高级概念

Machine translated content

Disclaimer

Contents

设计

March 2, 2021

使用 OpenShift 验证参考设计中的路由向 Citrix ADC 迁移服务

在不中断的情况下迁移服务。

VRD 用例 — 将 Citrix ADC 动态路由与 Kubernetes 结合使用：Acme Inc. Route Health Injection 和 BGP 集成到 Kubernetes 应用程序

使用 Citrix ADC 上的路由运行状况注入，Acme Inc. 和 Citrix 实施了一个解决方案，为通过现有 BGP + ECMP 路由结构访问的 Kubernetes 服务提供冗余。Acme Inc. 是 Citrix 的一家老客户，大量采用 Citrix ADC。Citrix ADC 是关键 Kubernetes 应用程序的主要负载平衡和业务连续性解决方案。Acme Inc. 目前有三个主要的数据中心。

适用于 Red Hat OpenShift 3.11 的 Citrix Cloud Native Networking 的经过验证的参考设计

Citrix ADC 堆栈满足了以下基本要求：应用程序可用性功能 (ADC)、安全功能隔离 (WAF)、灵活应用程序拓扑扩展（SSL 和 GSLB）以及主动可观察性（服务图）到高度协调的云原生时代环境中。这一经过验证的参考设计引导您完成适用于 Red Hat OpenShift 3.11 的 Citrix Cloud Native Networking 的部署。

Citrix ADC 和 Microsoft Azure 验证的参考设计

Citrix ADC 是一款一体化应用程序交付 Controller，可将应用程序运行速度提高五倍，降低应用程序拥有成本，优化用户体验，并确保应用程序始终可用。

Google Cloud 上的 Citrix ADC CPX、Citrix Ingress Controller 和 Application Delivery Management

Citrix ADC 群集验证的参考设计

Citrix ADC 群集是一组 Citrix ADC nCore 设备，作为单个系统映像一起工作。群集的每个设备都称为节点。Citrix ADC 群集可以包含至少 2 个或多达 32 个 Citrix ADC nCore 硬件或虚拟设备作为节点。

客户端流量在节点之间分布，以提供高可用性、高吞吐量和可扩展性。

Citrix ADC 池容量验证的参考设计

Citrix ADC 池容量是一种许可框架，由带宽池和 Citrix Application Delivery Management 托管并提供服务的虚拟实例池组成。

Kubernetes 中使用 Diamanti 和 Nirmata 验证的参考设计的 Citrix ADC CPX

Citrix ADC 是一款应用程序交付 Controller，可执行特定于应用程序的流量分析，以智能地分发、优化和保护 Web 应用程序的第 4 层 7 (L4—L7) 网络流量。其功能集可以大致包括交换、安全和保护以及服务器群优化功能。

Citrix ADC SSL 配置文件验证的参考设计

使用 SSL 配置文件指定 Citrix ADC 如何处理 SSL 通信。配置文件是 SSL 实体（如虚拟服务器、服务和服务组）的 SSL 参数设置的集合，并提供了易于配置和灵活性。您不限于只配置一组全局参数。您可以创建多个全局参数集（配置文件），并将不同的集分配给不同的 SSL 实体。

Citrix ADC 和 Amazon Web Services 验证的参考设计

Citrix Networking VPX 在 AWS 应用商店中作为 Amazon Machine Image (AMI) 提供。通过 AWS 上的 Citrix Networking VPX，客户能够利用 AWS 云计算功能，并使用 Citrix ADC 负载平衡和流量管理功能满足业务需求。AWS 上的 Citrix ADC 支持物理 Citrix ADC 设备的所有流量管理功能。在 AWS 中运行的 Citrix ADC 实例可以作为独立实例或高可用性对进行部署。

Citrix ADC 管理分区验证的参考设计

NetScaler 管理分区在单个 NetScaler 实例中启用软件级别的多租户。每个分区都有自己的控制平面和网络平面。本文档详细介绍了管理分区启用的典型使用案例，以及在客户环境中使用管理分区的指南。

Citrix Gateway SaaS 和 O365 云验证的参考设计

软件即服务 (SaaS) 是一种软件分发模式，可作为基于 Web 的服务远程交付软件。常用的 SaaS 应用程序，包括 Microsoft Office 365 订阅。

现在可以使用 Citrix Gateway 服务使用 Citrix Workspace 访问 SaaS 应用程序。Citrix Gateway 服务与 Citrix Workspace 相结合，可为已配置的 SaaS 应用程序、已配置的虚拟应用程序或任何其他工作区资源提供统一的用户体验。

使用 Citrix Gateway 服务交付的 SaaS 应用程序可为您提供简单、安全、可靠且可扩展的解决方案来管理应用程序。

使用 Access Control 验证的参考设计的 Citrix Gateway 服务 SSO

使用访问控制服务，管理员可以提供一致的体验，将单点登录、远程访问和内容检查集成到单个解决方案中，以实现端到端访问控制。IT 管理员可以通过简化的单点登录体验来管理对已批准的 SaaS 应用程序的访问。通过访问控制服务，管理员还可以通过筛选对特定网站和网站类别的访问来保护组织的网络和最终用户设备免遭恶意软件和数据泄漏。管理员可以强制执行增强的访问安全策略，以安全访问 SaaS 应用程序。经过身份验证后，员工可以从任何设备访问所有关键业务应用程序，无论他们是在办公场所、家中还是出行。

具有 Azure 负载平衡器前端 IP 验证的参考设计的 Citrix ADC 高可用性

利用 Azure 负载平衡器 (ALB) 作为前端 (FE) 负载平衡器，在 Azure 中实施 NetScaler 高可用性部署。

使用 OpenShift 验证参考设计中的路由向 Citrix ADC 迁移服务

May 20, 2020

OpenShift 群集中的静态路由和自动路由

静态路由（默认）-通过静态路由将 OpenShift 主机子网映射到外部 ADC

静态路由在使用 HAProxy 的传统 OpenShift 部署中很常见。在将服务从一个服务代理迁移到另一个服务代理时，可以与 Citrix Node Controller (CNC)、Citrix Ingress Controller (CIC) 和 CPX 并行使用静态路由，而不会中断正常运行的群集中部署的命名空间。

Citrix ADC 的静态路由配置示例：

```
    oc get hostsubnet (Openshift Cluster) snippet
        oc311-master.example.com 10.x.x.x 10.128.0.0/23
        oc311-node1.example.com 10.x.x.x 10.130.0.0/23
        oc311-node2.example.com 10.x.x.x 10.129.0.0/23

    show route (external Citrix VPX) snippet
        10.128.0.0 255.255.254.0 10.x.x.x STATIC
        10.129.0.0 255.255.254.0 10.x.x.x STATIC
        10.130.0.0 255.255.254.0 10.x.x.x STATIC
<!--NeedCopy-->
```

自动路由-使用 CNC（Citrix 节点控制器）自动执行到定义路径分片的外部路由

您可以通过两种方式将 Citrix ADC 与 OpenShift 集成，这两种方式都支持 OpenShift 路由器分片。

路由类型

- 不安全-外部负载均衡器到 CIC 路由器，HTTP 流量不会被加密。
- 安全边缘-外部负载均衡器到 CIC 路由器终止 TLS。
- 安全直通-外部负载均衡器到目的地终止 TLS
- 安全重新加密-外部负载均衡器到 CIC 路由器终止 TLS。CIC 路由器使用 TLS 加密到目的地。

请参阅 Citrix Ingress Controller 部署解决方案 中有关不同路径类型的更多信息。

部署具有 **OpenShift** 路由器分片支持的 **Citrix Ingress Controller**

Citrix Ingress Controller (CIC) 充当路由器，并将流量重定向到各种容器，以便在各种可用容器之间分配传入流量。

此迁移过程还可以是群集升级过程的一部分，从传统的 OpenShift 拓扑到使用 Citrix CNC、CIC 和 CPX 组件进行群集迁移和升级的自动化部署。

这个解决方案可以通过两种方法实现：

- CIC 路由器插件（窗格）
- CPX 路由器内开放式移位（侧车）

下面介绍了这两种方法以及迁移示例。

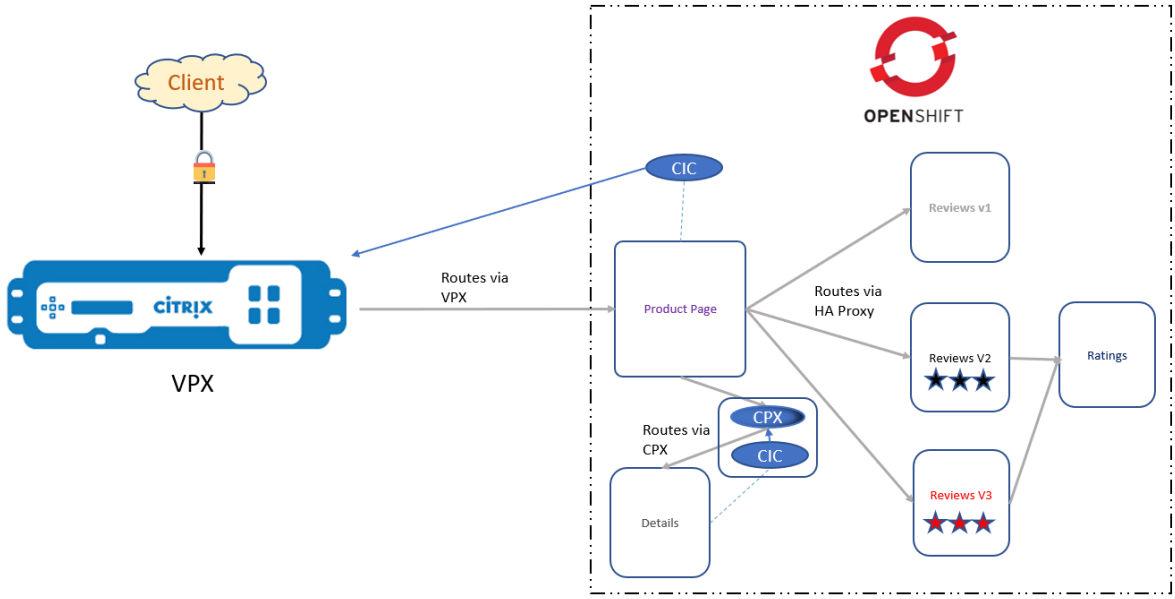
开放式 Shift 路由器分片 允许在多台 OpenShift 路由器之间分配一组路由。默认情况下，OpenShift 路由器从所有命名空间中选择所有路由。在路由器分片中，将标签添加到路由或命名空间中，标签选择器添加到路由器中以过滤路由。每个路由器分片仅选择具有与其标签选择参数匹配的特定标签的路由。

要为 OpenShift 上的 Citrix ADC 部署配置路由器分片，每个分片都需要一个 Citrix Ingress Controller 实例。根据分片所需的条件，Citrix Ingress Controller 实例通过路由或命名空间标签或两者作为环境变量进行部署。当 Citrix Ingress Controller 处理路由时，它会将路由的标签或路由的命名空间标签与其上配置的选择条件进行比较。如果路由满足条件，则会将相应的配置应用于 Citrix ADC，否则不会应用配置。

在路由器分片中，从整个路由池中选择路由的子集将基于选择表达式。选择表达式是多个值和操作的组合。请参阅此 Citrix 博客 中有关表达式、值和操作的更多信息。

书籍信息部署

Bookinfo 应用程序的体系结构如下图所示。CIC 作为一个 OpenShift 路由器插件部署在第一层，将 Citrix ADC VPX 配置为将南北流量路由到产品页面。在第二层中，Citrix ADC CPX 部署为 OpenShift 路由器，在详细信息和产品页面微服务之间路由东西流量，而产品页面、评论和评级微服务之间的东西流量则通过默认 HAProxy 路由器路由。



Citrix 组件

- VPX-向 DNS 呈现群集服务的入口 ADC。
- CIC-通过 CNC 路由向外部 Citrix ADC 提供路由标签和命名空间标签。
- CPX-在 OpenShift 群集中提供开放式 Shift 路由。

部署步骤

1. 为部署创建命名空间。`oc create ns sml`
2. 将 Bookinfo 应用程序部署到命名空间。`oc apply -f bookinfo.yaml`

```
##################################################################################################

# Details service
##################################################################################################

apiVersion: v1
kind: Service
metadata:
  name: details
  labels:
    app: details
    service: details
spec:
```



```
  ports:
  - port: 9080
    name: http
  selector:
    app: details
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: details-v1
  labels:
    app: details
    version: v1
spec:
  replicas: 1
  template:
    metadata:
      annotations:
        sidecar.istio.io/inject: "false"
      labels:
        app: details
        version: v1
    spec:
      containers:
"bookinfo.yaml" 224L, 5120C

<!--NeedCopy-->
```

3. 部署映射到我们的产品页面服务的路由文件。指定 `frontend-ip`（这是第 1 层 ADC 上的内容切换 VIP）

```
oc apply -f routes-productpage.yaml
```

```
apiVersion: v1
kind: Route
metadata:
   name: productpage-route
   namespace: sml
   annotations:
    ingress.citrix.com/frontend-ip: "X.X.X.X"
   labels:
    name: productpage
spec:
   host: bookinfo.com
   path: /
```



```
    port:
      targetPort: 80
    to:
      kind: Service
      name: productpage-service
<!--NeedCopy-->
```

4. 为 sml 命名空间部署 RBAC 文件，该文件为 CIC 提供运行所需的权限。RBAC 文件已命名空间。`oc apply -f rbac.yaml`

```
 kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
rules:
  - apiGroups: [””]
    resources: [”endpoints”, ”ingresses”, ”services”, ”pods”, ”secrets”, ”nodes”, ”routes”, ”namespaces”,”tokenreviews”,”subjectaccessreview”]
    verbs: [”get”, ”list”, ”watch”]
  # services/status is needed to update the loadbalancer IP in service status for integrating
  # service of type LoadBalancer with external-dns
  - apiGroups: [””]
    resources: [”services/status”]
    verbs: [”patch”]
  - apiGroups: [”extensions”]
    resources: [”ingresses”, ”ingresses/status”]
    verbs: [”get”, ”list”, ”watch”]
  - apiGroups: [”apiextensions.k8s.io”]
    resources: [”customresourcedefinitions”]
    verbs: [”get”, ”list”, ”watch”]
  - apiGroups: [”apps”]
    resources: [”deployments”]
    verbs: [”get”, ”list”, ”watch”]
  - apiGroups: [”citrix.com”]
    resources: [”rewritepolicies”, ”canarycrds”, ”authpolicies”, ”ratelimits”]
    verbs: [”get”, ”list”, ”watch”]
  - apiGroups: [”citrix.com”]
    resources: [”vips”]
    verbs: [”get”, ”list”, ”watch”, ”create”, ”delete”]
  - apiGroups: [”route.openshift.io”]
```



```
    resources: [”routes”]
    verbs: [”get”, ”list”, ”watch”]

---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cpx
subjects:
- kind: ServiceAccount
  name: cpx
  namespace: sml
---
apiVersion: v1
kind: ServiceAccount
metadata:
”rbac.yaml” 51L, 1513C
 <!--NeedCopy-->
```

5. 部署您的 CIC，将路由配置推送到 VPX。将参数路由标签与 `route-productpage.yaml` 中指定的标签匹配。有关路由标签语法的详细信息，请参阅此 博客。`oc apply -f cic-productpage-v2.yaml`

```
apiVersion: v1
kind: Pod
metadata:
  name: cic
  labels:
    app: cic
spec:
  serviceAccount: cpx
  containers:
  - name: cic
    image: ”quay.io/citrix/citrix-k8s-ingress-controller:1.7.6”
    securityContext:
       privileged: true
    env:
    - name: ”EULA”
      value: ”yes”
    # Set NetScaler NSIP/SNIP, SNIP in case of HA (mgmt has to be
```



```
          enabled)
18      - name: "NS_IP"
19        value: "X.X.X.X"
20      # Set NetScaler VIP that receives the traffic
21  #    - name: "NS_VIP"
22  #      value: "X.X.X.X"
23      - name: "NS_USER"
24        value: "nsroot"
25      - name: "NS_PASSWORD"
26        value: "nsroot"
27      - name: "NS_APPS_NAME_PREFIX"
28        value: "BOOK"
29      - name: "ROUTE_LABELS"
30        value: "name in (productpage)"
31  #    - name: "NAMESPACE_LABELS"
32  #      value: "app=hellogalaxy"
33      # Set username for Nitro
34  #    - name: "NS_USER"
35  #      valueFrom:
36  #       secretKeyRef:
37  #        name: nslogin
38  #        key: nsroot
39  #    # Set user password for Nitro
40  #    - name: "NS_PASSWORD"
41  #      valueFrom:
42  #       secretKeyRef:
43  #        name: nslogin
44  #        key: nsroot
45      args:
46  #    - --default-ssl-certificate
47  #      default/default-cert
48      imagePullPolicy: Always
49  ~
50  "cic-productpage-v2.yaml" 48L, 1165C
51  <!--NeedCopy-->
```

6. 现在，我们必须创建一个无头服务，使用群集中的 DNS 容器将查找详细信息的用户指向我们的 CPX。
   `oc apply -f detailsheadless.yaml`

```
1  ##################################################################################
2  # Details service
3  ##################################################################################
```



```
apiVersion: v1
kind: Service
metadata:
  name: details
spec:
  ports:
  - port: 9080
    name: http
  selector:
    app: cpx
<!--NeedCopy-->
```

7. 部署新服务以显示详细信息容器。`oc apply -f detailsservice.yaml`

```
##################################################################################

# Details service
##################################################################################

apiVersion: v1
kind: Service
metadata:
  name: details-service
  labels:
    app: details-service
    service: details-service
spec:
  clusterIP: None
  ports:
  - port: 9080
    name: http
  selector:
    app: details
<!--NeedCopy-->
```

8. 部署位于我们创建的详细信息服务前面的新路由定义。请注意，标签是“名称：详细信息”。`oc apply -f detailsroutes.yaml`

```
apiVersion: v1
kind: Route
metadata:
```



```
    name: details-route
    namespace: sml
    annotations:
     ingress.citrix.com/insecure-port: ”9080”
    labels:
     name: details
spec:
    host: details
    path: /
    port:
       targetPort: 9080
    to:
      kind: Service
      name: details-service
<!--NeedCopy-->
```

9. 为 E/W 流量部署 CPX。CIC 作为侧车部署，并使用 ROUTE_LABEL 参数配置，以便与 detailsroutes.yaml 中的标签匹配。`oc apply -f cpx.yaml`

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: cpx
  labels:
    app: cpx
    service: cpx
spec:
  replicas: 1
  template:
    metadata:
      name: cpx
      labels:
        app: cpx
        service: cpx
      annotations:
        NETSCALER_AS_APP: ”True”
    spec:
      serviceAccountName: cpx
      containers:
        - name: cpx
          image: ”quay.io/citrix/citrix-k8s-cpx-ingress:13.0-36.28
              ”
          securityContext:
```



```
24                privileged: true
25            env:
26            - name: "EULA"
27              value: "yes"
28            - name: "KUBERNETES_TASK_ID"
29              value: ""
30            - name: "MGMT_HTTP_PORT"
31              value: "9081"
32            ports:
33            - name: http
34              containerPort: 9080
35            - name: https
36              containerPort: 443
37            - name: nitro-http
38              containerPort: 9081
39            - name: nitro-https
40              containerPort: 9443
41  # readiness probe?
42            imagePullPolicy: Always
43          # Add cic as a sidecar
44          - name: cic
45            image: "quay.io/citrix/citrix-k8s-ingress-controller
                 :1.7.6"
46            env:
47            - name: "EULA"
48              value: "yes"
49            - name: "NS_IP"
50  "cpx.yaml" 75L, 1939C
51  <!--NeedCopy-->
```

微服务环境中的连续交付选择

持续集成 (CI) 是一种开发实践，要求开发人员每天多次将代码集成到共享存储库中。

持续交付 (CD) 是持续集成的自然延伸：这种方法使团队确保系统的每一个更改都是可释放的，并且我们只需按下按钮即可释放任何版本。

不同的 CD 选择以及它们的优点和缺点是：

- 重新创建 -终止版本 1 (V1)，然后推出版本 2 (V2)。
  - 优点
    - 易于设置。
    - 应用程序状态完全续订。
  - 缺点

        * 对用户的影响很大。预计停机时间取决于关机和启动持续时间。
- 动/滚动更新 -V2 缓慢推出并取代 V1。
    - 优点
        * 易于设置。
        * 版本在各实例之间缓慢释放。
        * 方便处理数据重新平衡的有状态应用程序。
    - 缺点
        * 部署/回滚可能需要时间。
        * 支持多个 API 很难。
        * 几乎没有控制交通。
- 蓝绿色 -V2 与 V1 一起释放，然后流量切换到 V2。
    - 优点
        * 即时部署/回滚。
        * 避免版本问题，因为整个应用程序一次更改。
    - 缺点
        * 昂贵，因为它需要两倍的资源。
        * 在发布到生产之前，应该对整个平台进行适当的测试。
        * 处理多个有状态的应用程序可能很困难。
- **Canary** -V2 发布给用户的子集，然后进行完整的部署。
    - 优点
        * 为用户子集发布的版本。
        * 方便的错误率和性能监控。
        * 快速回滚。
    - 缺点
        * 推出缓慢。
        * 处理多个有状态的应用程序可能很困难。
- **A/B** 测试 -V2 在特定条件下发布给用户子集。
    - 优点
        * 多个版本并行运行。
        * 完全控制流量分布。
    - 缺点
        * 需要智能负载均衡器。
        * 很难对给定会话的错误进行故障排除，因此分布式跟踪成为强制性的。
- 影子 -V2 接收 V11 的真实世界流量，不会影响响响应。
    - 优点
        * 使用生产流量对应用程序进行性能测试。
        * 对用户没有影响。
        * 在应用程序的稳定性和性能满足要求之前，不进行部署。
    - 缺点

* 昂贵，因为它需要两倍的资源。
* 不是真正的用户测试，可能会产生误导性。
* 复杂的设置。
* 某些情况下需要嘲笑服务。

参考资料

Citrix GitHub：“开放式 Shift 路由和入口”

Citrix Developer 文档：“部署解决方案”

Citrix 博客：“使用 Citrix ADC 启用 OpenShift 路由器分片支持”

打开 Shift 路由文档：

VRD 使用案例 — 将 Citrix ADC 动态路由与 Kubernetes 结合使用

May 20, 2020

Acme Inc. Route Health Injection 与 BGP 集成，适用于 Kubernetes 应用程序

Acme Inc. 是一家长期以来的 Citrix 客户，拥有较大的 Citrix ADC 占用空间。Citrix ADC 是关键 Kubernetes 应用程序的主要负载平衡和业务连续性解决方案。Acme Inc. 目前有三个主要的数据中心。

Acme Inc. 希望为关键 Kubernetes 应用程序提供冗余和高可用性，以便它们能够在三个数据中心的所有部署机架中提供更高的容错能力。

使用 Citrix ADC 上的路由运行状况注入，该解决方案可为通过现有 BGP + ECMP 路由结构访问的 Kubernetes 服务提供冗余。

除了路由运行状况注入外，许多 Kubernetes 应用程序还要求后端服务器接收真正的客户端 IP。传统的负载平衡与 Citrix ADC 源自 ADC 子网 IP 地址发往后端服务器的数据包。对于需要真正的客户端 IP 地址作为源地址的应用程序，Citrix ADC 提供了多种方法。这些方法包括 USIP（使用源 IP 模式）和 DSR（直接服务器返回）。

Acme Inc. IT 部门为 Kubernetes 应用程序提供测试 VIP 的 Citrix ADC VPX 实例。此测试环境用于使用客户端 IP 构建路由运行状况注入解决方案，并在将其部署到生产环境之前对其进行全面测试。

部署要求

Acme Inc. 和 Citrix 确定了几个不同的要求：

- 每个数据中心的 Citrix ADC VPX 单元，可连接到动态路由网络 (3)
- 最多三个虚拟服务器的 IP 地址，将在 Acme Inc. 动态路由中配置为 /32 路由

环境：

Kubernetes 测试 VIP

- 每个数据中心中要用作每个 Citrix ADC 单元路由运行状况注入 VIP 的下一跳的 SNIP 地址必须具有自己的 SNIP 地址并启用了动态路由。这是广告路由健康注入 VIP 的 Gateway。

识别 Kubernetes 信息，包括：

- 后端 Kubernetes Pod 和测试 VIP
- 所需端口和负载平衡参数
- SSL 证书（如果适用）

客户端 IP 配置

- 后端服务器必须接收真实的客户端 IP 地址
- 客户端 IP 选项部分讨论了多个可用选项

路线健康注射 **(RHI)**

采用路由运行状况注入的 **Citrix ADC** 动态路由

Citrix ADC 中的动态路由和路由运行状况注入的主要目的是将 VIP 的状态或运行状况传达给上游路由器。VIP 的状态取决于与其关联的虚拟服务器以及绑定到该 VIP 的服务。通过路由运行状况注入播发 VIP 通告与虚拟 IP 地址关联的虚拟服务器的状态相关联。

虚拟 IP 地址必须启用播发。这是通过将虚拟 IP 地址上的 `-hostroute` 选项设置为 `enabled` 来实现的。默认情况下，`-hostroute` 选项设置为 `disabled`。当您使用 `add ns ip` 命令添加 IP 地址或使用 `set ns ip` 命令修改现有 IP 地址时，可以启用 `-hostroute` 选项。

路由运行状况注入监控

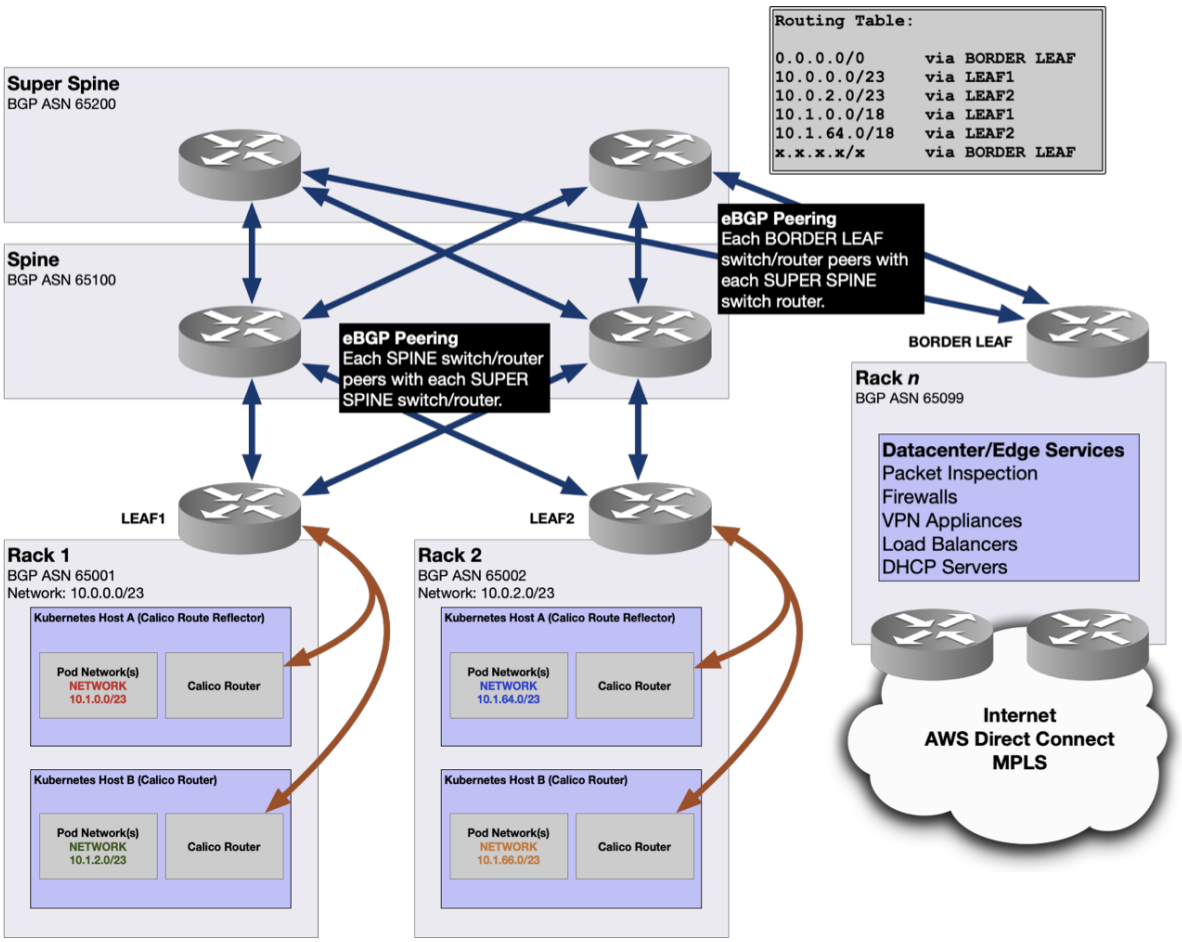
启用 `hostRoute` 此选项后，NetScaler 内核将根据与虚拟 IP 地址关联的虚拟服务器的状态将主机路由注入 ZebOS NSM（网络服务模块）。- `vserverroute health injectionLevel` 交换机控制虚拟服务器的状态与发送到网络服务模块 (NSM) 的虚拟 IP 主机路由之间的关系。

可用于虚拟服务器路由运行状况注入级别的三个选项：

- `ALL_VSERVERS` — 只有在与虚拟 IP 关联的所有虚拟服务器均为 UP 时，才会向 NSM 注入主机路由。
- `ONE_VSERVER` — 仅当与虚拟 IP 关联的任何一个虚拟服务器处于“开启”状态时，主机路由才会注入 NSM。
- `NONE` — 无论与虚拟 IP 关联的虚拟服务器的状态如何，都会向 NSM 注入主机路由。

> 注意：
>
> 默认情况下，–vserverRHILevel 设置为 ONE_VSERVER。

下图描述了与 Citrix ADC 上负载平衡虚拟服务器关联的虚拟 IP 地址的基本路由运行状况注入功能：



具有多个数据中心的路由运行状况注入选项

下面介绍了根据每个应用程序的特定要求选择的路由运行状况注入配置。可用选项如下：

活动 — 活动，BGP 为每个客户端确定最有效的路由（任播或 ECMP）

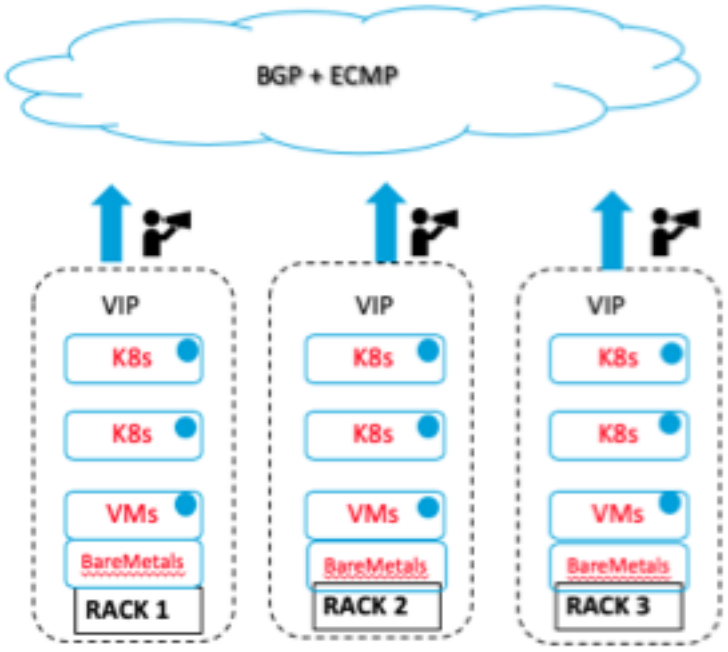
路由运行状况注入活动 — 活动：任播或 **ECMP**

路由健康注入活动/活动是任播或 ECMP。这是一个真正的主动-主动替代方案。在这种情况下，路由运行状况注入 VIP 的 /32 路由在所有数据中心中公布，而无需向 BGP 呈现成本或本地偏好。将向网络呈现三条路由，其中 Citrix ADC 的 SNIP 地址特定于每个数据中心，充当访问每个 VIP 的 Gateway。Acme Inc. 动态路由环境以相等的成本将客户端请求定向到数据中心，以便进行流量分配。如果三个数据中心之一的服务出现故障，绑定到负载平衡服务的监视器会导致虚拟服务器关闭。这反过来会删除数据中心出现故障的路由通告。所有客户端连接都会继续使用其余数据中心。

路由运行状况注入主动-主动配置的重要注意事项：

1. 对于基于 TCP 和 UDP 的服务，建议使用 ECMP 的路由健康注入，并且需要 Acme Inc. 网络团队进行 BGP 配置。使用 ECMP 的路由健康注入确实对上游路由器可支持的路由数量有限制 (64)。
2. 路由运行状况注入与基于 UDP 的服务配合使用，不推荐用于基于 TCP 的服务。

下图描述了活动-活动任意交流/ECMP 方案：



Citrix ADC 和客户端 **IP** 选项

Acme Inc. 对于其大量 Kubernetes 应用程序要求的重要要求之一是，后端服务器接收由 Citrix ADC 负载平衡的服务的真正客户端 IP 地址。Citrix ADC 上的典型负载平衡来源于 NetScaler 拥有的 SNIP（子网 IP）地址发往后端服务器的所有流量。对于某些应用程序，需要真正的客户端 IP。大多数利用路由运行状况注入的应用程序还需要将真正的客户端 IP 发送到后端服务器。

Citrix ADC 具有名为 “使用源 IP”(USIP) 的功能，可以全局绑定，也可以单独绑定到需要客户端 IP 到后端服务器的每个服务。这方面的问题是，一旦客户端收到具有不同源 IP 的数据包，就会发生非对称路由，并且数据包被丢弃。正因为如此，必须评估其他注意事项，并且需要在后端服务器上进行额外配置，以便 USIP 正常工作。

在 Citrix ADC 上实施使用源 IP 模式时，一个重要的注意事项是，必须在 Citrix ADC 模式下关闭浪涌保护模式。有关具有浪涌保护的 USIP 模式的详细信息，请参阅 Citrix 文章 此处。

Citrix ADC 提供了多种方法来实现这一目标，下一节将介绍这些方法。可用的选项包括：

- 使用 Citrix ADC 截取功能作为默认 Gateway 的 USIP 模式
- 直接服务器返回第 3 层
  - IP 隧道
  - TOS 标头上的客户端 IP 插入
- 直接服务器返回第 2 层
- TCP 标头的客户端 IP 插入

将 **Citrix ADC** 截取作为默认网关的 **USIP** 模式

在与 Acme Inc. IT 部门的多次会议期间，确定这种方法是大多数需要客户端 IP 的负载平衡服务的首选。此方法涉及更改负载平衡的每个后端服务器的默认 Gateway，并将其设置为托管负载平衡 VIP 的 Citrix ADC 单元的 SNIP 地址。此选项支持所有 Citrix ADC 功能，而不是直接服务器返回选项，因为 Citrix ADC 仅管理传入的客户端请求。此选项还需要 ADC 单元的带宽最多。
此方法具有以下基本要求：

- Citrix ADC 必须与所有正在负载平衡的后端服务器在同一 L2 子网中具有 SNIP 地址。
- SNIP 地址配置为所有后端服务器的默认 Gateway。多个 SNIP 地址可用于不同 L2 子网中的后端服务器。
- 必须在指向后端服务器的服务上启用 USIP 模式。

> 注意：
>
> USIP 也可以在 Citrix ADC 单元上全局启用，但 USIP 仅应用于启用 USIP 模式后创建的服务。

Citrix 建议向后端服务器添加更多网络接口，并为非客户端流量配置静态路由。

- 备份例程和需要带宽的其他进程不必穿越 Citrix ADC 单元。

直接服务器返回：第 **3** 层选项

使用 Citrix ADC 直接返回服务器是在负载平衡配置中获取后端服务器上的客户端 IP 地址的另一个配置选项。可以在第 3 层模式下配置直接服务器返回，因此允许在其他 L3 VLAN 上使用后端服务器，而 USIP 则需要从 ADC 到后端服务器的 L2 连接。由于响应流量未遍历 Citrix ADC 单元，直接服务器返回配置不支持某些 Citrix ADC 功能。此选项要求 Citrix ADC 单元的吞吐量最低。

直接服务器返回具有后端服务器所需的更复杂的配置，因为它们必须能够提取客户端 IP 并重写 TCP 标头以直接响应客户端。Citrix 当前支持两种不同的方法来配置第 3 层 DSR：

- 具有 IP 隧道的 DSR 模式（IP 通过 IP）
- 使用 TOS 的 DSR 模式（服务类型 TCP 标头字段）第 3 层

DSR 具有以下基本要求：

- 必须在服务上使用 USIP 配置 Citrix ADC。
- 后端服务器的环回地址配置为 Citrix ADC VIP 地址。
- 必须针对每种方法专门配置后端服务器：
  - IP 隧道：后端服务器必须从 ADC 解除数据包的封装，并提取客户端 IP 以便直接响应客户端。
  - TOS（服务类型）：后端服务器必须能够读取 TCP 数据包的 TOS 标头，并使用此信息直接回复客户端。
  - 任何一种方法都可能需要在后端服务器上进行自定义配置，并使用第三方应用程序。

第 3 层 DSR 可能需要配置防火墙和安全设备上的例外情况。

有关使用第 3 层直接返回服务器的更多信息，请在此处找到：

- 使用 TOS 的 DSR
- 具有 IP 隧道功能的 DSR

公开“负载均衡器”类型的服务

在公有云（如 AWS、GCP 或 Azure）上的 Kubernetes 部署中，本机支持负载均衡器类型的服务。在云部署中，当您创建负载均衡器类型的服务时，将为该服务分配云托管负载均衡器。然后使用负载均衡器公开服务。

对于 Kubernetes 的本地部署、裸机或公有云部署，您可以使用群集外部的 Citrix ADC 来平衡传入流量。Citrix Ingress Controller 提供了灵活的 IP 地址管理，可实现 Citrix ADC 的多租户功能。Citrix Ingress Controller 允许您使用单个 ADC 对多个服务进行负载平衡，并结合了各种入口功能。将 Citrix ADC 与 Citrix Ingress Controller 结合使用，您可以最大限度地利用公有云的负载均衡器资源，并显著降低运营开支。

当 Citrix ADC 位于 Kubernetes 群集（第 1 层）之外时，Citrix Ingress Controller 支持负载均衡器类型的服务。创建、更新或删除负载均衡器类型的服务时，Citrix Ingress Controller 将使用负载平衡虚拟服务器配置 Citrix ADC。

负载平衡虚拟服务器使用通过以下方式之一获取的 IP 地址（虚拟 IP 地址或 VIP）进行配置：

1. 使用 Citrix 提供的 IPAM Controller 自动为服务分配虚拟 IP 地址。该解决方案的设计方式使您可以轻松地将解决方案与 Infoblox 等外部 DNS 提供商集成。有关详细信息，请参阅与外部 DNS 的互操作性。

2. 通过在服务定义中使用 spec.loadBalancerIP 字段指定 IP 地址。Citrix Ingress Controller 使用 spec.loadBalancerIP 字段中提供的 IP 地址作为与服务对应的负载平衡虚拟服务器的 IP 地址。

```
apiVersion: v1
kind: Service
metadata:
    name: hello-world-service
spec:
    type: LoadBalancer
    loadBalancerIP: ””
ports:
- port: 80
    targetPort: 8080
selector:
    run: load-balancer-example
<!--NeedCopy-->
```

有关更详细的参考，请参阅 公开负载均衡器类型的服务。

适用于 **Red Hat OpenShift 3.11** 的 **Citrix Cloud Native Networking** 的经过验证的参考设计

May 20, 2020

Citrix ADC 堆栈满足了以下基本要求：应用程序可用性功能 (ADC)、安全功能隔离 (WAF)、灵活应用程序拓扑扩展（SSL 和 GSLB）以及主动可观察性（服务图）到高度协调的云原生时代环境中。

数字化转型推动了将现代应用程序部署转移到基于微服务的体系结构的需求。这些云原生体系结构利用应用容器、微服务和 Kubernetes。

现代应用程序的云原生方法也改变了开发生命周期，包括敏捷的工作流程、自动化部署工具集以及开发语言和平台。

现代应用程序部署的新时代也改变了传统的数据中心业务模式领域，包括月度和年度软件发布和合同、思洛存储器计算资源和预算以及供应商消费模式。

虽然所有这些现代化都在生态系统中进行，但仍然存在以下基本要求：应用程序可用性功能 (ADC)、安全功能隔离 (WAF)、敏捷应用程序拓扑 (SSL 和 GSLB) 的扩展以及主动可观察性 (服务图) 到高度协调的环境。

为什么选择 **Citrix** 实现现代应用程序交付

现代应用程序部署的 Citrix 软件方法需要在组织内的许多团队中集成敏捷的工作流程。敏捷应用程序开发和交付的优势之一是称为 CI/CD 的框架。

CI/CD 是为现代应用程序生命周期提供速度、安全性和可靠性的一种方式。

持续集成 (CI) 允许通用代码库，每天可实时更新几次，并集成到自动构建平台中。
持续集成的三个阶段是推送、测试、修复。

持续交付 (CD) 将部署管道直接集成到 CI 开发过程中，从而优化和改进现代应用程序的软件交付模式。

Citrix ADC 通过实施自动化金丝雀分析渐进式部署，与持续交付流程结合起来。

面向所有利益相关者的解决方案

Citrix 创建了一个专用的基于软件的解决方案，可满足部署现代应用程序时的跨功能需求，并集成了可观察性堆栈、安全框架和 CI/CD 基础架构的各种组件。

采用 CI/CD 技术部署现代应用程序的传统组织已经认识到需要为 CI/CD 所涉及的所有成员提供通用交付和可用性框架，这些资源通常被定义为业务部门“利益相关者”，而每个利益相关者都是投资于本组织的总体成功, 每个利益攸关方一般都有不同的要求和差异.

现代交付活动中利益相关者的一些常见例子包括：

- 平台团队-部署数据中心基础体系结构，如 IaaS、PaaS、SDN、ADC、WAF
- DevOps 和工程团队 — 开发和维护统一的代码库、自动化工具和软件体系结构

- 服务可靠性工程 (SRE) 团队 — 减少组织孤岛、错误管理、部署自动化和测量
- 安全运营团队 — 主动安全策略、事件管理、修补程序部署、产品组合强化

Citrix 软件堆栈解释

单一代码库-对您来说都是相同的代码
- 本地部署、公有云部署、私有云部署、GOV 云部署

- 平台选择-为了满足任何敏捷要求，请选择任何 Citrix ADC 型号
  - CPX — Citrix ADC CPX 是作为容器交付的 Citrix ADC
  - VPX — Citrix ADC VPX 产品是一款虚拟设备，可托管在各种虚拟化和云平台上，性能范围从 10 MB/秒到 100 Gb/秒。
  - MPX — Citrix ADC MPX 是一款基于硬件的应用程序交付设备，性能范围从 500 MB/秒到 200 Gb/秒。
  - SDX — Citrix ADC SDX 设备是一个多租户平台，您可以在其上置备和管理多个虚拟 Citrix ADC 计算机（实例）。
  - BLX — Citrix ADC BLX 设备是 Citrix ADC 的软件外形规格。它旨在在商用现成服务器 (COTS) 上的裸金属 Linux 上以本机方式运行
    Containerized Environments - create overlays and automatically configure your Citrix ADC
  - Citrix Ingress Controller - built around Kubernetes Ingress and automatically configures one or more Citrix ADC based on the Ingress resource configuration
  - Citrix 节点控制器 – create a VXLAN-based overlay network between the Kubernetes nodes and the Ingress Citrix ADC
  - Citrix IPAM 控制器 – automatically assign the load balancing virtual server on a Citrix ADC with an IP address (virtual IP address or VIP)
    Pooled Capacity Licensing – one global license
    - Ubiquitous global license pool decouples platforms and licenses for complete flexibility for design and performance
    Application Delivery Manger – the single pane of glass
    - Manage the fleet, orchestrate policies and applications, monitor and troubleshoot in real-time
    Flexible Topologies – traditional data center or modern clouds
    - Single tier, two-tier, and service mesh lite

Citrix ADC 值

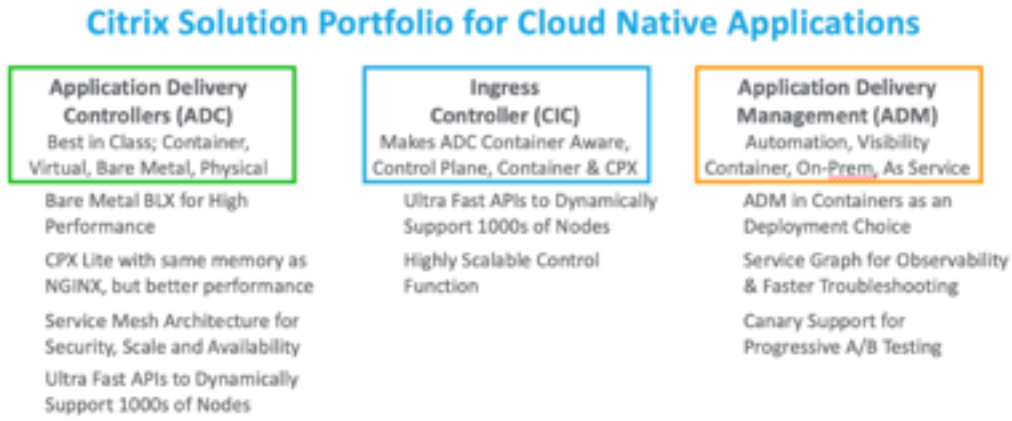
- Kubernetes 和 CNCF Open Source Tools 集成
- 完美代理 — 适用于现代应用的经验证的 Layer7 应用程序交付 Controller
  - 采用 Pod 或侧车部署的高性能 ADC 容器
  - 使用多个选项对 Kubernetes 群集进行低延迟访问
- 功能丰富的 API — 轻松实施和编排安全功能，无限制

- 用于 CI/CD 的高级流量转向和金丝雀部署
- 通过 TLS/SSL、WAF、DoS 和 API 保护经验证的安全性
- 丰富的第 7 层功能
- 针对旧式和现代应用程序部署的集成监控
- 实现可视性的可操作见解和服务图表

Citrix ADC 的优势

- 移动旧版应用程序而无需重写它们
- 开发人员可以使用 Kubernetes API 使用 Citrix ADC 策略保护应用程序（使用 CRD-开发人员友好）
- 为南北和服务网状部署高性能微服务
- 对所有微服务使用一个应用服务图表
- 通过 TCP、UDP、HTTP/S、SSL 更快地排除微服务问题
- 保护 API 并使用 Kubernetes API 进行配置
- 加强 CICD 流程，加强金丝雀部署

体系结构组件



Citrix ADC 套件的优势

Citrix 是选择。

无论您是使用旧式数据中心和组件，还是已经启动了新的云原生现代应用程序，Citrix ADC 都可以无缝集成到您可能有的任何平台要求中。我们为基于订阅的云平台和工具提供云原生 ADC 功能，允许通过简单的 Ingress Controller 调配将流量引导和调配到您的 Kubernetes 群集内部的流量，并处理从简单到复杂的 Service Mesh 体系结构。

Citrix 已验证。

经过验证的设计模板和示例应用程序允许您轻松参考所需状态和业务需求，从而快速、完整地解决。我们在一个中心位置记录并发布了配置示例，以便在 DevOps、SecOps 和平台团队之间轻松参考。

Citrix 是敏捷和现代。
创建基础体系结构，以便客户将 Citrix Cloud 本机堆栈的新功能与其现有的 ADC 和新模块（CNC、IPAM 等）结合使用

Citrix 是开放式的。
帮助客户了解我们与合作伙伴生态系统的集成。在本文档中，我们同时使用开源 CNCF 工具和 Citrix 企业级产品。

合作伙伴生态系统

本主题提供有关包括 Citrix ADC 和 Citrix Ingress Controller 在内的云原生部署中支持的各种 Kubernetes 平台、部署拓扑、功能和 CNI 的详细信息。

以下平台支持 Citrix Ingress Controller：

- Kubernetes 1.10 版在裸机或公共云（如 AWS、GCP 或 Azure）上自托管。
- Google Kubernetes Engine (GKE)
- Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Red Hat OpenShift 3.11 及更高版本
- Pivotal Container Service (PKS)
- Diamanti Enterprise Kubernetes 平台

我们的合作伙伴生态系统还包括以下内容：

- Prometheus — 用于指标、警报和洞察的监控工具
- Grafana — 用于分析和监控的平台
- Spinnaker — 多云连续交付和金丝雀分析的工具
- Elasticsearch — 应用程序或网站搜索服务
- Kibana — 弹性搜索数据的可视化工具和弹性堆栈导航工具
- Fluentd — 数据收集器工具

下一节的重点是使用 OpenShift 进行设计/体系结构。

OpenShift 概述

Red Hat OpenShift 是一个 Kubernetes 部署平台，专注于使用微服务和容器更快地构建和扩展应用。OpenShift 可以自动化、安装、升级和管理容器堆栈，简化 Kubernetes 并方便日常开发运营任务。

- 开发人员调配应用程序，可访问经验证的解决方案和合作伙伴，这些解决方案通过简化的工作流
- 操作可以使用 Web 控制台和内置日志记录和监视来管理和扩展环境。

图 1-6：OpenShift 高级体系结构。

OpenShift 的更多优势和组件包括：

- 基础架构的选择
- 主节点和工作人员节点

- 映像注册表
- 路由和服务层
- 开发人员操作（引入但超出了本文档的范围）

将 Red Hat OpenShift 与 Citrix 本机堆栈集成的使用案例包括：

- 传统应用程序支持
- 重写/响应程序策略作为 API 部署
- 微服务故障排除
- 通过安全补丁和功能增强功能的日常操作

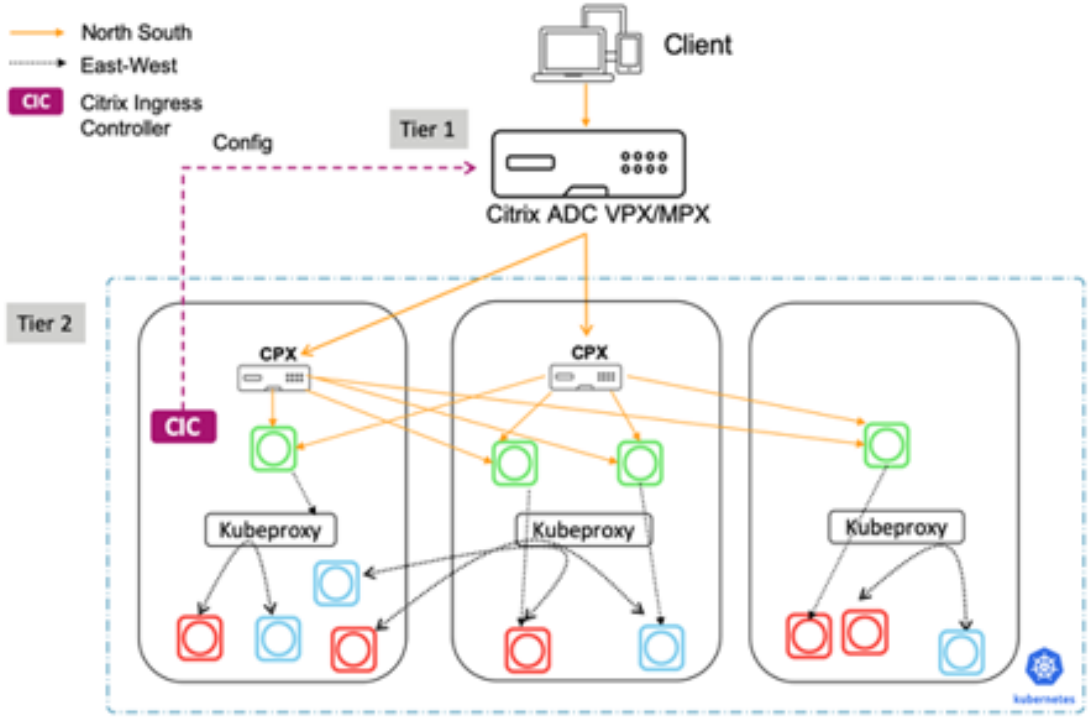
在本文档中，我们将介绍 Citric ADC 如何提供可靠的路由/服务层集成。

OpenShift 项目

OpenShift 添加的第一个新概念是项目，它有效地包装了一个命名空间，通过项目控制对命名空间的访问。访问权限通过基于用户和组的身份验证和授权模型进行控制。因此，OpenShift 中的项目在命名空间之间提供墙壁，确保用户或应用程序只能看到和访问允许的内容。

OpenShift 命名空间

Kubernetes 中的主要分组概念是命名空间。命名空间也是在多种用途之间划分群集资源的一种方法。话虽如此，Kubernetes 中的命名空间之间没有安全性。如果您是 Kubernetes 群集中的“用户”，则可以看到所有不同的命名空间及其中定义的资源。



OpenShift Software Defined Networking (SDN)

OpenShift Container Platform 使用软件定义的网络 (SDN) 方法提供统一的群集网络，以便在 OpenShift Container Platform 群集之间实现通信。此容器网络由 OpenShift SDN 建立和维护，后者使用开放 vSwitch (OVS) 配置覆盖网络。

OpenShift SDN 提供了三个 SDN 插件，用于配置容器网络：

- ovs-子网插件是原始插件，它提供了一个“扁平”容器网络，每个容器都可以与每个其他容器和服务进行通信。
- ovs-多租户插件为容器和服务提供了项目级隔离。每个项目都会收到一个唯一的虚拟网络 ID (VNID)，用于标识分配给项目的容器中的流量。来自不同项目的容器不能向不同项目的容器和服务发送数据包或接收数据包。
- 但是，接收 VNID 0 的项目更具特权，因为它们被允许与所有其他 POD 进行通信，并且所有其他 POD 都可以与其进行通信。在 OpenShift Container Platform 群集中，默认项目具有 VNID 0。这有助于某些服务（如负载平衡器）与群集中的所有其他窗格进行通信，反之亦然。
- ovs-networkpolicy plug-in 允许项目管理员使用 NetworkPolicy 对象配置自己的隔离策略。

OpenShift 路由和插件

OpenShift 管理员可以在 OpenShift 群集中部署路由器，该群集允许开发人员创建的路由用于由外部客户。OpenShift 中的路由层是可插拔的，默认情况下提供并支持两个可用的路由器插件。

OpenShift 路由器提供外部主机名映射以及负载平衡到 services，通过协议将区分信息直接传递给路由器实现；主机名必须按数序存在于协议中，以确定将其发送到何处。

路由器插件假定它们可以绑定到主机端口 80 和 443。这是为了允许外部流量路由到主机，然后通过路由器。路由器还假定网络配置为可以访问群集中的所有容器。

OpenShift路由器是针对 OpenShift 安装中的 services 的所有外部流量的入口点。OpenShift 提供并支持以下路由器插件：

- HAProxy 模板路由器是默认插件。它使用 openshift3/ose-haproxy-routerimage 在 OpenShift 上的容器内部与模板路由器插件一起运行 HAProxy 实例。它目前支持 HTTP (S) 流量和通过 SNI 启用 TLS 的流量。与大多数只侦听专用 IP 的容器不同，路由器的容器在主机网络接口上侦听。路由器将外部路由名称请求代理到由与路由关联的服务标识的实际容器的 IP。
- Citrix Ingress Controller 可以作为路由器插件部署在 OpenShift 群集中，以便与部署在您的环境中的 Citrix ADC 集成。通过 Citrix Ingress Controller，您可以将 Citrix ADC 的高级负载平衡和流量管理功能与 OpenShift 群集配合使用。请参阅 将 Citrix Ingress Controller 部署为 OpenShift 群集中的路由器插件。

OpenShift 路由和入口方法

在 OpenShift 群集中，外部客户端需要一种方法来访问播客提供的服务。OpenShift 提供两种资源用于与群集中运行的服务进行通信：路由 和 进入。

路由

在 OpenShift 群集中，路由公开给定域名上的服务或将域名与服务相关联。OpenShift 路由器根据路由中指定的规则将外部请求路由到 OpenShift 群集内的服务。使用 OpenShift 路由器时，还必须配置外部 DNS，以确保流量已登陆路由器上。

Citrix Ingress Controller 可以作为路由器插件部署在 OpenShift 群集中，以便与部署在您的环境中的 Citrix ADC 集成。通过 Citrix Ingress Controller，您可以将 Citrix ADC 的高级负载平衡和流量管理功能与 OpenShift 群集配合使用。
OpenShift 路由可以是安全的，也可以是不安全的。安全路由指定路由的 TLS 终止。

Citrix Ingress Controller 支持以下 OpenShift 路由：

- 不安全的路由：对于不安全的路由，HTTP 流量不会被加密。
- 边缘终止：对于边缘终止，TLS 在路由器终止。通过内部网络从路由器到端点的流量未加密。
- 直通终止：使用直通终止时，路由器不会参与 TLS 卸载，加密流量会直接发送到目的地。
- 重新加密终止：在重新加密终止时，路由器终止 TLS 连接，然后再建立到终端的另一个 TLS 连接。

根据您希望使用 Citrix ADC 的方式，有两种方法可以将 Citrix Ingress Controller 部署为 OpenShift 群集中的路由器插件：在群集内部作为 Citrix ADC CPX 或在群集外部作为 Citrix ADC MPX/VPX。

将 **Citrix ADC CPX** 部署为 **OpenShift** 群集中的路由器

Citrix Ingress Controller 与 Citrix ADC CPX 容器一起部署在同一个容器中。在此模式下，Citrix Ingress Controller 配置 Citrix ADC CPX。请参阅 将 Citrix ADC CPX 部署为 OpenShift 群集中的路由器。

将 **Citrix ADC MPX/VPX** 部署为 **OpenShift** 群集之外的路由器

Citrix Ingress Controller 作为独立容器进行部署，允许您从 OpenShift 群集外部控制 Citrix ADC MPX 或 VPX 装置。请参阅 将 Citrix ADC MPX/VPX 部署为 OpenShift 群集之外的路由器。

进入

Kubernetes 进入 提供了一种基于请求主机或路径将请求路由到服务的方法，将许多服务集中到单个入口点。

Citrix Ingress Controller 围绕 Kubernetes 入口构建，根据入口资源自动配置一个或多个 Citrix ADC 设备。

使用入口的路由可以通过以下方式完成：

- 基于主机名的路由
- 基于路径的路由
- 基于通配符的路由
- 精确路径匹配
- 非主机名路由
- 默认后端

有关示例和更多信息，请参阅 入口配置。

将 **Citrix Ingress Controller** 部署为 **OpenShift** 路由器插件

根据您希望使用 Citrix ADC 的方式，有两种方法可以将 Citrix Ingress Controller 部署为 OpenShift 群集中的路由器插件：

- 作为同一窗格中 Citrix ADC CPX 旁边的侧车容器：在此模式下，Citrix Ingress Controller 配置 Citrix ADC CPX。请参阅 将 Citrix ADC CPX 部署为 OpenShift 群集中的路由器。
- 作为 OpenShift 群集中的独立容器：在此模式下，您可以控制部署在群集外的 Citrix ADC MPX 或 VPX 装置。请参阅 将 Citrix ADC MPX/VPX 部署为 OpenShift 群集之外的路由器。

推荐的体系结构

我们建议客户在设计微服务体系结构时使用以下体系结构：

- Citrix 统一入口
- Citrix 2 层入口
- Citrix 服务网精简版

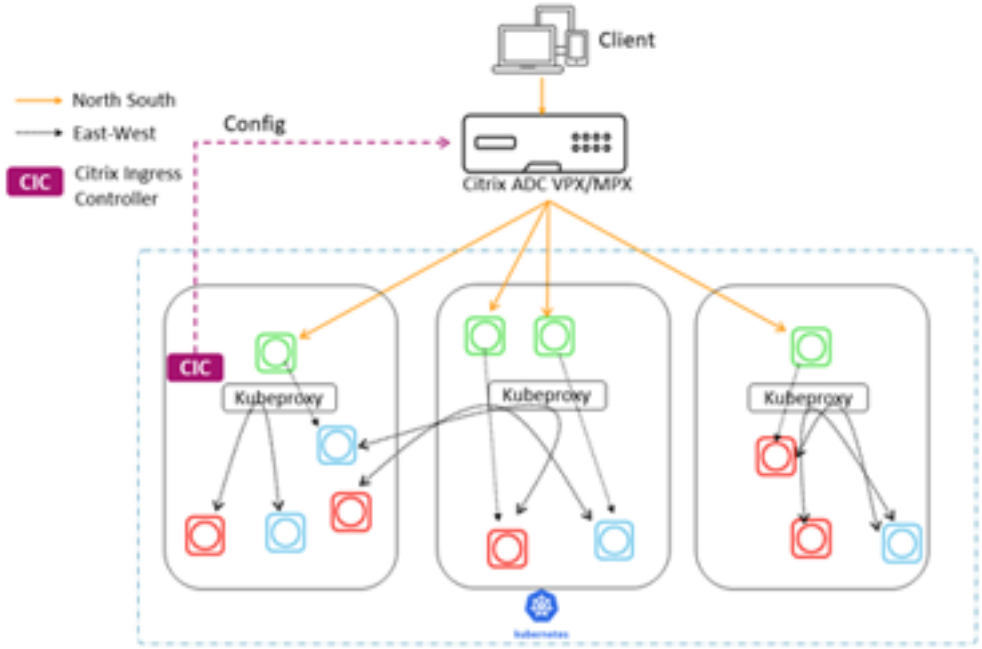
图 1-2：体系结构范围从相对简单到更复杂和功能丰富。

Citrix 统一入口

在统一入口部署中，Citrix ADC MPX 或 VPX 设备将来自客户端的北南流量代理到作为群集内微服务部署的企业级应用程序。Citrix Ingress Controller 部署为 Kubernetes 群集中的容器或侧车，以便根据对微服务或入口资源的更改自动配置 Citrix ADC 设备（MPX 或 VPX）。

您可以在应用程序仍然是一个整体时开始实施统一入口模型。只需将 Citrix ADC 定位为反向代理，即可在应用程序服务器前面实施后面介绍的功能。然后，您将处于将应用程序转换为微服务的良好位置。

微服务之间的通信通过您选择的机制（kube-proxy、IPVS 等）进行处理。



在不同类别中提供的功能

统一入口体系结构的功能分为三组。

第一组中的功能优化性能：

- 负载平衡
- 低延迟连接
- 高可用性

第二组中的功能提高了安全性并使应用程序管理更加轻松：

- 速率限制
- SSL/TLS 终止
- HTTP/2 支持
- 运行状况检查

最终组中的功能特定于微服务：

- 服务中央通信点
- API Gateway 功能

摘要

统一入口模型的功能包括强大的服务负载平衡、中央通信点、动态服务发现、低延迟连接、高可用性、速率限制、SSL/TLS 终止、HTTP/2 等。

统一入口模型可以轻松管理流量、负载平衡请求以及对后端微服务应用程序中的更改进行动态响应。

优势包括：

- 南北流量可扩展性良好，可用于观察和监控，并通过 Spinnaker 和 Citrix ADM 等工具提供持续交付
- 单一层可以统一管理网络和平台服务的基础体系结构团队，并减少跳跃以降低延迟
- 适用于不需要 Web App Firewall 和 SSL 卸载但可以稍后添加的内部应用程序

缺点包括：

- 使用 kube-代理没有东西安全性，但可以为 L4 分段添加印花布
- 库贝代理可扩展性未知
- 由于 kube-proxy 不提供可见性、控制或日志，从而减少开放式工具集成和持续交付，因此无法查看东西流量。
- 平台团队还必须精通网络

Citrix 2 层入口

2 层入口体系结构模型是云原生新手的一个很好的解决方案。在此模型中，第 1 层中的 Citrix ADC 管理传入流量，但将请求发送到由开发人员管理的 2 层 ADC，而不是直接发送到服务实例。第 2 层入口模型将由平台和开发人员团队编写的策略仅应用于入站流量，并启用云扩展和多租户。

图 1-4：具有一级 Citrix ADC VPX/MPX 和二级 Citrix ADC CPX 容器的 Citrix 2 层入口模型的图表。

第 **1** 层提供的功能

第一层 ADC 由传统网络团队管理，提供 L4 负载平衡、Citrix Web App Firewall、SSL 卸载和反向代理服务。第一层中的 Citrix ADC MPX 或 VPX 设备代理从客户端的流量（南北）到第 2 层中的 Citrix ADC CPX。

默认情况下，Citrix Ingress Controller 将在第 1 层对以下配置进行编程：

- 向用户反向代理应用程序：
- 内容交换虚拟服务器
- 虚拟服务器（前端，面向用户）
- 服务组
- SSL 卸载
- NetScaler 日志/调试
- 服务的健康监测

第 **2** 层提供的功能

虽然第一层 ADC 提供反向代理服务，但由平台团队管理的第 2 层 ADC 充当微服务的通信点，提供：

- 动态服务发现
- 负载平衡
- 可见性和丰富的指标

然后，第 2 层 Citrix ADC CPX 将流量路由到 Kubernetes 群集中的微服务。作为独立容器部署的 Citrix Ingress Controller 可以配置一级设备。而且，一个或多个 Citrix ADC CPX 窗格中的侧车 Controller 可在同一个窗格中配置关联的 Citrix ADC CPX。

摘要

2 层模型中的微服务网络体系结构使用两个针对不同角色配置的 ADC。第 1 层 ADC 充当面向用户的代理服务器，第 2 层 ADC 作为微服务的代理。

在两个不同层之间拆分不同类型的功能可提供快速、控制和优化安全性的机会。在第 2 层中，负载平衡快速、稳健且可配置。

使用此模型，ADC 管理员和开发人员之间有明确的分离。它是面向开发人员的 BYOL。

优势包括：

- 南北流量可扩展性良好，可用于观察和监控，并通过 Spinnaker 和 Citrix ADM 等工具提供持续交付
- 为云原生初学者提供最简单、更快速的部署，网络和平台团队的新学习有限

缺点包括：

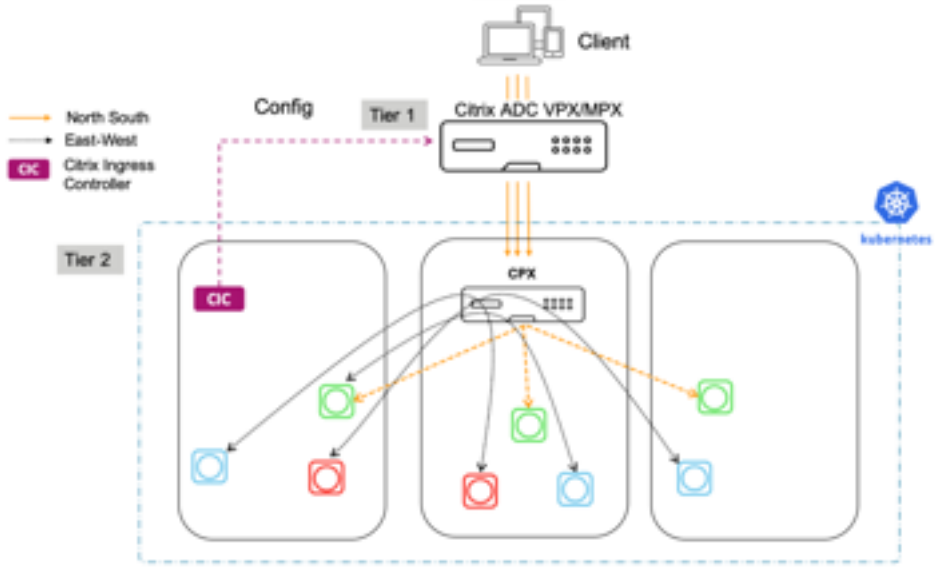
- 使用 kube-代理没有东西安全性，但可以为 L4 分段添加印花布
- 库贝代理可扩展性未知
- 由于 kube-proxy 不提供可见性、控制或日志，从而减少开放式工具集成和持续交付，所以东西方流量的可见性受到限制。

Citrix 服务网精简版

服务网格精简版是三种型号中最丰富的功能。它在内部安全、快速、高效且具有弹性，可用于实施入站和容器间流量的策略。

服务网格精简版模型适用于多种用例，其中包括：

- 健康和金融应用程序 — 法规和用户要求要求金融和健康应用程序将安全性和速度结合起来，金融和声誉价值涉及数十亿美元的财务和声誉价值。
- 电子商务应用程序 — 用户信任对电子商务来说是一个巨大的问题，速度是一个关键的竞争优势。因此，将速度和安全性结合起来至关重要。



摘要

优势包括：

- 使用负载平衡器
  CPX 将策略应用于入站流量和容器间流量，并部署完整的 L7 策略
- 为南北和东西流量提供更丰富的可观察性、分析性、持续交付和安全性
- 每个具有嵌入式 Citrix ADC 的容器的金丝雀
- 单一层可以统一管理网络和平台服务的基础体系结构团队，并减少跳跃以降低延迟

缺点包括：

- 要部署的更复杂的模型
- 平台团队必须精通网络

体系结构选择摘要

Citrix 统一入口

- 南北 **(NS)** 应用程序流量 - 一个 Citrix ADC 负责 K8 群集外的 L4 和 L7 NS 流量、安全性和外部负载平衡。
- 东西部 **(EW)** 应用程序流量 - kube-proxy 负责 L4 EW 流量。
- 安全性 - ADC 负责保护 NS 流量并对用户进行身份验证。Kube-代理负责 L4 EW 流量。
- 可扩展性和性能 - NS 流量具有良好的可扩展性，群集是一种选项。ew 流量和 kube-proxy 可扩展性未知。
- 可观察性 - ADC 为 NS 流量提供了出色的可观察性，但 EW 流量没有可观察性。

Citrix 2 层入口

- 南北 **(NS)** 应用程序流量 - 第 1 层 ADC 负责 SSL 卸载、Web App Firewall 和 L4 NS 流量。它适用于整体和 CN 应用。第 2 层 CPX 可管理 k8 和 L7 NS 流量的快速变化。
- 东西部 **(EW)** 应用程序流量 - kube 代理负责 L4 EW 流量。
- 安全性 - 一级 ADC 负责确保 NS 流量的安全。身份验证可以在任一 ADC 进行。EW 流量不受 Kube-代理的保护。添加印花布以进行 L4 分段。
- 可扩展性和性能 - NS 流量具有良好的可扩展性，群集是一种选项。ew 流量和 kube-proxy 可扩展性未知。
- 可观察性 - 第 1 层 ADC 为 NS 流量提供了出色的可观察性，但 EW 流量没有可观察性。

Citrix 服务网精简版

- 南北 **(NS)** 应用程序流量 - 第 1 层 ADC 负责 SSL 卸载、Web App Firewall 和 L4 NS 流量。它适用于整体和 CN 应用。第 2 层 CPX 可管理 k8 和 L7 NS 流量的快速变化。
- 东西部 **(EW)** 应用程序流量 - 第 2 层 CPX 或任何开源代理负责 L4 EW 流量。客户可以选择哪些应用程序使用 CPX，哪些应用程序使用 kube-proxy。
- 安全性 - 一级 ADC 负责确保 NS 流量的安全。身份验证可以在任一 ADC 进行。Citrix CPX 负责身份验证、SSL 卸载和保护 EW 流量。加密可以在应用程序级别应用。
- 可扩展性和性能 - NS 和 EW 流量具有良好的可扩展性，但它增加了 1 个内联跃点。
- 可观察性 - 一级 ADC 提供出色的 NS 流量可观察性。第 2 层中的 CPX 提供了 EW 流量的可观察性，但是可以禁用它以减少 CPX 内存或 CPU 占用空间。

如何部署

Citrix 统一入口

要使用 OpenShift 验证 Citrix 统一入口部署，请使用带有 Citrix ADC VPX 或 MPX 的示例“你好世界”应用程序。OpenShift 的默认命名空间“默认”用于此部署。

1. Citrix ADC 实例是手工构建的，并使用 NSIP/SNIP 进行配置的。可以从 此处 找到在 XenServer 上安装 Citrix ADC。
2. 将以下 YAML 文件示例复制到 OpenShift 目录中，并将其命名为 application.yaml。

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
```



```
  selector:
    matchLabels:
      run: load-balancer-example
  replicas: 2
  template:
    metadata:
      labels:
        run: load-balancer-example
    spec:
      containers:
        -  name: hello-world
          image: gcr.io/google-samples/node-hello:1.0
          ports:
            -  containerPort: 8080
              protocol: TCP
<!--NeedCopy-->
```

