使用] チェックボックスをオンにします。

詳細データを含むグラフに再び焦点を合わせるには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、レポートツールバーの [拡大] をクリックし、次のいずれかまたは両方の操作を行います。
   - 特定のタイムウィンドウのデータを表示するようにチャートを再フォーカスするには、カーソルを開始時刻から終了時刻までドラッグします。たとえば、特定の日の 1 時間分のデータを表示できます。

- グラフにフォーカスしてデータポイントのデータを表示するには、ズームインするグラフを一度クリックして、詳細情報を取得します。

3. 詳細データを表示する時間範囲が整ったら、レポートツールバーの [表形式表示] をクリックします。表形式ビューでは、行と列に数値形式でデータが表示されます。

グラフの数値データを表示するには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、レポートツールバーの [表形式表示] をクリックします。グラフィカルビューに戻るには、[グラフィカルビュー] をクリックします。

> 注: グリッド線のノッチにカーソルを合わせると、数値データをグラフィカルビューで表示することもできます。

グラフで時間をスクロールするには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、レポートツールバーの [スクロール] をクリックし、グラフ内をクリックし、新しい期間のデータを表示する方向にカーソルをドラッグします。たとえば、過去のデータを表示する場合は、左にドラッグします。

グラフの背景色とテキストの色を変更するには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、軸をカスタマイズするグラフの下の [カスタマイズ] をクリックします。
3. [グラフ] タブの **[**カテゴリ**]** で、次の 1 つまたは複数をクリックします。
   - 背景色を変更するには、[背景色**]** をクリックし、色、透明度、および効果のオプションを選択します。
   - テキストの色を変更するには、[テキストの 色**]** をクリックし、色、透明度、および効果のオプションを選択します。

グラフの軸をカスタマイズするには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、軸をカスタマイズするグラフの下の [カスタマイズ] をクリックします。
3. [グラフ] タブの [カテゴリ] で、次のいずれかまたは複数のをクリックします。
   - 左の Y 軸のスケールを変更するには、[左の **Y** 軸] をクリックし、目的のスケールを選択します。
   - 右の Y 軸のスケールを変更するには、[右の Y 軸] をクリックし、プロットするデータセットで日付セットを選択し、目的のスケールを選択します。

     > 注:
     > データセット 1 などのデータセット番号は、グラフの下部に表示されるカウンタの順序に対応しています。たとえば、CPU 使用率とメモリ使用率がグラフの一番下に 1 番目と 2 番目の順序で表

> 示されている場合、CPU 使用率はデータセット 1 に、メモリ使用率はデータセット 2 に等しくなります。

- 各データセットを独自の非表示の Y 軸にプロットするには、[複数の軸] をクリックし、[有効] をクリックします。

グラフのプロットエリアの背景色、エッジの色、およびグリッド線を変更するには

1. レポートツールの左ペインで、レポートを選択します。
2. 右ペインで、プロットエリアをカスタマイズするグラフの下の [カスタマイズ] をクリックします。
3. [印刷領域] タブの [カテゴリ] で、次の 1 つまたは複数をクリックします。
   - グラフの背景色とエッジの色を変更するには、[背景色] と [エッジの色] をクリックし、色、透明度、および効果のオプションを選択します。
   - グラフの水平グリッドまたは垂直グリッドを変更するには、[水平グリッド] または [垂直グリッド] をクリックし、グリッド、グリッドの幅、グリッドの色、透明度、および効果を表示するオプションを選択します。

データセットの色とグラフタイプを変更するには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、データセット (カウンタ) の表示をカスタマイズするグラフの下の [カスタマイズ] をクリックします。
3. [データセット] タブの [データセットの選択] で、グラフィック表示をカスタマイズするデータセット (カウンタ) を選択します。
   注: データセット 1 などのデータセット番号は、グラフの下部に表示されるカウンタの順序に対応しています。
   たとえば、
   CPU 使用率と
   メモリ使用率がグラフの一番下に 1 番目と 2 番目の順序で表示されている場合、
   CPU 使用率は
   データセット 1 に、
   メモリ使用率は
   データセット 2 に等しくなります。
4. [カテゴリ] で、次のいずれかの操作を行います。
   - 背景色を変更するには、[色] をクリックし、色、透明度、および効果のオプションを選択します。
   - グラフタイプを変更するには、[プロットタイプ] をクリックし、データセットに表示するグラフタイプを選択します。グラフを 3D として表示する場合は、[3D を使用] チェックボックスをオンにします。

Excel へのグラフデータのエクスポート

さらにデータを分析するために、グラフをカンマ区切り値 (CSV) 形式で Excel にエクスポートできます。

グラフデータを Excel にエクスポートするには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のウィンドウで、Excel にエクスポートするデータを含むグラフの下の [エクスポート] をクリックします。

グラフの削除

グラフを使用しない場合は、レポートからグラフを削除できます。カスタムレポートからのみグラフを完全に削除できます。組み込みレポートからグラフを削除し、変更を保持する場合は、レポートをユーザー設定レポートとして保存する必要があります。

グラフを削除するには

1. レポートツールの左ペインで、レポートを選択します。
2. 右側のペインで、削除するグラフの下の [削除] アイコンをクリックします。

例

先週の **CPU** 使用率とメモリ使用率の傾向レポートを表示するには

1. レポートツールの左側のウィンドウで、[組み込みレポート] の下の [システム] を展開します。
2. レポートの [CPU 対] をクリックします。メモリ使用量と HTTP 要求レート。
3. 右側のウィンドウで、レポートツールバーの [期間] をクリックし、[先週] をクリックします。

先週の **2** つのインターフェイス間で受信されたバイトレートと送信バイトレートを比較するには

1. 右側のウィンドウで、レポートツールバーの [作成] をクリックします。
2. [レポート名] ボックスに、カスタムレポートの名前 (Custom_Interfaces など) を入力し、[**OK**] をクリックします。このレポートは、デフォルトの [システム概要] グラフで作成されます。このグラフには、過去 1 時間にプロットされた CPU 使用率カウンタが表示されます。
3. [システムの概要] の [グラフツールバー] で、[カウンタ] をクリックします。
4. カウンタ選択ウィンドウの [タイトル] に、グラフの名前 (たとえば、Interfaces バイトデータ) を入力します。
5. [グラフのプロット] で、[システムエンティティの統計] をクリックし、[グループの選択] で [インターフェイス] を選択します。
6. [エンティティ] タブで、プロットする 1 つまたは複数のインタフェース名 (1/1 と 1/2 など) をクリックし、[>] ボタンをクリックします。
7. [カウンタ] タブで、[受信バイト数 (レート)] および [送信バイト数 (レート)] をクリックし、[>] ボタンをクリックします。
8. [**OK**] をクリックします。
9. レポートツールバーの [期間] をクリックし、[先週] をクリックします。

データ収集ユーティリティの停止と起動

データ収集ユーティリティnscollectは、Citrix ADC を起動すると自動的に実行されます。このユーティリティは、アプリケーション・パフォーマンス・データを取得し、ADC 上のデータ・ソース形式で保存します。最大 32 個のデータソースを作成できます。既定のデータソースは/var/log/db/**default**です。

データ・コレクション・ユーティリティは、グローバル・カウンタとエンティティ固有のカウンタのデータベースを作成し、このデータを使用してレポートを生成します。グローバルカウンタデータベースは、/var/log/db/<DataSourceName>で作成されます。エンティティ固有のデータベースは、Citrix ADC で構成されたエンティティに基づいて作成され、/var/log/db/<DataSourceName/EntityNameDB>のエンティティタイプごとに個別のフォルダが作成されます。

nscollectは 5 分ごとに 1 回データを取得します。データは、1 日間、1 時間あたり、過去 30 日間、1 日間 3 年間毎日保持されます。

データが正確に更新されない場合や、レポートに破損したデータが表示される場合は、データ収集ユーティリティを停止して再起動する必要があります。

nscollect を中止するには

コマンドプロンプトで入力します。

```
/netscaler/nscollect stop
```

Citrix ADC への現在の **SSH** セッションで **nscollect** を開始するには、以下の手順に従ってください。

コマンドプロンプトで入力します。

```
/netscaler/nscollect start
```

ローカル・システムで **nscollect** を起動するには、次の手順で行います。

コマンドプロンプトで入力します。

```
/netscaler/nscollect start &
```

CloudBridge Connector

October 7, 2021

注：現在の Citrix ADC 1000V リリースでは、この機能はサポートされていません。

Citrix ADC アプライアンスの CloudBridge Connector 機能は、エンタープライズデータセンターを外部クラウドおよびホスティング環境に接続し、クラウドをエンタープライズネットワークのセキュアな拡張にします。クラウドでホストされるアプリケーションは、1 つの連続したエンタープライズネットワークで実行されているかのように見えます。Citrix CloudBridge Connector を使用すると、クラウドプロバイダーが提供する容量と効率でデータセンターを強化できます。

CloudBridge Connector を使用すると、アプリケーションをクラウドに移行して、コストを削減し、信頼性を高めることができます。

データセンターとクラウド間で CloudBridge Connector を使用するだけでなく、これを使用して 2 つのデータセンターを接続し、大容量のセキュアで高速なリンクを実現できます。

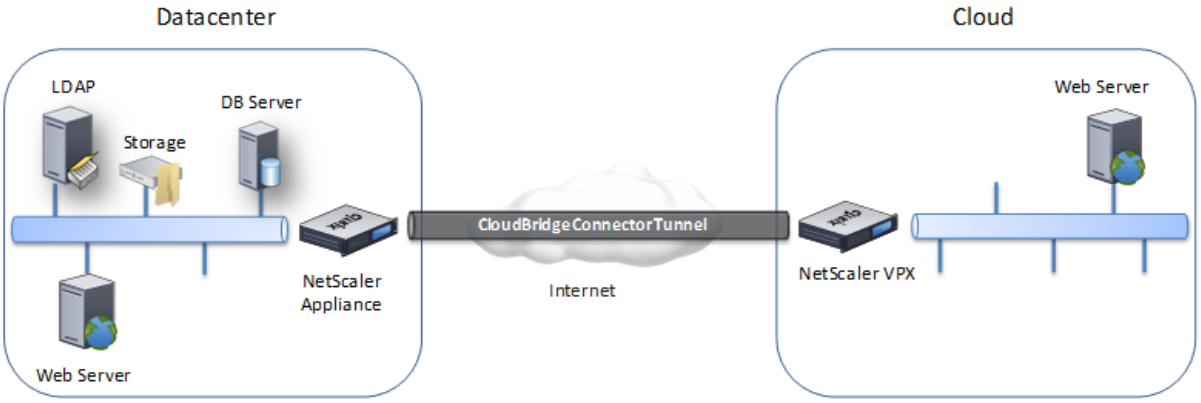
CloudBridge Connector について

Citrix CloudBridge Connector ソリューションを実装するには、CloudBridge Connector トンネルと呼ばれるトンネルを設定して、データセンターを別のデータセンターまたは外部クラウドに接続します。

データセンターを別のデータセンターに接続するには、各データセンターに 1 つずつ、2 つの Citrix ADC アプライアンス間に CloudBridge Connector トンネルを設定します。

データセンターを外部クラウド（Amazon AWS クラウドなど）に接続するには、データセンター内の Citrix ADC アプライアンスとクラウドに存在する仮想アプライアンス（VPX）の間に CloudBridge Connector トンネルを設定します。リモートエンドポイントは、CloudBridge Connector またはプレミアムライセンスを持つ Citrix ADC VPX にすることができます。

次の図は、データセンターと外部クラウドの間に設定された CloudBridge Connector トンネルを示しています。



CloudBridge Connector トンネルがセットアップされるアプライアンスは、CloudBridge Connector トンネルの エンドポイントまたは ピアと呼ばれます。

CloudBridge Connector トンネルは、次のプロトコルを使用します。

- 汎用ルーティングカプセル化（GRE）プロトコル
- オープン標準の IPSec プロトコルスイート（トランスポートモード）

GRE プロトコルは、さまざまなネットワークプロトコルからのパケットをカプセル化し、別のプロトコル経由で転送するメカニズムを提供します。GRE は次の目的で使用されます。

- 非 IP プロトコルおよびルーティング不可能なプロトコルを実行するネットワークを接続します。
- ワイドエリアネットワーク (WAN) を経由するブリッジ。

- 異なるネットワーク上で変更せずに送信する必要があるあらゆる種類のトラフィックに対して、トランスポートトンネルを作成します。

GRE プロトコルは、GRE ヘッダーと GRE IP ヘッダーをパケットに追加することによって、パケットをカプセル化します。

インターネットプロトコルセキュリティ（IPSec）プロトコルスイートは、CloudBridge Connector トンネル内のピア間の通信をセキュリティで保護します。

CloudBridge Connector トンネルでは、IPSec によって次のことが保証されます。

- データの整合性
- データ発信元認証
- データの機密性 (暗号化)
- リプレイ攻撃に対する保護

IPSec は、GRE カプセル化パケットが暗号化されるトランスポートモードを使用します。暗号化は、カプセル化セキュリティペイロード (ESP) プロトコルによって行われます。ESP プロトコルは、HMAC ハッシュ関数を使用してパケットの整合性を保証し、暗号化アルゴリズムを使用して機密性を保証します。パケットが暗号化され、HMAC が計算されると、ESP ヘッダーが生成されます。ESP ヘッダーは GRE IP ヘッダーの後に挿入され、ESP トレーラーは暗号化されたペイロードの最後に挿入されます。

CloudBridge Connector トンネル内のピアは、次のように、インターネットキー交換バージョン（IKE）プロトコル（IPSec プロトコルスイートの一部）を使用して、セキュアな通信をネゴシエートします。

- 2 つのピアは、次の認証方法のいずれかを使用して、相互に認証を行います。
  - 事前共有キー認証。事前共有キーと呼ばれるテキスト文字列は、各ピアに手動で設定されます。ピアの事前共有キーは、認証のために相互に照合されます。したがって、認証を成功させるには、各ピアで同じ事前共有キーを設定する必要があります。
  - デジタル証明書認証。発信側（送信者）ピアは、秘密キーを使用してメッセージ交換データに署名し、もう一方の受信側ピアは送信者の公開キーを使用して署名を検証します。通常、公開キーは X.509v3 証明書を含むメッセージで交換されます。この証明書は、証明書に示されているピアのアイデンティティが特定の公開キーに関連付けられていることを保証します。
- ピアは、次の上で合意に達するために交渉します。
  - 暗号化アルゴリズム。
  - 1 つのピアでデータを暗号化し、もう 1 つのピアでデータを復号化する暗号化キー。

セキュリティプロトコル、暗号化アルゴリズム、および暗号化キーに関する本契約は、セキュリティアソシエーション (SA) と呼ばれます。SA は一方向（単方向）です。たとえば、CB1 と CB2 の 2 つのピアがコネクタトンネルを介して通信している場合、CB1 には 2 つのセキュリティアソシエーションがあります。一方の SA は発信パケットの処理に使用され、もう一方の SA は着信パケットの処理に使用されます。

SA は、ライフタイムと呼ばれる指定された時間が経過すると期限切れになります。2 つのピアは、インターネットキー交換（IKE）プロトコル（IPSec プロトコルスイートの一部）を使用して、新しい暗号化キーをネゴシエートし、新しい SA を確立します。限られた寿命の目的は、攻撃者が鍵をクラックするのを防ぐことです。

次の表に、Citrix ADC アプライアンスでサポートされる IPSec のプロキシの一部を示します。

IPSec のプロパティ	サポートされているタイプ
IKE のバージョン	V1, V2
IKE DH グループ	Citrix ADC アプライアンスは、IKEv1 と IKEv2 の両方で DH グループ 2（1024 ビット MODP アルゴリズム）のみをサポートします。
IKE 認証方法	事前共有キー認証、デジタル証明書認証
暗号化アルゴリズム	AES（128 ビット）、AES256（256 ビット）、3DES
ハッシュアルゴリズム	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5

CloudBridge Connector トンネルのモニタリング

October 7, 2021

CloudBridge Connector トンネルのパフォーマンスを監視するための統計情報を表示できます。Citrix ADC アプライアンスで CloudBridge Connector のトンネルの統計情報を表示するには、GUI または Citrix ADC のコマンドラインを使用します。

次の表は、Citrix ADC アプライアンスで CloudBridge Connector トンネルを監視するための統計カウンタの一覧です。

統計カウンタ	**Specifies**
受信バイト	アプライアンスが最後に起動されてから、設定されたすべての CloudBridge Connector トンネルを介して Citrix ADC アプライアンスが受信した合計バイト数。
送信バイト	アプライアンスが最後に起動されてから、設定されたすべての CloudBridge Connector トンネルを介して Citrix ADC アプライアンスが送信した合計バイト数。

統計カウンタ	**Specifies**
受信済みパケット	アプライアンスが最後に起動されてから、設定されたすべての CloudBridge Connector トンネルを経由して Citrix ADC アプライアンスが受信したパケットの合計数。
送信済みパケット	アプライアンスが最後に起動されてから、設定されたすべての CloudBridge Connector トンネルを経由して Citrix ADC アプライアンスが送信したパケットの総数。
受信バイト数	すべての設定済み CloudBridge Connector トンネルを介して Citrix ADC アプライアンスが受信した 1 秒あたりのバイト数。
送信バイトレート	すべての設定済み CloudBridge Connector トンネルを介して Citrix ADC アプライアンスが送信した 1 秒あたりのバイト数
受信パケットレート	すべての設定済み CloudBridge Connector トンネルを介して Citrix ADC アプライアンスが受信した 1 秒あたりのバイト数
送信パケットレート	すべての設定済み CloudBridge Connector トンネルを介して Citrix ADC アプライアンスが受信した 1 秒あたりのバイト数

Citrix ADC アプライアンスが再起動されると、これらのカウンタはすべて 0 にリセットされます。次のフェーズでは増分しません。

- 設定済みの CloudBridge Connector トンネルでのインターネットキー交換（IKE）認証（事前共有キー）フェーズ。
- 設定済みの CloudBridge Connector トンネルでの IKE セキュリティアソシエーション (SA) 確立フェーズ。

Citrix ADC コマンドラインを使用して CloudBridge Connector のトンネルの統計情報を表示するには

コマンドプロンプトで入力します。

- 統計情報カウンタ

GUI を使用して CloudBridge Connector のトンネルの統計情報を表示するには

1. Web ブラウザーを使用して Citrix ADC アプライアンスの IP アドレスに接続して、GUI にアクセスします。
2. [設定] タブで、 [システム] > [**CloudBridge Connector**] に移動します。
3. [CloudBridge Connector] ページで、[**CloudBridge Connector** の作成/監視] をクリックします。「**IPSec Bytes** 」および「**IPSec Packets** 」チャートには、Citrix ADC アプライアンスで設定されているすべての

CloudBridge Connector トンネルのバイト受信レート、バイト送信レート、パケット受信レート、およびパケット送信レートが表示されます。

```
    > stat ipsec counters
    Secure tunnel(s) summary
                        Rate (/s)        Total
    Bytes Received    0    2811248
    Bytes Sent            0    157460630
    Packets Received    0    56787
    Packets Sent        0    200910
    Done
    >
<!--NeedCopy-->
```

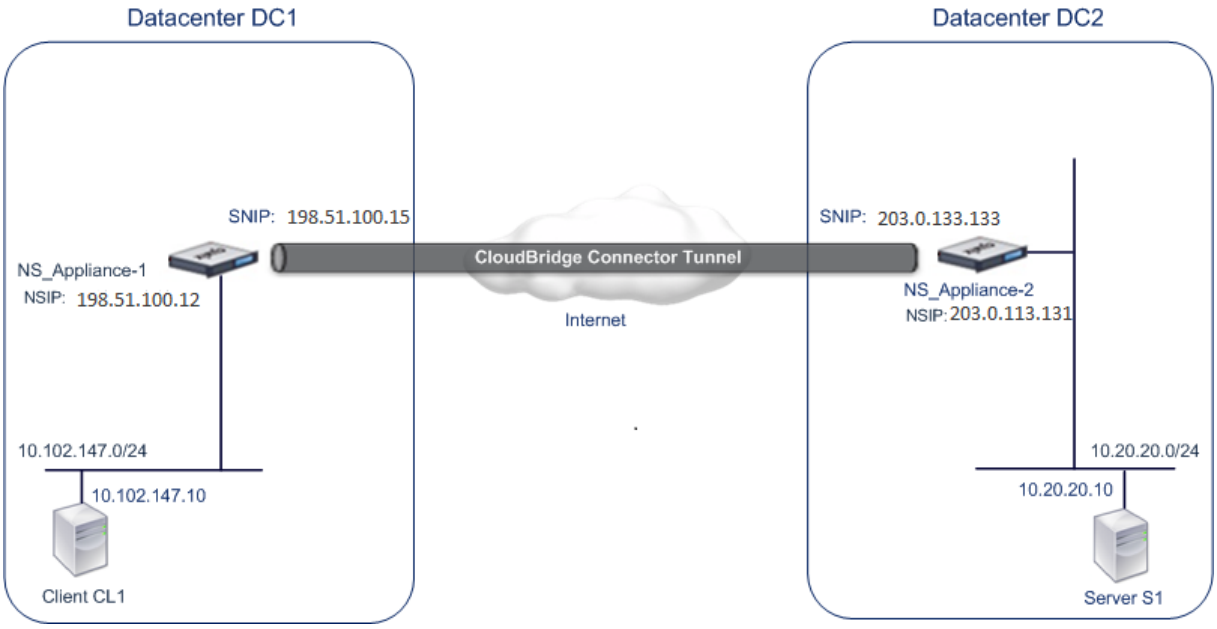
2 つのデータセンター間の CloudBridge Connector トンネルの設定

October 7, 2021

2 つの異なるデータセンター間で CloudBridge Connector トンネルを構成して、再構成せずにネットワークを拡張し、2 つのデータセンターの機能を活用できます。地理的に分離された 2 つのデータセンター間の CloudBridge Connector トンネルにより、冗長性を実装し、セットアップを障害から守ることができます。CloudBridge Connector トンネルは、データセンター全体でインフラストラクチャとリソースの最適な利用を実現します。2 つのデータセンターで利用可能なアプリケーションは、ユーザーに対してローカルとして表示されます。

データセンターを別のデータセンターに接続するには、あるデータセンターの Citrix ADC アプライアンスと、他のデータセンターの Citrix ADC アプライアンスとの間に CloudBridge Connector トンネルを設定します。

データセンター間の CloudBridge Connector トンネルの図のように、データセンター DC1 の Citrix ADC アプライアンスの NS_ アプライアンス-1 とデータセンター DC2 の Citrix ADC アプライアンスの NS_ アプライアンス-2 の間に CloudBridge Connector トンネルが設定されている例を考えてみます。



NS_ アプライアンス-1 および NS_ アプライアンス-2 は、L2 および L3 モードで機能します。これにより、データセンター DC1 と DC2 のプライベートネットワーク間の通信が可能になります。L3 モードでは、NS_ アプライアンス 1 と NS_ アプライアンス 2 は、データセンター DC1 のクライアント CL1 とデータセンター DC2 のサーバー S1 との間の通信を CloudBridge Connector のトンネルを介して有効にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

クライアント CL1 とサーバ S1 は異なるプライベートネットワーク上にあるため、NS_Appliance-1 と NS_Appliance-2 で L3 モードが有効になり、ルートは次のように更新されます。

- CL1 には、S1 に到達するための NS_ アプライアンス 1 へのルートがあります。
- NS_ アプライアンス-1 には、S1 に到達するための NS_ アプライアンス-2 へのルートがあります。
- S1 には、CL1 に到達するための NS_ アプライアンス 2 へのルートがあります。
- NS_ アプライアンス 2 には、CL1 に到達するための NS_ アプライアンス 1 へのルートがあります。

次の表に、データセンター DC1 の Citrix ADC アプライアンス NS_ アプライアンス-1 の設定を示します。

次の表に、データセンター DC2 の Citrix ADC アプライアンス NS_ アプライアンス 2 の設定を示します。

エンティティ	Name	詳細
NSIP アドレス		198.51.100.12
SNIP アドレス		198.51.100.15

エンティティ	Name	詳細
CloudBridge Connector tunnel	Cloud_Connector_DC1-DC2	1. CloudBridge Connector トンネルのローカルエンドポイント IP アドレス:198.51.100.15, 2. CloudBridge Connector トンネルのリモートエンドポイント IP アドレス:203.0.113.133。GRE トンネル詳細名 = クラウド接続 _DC1-DC2、IPSec プロファイルの詳細名 = クラウド接続 _DC1-DC2、暗号化アルゴリズム = AES、ハッシュアルゴリズム = HMAC SHA1

CloudBridge Connector トンネルを設定する際に考慮すべきポイント

CloudBridge Connector トンネルを設定する前に、次のタスクが完了していることを確認します。

1. 2 つのデータセンターそれぞれに Citrix ADC アプライアンスを展開してセットアップします。
2. CloudBridge Connector トンネルのエンドポイントの IP アドレスが相互にアクセス可能であることを確認します。

設定手順

あるデータセンターにある Citrix ADC アプライアンスと、他のデータセンターにある別の Citrix ADC アプライアンスとの間に CloudBridge Connector トンネルを設定するには、いずれかの Citrix ADC アプライアンスの GUI またはコマンドラインインターフェイスを使用します。

GUI を使用すると、最初の Citrix ADC アプライアンスで作成された CloudBridge Connector トンネル構成が、CloudBridge Connector トンネルのもう一方のエンドポイント（もう一方の Citrix ADC アプライアンス）に自動的にプッシュされます。したがって、対応する CloudBridge Connector トンネル構成を作成するために、他の Citrix ADC アプライアンスの GUI にアクセスする必要はありません。

各 Citrix ADC アプライアンスの CloudBridge Connector トンネル構成は、次のエンティティで構成されています。

- **IPSec** プロファイル：IPSec プロファイルエンティティは、CloudBridge Connector トンネル内の IPSec プロトコルで使用される、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPSec プロトコルパラメータを指定します。
- **GRE** トンネル：IP トンネルは、ローカル IP アドレス（ローカル Citrix ADC アプライアンスで設定されたパブリック SNIP アドレス）、リモート IP アドレス（リモート Citrix ADC アプライアンスで設定されたパブリ

ック SNIP アドレス)、CloudBridge Connector トンネルのセットアップに使用するプロトコル（GRE)、および IPSec を指定します。縦断図形。

- **PBR** ルールを作成し、**IP** トンネルを関連付ける：PBR エンティティは、一連の条件と IP トンネルエンティティを指定します。送信元 IP アドレスの範囲と宛先 IP の範囲は、PBR エンティティの条件です。トラフィックが CloudBridge Connector トンネルを通過するサブネットを指定するには、送信元 IP アドレスの範囲と宛先 IP アドレスの範囲を設定する必要があります。たとえば、最初のデータセンターのサブネット上のクライアントから発信され、2 番目のデータセンターのサブネット上のサーバーを宛先とする要求パケットがあるとします。このパケットが、最初のデータセンター内の Citrix ADC アプライアンス上の PBR エンティティの送信元および宛先 IP アドレス範囲と一致する場合、そのパケットは PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

コマンドラインインターフェイスを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> [-ikeVersion (V1 | V2)] [-encAlgo (AES | 3DES)...] [-hashAlgo <hashAlgo\> ...] [-lifetime <positive_integer>] (-psk | (-publickey<string> -privatekey <string>-peerPublicKey <string>))[-livenessCheckInterval <positive_intege>][-replayWindowSize \<positive_integer>] [-ikeRetryInterval <positive_integer>] [-retransmissiontime <positive_integer>]`
- `show ipsec profile <name>`

コマンドラインインターフェイスを使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

コマンドラインインターフェイスを使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

コマンドプロンプトで入力します。

- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

例

```
1  add ipsec profile Cloud_Connector_DC1-DC2  -encAlgo AES -hashAlgo
      HMAC_SHA1
```

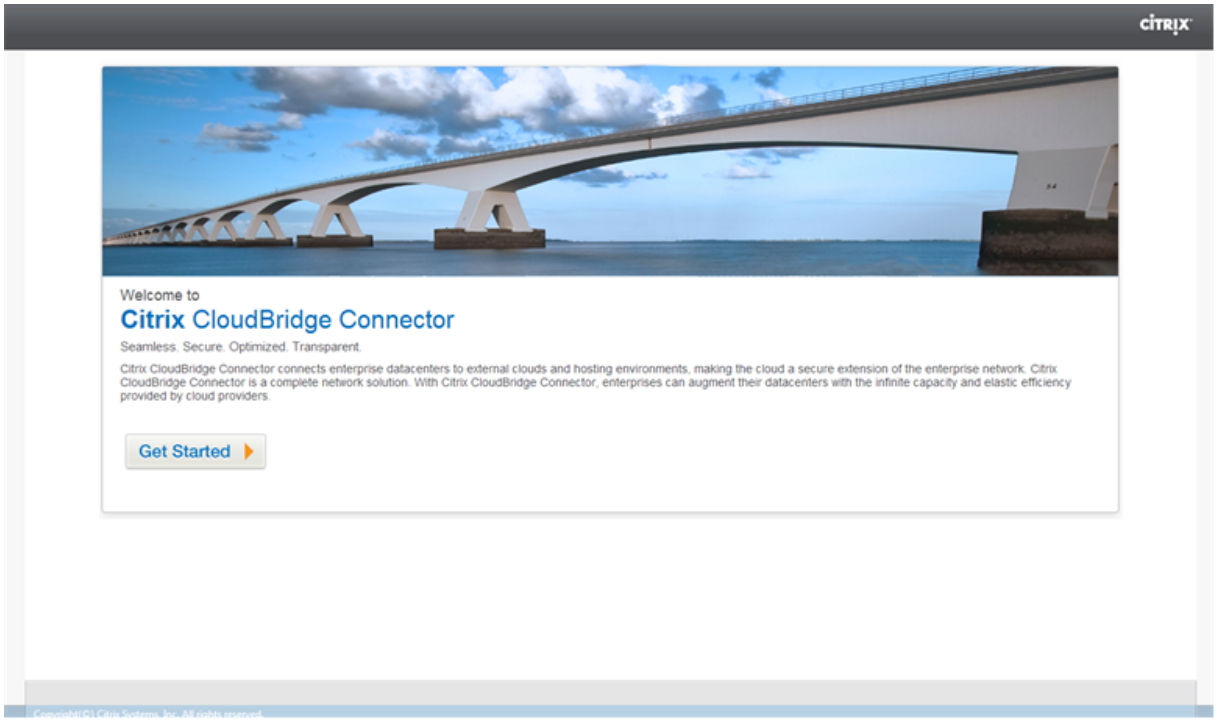


```
    Done
    > add ipTunnel Cloud_Connector_DC1-DC2 203.0.113.133
       255.255.255.255 198.51.100.15 -protocol GRE -ipsecProfileName
       Cloud_Connector_DC1-DC2

    Done
    > add ns pbr PBR-DC1-DC2 ALLOW -srcIP 198.51.100.15 -destIP
       203.0.113.133 ipTunnel Cloud_Connector_DC1-DC2

    Done
    > apply ns pbrs

    Done
<!--NeedCopy-->
```

GUI を使用して Citrix ADC アプライアンスで CloudBridge Connector トンネルを構成するには

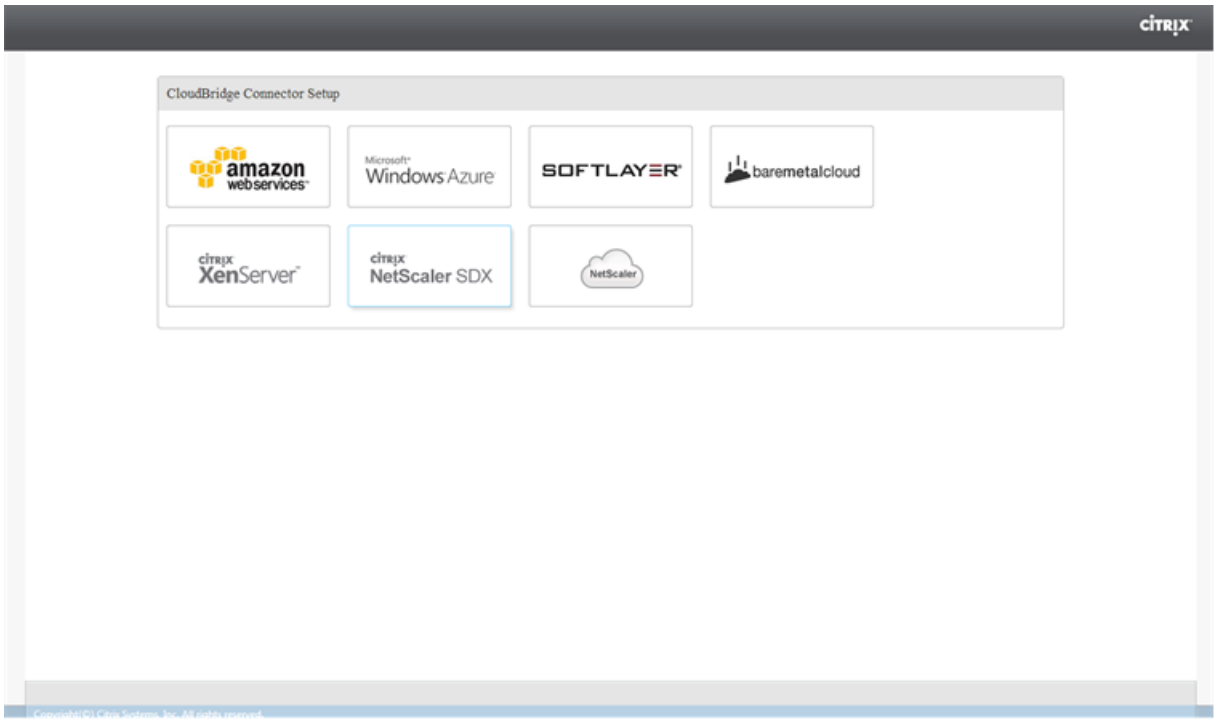
1. Web ブラウザのアドレス行に、Citrix ADC アプライアンスの NSIP アドレスを入力します。
2. アプライアンスのアカウント認証情報を使用して、Citrix ADC アプライアンスの GUI にログオンします。
3. [システム] > [**CloudBridge Connector**] に移動します。
4. 右側のペインの [はじめに] で、[**CloudBridge** の作成/監視] をクリックします。

   アプライアンスで CloudBridge Connector トンネルを初めて設定すると、ようこそ画面が表示されます。
5. ようこそ画面で、[は じめに] をクリックします。



> 注:
>
> Citrix ADC アプライアンスで CloudBridge Connector トンネルをすでに構成している場合は、ようこそ画面が表示されないため、［開始］をクリックしないでください。

1. **CloudBridge Connector** のセットアップペインで、［**Citrix ADC**］をクリックします。



1. Citrix ADC ペインで、リモート Citrix ADC アプライアンスのアカウント資格情報を入力します。［続行］をクリックします。
2. **CloudBridge Connector** 設定ペインで、次のパラメータを設定します。
   - **CloudBridge Connector** 名— ローカルアプライアンス上の CloudBridge Connector 設定の名前。ASCII アルファベットまたはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。CloudBridge Connector の設定が作成された後は変更できません。
3. [ローカル設定] で、次のパラメータを設定します。
   - サブネット **IP**：CloudBridge Connector トンネルのローカルエンドポイントの IP アドレス。
4. [リモート設定] で、次のパラメータを設定します。
   - サブネット **IP**：CloudBridge Connector トンネルのピアエンドポイントの IP アドレス。
5. [**PBR** 設定] で、次のパラメータを設定します。
   - 演算：等しい（=）または等しくない（! =) 論理演算子。
   - 送信元 **IP** 低：発信 IPv4 パケットの送信元 IP アドレスと照合する最小の送信元 IP アドレス。

- 送信元 **IP High**：発信 IPv4 パケットの送信元 IP アドレスと照合する最大送信元 IP アドレス。
- 演算：等しい（=）または等しくない（! =) 論理演算子。
- 宛先 IP Low*：発信 IPv4 パケットの宛先 IP アドレスと照合する最小の宛先 IP アドレス。
- 宛先 **IP** 高：発信 IPv4 パケットの宛先 IP アドレスと照合する最大の宛先 IP アドレス。

6. （オプション）[セキュリティ設定] で、CloudBridge Connector トンネルに次の IPSec プロトコルパラメータを設定します。
   - 暗号化アルゴリズム：CloudBridge トンネル内の IPSec プロトコルで使用される暗号化アルゴリズム。
   - ハッシュアルゴリズム：CloudBridge トンネル内の IPSec プロトコルで使用されるハッシュアルゴリズム。
   - [**Key**]：相互認証に 2 つのピアが使用する次の IPSec 認証方式のいずれかを選択します。
     - キーの自動生成：ローカルアプライアンスによって自動的に生成される PSK（事前共有キー）と呼ばれるテキスト文字列に基づく認証。ピアの PSK キーは、認証のために相互に照合されます。
     - 特定のキー：手動で入力した PSK に基づく認証。ピアの PSK は、認証のために相互に照合されます。
       * 事前共有セキュリティキー：事前共有キーベースの認証用に入力されたテキストストリング。
     - 証明書のアップロード：デジタル証明書に基づく認証。
       * 公開キー：IPSec セキュリティアソシエーションを確立する前に、ローカルの Citrix ADC アプライアンスをピアに対して認証するために使用するローカルデジタル証明書。同じ証明書が存在し、ピアの Peer Public Key パラメータに設定する必要があります。
       * 秘密キー：ローカルデジタル証明書の秘密キー。
       * ピア公開キー：ピアのデジタル証明書。IPSec セキュリティアソシエーションを確立する前に、ローカルエンドポイントに対してピアを認証するために使用されます。同じ証明書が存在し、ピアの Public key パラメータに設定する必要があります。
7. ［完了］をクリックします。

両方の Citrix ADC アプライアンスの新しい CloudBridge Connector トンネル構成が、それぞれの GUI の［ホーム］タブに表示されます。CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge Connector] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルのモニタリング

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

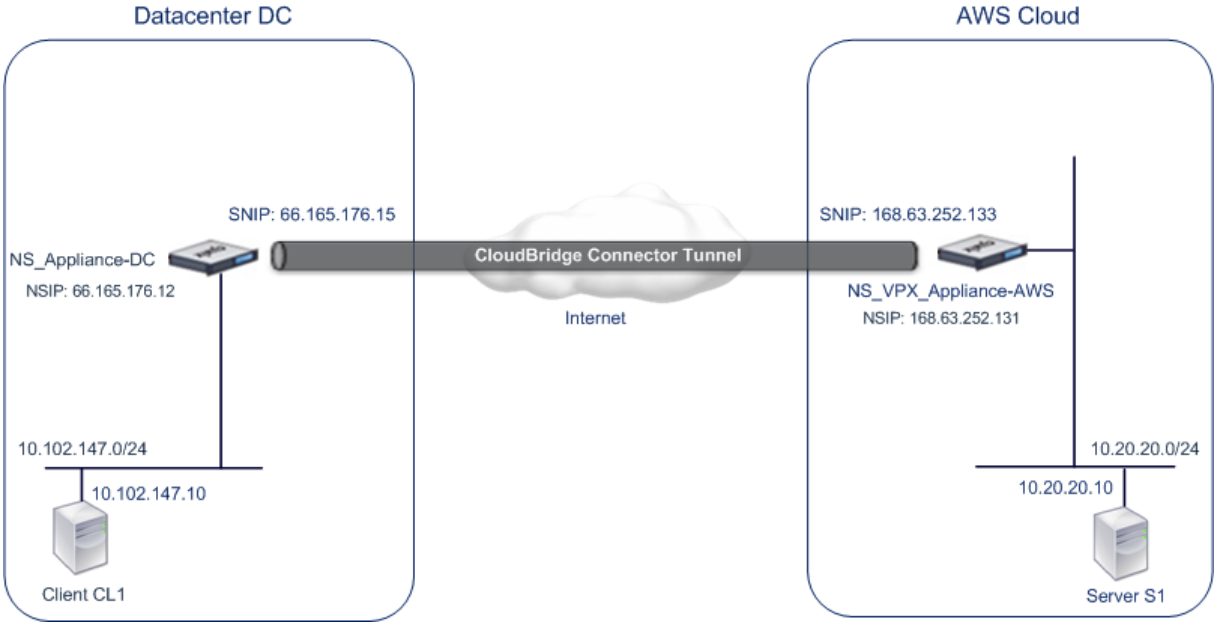
データセンターと **AWS** クラウド間の **CloudBridge Connector** の設定

October 7, 2021

データセンターと AWS クラウドの間に CloudBridge Connector トンネルを設定して、データセンターと AWS クラウドのインフラストラクチャとコンピューティング機能を活用できます。AWS を使用すると、初期設備投資や拡張ネットワークインフラストラクチャの維持コストなしで、ネットワークを拡張できます。必要に応じて、インフラストラクチャをスケールアップまたはスケールダウンできます。たとえば、需要が増えたときに、より多くのサーバー機能をリースできます。

データセンターを AWS クラウドに接続するには、データセンターに存在する Citrix ADC アプライアンスと、AWS クラウドに存在する Citrix ADC 仮想アプライアンス（VPX）の間に CloudBridge Connector トンネルを設定します。

データセンターと Amazon AWS クラウド間の CloudBridge Connector トンネルの図のように、データセンターの DC と Citrix ADC 仮想アプライアンス (VPX) NS_VPX_ アプライアンス-AWS の間で CloudBridge Connector トンネルが設定されている例を考えてみます。



NS_ アプライアンス-DC と NS_VPX_ アプライアンス-AWS の両方が L3 モードで機能します。これにより、データセンター DC のプライベートネットワークと AWS クラウド間の通信が可能になります。NS_ アプライアンス DC および NS_VPX_ アプライアンス-AWS は、データセンター DC のクライアント CL1 と AWS クラウドのサーバー S1 との間の通信を CloudBridge Connector のトンネルを介して有効にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

> 注:
>
> AWS は L2 モードをサポートしていないため、両方のエンドポイントで L3 モードのみを有効にする必要があ

> ります。

CL1 と S1 間の通信を正しく行うため、NS_ アプライアンス DC および NS_VPX_ アプライアンス AWS で L3 モードが有効になり、ルートは次のように更新されます。

- CL1 には、S1 に到達するための NS_ アプライアンス DC へのルートがあります。
- NS_ アプライアンス-DC は、S1 に到達するための NS_VPX_ アプライアンス-AWS へのルートを持っています。
- S1 には、CL1 に到達するための NS_VPX_ アプライアンス-AWS へのルートが必要です。
- NS_VPX_ アプライアンス-AWS には、CL1 に到達するための NS_ アプライアンス-DC へのルートがあります。

次の表に、データセンター DC における Citrix ADC アプライアンス NS_ アプライアンス DC の設定を示します。

エンティティ	Name	詳細
NSIP アドレス		66.165.176.12
SNIP アドレス		66.165.176.15
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	CloudBridge Connector トンネルのローカルエンドポイント IP アドレス:66.165.176.15、CloudBridge Connector トンネルのリモートエンドポイント IP アドレス:168.63.252.133、GRE トンネルの詳細-名前 = CC_Tunnel_DC-AWS

次の表は、AWS クラウド上の Citrix ADC VPX NS_VPX_ アプライアンス-AWS の設定を示しています。

エンティティ	Name	詳細
NSIP アドレス		10.102.25.30
NSIP アドレスにマッピングされたパブリック EIP アドレス		168.63.252.131
SNIP アドレス		10.102.29.30
SNIP アドレスにマッピングされたパブリック EIP アドレス		168.63.252.133

エンティティ	Name	詳細
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	CloudBridge Connector トンネルのローカルエンドポイント IP アドレス:168.63.252.133、CloudBridge Connector トンネルのリモートエンドポイント IP アドレス:66.165.176.15; **GRE** トンネルの詳細名 = CC_Tunnel_DC-AWS、IPSec プロファイルの詳細、名前 = CC_Tunnel_DC-AWS、暗号化アルゴリズム = AES、ハッシュアルゴリズム = HMACSHA

前提条件

CloudBridge Connector トンネルを設定する前に、次のタスクが完了していることを確認します。

1. AWS クラウド上で Citrix ADC 仮想アプライアンス（VPX）のインスタンスをインストール、設定、起動します。Citrix ADC VPX を AWS にインストールする手順については、「AWS での Citrix ADC VPX インスタンスのデプロイ」を参照してください。
2. Citrix ADC 物理アプライアンスを展開して構成するか、またはデータセンター内の仮想化プラットフォームで Citrix ADC 仮想アプライアンス（VPX）を Provisioning して構成します。
3. CloudBridge Connector トンネルのエンドポイントの IP アドレスが相互にアクセス可能であることを確認します。

Citrix ADC VPX ライセンス

インスタンスの初期起動後、Citrix ADC VPX for AWS にはライセンスが必要です。独自のライセンス (BYOL) を持参する場合は、http://support.citrix.com/article/CTX122426の VPX ライセンスガイドを参照してください。

次の操作を実行する必要があります。

1. Citrix Web サイトのライセンスポータルを使用して、有効なライセンスを生成します。
2. ライセンスをインスタンスにアップロードします。

有料マーケットプレイスインスタンスの場合は、ライセンスをインストールする必要はありません。該当する機能セットとパフォーマンスが自動的にアクティブ化されます。

構成の手順

データセンター内に存在する Citrix ADC アプライアンスと、AWS クラウド上に存在する Citrix ADC 仮想アプライアンス（VPX）との間に CloudBridge Connector トンネルを設定するには、Citrix ADC アプライアンスの GUI を使用します。

GUI を使用すると、Citrix ADC アプライアンスで作成された CloudBridge Connector トンネル構成が、CloudBridge Connector トンネルの他のエンドポイントまたはピア（AWS 上の Citrix ADC VPX）に自動的にプッシュされます。したがって、対応する CloudBridge Connector トンネル設定を作成するために、AWS 上の Citrix ADC VPX GUI（GUI）にアクセスする必要はありません。

両方のピア（データセンターに存在する Citrix ADC アプライアンスと、AWS クラウド上に存在する Citrix ADC 仮想アプライアンス（VPX））の CloudBridge Connector トンネル構成は、次のエンティティで構成されます。

- **IPSec** プロファイル：IPSec プロファイルエンティティは、CloudBridge Connector トンネルの両方のピアで IPSec プロトコルで使用される、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPSec プロトコルパラメータを指定します。
- **GRE** トンネル：IP トンネルは、ローカル IP アドレス（ローカルピアで設定されたパブリック SNIP アドレス）、リモート IP アドレス（リモートピアで設定されたパブリック SNIP アドレス）、CloudBridge Connector トンネルのセットアップに使用するプロトコル（GRE）、および IPSec プロファイルエンティティを指定します。
- **PBR** ルールを作成し、**IP** トンネルを関連付ける：PBR エンティティは、一連の条件と IP トンネルエンティティを指定します。送信元 IP アドレスの範囲と宛先 IP の範囲は、PBR エンティティの条件です。トラフィックが CloudBridge Connector トンネルを通過するサブネットを指定するには、送信元 IP アドレスの範囲と宛先 IP アドレスの範囲を設定する必要があります。たとえば、データセンター内のサブネット上のクライアントから発信され、AWS クラウド内のサブネット上のサーバー宛てのリクエストパケットがあるとします。このパケットが、データセンター内の Citrix ADC アプライアンス上の PBR エンティティの送信元および宛先 IP アドレス範囲と一致する場合、そのパケットは PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

コマンドラインインターフェイスを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> [-**ikeVersion** (V1 | V2)] [-**encAlgo** (AES | 3DES)...] [-**hashAlgo** <hashAlgo> ...] [-**lifetime** <positive_integer>] (-**psk** | (-**publickey** <string> -**privatekey** <string> -**peerPublicKey** <string>))[-**livenessCheckInterval** <positive_integer>] [-**replayWindowSize** <positive_integer>] [-**ikeRetryInterval** <positive_integer>] [-**retransmissiontime** <positive_integer>]`
- `**show ipsec profile** <name>`

コマンドラインインターフェイスを使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

コマンドラインインターフェイスを使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

コマンドプロンプトで入力します。

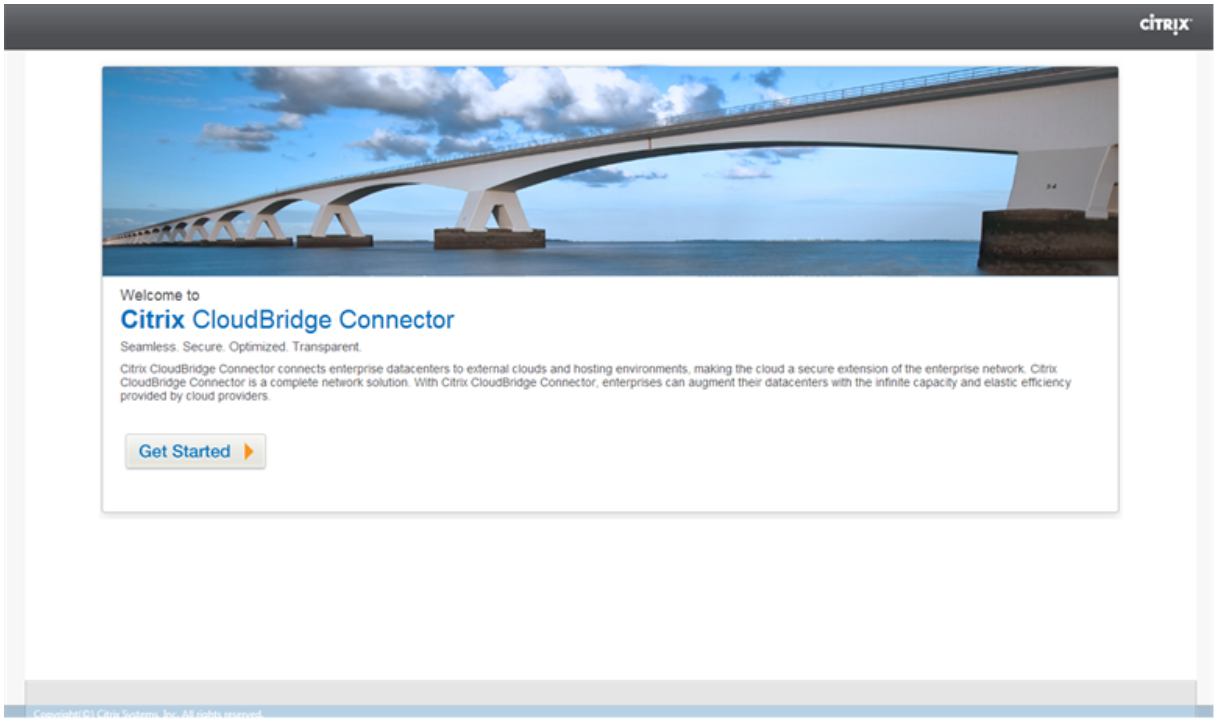
- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