3. 部署应用程序。

   ```
   oc apply -f application.yaml
   ```

4. 确保容器正在运行。

   ```
   oc get pods
   ```

5. 将以下 YAML 文件示例复制到 OpenShift 目录中，并将其命名为 service.yaml。

```
apiVersion: v1
kind: Service
metadata:
  name: hello-world-service
spec:
  type: NodePort
  ports:
  -  port: 80
    targetPort: 8080
  selector:
    run: load-balancer-example
<!--NeedCopy-->
```

6. 通过带服务的 NodePort 公开应用程序。

   ```
   oc apply -f service.yaml
   ```

7. 验证服务是否已创建。

   ```
   oc get service
   ```


8. 将以下 YAML 文件示例复制到 OpenShift 目录中，并将其命名为 ingress.yaml。您必须将注释 “ingress.citrix.com/frontend-ip” 更改为可用 IP 地址，以便在 Citrix ADC 上成为 VIP。

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: hello-world-ingress
  annotations:
   kubernetes.io/ingress.class: ”vpx”
   ingress.citrix.com/insecure-termination: ”redirect”
   ingress.citrix.com/frontend-ip: ”10.217.101.183”
spec:
  rules:
  -  host:  helloworld.com
    http:
      paths:
      -  path:
        backend:
          serviceName: hello-world-service
          servicePort: 80
<!--NeedCopy-->
```

9. 部署入口 YAML 文件。

```
oc apply -f ingress.yaml
```

10. 现在，我们已经使用服务公开了应用程序容器，并且可以使用 Ingress 将流量路由到它们。安装 Citrix Ingress Controller (CIC) 以将这些配置推送到我们的第 1 级 ADC VPX。在部署 CIC 之前，请部署一个 RBAC 文件，以便为 CIC 提供正确的运行权限。

> 注意：
>
> rbac yaml 文件指定命名空间，并且必须更改它，等待使用哪个命名空间。

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
rules:
  -  apiGroups: [””]
    resources: [”services”, ”endpoints”, ”ingresses”, ”pods”, ”
       secrets”, ”nodes”, ”routes”, ”routes/status”, ”tokenreviews
       ”, ”subjectaccessreviews”]
    verbs: [”\*”]
```



```
  - apiGroups: [”extensions”]
    resources: [”ingresses”, ”ingresses/status”]
    verbs: [”\*”]
  - apiGroups: [”citrix.com”]
    resources: [”rewritepolicies”]
    verbs: [”\*”]
  - apiGroups: [”apps”]
    resources: [”deployments”]
    verbs: [”\*”]

<!--NeedCopy-->
```

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cpx
subjects:
- kind: ServiceAccount
  name: cpx
  namespace: default

<!--NeedCopy-->
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: cpx
  namespace: default
<!--NeedCopy-->
```

11. 部署 RBAC 文件。

    ```
    oc apply -f rbac.yaml
    ```

12. 在部署 CIC 之前，请编辑 YAML 文件。在规范下，只要在第 1 层 ADC 的 SNIP 上启用了管理，就可以添加 NSIP 或 SNIP。请注意，参数 “入口类” 与入口 YAML 文件中指定的入口类注释相同。


```
apiVersion: v1
kind: Pod
metadata:
  name: hello-world-cic
  labels:
    app: hello-world-cic
spec:
  serviceAccountName: cpx
  containers:
  -  name: hello-world-cic
    image: ”quay.io/citrix/citrix-k8s-ingress-controller:1.1.3”
    env:
     # Set NetScaler NSIP/SNIP, SNIP in case of HA (mgmt has to be
         enabled)
     -  name: ”NS_IP”
       value: ”10.217.101.193”
     # Set username for Nitro
     # Set log level
     -  name: ”NS_ENABLE_MONITORING”
       value: ”NO”
     -  name: ”NS_USER”
       value: ”nsroot”
     -  name: ”NS_PASSWORD”
       value: ”nsroot”
     -  name: ”EULA”
       value: ”yes”
     -  name: ”LOGLEVEL”
       value: ”DEBUG”
    args:
      -  --ingress-classes
        vpx
      -  --feature-node-watch
        false
    imagePullPolicy: IfNotPresent
<!--NeedCopy-->
```

13. 部署 CIC。

    ```
    oc apply -f cic.yaml
    ```

14. 验证所有容器都在运行。

    ```
    oc get pods
    ```

15. 在本地计算机上使用 helloworld.com 的条目和入口 YAML 文件中指定的 Citrix ADC 上的 VIP 编辑主机文件。

16. 在浏览器中导航至 helloworld.com。此时应显示 “Hello Kubernetes!”。

> 注意：以下是删除命令
>
> - `oc delete pods (pod name)-n (namespace name)`
> - `oc delete deployment (deployment name)-n (namespace name)`
> - `oc delete service (service name)-n (namespace name)`
> - `oc delete ingress (ingress name)-n (namespace name)`
> - `oc delete serviceaccounts (serviceaccounts name)-n (namespace name)`

Citrix 2 层入口

要使用 OpenShift 验证 Citrix 2 层入口部署，请使用带有 Citrix ADC VPX 或 MPX 的示例 “你好世界” 应用程序。默认命名空间 “tier-2-adc” 用于此部署。** 注意：部署窗格、服务和入口时，必须使用参数 “-n (namespace name)” 指定命名空间。

1. Citrix ADC 实例是手工构建的，并使用 NSIP/SNIP 进行配置的。您可以在此处找到在 XenServer 上安装 Citrix ADC。如果已配置实例，请清除负载平衡或内容交换中推送到 ADC 的所有虚拟服务器，不要将 Hello-world 部署为统一入口。
2. 创建一个名为 “tier-2-adc” 的命名空间。
   `oc create namespace tier-2-adc`
3. 将以下 YAML 文件示例复制到 OpenShift 目录中并将其命名为 `application-2t.yaml`。

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  selector:
    matchLabels:
      run: load-balancer-example
  replicas: 2
  template:
    metadata:
      labels:
        run: load-balancer-example
    spec:
      containers:
        -  name: hello-world
          image: gcr.io/google-samples/node-hello:1.0
          ports:
            -  containerPort: 8080
              protocol: TCP
```



```
21
22 <!--NeedCopy-->
```

4. 在命名空间中部署应用程序。

```
oc apply -f application-2t.yaml -n tier-2-adc
```

5. 确保容器正在运行。

```
oc get pods
```

6. 将以下 YAML 文件示例复制到 OpenShift 目录中并将其命名为 `service-2t.yaml`。

```
apiVersion: v1
kind: Service
metadata:
  name: hello-world-service-2
spec:
  type: NodePort
  ports:

  -port: 80
    targetPort: 8080
  selector:
    run: load-balancer-example

<!--NeedCopy-->
```

7. 通过带服务的 NodePort 公开应用程序。

```
oc apply -f service-2t.yaml -n tier-2-adc
```

8. 验证服务是否已创建。

```
oc get service -n tier-2-adc
```

9. 将以下 YAML 文件示例复制到 OpenShift 目录中并将其命名为 `ingress-2t.yaml`。

```
    apiVersion: extensions/v1beta1
    kind: Ingress
    metadata:
      name: hello-world-ingress-2
      annotations:
       kubernetes.io/ingress.class: "cpx"
    spec:
      backend:
        serviceName: hello-world-service-2
```



```
        servicePort: 80

<!--NeedCopy-->
```

10. 部署入口 YAML 文件。

```
oc apply -f ingress-2t.yaml -n tier-2-adc
```

11. 部署一个 RBAC 文件，该文件为 CIC 和 CPX 提供正确的运行权限。

> 注意：
>
> rbac yaml 文件指定命名空间，并且必须更改它，等待使用哪个命名空间。

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
rules:
  -apiGroups: [””]
    resources: [”services”, ”endpoints”, ”ingresses”, ”pods”, ”
       secrets”, ”nodes”, ”routes”, ”routes/status”, ”tokenreviews
       ”, ”subjectaccessreviews”]
    verbs: [”\*”]
  -apiGroups: [”extensions”]
    resources: [”ingresses”, ”ingresses/status”]
    verbs: [”\*”]
  -apiGroups: [”citrix.com”]
    resources: [”rewritepolicies”]
    verbs: [”\*”]
  -apiGroups: [”apps”]
    resources: [”deployments”]
    verbs: [”\*”]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cpx
subjects:
-  kind: ServiceAccount
  name: cpx
  namespace: tier-2-adc
```



```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: cpx
  namespace: tier-2-adc
<!--NeedCopy-->
```

12. 部署 RBAC 文件。

    ```
    oc apply -f rbac-2t.yaml
    ```

13. 服务帐户需要提升权限才能创建 CPX。

    ```
    oc adm policy add-scc-to-user privileged system:serviceaccount:tier-2-adc:cpx
    ```

14. 编辑 CPX YAML 文件并将其命名为 `cpx-2t.yaml`。这将部署 CPX 和公开它的服务。请注意，入口类的参数与 `ingress-2t.yaml` 文件中的注释匹配。

```
    apiVersion: extensions/v1beta1
    kind: Deployment
    metadata:
      name: hello-world-cpx-2
    spec:
      replicas: 1
      template:
        metadata:
          name: hello-world-cpx-2
          labels:
            app: hello-world-cpx-2
            app1: exporter
          annotations:
            NETSCALER_AS_APP: "True"
        spec:
          serviceAccountName: cpx
          containers:
            -  name: hello-world-cpx-2
              image: "quay.io/citrix/citrix-k8s-cpx-ingress
                 :13.0-36.28"
              securityContext:
                 privileged: true
              env:
              -  name: "EULA"
              value: "yes"
              -  name: "KUBERNETES_TASK_ID"
```



```
                value: ""
              imagePullPolicy: Always
        # Add cic as a sidecar
            -  name: cic
              image: "quay.io/citrix/citrix-k8s-ingress-controller
                 :1.1.3"
              env:
              -  name: "EULA"
                value: "yes"
              -  name: "NS_IP"
                value: "127.0.0.1"
              -  name: "NS_PROTOCOL"
                value: "HTTP"
              -  name: "NS_PORT"
                value: "80"
              -  name: "NS_DEPLOYMENT_MODE"
                value: "SIDECAR"
              -  name: "NS_ENABLE_MONITORING"
                value: "YES"
              -  name: POD_NAME
                valueFrom:
                  fieldRef:
                    apiVersion: v1
                    fieldPath: metadata.name
              -  name: POD_NAMESPACE
                valueFrom:
                  fieldRef:
                    apiVersion: v1
                    fieldPath: metadata.namespace
              args:
                -  --ingress-classes
                  cpx
              imagePullPolicy: Always
        apiVersion: v1
        kind: Service
        metadata:
          name: lb-service-cpx
          labels:
            app: lb-service-cpx
        spec:
          type: NodePort
          ports:
          -  port: 80
            protocol: TCP
            name: http
```



```
70              targetPort: 80
71            selector:
72              app: hello-world-cpx-2
73
74  <!--NeedCopy-->
```

15. 部署 CPX。

    ```
    oc apply -f cpx-2t.yaml -n tier-2-adc
    ```

16. 验证容器是否正在运行以及服务是否已创建。

    ```
    oc get pods -n tier-2-adc
    oc get service -n tier-2-adc
    ```

17. 创建从 VPX 路由到 CPX 的入口。前端 IP 应该是 ADC 上的空闲 IP。为文件指定一个名称：`ingress-cpx-2t.yaml`。

```
1  apiVersion: extensions/v1beta1
2  kind: Ingress
3  metadata:
4    name: hello-world-ingress-vpx-2
5    annotations:
6     kubernetes.io/ingress.class: "helloworld"
7     ingress.citrix.com/insecure-termination: "redirect"
8     ingress.citrix.com/frontend-ip: "10.217.101.183"
9  spec:
10   rules:
11     -  host:  helloworld.com
12
13     http:
14       paths:
15       -  path:
16         backend:
17           serviceName: lb-service-cpx
18           servicePort: 80
19 <!--NeedCopy-->
```

18. 部署入口。

    ```
    oc apply -f ingress-cpx-2t.yaml -n tier-2-adc
    ```

19. 在部署 CIC 之前，请编辑 YAML 文件。在规范下，只要在第 1 层 ADC 的 SNIP 上启用了管理，就可以添加 NSIP 或 SNIP。


```
apiVersion: v1
kind: Pod
metadata:
  name: hello-world-cic
  labels:
    app: hello-world-cic
spec:
      serviceAccountName: cpx
      containers:
      -  name: hello-world-cic
        image: ”quay.io/citrix/citrix-k8s-ingress-controller:1.1.3
           ”
        env:
         # Set NetScaler NSIP/SNIP, SNIP in case of HA (mgmt has
            to be enabled)
         -  name: ”NS_IP”
       value: ”10.217.101.176”
     # Set username for Nitro
     # Set log level
     -  name: ”NS_ENABLE_MONITORING”
       value: ”NO”
     -  name: ”NS_USER”
       value: ”nsroot”
     -  name: ”NS_PASSWORD”
       value: ”nsroot”
     -  name: ”EULA”
       value: ”yes”
     -  name: ”LOGLEVEL”
       value: ”DEBUG”
    args:
      -  --ingress-classes
        helloworld
      -  --feature-node-watch
        false
    imagePullPolicy: IfNotPresent
<!--NeedCopy-->
```

20. 部署 CIC。

    ```
    oc apply -f cic-2t.yaml -n tier-2-adc
    ```

21. 验证所有容器都在运行。

    ```
    oc get pods -n tier-2-adc
    ```

22. 编辑本地计算机上的主机文件，其中包含一个 Helloworld.com 条目，以及在从 VPX 路由到 CPX 的入口 YAML 文件中指定的 Citrix ADC 上的 VIP。

23. 在浏览器中导航至 helloworld.com。此时应显示 “Hello Kubernetes!”。

Citrix 服务网精简版

服务网格精简版允许引入 CPX（或其他 Citrix ADC 设备）作为内置 HAProxy 功能的替代品。这使我们能够扩展 Kubernetes 的 N/S 功能，并提供 E/W 流量负载平衡、路由和可观察性。
Citrix ADC（MPX、VPX 或 CPX）可以为 E-W 流量提供以下优势，例如：

- 相互 TLS 或 SSL 卸载
- 基于内容的路由，允许或阻止基于 HTTP 或 HTTPS 标头参数的流量
- 高级负载平衡算法（例如，最少连接、最短响应时间等）。
- 通过测量黄金信号（错误、延迟、饱和度或流量量）来观察东西流量。Citrix ADM 的服务图表是监视和调试微服务的可观察性解决方案。
- 在此部署方案中，我们部署 Bookinfo 应用程序并观察其默认工作方式。然后，我们通过撕裂和替换默认的 Kubernetes 服务，并使用 CPX 和 VPX 来代理我们的 E/W 流量。

Citrix 服务网格精简版与 **CPX**

要使用 OpenShift 验证 Citrix 统一入口部署，请使用带有 Citrix ADC VPX 或 MPX 的示例 “你好世界” 应用程序。OpenShift 的默认命名空间 “默认” 用于此部署。

1. Citrix ADC 实例是手工构建的，并使用 NSIP/SNIP 进行配置的。可以从 此处 找到在 XenServer 上安装 Citrix ADC。
2. 为此部署创建命名空间。在此示例中，使用 `sml`。
   ```
   oc create namespace sml
   ```
3. 复制以下 YAML 以创建 Bookinfo 的部署和服务。把它命名为 bookinfo.yaml。

```
1  ##################################################################################
2  # Details service
3  ##################################################################################
4  apiVersion: v1
5  kind: Service
6  metadata:
7    name: details
8    labels:
9      app: details
10     service: details
11 spec:
12   ports:
```



```
13    -  port: 9080
14      name: http
15    selector:
16      app: details
17  ---
18  apiVersion: extensions/v1beta1
19  kind: Deployment
20  metadata:
21    name: details-v1
22    labels:
23      app: details
24      version: v1
25  spec:
26    replicas: 1
27    template:
28      metadata:
29        annotations:
30          sidecar.istio.io/inject: ”false”
31        labels:
32          app: details
33          version: v1
34      spec:
35        containers:
36        -  name: details
37          image: docker.io/maistra/examples-bookinfo-details-v1:0.12.0
38          imagePullPolicy: IfNotPresent
39          ports:
40          -  containerPort: 9080
41  ---
42  ##################################################################################
43  # Ratings service
44  ##################################################################################
45  apiVersion: v1
46  kind: Service
47  metadata:
48    name: ratings
49    labels:
50      app: ratings
51      service: ratings
52  spec:
53    ports:
54    -  port: 9080
55      name: http
```



```
    selector:
      app: ratings
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: ratings-v1
  labels:
    app: ratings
    version: v1
spec:
  replicas: 1
  template:
    metadata:
      annotations:
        sidecar.istio.io/inject: "false"
      labels:
        app: ratings
        version: v1
    spec:
      containers:
      -  name: ratings
        image: docker.io/maistra/examples-bookinfo-ratings-v1:0.12.0
        imagePullPolicy: IfNotPresent
        ports:
        -  containerPort: 9080
---
##################################################################################################

# Reviews service
##################################################################################################

apiVersion: v1
kind: Service
metadata:
  name: reviews
  labels:
    app: reviews
    service: reviews
spec:
  ports:
  -  port: 9080
    name: http
  selector:
    app: reviews
```



```
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: reviews-v1
  labels:
    app: reviews
    version: v1
spec:
  replicas: 1
  template:
    metadata:
      annotations:
        sidecar.istio.io/inject: ”false”
      labels:
        app: reviews
        version: v1
    spec:
      containers:
      -  name: reviews
        image: docker.io/maistra/examples-bookinfo-reviews-v1:0.12.0
        imagePullPolicy: IfNotPresent
        ports:
        -  containerPort: 9080
---
   apiVersion: extensions/v1beta1
   kind: Deployment
   metadata:
     name: reviews-v2
     labels:
       app: reviews
       version: v2
   spec:
     replicas: 1
     template:
       metadata:
         annotations:
           sidecar.istio.io/inject: ”false”
         labels:
           app: reviews
           version: v2
       spec:
         containers:
         -  name: reviews
           image: docker.io/maistra/examples-bookinfo-reviews-v2
```



```
                    :0.12.0
144             imagePullPolicy: IfNotPresent
145             ports:
146             -  containerPort: 9080
147   ---
148     apiVersion: extensions/v1beta1
149     kind: Deployment
150     metadata:
151       name: reviews-v3
152       labels:
153         app: reviews
154         version: v3
155     spec:
156       replicas: 1
157       template:
158         metadata:
159           annotations:
160             sidecar.istio.io/inject: "false"
161           labels:
162             app: reviews
163             version: v3
164         spec:
165           containers:
166           -  name: reviews
167             image: docker.io/maistra/examples-bookinfo-reviews-v3
                    :0.12.0
168             imagePullPolicy: IfNotPresent
169             ports:
170             -  containerPort: 9080
171     ---
172     ##################################################################################

173     # Productpage services
174     ##################################################################################

175     apiVersion: v1
176     kind: Service
177     metadata:
178       name: productpage-service
179     spec:
180       type: NodePort
181       ports:
182       -  port: 80
183         targetPort: 9080
184       selector:
```



```
185         app: productpage
186     ---
187     apiVersion: extensions/v1beta1
188     kind: Deployment
189     metadata:
190       name: productpage-v1
191       labels:
192         app: productpage
193         version: v1
194     spec:
195       replicas: 1
196       template:
197         metadata:
198           annotations:
199             sidecar.istio.io/inject: "false"
200           labels:
201             app: productpage
202             version: v1
203         spec:
204           containers:
205           -  name: productpage
206             image: docker.io/maistra/examples-bookinfo-productpage-v1
                    :0.12.0
207             imagePullPolicy: IfNotPresent
208             ports:
209             -  containerPort: 9080
210     ---
211 <!--NeedCopy-->
```

1. 在 sml 命名空间中部署 bookinfo.yaml。
   `oc apply -f bookinfo.yaml -n sml`
2. 复制并部署映射到产品页面服务的入口文件。此文件可以命名 `ingress-productpage.yaml`。前端 IP 应为 Citrix ADC VPX/MPX 上的免费 VIP。

```
1 apiVersion: extensions/v1beta1
2 kind: Ingress
3 metadata:
4   name: productpage-ingress
5   annotations:
6    kubernetes.io/ingress.class: "bookinfo"
7    ingress.citrix.com/insecure-termination: "redirect"
8    ingress.citrix.com/frontend-ip: "10.217.101.182"
9 spec:
```



```
  rules:
  -  host: bookinfo.com
    http:
      paths:
      -  path:
        backend:
          serviceName: productpage-service
          servicePort: 80
<!--NeedCopy-->
```

```
oc apply -f ingress-productpage.yaml -n sml
```

1. 在 sml 命名空间中为 RBAC 文件复制以下 YAML 并将其部署。在产品页面微服务前面用于 CIC 的文件命名为 `rbac-cic-pp.yaml`。

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
rules:
  -  apiGroups: [””]
    resources: [”services”, ”endpoints”, ”ingresses”, ”pods”, ”
       secrets”, ”routes”, ”routes/status”, ”nodes”, ”namespaces”]
    verbs: [”\*”]
  -  apiGroups: [”extensions”]
    resources: [”ingresses”, ”ingresses/status”]
    verbs: [”\*”]
  -  apiGroups: [”citrix.com”]
    resources: [”rewritepolicies”, ”vips”]
    verbs: [”\*”]
  -  apiGroups: [”apps”]
    resources: [”deployments”]
    verbs: [”\*”]
  -  apiGroups: [”apiextensions.k8s.io”]
    resources: [”customresourcedefinitions”]
    verbs: [”get”, ”list”, ”watch”]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: cpx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
```



```
  name: cpx
subjects:
-  kind: ServiceAccount
  name: cpx
  namespace: sml
apiVersion: rbac.authorization.k8s.io/v1
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: cpx
  namespace: sml
<!--NeedCopy-->
```

```
oc apply -f rbac-cic-pp.yaml -n sml
```

1. 提升服务帐户权限以部署 CIC 和 CPX。
   ```
   oc adm policy add-scc-to-user privileged system:serviceaccount:sml:cpx
   ```
2. 编辑本地计算机上的主机文件，其中 bookinfo.com 映射到在 `ingress-productpage.yaml` 中指定的前端 IP。
3. 使用 CIC 复制和部署产品页面。命名文件 `cic-productpage.yaml`。NS_IP 应为第 1 层 ADC 的 NS_IP。

```
apiVersion: v1
kind: Pod
metadata:
  name: productpage-cic
  labels:
    app: productpage-cic
spec:
      serviceAccountName: cpx
      containers:
      -  name: productpage-cic
        image: ”quay.io/citrix/citrix-k8s-ingress-controller:1.1.3
           ”
        env:
         # Set NetScaler NSIP/SNIP, SNIP in case of HA (mgmt has
            to be enabled)
         -  name: ”NS_IP”
           value: ”10.217.101.176”
         # Set username for Nitro
         # Set log level
         -  name: ”NS_ENABLE_MONITORING”
           value: ”NO”
```



```
20            -  name: "NS_USER"
21               value: "nsroot"
22            -  name: "NS_PASSWORD"
23               value: "nsroot"
24            -  name: "EULA"
25               value: "yes"
26            -  name: "LOGLEVEL"
27               value: "DEBUG"
28            -  name: "NS_APPS_NAME_PREFIX"
29               value: "BI-"
30          args:
31            -  --ingress-classes
32               bookinfo
33            -  --feature-node-watch
34               false
35          imagePullPolicy: IfNotPresent
36  <!--NeedCopy-->
```

```
oc apply -f cic-productpage.yaml -n sml
```

1. 导航至 Bookinfo.com 并单击普通用户。产品页面应提供其他微服务的详细信息、评论和评级。HAProxy 负责在微服务（东西）之间路由流量。
2. 删除详细信息前的服务。刷新 Bookinfo 网页，注意产品页面无法拉取微服务以获取详细信息。
   ```
   oc delete service details -n sml
   ```
3. 复制并部署无头服务，以便从产品页面到详细信息的流量通过 CPX。调用此文件 detailsheadless.yaml。

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: details
5  spec:
6    ports:
7    -  port: 9080
8      name: http
9    selector:
10     app: cpx
11 <!--NeedCopy-->
```

```
oc apply -f detailsheadless.yaml -n sml
```

1. 复制和部署一个新的详细信息服务，该服务应命名为 detailsservice.yaml，以为与详细信息微服务前面。


```
apiVersion: v1
kind: Service
metadata:
  name: details-service
  labels:
    app: details-service
    service: details-service
spec:
  clusterIP: None
  ports:
  -  port: 9080
    name: http
  selector:
    app: details
<!--NeedCopy-->
```

```
oc apply -f detailsservice.yaml -n sml
```

1. 使用入口公开详细信息服务并部署它。调用此文件 detailsingress.yaml。

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: details-ingress
  annotations:
   kubernetes.io/ingress.class: "cpx"
   ingress.citrix.com/insecure-port: "9080"
spec:
  rules:
  -  host: details
    http:
      paths:
      -  path:
        backend:
          serviceName: details-service
          servicePort: 9080
<!--NeedCopy-->
```

```
oc apply -f detailsingress.yaml -n sml
```

1. 复制并部署 CPXEastWest.yaml 文件。


```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: cpx
  labels:
    app: cpx
    service: cpx
spec:
  replicas: 1
  template:
    metadata:
      name: cpx
      labels:
        app: cpx
        service: cpx
      annotations:
        NETSCALER_AS_APP: "True"
    spec:
      serviceAccountName: cpx
      containers:
        - name: reviews-cpx
          image: "quay.io/citrix/citrix-k8s-cpx-ingress:13.0-36.28
              "
          securityContext:
             privileged: true
          env:
          - name: "EULA"
            value: "yes"
          - name: "KUBERNETES_TASK_ID"
            value: ""
          - name: "MGMT_HTTP_PORT"
            value: "9081"
          ports:
          - name: http
            containerPort: 9080
          - name: https
            containerPort: 443
          - name: nitro-http
            containerPort: 9081
          - name: nitro-https
            containerPort: 9443
          imagePullPolicy: Always
        # Add cic as a sidecar
        - name: cic
          image: "quay.io/citrix/citrix-k8s-ingress-controller
```



```
                :1.2.0”
45            env:
46            -  name: ”EULA”
47              value: ”yes”
48            -  name: ”NS_IP”
49              value: ”127.0.0.1”
50            -  name: ”NS_PROTOCOL”
51              value: ”HTTP”
52            -  name: ”NS_PORT”
53              value: ”80”
54            -  name: ”NS_DEPLOYMENT_MODE”
55              value: ”SIDECAR”
56            -  name: ”NS_ENABLE_MONITORING”
57              value: ”YES”
58            -  name: POD_NAME
59              valueFrom:
60                fieldRef:
61                  apiVersion: v1
62                  fieldPath: metadata.name
63            -  name: POD_NAMESPACE
64              valueFrom:
65                fieldRef:
66                  apiVersion: v1
67                  fieldPath: metadata.namespace
68            args:
69              -  --ingress-classes
70                cpx
71            imagePullPolicy: Always
72  <!--NeedCopy-->
```

```
oc apply -f CPXEastWest.yaml -n sml
```

1. 刷新 bookinfo.com，细节应该从细节中提取。CPX 已成功部署到代理 EW 流量。

带有 **VPX/MPX** 的 **Citrix** 服务网格精简版

1. 运行以下命令删除用作 EW 代理的 CPX。部署新文件以将 VPX 配置为产品页面和细节微服务之间的 EW 代理。

   ```
   oc delete -f detailsheadless.yaml -n sml
   oc delete -f detailsservice.yaml -n sml
   oc delete -f detailsingress.yaml -n sml
   oc delete -f CPXEastWest.yaml -n sml
   ```

2. 复制并部署服务，命名文件 detailstoVPX.yaml，以便将流量从产品页面发送回 VPX。IP 参数应该是 Citrix ADC VPX/MPX 上的免费 VIP。


```
---
kind: ”Service”
apiVersion: ”v1”
metadata:
  name: ”details”
spec:
  ports:
    -
      name: ”details”
      protocol: ”TCP”
      port: 9080
---
kind: ”Endpoints”
apiVersion: ”v1”
metadata:
  name: ”details”
subsets:
  -
    addresses:
      -
        ip: ”10.217.101.182” # Ingress IP in MPX
    ports:
      -
        port: 9080
        name: ”details”
<!--NeedCopy-->
```

```
oc apply -f detailstoVPX.yaml -n sml
```

1. 在详细信息微服务前面重新部署 detailsservice.yaml。
   ```
   oc apply -f detailsservice.yaml -n sml
   ```
2. 复制并部署入口以向 VPX 公开细节微服务。此文件命名为 `detailsVPXingress.yaml`。前端 IP 应与第 1 层 ADC 上的 VIP 匹配。

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: details-ingress
  annotations:
   kubernetes.io/ingress.class: ”vpx”
   ingress.citrix.com/insecure-port: ”9080”
```



```
    ingress.citrix.com/frontend-ip: ”10.217.101.182”
spec:
  rules:
  -  host: details
    http:
      paths:
      -  path:
        backend:
          serviceName: details-service
          servicePort: 9080
<!--NeedCopy-->
```

```
oc apply -f detailsVPXingress.yaml
```

1. 刷新 bookinfo.com 和细节应该从细节微服务拉. VPX 已成功部署到代理 EW 流量。

使用 **Openshift** 中的路由或入口类迁移到 **Citrix ADC** 的服务

使用路由分片的服务迁移

Citrix Ingress Controller (CIC) 充当路由器，并将流量重定向到各种容器，以便在各种可用容器之间分配传入流量。

此迁移过程还可以是群集升级过程的一部分，从传统的 Openshift 拓扑到使用 Citrix CNC、CIC 和 CPX 组件进行群集迁移和升级的自动化部署。

这个解决方案可以通过两种方法来实现：

- 插件中的 CIC 路由器 (Pod)
- CPX 路由器内部开路（侧车）

下面介绍了这两种方法以及迁移示例。

静态路由（默认） -通过静态路由将 Openshift 主机子网映射到外部 ADC。

静态路由在使用 HAProxy 的传统 Openshift 部署中很常见。在将服务从一个服务代理迁移到另一个服务代理时，可以在 parralell 中与 Citrix CNC、CIC 和 CPX 一起使用静态路由，而不会中断正常运行的群集中部署的命名空间。

Citrix ADC 的静态路由配置示例：

```
oc 获取主机子网（开放转换群集）代码片段
oc311-master.example.com 10.x.x.x 10.128.0.0/23
oc311-node1.example.com 10.x.x.x 10.130.0.0/23
oc311-node2.example.com 10.x.x.x 10.129.0.0/23

show route (external Citrix VPX) snippet
10.128.0.0 255.255.254.0 10.x.x.x STATIC
```



```
10.129.0.0 255.255.254.0 10.x.x.x STATIC
10.130.0.0 255.255.254.0 10.x.x.x STATIC
```

自动路由 -使用 CNC（Citrix 节点控制器）自动执行到定义路径分片的外部路由。

您可以通过两种方式将 Citrix ADC 与 OpenShift 集成，这两种方式都支持 OpenShift 路由器分片。

路由类型

- 不安全-外部负载均衡器到 CIC 路由器，HTTP 流量不会被加密。
- 安全边缘-外部负载均衡器到 CIC 路由器终止 TLS。
- 安全直通-外部负载均衡器到目的地终止 TLS
- 安全重新加密-外部负载均衡器到 CIC 路由器终止 TLS。使用 TLS 将 CIC 路由器加密到目的地。

部署示例 **#1**：作为 **OpenShift** 路由器插件部署的 **CIC**

对于路径，我们使用路径分片概念。在这里，CIC 充当路由器，并将流量重定向到各种容器，以便在各种可用容器之间分配传入流量。CIC 作为部署在群集之外的 Citrix ADC MPX 或 VPX 的路由器插件进行安装。

Citrix 组件：

- VPX-向 DNS 呈现群集服务的入口 ADC。
- CIC-通过 CNC 路由向外部 Citrix ADC 提供路由标签和命名空间标签。

路由分片的 **YAML** 文件参数示例：

Citrix 开放转移源文件位于 Github 这里

1. 使用 kubernetes 标签格式的值添加以下环境变量：路由标签和命名空间标签。CIC 名称空间标签中的路由分片表达式是一个可选字段。如果使用，它必须与 route.yaml 文件中提到的命名空间标签匹配。

```
        env:
         - name: "ROUTE_LABELS"
           value: "name=apache-web"
         - name: "NAMESPACE_LABELS"
           value: "app=hellogalaxy"

<!--NeedCopy-->
```

1. 通过 route.yaml 创建的路由将具有与 CIC 中的路由分片表达式匹配的标签将得到配置。

```
        metadata:
            name: apache-route
```



```
            namespace: hellogalaxy
            labels:
                name: apache-web
<!--NeedCopy-->
```

1. 使用 service.yaml 公开该服务。

```
        metadata:
            name: apache-service
        spec:
        type: NodePort
        #type=LoadBalancer
        ports:
            - port: 80
        targetPort: 80
        selector:
            app: apache
<!--NeedCopy-->
```

1. 部署一个简单的 Web 应用程序，其中包含与 service.yaml 中的选择器标签匹配。

部署示例 **#2**：作为 **OpenShift** 路由器部署的 **Citrix ADC CPX**

Citrix ADC CPX 可以与群集内的 Citrix Ingress Controller 一起部署为 OpenShift 路由器。有关将 CPX 或 CIC 部署为群集中的路由器的更多步骤，请参阅 通过 Citrix ADC 启用 OpenShift 路由分片支持。

Citrix 组件：

- VPX-向 DNS 呈现群集服务的入口 ADC。
- CIC-向外部 Citrix ADC 提供路由标签和命名空间标签以定义路由分片。
- CNC-提供分片到外部负载均衡器的自动路由配置。
- CPX-在开放换班群集内提供开放换班路由。

带入口类注释的服务迁移

入口类注释使用入口类注释概念，我们向入口类别信息添加注释，这将有助于将流量从外部 ADC 重定向到特定容器/节点。

入口类的 **YAML** 文件参数示例：**

Citrix 入口源文件位于 Github 这里


```
    env:
        args:
          - --ingress-classes
            vpx
<!--NeedCopy-->
```

入口配置文件还应该在元数据中有一个 kubernetes.io/ingress.class 注释字段，该字段将在创建时与 CIC 入口类参数字段匹配。

使用 **“ingress.classes”** 示例进入 **VPX** 部署示例 **

```
        kind: Ingress
            metadata:
                name: ingress-vpx
                annotations:
                    kubernetes.io/ingress.class: ”vpx”
<!--NeedCopy-->
```

Citrix 指标导出器

您可以使用 Citrix ADC 指标导出器和 Prometheus-Operator 来监视 Citrix ADC VPX 或 CPX 入口设备和 Citrix ADC CPX（东西）设备。请参阅 使用 Prometheus 和 Grafana 查看 Citrix ADC 的指标。

Citrix ADC 和 Microsoft Azure 验证的参考设计

May 20, 2020

Microsoft Azure 上的 Citrix ADC 可确保组织能够访问在云中部署的安全和优化的应用程序和资产，并提供灵活性，以建立适应环境不断变化的需求的网络基础。此经验证的设计指导组织了解如何在 Azure 中配置前端自动缩放，以便可靠且经济高效地交付应用程序。

概述 Citrix ADC VPX

Citrix ADC 是一款一体化应用程序交付 Controller，可加快内部和外部 Web 应用程序的性能。该设备可降低应用程序拥有成本，优化用户体验，并通过使用以下方式确保应用程序始终可用：

- 高级第 4-7 层服务负载平衡和流量管理
- 经验证的应用程序加速，例如 HTTP 压缩和缓存

- 集成的应用程序防火墙可确保应用程序安全
- 服务器卸载以显著降低成本并整合服务器

作为服务和应用程序交付领域无可争议的领导者，Citrix ADC 部署在全球数千个网络中。利用 Citrix ADC 优化、保护和控制企业和云服务的交付。设备直接部署在 Web 服务器和数据库服务器之前。Citrix ADC 将高速负载平衡和内容交换、HTTP 压缩、内容缓存、SSL 加速、应用程序流可视性和功能强大的应用程序防火墙集成到一个易于使用的集成平台中。通过端到端监视将网络数据转换为可操作的商业智能来满足 SLA 要简单得多。Citrix ADC 允许使用简单的声明性策略引擎来定义和管理策略，无需编程专业知识。

Microsoft Azure 中的 Citrix ADC 概述

Citrix ADC VPX 虚拟设备可作为映像在 Microsoft Azure 应用商店中提供。Microsoft Azure Resource Manager (ARM) 上的 Citrix ADC VPX 使客户能够使用 Azure 云计算功能并应用 Citrix ADC 负载平衡和流量管理功能以满足其业务需求。您可以将 Citrix ADC VPX 实例作为独立实例或作为主动-主动或主动-备用模式下的高可用性对部署 ARM 上。

局限性与用法指南

- Azure 体系结构不支持以下功能：
  - 群集
  - IPv6
  - Gratuitous ARP (GARP)
  - L2 模式
  - 已标记的 VLAN
  - 动态路由
  - 虚拟 MAC (vMAC)
  - USIP
  - CloudBridge Connector
- 不支持 Intranet IP (IIP) 功能，因为 Azure 不提供此功能所需的 IP 地址池。IIP 常用于 VOIP、SIP 或“服务器启动连接”部署中。
- 如果您希望可能需要随时关闭并临时取消分配 Citrix ADC VPX 虚拟机，请在创建虚拟机时分配静态内部 IP 地址。如果不分配静态内部 IP 地址，Azure 可能会在每次重新启动时为虚拟机分配一个不同的 IP 地址，并且虚拟机可能会变得无法访问。
- 在 Azure 部署中，仅支持以下 Citrix ADC VPX 型号：VPX 10、VPX 200 和 VPX 1000。这些虚拟设备可以部署在具有两个或更多核心以及 2GB 以上内存的任何实例类型中。请参阅Citrix ADC VPX 数据手册。
- Azure 在虚拟机预配期间生成的 部署 **ID** 对 ARM 中的用户不可见。您不能使用部署 ID 在 ARM 上部署 Citrix ADC VPX 设备。

用例

与需要将每个服务作为单独的虚拟设备部署的替代解决方案相比，Citrix ADC on Azure 将基本的应用程序交付功能结合在一个 VPX 实例中。这包括 L4 负载平衡、L7 流量管理、服务器卸载、应用程序加速、应用程序安全以及通过 Azure 应用商店方便提供的其他服务。此外，一切都有一个单一的政策框架。Citrix ADC 使用用于管理本地 Citrix ADC 部署的功能强大的工具集进行管理。最终结果是，Azure 上的 Citrix ADC 启用了几个引人注目的使用案例。Citrix ADC 不仅支持当今企业的迫切需求，还支持从传统计算基础架构向企业云数据中心的持续演变。

生产交付

许多企业积极采用 Azure 作为基础架构即服务 (IaaS) 产品，用于生产应用程序交付。现在，企业可以使用世界上最大的网站和云服务提供商所使用的相同云网络平台来前端这些应用程序。可以利用广泛的卸载、加速和安全功能来提高性能并降低成本。

混合云设计

借助 Azure 上的 Citrix ADC，跨企业数据中心并扩展到 Azure 的混合云可以受益于相同的 Citrix ADC 云网络平台，从而大大简化了应用程序和工作负载在私有数据中心和 Azure 之间来回转换的过程。Citrix ADC 在 Azure 上可以充分利用 Citrix ADC 的全套功能，从使用数据流的智能数据库负载平衡到使用 AppFlow® 前所未有的应用程序可见性，以及使用操作分析进行实时监视和响应。

业务连续性

希望使用 Azure 作为其灾难恢复和业务连续性计划一部分的企业可以依赖 Citrix ADC 全局服务器负载平衡，以持续监视企业数据中心和 Azure 环境的可用性和性能，从而确保用户始终被发送到最佳位置。

开发和测试

在本地运行生产交付但使用 Azure 进行开发和测试的企业现在可以将 Citrix ADC 包含在其 Azure 测试环境中，这样可以由于在其测试环境中更好地模拟生产实施，从而加快生产速度。在每个用例中，网络体系结构师还可以利用 Citrix CloudBridge（配置为独立实例或 Citrix ADC 白金版实例的功能）来保护和优化企业数据中心与 Azure 云之间的连接，从而加快数据传输/同步速度和最小化网络成本

网络体系结构

在 ARM 中，Citrix ADC VPX 虚拟机 (VM) 驻留在虚拟网络中。默认情况下，在 Azure 中预配的 Citrix ADC VPX 以下一节所述单个 IP 模式运行。

将在每个 Citrix ADC 虚拟机上创建一个虚拟 NIC。在虚拟网络中配置的网络安全组绑定到 NIC。它们一起控制流入虚拟机和流出虚拟机的流量。

网络安全组将请求转发到 Citrix ADC VPX 实例，VPX 实例将这些请求发送到服务器。来自服务器的响应相反地遵循相同的路径。您可以将网络安全组配置为控制单个 VPX VM，或者控制子网和虚拟网络，并控制多个 VPX VM 部署中的流量。

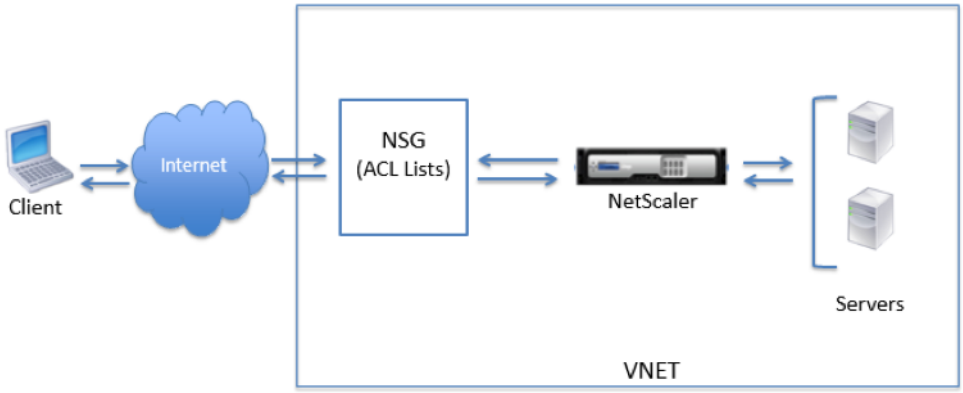
NIC 包含网络配置详细信息，如虚拟网络、子网、内部 IP 地址和公用 IP 地址。

在 ARM 中操作时，最好知晓用于访问 VM 的以下 IP 地址：

- 公有 IP (PIP) 地址是直接在 Citrix ADC 虚拟机的虚拟 NIC 上配置的面向 Internet 的 IP 地址。PIP 允许您从外部网络直接访问 VM，而无需在网络安全组上配置入站和出站规则。
- Citrix ADC IP (NSIP) 地址是在 VM 上配置的内部 IP 地址。该地址不可路由。
- 虚拟 IP 地址 (VIP) 是使用 NSIP 和端口号配置的。客户端通过 PIP 地址访问 Citrix ADC 服务，当请求到达 Citrix ADC VPX 虚拟机或 Azure 负载平衡器的 NIC 时，VIP 将被转换为内部 IP (NSIP) 和内部端口号。
- 内部 IP 地址是虚拟网络地址空间池中虚拟机的私有内部 IP 地址。此 IP 地址无法从外部网络进行访问。此 IP 地址默认是动态的，除非您将其设置为静态。根据在网络安全组上创建的规则，来自 Internet 的流量将路由到此地址。网络安全组和 NIC 有选择地将正确类型的流量发送到 NIC 上的右端口，这取决于在 VM 上配置的服务。

> 注意：
>
> 在本文档中，PIP、VIP 和实例级 PIP (ILPIP) 的含义相同，可互换使用。

下图显示了流量如何通过在 ARM 中预配的 Citrix ADC VPX 实例从客户端流向服务器。



Citrix ADC VPX 在 **Azure** 上的工作原理

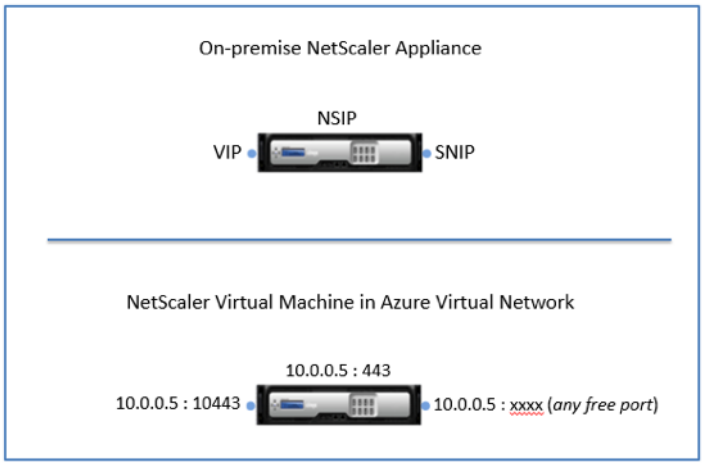
在本地部署中，Citrix ADCVPX 实例至少需要三个 IP 地址：

- 管理 IP 地址，称为 Citrix ADC IP (NSIP) 地址
- 子网 IP (SNIP) 地址，用于与服务器场通信
- 虚拟服务器 IP (VIP) 地址，用于接收客户端请求

在 Azure 部署中，在通过 DHCP 预配期间，仅为实例分配一个 IP 地址（私有（内部）地址）。

为避免此限制，可以使用单个 IP 体系结构在 Azure 中部署 Citrix ADC VPX 实例。这样，Citrix ADC 设备的三个 IP 功能可以多路复用到一个 IP 地址上。此单一 IP 地址使用不同的端口号来执行 NSIP、SNIP 和 VIP 功能。

下图说明如何使用单一 IP 地址来执行 NSIP、SNIP 和 VIP 的功能。

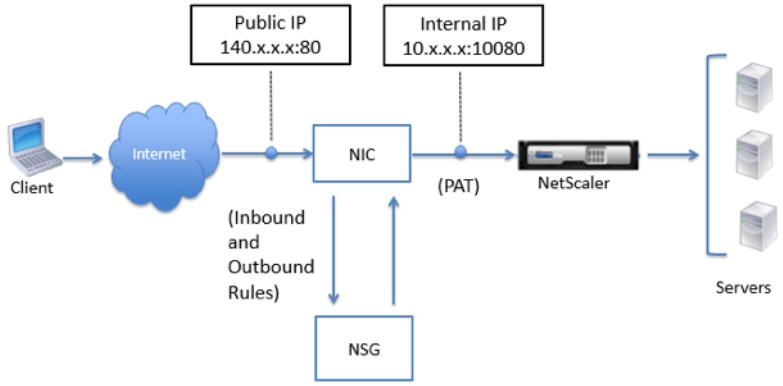


通过端口地址转换的流量

在 Azure 部署中，将 Citrix ADC VPX 实例预配为虚拟机 (VM) 时，Azure 会向 Citrix ADC 虚拟机分配公有 IP 地址和内部 IP 地址（不可路由）。在 Citrix ADC 实例的网络安全组中定义入站和出站规则，以及每个定义的规则的公共端口和专用端口。Citrix ADC 实例侦听内部 IP 地址和专用端口。

在 Citrix ADC VPX 虚拟机的虚拟 NIC 上接收任何外部请求。网卡绑定到网络安全组，该组指定转换请求的目标地址和端口（公共 IP 地址和端口）的专用 IP 和专用端口组合。ARM 执行端口地址转换 (PAT) 以将公有 IP 地址和端口映射到 Citrix ADC 虚拟机的内部 IP 地址和专用端口。最后，ARM 然后将流量转发到 VM。

下图显示了 Azure 如何执行 PAT 以将流量定向到 Citrix ADC 内部 IP 地址和专用端口。



在此示例中，为虚拟机分配的公有 IP 地址为 140.x.x.x，内部 IP 地址为 10.x.x.x。定义入站和出站规则时，公共 HTTP 端口 80 定义为接收客户端请求的端口。相应的专用端口 10080 被定义为 Citrix ADC 虚拟机侦听的端口。客户端请求在公用 IP 地址 140.x.x.x 和端口 80 上接收。Azure 执行 PAT 以将此地址和端口映射到专用端口 10080 上的内部 IP 地址 10.x.x.x 并转发客户端请求。

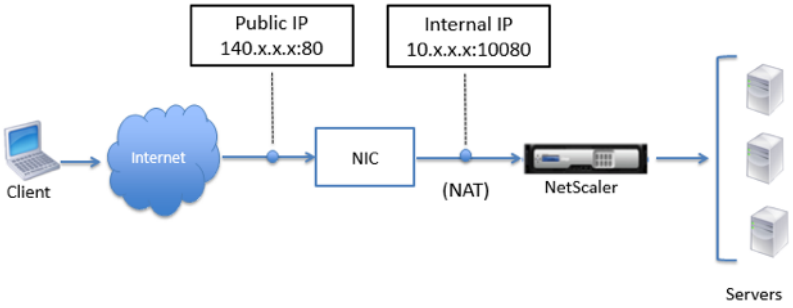
有关端口使用准则的信息，请参阅 端口用法指南 部分。

有关网络安全组和访问控制列表的信息，请单击 此处。

通过网络地址转换的流量

您还可以为 Citrix ADC 虚拟机（实例级别）请求公有 IP (PIP) 地址。如果您在 VM 级别使用此直接 PIP，则无需定义入站和出站规则来拦截网络流量。来自 Internet 的传入请求直接在虚拟机上接收。Azure 执行网络地址转换 (NAT) 并将流量转发到 Citrix ADC 实例的内部 IP 地址。

下图显示了 Azure 如何执行网络地址转换以映射 Citrix ADC 内部 IP 地址。



在此示例中，分配给网络安全组的公有 IP 为 140.x.x.x，内部 IP 地址为 10.x.x.x。定义入站和出站规则时，公共 HTTP 端口 80 定义为接收客户端请求的端口。相应的专用端口 10080 被定义为 Citrix ADC 虚拟机侦听的端口。客户端请求在公用 IP 地址 (140.x.x.x) 上接收。Azure 执行网络地址转换，以将 PIP 映射到端口 10080 上的内部 IP 地址 10.x.x.x，并转发客户端请求。

> 注意：
>
> 高可用性的 Citrix ADC VPX 虚拟机由外部或内部负载平衡器控制。这些负载平衡器具有定义的入站规则，以控制负载均衡流量。首先，外部流量被这些负载平衡器拦截。然后根据配置的负载平衡规则转移流量。这包括在负载平衡器上定义的后端池、NAT 规则和运行状况探测。

分配多个 **IP** 地址

Azure 虚拟机 (VM) 附加了一个或多个网络接口 (NIC)。任何 NIC 都可以为其分配一个或多个静态或动态公共和私有 IP 地址。为虚拟机分配多个 IP 地址可实现以下功能：

- 在一台服务器上托管多个具有不同 IP 地址和 SSL 证书的网站或服务。
- 用作网络虚拟设备，例如防火墙或负载平衡器。
- 将任何 NIC 的任何私有 IP 地址添加到 Azure 负载平衡器后端池的功能。过去，只能将主网卡的主 IP 地址添加到后端池中。要了解有关如何对多个 IP 配置进行负载平衡的详细信息，请阅读 负载平衡多个 IP 配置文章。

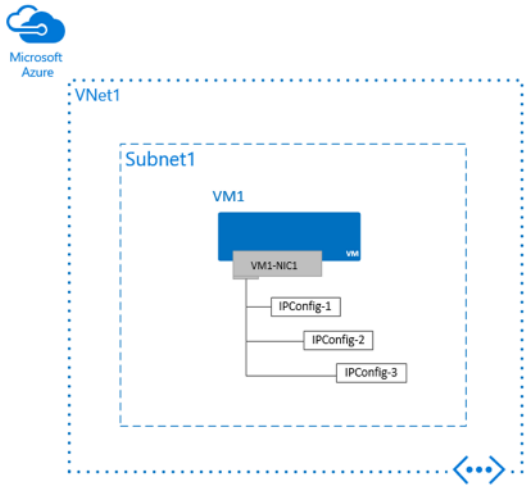
情景

创建具有单个 NIC 的 VM 并将其连接到虚拟网络。虚拟机需要三个不同的私有 IP 地址和两个公有 IP 地址。

IP 地址分配给以下 IP 配置：

- IPConfig-1：分配动态专用 IP 地址（默认）和静态公用 IP 地址。
- IPConfig-2：分配静态专用 IP 地址和静态公用 IP 地址。

- IPConfig-3：分配动态专用 IP 地址，而不分配公用 IP 地址。



连接到虚拟机的每个 NIC 都有一个或多个 IP 配置与之关联。每个配置都分配一个静态或动态私有 IP 地址。每个配置还可能有一个与其关联的公有 IP 地址资源。公有 IP 地址资源具有分配给它的动态或静态公有 IP 地址。若要了解有关 Azure 中 IP 地址的详细信息，请阅读 Azure 文章中的 IP 地址。您最多可以为每个 NIC 分配 250 个私有 IP 地址。虽然您可以为每个 NIC 分配多个公有 IP 地址，但在 Azure 订阅中可以使用的公有 IP 地址数量有限制。有关详细信息，请参阅 Azure 限制文章。

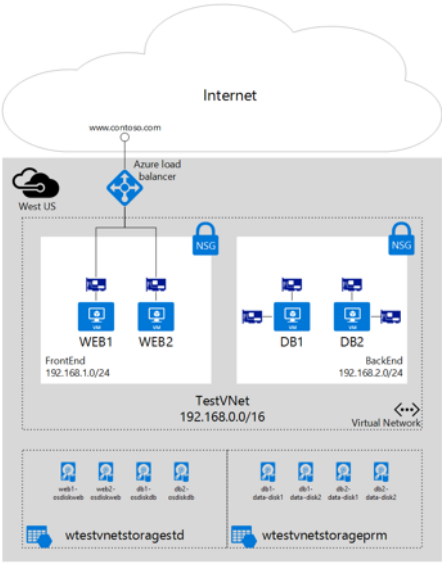
> 注意：
>
> 多个 IP 地址不能分配给通过传统部署模型创建的资源。

创建具有多个 **NIC** 接口的虚拟机

您可以在 Azure 中创建虚拟机 (VM)，并将多个网络接口 (NIC) 附加到每个虚拟机。多 NIC 是许多网络虚拟设备的要求，如应用程序交付和 WAN 优化解决方案。多 NIC 还提供更多的网络流量管理功能。示例包括隔离前端 NIC 和后端 NIC 之间的流量，以及将数据平面流量与管理平面流量分开。

情景

本文档演示了在特定场景中使用虚拟机中的多个 NIC 的部署。在这种情况下，你有一个双层 IaaS 工作负载托管在 Azure 中。每个层都部署在虚拟网络 (VNet) 中自己的子网中。前端层由多个 Web 服务器组成，分组在负载平衡器集中，以实现高可用性。后端层由多个数据库服务器组成。这些数据库服务器部署了两个 NIC，一个用于数据库访问，另一个用于管理。该方案还包括网络安全组，用于控制允许进入每个子网的通信量，以及部署中的 NIC。下图显示了此方案的基本体系结构。



端口使用指南

在创建 Citrix ADC 虚拟机时或预配虚拟机后，可以在网络安全组中配置其他入站和出站规则。每个入站和出站规则都与一个公用端口和一个专用端口相关联。

在配置网络安全组规则之前，请注意以下有关可以使用的端口号的准则：

1. 以下端口由 Citrix ADC 虚拟机保留。在对来自 Internet 的请求使用公用 IP 地址时，不能将这些端口定义为专用端口。

   端口 21、22、80、443、8080、67、161、179、500、520、3003、3008、3009、3010、3011、4001、5061、9000、7000。

   但是，如果您希望面向 Internet 的服务（如 VIP）使用标准端口（例如端口 443），则必须使用网络安全组创建端口映射。然后，标准端口将映射到 Citrix ADC 上为此 VIP 服务配置的其他端口。

   例如，VIP 服务可能正在 Citrix ADC 实例上的端口 8443 上运行，但映射到公共端口 443。因此，当用户通过公用 IP 访问端口 443 时，请求被定向到专用端口 8443。

2. 公用 IP 地址不支持动态打开端口映射的协议，例如被动 FTP 或 ALG。
3. Azure 负载平衡器不适用于公有 IP 地址。高可用性不适用于使用与 VPX 实例关联的 PIP 而不是在负载平衡器上配置的 PIP 的流量。有关在 ARM 中配置 Citrix ADC VPX HA 的详细信息，请参阅 Azure 中在高可用性模式下配置 Citrix ADC VPX。
4. 在 Citrix ADC 网关部署中，您无需配置 SNIP 地址，因为当未配置 SNIP 时，NSIP 可用作 SNIP。

> 注意：您必须使用 NSIP 地址和某些非标准端口号来配置 VIP 地址。对于后端服务器上的回调配置，必须指定 VIP 端口号以及 VIP URL（例如，url: 端口）。
> 注意：在 ARM 中，Citrix ADC VPX 虚拟机与两个 IP 地址相关联。即，公用 IP 地址和内部 IP 地址。外部流量连接到 PIP，而内部 IP 地址或 NSIP 是不可路由的。要在 VPX 中配置 VIP，请使用此内部 IP 地址和端口号的组合。

> 示例：如果 VPN 虚拟服务器 FQDN 是 vip.test.com，并且 VPN 虚拟服务器正在端口 8443 上运行，则回调 URL 为：https://vip.test.com:8443。

配置步骤

以下步骤概述了如何配置必要的资源组、安全组、虚拟网络和转换配置，以便具有正常工作的 ADC。

预配资源组

在 Microsoft Azure 门户页面 上，使用用户名和密码登录 Azure Resource Manager 门户。（在 ARM 门户中，单击一个窗格中的选项将在右侧打开一个新窗格。从一个窗格导航到另一个窗格以配置您的设备。）

创建资源组以充当所有资源的容器。使用资源组以组形式部署、管理和监视资源。

创建网络安全组

创建网络安全组以分配入站和出站规则，以控制虚拟网络中的入站和出站流量。通过网络安全组，您可以为单个虚拟机定义安全规则，还可以定义虚拟网络子网的安全规则。

配置虚拟网络和子网

ARM 中的虚拟网络为您的服务提供了一个安全和隔离层。作为同一虚拟网络的一部分的 VM 和服务可以相互访问。

例如，创建一个虚拟网络，其中一个保留的 CIDR 块为 192.168.0.0/16 和两个具有 CIDR 块分别为 192.168.1.0/24 和 192.168.2.0/24 的子网。

1. 在“创建虚拟网络”窗格中，输入以下值，然后单击“创建”。
    - 虚拟网络的名称
    - Address space（地址空间）– 键入虚拟网络的预留 IP 地址块
    - 子网 — 键入第一个子网的名称（您在此步骤稍后创建第二个子网）
    - Subnet address range（子网地址范围）– 键入子网的预留 IP 地址块
    - Resource group（资源组）– 从下拉列表中选择之前创建的资源组

配置第二个子网

1. 从“所有资源”窗格中选择新创建的虚拟网络，然后在“设置”窗格中单击“子网”。
2. 单击 **+** 子网，然后输入以下详细信息来创建第二个子网。
    - 第二个子网的名称
    - Address range（地址范围）– 键入子网的预留 IP 地址块
    - 网络安全组 — 从下拉列表中选择网络安全组

配置存储帐户

ARM IaaS 基础结构存储包括我们能够在其中以 blob、表格、队列和文件格式存储数据的所有服务。还可以使用 ARM 中这些格式的存储数据创建应用程序。

创建存储帐户以存储所有数据

1. 单击 **+** 新建 **>** 存储 **>** 存储帐户。
2. 在“创建存储帐户”窗格中，输入以下详细信息：
    - 帐户的名称
    - 部署模式 — 确保选择资源管理器
    - 帐户类型 — 从下拉列表中选择常规用途
    - 复制 — 从下拉列表中选择本地冗余存储
    - Resource group（资源组）– 从下拉列表中选择新创建的资源组
3. 单击创建。

配置可用性集

可用性集可保证在进行计划内维护或非计划内维护时至少一个 VM 保持启动并运行。同一个“可用性集”下的两个或多个虚拟机放置在不同的故障域上，以实现冗余服务。

1. 单击 **+** 新建并搜索可用性集。
2. 从列表中选择可用性集实体。单击创建。
3. 在“创建可用性集”窗格中，输入以下详细信息：
    - 可用性集的名称
    - Resource group（资源组）– 从下拉列表中选择新创建的资源组
4. 单击“创建”

预配 **Citrix ADC** 实例

在虚拟网络中创建 Citrix ADC VPX 的实例。接下来，从 Azure 应用商店获取 Citrix ADC VPX 映像。使用 Azure Resource Manager 门户创建 Citrix ADC VPX 实例。

在开始创建 Citrix ADC VPX 实例之前，请确保已创建具有实例所在的所需子网的虚拟网络。您可以在虚拟机预配期间创建虚拟网络，但无法灵活创建不同的子网。有关详细信息，请参阅 使用 Azure 门户创建虚拟网络 文章。

可选：配置 DNS 服务器和 VPN 连接以允许虚拟机访问互联网资源。

> 注意： Citrix 建议在预配 Citrix ADC VPX 虚拟机之前创建资源组、网络安全组、虚拟网络和其他实体。这样，网络信息在预配期间可用。

1. 单击 **+** 新建 **>** 网络连接。
2. 单击查看全部，然后在网络窗格中单击 **Citrix ADC VPX** 自带许可证。
3. 单击创建。

   > 注意：作为在 ARM 门户上查找任何实体的快速方法，您还可以在 Azure 应用商店搜索框中键入实体的名称，然后按 <Enter>。在搜索框中键入 **Citrix ADC** 以查找 Citrix ADC 映像。

4. 选择 **Citrix ADC 12.0 VPX** 自带许可证。
5. 填充您的详细信息。
6. 通过验证后，购买并部署我的 Citrix ADC。
7. 建议将 IP 地址设置为静态。

> 注意：确保选择最新的图像。您的 Citrix ADC 映像名称中可能包含发行号。

使用 **PowerShell** 创建具有多个 **IP** 地址的虚拟机

以下步骤说明了如何创建具有多个 IP 地址的示例 VM（如场景中所述）。根据实施所需更改变量名称和 IP 地址类型。

涵盖的配置步骤包括：

1. 创建具有多个 IP 地址的虚拟机
2. 将 IP 地址添加到虚拟机
3. 向 VM 操作系统添加 IP 地址
4. 验证 (Windows)
5. 验证 (Linux)

请参阅下面的 Microsoft Azure 文档：使用 PowerShell 为虚拟机分配多个 IP 地址。

配置 **Citrix ADC** 端口转换

1. 单击虚拟机的网络接口，即 Citrix ADC。
2. 单击您的 网络安全组。
3. 单击入站安全规则。
4. 在安全组中允许进行入站连接的 **SSH** 和 **HTTP** 。

此时，您可以登录到 Citrix ADC 实例并配置 Azure 环境所需的功能和设置。

> 注意：首次登录 Citrix ADC 时，向导可能会要求提供子网 IP 地址。这在 Citrix ADC Azure 实例上不是必需的，因为它们只对所有功能使用一个 IP 地址。出现提示时跳过此步骤，并继续进入默认配置登录页。

Microsoft Azure Resource Manager 门户

应用程序的基础架构通常由许多组件组成，可能是虚拟机、存储帐户和虚拟网络，或者是 Web 应用程序、数据库、数据库服务器和第三方服务。您不会将这些组件视为单独的实体。相反，您将它们视为单个实体的相关部分和相互依存部分。您希望将它们作为组进行部署、管理和监视。Azure Resource Manager 使您可以作为一个组使用解决方案中的资源。您可以在单个协调操作中部署、更新或删除解决方案的所有资源。您可以使用模板进行部署，该模板可以适用于不同的环境，如测试、暂存和生产。资源管理器提供了安全性、审核和标记功能，以帮助您在部署后管理资源。

术语

如果您是 Azure Resource Manager 的新用户，则可能不熟悉一些术语：

- 资源 -可通过 Azure 获得的可管理项目。一些常见的资源是虚拟机、存储帐户、Web 应用程序、数据库和虚拟网络，但还有更多资源。
- 资源组 -包含 Azure 解决方案相关资源的容器。资源组可以包括解决方案的所有资源，或者仅包括要作为一个组管理的资源。您可以根据对您的组织最有意义的内容来决定如何将资源分配给资源组。请参阅资源组。
- 资源提供程序 - 提供可通过资源管理器部署和管理的资源的服务。每个资源提供程序都提供了用于处理部署的资源的操作。一些常见的资源提供程序是 `Microsoft.Compute`，它提供虚拟机资源 `Microsoft.Storage`，它提供存储帐户资源 `Microsoft.Web`，以及提供与 Web 应用程序。请参阅资源提供程序。
- 资源管理器模板 -一个 JavaScript 对象表示法 (JSON) 文件，用于定义要部署到资源组的一个或多个资源。它还定义了部署的资源之间的依赖关系。该模板可用于一致且重复地部署资源。请参阅模板部署。
- 声明式语法 -语法，可以让你声明“这是我打算创建的”，而不必编写编程命令序列来创建它。资源管理器模板是声明语法的示例。在文件中，定义要部署到 Azure 的基础结构的属性。

参考资料：Azure Resource Manager 概述

Citrix ADC 多 NIC 概述

在 Azure 平台上运行的 Citrix ADC 虚拟实例能够将多个虚拟 NIC 附加到独立的虚拟 Citrix ADC 设备。

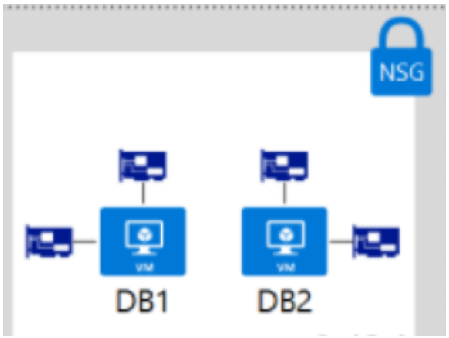
当 Azure 中需要分布式体系结构（如应用程序层和数据库层）时，这是常见的情况。

对于多个 NIC，第二个常见的 Citrix ADC 用例是希望在 Azure 环境中隔离网络区域。隔离的一个例子可以是，允许互联网源流量在一个接口（DMZ 或公共）上终止，而内部 Web 和应用程序服务则为私有。

在此双臂网络方案中，Citrix ADC 虚拟设备需要至少显示两个虚拟 NIC。一个用于公用网络的虚拟 NIC，一个用于专用网络的虚拟 NIC。

此外，Azure 中的多 NIC 配置要求使用多个子网来容纳隔离 NIC。此组件使用 Azure 门户的 VNET 组件进行配置。

示例：多个虚拟网卡连接到 Azure 虚拟机



> 注意：不同的 VM 大小支持不同数量的 NIC，因此相应地调整虚拟机的大小。https://docs.microsoft.com/en-us/azure/virtual-machines/linux/sizes

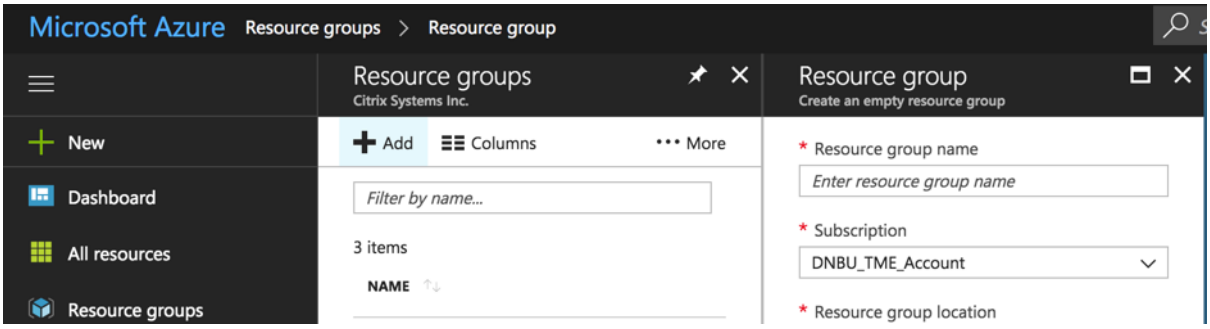
Citrix ADC 多 NIC 配置

以下配置描述了如何轻松创建必要的子网和虚拟网络，以便在 ADC 上具有多 NIC 配置。

创建资源组

PowerShellCopy

```
New-AzureRmResourceGroup -Name ”myResourceGroup” -Location ”EastUS”
<!--NeedCopy-->
```



创建 **VNET** 和子网

常见的情况是虚拟网络具有两个或多个子网。一个子网可以用于前端流量，另一个用于后端流量。要连接到两个子网，然后在 VM 上使用多个 NIC。

使用 `New-AzureRmVirtualNetworkSubnetConfig` 定义两个虚拟网络子网。以下示例定义了 `mySubnetFrontEnd` 和 `mySubnetBackEnd` 的子网：

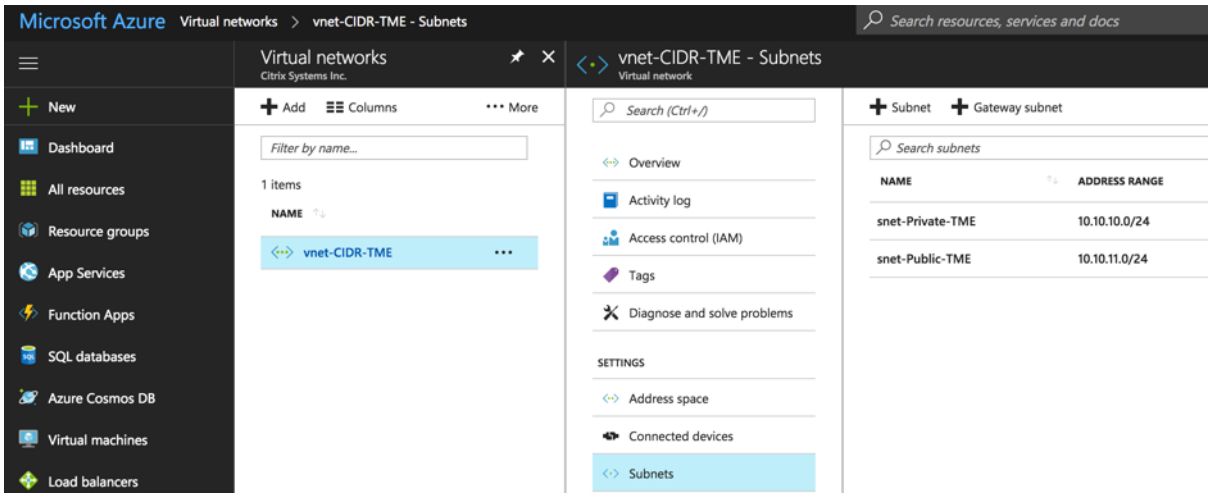
PowerShellCopy


```
$mySubnetFrontEnd = New-AzureRmVirtualNetworkSubnetConfig -Name ”
    mySubnetFrontEnd” `
 -AddressPrefix ”192.168.1.0/24”
$mySubnetBackEnd = New-AzureRmVirtualNetworkSubnetConfig -Name ”
    mySubnetBackEnd” `
 -AddressPrefix ”192.168.2.0/24”
 <!--NeedCopy-->
```

使用 `New-AzureRmVirtualNetwork` 创建虚拟网络和子网。以下示例创建一个名为 myVnet 的虚拟网络：

PowerShellCopy

```
$myVnet = New-AzureRmVirtualNetwork -ResourceGroupName ”myResourceGroup
    ” `
 -Location ”EastUs” `
 -Name ”myVnet” `
 -AddressPrefix ”192.168.0.0/16” `
 -Subnet $mySubnetFrontEnd,$mySubnetBackEnd
 <!--NeedCopy-->
```



创建网络安全组

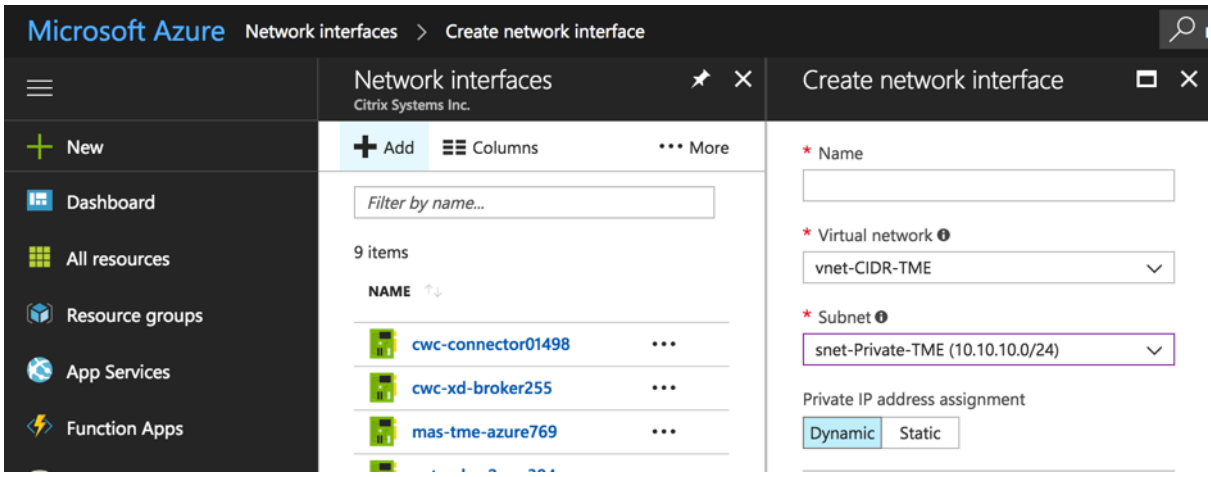
通常，您还创建一个网络安全组来筛选到虚拟机的网络流量，并创建一个负载平衡器以在多个虚拟机之间分配流量。

创建和配置 **vNIC**

使用 `New-AzureRmNetworkInterface` 创建两个 NIC。将一个 NIC 附加到前端子网，将一个 NIC 附加到后端子网。以下示例创建名为 myNic1 和 myNic2 的 NIC：

PowerShellCopy

```
$frontEnd = $myVnet.Subnets|?{
 $\_.Name -eq 'mySubnetFrontEnd' }

$myNic1 = New-AzureRmNetworkInterface -ResourceGroupName ”
    myResourceGroup” `
 -Name ”myNic1” `
 -Location ”EastUs” `
 -SubnetId $frontEnd.Id
$backEnd = $myVnet.Subnets|?{
 $\_.Name -eq 'mySubnetBackEnd' }

$myNic2 = New-AzureRmNetworkInterface -ResourceGroupName ”
    myResourceGroup” `
 -Name ”myNic2” `
 -Location ”EastUs” `
 -SubnetId $backEnd.Id
 <!--NeedCopy-->
```



创建 **VM** 并附加 **vNIC**

现在开始构建您的 VM 配置。每个 VM 大小对可以添加到虚拟机的 NIC 总数都有一个限制。有关详细信息，请参阅 Windows 虚拟机大小。

附加您之前通过 `Add-AzureRmVMNetworkInterface` 创建的两个 NIC：

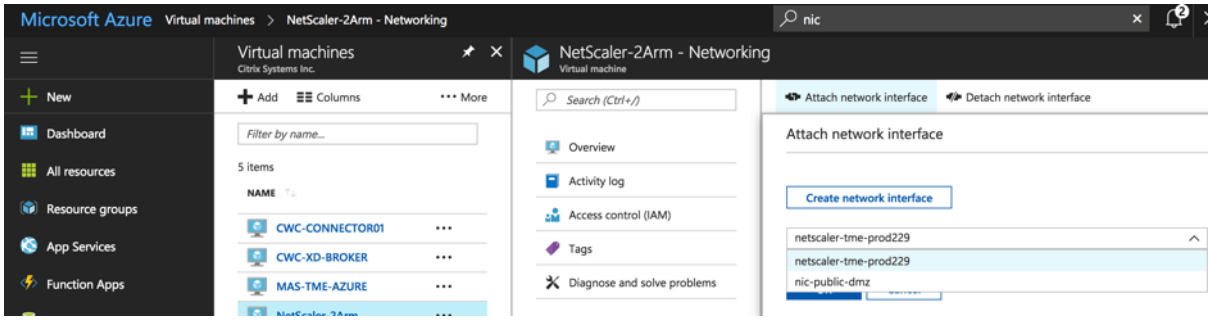
PowerShellCopy


```
$vmConfig = Add-AzureRmVMNetworkInterface -VM $vmConfig -Id $myNic1.Id
    -Primary
$vmConfig = Add-AzureRmVMNetworkInterface -VM $vmConfig -Id $myNic2.Id
<!--NeedCopy-->
```

最后，使用 `New-AzureRmVM` 创建您的 VM：

PowerShellCopy

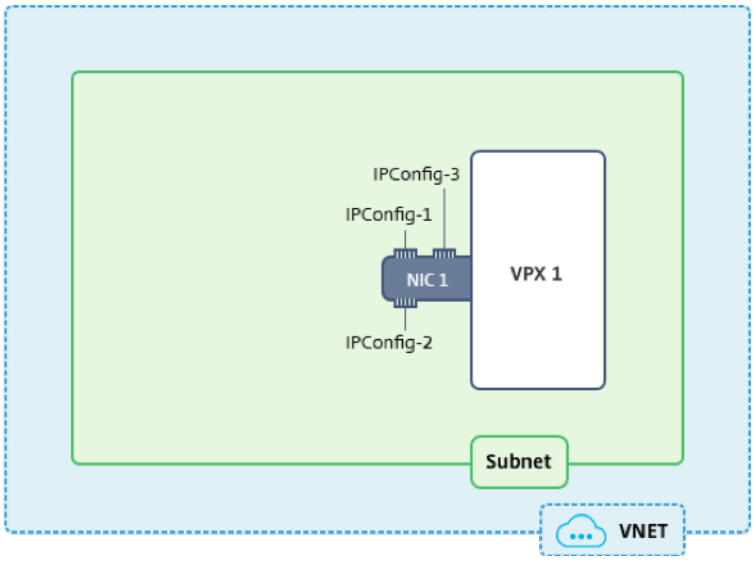
```
New-AzureRmVM -VM $vmConfig -ResourceGroupName ”myResourceGroup” -
    Location ”EastUs”
<!--NeedCopy-->
```



Citrix ADC 多 IP 地址概述

在此用例中，可以使用连接到虚拟网络 (VNET) 的单个或多个 vNIC 配置独立 Citrix ADC VPX 设备。vNIC 与三个 IP 配置 (ipconfig) 相关联，每个配置都有不同的用途。

示例：连接到 vNIC 的多个 VIP



为 NIC 分配多个 IP 配置时，必须将一个配置分配为 -Primary。

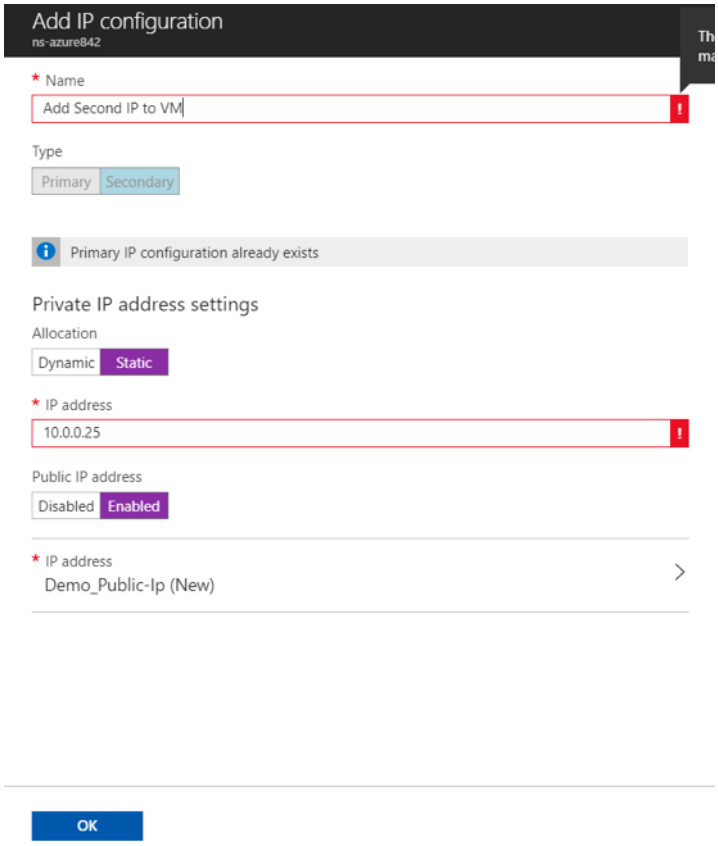
```
$MyNIC.IpConfigurations | Format-Table Name, PrivateIPAddress,
    PublicIPAddress, Primary
<!--NeedCopy-->
```

> 注意：
>
> 公共 IP 地址收取象征性费用。要了解有关 IP 地址定价的更多信息，请阅读 IP 地址定价页面。订阅中可使用的公有 IP 地址数量有限制。要了解有关限制的详细信息，请阅读 Azure 限制文章。

添加私有 **IP** 地址

要将私有 IP 地址添加到 NIC，您必须创建 IP 配置。以下命令使用 10.0.0.7 的静态 IP 地址创建配置。指定静态 IP 地址时，它必须是子网未使用的地址。我们建议您先测试该地址，通过输入 `Test-AzureRmPrivateIPAddressAvailability -IPAddress 10.0.0.7 -VirtualNetwork $myVnet` 命令来确保该地址可用。如果 IP 地址可用，则返回输出 `True`。如果地址不可用，则输出将返回 `False`，并包含可用地址列表。

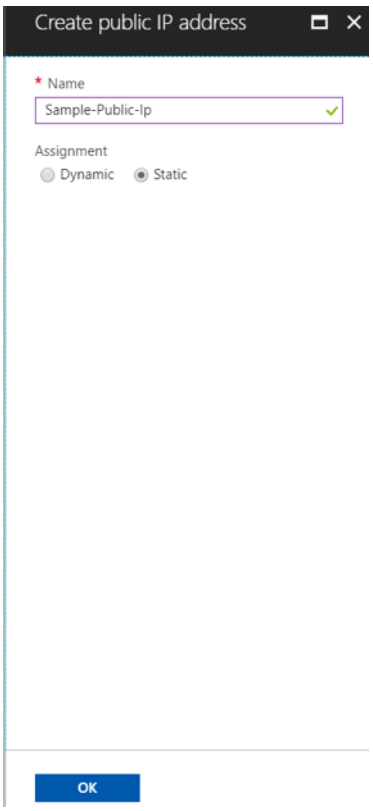
```
Add-AzureRmNetworkInterfaceIpConfig -Name IPConfig-4 -NetworkInterface
    `
$MyNIC -Subnet $Subnet -PrivateIpAddress 10.0.0.7
<!--NeedCopy-->
```

添加公有 **IP** 地址

通过将公有 IP 地址资源与新 IP 配置或现有 IP 配置关联来添加公有 IP 地址。根据需要完成以下部分之一中的步骤。

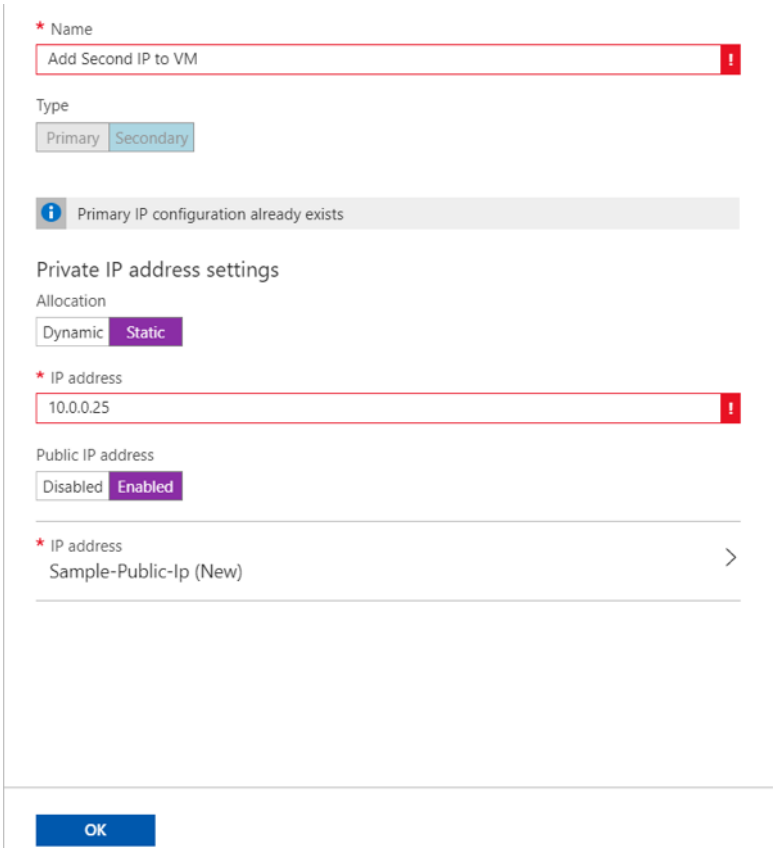
```
$MyPublicIp3 = New-AzureRmPublicIpAddress `
 -Name ”MyPublicIp3” `
 -ResourceGroupName $RgName `
 -Location $Location -AllocationMethod Static
 <!--NeedCopy-->
```

将公有 **IP** 地址资源与现有虚拟机关联起来

公有 IP 地址资源只能与尚未关联的 IP 配置相关联

```
Set-AzureRmNetworkInterfaceIpConfig `
 -Name IpConfig-3 `
 -NetworkInterface $mynic `
 -Subnet $Subnet `
 -PublicIpAddress $myPublicIp3
<!--NeedCopy-->
```

Citrix ADC 高可用性概述

您可以在 Azure 上的主动-被动高可用性 (HA) 设置中部署具有多个 NIC 的一对 Citrix ADC 虚拟设备。每个 NIC 可以包含多个 IP 地址。主动-被动部署需要：

- HA 独立网络配置 (INC) 配置
- Azure 负载平衡器 (ALB) 在直接服务器返回 (DSR) 模式下

所有流量都通过主节点。在主节点发生故障前，辅助节点一直处于备用模式。

在主动-被动部署中，ALB 浮动公有 IP (PIP) 地址将作为每个 Citrix ADC 节点的 VIP 地址添加。在 HA-INC 配置中，VIP 地址是浮动的，而 SNIP 地址是实例特定的。ALB 通过每 5 秒发送运行状况探头来监视每个 Citrix ADC 实例。ADC 只将流量重定向到定期发送运行状况探测响应的实例。因此，在 HA 设置中，主节点响应运行状况探测，而辅助节点不响应。如果主实例错过了两个连续运行状况探测，ALB 不会将流量重定向到该实例。故障转移时，新的主服务器开始响应运行状况探测，ALB 会将流量重定向到该探测。标准的 Citrix ADC HA 故障切换时间为三秒。切换流量可能需要的故障转移总时间最长为 13 秒。

您可以通过以下两种方式在主动-被动高可用性模式下部署 Citrix ADC 对：

- Citrix ADC 标准 HA 模板：使用此选项配置具有三个子网和六个 NIC 的默认选项的 HA 对。
- Windows PowerShell 命令：使用此选项可根据子网和 NIC 要求配置高可用性对。

Citrix ADC 高可用性配置 - PowerShell

检查 使用 PowerShell 命令配置具有多个 IP 地址和 NIC 的 HA 设置 中的 Azure PowerShell 命令。

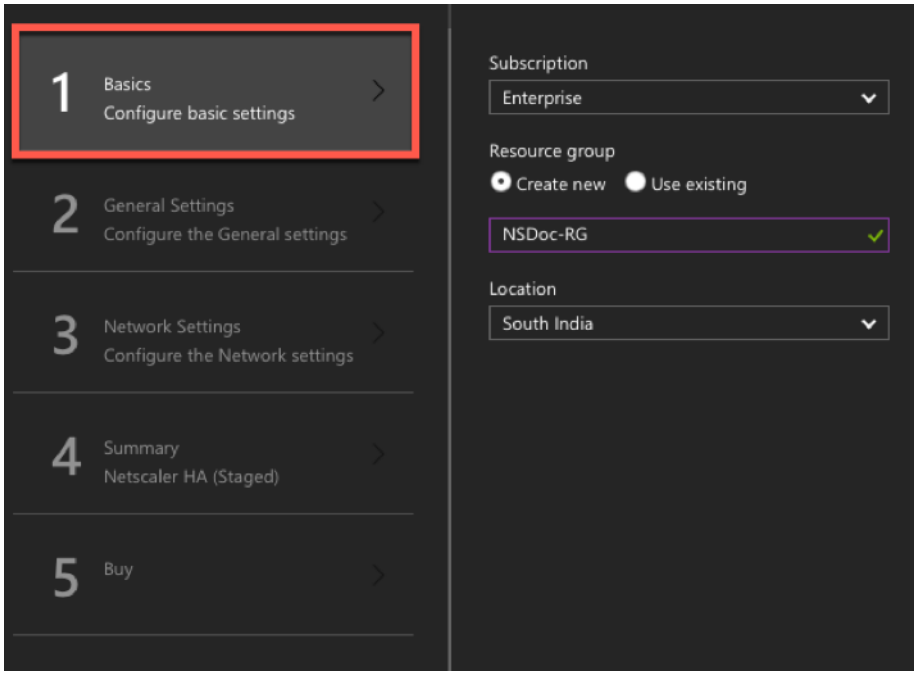
Citrix ADC 高可用性配置 - Azure 门户

您可以使用标准模板在 HA-INC 模式下快速高效地部署一对 Citrix ADC 实例。模板创建两个节点，其中包含三个子网和六个 NIC。子网用于管理、客户端和服务器端流量，每个子网都有两个 VPX 实例的 NIC。

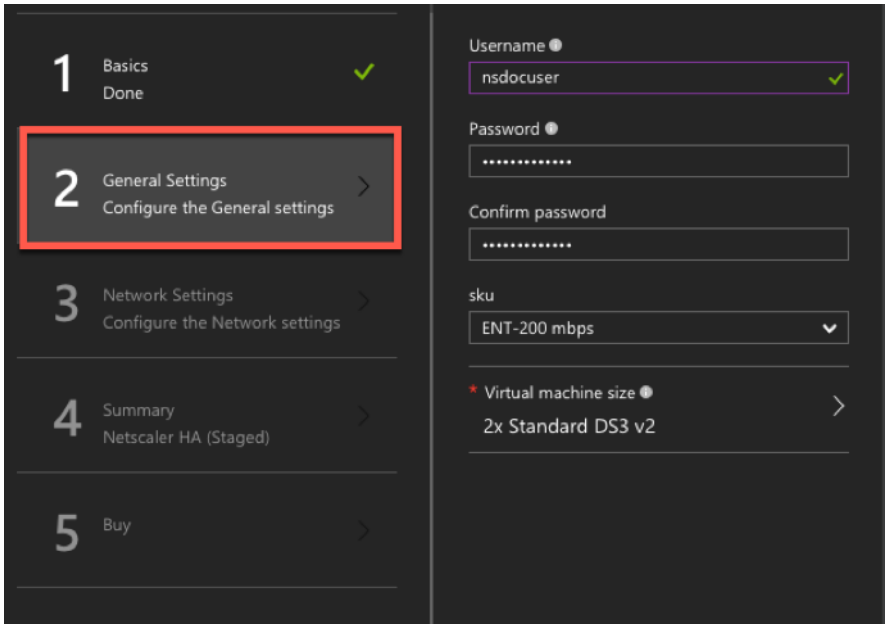
在 Azure 应用商店中可用。Citrix ADC 12.0 高可用性对模板

要使用模板，请执行以下操作：

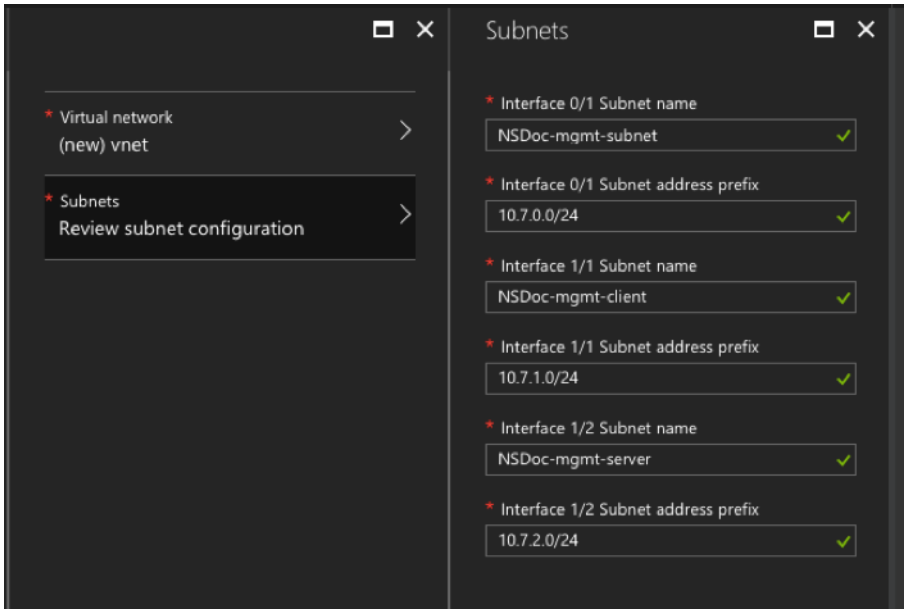
1. 在 Azure 应用商店中，选择并启动 Citrix 解决方案模板。此时将显示模板。
2. 确保部署类型为资源管理器，然后选择创建。
3. 此时将显示“基础知识”页面。创建资源组，然后选择确定。



4. 此时将显示“常规设置”页面。键入详细信息，然后选择确定。



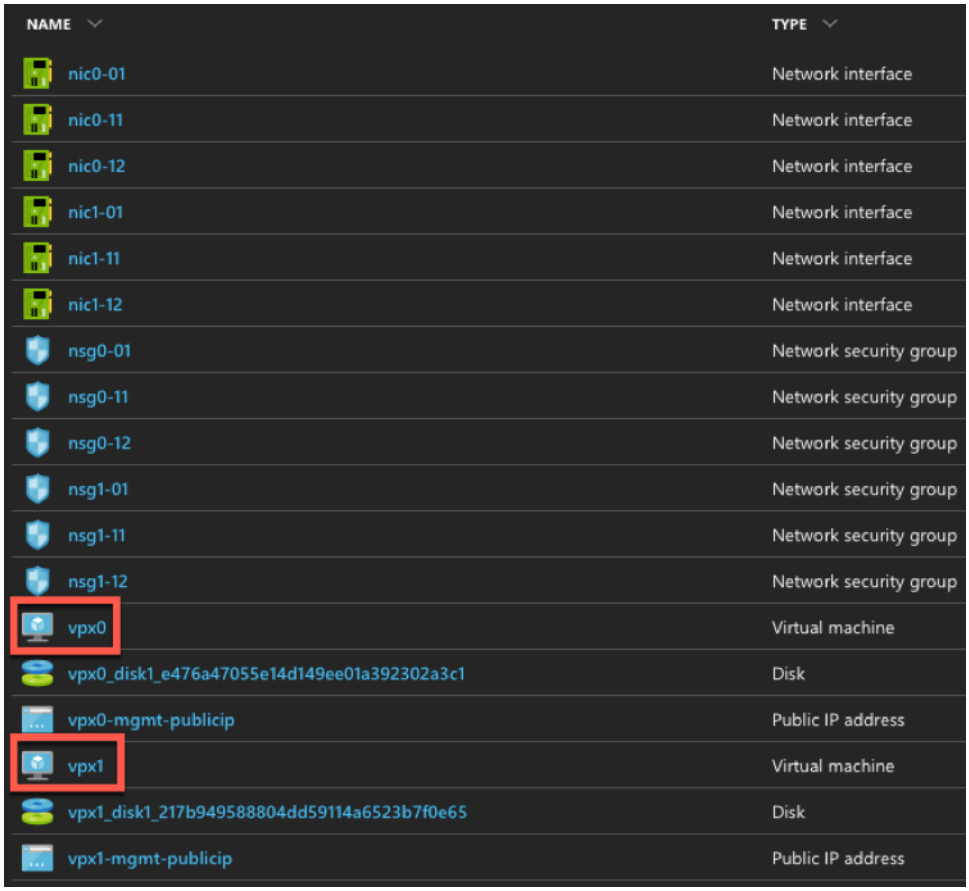
5. 此时将显示“网络设置”页面。检查 vnet 和子网配置，编辑所需的设置，然后选择确定。



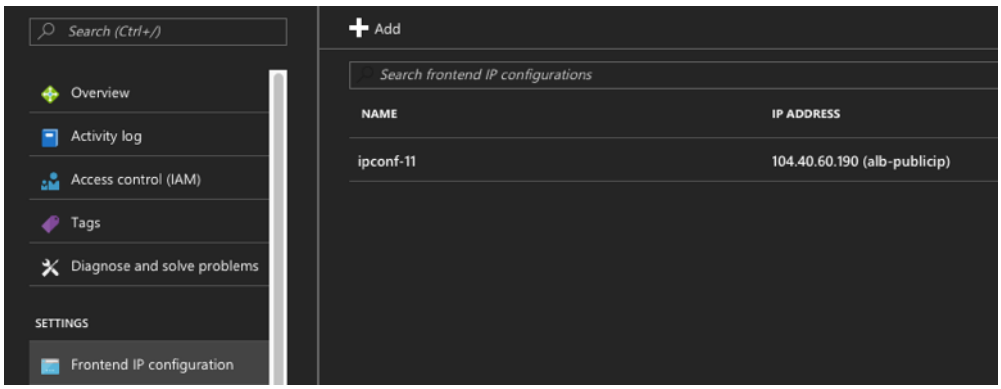
6. 此时将显示摘要页面。检查配置并相应地进行编辑。选择确定进行确认。
7. 此时将显示“购买”页面。选择“购买”以完成部署。

可能需要一段时间采用所需配置来创建 Azure 资源组。完成后，选择资源组以查看配置详细信息。这可能包括 Azure 门户中的 LB 规则、后端池、运行状况探测等。HA 对显示为 VPX0 和 VPX1。

如果需要对 HA 设置进行进一步修改，例如创建更多安全规则和端口，则可以从 Azure 门户执行此操作。



接下来，需要在每个节点上使用 ALB 公有 IP (PIP) 地址配置负载平衡虚拟服务器。要查找 ALB PIP，请选择 **ALB >** 前端 **IP** 配置。



Citrix ADC GSLB 和基于域的服务使用云负载平衡器进行后端自动缩放

GSLB 和 **DBS** 概述

Citrix ADC GSLB 支持将 DBS（基于域的服务）用于云负载平衡器。这允许使用云负载平衡器解决方案自动发现动态云服务。此配置允许 Citrix ADC 在主动-主动环境中实现全局服务器负载平衡基于域名的服务 (GSLB DBS)。DBS 允

许从 DNS 发现中扩展 Amazon Web Services (AWS) 和 Microsoft Azure 环境中的后端资源。本部分内容介绍了 AWS 中的 Citrix ADC 与 Azure Auto Scaling 环境之间的集成。本文档的最后一部分详细介绍了设置跨两个特定于 AWS 或 Azure 区域的不同可用区 (AZ) 的 HA 对 Citrix ADC 的能力。

必备项：

Citrix ADC GSLB 服务组的必备条件包括运行良好的 Amazon Web Services/Microsoft Azure 环境，该环境具有配置安全组、Linux Web 服务器、AWS 内的 Citrix ADC、弹性 IP 和弹性负载平衡器的知识和能力。

GSLB DBS 服务集成要求为 AWS ELB 和 Microsoft Azure ALB 负载平衡器实例使用 Citrix ADC 版本 12.0.57。

Citrix ADC GSLB 服务组功能增强

GSLB 服务组实体：Citrix ADC 版本 12.0.57

引入了 GSLB 服务组，它支持使用 DBS 动态发现自动缩放。

DBS 功能组件（基于域的服务）必须绑定到 GSLB 服务组

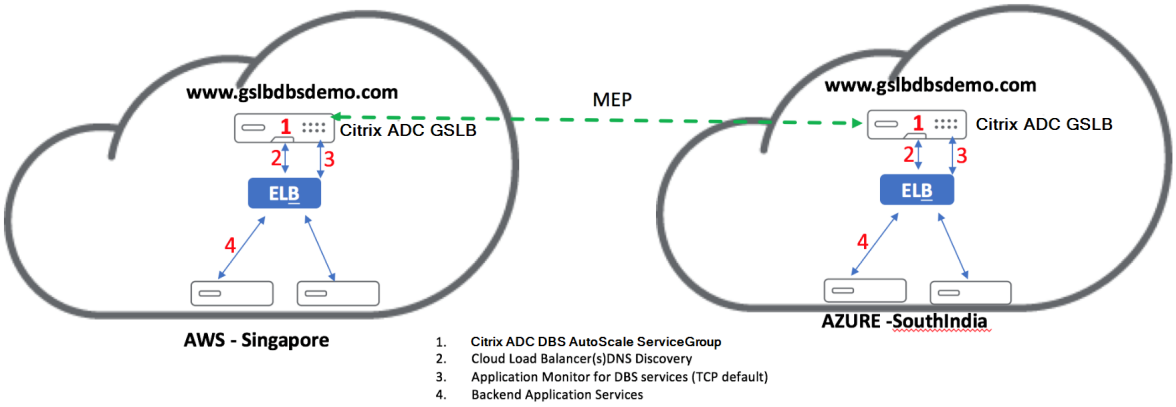
示例：