例

```
1      > add ipsec profile CC_Tunnel_DC-AWS -encAlgo AES -hashAlgo
           HMAC_SHA1
2
3      Done
4      > add ipTunnel CC_Tunnel_DC-AWS 168.63.252.133 255.255.255.0
           66.165.176.15  - protocol GRE -ipsecProfileName CC_Tunnel_DC-AWS
5
6      Done
7      > add ns pbr PBR-DC-AWS ALLOW  - srcIP 66.165.176.15  - destIP
           168.63.252.133 ipTunnel CC_Tunnel_DC-AWS
8
9      Done
10     > apply ns pbrs
11
12     Done
13 <!--NeedCopy-->
```

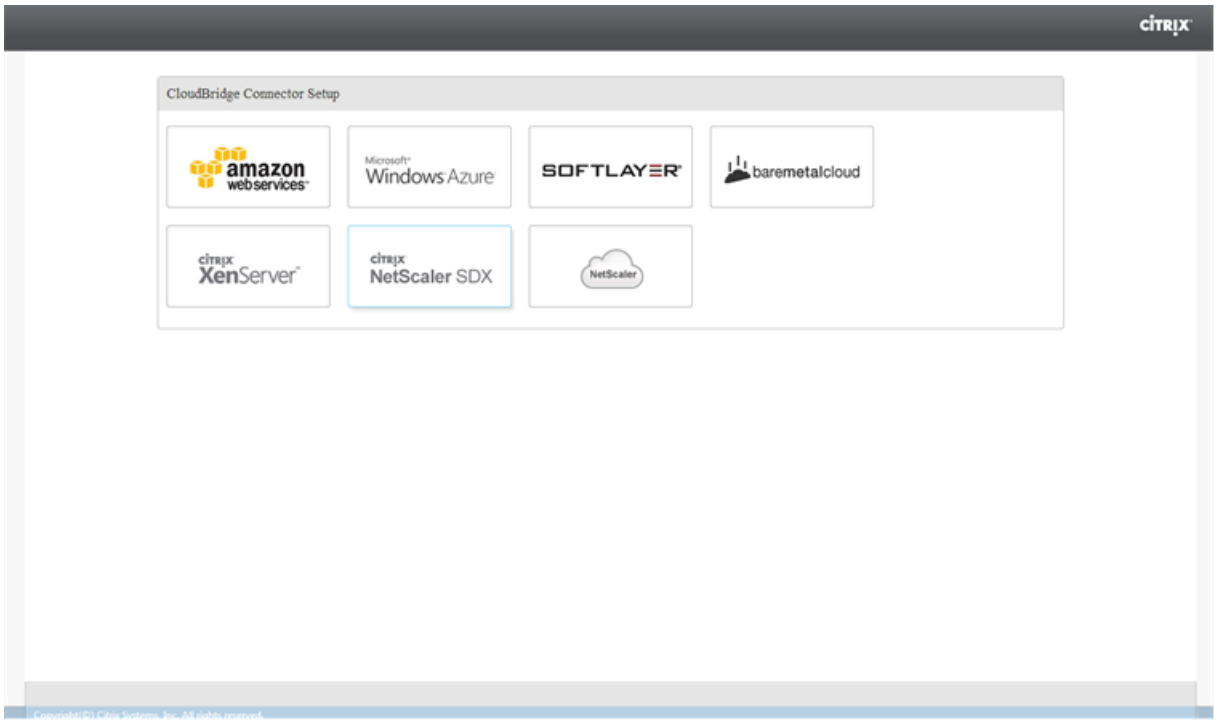
GUI を使用して Citrix ADC アプライアンスで CloudBridge Connector トンネルを構成するには

1. Web ブラウザのアドレス行に、Citrix ADC アプライアンスの NSIP アドレスを入力します。
2. アプライアンスのアカウント認証情報を使用して、Citrix ADC アプライアンスの GUI にログオンします。
3. [システム] > [**CloudBridge Connector**] に移動します。
4. 右側のペインの [はじめに] で、[**CloudBridge** の作成/監視] をクリックします。
5. アプライアンスで CloudBridge Connector トンネルを初めて設定すると、ようこそ画面が表示されます。
6. ようこそ画面で、[は じめに] をクリックします。



> 注:
>
> Citrix ADC アプライアンスで CloudBridge Connector トンネルをすでに構成している場合は、ようこそ画面が表示されないため、［開始］をクリックしないでください。

1. [**CloudBridge Connector** の設定] ペインで、[**Amazon** ウェブサービス] をクリックします。



1. [**Amazon**] ペインで、AWS アカウントの認証情報（AWS アクセスキー ID および AWS シークレットアクセスキー）を入力します。これらのアクセスキーは、AWS GUI コンソールから入手できます。［続行］をクリックします。

> 注
>
> 以前は、別のリージョンが選択されていても、セットアップウィザードは常に同じ AWS リージョンに接続します。その結果、選択した AWS リージョンで実行されている Citrix ADC VPX への CloudBridge Connector トンネルを設定すると失敗していました。この問題は現在修正されています。

1. [**Citrix ADC**] ペインで、AWS で実行されている Citrix ADC 仮想アプライアンスの NSIP アドレスを選択します。次に、Citrix ADC 仮想アプライアンスのアカウント資格情報を入力します。［続行］をクリックします。
2. **CloudBridge Connector** 設定ペインで、次のパラメータを設定します。
   - **CloudBridge Connector** 名— ローカルアプライアンス上の CloudBridge Connector 設定の名前。ASCII アルファベットまたはアンダースコア (_) 文字で始まり、ASCII 英数字、アンダースコア、ハッシュ (#)、ピリオド (.)、スペース、コロン (:)、アットマーク (@)、等しい (=)、およびハイフン (-) 文字のみを含める必要があります。CloudBridge Connector の設定が作成された後は変更できません。
3. [ローカル設定] で、次のパラメータを設定します。
   - サブネット **IP**：CloudBridge Connector トンネルのローカルエンドポイントの IP アドレス。SNIP タイプのパブリック IP アドレスである必要があります。
4. [リモート設定] で、次のパラメータを設定します。
   - サブネット **IP**— AWS 側の CloudBridge Connector のトンネルエンドポイントの IP アドレス。AWS 上の Citrix ADC VPX インスタンスでは、SNIP タイプの IP アドレスである必要があります。
   - **NAT**：AWS 上の Citrix ADC VPX インスタンスで設定された SNIP にマッピングされた AWS 内のパブリック IP アドレス（EIP）。
5. [**PBR** 設定**]** で、次のパラメータを設定します。
   - 演算：等しい（=）または等しくない（! =) 論理演算子。
   - 送信元 **IP** 低：発信 IPv4 パケットの送信元 IP アドレスと照合する最小の送信元 IP アドレス。
   - 送信元 **IP High**：発信 IPv4 パケットの送信元 IP アドレスと照合する最大送信元 IP アドレス。
   - 演算：等しい（=）または等しくない（! =) 論理演算子。
   - 宛先 **IP** 低：発信 IPv4 パケットの宛先 IP アドレスと照合する最小の宛先 IP アドレス。
   - 宛先 **IP** 高：発信 IPv4 パケットの宛先 IP アドレスと照合する最大の宛先 IP アドレス。
6. （オプション）[セキュリティ設定**]** で、CloudBridge Connector トンネルに次の IPSec プロトコルパラメータを設定します。
   - 暗号化アルゴリズム：CloudBridge トンネル内の IPSec プロトコルで使用される暗号化アルゴリズム。
   - ハッシュアルゴリズム：CloudBridge トンネル内の IPSec プロトコルで使用されるハッシュアルゴリズム。
   - [**Key**]：相互認証に 2 つのピアが使用する次の IPSec 認証方法のいずれかを選択します。

- キーの自動生成— ローカルアプライアンスによって自動的に生成される事前共有キー（PSK）と呼ばれるテキスト文字列に基づく認証。ピアの PSK キーは、認証のために相互に照合されます。
- 特定のキー：手動で入力した PSK に基づく認証。ピアの PSK は、認証のために相互に照合されます。
  - 事前共有セキュリティキー：事前共有キーベースの認証用に入力されたテキストストリング。
- 証明書のアップロード：デジタル証明書に基づく認証。
  - **[Public Key]**：IPSec セキュリティアソシエーションを確立する前に、リモートピアに対してローカルピアを認証するために使用されるローカルデジタル証明書。同じ証明書が存在し、ピアの Peer Public Key パラメータに設定する必要があります。
  - 秘密キー：ローカルデジタル証明書の秘密キー。
  - ピア公開キー：ピアのデジタル証明書。IPSec セキュリティアソシエーションを確立する前に、ローカルエンドポイントに対してピアを認証するために使用されます。同じ証明書が存在し、ピアの Public key パラメータに設定する必要があります。

7. ［完了］をクリックします。

データセンター内の Citrix ADC アプライアンスの新しい CloudBridge Connector トンネル構成が、GUI の［ホーム］タブに表示されます。AWS クラウド内の Citrix ADC VPX アプライアンス上の対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

Citrix ADC アプライアンスと **AWS** 上の仮想プライベート **Gateway** 間の **CloudBridge Connector** トンネルの設定

October 7, 2021

データセンターを Amazon Web Services（AWS）に接続するには、データセンター内の Citrix ADC アプライアンスと AWS の仮想プライベート Gateway の間に CloudBridge Connector トンネルを設定します。Citrix ADC アプライアンスと仮想プライベート Gateway は、CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

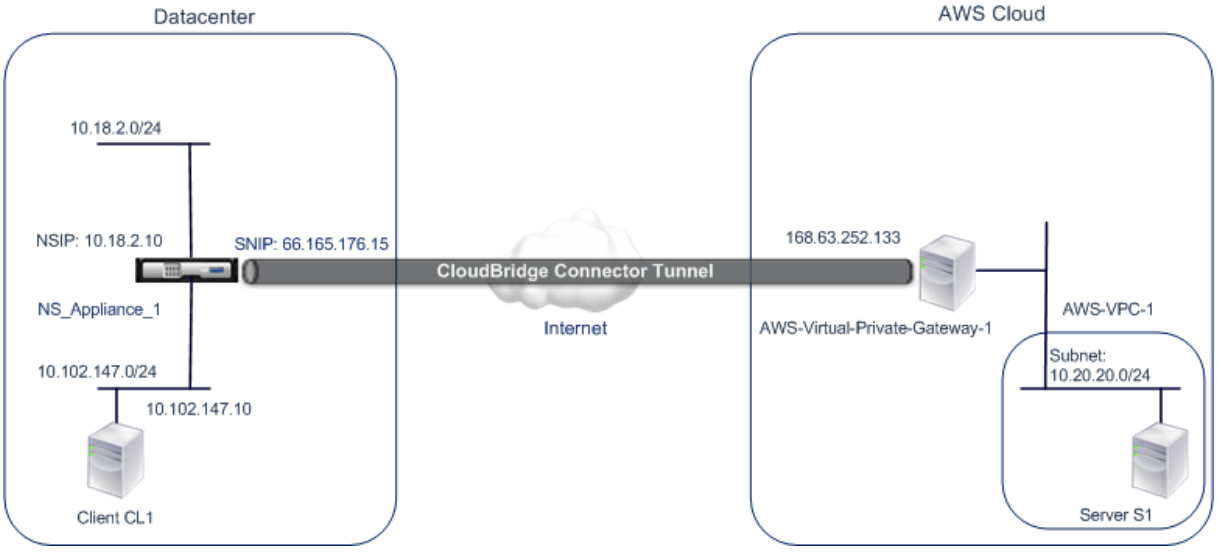
> 注:
>
> また、データセンター内の Citrix ADC アプライアンスと（仮想プライベート Gateway ではなく）AWS 上の Citrix ADC VPX インスタンスとの間に CloudBridge Connector トンネルを設定することもできます。詳細については、「データセンターと AWS クラウド間の CloudBridge Connector の設定」を参照してください。

AWS の仮想プライベートゲートウェイは、CloudBridge Connector トンネルに対して次の IPSec 設定をサポートしています。したがって、CloudBridge Connector トンネル用に Citrix ADC アプライアンスを構成するときは、同じ IPSec 設定を指定する必要があります。

IPSec のプロパティ	設定
IPSec モード	トンネルモード
IKE のバージョン	バージョン 1
IKE 認証方法	事前共有キー
暗号化アルゴリズム	AES
ハッシュ関数	HMAC SHA1

CloudBridge Connector のトンネル設定とデータフローの例

CloudBridge Connector トンネル内のトラフィックフローの図として、データセンター内の Citrix ADC アプライアンス NS_Appliance-1 と AWS クラウド上の仮想プライベート Gateway AWS 仮想プライベート Gateway 1 との間に CloudBridge Connector トンネルが設定されている例を考えてみます。



NS_Appliance-1 は L3 ルーターとしても機能し、データセンター内のプライベートネットワークが CloudBridge Connector トンネルを介して AWS クラウド内のプライベートネットワークに到達できるようにします。NS_Appliance-1 は、ルーターとして、データセンター内のクライアント CL1 と AWS クラウド内のサーバー S1 と

の間の通信を CloudBridge Connector トンネル経由で可能にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル設定には、NS_AWS_IPSec_Profile という名前の IPSec プロファイルエンティティ、NS_AWS_Tunnel という名前の CloudBridge Connector トンネルエンティティ、および NS_AWS_Pbr という名前のポリシーベースのルーティング (PBR) エンティティが含まれます。

IPSec プロファイルエンティティ NS_AWS_IPSec_Profile は、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズムなどの IPSec プロトコルパラメータを指定します。このパラメータは、CloudBridge Connector トンネル内の IPSec プロトコルで使用されます。NS_AWS_IPSec_ プロファイルは、IP トンネルエンティティ NS_AWS_Tunnel にバインドされています。

CloudBridge Connector トンネルエンティティ NS_AWS_Tunnel は、ローカル IP アドレス（Citrix ADC アプライアンス上で構成されたパブリック IP—SNIP アドレス)、リモート IP アドレス (AWS-Virtual-Private-Gateway-1 の IP アドレス)、および CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPSec）を指定します。NS_AWS_Tunnel は、ポリシーベースルーティング (PBR) エンティティ NS_AWS_Pbr にバインドされています。

PBR エンティティ NS_AWS_Pbr は、一連の条件と CloudBridge Connector トンネルエンティティ (NS_AWS_Tunnel) を指定します。送信元 IP アドレスの範囲と宛先 IP アドレスの範囲は、NS_AWS_Pbr の条件です。送信元 IP アドレスの範囲と送信先 IP アドレスの範囲は、それぞれデータセンターではサブネットとして、AWS クラウドではサブネットとして指定されます。データセンター内のサブネットのクライアントから発信され、AWS クラウド上のサブネットのサーバー宛てのリクエストパケットは、NS_AWS_Pbr の条件と一致します。このパケットは、CloudBridge Connector の処理のために考慮され、PBR エンティティにバインドされた CloudBridge Connector トンネル（NS_AWS_Tunnel）を介して送信されます。

次の表に、この例で使用される設定を示します。

データセンター側の CloudBridge Connector トンネルエンドポイント（NS_ アプライアンス-1）の IP アドレス	66.165.176.15
AWS の CloudBridge Connector トンネルエンドポイント（AWS-Virtual-Private-Gateway-1）の IP アドレス	168.63.252.133
データセンターのサブネット。トラフィックは CloudBridge Connector のトンネルを通過します。	10.102.147.0/24
AWS サブネット。トラフィックは CloudBridge Connector トンネルを通過します。	10.20.20.0/24

Amazon AWS での設定

Customer Gateway	AWS-Customer-Gateway-1	ルーティング = 静的、IP アドレス = インターネットでルーティング可能な CloudBridge Connector トンネルエンドポイント IP アドレス、Citrix ADC 側のアドレス = 66.165.176.15
仮想プライベートゲートウェイ	AWS-Virtual-Private-Gateway-1	関連付けられた VPC = AWS VPC
VPN 接続	AWS-VPN-Connection-1	Customer Gateway = AWS-Customer-Gateway-1, Virtual Private Gateway= Virtual-Private-Gateway-1, Routing Options: Type = Static, Static IP Prefixes = Subnets on the Citrix ADC side = 10.102.147.0/24

データセンター **1** の **Citrix ADC** アプライアンス **NS_** アプライアンス **1** での設定:

アプライアンス	設定	
SNIP1（参考目的のみ）	66.165.176.15	
セキュリティプロファイル	NS_AWS_IPSec_ プロファイル	IKE バージョン = v1、暗号化アルゴリズム = AES、ハッシュアルゴリズム = HMAC SHA1
CloudBridge Connector トンネル	NS_AWS_ トンネル	リモート IP = 168.63.252.133、ローカル IP = 66.165.176.15、トンネルプロトコル = IPSec プロファイル

アプライアンス	設定
ポリシー・ベースルート	送信元 IP 範囲 = データセンター内のサブネット =10.102.147.0-010.102.147.255、送信先 IP 範囲 = サブネットのサブネット =10.20.20.0-010.20.255、IP トンネル =NS_AWS_ トンネル

CloudBridge Connector のトンネル設定について考慮すべきポイント

Citrix ADC アプライアンスと AWSGateway 間の CloudBridge Connector トンネルを設定する前に、次の点を考慮してください。

1. AWS では、CloudBridge Connector トンネルに対して次の IPSec 設定がサポートされています。したがって、CloudBridge Connector トンネル用に Citrix ADC アプライアンスを構成するときは、同じ IPSec 設定を指定する必要があります。
    - IKE version = v1
    - Encryption algorithm = AES
    - Hash algorithm = HMAC SHA1
2. Citrix ADC 側でファイアウォールを構成して、次のことを許可する必要があります。
    - ポート 500 の任意の UDP パケット
    - ポート 4500 に対する任意の UDP パケット
    - 任意の ESP（IP プロトコル番号 50）パケット
3. Citrix ADC でトンネル設定を指定する前に Amazon AWS を設定する必要があります。これは、トンネルの AWS エンド（Gateway）と PSK のパブリック IP アドレスは、AWS でトンネル設定を設定すると自動的に生成されるためです。この情報は、Citrix ADC アプライアンスのトンネル構成を指定するために必要です。
4. AWS Gateway は、ルート更新のための静的ルートと BGP プロトコルをサポートしています。Citrix ADC アプライアンスは、AWSGateway への CloudBridge Connector トンネル内の BGP プロトコルをサポートしていません。したがって、トンネルを通るトラフィックを適切にルーティングするには、CloudBridge Connector トンネルの両側で適切な静的ルートを使用する必要があります。

CloudBridge Connector トンネル用に **Amazon AWS** を設定する

Amazon AWS で CloudBridge Connector のトンネル設定を作成するには、Amazon AWS マネジメントコンソールを使用します。これは、Amazon AWS でリソースを作成および管理するためのウェブベースのグラフィカルインターフェイスです。

AWS クラウドで CloudBridge Connector のトンネル設定を開始する前に、次のことを確認してください。

- Amazon AWS クラウドのユーザーアカウントがあります。
- CloudBridge Connector トンネルを介して Citrix ADC 側のネットワークに接続するネットワークを持つ仮想プライベートクラウドがあります。
- Amazon AWS マネジメントコンソールに精通していること。

> 注:
>
> CloudBridge Connector のトンネル用に Amazon AWS を設定する手順は、Amazon AWS のリリースサイクルに応じて時間の経過とともに変化する可能性があります。最新の手順については、Amazon AWS のドキュメントを参照することをお勧めします 。

Citrix ADC と AWS Gateway 間の CloudBridge Connector トンネルを設定するには、AWS マネジメントコンソールで以下のタスクを実行します。

- カスタマーゲートウェイを作成します。カスタマー Gateway は、CloudBridge Connector のトンネルエンドポイントを表す AWS エンティティです。Citrix ADC アプライアンスと AWS Gateway 間の CloudBridge Connector トンネルの場合、カスタマー Gateway は AWS 上の Citrix ADC アプライアンスを表します。カスタマー Gateway は、名前、トンネルで使用されるルーティングのタイプ（静的または BGP）、Citrix ADC 側の CloudBridge Connector トンネルエンドポイント IP アドレスを指定します。IP アドレスには、インターネットでルーティング可能な Citrix ADC が所有するサブネット IP（SNIP）アドレスを使用できます。Citrix ADC アプライアンスが NAT デバイスの背後にある場合は、SNIP アドレスを表すインターネットでルーティング可能な NAT IP アドレスを使用できます。
- 仮想プライベートゲートウェイを作成し、**VPC** にアタッチします。仮想プライベート Gateway は、AWS 側の CloudBridge Connector トンネルエンドポイントです。仮想プライベート Gateway を作成するときに、名前を割り当てるか、AWS に名前を割り当てることを許可します。次に、仮想プライベート Gateway を VPC に関連付けます。この関連付けにより、VPC のサブネットは CloudBridge Connector トンネルを介して Citrix ADC 側のサブネットに接続できるようになります。
- **VPN** 接続を作成します。VPN 接続では、CloudBridge Connector トンネルが作成されるカスタマー Gateway と仮想プライベート Gateway を指定します。また、Citrix ADC 側のネットワークの IP プレフィックスも指定します。（静的ルートエントリを介して）仮想プライベート Gateway に認識されている IP プレフィックスのみが、トンネルを介して VPC からトラフィックを受信できます。また、仮想プライベート Gateway は、指定された IP プレフィックスを宛先としないトラフィックをトンネル経由でルーティングしません。VPN 接続を設定した後、接続が作成されるまで数分待たなければならない場合があります。
- ルーティングオプションを設定します。VPC のネットワークが CloudBridge Connector トンネルを経由して Citrix ADC 側のネットワークに到達するには、VPC のルーティングテーブルを設定して、Citrix ADC 側のネットワークへのルートを含めて、それらのルートを仮想プライベート Gateway に向ける必要があります。

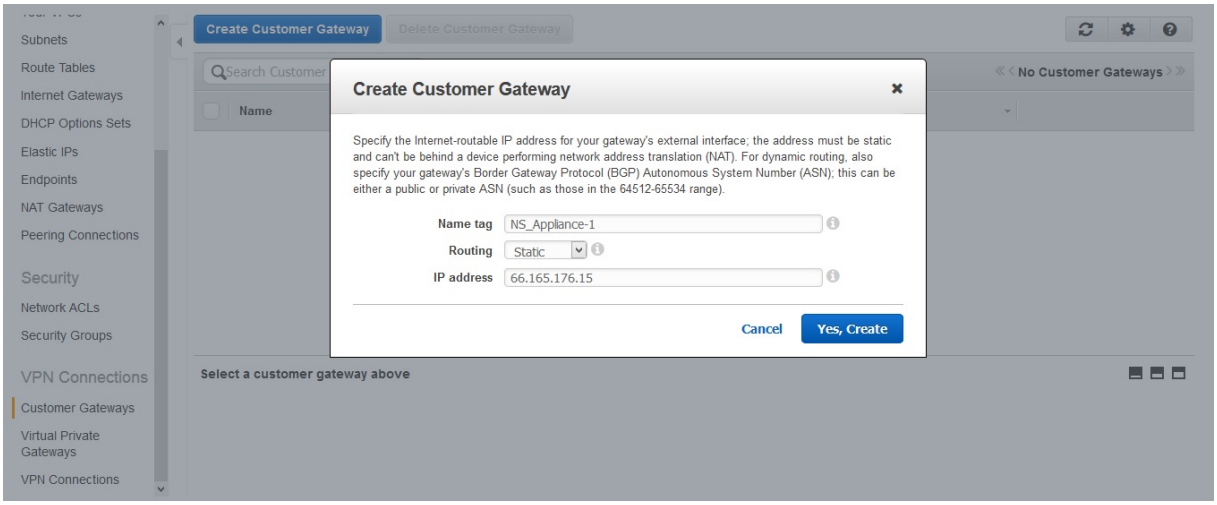
次のいずれかの方法で、VPC のルーティングテーブルにルートを含めることができます。

- ルート伝播を有効にします。ルーティングテーブルのルート伝播を有効にして、ルートが自動的にテーブルに伝播されるようにできます。VPN 設定に指定するスタティック IP プレフィクスは、VPN 接続の作成後にルーティングテーブルに伝播されます。
- スタティックルートを手動で入力します。ルートの伝播を有効にしない場合は、Citrix ADC 側でネットワークの静的ルートを手動で入力する必要があります。

- ダウンロード設定。AWS で CloudBridge Connector トンネル（VPN 接続）設定が作成されたら、VPN 接続の設定ファイルをローカルシステムにダウンロードします。Citrix ADC アプライアンスで CloudBridge Connector トンネルを構成するために、構成ファイル内の情報が必要になる場合があります。

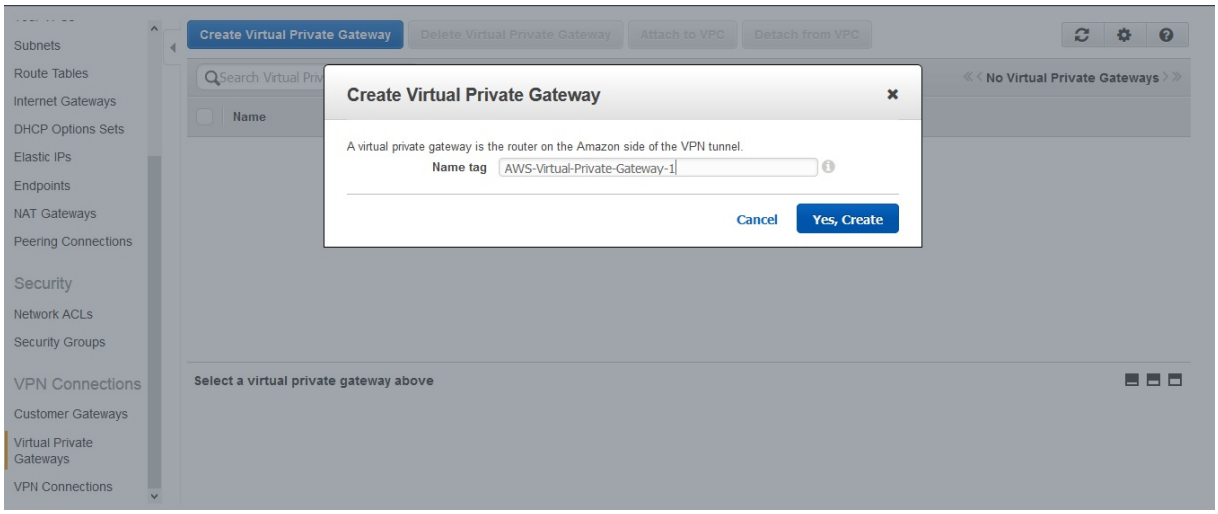
カスタマー Gateway を作成するには

1. https://console.aws.amazon.com/vpc/で Amazon VPC コンソールを開きます。
2. [**VPN** 接続] > [カスタマーゲートウェイ] に移動し、[カスタマーゲートウェイの作成] をクリックします。
3. [カスタマーゲートウェイの作成] ダイアログボックスで、次のパラメータを設定し、[はい、作成] をクリックします。
   - **Name** タグ。カスタマー Gateway の名前。
   - ルーティングリスト。CloudBridge Connector トンネルを介して相互にルートをアドバタイズするための、Citrix ADC アプライアンスと AWS 仮想プライベート Gateway 間のルーティングのタイプ。「ルーティング」リストから「静的ルーティング」を選択します。注：Citrix ADC アプライアンスは、AWSGateway への CloudBridge Connector トンネル内の BGP プロトコルをサポートしていません。したがって、トンネルを通るトラフィックを適切にルーティングするには、CloudBridge Connector トンネルの両側で適切な静的ルートを使用する必要があります。
   - **IP** アドレス。Citrix ADC 側のインターネットルーティング可能な CloudBridge Connector トンネルエンドポイント IP アドレスです。IP アドレスには、インターネットでルーティング可能な Citrix ADC が所有するサブネット IP（SNIP）アドレスを使用できます。Citrix ADC アプライアンスが NAT デバイスの背後にある場合は、SNIP アドレスを表すインターネットでルーティング可能な NAT IP アドレスを使用できます。

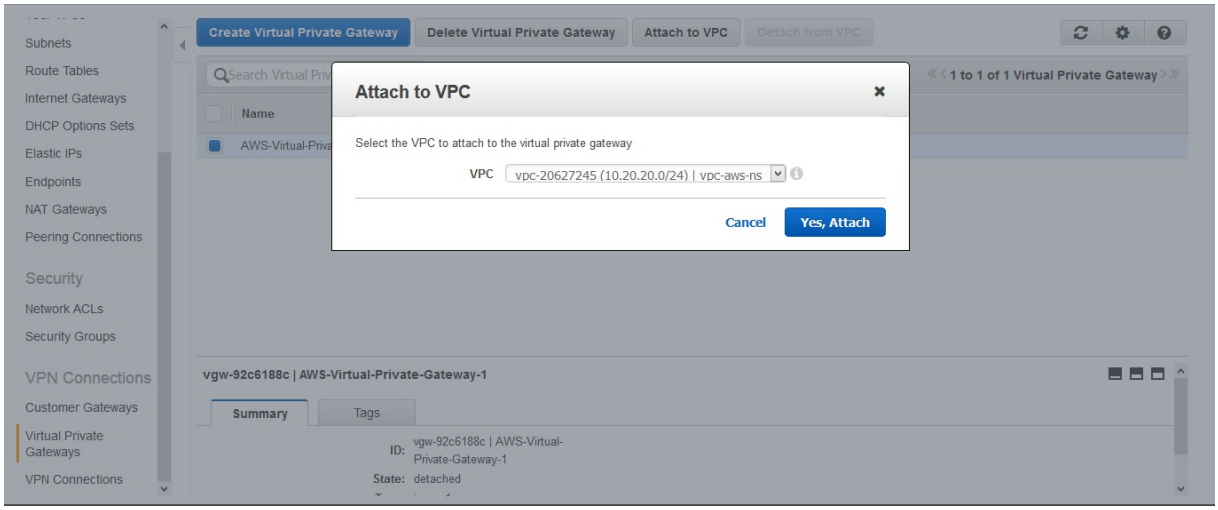


仮想プライベート Gateway を作成して VPC にアタッチするには

1. [**VPN** 接続] > [仮想プライベートゲートウェイ] に移動し、[仮想プライベートゲートウェイの作成] をクリックします。
2. 仮想プライベート Gateway の名前を入力し、[Yes, Create] をクリックします。



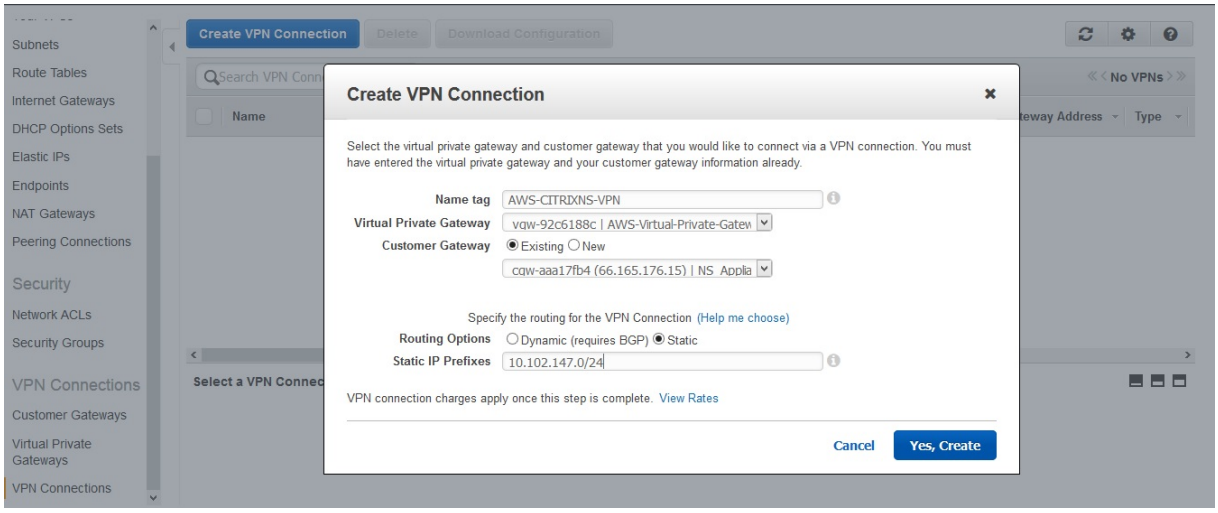
1. 作成した仮想プライベート Gateway を選択し、[Attach to VPC] をクリックします。
2. [Attach to VPC] ダイアログボックスで、リストから VPC を選択し、[Yes, Attach] を選択します。



VPN 接続を作成するには:

1. [VPN 接続] > [VPN 接続] に移動し、[VPN 接続の作成] をクリックします。
2. [VPN 接続の作成] ダイアログボックスで次のパラメータを設定し、[Yes, Create] を選択します。
   - **Name** タグ。VPN 接続の名前。
   - 仮想プライベートゲートウェイ。前に作成した仮想プライベート Gateway を選択します。
   - カスタマーゲートウェイ。[既存] を選択します。次に、ドロップダウンリストから、以前に作成したカスタマー Gateway を選択します。
   - ルーティングオプション。仮想プライベート Gateway とカスタマー Gateway（Citrix ADC アプライアンス）の間のルーティングのタイプ。[静的] を選択します。［静的 IP プレフィックス］フィールドで、

Citrix ADC 側のサブネットの IP プレフィックスをカンマで区切って指定します。



ルート伝播を有効にするには、次の手順を実行します。

1. [ルートテーブル] に移動し、トラフィックが CloudBridge Connector トンネルを通過するサブネットに関連付けられているルーティングテーブルを選択します。

> 注
>
> デフォルトでは、これは VPC のメインルーティングテーブルです。

1. 詳細ペインの [**Route Propagation**] タブで、[**Edit]** を選択し、仮想プライベート Gateway を選択して、**[Save]** を選択します。

スタティックルートを手動で入力するには、次の手順を実行します。

1.「ルートテーブル」に移動し、ルーティングテーブルを選択します。
2. [ルート] タブで、[編集] をクリックします。
3. [**Destination**] フィールドに、CloudBridge Connector トンネル（VPN 接続）で使用される静的ルートを入力します。
4. [**Target**] リストから仮想プライベート Gateway ID を選択し、[**Save**] をクリックします。

設定ファイルをダウンロードするには、次の手順に従います。

1. [**VPN** 接続] に移動し、VPN 接続を選択し、[設定のダウンロード] をクリックします。
2. [ダウンロード構成] ダイアログボックスで、次のパラメータを設定し、[はい、ダウンロード] をクリックします。
   - ベンダー。[汎用] を選択します。
   - プラットフォーム。[汎用] を選択します。
   - ソフトウェア。「ベンダーに依存しない」を選択します。

CloudBridge Connector トンネル用の **Citrix ADC** アプライアンスの構成

Citrix ADC アプライアンスと AWS クラウド上の仮想プライベート Gateway との間に CloudBridge Connector トンネルを設定するには、Citrix ADC アプライアンスで以下のタスクを実行します。
Citrix ADC コマンドラインまたは GUI のいずれかを使用できます。

- **IPSec** プロファイルを作成します。IPSec プロファイルエンティティは、CloudBridge Connector トンネル内の IPSec プロトコルで使用される IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPSec プロトコルパラメータを指定します。
- **IPSec** プロトコルを使用する **IP** トンネルを作成し、**IPSec** プロファイルを関連付けます。IP トンネルは、ローカル IP アドレス（Citrix ADC アプライアンスで設定された SNIP アドレス）、リモート IP アドレス（AWS の仮想プライベート Gateway のパブリック IP アドレス）、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPSec）、および IPSec プロファイルエンティティを指定します。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成し、**IP** トンネルに関連付けます。PBR エンティティは、一連のルールと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレスの範囲と宛先 IP アドレスの範囲は、PBR エンティティの条件です。送信元 IP アドレス範囲を設定して、トラフィックがトンネルを通過する Citrix ADC 側のサブネットを指定します。送信先 IP アドレス範囲を設定して、トラフィックが CloudBridge Connector トンネルを通過する AWS VPC サブネットを指定します。Citrix ADC 側のサブネットのクライアントから発信され、AWS クラウドサブネットのサーバーに送信され、PBR エンティティの送信元および宛先 IP 範囲と一致するすべてのリクエストパケットは、PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

Citrix ADC コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで、次のように入力します。

- `add ipsec profile <name> -psk <string> -**ikeVersion** v1`
- `show ipsec profile** <name>`

Citrix ADC コマンドラインを使用して IPSEC トンネルを作成し、そのトンネルに IPSEC プロファイルをバインドするには

コマンドプロンプトで、次のように入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

Citrix ADC コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

コマンドプロンプトで、次のように入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP** <subnet-range> -*ipTunnel <tunnelName>`
- `apply pbrs`

- `show pbr <pbrName>`

次のコマンドは、「CloudBridge Connector の構成とデータフローの例」で使用される Citrix ADC アプライアンス NS_Appliance-1 のすべての設定を作成します。

```
1      > add ipsec profile NS_AWS_IPSec_Profile -psk
          DkiMgMdcbqvYREEuIvxsbKkW0Foyabcd -ikeVersion v1  - lifetime
          31536000
2      Done
3      > add iptunnel NS_AWS_Tunnel 168.63.252.133 255.255.255.255
          66.165.176.15  - protocol IPSEC  - ipsecProfileName
          NS_AWS_IPSec_Profile
4
5      Done
6      > add pbr NS_AWS_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
          10.20.0.0-10.20.255.255  - ipTunnel NS_AWS_Tunnel
7      Done
8
9      > apply pbrs
10
11     Done
12 <!--NeedCopy-->
```

GUI を使用して IPSEC プロファイルを作成するには

1. [システム] > [**CloudBridge Connector**] > [**IPsec** プロファイル**]** に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**IPSec** プロファイルの追加] ダイアログボックスで、次のパラメータを設定します。
   - Name
   - 暗号化アルゴリズム
   - ハッシュアルゴリズム
   - IKE プロトコルバージョン（V1 を選択）
4. [事前共有キー認証] 方法を選択し、[事前共有キーが存在します**]** パラメータを設定します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2. [**IPv4** トンネル] タブで、[追加] をクリックします。
3. [**Add IP Tunnel**] ダイアログボックスで、次のパラメータを設定します。
   - Name

- リモート IP
- リモートマスク
- ローカル IP タイプ ([ローカル IP タイプ] ドロップダウンリストで、[サブネット IP] を選択します)。
- ローカル IP（選択した IP タイプの構成済みの IP はすべて、[ローカル IP] ドロップダウンリストに表示されます。リストから目的の IP を選択します)。
- プロトコル
- IPSec プロファイル

4. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. **[PBR** の作成**]** ダイアログボックスで、次のパラメータを設定します。

   - Name
   - 操作（アクション）
   - ネクストホップタイプ（IP トンネルの選択）
   - IP トンネル名
   - 送信元 IP の低
   - 送信元 IP 高
   - 宛先 IP の低
   - 宛先 IP 高

4. [**Create**] をクリックしてから、[**Close**] をクリックします。

Citrix ADC アプライアンスの対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。

CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge Connector] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。
Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

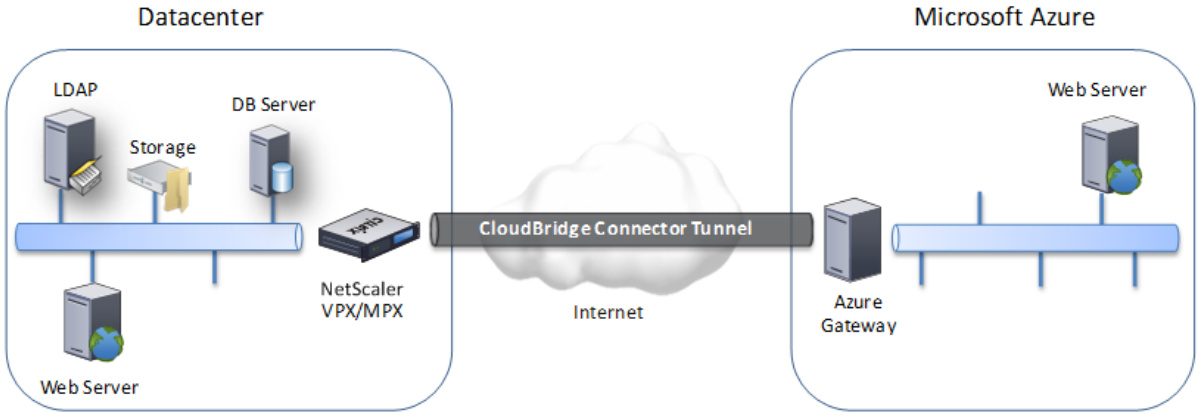
データセンターと **Azure** クラウド間の **CloudBridge Connector** トンネルの構成

June 1, 2022

Citrix ADC アプライアンスは、エンタープライズデータセンターと Microsoft クラウドホスティングプロバイダー Azure の間の接続を提供し、Azure をエンタープライズネットワークのシームレスな拡張にします。Citrix ADC は、エンタープライズデータセンターと Azure クラウド間の接続を暗号化して、両者の間で転送されるすべてのデータを安全にします。

CloudBridge Connector トンネルのしくみ

データセンターを Azure クラウドに接続するには、データセンター内に存在する Citrix ADC アプライアンスと、Azure クラウドに存在する Gateway との間に CloudBridge Connector トンネルを設定します。データセンター内の Citrix ADC アプライアンスと Azure クラウド内の Gateway は、CloudBridge Connector トンネルのエンドポイントであり、CloudBridge Connector トンネルのピアと呼ばれています。



データセンターと Azure クラウド間の CloudBridge Connector トンネルは、オープン標準のインターネットプロトコルセキュリティ (IPSec) プロトコルスイートをトンネルモードで使用して、CloudBridge Connector トンネル内のピア間の通信を保護します。CloudBridge Connector トンネルでは、IPSec によって次のことが保証されます。

- データの整合性
- データ発信元認証
- データの機密性 (暗号化)
- リプレイ攻撃に対する保護

IPSec は、トンネルモードを使用します。このモードでは、IP パケット全体が暗号化され、カプセル化されます。暗号化では、カプセル化セキュリティペイロード (ESP) プロトコルが使用されます。ESP プロトコルは、HMAC ハッシュ関数を使用してパケットの整合性を保証し、暗号化アルゴリズムを使用して機密性を保証します。ESP プロトコルは、ペイロードを暗号化して HMAC を計算した後、ESP ヘッダーを生成し、暗号化された IP パケットの前に挿入します。また、ESP プロトコルは ESP トレーラを生成し、パケットの最後に挿入します。

次に、IPSec プロトコルは、ESP ヘッダーの前に IP ヘッダーを追加することによって、結果のパケットをカプセル化します。IP ヘッダーでは、宛先 IP アドレスは CloudBridge Connector ーピアの IP アドレスに設定されます。

CloudBridge Connector トンネル内のピアは、次のように、インターネットキー交換バージョン 1 (IKEv1) プロトコル (IPSec プロトコルスイートの一部) を使用して、安全な通信をネゴシエートします。

1. 2 つのピアは、事前共有キー認証を使用して相互に認証を行います。事前共有キー認証では、ピアは Pre-Shared Key（PSK; 事前共有キー）と呼ばれるテキスト文字列を交換します。事前共有キーは、認証のために相互に照合されます。したがって、認証を成功させるには、各ピアで同じ事前共有キーを設定する必要があります。
2. ピアは、次の上で合意に達するために交渉します。

   - 暗号化アルゴリズム
   - 1 つのピアでデータを暗号化し、もう 1 つのピアでデータを復号化する暗号化キー。

   セキュリティプロトコル、暗号化アルゴリズム、および暗号化キーに関する本契約は、セキュリティアソシエーション (SA) と呼ばれます。SA は一方向（単方向）です。たとえば、データセンター内の Citrix ADC アプライアンスと Azure クラウド内の Gateway との間に CloudBridge Connector トンネルが設定されている場合、データセンターアプライアンスと Azure Gateway の両方に 2 つの SA があります。一方の SA は発信パケットの処理に使用され、もう一方の SA は着信パケットの処理に使用されます。SA は、ライフタイムと呼ばれる指定された時間間隔が経過すると期限切れになります。

CloudBridge Connector のトンネル設定とデータフローの例

CloudBridge Connector トンネルの図のように、CloudBridge Connector トンネルが Azure クラウド内のデータセンター内の Citrix ADC アプライアンスの CB_Appliance-1 と Gateway Azure_Gateway-1 の間にセットアップされている例を考えてみましょう。

CB_Appliance-1 は L3 ルーターとしても機能し、データセンター内のプライベートネットワークが CloudBridge Connector トンネルを介して Azure クラウド内のプライベートネットワークに到達できるようにします。CB_Appliance-1 は、ルーターとして、データセンター内のクライアント CL1 と Azure クラウド内のサーバー S1 との間の通信を CloudBridge Connector トンネル経由で可能にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

CB_Appliance-1 では、CloudBridge Connector のトンネル設定には、CB_Azure_IPSec_Profile という名前の IPSec プロファイルエンティティ、CB_Azure_Tunnel という名前の CloudBridge Connector トンネルエンティティ、および CB_Azure_Pbr という名前のポリシーベースのルーティング (PBR) エンティティが含まれます。

IPSec プロファイルエンティティ CB_Azure_IPSec_Profile は、CloudBridge Connector トンネル内の IPSec プロトコルで使用される IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズムなどの IPSec プロトコルパラメータを指定します。CB_Azure_IPSec_ プロファイルは、IP トンネルエンティティ CB_Azure_Tunnel にバインドされています。

CloudBridge Connector トンネルエンティティ CB_Azure_Tunnel は、ローカル IP アドレス（Citrix ADC アプライアンス上で構成されたパブリック IP（SNIP）アドレス)、リモート IP アドレス（Azure_Azure_Gateway-1 の

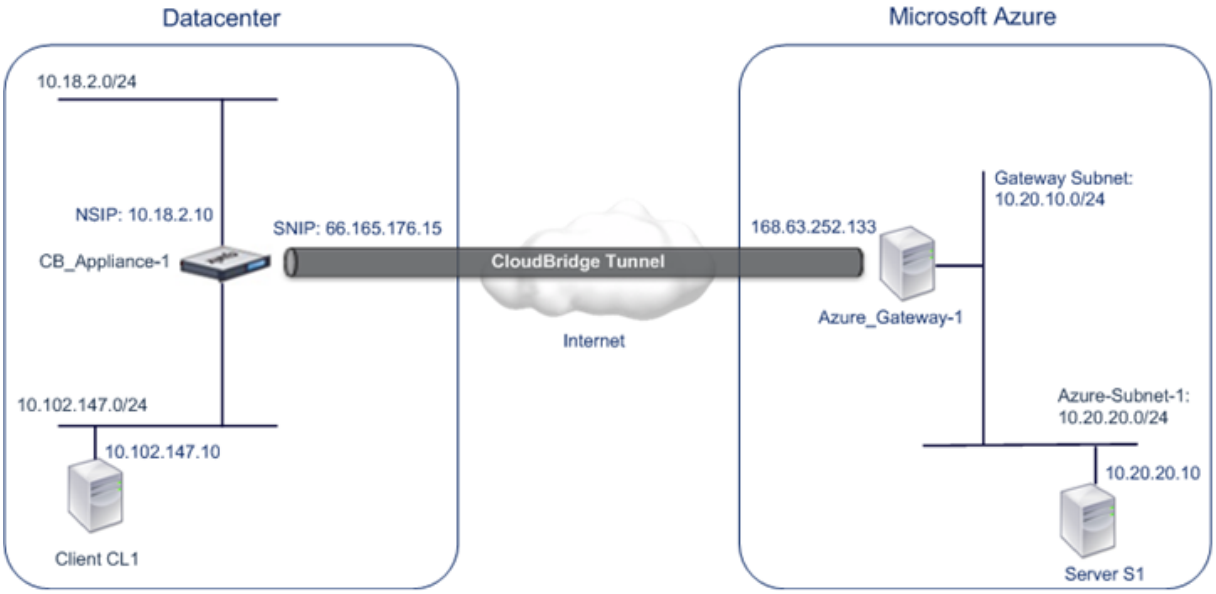
IP アドレス)、および CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPSec）を指定します。CB_Azure_ トンネルは、PBR エンティティ CB_Azure_Pbr にバインドされています。

PBR エンティティ CB_Azure_Pbr は、一連の条件と CloudBridge Connector トンネルエンティティ (CB_Azure_Tunnel) を指定します。送信元 IP アドレスの範囲と宛先 IP アドレスの範囲は、CB_Azure_Pbr の条件です。送信元 IP アドレスの範囲と宛先 IP アドレスの範囲は、それぞれデータセンターではサブネットとして指定され、Azure クラウドではサブネットとして指定されます。データセンターのサブネットのクライアントから発信され、Azure クラウド上のサブネットのサーバー宛ての要求パケットは、CB_Azure_Pbr の条件と一致します。このパケットは、CloudBridge の処理のために考慮され、PBR エンティティにバインドされた CloudBridge Connector トンネル（CB_Azure_Tunnel）を介して送信されます。

Microsoft Azure では、CloudBridge Connector のトンネル構成には、マイデータセンターネットワークという名前のローカルネットワークエンティティ、クラウドブリッジトンネル用の Azure ネットワークという名前の仮想ネットワークエンティティ、Azure_Gateway-1 という名前の Gateway が含まれます。