```
1  > add server sydney_server LB-Sydney-xxxxxxxxxx.ap-southeast-2.elb.
       amazonaws.com
2  > add gslb serviceGroup sydney_sg HTTP -autoscale DNS -siteName sydney
3  > bind gslb serviceGroup sydney_sg sydney_server 80
4  <!--NeedCopy-->
```

基于域名的服务 **— Azure ALB**

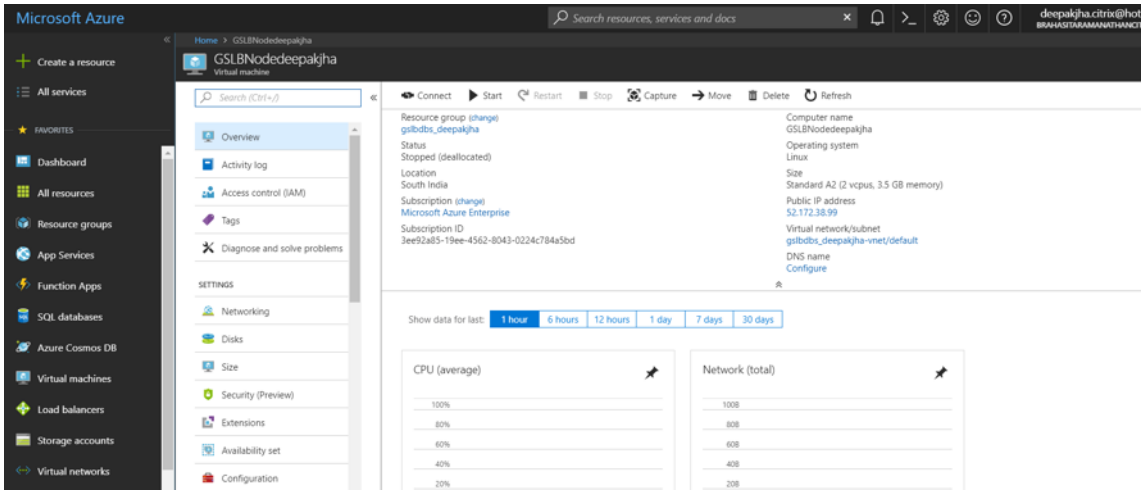
GLSB DBS 利用 Azure 负载平衡器的 FQDN 动态更新 GSLB 服务组，以包括在 Azure 中创建和删除的后端服务器。要配置此功能，我们将 Citrix ADC 指向 Azure 负载平衡器，以动态路由到 Azure 中的不同服务器。我们可以做到这一点，而无需每次在 Azure 中创建和删除实例时手动更新 Citrix ADC。用于 GSLB 服务组的 Citrix ADC DBS 功能使用 DNS 感知服务发现来确定 autoscaler 组中标识的 DBS 命名空间的成员服务资源。

图表：使用云负载平衡器的 **Citrix ADC GSLB DBA AutoScale** 组件

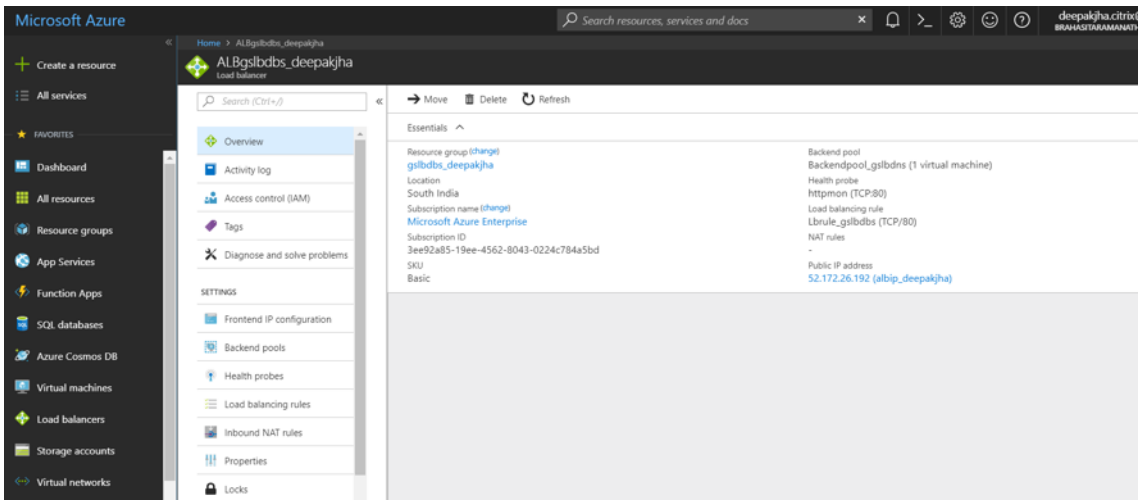


配置 **Azure** 组件

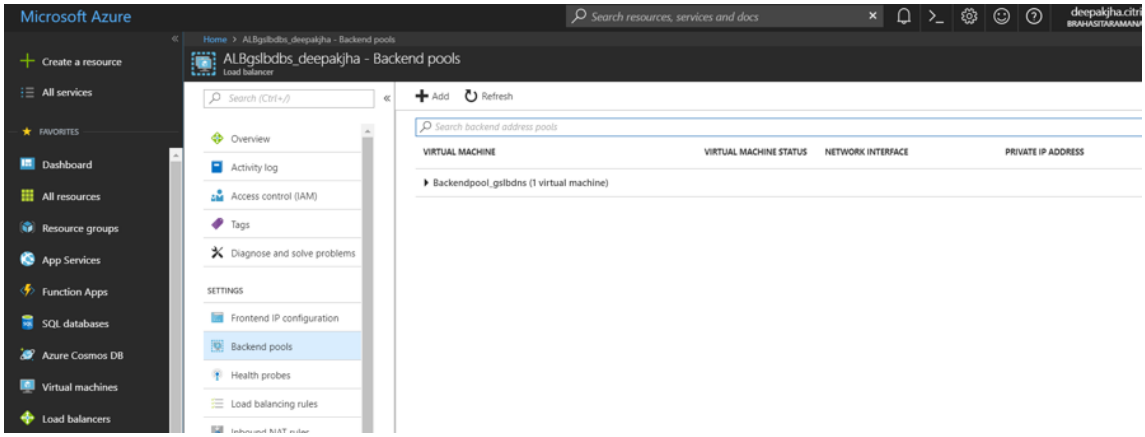
1. 登录 Azure 门户并从 Citrix ADC 模板创建新虚拟机



2. 创建 Azure 负载平衡器



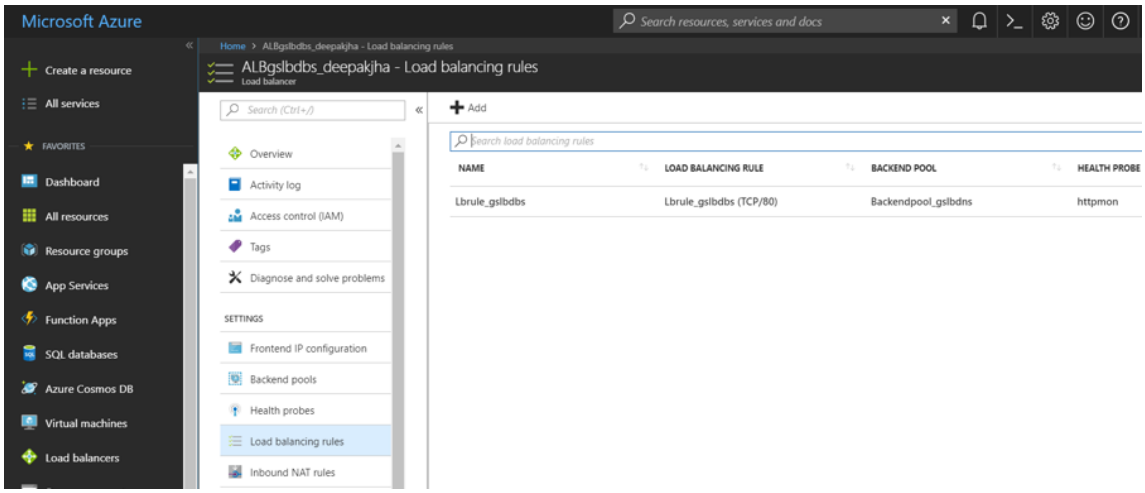
3. 添加已创建的 Citrix ADC 后端池



4. 为端口 80 创建运行状况探测器。

   利用从负载平衡器创建的前端 IP 创建负载平衡规则。

   a) 协议：TCP
   b) 后端端口：80
   c) 后端池：在步骤 1 中创建 Citrix ADC
   d) 运行状况探测器：在步骤 4 中创建
   e) 会话持久性：无



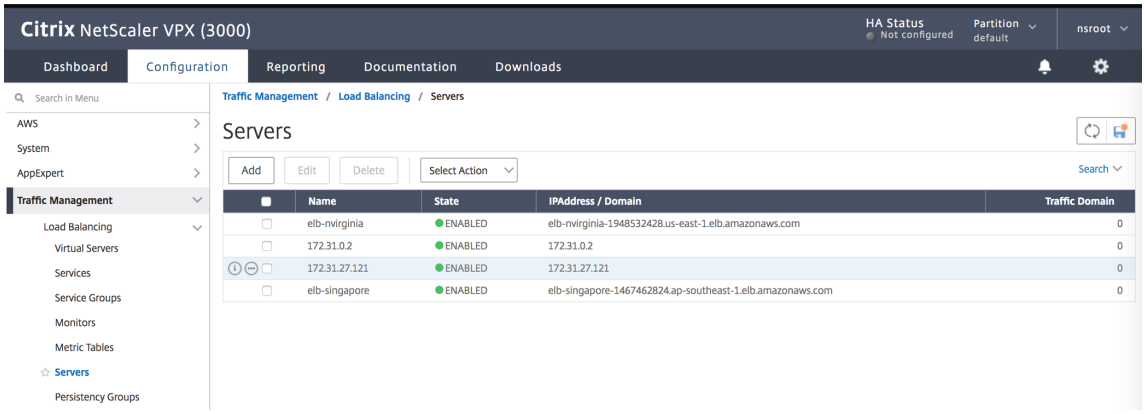
配置基于 **Citrix ADC GSLB** 域的服务

以下配置总结了在启用了 GSLB 的环境中启用基于域的服务以实现自动缩放 ADC 所需的对象。

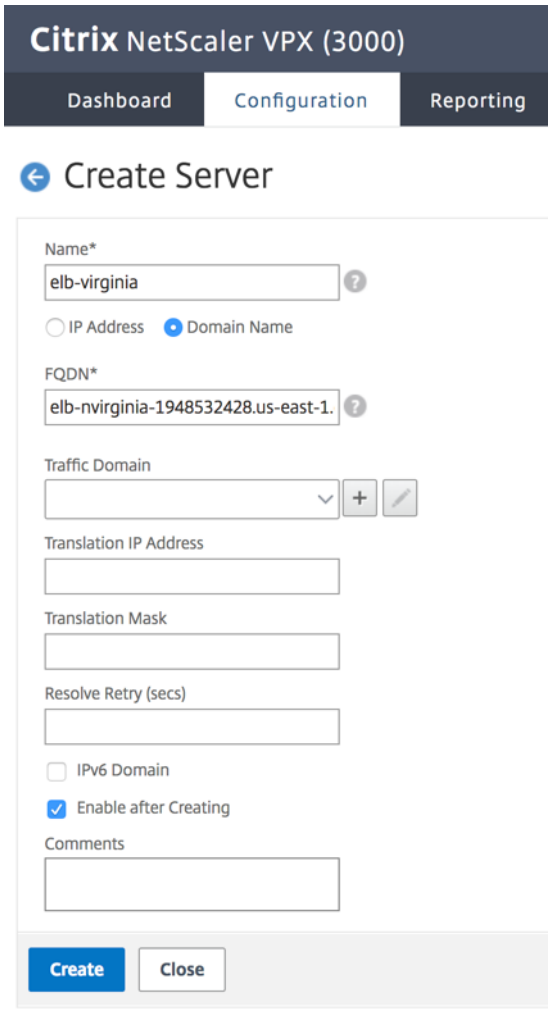
流量管理配置

> 注意：需要配置 Citrix ADC，名称服务器或 DNS 虚拟服务器 通过它解析 DBS 服务组的 ELB/ALB 域。

1. 导航到流量管理 -> 负载平衡 -> 服务器



2. 单击“添加”以创建服务器，提供与 Azure 负载平衡器 (ALB) 中的 A 记录（域名）相对应的名称和 FQDN



3. 重复步骤 2 以从 Azure 中的第二个资源添加第二个 ALB。

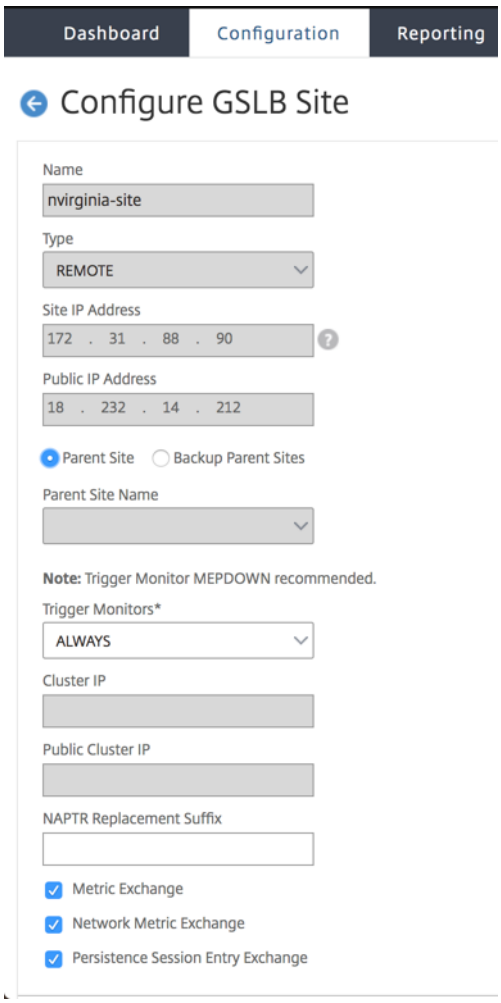
GSLB 配置

1. 单击添加按钮以配置 GSLB 站点
2. 为站点命名。

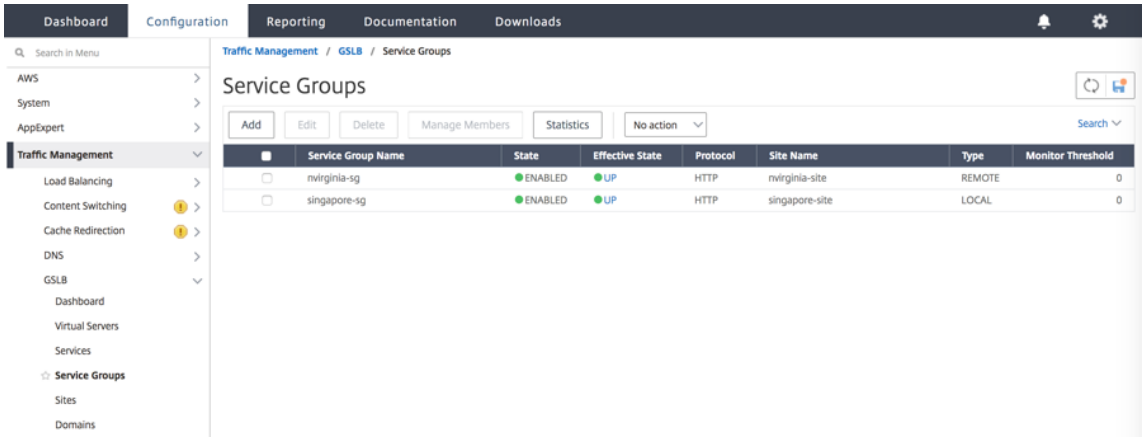
   “类型”根据您在哪个 Citrix ADC 配置为“远程”或“本地”。站点 IP 地址是 GSLB 站点的 IP 地址。GSLB 站点使用此 IP 地址与其他 GSLB 站点进行通信。在使用某个特定 IP 托管在外部防火墙或 NAT 设备上的云服务时，需要公有 IP 地址。该站点应配置为父站点。确保“触发器监视器”设置为“始终”。此外，请务必勾选“衡量指标交换”、“网络衡量指标交换”和“持久性会话条目交换”底部的三个框。

   建议将触发器设置设置为 MEPDOWN，请参阅配置 GSLB 服务组。

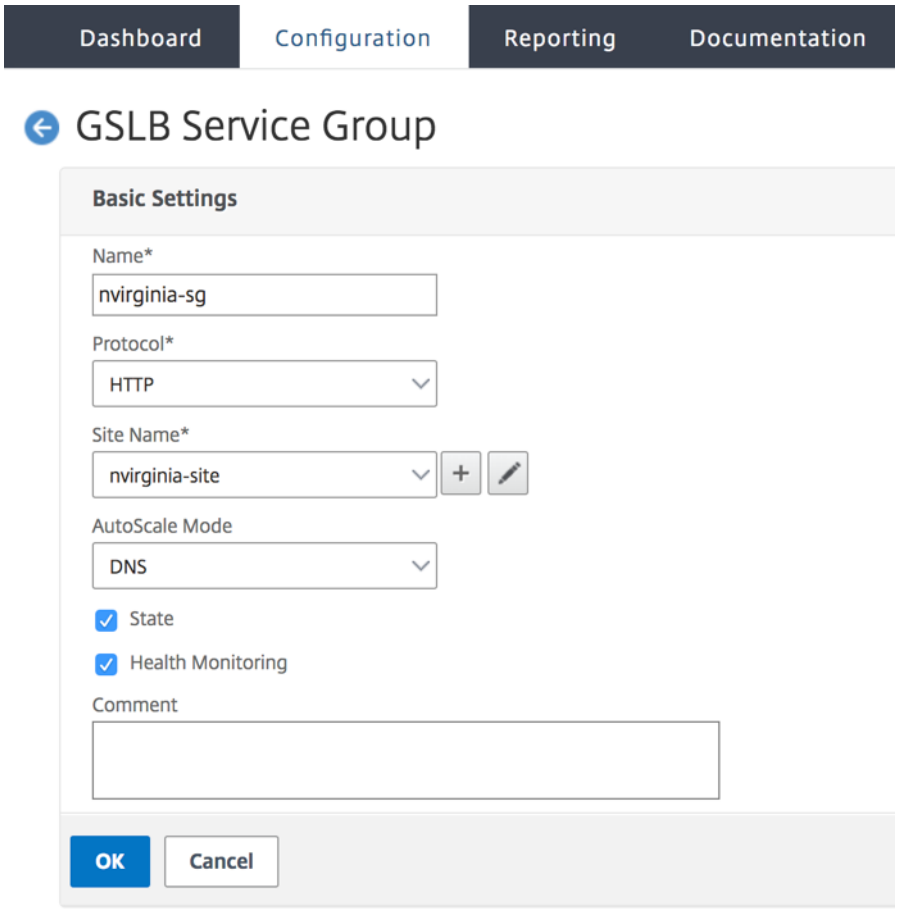
   ![image-citrix-adc-and-microsoft-azure-28]/en-us/advanced-concepts/media/image-citrix-adc-and-microsoft-azure-28.png)



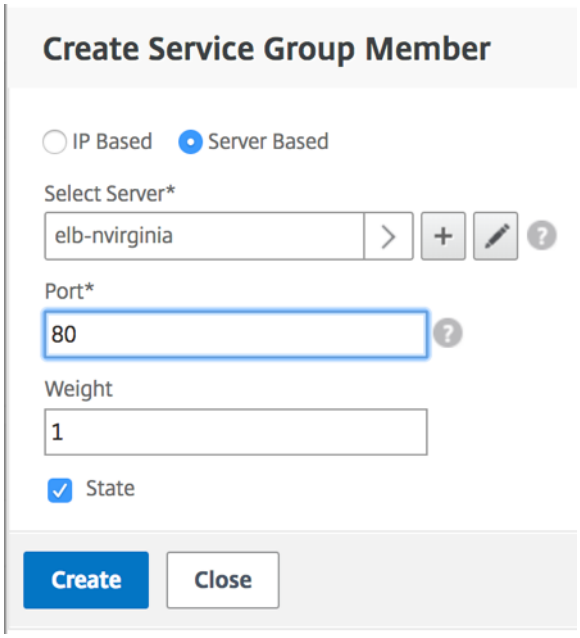
3. 单击创建，重复步骤 3 和 4 以为 Azure 中的其他资源位置配置 GSLB 站点（这可以在同一 Citrix ADC 上配置）
4. 导航到 流量管理**-> GSLB ->** 服务组



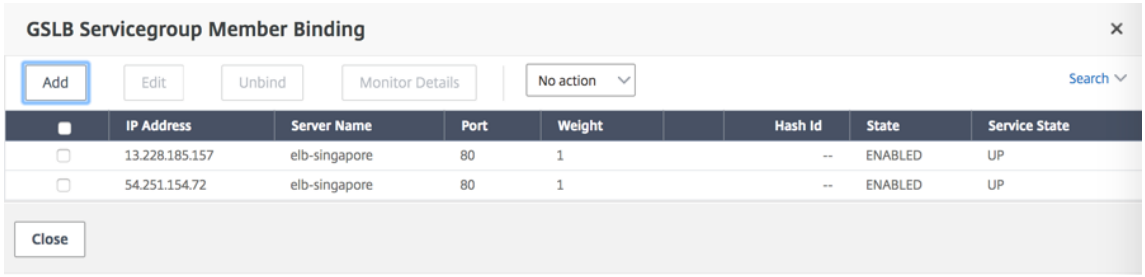
单击“添加”以添加新的服务组。命名服务组，使用 HTTP 协议，然后在“站点名称”下选择在前面的步骤中创建的相应站点。请务必将自动缩放模式配置为 DNS，并勾选“状态”和“运行状况监视”对应的复选框。单击“确定”以创建服务组



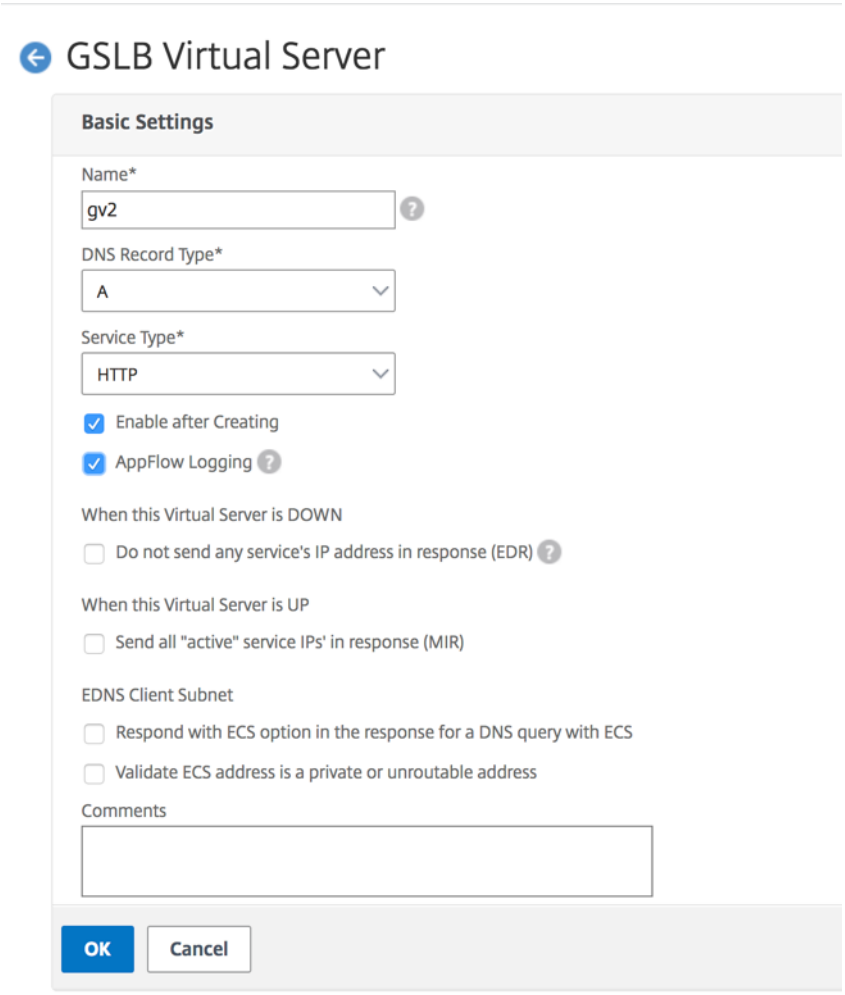
5. 单击“服务组成员”并选择“基于服务器”。选择在运行指南开始部分配置的各自弹性负载平衡服务器。将流量配置为通过端口 80。单击创建。



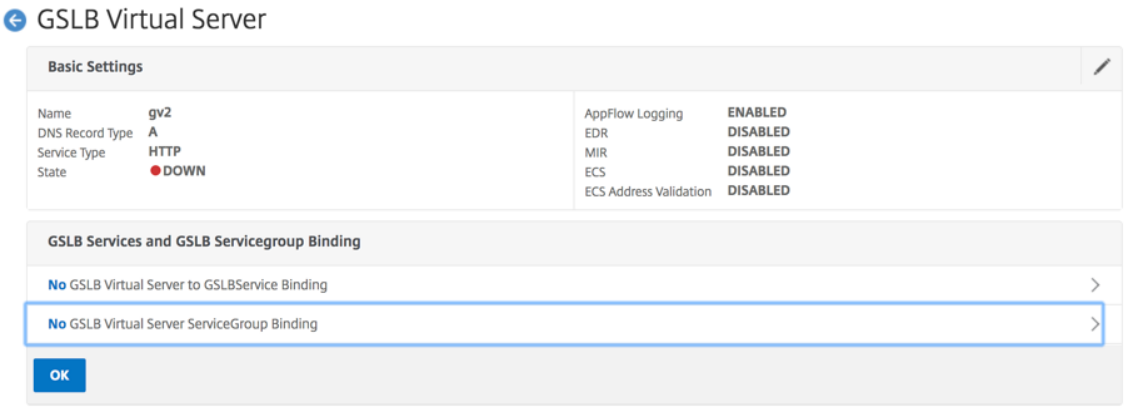
6. 服务组成员绑定应填充从弹性负载平衡器接收的 2 个实例。



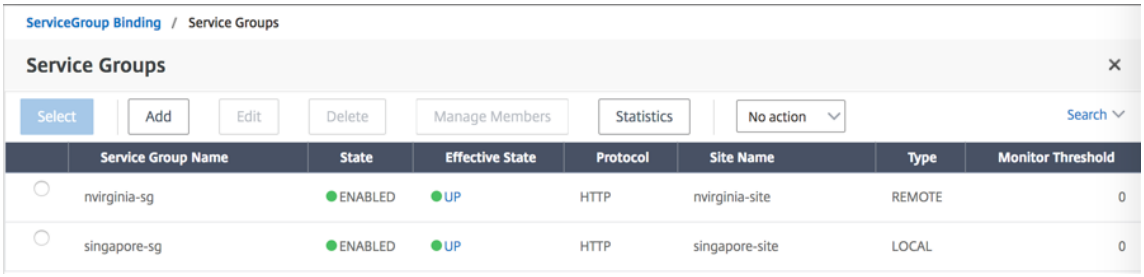
7. 重复步骤 5 和 6 为 Azure 中的第二个资源位置配置服务组。（这可以从相同的 Citrix ADC GUI 来完成）。
8. 最后一步是设置 GSLB 虚拟服务器。导航到流量管理**-> GSLB->** 虚拟服务器。
9. 单击 “添加” 以创建虚拟服务器。命名服务器，将 DNS 记录类型设置为 A，服务类型设置为 HTTP，并选中 “创建后启用” 和 “AppFlow 日志记录” 复选框。单击确定以创建 GSLB 虚拟服务器。



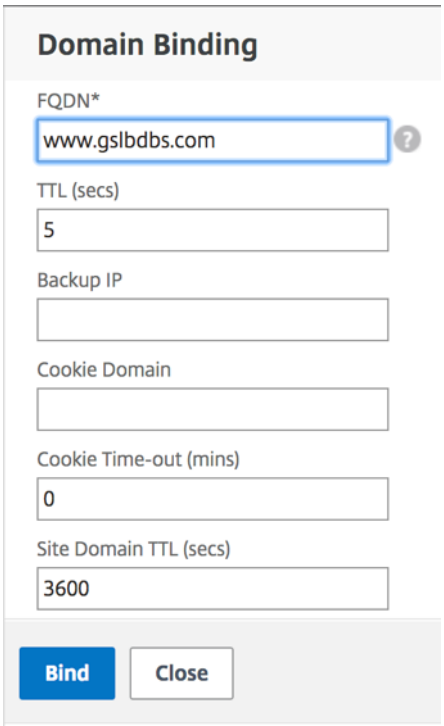
10. 创建 GSLB 虚拟服务器后，单击 无 **GSLB** 虚拟服务器服务组绑定。



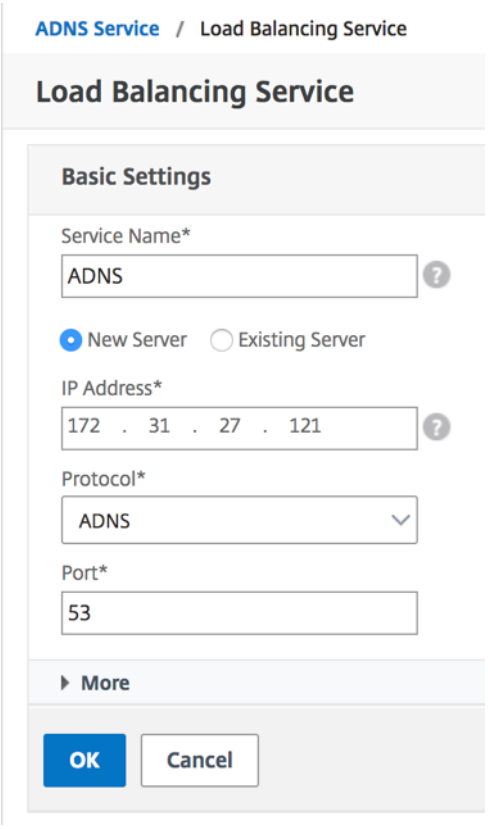
11. 在“服务组绑定”下，使用“选择服务组名称”选择并添加在上述步骤中创建的服务组。



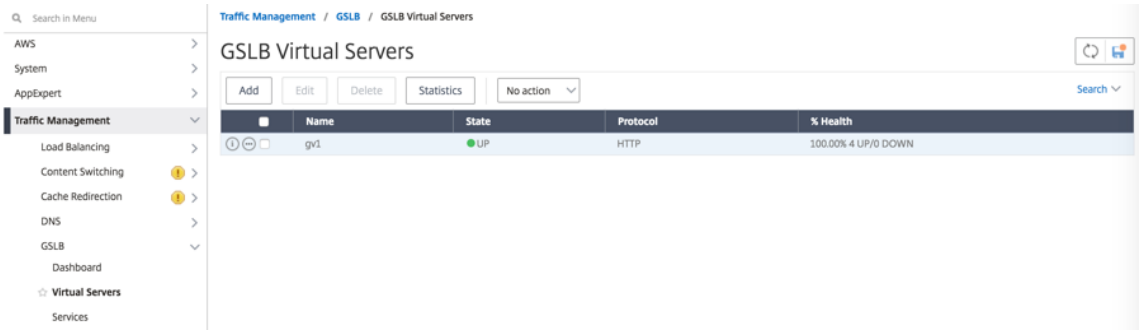
12. 接下来，通过单击无 **GSLB** 虚拟服务器域绑定配置 GSLB 虚拟服务器域绑定。配置 FQDN 和绑定，其余设置可保留为默认值。



13. 通过单击“无服务”来配置 ADNS 服务。添加服务名称，单击“新建服务器”，然后输入 ADNS 服务器的 IP 地址。此外，如果您的 ADNS 已配置，您可以选择“现有服务器”，然后从下拉菜单中选择您的 ADNS。确保协议是 ADNS，并且流量通过端口 53。



14. 将方法配置为 LEASTCONNECTION，并将备份方法配置为 ROUNDROBIN。
15. 单击完成并验证您的 GSLB 虚拟服务器是否显示为“已启动”。



使用 **Citrix Application Delivery Management** 中的 **AutoScale** 组在 **Azure** 中使用 **VPX** 进行前端自动缩放

AutoScaling 是一种云计算方法，根据实际使用情况自动添加或删除资源。当您的站点或应用程序需要按需分配资源以满足不断变化的客户端请求或处理作业时，AutoScaling 非常有用。

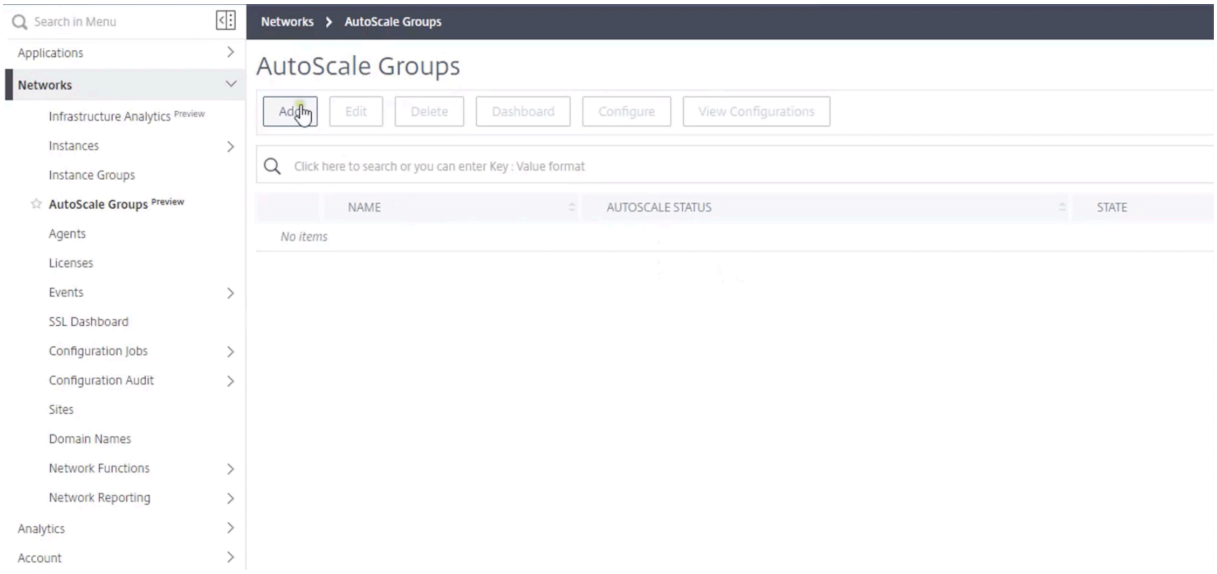
对 Web 应用程序或服务的需求可能会有很大差异。为不同的流量需求保持正确数量的 Citrix ADC 实例非常重要。您可以根据需求增加或减少 Microsoft Azure 上的网络资源。因此，它在不影响性能的情况下提供了成本优化。

Citrix Application Delivery Management (ADM) 自动缩放可保持 Citrix ADC 实例的确切数量，以应付不断波动

的资源消耗。Citrix ADM 根据资源消耗波动来确定流量流，并决定在 Citrix ADC 实例中动态扩展或扩展。

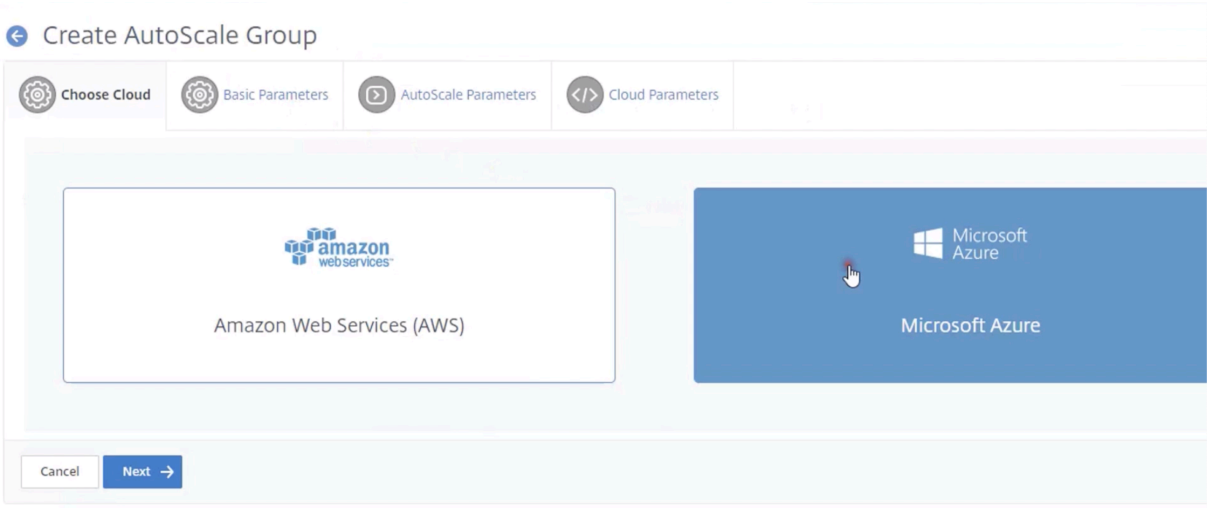
创建 **AutoScale** 组

- 在 ADC 管理控制台实用程序中，导航到网络 **>** 自动缩放。
- 选择添加。



选择云

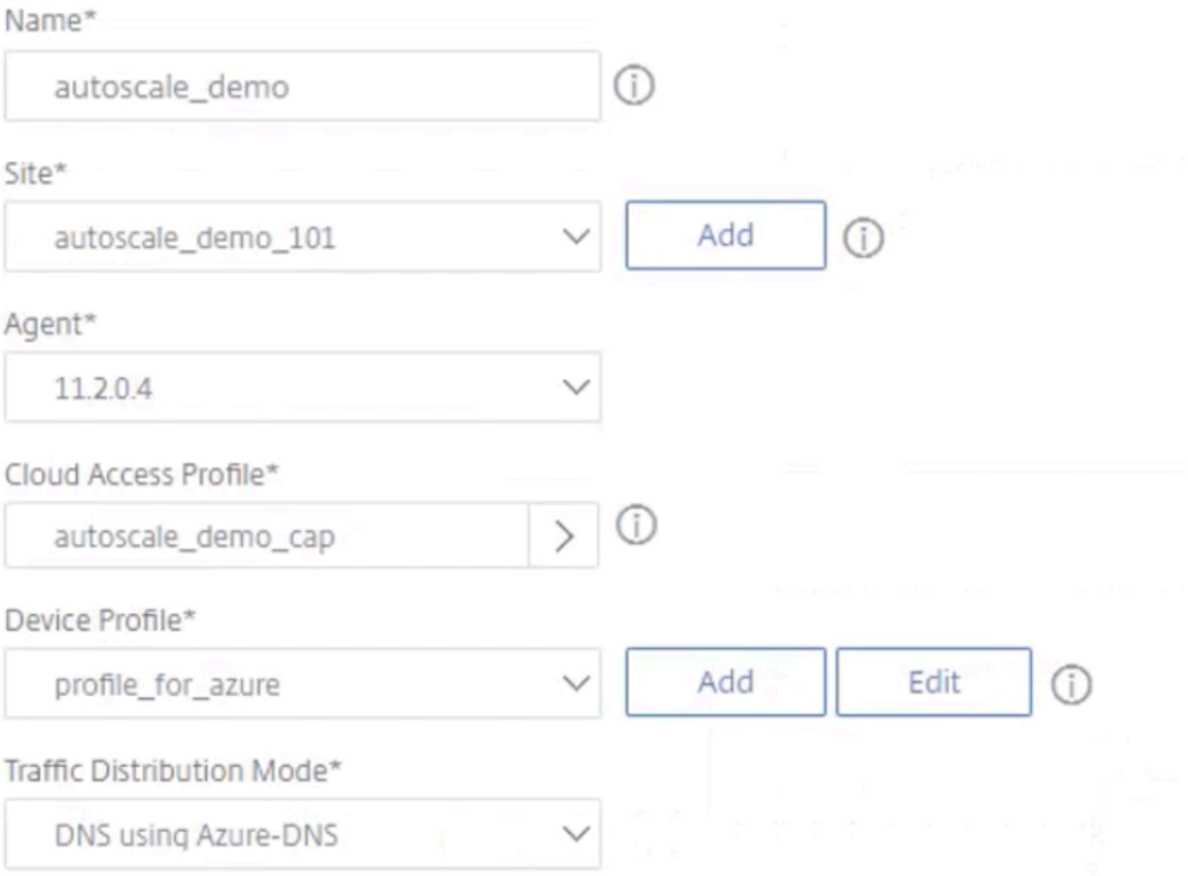
在创建 **AutoScale** 组选项卡中，单击 **Microsoft Azure**，然后单击添加。



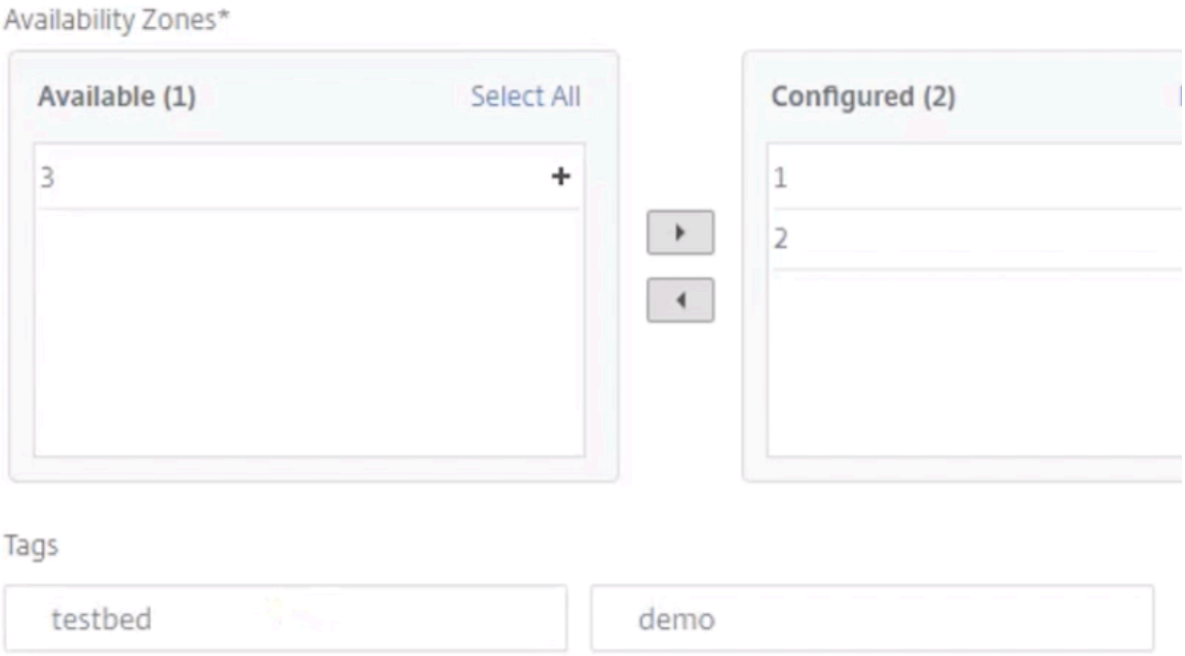
基本参数

配置 AutoScale 组的基本参数：

1. 名称：autoscale_demo
2. 站点：autoscale_demo_101
3. 代理：11.2.0.4
4. 云访问配置文件：autoscale_demp_cap
5. 设备：用于 Azure 的配置文件
6. 流量分配模式：使用 Azure DNS 的 DNS



尚未选择“下一步”。通过单击加号添加可用区 1 和 2，并将标签设置为“测试床”。

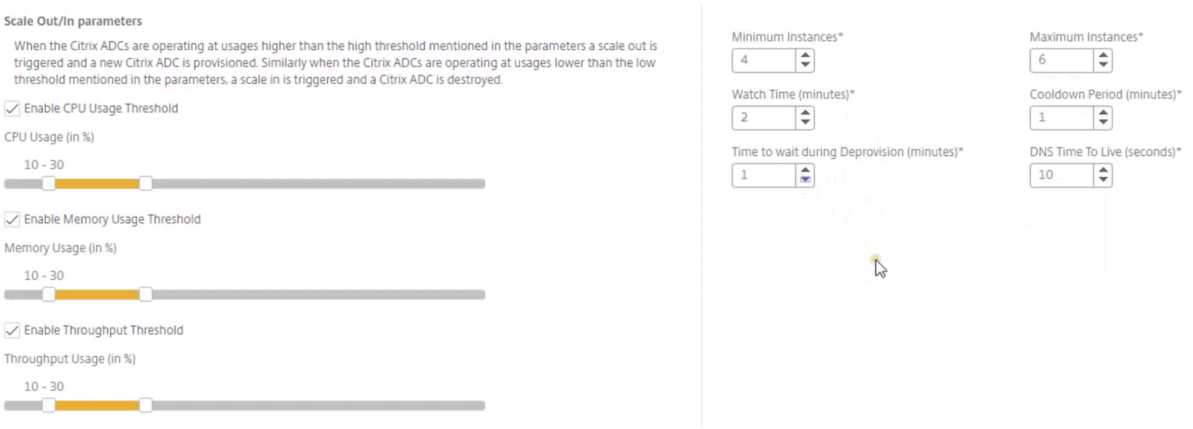


选择下一步。

AutoScale 参数

配置 AutoScale 组的 AutoScale 参数：

1. CPU 使用率：10–30
2. 内存使用率：10–30
3. 吞吐量使用量：10–30
4. 最小实例：4
5. 最大实例数：6
6. 观看时间：2
7. 冷却时间：1
8. 取消置备期间的等待时间：1
9. DNS 生存时间：10

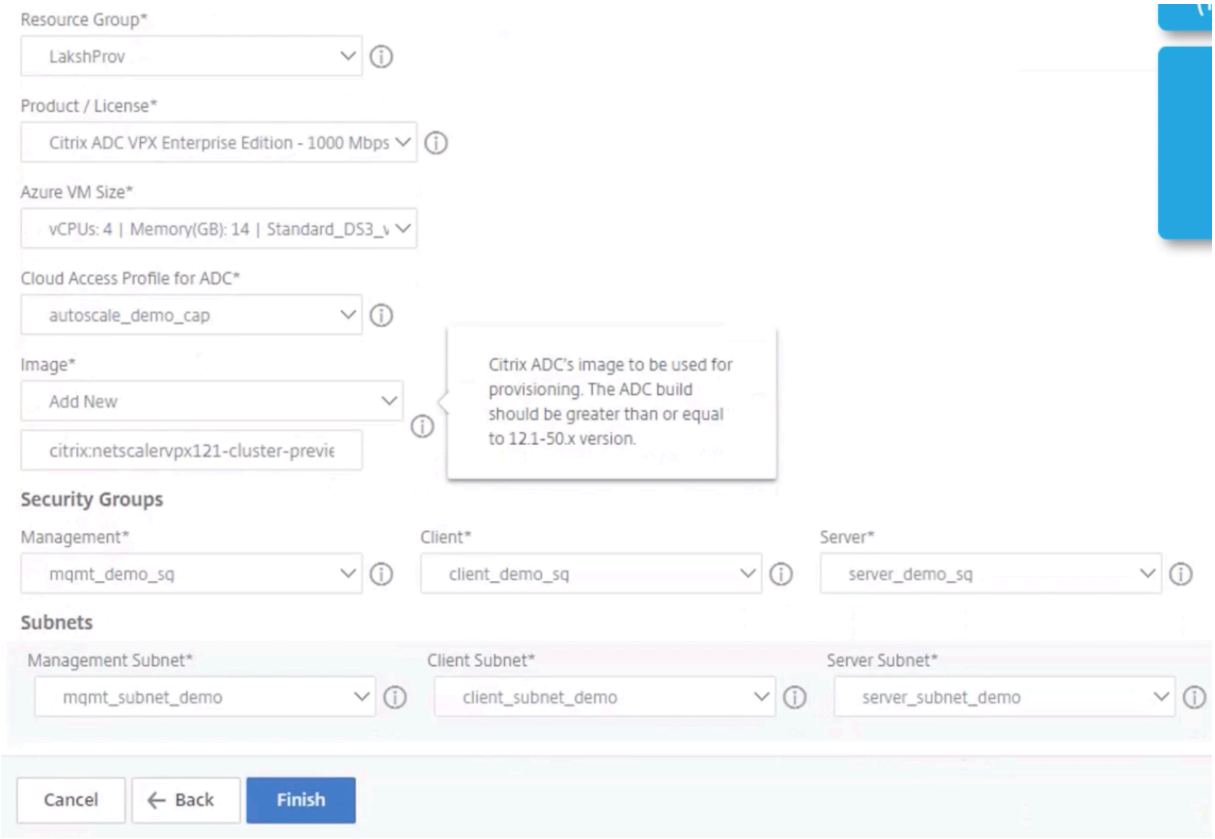


单击下一步。

云参数

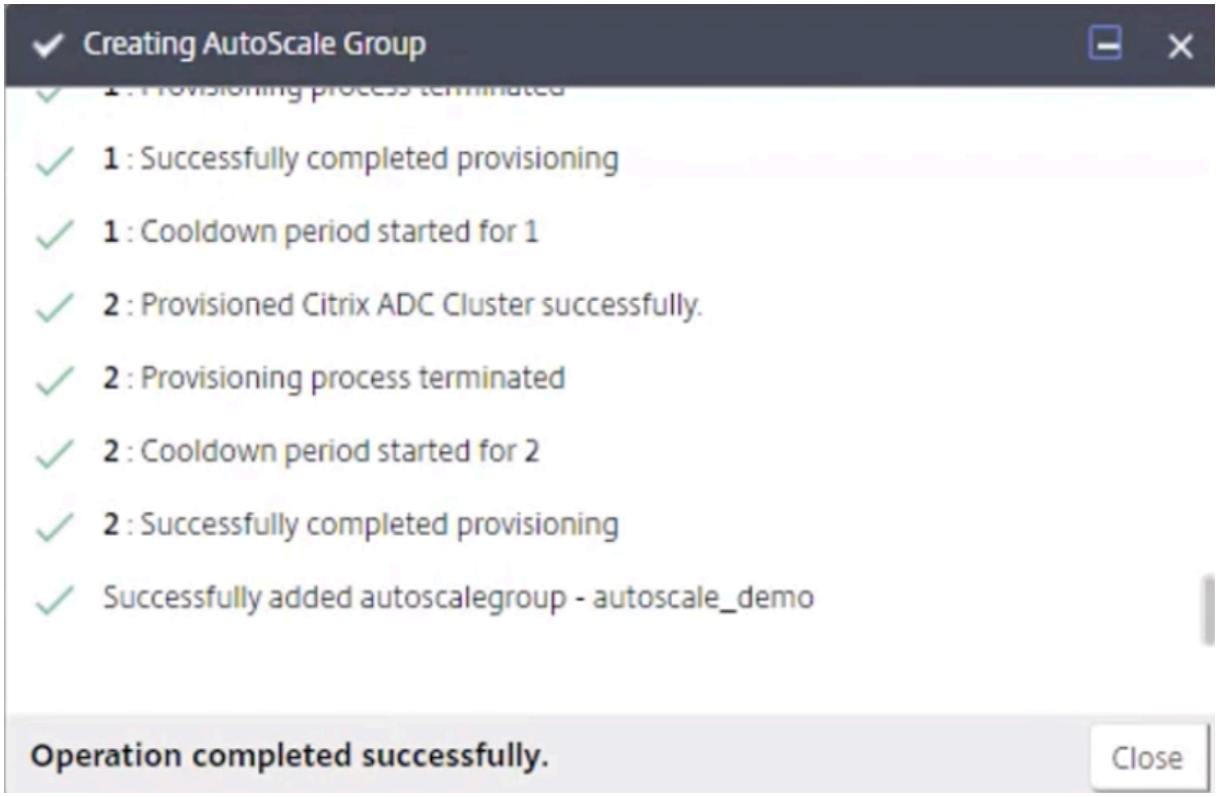
配置 AutoScale 组的云参数：

1. 资源组：LakshProv
2. 产品/许可证：Citrix ADC VPX Enterprise Edition - 1000 Mbps
3. Azure VM 大小：vCPU：4 | 内存 (GB)：14 | Standard_DS3_V2
4. ADC 的云访问配置文件：autoscale_demo_cap
5. 图像：默认值
6. 管理：mgmt_demo_sq
7. 客户：client_demo_sq
8. 服务器：server_demo_sq
9. 管理子网：mgmt_subnet_demo
10. 客户端子网：client_subnet_demo
11. 服务器子网：server_subnet_demo



选择下一步。

此时将显示一个对话框。确保 AutoScale 组成功创建。



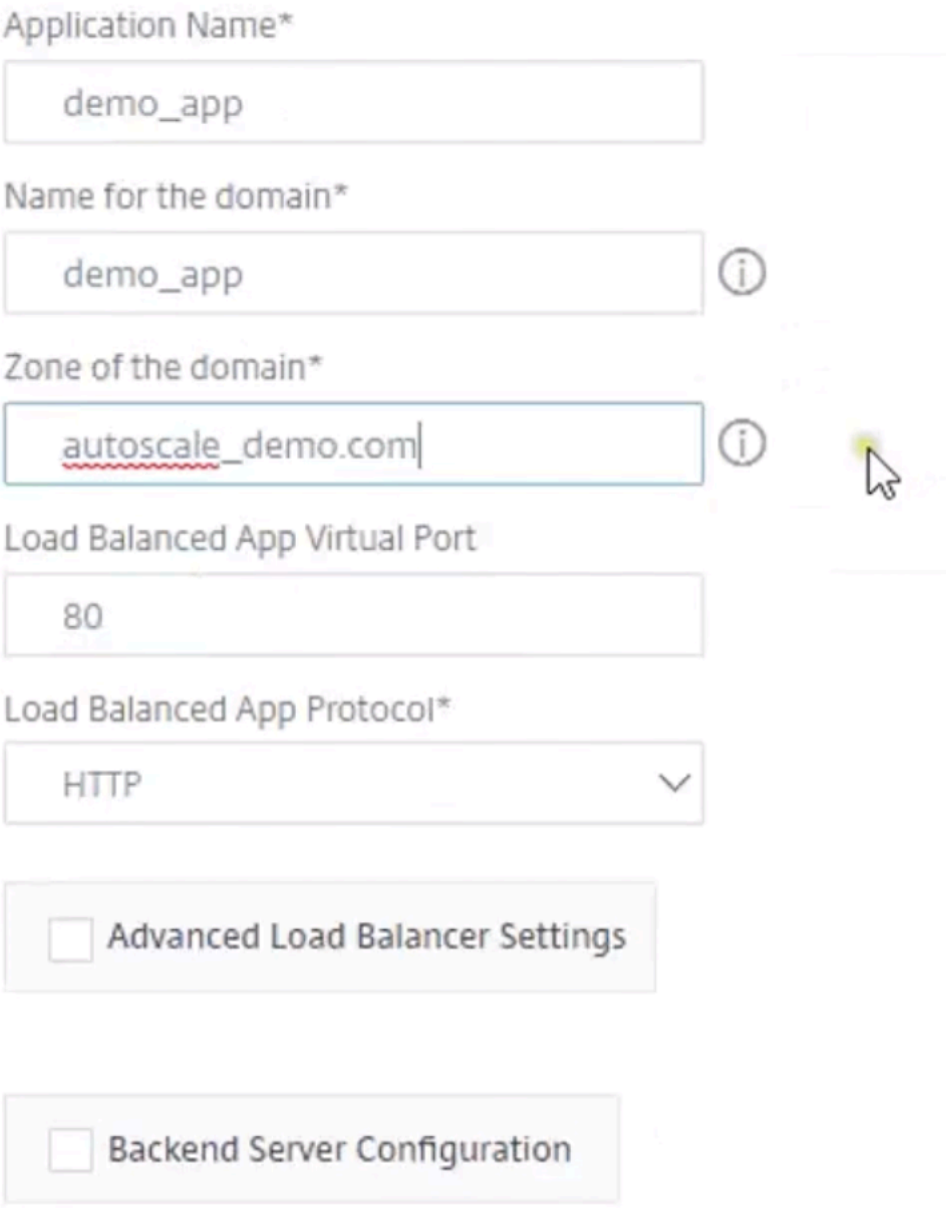
配置 **AutoScale** 组

1. 在 ADC 管理控制台实用程序中，导航到网络 **>** 自动缩放。
2. 选择已创建的自动缩放，并确保其已启用。
3. 单击 配置。
4. 在 “选择 StyleBook” 页面上选择 `HTTP/SSL LoadBalancing StyleBook`。

配置详细信息

输入要使用 StyleBook 创建的配置详细信息：

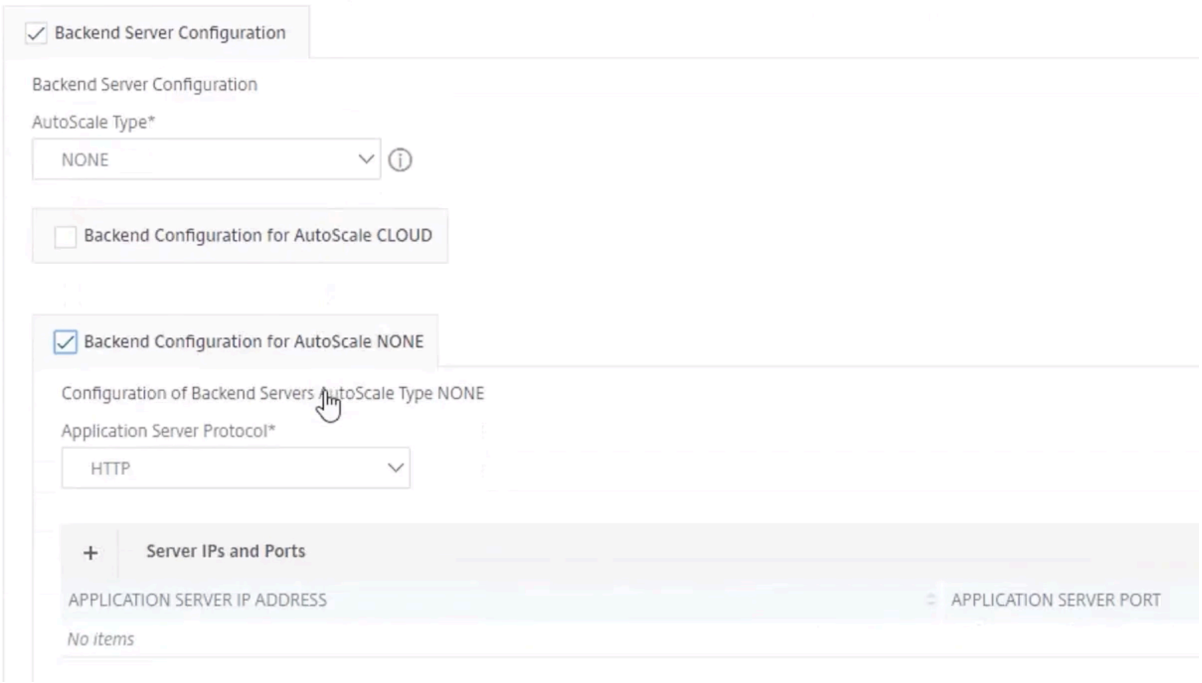
1. 应用程序名称：演示应用程序
2. 域的名称：应用程序
3. 域的区域：autoscale_demo.com
4. 负载平衡应用程序虚拟端口：80
5. 负载平衡应用协议：HTTP



检查 后端服务器配置。

1. 自动扩展类型：无

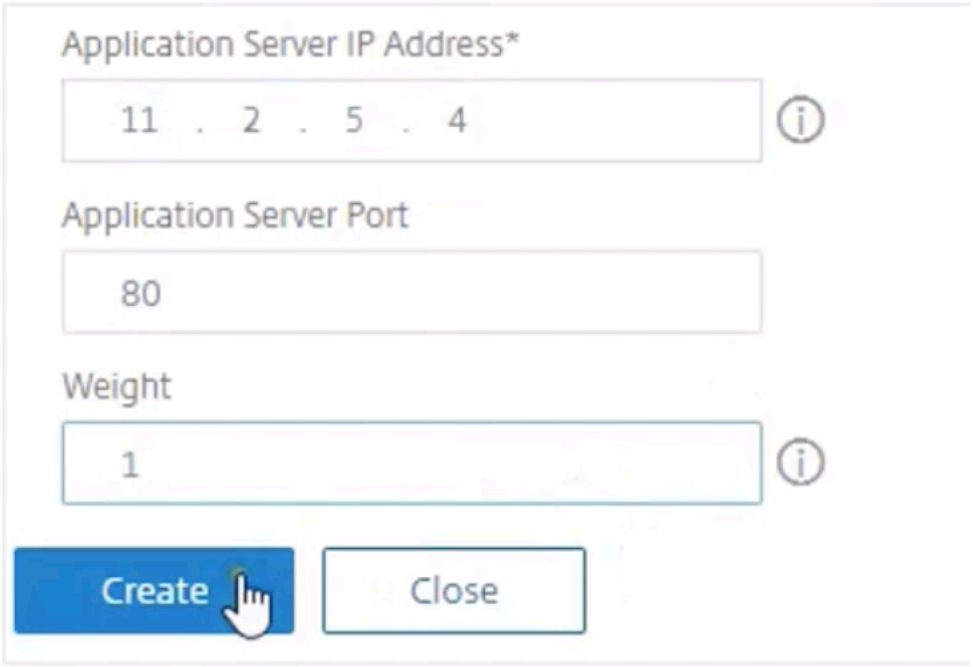
选中后端配置是否自动缩放 **NONE**



单击 服务器 **IP** 和端口旁边的加号。

1. 应用程序服务器 IP 地址：11.2.5.4
2. 应用程序服务器端口：80
3. 重量：1

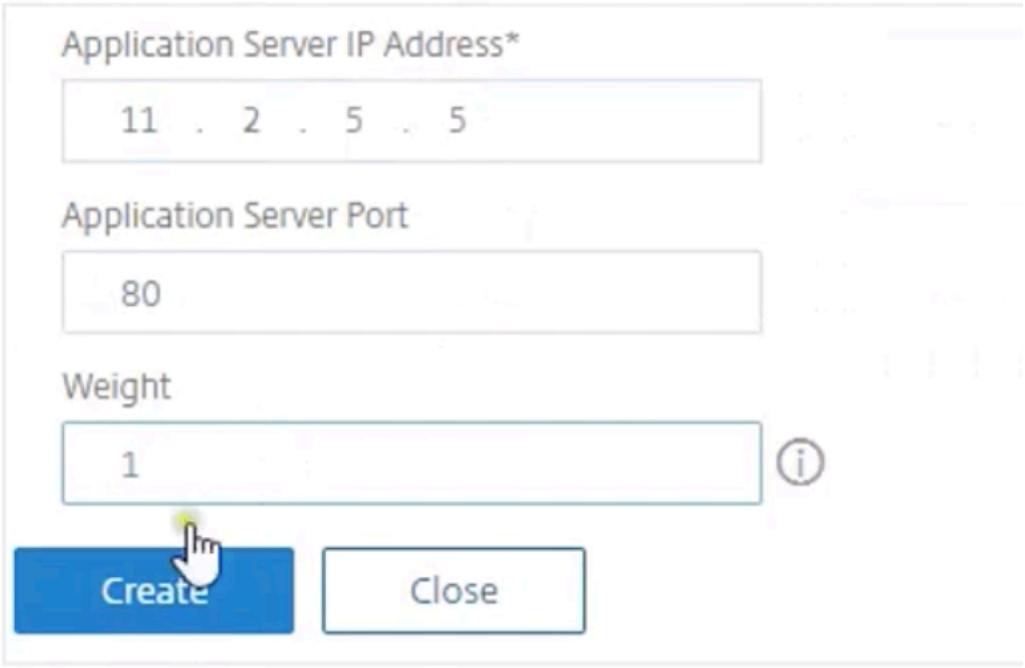
单击创建。



重复上一步：

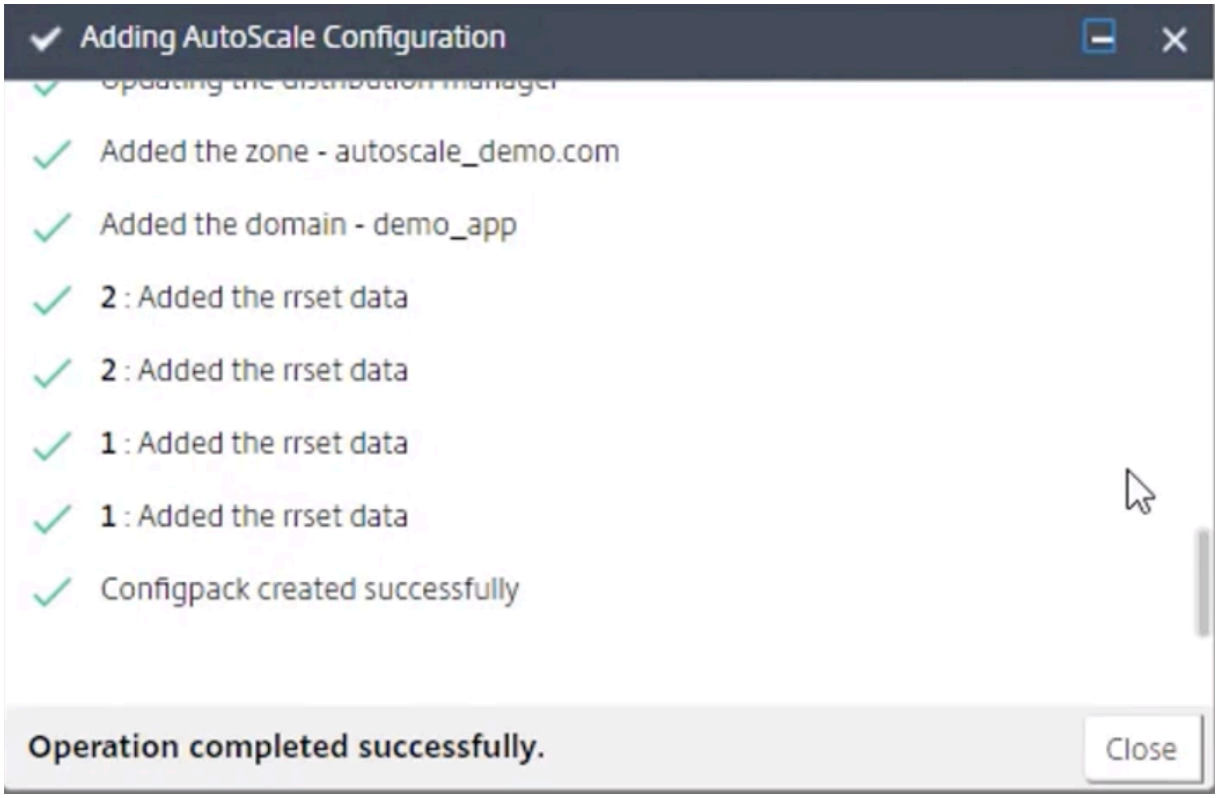
1. 应用程序服务器 IP 地址：11.2.5.5
2. 应用程序服务器端口：80
3. 重量：1

单击创建。



在主 StyleBook 配置页面上，单击创建。

此时将显示一个对话框。确保 StyleBook 配置已成功创建。



其他参考资料

在 Microsoft Azure 上部署 Citrix ADC VPC 实例

Microsoft Azure 文档

使用 PowerShell 为虚拟机分配多个 IP 地址

使用 PowerShell 创建具有多个 NIC 的虚拟机

在 Azure 公有云中使用 Citrix ADC VPX 自动缩放后端服务 VM

Citrix 产品文档

为 Citrix ADC VPX 独立实例配置多个 IP 地址

使用 PowerShell 命令在独立模式下为 Citrix ADC VPX 实例配置多个 IP 地址

为独立 VPX 实例配置多个 Azure VIP

Google Cloud 上的 Citrix ADC CPX、Citrix Ingress Controller 和 Application Delivery Management

March 2, 2021

GCP K8 体系结构和组件的 Citrix 产品概述

GCP 的 5 个主要 Citrix 组件

1. **Citrix ADC VPX** 作为第 **1** 层 **ADC**，用于基于入站的 **Internet** 客户端流量。

   GCP 中的 VPX 实例使您能够利用 GCP 计算功能，并使用 Citrix 负载平衡和流量管理功能来满足您的业务需求。您可以在 GCP 中部署 VPX 作为独立实例。支持单个和多个网络接口卡 (NIC) 配置。

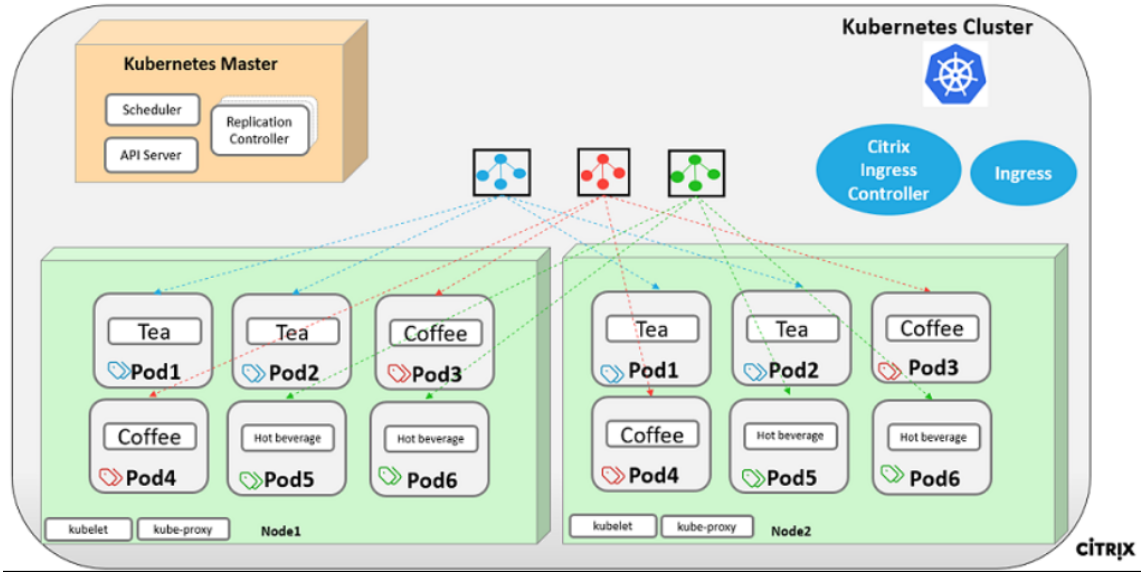
2. 使用 **Google Kubernetes Engine (GKE)** 组成容器平台的 **Kubernetes** 群集。

   Kubernetes 引擎是用于部署容器化应用程序的托管、生产就绪型环境。它可以快速部署和管理您的应用程序和服务。

3. 使用 **YAML** 文件库部署示例 **Citrix Web** 应用程序。

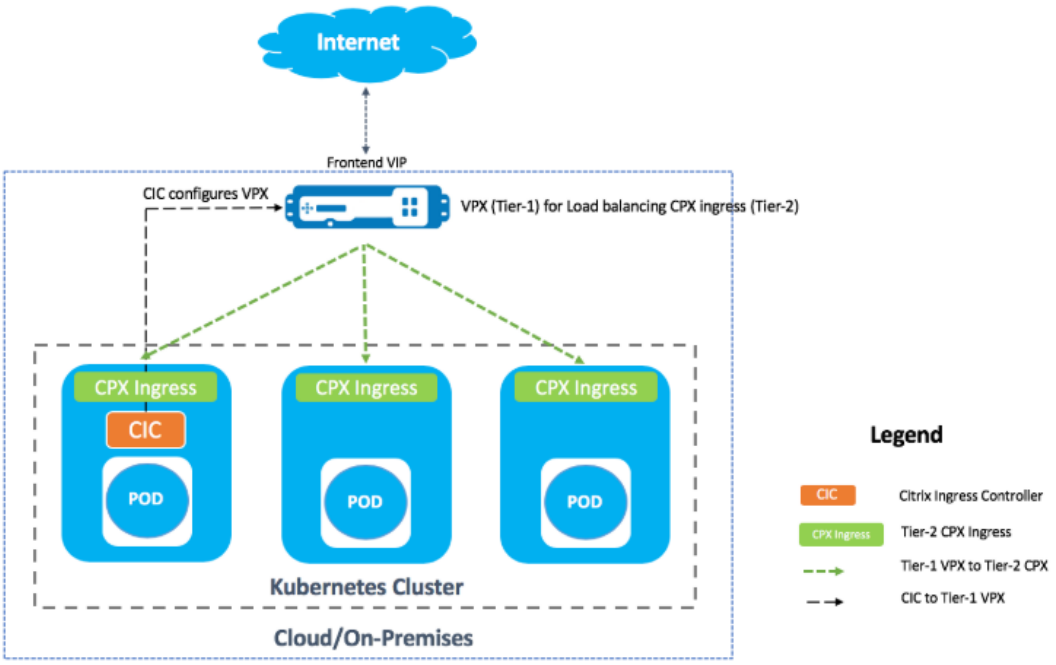
   Citrix 提供了一个示例微服务 Web 应用程序来测试 GCP 上的双层应用程序拓扑。为了进行概念验证，我们还在示例文件中包含了以下组件：

   - Kubernetes YAML 文件中的示例 Hotdrink Web Service
   - Kubernetes YAML 文件中的示例 Colddrink Web Service
   - Kubernetes YAML 文件中的示例 Guestbook Web Service
   - Kubernetes YAML 文件中的示例 Grafana Charting Service
   - Kubernetes YAML 文件中的示例 Prometheus Logging Service



4. 将第 **1** 层 **Citrix ADC** 自动化的 **Citrix Ingress Controller** 部署到 **GKE** 群集。

   围绕 Kubernetes 构建的 Citrix Ingress Controller 会根据入口资源配置自动配置一个或多个 Citrix ADC。Ingress Controller 是一种控制器，它监视 Kubernetes API 服务器以获取入口资源的更新，并相应地重新配置入口负载平衡器。Citrix Ingress Controller 可以直接使用 YAML 文件部署，也可以通过 Helm 图部署。



   Citrix 为第 1 层 VPX 实例的 Citrix Ingress Controller 自动化提供了示例 YAML 文件。这些文件在第 1 层 VPX 上自动执行多个配置，包括：

   - 重写策略和操作
   - 响应者策略和操作
   - 内容切换 URL 规则
   - 添加/删除 CPX 负载平衡服务

   用于 GCP 的 Citrix Ingress Controller YAML 文件位于此处：
   https://github.com/citrix/example-cpx-vpx-for-kubernetes-2-tier-microservices/tree/master/gcp

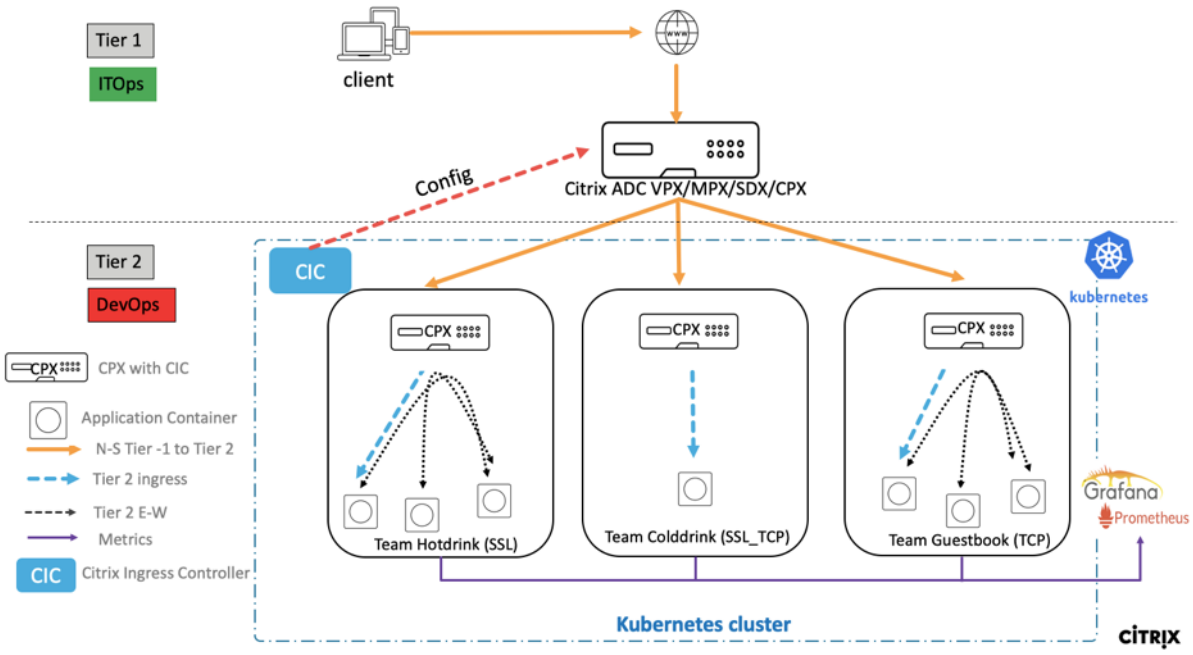
GCP 上的两层入口部署

在双层入口部署中，将 Citrix ADC VPX/MPX 部署在 Kubernetes 群集（第 1 层）外部，并在 Kubernetes 群集（第 2 层）内部部署 Citrix ADC CPX。

第 1 层 VPX/MPX 将在 Kubernetes 群集内部平衡第 2 层 CPX 的负载。这是一个广泛遵循的通用部署模型，无论平台是 Google 云、Amazon Web Services、Azure 还是本地部署。

第 1 层 VPX/MPX 的自动化

第 1 层 VPX/MPX 会自动对第 2 层 CPX 进行负载平衡。Citrix Ingress Controller 通过在 Kubernetes 群集中作为容器运行来完成自动化配置。它为第 1 层 VPX/MPX 配置单独的入口类，以便配置不会与其他入口资源重叠。



Citrix 部署概述

在 GCP 上安装和配置第 1 层 Citrix ADC

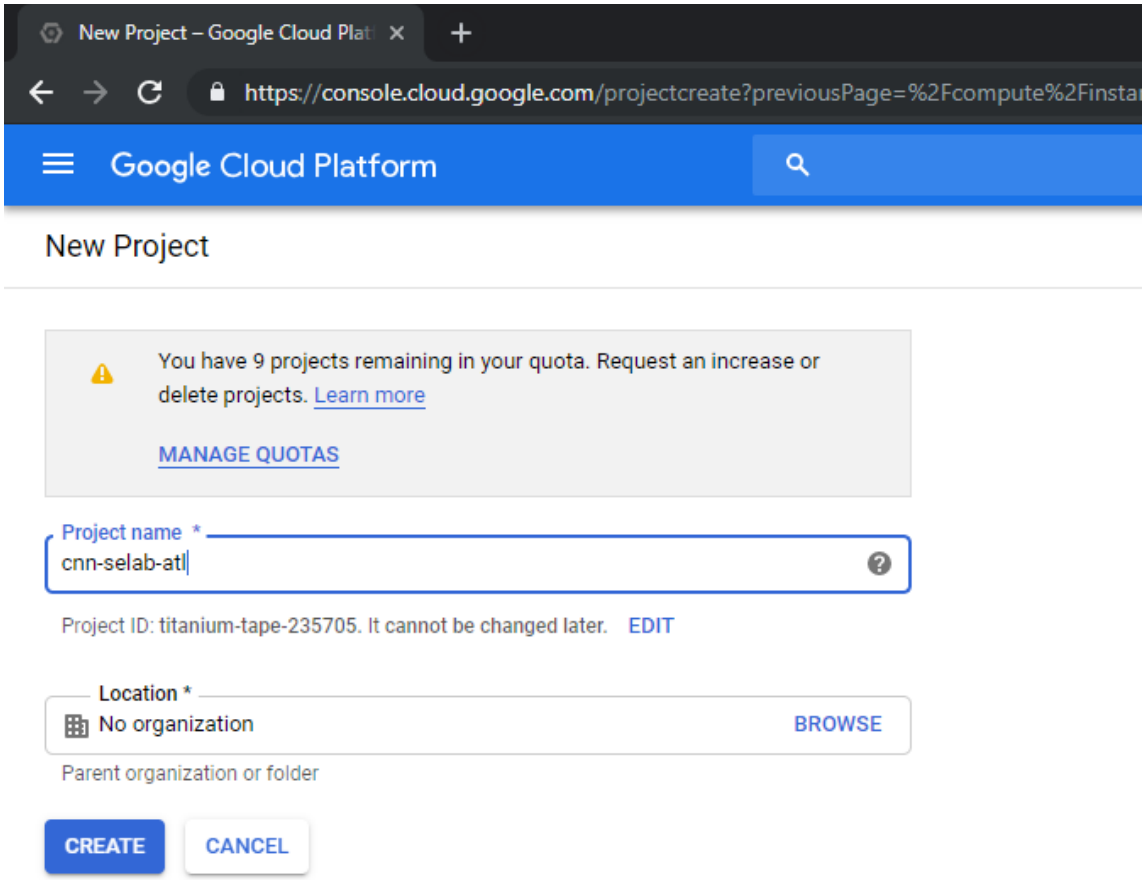
您可以使用以下方法之一部署 Citrix ADC：

- **Google 云端平台 GUI**：有关通过 GUI 在 Google 云端平台上配置第 1 层 Citrix ADC 的信息，请参阅 部署 Citrix ADC VPX 实例 。
- **Google Deployment Manager**：有关通过 GDM 模板在 Google 云端平台上配置第 1 层 Citrix ADC 的信息，请参阅 使用 GDM 模板部署 Citrix ADC VPX 实例 。

现在，您需要使用 3-NIC GDM 模板部署 Citrix VPX (tier-1-adc)。

先决条件（强制性）：

1. 仅使用您的 Citrix 邮件 ID 创建 GCP 帐户 http://console.cloud.google.com
2. 在 GCP 控制台上创建 cnn-Selab-atl 作为项目名称：



3. 在您的设备上安装该 gcloud 实用程序。按照链接查找实用程序：https://cloud.google.com/sdk/install。
4. 使用 gcloud API **gcloud auth login** 对您的 Google 帐户进行身份验证。
5. 在您的客户端上安装库贝特尔：https://kubernetes.io/docs/tasks/tools/install-kubectl/
6. 在 gcloud 实用程序上运行以下命令以创建映像。