ローカル（Azure に対してローカル）ネットワークエンティティ My-Datacenter-Network は、データセンター側の Citrix ADC アプライアンスの IP アドレスと、トラフィックが CloudBridge Connector トンネルを通過するデータセンターサブネットを指定します。仮想ネットワークエンティティ Azure-ネットワーク for CloudBridge-Tunnel は、Azure で Azure-Subnet-1 という名前のプライベートサブネットを定義します。サブネットのトラフィックは CloudBridge Connector トンネルを通過します。サーバ S1 は、このサブネットでプロビジョニングされます。

ローカルネットワークエンティティマイデータセンター-ネットワークは、仮想ネットワークエンティティ Azure ネットワーク-for-CloudBridge-Tunnel に関連付けられています。この関連付けは、Azure の CloudBridge Connector トンネル構成のリモートおよびローカルネットワークの詳細を定義します。Gateway Azure_Gateway-1 は、この関連付けが CloudBridge Connector トンネルの Azure エンドで CloudBridge のエンドポイントになるために作成されました。



設定の詳細については、CloudBridge Connector トンネル設定 pdf を参照してください。

CloudBridge Connector のトンネル設定について考慮すべきポイント

データセンター内の Citrix ADC アプライアンスと Microsoft Azure 間の CloudBridge Connector トンネルを構成する前に、次の点を考慮してください。

1. Citrix ADC アプライアンスには、CloudBridge Connector トンネルのトンネルエンドポイントアドレスとして使用するパブリック側の IPv4 アドレス（SNIP タイプ）が必要です。また、Citrix ADC アプライアンスは、NAT デバイスの背後に配置しないでください。
2. Azure では、CloudBridge Connector トンネルに対して次の IPSec 設定がサポートされています。したがって、CloudBridge Connector トンネル用に Citrix ADC を構成する際に、同じ IPSec 設定を指定する必要があります。
    - IKE version = v1
    - Encryption algorithm = AES
    - Hash algorithm = HMAC SHA1
3. 次のことを許可するように、データセンターエッジでファイアウォールを構成する必要があります。
    - ポート 500 の任意の UDP パケット
    - ポート 4500 に対する任意の UDP パケット
    - 任意の ESP（IP プロトコル番号 50）パケット
4. IKE キー再生成は、CloudBridge Connector トンネルエンドポイント間で新しい SA を確立するための新しい暗号化キーの再ネゴシエーションであり、サポートされていません。セキュリティアソシエーション（SA）の有効期限が切れると、トンネルは DOWN 状態になります。したがって、SA のライフタイムには非常に大きな値を設定する必要があります。
5. Azure でトンネル構成をセットアップすると、トンネルの Azure エンド（Gateway）のパブリック IP アドレスと PSK が自動的に生成されるため、Citrix ADC でトンネル構成を指定する前に Microsoft Azure を構成する必要があります。この情報は、Citrix ADC でトンネル構成を指定するために必要です。

CloudBridge Connector トンネルの設定

データセンターと Azure の間に CloudBridge Connector トンネルをセットアップするには、データセンターに CloudBridgeVPX/MPX をインストールし、CloudBridge Connector トンネル用に Microsoft Azure を構成してから、データセンターで CloudBridge Connector トンネル用の Citrix ADC アプライアンスを構成する必要があります。

データセンター内の Citrix ADC アプライアンスと Microsoft Azure 間の CloudBridge Connector トンネルを構成するには、次のタスクを実行します。

1. データセンターで **Citrix ADC** アプライアンスをセットアップします。このタスクでは、Citrix ADC 物理アプライアンス（MPX）の展開と構成、またはデータセンター内の仮想化プラットフォームでの Citrix ADC 仮想アプライアンス（VPX）の Provisioning と構成を行います。
2. **CloudBridge Connector** トンネルの **Microsoft** の **Azure** を構成します。このタスクでは、Azure でローカルネットワーク、仮想ネットワーク、および Gateway エンティティを作成します。ローカルネットワークエンティティは、データセンター側の CloudBridge Connector トンネルエンドポイント（Citrix ADC ア

プライアンス）の IP アドレスと、トラフィックが CloudBridge Connector トンネルを通過するデータセンターサブネットを指定します。仮想ネットワークは、Azure 上のネットワークを定義します。仮想ネットワークの作成には、トラフィックが形成される CloudBridge Connector トンネルを通過するサブネットの定義が含まれます。次に、ローカルネットワークを仮想ネットワークに関連付けます。最後に、CloudBridge Connector トンネルの Azure エンドでエンドポイントとなる Gateway を作成します。

3. データセンターで **CloudBridge Connector** トンネル用の **Citrix ADC** アプライアンスを構成します。このタスクでは、データセンターの Citrix ADC アプライアンスで IPSec プロファイル、IP トンネルエンティティ、および PBR エンティティを作成します。IPSec プロファイルエンティティは、CloudBridge Connector トンネルで使用する IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPSec プロトコルパラメータを指定します。IP トンネルは、CloudBridge Connector トンネルエンドポイント（データセンター内の Citrix ADC アプライアンスおよび Azure の Gateway）と CloudBridge Connector トンネルで使用されるプロトコルの両方の IP アドレスを指定します。次に、IPSec プロファイルエンティティを IP トンネルエンティティに関連付けます。PBR エンティティは、CloudBridge Connector トンネルを介して相互に通信する、データセンター内と Azure クラウド内の 2 つのサブネットを指定します。次に、IP トンネルエンティティを PBR エンティティに関連付けます。

CloudBridge Connector トンネルの **Microsoft Azure** の構成

Microsoft Azure で CloudBridge Connector のトンネル構成を作成するには、Microsoft Windows Azure 管理ポータルを使用します。このポータルは、Microsoft Azure でリソースを作成および管理するための Web ベースのグラフィカルインターフェイスです。

Azure クラウドで CloudBridge Connector トンネルの構成を開始する前に、次のことを確認してください。

- Microsoft の Azure のユーザーアカウントがあります。
- あなたは、Microsoft の Azure の概念を理解しています。
- Microsoft の Windows Azure 管理ポータルに精通していること。

データセンターと Azure クラウド間の CloudBridge Connector トンネルを構成するには、Microsoft Windows Azure 管理ポータルを使用して Microsoft Azure で次のタスクを実行します。

- ローカルネットワークエンティティを作成します。データセンターのネットワークの詳細を指定するために、Windows Azure でローカルネットワークエンティティを作成します。ローカルネットワークエンティティは、データセンター側の CloudBridge Connector トンネルエンドポイント（Citrix ADC）の IP アドレスを指定し、トラフィックが CloudBridge Connector トンネルを通過するデータセンターサブネットを指定します。
- 仮想ネットワークを作成します。Azure でネットワークを定義する仮想ネットワークエンティティを作成します。この作業には、プライベートアドレス空間の定義が含まれます。ここでは、アドレス空間で指定された範囲に属するプライベートアドレスおよびサブネットの範囲を指定します。サブネットのトラフィックは CloudBridge Connector トンネルを通過します。次に、ローカルネットワークエンティティを仮想ネットワークエンティティに関連付けます。この関連付けにより、Azure は仮想ネットワークとデータセンターネットワーク間の CloudBridge Connector トンネルの構成を作成できます。この仮想ネットワーク用の Azure

の Gateway (作成予定) は、CloudBridge Connector トンネルの Azure エンドにある CloudBridge エンドポイントになります。次に、作成する Gateway のプライベートサブネットを定義します。このサブネットは、仮想ネットワークエンティティのアドレス空間で指定された範囲に属します。

- **Windows Azure** で **Gateway** を作成します。CloudBridge Connector トンネルの Azure エンドでエンドポイントとなる Gateway を作成します。Azure は、パブリック IP アドレスのプールから、作成された Gateway に IP アドレスを割り当てます。
- **Gateway** のパブリック **IP** アドレスと事前共有キーを収集します。Azure 上の CloudBridge Connector トンネル構成の場合、Gateway のパブリック IP アドレスと事前共有キー (PSK) が Azure によって自動的に生成されます。この情報を書き留めておきます。これは、データセンター内の Citrix ADC で CloudBridge Connector トンネルを構成する場合に必要です。

> 注:
>
> CloudBridge Connector トンネル用に Microsoft Azure を構成するための手順は、Microsoft Azure のリリースサイクルによって時間が経つにつれ、変更されることがあります。最新の手順については、Microsoft Azure のドキュメントを参照してください。

CloudBridge Connector トンネル用のデータセンターでの **Citrix ADC** アプライアンスの構成

データセンターと Azure クラウド間の CloudBridge Connector トンネルを構成するには、データセンター内の Citrix ADC で次のタスクを実行します。Citrix ADC コマンドラインまたは GUI のいずれかを使用できます。

- **IPSec** プロファイルを作成します。IPSec プロファイルエンティティは、CloudBridge Connector トンネルの IPSec プロトコルで使用される IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、PSK などの IPSec プロトコルパラメータを指定します。
- **IPSec** プロトコルを使用して **IP** トンネルを作成し、**IPSec** プロファイルを関連付けます。IP トンネルは、ローカル IP アドレス（Citrix ADC アプライアンス上で構成されたパブリック SNIP アドレス）、リモート IP アドレス（Azure の Gateway のパブリック IP アドレス）、CloudBridge Connector トンネルのセットアップに使用されるプロトコル（IPSec）、および IPSec プロファイルエンティティを指定します。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成し、**IP** トンネルをそのルールに関連付けます。PBR エンティティは、一連の条件と IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレスの範囲と宛先 IP の範囲は、PBR エンティティの条件です。送信元 IP アドレスの範囲を設定して、トラフィックがトンネルを通過するデータセンターのサブネットを指定し、トラフィックが CloudBridge Connector トンネルを通過する Azure サブネットを指定する宛先 IP アドレスの範囲を設定する必要があります。データセンターのサブネットのクライアントから発信され、Azure クラウド上のサブネットのサーバー宛ての要求パケットは、PBR エンティティの送信元および宛先 IP 範囲と一致します。このパケットは、CloudBridge Connector のトンネル処理のために考慮され、PBR エンティティに関連付けられた CloudBridge Connector トンネルを介して送信されます。

GUI では、これらのすべてのタスクを CloudBridge Connector ウィザードと呼ばれる 1 つのウィザードにまとめます。

Citrix ADC コマンドラインを使用して IPSEC プロファイルを作成するには:

コマンドプロンプトで、次のように入力します。

```
add ipsec profile <name> -psk <string> -ikeVersion v1
```

Citrix ADC コマンドラインを使用して IPSEC トンネルを作成し、そのトンネルに IPSEC プロファイルをバインドするには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

```
add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -
ipsecProfileName <string>
```

Citrix ADC コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

```
add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> ipTunnel
 <tunnelName> apply pbrs
```

構成例

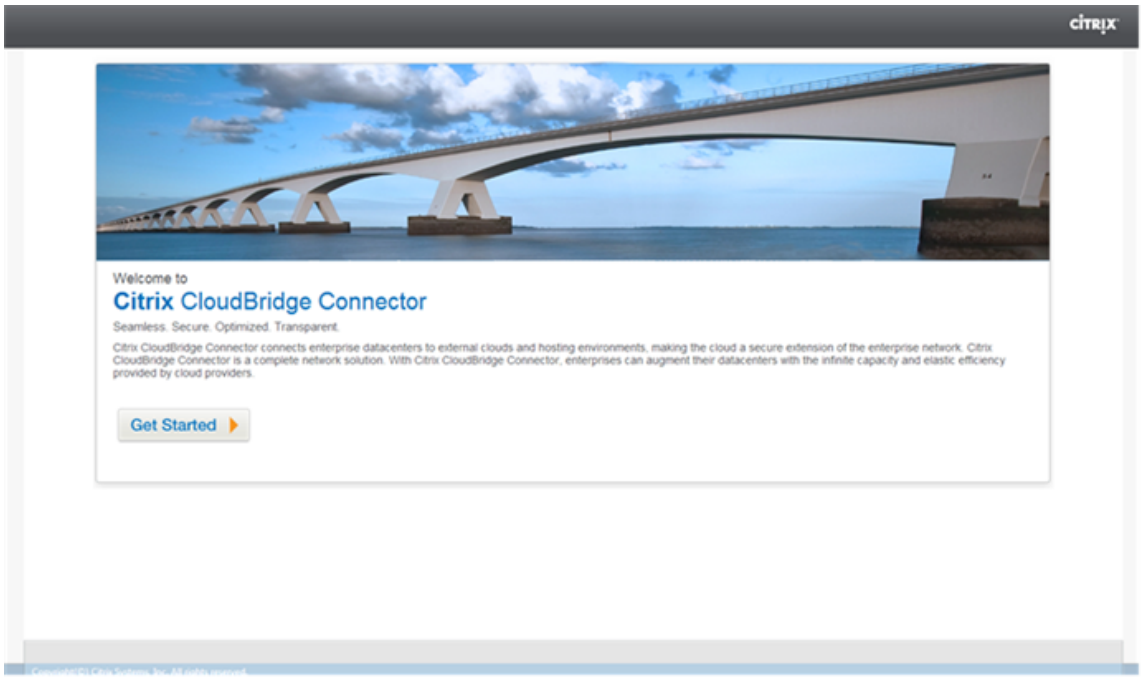
次のコマンドは、「CloudBridge Connector の構成とデータフローの例」で使用される Citrix ADC アプライアンスの CB_Appliance-1 のすべての設定を作成します。

```
1  > add ipsec profile CB_Azure_IPSec_Profile -psk
       DkiMgMdcbqvYREEuIvxsbKkW0FOyDiLM  -ikeVersion v1  - lifetime 31536000
2  Done
3
4  >  add iptunnel CB_Azure_Tunnel 168.63.252.133 255.255.255.255
       66.165.176.15  - protocol IPSEC  - ipsecProfileName
       CB_Azure_IPSec_Profile
5  Done
6
7  > add pbr CB_Azure_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
       10.20.0.0-10.20.255.255  - ipTunnelCB_Azure_Tunnel
8  Done
9
10 > apply pbrs
11 Done
12 <!--NeedCopy-->
```

GUI を使用して Citrix ADC アプライアンスで CloudBridge Connector トンネルを構成するには

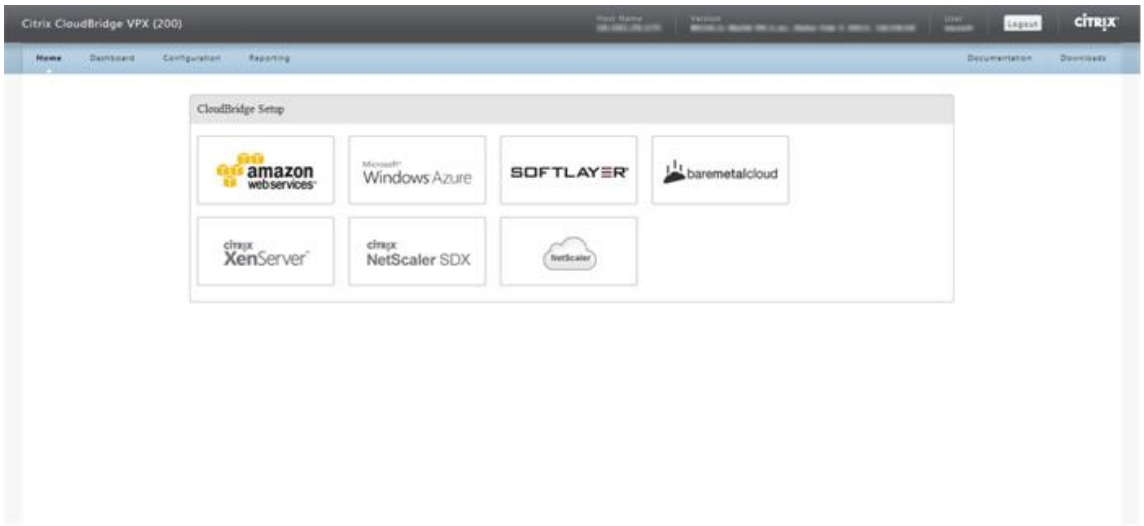
1. Web ブラウザを使用して GUI にアクセスし、データセンター内の Citrix ADC アプライアンスの IP アドレスに接続します。
2. [システム] > [**CloudBridge Connector**] に移動します。

3. 右側のペインの [はじめに] で、[**CloudBridge** の作成/監視] をクリックします。
4. [開始] をクリックします。

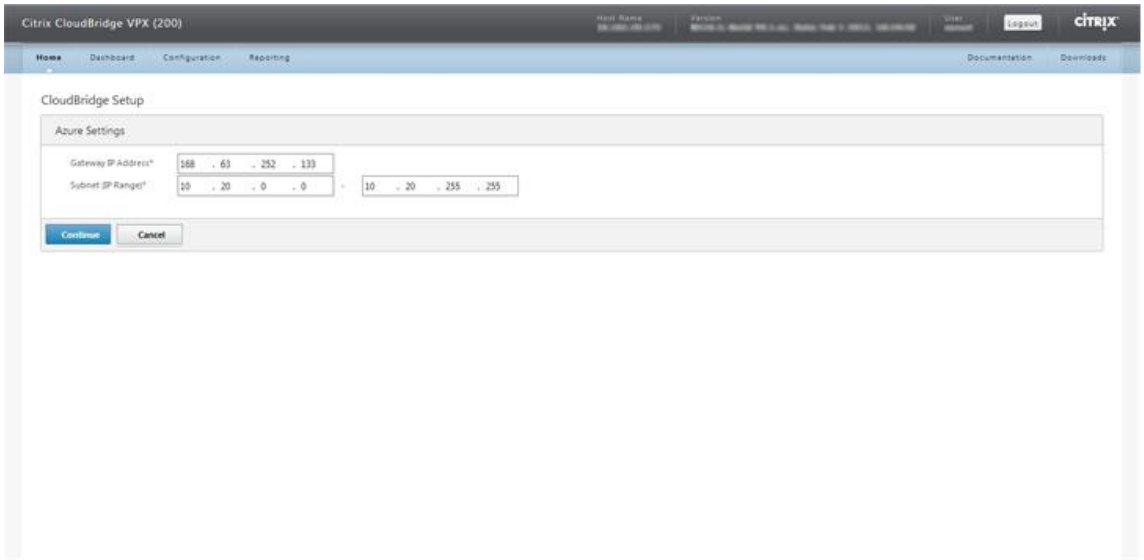


> 注：Citrix ADC アプライアンスで CloudBridge Connector トンネルがすでに設定されている場合、この画面は表示されず、CloudBridge Connector の設定ペインが表示されます。

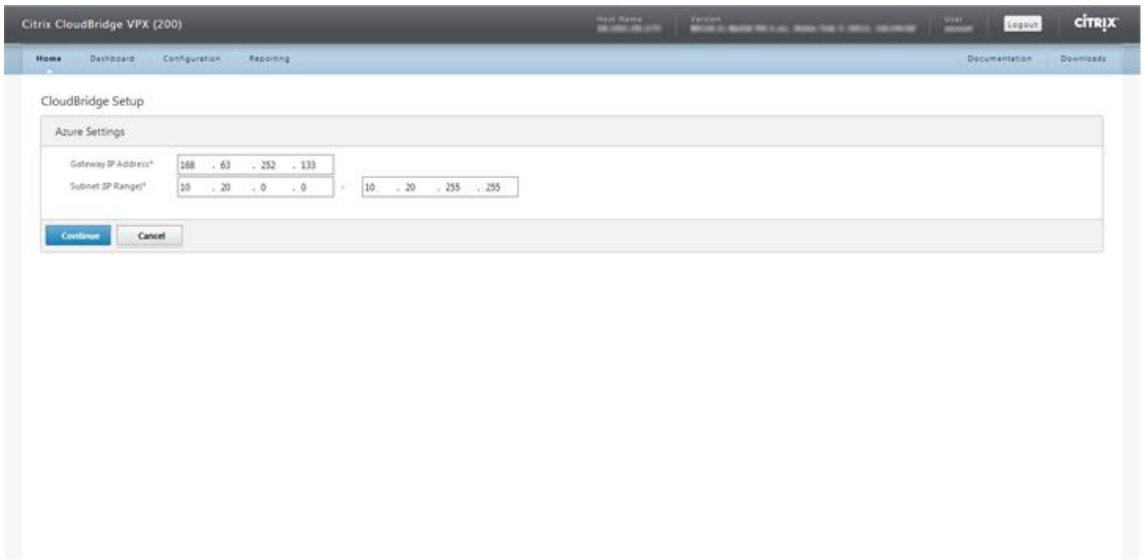
5. [セットアップ] ウィンドウで、[**Microsoft Windows Azure**] をクリックします。



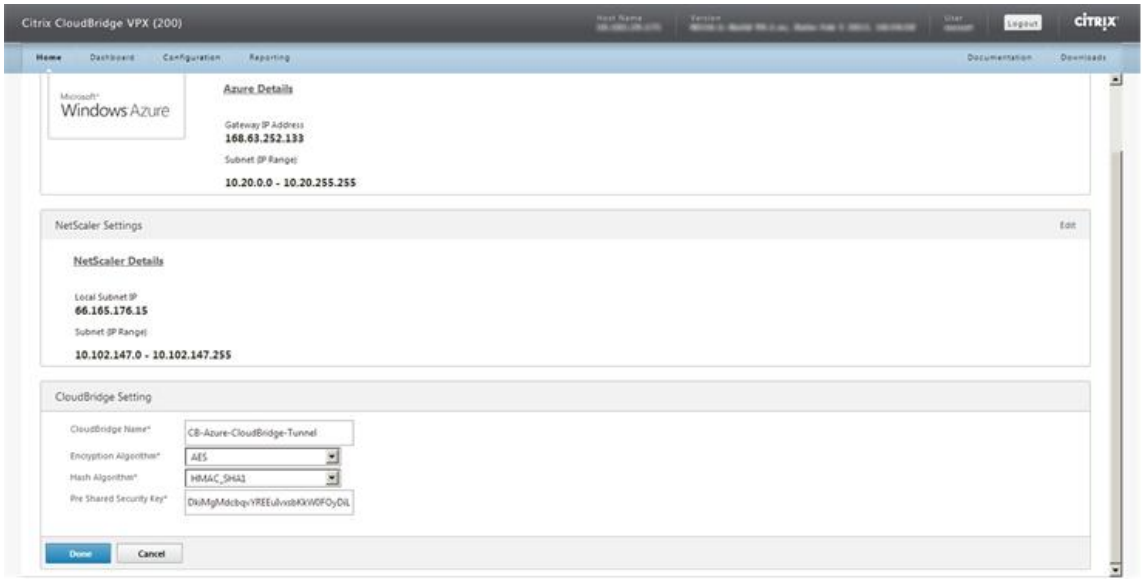
6. [Azure 設定] ウィンドウの [ゲートウェイ **IP** アドレス] フィールドに、Azure ゲートウェイの IP アドレスを入力します。その後、Citrix ADC アプライアンスと Gateway の間に CloudBridge Connector トンネルが設定されます。[サブネット（**IP** 範囲）] テキストボックスで、(Azure クラウド内の）サブネット範囲を指定します。この範囲のトラフィックは、CloudBridge コネクタトンネルを通過します。[続行] をクリックします。



7. [Citrix ADC 設定] ペインの [ローカルサブネット **IP]** ドロップダウンリストから、Citrix ADC アプライアンスで構成されているパブリックにアクセス可能な SNIP アドレスを選択します。[サブネット（**IP** 範囲）**]** テキストボックスで、ローカルサブネット範囲を指定します。この範囲のトラフィックは CloudBridge コネクタトンネルを通過します。［続行］をクリックします。



8. [**CloudBridge** の設定] ペインの [CloudBridge 名] テキストボックスに、作成する CloudBridge の名前を入力します。



9. [暗号化アルゴリズム] および [ハッシュアルゴリズム] ドロップダウンリストから、それぞれ AES アルゴリズムと HMAC_SHA1 アルゴリズムを選択します。[事前共有セキュリティキー] テキストボックスに、セキュリティキーを入力します。
10. ［完了］をクリックします。

CloudBridge Connector トンネルの監視

データセンター内の Citrix ADC アプライアンスと Microsoft Azure の間の CloudBridge Connector トンネルのパフォーマンスを監視するための統計情報を表示できます。Citrix ADC アプライアンスで CloudBridge Connector トンネル統計を表示するには、GUI または Citrix ADC コマンドラインを使用します。Microsoft の Azure で CloudBridge Connector のトンネルの統計情報を表示するには、Microsoft の Windows 管理ポータルを使用します。

Citrix ADC アプライアンスでの **CloudBridge Connector** トンネルの統計情報の表示

Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、CloudBridge Connector トンネルのモニタリングを参照してください。

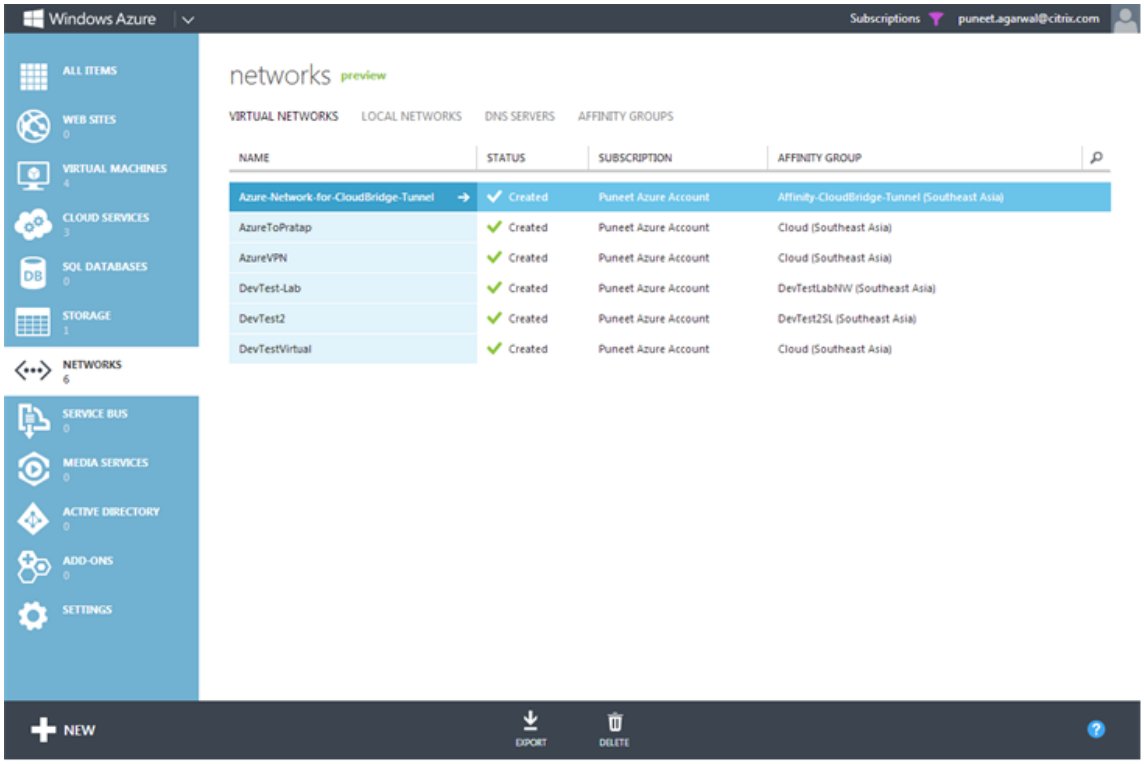
Microsoft の **Azure** での **CloudBridge Connector** のトンネルの統計情報の表示

次の表に、Microsoft Azure で CloudBridge Connector のトンネルを監視するための統計カウンタを示します。

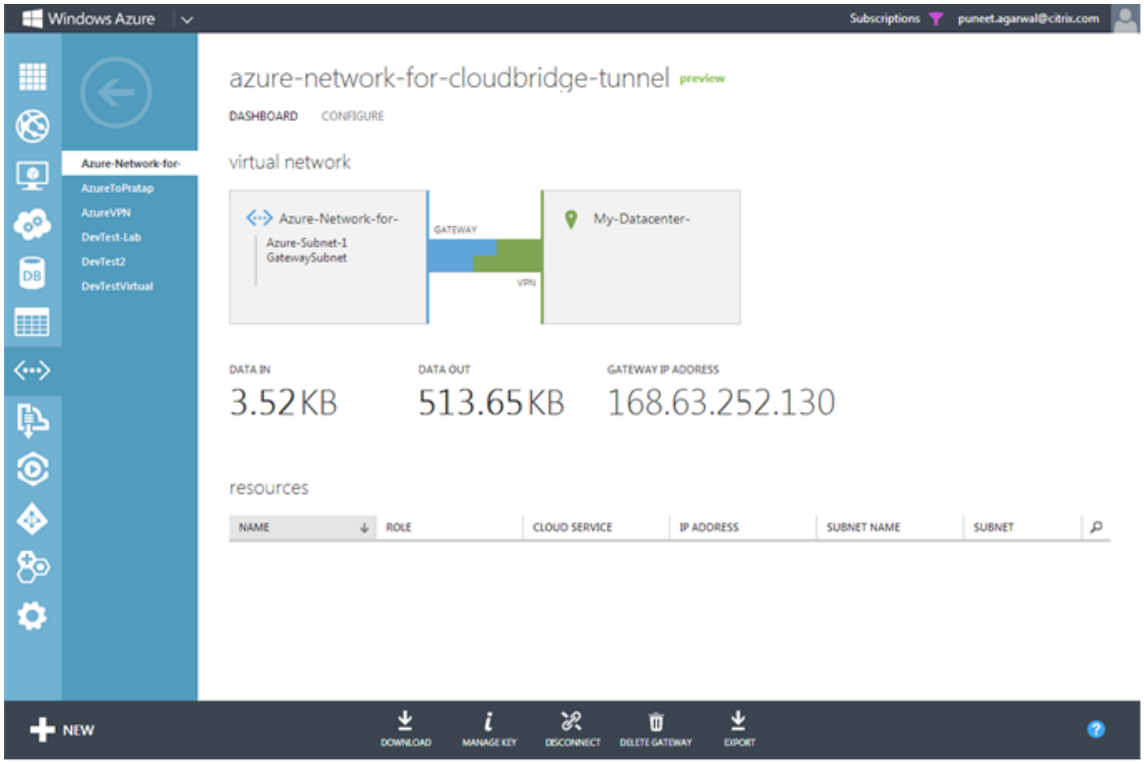
統計カウンタ	Specifies
DATA IN	Gateway が作成されてから CloudBridge Connector トンネルを介して Azure Gateway が受信した合計キロバイト数。
DATA OUT	Gateway が作成されてから、CloudBridge Connector トンネルを介して Azure Gateway が送信した合計キロバイト数。

Microsoft の Windows Azure 管理ポータルを使用して CloudBridge Connector のトンネルの統計情報を表示するには

1. Microsoft Azure アカウントの認証情報を使用して、Windows Azure 管理ポータルにログオンします 。
2. 左側のウィンドウで、[ネットワーク] をクリックします。
3. [仮想ネットワーク] タブの [名前] 列で、統計情報を表示する CloudBridge Connector トンネルに関連付けられた仮想ネットワークエンティティを選択します。



4. 仮想ネットワークの「ダッシュボード」ページで、CloudBridge Connector トンネルの「DATA IN」および「DATA OUT」カウンタを表示します。



データセンターとソフトレイヤーエンタープライズクラウド間の **CloudBridge Connector** トンネルの設定

October 7, 2021

GUI には、データセンター内の Citrix ADC アプライアンスと、SoftLayer エンタープライズクラウド上の Citrix ADC VPX インスタンス間の CloudBridge Connector トンネルを簡単に構成できるウィザードが含まれています。

データセンターで Citrix ADC アプライアンスのウィザードを使用すると、Citrix ADC アプライアンスで作成された CloudBridge Connector トンネル構成が、CloudBridge Connector トンネルの他のエンドポイントまたはピア（SoftLayer 上の Citrix ADC VPX）に自動的にプッシュされます。

データセンターで Citrix ADC アプライアンスのウィザードを使用して、以下の手順を実行して CloudBridge Connector トンネルを構成します。

1. ユーザーログオン資格情報を指定して、Softlayer エンタープライズクラウドに接続します。
2. Citrix ADC VPX アプライアンスを実行している Citrix XenServer を選択します。
3. Citrix ADC VPX アプライアンスを選択します。
4. CloudBridge Connector のトンネルパラメーターを次の対象に指定します。
    - GRE トンネルを設定します。
    - GRE トンネルに IPsec を設定します。
    - 名前を指定して、CloudBridge Connector の論理表現であるネットブリッジを作成します。

- GRE トンネルをネットブリッジにバインドします。

GUI を使用して CloudBridge Connector トンネルを設定するには

1. アプライアンスのアカウント認証情報を使用して、データセンター内の Citrix ADC アプライアンスの GUI にログオンします。
2. [システム] > [**CloudBridge Connector**] に移動します。
3. 右側のペインの [はじめに] で、[**CloudBridge Connector** の作成/監視] をクリックします。
4. [開始] をクリックします。

> 注:
>
> Citrix ADC アプライアンスで CloudBridge Connector トンネルがすでに設定されている場合、この画面は表示されず、CloudBridge Connector のセットアップペインが表示されます。

1. CloudBridge Connector のセットアップペインで、[Softlayer] をクリックし、ウィザードの指示に従います。

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

Citrix ADC アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルの設定

October 7, 2021

Citrix ADC アプライアンスと Cisco デバイス間の CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。Citrix ADC アプライアンスと Cisco IOS デバイスは、CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

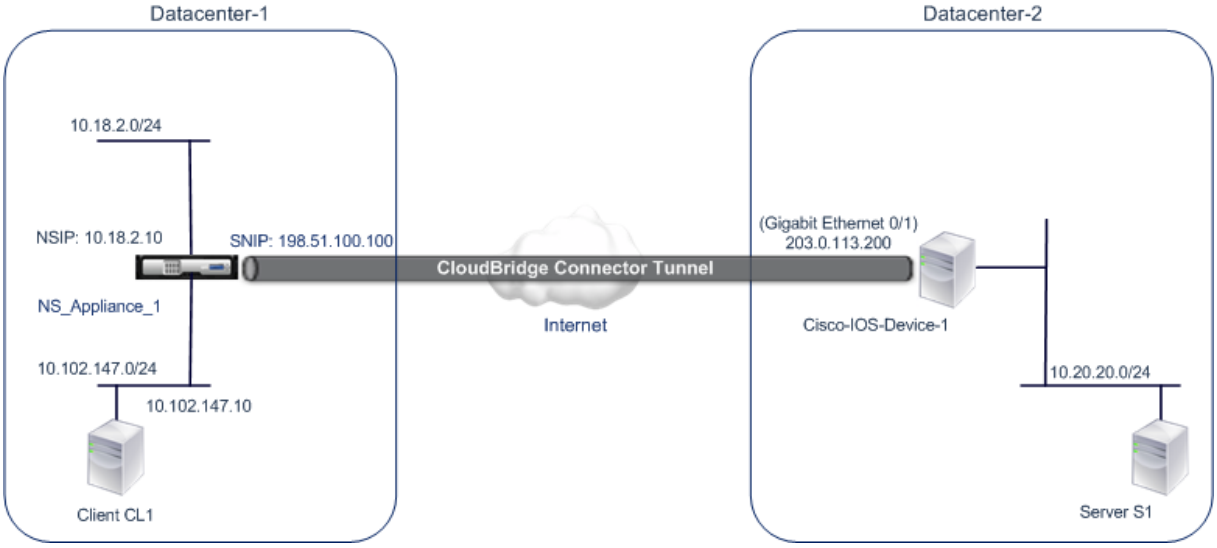
CloudBridge Connector のトンネル設定とデータフローの例

CloudBridge Connector トンネルのトラフィックフローの図のように、CloudBridge Connector トンネルが次のデバイス間でセットアップされる例について考えてみます。

- データセンター 1 として指定されたデータセンター内の Citrix ADC アプライアンス NS_ アプライアンス-1
- データセンター-2 として指定されたデータセンター内の Cisco IOS デバイス Cisco IOS-デバイス 1

NS_Appliance-1 および Cisco-IOS-Device-1 は、CloudBridge Connector トンネルを介したデータセンター 1 とデータセンター 2 のプライベートネットワーク間の通信を可能にします。この例では、NS_Appliance-1 および Cisco-IOS-Device-1 により、データセンター-1 のクライアント CL1 とデータセンター-2 のサーバ S1 との間の通信が CloudBridge Connector トンネルを介して有効になります。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector トンネル設定には、IPSec プロファイルエンティティ NS_Cisco_IPSec_Profile、CloudBridge Connector トンネルエンティティ NS_Cisco_Tunnel、およびポリシーベースルーティング（PBR）エンティティ NS_Cisco_Pbr が含まれます。



詳細については、Citrix ADC アプライアンスと Cisco IOS デバイスの設定間の CloudBridge Connector トンネルを参照してください 。

CloudBridge Connector のトンネル設定について考慮すべきポイント

Citrix ADC アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルを設定する前に、次の点を考慮してください。

- Citrix ADC アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルでは、次の IPSec 設定がサポートされています。

IPSec のプロパティ	設定
IPSec モード	トンネルモード
IKE のバージョン	バージョン 1
IKE DH グループ	DH グループ 2 (1024 ビット MODP アルゴリズム)
IKE 認証方式	事前共有キー
IKE 暗号化アルゴリズム	AES, 3DES

IPSec のプロパティ	設定
IKE ハッシュアルゴリズム	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5
ESP 暗号化アルゴリズム	AES, 3DES
ESP ハッシュアルゴリズム	HMAC SHA1, HMAC SHA256, HMAC SHA256, HMAC SHA256, HMAC MD5

- Citrix ADC アプライアンスと Cisco IOS デバイスで、CloudBridge Connector の両端で同じ IPSec 設定を指定する必要があります。
- Citrix ADC は、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル内）を提供します。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。したがって、Cisco デバイスでは、IKE（IKE ポリシーの作成時）および ESP（IPSec トランスフォームセットの作成時）に同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- 次のことを許可するには、Citrix ADC 側と Cisco デバイス側でファイアウォールを設定する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 に対する任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネル用の **Cisco IOS** デバイスの設定

Cisco IOS デバイスで CloudBridge Connector トンネルを設定するには、Cisco IOS コマンドラインインターフェイスを使用します。これは、Cisco デバイスの設定、モニタリング、および保守のための主要なユーザインターフェイスです。

Cisco IOS デバイスで CloudBridge Connector トンネルの設定を開始する前に、次の点を確認してください。

- Cisco IOS デバイスに、管理者クレデンシャルを持つユーザアカウントがあること。
- Cisco IOS コマンドラインインターフェイスに精通していること。
- Cisco IOS デバイスは稼動しており、インターネットに接続されており、トラフィックが CloudBridge Connector トンネルを介して保護されるプライベートサブネットにも接続されています。

> 注:
>
> Cisco IOS デバイスで CloudBridge Connector トンネルを設定する手順は、Cisco のリリースサイクルによって時間が経つにつれ、変更されることがあります。詳細については、Cisco の公式製品ドキュメントに従うことをお勧めします。詳細については、「IPsec VPN トンネルの構成」を参照してください。

Citrix ADC アプライアンスと **Cisco IOS** デバイスの間に **CloudBridge** コネクタトンネルを設定するには、**Cisco** デバイスの **IOS** コマンドラインで次のタスクを実行します。

- IKE ポリシーを作成します。
- IKE 認証用の事前共有キーを設定します。
- トランスフォームセットを定義し、トンネルモードで IPSec を設定します。
- 暗号アクセスリストを作成する
- クリプトマップを作成する
- インターフェイスにクリプトマップを適用する

次の手順の例では、「CloudBridge Connector の設定とデータフローの例」で説明されている`Cisco IOS device Cisco-IOS-Device-1`の設定を作成します。

IKE ポリシーを作成するには、IKE ポリシー pdf を参照してください。

Cisco IOS コマンドラインを使用して事前共有キーを設定するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
crypto isakmp identity address	Cisco-ios-device-1(config)# crypto isakmp identity address	IKE ネゴシエーション中にピア（Citrix ADC アプライアンス）と通信するときに使用する Cisco IOS デバイスの ISAKMP アイデンティティ（アドレス）を指定します。次に、address キーワードを指定する例を示します。このキーワードでは、IP アドレス 203.0.113.200（Cisco-IOS-Device-1 のギガビットイーサネットインターフェイス 0/1）をデバイスのアイデンティティとして使用します。
crypto isakmp key keystringaddress peer-address	Cisco-ios-device-1 (config)# crypto isakmp key examplepresharedkey address 198.51.100.100	IKE 認証の事前共有キーを指定します。この例では、Citrix ADC アプライアンス NS_ アプライアンス 1（198.51.100.100）で使用する共有キーの例事前共有キーを構成します。Cisco IOS デバイスと Citrix ADC アプライアンス間で IKE 認証を成功させるには、Citrix ADC アプライアンスで同じ事前共有キーを設定する必要があります。

Cisco IOS コマンドラインを使用して暗号アクセスリストを作成するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
access-listaccess-list-number permit IPsource source-wildcard destination destination-wildcard	Cisco-ios-device-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	CloudBridge Connector トンネルを介して IP トラフィックを保護するサブネットを決定する条件を指定します。この例では、サブネット 10.20.0/24（Cisco-IOS-デバイス 1 側）および 10.102.147.0/24（NS_ アプライアンス 1 側）からのトラフィックを保護するように、アクセスリスト 111 を設定します。

Cisco IOS コマンドラインを使用して、トランスフォームを定義して **IPSec** トンネルモードを設定するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。
コマンド	例	コマンドの説明
–	–	–
暗号化 ipsec transform-setname esp_authentication_Transform Esp_Authentication_transform 注:Esp_Authentication_Transform は次の値を取ることができます:esp-sha-hmac, esp-sha256-hmac, esp-sha512-hmac, esp-md5-hmac。Esp_encryption_Transform は次の値を取ることができます:esp-aes または esp-3des	Cisco-ios-device-1(config)# crypto ipsec transform-set NS-CISCO-TS esp-sha256-hmac esp-3des	トランスフォームセットを定義し、CloudBridge Connector トンネルピア間のデータ交換時に使用する ESP ハッシュアルゴリズム（認証用）と ESP 暗号化アルゴリズムを指定します。次の例では、トランスフォームセット NS-CISCO-TS を定義し、ESP 認証アルゴリズムを esp-sha256-hmac、ESP 暗号化アルゴリズムを esp-3des に指定しています。
モードトンネル	Cisco-iOS-device-1（設定暗号トランス）# モードトンネル	IPSec をトンネルモードに設定します。
exit	Cisco-iOS-device-1 (設定-crypto-trans) # 終了、Cisco-iOS-device-1 (設定) #	グローバル構成モードに戻ります。

Cisco IOS コマンドラインを使用してクリプトマップを作成するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
crypto mapmap-name seq-num ipsec-isakmp	Cisco-ios-device-1 (config)# crypto map NS-CISCO-CM 2 ipsec-isakmp	クリプトマップ構成モードを開始し、クリプトマップのシーケンス番号を指定し、IKE を使用してセキュリティアソシエーション（SA）を確立するようにクリプトマップを設定します。次に、クリプトマップ NS-CISCO-CM のシーケンス番号 2 および IKE を設定する例を示します。
set peer ip-address	Cisco-ios-device-1 (config-crypto-map)# set peer 172.23.2.7	ピア（Citrix ADC アプライアンス）を IP アドレスで指定します。この例では、198.51.100.100 を指定します。これは、Citrix ADC アプライアンス上の CloudBridge Connector のエンドポイント IP アドレスです。
match addressaccess-list-id	Cisco-ios-device-1 (config-crypto-map)# match address 111	拡張アクセスリストを指定します。このアクセスリストは、CloudBridge Connector トンネルを介して IP トラフィックを保護するサブネットを決定する条件を指定します。次に、アクセスリスト 111 を指定する例を示します。
トランスフォームセットのトランスフォームセット名の設定	Cisco-ios-device-1 (config-crypto-map)# set transform-set NS-CISCO-TS	このクリプトマップエントリに許可されるトランスフォームセットを指定します。次に、トランスフォームセット NS-CISCO-TS を指定する例を示します。
exit	Cisco-ios-device-1 (config-crypto-map)# exit Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

Cisco IOS コマンドラインを使用してインターフェイスにクリプトマップを適用するには、次の手順を実行します。

Cisco IOS デバイスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
interfaceinterface-ID	Cisco-ios-device-1(config)# interface GigabitEthernet 0/1	クリプトマップを適用する物理インターフェイスを指定し、インターフェイス構成モードを開始します。この例では、Cisco デバイス Cisco-IOS-Device-1 のギガビットイーサネットインターフェイス 0/1 を指定します。IP アドレス 203.0.113.200 はすでにこのインターフェイスに設定されています。
crypto mapmap-name	Cisco-ios-device-1 (config-if)# crypto map NS-CISCO-CM	物理インターフェイスにクリプトマップを適用します。次に、クリプトマップ NS-CISCO-CM を適用する例を示します。
exit	Cisco-ios-device-1 (config-if)# exit, Cisco-ios-device-1 (config)#	グローバル構成モードに戻ります。

CloudBridge Connector トンネル用の **Citrix ADC** アプライアンスの構成

Citrix ADC アプライアンスと Cisco IOS デバイス間の CloudBridge Connector トンネルを設定するには、Citrix ADC アプライアンスで次のタスクを実行します。Citrix ADC コマンドラインまたは Citrix ADC グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- IPSec プロファイルを作成します。
- IPSec プロトコルを使用する IP トンネルを作成し、IPSec プロファイルを関連付けます。
- PBR ルールを作成し、IP トンネルに関連付けます。

Citrix ADC コマンドラインを使用して **IPSEC** プロファイルを作成するには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1`
- `show ipsec profile <name>`

Citrix ADC コマンドラインを使用して **IPSEC** トンネルを作成し、**IPSEC** プロファイルをバインドするには:

コマンドプロンプトで、次のように入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`

- add ipTunnel <name>

Citrix ADC コマンドラインを使用して **PBR** ルールを作成し、**IPSEC** トンネルをそれにバインドするには、次の手順を実行します。

コマンドプロンプトで、次のように入力します。

- add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>
- apply pbrs
- show pbrs <pbrName>

次のコマンドは、「**CloudBridge Connector** の設定とデータフローの例」で説明されている**Citrix ADC appliance NS_Appliance-1**の設定を作成します。

```
1      >  add ipsec profile NS_Cisco_IPSec_Profile -psk
          examplepresharedkey -ikeVersion v1  - lifetime 315360  - encAlgo 3
          DES
2      Done
3      >  add iptunnel NS_Cisco_Tunnel 203.0.113.200 255.255.255.255
          198.51.100.100  - protocol IPSEC  - ipsecProfileName
          NS_Cisco_IPSec_Profile
4
5      Done
6      > add pbr NS_Cisco_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
          10.20.0.0-10.20.255.255  - ipTunnel NS_Cisco_Tunnel
7
8      Done
9      > apply pbrs
10
11     Done
12 <!--NeedCopy-->
```

GUI を使用して **IPSEC** プロファイルを作成するには、次の手順を実行します。

1. [システム] > [**CloudBridge Connector**] > [**IPsec** プロファイル**]** に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**IPSec** プロファイルの追加] ダイアログボックスで、次のパラメータを設定します。
   - Name
   - 暗号化アルゴリズム
   - ハッシュアルゴリズム
   - IKE プロトコルバージョン
4. 相互に認証するために **2** つの **CloudBridge Connector** トンネルピアで使用される **IPsec** 認証方法を設定します。事前共有キー認証方法を選択し、[事前共有キーが存在する] パラメータを設定します。

5. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して **IP** トンネルを作成し、**IPSEC** プロファイルをそのトンネルにバインドするには、次の手順を実行します。

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2. [**IPv4** トンネル] タブで、[追加] をクリックします。
3. [**Add IP Tunnel**] ダイアログボックスで、次のパラメータを設定します。
   - Name
   - リモート IP
   - リモートマスク
   - ローカル IP タイプ ([ローカル IP タイプ] ドロップダウンリストで、[サブネット IP] を選択します)。
   - ローカル IP（選択した IP タイプの構成済みの IP はすべて、[ローカル IP] ドロップダウンリストに表示されます。リストから目的の IP を選択します)。
   - プロトコル
   - IPSec プロファイル
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. **[PBR** の作成**]** ダイアログボックスで、次のパラメータを設定します。
   - Name
   - 操作（アクション）
   - ネクストホップタイプ（IP トンネルの選択）
   - IP トンネル名
   - 送信元 IP の低
   - 送信元 IP 高
   - 宛先 IP の低
   - 宛先 IP 高
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して **PBR** を適用するには、次の手順を実行します。

1. **System** > **Network** > **PBRs** に移動します。
2. **[PBR]** タブで **PBR** を選択し、**[**アクション**]** リストで **[**適用**]** を選択します。

Citrix ADC アプライアンスの対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge Connector] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

CloudBridge Connector トンネルのモニタリング

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

Citrix ADC アプライアンスとフォーティネットフォーティゲートアプライアンス間の CloudBridge Connector トンネルの構成

October 7, 2021

Citrix ADC アプライアンスと Fortinet FortiGate アプライアンスの間に CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。Citrix ADC アプライアンスと FortiGate アプライアンスは、CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

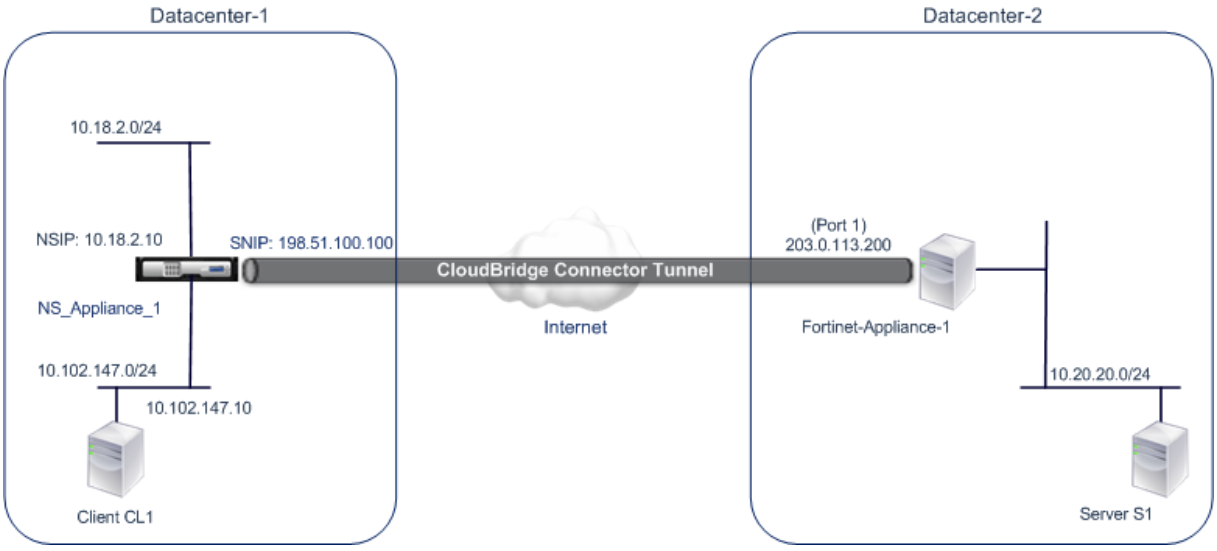
CloudBridge Connector トンネル設定の例

CloudBridge Connector トンネルのトラフィックフローの図のように、CloudBridge Connector トンネルが次のデバイス間でセットアップされる例について考えてみます。

- データセンター 1 として指定されたデータセンター内の Citrix ADC アプライアンス NS_ アプライアンス-1
- データセンター-2 として指定されたデータセンター内の FortiGate アプライアンス-1

NS_Appliance-1 および FortiGate-Appliance-1 は、データセンター 1 とデータセンター-2 のプライベートネットワーク間で CloudBridge Connector トンネルを介して通信できるようにします。この例では、NS_ アプライアンス 1 と FortiGate-アプライアンス 1 は、データセンター-1 のクライアント CL1 と、データセンター-2 のサーバー S1 との間の通信を CloudBridge Connector トンネルを介して有効にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル設定には、IPSec プロファイルエンティティ NS_Fortinet_IPSec_Profile、CloudBridge Connector トンネルエンティティ NS_Fortinet_Tunnel、およびポリシーベースルーティング (PBR) エンティティ NS_Fortinet_Pbr が含まれます。



詳細については、CloudBridge Connector トンネル設定表 pdf を参照してください。

データセンター 2 の Fortinet FortiGate-Appliance-1 の設定については、表を参照してください。

CloudBridge Connector のトンネル設定について考慮すべきポイント

Citrix ADC アプライアンスと FortiGate アプライアンスの間に CloudBridge Connector トンネルを構成する前に、次の点を考慮してください。

- 以下の IPSec 設定は、Citrix ADC アプライアンスと FortiGate アプライアンス間の CloudBridge Connector トンネルでサポートされています。

IPSec のプロパティ	設定
IPSec モード	トンネルモード
IKE のバージョン	バージョン 1
IKE DH グループ	DH グループ 2 (1024 ビット MODP アルゴリズム)
IKE 認証方式	事前共有キー
IKE 暗号化アルゴリズム	AES
IKE ハッシュアルゴリズム	HMAC SHA1
ESP 暗号化アルゴリズム	AES
ESP ハッシュアルゴリズム	HMAC SHA1

- CloudBridge Connector の両端の Citrix ADC アプライアンスと FortiGate アプライアンスで同じ IPSec 設定を指定する必要があります。

- Citrix ADC は、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル内）を提供します。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。したがって、FortiGate アプライアンスでは、IKE（フェーズ 1 設定）および ESP（フェーズ 2 設定）で同じハッシュアルゴリズムと暗号化アルゴリズムを指定する必要があります。
- Citrix ADC 端と FortiGate 端でファイアウォールを構成して、次のことを許可する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 に対する任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット
- FortiGate アプライアンスは、ポリシーベースとルートベースの 2 種類の VPN トンネルをサポートしています。FortiGate アプライアンスと Citrix ADC アプライアンスの間では、ポリシーベースの VPN トンネルのみがサポートされます。

CloudBridge Connector トンネル用の **FortiGate** アプライアンスの設定

FortiGate アプライアンスで CloudBridge Connector トンネルを設定するには、FortiGate アプライアンスを設定、監視、および保守するための主要なユーザーインターフェイスである Fortinet ウェブベースのマネージャーを使用します。

FortiGate アプライアンスで CloudBridge Connector のトンネル設定を開始する前に、次の点を確認してください。

- FortiGate アプライアンスで管理者資格情報を持つユーザーアカウントがあること。
- あなたは Fortinet の Web ベースのマネージャーに精通しています。
- FortiGate アプライアンスは稼働しており、インターネットに接続されており、トラフィックが CloudBridge Connector トンネルを介して保護されるプライベートサブネットにも接続されています。

> 注
>
> FortiGate アプライアンスで CloudBridge Connector トンネルを設定する手順は、Fortinet のリリースサイクルによって時間が経つにつれ、変更されることがあります。IPsec VPN トンネルの構成に関する公式の Fortinet 製品マニュアルに従うことをお勧めします。

Citrix ADC アプライアンスと FortiGate アプライアンス間の CloudBridge Connector トンネルを構成するには、Fortinet ウェブベースのマネージャーを使用して、FortiGate アプライアンスで以下のタスクを実行します。

- ポリシーベースの **IPSec VPN** 機能を有効にします。FortiGate アプライアンスでポリシーベースの VPN トンネルを作成するには、この機能を有効にします。FortiGate アプライアンスと Citrix ADC アプライアンスの間では、ポリシーベースの VPN トンネルのみがサポートされています。FortiGate アプライアンスのポリシーベースの VPN トンネル設定には、フェーズ 1 の設定、フェーズ 2 の設定、および IPSec セキュリティポリシーが含まれます。
- フェーズ **1** のパラメータを定義します。フェーズ 1 パラメータは、Citrix ADC アプライアンスへの安全なトンネルを形成する前に、FortiGate アプライアンスによって IKE 認証に使用されます。

- フェーズ **2** のパラメータを定義します。フェーズ 2 パラメーターは、IKE セキュリティアソシエーション (SA) を確立することにより、Citrix ADC アプライアンスへの安全なトンネルを形成するために FortiGate アプライアンスによって使用されます。
- プライベートサブネットを指定します。FortiGate 側と Citrix ADC 側のプライベートサブネットを定義します。このサブネットはトンネルを介して転送されます。
- トンネルの **IPSec** セキュリティポリシーを定義します。セキュリティポリシーでは、FortiGate アプライアンス上のインターフェイス間で IP トラフィックが通過することを許可します。IPSec セキュリティポリシーでは、プライベートサブネットへのインターフェイスと、トンネルを介して Citrix ADC アプライアンスを接続するインターフェイスを指定します。

Fortinet Web ベースマネージャを使用してポリシーベースの IPSec VPN 機能を有効にするには

1. [システム] > [設定] > [機能] に移動します。
2. [機能設定**]** ページで、[詳細を表示] を選択し、[ポリシーベースの **IPSec VPN**] をオンにします。

Fortinet Web ベースのマネージャーを使用してフェーズ 1 のパラメーターを定義するには

1. [**VPN**] > [**IPsec**] > [自動キー **(IKE)**] に移動し、[フェーズ **1** の作成] をクリックします。
2. [新しいフェーズ **1**] ページで、次のパラメータを設定します。
    - [名前]: このフェーズ 1 構成の名前を入力します。
    - リモートゲートウェイ:[静的 *IP* アドレス*]* を選択します。
    - モード：メイン（*ID* 保護）を選択します。
    - 認証方法:[事前共有キー] を選択します。
    - 事前共有キー: 事前共有キーを入力します。Citrix ADC アプライアンスでは、同じ事前共有キーを構成する必要があります。
    - ピアオプション：Citrix ADC アプライアンスを認証するための次の IKE パラメータを設定します。
        - IKE バージョン: *1* を選択します。
        - モード設定: このオプションが選択されている場合は、このオプションを選択解除します。
        - [ローカルゲートウェイ IP]: [メインインターフェイス *IP*] を選択します。
        - P1 提案：Citrix ADC アプライアンスへの安全なトンネルを形成する前に、IKE 認証の暗号化アルゴリズムと認証アルゴリズムを選択します。
            * 1-暗号化：*AES128* を選択します。
            * 認証: *SHA1* を選択します。
            * [キーライフ]: フェーズ 1 のキー寿命の時間 (秒単位) を入力します。
            * DH グループ：2 を選択します。
        - X-Auth: [無効にする] を選択します。
        - ディードピアの検出: このオプションを選択します。
3. [**OK**] をクリックします。

Fortinet Web ベースのマネージャーを使用してプライベートサブネットを指定するには

1. [ファイアウォールオブジェクト**]** > [アドレス] > [アドレス] に移動し、[新規作成] を選択します。
2. [新しいアドレス] ページで、次のパラメータを設定します。
    - 名前:FortiGate 側サブネットの名前を入力します。

- タイプ:[サブネット] を選択します。
- サブネット/ IP 範囲：FortiGate 側サブネットのアドレスを入力します。
- [Interface]：このサブネットへのローカルインターフェイスを選択します。

3. [**OK**] をクリックします。
4. 手順 1〜3 を繰り返して、Citrix ADC 側のサブネットを指定します。

Fortinet Web ベースのマネージャーを使用してフェーズ 2 のパラメーターを定義するには

1. [**VPN**] > [**IPsec**] > [自動キー **(IKE)**] に移動し、[フェーズ **2** の作成] をクリックします。
2. [新しいフェーズ **2**] ページで、次のパラメータを設定します。
   - [名前]: このフェーズ 2 構成の名前を入力します。
   - フェーズ 1: ドロップダウンリストからフェーズ 1 構成を選択します。
3. 「詳細」をクリックし、次のパラメータを設定します。
   - P2 提案：Citrix ADC アプライアンスへの安全なトンネルを形成するための暗号化アルゴリズムと認証アルゴリズムを選択します。
     - 1-暗号化：*AES128* を選択します。
     - 認証: *SHA1* を選択します。
     - [リプレイ検出を有効にする]: このオプションを選択します。
     - Perfect Forward Secrecy (PFS) を有効にする: このオプションを選択します。
     - DH グループ：*2* を選択します。
   - [キーライフ]: フェーズ 2 のキー寿命の時間 (秒単位) を入力します。
   - 自動キー Keep-Alive: このオプションを選択します。
   - 自動ネゴシエート: このオプションを選択します。
   - クイックモードセレクター：トラフィックがトンネルを通過する FortiGate 側と Citrix ADC 側のプライベートサブネットを指定します。
     - 送信元アドレス: ドロップダウンリストから FortiGate-側のサブネットを選択します。
     - 送信元ポート：*0* を入力します。
     - 宛先アドレス：ドロップダウンリストから Citrix ADC 側のサブネットを選択します。
     - 宛先ポート：*0* を入力します。
     - プロトコル：*0* を入力します。
4. [**OK**] をクリックします。

Fortinet Web ベースのマネージャーを使用して IPSec セキュリティポリシーを定義するには

1. [ポリシー**] > [**ポリシー ****] > [**ポリシー**]** に移動し、**[**新規作成 **] をクリックします。
2. [ポリシーの編集**]** ページで、次のパラメータを設定します。
   - ポリシータイプ：[*VPN*] を選択します。
   - [ポリシーサブタイプ]: [*IPSec*] を選択します。
   - [ローカルインターフェイス]: 内部 (プライベート) ネットワークへのローカルインターフェイスを選択します。
   - Local Protected Subnet：トラフィックがトンネルを通過するドロップダウンリストから FortiGate-側のサブネットを選択します。

- 発信 VPN インターフェイス: 外部 (パブリック) ネットワークへのローカルインターフェイスを選択します。
- リモート保護サブネット： トンネルを通過するトラフィックの Citrix ADC 側のサブネットをドロップダウンリストから選択します。
- スケジュール: 特定の要件を満たすために変更が必要でない限り、デフォルト設定 (常に) を維持します。
- サービス: 特定の要件を満たすために変更が必要でない限り、デフォルト設定 (*ANY*) を維持します。
- [VPN トンネル]: [既存の使用] を選択し、ドロップダウンリストからトンネルを選択します。
- [リモートサイトからのトラフィックの開始を許可する]: リモートネットワークからのトラフィックがトンネルの開始を許可するかどうかを選択します。

3. [**OK**] をクリックします。

CloudBridge Connector トンネル用の **Citrix ADC** アプライアンスの構成

Citrix ADC アプライアンスと FortiGate アプライアンスの間に CloudBridge Connector トンネルを設定するには、Citrix ADC アプライアンスで以下のタスクを実行します。Citrix ADC コマンドラインまたは Citrix ADC グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- **IPSec** プロファイルを作成します。IPSec プロファイルエンティティは、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、認証方法などの IPSec プロトコルパラメータを指定します。このパラメータは、CloudBridge Connector トンネル内の IPSec プロトコルで使用されます。
- **IPSec** プロトコルを使用する **IP** トンネルを作成し、**IPSec** プロファイルを関連付けます。IP トンネルは、ローカル IP アドレス（Citrix ADC アプライアンス上で構成された CloudBridge Connector トンネルエンドポイント IP アドレス（SNIP タイプ））、リモート IP アドレス（FortiGate アプライアンス上で構成された CloudBridge Connector トンネルエンドポイント IP アドレス)、CloudBridge のセットアップに使用するプロトコル（IPSec）を指定します。コネクタトンネル、および IPSec プロファイルエンティティ。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成し、**IP** トンネルに関連付けます。PBR エンティティは、一連のルールと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレスの範囲と宛先 IP アドレスの範囲は、PBR エンティティの条件です。送信元 IP アドレスの範囲を設定して、トンネル経由でトラフィックを保護する Citrix ADC 側のサブネットを指定します。宛先の IP アドレス範囲を設定して、トンネル経由でトラフィックを保護する FortiGate アプライアンス側のサブネットを指定します。

Citrix ADC コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

Citrix ADC コマンドラインを使用して IPSEC トンネルを作成し、そのトンネルに IPSEC プロファイルをバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName** <string>`
- `show ipTunnel <name>`

Citrix ADC コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

コマンドプロンプトで入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

GUI を使用して IPSEC プロファイルを作成するには

1. [システム] > [**CloudBridge Connector**] > [**IPsec** プロファイル**]** に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**IPSec** プロファイルの追加**]** ページで、次のパラメータを設定します。
   - Name
   - 暗号化アルゴリズム
   - ハッシュアルゴリズム
   - IKE プロトコルバージョン
   - Perfect Forward Secrecy（このパラメータを有効にする）
4. 2 つの CloudBridge Connector トンネルピアが相互認証に使用する IPSec 認証方法を設定します。事前共有キー認証方法を選択し、事前共有キーが存在パラメータを設定します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2. [**IPv4** トンネル] タブで、[追加] をクリックします。
3. **[IP** トンネルの追加**]** ページで、次のパラメータを設定します。
   - Name
   - リモート IP
   - リモートマスク
   - [ローカル IP タイプ] ([ローカル IP タイプ] ドロップダウンリストで、[サブネット *IP*] を選択します)。
   - ローカル IP（選択した IP タイプの構成済みの IP アドレスはすべて、[ローカル IP] ドロップダウンリストに表示されます。リストから目的の IP を選択します)。
   - プロトコル
   - IPSec プロファイル
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。

2. [**PBR**] タブで、[追加] をクリックします。
3. [PBR の作成（**Create PBR** ）] ページで、次のパラメータを設定します。
    - Name
    - 操作（アクション）
    - ネクストホップタイプ（*IP* トンネルの選択）
    - IP トンネル名
    - 送信元 IP の低
    - 送信元 IP 高
    - 宛先 IP の低
    - 宛先 IP 高
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

Citrix ADC アプライアンスの対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。

CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge Connector] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

次のコマンドは、「CloudBridge Connector 構成の例」で、Citrix ADC アプライアンスの NS_Appliance-1 の設定を作成します。

```
1      >  add ipsec profile NS_Fortinet_IPSec_Profile -psk
          examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
          HMAC_SHA1  - lifetime 315360 -perfectForwardSecrecy ENABLE
2
3       Done
4      >  add iptunnel NS_Fortinet_Tunnel 203.0.113.200 255.255.255.255
          198.51.100.100  - protocol IPSEC  - ipsecProfileName
          NS_Fortinet_IPSec_Profile
5
6       Done
7      > add pbr NS_Fortinet_Pbr -srcIP 10.102.147.0-10.102.147.255  -
          destIP 10.20.0.0-10.20.255.255  - ipTunnel NS_Fortinet_Tunnel
8
9       Done
10     > apply pbrs
11
12      Done
13 <!--NeedCopy-->
```


CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

CloudBridge Connector のトンネルの診断とトラブルシューティング

October 7, 2021

CloudBridge Connector トンネル設定に問題がある場合は、トンネルを設定する前にすべての前提条件を満たしていることを確認してください。その場合は、トンネルエンドポイント IP アドレス、NAT 設定、トンネルのセットアップ方法、またはデータトラフィックに問題がある可能性があります。

CloudBridge Connector トンネルのトラブルシューティング

CloudBridge Connector トンネルが正しく機能しない場合は、トンネルの確立またはデータトラフィックに問題がある可能性があります。発生している問題のタイプが不明な場合は、ログファイルでエラーメッセージを探し、そのエラーメッセージがトンネル確立の問題の一覧に含まれているかどうかを確認します。エラーメッセージが見つからない場合は、データトラフィックに関連する可能性のある問題の一覧を確認してください。

トンネルの確立に関する問題

IPSec トンネルの構成要件が満たされ、CloudBridge Connector トンネルが設定された後、トンネルのステータスが UP でない場合は、トンネルのエンドポイントとして設定されている片方または両方の Citrix ADC アプライアンスの iked.log ファイルでデバッグ情報を探します。

いずれかのアプライアンスで、Citrix ADC シェルプロンプトで次のコマンドを入力します。

```
cat /tmp/iked.debug | tee /var/iked.log
```

トラブルシューティング pdf には、一般的なエラーとその解決策が記載されています。

データトラフィックに関連する問題

CloudBridge Connector トンネルのデータがトンネルのエンドポイント間で適切に交換されない場合は、次の操作を行います。

- GRE プロトコルと IPSec プロトコルを使用する CloudBridge Connector トンネルの場合:
  - 両方の CloudBridge Connector トンネルエンドポイントで L2 モードが有効になっていることを確認します。L2 モードを有効にするには、Citrix ADC コマンドラインインターフェイスで次のコマンドを入力します。


```
enable mode L2
```

* CloudBridge Connectorのトンネルエンドポイントの1つがCloudBridge仮想アプライアンス(VPX)であり、VMware ESXi Hypervisorでプロビジョニングされている場合は、CloudBridge VPXアプライアンスに関連付けられたvSwitchに対して、無差別モードが[承諾]に設定されていることを確認します。

- CloudBridge Connectorトンネルを介してVLANを拡張する場合は、各トンネルエンドポイントの拡張VLANエンティティで1対1のマッピングを確認します。
- IPトンネルエンティティが、各トンネルエンドポイントの正しいnetbridgeエンティティにバインドされていることを確認します。
- Citrix ADCコマンドラインインターフェイスで次のコマンドを入力して、ピアCloudBridge Connectorトンネルエンドポイントに対するARPエントリがローカルトンネルのエンドポイントに存在することを確認します。

  ```
  show arp
  ```

- 出力に不完全なARPエントリが表示されている場合、双方向トラフィックはトンネルを通過しません。双方向トラフィックが流れている場合、ARPエントリには、トンネルの反対側にあるデバイスのトンネルインターフェイスの名前が表示されます。
- 両方のトンネルエンドポイントからIPトンネルエンティティを削除し、同じパラメータで再度追加しますが、IPSecプロファイルをNONEに設定して、トンネルがGREプロトコルだけを使用するようにします。

  IPトンネル（GREプロトコルを使用する）で次のことを確認した後、各トンネルエンドポイントの各IPトンネルエンティティに対して有効なIPSecプロファイルを指定して、IPSecパラメータを使用してトンネルを設定します。

  トンネルを通過する適切なPINGまたはTCPフロー。

  トンネルを通過するデータトラフィックの適切なフロー。

  設定されたトンネル（GREプロトコルとIPSecプロトコルを使用する）がUP状態になった後、データトラフィックがトンネルを適切に通過せず、トンネルエンドポイントの一方または両方の前にNATデバイスが配置されている場合は、NATデバイスの入力パケットと出力パケットを分析します。

- Citrix ADCアプライアンスがルーターまたはゲートウェイとして使用されている場合。
  - Citrix ADCアプライアンスでL3モードが有効になっていることを確認します。L3モードを有効にするには、CloudBridgeコマンドラインで次のコマンドを実行します。
  - 有効モードL3
  - サブネットがnetbridgeエンティティにバインドされている場合は、正しいIPトンネルエンティティもネットブリッジにバインドされていることを確認します。
  - Citrix ADCコマンドラインで次のコマンドを実行して、パケット（入力と出力）がドロップされている場所を確認します。

    ```
    stat ipsec counters
    ```

  - 両方のトンネルエンドポイントに正しいルートが設定されていることを確認します。
  - Citrix ADCアプライアンスの前にNATデバイスが展開されていない場合は、ファイアウォールがESP（IPプロトコル番号50）パケットとポート4500のUDPパケットを許可するように構成されているこ

とを確認します。

上記のいずれの方法でもトンネルエンドポイント間のトラフィック交換が成功しない場合は、Citrix テクニカルサポートにお問い合わせください。

Citrix テクニカルサポートに問い合わせる前のチェックリスト

迅速な解決のために、Citrix テクニカルサポートに連絡する前に、以下の項目を準備しておいてください。

- 展開とネットワークトポロジの詳細。
- Citrix ADC シェルプロンプトで次のコマンドを入力して収集されたログファイル。
  ```
  cat /tmp/iked.debug | tee /var/log/iked.log
  ```
- Citrix ADC コマンドラインで次のコマンドを入力して取得した技術サポートバンドル。
  ```
  show techsupport
  ```
- 両方の CloudBridge Connector トンネルエンドポイントでキャプチャされたパケットトレース。パケットトレースを開始するには、Citrix ADC コマンドラインで次のコマンドを入力します。
  ```
  start nstrace -size 0
  ```
  パケットトレースを停止するには、Citrix ADC コマンドラインで次のコマンドを入力します。
  ```
  stop nstrace
  ```
- Citrix ADC コマンドプロンプトで入力した次のコマンドの出力。
  ```
  show arp
  ```

CloudBridge Connector の相互運用性 — ストロングスワン

October 7, 2021

StrongSwan は Linux プラットフォーム用のオープンソースの IPSec 実装です。Citrix ADC アプライアンスと StrongSwan アプライアンスの間に CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。Citrix ADC アプライアンスと StrongSwan アプライアンスは、CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

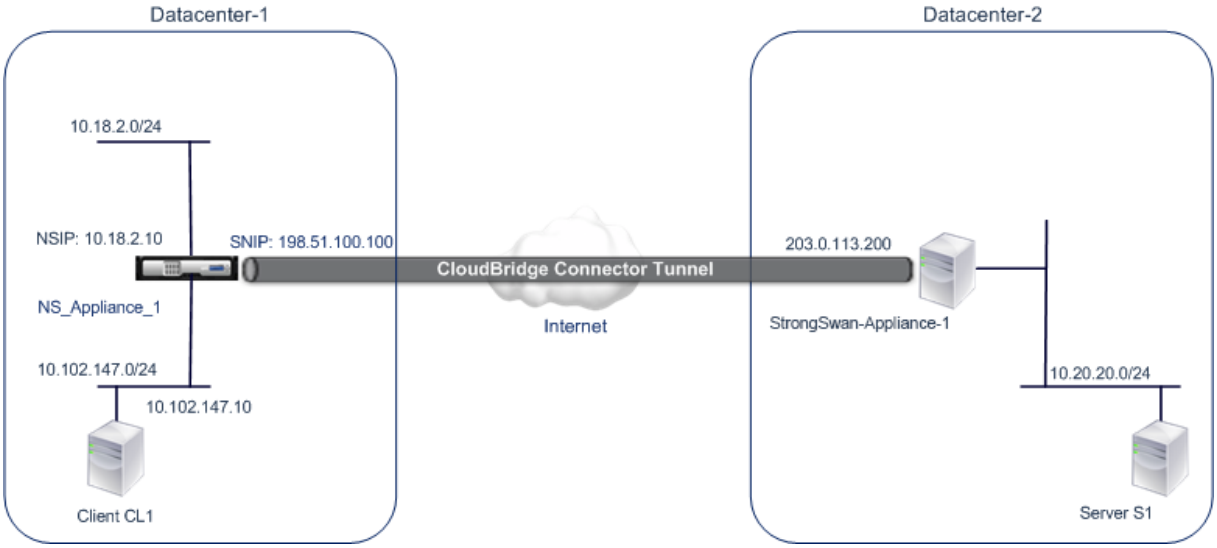
CloudBridge Connector のトンネル設定の例

CloudBridge Connector トンネルのトラフィックフローの図のように、CloudBridge Connector トンネルが次のデバイス間でセットアップされる例について考えてみます。

- データセンター 1 として指定されたデータセンター内の Citrix ADC アプライアンス NS_ アプライアンス-1
- StrongSwan アプライアンス StrongSwan アプライアンス-1 は、データセンター-2 として指定されたデータセンターにあります。

NS_Appliance-1 と StrongSwan-Appliance-1 は、データセンター 1 とデータセンター-2 のプライベートネットワーク間で CloudBridge Connector トンネルを介して通信できるようにします。この例では、NS_Appliance-1 と StrongSwan-Appliance-1 では、データセンター-1 のクライアント CL1 と、データセンター-2 のサーバー S1 が CloudBridge Connector トンネルを介して通信できるようにします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_ アプライアンス 1 では、CloudBridge Connector のトンネル構成には、IPSec プロファイルエンティティ NS_StrongSwan_IPSec_Profile、CloudBridge Connector トンネルエンティティ NS_StrongSwan_Tunnel、およびポリシーベースルーティング (PBR) エンティティ NS_StrongSwan_Pbr が含まれます。



次の表に、この例で使用される設定を示します。

CloudBridge Connector のトンネルのセットアップの主な設定

エンティティ	詳細
データセンター 1 の CloudBridge Connector トンネルエンドポイント（NS_ アプライアンス-1）の IP アドレス	198.51.100.100
データセンター-2 の CloudBridge Connector トンネルエンドポイント (ストロングスワン-アプライアンス 1) の IP アドレス	203.0.113.200
データセンター — CloudBridge Connector トンネルを介してトラフィックを保護する対象のサブネット	10.102.147.0/24
データセンター-CloudBridge Connector トンネルを介してトラフィックを保護する対象のサブネット	10.20.20.0/24

データセンター 1 の Citrix ADC アプライアンス NS_ アプライアンス 1 での設定

SNIP1（参考目的のみ）	198.51.100.100	
IPSec プロファイル	NS_StrongSwan_IPSec_ プロファイル	IKE バージョン:v1、暗号化アルゴリズム:AES、ハッシュアルゴリズム:HMAC_SHA1
psk = サンプル共有キー (注: これは事前共有キーの例です。この文字列は、CloudBridge Connector 構成で使用することはお勧めしません)		
CloudBridge Connector トンネル	NS_StrongSwan_Tunnel	リモート IP = 203.0.113.200、ローカル IP= 198.51.100.100、トンネルプロトコル =IPSEC、IPSec プロファイル =NS_StrongSwan_Profile
ポリシーベースのルート	NS_StrongSwan_Pbr	送信元 IP 範囲 = データセンターのサブネット-1=10.102.147.0-010.102.147.255、送信先 IP 範囲 = データセンターのサブネット-2=10.20.0-010.20.255、IP トンネル =NS_ ストロングスワン _ トンネル

CloudBridge Connector のトンネル設定について考慮すべきポイント

CloudBridge Connector トンネルの設定を開始する前に、次のことを確認してください。

- あなたは、Linux の設定に関する基本的な知識を持っています。
- IPSec プロトコルスイートに関する基本的な知識があります。
- StrongSwan アプライアンスは稼働しており、インターネットに接続されており、トラフィックが CloudBridge Connector トンネルを介して保護されるプライベートサブネットにも接続されています。
- Citrix ADC アプライアンスは稼働しており、インターネットに接続されており、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。
- 以下の IPSec 設定は、Citrix ADC アプライアンスと StrongSwan アプライアンスの間の CloudBridge Connector トンネルでサポートされています。
  - IPSec モード: トンネルモード
  - IKE バージョン: バージョン 1
  - IKE 認証方法：事前共有キー
  - IKE 暗号化アルゴリズム:AES

  - IKE ハッシュアルゴリズム:HMAC SHA1
  - ESP 暗号化アルゴリズム:AES
  - ESP ハッシュアルゴリズム:HMAC SHA1
- Citrix ADC アプライアンスと StrongSwan アプライアンスで、CloudBridge Connector トンネルの両端で同じ IPSec 設定を指定する必要があります。
- Citrix ADC は、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル内）を提供します。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。したがって、StrongSwan アプライアンスでは、IPSec.conf ファイルの IKE パラメータと ESP パラメータで同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- Citrix ADC 端と StrongSwan 端でファイアウォールを構成して、次のことを許可する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 に対する任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネル用のストロングスワンを設定する

Citrix ADC アプライアンスと StrongSwan アプライアンスの間に CloudBridge Connector トンネルを構成するには、StrongSwan アプライアンスで次のタスクを実行します。

- **ipsec.conf** ファイルに **IPsec** 接続情報を指定します。ipsec.conf ファイルは、strongSwan アプライアンスの IPsec 接続に関するすべての制御情報と構成情報を定義します。
- **ipsec.secrets** ファイルに事前共有キーを指定します。ipsec.secrets ファイルは、strongSwan アプライアンスの IPsec 接続用の IKE/IPsec 認証用のシークレットを定義します。

StrongSwan アプライアンスで IPsec VPN（CloudBridge Connector トンネル）を設定する手順は、StrongSwan のリリースサイクルによって時間とともに変化する可能性があります。IPsec VPN トンネルの構成に関する公式の StrongSwan ドキュメントに従うことをお勧めします。

ipsec.conf ファイルのサンプル抜粋に続いて、IPsec VPN トンネルを設定するための IPsec 情報を指定します。詳細については、「CloudBridge Connector 設定の例」トピックを参照してください。詳細については、「CloudBridge Connector の設定 」を参照してください。

ipsec.secrets ファイルのサンプル抜粋に続いて、IPsec VPN トンネルを設定するための IKE 認証事前共有キーを指定します（「CloudBridge Connector 設定の例」トピックを参照）。

`/etc/ipsec.secrets` PSK ‘examplepresharedkey’ #pre-shared key for IPsec IKE authentication

CloudBridge Connector トンネル用の **Citrix ADC** アプライアンスの構成

Citrix ADC アプライアンスと StrongSwan アプライアンスの間に CloudBridge Connector トンネルを構成するには、Citrix ADC アプライアンスで以下のタスクを実行します。Citrix ADC コマンドラインまたは Citrix ADC グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- **IPSec** プロファイルを作成します。IPSec プロファイルエンティティは、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、認証方法などの IPSec プロトコルパラメータを指定します。このパラメータは、CloudBridge Connector トンネル内の IPSec プロトコルで使用されます。
- **IPSec** プロトコルを使用する **IP** トンネルを作成し、**IPSec** プロファイルを関連付けます。IP トンネルは、ローカル IP アドレス（Citrix ADC アプライアンス上で構成された CloudBridge Connector トンネルエンドポイント IP アドレス（SNIP タイプ））、リモート IP アドレス（StrongSwan アプライアンス上で構成された CloudBridge Connector トンネルエンドポイント IP アドレス）、CloudBridge のセットアップに使用するプロトコル（IPSec）を指定します。コネクタトンネル、および IPSec プロファイルエンティティ。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成し、**IP** トンネルに関連付けます。PBR エンティティは、一連のルールと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレスの範囲と宛先 IP アドレスの範囲は、PBR エンティティの条件です。送信元 IP アドレスの範囲を設定して、トンネル経由でトラフィックを保護する Citrix ADC 側のサブネットを指定します。宛先の IP アドレス範囲を設定して、トンネル経由でトラフィックを保護する StrongSwan 側のサブネットを指定します。

Citrix ADC コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1`
- `show ipsec profile <name>`

Citrix ADC コマンドラインを使用して IPSEC トンネルを作成し、そのトンネルに IPSEC プロファイルをバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

Citrix ADC コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

コマンドプロンプトで入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

GUI を使用して IPSEC プロファイルを作成するには

1. **System** > **CloudBridge Connector** > **IPSec Profile** に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**IPSec** プロファイルの追加] ページで、次のパラメータを設定します。
    - Name

- 暗号化アルゴリズム
- ハッシュアルゴリズム
- IKE プロトコルバージョン

4. 2 つの CloudBridge Connector トンネルピアが相互認証に使用する IPSec 認証方法を設定します。事前共有キー認証方法を選択し、事前共有キーが存在パラメータを設定します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2. [**IPv4** トンネル] タブで、[追加] をクリックします。
3. [IP トンネルの追加] ページで、次のパラメータを設定します。
   - Name
   - リモート IP
   - リモートマスク
   - [ローカル IP タイプ] ([ローカル IP タイプ] ドロップダウンリストで、[サブネット *IP*] を選択します)。
   - [ローカル IP] (選択した IP タイプの構成済みの IP アドレスはすべて [ローカル IP] ドロップダウンリストに表示されます。リストから目的の IP を選択します)。
   - プロトコル
   - IPSec プロファイル
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. [PBR の作成（**Create PBR** ）] ページで、次のパラメータを設定します。
   - Name
   - 操作（アクション）
   - ネクストホップタイプ（*IP* トンネルの選択）
   - IP トンネル名
   - 送信元 IP の低
   - 送信元 IP 高
   - 宛先 IP の低
   - 宛先 IP 高
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

Citrix ADC アプライアンスの対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge Connector] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

次のコマンドは、「CloudBridge Connector 構成の例」で、Citrix ADC アプライアンスの NS_Appliance-1 の設定を作成します。


```
    > add ipsec profile NS_StrongSwan_IPSec_Profile -psk
       examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
       HMAC_SHA1


    Done

    > add iptunnel NS_StrongSwan_Tunnel 203.0.113.200 255.255.255.255
       198.51.100.100  - protocol IPSEC  - ipsecProfileName
       NS_StrongSwan_IPSec_Profile


    Done

    > add pbr NS_StrongSwan_Pbr -srcIP 10.102.147.0-10.102.147.255  -
       destIP 10.20.0.0-10.20.255.255  - ipTunnel NS_StrongSwan_Tunnel


    Done

    > apply pbrs


    Done
<!--NeedCopy-->
```

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

CloudBridge Connector の相互運用性 — **F5** ビッグ **IP**

October 7, 2021

Citrix ADC アプライアンスと F5 BIG-IP アプライアンスの間に CloudBridge Connector トンネルを構成して、2 つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。Citrix ADC アプライアンスと F5 BIG-IP アプライアンスは、CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

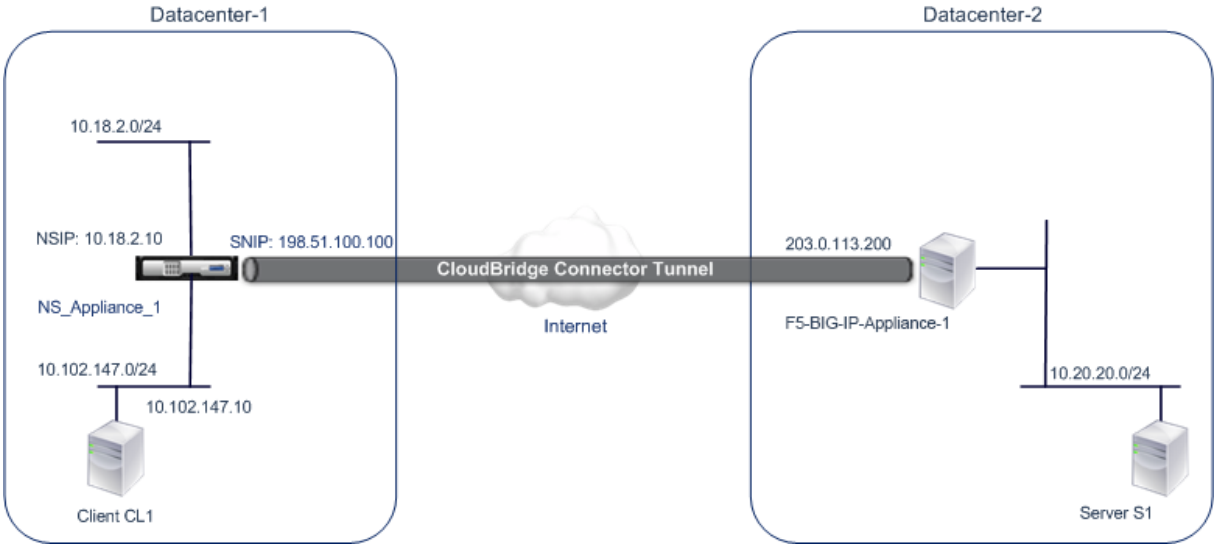
CloudBridge Connector のトンネル設定の例

CloudBridge Connector トンネルのトラフィックフローの図のように、CloudBridge Connector トンネルが次のデバイス間でセットアップされる例について考えてみます。

- データセンター 1 として指定されたデータセンター内の Citrix ADC アプライアンス NS_ アプライアンス-1
- データセンター-2 として指定されたデータセンター内の F5 BIG-IP アプライアンス F5-BIG-IP アプライアンス 1

NS_Appliance-1 および F5-BIG-IP-Appliance-1 は、データセンター 1 とデータセンター 2 のプライベートネットワーク間で CloudBridge Connector トンネルを介して通信できるようにします。この例では、NS_ アプライアンス 1 と F5-BIG-IP-アプライアンス 1 は、データセンター-1 のクライアント CL1 と、データセンター-2 のサーバー S1 との間の通信を CloudBridge Connector トンネルを介して有効にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector のトンネル設定には、IPSec プロファイルエンティティ NS_F5-BIG-IPSec_Profile、CloudBridge Connector トンネルエンティティ NS_F5-BIG-IP_Tunnel、ポリシーベースルーティング（PBR）エンティティ NS_F5-BIG-IP_Pbr が含まれます。



詳細については、F5 big IP pdf を参照してください。

CloudBridge Connector のトンネル設定について考慮すべきポイント

- Citrix ADC アプライアンスは稼働しており、インターネットに接続されており、CloudBridge Connector トンネルを介してトラフィックを保護するプライベートサブネットにも接続されています。
- F5 BIG-IP アプライアンスは稼働しており、インターネットに接続されており、トラフィックが CloudBridge Connector トンネルを介して保護されるプライベートサブネットにも接続されています。
- Citrix ADC アプライアンスと F5 BIG-IP アプライアンス間の CloudBridge Connector トンネルでは、以下の IPSec 設定がサポートされています。
  - IPSec モード: トンネルモード

  - IKE バージョン: バージョン 1
  - IKE 認証方法：事前共有キー
  - IKE 暗号化アルゴリズム:AES
  - IKE ハッシュアルゴリズム:HMAC SHA1
  - ESP 暗号化アルゴリズム:AES
  - ESP ハッシュアルゴリズム:HMAC SHA1
- CloudBridge Connector トンネルの両端で、Citrix ADC アプライアンスと F5 BIG-IP アプライアンスで同じ IPSec 設定を指定する必要があります。
- Citrix ADC は、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル内）を提供します。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。したがって、F5 BIG-IP アプライアンスでは、IKE（フェーズ 1 構成）と ESP（フェーズ 2 構成）で同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- 次のことを許可するには、Citrix ADC 側と F5 BIG-IP 側でファイアウォールを構成する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 に対する任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネルの **F5 BIG-IP** の設定

Citrix ADC アプライアンスと F5 BIG-IP アプライアンス間の CloudBridge Connector トンネルを設定するには、F5 BIG-IP アプライアンスで次のタスクを実行します。

- **IPsec** 用の転送仮想サーバーを作成します。転送仮想サーバは、IPsec トンネルの IP トラフィックを代行受信します。
- **IKE** ピアを作成します。IKE ピアは、ローカルおよびリモートの IPsec トンネルエンドポイントを指定します。また、IPsec IKE フェーズ 1 で使用するアルゴリズムとクレデンシャルも指定します。
- カスタム **IPsec** ポリシーを作成します。ポリシーは、IPsec トンネルの形成に使用する IPSec プロトコル (ESP) とモード (トンネル) を指定します。また、IKE IPsec フェーズ 2 で使用するアルゴリズムとセキュリティパラメータも指定します。
- 双方向 **IPsec** トラフィックセレクタを作成します。トラフィックセレクタは、IP トラフィックが IPSec トンネルを通過する F5 BIG-IP 側と Citrix ADC 側のサブネットを指定します。

F5 BIG-IP アプライアンスで IPsec VPN（CloudBridge Connector トンネル）を設定する手順は、F5 リリースサイクルによっては時間が経つにつれ、変更されることがあります。IPSec VPN トンネルの構成については、F5 BIG-IP の公式ドキュメントに従うことをお勧めします。

https://f5.com

F5 BIG-IP GUI を使用して IPsec 用の転送仮想サーバーを作成するには

1. **[メイン]** タブで、**[ローカルトラフィック]** > **[仮想サーバー]** をクリックし、**[作成]** をクリックします。
2. 「新規仮想サーバー・リスト」画面で、次のパラメータを設定します。

- **Name**：仮想サーバの一意の名前を入力します。
- タイプ。[転送 **(IP)**] を選択します。
- 宛先アドレス。ワイルドカードネットワークアドレスを CIDR 形式で入力します。たとえば、IPv4 の場合は 0.0.0.0/0 と入力して、すべてのトラフィックを受け入れます。
- サービスポート。リストから [すべてのポート] を選択します。
- プロトコルリスト。リストから [すべてのプロトコル] を選択します。
- **VLAN** およびトンネルトラフィック。デフォルトの [すべての **VLAN** とトンネル] のままにします。

3. [完了] をクリックします。

F5 BIG-IP GUI を使用してカスタム IPsec ポリシーを作成するには

1. [メイン] タブで、[ネットワーク] > [**IPsec**] > [**IPsec** ポリシー] をクリックし、[作成] をクリックします。
2. [新しいポリシー] 画面で、次のパラメータを設定します。
   - **Name**：ポリシーの一意の名前を入力します。
   - **IPsec** プロトコル。デフォルトの選択である ESP を保持します。
   - モード。[トンネル] を選択します。画面が更新され、関連する追加設定が表示されます。
   - トンネルローカルアドレス。ローカル IPsec トンネルエンドポイントの IP アドレスを入力します（F5 BIG-IP アプライアンスで構成）。
   - トンネルリモートアドレス。リモートの IPsec トンネルのエンドポイントの IP アドレス（Citrix ADC アプライアンスで構成）を入力します。
3. IKE フェーズ 2 パラメータの場合は、デフォルト値をそのまま使用するか、展開に適したオプションを選択します。
4. [完了] をクリックします。

F5 BIG-IP GUI を使用して双方向 IPsec トラフィックセレクタを作成するには

1. **[** メイン**]** タブで、**[** ネットワーク**]** > **IPsec** > トラフィックセレクターをクリックし、**[** 作成**]** をクリックします。
2. [新しいトラフィックセレクタ**]** 画面で、次のパラメータを設定します。
   - **Name**：トラフィックセレクタの一意の名前を入力します。
   - 注文。既定値 **([**最初**])** をそのまま使用します。この設定では、トラフィックセレクタが [Traffic Selector List] 画面に表示される順序を指定します。
3. 「構成」リストから「詳細」を選択し、次のパラメータを設定します。
   - 送信元 **IP** アドレス。[ホスト**]** または [ネットワーク**]** をクリックし、[アドレス**]** フィールドに、IPsec トンネルを介してトラフィックを保護する F5BIG-IP 側サブネットのアドレスを入力します。
   - 送信元ポート。[***** すべてのポート] を選択します。
   - 宛先 **IP** アドレス。［ホスト］をクリックし、［アドレス］フィールドに、トラフィックが IPSec トンネル上で保護される Citrix ADC 側のサブネットのアドレスを入力します。
   - 宛先ポート。[***** すべてのポート] を選択します。
   - プロトコル。[***** すべてのプロトコル] を選択します。
   - **[**方向**]:** [両方] を選択します。
   - 操作。［保護］を選択します。[**IPsec** ポリシー名**]** の設定が表示されます。

- **IPsec** ポリシー名。作成したカスタム IPsec ポリシーの名前を選択します。

4. [完了] をクリックします。

CloudBridge Connector トンネル用の **Citrix ADC** アプライアンスの構成

Citrix ADC アプライアンスと F5 BIG-IP アプライアンスの間に CloudBridge Connector トンネルを設定するには、Citrix ADC アプライアンスで次のタスクを実行します。Citrix ADC コマンドラインまたは Citrix ADC グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- **IPSec** プロファイルを作成します。IPSec プロファイルエンティティは、IKE バージョン、暗号化アルゴリズム、ハッシュアルゴリズム、認証方法などの IPSec プロトコルパラメータを指定します。このパラメータは、CloudBridge Connector トンネル内の IPSec プロトコルで使用されます。
- **IPSec** プロトコルを使用する **IP** トンネルを作成し、**IPSec** プロファイルを関連付けます。IP トンネルは、ローカル IP アドレス（Citrix ADC アプライアンス上で構成された CloudBridge Connector トンネルエンドポイント IP アドレス（SNIP タイプ））、リモート IP アドレス（F5 ビッグ IP アプライアンス上で構成された CloudBridge Connector トンネルエンドポイント IP アドレス）、CloudBridge のセットアップに使用するプロトコル（IPSec）を指定します。コネクタトンネル、および IPSec プロファイルエンティティ。作成された IP トンネルエンティティは、CloudBridge Connector トンネルエンティティとも呼ばれます。
- **PBR** ルールを作成し、**IP** トンネルに関連付けます。PBR エンティティは、一連のルールと IP トンネル (CloudBridge Connector トンネル) エンティティを指定します。送信元 IP アドレスの範囲と宛先 IP アドレスの範囲は、PBR エンティティの条件です。送信元 IP アドレスの範囲を設定して、トンネル経由でトラフィックを保護する Citrix ADC 側のサブネットを指定し、送信先 IP アドレス範囲を設定して、トンネル経由でトラフィックを保護する F5 BIG-IP 側のサブネットを指定します。

Citrix ADC コマンドラインを使用して IPSEC プロファイルを作成するには

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecyENABLE`
- `show ipsec profile** <name>`

Citrix ADC コマンドラインを使用して IPSEC トンネルを作成し、そのトンネルに IPSEC プロファイルをバインドするには

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

Citrix ADC コマンドラインを使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

コマンドプロンプトで入力します。

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`

- `apply pbrs`
- `show pbr <pbrName>`

GUI を使用して IPSEC プロファイルを作成するには

1. [システム] > [**CloudBridge Connector**] > [**IPSec** プロファイル**]** に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**IPSec** プロファイルの追加**]** ページで、次のパラメータを設定します。
   - Name
   - 暗号化アルゴリズム
   - ハッシュアルゴリズム
   - IKE プロトコルバージョン
4. 2 つの CloudBridge Connector トンネルピアが相互認証に使用する IPSec 認証方法を設定します。事前共有キー認証方法を選択し、事前共有キーが存在パラメータを設定します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して IP トンネルを作成し、IPSEC プロファイルをそのトンネルにバインドするには

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。
2. [**IPv4** トンネル] タブで、[追加] をクリックします。
3. **[IP** トンネルの追加**]** ページで、次のパラメータを設定します。
   - Name
   - リモート IP
   - リモートマスク
   - [ローカル IP タイプ] ([ローカル IP タイプ] ドロップダウンリストで、[サブネット *IP*] を選択します)。
   - ローカル IP（選択した IP タイプの構成済みの IP アドレスはすべて、[ローカル IP] ドロップダウンリストに表示されます。リストから目的の IP を選択します)。
   - プロトコル
   - IPSec プロファイル
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して PBR ルールを作成し、IPSEC トンネルをバインドするには

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. [PBR の作成（**Create PBR** ）] ページで、次のパラメータを設定します。
   - Name
   - 操作（アクション）
   - ネクストホップタイプ（*IP* トンネルの選択）
   - IP トンネル名
   - 送信元 IP の低
   - 送信元 IP 高
   - 宛先 IP の低
   - 宛先 IP 高

4. [**Create**] をクリックしてから、[**Close**] をクリックします。

Citrix ADC アプライアンスの対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge Connector] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

次のコマンドは、「CloudBridge Connector 構成の例」で、Citrix ADC アプライアンスの NS_Appliance-1 の設定を作成します。:

```
1      > add ipsec profile NS_F5-BIG-IP_IPSec_Profile -psk
           examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
           HMAC_SHA1 -lifetime 315360 -perfectForwardSecrecy ENABLE
2
3
4      Done
5
6      > add iptunnel NS_F5-BIG-IP_Tunnel 203.0.113.200 255.255.255.255
           198.51.100.100 -protocol IPSEC -ipsecProfileName NS_F5-BIG-
           IP_IPSec_Profile
7
8
9      Done
10
11     > add pbr NS_F5-BIG-IP_Pbr -srcIP 10.102.147.0-10.102.147.255 -
           destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_F5-BIG-IP_Tunnel
12
13
14     Done
15
16     > apply pbrs
17
18
19     Done
20 <!--NeedCopy-->
```

CloudBridge Connector トンネルの監視

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

CloudBridge Connector の相互運用性 — Cisco ASA

October 7, 2021

Citrix ADC アプライアンスと Cisco ASA アプライアンスの間に CloudBridge Connector トンネルを構成して、2つのデータセンターを接続したり、ネットワークをクラウドプロバイダーに拡張したりできます。Citrix ADC アプライアンスと Cisco ASA アプライアンスは、CloudBridge Connector トンネルのエンドポイントを形成し、ピアと呼ばれます。