```
gcloud compute images create netscaler12-1 --source-uri=gs://tme-
    cpx-storage/NSVPX-GCP-12.1-50.28_nc.tar.gz --guest-os-features=
    MULTI_IP_SUBNET
<!--NeedCopy-->
```

创建映像可能需要一些时间。创建映像后，它将显示在 GCP 控制台的 “计算”>“计算引擎” 下。

在 GCP 上部署 Citrix VPX (tier-1-adc)

1. **GCP VPC** 实例：
   出于安全目的解决外部网络、内部网络和 DMZ 网络的分离问题。我们必须创建三个 NIC，如下表所示：

网络	备注
192.168.10.0/24	管理网络 (vpx-snet-mgmt)
172.16.10.0/24	客户端网络 (vpx-snet-vip)
10.10.10.0/24	服务器网络 (vpx-snet-snip)

> 注意：
>
> 在部署任何虚拟机实例之前，构建三臂网络 VPC。

VPC 可以使用 gcloud API 或者通过 Google 云端平台控制台由 SDK 进行创建

通过 ***gcloud API*** 创建 ***VPC***

为管理或 NSIP 流量创建 VPC

```
gcloud compute --project=cnn-selab-atl networks create vpx-snet-mgmt --subnet-mode=custom
gcloud compute --project=cnn-selab-atl networks subnets create vpx-snet-mgmt --network=vpx-snet-mgmt --region=us-east1 --range=192.168.10.0/24
<!--NeedCopy-->
```

为客户端或 VIP 流量创建 VPC

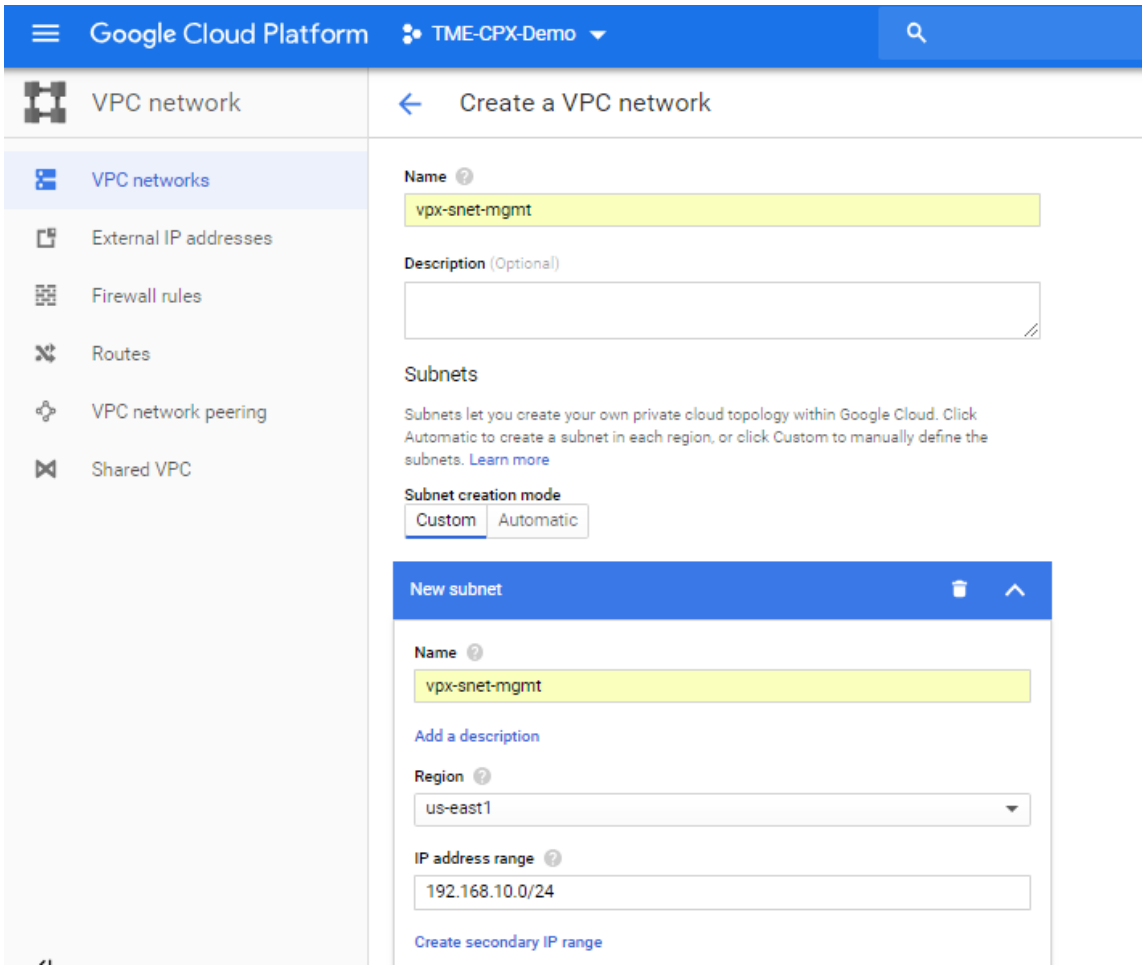
```
gcloud compute --project=cnn-selab-atl networks create vpx-snet-vip --subnet-mode=custom
gcloud compute --project=cnn-selab-atl networks subnets create vpx-snet-vip --network=vpx-snet-vip --region=us-east1 --range=172.16.10.0/24
<!--NeedCopy-->
```

为托管您的 kubernetes 群集的服务器或 SNIP 流量创建 VPC

```
gcloud compute --project=cnn-selab-atl networks create vpx-snet-snip --subnet-mode=custom
gcloud compute --project=cnn-selab-atl networks subnets create vpx-snet-snip --network=vpx-snet-snip --region=us-east1 --range=10.10.10.0/24
<!--NeedCopy-->
```


通过 ***GCP GUI*** 控制台创建 ***VPC***

在 Google 控制台中，选择网络连接 **> VPC** 网络 **>** 创建 **VPC** 网络，然后输入必填字段，如下所示。然后单击创建。



同样，为客户端和服务器端 NIC 创建 VPC 网络以创建三个子网。

> 注意：
>
> 所有三个 VPC 网络都应位于同一区域，在这种情况下为 us-east1。

VPC networks CREATE VPC NETWORK REFRESH

Name	Region	Subnets	Mode	IP addresses ranges	Gateways
vpx-snet-vip		1	Custom		
	us-east1	vpx-snet-vip		172.16.10.0/24	172.16.10.1
vpx-snet-snip		1	Custom		
	us-east1	vpx-snet-snip		10.10.10.0/24	10.10.10.1
vpx-snet-mgmt		1	Custom		
	us-east1	vpx-snet-mgmt		192.168.10.0/24	192.168.10.1

2. 在 **VPC** 网络下创建三个网络和三个子网后，请使用 GDM 模板部署 Citrix ADC VPX 实例。确保 **configuration.yml** 和 **template.py** 位于同一文件夹或目录中。使用 Google 软件开发工具包中的以下命令来部署实例。

```
1 gcloud deployment-manager deployments create tier1-vpx --config
     configuration.yml
2 <!--NeedCopy-->
```

3. 成功部署后，转到计算引擎检查 **citrix-adc-tier1-vpx** 部分，并验证内部 IP。

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP
nic0	vpx-snet-mgmt	vpx-snet-mgmt	192.168.10.20	—	static-external-mgmt-ip-tier1-vpx (35.196.91.86)
nic1	vpx-snet-vip	vpx-snet-vip	172.16.10.20	—	static-external-traffic-ip-tier1-vpx (34.73.141.54)
nic2	vpx-snet-snip	vpx-snet-snip	10.10.10.20	—	None

Citrix Ingress Controller 可以自动执行层 1 VPX 中的静态路由配置。配置应属于 Kubernetes 群集的同一子网/虚

拟私有云的子网 IP (SNIP) 地址。

> 注意：
>
> 部署的第 1 层 VPX/MPX 将负载平衡 Kubernetes 群集内的 CPX。在第 1 层 VPX 中配置 SNIP。

在第 1 层 VPX 上的 PuTTY 会话中，完成以下命令以添加 SNIP 并启用对 SNIP 的管理访问：

```
clear config -force full
add ns ip 10.10.10.20 255.255.255.0 -type snip -mgmt enabled
enable ns mode mbf
<!--NeedCopy-->
```

使用 **GKE** 部署 **Kubernetes** 群集

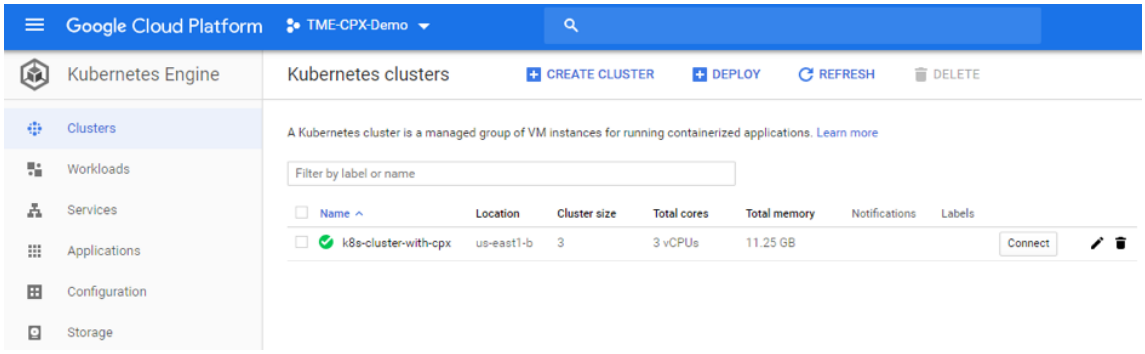
人们可以通过 ***Google*** 云端 ***SDK*** 或通过 ***Google*** 云端平台 ***GUI*** 控制台部署 Kubernetes 群集。

用于创建 **k8s** 群集的 **Gcloud API** 命令

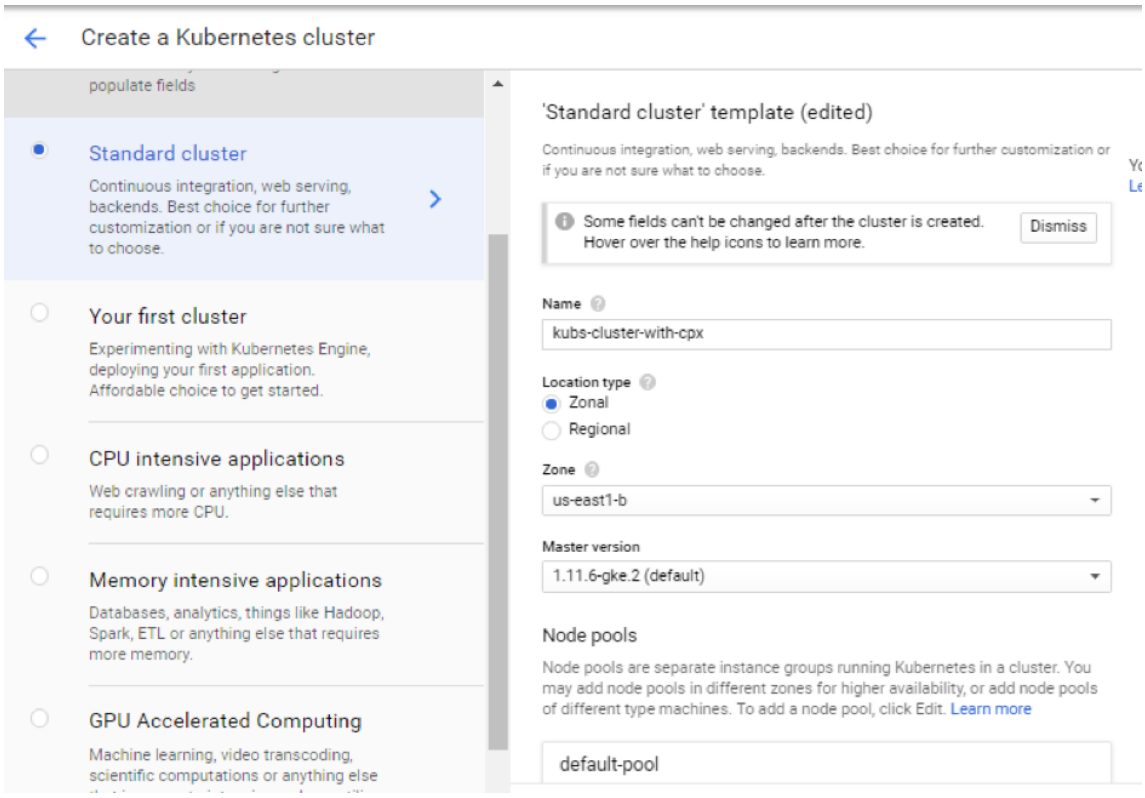
```
gcloud beta container --project "cnn-selab-atl" clusters create "k8s-
    cluster-with-cpx" --zone "us-east1-b" --username "admin" --cluster-
    version "1.11.7-gke.12" --machine-type "n1-standard-1" --image-type
    "COS" --disk-type "pd-standard" --disk-size "100" --scopes "https://
    www.googleapis.com/auth/devstorage.read_only","https://www.
    googleapis.com/auth/logging.write","https://www.googleapis.com/auth/
    monitoring","https://www.googleapis.com/auth/servicecontrol","https
    ://www.googleapis.com/auth/service.management.readonly","https://www
    .googleapis.com/auth/trace.append" --num-nodes "3" --enable-cloud-
    logging --enable-cloud-monitoring --no-enable-ip-alias --network "
    projects/cnn-selab-atl/global/networks/vpx-snet-snip" --subnetwork "
    projects/cnn-selab-atl/regions/us-east1/subnetworks/vpx-snet-snip"
    --addons HorizontalPodAutoscaling,HttpLoadBalancing --enable-
    autoupgrade --enable-autorepair
<!--NeedCopy-->
```

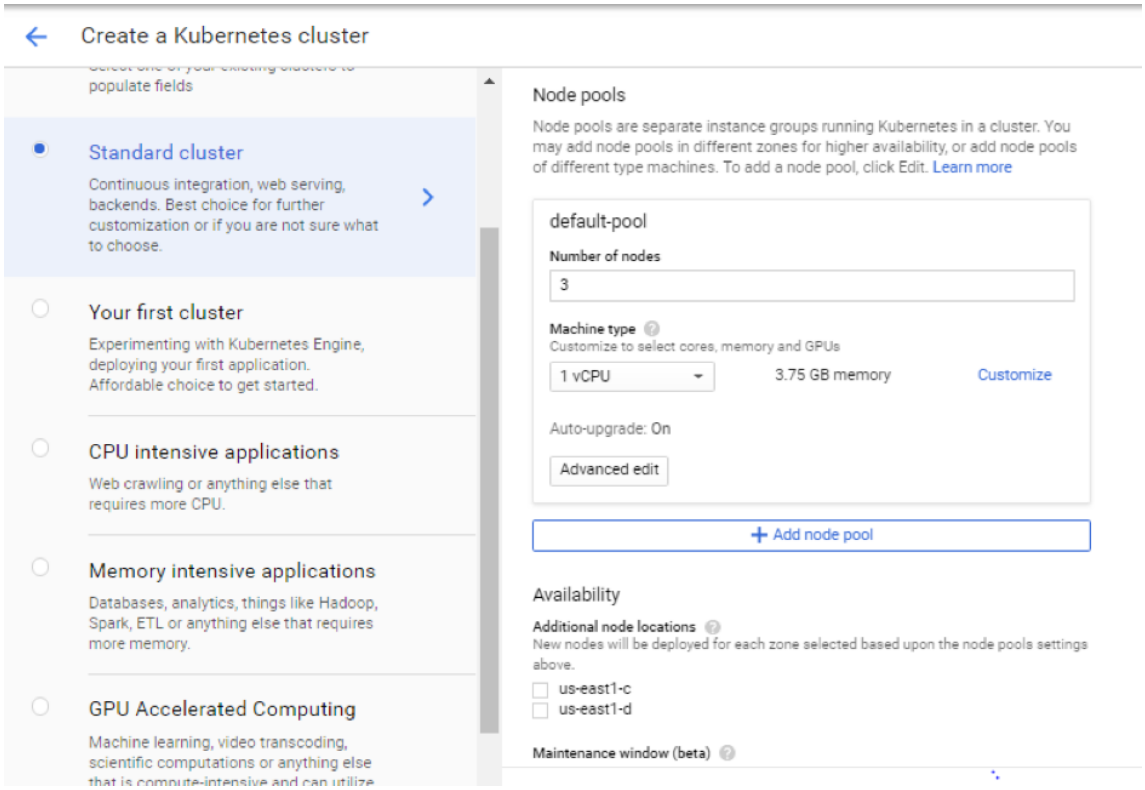
Google 云端平台 **GUI** 控制台步骤

1. 在 GCP 控制台上搜索 Kubernetes 引擎，然后单击 创建群集。

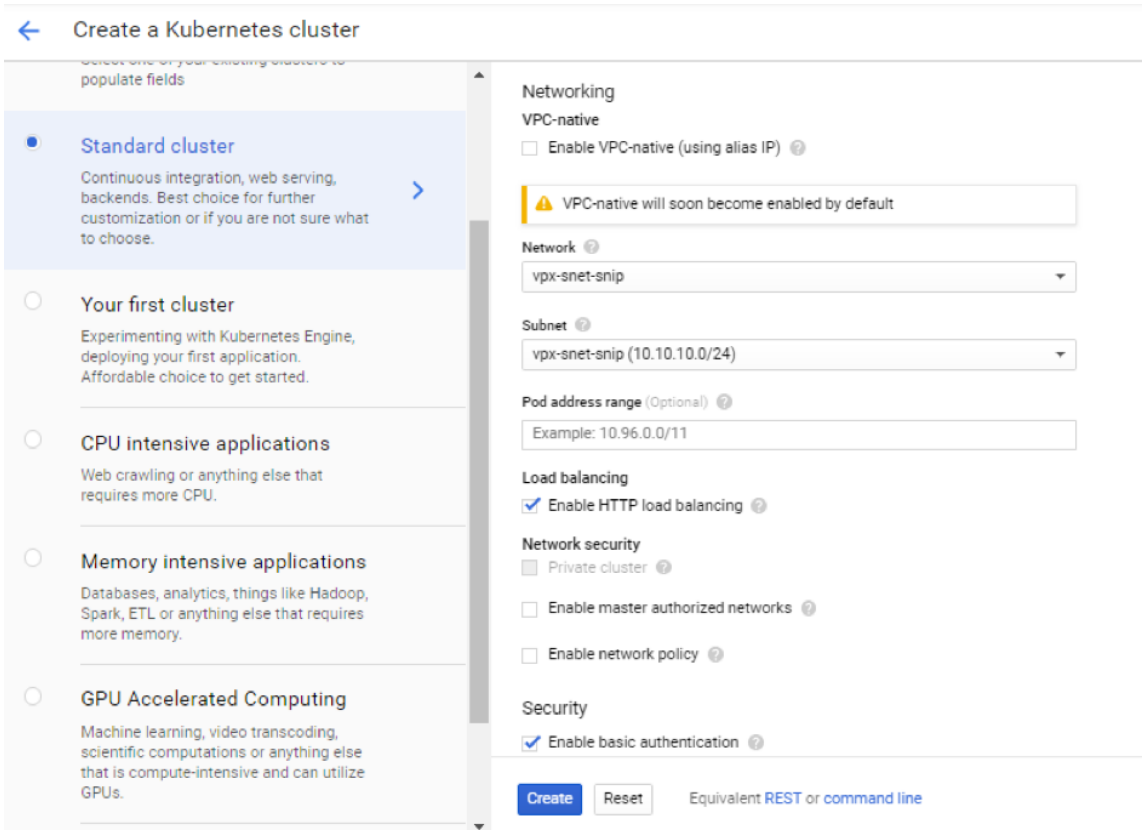


2. 在您的 VPX SNIP 所在的同一子网中创建群集 (vpx-snet-snip)。此群集可自动将配置从 K8s 群集中的 Citrix Ingress Controller 向 1 级 ADC 推送。

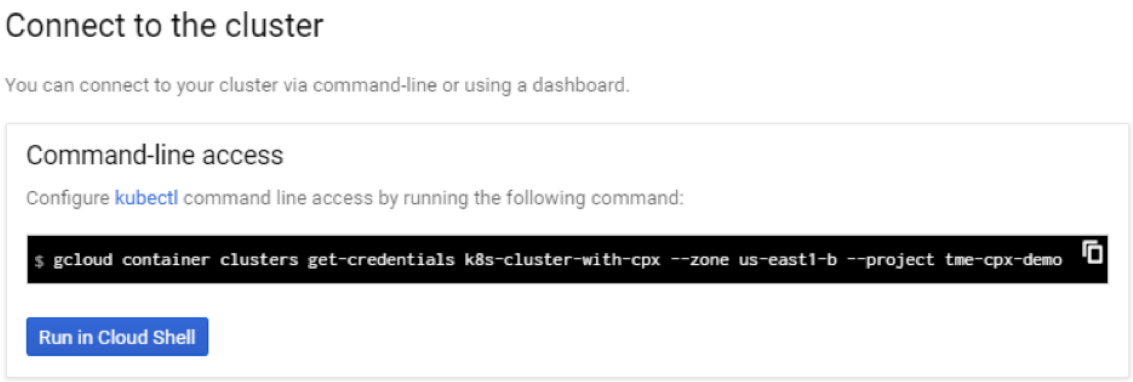




3. 单击“高级选项”将子网更改为`vpx-snet-snip`，然后选择以下字段。



4. 要从云 SDK 访问此群集，请单击 Kubernetes 连接到群集按钮，然后将命令粘贴到云 SDK 中。



5. 通过运行以下命令验证 GKE 群集部署：

```
kubectl get nodes
<!--NeedCopy-->
```

```
C:\Program Files (x86)\Google\Cloud SDK\GCP_SE_2019\Multitier-Hairpin>kubectl get nodes
NAME                                              STATUS    ROLES     AGE       VERSION
gke-k8s-cluster-with-cpx-default-pool-8c0d6de1-l15t   Ready     <none>    2d        v1.11.6-gke.2
gke-k8s-cluster-with-cpx-default-pool-8c0d6de1-xbvf   Ready     <none>    2d        v1.11.6-gke.2
gke-k8s-cluster-with-cpx-default-pool-8c0d6de1-xs47   Ready     <none>    2d        v1.11.6-gke.2
```

使用示例 **YAML** 文件库部署示例应用程序

Citrix ADC 提供双层体系结构部署解决方案，用于对微服务中部署并通过 Internet 访问的企业级应用程序进行负载平衡。层 1 具有重负载平衡器，如 VPX/SDX/MPX，用于平衡南北流量。第 2 层具有 CPX 部署，用于管理微服务和负载平衡东西流量。

1. 如果您在 GKE 中运行群集，请确保您已使用群集角色绑定来配置群集管理员。您可以使用以下命令执行此操作。

```
kubectl create clusterrolebinding citrix-cluster-admin --
    clusterrole=cluster-admin --user=<email-id of your google
    account>.
<!--NeedCopy-->
```

2. 访问您拥有部署 YAML 文件的当前目录。运行以下命令以获取节点状态。

```
kubectl get nodes
<!--NeedCopy-->
```



```
C:\Program Files (x86)\Google\Cloud SDK\GCP_SE_2019\Multitier-Hairpin>kubectl get nodes
NAME                                                STATUS    ROLES     AGE       VERSION
gke-k8s-cluster-with-cpx-default-pool-8c0d6de1-l15t   Ready     <none>    2d        v1.11.6-gke.2
gke-k8s-cluster-with-cpx-default-pool-8c0d6de1-xbvf   Ready     <none>    2d        v1.11.6-gke.2
gke-k8s-cluster-with-cpx-default-pool-8c0d6de1-xs47   Ready     <none>    2d        v1.11.6-gke.2
```

3. 创建命名空间：

```
kubectl create -f namespace.yaml
<!--NeedCopy-->
```

验证命名空间命令：

```
kubectl get namespaces
<!--NeedCopy-->
```

```
C:\Program Files (x86)\Google\Cloud SDK\GCP_SE_2019\Multitier-Hairpin>kubectl get namespaces
NAME             STATUS    AGE
default          Active    2d
kube-public      Active    2d
kube-system      Active    2d
monitoring       Active    10h
team-colddrink   Active    10h
team-guestbook   Active    10h
team-hotdrink    Active    10h
tier-2-adc       Active    10h
```

4. 在默认命名空间中部署 rbac.yaml。

```
kubectl create -f rbac.yaml
<!--NeedCopy-->
```

5. 使用以下命令为 hotdrink、colddrink 和 guestbook 微服务部署 CPX。

```
kubectl create -f cpx.yaml -n tier-2-adc
kubectl create -f hotdrink-secret.yaml -n tier-2-adc
<!--NeedCopy-->
```

6. 部署 three-hotdrink beverage 微服务 — — 带有发夹体系结构的 SSL 型微服务。

```
kubectl create -f team_hotdrink.yaml -n team-hotdrink
kubectl create -f hotdrink-secret.yaml -n team-hotdrink
<!--NeedCopy-->
```


7. 部署 colddrink beverage 微服务 — SSL_TCP 类型的微服务。

```
kubectl create -f team_colddrink.yaml -n team-colddrink
kubectl create -f colddrink-secret.yaml -n team-colddrink
<!--NeedCopy-->
```

8. 部署 guestbook - 一个 NoSQL 类型的微服务。

```
kubectl create -f team_guestbook.yaml -n team-guestbook
<!--NeedCopy-->
```

9. 验证为上述三个应用程序部署的 CPX。首先，获取部署为 tier-2-adc 的 CPX 容器，然后获得 CPX 的 CLI 访问权限。

   ‘‘‘

   To get CPX pods in tier-2-adc namespace, enter: kubectl get pods -n tier-2-adc

   要获得 CLI 访问（bash）到 CPX 窗格 (hotdrinks-cpx pod)，请输入：`kubectl exec -it ”copy and paste hotdrink CPX pod name from the above step”bash -n tier-2-adc`。

   例如，

   `kubectl exec -it cpx-ingress-hotdrinks-768b674f76-pcnw4 bash -n tier-2-adc`

   要检查 CS 虚拟服务器是否在 hotdrink-cpx 中运行，请在根访问 CPX 后输入以下命令：`cli-script”sh csvs”`。

   例如，

   `root@cpx-ingress-hotdrinks-768b674f76-pcnw4:/## cli_script.sh ”sh csvs”`

10. 将 VPX 入口和 Ingress Controller 部署到层 2 命名空间，该命名空间会自动配置 VPX。Citrix Ingress Controller (CIC) 自动执行 tier-1-adc (VPX)。

```
kubectl create -f ingress_vpx.yaml -n tier-2-adc
kubectl create -f cic_vpx.yaml -n tier-2-adc
<!--NeedCopy-->
```

11. 将 DNS 条目添加到本地计算机的主机文件中，以便通过 Internet 访问微服务。

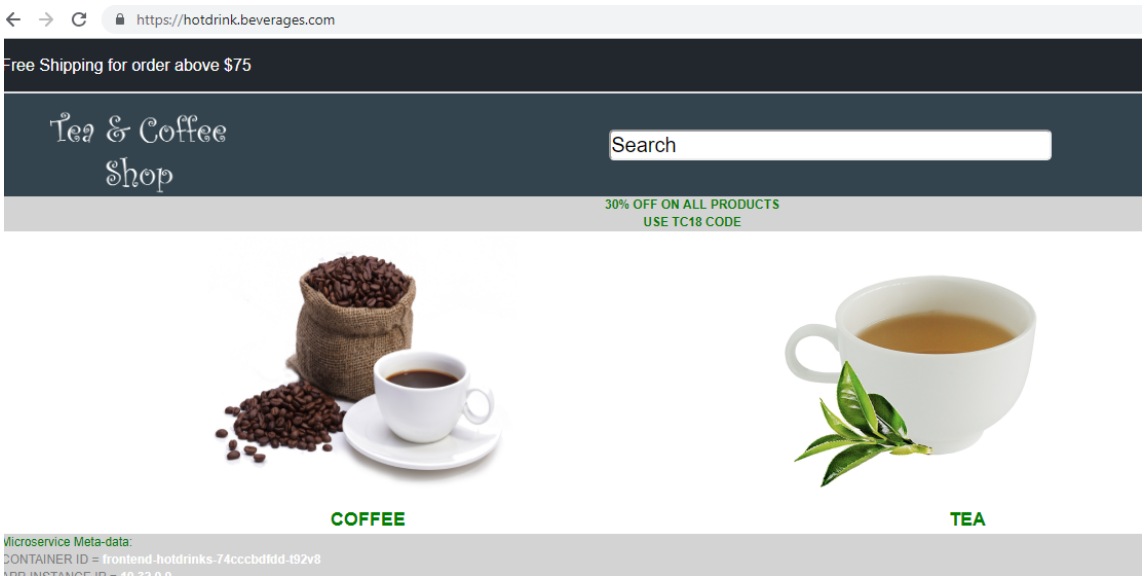
    对于 Windows 客户端，请转到 **C:\Windows\System32\drivers\etc\hosts**

    对于 macOS 客户端，请在终端中输入：**sudo nano /etc/hosts‘**

在主机的文件中添加以下条目并保存文件。

```
hotdrink.beverages.com   xxx.xxx.xxx.xxx (static-external-traffic
    -ip-tier1-vpx)
colddrink.beverages.com  xxx.xxx.xxx.xxx (static-external-traffic
    -ip-tier1-vpx)
guestbook.beverages.com  xxx.xxx.xxx.xxx (static-external-traffic
    -ip-tier1-vpx)
grafana.beverages.com    xxx.xxx.xxx.xxx (static-external-traffic
    -ip-tier1-vpx)
prometheus.beverages.com xxx.xxx.xxx.xxx (static-external-traffic
    -ip-tier1-vpx)
<!--NeedCopy-->
```

12. 现在，您可以通过互联网访问每个应用程序。例如 https://hotdrink.beverages.com。



为示例应用程序启用重写策略和响应程序策略

现在是时候通过自定义资源定义 (CRD) 在 VPX 上推送重写策略和响应程序策略。

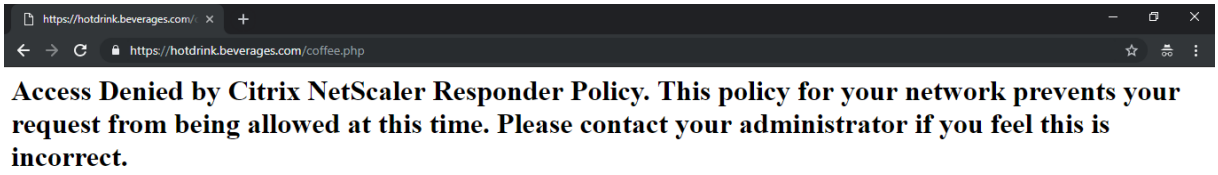
1. 部署 CRD 以将重写策略和响应程序策略推送到默认命名空间中的第 1-adc。

```
    kubectl create -f crd_rewrite_responder.yaml
    <!--NeedCopy-->
```


1. 黑名单 **URL** 配置 hotdrink.beverages.com 上的响应程序策略以阻止对咖啡页的访问。

```
kubectl create -f responderpolicy_hotdrink.yaml -n tier-2-adc
<!--NeedCopy-->
```

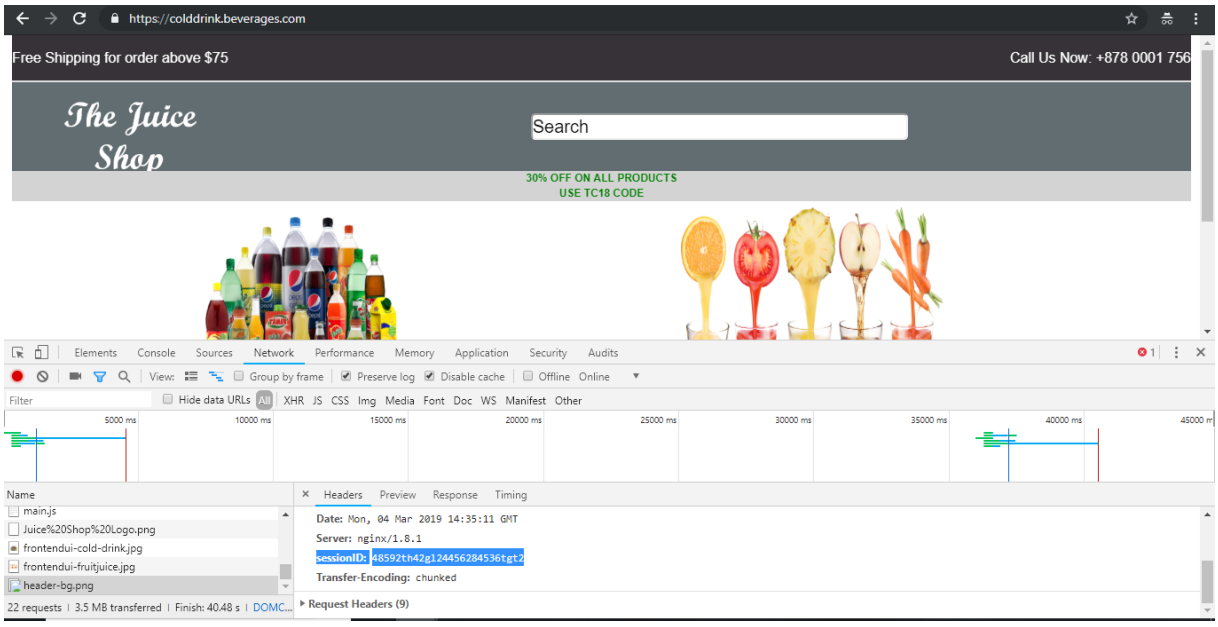
部署响应者策略后，访问 hotdrink.beverages.com 上的咖啡页。然后您会收到以下消息。



1. 标头插入配置 colddrink.beverages.com 上的重写策略以在标头中插入会话 ID。

```
kubectl create -f rewritepolicy_colddrink.yaml -n tier-2-adc
<!--NeedCopy-->
```

部署重写策略后，访问在浏览器上启用了开发人员模式的 colddrink.beverages.com。在 Chrome 中，按 F12 并保留网络类别中的日志以查看会话 ID，该 ID 是由第 1-adc (VPX) 上的重写策略插入的。



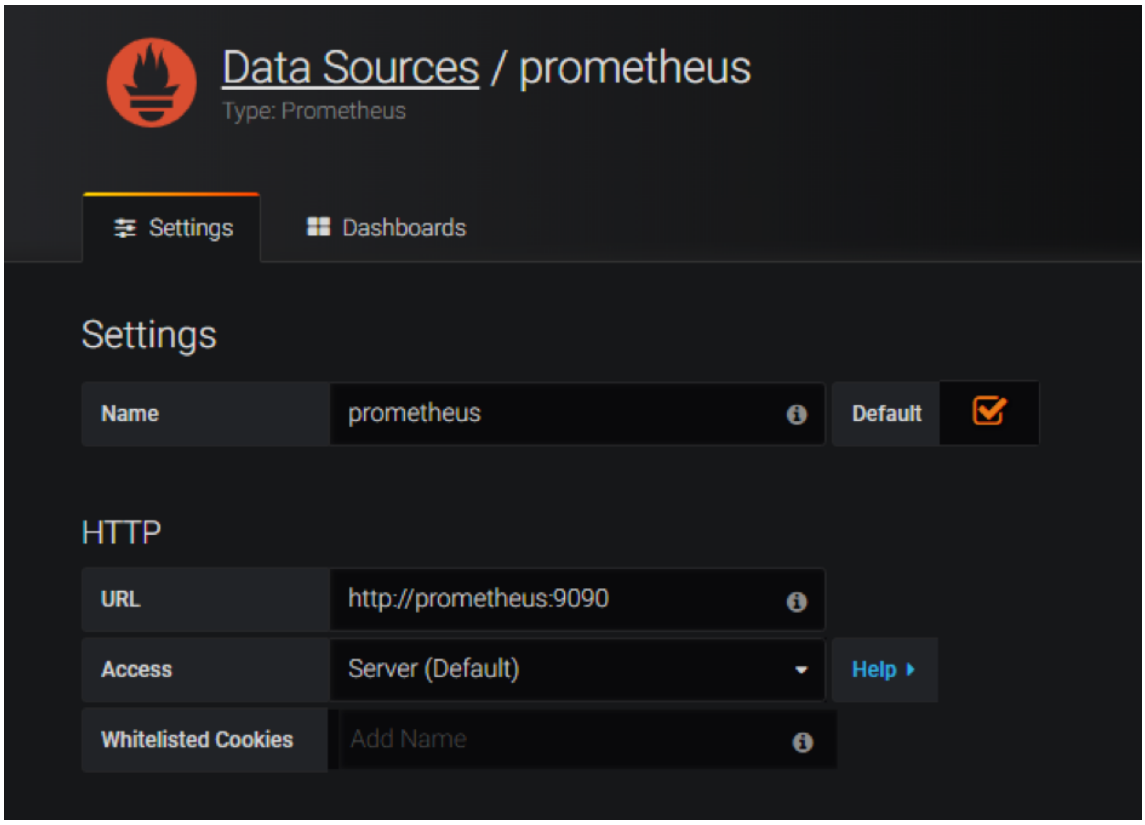
开源工具集

1. 部署 Cloud Native Computing Foundation (CNCF) 监视工具，例如 Prometheus 和 Grafana，以收集 ADC 代理统计数据。

```
kubectl create -f monitoring.yaml -n monitoring
kubectl create -f ingress_vpx_monitoring.yaml -n monitoring
<!--NeedCopy-->
```

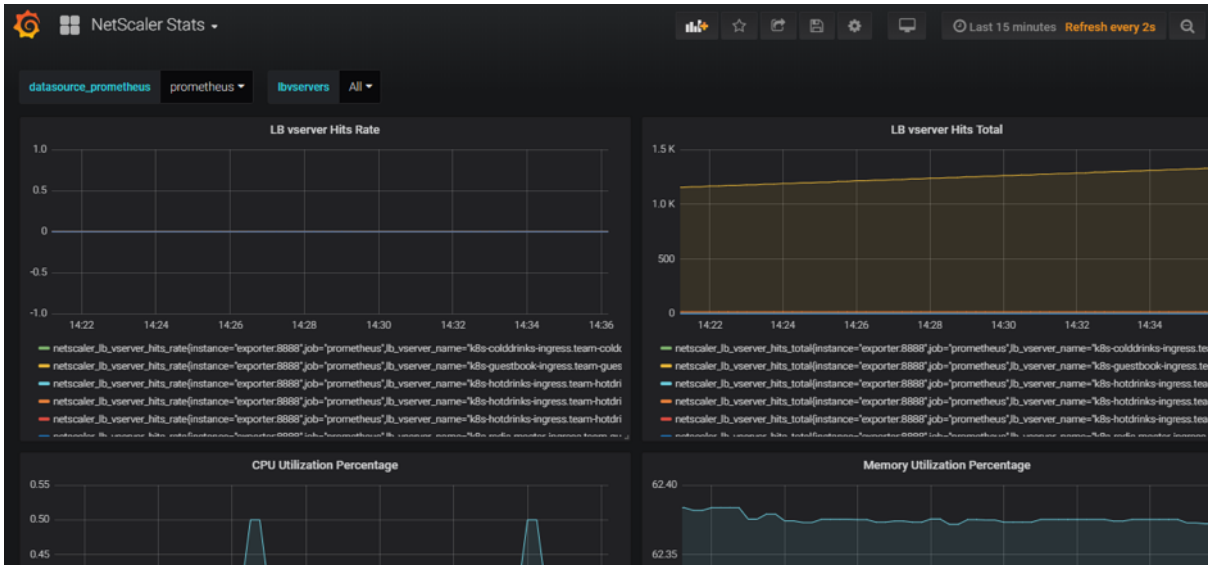
Prometheus 日志聚合器

1. 登录 http://grafana.beverages.com 并完成以下一次性设置。
   a) 使用管理员凭据登录门户。
   b) 单击添加数据源，然后选择 **Prometheus** 数据源。
   c) 配置以下设置，然后单击“保存并测试”按钮。



Grafana 可视化控制板

1. 从左侧面板中，选择“导入”选项并上传文grafana_config.json 件yamlFiles 夹中提供的文件。现在，您可以看到包含基本 ADC 统计信息的 Grafana 控制板。

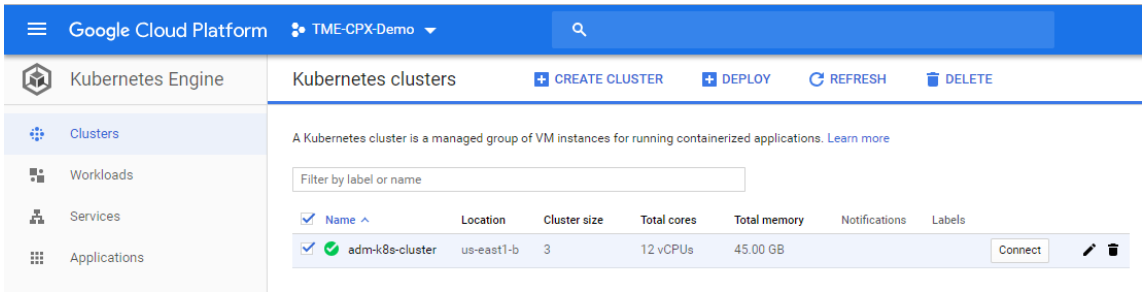


删除部署

1. 要删除 Citrix VPX (tier-1-adc) 部署，请转到 Google SDK CLI 控制台以删除实例：

```
gcloud deployment-manager deployments delete tier1-vpx
<!--NeedCopy-->
```

2. 要删除 GKE Kubernetes 群集，请转到 GCP 控制台，选择 kubernetes 群集，然后单击删除以擦除群集。



Citrix ADC 群集验证的参考设计

May 20, 2020

Citrix ADC 群集是一组 Citrix ADC nCore 设备，作为单个系统映像一起工作。群集的每个设备都称为节点。Citrix ADC 群集可以包含至少 2 个或多达 32 个 Citrix ADC nCore 硬件或虚拟设备作为节点。

客户端流量在节点之间分布，以提供高可用性、高吞吐量和可扩展性。

要创建群集，请将所需的 Citrix ADC 装置添加为群集节点，设置节点之间的通信，设置客户端和服务器网络的链接，配置 Citrix ADC 装置，并配置客户端和服务器通信的分布。

群集支持的 **Citrix ADC** 功能

下表列出了群集上完全支持的 Citrix ADC 功能，这些功能仅适用于单个群集节点，而且群集不支持的功能。

Citrix ADC 功能支持列表：

支持的功能	节点级别支持的功能	不受支持的功能
负载平衡	浪涌保护	DNS 负载平衡
负载平衡持久性	Sure Connect	FTP 负载平衡
SIP	优先队列	全局服务器负载平衡 (GSLB)
maxClient	HTTP 拒绝服务保护 (HTTP DoSP)	Citrix ADC 推送
溢出	集成缓存	RTSP
SSL PI 策略	Call Home	有状态连接故障转移
内容交换		正常关机
缓存重定向		DBS Auto Scaling
压缩控制		使用 TOS 的 DSR
内容过滤		基于带宽的溢出
OSPF（IPv4 和 IPv6）		更精细的启动-RR 控制
RIP（IPv4 和 IPv6）		速率限制
BGP（IPv4 和 IPv6）		Stream Analytics
HTML 注入		网络配置文件
TCP 缓存		DNS 缓存
分布式拒绝服务 (DDoS)		SSL-VPN
基本网络（IPv4 和 IPv6）		SSL CPE 策略
VLAN		应用程序防火墙
ICMP		AAA
碎片化		云桥接通道
基于 MAC 的转发 (MBF)		第 2 层模式

支持的功能	节点级别支持的功能	不受支持的功能
RNAT		FIPS
INAT		XML XSM
KRPC		AAA-TM
ACL		VMAC/VRRP
简单 ACL		链路负载平衡
PBR		IP 隧道
SNMP GET/SET，Walk		DHCP RA
SNMP 陷阱		Bridge Group
策略基础结构 (PE/PI)		网络桥接
NITRO API		Citrix ADC 上的 Web Interface
AppExpert		EdgeSight 监视
重写		BR LB
响应方		ISIS 路由
使用源 IP (USIP)		FIS（故障转移接口集）
AppFlow 导出程序和 AppFlow 收集器（客户端）与瀑布图		
DataStream		
MSR		
基于策略的 RNAT		
网络日志记录		
审计（系统日志和索托日志）		
路径 MTU 发现		
客户端保持活动状态		

硬件和软件要求

添加到 Citrix ADC 群集的装置必须满足以下要求：

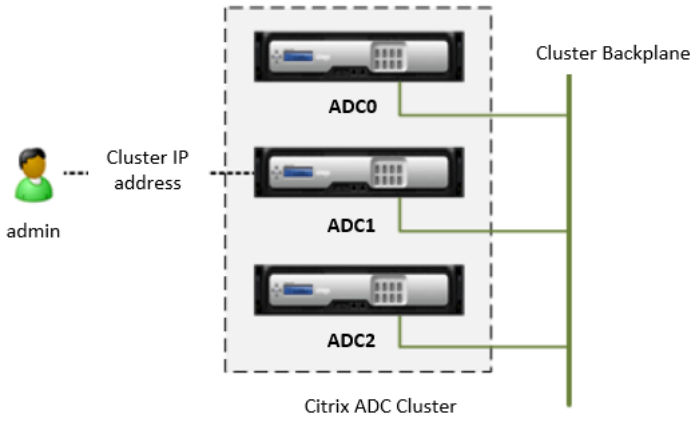
- 是 Citrix ADC nCore 设备。不支持 Citrix ADC Classic 设备的群集。
- 具有相同的平台类型（物理设备或 VPX 实例）。
- 具有相同的硬件类型（对于物理设备）。

- 位于同一子网上。
- 拥有群集许可证文件。
- 具有相同的许可证（标准许可证、企业级或白金级许可证以及任何附加许可证）。
- 具有相同的软件版本和版本。
- 初始配置并连接到公共客户端和服务器端网络。

群集的工作原理

Citrix ADC 群集是通过对满足中指定要求的 Citrix ADC 装置进行分组来形成的 硬件和软件要求。其中一个群集节点被指定为配置协调器 (CCO)。顾名思义，CCO 通过群集的管理 IP 地址（称为群集 IP 地址）来协调所有群集配置。

必须通过群集 IP 地址访问 CCO 来配置群集，如下图所示：



> 注意：
>
> 无法通过通过 Citrix ADC IP (ADCIP) 地址访问单个节点来配置单个节点。通过 ADCIP 地址访问的节点在只读模式下可用。这意味着您只能查看配置和统计信息。但是，有些命令可以在单个节点上执行。有关详细信息，请参阅单个节点上支持的操作。

您在群集上定义的 VIP 地址可用于群集的所有节点（条带地址）。您可以将 SNIP 地址定义为在所有节点（条带地址）上可用，也可以仅在单个节点（斑点地址）上使用。群集中流量分布的详细信息取决于使用的算法，但在每种情况下，相同的逻辑实体都会处理流量。

群集同步

将节点添加到群集时，Citrix ADC 配置和 CCO 上可用的文件（例如 SSL 证书、许可证和 DNS）将在新添加的群集节点上同步。这可确保群集的所有节点上始终同步配置和文件。

当现有群集节点重新加入群集时（在群集失败或故意禁用之后），群集将检查该节点上的可用配置。如果重新加入的节点和 CCO 上的可用配置不匹配，则使用以下方法之一同步节点：

- 完全同步。如果配置之间的差异超过 255 个命令，则在 CCO 上实现的所有配置都应用于重新加入群集的节点。在同步期间，节点在操作上保持不可用。

- 增量同步。如果配置之间的差异小于或等于 255 个命令，则仅将不可用的配置应用于重新加入群集的节点。节点的运行状态不受影响。

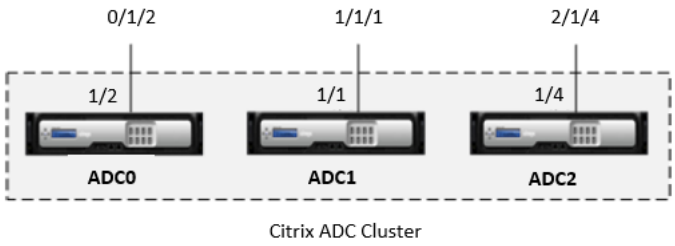
通过群集 IP 地址在 CCO 上执行的配置将自动传播到群集节点。由于群集配置基于可用节点的仲裁，因此只有当大多数节点处于同步状态时，才能将命令（在群集 IP 地址上执行）传播到其他群集节点。如果大多数节点未同步或正在同步过程中，则它们不能接受新命令，因此在同步完成之前不会传播命令。

群集连接

要标识接口所属的节点，标准 Citrix ADC 接口命名约定以节点 ID 作为前缀。也就是说，接口标识符 **c/u**（其中 **c** 是 Controller 编号，**u** 是单位编号）变为 **n/c/u**，其中 **n** 是节点 ID。

例如，在下图中，节点 0 的接口 **1/2** 表示为 **0/1/2**，节点 1 的接口 **1/1** 表示为 **1/1/1**，节点 2 的接口 **1/4** 表示为 **2/1/4**。

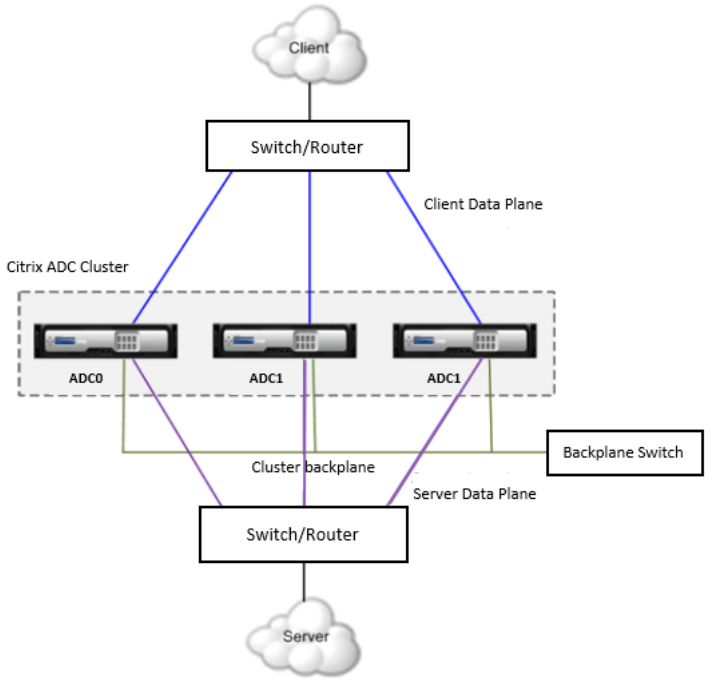
群集中的网络接口命名约定



群集通过群集节点和客户端连接设备之间的物理连接与客户端通信。这些物理连接的逻辑分组称为客户端数据平面。同样，群集通过群集节点和服务器端连接设备之间的物理连接与服务器通信。这些物理连接的逻辑分组称为服务器数据平面。

除了分别通过客户端数据平面和服务器数据平面与客户端和服务器通信之外，群集节点还通过群集背板相互通信。背板包括来自每个群集节点和背板交换机的物理连接，是群集系统的主干。

群集通信接口



上图显示了构成客户端数据平面、服务器数据平面和群集背板的物理连接的逻辑分组。

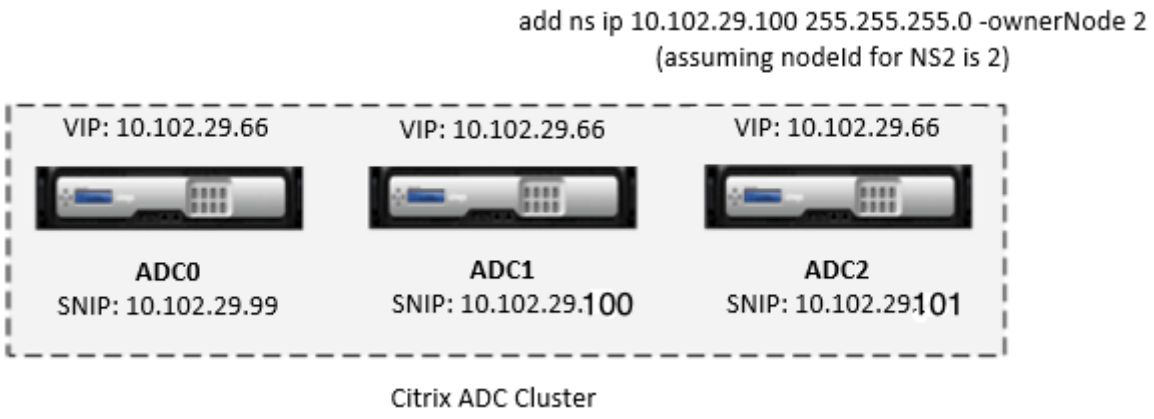
条带化和发现的 **IP** 地址

在群集部署中，VIP 和 SNIP 地址可以进行条带化或发现。

- 群集的所有节点上都处于活动状态的条带 *IP* 地址。在群集上配置的未指定所有者节点的 IP 地址在所有群集节点上都处于活动状态。
- 发现的 *IP* 地址在一个节点上处于活动状态，并且独占拥有。通过指定所有者节点在群集上配置的 IP 地址仅在指定为所有者的节点上处于活动状态。

下图显示了三节点群集中的条带化 IP 地址和发现 IP 地址。

具有条带和发现 **IP** 地址的三节点群集



在上图中，VIP 地址 10.102.29.66 在所有群集节点上进行条带化处理，而 SNIP 地址 10.102.29.99 在 ADC0 和 ADC1 上进行条带化处理。ADC2 具有一个发现的 SNIP 地址。

下表显示了 Citrix ADC 拥有的可以条带化或发现的 IP 地址：

条带化和发现的 **IP** 地址

Citrix ADC 拥有的 IP 地址	条带 IP 地址	发现的 IP 地址
ADCIP	否	是
群集 IP 地址	否	否
VIP	是	否
SNIP	是	是（推荐）

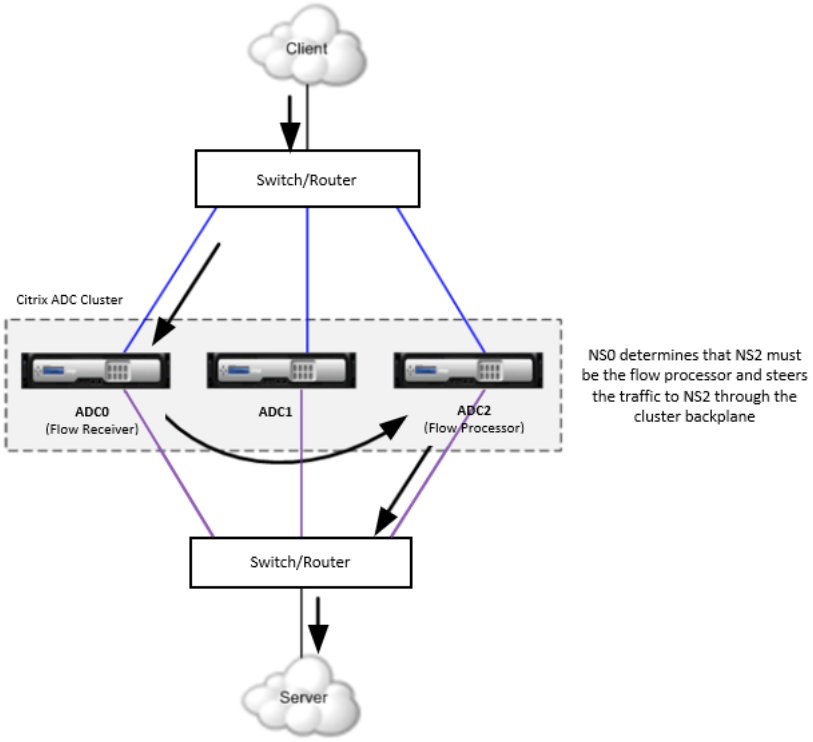
> 注意：
>
> - 群集 IP 地址不是条带化或发现的 IP 地址。它是一个由 CCO 拥有的浮动 IP 地址，该地址不是固定节点。
> - Citrix 建议您仅使用发现的 IP 地址。仅当缺少 IP 地址时，才能使用条带 IP 地址。使用条带 IP 地址可能会导致 ARP 通量问题。

流量分布

Citrix ADC 群集使用等成本多路径 (ECMP) 或群集链路聚合组 (CLAG) 流量分发机制来确定从外部连接设备接收流量（流量接收器）的节点。其中每个机制都使用不同的算法来确定流量接收器。然后，流量接收器使用内部群集逻辑来确定处理流量的节点（流量处理器）。

> 注意：
>
> 流量接收器和流量处理器必须是能够为流量提供服务的节点。

群集中的流量分布



上图显示了流经群集的客户端请求。客户端向条带化虚拟 IP (VIP) 地址发送请求。在客户端数据平面上配置的流量分配机制选择其中一个群集节点作为流量接收器。流量接收器接收流量，确定必须处理流量的节点，并通过群集背板将请求引导到该节点（除非流量接收器选择自己作为流量处理器）。

流量处理器与服务器建立连接。服务器处理请求并将响应发送到向服务器发送请求的子网 IP (SNIP) 地址。

- 如果 SNIP 地址是条带 IP 地址，则在服务器数据平面上配置的流量分配机制将选择一个群集节点（拥有 SNIP 地址）作为流量接收器。流量接收器接收流量，确定流量处理器，并通过群集背板将请求引导到流量处理器。
- 如果 SNIP 地址是发现的 IP 地址，则拥有该 SNIP 地址的节点会收到来自服务器的响应。

在非对称群集拓扑中（所有群集节点都未连接到外部交换机），必须专门使用 链路集或与 ECMP 或 CLAG 结合使用。有关详细信息，请参阅使用链接集。

群集和节点状态

群集节点分类包括三种类型的状态：管理状态、操作状态和运行状态。

- 管理状态。将节点添加到群集时会配置管理员状态。它指示节点的用途，该节点可以处于以下状态之一：
  - **ACTIVE**。处于此状态的节点如果流量正常运行，则为流量提供服务。
  - **PASSIVE**。处于此状态的节点不提供流量，但与群集同步。这些节点在维护活动期间很有用，因为它们可以在不从群集中删除节点的情况下进行升级。
  - **SPARE**。处于此状态的节点不提供流量，但与群集同步。备用节点充当群集的备份节点。如果其中一个 ACTIVE 节点不可用，则其中一个备用节点的操作状态变为 ACTIVE，并且该节点开始提供流量。

- 操作状态。当节点是群集的一部分时，其操作状态可以更改为 “ACTIVE”、“INACTIVE” 或 “UNKNOWN”。节点处于 “INACTIVE” 或 “UNKNOWN” 状态有多种原因。查看 `ns.log` 文件或错误计数器以帮助确定确切的原因。
- 运行状况。根据其运行状况，节点可以是 UP 或 NOT UP。要查看节点处于 NOT UP 状态的原因，请从群集 IP 地址为该节点运行 **show cluster node** 命令。

只有管理员状态为 “ACTIVE”、操作状态为 “ACTIVE”、运行状态为 “UP” 的节点才能提供流量。只有当最少 (n/2 + 1) 节点（其中 n 为群集节点数）能够服务流量时，群集才起作用。

设置 **Citrix ADC** 群集

要设置 Citrix ADC 群集，首先设置群集背板。然后，通过将第一个节点添加到群集（成为初始配置协调器 (CCO)），并通过为该节点分配群集 IP 地址来创建群集。在 CCO 上定义了群集 IP 地址后，您可以向群集添加更多节点。

要添加到群集的每个设备都必须：

- 是 Citrix ADC nCore 设备。不支持 Citrix ADC Classic 设备的群集。
- 具有相同的平台类型（物理设备或 VPX 实例）。
- 具有相同的硬件类型（对于物理设备）。
- 位于同一子网上。
- 拥有群集许可证文件。
- 具有相同的许可证（标准许可证、企业级或白金级许可证以及任何附加许可证）。
- 具有相同的软件版本和版本。
- 初始配置并连接到公共客户端和服务器端网络。

只有满足上述所有条件的装置才能成为 Citrix ADC 群集的一部分。

设置群集背板

群集中的节点通过群集后台相互通信。背板是一组连接，其中每个节点的一个接口连接到一个公共交换机（称为群集背板交换机）。群集的每个节点都使用一个特殊的 MAC 地址通过群集背板与其他节点进行通信。

> 注意：
>
> 在 XenServer 上部署的 VPX 设备群集（启用了 MAC 欺骗功能）中，NIC（XenServer 虚拟交换机）可以丢弃在背板上发送的数据包。因此，您必须确保在 XenServer 上禁用 MAC 欺骗。
> 您必须确保群集背板交换机支持大于 1,500 字节的数据包。

需要记住的几个要点：

- 请勿使用装置的管理界面 (0/1) 作为背板接口。
- 用于背板的接口不得用于客户端数据平面或服务器数据平面。
- 群集中所有节点的背板接口必须连接到同一交换机，并绑定到同一 L2 VLAN。默认情况下，背板接口在群集上配置的所有 L3 VLAN 上都存在。
- 如果您有多个具有相同群集实例 ID 的 Citrix ADC 群集，请确保每个群集的背板接口绑定到不同的 VLAN。

- Citrix 建议您仅为背板专用一个单独的交换机，以便无缝处理大量流量。
- 背板接口始终受到监视，无论该接口的 HA 监视设置如何。

要设置群集背板，请对每个节点执行以下操作：

1. 确定要用于背板的网络接口。
2. 将所选网络接口的以太网或光缆连接到群集背板交换机。

例如，要使用接口 1/2 作为节点 4 的背板接口，请将电缆从节点 4 的 1/2 接口连接到背板交换机。

> 注意：
>
> 您可以配置链路聚合 (LA) 通道以优化群集背板的吞吐量。

创建 **Citrix ADC** 群集

要创建群集，您必须在添加到群集的第一个设备上创建群集实例并配置群集 IP 地址。此节点称为配置协调器 (CCO)。所有群集配置都在此节点上执行，
方法是通过群集 IP 地址访问它。CCO 未固定到一个特定的群集节点。它可以随着时间的推移而改变。例如，如果 CCO 出现故障，群集会选择其他节点之一作为新的 CCO，然后它拥有群集 IP 地址。

添加群集实例时，在该节点上内部执行 **clear ns config** 扩展命令。此外，从节点清除 SNIP 地址和所有 VLAN 配置（默认 VLAN 和 ADCVLAN 除外）。

> 注意：
>
> 在创建群集之前，请确保已为该节点设置了背板
> 接口。

使用 **Citrix ADC** 命令行创建群集的步骤

> 注意：
>
> 以下命令仅包含强制参数。有关 CLI 命令的详细信息，请参阅每个命令可用的手册页。键入 `man <command syntax>`。例如，要获取 add 群集实例命令的 手册页，请键入`man add cluster instance` 。

1. 登录到要添加到群集的 Citrix ADC 装置（例如，具有 ADCIP 地址 10.102.29.60 的装置）。
2. 添加群集实例。群集实例是标识群集的实体。键入 `add cluster instance <clId>`。其中，**clId** 是标识群集的唯一编号。最小值:1。最大值：16。

> 注意：
>
> 确保群集实例 ID 在 LAN 中是唯一的。

1. 将 Citrix ADC 装置添加到群集中。键入 `add cluster node <nodeId> <IPAddress> [-state <state>] [-backplane <interface_name>]`。其中，
   - **nodeId** 是标识群集上设备的唯一编号。每个节点必须具有不同的节点 ID。最小值：0。最大值:31.

- **IPAddress** 为 Citrix ADC 设备的 IP 地址。仅支持 IPv4 地址。
- **state** 是群集节点的配置状态。可能的值：ACTIVE、PASSIVE、SPARE。默认值：PASSIVE。

> 注意：
>
> 如果要在节点提供流量之前执行节点特定配置（例如添加斑点 IP 地址），请将状态设置为“PASSIVE”（默认状态）。执行特定于节点的配置后，使用 **set cluster node** 命令将节点状态更改为“活动”。

- 背板是节点的背板接口。例如，如果节点 0 使用接口 1/1，则此参数的值为 0/1/1。

例如，

```
add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1
<!--NeedCopy-->
```

2. 在此节点上添加群集 IP 地址（例如 10.102.29.61）。键入 `add ns ip <IPAddress> <netmask> -type clip`。其中，
   - **IPAddress** 是 Citrix ADC 群集的群集 IP 地址。仅支持 IPv4 地址。
   - 子网掩码是群集 IP 地址的子网掩码。该值必须是 255.255.255.255。

   例如，