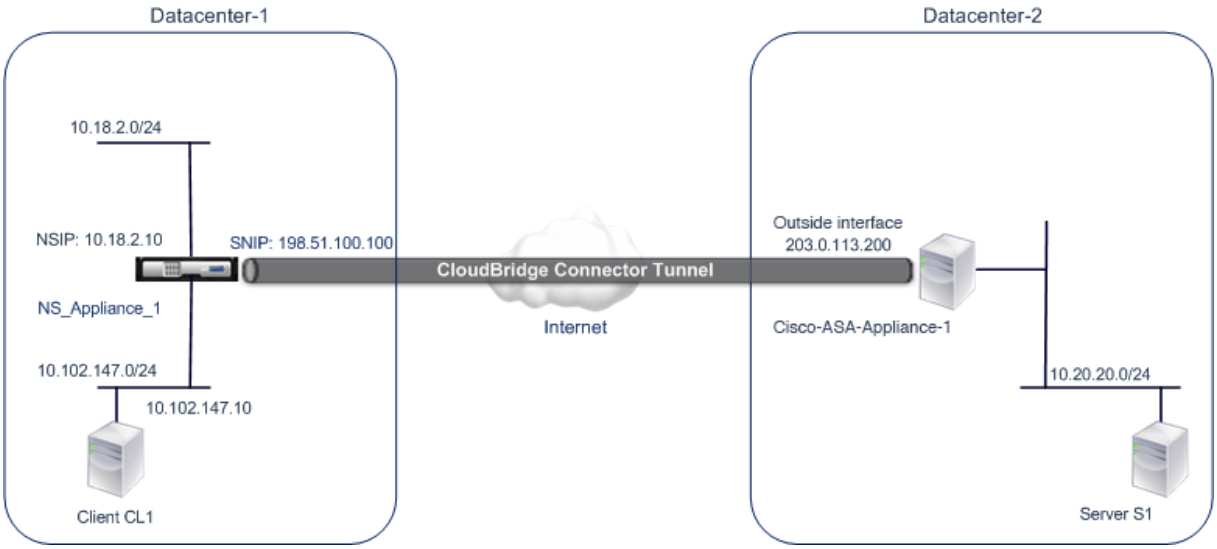
CloudBridge Connector のトンネル設定の例

CloudBridge Connector トンネル内のトラフィックフローの図のように、CloudBridge Connector トンネルが次のアプライアンス間でセットアップされる例について考えてみます。

- データセンター 1 として指定されたデータセンター内の Citrix ADC アプライアンス NS_ アプライアンス-1
- データセンター-2 として指定されたデータセンター内の Cisco ASA アプライアンス Cisco ASA アプライアンス 1

NS_ アプライアンス 1 および Cisco ASA-アプライアンス 1 は、CloudBridge Connector トンネルを介したデータセンター 1 とデータセンター 2 のプライベートネットワーク間の通信を可能にします。この例では、NS_ アプライアンス 1 と Cisco ASA-アプライアンス 1 は、データセンター-1 のクライアント CL1 とデータセンター-2 のサーバ S1 との間の通信を CloudBridge Connector トンネル経由で有効にします。クライアント CL1 とサーバー S1 は、異なるプライベートネットワーク上にあります。

NS_Appliance-1 では、CloudBridge Connector トンネル設定には、IPSec プロファイルエンティティ NS_Cisco-ASA_IPSec_Profile、CloudBridge Connector トンネルエンティティ NS_Cisco-ASA_Tunnel、およびポリシーベースルーティング（PBR）エンティティ NS_Cisco-ASA_Pbr が含まれます。



CloudBridge Connector のトンネル設定について考慮すべきポイント

CloudBridge Connector トンネルの設定を開始する前に、次のことを確認してください。

- Citrix ADC アプライアンスと Cisco ASA アプライアンス間の CloudBridge Connector トンネルでは、次の IPSec 設定がサポートされています。

IPSec のプロパティ	設定
IPSec モード	トンネルモード
IKE のバージョン	バージョン 1
IKE 認証方式	事前共有キー
IKE 暗号化アルゴリズム	AES, 3DES
IKE ハッシュアルゴリズム	HMAC SHA1, HMAC MD5
ESP 暗号化アルゴリズム	AES, 3DES
ESP ハッシュアルゴリズム	HMAC SHA1, HMAC MD5

- CloudBridge Connector トンネルの両端で、Citrix ADC アプライアンスと Cisco ASA アプライアンスで同じ IPSec 設定を指定する必要があります。
- Citrix ADC は、IKE ハッシュアルゴリズムと ESP ハッシュアルゴリズムを指定するための共通パラメータ（IPSec プロファイル内）を提供します。また、IKE 暗号化アルゴリズムと ESP 暗号化アルゴリズムを指定するためのもう 1 つの共通パラメータも用意されています。したがって、Cisco ASA アプライアンスでは、IKE（フェーズ 1 構成）および ESP（フェーズ 2 構成）で同じハッシュアルゴリズムと同じ暗号化アルゴリズムを指定する必要があります。
- 次のことを許可するには、Citrix ADC 側と Cisco ASA 側でファイアウォールを設定する必要があります。
  - ポート 500 の任意の UDP パケット
  - ポート 4500 に対する任意の UDP パケット
  - 任意の ESP（IP プロトコル番号 50）パケット

CloudBridge Connector トンネル用の **Cisco ASA** の設定

Cisco ASA アプライアンスで CloudBridge Connector トンネルを設定するには、Cisco ASA コマンドラインインターフェイスを使用します。これは、Cisco ASA アプライアンスの設定、モニタリング、および保守のための主要なユーザインターフェイスです。

Cisco ASA アプライアンスで CloudBridge Connector トンネル設定を開始する前に、次の点を確認してください。

- Cisco ASA アプライアンスで管理者クレデンシャルを持つユーザアカウントがあること。
- Cisco ASA コマンドラインインターフェイスに精通していること。
- Cisco ASA アプライアンスは稼動しており、インターネットに接続されており、トラフィックが CloudBridge Connector トンネルを介して保護されるプライベートサブネットにも接続されています。

> 注
>
> Cisco ASA アプライアンスで CloudBridge Connector トンネルを設定する手順は、Cisco のリリースサイクルによっては、時間の経過とともに変化する場合があります。IPSec VPN トンネルの構成に関する公式の Cisco ASA 製品マニュアルに従うことをお勧めします。

- http://www.cisco.com

Citrix ADC アプライアンスと Cisco ASA アプライアンスの間に CloudBridge Connector トンネルを設定するには、Cisco ASA アプライアンスのコマンドラインで次のタスクを実行します。

- **IKE** ポリシーを作成します。IKE ポリシーは、IKE ネゴシエーション（フェーズ 1）で使用されるセキュリティパラメータの組み合わせを定義します。たとえば、IKE ネゴシエーションで使用されるハッシュアルゴリズム、暗号化アルゴリズム、認証方式などのパラメータは、このタスクで設定されます。
- 外部インターフェイスで **IKE** を有効にします。トンネルトラフィックがトンネルピアに流れる外部インターフェイスで IKE を有効にします。
- トンネルグループを作成します。トンネルグループは、トンネルのタイプと事前共有キーを指定します。トンネルタイプは ipsec-l2l に設定する必要があります。これは、IPsec LAN から LAN への対応を表します。事前共有キーは、CloudBridge Connector トンネルのピアが相互に認証するために使用するテキスト文字列です。事前共有キーは、IKE 認証のために相互に照合されます。したがって、認証を成功させるには、Cisco ASA アプライアンスと Citrix ADC アプライアンスで同じ事前共有キーを設定する必要があります。
- トランスフォームセットを定義します。トランスフォームセットは、IKE ネゴシエーションが成功した後、CloudBridge Connector トンネルを介したデータの交換に使用されるセキュリティパラメータ（フェーズ 2）の組み合わせを定義します。
- アクセスリストを作成します。暗号化アクセスリストは、CloudBridge トンネルを介して IP トラフィックを保護するサブネットを定義するために使用されます。アクセスリストの送信元パラメータと宛先パラメータは、CloudBridge Connector トンネル上で保護される Cisco アプライアンス側および Citrix ADC 側のサブネットを指定します。アクセスリストは permit に設定する必要があります。Cisco アプライアンス側のサブネット内のアプライアンスから発信され、Citrix ADC 側のサブネット内のアプライアンスを宛先とし、アクセスリストの送信元パラメータと宛先パラメータに一致する要求パケットは、CloudBridge Connector トンネルを介して送信されます。
- クリプトマップを作成します。クリプトマップは、セキュリティアソシエーション（SA）の IPSec パラメータを定義します。これには、CloudBridge トンネルでトラフィックを保護するサブネットを識別する暗号化アクセスリスト、IP アドレスによるピア（Citrix ADC）識別、ピアセキュリティ設定に一致するようにトランスフォームセットが含まれます。
- 外部インターフェイスにクリプトマップを適用します。この作業では、トンネルトラフィックがトンネルピアに流れる外部インターフェイスにクリプトマップを適用します。インターフェイスにクリプトマップを適用すると、Cisco ASA アプライアンスは、すべてのインターフェイストラフィックをクリプトマップセットに対して評価し、接続またはセキュリティアソシエーションのネゴシエーション中に指定したポリシーを使用するように指示します。

次の手順の例では、CloudBridge Connector の設定とデータフローの例で使用する Cisco ASA アプライアンス Cisco-ASA-アプライアンス 1 の設定を作成します。

Cisco ASA コマンドラインを使用して IKE ポリシーを作成するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
crypto ikev1 policy priority	Cisco-ASA-appliance-1(config)# crypto ikev1 policy 1	IKE ポリシー構成モードを開始し、作成するポリシーを指定します。(各ポリシーは、割り当てたプライオリティ番号によって一意に識別されます)。この例では、ポリシー 1 を設定します。
encryption (3des | aes)	Cisco-ASA-appliance-1 (config-ikev1-policy)# encryption 3des	暗号化アルゴリズムを指定します。この例では、3DES アルゴリズムを設定します。
hash (sha | md5)	Cisco-ASA-appliance-1 (config- ikev1-policy)# hash sha	ハッシュアルゴリズムを指定します。この例では、SHA を設定します。
authenticationpre-share	Cisco-ASA-appliance-1 (config- ikev1-policy)# authentication pre-share	事前共有認証方法を指定します。
group 2	Cisco-ASA-appliance-1 (config- ikev1-policy)# group 2	1024 ビットの Diffie-Hellman グループ識別子 (2) を指定します。
lifetime seconds	Cisco-ASA-appliance-1 (config- ikev1-policy)# lifetime 28800	セキュリティアソシエーションの有効期間を秒単位で指定します。この例では、Citrix ADC アプライアンスの有効期間のデフォルト値である 28800 秒を設定します。

Cisco ASA コマンドラインを使用して外部インターフェイスで IKE を有効にするには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
crypto ikev1 enable outside	Cisco-ASA-appliance-1(config)# crypto ikev1 enable outside	トンネルトラフィックがトンネルピアに流れるインターフェイスで IKEv1 を有効にします。この例では、outside という名前のインターフェイスで IKEv1 を有効にします。

To create a tunnel group by using the Cisco ASA command line

Cisco ASA アプライアンスのコマンドプロンプトで、Cisco ASA コマンドラインを使用して、接続された pdf トンネルグループの show のように、グローバル構成モードで次のコマンドを入力します。

Cisco ASA コマンドラインを使用してクリプトアクセスリストを作成するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを次の順序で入力します。

コマンド	例	コマンドの説明
access-list access-list-number permit IP source source-wildcard destination destination-wildcard	Cisco-ASA-appliance-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	CloudBridge Connector トンネルを介して IP トラフィックを保護するサブネットを決定する条件を指定します。この例では、サブネット 10.20.0/24（Cisco ASA アプライアンス 1 側）および 10.102.147.0/24（NS_ アプライアンス 1 側）からのトラフィックを保護するように、アクセスリスト 111 を設定します。

Cisco ASA コマンドラインを使用してトランスフォームセットを定義するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを入力します。ASA コマンドラインテーブルを使用したトランスフォームセット pdf を参照してください。

Cisco ASA コマンドラインを使用してクリプトマップを作成するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
crypto map map-name seq-num match address access-list-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 match address 111	クリプトマップを作成し、アクセスリストを指定します。次に、シーケンス番号 1 のクリプトマップ NS-CISCO-CM を設定し、アクセスリスト 111 を NS-CISCO-CM に割り当てる例を示します。
crypto map map-name seq-num set peer ip-address	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set peer 198.51.100.100	ピア（Citrix ADC アプライアンス）を IP アドレスで指定します。この例では、198.51.100.100 を指定します。これは、Citrix ADC アプライアンス上のトンネルエンドポイントの IP アドレスです。
crypto map map-name seq-num set ikev1 transform-set transform-set-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set ikev1 transform-set NS-CISCO-TS	このクリプトマップエントリに許可されるトランスフォームセットを指定します。次に、トランスフォームセット NS-CISCO-TS を指定する例を示します。

Cisco ASA コマンドラインを使用してインターフェイスにクリプトマップを適用するには

Cisco ASA アプライアンスのコマンドプロンプトで、グローバル構成モードで次のコマンドを、表示されている順序で入力します。

コマンド	例	コマンドの説明
crypto map map-nameinterface interface-name	Cisco-ASA-appliance-1(config)# crypto map NS-CISCO-CM interface outside	CloudBridge Connector トンネルのトラフィックが流れるインターフェイスにクリプトマップを適用します。次に、クリプトマップ NS-CISCO-CM を外部インターフェイスに適用する例を示します。

CloudBridge Connector トンネル用の **Citrix ADC** アプライアンスの構成

Citrix ADC アプライアンスと Cisco ASA アプライアンスの間に CloudBridge Connector トンネルを設定するには、Citrix ADC アプライアンスで次のタスクを実行します。Citrix ADC コマンドラインまたは Citrix ADC グラフィカルユーザーインターフェイス（GUI）のいずれかを使用できます。

- IPSec プロファイルを作成します。
- IPSec プロトコルを使用する IP トンネルを作成し、IPSec プロファイルを関連付けます。
- PBR ルールを作成し、IP トンネルに関連付けます。

Citrix ADC コマンドラインを使用して **IPSEC** プロファイルを作成するには:

コマンドプロンプトで入力します。

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

Citrix ADC コマンドラインを使用して **IPSEC** トンネルを作成し、そのトンネルに **IPSEC** プロファイルをバインドするには、次の手順を実行します:

コマンドプロンプトで入力します。

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

Citrix ADC コマンドラインを使用して **PBR** ルールを作成し、**IPSEC** トンネルをバインドするには:

コマンドプロンプトで入力します。

- `**add pbr** <pbrName> **ALLOW** -**srcIP** <subnet-range> -**destIP** <subnet-range>`
- `**ipTunnel** <tunnelName>`
- `**apply pbrs**`
- `**show pbr** <pbrName>`

GUI を使用して **IPSEC** プロファイルを作成するには:

1. [システム] > [**CloudBridge Connector**] > [**IPsec** プロファイル**]** に移動します。
2. 詳細ペインで、[**Add**] をクリックします。
3. [**IPSec** プロファイルの追加**]** ページで、次のパラメータを設定します。
   - Name
   - 暗号化アルゴリズム
   - ハッシュアルゴリズム
   - IKE プロトコルバージョン
   - Perfect Forward Secrecy (このパラメータを有効にする)
4. 相互に認証するために 2 つの CloudBridge Connector トンネルピアで使用される IPsec 認証方法を設定します。事前共有キー認証方法を選択し、[事前共有キーが存在する] パラメータを設定します。
5. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して **IP** トンネルを作成し、**IPSEC** プロファイルをそれにバインドするには:

1. [システム] > [**CloudBridge Connector**] > [**IP** トンネル] に移動します。

2. [**IPv4** トンネル] タブで、[追加] をクリックします。
3. **[IP** トンネルの追加**]** ページで、次のパラメータを設定します。
   - Name
   - リモート IP
   - リモートマスク
   - [ローカル IP タイプ] ([ローカル IP タイプ] ドロップダウンリストで、[サブネット IP] を選択します)。
   - ローカル IP（選択した IP タイプの構成済みの IP アドレスはすべて、[ローカル IP] ドロップダウンリストに表示されます。リストから目的の IP を選択します)。
   - プロトコル
   - IPSec プロファイル
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

GUI を使用して **PBR** ルールを作成し、**IPSEC** トンネルをバインドするには:

1. [システム] > [ネットワーク] > [**PBR**] に移動します。
2. [**PBR**] タブで、[追加] をクリックします。
3. [**PBR** の 作成**]** ページで、次のパラメーターを設定します。
   - Name
   - 操作（アクション）
   - ネクストホップタイプ（IP トンネルの選択）
   - IP トンネル名
   - 送信元 IP の低
   - 送信元 IP 高
   - 宛先 IP の低
   - 宛先 IP 高
4. [**Create**] をクリックしてから、[**Close**] をクリックします。

Citrix ADC アプライアンスの対応する新しい CloudBridge Connector トンネル構成が GUI に表示されます。CloudBridge Connector トンネルの現在のステータスは、[設定済み CloudBridge Connector] ペインに表示されます。緑色のドットは、トンネルがアップしていることを示します。赤い点は、トンネルがダウンしていることを示します。

次のコマンドは、「CloudBridge Connector 構成の例」で、Citrix ADC アプライアンスの NS_Appliance-1 の設定を作成します:

```
1  >  add ipsec profile NS_Cisco-ASA_IPSec_Profile -psk
       examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
       HMAC_SHA1  - lifetime 315360 -perfectForwardSecrecy ENABLE
2
3  Done
4
5  >  add iptunnel NS_Cisco-ASA_Tunnel 203.0.113.200 255.255.255.255
```



```
    198.51.100.100  -protocol IPSEC  -ipsecProfileName NS_Cisco-
    ASA_IPSec_Profile
6
7
8 Done
9
10 > add pbr NS_Cisco-ASA_Pbr -srcIP 10.102.147.0-10.102.147.255  -destIP
    10.20.0.0-10.20.255.255  -ipTunnel NS_Cisco-ASA_Tunnel
11
12
13 Done
14
15 > apply pbrs
16
17 Done
18
19 <!--NeedCopy-->
```

CloudBridge Connector トンネルのモニタリング

CloudBridge Connector のトンネル統計カウンタを使用して、Citrix ADC アプライアンス上の CloudBridge Connector トンネルのパフォーマンスを監視できます。Citrix ADC アプライアンスでの CloudBridge Connector トンネル統計の表示の詳細については、「CloudBridge Connector トンネルの監視」を参照してください。

高可用性

October 7, 2021

2 台の Citrix ADC アプライアンスを高可用性（HA）で展開すると、どのトランザクションでも中断のない操作を実現できます。一方のアプライアンスをプライマリノードとして構成し、もう一方のアプライアンスをセカンダリノードとして設定すると、プライマリノードは接続を受け入れ、サーバを管理し、セカンダリノードはプライマリノードを監視します。何らかの理由でプライマリノードが接続を受け付けることができなくなると、セカンダリノードが処理を引き継ぎます。

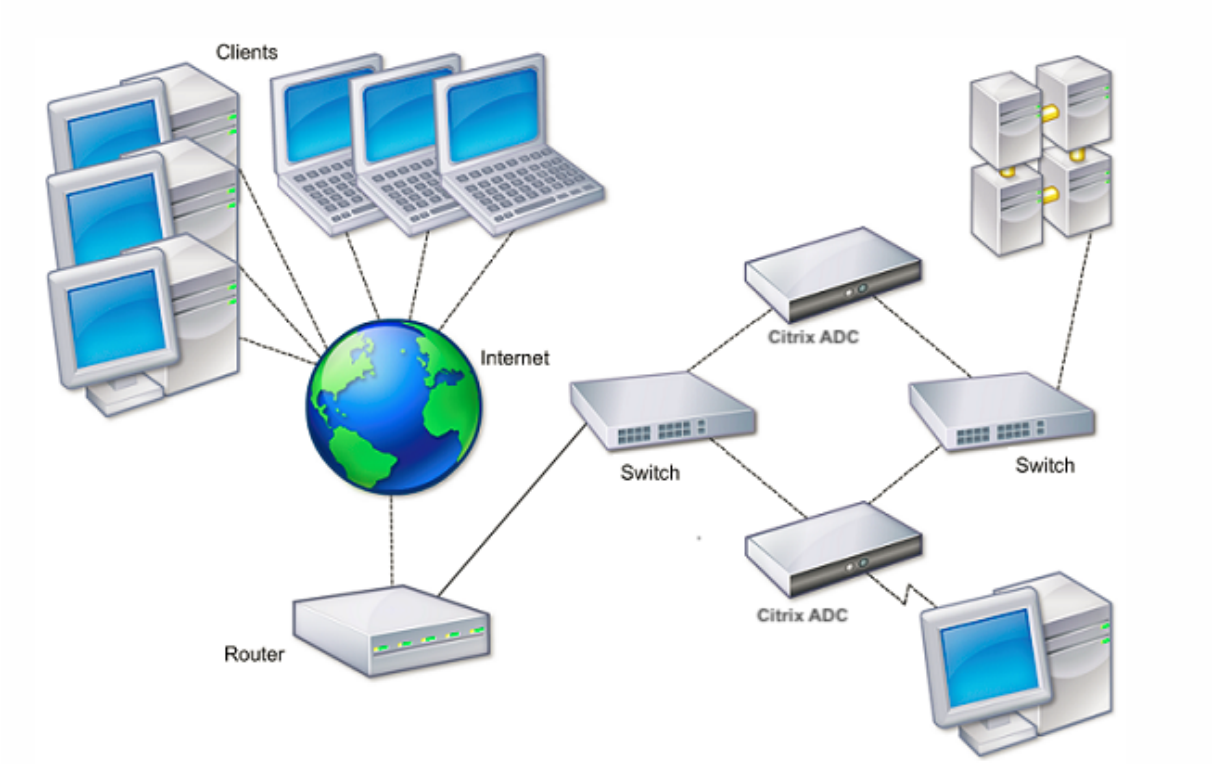
セカンダリノードは、定期的なメッセージ（ハートビートメッセージまたはヘルスチェックとも呼ばれる）を送信してプライマリを監視し、プライマリノードが接続を受け付けているかどうかを判断します。ヘルスチェックが失敗した場合、セカンダリノードは指定された期間接続を再試行します。その後、プライマリノードが正常に機能していないと判断されます。次に、セカンダリ・ノードがプライマリ・ノードを引き継ぎます（フェイルオーバーと呼ばれるプロセス）。

フェイルオーバー後、すべてのクライアントが管理対象サーバーへの接続を再確立する必要がありますが、セッション永続性ルールはフェイルオーバー前と同じように維持されます。

Web サーバーのロギングの永続性を有効にすると、フェールオーバーによってログデータが失われることはありません。ロギングの永続性を有効にするには、ログサーバー設定が log.conf ファイルに両方のシステムのエントリを保持する必要があります。

次の図は、HA ペアを使用したネットワーク構成を示しています。

図 1: 高可用性構成の Citrix ADC アプライアンス



HA を設定するには、まず、両方のノードが同じサブネットにある基本的なセットアップを作成します。次に、ノードがヘルスチェック情報を通信する間隔、ノードが同期を維持するプロセス、プライマリからセカンダリへのコマンドの伝播をカスタマイズできます。フェールセーフモードを設定して、どちらのノードもプライマリでない状況を防ぐことができます。Citrix ADC 無償 ARP メッセージを受け付けないデバイスが環境に含まれている場合は、仮想 MAC アドレスを構成する必要があります。より複雑な構成の準備ができたら、異なるサブネットに HA ノードを設定できます。

HA セットアップの信頼性を向上させるために、ルートモニタを設定し、冗長リンクを作成できます。トラブルシューティングやメンテナンスタスクの実行など、状況によっては、ノードを強制的にフェイルオーバーする（プライマリステータスを他のノードに割り当てる）場合や、セカンダリノードを強制的にセカンダリにしたり、プライマリノードをプライマリにしたりしたい場合があります。

高可用性のセットアップで考慮すべきポイント

October 7, 2021

> 注
>
> HA セットアップでシステムを設定するための次の要件。

- 高可用性構成では、プライマリとセカンダリの Citrix ADC アプライアンスが同じモデルである必要があります。異なる Citrix ADC モデルは HA ペアではサポートされていません。
- HA セットアップでは、両方のノードで同じバージョンの Citrix ADC を実行する必要があります。
- プライマリシステムとセカンダリシステムの両方で構成ファイル（ns.conf）のエントリが一致する必要があります。ただし、次の例外があります。
  - プライマリシステムとセカンダリシステムは、それぞれ固有の IP アドレス（NSIP）を使用して構成する必要があります。
  - HA ペアでは、1 つのノードのノード ID と関連する IP アドレスが他のノードを指している必要があります。たとえば、ノード NS1 と NS2 がある場合、一意のノード ID と IP アドレス NS2 を使用して NS1 を構成し、一意のノード ID と IP アドレス NS1 を使用して NS2 を構成する必要があります。
- GUI または CLI を直接経由しない方法（たとえば、SSL 証明書のインポート、スタートアップスクリプトへの変更）を使用して、いずれかのノードに構成ファイルを作成する場合は、構成ファイルを他のノードにコピーするか、そのノードに同じファイルを作成する必要があります。
- 最初は、すべての Citrix ADC アプライアンスに同じ RPC ノードパスワードが設定されています。RPC ノードは、構成およびセッション情報のシステム間通信に使用される内部システムエンティティです。セキュリティのため、既定の RPC ノードのパスワードを変更する必要があります。

  各 Citrix ADC には 1 つの RPC ノードが存在します。このノードはパスワードを格納します。このパスワードは、連絡先のシステムによって提供されたパスワードと照合されます。他のシステムと通信するには、Citrix ADC ごとにシステムに関する知識が必要です。これには、これらのシステムでの認証方法も含まれます。RPC ノードは、この情報を保持します。この情報には、他のシステムの IP アドレスや認証に必要なパスワードが含まれます。

  RPC ノードは、ノードの追加またはグローバルサーバー負荷分散 (GSLB) サイトの追加時に暗黙的に作成されます。RPC ノードを手動で作成または削除することはできません。

  > 注:
  >
  > 高可用性セットアップの Citrix ADC アプライアンスがワンアームモードで構成されている場合は、スイッチまたはハブに接続されているシステムインターフェイスを除くすべてのシステムインターフェイスを無効にする必要があります。

  IPv6 HA 設定では、次の考慮事項が適用されます。

  - IPv6PT ライセンスを両方の Citrix ADC アプライアンスにインストールする必要があります。
  - IPv6PT ライセンスをインストールした後、GUI またはコマンドラインインターフェイスを使用して IPv6 機能を有効にします。
  - どちらの Citrix ADC アプライアンスも、グローバルな NSIP IPv6 アドレスが必要です。さらに、2 つのノード間のネットワークエンティティ (スイッチやルーターなど) が IPv6 をサポートしている必要が

あります。

高可用性の設定

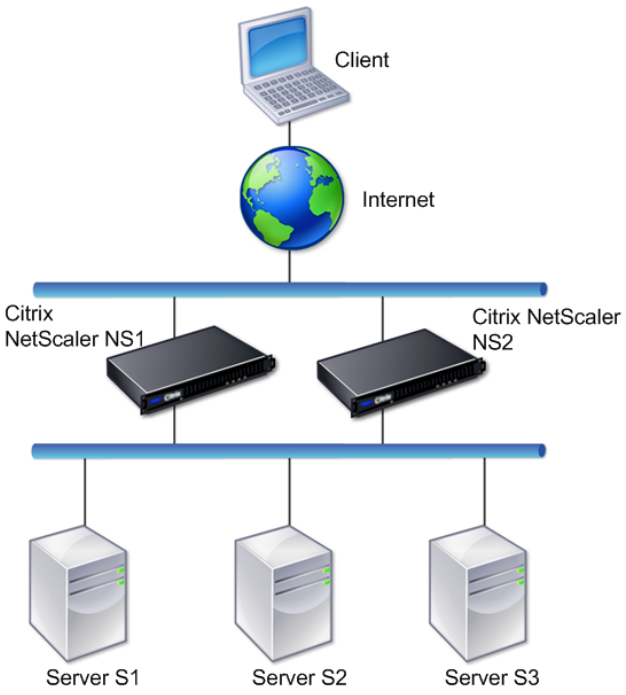
December 7, 2021

高可用性構成を設定するには、2 つのノードを作成します。各ノードは、もう一方の Citrix ADC IP（NSIP）アドレスをリモートノードとして定義します。まず、高可用性を構成する 2 つの Citrix ADC アプライアンスのいずれかにログオンし、ノードを追加します。他のアプライアンスの Citrix ADC IP（NSIP）アドレスを新しいノードのアドレスとして指定します。次に、もう一方のアプライアンスにログオンし、最初のアプライアンスの NSIP アドレスを持つノードを追加します。アルゴリズムは、どのノードがプライマリになり、どのノードがセカンダリになるかを決定します。

> 注:
>
> Citrix ADC GUI には、2 番目のアプライアンスにログオンする必要がないようにするオプションが用意されています。

次の図は、両方のノードが同じサブネット内にある簡単な HA セットアップを示しています。

図 1: 高可用性構成で接続された 2 つの Citrix ADC アプライアンス



リモートノードの追加

高可用性セットアップのノードとしてリモート Citrix ADC アプライアンスを追加するには、一意のノード ID とアプライアンスの NSIP アドレスを指定します。HA ノードを追加するときは、接続されていない、またはトラフィックに使用されていないインターフェイスごとに HA モニタを無効にする必要があります。CLI ユーザの場合、これは別の手順です。

> 注:
>
> 高可用性設定の各ノードが同じ設定になるようにするには、SSL 証明書、スタートアップスクリプト、およびその他の設定ファイルをプライマリノード上のものと同期させる必要があります。

コマンドラインインターフェイスを使用してノードを追加するには

コマンドプロンプトで入力します。

- `add ha node <id> <IPAddress>`
- `show ha node`

例

```
> add ha node 10 203.0.113.32
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **HA** モニタを無効にするには

コマンドプロンプトで入力します。

- `set interface <ifNum> [-haMonitor ( ON | OFF )]`
- `show interface <ifNum>`

例

```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

GUI を使用してリモートノードを追加するには

[システム] > [高可用性] に移動し、[ノード] タブで新しいリモートノードを追加するか、既存のノードを編集します。

ノードの無効化または有効化

セカンダリノードのみを無効または有効にできます。セカンダリノードを無効にすると、プライマリノードへのハートビートメッセージの送信が停止するため、プライマリノードはセカンダリのステータスを確認できなくなります。ノードを有効にすると、そのノードは高可用性構成に参加します。

コマンドラインインターフェイスを使用してノードを無効または有効にするには

コマンドプロンプトで、次のコマンドのいずれかを入力します。

- `set ha node -hastatus DISABLED`
- `set ha node -hastatus ENABLED`

GUI を使用してノードを無効または有効にするには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. [高可用性ステータス] リストで、**[**有効**] (HA** にアクティブに参加する**)** または [無効**] (HA** に参加しない**)** を選択します。

ノードを削除する

ノードを削除すると、そのノードは高可用性構成ではなくなります。

コマンドラインインターフェイスを使用してノードを削除するには

コマンドプロンプトで入力します。

```
rm ha node <id>
```

例

```
1 > rm ha node 10
2   Done
3 <!--NeedCopy-->
```

GUI を使用してノードを削除するには

[システム] > [高可用性] に移動し、[ノード] タブでノードを削除します。

通信間隔の構成

October 7, 2021

hello 間隔は、ハートビートメッセージがピアノードに送信される間隔です。デッドインターバルは、ハートビートパケットが受信されなかった場合に、ピアノードが DOWN とマークされるまでの時間間隔です。ハートビートメッセージは、HA ペアの他のノードのポート 3003 に送信される UDP パケットです。デッドインターバルは、hello インターバルの倍数として設定する必要があります。

コマンドラインインターフェイスを使用して **hello** 間隔とデッドインターバルを設定するには

コマンドプロンプトで入力します。

- `set HA node [-helloInterval <msecs>] [-deadInterval <secs>]`
- `show HA node <id>`

GUI を使用して **hello** 間隔とデッドインターバルを設定するには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. 次のパラメーターを設定します。
    - ハロー間隔 (ミリ秒)
    - デッド間隔 (秒)

同期の設定

October 7, 2021

同期は、セカンダリノード上のプライマリノードの構成を複製するプロセスです。同期の目的は、発生したフェイルオーバーの数に関係なく、プライマリノードとセカンダリノード間の構成情報が失われないようにすることです。同期ではポート 3010 が使用されます。

同期は、次のいずれかの状況によってトリガーされます。

- HA セットアップのセカンダリノードは、再起動後に起動します。
- プライマリノードは、フェールオーバー後にセカンダリになります。

自動同期はデフォルトで有効になっています。同期を強制することもできます。

同期の無効化または有効化

自動 HA 同期は、HA ペアの各ノードでデフォルトで有効になっています。どちらのノードでも有効または無効にできます。

コマンドラインインターフェイスを使用して自動同期を無効または有効にするには

コマンドプロンプトで入力します。

- `set HA node -haSync DISABLED`
- `set HA node -haSync ENABLED`

GUI を使用して同期を無効または有効にするには

1. [**System**] > [**High Availability**] に移動します。
2. [HA 同期] で、[セカンダリノードがプライマリから構成をフェッチする] オプションをオフまたは選択します。

セカンダリノードとプライマリノードとの同期を強制する

Citrix ADC は、自動同期に加えて、強制同期もサポートしています。プライマリノードまたはセカンダリノードのいずれかから同期を強制できます。セカンダリノードからの同期を強制すると、プライマリノードとの構成の同期が開始されます。

ただし、同期がすでに進行中の場合、強制同期は失敗し、警告が表示されます。強制同期は、次のいずれかの状況でも失敗します。

- スタンドアロンシステム上で同期を強制します。
- セカンダリノードは無効です。
- HA 同期は、セカンダリノードで無効になっています。

コマンドラインインターフェイスを使用して同期を強制するには

コマンドプロンプトで入力します。

```
force HA sync
```

GUI を使用して同期を強制するには

1. [**System**] > [**High Availability**] に移動します。
2. [ノード] タブの [操作] ボックスの一覧で、[同期の強制] をクリックします。

高可用性セットアップでの構成ファイルの同期

October 7, 2021

高可用性セットアップでは、1 分間隔でプライマリノードのすべての構成ファイルがセカンダリノードに自動で同期されます。設定ファイルの同期は、コマンド・ライン・インタフェースまたはプライマリ・ノードまたはセカンダリ・ノードの GUI を使用して手動で実行できます。

セカンダリに固有のファイル（プライマリに存在しない）は、同期中に削除されません。

コマンドラインインターフェイスを使用して高可用性セットアップのファイルを同期するには

コマンドプロンプトで入力します。

```
sync HA files <mode>
```

例

```
1 > sync HA files all
2  Done
3 <!--NeedCopy-->
```

```
1 > sync HA files ssl
2  Done
3 <!--NeedCopy-->
```

パラメータの説明（**CLI** プロシージャにリストされているコマンド）

```
sync ha files <mode>
```

```
mode
```

次のいずれかの同期モードを指定します。

- **all** -システム構成、Access Gateway ブックマーク、SSL 証明書、SSL CRL リスト、HTML インジェクションスクリプト、およびアプリケーションファイアウォール XML オブジェクトに関連するファイルを同期します。
- **bookmarks** -Access Gateway のすべてのブックマークを同期します。
- **ssl** -SSL 機能のすべての証明書、キー、および CRL を同期します。
- **htmlinjection** -HTML インジェクション機能に設定されているすべてのスクリプトを同期します。
- **imports** -アプリケーションファイアウォール用に構成されたすべての XML オブジェクト (WSDLs、スキーマ、エラーページなど) を同期します。
- **misc** -すべてのライセンスファイルと rc.conf ファイルを同期します。
- **all_plus_misc** -システム構成、Access Gateway ブックマーク、SSL 証明書、SSL CRL リスト、HTML インジェクションスクリプト、アプリケーションファイアウォールの XML オブジェクト、ライセンス、および rc.conf ファイルに関連するファイルを同期します。

GUI を使用して高可用性セットアップのファイルを同期するには

[システム] > [診断] に移動し、[ユーティリティ] グループで [**HA** ファイルの同期の開始] をクリックします。

コマンド伝播の設定

October 7, 2021

HA セットアップでは、プライマリノードで発行されたコマンドは、プライマリノードで実行される前に自動的にセカンダリに伝播し、実行されます。コマンドの伝播が失敗した場合、またはセカンダリでコマンドの実行が失敗した場合、プライマリノードはコマンドを実行し、エラーをログに記録します。コマンド伝播ではポート 3010 が使用されます。

HA ペア設定では、プライマリノードとセカンダリノードの両方でコマンドの伝播がデフォルトで有効になっています。HA ペアのいずれかのノードで、コマンド伝播を有効または無効にできます。1 次ノードでコマンド伝達を無効にすると、コマンドは二次ノードに伝達されません。セカンダリノードでコマンドの伝播を無効にすると、プライマリノードから伝播されたコマンドはセカンダリノードで実行されません。

> 注
>
> 伝播を再び有効化したら、必ず同期化を強制してください。

伝播を無効にしている間に同期が発生した場合、伝播を無効にする前に行った構成関連の変更は、セカンダリ・ノードと同期されます。これは、同期の進行中に伝播が無効になっている場合にも当てはまります。

コマンドラインインターフェイスを使用してコマンドの伝達を無効または有効にするには

コマンドプロンプトで入力します。

- set HA node -haProp DISABLED
- HA ノードを設定-haProp を設定

GUI を使用してコマンドの伝播を無効または有効にするには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. [プライマリノードが構成をセカンダリオプションに伝達する] を選択解除または選択します。

VLAN への高可用性同期トラフィックの制限

October 7, 2021

高可用性 (HA) 配置では、高可用性 (HA) 構成の維持に関連するトラフィックは、2 つの HA ノード間でフローされます。このトラフィックには、次のタイプがあります。

- 構成の同期
- 構成の伝播
- 接続ミラーリング
- ロードバランシングの永続性設定の同期
- 永続的なセッション同期
- セッション・ステートの同期

2 つのノード間でこの HA 関連トラフィックの適切なフローは、HA デプロイメントの機能に不可欠です。通常、HA 関連のトラフィックはボリュームが小さくなりますが、フェールオーバー中に非常に高くなる可能性があります。ステートフル接続のフェイルオーバーが有効になっていて、フェイルオーバーの前にプライマリだったノードが多数の接続を処理していた場合、この処理は非常に高くなります。

デフォルトでは、HA 関連のトラフィックは、NSIP アドレスがバインドされている VLAN を通過します。このトラフィックの急増に対応するために、HA 関連のトラフィックを管理トラフィックから分離し、フローを別の VLAN に制限できます。この VLAN を HA 同期 VLAN と呼びます。

HA SYNC VLAN を設定する前に考慮すべきポイント

- HA SYNC VLAN の設定は、伝播も同期もされません。つまり、HA SYNC VLAN はノード固有であり、各ノードで個別に設定されます。
- フルモードだけで設定をクリアすると、HA SYNC VLAN 設定は削除されます。
- 両方のノードがプライマリノードとして機能する状況を避けるために、HA SYNC VLAN の一部であるインターフェイスに対して HA MON を OFF に設定する必要があります。
- 管理インターフェイス（たとえば、0/1 および 0/2）は HA SYNC VLAN の一部であってはならず、HA 関連のトラフィックが管理インターフェイスを通過しないようにします。
- 管理インターフェイスで高可用性ハートビートメッセージを無効にし、HA SYNC VLAN インターフェイスで有効にすることをお勧めします。これらの推奨事項を満たすと、データインターフェイスで高可用性ハートビートメッセージも有効にできます。

  インターフェイスでの高可用性ハートビートメッセージの無効化の詳細については、Citrix ADC アプライアンスでの高可用性ハートビートメッセージの管理を参照してください。

Citrix ADC ノードで HA SYNC VLAN を構成するには、ローカルノードエンティティの HA SYNC VLAN パラメータを使用して、設定済みの VLAN を指定します。

コマンドラインを使用してローカルノードで **HA SYNC VLAN** を設定するには、次の手順を実行します。

コマンドプロンプトで入力します。

- `set ha node -syncvlan <VLANID>`

- show node

パラメータの説明:

syncvlan（同期 **VLAN**）: HA 関連のトラフィックが送信される VLAN。これには、同期、伝播、接続ミラーリング、ロードバランシングの永続性、設定の同期、永続セッション同期、およびセッション状態の同期のためのトラフィックが含まれます。ただし、HA ハートビートは任意のインターフェイスを使用できます。

GUI を使用してノードに **HA SYNC VLAN** を設定するには、次の手順を実行します。

1. ［**System**］ > ［**High Availability**］に移動します。
2. ローカルノードの変更中に **Sync VLAN** パラメータを設定します。

フェールセーフモードの設定

October 7, 2021

HA 構成では、フェールセーフモードでは、両方のノードが健全性チェックに不合格になったときに 1 つのノードが常にプライマリになります。これは、ノードが部分的にしか使用できない場合に、可能な限りトラフィックを処理するためのバックアップ方法が有効になるようにするためです。HA フェイルセーフモードは、各ノードで個別に設定されます。

次の表は、フェイルセーフのケースの一部を示しています。NOT_UP 状態は、ノードがヘルスチェックに失敗したが、部分的に利用可能であることを意味します。UP 状態は、ノードがヘルスチェックに合格したことを意味します。

ノード A（プライマリ）のヘルス状態	ノード B（セカンダリ）のヘルス状態	デフォルトの HA 動作	フェールセーフが有効化された HA の動作	説明
NOT_UP (最後に失敗しました)	NOT_UP (最初に失敗しました)	A（セカンダリ）、B（セカンダリ）	A（プライマリ）、B（セカンダリ）	両方のノードで障害が発生すると、最後のプライマリだったノードはプライマリのままになります。
NOT_UP (最初に失敗しました)	NOT_UP (最後に失敗しました)	A（セカンダリ）、B（セカンダリ）	A（セカンダリ）、B（プライマリ）	両方のノードで障害が発生すると、最後のプライマリだったノードはプライマリのままになります。

ノード A（プライマリ）のヘルス状態	ノード B（セカンダリ）のヘルス状態	デフォルトの HA 動作	フェールセーフが有効化された HA の動作	説明
上へ	上へ	A（プライマリ）、B（セカンダリ）	A（プライマリ）、B（セカンダリ）	両方のノードがヘルスチェックに合格した場合、フェイルセーフを有効にした場合の動作は変更されません。
上へ	NOT_UP	A（プライマリ）、B（セカンダリ）	A（プライマリ）、B（セカンダリ）	セカンダリノードのみで障害が発生した場合、フェイルセーフを有効にした場合の動作は変更されません。
NOT_UP	上へ	A（セカンダリ）、B（プライマリ）	A（セカンダリ）、B（プライマリ）	プライマリだけが故障した場合、フェイルセーフを有効にした場合の動作は変更されません。
NOT_UP	上 (セカンダリ滞留時)	A（セカンダリ）、B（セカンダリ）	A（プライマリ）、B（セカンダリ）	セカンダリが STAYSECONDARY として設定されている場合、プライマリは、障害が発生してもプライマリのままです。

コマンドラインインターフェイスを使用してフェールセーフモードを有効にするには

コマンドプロンプトで入力します。

```
set HA node [-failSafe ( **ON** | **OFF** )]
```

例

```
set ha node -failsafe ON
<!--NeedCopy-->
```


GUI を使用してフェールセーフモードを有効にするには

1. [システム] > [高可用性] に移動し、[ノード] タブでノードを開きます。
2. [フェイルセーフモード] で、[両方のノードが正常でない場合でも **1** つのプライマリノードを保持] オプションを選択します。

仮想 **MAC** アドレスの設定

October 7, 2021

仮想 MAC アドレスは、HA セットアップのプライマリノードとセカンダリノードによって共有されるフローティングエンティティです。

HA セットアップでは、プライマリノードは MIP、SNIP、VIP などのフローティング IP アドレスをすべて所有します。プライマリノードは、これらの IP アドレスに対するアドレス解決プロトコル (ARP) 要求に対して、独自の MAC アドレスで応答します。その結果、外部デバイス（アップストリームルーターなど）の ARP テーブルは、Floating IP アドレスとプライマリノードの MAC アドレスで更新されます。

フェイルオーバーが発生すると、セカンダリノードが新しいプライマリノードとして引き継がれます。次に、Gratuitous ARP（GARP）を使用して、プライマリから取得したフローティング IP アドレスをアドバタイズします。ただし、新しいプライマリがアドバタイズする MAC アドレスは、自身のインターフェイスの MAC アドレスです。

一部のデバイス（特に少数のルーター）は、Citrix ADC アプライアンスによって生成された GARP メッセージを受け付けません。その結果、一部の外部デバイスは、古いプライマリノードによってアドバタイズされた古い IP と MAC のマッピングを保持します。これにより、サイトがダウンする可能性があります。

この問題は、HA ペアの両方のノードに仮想 MAC を設定することで解決できます。両方のノードは、同じ MAC アドレスを持ちます。したがって、フェールオーバーが発生しても、セカンダリノードの MAC アドレスは変更されず、外部デバイスの ARP テーブルを更新する必要はありません。

仮想 MAC を作成するには、まず仮想ルータ ID（VRID）を作成し、インターフェイスにバインドする必要があります。(HA セットアップでは、VRID を両方のノードのインターフェイスにバインドする必要があります)。VRID がインターフェイスにバインドされると、システムは VRID を最後のオクテットとする仮想 MAC を生成します。

ここでは、次の詳細について説明します。

- IPv4 仮想 MAC の設定
- IPv6 仮想 MAC の設定

IPv4 仮想 **MAC** の設定

IPv4 仮想 MAC アドレスを作成してインターフェイスにバインドすると、インターフェイスから送信されるすべての IPv4 パケットは、インターフェイスにバインドされた仮想 MAC アドレスを使用します。インターフェイスにバインドされた IPv4 仮想 MAC がない場合は、インターフェイスの物理 MAC アドレスが使用されます。

汎用仮想 MAC は00:00:5e:00:01:<VRID>の形式です。たとえば、値 60 の VRID を作成してインターフェイスにバインドすると、その仮想 MAC は 00:00:5 e: 00:01:3 c になります。3c は VRID の 16 進表現です。1 ～255 の値で 255 個の VRID を作成できます。

IPv4 仮想 **MAC** の作成または変更

IPv4 仮想 MAC を作成するには、仮想ルータ ID を割り当てます。その後、仮想 MAC をインターフェイスにバインドできます。複数の VRID を同じインターフェイスにバインドすることはできません。仮想 MAC 設定を確認するには、仮想 MAC および仮想 MAC にバインドされたインターフェイスを表示して調べる必要があります。

コマンドラインインターフェイスを使用して仮想 **MAC** を追加するには

コマンドプロンプトで入力します。

- `add vrID`
- `bind vrid <id> -ifnum <interface_name>`
- `show vrID`

例

```
> add vrID 100
 Done
> bind vrid 100 -ifnum 1/1 1/2 1/3
 Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して仮想 **MAC** からインターフェイスをバインド解除するには

コマンドプロンプトで入力します。

- `unbind vrid <id> -ifnum <interface_name>`
- `show vrID`

GUI を使用して仮想 **MAC** を設定するには

[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC**] タブで新しい仮想 MAC を追加するか、既存の仮想 MAC を編集します。

IPv4 仮想 **MAC** の削除

IPv4 仮想 MAC を削除するには、その仮想ルータ ID を削除します。

コマンドラインインターフェイスを使用して **IPv4** 仮想 **MAC** を削除するには

コマンドプロンプトで入力します。

```
rm vrid <id>
```

例

```
rm vrid 100s
<!--NeedCopy-->
```

GUI を使用して **IPv4** 仮想 **MAC** を削除するには

[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC**] タブで IPv4 仮想 MAC を削除します。

IPv6 仮想 **MAC** の設定

Citrix ADC は、IPv6 パケット用の仮想 MAC6 をサポートしています。IPv4 仮想 MAC がインターフェイスにバインドされている場合でも、任意のインターフェイスを仮想 MAC6 にバインドできます。インターフェイスから送信される IPv6 パケットは、そのインターフェイスにバインドされた仮想 MAC6 を使用します。インターフェイスにバインドされた仮想 MAC6 がない場合、IPv6 パケットは物理 MAC を使用します。

仮想 **MAC6** の作成または変更

IPv6 仮想 MAC を作成するには、IPv6 仮想ルータ ID を割り当てます。その後、仮想 MAC をインターフェイスにバインドできます。複数の IPv6 VRID を 1 つのインターフェイスにバインドすることはできません。仮想 MAC6 の設定を確認するには、仮想 MAC6 および仮想 MAC6 にバインドされたインターフェイスを表示して調べる必要があります。

コマンドラインインターフェイスを使用して仮想 **MAC6** を追加するには

コマンドプロンプトで入力します。

- `add vrID6 <id>`
- `bind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

例

```
> add vrID6 100
Done
```



```
3  > bind vrID6 100 -ifnum 1/1 1/2 1/3
4   Done
5  <!--NeedCopy-->
```

コマンドラインインターフェイスを使用して仮想 **MAC6** からインターフェイスをバインド解除するには

コマンドプロンプトで入力します。

- `unbind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

GUI を使用して仮想 **MAC6** を設定するには

[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC6**] タブで新しい仮想 MAC6 を追加するか、既存の仮想 MAC6 を編集します。

仮想 **MAC6** の削除

IPv4 仮想 MAC を削除するには、その仮想ルータ ID を削除します。

コマンドラインインターフェイスを使用して仮想 **MAC6** を削除するには

コマンドプロンプトで入力します。

`rm vrid6 <id>`

例

```
1  rm vrid6 100s
2  <!--NeedCopy-->
```

GUI を使用して仮想 **MAC6** を削除するには

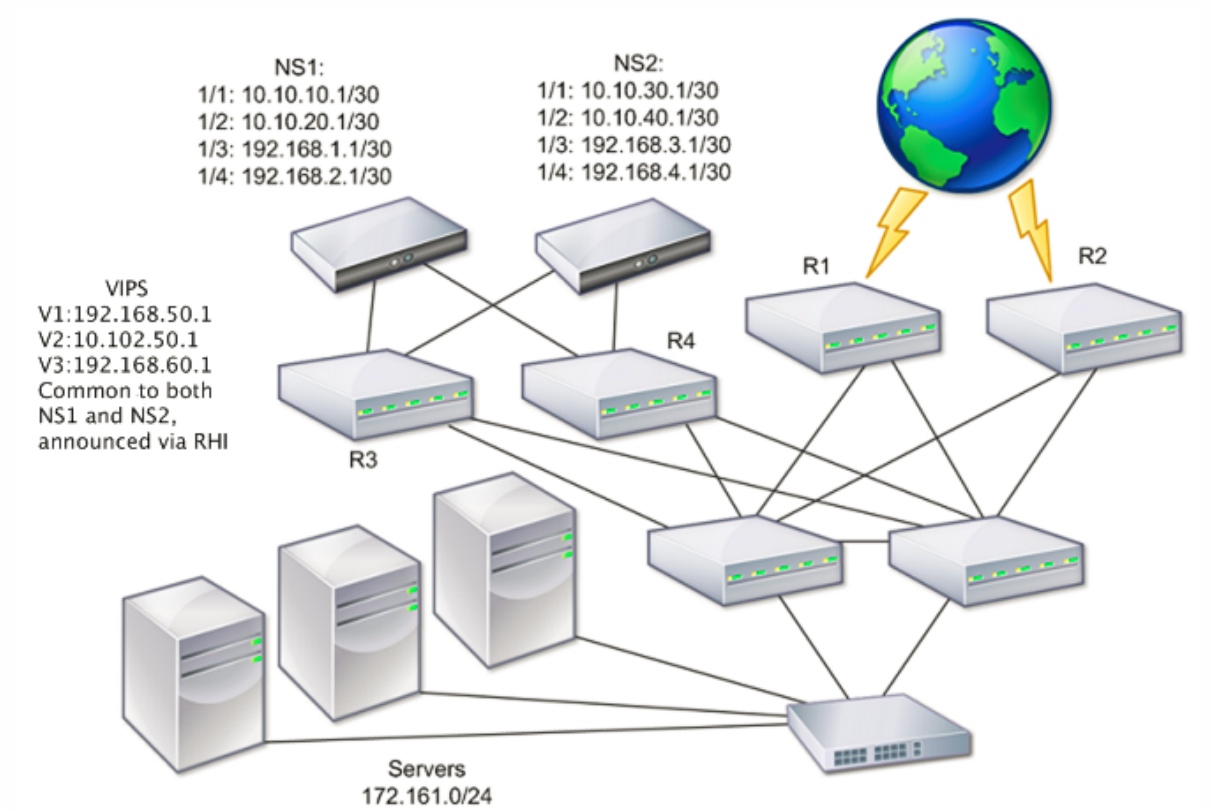
[システム] > [ネットワーク] > [**VMAC**] に移動し、[**VMAC6**] タブで仮想ルータ ID を削除します。

異なるサブネットでの高可用性ノードの構成

October 7, 2021

次の図は、2 つのシステムが異なるサブネットに配置された HA 配置を示しています。

図 1: ルーテッドネットワークでの高可用性



この図では、システム NS1 と NS2 が、2 つの異なるサブネット上の 2 つの別々のルータ R3 と R4 に接続されています。Citrix ADC アプライアンスは、ルーターを介してハートビートパケットを交換します。この設定は、任意の数のインターフェイスが関与する配置に対応するように拡張できます。

> 注:
>
> ネットワークでスタティックルーティングを使用する場合は、ハートビートパケットが正常に送受信されるように、すべてのシステム間にスタティックルートを追加する必要があります。（システムでダイナミックルーティングを使用する場合、スタティックルートは不要です）。

HA ペアのノードが 2 つの別々のネットワーク上に存在する場合、プライマリノードとセカンダリノードは独立したネットワーク構成を持つ必要があります。これは、異なるネットワーク上のノードが SNIP アドレス、VLAN、ルートなどのエンティティを共有できないことを意味します。このタイプの構成は、HA ペアのノードで設定可能なパラメータが異なるため、独立ネットワーク構成（INC）または対称ネットワーク構成（SNC）と呼ばれます。

次の表は、INC の設定可能なエンティティとオプションの概要と、各ノードでの設定方法を示しています。

NetScaler エンティティ	オプション
IP（NSIP/SNIP）	ノード固有。そのノードでのみアクティブです。
VIP	フローティング。
VLAN	ノード固有。そのノードでのみアクティブです。

NetScaler エンティティ	オプション
ルート	ノード固有。そのノードでのみアクティブです。リンクロードバランシングルートはフローティングです。
ACL	フローティング (共通)。両方のノードでアクティブです。
動的ルーティング	ノード固有。そのノードでのみアクティブです。また、セカンダリノードはルーティングプロトコルを実行し、アップストリームルータとピアリングする必要があります。
L2 モード	フローティング (共通)。両方のノードでアクティブです。
L3 モード	フローティング (共通)。両方のノードでアクティブです。
リバース NAT (RNAT)	VIP アドレスがフローティング（共通）であるため、NAT IP アドレスを仮想サーバの IP アドレス（VIP）に設定した RNAT 設定。

同じサブネット内の HA ノードを構成する場合と同様に、異なるサブネットに HA ノードを構成するには、2 つの Citrix ADC アプライアンスのそれぞれにログオンし、もう一方のアプライアンスを表すリモートノードを追加します。

リモートノードの追加

HA ペアの 2 つのノードが異なるサブネット上に存在する場合、各ノードは異なるネットワーク構成を持つ必要があります。したがって、HA ペアとして機能するように 2 つの独立したシステムを設定するには、設定プロセス中に INC モードを指定する必要があります。

HA ノードを追加するときは、接続されていないインターフェイスやトラフィックに使用されていないインターフェイスごとに HA モニタを無効にする必要があります。CLI ユーザーの場合、これは別の手順です。

コマンドラインインターフェイスを使用してノードを追加するには

コマンドプロンプトで入力します。

- `add ha node <id> <IPAddress> -inc ENABLED`
- `show ha node`

例


```
> add ha node 3 10.102.29.170 -inc ENABLED
 Done
> add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
  Done
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **HA** モニタを無効にするには

コマンドプロンプトで入力します。

- `set interface <ifNum> [-haMonitor ( **ON** | **OFF** )]`
- `show interface <ifNum>`

例

```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

GUI を使用してリモートノードを追加するには

1. [システム] > [高可用性] に移動し、[ノード] タブで新しいリモートノードを追加します。
2. [ダウンしているインターフェイス/チャネルの HA モニタをオフにする] と [セルフモードで INC (独立ネットワーク構成) モードをオンにする] オプションを選択してください。

ノードの削除

ノードを削除すると、ノードは高可用性構成になりません。

コマンドラインインターフェイスを使用してノードを削除するには

コマンドプロンプトで入力します。

```
rm ha node <id>
```

例

```
> rm ha node 2
  Done
<!--NeedCopy-->
```


GUI を使用してノードを削除するには

[システム] > [高可用性] に移動し、[ノード] タブでノードを削除します。

> 注:
>
> ネットワークビジュアライザーを使用して、高可用性（HA）ペアとして構成されている Citrix ADC アプライアンスを表示し、高可用性構成タスクを実行できます。

ルートモニタの設定

October 7, 2021

ルートモニタを使用すると、動的に学習されたルートまたはスタティックルートがテーブルに含まれているかどうかに関係なく、HA 状態を内部ルーティングテーブルに依存させることができます。HA 構成では、各ノードのルートモニタが内部ルーティングテーブルを監視し、特定のネットワークに到達するためのルートエントリが常に存在することを確認します。ルートエントリが存在しない場合、ルートモニタの状態は DOWN に変わります。

Citrix ADC アプライアンスにネットワークに到達するためのスタティックルートのみがあり、ネットワークのルートモニターを作成する場合は、スタティックルートの監視対象スタティックルート（MSR）を有効にする必要があります。MSR は、内部ルーティングテーブルから到達不能なスタティックルートを削除します。スタティックルートで MSR が無効になっている場合、到達不能なスタティックルートが内部ルーティングテーブルに残り、ルートモニタの目的がなくなります。

ルートモニタは、非 INC モードと INC モードの両方でサポートされます。

非 INC モードの HA でのルートモニタ	INC モードの HA でのルートモニタ
ルートモニタはノードによって伝播され、同期中に交換されます。	ルートモニタは、ノードによって伝播されず、同期中に交換されることもありません。
ルートモニタは、現在のプライマリノードでのみアクティブです。	ルートモニタは、プライマリノードとセカンダリノードの両方でアクティブです。
Citrix ADC アプライアンスは、ルートエントリが内部ルーティングテーブルに存在するかどうかに関係なく、常にルートモニターの状態を UP として表示します。	Citrix ADC アプライアンスは、対応するルートエントリが内部ルーティングテーブルに存在しない場合、ルートモニターの状態を DOWN と表示します。

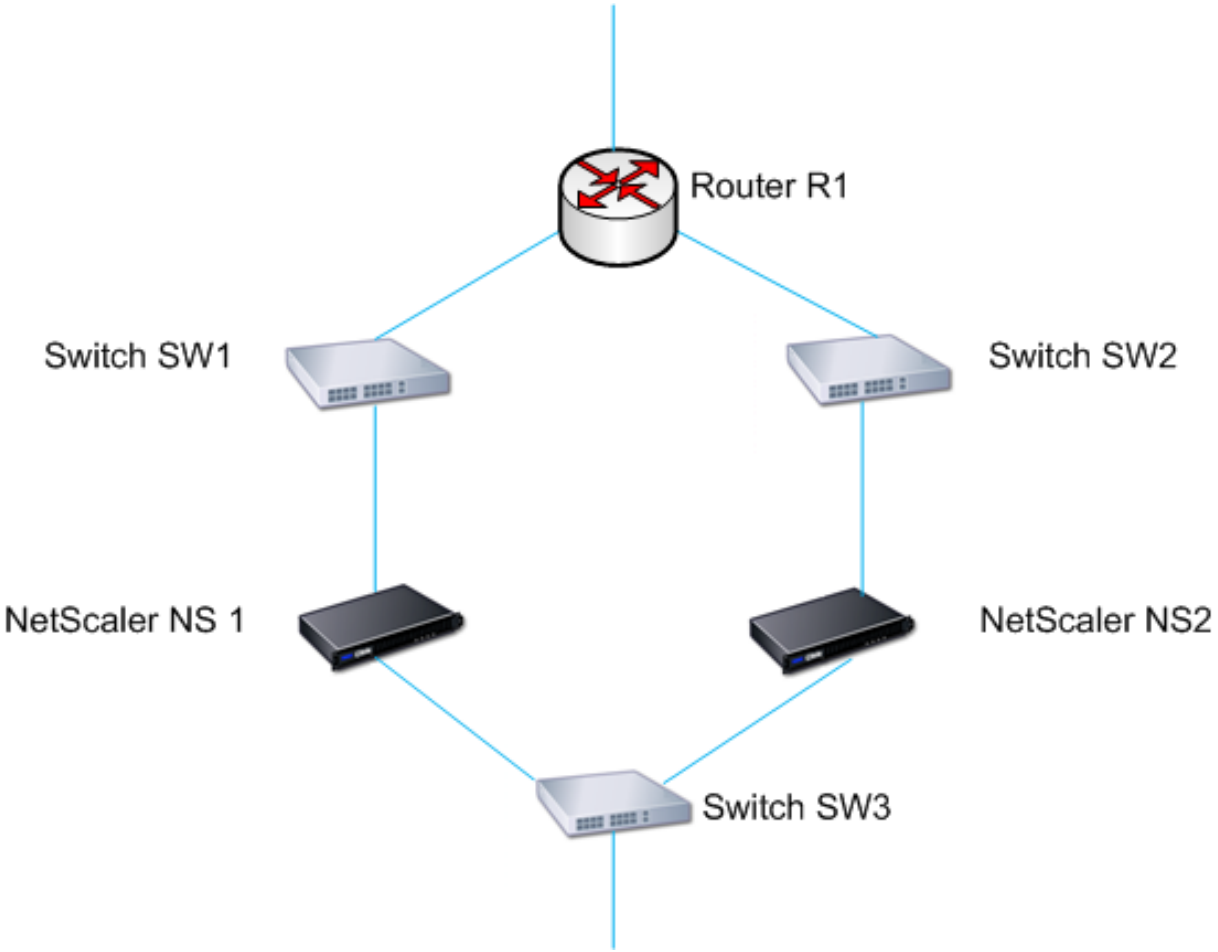
非 INC モードの HA でのルートモニタ	INC モードの HA でのルートモニタ
ルートモニタは 180 秒後にルートのモニタリングを開始します。[これは、ダイナミックルートの学習を可能にするために行われます。これには 180 秒かかることがあります。リブート、] フェールオーバー、v6 ルートの set route6 コマンド、v4 ルートの set route msr enable/disable コマンド、新しいルートモニタの追加。	-

ルートモニタは、プライマリノードからの Gateway の非到達可能性を HA フェールオーバーの条件の 1 つにする非 INC モードの HA 構成で役立ちます。

同じサブネット内に Citrix ADC アプライアンス NS1 と NS2 があり、ルーター R1 とスイッチ SW1、SW2、SW3 を持つ 2 アームトポロジにおける非 Inc モードの HA セットアップの例を考えてみましょう。

このセットアップでは R1 が唯一のルータであるため、現在のプライマリノードから R1 に到達できないときは必ず HA セットアップがフェールオーバーするようにします。各ノードでルートモニタ（それぞれ RM1 と RM2 など）を設定して、そのノードからの R1 の到達可能性を監視できます。

図 1:



NS1 を現在のプライマリノードとして使用すると、実行フローは次のようになります。

1. NS1 上のルートモニタ RM1 は、ルータ R1 のルートエントリの存在について、NS1 の内部ルーティングテーブルを監視します。NS1 および NS2 は、スイッチの SW1 または SW3 を介して定期的にハートビートメッセージを交換します。
2. スイッチ SW1 がダウンすると、NS1 のルーティングプロトコルは R1 に到達できないことを検出するため、内部ルーティングテーブルから R1 のルートエントリを削除します。NS1 および NS2 は、スイッチの SW3 を介して定期的にハートビートメッセージを交換します。
3. R1 のルートエントリが内部ルーティングテーブルに存在しないことを検出すると、RM1 はフェールオーバーを開始します。NS1 と NS2 の両方から R1 へのルートがダウンしている場合、いずれかのアプライアンスが R1 に到達して接続性を復元できるまで、180 秒ごとにフェイルオーバーが行われます。

高可用性ノードへのルートモニターの追加

1 つの手順でルートモニタを作成し、HA ノードにバインドします。

コマンドラインインターフェイスを使用してルートモニターを追加するには

コマンドプロンプトで入力します。

- bind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])
- show HA node

例

```
> bind HA node 0 -routeMonitor 10.102.71.0 255.255.255.0
 Done
> bind HA node 0 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
 Done
<!--NeedCopy-->
```

GUI を使用してルートモニタを追加するには

[システム] > [高可用性] に移動し、[ルートモニター] タブで [設定] をクリックします。

ルートモニタの削除

コマンドラインインターフェイスを使用してルートモニタを削除するには

コマンドプロンプトで入力します。

- unbind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])
- show ha node

例

```
unbind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
unbind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
<!--NeedCopy-->
```

GUI を使用してルートモニタを削除するには

[システム] > [高可用性] に移動し、[ルートモニタ] タブでルートモニタを削除します。

非 **INC** モードでのルートモニタによるフェールオーバーの制限

October 7, 2021

非 INC モードの HA 設定で、ルートモニタが両方のノードで失敗した場合、いずれかのノードがそれぞれのルートモニタによって監視されるすべてのルートに到達できるまで 180 秒ごとにフェールオーバーが発生します。

ただし、ノードに対して、ノードの [最大反転数] パラメータと [最大反転時間] パラメータを設定することで、特定の間隔でのフェイルオーバー回数を制限できます。いずれかの制限に達すると、フェールオーバーは発生しなくなり、そのノードでルートモニタが失敗しても、ノードはプライマリ（ただし、ノードの状態は NOT UP）として割り当てられます。NOT UP としてプライマリとノード状態として HA 状態のこの組み合わせは、スティックプライマリ状態と呼ばれています。

ノードが監視対象のすべてのルートに到達できる場合、次の監視障害によって、ノードの [最大反転数] パラメータと [最大フリップ時間] パラメータがリセットされ、[最大フリップ時間] パラメータで指定された時間が開始されます。

これらのパラメータは、各ノードで個別に設定されるため、伝播も同期もされません。

フェイルオーバー数を制限するためのパラメータ

- フリップの最大数 **(**最大フリップ**)**

  ルートモニタの障害が原因でフェールオーバーが発生した場合、非 INC モードの HA ノードの Maximum Flip Time インターバル内で許容されるフェールオーバーの最大数。

- 最大反転時間 **(**最大反転時間**)**

  非 INC モードの HA ノードで、ルートモニタの障害に起因するフェールオーバーが許可される時間（秒単位）。

コマンドラインインターフェイスを使用してフェールオーバーの数を制限するには

コマンドプロンプトで入力します。

- `set HA node [-maxFlips < positive_integer>] [-maxFlipTime <positive_integer >]`
- `show HA node [< id>]`

GUI を使用してフェールオーバーの数を制限するには

1. [システム] > [高可用性] に移動し、[ノード] タブでローカルノードを開きます。
2. 次のパラメーターを設定します。

- フリップの最大数
- 最大フリップ時間

```
1      > set ha node -maxFlips 30 -maxFlipTime 60
2      Done
3      > sh ha node
```



```
    1) Node ID: 0
    IP: 10.102.169.82 (NS)
    Node State: UP
    Master State: Primary
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen :None
    Interfaces causing Partial Failure:None
    SSL Card Status: NOT PRESENT
    Hello Interval: 200 msecs
    Dead Interval: 3 secs
    Node in this Master State for: 0:4:24:1 (days:hrs:min:sec)

    2) Node ID: 1
    IP: 10.102.169.81
    Node State: UP
    Master State: Secondary
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen : None
    Interfaces causing Partial Failure: None
    SSL Card Status: NOT PRESENT

    Local node information:
    Configured/Completed Flips: 30/0
    Configured Flip Time: 60
    Critical Interfaces: 1/1

    Done
<!--NeedCopy-->
```


スティッキプライマリステートの **SNMP** アラーム

ノードがスティッキプライマリになることを警告する場合は、高可用性セットアップのノードで HA-STICKY-PRIMARY SNMP アラームを有効にします。ノードがスティッキプライマリになると、トラップメッセージ（stickyPrimary（1.3.6.1.4.1.5951.1.0.138））を生成して警告し、設定されたすべての SNMP トラップ宛先に送信します。SNMP アラームとトラップ先の構成の詳細については、「SNMPv1 および SNMPv2 トラップを生成するように Citrix ADC を構成する」を参照してください。

よくある質問

2 つの Citrix ADC アプライアンス NS-1 と NS-2 を非 INC モードで高可用性セットアップする例を考えてみましょう。両方のノードでフリップの最大数と最大フリップ時間が同じ値で設定されています。

次の表に、この例で使用される設定を示します。

エンティティ	詳細
NS-1 の IP アドレス	10.102.173.211
NS-2 の IP アドレス	10.102.173.212
フリップの最大数	2
最大フリップ時間	200

フリップの最大数と最大フリップ時間の設定については、PDF を参照してください。

フェールオーバーインターフェイスセットの設定

October 7, 2021

フェールオーバーインターフェイスセット (FIS) は、インターフェイスの論理グループです。HA 構成では、FIS を使用することは、インターフェイスをグループ化してフェールオーバーを防止する方法です。これにより、1 つのインターフェイスに障害が発生しても、他の機能しているインターフェイスを使用できるようになります。FIS は、Citrix ADC クラスタのノードに対して構成することもできます。

FIS にバインドされていない HA MON インターフェイスは、いずれかに障害が発生するとフェールオーバーがトリガーされるため、クリティカルインターフェイス（CI）と呼ばれます。

> 注:
>
> FIS では、アクティブ構成とスタンバイ構成は作成されません。また、同じ VLAN にリンクに接続するときのブリッジングループも防止されません。

FIS を作成または修正する

コマンドラインインターフェイスを使用して **FIS** を追加し、インターフェイスをバインドするには

コマンドプロンプトで入力します。

- `add fis <name>`
- `bind fis \<name\> \<ifnum\> ...`
- `show fis \<name\>`

例

```
> add fis fis1
 Done
> bind fis fis1 1/3 1/5
 Done
<!--NeedCopy-->
```

非バインドインターフェイスが有効で HA MON がオンの場合、非バインドインターフェイスはクリティカルインターフェイス（CI）になります。

コマンドラインインターフェイスを使用して **FIS** からインターフェイスをバインド解除するには

コマンドプロンプトで入力します。

- `unbind fis \<name\> \<ifnum\> ...`
- `show fis \<name\>`

例

```
> unbind fis fis1 1/3
 Done
<!--NeedCopy-->
```

GUI を使用して **FIS** を構成するには

[システム] > [高可用性] に移動し、[フェールオーバーインターフェイスセット] タブで新しい FIS を追加するか、既存の FIS を編集します。

FIS の削除

FIS が削除されると、そのインターフェイスはクリティカルインターフェイスとしてマークされます。

コマンドラインインターフェイスを使用して **FIS** を削除するには

コマンドプロンプトで入力します。

```
rm fis <name>
```

例

```
> rm fis fis1
  Done
<!--NeedCopy-->
```

GUI を使用して **FIS** を削除するには

[システム] > [高可用性] に移動し、[フェールオーバーインターフェイスセット] タブで FIS を削除します。

フェイルオーバーの原因を理解する

October 7, 2021

次のイベントにより、ハイアベイラビリティ設定でフェールオーバーが発生する可能性があります。

1. セカンダリノードが、セカンダリに設定されたデッドインターバルを超える期間、プライマリからハートビートパケットを受信しない場合。(注 1 を参照)。
2. プライマリノードで SSL カードのハードウェア障害が発生します。
3. プライマリノードは、ネットワークインターフェイス上で 3 秒間ハートビートパケットを受信しません。
4. プライマリノードで、フェールオーバーインターフェイスセット（FIS）またはリンク集約（LA）チャネルの一部ではなく、HA モニタ（HAMON）が有効になっているネットワークインターフェイスで障害が発生します。(注 2 を参照)。
5. プライマリノードでは、FIS 内のすべてのインターフェイスで障害が発生します。(注 2 を参照)。
6. プライマリノードで、HAMON が有効になっている LA チャネルで障害が発生します。(注 2 を参照)。
7. プライマリノードでは、すべてのインターフェイスで障害が発生します（注 2 を参照)。この場合、フェールオーバーは HAMON の設定に関係なく発生します。
8. プライマリノードでは、すべてのインターフェイスが手動で無効になります。この場合、フェールオーバーは HAMON の設定に関係なく発生します。
9. いずれかのノードで force failover コマンドを発行して、フェールオーバーを強制します。
10. プライマリノードにバインドされているルートモニタがダウンします。

> 注 **1**:
>
> デッドインターバルの設定の詳細については、「通信間隔の設定」を参照してください。ノードがピアノードか

> らハートビートパケットを受信しない原因としては、次のようなものがあります。
>
> - ネットワーク構成の問題により、ハートビートが HA ノード間でネットワークを通過するのを防ぎます。
> - ピアノードでハードウェアまたはソフトウェア障害が発生し、その原因でフリーズする（ハング）、リブートしたり、ハートビートパケットの処理や転送を停止したりします。
>
> 注 **2**:
>
> この場合、fail は、show interface コマンドまたは GUI からわかるように、インターフェイスが有効になっていて DOWN 状態になったことを意味します。有効なインターフェイスがダウン状態になる原因としては、LINK DOWN および TXSTAL があります。

ノードのフェイルオーバーを強制する

October 7, 2021

たとえば、プライマリノードを交換またはアップグレードする必要がある場合に、フェールオーバーを強制することができます。プライマリノードまたはセカンダリノードのいずれかからフェイルオーバーを強制できます。強制フェールオーバーは継承されたり、同期されたりしません。強制フェールオーバー後の同期ステータスを表示するには、ノードのステータスを表示します。

次の状況では、強制フェールオーバーを実行できません。

- スタンドアロンシステムにフェールオーバーを強制する。
- セカンダリノードは無効です。
- セカンダリノードは、セカンダリノードを維持するように構成されています。

Citrix ADC アプライアンスは、強制フェイルオーバーコマンドの実行時に潜在的な問題を検出すると、警告メッセージを表示します。メッセージには警告の要因に関する情報が含まれており、手順を進める前に確認が求められます。

プライマリノード、セカンダリノード、およびノードがリッスンモードのときにフェイルオーバーを強制できます。

- プライマリノードでのフェールオーバーの強制実行。

  プライマリノードでフェイルオーバーを強制すると、プライマリがセカンダリになり、セカンダリがプライマリになります。強制フェイルオーバーは、プライマリノードがセカンダリノードが稼働していると判断できる場合にのみ可能です。

  セカンダリノードが DOWN の場合、強制フェールオーバーコマンドは次のエラーメッセージを返します。「無効なピアの状態のため操作できません。修正して再試行してください。」

  セカンダリシステムが要求状態または非アクティブの場合、次のエラーメッセージが表示されます。

  ```
  Operation not possible now. Please wait for the system to stabilize
  before retrying.
  ```

- セカンダリ・ノードでのフェイルオーバーの強制実行。

セカンダリノードから force failover コマンドを実行すると、セカンダリノードはプライマリノードになり、プライマリノードはセカンダリノードになります。強制フェールオーバーは、セカンダリノードの正常性が良好で、セカンダリとして構成されていない場合にのみ発生します。

セカンダリノードがプライマリノードになれない場合、またはセカンダリノードが（STAYSECERNDARY オプションを使用して）セカンダリとして構成されている場合、ノードは次のエラーメッセージが表示されます。

```
Operation not possible as my state is invalid. View the node for more information.
```

- ノードがリッスンモードである場合のフェールオーバーの強制実行。

  HA ペアの 2 つのノードで異なるバージョンのシステムソフトウェアが実行されている場合、上位バージョンを実行しているノードはリッスンモードに切り替わります。このモードでは、コマンドの伝播も同期も機能しません。

  両方のノードでシステムソフトウェアをアップグレードする前に、いずれかのノードで新しいバージョンをテストします。これを行うには、すでにアップグレードされているシステムでフェイルオーバーを強制する必要があります。アップグレードされたシステムはプライマリノードとして引き継がれますが、コマンドの伝播や同期は行われません。また、すべての接続を再確立する必要があります。

> 重要
>
> HA 同期処理の進行中にフェールオーバーを強制すると、HA セットアップのアクティブなデータセッションの一部が失われる可能性があります。したがって、HA 同期処理が完了するのを待ってから、強制フェールオーバー操作を実行します。

コマンドラインインターフェイスを使用してノードでフェイルオーバーを強制するには、次の手順を実行します。

コマンドプロンプトで入力します。

```
force HA failover
```

GUI を使用してノードでフェイルオーバーを強制するには、次の手順を実行します。

[システム] > [高可用性] に移動し、[ノード] タブでノードを選択し、[アクション] リストで [強制フェイルオーバー] を選択します。

セカンダリノードを強制的にセカンダリ状態にする

October 7, 2021

HA セットアップでは、プライマリノードの状態に関係なく、セカンダリノードをセカンダリのまま強制的に維持できます。

たとえば、プライマリノードをアップグレードする必要があり、プロセスに数秒かかるとします。アップグレード中に、プライマリノードが数秒間ダウンする場合がありますが、セカンダリノードが引き継ぐ必要はありません。プライマリノードで障害が検出された場合でも、セカンダリノードのままにしておきます。

セカンダリノードを強制的にセカンダリのまま維持すると、プライマリノードがダウンしても、セカンダリのまま維持されます。HA ペアの一方のノードのステータスをセカンダリのまま強制的に維持すると、そのノードは、HA 状態マシン遷移には参加しません。ノードのステータスは、STAYSECONDARY として表示されます。

ノードをセカンダリのまま強制的に維持する方法は、スタンドアロンノードとセカンダリノードのどちらでも機能します。スタンドアロンノードでは、ノードを追加して HA ペアを作成する前に、このオプションを使用する必要があります。新しいノードを追加すると、既存のノードはトラフィックの処理を停止し、セカンダリノードになります。新しいノードがプライマリノードになります。

> 注:
>
> システムをセカンダリのまま強制的に維持する場合、その強制を実施するプロセスは、伝播も同期もされません。コマンドを実行するノードのみが対象となります。

コマンドラインインターフェイスを使用してセカンダリノードを強制的にセカンダリにするには

コマンドプロンプトで入力します。

```
set ha node -hastatus STAYSECONDARY
```

GUI を使用してセカンダリノードを強制的にセカンダリにするには

[システム] > [高可用性] に移動し、[ノード] タブで、ローカルノードを開き、[**STAY SECONDARY**] を選択します。

プライマリノードを強制的にプライマリに留める

October 7, 2021

HA セットアップでは、フェールオーバー後も正常なプライマリノードを強制的にプライマリに保つことができます。このオプションは、HA ペアのプライマリノードのいずれかで有効にできます。このオプションを使用すると、プライマリノードが正常である限り、プライマリノードがプライマリ状態になります。

スタンドアロンノードでは、ノードを追加して HA ペアを作成する前に、このオプションを使用する必要があります。新しいノードを追加しても、既存のノードがプライマリノードとして機能し続け、新しいノードがセカンダリノードになります。

コマンドラインインターフェイスを使用してプライマリノードを強制的にプライマリに維持するには

コマンドプロンプトで入力します。

```
set ha node -hastatus STAYPRIMARY
```


GUI を使用してプライマリノードを強制的にプライマリに維持するには

[システム] > [高可用性] に移動し、[ノード] タブで、ローカルノードを開き、[**STAY PRIMARY**] を選択します。

高可用性ヘルスチェックの計算について

October 7, 2021

次の表は、ヘルスチェックの計算で調べた要因をまとめたものです。

- フェールオーバーインターフェイスセットの状態
- クリティカルインターフェイスの状態
- ルートモニタの状態

次の表は、ヘルスチェックの計算をまとめたものです。

フェールオーバーインターフェイスセット	クリティカルインターフェイス	ルートモニタ	条件
×	□	×	システムにクリティカルインターフェイスがある場合は、それらのクリティカルインターフェイスがすべて稼動している必要があります。
□	□	×	システムにフェールオーバーインターフェイスセットがある場合は、すべてのフェールオーバーインターフェイスセットが UP になっている必要があります。
□	□	□	システムにルートモニタが設定されている場合は、すべてのモニタ対象ルートがフェールオーバーインターフェイスセットに存在している必要があります。

高可用性に関する **FAQ**

October 7, 2021

1. HA 構成のノード間で HA 関連情報を交換するために使用するさまざまなポートは何ですか。

   HA 構成では、両方のノードは次のポートを使用して HA 関連情報を交換します。

   - ハートビートパケットを交換するための UDP ポート 3003。
   - 同期およびコマンド伝播用のポート 3010。

2. 同期をトリガーする条件は何ですか。

   同期は、次のいずれかの条件によってトリガーされます。

   - セカンダリが受信したプライマリノードのインカネーション番号が、セカンダリノードのインカネーション番号と一致しません。
     注：高可用性構成の両方のノードは、ノード構成ファイル内の構成の数をカウント
     インカネーション番号と呼ばれるカウンタを維持します。各ノードは、ハートビートメッセージ内の別のノードにインカネーション番号を送信します。次のコマンドでは、インカネーション番号は増分されません。
     a) すべての HA 設定関連コマンド。たとえば、HA ノードを追加し、HA ノードを設定し、HA ノードをバインドします。
     b) すべてのインターフェイス関連コマンド。たとえば、インターフェイスを設定し、インターフェイスを設定解除します。
     c) すべてのチャネル関連コマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。
   - 再起動後、セカンダリノードが起動します。
   - プライマリノードは、フェールオーバー後にセカンダリになります。

3. INC モードまたは非 INC モードの HA 構成で同期または伝播されない構成は何ですか。

   次のコマンドは、セカンダリノードに伝達も同期もされません。

   - すべてのノード固有の HA 構成コマンド。たとえば、HA ノードを追加し、HA ノードを設定し、HA ノードをバインドします。
   - インターフェイス関連のすべての構成コマンド。たとえば、インターフェイスを設定し、インターフェイスを設定解除します。
   - チャネル関連のすべての構成コマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。

   > 注：
   >
   > 以下の構成は、INC モードの HA でのみ同期も伝播もされません。各ノードには独自のものがあります：


```
- SNIP
```

- VLAN
- ルート（LLB ルートを除く）
- ルートモニタ
- RNAT ルール（NAT IP として VIP を使用するすべての RNAT ルールを除く）
- 動的ルーティング設定
- ネットプロファイル

4. セカンダリノードに追加された構成はプライマリで同期されますか?

   いいえ。セカンダリノードに追加された構成は、プライマリノードと同期されません。

5. 両方のノードが HA 構成でプライマリであると主張する理由は何ですか?

   最も可能性の高い理由は、プライマリノードとセカンダリノードは両方とも正常ですが、セカンダリはプライマリからハートビートパケットを受信しないことです。この問題は、ノード間のネットワークにある可能性があります。

6. 異なるシステムクロック設定で 2 つのノードを展開すると、HA 構成で問題が発生しますか?

   2 つのノードのシステムクロックの設定が異なると、次の問題が発生する可能性があります。

   - ログファイルエントリのタイムスタンプが一致しません。この状況では、問題のログエントリを分析することが困難になります。
   - フェイルオーバー後、負荷分散のためのあらゆる種類のクッキーベースの永続性に問題がある可能性があります。時間が大きく異なると、Cookie が予想よりも早く期限切れになり、永続セッションが終了する可能性があります。
   - 同様の考慮事項は、ノードの時間に関連する決定にも適用されます。

7. *Force HA sync* コマンドが失敗した場合の条件は何ですか。

   強制同期は、次のいずれかの状況で失敗します。

   - 同期が既に進行中の場合は、同期を強制します。
   - スタンドアロン Citrix ADC アプライアンスで同期を強制します。
   - セカンダリノードは無効です。
   - HA 同期は、現在のセカンダリノードで無効になっています。
   - 現在のプライマリノードで HA 伝播が無効になり、プライマリからの同期が強制されます。

8. *sync HA files* コマンドの失敗条件は何ですか。

   設定ファイルの同期は、次のいずれかの状況で失敗します。

   - スタンドアロンシステムの場合。
   - セカンダリノードが無効になっている場合。

9. HA 構成で、セカンダリノードがプライマリとして引き継ぐ場合、元のプライマリがオンラインに戻ると、セカンダリステータスに戻りますか。

いいえ。セカンダリノードがプライマリとして引き継ぐと、元のプライマリノードが再びオンラインに戻っても、プライマリノードはプライマリのままになります。ノードのプライマリとセカンダリのステータスを交換するには、*force failover* コマンドを実行します。

10. forcefailover コマンドが失敗する条件は何ですか。

    次の状況では、強制フェールオーバーを実行できません。

    - スタンドアロンシステムにフェールオーバーを強制する。
    - セカンダリノードは無効です。
    - セカンダリノードは、セカンダリノードを維持するように構成されています。
    - プライマリノードはプライマリノードとして構成されています。
    - ピアノードの状態が不明です。

高可用性に関する問題のトラブルシューティング

October 7, 2021

最も一般的な高可用性の問題は、高可用性機能がまったく動作しない、または断続的にしか動作しないことです。次に、一般的な高可用性の問題と、考えられる原因と解決策を示します。

- 問題
  Citrix ADC アプライアンスが高可用性セットアップで Citrix ADC アプライアンスをペアリングできない。
  - 原因
    ネットワーク接続
    解像度
    両方のアプライアンスがスイッチに接続され、インターフェイスが有効になっていることを確認します。
  - 原因
    デフォルトの管理者アカウントのパスワードが一致しません
    解像度
    両方のアプライアンスのパスワードが同じであることを確認します。
  - 原因
    IP の競合
    解像度
    両方のアプライアンスが一意の Citrix ADC IP（NSIP）アドレスを持っていることを確認します。アプライアンスの NSIP アドレスは同じであってはなりません。
  - 原因
    ノード ID が一致しません
    解像度
    両方のアプライアンスのノード ID 構成が一意であることを確認します。アプライアンスのノード ID 構成が同じであってはなりません。さらに、ノード ID の値を 1～64 の範囲で割り当てる必要があります。

- 原因

  RPC ノードのパスワードが一致しません

  解像度

  両方のノードが同じ RPC ノードのパスワードを持っていることを確認します。
- 原因

  管理者がリモートノードを無効にしました

  解像度

  リモートノードを有効にします。
- 原因

  ファイアウォールアプリケーションがハートビートパケットをブロックしました

  解像度

UDP ポート 3003 が許可されていることを確認します。

- 問題

  両方のアプライアンスがプライマリアプライアンスであると主張します。
  - 原因

    アプライアンス間のハートビートパケットがありません

    解像度

    UDP ポート 3003 がアプライアンス間の通信でブロックされていないことを確認します。
- 問題

  Citrix ADC アプライアンスが構成を同期できない。
  - 原因

    ファイアウォールアプリケーションが、必要なポートをブロックしています。

    解像度

    UDP ポート 3010（または安全な同期を持つ UDP ポート 3008）がアプライアンス間の通信でブロックされていないことを確認します。
  - 原因

    管理者が同期を無効にしました。

    解像度

    問題のあるアプライアンスで同期を有効にします。
  - 原因

    アプライアンスには、異なる Citrix ADC リリースまたはビルドがインストールされています。

    解像度

    アプライアンスを同じ Citrix ADC リリースまたはビルドにアップグレードします。
- **Issue**

  アプライアンス間でコマンド伝達が失敗する。
  - 原因

    ファイアウォールアプリケーションがポートをブロックしています。

    解像度

    UDP ポート 3011（または安全な伝播を持つ UDP ポート 3009）がアプライアンス間の通信でブロック

されていないことを確認します。

- 原因

  管理者がコマンドの伝播を無効にしました。

  解像度

  問題のあるアプライアンスでコマンド伝播を有効にします。

- 原因

  アプライアンスには、異なる Citrix ADC リリースまたはビルドがインストールされています。

  解像度

  アプライアンスを同じ Citrix ADC リリースまたはビルドにアップグレードします。

- 問題

  高可用性ペアの Citrix ADC アプライアンスは、強制フェイルオーバープロセスを実行できません。

  - 原因

    セカンダリノードが無効になっています。

    解像度

    セカンダリノードを有効にします。

  - 原因

    セカンダリノードは、セカンダリノードを維持するように構成されています。

    解像度

    セカンダリノードのセカンダリ高可用性ステータスを [セカンダリ滞在] から [有効] に設定します。

- 問題

  セカンダリアプライアンスは、フェールオーバープロセスの後にトラフィックを受信しません。

  - 原因

    アップストリームルーターは、Citrix ADC アプライアンスの GARP メッセージを認識しません。

    解像度

    セカンダリアプライアンスで仮想 MAC アドレスを設定します。

Citrix ADC アプライアンスでの高可用性ハートビートメッセージの管理

September 2, 2022

高可用性構成の 2 つのノードは、有効になっているすべてのインターフェイスで相互にハートビートメッセージを送受信します。ハートビートメッセージは、これらのインターフェイスの HA MON 設定に関係なく流れます。NSVLAN またはその両方（NSVLAN と SYNC）がアプライアンスに設定されている場合、ハートビートメッセージは、NSVLAN および SYNCVLAN の一部である有効なインターフェイスを介してのみ流れます。

ノードが有効なインターフェイスでハートビートメッセージを受信しない場合、指定された SNMP マネージャに重大なアラートが送信されます。これらの重大なアラートは、ピアノードへの接続の一部として構成されていないインターフェイスについて、誤警報を発し、管理者から不必要な注意を引きます。

この問題を解決するために、インターフェイスとチャネルの HAHeartbeat オプションを使用して、それらの HA ハ

ートビートメッセージフローを有効または無効にします。

コマンドラインインターフェイスを使用してインターフェイス上の高可用性ハートビートメッセージを管理するには

コマンドプロンプトで入力します。

- `set interface <ID> [-HAHeartBeat ( ON | OFF )]`
- `show interface <ID>`

コマンドラインインターフェイスを使用してチャネル上の高可用性ハートビートメッセージを管理するには

コマンドプロンプトで入力します。

- `set channel <ID> [-HAHeartBeat ( ON | OFF )]`
- `show channel <ID>`

GUI を使用してインターフェイスの高可用性ハートビートメッセージを管理するには

1. システム > ネットワーク > インターフェイスに移動します。
2. **HA** ハートビートパラメータを有効または無効にします。

GUI を使用してチャネル上の高可用性ハートビートメッセージを管理するには

1. システム > ネットワーク > チャネルに移動します。
2. **HA** ハートビートパラメータを有効または無効にします。

高可用性セットアップでの **Citrix ADC** 取り外しと交換

October 7, 2021

このトピックは、RMA の交換に対処するのに役立ちます。また、このトピックでは、設定のバックアップ、付属のソフトウェアバージョンのアップグレードまたはダウングレード、および ADC での RPC パスワードの設定方法についても説明します。

考慮すべきポイント

次の設定は、INC（独立ネットワーク構成）または非 INC モードの高可用性構成で同期または伝播されません。

- すべてのノード固有の HA 構成コマンド。たとえば、HA ノードを追加し、HA ノードを設定し、HA ノードをバインドします。
- インターフェイス関連のすべての構成コマンド。たとえば、インターフェイスを設定し、インターフェイスを設定解除します。
- チャネル関連のすべての構成コマンド。たとえば、チャネルの追加、チャネルの設定、チャネルのバインドなどを行います。
- すべてのインターフェイス HA モニタリング構成コマンド。

次の設定は、INC モード（独立ネットワーク設定）の HA 設定では同期も伝播もされません。

- SNIP
- VLAN
- ルート（LLB ルートを除く）
- ルートモニタ
- RNAT ルール（NAT IP として VIP を使用するすべての RNAT ルールを除く）
- 動的ルーティング設定

手順

高可用性セットアップで Citrix ADC を交換するには、以下の手順を実行します。

- アクティブな Citrix ADC セカンダリノードを削除する
- 交換用セカンダリノードの構成
- 交換用 ADC のソフトウェアビルドの検証と更新
- プライマリに一致するように新しいセカンダリにパスワードを設定する
- 交換用 ADC へのライセンスの追加
- プライマリと新しいセカンダリノード間の HA ペアの作成

アクティブなセカンダリ・ノードの削除

1. 両方の ADC にログオンし、次のコマンドを実行して、プライマリノードとセカンダリノードを確認します。

```
show ha node
<!--NeedCopy-->
```

2. プライマリ ADC にログオンし、プライマリノードの設定をバックアップし、変更前にファイルを ADC からコピーします。これらのファイルは、「/var /ns_sys_backup/」ディレクトリの下にあります。

   手順は次のとおりです。

   a) ADC の構成実行をメモリに保存します。

```
save ns config
<!--NeedCopy-->
```

   b) 完全バックアップファイルパッケージを作成します。

```
create system backup -level full
<!--NeedCopy-->
```


c) 基本的なバックアップファイルパッケージを作成します。

```
create system backup -level basic
<!--NeedCopy-->
```

3. すべてのバックアップファイルが生成されたら、先に進む前にデバイスからファイルをコピーしてください。

   Windows ターミナルからコマンドプロンプトを開き、バックアップファイルを ADC からローカルハードドライブにコピーします。これは、次のコマンドを使用して行うことができます。

```
pscp <username>@<NSIP>:<Target file source>  <Target file
   destination>
<!--NeedCopy-->
```

   例:

```
pscp nsroot@10.125.245.78:/var/ns_sys_backup/backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz c:\nsbackup\backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz
<!--NeedCopy-->
```

   プロンプトが表示されたら、指定した管理者アカウントのパスワードを入力し、Enter キーを押します。先に進む前に、すべてのバックアップバンドルがローカル PC にコピーされるまで、この手順を繰り返します。

4. SSH をセカンダリ ADC に接続し、ユニットを「STAYSECONDARY」ステータスに設定します。これにより、スワップ中に障害が検出された場合に、ユニットが主要な役割を引き受けることを強制しません。この手順を実行する前に、セカンダリ ADC に接続されていることを確認してください。

```
set ha node  -haStatus <state>
set ha node  -haStatus STAYSECONDARY
<!--NeedCopy-->
```

5. セカンダリ ADC の ノード状態が正常に「STAYSECONDARY」と表示されたら、プライマリ ADC に切り替えてセカンダリノードを削除し、次のコマンドを実行します。

```
save ns config
<!--NeedCopy-->
```


プライマリ ADC にログインした状態で、次のコマンドを実行します。

a) 次のコマンドを実行して、セカンダリ HA ノードを表す数値を特定します。

```
show ha node
<!--NeedCopy-->
```

b) 次のコマンドを実行して、プライマリ HA ペアからセカンダリ ADC を取り外します。

```
rm ha node <node ID>
<!--NeedCopy-->
```

c) 次のコマンドを実行して、設定を保存します。

```
save ns config
<!--NeedCopy-->
```

d) セカンダリ ADC を取り外した状態で、シャットダウンし、切断し、ネットワークからセカンダリ ADC を取り外します。

注。切断する前に、必ずすべての接続にラベルを付けてください。

代替セカンダリ・ノードの構成

1. 交換用 ADC を取り付けた状態で、新しいデバイスの電源を入れます。この時点では、ネットワーク接続は接続しないでください。
2. 起動が完了したら、コンソール・ポートを使用して ADC に接続し、ユニットへの接続に使用する NSIP を設定します。
3. プロンプトが表示されたら、[**4**] を選択します。

   注。この例では、交換用 ADC に異なる NSIP を使用しています。元のセカンダリユニットの IP を使用する場合は、新しい ADC をプライマリ HA ユニットにバインドする前に、交換時に変更することができます。
4. これで ADC が起動します。次に、管理トラフィックに使用されるネットワークインターフェイスを接続し、IP アドレスがネットワークから到達可能であることを確認します。

交換用 **ADC** のソフトウェアビルドの検証と更新

新しいユニットをプライマリ ADC に同期する前に、両方の ADC が同じビルドを実行していることを確認する必要があります。

1. ADC のバージョンを確認するには、次のコマンドを実行します。

```
show version
<!--NeedCopy-->
```

2. 新しいセカンダリ ADC で、アップグレードに使用するサブフォルダを **/var** に作成します。
3. ［Citrix Downloads］ に移動し、プライマリ ADC で実行されているビルドバージョンに一致する適切なパッケージをダウンロードします。
4. .tgz ファイルをダウンロードし、展開します。

```
tar -xvzf ”file.tgz”
<!--NeedCopy-->
```

5. 抽出したファイルをセカンダリ ADC にコピーします。Windows ターミナルで、「コマンドプロンプト」を開き、抽出された.tgz ビルドパッケージを含むディレクトリに移動し、次の pscp コマンドを実行します。

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

例:

```
C:\inetpub>pscp c:\inetpub\build-12.1-47.14_nc.tgz nsroot@10
   .20.245.80:/var/NS_upg_12.1_47.14/build-12.1-47.14_nc.tgz
<!--NeedCopy-->
```

6. ファイルが転送されたら、セカンダリ ADC に戻り、アップグレードします。詳細な手順については、「Citrix ADX スタンドアロンアプライアンスのアップグレード」を参照してください。
7. 新しいセカンダリがリブートしたら、SSH でユニットに戻り、アップグレードが成功し、ビルドがプライマリのビルドと一致することを確認します。

プライマリと一致する代替セカンダリノードのパスワードを設定

注: この時点で、新しいセカンダリ ADC の管理 IP（NSIP）アドレスを変更する場合は、先に進む前に変更してください。

新しいセカンダリ ADC のパスワードを変更し、現在プライマリ ADC にあるパスワードと一致させます。

1. デフォルトの管理者（nsroot）アカウントのパスワードがプライマリ ADC と同じであることを確認します。これは、SSH 経由で新しいセカンダリユニットにログインしているときに、次のコマンドを使用して行います。

```
set system user <user> <password>
<!--NeedCopy-->
```

   このコマンドは、指定したユーザーのパスワードを設定/リセットします。

2. プライマリ ADC と新しいセカンダリ ADC に SSH 接続し、パスワードが一致することを確認します。

交換用セカンダリノードへのライセンスの追加

新しい ADC が更新され、ペアリングできる状態になったら、交換ノード用の適切なライセンスをダウンロードしてインストールします。

1. https://www.citrix.comに移動して、新しい交換用ユニットのライセンスを要求し、ダウンロードします。
2. 適切なライセンスをすべてダウンロードしたら、新しいセカンダリ ADC に SSH で接続し、次のコマンドを入力してライセンスの現在の状態を確認します。

```
show license
<!--NeedCopy-->
```

3. Windows ターミナル・コマンド・プロンプトから、次のコマンドを使用して、ライセンス・ファイルを新しいセカンダリ・ADC にアップロードする必要があります。

   注。複数のライセンスがある場合は、すべてのライセンスがアップロードされるまでこの手順を繰り返します。

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

   例:

```
C:\inetpub>pscp c:\inetpub\NS-VPX-3K-LIC-020030ad0024.lic
   nsroot@10.125.245.80:/nsconfig/license/NS-VPX-3K-LIC-020030
   ad0024.lic
<!--NeedCopy-->
```


4. SSH を新しいセカンダリ ADC に接続し、次のコマンドを使用してウォームリブートを実行します。