```
add ns ip 10.102.29.61 255.255.255.255 -type clip
<!--NeedCopy-->
```

3. 启用群集实例以创建群集。键入 `enable cluster instance <clId>`。其中，**clId** 是标识必须启用的群集实例的编号。
4. 保存配置。键入 `save ns config`。
5. 热重新启动设备。键入 `reboot -warm`。

使用 show 群集实例命令验证群集配置。命令的输出必须将 CCO 的 ADCIP 地址显示为群集的节点。

使用配置实用程序创建群集

1. 登录到要添加到群集的 Citrix ADC 装置（例如，具有 ADCIP 地址 10.102.29.60 的装置）。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格中的“开始”下，单击“管理群集”。
4. 在“群集配置”对话框中，设置以下参数：
   - 群集实例 **ID** -标识群集的唯一编号。最小值:1。最大值：16。

- 群集 **IP** 地址 — Citrix ADC 群集的 IP 地址。仅支持 IPv4 地址。
- 背板 - 节点的背板接口。例如，如果节点 0 使用接口 1/1，则此参数的值为 1/1。

5. 单击创建。
6. 在“配置群集实例”对话框中，请确保已选中“启用群集实例”复选框。
7. 在“群集节点”窗格中，选择节点，然后单击“打开”。
8. 在“配置群集节点”对话框中，设置 **ate** 。
9. 单击“确定”，然后单击“保存”。
10. 热重新启动设备。

向群集添加节点

您可以无缝扩展群集的大小，以便最多包含 32 个节点。将设备添加到群集后，系统将根据 CCO 上的可用许可证检查该设备上的许可证。如果许可证匹配，则设备将添加到
群集中。将清除节点的现有配置，群集配置将与节点同步。同步过程中可能会间歇性地下降流量。

要向群集添加节点，您必须首先在群集上配置节点（添加节点），然后在节点上配置群集（加入群集）。

如果使用 Citrix ADC 命令行，请首先登录到群集 IP 地址以添加节点。然后，登录到该节点并将该节点加入群集。如果使用配置实用程序，则只能登录到群集 IP 地址才能添加节点。新添加的节点将自动加入到群集中。或者，您可以从命令行添加节点，然后使用配置实用程序将节点加入群集。

> 注意：
>
> - 添加节点之前，请确保已为该节点设置了背板接口。
> - 当您将新节点添加到只有发现 IP 的群集时，同步会在将发现的 IP 地址分配给该节点之前进行。在这种情况下，L3 VLAN 绑定和静态路由可能会丢失。为避免这种丢失，请在新添加节点的 ADCIP 上添加条带 IP 或添加 L3 VLAN 绑定和静态路由。
> - 将具有预配置链路聚合 (LA) 通道的 Citrix ADC 装置添加到群集中时，LA 通道将继续存在于群集环境中。LA 通道从 LA/x 重命名为 nodeId/LA/x，其中 LA/x 是 LA 通道标识符。

使用 **Citrix ADC** 命令行向群集添加节点的步骤

1. 登录到群集 IP 地址并执行以下操作：
   - 将 Citrix ADC 装置（例如 10.102.29.70）添加到群集中。键入 `add cluster node <nodeId> <IPAddress> [-state <state>] [-backplane <interface_name>]`。其中，
     - **nodeId** 是一个唯一整数，用于标识群集上的设备。每个节点必须具有不同的节点 ID。最小值：0。最大值:31.
     - **IPAddress** 为 Citrix ADC 设备的 IP 地址。仅支持 IPv4 地址。
     - **state** 是群集节点的配置状态。可能的值：ACTIVE、PASSIVE、SPARE。默认值：PASSIVE。

       > 注意：
       > 如果要在节点提供流量之前执行节点特定配置（例如添加斑点 IP 地址），请将状态设置为“PASSIVE”（默认状态）。执行节点特定配置后，使用 **set cluster node** 命令将节点状态更改为 Active。

> – **interface_name** 是节点的背板接口。例如，如果节点 1 使用接口 1/1，则此参数的值为 1/1/1。

例如，

```
add cluster node 1 10.102.29.70 -state PASSIVE -backplane
    1/1/1
<!--NeedCopy-->
```

- 通过输入 `save ns config` 保存配置。

2. 登录到新添加的节点（例如 10.102.29.70）并执行以下操作：
   - 将节点加入群集。键入 `join cluster -clip <ip_addr> -password <password>`。其中，
     - **clip** 是 Citrix ADC 群集的 IP 地址。仅支持 IPv4 地址。
     - **password** 是 CCO 的 nsroot 密码。

     例如，

```
join cluster -clip 10.102.29.61 -password nsroot
<!--NeedCopy-->
```

   - 通过输入 `save ns config` 保存配置。
   - 通过输入 `reboot -warm` 对设备执行暖重新启动。

使用配置实用程序向群集添加节点

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格中的“开始”下，单击“管理群集”。
4. 单击“添加”以添加新节点（例如，10.102.29.70）。
5. 在“创建群集节点”对话框中，设置以下参数：
   - 节点 **ID** -标识群集上装置的唯一整数。每个节点必须具有不同的节点 ID。最小值：0。最大值:31.
   - **IP** 地址 - Citrix ADC 设备的 IP 地址。仅支持 IPv4 地址。
   - 背板 - 节点的背板接口。例如，如果节点 1 使用接口 1/1，则此参数的值为 1/1。
   - 状态 - 群集节点的配置状态。可能的值：ACTIVE、PASSIVE、SPARE。默认值：PASSIVE。

     > 注意：
     > 如果要在节点提供流量之前执行节点特定配置（例如添加斑点 IP 地址），请将状态设置为“PASSIVE”（默认状态）。执行节点特定配置后，将节点状态更改为“ACTIVE”。

6. 单击**“创建”**。将出现一个对话框，通知您设备将进行热重新启动。单击“是”以确认。

使用配置实用程序将先前添加的节点加入群集

如果已使用 Citrix ADC 命令行向群集添加节点，但尚未将节点加入群集，则可以使用以下过程将节点加入群集。

1. 登录到要加入群集的节点（例如 10.102.29.70）。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格的“开始”下，单击“加入群集”。
4. 在“加入到现有群集”对话框中，设置以下参数：
   - 群集 **IP** - Citrix ADC 群集的 IP 地址。仅支持 IPv4 地址。
   - 密码 - CCO 的 `nsroot` 密码。
5. 单击确定。

删除群集节点

从群集中删除节点是一个两步的过程：

1. 从节点中删除对群集实例的引用。此命令在内部执行该节点上的 **clear ns config extended** 命令。此外，从节点清除 SNIP 地址和所有 VLAN 配置（默认 VLAN 和 ADCVLAN 除外）。
2. 从群集中删除节点。

> 注意：
>
> - 当您删除节点为 CCO 时，任何当前群集 IP 地址会话都将失效。另一个群集节点成为 CCO，并且群集 IP 地址分配给该节点。您必须使用群集 IP 地址启动新会话。
> - 要删除群集（和所有节点），您必须分别删除每个节点。删除最后一个节点时，群集 IP 地址将被删除。

使用 **Citrix ADC** 命令行删除群集节点的步骤

1. 登录到要从群集中删除的节点，然后执行以下操作：
   - 删除对群集实例的引用。键入 `rm cluster instance <clId>`。其中，**clId** 是标识要从中移除节点的群集的整数。
   - 通过输入 `save ns config` 保存配置。

   > 注意：
   >
   > 要删除群集的最后一个节点，您只能从该节点中删除群集实例。节点会自动从群集中移除。
2. 登录到群集 IP 地址并执行以下操作：
   - 删除从中删除群集实例的节点。键入 `rm cluster node <nodeId>`。其中，**nodeId** 是标识要移除的节点的整数。
   - 通过输入 `save ns config` 保存配置。

   > 注意：
   >
   > 确保不从本地节点运行 **rm cluster node** 命令，因为这会导致 CCO 和节点之间的配置不一致。

使用配置实用程序删除群集节点

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格的入门下，单击管理群集。
4. 选择要从群集中删除的节点，然后单击“删除”。
5. 单击确定。

查看群集的详细信息

您可以从群集 IP 地址查看群集实例和群集节点的详细信息。

使用 **Citrix ADC** 命令行查看群集实例的详细信息

在群集 IP 地址的 Citrix ADC 命令提示符下，键入 `sh cluster instance <clId>`。其中，**clId** 是标识要查看其详细信息的群集实例的整数。

```
> show cluster instance 1
1)Cluster ID: 1
  Dead Interval: 3 secs
  Hello Interval: 200 msecs
  Preemption: DISABLED
  Propagation: ENABLED
  Cluster Status: ENABLED(admin), ENABLED(operational), UP
  Member Nodes:
  Node ID    Node IP    Health Admin State Operation State
  -------     -------     ------  -----------  ---------------
1)   0    10.102.29.60*   UP      ACTIVE      ACTIVE(CCO)
2)   1    10.102.29.70    UP      ACTIVE      ACTIVE
Done
<!--NeedCopy-->
```

> 注意：从非 CCO 节点的 ADCIP 地址执行此命令将显示此节点上群集的状态。

使用 **Citrix ADC** 命令行查看群集节点的详细信息

在群集 IP 地址的 Citrix ADC 命令提示符处，键入 `sh cluster node <nodeId>`。其中 **nodeId** 是标识要查看其详细信息的节点的整数。

```
>show cluster node 1
Node ID: 1
```



```
IP: 10.102.29.70
Backplane: 1/1/1
Health: UP
Admin state: ACTIVE
Operational State: ACTIVE
Sync State: ENABLED
<!--NeedCopy-->
```

使用配置实用程序查看群集实例的详细信息

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格中的“开始”下，单击“管理群集”。
4. 在“配置 群集实例”对话框中，查看群集的详细信息。

使用配置实用程序查看群集节点的详细信息

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，单击群集，然后单击节点。
3. 在“群集节点”列表中，查看节点详细信息。要获取节点的更详细视图，请单击节点。

跨群集节点分配流量

创建 Citrix ADC 群集并执行所需配置后，必须在客户端数据平面（用于客户端流量）或服务器数据平面（用于服务器流量）上部署等价多路径 (ECMP) 或群集链路聚合组 (CLAG)。这些机制在群集节点之间分配外部流量。

使用等价多路径

使用等价多路径机制，路由器具有到 VIP 地址的等价路由，下一跳作为群集的活动节点。路由器使用基于无状态哈希的机制在路由之间分配流量。

> 注意：
>
> 路由限制为上游路由器支持的最大 ECMP 路由数。

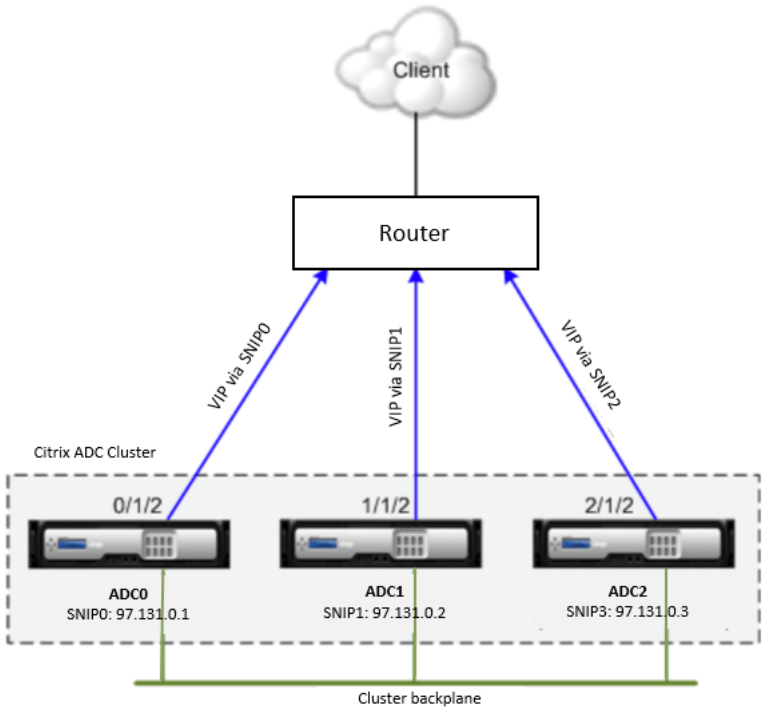
要使用 ECMP，必须首先在群集 IP 地址上启用所需的路由协议（OSPF、RIP 或 BGP）。必须将接口和发现的 IP 地址（启用了动态路由）绑定到 VLAN。使用 vtysh 外壳配置所选路由协议，并在 ZebOS 上重新分配内核路由。

必须在群集 IP 地址和外部连接设备上执行类似的配置。

您必须具备路由协议的详细知识才能使用 ECMP。有关详细信息，请参阅配置动态路由 (/zh-c 不适用 dvanced-concepts/downloads/citrix-Citrix ADC-clustering-guide-v2 copy.pdf) 部分。

> 注意：
>
> 确保群集上的许可证支持动态路由，否则 ECMP 流量分配不起作用。例如，标准的 Citrix ADC 许可证不支持动态路由。

ECMP 拓扑



如上图所示，ECMP 路由器可以通过 SNIP0、SNIP1 或 SNIP2 到达 VIP 地址。

使用 **Citrix ADC** 命令行在 **Citrix ADC** 群集上配置 **ECMP** 的步骤

1. 登录到群集 IP 地址。
2. 启用路由协议（OSPF、RIP 或 BGP）。

   ```
   enable ns feature <routing protocol>
   ```

   例如，

   ```
   1 enable ns feature ospf
   2 <!--NeedCopy-->
   ```

3. 添加 VLAN。

   ```
   add vlan <vlan id>
   ```


例如，

```
add vlan 97
<!--NeedCopy-->
```

4. 将群集节点的接口绑定到 VLAN。

   ```
   bind vlan <vlan id> -ifnum <interface_name>
   ```

   例如，

   ```
   bind vlan 97 -ifnum 0/1/2 1/1/2 2/1/2
   <!--NeedCopy-->
   ```

5. 在每个节点上添加一个被发现的 SNIP 地址，并在其上启用动态路由。

   ```
   add ns ip <SNIP> <netmask> -ownerNode <node id> -dynamicRouting ENABLED
   ```

   例如，

   ```
   add ns ip 97.131.0.1 255.0.0.0 -ownerNode 0 -dynamicRouting
   ENABLED -type SNIP
   add ns ip 97.131.0.2 255.0.0.0 -ownerNode 1 -dynamicRouting
   ENABLED -type SNIP
   add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2 -dynamicRouting
   ENABLED -type SNIP
   <!--NeedCopy-->
   ```

6. 将其中一个发现的 SNIP 地址绑定到 VLAN。将一个发现的 SNIP 地址绑定到 VLAN 时，该子网中的群集上定义的所有其他发现的 SNIP 地址都会自动绑定到 VLAN。

   ```
   bind vlan <vlan id> -ipAddress <SNIP> <netmask>
   ```

   例如，

   ```
   bind vlan 97 -ipAddress 97.131.0.1 255.0.0.0
   <!--NeedCopy-->
   ```

   > 注意：
   >
   > 您可以使用群集节点的 ADCIP 地址，而不是添加 SNIP 地址。如果是这样，则不必执行步骤 3-6。

7. 使用 vtysh 外壳在 ZebOS 上配置路由协议。在节点 ID 0、1 和 2 上配置 OSPF 路由协议。

```
!
interface vlan97
!
router ospf
owner-node 0
ospf router-id 97.131.0.1
exit-owner-node
owner-node 1
ospf router-id 97.131.0.2
exit-owner-node
owner-node 2
ospf router-id 97.131.0.3
exit-owner-node
redistribute kernel
network 97.0.0.0/8 area 0
!
<!--NeedCopy-->
```

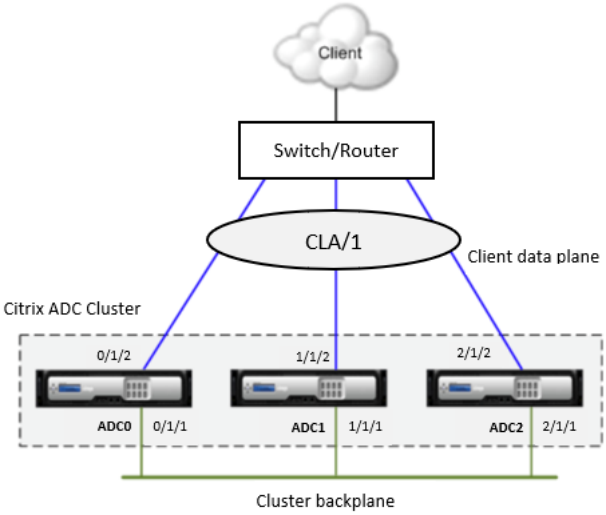
使用群集链路聚合组 (CLAG)

如名称所示，群集链路聚合组是一组群集节点的接口。它是 Citrix ADC 链路聚合的扩展。唯一的区别是，虽然链路聚合要求接口来自同一设备，但在 CLAG 中，接口来自群集的不同节点。

有关链路聚合的更多信息，请参阅配置链路聚合第节。

CLAG 可以是静态的，也可以是动态的。例如，假设一个三节点群集，其中所有三个节点都连接到上游交换机。通过绑定接口 0/1/2、1/1/2 和 2/1/2 形成了一个 CLAG 通道 (CLA/1)。

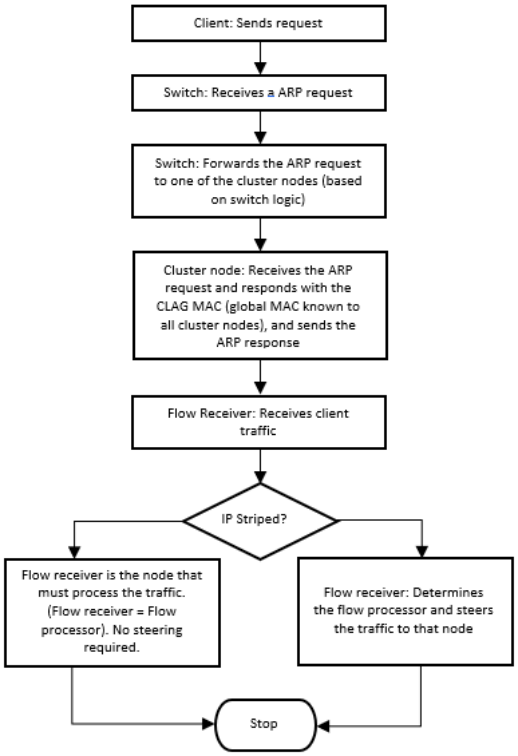
群集链路聚合组拓扑



CLAG 通道具有以下属性：

- 每个通道都有一个由群集节点同意的唯一 MAC。
- 通道可以绑定本地和远程节点的接口。
- 群集中最多支持四个 CLAG 通道。
- 背板接口不能成为 CLAG 通道的一部分。
- 当接口绑定到 CLAG 通道时，通道参数优先于网络接口参数。网络接口只能绑定到一个通道。

使用 **CLAG** 的流量分配流



静态群集链路聚合组

必须在群集 IP 地址和外部连接设备上配置静态 CLAG 通道。如果可能，请将上游交换机配置为基于 IP 地址或端口而不是 MAC 地址分配流量。

有关配置静态 LA 通道的详细信息，请参阅 手动配置链路聚合 部分。

使用 **Citrix ADC** 命令行配置静态 **CLAG** 通道的步骤

1. 登录到群集 IP 地址。

   > 注意：
   >
   > 在外部交换机上配置 CLAG 通道之前，请确保在群集 IP 地址上配置 CLAG 通道。否则，即使未配置 CLAG 通道，交换机也会将流量转发到群集。这可能导致流量丢失。

2. 创建一个 CLAG 通道。添加通道 <clag channel id> -speed <speed>, 其中,

   - <clag channel id> 是标识 CLAG 通道的唯一编号。必须为 CLA/X 的形式，其中 x 的范围为 1 到 4。
   - <speed> 是 CLAG 成员接口的速度。

   例如，

   ```
   add channel CLA/1 -speed 1000
   <!--NeedCopy-->
   ```

   > 注意：
   >
   > 您不得将速度指定为 AUTO。您需要将速度明确指定为 10、100、1000 或 10000。只有与 CLAG 中 <speed> 属性匹配的速度的接口才会添加到活动通讯组列表中。

3. 将所需接口绑定到 CLAG 通道。确保接口未用于群集背板。Bind channel <clag channel id> <interface_name...>，其中，

   - <clag channel id> 标识要将接口绑定到的 CLAG 通道。
   - <interface_name> 指定要绑定到 CLAG 通道的接口。

   例如，

   ```
   bind channel CLA/1 1/1/2 2/1/2 3/1/2
   <!--NeedCopy-->
   ```

4. 验证 CLAG 通道配置。Show channel <clag channel id>

   例如，


```
show channel CLA/1
<!--NeedCopy-->
```

> 注意：
>
> 您可以使用绑定 vlan 命令将 CLAG 通道绑定到 VLAN。CLAG 通道的接口自动绑定到 VLAN。

动态群集链路聚合组

动态 CLAG 使用链路聚合控制协议 (LACP)。有关配置动态 LA 通道的详细信息，请参阅 使用链路聚合控制协议配置链路聚合 部分。

必须在群集 IP 地址和外部连接设备上执行类似的配置。如果可能，请将上游交换机配置为基于 IP 地址或端口而不是 MAC 地址分配流量。

需要记住的几个要点：

- 启用 LACP（通过将 LACP 模式指定为 ACTIVE 或 PASSIVE）。

  > 注意：
  >
  > 确保在 Citrix ADC 群集和外部连接设备上均未将 LACP 模式设置为 PASSIVE。

- 在希望成为通道一部分的每个接口上指定相同的 LACP 键。为了创建 CLAG 通道，LACP 密钥的值可以介于 5 到 8 之间。

  例如，如果在接口 1/1/2 和 2/1/2 设置为 5 上的 LACP 密钥，则会创建 CLA/1。接口 1/1/2 和 2/1/2 自动绑定到 CLA/1。同样，如果将 LACP 键设置为 6，则会创建 CLA/2 通道。

- 将 LAG 类型指定为群集。

使用 **Citrix ADC** 命令行配置动态 **CLAG** 通道的步骤

在群集 IP 地址上，对于要添加到 CLAG 通道的每个接口，键入：`set interface <interface id> -lacpMode <lacpMode> -lacpKey <lacpKey> -lagType Cluster`

为 3 个接口配置 CLAG 通道。

```
set interface 0/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
set interface 1/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
set interface 2/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
<!--NeedCopy-->
```


使用链接集

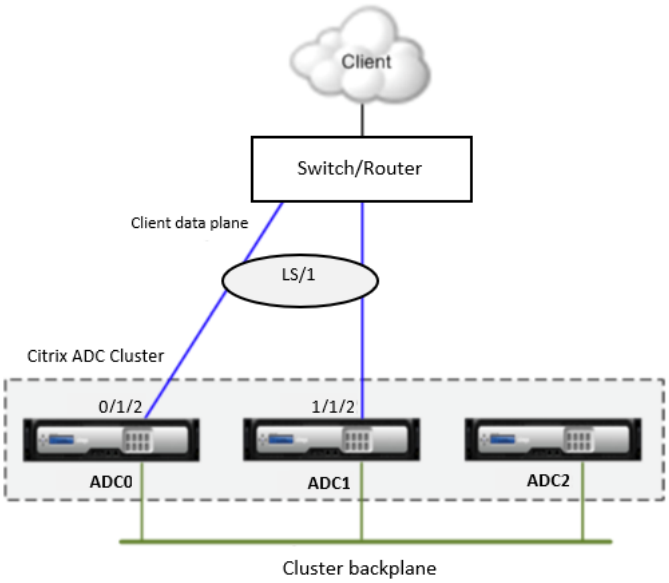
当某些群集节点没有物理连接到外部网络时，必须使用链接集。在此类群集拓扑中，未连接的群集节点使用链接集中指定的接口通过群集背板与外部网络进行通信。链接集通常用于连接设备的端口不足以连接群集节点的情况。

必须仅在群集 IP 地址上配置链接集。

例如，假设一个三节点群集，其中上游交换机只有两个可用端口。使用链接集，您可以将两个节点连接到交换机，并保持第三个节点未连接。在下图中，链路集 (LS/1) 是通过绑定接口 0/1/2 和 1/1/2 来形成的。ADC2 是群集的未连接节点。

> 注意：
>
> 使用链接集可提高需要基于 MAC 的转发 (MBF) 的拓扑的性能。

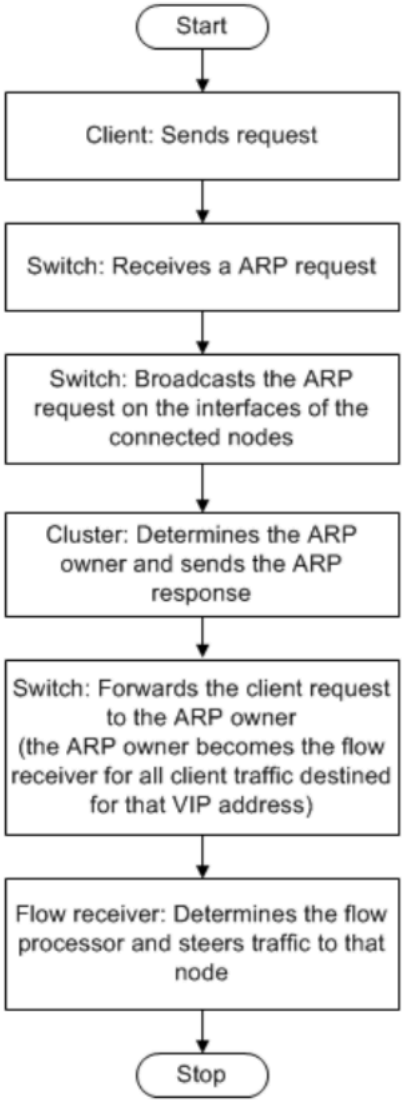
链接集拓扑



链路集通知 ADC2 它可以使用接口 0/1/2 和 1/1/2 与网络设备进行通信。所有进出 ADC2 的流量现在都通过接口 0/1/2 或 1/1/2 进行路由。

使用链接集的流量分配流

Figure 1-10. Traffic distribution flow using linksets



使用 **Citrix ADC** 命令行配置链接集的步骤

1. 登录到群集 IP 地址。
2. 创建链接集。要添加链接集，请输入 `linkset id`。`Linkset id` 是链接集的唯一标识符。它必须为 LS/X 的形式。例如，

```
1 add linkset LS/1
2 <!--NeedCopy-->
```

3. 将所需接口绑定到链接集。确保这些接口未用于群集背板。要绑定链接集，请输入 <`linkset id`> `-ifnum`

<interface_name...>。Interface_name 指定要绑定到链接集的接口。例如，

```
bind linkset LS/1 -ifnum 0/1/2 1/1/2
<!--NeedCopy-->
```

4. 验证链接集配置。要查看链接集，请输入 linkset id。Linkset id 是要验证的链接集的标识符。例如，

```
show linkset LS/1
<!--NeedCopy-->
```

> 注意：
>
> 您可以使用绑定 vlan 命令将链接集绑定到 VLAN。链路集的接口会自动绑定到 VLAN。

使用配置实用程序配置链接集

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开网络，然后单击链接集。
3. 在详细信息窗格中，单击 **Add**（添加）。
4. 在“创建链接集”对话框中：
   a) 通过设置链接集参数指定链接集的名称。
   b) 指定要添加到链接集的接口，然后单击添加。对要添加到链接集的每个接口重复此步骤。
5. 单击 **Create**（创建），然后单击 **Close**（关闭）。

管理 **Citrix ADC** 群集

创建群集并配置所需的流量分配机制后，群集可以为流量提供服务。在群集的生命周期中，您可以执行群集管理任务，例如禁用群集的节点、发现 Citrix ADC 装置、查看统计信息、同步群集配置、群集文件和跨节点的时间，以及升级或降级群集节点的软件。

禁用群集节点

您可以通过禁用该节点上的群集实例来临时从群集中删除节点。禁用的节点不与群集配置同步，无法为流量提供服务。

使用 **Citrix ADC** 命令行禁用群集节点的步骤

在要禁用的节点的 Citrix ADC 命令提示符处。要禁用群集实例，请输入 clId。**clId** 标识要禁用的群集实例。

> 注意：
>
> 要禁用群集，请在群集 IP 地址上运行禁用群集实例命令。

使用配置实用程序禁用群集节点

1. 登录到要禁用的节点。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格中的“开始”下，单击“管理群集”。
4. 在配置群集实例对话框中，取消选中启用群集实例复选框。
5. 单击确定。

> 注意：
>
> 要禁用群集，请在群集 IP 地址上运行禁用群集实例命令。要在所有节点上禁用群集实例，请登录到群集并执行上述步骤。

发现 **Citrix ADC** 设备

您可以发现 Citrix ADC 装置与 CCO 的 ADCIP 地址位于同一子网中。然后，可将发现的装置添加到群集中。

> 注意：
>
> 此操作仅可通过配置实用程序执行。

使用 **Citrix ADC** 配置实用程序发现装置

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，单击群集，然后单击节点。
3. 在详细信息窗格中，单击页面底部的“发现 **Citrix ADC**”。
4. 在“发 现 **Citrix ADC**”对话框中，设置以下参数：
   - **IP** 地址范围 -指定要在其中发现 Citrix ADC 装置的 IP 地址范围。例如，通过将此选项指定为 10.102.29.4 - 15，您可以搜索介于 10.102.29.4 到 10.102.29.15 之间的所有 ADCIP 地址。
   - 背板接口 -指定要用作背板接口的接口。这是可选参数。如果未指定此参数，则必须在将节点添加到群集后对其进行更新。
5. 单击确定。
6. 选择要添加到群集的 Citrix ADC 装置。
7. 单击确定。

查看群集的统计信息

您可以查看群集实例和群集节点的统计信息，以评估性能或对群集的操作进行故障排除。

使用 **Citrix ADC** 命令行查看群集实例的统计信息

在群集 IP 地址的 Citrix ADC 命令提示符下，键入：

```
stat cluster instance <clId>
```

```
1 >stat cluster instance
2 Cluster Instance Summary
3 Cluster Size                                    3
4 Cluster Status                              ENABLED
5 Cluster Config Coordinator (CCO)        10.102.29.80
6 Current DFD Sessions                            0
7 Total Steered Packets                           0
8 Done
9 <!--NeedCopy-->
```

要使用错误统计信息显示群集实例的统计信息，请在群集 IP 地址的 Citrix ADC 命令提示符处键入：

```
stat cluster instance -detail <clId>
```

```
1  > stat cluster instance -detail
2  Cluster Statistics
3  Summary
4  Cluster Size                                   3
5  Cluster Status                             ENABLED
6  Cluster Config Coordinator (CCO)       10.102.29.80
7  Current DFD Sessions                           0
8  Total Steered Packets                          0
9  Error Statistics
10 DFD Dropped Packets                            0
11 Propagation timeout                            0
12 Done
13 <!--NeedCopy-->
```

使用 **Citrix ADC** 命令行查看群集节点的统计信息

在群集 IP 地址的 Citrix ADC 命令提示符下，键入：`stat cluster node`。

```
1 > stat cluster node
2 Cluster Node Summary
3 NodeID NodeIP       State    Health  Sync State HB Tx HB Rx
```



```
0       10.102.29.70 ACTIVE   UP       ENABLED    4489  2247
1       10.102.29.80 ACTIVE   UP       ENABLED    2659  4805
2       10.102.29.60 INACTIVE UNKNOWN UNKNOWN    7145  0
Done
<!--NeedCopy-->
```

要显示单个群集节点的统计信息，请在群集 IP 地址的 Citrix ADC 命令提示符处键入：`stat cluster node <nodeid>`。

```
> stat cluster node 1
Node ID : 1
Node IP                                  10.102.29.80
Master State                                ACTIVE
Health                                          UP
Sync State                                  ENABLED
Heartbeats Sent                               3025
Heartbeats received                           5537
NNM Statistics
NNM current connections                          7
NNM total transmitted messages                  15
NNM total received messages                     18
Error Statistics
NNM Multicast/Broadcast req err                  0
Done
<!--NeedCopy-->
```

使用配置实用程序查看群集实例的统计信息

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格中的页面中，单击“统计”。

使用配置实用程序查看群集节点的统计信息

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，单击群集，然后单击节点。
3. 在详细信息窗格中，选择一个节点，然后单击“统计”以查看该节点的统计信息。要查看所有节点的统计信息，请单击“统计信息”，而不选择特定节点。

同步群集配置

在以下情况下，CCO 上可用的 Citrix ADC 配置将同步到群集的其他节点：

- 节点加入群集。
- 节点重新加入群集。
- 在 CCO 上执行一个新命令。

此外，您可以强制同步 CCO（完全同步）上可用的配置到特定群集节点。确保一次同步一个群集节点，否则群集可能会受到影响。

使用 **Citrix ADC** 命令行同步群集配置的步骤

在要同步 CCO 配置的装置的 Citrix ADC 命令提示符下，键入：**force cluster sync**。

使用配置实用程序同步群集配置

1. 登录到要同步 CCO 配置的装置。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格中的“实用工具”下，单击“强制群集同步”。
4. 单击确定。

同步群集文件

CCO 上可用的文件称为群集文件。将节点添加到群集时，这些文件会在其他群集节点上自动同步，并在群集的生命周期内定期进行同步。此外，您可以手动同步群集文件。

同步的 CCO 中的目录和文件包括：

```
- /nsconfig/ssl/
- /var/netscaler/ssl/
- /var/vpn/bookmark/
- /nsconfig/dns/
- /nsconfig/htmlinjection/
- /netscaler/htmlinjection/ens/
- /nsconfig/monitors/
- /nsconfig/nstemplates/
- /nsconfig/ssh/
- /nsconfig/rc.netscaler
- /nsconfig/resolv.conf
- /nsconfig/inetd.conf
- /nsconfig/syslog.conf
- /nsconfig/snmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/httpd.conf
- /nsconfig/sshd_config
- /nsconfig/hosts
```



```
- /nsconfig/enckey
- /var/nslw.bin/etc/krb5.conf
- /var/nslw.bin/etc/krb5.keytab
- /var/lib/likewise/db/
- /var/download/
- /var/wi/tomcat/webapps/
- /var/wi/tomcat/conf/Catalina/localhost/
- /var/wi/java_home/lib/security/cacerts
- /var/wi/java_home/jre/lib/security/cacerts
- /nsconfig/license/
- /nsconfig/rc.conf-
<!--NeedCopy-->
```

使用 **Citrix ADC** 命令行同步群集文件的步骤

在群集 IP 地址的 Citrix ADC 命令提示符处，键入 `sync cluster files <mode>`。

- 模式指定要同步的目录或文件。可能的值是：all、bookmarks、ssl、htmlinjection、imports、misc、dns、all_plus_misc。默认值：全部。

使用配置实用程序同步群集文件

1. 登录到群集。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格中的“实用工具”下，单击“同步群集文件”。
4. 在“同步群集文件”对话框中，在“模式”下拉框中选择要同步的文件。
5. 单击确定。

同步群集节点上的时间

Citrix ADC 群集使用精密时间协议 (PTP) 来跨群集节点同步时间。PTP 使用多播数据包来同步时间。如果在时间同步中存在某些问题，则必须禁用 PTP 并在群集上配置网络时间协议 (NTP)。

使用 **Citrix ADC** 命令行启用/禁用 **PTP**

在群集 IP 地址的 Citrix ADC 命令提示符下，键入：`set ptp -state disable`。

使用配置实用程序启用/禁用 **PTP**

1. 登录到群集 IP 地址。
2. 在导航窗格中，展开系统，然后单击群集。
3. 在详细信息窗格的 实用程序下，单击 配置 **PTP** 设置。

4. 在“启用/禁用 **PTP**”对话框中，选择是启用还是禁用 PTP。
5. 单击确定。

升级或降级群集软件

所有群集节点必须运行相同的软件版本。要升级或降级群集的软件，必须升级或降级每个节点上的软件，一次一个节点。

当节点上的软件升级或降级时，不会从群集中删除该节点。节点仍然是群集的一部分，并且不间断地为客户端流量提供服务，但节点在升级或降级后重新启动时出现停机时间除外。但是，由于群集节点之间的软件版本不匹配，配置传播将被禁用，并且只有在所有群集节点都是相同版本后才会启用。

由于在降级群集时升级过程中禁用了配置传播，因此在此期间您无法通过群集 IP 地址执行任何配置。但是，您可以通过单个节点的 ADCIP 地址执行节点级配置，但必须确保在所有节点上执行相同的配置，以保持它们同步。

> 注意：
>
> 升级或降级群集软件版本时，无法添加群集节点。

升级或降级群集节点的软件

1. 确保群集是稳定的，并且所有节点上的配置都同步。
2. 升级或降级群集的软件。
   - 升级或降级群集节点的软件。有关升级和降级装置软件的详细信息，请参阅升级或降级系统软件。
   - 重新启动设备。
   - 对每个其他群集节点重复上述两个步骤。

   > 注意：
   >
   > Citrix 建议您在升级下一个节点之前等待上一个节点处于活动状态。

用例

本主题提供了部署 Citrix ADC 群集的一些用例。

- 创建双节点群集
- 在群集中使用缓存重定向
- 将 CLAG 与链接集一起使用
- 用于客户端和服务器的通用接口以及用于背板的专用接口
- 用于客户端、服务器和背板的通用交换机
- 用于客户端和服务器的通用交换机和用于背板的专用交换机
- 每个节点的多个交换机
- 每个节点都有不同的交换机
- 示例群集配置

创建双节点群集

双节点群集是以下规则的例外：只有当最少 (n/2 + 1) 节点（其中 **n** 为群集节点数）能够服务流量时，群集才起作用。如果将该公式应用于双节点群集，则如果一个节点出现故障（n/2 +1=2），群集将失败。

即使只有一个节点能够提供流量，双节点群集也可以正常工作。创建两个节点群集与创建任何其他群集相同。必须将一个节点添加为配置协调器，另一个节点添加为另一个群集节点。

> 注意：
>
> 双节点群集不支持增量配置同步。仅支持完全同步。

将 **HA** 设置迁移到群集设置

通过从 HA 设置中删除装置，然后创建 Citrix ADC 群集，可以将现有的高可用性 (HA) 设置迁移到群集设置中。例如，考虑使用 ADCIP 地址为 10.102.97.131 和 10.102.97.132 的高可用性设置。

使用 **Citrix ADC** 命令行将 **HA** 设置转换为群集设置

1. 登录到每个 HA 节点，并将其从 HA 设置中删除。键入 `rm HA node <nodeId>`。例如 `rm HA node 1`。
2. 转到其中一个 HA 节点上的 shell 并将 `ns.conf` 复制到另一个 `.conf` 文件。例如 `ns_backup.conf`。
3. 按如下方式编辑新的配置文件：
   - 删除群集不支持的所有功能。有关不受支持的功能的列表，请参阅 群集支持的 Citrix ADC 功能。
   - 删除具有接口的配置，或将接口名称从 **c/u** 约定更新为 **n/c/u** 约定。
4. 在两个节点上，确定要用于群集背板的网络接口。
5. 将其中一个节点（例如 10.102.97.131）设置为 CCO 节点。有关详细说明，请参阅 设置 Citrix ADC 群集。
6. 登录到群集 IP 地址并应用备份配置文件中的配置。键入 `batch -f <fileName>`。例如 `batch -f ns_backup.conf`。
7. 保存配置。键入 `save ns config`。
8. 将另一个节点添加到群集中。有关详细说明，请参阅 向群集添加节点。

HA 设置的装置将迁移到群集设置中。

在群集中使用缓存重定向

Citrix ADC 群集中的缓存重定向工作方式与独立 Citrix ADC 设备上的工作方式相同。唯一的区别是在群集 IP 地址上完成了配置。有关详细信息，请参阅缓存重定向部分。

在透明模式下使用缓存重定向时要记住的要点：

- 在配置缓存重定向之前，请确保已将所有节点连接到外部交换机，并且已配置了链接集。否则，客户端请求将被删除。

- 在负载平衡虚拟服务器上启用 MAC 模式时，请确保在群集上启用 MBF 模式（使用 `enable ns mode MBF` 命令）。否则，请求将直接发送到源服务器，而不是发送到缓存服务器。

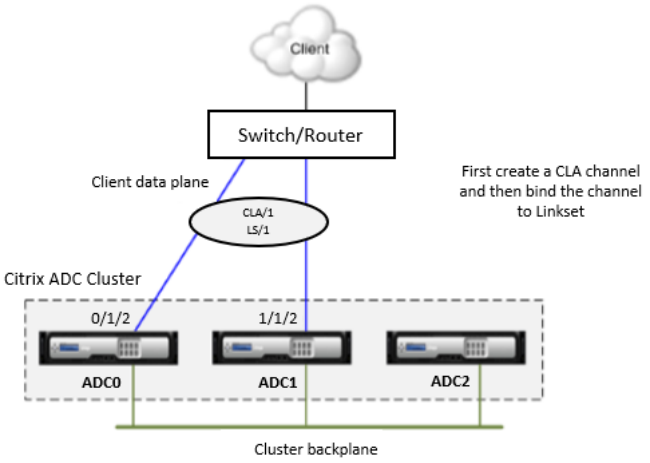
将 **CLAG** 与链接集一起使用

在非对称群集拓扑中，某些群集节点未连接到上游网络。在这种情况下，您必须使用链接集。要优化性能，可以将连接到交换机的接口绑定为 CLA 通道，然后将 CLA 通道绑定到链路集。

要了解如何使用 CLAG 和链接集的组合，请考虑一个三节点群集，其上游交换机只有两个可用端口。您可以将其中两个群集节点连接到交换机，并保持其他节点未连接。

> 注意：
>
> 同样，您也可以在非对称拓扑中使用 ECMP 和链接集的组合。

链接集和群集链路聚合组拓扑

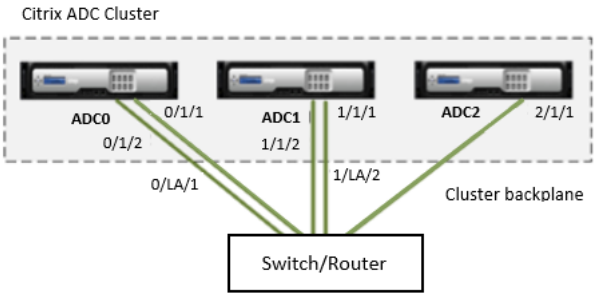


使用 **Citrix ADC** 命令行使用 **CLAG** 和链接集的步骤

1. 登录到群集 IP 地址。
2. 将连接的接口绑定到 CLA 通道 `add channel CLA/1 -ifnum 0/1/2 1/1/2`
3. 将 CLA 通道绑定到链接集 `add linkset LS/1 -ifnum CLA/1`

LA 通道上的背板

在此部署中，LA 通道用于群集背板。



ADC0 - nodeId：0，ADCIP：10.102.29.60

ADC1 - nodeId：1，ADCIP：10.102.29.70

ADC2 - nodeId：2，ADCIP：10.102.29.80

将背板接口作为 **LA** 通道部署群集

1. 创建一个由节点 ADC0、ADC1 和 ADC2 组成的群集。

   - 登录到要添加到群集的第一个节点，然后执行以下操作：

```
create cluster instance 1
add cluster node 0 10.102.29.60 -state ACTIVE
enable cluster instance 1
add ns ip 10.102.29.61 255.255.255.255 -type CLIP
save ns config
reboot -warm
<!--NeedCopy-->
```

   - 登录到群集 IP 地址并执行以下操作：

```
add cluster node 1 10.102.29.70 -state ACTIVE
add cluster node 2 10.102.29.80 -state ACTIVE
<!--NeedCopy-->
```

   - 登录到节点 10.102.29.70 和 10.102.29.80，以便将节点加入到群集。

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
<!--NeedCopy-->
```


如上所述，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

2. 登录到群集 IP 地址并执行以下操作：

   - 为节点 ADC0 和 ADC1 创建 LA 通道。

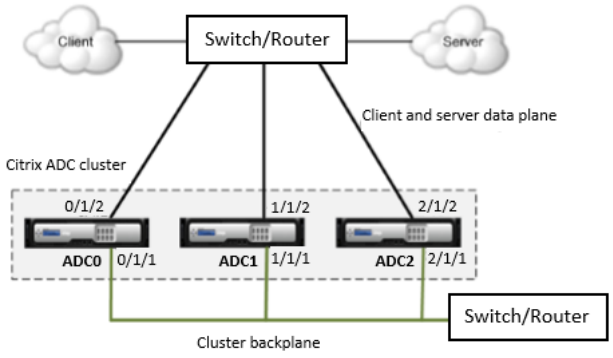
```
add channel 0/LA/1 -ifnum 0/1/1 0/1/2
add channel 1/LA/2 -ifnum 1/1/1 1/1/2
<!--NeedCopy-->
```

   - 配置群集节点的背板。

```
set cluster node 0 -backplane 0/LA/1
set cluster node 1 -backplane 1/LA/2
set cluster node 2 -backplane 2/1/1
<!--NeedCopy-->
```

用于客户端和服务器的通用接口以及用于背板的专用接口

这是 Citrix ADC 群集的单臂部署。在此部署中，客户端和服务器网络使用相同的接口与群集进行通信。群集背板使用专用接口进行节点间通信。



ADC0 - nodeId：0，ADCIP：10.102.29.60

ADC1 - nodeId：1，ADCIP：10.102.29.70

ADC2 - nodeId：2，ADCIP：10.102.29.80

部署具有用于客户端和服务器的公用接口以及用于群集背板的不同接口的群集

1. 创建一个由节点 ADC0、ADC1 和 ADC2 组成的群集。

   - 登录到要添加到群集的第一个节点，然后执行以下操作：


```
create cluster instance 1
add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
enable cluster instance 1
add ns ip 10.102.29.61 255.255.255.255 -type CLIP
save ns config
reboot -warm
<!--NeedCopy-->
```

- 登录到群集 IP 地址并执行以下操作：

```
add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
<!--NeedCopy-->
```

- 登录到节点 10.102.29.70 和 10.102.29.80，以便将节点加入到群集。

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
<!--NeedCopy-->
```

如上所述，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

2. 在群集 IP 地址上，为背板接口以及客户端和服务器接口创建 VLAN。

```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1
//For the interfaces that are connected to the client and server
   networks
add vlan 20
bind vlan 20 0/1/2 1/1/2 2/1/2
<!--NeedCopy-->
```

3. 在交换机上，为与背板接口以及客户端和服务器接口对应的接口创建 VLAN。为思科 C3750 版本 12.2 (40) SE 交换机提供了以下配置示例。必须在其他交换机上执行类似的配置。

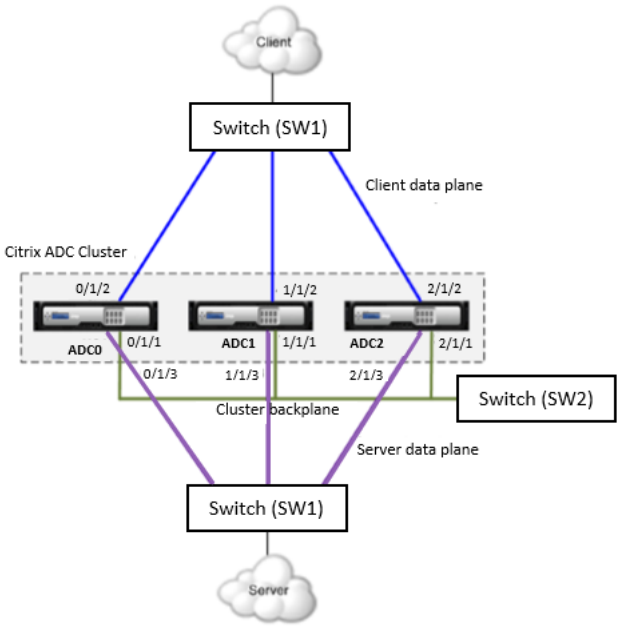
```
//For the backplane interfaces. Repeat for each interface...
```



```
interface GigabitEthernet1/0/1
switchport access vlan 100
switchport mode access
end
//For the interfaces connected to the client and server networks.
   Repeat for each interface...
interface GigabitEthernet1/0/3
switchport access vlan 200
switchport mode access
end
<!--NeedCopy-->
```

用于客户端、服务器和背板的通用交换机

在此部署中，客户端、服务器和背板使用同一交换机上的专用接口与 Citrix ADC 群集进行通信。



ADC0 - nodeId：0，ADCIP：10.102.29.60

ADC1 - nodeId：1，ADCIP：10.102.29.70

ADC2 - nodeId：2，ADCIP：10.102.29.80

使用客户端、服务器和背板的公用交换机部署群集

1. 创建一个由节点 ADC0、ADC1 和 ADC2 组成的群集。
    - 登录到要添加到群集的第一个节点，然后执行以下操作：


```
create cluster instance 1
add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
enable cluster instance 1 add ns ip 10.102.29.61 255.255.255.255 -
    type CLIP
save ns config
reboot -warm
<!--NeedCopy-->
```

- 登录到群集 IP 地址并执行以下操作：

```
add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
<!--NeedCopy-->
```

- 登录到节点 10.102.29.70 和 10.102.29.80，以便将节点加入到群集。

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
<!--NeedCopy-->
```

如上所述，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

2. 在群集 IP 地址上，为背板、客户端和服务器接口创建 VLAN。

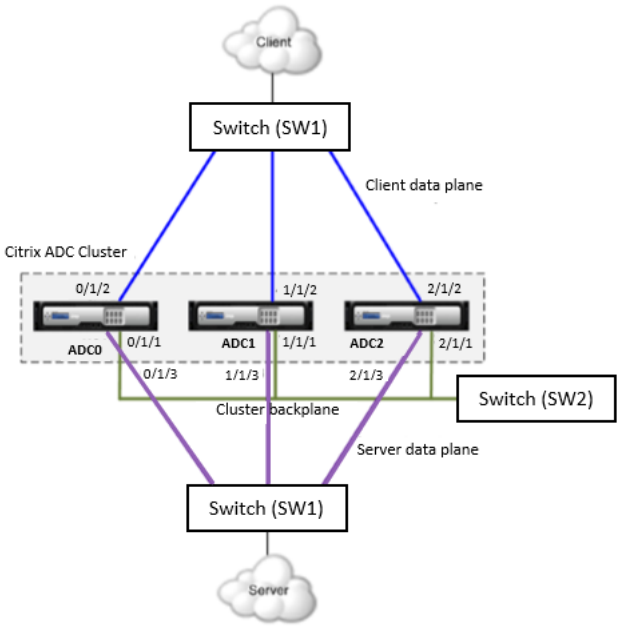
```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1
//For the client-side interfaces
add vlan 20 bind vlan 20 0/1/2 1/1/2 2/1/2
//For the server-side interfaces
add vlan 30
bind vlan 30 0/1/3 1/1/3 2/1/3
<!--NeedCopy-->
```

3. 在交换机上，为与背板接口以及客户端和服务器接口对应的接口创建 VLAN。为思科 C3750 版本 12.2 (40) SE 交换机提供了以下配置示例。必须在其他交换机上执行类似的配置。


```
//For the backplane interfaces. Repeat for each interface...
interface GigabitEthernet1/0/1
switchport access vlan 100
switchport mode access
end
//For the client interfaces. Repeat for each interface...

interface GigabitEthernet1/0/3
switchport access vlan 200
switchport mode access
end
//For the server interfaces. Repeat for each interface...

interface GigabitEthernet1/0/6
switchport access vlan 300
switchport mode access
end
<!--NeedCopy-->
```

客户端和服务器通用交换机以及背板专用交换机

在此部署中，客户端和服务器使用同一交换机上的不同接口与 Citrix ADC 群集进行通信。群集背板使用专用交换机进行节点间通信。



ADC0 - nodeId：0，ADCIP：10.102.29.60

ADC1 - nodeId：1，ADCIP：10.102.29.70

ADC2 - nodeId：2，ADCIP：10.102.29.80

为客户端和服务器部署具有相同交换机和群集背板的不同交换机的群集

1. 创建一个由节点 ADC0、ADC1 和 ADC2 组成的群集。

   - 登录到要添加到群集的第一个节点，然后执行以下操作：

```
create cluster instance 1
add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
enable cluster instance 1
add ns ip 10.102.29.61 255.255.255.255 -type CLIP
save ns config
reboot -warm
<!--NeedCopy-->
```

   - 登录到群集 IP 地址并执行以下操作：

```
add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
<!--NeedCopy-->
```

   - 登录到节点 10.102.29.70 和 10.102.29.80，以便将节点加入到群集。

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
<!--NeedCopy-->
```

   如上所述，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

2. 在群集 IP 地址上，为背板、客户端和服务器接口创建 VLAN。

```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1
//For the client-side interfaces
add vlan 20
bind vlan 20 0/1/2 1/1/2 2/1/2
//For the server-side interfaces
```



```
add vlan 30
bind vlan 30 0/1/3 1/1/3 2/1/3
<!--NeedCopy-->
```

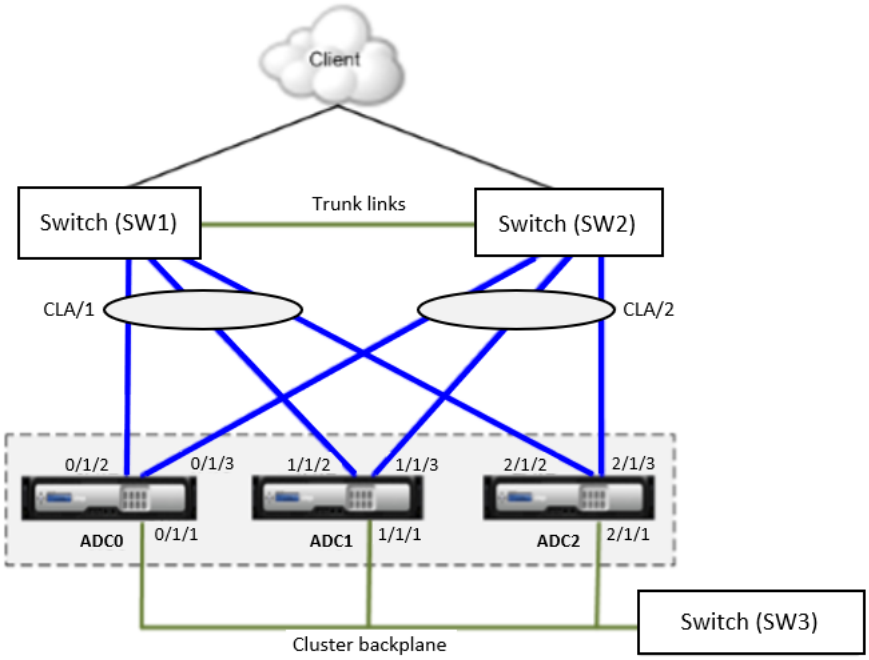
1. 在交换机上，为与背板接口以及客户端和服务器接口对应的接口创建 VLAN。为思科 C3750 版本 12.2 (40) SE 交换机提供了以下配置示例。必须在其他交换机上执行类似的配置。

```
//For the backplane interfaces. Repeat for each interface...
interface GigabitEthernet1/0/1
switchport access vlan 100
switchport mode access
end

//For the client interfaces. Repeat for each interface...
interface GigabitEthernet1/0/3
switchport access vlan 200
switchport mode access
end

//For the server interfaces. Repeat for each interface...
interface GigabitEthernet1/0/6
switchport access vlan 300
switchport mode access
end
<!--NeedCopy-->
```

每个节点使用多个交换机

在此部署中，我们引入了两个客户端交换机，以确保客户端交换机的冗余性。交换机通过中继链路相互连接。一台交换机发生故障不会影响群集的整体工作。

> 注意：
>
> 同样的部署策略也可用于服务器端连接。



ADC0 - nodeId：0，ADCIP：10.102.29.60

ADC1 - nodeId：1，ADCIP：10.102.29.70

ADC2 - nodeId：2，ADCIP：10.102.29.80

> 注意：
>
> 使用中继链路时，流量可能会循环流动。为避免这种情况，必须确保网络拓扑配置为避免环路。

部署一个群集，每个节点连接到两个交换机，并且交换机通过中继链接连接

1. 创建一个由节点 ADC0、ADC1 和 ADC2 组成的群集。
   - 登录到要添加到群集的第一个节点，然后执行以下操作：

```
create cluster instance 1
add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
enable cluster instance 1
add ns ip 10.102.29.61 255.255.255.255 -type CLIP
save ns config
reboot -warm
<!--NeedCopy-->
```

   - 登录到群集 IP 地址并执行以下操作：


```
add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
<!--NeedCopy-->
```

- 登录到节点 10.102.29.70 和 10.102.29.80，以便将节点加入到群集。

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
<!--NeedCopy-->
```

如上所述，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

2. 登录到群集 IP 地址并执行以下操作：

   - 为背板接口创建 VLAN。

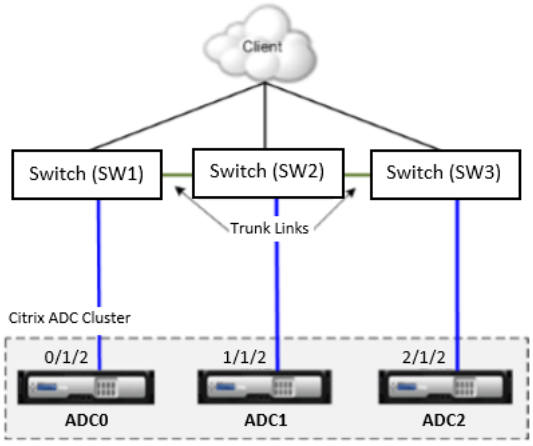
```
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1
<!--NeedCopy-->
```

   - 为使用 SW1 和 SW2 的客户端接口创建一个 CLAG。

```
add channel CLA/1 -ifnum 0/1/2 1/1/2 2/1/2 -speed 1000
add channel CLA/2 -ifnum 0/1/3 1/1/3 2/1/3 -speed 1000
<!--NeedCopy-->
```

每个节点都有不同的交换机

在此部署中，每个群集节点都连接到不同的交换机，并在交换机之间配置中继链路。



群集配置将与其他部署方案相同。大多数客户端配置将在客户端交换机上完成。

示例群集配置

以下示例可用于配置具有 ECMP、CLAG 或链接集的四节点群集。

1. 创建群集。

   - 登录到第一个节点。
   - 添加群集实例。

```
add cluster instance 1
<!--NeedCopy-->
```

   - 将第一个节点添加到群集。

```
add cluster node 0 10.102.33.184 -backplane 0/1/1
<!--NeedCopy-->
```

   - 启用群集实例。

```
enable cluster instance 1
<!--NeedCopy-->
```

   - 添加群集 IP 地址。

```
add ns ip 10.102.33.185 255.255.255.255 -type CLIP
```



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

- 保存配置。

```
save ns config
<!--NeedCopy-->
```

- 热重新启动设备。

```
reboot -warm
<!--NeedCopy-->
```

2. 将其他三个节点添加到群集中。
   - 登录到群集。
   - 将第二个节点添加到群集。

```
add cluster node 1 10.102.33.187 -backplane 1/1/1
<!--NeedCopy-->
```

   - 将第三个节点添加到群集。

```
add cluster node 2 10.102.33.188 -backplane 2/1/1
<!--NeedCopy-->
```

   - 将第四个节点添加到群集。

```
add cluster node 3 10.102.33.189 -backplane 3/1/1
<!--NeedCopy-->
```

3. 将添加的节点加入群集。此步骤不适用于第一个节点。
   - 登录到每个新添加的节点。
   - 将节点加入群集。

```
join cluster -clip 10.102.33.185 -password nsroot
<!--NeedCopy-->
```


- 保存配置。

```
save ns config
<!--NeedCopy-->
```

- 热重新启动设备。

```
reboot -warm
<!--NeedCopy-->
```

4. 通过群集 IP 地址配置 Citrix ADC 群集。

```
// Enable load balancing feature enable ns feature lb
// Add a load balancing virtual server add lb vserver
   first_lbvserver http
....
....
<!--NeedCopy-->
```

5. 为群集配置以下任一通信分发机制（ECMP、链接集、CLAG）。

- **ECMP**。
  - 登录到群集。
  - 启用 OSPF 路由协议。

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

  - 添加 VLAN。

```
add vlan 97
<!--NeedCopy-->
```

  - 将群集节点的接口绑定到 VLAN。

```
bind vlan 97 -ifnum 0/1/4 1/1/4 2/1/4 3/1/4
<!--NeedCopy-->
```

  - 在每个节点上添加一个斑点 SNIP，并在其上启用动态路由。


```
add ns ip 1.1.1.10 255.255.255.0 -ownerNode 0 dynamicRouting
    ENABLED
add ns ip 1.1.1.11 255.255.255.0 -ownerNode 1 dynamicRouting
    ENABLED
add ns ip 1.1.1.12 255.255.255.0 -ownerNode 2 dynamicRouting
    ENABLED
add ns ip 1.1.1.13 255.255.255.0 -ownerNode 3 dynamicRouting
    ENABLED
<!--NeedCopy-->
```

  - 将其中一个 SNIP 地址绑定到 VLAN。

```
bind vlan 97 -ipAddress 1.1.1.10 255.255.255.0
<!--NeedCopy-->
```

  - 使用 vtysh 外壳在 ZebOS 上配置路由协议。

- 链接集。假定具有 `nodeId` 3 的节点未连接到交换机。必须配置链接集，以便未连接的节点可以使用其他节点接口与交换机通信。
  - 登录到群集。
  - 添加链接集。

```
add linkset LS/1
<!--NeedCopy-->
```

  - 将连接的接口绑定到链接集。

```
bind linkset LS/1 -ifnum 0/1/6 1/1/6 2/1/6
<!--NeedCopy-->
```

- 静态 **CLAG**。
  - 登录到群集。
  - 添加 CLA 通道。

```
add channel CLA/1 -speed 1000
<!--NeedCopy-->
```

  - 将接口绑定到 CLA 通道。

```
bind channel CLA/1 0/1/5 1/1/5 2/1/5 3/1/5
```



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

- 在交换机上执行等效配置。

- 动态 **CLAG**。
  - 登录到群集。
  - 将接口添加到 CLA 通道。

```
1   set interface 0/1/5 -lacpmode active -lacpkey 5 -lagtype
       cluster
2   set interface 1/1/5 -lacpmode active -lacpkey 5 -lagtype
       cluster
3   set interface 2/1/5 -lacpmode active -lacpkey 5 -lagtype
       cluster
4   set interface 3/1/5 -lacpmode active -lacpkey 5 -lagtype
       cluster
5   <!--NeedCopy-->
```

  - 在交换机上执行等效配置。

6. 将群集节点的状态更新为 “ACTIVE”

```
1  set cluster node 0 -state ACTIVE
2  set cluster node 1 -state ACTIVE
3  set cluster node 2 -state ACTIVE
4  set cluster node 3 -state ACTIVE
5  <!--NeedCopy-->
```

Citrix ADC 群集故障排除

如果 Citrix ADC 群集中出现故障，故障排除的第一步是通过分别运行 `show cluster instance <clId>` 和 `show cluster node <nodeId>` 命令获取有关群集实例和群集节点的信息。

如果您无法通过使用上述两种方法找到问题，则可以使用以下方法之一：

- 隔离故障源。尝试绕过群集以访问服务器。如果尝试成功，则问题可能与群集设置有关。
- 检查最近执行的命令。运行 历史命令以检查群集上最近执行的配置。您还可以查看 `ns.conf` 文件以验证已实施的配置。
- 检查 **ns.log** 文件。使用每个节点的 `/var/log/` 目录中可用的日志文件来标识已执行的命令、命令的状态和状态更改。
- 检查 **newnslog** 文件。使用每个节点的 `/var/nslog/` 目录中可用的 newnslog 文件来标识群集节点上发生的事件。您可以将多个 newnslog 文件作为单个文件查看，方法是将这些文件复制到单个目录中，然后运行以下命令：


```
nsconmsg -K newnslog-node<id> -K newnslog.node<id> -d current
<!--NeedCopy-->
```

如果仍然无法解决此问题，您可以尝试跟踪群集上的数据包，或使用 **show tech support scope cluster** 命令将报告发送给技术支持团队。

跟踪 **Citrix ADC** 群集的数据包

Citrix ADC 操作系统提供了一个名为 nstrace 的实用程序，用于获取装置接收和发出的数据包的转储。实用程序将数据包存储在跟踪文件中。您可以使用这些文件来调试数据包流向群集节点的问题。必须使用 Wireshark 应用程序查看跟踪文件。对于在本机 (.cap) 模式下收集的跟踪，请务必使用 Wireshark 的内部版本，该版本可以理解本机数据包。

nstrace 实用程序的一些突出方面是：

- 可以配置为使用经典表达式和默认表达式有选择地跟踪数据包。
- 可以采用多种格式捕获跟踪：nstrace 格式 (.cap) 和 TCP 转储格式 (.pcap)。
- 可以聚合 CCO 上所有群集节点的跟踪文件。
- 可以将多个跟踪文件合并到一个跟踪文件中。

您可以从 Citrix ADC 命令行或 Citrix ADC 外壳中使用 nstrace 实用程序。

跟踪独立装置的数据包

在设备上运行启动 nstrace 命令。该命令在 /var/nstrace/<date-timestamp> 目录中创建跟踪文件。跟踪文件名的格式为 nstrace<id>.cap。

您可以通过执行 **show nstrace** 命令来查看状态。您可以通过执行 **stop nstrace** 命令停止跟踪数据包。

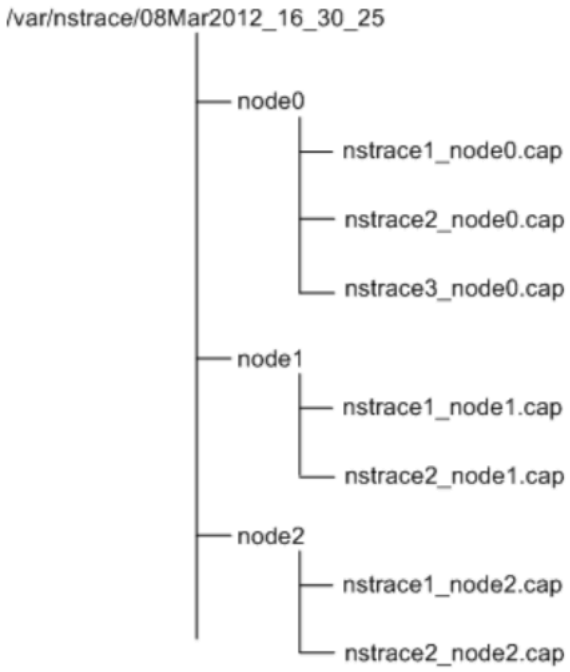
> 注意：
>
> 您还可以通过执行 nstrace.sh 文件从 Citrix ADC 外壳中运行 nstrace 实用程序。Citrix 建议通过 Citrix ADC 命令行使用 nstrace 实用程序。

跟踪群集的数据包

您可以跟踪所有群集节点上的数据包，并获取 CCO 节点上的所有跟踪文件。

在群集 IP 地址上运行 **start nstrace** 命令。在所有群集节点上传播并执行该命令。跟踪文件存储在 /var/nstrace/<date-timestamp> 目录中的单个群集节点中。跟踪文件名的格式为 nstrace<id>_node<id>.cap。

您可以使用每个节点的跟踪文件来调试节点操作。但是，如果要将所有群集节点的跟踪文件放在一个位置，则必须在群集 IP 地址上运行 **stop nstrace** 命令。所有节点的跟踪文件都下载到 /var/nstrace/<date-timestamp> 目录中的群集配置协调器 (CCO) 节点上，如下所示：



合并多个跟踪文件

您可以从从群集节点获取的跟踪文件中准备单个文件。单个跟踪文件为您提供群集数据包跟踪的累积视图。单个跟踪文件中的跟踪条目根据群集上收到数据包的时间进行排序。

要合并跟踪文件，请在 Citrix ADC shell 中键入：`nstracemerge.sh -srcdir <DIR> -dstdir <DIR> -filename <name> -filesize <num>`。

- **srcdir** 是合并跟踪文件的目录。此目录中的所有跟踪文件都合并为一个文件。
- **dstdir** 是创建合并跟踪文件的目录。
- **filename** 是创建的跟踪文件的名称。
- **filesize** 是跟踪文件的大小。

以下是使用 nstrace 实用程序过滤数据包的一些示例。

- 要跟踪三个节点的背板接口上的数据包，请执行以下操作：
  - 使用经典表达式：

```
1 start nstrace -filter /"INTF == 0/1/1 && INTF == 1/1/1 && INTF==
     2/1/1"
2 <!--NeedCopy-->
```

  - 使用默认表达式：


```
start nstrace -filter "CONNECTION.INTF.EQ("0/1/1") &&CONNECTION.
    INTF.EQ("1/1/1") && CONNECTION.INTF.EQ("2/1/1")"
<!--NeedCopy-->
```

- 从源 IP 地址 10.102.34.201 或源端口大于 80 且服务名称不是 "s1" 的系统跟踪数据包：

  – 使用经典表达式

```
start nstrace -filter "SOURCEIP == 10.102.34.201 || (SVCNAME !=
    s1 && SOURCEPORT > 80)"
<!--NeedCopy-->
```

  – 使用默认表达式

```
start nstrace -filter "CONNECTION.SRCIP.EQ(10.102.34.201) ||(
    CONNECTION.SVCNAME.NE("s1") && CONNECTION.SRCPORT.GT(80))"
<!--NeedCopy-->
```

故障排除常见问题

在将节点加入群集时，我收到以下消息：" 错误：无效的接口名称/数字。" 我该怎么做才能解决此错误?