```
reboot  -w
<!--NeedCopy-->
```

装置が再起動されたら、装置に SSH 接続し、もう一度 show license コマンドを実行します。この時点で、ライセンスを適用する必要があります。

プライマリと新しいセカンダリノード間の高可用性の設定

この時点で、Citrix ADC ユニットを高可用性ペアに加入する準備が整いました。詳細については、「高可用性の構成」を参照してください。

再試行をリクエストする

December 7, 2021

Citrix ADC アプライアンスが HTTP 要求を受信したが、バックエンドサーバーとの接続に失敗した場合、アプライアンスは再試行ディレクティブを使用します。リクエストの再試行は、接続障害のシナリオに対処し、アプライアンスが次に使用可能なサービスを選択してリクエストを転送できるようにします。リクエストの再試行を行うことで、クライアントはラウンドトリップ時間 (RTT) を節約できます。

再試行要求機能は、次の接続障害シナリオに適用できます。

- HTTP リクエストの受信時にバックエンドサーバーが TCP 接続をリセットした場合。詳細については、「再試行のリクエスト」を参照してください。
- 接続の確立中にバックエンドサーバーが TCP 接続をリセットした場合。詳細については、「再試行のリクエスト」を参照してください。
- アプライアンスが HTTP リクエストを送信したときに、バックエンドサーバーからの応答が（設定されたタイムアウト値に基づいて）タイムアウトした場合。詳細については、「再試行のリクエスト」を参照してください。

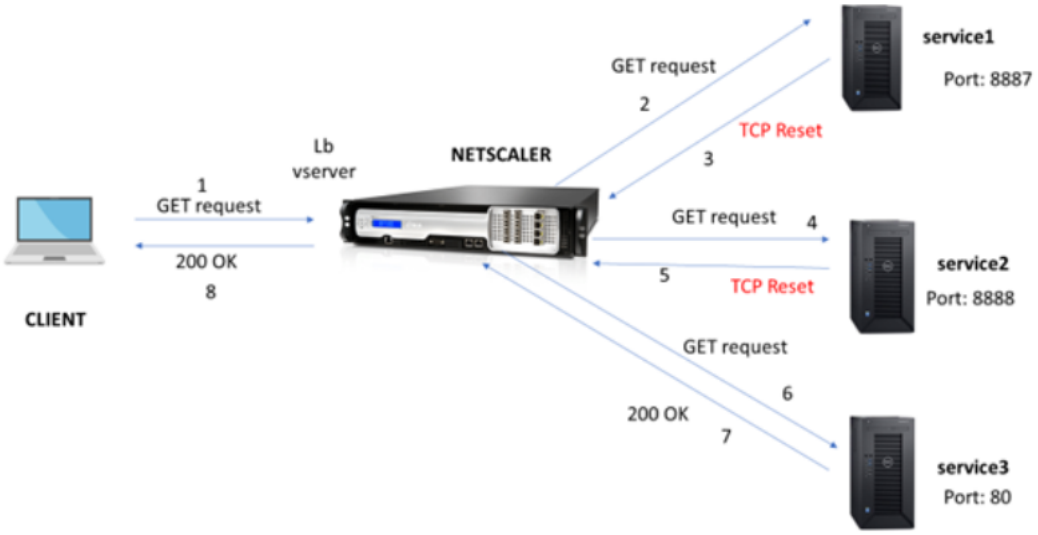
バックエンドサーバーが **TCP** 接続をリセットした場合に再試行を要求する

October 7, 2021

バックエンドサーバーが TCP 接続をリセットすると、要求の再試行機能は、リセットをクライアントに送信する代わりに、次に使用可能なサーバーに要求を転送します。リロードバランシングを実行することにより、アプライアンスが次に利用可能なサービスに対して同じ要求を開始したときに、クライアントは RTT を保存します。

バックエンドサーバーが **TCP** 接続をリセットするときの要求の再試行のしくみ

次の図は、コンポーネントが互いにどのように相互作用するかを示しています。



1. プロセスは、アプライアンスで appqoe 機能を有効にすることから始まります。
2. クライアントが HTTP または HTTPS 要求を送信すると、負荷分散仮想サーバーはその要求をバックエンドサーバーに送信します。
3. 要求されたサービスが利用できない場合、バックエンドサーバーは TCP 接続をリセットします。
4. appqoe 構成で「再試行」が有効になっていて、必要な再試行回数が指定されている場合、負荷分散仮想サーバーは、構成された負荷分散アルゴリズムを使用して、次に使用可能なアプリケーションサーバーに要求を転送します。
5. 負荷分散仮想サーバーが応答を受信した後、アプライアンスは応答をクライアントに転送します。
6. 使用可能なバックエンドサーバーが再試行回数以下であり、すべてのサーバーがリセットを送信した場合、アプライアンスは 500 内部サーバーエラーに応答します。使用可能なサーバーが 5 つあり、再試行回数が 6 に設定されているシナリオを考えてみます。5 つのサーバーすべてが接続をリセットすると、アプライアンスは 500 内部サーバーエラーをクライアントに返します。
7. 同様に、バックエンドサーバーの数が再試行回数を超え、バックエンドサーバーが接続をリセットした場合、アプライアンスはリセットをクライアントに転送します。3 つのバックエンドサーバーがあり、再試行回数が 2 に設定されているシナリオを考えてみます。3 つのサーバーが接続をリセットすると、アプライアンスはリセット応答をクライアントに送信します。

GET メソッドの要求再試行を構成します

GET メソッドの再試行機能を構成するには、次の手順を実行する必要があります。

1. AppQoE を有効にする

2. AppQoE アクションの追加
3. AppQoE ポリシーの追加
4. AppQoE ポリシーを負荷分散仮想サーバーにバインドする

AppQoE を有効にする

コマンドプロンプトで入力します。

```
enable ns feature appqoe
```

AppQoE アクションの追加

AppQoE アクションを構成して、TCP リセット後にアプライアンスを再試行するかどうかと再試行の回数を指定する必要があります。

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries <
positive_integer>]
```

例**:**

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```

RetryonReset バックエンドサーバーが TCP 接続をリセットした場合は、再試行を有効にします。
numretries. 再試行回数。

AppQoE ポリシーの追加

AppQoE を実装するには、特定のキューで着信 HTTP または SSL 要求に優先順位を付けるように AppQoE ポリシーを構成する必要があります。

コマンドプロンプトで入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
```

例**:**

```
add appqoe policy reset_policy -rule http.req.method.eq(get)-action reset_action
```

appqoe ポリシーを負荷分散仮想サーバーにバインドする

バックエンドサーバーが TCP パケット要求をリセットし、負荷分散仮想サーバーがその要求を次に利用可能なサービスに転送するようにする場合は、負荷分散仮想サーバーを AppQoE ポリシーにバインドする必要があります。

コマンドプロンプトで入力します。

```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```


例:

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```

POST 要求の要求再試行を構成します

バックエンドサーバーにデータを書き込むバランスリクエストをリロードするときは、常に注意を払う必要があります。このようなリクエストの場合は、コンテンツの長さが短いことを確認してください。コンテンツの長さが長いと、リソースが消費される可能性があります。以下の手順に従って、POST 要求のリロードバランシングを構成します。

1. AppQoE を有効にする
2. AppQoE アクションの追加
3. AppQoE ポリシーの追加
4. appQoE ポリシーを負荷分散仮想サーバーにバインドします

AppQoE を有効にする

コマンドプロンプトで入力します。

```
enable ns feature appqoe
```

Appqoe アクションを追加

TCP リセットと再試行回数の後に再試行するには、AppQoE アクションを追加する必要があります。

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries <
positive_integer>]
```

例:

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```

Appqoe ポリシーを追加する

AppQoE を実装するには、AppQoE ポリシーを構成して、特定のキューで接続をキューに入れる方法を定義する必要があります。

コマンドプロンプトで入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
```

例:

```
add appqoe policy reset_policy -rule HTTP.REQ.CONTENT_LENGTH.le(2000)-
action reset_action
```


> 注:
>
> コンテンツの長さが 2000 未満の場合、要求の再試行機能を制限する場合は、この構成を使用できます。

負荷分散仮想サーバーを **AppQoE** ポリシーにバインドする

バックエンドサーバーが TCP パケット要求をリセットし、負荷分散仮想サーバーが特定のキューを介して次に利用可能なサービスに要求を転送するようにする場合は、負荷分散仮想サーバーを AppQoE ポリシーにバインドする必要があります。

コマンドプロンプトで入力します。

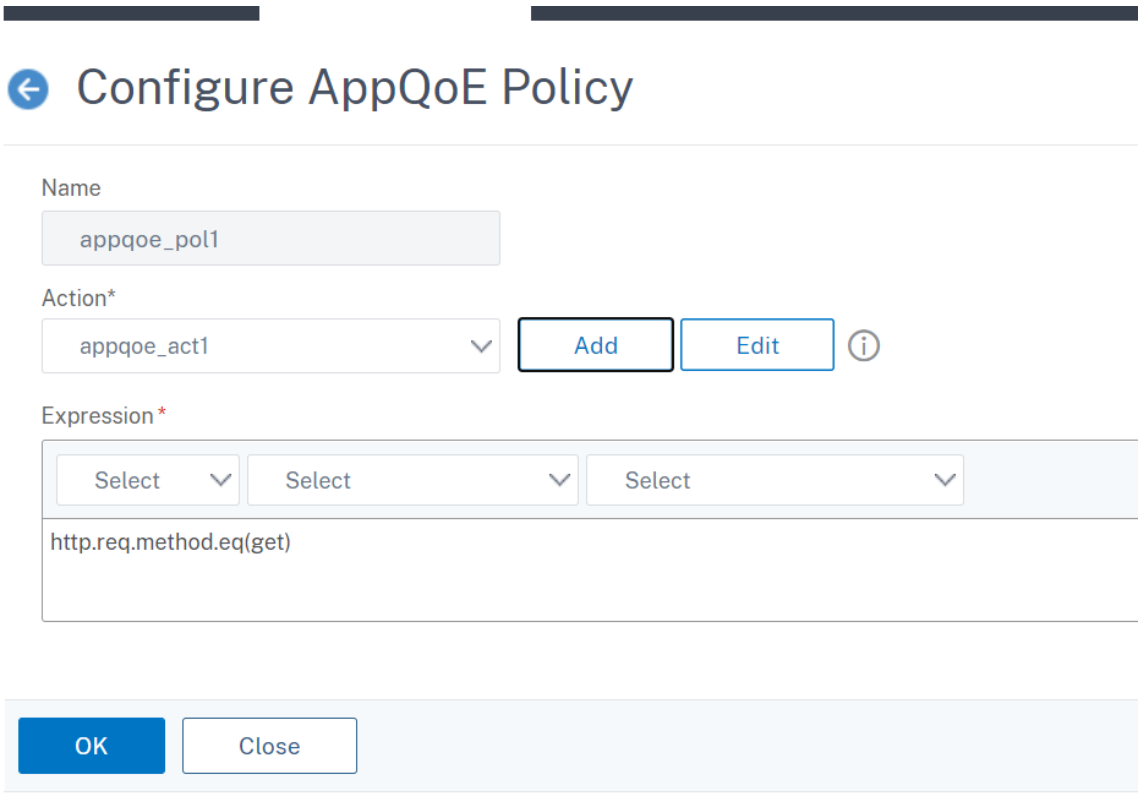
```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```

例**:**

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```

Citrix ADC GUI を使用して、要求の再試行のために **AppQoE** ポリシーを構成します

1. **AppExpert** > **AppQoE** > ポリシーに移動します。
2. [**AppQoE** ポリシー] ページで、[追加] をクリックします。
3. **AppQoE** ポリシーの作成ページで、次のパラメーターを設定します。
   a。名称。AppQoE ポリシー名
   b。アクション。アクションを追加または編集します。アクションを作成するには、。
   c. 式。`HTTP.REQ.CONTENT_LENGTH.le (2000)` ポリシー式を選択または入力します。
4. **[**作成**]** して **[**閉じる**]** をクリックします。

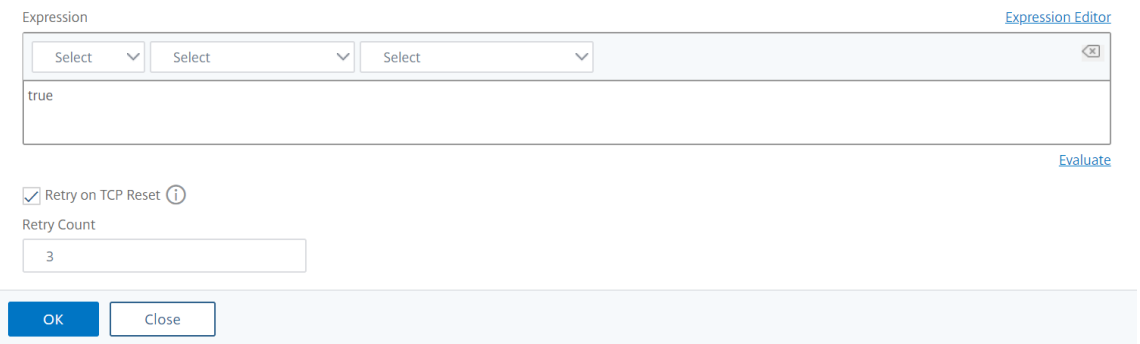


Citrix ADC GUI を使用して、要求の再試行バランシング用の **AppQoE** アクションを構成します

1. **AppExpert** > **AppQoE** > アクションに移動します。
2. [**AppQoE** アクション] ページで、[追加] をクリックします。
3. **[AppQoE** アクションの作成**]** ページで、TCP リセット時に再試行するための次のパラメーターを設定します。

   a。TCP リセットを再試行します。TCP リセットの再試行アクションを有効にするには、このチェックボックスをオンにします。

   b。再試行回数。再試行回数を入力します。
4. **[**作成**]** して **[**閉じる**]** をクリックします。



TCP SYN の確立時にバックエンドサーバーがリセットされたときに、**GET** メソッドの要求の再試行を構成します

CLI と GUI の構成は、GET メソッドの場合と同様の手順です。詳細については、「GET メソッドのリクエスト試行を設定する」セクションを参照してください。バックエンドサーバーが接続セクションをリセットしたとき。

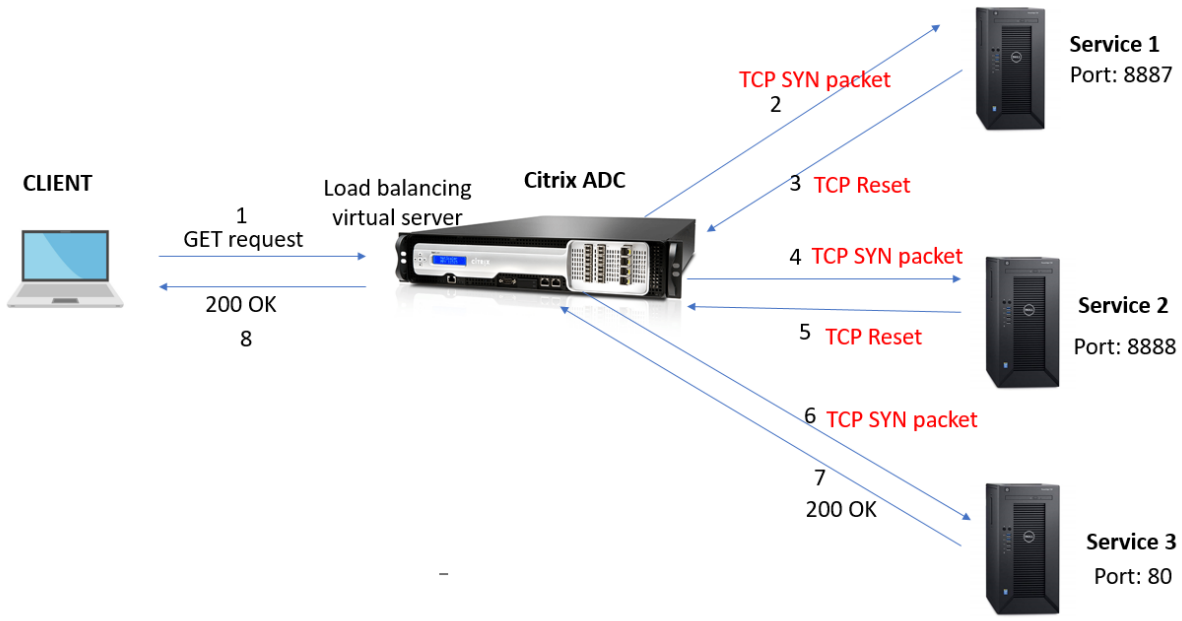
接続の確立中にバックエンドサーバーが **TCP** 接続をリセットした場合は、再試行を要求します

October 7, 2021

バックエンドサーバーが TCP 接続をリセットすると、要求の再試行機能は、リセットをクライアントに送信する代わりに、次に使用可能なサーバーに要求を転送します。リロードバランシングを実行することにより、アプライアンスが次に利用可能なサービスに対して同じ要求を開始したときに、クライアントは RTT を保存します。

バックエンドサーバーが **SYN** 確立時に **TCP** 接続をリセットするときの要求の再試行のしくみ

次の図は、コンポーネントが相互作用することを示しています。



1. プロセスは、アプライアンスで appqoe 機能を有効にすることから始まります。
2. クライアントが HTTP または HTTPS 要求を送信すると、負荷分散仮想サーバーがバックエンドサーバーへの接続を開始します。
3. 要求されたサービスが TCPSYN の確立で利用できない場合、バックエンドサーバーは TCP 接続をリセットします。

4. appqoe 構成で「再試行」が有効になっていて、必要な再試行回数が指定されている場合、負荷分散仮想サーバーは、構成された負荷分散アルゴリズムを使用して、次に使用可能なアプリケーションサーバーに要求を転送します。
5. 負荷分散仮想サーバーが応答を受信した後、アプライアンスは応答をクライアントに転送します。
6. 使用可能なバックエンドサーバーが再試行回数以下であり、すべてのサーバーがリセットを送信した場合、アプライアンスは 500 内部サーバーエラーに応答します。使用可能なサーバーが 5 つあり、再試行回数が 6 に設定されているシナリオを考えてみます。5 つのサーバーすべてが接続をリセットすると、アプライアンスは 500 内部サーバーエラーをクライアントに返します。
7. 同様に、バックエンドサーバーの数が再試行回数を超え、バックエンドサーバーが TCP SYN の確立時に接続をリセットした場合、アプライアンスはリセットをクライアントに転送します。3 つのバックエンドサーバーがあり、再試行回数が 2 に設定されているシナリオを考えてみます。3 つのサーバーが接続をリセットすると、アプライアンスはリセットパケットをクライアントに送信します。

TCP SYN の確立時にバックエンドサーバーがリセットされたときの要求の再試行（**GET** および **POST** メソッド）を構成します

CLI および GUI の構成は、GET および POST メソッドの場合と同様の手順です。詳細については、「GET メソッドのリクエスト再試行の構成」の「バックエンドサーバーが接続セクションをリセットするときの POST メソッドのリクエスト再試行を構成する」を参照してください。tyuu

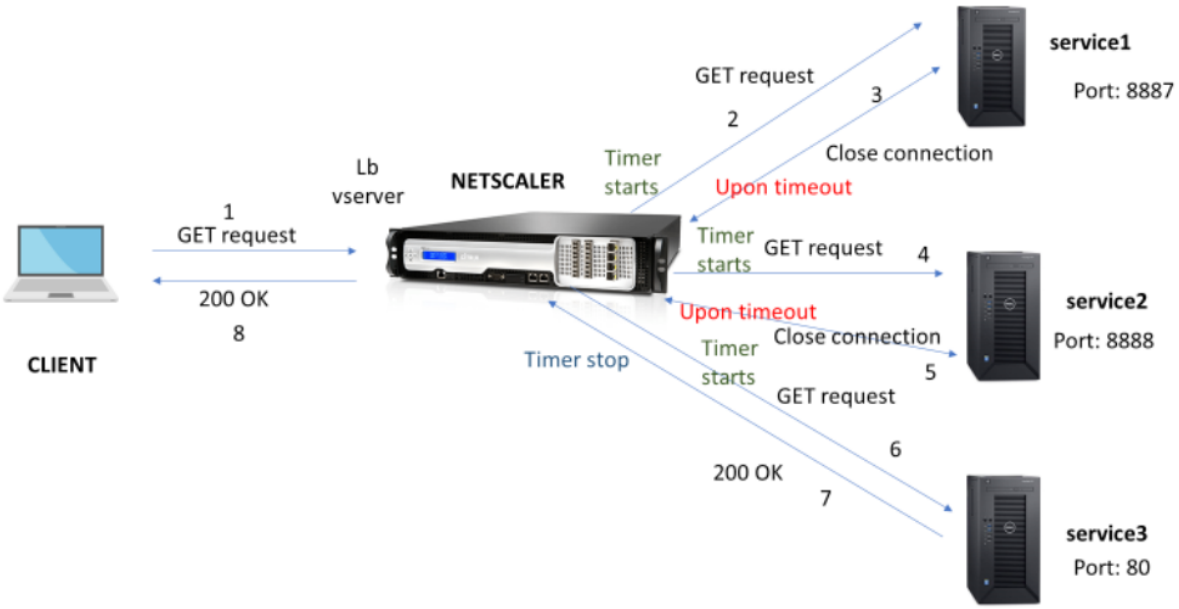
バックエンドサーバーの応答がタイムアウトした場合は、再試行を要求します

October 7, 2021

リクエストの再試行は、バックエンドサーバーがリクエストに応答するのに時間がかかる場合、アプライアンスがタイムアウト時に再ロードバランシングを実行し、次に利用可能なサーバーにリクエストを転送するもう 1 つのシナリオで使用できます。

バックエンドサーバーの応答がタイムアウトした場合の要求の再試行のしくみ

次の図は、コンポーネントが相互作用することを示しています。



1. プロセスは、アプライアンスで appqoe 機能を有効にすることから始まります。
2. appqoe 構成には、ミリ秒単位の「retryOnTimeout」パラメーターがあります。
3. アプライアンスが要求を送信し、サーバーが応答するのにさらに時間がかかる場合、アプライアンスは構成されたタイムアウト値に基づいて再負荷分散を実行します。アプライアンスは接続をリセットし、別のサービスを選択して、サーバーの応答を待つ代わりに要求を転送します。
4. 負荷分散仮想サーバーが応答を受信した後、アプライアンスは応答をクライアントに転送します。タイムアウトパラメータを使用すると、アプライアンスはサーバーの応答を待機し続けることができなくなり、RTT が増加します。
5. 使用可能なバックエンドサーバーが再試行回数以下であり、すべてのサーバーが要求に対してタイムアウトした場合、アプライアンスは 500 内部サーバーエラーに応答します。使用可能なサーバーが 5 つあり、再試行回数が 6 に設定されているシナリオを考えてみます。5 つのサーバーすべてが要求に対してタイムアウトした場合、アプライアンスは 500 の内部サーバーエラーをクライアントに返します。
6. 同様に、バックエンドサーバーの数が再試行回数を超え、バックエンドサーバーが要求に応じてタイムアウトした場合、アプライアンスは、サーバーが応答を送信するか、クライアントのアイドル接続がタイムアウトするまで、最後のサービスを待機し続けます。3 つのバックエンドサーバーがあり、再試行回数が 2 に設定されているシナリオを考えてみます。3 つのサーバーすべてが要求に応じてタイムアウトした場合、アプライアンスは、サーバーが応答を送信するか、クライアントのアイドル接続がタイムアウトするまで、3 番目のサービスを待機し続けます。

バックエンドサーバーの応答がタイムアウトした場合のリクエストの再試行（**GET** および **POST** メソッド）を構成する

タイムアウト時に GET メソッドの要求再試行を構成するには、次の手順を完了する必要があります。

1. appqoe を有効にする
2. appqoe アクションを構成します
3. appqoe ポリシーを追加する
4. appqoe ポリシーを負荷分散仮想サーバーにバインドする

> 注:
>
> タイムアウト時の要求の再試行シナリオは、POST メソッドにも適用できます。

appqoe を有効にする

コマンドプロンプトで入力します。

```
enable ns feature appqoe
```

タイムアウトの **appqoe** アクションを追加します

タイムアウト時に再試行するように appqoe アクションを構成し、再試行の回数を定義する必要があります。

コマンドプロンプトで入力します。

```
add appqoe action <name> -retryOnTimeout <msecs> -numRetries <positive_integer
>
```

例**:**

```
add appqoe action appact1 -retryOnTimeout 35 -numRetries 5
```

appqoe ポリシーを追加する

appqoe を実装するには、appqoe ポリシーを構成して、接続をキューに入れる方法を定義する必要があります。

コマンドプロンプトで入力します。

```
add appqoe policy <name> -rule <rule> -action <name>
```

例**:**

```
add appqoe policy timeout_policy -rule http.req.method.eq(get)-action
appact1
```

appqoe ポリシーを負荷分散仮想サーバーにバインドする

バックエンドサーバーの応答に時間がかかり、負荷分散仮想サーバーが要求を次に利用可能なサービスに転送するようにする場合は、appqoe ポリシーを分散仮想サーバーにバインドする必要があります。

コマンドプロンプトで入力します。


```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```

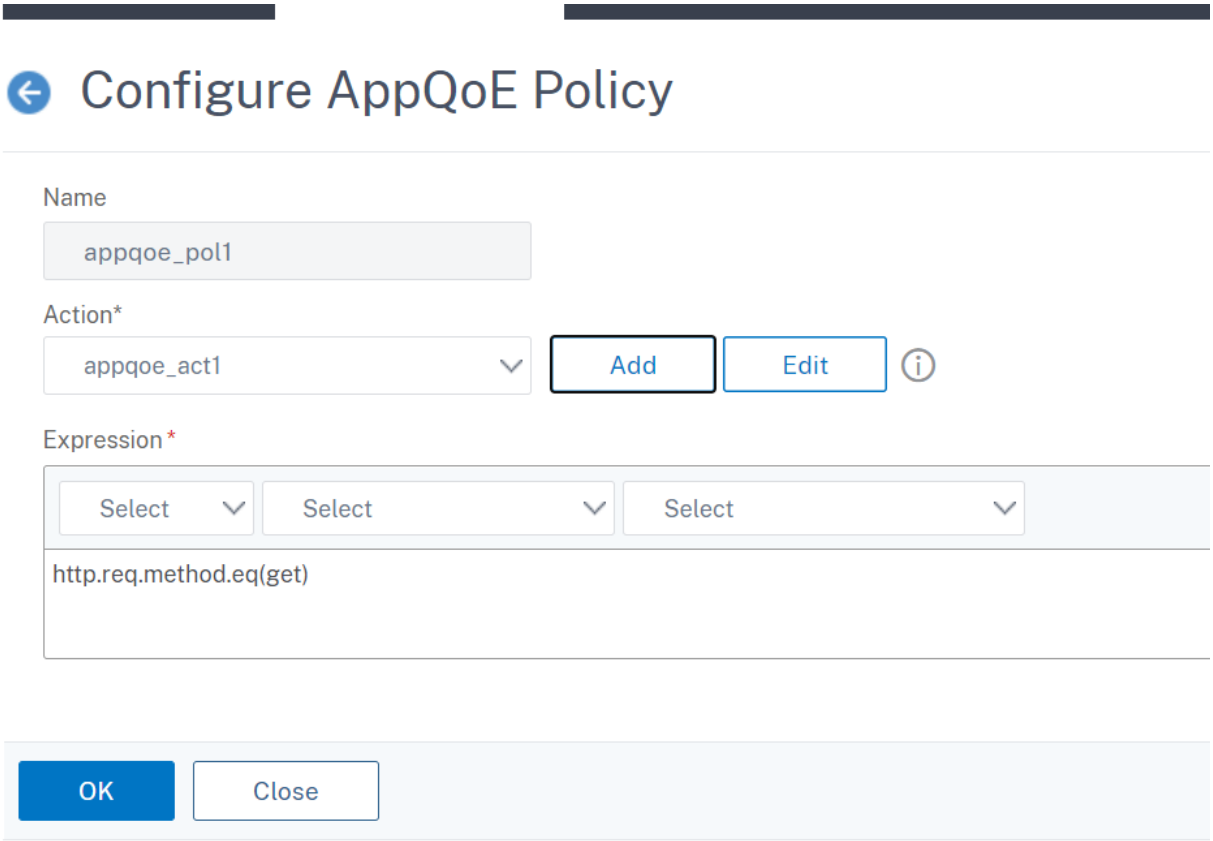
例:

```
bind lb vserver v1 -policyName timeout_policy -type REQUEST -priority 1
```

Citrix ADC GUI を使用して、要求の再試行のために **AppQoE** ポリシーを構成します

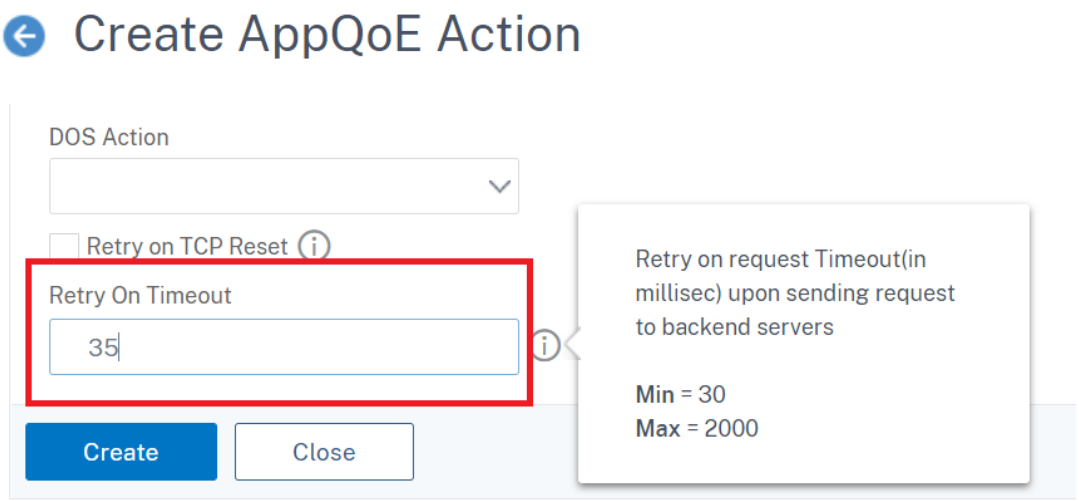
1. **AppExpert >** 書き換え **>** ポリシーに移動します。
2. [**AppQoE** ポリシー] ページで、[追加] をクリックします。
3. [**AppQoE** ポリシーの作成] ページで、次のパラメーターを設定します。

   a. 名称。AppQoE ポリシー名
   b。アクション。アクションを追加または編集します。新しいアクションを作成するには、「AppQoE アクションの作成」セクションを参照してください。
   c. 式。「http.req.method.eq（get)」ポリシー式を選択または入力します。
4. **[作成]** して **[閉じる]** をクリックします。



Citrix ADC GUI を使用して、要求の再試行のために **AppQoE** アクションを構成します

1. **AppExpert > AppQoE > Action** に移動します。
2. [**AppQoE** アクション] ページで、[追加] をクリックします。
3. [**AppQoE** アクションの作成] ページで、バックエンドサーバーの応答タイムアウト時に再試行するための次のパラメーターを設定します。
   a。タイムアウトで再試行します。バックエンドサーバーにリクエストを送信すると、リクエストのタイムアウト（ミリ秒単位）で再試行します。
4. **[作成]** して **[閉じる]** をクリックします。



TCP の最適化

October 7, 2021

TCP は、次の最適化手法と輻輳制御戦略（またはアルゴリズム）を使用して、データ伝送におけるネットワークの輻輳を回避します。

輻輳制御方針

TCP は、インターネット接続の確立と管理、伝送エラーの処理、および Web アプリケーションとクライアントデバイスのスムーズな接続に長い間使用されてきました。しかし、パケット損失はネットワークの輻輳だけに依存せず、輻輳が必ずしもパケット損失を引き起こすわけではないため、ネットワークトラフィックの制御が困難になっています。したがって、輻輳を測定するには、TCP アルゴリズムはパケット損失と帯域幅の両方に焦点を当てる必要があります。

比例レート回復（**PRR**）アルゴリズム

TCP 高速回復メカニズムは、パケット損失に起因する Web 遅延を低減します。新しい比例速度回復（PRR）アルゴリズムは、損失回復中に TCP データを評価する高速回復アルゴリズムです。輻輳制御アルゴリズムによって選択されたターゲットウィンドウに適した分数を使用して、Rate-Halving の後にパターン化されます。これにより、ウィンドウの調整が最小限に抑えられ、リカバリ終了時の実際のウィンドウ・サイズは Slow-Start しきい値（ssthresh）に近い値になります。

TCP ファスト・オープン（**TFO**）

TCP ファストオープン (TFO) は、TCP の最初のハンドシェイク中に、クライアントとサーバー間で迅速かつ安全なデータ交換を可能にする TCP メカニズムです。この機能は、Citrix ADC アプライアンスの仮想サーバーにバインドされた TCP プロファイルで TCP オプションとして使用できます。TFO は、Citrix ADC アプライアンスが生成する TCP 高速オープンクッキー（セキュリティクッキー）を使用して、仮想サーバーへの TFO 接続を開始するクライアントを検証および認証します。この TFO メカニズムを使用すると、1 回のフルラウンドトリップに必要な時間だけアプリケーションのネットワーク遅延を減らすことができます。これにより、短い TCP 転送で発生する遅延が大幅に削減されます。

TFO の仕組み

クライアントが TFO 接続を確立しようとすると、それ自体を認証する最初の SYN セグメントで TCP ファストオープン Cookie が含まれます。認証が成功すると、Citrix ADC アプライアンスの仮想サーバーがスリーウェイハンドシェイクの最終 ACK セグメントを受信していなくても、SYN-ACK セグメントにデータを含めることができます。これにより、通常の TCP 接続と比較して最大 1 回のラウンドトリップが節約されます。この場合、データを交換する前に 3 ウェイハンドシェイクが必要になります。

クライアントとバックエンドサーバーは、次の手順を実行して TFO 接続を確立し、最初の TCP ハンドシェイク中にデータを安全に交換します。

1. クライアントが自身を認証するための TCP ファスト・オープン Cookie を持っていない場合、SYN パケット内のファスト・オープン Cookie 要求を Citrix ADC アプライアンス上の仮想サーバーに送信します。
2. 仮想サーバーにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、アプライアンスはクッキーを生成し（シークレットキーでクライアントの IP アドレスを暗号化して）、生成された高速オープン Cookie を TCP オプションフィールドに含む SYN-ACK でクライアントに応答します。
3. クライアントは、アプライアンス上の同じ仮想サーバーへの将来の TFO 接続用に Cookie をキャッシュします。
4. クライアントは、同じ仮想サーバーへの TFO 接続を確立しようとすると、HTTP データと共に (TCP オプションとして) キャッシュされた高速オープン Cookie を含む SYN を送信します。
5. Citrix ADC アプライアンスはクッキーを検証し、認証に成功すると、サーバーは SYN パケット内のデータを受け入れ、SYN-ACK、TFO クッキー、HTTP レスポンスでイベントを確認します。

> 注:
>
> クライアント認証が失敗した場合、サーバーはデータをドロップし、セッションのタイムアウトを示す SYN のみでイベントを確認応答します。

1. サーバー側では、サービスにバインドされた TCP プロファイルで TFO オプションが有効になっている場合、Citrix ADC アプライアンスは、接続しようとしているサービスに TCP 高速オープン Cookie が存在するかどうかを決定します。
2. TCP ファスト・オープン・クッキーが存在しない場合、アプライアンスは SYN パケットでクッキー要求を送信します。
3. バックエンドサーバーが Cookie を送信すると、アプライアンスは Cookie をサーバー情報キャッシュに保存します。
4. アプライアンスが所定の宛先 IP ペアのクッキーをすでに持っている場合、古いクッキーを新しいものと置き換えます。
5. 仮想サーバが同じ SNIP アドレスを使用して同じバックエンドサーバに再接続しようとしたときに、サーバ情報キャッシュで cookie が使用可能な場合、アプライアンスは SYN パケット内のデータを cookie と結合し、バックエンドサーバに送信します。
6. バックエンドサーバーは、データと SYN の両方でイベントを確認します。

注：サーバーが SYN セグメントのみでイベントを確認した場合、Citrix ADC アプライアンスは、元のパケットから SYN セグメントと TCP オプションを削除した後、直ちにデータパケットを再送信します。

TCP ファーストオープーの設定

TCP ファストオープン（TFO）機能を使用するには、関連する TCP プロファイルで TCP ファストオープンオプションを有効にし、TFO Cookie タイムアウトパラメータにそのプロファイルのセキュリティ要件に適した値を設定します。

CLI を使用して **TFO** を有効または無効にする

コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存のプロファイルで TFO を有効または無効にします。

注**:** デフォルト値は DISABLED です。

```
add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
unset tcpprofile <TCP Profile Name> - tcpFastOpen
Examples
add tcpprofile Profile1 - tcpFastOpen
Set tcpprofile Profile1 - tcpFastOpen Enabled
unset tcpprofile Profile1 - tcpFastOpen
```



```
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **TCP** ファストオープン **cookie** タイムアウト値を設定するには

コマンドプロンプトで入力します。

```
    set tcpparam  - tcpfastOpenCookieTimeout <Timeout Value>
    Example
    set tcpprofile  - tcpfastOpenCookieTimeout 30secs
<!--NeedCopy-->
```

GUI を使用して **TCP** ファストオープンを構成するには

1. [設定] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. [**TCP** プロファイルの構成] ページで、[**TCP** 高速オープン] チェックボックスをオンにします。
3. [**OK**]、[完了] の順にクリックします。

GUI を使用して **TCP** ファストクッキーのタイムアウト値を設定するには

[設定**]** > [システム] **> [**設定**] > [TCP** パラメータの変更] に移動し、**[TCP** パラメータの構成**]** ページに移動して、TCP ファストオープン Cookie タイムアウト値を設定します。

TCP HyStart

新しい TCP プロファイルパラメータ HyStart を使用すると、HyStart アルゴリズムが有効になります。HyStart アルゴリズムは、終了するセーフポイント（ssthresh）を動的に決定するスロースタートアルゴリズムです。これにより、大量のパケット損失を伴わずに、輻輳回避への移行が可能になります。この新しいパラメータは、デフォルトでは無効になっています。

輻輳が検出されると、HyStart は輻輳回避フェーズに入ります。これを有効にすると、パケット損失が大きい高速ネットワークのスループットが向上します。このアルゴリズムは、トランザクションの処理中に最大帯域幅に近い状態を維持するのに役立ちます。したがって、スループットを向上させることができます。

TCP HyStart の構成

HyStart 機能を使用するには、関連する TCP プロファイルで Cubic HyStart オプションを有効にします。

コマンドラインインターフェイス（**CLI**）を使用して **HyStart** を構成するには

コマンドプロンプトで、次のいずれかのコマンドを入力して、新規または既存の TCP プロファイルで HyStart を有効または無効にします。

```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

例:

```
    add tcpprofile profile1 -hystart ENABLED
    set tcpprofile profile1 -hystart ENABLED
    unset tcprofile profile1 -hystart
<!--NeedCopy-->
```

GUI を使用して HyStart サポートを構成するには

1. [設定] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. [**TCP** プロファイルの設定] ページで、[キュービックハイスタート] チェックボックスをオンにします。
3. [**OK**]、[完了] の順にクリックします。

TCP バーストレート制御

TCP 制御メカニズムは、高速モバイルネットワーク上でバーストなトラフィックフローにつながり、ネットワーク全体の効率に悪影響を及ぼす可能性があります。輻輳やデータのレイヤ 2 再送信などのモバイルネットワークの状態により、TCP 確認応答が送信者に集中して到着し、送信のバーストをトリガーします。短いパケット間ギャップで送信されるこれらの連続したパケットのグループは、TCP パケットバーストと呼ばれます。トラフィックバーストを克服するために、Citrix ADC アプライアンスは TCP バーストレート制御技術を使用します。この方法では、データがバーストに送信されないように、ラウンドトリップ時間全体にわたってデータをネットワークに均等に配置します。このバーストレート制御技術を使用すると、スループットが向上し、パケットドロップレートが低下します。

TCP バーストレート制御の仕組み

Citrix ADC アプライアンスでは、この手法により、パケットの送信がラウンドトリップ時間（RTT）の全期間に均等に分散されます。これは、バーストを減らすために進行中の TCP セッションのパケットを出力するさまざまなネットワーク条件を識別する TCP スタックとネットワークパケットスケジューラを使用することによって実現されます。

送信側では、確認応答を受信した直後にパケットを送信する代わりに、送信側はスケジューラ（動的設定）または TCP プロファイル（固定設定）によって定義されたレートでパケットを拡散するようにパケットの送信を遅らせることができます。

TCP バーストレート制御の設定

関連する TCP プロファイルで TCP バーストレート制御オプションを使用し、バーストレート制御パラメータを設定します。

コマンドラインを使用して **TCP** バーストレート制御を設定するには

コマンドプロンプトで、新しい、または既存のプロファイルで、次の TCP バーストレート制御コマンドのいずれかを設定します。

注**:** デフォルト値は DABLED です。

```
add tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
    | Fixed

set tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
    | Fixed

unset tcpprofile <TCP Profile Name> -burstRateControl Disabled |
    Dynamic | Fixed
<!--NeedCopy-->
```

各項目の意味は次のとおりです。

無効 — バーストレート制御が無効になっている場合、Citrix ADC アプライアンスは maxBurst 設定以外のバースト管理を実行しません。

固定 — TCP バーストレート制御が「固定」の場合、アプライアンスは、TCP プロファイルに記載されている TCP 接続ペイロード送信レート値を使用します。

Dynamic：バーストレート制御が「Dynamic」の場合、接続はさまざまなネットワーク条件に基づいて規制され、TCP バーストが減少します。このモードは、TCP 接続が ENDPOINT モードの場合にのみ機能します。ダイナミックバーストレート制御が有効の場合、TCP プロファイルの maxBurst パラメータは有効になりません。

```
add tcpProfile  profile1 -burstRateControl Disabled

set tcpProfile profile1 -burstRateControl Dynamic

```



```
unset tcpProfile profile1 -burstRateControl Fixed
<!--NeedCopy-->
```

コマンドラインインターフェイスを使用して **TCP** バーストレート制御パラメータを設定するには

コマンドプロンプトで入力します。

```
set ns tcpprofile nstcp_default_profile  - burstRateControl <type of
    burst rate control>  - tcprate <TCP rate> -rateqmax <maximum
   bytes in queue>

T1300-10-2> show ns tcpprofile nstcp_default_profile
        Name: nstcp_default_profile
        Window Scaling status:  ENABLED
        Window Scaling factor: 8
        SACK status: ENABLED
        MSS: 1460
        MaxBurst setting: 30 MSS
        Initial cwnd setting: 16 MSS
        TCP Delayed-ACK Timer: 100 millisec
        Nagle's Algorithm: DISABLED
        Maximum out-of-order packets to queue: 15000
        Immediate ACK on PUSH packet: ENABLED
        Maximum packets per MSS: 0
        Maximum packets per retransmission: 1
        TCP minimum RTO in millisec: 1000
        TCP Slow start increment: 1
        TCP Buffer Size: 8000000 bytes
        TCP Send Buffer Size: 8000000 bytes
        TCP Syncookie: ENABLED
        Update Last activity on KA Probes: ENABLED
        TCP flavor: BIC
        TCP Dynamic Receive Buffering: DISABLED
        Keep-alive probes: ENABLED
        Connection idle time before starting keep-alive probes: 900
            seconds
        Keep-alive probe interval: 75 seconds
        Maximum keep-alive probes to be missed before dropping
           connection: 3
        Establishing Client Connection: AUTOMATIC
        TCP Segmentation Offload: AUTOMATIC
        TCP Timestamp Option: DISABLED
        RST window attenuation (spoof protection): ENABLED
```



```
33              Accept RST with last acknowledged sequence number: ENABLED
34              SYN spoof protection: ENABLED
35              TCP Explicit Congestion Notification: DISABLED
36              Multipath TCP: DISABLED
37              Multipath TCP drop data on pre-established subflow:
                    DISABLED
38              Multipath TCP fastopen: DISABLED
39              Multipath TCP session timeout: 0 seconds
40              DSACK: ENABLED
41              ACK Aggregation: DISABLED
42              FRTO: ENABLED
43              TCP Max CWND : 4000000 bytes
44              FACK: ENABLED
45              TCP Optimization mode: ENDPOINT
46              TCP Fastopen: DISABLED
47              HYSTART: DISABLED
48              TCP dupack threshold: 3
49              Burst Rate Control: Dynamic
50              TCP Rate: 0
51              TCP Rate Maximum Queue: 0
52  <!--NeedCopy-->
```

GUI を使用して **TCP** バーストレート制御を設定するには

1. [設定] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. [**TCP** プロファイルの設定] ページで、ドロップダウンリストから [**TCP** バースト制御] オプションを選択します。
   a) BurstRateCntrl
   b) CreditBytePrms
   c) RateBytePerms
   d) RateSchedulerQ
3. [**OK**]、[完了] の順にクリックします。

ラップされたシーケンス（**PAWS**）アルゴリズムに対する保護

デフォルトの TCP プロファイルで TCP タイムスタンプオプションを有効にすると、Citrix ADC アプライアンスはラップされたシーケンス（PAWS）に対する保護アルゴリズムを使用して、シーケンス番号が現在の TCP 接続の受信ウィンドウ内にある古いパケットを特定して拒否します（が最大値に達し、0 から再起動しました）。

ネットワークの輻輳によって SYN 以外のデータパケットが遅延し、パケットが到着する前に新しい接続を開くと、シーケンス番号の折り返しにより、新しい接続がパケットを有効として受け入れ、データが破損する可能性があります。ただし、TCP タイムスタンプオプションが有効の場合、パケットは破棄されます。

デフォルトでは、TCP タイムスタンプオプションは無効です。これを有効にすると、アプライアンスはパケットのヘッダーにある TCP タイムスタンプ（SEG.TSval）と最近のタイムスタンプ（Ts.recent）の値を比較します。SEG.TSval が Ts.recent 以上の場合は、パケットが処理されます。それ以外の場合、アプライアンスはパケットをドロップし、修正確認を送信します。

PAWS のしくみ

PAWS アルゴリズムは、同期接続のすべての着信 TCP パケットを次のように処理します。

1. SEG.TSval<Ts.recent:の場合着信パケットは受け入れられません。PAWS は（RFC-793 で指定されているように）確認応答を送信し、パケットをドロップします。注：ハーフオープン接続を検出して回復するための TCP のメカニズムを保持するには、ACK セグメントを送信する必要があります。
2. パケットがウィンドウ外にある場合：通常の TCP 処理と同様に、PAWS はパケットを拒否します。
3. SEG.TSval> の場合Ts.recent: PAWSパケットを受け入れ、それを処理します。
4. SEG.TSval < = Last.ACK.sentの場合（到着セグメントが満たす）：PAWS はSEG.TSval値をTs.recentにコピーする必要があります（Ts にコピーされますか）。db の最近のフィールド?。
5. パケットが順番にある場合：PAWS はパケットを受け入れます。
6. パケットが順番にない場合：パケットは、通常のウィンドウ内、アウトオブシーケンス TCP セグメントとして扱われます。たとえば、後で配信するためにキューに入れられる場合があります。
7. Ts.recent 値が 24 日を超えてアイドル状態の場合：PAWS タイムスタンプチェックが失敗した場合、Ts.recent の有効性がチェックされます。Ts.recent 値が無効であることが判明した場合、セグメントは受け入れられ、PAWS rule は Ts.recent を新しいセグメントの TSval 値で更新します。

コマンドラインインターフェイスを使用して **TCP** タイムスタンプを有効または無効にするには

コマンドプロンプトで入力します。

```
1 `set nstcpprofile nstcp_default_profile -TimeStamp (ENABLED | DISABLED)
    `
```

GUI を使用して TCP タイムスタンプを有効または無効にするには

[システム] > [プロファイル] > [**TCP** プロファイル] に移動し、デフォルトの TCP プロファイルを選択して [編集] をクリックし、[**TCP** タイムスタンプ] チェックボックスをオンまたはオフにします。

最適化テクニック

TCP は、フロー制御を最適化するために、次の最適化手法と方法を使用します。

ポリシーベースの **TCP** プロファイルの選択

今日のネットワークトラフィックは、かつてないほど多様で帯域幅が集中しています。トラフィックが増加すると、サービス品質（QoS）が TCP パフォーマンスに与える影響が大きくなります。QoS を強化するために、ネットワークトラフィックのクラスごとに異なる TCP プロファイルを使用して AppQoE ポリシーを構成できるようになりました。AppQoE ポリシーは、仮想サーバーのトラフィックを分類して、3G、4G、LAN、WAN などの特定のタイプのトラフィックに最適化された TCP プロファイルを関連付けます。

この機能を使用するには、TCP プロファイルごとにポリシーアクションを作成し、AppQoE ポリシーにアクションを関連付け、負荷分散仮想サーバーにポリシーをバインドします。

サブスクライバ属性を使用して TCP 最適化を実行する方法については、ポリシーベースの TCP プロファイルを参照してください。

ポリシーベースの **TCP** プロファイル選択の設定

ポリシーベースの TCP プロファイル選択の設定は、次の作業で構成されます。

- AppQoE を有効にします。TCP プロファイル機能を設定する前に、AppQoE 機能を有効にする必要があります。
- AppQoE アクションを追加しています。AppQoE 機能を有効にした後、TCP プロファイルを使用して AppQoE アクションを設定します。
- AppQoE ベースの TCP プロファイル選択の設定。異なるクラスのトラフィックに対して TCP プロファイル選択を実装するには、Citrix ADC が接続を区別し、正しい AppQoE アクションを各ポリシーにバインドできる AppQoE ポリシーを構成する必要があります。
- 仮想サーバーへの AppQoE ポリシーのバインド。AppQoE ポリシーを構成したら、1 つまたは複数の負荷分散、コンテンツスイッチング、またはキャッシュリダイレクト仮想サーバーにポリシーをバインドする必要があります。

コマンドラインインターフェイスを使用した構成

コマンドラインインターフェイスを使用して **AppQoE** を有効にするには

コマンドプロンプトで次のコマンドを入力して、機能を有効にし、有効になっていることを確認します。

- `enable ns feature appqoe`
- `show ns feature`

コマンドラインインターフェイスを使用して **AppQoE** アクションの作成中に **TCP** プロファイルをバインドするには

コマンドプロンプトで、`tcpprofiletobind` オプションを指定して次の AppQoE アクションコマンドを入力します。

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS | NS )
[<CustomFile>] [-altContentSvcName <string>] [-altContentPath <string>] [-
maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth <positive_integer
>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-
dosAction ( SimpleResponse |HICResponse )] [-tcpprofiletobind <string>]
show appqoe action
```

コマンドラインインターフェイスを使用して **AppQoE** ポリシーを構成するには

コマンドプロンプトで入力します。

```
add appqoe policy <name> -rule <expression> -action <string>
```

コマンドラインインターフェイスを使用して **AppQoE** ポリシーを負荷分散、キャッシュリダイレクト、コンテンツスイッチング仮想サーバーにバインドするには

コマンドプロンプトで入力します。

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <priority
>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <priority
>
```

例

```
1     add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
         ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500
          -slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
         sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
         ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack
         ENABLED -tcpmode ENDPOINT
2     add appqoe action appact1 -priority HIGH -tcpprofile tcp1
3     add appqoe policy apppol1 -rule ”client.ip.src.eq(10.102.71.31)” -
         action appact1
4     bind lb vserver lb2 -policyName apppol1 -priority 1 -
         gotoPriorityExpression END -type REQUEST
5     bind cs vserver cs1 -policyName apppol1 -priority 1 -
         gotoPriorityExpression END -type REQUEST
6  <!--NeedCopy-->
```


GUI を使用したポリシーベースの **TCP** プロファイルの設定

GUI を使用して AppQoE を有効にするには

1. [システム] > [設定**]** に移動します。
2. 詳細ペインで、[高度な機能の構成] をクリックします。
3. [高度な機能の構成] ダイアログボックスで、[**AppQoE**] チェックボックスをオンにします。
4. [**OK**] をクリックします。

GUI を使用して **AppQoE** ポリシーを構成するには

1.「アプリ エキスパート」>「**AppQoE**」**>**「アクション」に移動します。
2. 詳細ウィンドウで、次のいずれかの操作を行います。
3. 新しいアクションを作成するには、[追加] をクリックします。
4. 既存のアクションを変更するには、アクションを選択し、[編集] をクリックします。
5.「**AppQoE** アクションの作成」または「**AppQoE** アクションの設定」画面で、パラメーターの値を入力または選択します。ダイアログ・ボックスの内容は、「AppQoE アクションを構成するためのパラメータ」で説明されているパラメータに対応しています（アスタリスクは必須パラメータを示します）。
   a) 名前—name
   b) アクション・タイプ-respondWith
   c) プライオリティ—priority
   d) ポリシーキューの深さ：polqDepth
   e) キューの深さ：priqDepth
   f) DOS アクション：dosAction
6. [作成] をクリックします。

GUI を使用して **AppQoE** ポリシーをバインドするには

1. [トラフィック管理] > [負荷分散] > [仮想サーバー] に移動し、サーバーを選択して [編集] をクリックします。
2. [ポリシー] セクションで、[+] をクリックして AppQoE ポリシーをバインドします。
3. [ポリシー] スライダで、次の操作を行います。
   a) ドロップダウンリストから [AppQoE] としてポリシータイプを選択します。
   b) ドロップダウンリストからトラフィックタイプを選択します。
4. [ポリシーのバインド] セクションで、次の操作を行います。
   a) [新規] をクリックして、新しい AppQoE ポリシーを作成します。
   b) [既存のポリシー] をクリックして、ドロップダウンリストから AppQoE ポリシーを選択します。
5. バインドの優先順位を設定し、[仮想サーバにポリシーに バインド] をクリックします。
6. [完了] をクリックします。

SACK ブロック生成

1 つのデータウィンドウで複数のパケットが失われると、TCP のパフォーマンスが低下します。このようなシナリオでは、選択的確認応答（SACK）メカニズムと選択的繰り返し再送信ポリシーを組み合わせると、この制限が克服されます。着信順不同パケットごとに、SACK ブロックを生成する必要があります。

順序外パケットが再構成キューブロックに収まる場合は、ブロックにパケット情報を挿入し、完全なブロック情報を SACK-0 に設定します。アウトオブオーダーパケットが再構成ブロックに収まらない場合は、パケットを SACK-0 として送信し、前の SACK ブロックを繰り返します。順序外パケットが重複していて、パケット情報が SACK-0 に設定されている場合は、ブロックを D-SACK します。

注：パケットは、受信確認されたパケット、またはすでに受信された順序どおりのパケットである場合、D-SACK と見なされます。

クライアントのリネージング

Citrix ADC アプライアンスは、SACK ベースのリカバリ中にクライアントの再接続を処理できます。

PCB 上のエンドポイントをマーキングするためのメモリチェックは、使用可能な合計メモリを考慮していません

Citrix ADC アプライアンスでは、メモリ使用量のしきい値が、使用可能なメモリの合計を使用する代わりに 75% に設定されている場合、新しい TCP 接続は TCP 最適化をバイパスします。

SACK ブロックが欠落しているため、不必要な再送信

非エンドポイントモードでは、DUPACKS を送信するときに、順序どおりのパケットが少ない場合に SACK ブロックが欠落すると、サーバからの追加の再送信がトリガーされます。

過負荷のために最適化をバイパスした接続の **SNMP**

過負荷が原因で TCP 最適化をバイパスした接続数を追跡するために、Citrix ADC アプライアンスに次の SNMP ID が追加されました。

1. 1.3.6.1.4.1.5951.4.1.1.46.131 (tcpOptimizationEnabled). TCP 最適化で有効にされた接続の合計数を追跡します。
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed). 接続の総数を追跡するには、TCP 最適化をバイパスします。

動的受信バッファ

TCP パフォーマンスを最大化するために、Citrix ADC アプライアンスは TCP 受信バッファサイズを動的に調整できるようになりました。

テールロスプローブアルゴリズム

再送信タイムアウト（RTO）は、トランザクションの末尾でのセグメントの損失です。RTO は、特に短い Web トランザクションで、アプリケーションの待ち時間の問題がある場合に発生します。トランザクションの終了時にセグメントの損失を回復するために、TCP はテール損失プローブ（TLP）アルゴリズムを使用します。
TLP は送信者専用アルゴリズムです。TCP 接続が一定期間確認応答を受信しない場合、TLP は最後の確認応答されていないパケットを送信します（損失プローブ）。元の送信でテール損失が発生した場合、損失プローブからの確認応答によって SACK または FACK 回復がトリガーされます。

テールロスプローブの設定

Tail Loss Probe（TLP）アルゴリズムを使用するには、TCP プロファイルで TLP オプションを有効にし、パラメータにそのプロファイルのセキュリティ要件に適合する値を設定する必要があります。

コマンドラインを使用して **TLP** を有効にする

コマンドプロンプトで、次のコマンドのいずれかを入力して、新規または既存のプロファイルで TLP を有効または無効にします。

> 注:
>
> デフォルト値は DISABLED です。

```
add tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED
set tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED
unset tcpprofile <TCP Profile Name> - taillossprobe
```

例:

```
add tcpprofile nstcp_default_profile - taillossprobe
set tcpprofile nstcp_default_profile -taillossprobe Enabled
unset tcpprofile nstcp_default_profile -taillossprobe
```

Citrix ADC GUI を使用してテールロスプローブアルゴリズムを構成します

1. [設定] > [システム] > [プロファイル] に移動し、[編集] をクリックして TCP プロファイルを変更します。
2. [**TCP** プロファイルの構成] ページで、[テールロスプローブ**]** チェックボックスをオンにします。
3. [**OK]**、[完了**]** の順にクリックします。

Citrix ADC のトラブルシューティングソリューション

October 7, 2021

このトピックでは、アプライアンスで発生する問題を解決するために必要な基本的なトラブルシューティングソリューションについて説明します。NetScaler アプライアンスと、NetScaler アプライアンスがネットワークとどのように統合されるか、基本的なシステム機能にどのような問題があるかを理解できます。

Citrix ADC でパケットトレースを記録する方法

June 24, 2022

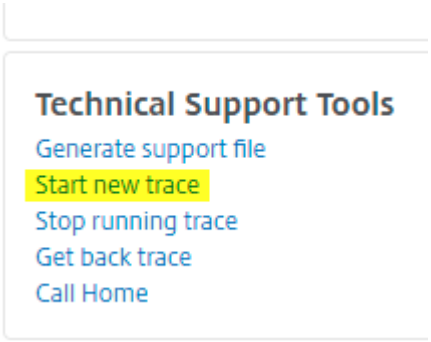
このトラブルシューティング記事では、管理者が Citrix ADC GUI を使用してネットワークパケットトレースを記録する方法について説明します。

確認事項

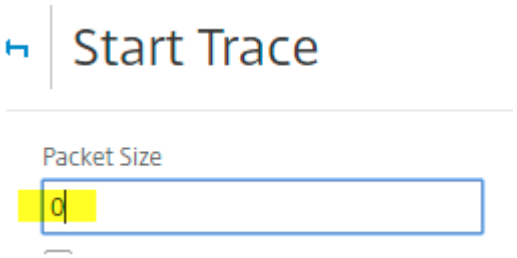
- Citrix では、次の Web ページにある「自動ビルドセクション」から最新の Wireshark バージョンを使用することをお勧めします: http://www.wireshark.org/download/automated。
- Citrix ADC バージョン 11.1 以降では、キャプチャを復号化して ECC（楕円曲線暗号化）を確認するために、セッションの再利用と DH パラメーターが仮想サーバーから無効になっています。トレースをキャプチャする前に行う必要があります。

NetScaler バージョン 11.1 でパケットトレースを記録する

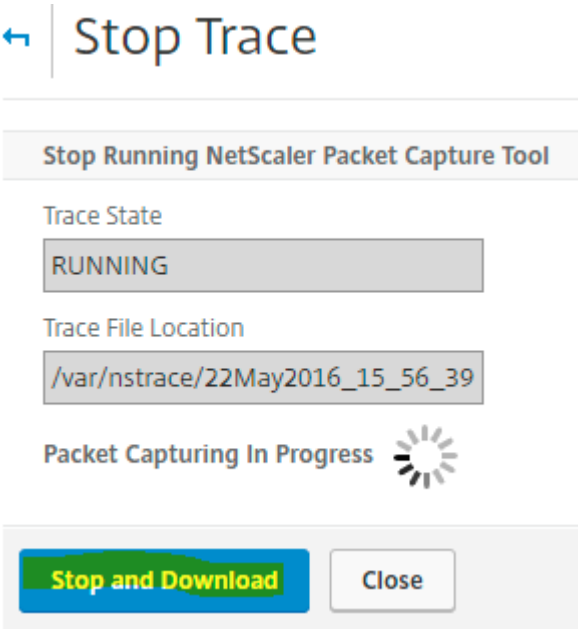
1. [システム] > [診断] ページに移動します。
2. 次のスクリーンショットに示すように、[診断] ページの [新しいトレースの開始] リンクをクリックします。



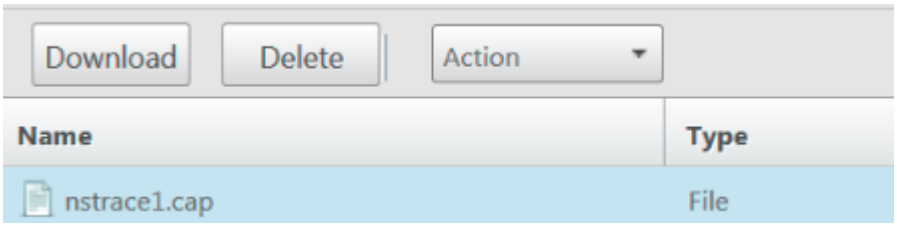
3. [パケットサイズ] フィールドでパケットサイズを **0** に更新します。



4. [開始] をクリックして、ネットワークパケットトレースの記録を開始します。
5. テストが完了したら、[停止してダウンロード] をクリックして、ネットワークパケットトレースの記録を停止します。

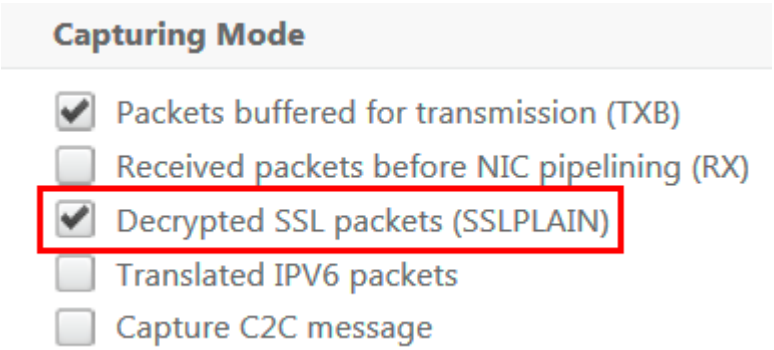


6. 必要なファイルを選択して [選択] をクリックし、[ダウンロード] をクリックします。



7. Wireshark ユーティリティでネットワークパケットトレースファイルを開き、ファイルの内容を表示します。

> 注: 秘密キーなしでパケットトレースを復号化するには、[復号化された SSL パケット (SSLPLAIN)] を選択します。



SSL マスターキーのキャプチャ

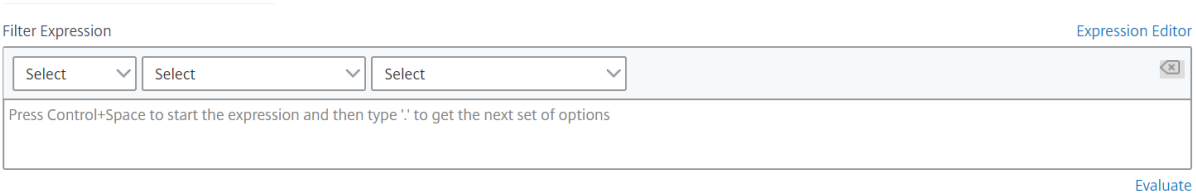
11.0、11.1 以降のバージョンには、その特定のセッション/nstrace に対してのみ有効なセッションキーをキャプチャするオプションがあります。このオプションは、秘密キーを共有しない場合や、SSLPLAIN モードを使用する場合に使用できます。詳しくは、https://support.citrix.com/article/CTX135889を参照してください。

秘密キーを共有せずにセッションキーをエクスポートする

ほとんどのシナリオでは、秘密鍵は利用できないか、共有されていません。このようなシナリオでは、秘密鍵の代わりに **SSL** セッション鍵をエクスポートすることを提案できます。
「SSL 秘密キーを共有せずに SSL セッションキーをエクスポートおよび使用して SSL トレースを復号する方法」
https://support.citrix.com/article/CTX135889を参照してください。

フィルター

また、トレース中は IP ベースのフィルタを追加することを常に推奨します。このプロセスにより、関心のあるトラフィックだけが確実にキャプチャされ、トラブルシューティングが容易になります。フィルターを追加すると、トレース中のアプライアンスの負荷も軽減されます。

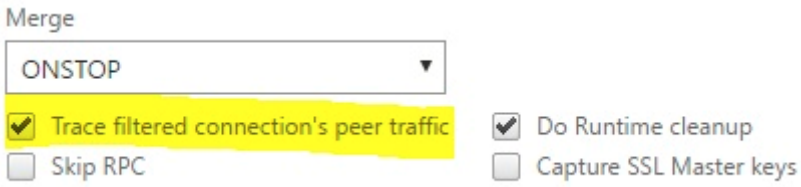


適切なキャプチャを取得するには、単純な IP ベースのフィルターで十分です。`nstrace`フィルターと例の詳細については、Citrix のドキュメントページを参照してください 。

仮想サーバー **IP** フィルタ（フロントエンドとバックエンドの両方）でパケットトレースをキャプチャするユースケース

仮想サーバの IP アドレスのフィルタを使用し、CLI で「—link」オプションを有効にするか、GUI で [フィルタリングされた接続ピアトラフィックをトレース] オプション (10.1 以降で利用可能) を選択すると、IP アドレスのフロントエンドトラフィックとバックエンドトラフィックの両方をキャプチャできます。

```
start nstrace -size 0 -filter "CONNECTION.IP.EQ(1.1.1.1)" -link ENABLED

show nstrace
        State:  RUNNING          Scope:  LOCAL            TraceLocation
           :  "/var/nstrace/24Mar2017_16_00_19/..." Nf:  24
                           Time:  3600             Size:  0
                          Mode:  TXB NEW_RX
        Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  24
           Mar2017_16_00_19 Filter:  "CONNECTION.IP.EQ(1.1.1.1)" Link:
            ENABLED           Merge:  ONSTOP           Doruntimecleanup
           :  ENABLED
        TraceBuffers:  5000      SkipRPC:  DISABLED       Capsslkeys:
           DISABLED    InMemoryTrace:  DISABLED
<!--NeedCopy-->
```



周期的なトレースのキャプチャ

断続的な問題のトラブルシューティングは常に困難です。周期的なトレースは、断続的な問題に最適です。トレースは、問題が発生するまでに数時間または数日にわたって実行できます。また、特定のフィルタを使用して、長期間実行する前に生成されるトレースファイルのサイズを評価することもできます。

CLI から以下のコマンドを実行します。

```
start nstrace -nf 60 -time 30 -size 0
This particular trace will create 60 files each of them for 30 sec.
   This means the files will start getting overwritten after 60 trace
   files or 30 mins
Show nstrace à To check the status of the nstrace
```



```
Stop nstrace à To stop the nstrace.

<!--NeedCopy-->
```

ベストプラクティス

1 秒あたり GB のトラフィックを処理するユニットでは、トラフィックのキャプチャは非常にリソースを大量に消費するプロセスです。リソースへの影響は、主に CPU とディスク容量の点にあります。ディスク容量への影響は、フィルタリング式を使用することで軽減できます。ただし、アプライアンスはパケットをキャプチャする前にフィルタに従ってパケットを処理する必要があるため、CPU への影響は残り、わずかに増加することがあります。

トレースに関するベストプラクティスは次のとおりです。

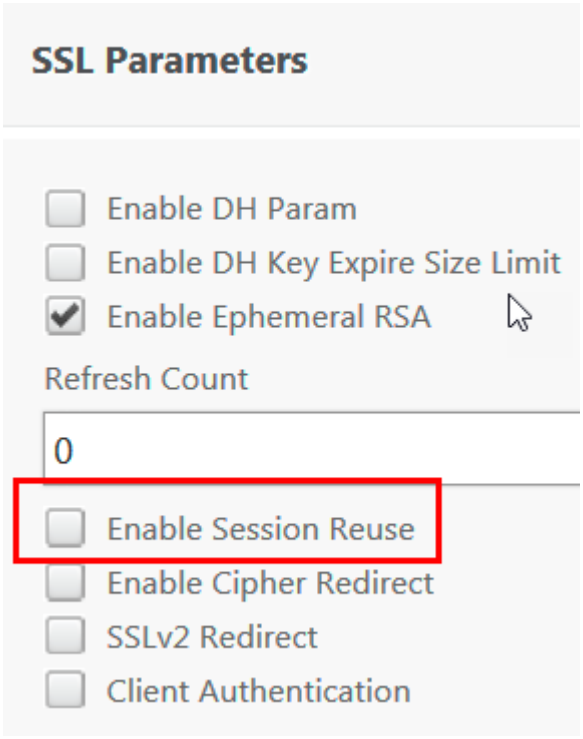
1. 対象のパケットが確実にキャプチャされる場合は、トレースを実行する期間をできるだけ制限する必要があります。
2. 営業時間外など、ユーザー数 (したがってトラフィック) が大幅に減少したときにトレースアクティビティが発生するようにスケジュールします。

その他のリソース

GUI から仮想サーバーでのセッション再利用を無効にする

トレースをキャプチャしてトレースで SSL ハンドシェイクを完了すると、セッションの再利用は無効になります。有効にすると、トレースで部分的なハンドシェイクをキャプチャできます。トレース収集後に必ずこのオプションを有効にしてください。
永続化メソッドが sslsession の場合、SSL セッションの再利用を無効にしないでください。既存の接続の永続性が損なわれるためです。詳細については、https://support.citrix.com/article/CTX121925を参照してください。

1. 仮想サーバーを開き、[SSL パラメータ] に移動します。
2. セッション再利用を有効にするが有効な場合は無効にします。



CLI から仮想サーバーでのセッション再利用を無効にする

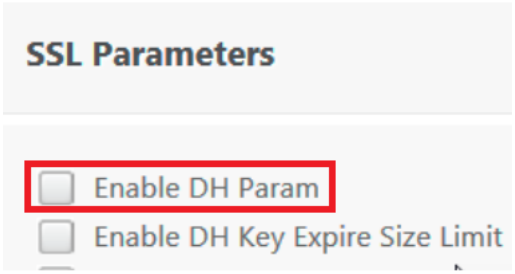
1. アプライアンスコンソールに SSH 接続します。
2. 次のコマンドを実行して、仮想サーバーから DH Param を無効にします。

```
set ssl vserver ”vServer_Name”-sessReuse DISABLED
```

GUI から仮想サーバの **DH** パラメータを無効にする

DH パラメータについて理解するにはhttps://support.citrix.com/article/CTX213335を参照してください。

1. 仮想サーバーを開き、[SSL パラメータ] に移動します。
2. DH Param が有効な場合は無効にします。



CLI から仮想サーバの **DH** パラメータを無効にする

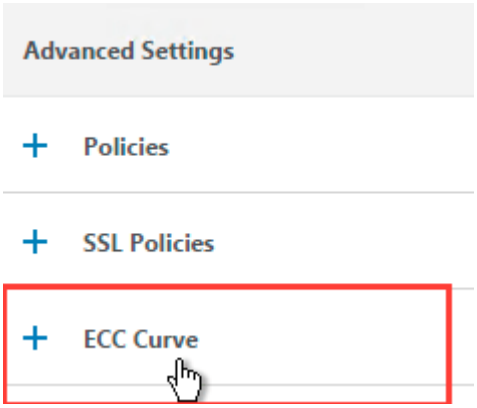
1. アプライアンスコンソールに SSH 接続します。
2. 次のコマンドを実行して、仮想サーバーから DH Param を無効にします。

```
set ssl vserver ”vServer_Name”-dh DISABLED
```

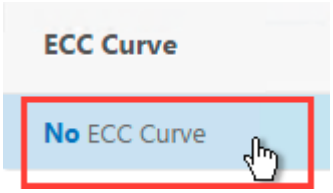
GUI から仮想サーバ上の **ECC** カーブを無効にする

キャプチャされた SSL トレースを秘密鍵で復号化するために、ECC カーブが無効になっています。関連する SSL 暗号が使用されている場合は、キーを無効にしないでください。ECC カーブの詳細については、https://support.citrix.com/article/CTX205289を参照してください。

1. 仮想サーバを開き、[ECC Curve] に移動します。



2. 仮想サーバにバインドされた ECC Curve がない場合、他のアクションは必要ありません。



3. ECC カーブが仮想サーバにバインドされている場合は、ECC カーブをクリックして仮想サーバからバインド解除します。

CLI から仮想サーバ上の **ECC** カーブを無効にする

1. アプライアンスコンソールに SSH 接続します。
2. 仮想サーバにバインドされている ECC Curve ごとに、次のコマンドを実行します。

```
unbind ssl vserver ”vServer_Name”-eccCurveName ”ECC_Curve_Name”
```


Citrix ADC アプライアンスの問題をログに記録するための VAR ディレクトリ上の領域を解放する方法

October 7, 2021

次の記事では、管理者が Citrix /var ADC アプライアンスのディレクトリから領域を解放する方法について説明します。Citrix GUI にアクセスできない場合は、手順に従うことができます。

アプライアンスの /var ディレクトリのディスク容量が少ない場合は、Citrix GUI にサインインできないことがあります。このシナリオでは、古いログファイルを削除して /var ディレクトリに空き領域を作成できます。

確認事項

- ファイルをアプライアンスから削除する前に、必ずファイルをバックアップしてください。

Citrix ADC アプライアンスの/varディレクトリ内の領域を解放するには、次の手順を実行します。

1. SSH を使用して Citrix ADC の CLI にログオンします。このタスクを完了する方法の詳細については、Citrix ADC ドキュメントを参照してください。
2. Citrix ADC CLI にログオンした後、次のコマンドを使用してシェルプロンプトに切り替えます。`shell`
3. 次のコマンドを実行して、Citrix ADC アプライアンスの空き容量を確認します。`df -h`
4. /varディレクトリのメモリ容量が最大 90% 満たされている場合は、このディレクトリからファイルを削除する必要があります。
   - 次のコマンドを実行して、/var ディレクトリの内容を表示します。
     ```
     cd /var
     ls -l
     ```

   通常、目的のディレクトリは次のとおりです。

```
/var/nstrace - This directory contains trace files.This is the
   most common reason for HDD being filled on the Citrix ADC
   appliance. This is due to an nstrace being left running for
   indefinite amount of time. All traces that are not of interest
   can and should be deleted. To stop an nstrace, go back to the
   CLI and issue stop nstrace command.

/var/log - This directory contains system specific log files.

/var/nslog - This directory contains Citrix ADC log files.

```



```
/var/tmp/support - This directory contains technical support files, also known as, support bundles. All files not of interest should be deleted.

/var/core - Core dumps are stored in this directory. There will be directories within this directory and they will be labeled with numbers starting with 1. These files can be quite large in size. Clear all files unless the core dumps are recent and investigation is required.

/var/crash - Crash files, such as process crashes are stored in this directory. Clear all files unless the crashes are recent and investigation is required.

/var/nsinstall -  Firmware is placed in this directory when upgrading. Clear all files, except the firmware that is currently being used.
```

- ディレクトリのいずれかがより多くのスペースを使用しているかどうかを確認します。

```
du -hs *
44k     cache
2.0k    clusterd
2.0k    configdb
6.0k    core
989M    crash
4.0k    cron
2.0k    dev
6.0k    download
2.0k    gui
2.0k    install
2.0k    krb
2.0k    learnt_data
122M    log
366M    NetScaler
14k     ns_gui
86k     ns_sys_backup
631M    nsinstall
883M    nslog
32k     nsproflog
2.0k    nssynclog
16k     nstemplates
36k     nstmp
```



```
4.5G    nstrace
8.1M    opt
6.0k    pubkey
52k     run
28M     safenet
72M     tmp
2.0k    vmtools
14k     vpn
```

- 不要なファイルを削除します。

```
rm -r nstrace/*
```

For more help on deleting files see FreeBSD Man Pages.

- Delete the files which are not required.
  `rm -r nstrace/*`
  For more help on deleting files see FreeBSD Man Pages.
- If the log or `nslog` directory is using more space, then run the following commands to open the log directory and view its contents:

```
cd /var/log
ls -l
cd /var/nslog
ls -l
```

1. すべてのファイルが圧縮されていることを確認します。これは、.tar.gz ファイル名の拡張子で示されます。
2. Citrix ADM または Command Center を使用している場合は、/var/ns_system_backup ディレクトリを確認します。Citrix ADM または Command Center が作成したバックアップファイルがクリアされていることを確認します。

その他のリソース

前の手順で説明したコマンドの詳細については、http://ss64.com/bash/を参照してください。

Citrix ADC アプライアンスからコアファイルまたはクラッシュしたファイルをダウンロードする方法

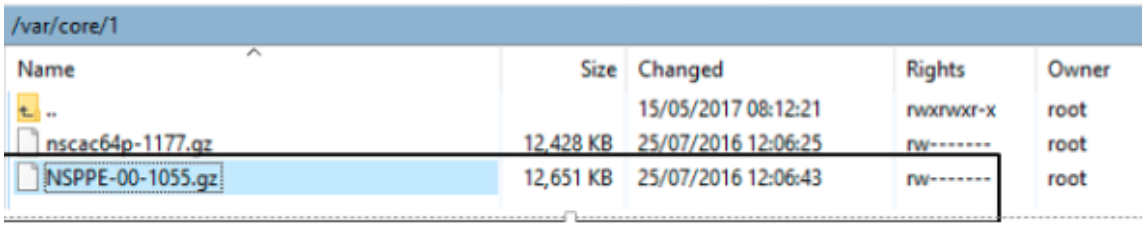
October 7, 2021

このトラブルシューティングの記事では、管理者が Citrix ADC アプライアンスからコアファイルやクラッシュファイルをダウンロードする方法について説明します。

SFTP クライアントを使用して Citrix ADC アプライアンスからコアファイルまたはクラッシュファイルをダウンロードする

NetScaler アプライアンスからコアファイルまたはクラッシュファイルをダウンロードするには、次の手順に従います。

1. WinSCP を開き、NetScaler 管理 IP アドレスにログオンします。
2. /var/core/1に移動し、ファイルをダウンロードします。



> 注:
>
> 最新のクラッシュファイルまたはコアファイルをダウンロードするには、コマンドインターフェイスから WinSCP ツールを使用することもできます。ファイルは、コアディレクトリまたはクラッシュディレクトリのいずれかに配置できます。

パフォーマンス統計とイベントログを収集する方法

October 7, 2021

/var/nslogディレクトリにあるアーカイブnewnslogファイルから、仮想サーバと関連サービスのパフォーマンス統計を収集できます。newnslog ファイルは、/netscaler/nsconmsgを実行して解釈されます。

CLI を使用してパフォーマンス統計とイベントログを収集する

Citrix ADC シェルプロンプトから nsconmsg コマンドを実行して、イベントを報告できます。

コマンドプロンプトで入力します。


```
/netscaler/nsconmsg -K /var/nslog/newnslog -d event
```

```
1 Displaying event information
2 NetScaler V20 Performance Data
3 NetScaler NS10.5: Build 57.7.nc, Date: May 14 2015, 07:35:21
4 rtime: Relative time between two records in milliseconds
5 seqno rtime event-message                          event-time
6 11648 16310 PPE-0 MonServiceBinding_10.104.20.110:443_(tcp-default)
7 <!--NeedCopy-->
```

特定の”**newnslog**“ファイルの対象期間を表示する

コマンドプロンプトで入力します。

```
/netscaler/nsconmsg -K /var/nslog/newnslog -d setime
```

現在のデータは /var/nslog/newnslog ファイルに追加されます。NetScaler は、デフォルトで 2 日ごとに newnslog ファイルを自動的にアーカイブします。アーカイブされたデータを読み取るには、次の例に示すようにアーカイブを抽出する必要があります。

-NetScaler シェルプロンプトから特定のディレクトリに移動するコマンド。`cd /var/nslog`

`tar xvfz newnslog.100.tar.gz` -tar ファイルを抽出するコマンド。

`/netscaler/nsconmsg -K newnslog.100 -d setime` -この例newnslog.100では、特定のファイルでカバーされている時間スパンをチェックするコマンド。

`ls -l` コマンドは、すべてのログファイルとそれらのファイルに関連付けられたタイムスタンプをチェックします。

```
root@NETSCALER## cd /var/nslog
root@NETSCALER## ls -l
```

```
1  wheel    461544 Aug  7  2014 newnslog.1.tar.gz
2 -rw-r--r--   1 root       wheel    191067 Aug  7  2014 newnslog.10.tar.
     gz
3 -rw-r--r--   1 root       wheel  11144873 Apr 26 22:04 newnslog.100.tar
     .gz
4 -rw-r--r--   1 root       wheel  11095053 Apr 28 22:04 newnslog.101.tar
     .gz
5 -rw-r--r--   1 root       wheel  11114284 Apr 30 22:04 newnslog.102.tar
     .gz
6 -rw-r--r--   1 root       wheel  11146418 May  2 22:04 newnslog.103.tar
     .gz
```



```
-rw-r--r--   1 root       wheel  11104227 May  4 22:04 newnslog.104.tar
    .gz
-rw-r--r--   1 root       wheel  11297419 May  6 22:04 newnslog.105.tar
    .gz
-rw-r--r--   1 root       wheel  11081212 May  8 22:04 newnslog.106.tar
    .gz
-rw-r--r--   1 root       wheel  11048542 May 10 22:04 newnslog.107.tar
    .gz
-rw-r--r--   1 root       wheel  11101869 May 12 22:04 newnslog.108.tar
    .gz
-rw-r--r--   1 root       wheel  11378787 May 14 22:04 newnslog.109.tar
    .gz
-rw-r--r--   1 root       wheel  44989298 Apr 11  2014 newnslog.11.gz
<!--NeedCopy-->
```

ファイル内の期間を表示する

次の例に示すように、nsconmsg コマンドを使用して、指定されたファイル内の期間のみを表示します。

```
/netscaler/nsconmsg -K /var/nslog/newnslog -s time=22Mar2007:20:00 -T 7 -s
ConLb=2 -d oldconmsg
```

各項目の意味は次のとおりです。

s - time=22Mar2007:20:00:00 is start at March 22, 2007 at exactly 20:00.

T 7 -7 秒のデータを表示します

s -負荷分散統計の詳細レベルを表示します。

d -統計情報を表示します。

> 注:
>
> ADC リリース 12.1 から、” 時間 “ の秒も追加する必要があります。例: 22Mar2007:20:00:00

-d oldconmsgパラメータによって提供される統計情報は、7 秒ごとに記録されます。次に、出力例を示します。

```
VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec) Mbps(1.02)
    Pers(OFF) Err(0)
Pkt(186/sec, 610 bytes) actSvc(4) DefPol(NONE) override(0)
Conn: Clt(253, 1/sec, OE[252]) Svr(3)
S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5)
    Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms)
Other: Pkt(36/sec, 712 bytes) Wt(10000) RHits(31555)
Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) RP(11) SQ(0)
```



```
S(10.128.49.39:80:UP) Hits(9731048, 4/sec, P[2929279, 0/sec]) ATr(9)
    Mbps(0.27) BWlmt(0 kbits) RspTime(161.69 ms)
Other: Pkt(41/sec, 756 bytes) Wt(10000) RHits(31555)
Conn: CSvr(32, 0/sec) MCSvr(19) OE(13) RP(4) SQ(0)
S(10.128.49.38:80:UP) Hits(9341366, 5/sec, P[2700778, 0/sec]) ATr(4)
    Mbps(0.27) BWlmt(0 kbits) RspTime(120.50 ms)
Other: Pkt(42/sec, 720 bytes) Wt(10000) RHits(31556)
Conn: CSvr(37, 0/sec) MCSvr(19) OE(13) RP(9) SQ(0)
S(10.128.49.37:80:UP) Hits(9685018, 4/sec, P[2844418, 0/sec]) ATr(3)
    Mbps(0.23) BWlmt(0 kbits) RspTime(125.38 ms)
Other: Pkt(38/sec, 670 bytes) Wt(10000) RHits(31556)
Conn: CSvr(32, 0/sec) MCSvr(20) OE(10) RP(7) SQ(0)
<!--NeedCopy-->
```

> 注:
>
> 個々のサービスのクライアント接続数は、仮想サーバーのクライアント接続数には加算されません。これは、Citrix ADC アプライアンスとバックエンドサービス間のセッションの再利用が原因です。

仮想サーバーの出力

```
VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec)Mbps(1.02)Pers(
OFF)Err(0)Pkt(186/sec, 610 bytes)actSvc(4)DefPol(NONE)override(0)Conn: Clt
(253, 1/sec, OE[252])Svr(3)
```

次のリストは、仮想サーバーの統計について説明しています。

1. IP (IP address:port:state:Load balancing method)。構成された仮想 IP アドレスの IP アドレスとポート。仮想サーバの状態または仮想 IP アドレスが、UP、DOWN、または OUT OF SERVICE 。負荷分散メソッドが仮想 IP アドレス）に設定されている
2. Hits (##)。仮想サーバーに到達した要求の数。
3. Mbps (##)。仮想サーバー（Rx + Tx）の総トラフィック量を Mbits/s に変換
4. Pers: 構成されている永続性のタイプ。
5. Err (##)。仮想サーバーによってエラーページが生成された回数。
6. Pkt (##/sec, ## bytes): 仮想サーバーを通過するネットワークトラフィックの量（パケットとして）と仮想サーバーを通過する平均パケットサイズ。
7. actSvc(##)。仮想サーバーにバインドされているアクティブなサービスの数。
8. DefPol (RR)。デフォルトの負荷分散方法がアクティブかどうかを示します。デフォルトの負荷分散方法は、他の方法の動作をスムーズにするために、いくつかの初期要求に使用されます。
9. Clt (##, ##/sec)。仮想サーバーへの現在のクライアント接続の数レート。
10. OE [##]。オープン確立状態の仮想サーバーからのサーバー接続の数。
11. Svr (##)。仮想サーバーからの現在のサーバー接続の数。

上記の出力で、Svr(3) は、コマンドが統計サンプルを収集することを示します。合計 4 つのサービスがあります

が、仮想サーバーからバックエンドサーバーへのアクティブな接続は 3 つあります。クライアントが仮想サーバーとの接続を確立するとき、コマンドが情報を収集するときにクライアントがトラフィックを送受信する必要はありません。したがって、OE[]数よりも小さいSvrカウンターの表示が一般的です。Svrカウンターは、データをアクティブに送受信しているアクティブな接続の数を表します。マップされた IP アドレス（MIP）またはサブネット IP アドレス（SNIP）は、関連付けられたバックエンドサーバーに接続されます。また、Citrix ADC は、バックエンドサーバーに接続されている仮想サーバーを追跡し、カウンターを計算します。

仮想サービスの出力

```
S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5)
    Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms)
Other: Pkt(36/sec, 712 bytes) Wt(10000) RHits(31555)
Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) RP(11) SQ(0)
<!--NeedCopy-->
```

次のリストは、サービス統計について説明しています。

1. S （IP address:port:state)。IP アドレス、ポート、およびサービスの状態（DOWN、UP、OUT OF SERVICE など)。
2. Hits （##, P[##])。サービスに向けられた要求の数、構成されたサーバーの永続性のためにサービスに向けられた要求の数。
3. ATr （##)。サービスへのアクティブな接続の数。

> 注:
>
> アクティブな接続とは、サービスへの未処理の要求があるか、現在トラフィックアクティビティがある接続です。

1. Mbps （##.####)。サービス（Rx + Tx）の総トラフィック量を Mbits/s に変換
2. BWlmt （## kbits): 定義された帯域幅制限。
3. RspTime （## ms)。サービスの平均応答時間（ミリ秒単位)。
4. Pkt(##/sec, ##bytes)。サービスに向かう 1 秒あたりのパケット数で表したトラフィック量。パケットの平均サイズ。
5. Wt （##)。負荷分散アルゴリズムで使用される重みインデックス。

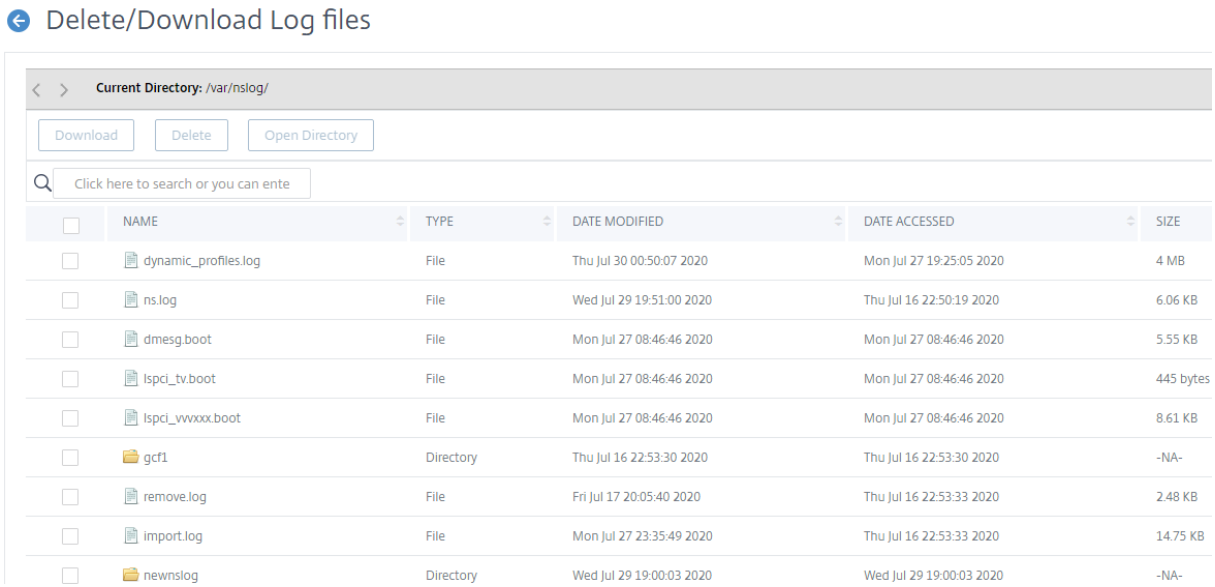
> 注:
>
> この値を 10,000 で割ると、サービスの実際に構成された重みが得られます。

1. RHits （##)。ラウンドロビン負荷分散アルゴリズムで使用される実行中の要求カウンター。
2. CSvr （##, ##/sec)。サービスレートへの接続数。
3. MCSvr （##)。サービスへの接続の最大数。
4. OE （##)。確立された状態のサービスへの接続数。
5. RP （##)。再利用プールに存在するサービスへの接続数。

6. SQ （##）。サージキューで待機しているサービスへの接続数。

Citrix ADCGUI を使用してパフォーマンス統計とイベントログを収集する

1. **System > Diagnostics > Maintenance > Delete/Download log files** に移動します。
2. ファイルを選択し、[ダウンロード] をクリックしてファイルをダウンロードします。



ログファイルのローテーションを構成する方法

October 7, 2021

Citrix ADC アプライアンスは、複数のディレクトリにさまざまな形式でログを生成します。これらのログの一部はデフォルトではローテーションされておらず、サイズが大きくなりすぎてディスク領域を消費する可能性があります。付属のログローテーションユーティリティ（newsyslog）を使用すると、関連情報のみを保持して管理と管理を容易にすることで、これらのログを一貫して管理できます。

Citrix ADC ファームウェアに含まれている newsyslog ユーティリティは、ログファイルをアーカイブし、システムログをローテーションして、ローテーション中に現在のログが空になるようにします。システム crontab は、このユーティリティを 1 時間ごとに実行し、ローテーションするファイルと条件を指定する構成ファイルを読み取ります。アーカイブされたファイルは、必要に応じて圧縮される場合があります。

既存の設定は/etc/newsyslog.confにあります。ただし、このファイルはメモリファイルシステムにあるため、管理者が変更を/nsconfig/newsyslog.confに保存して、構成が NetScaler の再起動後も維持する必要があります。

このファイルに含まれるエントリの形式は次のとおりです。

```
logfilename [owner:group] mode count size when flags [/pid_file] [sig_num]
```


> 注:
>
> 角括弧内のフィールドはオプションであり、省略できます。

ファイルの各行は、ログファイルとローテーションが発生する必要がある条件を表します。

一例では、size フィールドが示すことの大きさ ns.log 100 としてキロバイト。count フィールドは、アーカイブされた ns.log ファイルの数が 25 であることを示します。100K のサイズと 25 のカウントがデフォルトのサイズとカウント値です。

> 注:
>
> フィールドがアスタリスクで構成されている場合（* ）、ns.log ファイルが時間に基づいてローテーションされないことを意味します。crontab ジョブは 1 時間ごとに、ns.log のサイズがこのファイルで構成されているサイズ以上であるかどうかをチェックする newsyslog ユーティリティを実行します。この例では、100 K 以上の場合、そのファイルをローテーションします。

```
root@ns# cat /etc/newsyslog.conf
# Netscaler newsyslog.conf

# This file is present in the memory filesystem by default, and any
    changes
# to this file will be lost following a reboot. If changes to this file
# require persistence between reboots, copy this file to the /nsconfig
# directory and make the required changes to that file.
#
# logfilename [owner:group] mode count size when flags [/pid_file] [
    sig_num]
/var/log/cron 600 3 100 * Z
/var/log/amd.log 644 7 100 * Z
/var/log/auth.log 600 7 100 * Z
/var/log/ns.log 600 25 100 * Z
<!--NeedCopy-->
```

size フィールドを変更して ns.log ファイルの最小サイズを変更したり、フィールドを変更して ns.log ファイルを特定の時間に基づいて回転させたりすることができます。

毎日、毎週、または毎月の仕様は、それぞれ、[Dhh]および[Dhh [Mdd]]として与えられます。オプションの時刻フィールドは、デフォルトで深夜に設定されています。これらの仕様の範囲と意味は次のとおりです。

```
Hh hours, range 0 ... 23
w day of week, range 0 ... 6, 0 = Sunday
dd day of month, range 1 ... 31, or the letter L or l to specify the
    last day of the month.
```



```
<!--NeedCopy-->
```

例:

デフォルトでローテーションされるログの説明を含むいくつかの例を次に示します。

```
/var/log/auth.log 600 7 100 * Z
```

ファイルが 100K に達すると、認証ログがローテーションされ、auth.log の最後の 7 つのコピーがアーカイブされ、gzip（Z フラグ）で圧縮され、結果のアーカイブに次のアクセス許可が割り当てられます–rw ——- 。

```
/var/log/all.log 600 7 * @T00 Z
```

キャッチオールログは毎晩深夜に 7 回ローテーションされます (@T00) そして gzip で圧縮されます。結果のアーカイブには、次のアクセス許可–rw-r ——が割り当てられます。

```
/var/log/weekly.log 640 5 * $W6D0 Z
```

毎週のログは、毎週月曜日の深夜に 5 回ローテーションされます。結果のアーカイブには、アクセス許可が割り当てられます。

一般的な回転パターン:

- D0。毎晩深夜にローテーション
- D23。毎日ローテーション 23:00
- W0D23。毎週日曜日にローテーション 23:00
- W5。毎週金曜日の深夜にローテーション
- MLD6。毎月最終日にローテーションする 6:00
- M5。月の 5 日ごとの深夜にローテーションします

間隔と時間の指定が両方とも指定されている場合は、両方の条件が満たされている必要があります。つまり、ファイルは指定された間隔と同じかそれより古い必要があり、現在の時刻は時刻の指定と一致する必要があります。

最小ファイルサイズを制御できますが、newsyslog ユーティリティが次の 1 時間のスロットで順番を取得するまでのファイルサイズに制限はありません。

newsyslog をデバッグします。

newsyslogユーティリティの動作をデバッグするには、verbose フラグを追加します。

```
root@dj_ns# newsyslog -v
/var/log/cron <3Z>: size (Kb): 31 [100] --> skipping
/var/log/amd.log <7Z>: does not exist, skipped.
/var/log/auth.log <7Z>: size (Kb): 2 [100] --> skipping
/var/log/kerberos.log <7Z>: does not exist, skipped.
/var/log/lpd-errs <7Z>: size (Kb): 0 [100] --> skipping
/var/log/maillog <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/sendmail.st <10>: age (hr): 0 [168] --> skipping
```



```
/var/log/messages <5Z>: size (Kb): 7 [100] --> skipping
/var/log/all.log <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/slip.log <3Z>: size (Kb): 0 [100] --> skipping
/var/log/ppp.log <3Z>: does not exist, skipped.
/var/log/security <10Z>: size (Kb): 0 [100] --> skipping
/var/log/wtmp <3>: --> will trim at Wed Apr 1 04:00:00 2009
/var/log/daily.log <7Z>: does not exist, skipped.
/var/log/weekly.log <5Z>: does not exist, skipped.
/var/log/monthly.log <12Z>: does not exist, skipped.
/var/log/console.log <5Z>: does not exist, skipped.
/var/log/ns.log <5Z>: size (Kb): 18 [100] --> skipping
/var/log/nsvpn.log <5Z>: size (Kb): 0 [100] --> skipping
/var/log/httperror.log <5Z>: size (Kb): 1 [100] --> skipping
/var/log/httpaccess.log <5Z>: size (Kb): 1 [100] --> skipping
root@dj_ns#
<!--NeedCopy-->
```

Citrix ADC アプライアンスの /flash ディレクトリ上のスペースを解放する方法

October 7, 2021

このトラブルシューティングの記事では、管理者が Citrix ADC アプライアンスからコアファイルやクラッシュファイルをダウンロードする方法について説明します。

Citrix ADC アプライアンスの/flash ディレクトリのスペースを解放する手順

1. SSH を使用して Citrix ADC の CLI にログオンします。
2. Citrix ADC CLI にログオンした後、次のコマンドを使用してシェルプロンプトに切り替えます。`shell`。
3. `df -h`コマンドを実行して、Citrix ADC アプライアンスの空き容量を確認します。
4. /flash ディレクトリの容量が 90% 未満または非常に低い場合は、このディレクトリからファイルを削除する必要があります。
5. 次のコマンドを実行して、/flash ディレクトリの内容を表示します。

```
cd /flash
ls -l
```


6. NetScaler ソフトウェアリリースのさまざまなバージョンの複数のファイルが見つかる場合があります。この場所にあるファイルが、アプライアンスの現在のバージョンの NetScaler ソフトウェアに適用できるファイルであることを確認してください。次のコマンドを実行して、アプライアンスから他のファイルを削除します。

```
rm <filename>
```

> 注
>
> 古いバージョンのカーネルのみを削除します。/flash ディレクトリには、NetScaler ソフトウェアリリースの現在のバージョンまたはビルドが使用しているファイルと kernel.gz ファイルが含まれている必要があります。これらのファイルをから削除しないことをお勧めします /flash ディレクトリ。

参考資料

December 7, 2021

このリファレンス情報を使用して、次の Citrix ADC コンポーネントについて詳しく理解してください。

Citrix ADC SNMP OID -Citrix ADC アプライアンスから情報を取得するために使用できる SNMP OID の詳細。

Citrix ADC Syslog メッセージ -Citrix ADC アプライアンスによって提供される Syslog メッセージの詳細。

Citrix ADC CLI コマンド -CLI を使用して Citrix ADC アプライアンスを構成するために使用できるコマンドの詳細。`man <ns-command-name>`コマンドを入力して、CLI で各コマンドの詳細を表示することもできます。

Citrix NITRO API リファレンス -REST API を使用して Citrix ADC アプライアンスで実行できるすべての操作の詳細。

Citrix ADC の詳細ポリシー式 -高度なポリシーの定義に使用できる式の詳細。

Locations
Corporate Headquarters | 851 Cypress Creek Road Fort Lauderdale, FL 33309, United States
Silicon Valley | 4988 Great America Parkway Santa Clara, CA 95054, United States