- 如果您在使用 添加 **cluster** 节点命令添加节点时提供了无效或不正确的背板接口，则会出现此错误。要解决此错误，请验证您在添加节点时提供的接口。请确保您尚未将设备的管理界面指定为背板接口，并且该接口的 **nodeId** 位与节点的 ID 相同。例如，如果 **nodeId** 为 **3** ，则背板接口必须为3/<c>/<u> 。

在将节点加入群集时，我收到以下消息：" 错误：无法启用群集，因为本地节点不是群集的成员。" 我该怎么做才能解决此错误?

- 当您尝试加入节点而不将节点的 ADCIP 添加到群集时，会出现此错误。若要解决此错误，必须首先使用添加群集节点命令将 节点的 **ADCIP** 地址添加到群集，然后执行 连接群集命令。

在将节点加入群集时，我收到以下消息：" 错误：连接被拒绝。" 我该怎么做才能解决此错误?

由于以下原因，可能会发生此错误：

- 连接问题。节点无法连接到群集 IP 地址。尝试从您尝试加入的节点 ping 群集 IP 地址。
- 重复的群集 **IP** 地址。检查某个非群集节点上是否存在群集 IP 地址。如果是，请创建一个新的群集 IP 地址，然后尝试重新加入群集。

在将节点加入群集时，我收到以下消息：”错误：**CCO** 和本地节点之间的许可证不匹配。“ 我该怎么做才能解决此错误**?**

- 要加入到群集的设备必须具有与 CCO 相同的许可证。如果要加入的节点上的许可证与 CCO 上的许可证不匹配，则会出现此错误。要解决此错误，请在两个节点上运行以下命令并比较输出。

  从命令行运行：

  - `show ns hardware`
  - `show ns license`

  从 shell 中运行：

  - `nsconmsg -g feature -d stats`
  - `ls /nsconfig/license`
  - 查看 `/var/log/license.log` 文件的内容

当群集节点的配置未与群集配置同步时，我必须做什么**?**

- 通常，这些配置会在所有群集节点之间自动同步。但是，如果您认为配置未在特定节点上同步，则必须通过从要同步的节点执行强制群集 sync 命令来强制同步。有关详细信息，请参阅同步群集配置。

不同步群集节点上的配置。我如何确保配置始终保持同步**?**

- 要确保在节点之间同步群集配置，请在每次配置后运行 **save ns config** 命令。否则，当群集节点重新启动时，这些配置可能不可用。

配置群集节点时，我收到以下消息：“错误：会话为只读；连接到群集 **IP** 地址以修改配置。“

- 群集上的所有配置都必须通过群集 IP 地址完成，并且配置将传播到其他群集节点。通过各个节点的 Citrix ADC IP (ADCIP) 地址建立的所有会话都是只读的。

为什么节点运行状况显示 **“UP”** 时，节点状态显示 **“INACTIVE”?**

- 由于多种原因，运行正常的节点可能处于 “INACTIVE” 状态。扫描 `ns.log` 或错误计数器可帮助您确定确切的原因。

当节点的运行状况显示 **“Not UP”** 时，如何解决节点的运行状况**?**

- 节点运行状况 **Not UP** 表示节点存在一些问题。要了解根本原因，您必须运行 **sh cluster node** 命令。此命令显示节点属性和节点失败的原因。

当我运行 **set** 虚拟服务器命令时，我收到以下消息，“没有这样的资源。“ 我该如何解决此问题**?**

- 群集中不支持 **set vserver** 命令。也不支持 **unset vserver**、**enable vserver**、**disable vserver** 和 **rm vserver** 命令。但是，支持 显示虚拟服务器命令。

我无法通过 **Telnet** 会话配置群集。我该怎么办**?**

- 在 telnet 会话中，只能在只读模式下访问群集 IP 地址。因此，您不能通过 telnet 会话配置群集。

我注意到群集节点之间存在显著的时间差异。我该如何解决此问题**?**

当 PTP 数据包由于背板交换而丢弃，或者如果物理资源在虚拟环境中过度投入时，时间将不会同步。

要同步时间，必须对群集 IP 地址执行以下操作：

1. 禁用 PTP。

```
set ptp -state disable
<!--NeedCopy-->
```

2. 为群集配置网络时间协议 (NTP)。有关详细信息，请参阅使用 CLI 或配置实用程序设置时钟同步部分。

常见问题解答

群集中有多少个 **Citrix ADC** 设备**?**

- Citrix ADC 群集可以包含至少 2 个或多达 32 个 Citrix ADC nCore 硬件或虚拟设备。

我有多个独立节点，每个节点都有不同的配置。我可以将它们添加到单个群集吗**?**

- 是。您最多可以向群集添加 32 个节点。但是，将节点添加到群集时，装置的现有配置将被清除。要使用单个装置的配置，必须手动准备所有配置的单个 `*.conf` 文件，编辑配置以删除群集不支持的功能，更新接口的命名约定，然后应用配置添加到 CCO，通过使用批处理命令。

是否可以将独立 **Citrix ADC** 设备或 **HA** 设置的配置迁移到群集设置**?**

- 否。将节点添加到群集设置中时，将在该设备上执行 **clear ns config** 命令（带有扩展选项）。此外，还会清除 SNIP 地址和所有 VLAN 配置（默认 VLAN 和 ADCVLAN 除外）。因此，Citrix 建议在将设备添加到群集之前先备份这些配置。

是否可以自动检测 **Citrix ADC** 设备，以便将其添加到群集**?**

- 是。配置实用程序允许您发现与 CCO 的 ADCIP 地址位于同一子网中的 Citrix ADC 装置。有关详细信息，请参阅发现 Citrix 设备。

群集是否为许可功能**?**

- 是的，群集是许可的功能。您必须在要添加到群集的所有设备的/nsconfig/license/ 目录中拥有群集许可证文件的副本。此外，要添加到群集的所有装置也必须具有相同的许可证文件可用。

Citrix ADC 设备是否可以成为多个群集的一部分**?**

- 否。设备只能属于一个群集。

未连接到客户端或服务器网络的节点是否仍然提供流量**?**

- 是。Citrix ADC 群集支持称为链路集的流量分配机制，该机制允许未连接的节点通过连接节点的接口为流量提供服务。未连接的节点通过群集背板与连接的节点进行通信。

如果节点上的群集许可证已过期，会发生什么情况**?**

- 如果节点上的群集许可证在节点运行时过期，则群集不受影响。但是，当您重新启动该节点时，群集在此节点上被禁用，因此该节点将无法为流量提供服务。要纠正问题并使节点处于活动状态，您必须上传新的许可证并热重新启动设备。

为什么群集的网络接口使用 **3** 元组（**n/u/c**）表示法而不是常规 **2** 元组（**u/c**）表示法**?**

- 当设备是群集的一部分时，您必须能够识别网络接口所属的节点。因此，群集节点的网络接口命名约定从 u/c 修改为 n/u/c，其中 n 表示节点 Id。

什么是条纹 **IP** 地址**?**

- 默认情况下，在群集上定义的任何 IP 地址（VIP 或 SNIP）都是条带 IP 地址。条带 IP 地址在群集的所有节点上都处于活动状态。

什么是发现的 **IP** 地址**?** 我可以在运行时更改已发现 **IP** 地址的所有权吗**?**

- 发现的 IP 地址是指活动的 IP 地址，并且专门由群集的一个节点拥有。必须通过群集 IP 地址定义斑点 IP 地址，方法是在 **add ns ip** 命令中指定所有者节点。

您不能在运行时更改发现的 IP 地址的所有权。要更改所有权，必须首先删除 IP 地址，然后通过指定新所有者来重新添加该 IP 地址。

什么是 **CCO?**

- CCO 是 *Configuration Coordinator* 的缩写形式。此节点拥有群集 IP 地址并协调所有群集配置。

什么是群集 **IP** 地址**?** 什么是其子网掩码**?**

- 群集 IP 地址是 Citrix ADC 群集的管理地址。所有群集配置都必须通过此地址访问群集来执行。群集 IP 地址的子网掩码固定在 255.255.255.255。

当我将第一个节点添加到群集时，它是配置协调器（**CCO**）。现在，另一个节点显示为 **CCO**。为什么**?**

- 创建群集后，第一个节点将成为 CCO。群集 IP 地址由该节点拥有。但是，CCO 不是一个固定的节点。由于各种原因，它可以随着时间的推移而改变。在这种情况下，群集会选择新的 CCO，并将群集 IP 地址分配给新的 CCO。

是否可以从群集节点的 **ADCIP** 地址执行命令**?**

- 否。通过 Citrix ADC IP (ADCIP) 地址访问各个群集节点是只读的。这意味着当您登录到群集节点的 ADCIP 地址时，您只能查看配置和统计信息。您不能执行任何配置。但是，您可以从群集节点的 ADCIP 地址执行一些操作。有关详细信息，请参阅单个节点上支持的操作。

是否可以禁用群集节点之间的配置传播**?**

- 否，您不能明确禁用群集节点之间的群集配置传播。但是，配置传播可以在软件升级或降级过程中自动禁用帐户版本不匹配.

如何删除群集和群集的所有节点**?**

- 要删除群集和群集的所有节点，您必须按照中所述分别删除每个节点删除群集节点。

Citrix ADC 设备是群集的一部分时，是否可以更改 **ADCIP** 地址或更改 **ADCVLAN?**

- 否。要进行此类更改，您必须首先从群集中移除设备，执行更改，然后将设备添加到群集中。

Citrix ADC 群集是否支持 **L2** 和 **L3** 虚拟局域网 **(VLAN)?**

- 是的，Citrix ADC 群集支持群集节点之间的 VLAN。必须在群集 IP 地址上配置 VLAN。
  - **L2 VLAN**。您可以通过绑定属于群集不同节点的接口来创建第 2 层 VLAN。
  - **L3 VLAN**。您可以通过绑定属于群集不同节点的 IP 地址来创建 layer3 VLAN。IP 地址必须属于同一子网。确保满足以下条件之一。否则，L3 VLAN 绑定可能会失败：
    * 所有节点的 IP 地址与绑定到 VLAN 的节点位于同一子网上。
    * 群集具有条带 IP 地址，并且该 IP 地址的子网绑定到 VLAN。

当您将新节点添加到只有发现 IP 的群集时，同步会发生在已发现的 IP 地址分配给该节点之前。在这种情况下，L3 VLAN 绑定可能会丢失。为避免这种丢失，请在新添加节点的 ADCIP 上添加条带 IP 或添加 L3 VLAN 绑定。

为什么 **Citrix ADC** 设备添加到群集时会删除 **VLAN** 和 **VLAN** 绑定**?**

- 将 Citrix ADC 装置添加到群集设置中时，将在该装置上执行清除 ns 配置命令（带有扩展选项）。此外，将清除 SNIP 地址和所有 VLAN 配置（默认 VLAN 和 ADCVLAN 除外）。

如何在 **Citrix ADC** 群集上配置 **SNMP?**

- SNMP 监视群集以及群集的所有节点，其方式与监视独立 Citrix ADC 设备的方式相同。唯一的区别是必须通过群集 IP 地址配置群集上的 SNMP。在生成特定于硬件的陷阱时，还包括两个额外的变量绑定来标识群集的节点：节点 ID 和节点的 ADCIP。

有关配置 SNMP 的详细信息，请参阅 SNMP 部分。

联系技术支持以了解群集相关问题时，我必须提供哪些详细信息**?**

- Citrix ADC 提供了一个 **show techsupport -scope cluster** 命令，用于提取所有群集节点的配置数据、统计信息和日志。您需要在群集 IP 地址上运行此命令。

此命令的输出保存在名为 `collector_cluster_<nsip_CCO>_P_<date-timestamp>.tar.gz` 的文件中，该文件位于 CCO 的 `/var/tmp/support/cluster/` 目录中。将此归档文件发送给技术支持团队以调试问题。

单个节点上支持的操作

所有群集配置都通过群集 IP 地址在 CCO 上执行，并将这些配置传播到群集节点。但是，通过各个群集节点的 Citrix ADC IP (ADCIP) 地址访问这些节点，可以对其执行某些操作。

```
1  -  enable cluster instance w disable cluster instance
2  -  set cluster instance
3  -  rm cluster instance
4  -  set cluster node
5  -  rm cluster node
6  -  force cluster sync
7  -  sync cluster files
8  -  send arp all
9  -  start nstrace
10 -  stop nstrace
11 -  show nstrace
12 -  set interface
13 -  enable interface w disable interface w save ns config
14 -  reboot
15 <!--NeedCopy-->
```


> 注意：
>
> 允许使用所有 show 和 stat 命令，因为它们不涉及任何配置更改。

Citrix ADC 池容量验证的参考设计

May 20, 2020

Citrix ADC 池容量是一种许可框架，由带宽池和 Citrix Application Delivery Management (ADM) 托管并提供服务的虚拟实例池组成。在此公共池中，数据中心中的每个 Citrix ADC 都会根据需要检出一个虚拟实例许可证，并且仅检出任意多大的带宽。无论平台或外形如何，它都可以执行此操作（MPX-Z 除外，它只检出带宽许可证）。许可证文件和带宽未绑定到 Citrix ADC。当 Citrix ADC 不再需要这些资源时，它会将这些资源签回公共池，从而使需要这些资源的其他 ADC 可用。

此许可框架通过确保 ADC 不分配多余的未使用带宽，最大限度地提高了带宽利用率。Citrix ADC 能够检查许可证和带宽进出公共池的能力，使用户和管理员能够自动实例 Provisioning。用户和管理员可以在运行时增加或减少分配给实例的带宽，而不会影响流量。此外，池中的 Citrix ADC 许可证也可以从一个实例传输到另一个实例，并且这些许可证可以由所有外形规格（MPX、SDX、VPX 和 CPX）共享。

组件

池容量将软件与底层硬件分离。这种方法允许一种可从现有平台转移到新平台的许可模式。池容量由四个组成部分组成：

1. 零容量硬件，没有带宽，没有实例，也没有功能。
2. 带有软件版本（标准、高级和高级）的带宽池，可在所有 Citrix ADC 形式因素（包括 MPX、SDX、VPX 和 CPX）之间共享。
3. 实例池，它是跨软件/虚拟 Citrix ADC 外形规格（包括在 SDX 上运行的 VPX、独立 VPX 和 CPX 上共享的实例池。
4. Citrix ADM，用于管理带宽和实例许可证。Citrix ADM 的此功能免费向客户收取任何费用。

池容量的组件将在本文后面详细讨论。

永久许可证

永久许可证是不会过期的许可证。使用永久许可证，用户支付一次性费用，并有权永久使用许可证。以下是一些需要考虑的事项：

- 永久许可证通常存在限制，例如定期维护支持成本。
- 许可证绑定到特定的硬件平台，通常无法移动。
- 随着技术的变化，许可证可能会过时。
- 特定功能通过特定版本的永久许可证授权（如带宽）启用。对于 Citrix ADC，这些特定版本的许可证包括标准版、高级版和高级版。

相比之下，池容量许可证并不绑定到特定的硬件平台，并且可以从现有平台转移到新平台。

集合容量优势

用例 **1**：移动到云

集合容量通过在现有基础架构上提供投资保护，促进混合云的采用。客户可以选择将部分容量从内部部署移动到云，从而降低基于云的 Citrix ADC 设备的成本。

用例 **2**：硬件刷新周期

以前在传统部署中部署了 Citrix ADC 的客户现在必须在更新环境时重新购买所有内容。使用池化的 Citrix ADC 许可证，刷新周期只需要在保留软件的同时刷新硬件。当硬件刷新时，软件许可证可以轻松地从旧设备转移到新的硬件/软件设备。这大大降低了刷新周期的成本，并使客户能够比传统的 5 年间隔更早地查看刷新周期。

用例 **3**：部署 **DevOps (VPX/CPX)**

投资于零容量设备的客户可以购买 CPX 设备，并将部分容量转移到微服务环境。他们还可以购买额外的容量来支持新体系结构。总体而言，这是从基于本版或基于硬件的体系结构到微服务或基于软件的体系结构的过渡更具成本效益的。

池容量的工作原理

说明

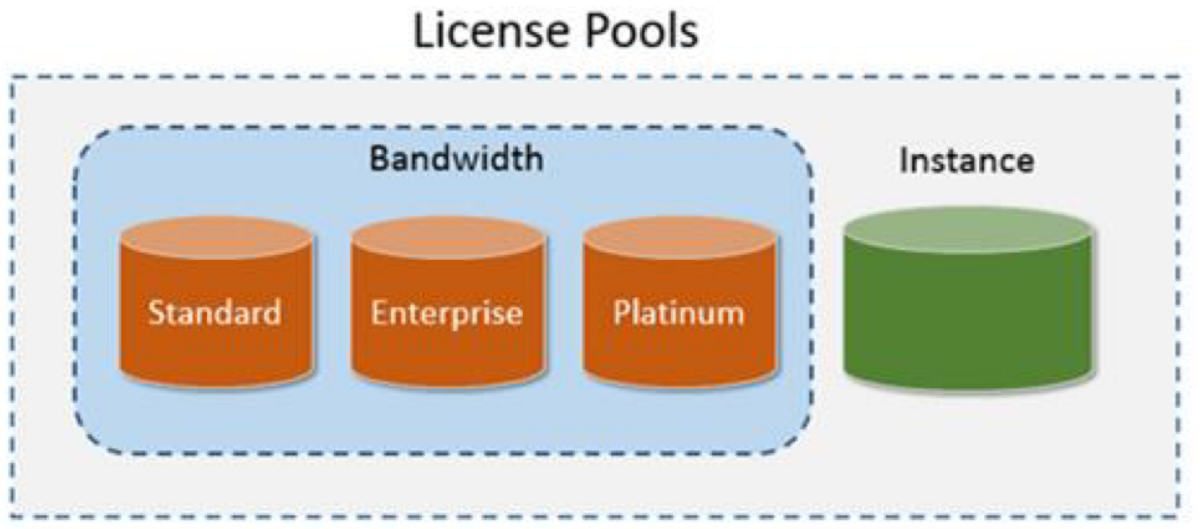
池容量是将软件与底层硬件分离的许可框架。这种方法允许一种可从现有平台转移到新平台的许可模式，并通过确保分配 ADC 的带宽不会超出其要求，最大限度地提高带宽利用率。Citrix ADC 能够检查公用池进出许可证和带宽，也使用户能够自动执行实例预配。

用户可以在运行时增加或减少分配给 Citrix ADC 的带宽，而不会影响流量。用户还可以将池中的 Citrix ADC 许可证从一个 Citrix ADC 传输到另一个。

许可证流

客户购买 Citrix ADC 池容量许可证并从 我的帐户登录页面 中下载许可证。

然后，将这些许可证导入到 Citrix ADM 中。



零容量硬件

当通过 Citrix ADC 池容量进行管理时，SDX 实例称为“零容量硬件”，因为这些实例在从带宽和实例池中检出资源之前无法正常工作。因此，这些平台被称为 SDX-Z 装置。

同样，MPX 装置在通过 Citrix ADC 池容量进行管理时也称为“零容量硬件”，因为它们在从带宽池中检出资源之前无法正常工作。因此，这些平台称为 MPX-Z 装置。

零容量硬件设备需要平台许可证才能从公共池中检出带宽和/或实例许可证。用户必须首先使用硬件序列号或许可证访问代码手动安装平台许可证。

目前，运行 Citrix ADC 软件 11.1 或更高版本的以下零容量平台支持 Citrix ADC 池容量进行新购买和升级：

- MPX-14000Z
- MPX-14000Z-40G
- MPX-15000Z
- MPX-15000Z-50G
- MPX-25000Z-40G
- MPX-26000Z
- MPX-26000Z-100G
- SDX-14000Z
- SDX-14000Z-40G
- SDX-15000Z-50G
- SDX-25000Z-40G
- SDX-26000Z-100G

目前，运行 Citrix ADC 软件 12.0 或更高版本 (MPX) 和 11.1 或更高版本 (SDX) 的以下零容量平台支持 Citrix ADC 池容量进行新购买和升级：

- MPX-14000Z-40S
- MPX-14000Z-40C
- MPX-14000 FIPS
- MPX-25000ZA
- MPX-26000Z-50S
- SDX-14000Z-40S
- SDX-14000Z-40C
- SDX-14000 FIPS
- SDX-25000ZA

目前，运行 Citrix ADC 软件 12.0 或更高版本的以下零容量平台支持 Citrix ADC 池容量进行新购买和升级：

- MPX-8900Z
- SDX-8900Z

目前，运行 Citrix ADC 软件 12.0 或更高版本 (MPX) 和 11.1 或更高版本 (SDX) 的以下零容量平台仅支持 Citrix ADC 池容量进行升级：

- MPX-115xx (11515 - 11542)
- MPX-89xx/80xx
- MPX-22xxx
- MPX-24xxx
- SDX-115xx (11515 - 11542)
- SDX-89xx/80xx
- SDX-22xxx
- SDX-24xxx

目前，运行 Citrix ADC 软件 11.1 或更高版本的以下零容量平台仅支持 Citrix ADC 池容量进行新购买：

- VPX
- CPX

独立 **Citrix ADC VPX** 实例

在以下虚拟机管理程序上运行 Citrix ADC 软件 11.1 或更高版本的 Citrix ADC VPX 实例支持池容量：

- VMware ESX 6.0
- Citrix XenServer
- Linux KVM

在以下虚拟机管理程序和云平台上运行 Citrix ADC 软件 12.0 或更高版本的 Citrix ADC VPX 实例支持池容量：

- Microsoft Hyper-V
- Amazon AWS
- Microsoft Azure

> 注意：
>
> 要启用 Citrix ADM 与 Microsoft Azure 或 AWS 之间的通信，必须配置 IPSEC 隧道。有关详细信息，请参阅将部署在云中的 NetScaler VPX 实例添加到 NetScaler MAS。

独立 **Citrix ADC CPX** 实例

部署在 Docker 主机上的 Citrix ADC CPX 实例支持池容量。与零容量硬件不同，CPX 不需要平台许可证。为了处理流量，它必须从池中签出实例许可证。

带宽池

带宽池是 Citrix ADC 可以共享的总带宽（包括物理和虚拟）。带宽池由每个软件版本（标准版、高级版和高级版）的单独池组成。给定的 Citrix ADC 不能同时检出来自不同池的带宽。Citrix ADC 可从中检出带宽的带宽池取决于其获得许可的软件版本。从池中签出时，许可证将解锁资源，如 CPU/PE、SSL 内核、每秒数据包和带宽。

实例池

实例池定义了可通过 Citrix ADC 池容量管理的 VPX 实例或 CPX 实例的数量，或 SDX-Z 中的 VPX 实例数量。

> 注意：
>
> SDX-Z 的管理服务不使用实例。

Citrix ADM

Citrix ADC 池容量使用 Citrix ADM 管理池容量许可证：带宽池许可证和实例池许可证。用户可以使用 Citrix ADM 管理池容量许可证而无需 ADM 许可证。

从带宽和/或实例池中签出许可证时，零容量硬件平台上的 Citrix ADC 外形规格和硬件型号决定：

- Citrix ADC 在正常工作之前必须检出的最小带宽和实例数。
- Citrix ADC 可以检出的最大带宽和实例数。

- 每个带宽签出的最低带宽单位。最小带宽单位是 Citrix ADC 必须从池中检出的最小带宽单位。任何签出都必须是最低带宽单位的整数倍数。例如，如果 Citrix ADC 的最小带宽单位为 1 Gbps，则可以检出 100 Gbps，但无法检出 200 Mbps 或 150.5 Gbps。最低带宽单位与最低带宽要求不同。Citrix ADC 只有在获得至少具有最小带宽许可后才能运行。一旦达到最小带宽，实例就可以使用最小带宽单位检出额外带宽。

下表汇总了所有支持的 Citrix ADC 平台的最大带宽/实例、最小带宽/实例和最小带宽单位：

最低系统要求：**MPX** 和 **SDX**

产品系列	最大带宽 (Gbps)	最小带宽 (Gbps)	最小实例	最大实例数	最小带宽单位
MPX 8005Z	30	5	不适用	不适用	1 Gbps
MPX 8900Z	33	5	不适用	不适用	1 Gbps
MPX 14000Z 系列	100	20	不适用	不适用	1 Gbps
MPX 14000Z 40G 系列	100	20	不适用	不适用	1 Gbps
MPX 14000Z FIPS 系列	100	20	不适用	不适用	1 Gbps
MPX 14000Z 40S 系列	100	20	不适用	不适用	1 Gbps
MPX 15000Z 系列	100	20	不适用	不适用	1 Gbps
MPX 15000Z 50G 系列	100	20	不适用	不适用	1 Gbps
MPX 115XX 系列	42	15	不适用	不适用	1 Gbps
MPX 22XXX 系列	120	40	不适用	不适用	1 Gbps
MPX 24000Z 系列	150	100	不适用	不适用	1 Gbps
MPX 25000Z 40G	200	100	不适用	不适用	1 Gbps
MPX 25000ZA	200	100	不适用	不适用	1 Gbps
MPX 26000Z 系列	200	100	不适用	不适用	1 Gbps

产品系列	最大带宽 (Gbps)	最小带宽 (Gbps)	最小实例	最大实例数	最小带宽单位
MPX 26000Z 100G 系列	200	100	不适用	不适用	1 Gbps
MPX 26000Z 50S 系列	200	100	不适用	不适用	1 Gbps
SDX 8015Z	15	2	1	2	1 Gbps
SDX 89XX 系列	33	10	2	7	1 Gbps
SDX 115XX 系列	42	7	2	20	1 Gbps
SDX 14000Z 系列	100	10	2	25	1 Gbps
SDX 14000Z 40G 系列	100	10	2	25	1 Gbps
SDX 14000Z 40S 系列	100	10	2	25	1 Gbps
SDX 14000Z FIPS 系列	100	10	2	25	1 Gbps
SDX 15000Z 50G	100	10	2	55	1 Gbps
SDX 15000Z	100	10	2	55	1 Gbps
SDX 22XXX 系列	120	20	20	80	1 Gbps
SDX 25000Z 40G	200	50	20	115	1 Gbps
SDX 25000ZA	200	50	20	115	1 Gbps
SDX 26000Z 100G	200	50	10	115	1 Gbps
SDX 26000Z	200	50	10	115	1 Gbps
SDX 26000Z 50S	200	50	10	115	1 Gbps
SDX 24000Z 系列	150	50	20	80	1 Gbps

适用的 **Citrix ADC CPX** 型号

带宽/实例带宽单位	CPX
最大带宽 (Gbps)	1
最低带宽 (Gbps)	不适用
最小实例数	1
最大实例数	不适用
最低带宽单位	不适用

对于 **Hypervisor** 和云服务上的 **Citrix ADC VPX**

带宽/实例带宽单位	Citrix XenServer	VMware ESXi	Linux KVM	Microsoft Hyper-v	AWS	AZURE
最大带宽 (Gbps)	40 Gbps	100 Gbps	100 Gbps	3 Gbps	5 Gbps	3 Gbps
最低带宽 (Gbps)	10 Mbps	10 Mbps	10 Mbps	10 Mbps	10 Mbps	10 Mbps
最小实例数	1	1	1	1	1	1
最大实例数	1	1	1	1	1	1
最低带宽单位	10 Mbps	10 Mbps	10 Mbps	10 Mbps	10 Mbps	10 Mbps

不同外形规格的许可证要求

许可证要求	MPX	SDX	VPX	CPX
零容量硬件购买	X	X		
带宽和版本订阅	X	X	X	
实例订阅		X	X	X

有关支持的平台、支持的最小带宽/实例、支持的最大带宽/实例以及支持的平台的最小带宽单位的更多信息，请参阅 MPX/CPX/VPX 的带宽和实例信息。

配置 **Citrix ADC** 池容量

池容量允许用户：

- 根据需要将许可证池中的许可证分配给 Citrix ADC。
- 将池容量许可证文件（带宽池或实例池）上传到 ADM。
- 根据实例的最小容量和最大容量从 Citrix ADM 分配许可证。

Citrix Application Delivery Management (ADM)

用户可以将 Citrix ADM 配置为用于 Citrix ADC 池容量的许可证服务器。Citrix ADC 实例可通过两种方式获取带宽和/或实例许可证：

- 第一个许可证签出请求应从 Citrix ADC (SDX/MPX/VPX) 发起，以获取其带宽和/或实例许可证。
- 用户可以稍后从 Citrix ADC 或 Citrix ADM 启动许可证签出。

> 注意：
>
> 仅当池许可证添加到 Citrix ADM 时，才会在 Citrix ADM 上显示池容量。

Citrix ADM 许可证池状态

- 已分配：许可证状态正常。
- 宽限： Citrix ADC 实例处于许可宽限期 30 天。
- 同步进行中： Citrix ADM 以 2 分钟的间隔从 Citrix ADC 获取信息。
- 同步进行中：在 Citrix ADM 和 Citrix ADC 之间同步许可证可能需要长达 15 分钟。Citrix ADM 可能已重新启动或触发 ADM HAS 故障转移。
- 部分分配： Citrix ADC 无法接受分配的容量，因为它可能以最大分配的速度运行。例如，Citrix ADC 的许可证池容量为 10 Gbps。当 ADC 重新启动时，10 Gbps 将被签回 ADM 许可证服务器。当 Citrix ADC 恢复在线时，它会尝试自动检查先前分配的 10 Gbps。同时，其他 ADC 可能已检出该带宽。如果许可证池没有足够的容量来分配全部 10 Gbps 或者甚至部分容量给此 ADC，则会显示部分已分配。
- 未受管理：为了便于管理，未将 Citrix ADC 添加到 ADM 中。这不会影响 Citrix ADC 许可，但会影响 ADM 的许可证监视。
- 连接丢失：无法从 ADM 访问 Citrix ADC 以实现可管理性。例如，存在网络连接问题、NITRO 无法正常工作或 Citrix ADC 密码不匹配。如果 NITRO 无法正常工作或 Citrix ADC 密码不匹配，则不会影响 Citrix ADC 许可。但是，它可能会影响 ADM 的许可证监视。
- 已分配：不适用于 **ADC**：如果许可证已从 ADC 签出或签出，则 Citrix ADC 可能需要重新启动，但 Citrix ADC 尚未重新启动。
- 未分配： ADC 实例中未分配许可证。

Citrix ADM 许可证池：常见问题

- ADC 将许可证服务器显示为“无法访问”：
  - 与许可证服务器（ADM 或 ADM 服务代理）的连接已断开 15 分钟以上。
  - ADC 处于宽限模式。
- ADC 将许可证服务器状态显示为“可 访问”，但用户尝试更改分配不起作用：
  - 与许可证服务器的连接最近断开，但 ADC 仍然没有错过第二个心跳。因此，它不是在恩典（尚未）。
  - 按“更改分配”将返回 0 0，这可能会显示配置的容量已丢失。
- ADC 显示容量/实例计数，但许可证服务器处于可访问/无法访问状态：
  - 与许可证服务器的连接已恢复，但 ADC 仍未错过第二个检测信号/或发送重新连接探测器。
  - 按“更改分配”返回一些数字，但不考虑配置的容量。
- ADC 说，使用 ADM 服务配置池许可时无法连接到许可证服务器：
  - 检查防火墙规则：27000 和 7279。
  - 代理未注册或 ADM 服务没有上传许可证文件（或文件错误）。

用例：**Citrix ADM** 许可证池使用情况报告

Citrix ADM 许可证池使用情况报告将确定每月高峰，供客户计划许可证使用量的增加，并规划下一次许可证池购买。

- 轮询：
  - 每 15 分钟从 ADC 轮询一次许可证数据。
- 每小时仅保持峰值：
  - 每个设备将存储一小时内的最大许可证使用要求。
- 报告：
  - 要生成的 GUI 报告显示指定时间范围内每台设备的使用情况。
- 导出:
  - 在指定时间范围内将计量数据导出为 CSV 或 XLS 的功能。
- 正在清除：
  - 清除作业将在每月的 1 日上午 12:10 运行。
  - 清除期间是可配置的（默认期间为 2 个月）。

在 **Citrix Application Delivery Management (ADM)** 上安装许可证文件

1. 在 Web 浏览器中，键入 Citrix ADM 的 IP 地址。例如 `http://192.168.100.1`。
2. 在“用 户名和密码”字段中，输入管理员凭据。
3. 在“配置”选项卡上，导航到“网络**”>“**许可证**”>“**设置”，然后单击“添加新许可证”。
4. 在“许可证文件”部分，选择以下选项之一：
   - 从本地计算机上载许可证文件-如果用户的本地计算机上已存在许可证文件，则用户可以将其上载到 Citrix ADM。要添加许可证文件，用户可以单击“浏览”以选择许可证文件 (.lic)。然后单击完成。

     > 注意：
     >
     > 如果上传的许可证文件未在 Citrix ADC 池容量中添加许可证，则可以选择许可证文件，然后单击“应用许可证”将许可证添加到池中。

   - 使用许可证访问代码-Citrix 通过电子邮件向客户购买的许可证访问代码 (LAC) 发送电子邮件。要添加许可证文件，请在文本框中输入 **LAC**，然后单击获取许可证。

     > 注意：
     >
     > 用户可以随时从许可证设置向 Citrix ADM 添加更多许可证。

从 **Citrix ADM** 分配 **Citrix ADC** 池容量许可证

必备条件：用户必须在 Citrix ADM 中注册 Citrix ADC 实例之前，才能通过 Citrix ADM 管理其实例池许可证。在 Citrix ADC GUI 中，导航到“系统**”>“**许可证**”>“**管理许可证”，然后在添加 **Citrix ADM IP** 时选中 **“**注册到 **Citrix ADM** 以实现可管理性”复选框。

> 注意：
>
> 如果用户尚未向 Citrix ADM 注册 Citrix ADC 实例，则可以从 Citrix ADM 中签出许可证。但是，它们无法从 Citrix ADM 分配到启用 Citrix ADC 池容量的实例。
>
> 在“用户名**”**和 **“**密码”字段中，输入 Citrix ADM 凭据。
>
> 如果 Citrix ADC (SDX/MPX/VPX) 密码不是默认密码，则此选项不起作用。

向许可证服务器注册实例后，按如下方式分配许可证

1. 在 Web 浏览器中，键入 Citrix ADM 的 IP 地址。例如 `http://192.168.100.1`。
2. 在“用 户名和密码”字段中，输入管理员凭据。
3. 在“配置”选项卡上，导航到“网络**”>“**许可证**”>“**池容量”。
4. 单击要管理的许可证池。
5. 单击 **>** 按钮，从可用实例列表中选择一个 Citrix ADC 实例。

6. 如果用户想要更改或释放许可证分配，请单击 “更改分配” 或 “释放分配”。
7. 如果用户单击 “更改分配”，将显示一个弹出窗口，其中包含许可证服务器中的可用许可证。
8. 用户可以通过设置 “分配” 下拉选项来选择 Citrix ADC 实例的带宽或实例分 配。进行所需的选择后，单击 “分配”。
9. 用户还可以从 “更改许可证分配” 窗口中的下拉选项 更改已分配的许可证版本。

具有许可证池的 **Citrix ADM** 高可用性 **(HA)**

以前，许可证池许可证已被节点锁定，并与 ADM 主节点的主机 ID 相关联。每当向辅助节点发生故障切换时，Citrix ADC 都会进入 30 天的宽限期，以避免因 ADM 无法访问事件而造成的任何中断。这允许 Citrix ADC 运行 30 天，即使无法访问 Citrix ADM 也是如此。但是，如果无法访问，新的 Citrix ADC 实例将无法从 ADM 许可证服务器中签出许可证，这意味着在 30 天宽限期内没有新的许可证签出。客户必须从 Citrix 许可证系统生成许可证文件的副本，以便在主节点没有恢复并且 30 天过去时间（这意味着他们正在生成新的许可证文件）时使此许可证工作。

解决方案

使用许可证池 HA 解决方案，如果主节点不回来，客户不必生成具有 ADM 故障转移到辅助节点的新许可证文件。在故障转移后，新的许可证签出将继续工作。许可证池许可证和 ADM 许可证现在与 Citrix ADM 主节点和辅助节点之间共享的虚拟主机 ID 相关联。

虚拟主机 **ID**

Citrix ADM 主节点和辅助节点共享相同的虚拟主机 ID。HA 部署中主节点或第一台 Citrix ADM 服务器的真实主机 ID 用作虚拟主机 ID。虚拟主机 ID 在 ADM 部署中自动生成，并以加密格式存储在 ADM 数据库中，客户无法更改。虚拟主机 ID 优先于真正的主机 ID。许可证文件从 ADM 主节点同步到辅助节点。Citrix ADC 使用 ADM 浮动 IP 地址签出许可证。从主节点故障切换到辅助节点时，许可证文件和虚拟主机 ID 将从主节点同步到辅助节点，以及浮动 IP 地址。

中断 **HA** 行为

如果客户启动 ADM 中断 HA 操作，则两个 ADM 节点都会保留虚拟主机 ID，然后启动中断 HA 工作流。节点 1 和节点 2 都可以继续签出许可证。自从 ADM 中删除浮动 IP 地址后，现有 Citrix ADC 将进入 30 天的宽限期。

大脑分裂

Citrix ADM 通过定期发送心跳来监视 ADM HA 节点的可用性。如果由于网络问题，检测信号无法到达另一个节点，则两个 ADM 节点都将自己提升为 ADM 主节点。在这种情况下，许可证服务器正在两个节点上运行。Citrix ADC 可以使用 ADM 服务器节点 IP 从两个节点签出许可证，因为这两个节点共享相同的虚拟主机 ID。节点 1 和节点 2 升级为 ADM 主节点。许可证服务器在具有相同虚拟主机 ID 的两台服务器上运行。许可证容量增加了一倍。生成 Citrix ADM 脑分裂相关事件和 ADM HA 宽限期相关事件。

从大脑分裂中恢复

在客户管理员发现并修复网络问题后，Citrix ADM 可以从大脑分裂状况中恢复。从 ADM 分裂脑恢复的工作流程如下。网络恢复后，Citrix ADM 会自动将 ADM 节点 1 检测为 ADM 主节点。Citrix ADM 从 ADM 节点 2 启动加入高可用性工作流程。Citrix ADM 节点 1 真实主机 ID 被选为虚拟主机 ID。Citrix ADM 恢复到正常的 HA 方案，并将许可证文件和虚拟主机 ID 同步到 ADM 节点 2。

在 **MPX-Z** 上配置池容量

MPX-Z 是启用了 Citrix ADC 池容量的 Citrix ADC MPX 设备。MPX-Z 支持高级版、高级版或标准版许可证的带宽池。MPX-Z 需要其平台许可证才能连接到许可证服务器。用户可以通过从本地计算机上传许可证文件或使用实例的硬件序列号或 Citrix ADC 实例 GUI 的系统 **>** 许可证部分的许可证访问代码来安装 MPX-Z 平台许可证。如果用户删除 MPX-Z 平台许可证，则会禁用池容量功能，并将所有签出的许可证签入许可证服务器。

用户可以动态修改 MPX-Z ADC 的带宽，而无需重新启动。仅当用户想要更改许可证版本时，才需要重新启动。

> 注意：
>
> 当用户重新启动 Citrix ADC 时，它会自动签出其配置容量所需的池许可证。

在 **VPX** 实例上配置池容量

启用池容量的 Citrix ADC VPX 实例可以从带宽池（高级/高级/标准版）中签出许可证。用户可以使用 Citrix ADC GUI 从许可证服务器签出许可证。

用户可以动态修改 VPX 实例的带宽，而无需重新启动。仅当用户想要更改许可证版本时，才需要重新启动。

> 注意：
>
> 当用户重新启动实例时，实例会自动签出其配置容量所需的池许可证。

将池许可证分配给 **MPX-Z** 或 **VPX** 实例

要分配许可证，请执行以下操作：

1. 在 Web 浏览器中，键入 Citrix ADC 实例的 IP 地址。例如 `http://192.168.100.1`。
2. 在“用户名”和“密码”字段中，输入管理员凭据。
3. 在“配置”选项卡上，导航到“系统”>“许可证”>“管理许可证”，单击“添加新许可证”，然后选择“使用池许可”。
4. 在“服务器 名称/IP 地址”字段中输入许可证服务器的详细信息。
5. 如果用户希望通过 Citrix ADM 管理其实例的池许可证，请选中“向 **Citrix ADM** 注册以便可管理性”复选框，然后输入 Citrix ADM 凭据。
6. 选择许可证版本和所需的带宽，然后单击获取许可证。
7. 用户可以通过选择“更改分配”或“发布分 配”来更改或 释放许可证分配。

8. 如果用户单击 “更改分配”，弹出窗口将显示许可证服务器上可用的许可证。

> 注意：
>
> 如果用户更改带宽分配，则不需要重新启动，但如果用户更改许可证版本，则需要热重新启动。

9. 用户可以从 “分配” 下拉列表中为 Citrix ADC 实例分 配带宽或实例。然后单击 获取许可证。
10. 用户可以从弹出窗口的下拉列表中选择许可证版本和所需的带宽。

> 注意：
>
> 带宽分配应该是最低带宽单位的倍数。

在 **SDX-Z** 上配置池容量

SDX-Z 实例是启用了池容量的 Citrix ADC SDX 实例。SDX-Z 支持高级版、高级版和标准版以及实例池的带宽池。用户应用 SDX-Z 平台许可证后，管理服务提供了从许可证服务器检出许可证并返回许可证服务器，以及为在 SDX-Z 平台上运行的 Citrix ADC 实例分配带宽容量的选项。

> 注意：
>
> 在 SDX-Z 上运行的 Citrix ADC VPX 实例无法直接将许可证从许可证服务器签出或签入许可证服务器。这可以由 SDX 中的管理服务完成。

用户可以通过从本地计算机上传许可证文件或使用实例的硬件序列号或许可证访问代码来安装 SDX-Z 平台许可证。

如果用户删除 SDX-Z 平台许可证，则会禁用池容量功能，并将所有许可证签回许可服务器。

> 注意：
>
> 如果用户重新启动实例，则实例会签出其配置容量所需的池许可证。

SDX 上的池容量

实例池

SDX 设备可以预配置 SDX 设备的实例池中可用数量相同的实例。

带宽池

在 Citrix ADC 实例预配期间，会为该实例分配带宽。用户可以选择版本和所需的带宽来预配虚拟 Citrix ADC 实例。仅当实例具有适用于所请求版本的足够带宽时，管理服务才允许继续进行置备。如果带宽不足，将通知用户。

> 注意：
>
> 带宽修改不需要重新启动实例。

将池许可证分配给 **SDX-Z** 实例

要分配许可证，请执行以下操作：

1. 在 Web 浏览器中，键入 Citrix ADC SDX-Z 实例的 IP 地址。例如 `http://192.168.100.1`。
2. 在“用户名”和“密码”字段中，输入管理员凭据。
3. 在“配置”选项卡上，导航到“系统”>“许可证”，然后转到“池容量”。
4. 在“服务器 名称/IP 地址”字段中输入许可证服务器的详细信息。
5. 如果用户希望通过 Citrix ADM 管理其实例的池许可证，请选中“向 **Citrix ADM** 注册以便可管理性”复选框，然后输入 Citrix ADM 凭据。
6. 用户可以通过选择“更改分配”或“发布分 配”来更改或 释放许可证分配。

   > 注意：
   >
   > 签出的许可证由管理服务存储在单独的池中。

7. 要更改 **SDX-Z** 实例中特定 **VPX** 实例的许可证分配，请从“实例”部分选择该实例，然后单击“更改分配”。新窗口将显示可用许可证。
8. 用户可以从功能许可证下拉列表中更改实例的带宽版本，并在吞吐量 **(mbps)** 字段中更改所需的带宽。然后单击完成。

> 注意：
>
> 带宽分配应该是对应尺寸规格的最低带宽单位的整数倍数。

在 **CPX** 实例上配置池容量

在预配 Citrix ADC CPX 实例时，用户可以将 Citrix ADC CPX 实例配置为使用 Citrix ADC 池容量。在码头窗口中，用户必须提供 Citrix ADC 许可服务器 (Citrix ADM) 详细信息。Citrix ADC CPX 实例会从实例池中签出许可证。

> 注意：
>
> 默认情况下，Citrix ADC CPX 实例会从实例池中检出实例许可证，吞吐量会自动设置为 1000 Mbps。用户无法修改分配给实例的 1000 Mbps 带宽。

用户可以从 Docker App Store 下载 Citrix ADC CPX。在 Docker 主机上，要下载 Citrix ADC CPX，请运行以下命令：

```
docker pull store/citrix/netscalercpx:[version number]
```

要在 Provisioning Citrix ADC CPX 实例的同时配置池容量，请执行以下操作：

在预配 Citrix ADC CPX 实例时，将 Citrix ADC 授权服务器 (Citrix ADM) 定义为泊坞窗主机中的环境变量，然后按如下所示运行命令：


```
docker run -dt -P -e LS_IP=<LS_IP_ADDRESS> -e LS_PORT=<LS_PORT> --name <
container_name> --ulimit core=-1 -e EULA=yes -v <host_dir>:/cpx --cap-add=
NET_ADMIN >REPOSITORY<:>TAG<
```

地点：

- <LS_IPADDRESS> 是 Citrix ADC Licensing Server (Citrix ADM) 的 IP 地址。
- <LS_PORT> 是 Citrix ADC 许可服务器的端口。默认情况下，端口为 27000。

最佳实践、角落案例和常见问题解答

升级 **SDX** 许可证-永久到池

当 SDX 上的许可证从永久许可升级到池许可时，SDX 不需要重新启动。SDX 和 VPX 都不需要重新启动才能转移到池许可。SVM 会自动将一个或多个 VPX 转换为池许可证。

用户应确保以下顺利过渡：

- 确保 SDX 具有适当的零容量许可证。
- 确保 Citrix ADM 服务器具有足够的容量以供 SDX 中 VPX 实例中使用的许可证版本使用。
- 确保从 SVM 中的 ADM 检出所有 VPX 实例的足够带宽容量。
  - 例如：如果 SDX 有 10 个 VPX 实例，并且它们一起消耗 40 Gbps 高级和 20 Gbps 高级，请确保首先通过 SVM 签出，以便 VPX 实例可以获得这些许可证。

在 **30** 天宽限期内运行 **Citrix ADC** 实例

如果 Citrix ADC 实例在从池接收许可证后断开与 Citrix ADM 的连接，则允许它在 30 天的宽限期内继续运行，同时尝试重新建立与许可证服务器的连接。即使 Citrix ADC 重新启动，许可证仍保持在 30 天的宽限期内，并且实例将继续运行。

客户端启动的签入（随机）：**Citrix ADM** 启动连接的方案

对于客户端启动的签入（随机），是否存在 Citrix ADM 将启动此连接的情况？

Citrix ADM（许可证服务器）和 Citrix ADC（许可证客户端）交换心跳数据包，以监视在客户端和服务器之间建立的连接的运行状况。此时间段是随机化的，以避免所有 Citrix ADC 客户端同时向 Citrix ADM 许可证服务器发送请求。

如果客户端和服务器之间的许可证连接存在问题，则执行以下操作：

- 如果 Citrix ADM 未收到来自 Citrix ADC 客户端的心跳数据包，则 Citrix ADM 服务器会声明退回分配给该特定 Citrix ADC 的许可证。
- 如果 Citrix ADC 未收到心跳数据包，则 Citrix ADC 将移至 30 天的许可宽限期。
- 如果 Citrix ADC 收到与 Citrix ADM 许可证服务器建立的许可证服务器连接信号，则 Citrix ADC 将再次从 Citrix ADM 中签出许可证。

Citrix ADC 重新启动期间的共享带宽分配

如果在 Citrix ADC 重新启动期间分配了带宽，则池带宽许可证是否部分分配（最多不超过池中的可用带宽），还是不分配许可证？

Citrix ADC 最初尝试检出用户配置的池容量。如果此尝试失败，Citrix ADC 将尝试检出 Citrix ADM 中的可用池容量。

> 注意：
>
> 此功能仅适用于 MPX 和 VPX。如果 Citrix ADM 没有足够的容量，SDX 会尝试部分许可证签出。

许可证不匹配警报（**Citrix ADC** 收到部分许可证或没有许可证）

如果出现不匹配（例如，Citrix ADC 未收到许可证或仅收到部分许可证），Citrix ADM 是否能够将这种情况标记为进行调节？

如果 Citrix ADC 未收到许可证或部分许可证的许可证不匹配，则 Citrix ADM 必须标记此情况以进行协调。在以下情况下，可能会发生许可证不匹配：

- 如果 Citrix ADC 重新启动，Citrix ADC 会在重新启动后再次签出许可证。这将清除池容量不匹配事件。
- 如果 Citrix ADM 重新启动，则 Citrix ADC 和 Citrix ADM 会在心跳间隔内同步许可证信息，并清除此事件。
- 如果重新启动/许可证服务器重新连接后 Citrix ADC 签出失败，则不会自动恢复。用户需要再次从池中手动签出许可证。

Citrix ADM 的高可用性故障转移

在 Citrix ADM 的 HA 故障转移期间，许可文件如何同步以及可能发生哪些故障（例如，在主节点上更新时，Citrix ADC 上的 SSL 证书有时不会复制到辅助节点）？

Citrix ADM 高可用性 (HA) 支持池许可从 12.1-50.x 以后的软件版本中获得。Citrix ADM 会定期同步主 Citrix ADM 中上载的文件到辅助 Citrix ADM。因此，文件同步是在 HA 故障切换事件发生之前完成的。因此，不可能发生文件同步失败。例如，Citrix ADC 上在主 Citrix ADM 上已更新但尚未复制到辅助 Citrix ADM 上的 SSL 证书。

辅助 **Citrix ADM** 数据库运行状况检查

是否有辅助数据库问题的运行状况检查？辅助 Citrix ADM 是否验证要共享的信息是否正常，以避免复制不正常的信息？

许可证信息保存在许可证服务器内存中（在 Citrix ADM 中）。此信息未同步到 Citrix ADM 辅助信息。所有许可证签出/签入都是根据许可证服务器的内存信息执行的。Citrix ADM 数据库仅用于存储从许可证服务器（在 Citrix ADM 中）和 Citrix ADC 实例收集的报告。

Citrix ADM 仅同步许可证文件从 Citrix ADM 主要文件到辅助文件（从 12.1-50.x 版本开始）。

在 Citrix ADM HA 故障切换期间，Citrix ADC 会在心跳间隔后从 ADM 中签出许可证，并在心跳间隔后更新许可证服务器内存。

许可证不可用的反向宽限期

许可证不可用是否有反向宽限期，允许实例在宽限期内保持许可，而不是立即关闭？例如，Citrix ADC 尝试签入，而 Citrix ADM 则表示没有可用的有效许可证。

目前正在研究这一特定问题的解决办法。当我们对此问题有建议的解决方案时，我们将通知用户。

用于在 **Citrix ADM** 上进行许可的可配置系统 **ID**

是否支持用于在 Citrix ADM 上进行许可的可配置系统 ID（而不是基于 MAC 地址的系统）？

目前尚未计划支持用于许可的可配置系统 ID。

文件一致性检查或机制

对于文件（包括从主数据库复制到辅助 Citrix ADM 的许可证），是否有任何一致性/损坏检查或机制来确保主数据库损坏不会将问题复制到辅助？

Citrix ADM 在文件系统中维护许可证文件，并使用 RSYNC 实用程序进行同步。因此，数据库问题不会影响许可证文件。

用于许可证登入/签出的 **Citrix ADM** 代理使用

> 注意：
>
> 目前，每个给定租户只支持 1 个代理程序用于公有云中的池容量。

突发许可

突发许可是通过向池容量添加消耗元素来增强池容量。客户可能无法准确预测其带宽需求。这可能是由于各种原因或特殊情况，例如合并或收购或特殊事件，如黑色星期五销售，这些事件可能会导致流量高峰并超出客户当前拥有的容量。一般来说，该行业正朝着由许多云产品驱动的消费模式（按用量付费）发展。

池容量的突发许可允许客户购买池容量作为基本订阅，并可在需要时选择突增超过购买的池容量。对于基本订阅池，客户必须提前支付订阅期限（1 年、3 年或 5 年）的费用。对于突发池，客户可以无需提前付费使用，并将根据消费后每年的实际使用量收取费用。

借助突发许可，Citrix ADC 将看到一个池、基本池和突发池的组合视图。如果 Citrix ADC 无法从基本池中签出，它将尝试从突发池中签出。

ADM 中的突发许可证报告

将为基础池和拆分池生成许可证使用情况报告。每月将生成许可证使用情况报告，以报告许可证池中每小时的最大许可证使用情况。这些月度报告的清除间隔最长为三年。将生成年度许可证使用情况报告，以报告许可证池中每月的最大许可证使用情况。这些年度报告的清除间隔最长为六年。客户可以在指定时间范围内将计量数据导出为 CSV、XLS 和 PDF。

BLX 池授权

BLX 使用与 Citrix ADC VPX 相同的许可证池。

虚拟 **CPU** 许可概述

数据中心正在转向更新的技术，这些技术可简化网络功能，同时提供更低的成本和更高的可扩展性。较新的数据中心体系结构必须至少包含以下功能：

- 软件定义网络 (SDN)。
- 网络功能虚拟化 (NFV)。
- 网络虚拟化 (NV)。
- 微型服务.

这样一个运动还要求软件要求具有动态、灵活和敏捷性，以满足不断变化的业务需求。许可证还将由一个中央管理工具管理，并充分了解使用情况。

以前，Citrix 软件 ADC 许可证是根据实例的带宽消耗分配的。根据其绑定的许可证版本（标准版、高级版或高级版），Citrix 软件 ADC 被限制使用特定的带宽和其他性能指标。为了增加可用带宽，用户必须升级到提供更多带宽的许可证版本。在某些情况下，带宽要求可能较小，但对于其他 L7 性能（如 SSL TPS、压缩吞吐量等）的要求更高。在这种情况下，升级 Citrix 软件 ADC 许可证可能不适合。但是，用户可能仍然需要购买带宽较大的许可证，以解锁 CPU 密集型处理所需的系统资源。Citrix 软件 ADM 现在支持基于虚拟 CPU (vCPU) 数量分配许可证。

使用基于 vCPU 的许可功能，许可证可指定特定 Citrix 软件 ADC VPX 有权访问的 vCPU 数量。Citrix 软件 ADC VPX 只能从许可证服务器动态签出许可证，只能针对可以运行软件 ADC 的 vCPU 数量。vCPU 许可证支持所有软件 ADC 外形规格，包括 VPX、CPX 和 BLX。

与池许可证容量和 CICO（签入、签出）许可证功能类似，Citrix SW ADM 许可证服务器管理一组单独的 vCPU 许可证。此外，为 vCPU 许可证管理的三个版本是标准版、高级版和高级版本。这些版本解锁了与带宽许可证版本解锁的功能集相同的功能集。

vCPU 数量可能发生变化或许可证版本发生变化时。在这种情况下，用户必须始终关闭实例，然后才能发起新许可证集的请求。用户必须在签出许可证后重新启动 Citrix 软件 ADC。

使用 **GUI** 在 **Citrix** 软件 **ADC VPX** 中配置许可服务器：

1. 在 Citrix SW ADC VPX 中，导航到 “系统”>“许可证”，然后单击 “管理许可证”。
2. 在 “许可证” 页上，单击 “添加新许可证”。
3. 在 “许可证” 页面上，选择 “使用远程许可” 选项。
4. 从 “远程 许可模式” 列表中选择 **CPU** 许可。
5. 键入许可证服务器的 IP 地址和端口号。
6. 单击 继续。

> 注意：用户必须始终向 Citrix 软件 ADM 注册 Citrix 软件 ADC VPX 实例。如果尚未完成，请启用“向 Citrix 软件 ADM 注册”并键入 Citrix ADM 登录凭据。

7. 在“分配许可证”窗口中，选择许可证类型。此窗口将显示总数和可用的 vCPU 以及可分配的 CPU。单击获取许可证。
8. 单击下一页上的 重新启动以申请许可证。

> 注意：用户还可以释放当前许可证并从其他版本中签出。例如，用户已经在其实例上运行标准版许可证。他们可以释放该许可证，然后从高级版中签出。
> 注意：用户必须确保为每个 vCPU 分配正确的内存量（2 Gb）。检查每个 vCPU 分配的内存。如果它们不正确，请增加内存并重新启动 Citrix 软件 ADC VPX 实例。

使用 **CLI** 在 **Citrix** 软件 **ADC VPX** 中配置许可服务器：

在 Citrix 软件 ADC VPX 控制台中，为以下两个任务键入以下命令：

1. 要将许可服务器添加到 Citrix 软件 ADC VPX，请执行以下操作：
   - 键入许可方 IP 地址。例如 http://192.168.100.1。
2. 要申请许可证，请执行以下操作：
   - 设置容量-vcpu-版白金
   - 出现提示时，键入以下命令重新启动实例：**reboot -w**

管理 **Citrix** 软件 **ADM** 上的 **vCPU** 许可证

1. 在 Citrix SW ADM 中，导航到网络 > 许可证 > 虚拟 **CPU** 许可证。
2. 此页面显示为每种类型的许可证版本分配的许可证。
3. 单击每个圆环（标准、高级、高级）中的数字可查看使用此许可证的 Citrix SW ADC 实例。

适用于 **Citrix** 软件 **ADC CPX** 的 **vCPU** 许可

在预配 Citrix 软件 ADC CPX 实例时，用户可以根据实例中的 CPU 使用情况将 Citrix 软件 ADC CPX 实例 Provisioning 为从许可证服务器签出许可证。

Citrix 软件 ADC CPX 依赖于在 Citrix 软件 ADM 上运行的许可证服务器来管理许可证。Citrix 软件 ADC CPX 在许可证服务器启动时从许可证服务器中签出许可证。当 Citrix 软件 ADC CPX 关闭时，许可证会签回许可证服务器。

用户可以从 Docker App Store 下载 Citrix SW ADC CPX。在 Docker 主机上，要下载 Citrix 软件 ADC CPX，请运行以下命令：

- docker pull store/citrix/netscalercpx:[版本号]

CPX 许可证有三种许可证类型：

1. CPX 和 VPX 支持的虚拟 CPU 订阅许可证

2. 池容量许可证
3. CP1000 许可证仅支持 CPX 的单到多个 vCPU

要在置备 **Citrix** 软件 **ADC CPX** 实例的同时 **Provisioning vCPU** 订阅许可证，请执行以下操作：

用户需要指定 Citrix ADC CPX 实例使用的 vCPU 许可证数。

- 此值通过 Docker、Kubernetes 或中索斯/马拉松作为环境变量输入。
- 目标变量是 **CPX_CORES**。CPX 可以支持 1 到 7 个内核。

要指定 2 个内核，用户可以执行 docker 运行命令，如下所示：

- docker run -dt -P –privileged=true –ulimit core=-1 -v<host_dir>:/cpx -e EULA=yes -e CPX_CORES=2

Provisioning Citrix ADC CPX 实例时，请在码头运行命令中将 Citrix 软件 ADC 许可服务器定义为环境变量，如下所示：

- docker run -dt -P –privileged=true –ulimit core=-1 -v<host_dir>:/cpx -e EULA=yes -e CPX_CORES=2 -e LS_IP=<LS_IP_ADDRESS> -e LS_PORT=<LS_PORT> cpx:11.1

其中，

- <LS_IP_ADDRESS> 是 Citrix ADC 授权服务器的 IP 地址。
- ´<LS_PORT> 是 Citrix ADC 许可服务器的端口。默认情况下，端口为 27000。

> 注意：默认情况下，Citrix 软件 ADC CPX 实例会从 vCPU 订阅池中签出许可证。如果实例使用 “n”CPU 运行，CPX 实例会检出 “n” 许可证数量。

要在预配 **Citrix** 软件 **ADC CPX** 实例时 **Provisioning Citrix** 软件 **ADC** 池容量或 **CP1000** 许可证，请执行以下操作：

如果用户希望使用池许可（基于带宽）或 CPX 专用池（CP1000 或基于私有池）签出 CPX 实例的许可证，则用户必须相应地提供环境变量。

例如，

- docker run -dt -P –privileged=true –ulimit core=-1 -v<host_dir>:/cpx -e EULA=yes -e CPX_CORES=2 -e LS_IP=<LS_IP_ADDRESS> -e LS_PORT=<LS_PORT> -e PLATFORM=CP1000 cpx:11.1

CP1000. 此命令触发从 CP1000 池（CPX 专用池）检出。然后，Citrix 软件 ADC CPX 实例检索为 CPX_CORES 指定的 “n” 核心数量的实例。最常见的用例是为单个实例的检出指定 n = 1。多核 CPX 使用案例检出 “n”vCPU（其中 “n” 是从 1 到 7）。

- docker run -dt -P –privileged=true –ulimit core=-1 -v<host_dir>:/cpx -e EULA=yes -e CPX_CORES=2 -e LS_IP=<LS_IP_ADDRESS> -e LS_PORT=<LS_PORT> -e BANDWIDTH=2000 cpx:11.1

集合容量。此命令从实例池中检出一个许可证，并消耗来自白金带宽池的 1000 Mbps 带宽，但使 CPX 能够运行高达 2000 Mbps。在池许可中，不收费前 1000 Mbps 的费用。

> 注意：从带宽池检出时，指定所需目标带宽的相应 vCPU 数，如下表所述：

内核数量 (vCPU)	最大带宽
1	1000 Mbps
2	2000 Mbps
3	3500 Mbps
4	5000 Mbps
5	6500 Mbps
6	8000 Mbps
7	9300 Mbps

Citrix ADM 服务池许可概述

Citrix ADM 服务池许可是 Citrix ADM 服务的一项功能。Citrix ADM 服务池许可使客户能够将池许可证与 ADM 服务一起使用。借助 ADM Service 池许可，用户可以管理分布在多个数据中心之间的多个 ADC 的许可证分配。Citrix ADM 服务池许可支持每个数据中心的多个代理。许可由整个 ADM 服务进行管理，而不仅仅是一个代理。因此，用户应将代理视为基于云的许可证服务器的网络代理。Citrix ADM 服务池许可还支持许可证访问代码，用于从 Citrix 门户获取许可证。Citrix ADM 服务提供了一个控制板，可帮助用户管理容量分配和查看许可证使用情况。

上传到 ADM 服务的许可证应为池许可证类型，并且应锁定到云虚拟主机 ID。代理程序应打开入站端口 27000 和 7279。如果代理出现故障，连接到代理的 ADC 将进入宽限模式。如果代理出现故障，未连接到代理的 ADC 将在大约 20 分钟内不会反映任何配置更改。他们将继续正常运作。更改 ADC（或许可证版本）上的许可证类型需要热重新启动。许可证的容量更改不需要重新启动。

在使用 ADM 服务的任何功能之前，必须通过注册/添加过程将代理和 ADC 告知 Citrix ADM 服务。ADM 服务注册/添加工作流程如下所示：

> 注意：前三个步骤是指 ADM 上下文。

1. 在 ADM 服务中注册代理。
2. 将 ADC 实例添加到这些代理（在 ADM 服务上）。
3. 上传 ADM 服务上的许可证。

   > 注意：接下来的三个步骤是指 ADC 上下文。

4. 在 ADC GUI 中选择远程许可。
5. 输入 ADC 注册到的代理的 IP 地址。

6. 分配（在 ADC 上）

> 注意： ADM 服务现在可用于监视所有 ADC 中的池许可证以及更改分配。

Kubernetes 中使用 Diamanti 和 Nirmata 验证的参考设计的 Citrix ADC CPX

May 20, 2020

待测试的特性和功能

测试用例：**CPX** 作为南北和 **Hairpin** 东西的 **Ingress Controller** 和设备：

将除 **VPX** 之外的所有测试用例设置为南北：

- 群集中的两个 CPX（CPX-1、CPX-2）
- ADM 作为许可服务器
- 群集中的 Prometheus 导出程序容器
- Prometheus 服务器和 Grafana（作为 Kubernetes 中的 pod 或 Kubernetes 服务器外部的 pod）
- 几个前端应用程序
- 几个后端应用程序

I. VPX 作为南北

1. SDX 前端 Diamanti 平台上的 VPX
    - 测试 SSL 卸载并重新加密，为每个 SSL 连接插入 X-forward
    - 在 SSL 会话中插入 X-forward

II. CPX 作为南北设备

1. CPX-1。设置 HTTPS 入口，并支持两个或三个具有指定入口类的 HTTPS 应用程序：
    - 演示多个内容交换策略的创建：每个前端应用一个。
    - 每个 CPX 演示多个通配符证书：每个应用程序一个通配符证书。
    - 演示 CPX 将流量卸载和重新加密到前端应用程序。
    - 演示不同的负载平衡算法。
    - 演示一个容器的持久性。
2. CPX-1。使用指定的入口类别设置单独的 TCP 入口：
    - 插入 TCP 应用程序，如 MongoDB。
    - 显示 TCP VIP 创建。
    - 显示 TCP 客户端流量击中 MongoDB 窗格。

- 显示默认 TCP 应用程序运行状况检查。

3. CPX-1。使用指定的入口类别设置单独的 TCP-SSL 入口：
   - 演示 TCP-SSL VIP 的 SSL 卸载和重新加密。
   - 重复测试用例 2。
4. 每个应用程序 CPX。使用单独的入口类：
   - 使用 CPX-2 仅支持一个应用程序，重复测试用例 1—3。
5. 每个团队的 CPX。使用入口类：
   - 为 2 支球队分配不同的入队类。
   - 演示测试用例 1 作为 CPX 可以为单个团队配置入口规则的证据。
6. 自动缩放前端 Pod：
   - 增加前端 pod 的流量，并确保 pod 自动缩放。
   - 显示 CPX-1 将新的窗格添加到服务组。
   - 演示 HTTPS 入口 VIP。
7. 4—7 个 vCPU 支持：
   - 使用 4 个或 7 个 vCPU 配置 CPX-1。
   - 显示 HTTPS TPS 的性能测试，整个加密 BW。

III. CPX 作为发夹东西设备

1. CPX-1。为南北流量创建 HTTPS 入口，如第 I.1 节所述：
   - 向前端应用程序公开后端应用程序。
   - 显示两个应用程序之间的流量。
   - 将后端应用程序显示给另一个后端应用程序。
   - 显示应用程序之间的流量。
2. CPX-1。按照步骤 1 的说明进行操作。此外，显示端到端加密：
   - 后端应用程序到后端应用程序使用 CPX-1 进行加密进行卸载和重新加密。
3. 自动缩放后端 pod：
   - 演示 CPX-1 将后端自动缩放的后端 pod 添加到服务组。

IV. CPX 与 **Prometheus** 和 **Grafana** 集成

1. 将 Prometheus 容器插入 Kubernetes 群集：
   - 将容器配置为每个应用程序导出的推荐计数器。
   - 演示向 Prometheus 服务器发送计数器数据的导出程序容器。

- 显示 Grafana 控制板，说明从 CPX 来自 Prometheus 服务器的数据。
- 目标是展示开发人员可以使用常用于 DevOps 的云原生工具。

2. 演示集成 Kubernetes 滚动部署：

   - 在 Nirmata 中插入新版本的应用程序。
   - 显示 Kubernetes 将新应用程序版本部署到群集中。
   - 演示 CPX 响应 Kubernetes 的滚动部署命令，将 100% 的流量从旧版应用程序转移到新版本的应用程序。

适用于 **Citrix ADC CPX** 部署的 **Citrix** 解决方案

1. 自定义协议：默认情况下，CITRIX INGRESS CONTROLLER 使用默认协议 (HTTP/SSL) 自动配置。CITRIX INGRESS CONTROLLER 支持使用注释配置自定义协议 (TCP/SSL-TCP/UDP)。

   注释：

   `ingress.citrix.com/insecure-service-type: "tcp"` [选择 LB 协议的注释]

   `ingress.citrix.com/insecure-port: "53"` [支持自定义端口的注释]

2. 微调 **CS/LB/Servicegroup** 参数：默认情况下，CITRIX INGRESS CONTROLLER 使用默认参数配置 ADC。借助 NetScaler ADC 实体参数 (**lb/servicegroup**) 注释，可以对参数进行微调。

   注释：

   LB 方法：`ingress.citrix.com/lbvserver: '{ "app-1":{ "lbmethod":"ROUNDROBIN" } } '`

   持 久 性：`ingress.citrix.com/lbvserver: '{ "app-1":{ "persistencetype":"sourceip" } } '`

   NITRO API

3. 每个应用程序的 **SSL** 加密： CITRIX INGRESS CONTROLLER 可以通过智能注释选择性地为应用程序启用 SSL 加密。

   注释：

   `ingress.citrix.com/secure_backend: '{ "web-backend": "True" }` [针对每个应用程序有选择性地启用加密的注释]

4. 入口的默认证书： CITRIX INGRESS CONTROLLER 可以将默认证书作为参数。如果入口定义没有密码，则采用默认证书。密码需要在命名空间中创建一次，然后名称空间中的所有入口都可以使用它。

5. **Citrix** 多个入口类支持：默认情况下，CITRIX INGRESS CONTROLLER 侦听 k8s 群集中的所有入口对象。我们可以通过入口类注释来控制 ADC（第 1 层 MPX/VPX 和第 2 层 CPX）的配置。这有助于每个团队独立管理其 ADC 的配置。入口类有助于部署解决方案，以便为特定命名空间以及一组名称空间配置 ADC。与其他供应商提供的支持相比，该支持更为通用。

注释：

`kubernetes.io/ingress.class: "citrix"` [通知 CITRIX INGRESS CONTROLLER 仅配置属于特定类的入口]

6. 可视性： Citrix `k8s` 解决方案与 Prometheous/Grafana 等 `cncf` 可视性工具集成，用于指标收集，以支持更好的调试和分析。Citrix prometheus 导出程序可以将指标提供给 Prometheus 以作为时间序列表对 Grafana 可见。

有关使用微服务体系结构的详细信息，请参阅 GitHub 中的 README.md 文件。您可以在 配置 文件夹中找到 `.yaml` 文件。

POC 故事行

有三个团队在 kubernetes 群集上运行他们的应用程序。在 Citrix 入口类的帮助下，每个团队的配置在不同的 CPX 上独立管理。

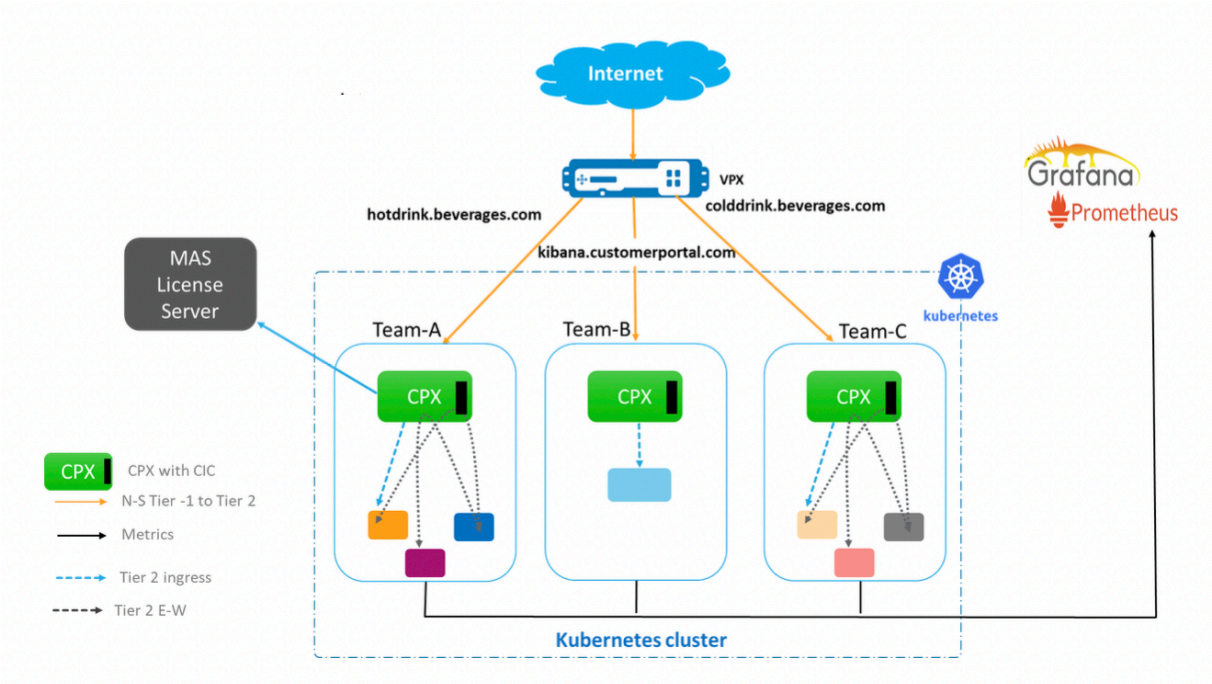
每个团队的应用程序在单独的命名空间（team-hotdrink、team-colddrink 和 team-redis）中运行，并且所有 CPX 都在 CPX 命名空间中运行。

`team-hotdrink`： SSL/HTTP 入口，持久性，单方法，每个应用程序的加密/代码，默认证书。

`team-colddrink`： SSL-TCP 入口

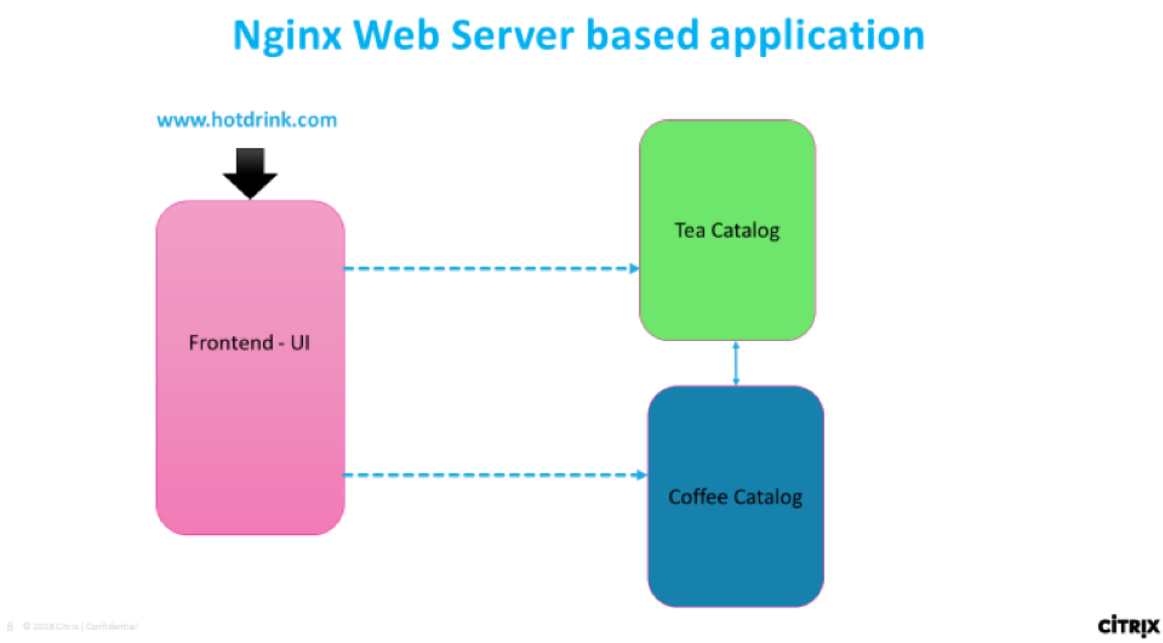
`team-redis`： TCP 入口

POC 设置

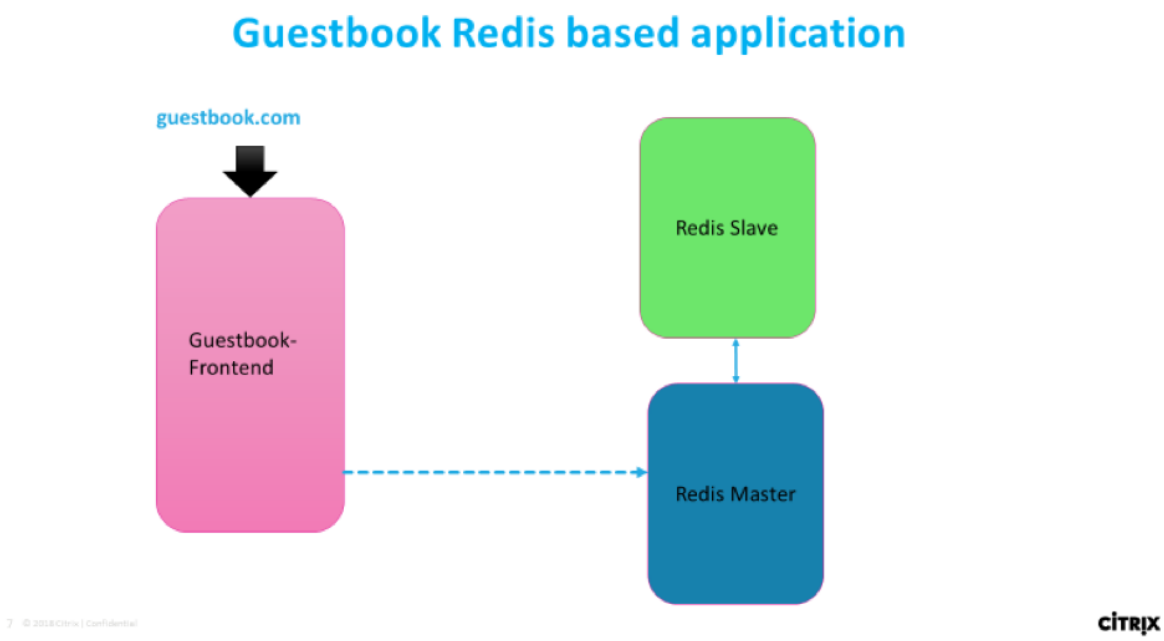


应用程序流程

HTTP/SSL/SSL-TCP 用例：



TCP 使用案例：



获取码头图像

提供的 YAML 命令正在从码头存储库中获取图像。

图像也可以被拉取并存储在本地存储库中。您可以通过在 YAML 中编辑图像参数来使用它们。

使用 **Nirmata** 的分步应用程序和 **CPX** 部署

1. 在 YAML 中上传群集角色和群集角色绑定，并使用 Nirmata (rbac.yaml) 在群集中进行应用。

   a) 转到“群集”选项卡。

   b) 选择群集。

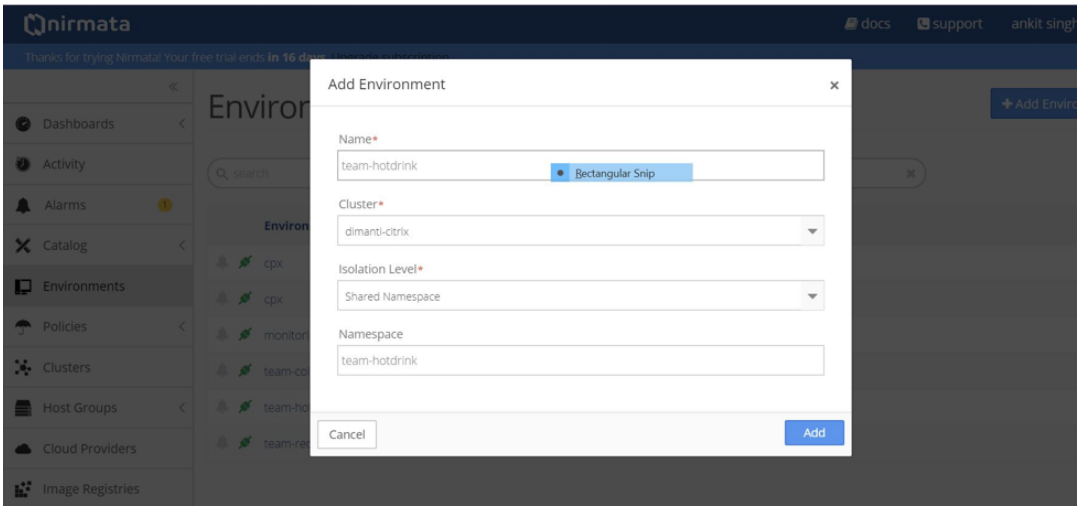
   c) 在设置中，从“应用 YAML”选项 应用 **YAML** 。

2. 创建运行 CPX 和应用程序的环境。

   a) 转到环境选项卡。

   b) 单击“添加环境”选项卡。

   - 选择群集并在共享命名空间中创建环境。



   c) 为不同团队运行 Prometheus、CPX 和应用创建以下环境。

   - 创建环境：cpx
   - 创造环境：team-hotdrink
   - 创造环境：team-colddrink
   - 创建环境：team-redis

3. 使用 Nirmata 上传 `.yaml` 应用程序。

   a) 转到目录选项卡。

   b) 单击添加应用程序。

   c) 单击添加添加应用程序。

   添加应用程序：team-hotdrink (team_hotdrink.yaml)。应用程序名称：`team-hotdrink`。

添加应用程序：team-colddrink (team_coldrink.yaml)。应用程序名称：`team-colddrink`。

添加应用程序：team-redis (team_redis.yaml)。应用程序名称：`team-redis`。

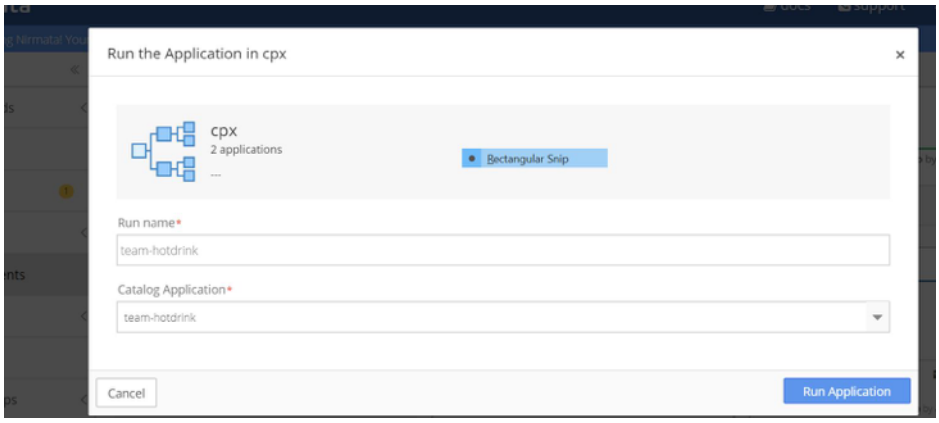
添加应用程序：cpx-svcacct (cpx_svcacct.yaml)。应用程序名称：cpx-svcacct。

> 注意：
>
> 具有内置 CITRIX INGRESS CONTROLLER 的 CPX 需要在运行该帐户的命名空间中有一个服务帐户。对于 Nirmata 中的当前版本，请在 cpx 环境中使用 `cpx_svcacct.yaml` 进行创建。

添加应用程序：cpx (cpx_wo_sa.yaml)。应用程序名称：cpx。

4. 使用 Nirmata 运行 CPX。

   a) 转到环境选项卡并选择正确的环境。
   b) 单击运行应用程序运行应用程序。
   c) 在 cpx 环境中，运行 `cpx-svcacct` 应用程序。`cpx-svcacct` 使用目录应用程序`cpx-svcacct`中的运行名称进行选择。
   d) 在 cpx 环境中，运行 cpx 应用程序。从目录应用程序中选择 cpx。



> 注意：
>
> CPX 部署需要一些小的解决方法，因为安装程序使用的是 Nirmata 的早期版本。
>
> a) 创建 CPX 部署时，请勿设置 `serviceAccountName`。`serviceAccountName` 可以稍后添加。解决方法是自动重新部署容器。
> b) 直接在环境中导入入口的 TLS 密钥。这可确保保留类型字段。

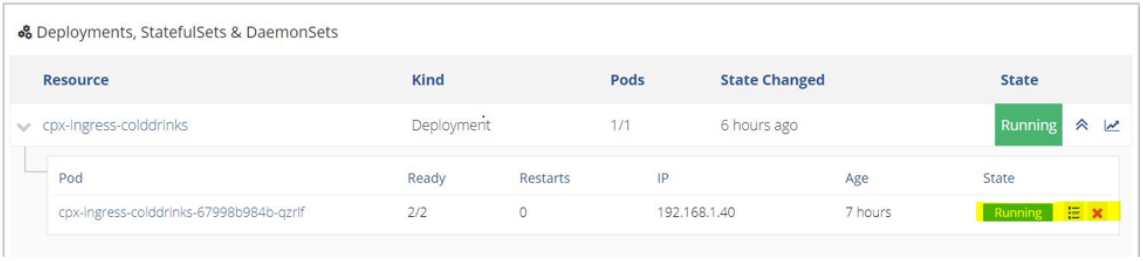
   a) 运行应用程序后，转到 **CPX** 应用程序。
   b) 在部署 **> StatefulSets** 和 **DaemonSets** 选项卡下，单击 `cpx-ingress-colddrinks` 部署。
   c) 在下一页上，编辑 **Pod** 模板。在服务帐户中输入 **CPX**。
   d) 返回 **CPX** 应用程序。
   e) 对 `cpx-ingress-hotdrinks` 和 `cpx-ingress-redis` 部署重复相同的过程。

应用服务帐户，重新部署容器。等待窗格出现，并确认服务帐户是否已应用。

同样可以通过在 **Diamanti** 群集中使用以下命令进行验证。


```
[diamanti@diamanti-250 ~]$ kubectl get deployment -n cpx -o yaml |
    grep -i account
        serviceAccount: cpx
        serviceAccountName: cpx
        serviceAccount: cpx
<!--NeedCopy-->
```

> 注意：如果未应用 serviceAccount，则取消 CPX 容器。重新创建时使用的部署，出现了 serviceAccount。



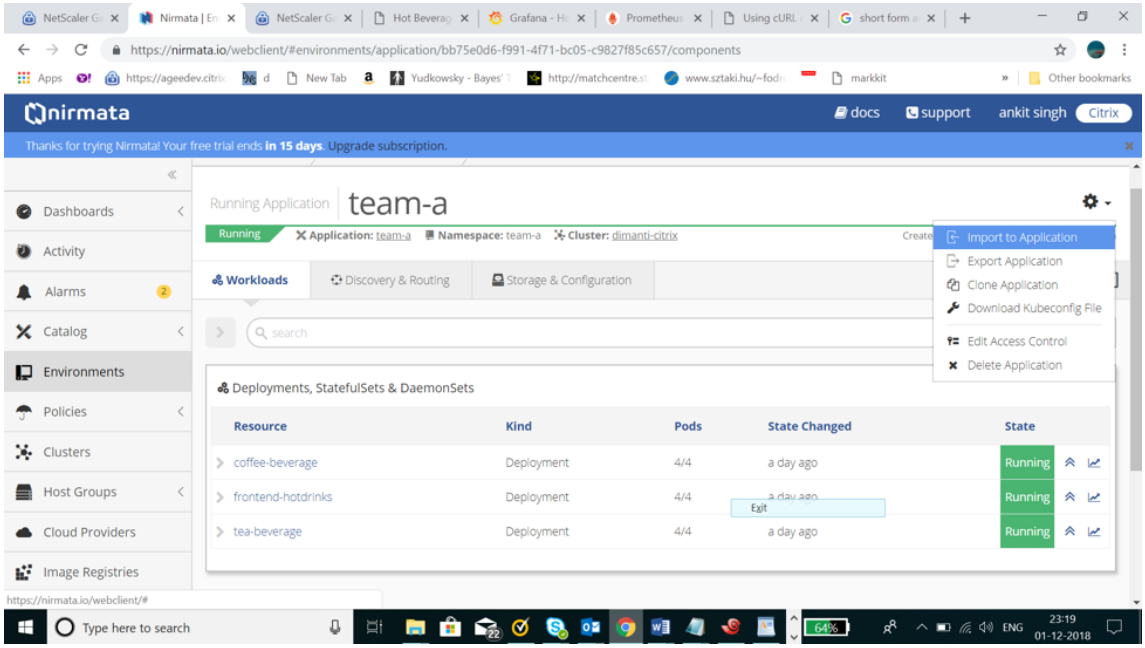
5. 使用 Nirmata 运行应用程序。

   team-hotdrink 应用程序：

   a) 转到环境选项卡并选择正确的环境：team-hotdrink。

   b) 在 team-hotdrink 环境中，使用 team-hotdrink 运行名称运行 team-hotddrink 应用程序。从目录应用程序中选择 team-hotdrink。

   c) 转到 team-hotdrink 应用程序。在屏幕右上角，单击“设置”，然后选择“导入到应用程序”。上传 hotdrink-secret.yaml。



team-colddrink 应用程序：

a) 转到环境选项卡并选择正确的环境：team-colddrink。
b) 在 team-colddrink 环境中，使用运行名称 team-coldddrink 运行 team-colddrink 应用程序。从目录应用程序中选择 team-hotdrink。
c) 转到 team-colddrink 应用程序。在屏幕右上角，单击“设置”，然后选择“导入到应用程序”。上传 colddrink-secret.yaml。

team-redis 应用程序：

a) 转到环境选项卡并选择正确的环境：team-redis。
b) 在 team-colddrink 环境中，使用 team-redis 运行名称运行应用程序。从目录应用程序中选择 team-redis。
   - 在 team-redis 环境中，使用 team-redis 运行名称运行应用程序。

在 **VPX** 上显示第 **2** 层 **CPX** 的命令

第 1 层 VPX 应在发送至第 2 层 CPX 时执行 SSL 加密/解密并插入 X-forward 标头。应手动执行第 1 层配置。可以在服务组中使用 -cip ENABLED 插入 X-forward 标头。打开 config.txt。

创建一个服务器：

在 Citrix ADC 中上载 certkey：wild.com-key.pem， wild.com-cert.pem

```
add cs vserver frontent_grafana HTTP <CS_VSERVER_IP> 80 -cltTimeout 180
<!--NeedCopy-->
```

在第 **1** 层 **VPX** 上公开 **www.hotdrinks.com**、**www.colddrinks.com**、**www.guestbook.com**：

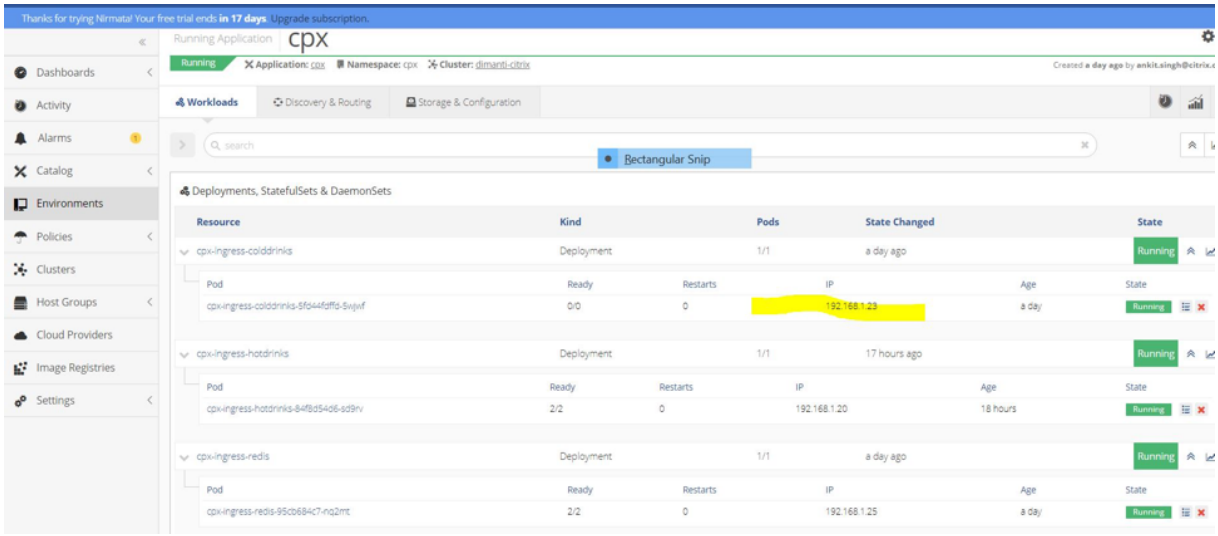
```
add serviceGroup team_hotdrink_cpx SSL -cip ENABLED
add serviceGroup team_colddrink_cpx SSL -cip ENABLED
add serviceGroup team_redis_cpx HTTP
add ssl certKey cert -cert ”wild-hotdrink.com-cert.pem” -key ”wild-
    hotdrink.com-key.pem”
add lb vserver team_hotdrink_cpx HTTP 0.0.0.0 0
add lb vserver team_colddrink_cpx HTTP 0.0.0.0 0
add lb vserver team_redis_cpx HTTP 0.0.0.0 0
add cs vserver frontent SSL 10.106.73.218 443
add cs action team_hotdrink_cpx -targetLBVserver team_hotdrink_cpx
add cs action team_colddrink_cpx -targetLBVserver team_colddrink_cpx
add cs action team_redis_cpx -targetLBVserver team_redis_cpx
```



```
12 add cs policy team_hotdrink_cpx -rule "HTTP.REQ.HOSTNAME.SERVER.EQ("www
   .hotdrinks.com") && HTTP.REQ.URL.PATH.STARTSWITH("/")" -action
   team_hotdrink_cpx
13 add cs policy team_colddrink_cpx -rule "HTTP.REQ.HOSTNAME.SERVER.EQ("
   www.colddrinks.com") && HTTP.REQ.URL.PATH.STARTSWITH("/")" -action
   team_colddrink_cpx
14 add cs policy team_redis_cpx -rule "HTTP.REQ.HOSTNAME.SERVER.EQ("www.
   guestbook.com") && HTTP.REQ.URL.PATH.STARTSWITH("/")" -action
   team_redis_cpx
15 bind lb vserver team_hotdrink_cpx team_hotdrink_cpx
16 bind lb vserver team_colddrink_cpx team_colddrink_cpx
17 bind lb vserver team_redis_cpx team_redis_cpx
18 bind cs vserver frontent -policyName team_hotdrink_cpx -priority 10
19 bind cs vserver frontent -policyName team_colddrink_cpx -priority 20
20 bind cs vserver frontent -policyName team_redis_cpx -priority 30
21 bind serviceGroup team_hotdrink_cpx 10.1.3.8 443
22 bind serviceGroup team_colddrink_cpx 10.1.2.52 443
23 bind serviceGroup team_redis_cpx 10.1.2.53 80
24 bind ssl vserver frontent -certkeyName cert
25 <!--NeedCopy-->
```

将 **IP** 地址更新到 **CPX** 窗格 **IP** 的服务组：

```
1 root@ubuntu-211:~/demo-nimata/final/final-v1# kubectl get pods -n cpx -
   o wide
2 NAME                                        READY     STATUS    RESTARTS
      AGE       IP              NODE
3 cpx-ingress-colddrinks-5bd94bff8b-7prdl   1/1       Running   0
             2h        10.1.3.8   ubuntu-221
4 cpx-ingress-hotdrinks-7c99b59f88-5kclv    1/1       Running   0
             2h        10.1.2.52  ubuntu-213
5 cpx-ingress-redis-7bd6789d7f-szbv7        1/1       Running   0
             2h        10.1.2.53  ubuntu-213
6 <!--NeedCopy-->
```

- 要访问 **www.hotdrinks.com**、**www.colddrinks.com**、**www.guestbook.com**，主机文件（要访问网页的计算机）应附加以下值：

  ```
  <CS_VSERVER_IP> www.hotdrinks.com

  <CS_VSERVER_IP> www.colddrinks.com

  <CS_VSERVER_IP> www.guestbook.com
  ```

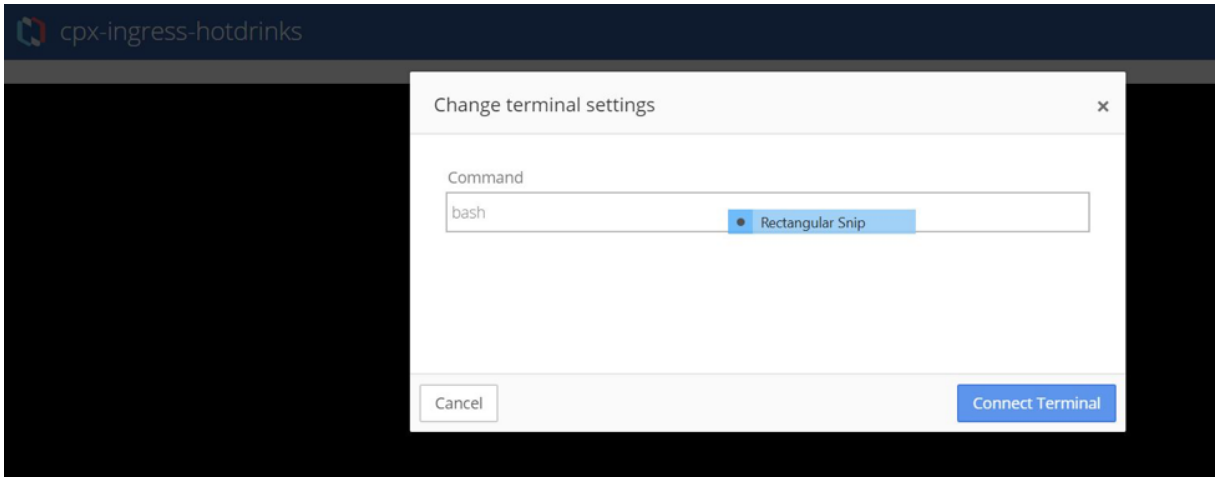
  完成此操作后，您可以通过访问 **www.hotdrinks.com, www.colddrinks.com, www.guestbook.com** 访问应用程序

验证第 **2** 层 **CPX** 配置

要验证 CPX 配置，请转到 CPX 环境。选择正在运行的 CPX 应用程序。

选择 `cpx-ingress-hotdrinks` 部署，然后单击 `cpx-ingress-hotdrinks-xxxx-xxxx` pod。

在下一页上，转到正在运行的容器并通过键入 “bash” 命令为 `cpx-ingress-hotdrinks` 启动终端。



连接终端后，通过 `cli_script.sh` 使用常规 NetScaler 命令验证配置。

- `cli_script.sh "sh cs vs"`
- `cli_script.sh "sh lb vs"`
- `cli_script.sh "sh servicegroup"`

可以以相同的方式为 `team-colddrink` 和 `team-mongodb` 的其他 CPX 部署完成验证。

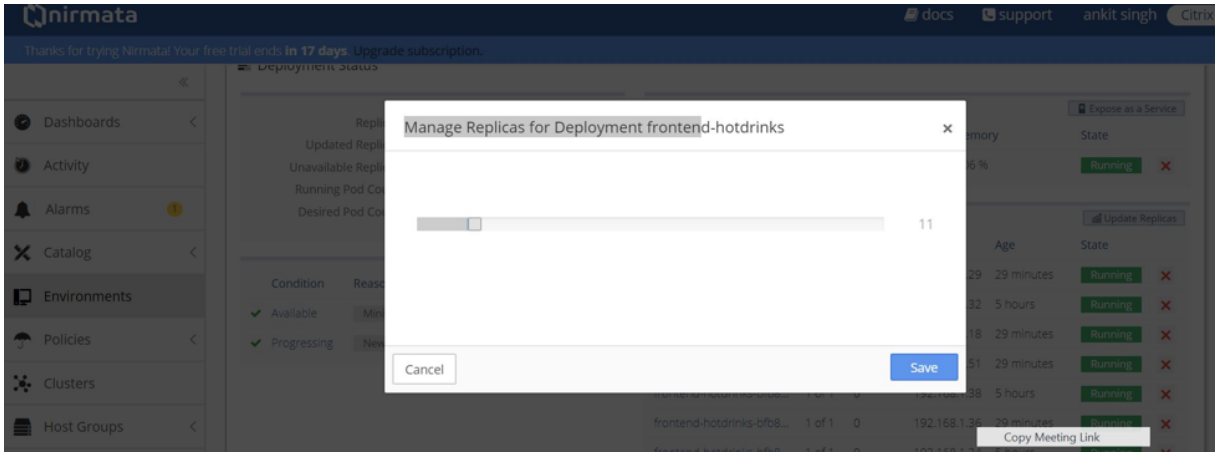
执行向上扩展/缩小扩展

要执行向上/缩小扩展，请执行以下操作：

1. 转至 `team-hotdrink` 环境。选择 `team-hotdrink` 运行的应用程序。
2. 单击 `frontend-hotdrinks` 部署。
3. 在下一页上，单击更新复制副本。将其增加到 10。

参考：验证第 **2** 层 **CPX** 配置以检查 **CPX** 中的配置（部署：**CPX** 摄入**-**热饮）。

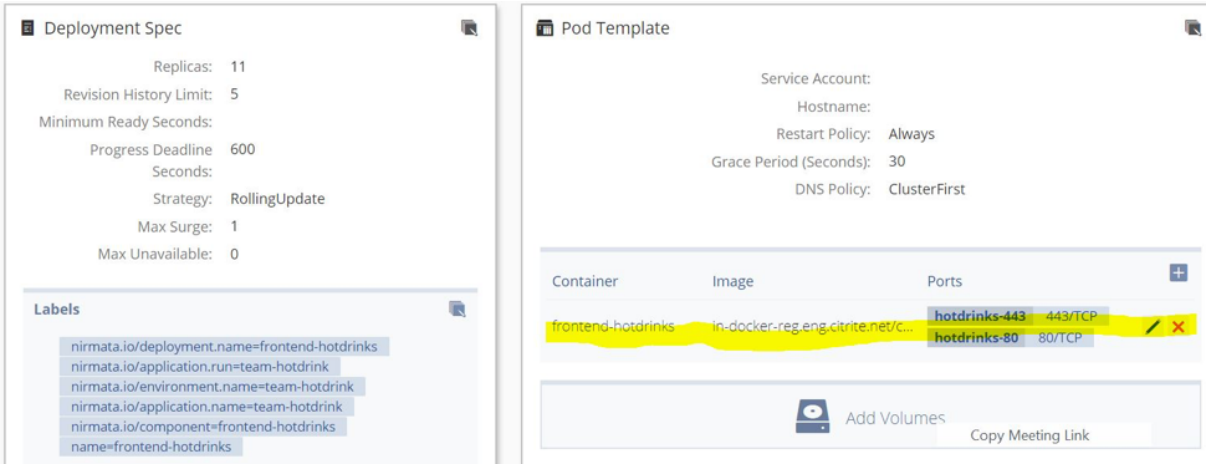
1. 转到 CPX 环境。选择正在运行的 CPX 应用程序。
2. 单击 `cpx-ingress-hotdrinks` 部署。
3. 单击 `cpx-ingress-hotdrinks-xxxx-xxxx` 窗格。
4. 在下一页上，转到正在运行的容器并通过键入 “bash” 命令为 `cpx-ingress-hotdrinks` 启动终端。
5. `cli_script.sh "sh servicegroup < servicegroup name >"`。



执行滚动更新

要执行滚动更新，请执行以下操作：

1. 转至 `team-hotdrink` 环境。选择 `team-hotdrink` 运行的应用程序。
2. 部署正面热饮。
3. 在下一页，转到 **Pod** 模板。
4. 将图像更新为：*quay.io/citrix/hotdrinks-v2: latest*。
5. 让更新完成。
6. 再次访问应用程序。滚动更新后，新页面应随附更新的图像。



部署 **Prometheus**

NetScaler 衡量指标导出程序、Prometheus 和 Grafana 用于自动检测和收集入口的 CPX 的衡量指标。

部署 **Prometheus** 的步骤：

创建运行 CPX 和应用程序的环境：

1. 转到环境选项卡。

2. 单击添加环境。
3. 创建运行导出程序、Prometheus 和 Grafana 的环境。
    - 创建环境：监视。

使用 Nirmata 上传 `.yaml` 文件：

1. 转到目录选项卡。
2. 单击添加应用程序。
3. 单击添加添加应用程序。
    - 添加应用程序：monitoring (monitoring.yaml)。

运行 Prometheus 应用程序：

1. 转到“环境”选项卡并选择正确的环境：**monitoring** 。
2. 使用名称 **monitoring** 单击运行应用程序。
3. 这将部署导出程序、Prometheus 和 Grafana 豆荚，并开始收集指标。
4. 现在 Prometheus 和 Grafana 需要通过 VPX 曝光。

VPX 上用于显示 **Prometheus** 和 **Grafana** 的命令：

创建 CSV 服务器：

```
add cs vserver frontent_grafana HTTP <CS_VSERVER_IP> 80 -cltTimeout 180
<!--NeedCopy-->
```

显示 Prometheus：

```
add serviceGroup prometheus HTTP
add lb vserver prometheus HTTP 0.0.0.0 0
add cs action prometheus -targetLBVserver prometheus
add cs policy prometheus -rule "HTTP.REQ.HOSTNAME.SERVER.EQ("www.
    prometheus.com") && HTTP.REQ.URL.PATH.STARTSWITH("/")" -action
    prometheus
bind lb vserver prometheus prometheus
bind cs vserver frontent_grafana -policyName prometheus -priority 20
bind serviceGroup prometheus <PROMETHEUS_POD_IP> 9090
<!--NeedCopy-->
```

> 注意：
>
> Get the prometheus-k8s-0 pod IP using“kubectl get pods -n monitoring -o wide”

显示 Grafana：


```
add serviceGroup grafana HTTP
add lb vserver grafana HTTP 0.0.0.0 0
add cs action grafana -targetLBVserver grafana
add cs policy grafana -rule "HTTP.REQ.HOSTNAME.SERVER.EQ("www.grafana.com") && HTTP.REQ.URL.PATH.STARTSWITH("/")" -action grafana
bind lb vserver grafana grafana
bind cs vserver frontent_grafana -policyName grafana -priority 10
bind serviceGroup grafana <GRAFANA_POD_IP>  3000
<!--NeedCopy-->
```

> 注意：
>
> Get the grafana-xxxx-xxx pod IP using `kubectl get pods -n monitoring -o wide`

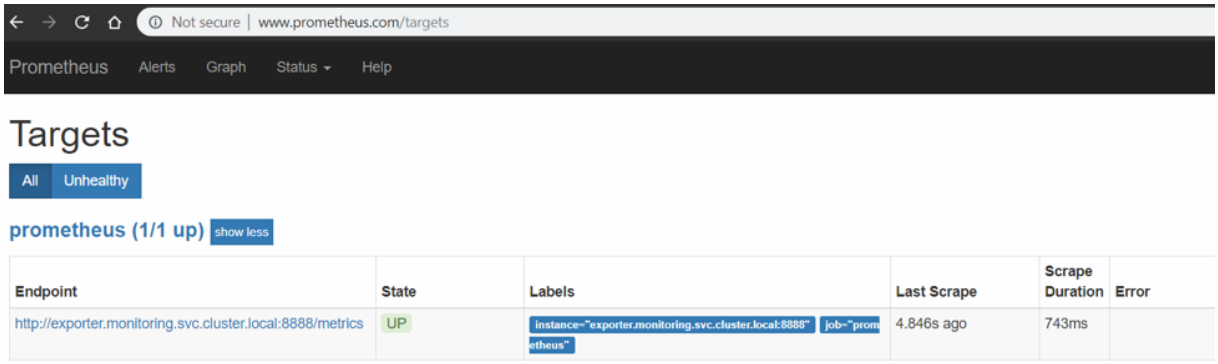
- 现在，Prometheus 和 Grafana 页面已经公开通过 VPX 的 cs vserver 访问。
- 要访问 Prometheus 和 Grafana，主机文件（要访问页面的机器的）应该附加以下值：

  ```
  <CS_VSERVER_IP> www.grafana.com
  <CS_VSERVER_IP> www.prometheus.com
  ```

- 完成此操作后，请访问 **www.prometheus.com** 访问 Prometheus。通过访问 **www.grafana.com** 访问 Grafana。

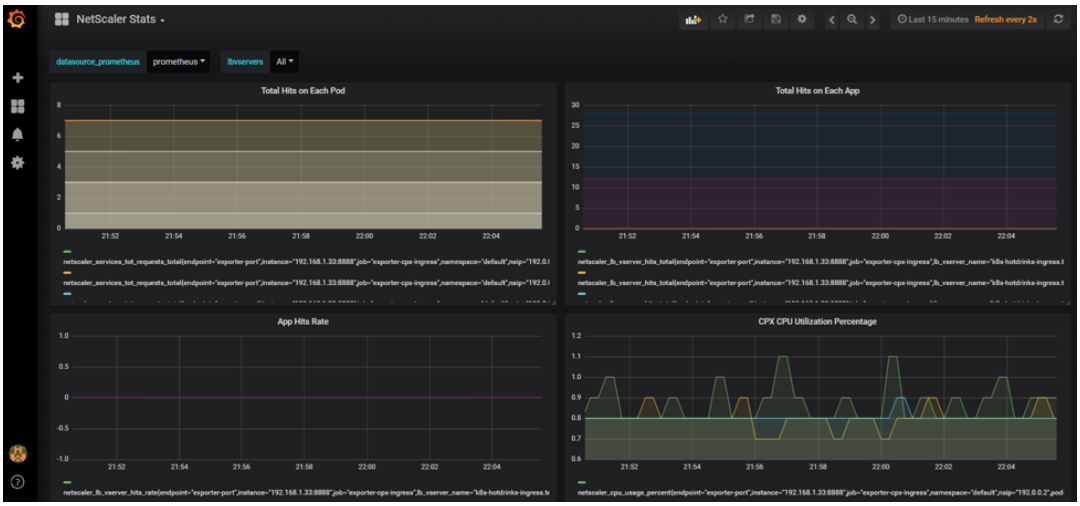
可视化指标：

- 为确保 Prometheus 检测到导出程序，请访问 **www.prometheus.com/targets**。它应该包含监视 CPX 和 VPX 设备的所有导出程序的列表。确保所有 Exporter 都处于 **UP** 状态。请参见以下示例：



- 现在，您可以使用 Grafana 绘制正在收集的值。要做到这一点：
  1. 转至 **www.grafana.com**。确保在主机文件中添加了适当的条目。
  2. 使用默认用户名 **admin** 和密码 **admin** 登录。
  3. 登录后，单击主控板中的“添加数据源”。
  4. 选择 **Prometheus** 选项。

5. 提供/更改以下详细信息：
   - 名称：Prometheus（全部小写）。
   - URL：`http://prometheus:9090`。
   - 将剩余条目保留为默认值。
6. 单击保存并测试。等待几秒钟，直到屏幕底部显示“数据源正常工作”消息。
7. 单击左侧面板上的+ 图标，导入预先设计的 Grafana 模板。选择“导入”。
8. 单击“上传 **json**”按钮，然后选择 **sample_grafana_dashboard.json** 文件（保留名称、文件夹和唯一标识符不变）。
9. 从 **Prometheus** 下拉菜单中选择 Prometheus，然后单击 导入。
10. 这将上传类似于下图的控制板：



许可证和性能测试

运行 **CPX** 进行性能和许可。

CPX 核心数和许可证服务器详细信息在以下环境变量中给出。

用于选择核心数量的环境变量

- 名称**:“CPX_CORES”**
- 值**:“3”**

用于选择许可证服务器的环境变量

- 名称**:“LS_IP”**

- 值：**“X.X.X”**

  Diamanti annotations:
  `diamanti.com/endpoint0: '{ ”network”:”lab-network”,”perfTier”:”high” }`

  通过上面设置正确的 **IP** 指向校正许可证服务器。

  1. 在文件中添加上述环境变量以及 Diamanti 特定的注释。`cpx-perf.yaml`
  2. 转到环境选项卡并创建 `cpx-perf` 环境。

  使用 **Nirmata** 上传 **YAML** 应用程序。

  1. 转到目录选项卡。
  2. 单击添加应用程序。
  3. 单击添加添加应用程序：`cpx-perf.yaml`。应用程序名称：`cpx-perf`。

  运行 **CPX**:

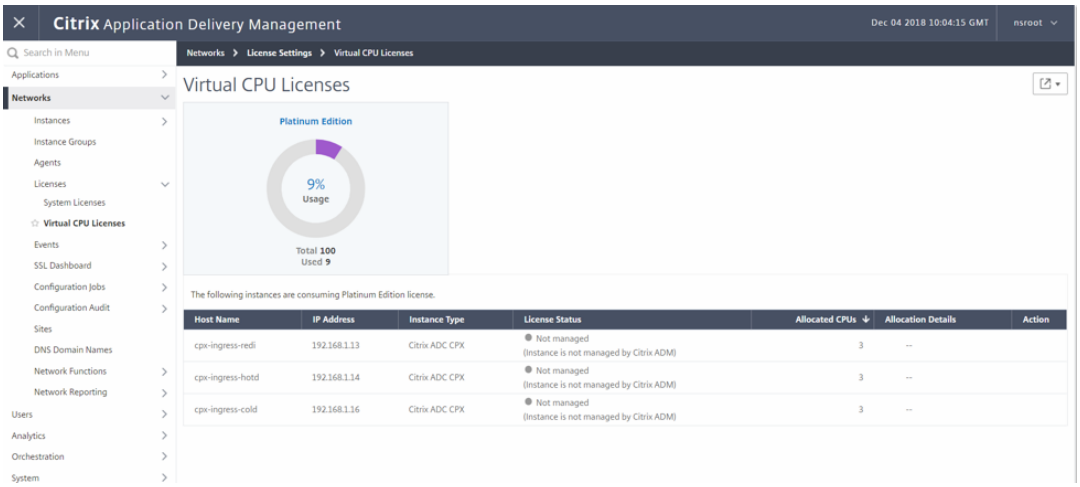
  1. 转到环境选项卡并选择 `cpx-perf` 环境。
  2. 在 `cpx-perf` 环境中，运行 `cpx-svcacct` 应用程序。
  3. 在 `cpx-perf` 环境中，运行 `cpx-perf` 应用程序。
  4. 运行应用程序后，转到 `cpx-perf` 应用程序。
  5. 在部署 **> StatefulSets** 和 **DaemonSets** 选项卡下，单击 `cpx-ingress-perf` 部署。在下一页上，编辑 Pod 模板。在服务帐户中输入 **CPX**。
  6. 验证许可证是否正常工作以及是否正在 Citrix ADM 中执行许可证签出。
     - 要在 CPX 上进行验证，请执行以下步骤：
       * kubectl get pods -n cpx
       * kubectl exec -it <CPX_POD_NAME> -n cpx bash
       * cli_script.sh ‘sh licenseserver’
       * cli_script.sh ‘sh capacity’
     - 查看类似的输出：

```
root@cpx-ingress-colddrinks-66f4d75f76-kzf8w:/# cli_script.sh
    'sh licenseserver'
exec: sh licenseserver
1)  ServerName:  10.217.212.228Port:  27000           Status:
      1             Grace:  0              Gptimeleft:  0
Done
root@cpx-ingress-colddrinks-66f4d75f76-kzf8w:/# cli_script.sh
    'sh capacity'
exec: sh capacity
```



```
7     Actualbandwidth:  10000 VcpuCount:  3          Edition:
          Platinum      Unit:  Mbps             Maxbandwidth:
          10000    Minbandwidth:  20       Instancecount:  0
8  Done
9  <!--NeedCopy-->
```

- 要在 ADM 上进行验证，请转到许可证服务器并导航到网络 **>** 许可证 **>** 虚拟 **CPU** 许可证。
- 在这里，您应该看到许可 CPX 以及核心计数。



批注表

注释	可能的价值	说明	默认值（如果有）
kubernetes.io/ingress.c	入口类名称	这是一种将特定入口资源与 Ingress Controller 关联的方法。例如，`kubernetes.io/ingress.class:"Citrix"`	配置所有入口

注释	可能的价值	说明	默认值（如果有）
网站/安全后端	使用.json 格式，列出用于安全后端的服务	如果希望 Citrix ADC 使用安全的 HTTPS 连接连接您的应用程序，请使用 **True**。如果希望 Citrix ADC 使用不安全的 HTTP 连接来连接您的应用程序，请使用 **False**。例如，`ingress.citrix.com/secure_backend: { 'app1':"True", 'app2':"False", 'app3':"True"}`	“False”
ingress.citrix.com/lbvse	在 JSON 表单中，lbvserver 的设置	它提供了智能注释功能。使用此功能，高级用户（了解 NetScaler LB 虚拟服务器和服务组选项）可以直接应用这些选项。值必须采用.json 格式。对于入口中的每个后端应用程序，请提供一个密钥值对。密钥名称应与相应的 CLI 名称匹配。例如，`ingress.citrix.com/lbvserver: ' { "app-1":{ "lbmethod":"ROUNDROBIN" } } '`	默认值

Citrix ADC SSL 配置文件验证的参考设计

May 20, 2020

概述

Citrix ADC 摘要

Citrix ADC 是一款一体化应用程序交付 Controller，可使应用程序运行速度提高五倍，降低应用程序拥有成本，优化用户体验，并通过使用以下方式确保应用程序始终可用：

- 高级 L4-7 负载平衡和流量管理
- 经验证的应用程序加速，例如 HTTP 压缩和缓存
- 集成的应用程序防火墙可确保应用程序安全
- 服务器卸载以显著降低成本并整合服务器

作为服务和应用交付领域无可争议的领导者，Citrix ADC 部署在全球数千个网络中，以优化、保护和控制所有企业和云服务的交付。Citrix ADC 直接部署在 Web 和数据库服务器前，将高速负载平衡和内容交换、http 压缩、内容缓存、SSL 加速、应用程序流可见性和强大的应用程序防火墙集成到一个易于使用的集成平台中。通过端到端监视，将网络数据转换为可操作的商业智能，大大简化了会议 SLA。Citrix ADC 允许使用简单的声明性策略引擎来定义和管理策略，无需编程专业知识。

概述 Citrix ADC SSL 配置文件

您可以使用 SSL 配置文件指定 Citrix ADC 如何处理 SSL 通信。配置文件是 SSL 实体（如虚拟服务器、服务和服务组）的 SSL 参数设置的集合，并提供了易于配置和灵活性。您不限于只配置一组全局参数。您可以创建多个全局参数集（配置文件），并将不同的集分配给不同的 SSL 实体。SSL 配置文件分为两类：

- 前端配置文件，包含适用于前端实体的参数。也就是说，它们应用于接收来自客户端的请求的实体。
- 后端配置文件，包含适用于后端实体的参数。也就是说，它们应用于向服务器发送客户端请求的实体。

与 TCP 或 HTTP 配置文件不同，SSL 配置文件是可选的。启用 SSL 配置文件（全局参数）后，所有 SSL 端点都会继承默认配置文件。同一个配置文件可以跨多个实体重复使用。如果实体没有附加配置文件，则应用在全局级别设置的值。对于动态学习的服务，应用当前的全局值。

与需要在单个 SSL 终端上配置 SSL 参数、密码和 ECC 曲线的替代方式相比，Citrix ADC 上的 SSL 配置文件通过充当所有相关终端的单点 SSL 配置来简化配置管理。此外，使用 SSL 配置文件解决了密码重新排序和重新排序密码时停机等配置问题。

SSL 配置文件有助于在传统上无法设置这些参数和绑定的 SSL 端点上设置所需的 SSL 参数和密码绑定。SSL 配置文件也可以在安全监视器上设置。

下表列出了作为每个配置文件一部分的参数：

前端配置文件	后端配置文件
cipherRedirect、cipherURL	denySSLReneg
clearTextPort*	encryptTriggerPktCount
clientAuth、clientCert	nonFipsCiphers

前端配置文件	后端配置文件
denySSLReneg	pushEncTrigger
dh、dhFile、dhCount	pushEncTriggerTimeout
dropReqWithNoHostHeader	pushFlag
encryptTriggerPktCount	quantumSize
eRSA、eRSACount	serverAuth
insertionEncoding	commonName
nonFipsCiphers	sessReuse, sessTimeout
pushEncTrigger	SNIEnable
pushEncTriggerTimeout	ssl3
pushFlag	sslTriggerTimeout
quantumSize	strictCAChecks
redirectPortRewrite	TLS 1.0、TLS 1.1、TLS 1.2
sendCloseNotify	
sessReuse, sessTimeout	
SNIEnable	
ssl3	
sslRedirect	
sslTriggerTimeout	
strictCAChecks	
tls1、tls11、tls12	

* clearTextPort 参数仅适用于 SSL 虚拟服务器。

如果您尝试设置不是配置文件一部分的参数（例如，如果您尝试在后端配置文件中设置 clientAuth 参数），则会显示一条错误消息。

某些 SSL 参数（如 CRL 内存大小、OCSP 高速缓存大小、UndefAction 控制以及 UndefAction 数据）不属于上述任何配置文件，因为这些参数独立于实体。这些参数出现在 “流量管理**”>“SSL”>“**高级 **SSL** 设置**”** 中。

SSL 配置文件支持以下操作：

- 添加-在 Citrix ADC 上创建 SSL 配置文件。指定配置文件是前端还是后端。前端为默认值。
- 集 (Set)-修改现有配置文件的设置。

- Unset (Unset)-将指定参数设置为其默认值。如果未指定任何参数，则会显示一条错误消息。如果取消对实体的配置文件的设置，则该配置文件将取消与实体的绑定。
- 删除-删除配置文件。无法删除任何实体正在使用的配置文件。清除配置将删除所有实体。因此，配置文件也会被删除。
- 绑定-将配置文件绑定到虚拟服务器。
- 取消绑定-从虚拟服务器取消绑定配置文件。
- 显示-显示 Citrix ADC 上可用的所有配置文件。如果指定了配置文件名称，则会显示该配置文件的详细信息。如果指定了实体，则会显示与该实体关联的配置文件。

SSL 配置文件用例

SSL 默认配置文件

Citrix ADC 设备附带两个内置的默认配置文件 —

1. ns_default_ssl_profile_frontend — 所有 SSL 类型虚拟服务器和内部服务的默认前端配置文件。
2. ns_default_ssl_profile_backend — SSL 类型服务、服务组和安全监视器的默认后端配置文件。

创建的任何新端点都会绑定相应的默认 SSL 配置文件。

可以更改默认 SSL 配置文件的 SSL 参数和密码。这可确保客户可以在相应端点引用的一个点更改设置和绑定。

> 重要提示：
>
> 在升级软件并启用默认配置文件之前保存您的配置。
>
> 将软件升级到支持增强型配置文件基础结构的版本，然后启用默认配置文件。根据您的特定部署，您可以采取两种方法之一。如果您的部署具有跨端点的通用 SSL 配置，请参阅用例 1。如果您的部署具有大型 SSL 配置，并且 SSL 参数和密码在端点之间不常见，请参阅用例 2。
>
> 升级软件后，如果启用配置文件，则无法撤消更改。也就是说，无法禁用配置文件。因此，撤消更改的唯一方法是使用旧配置重新启动。
>
> 注意：单个操作（启用默认配置文件或 set ssl parameter -defaultProfile ENABLED）启用（绑定）默认前端配置文件和默认后端配置文件。
>
> 注意：默认的 SSL 配置文件现在可用于从 v11.1 开始的群集

要使用 Citrix ADC 命令行保存配置，请在命令提示符处键入：

```
1 >save config
2
3 >shell
```

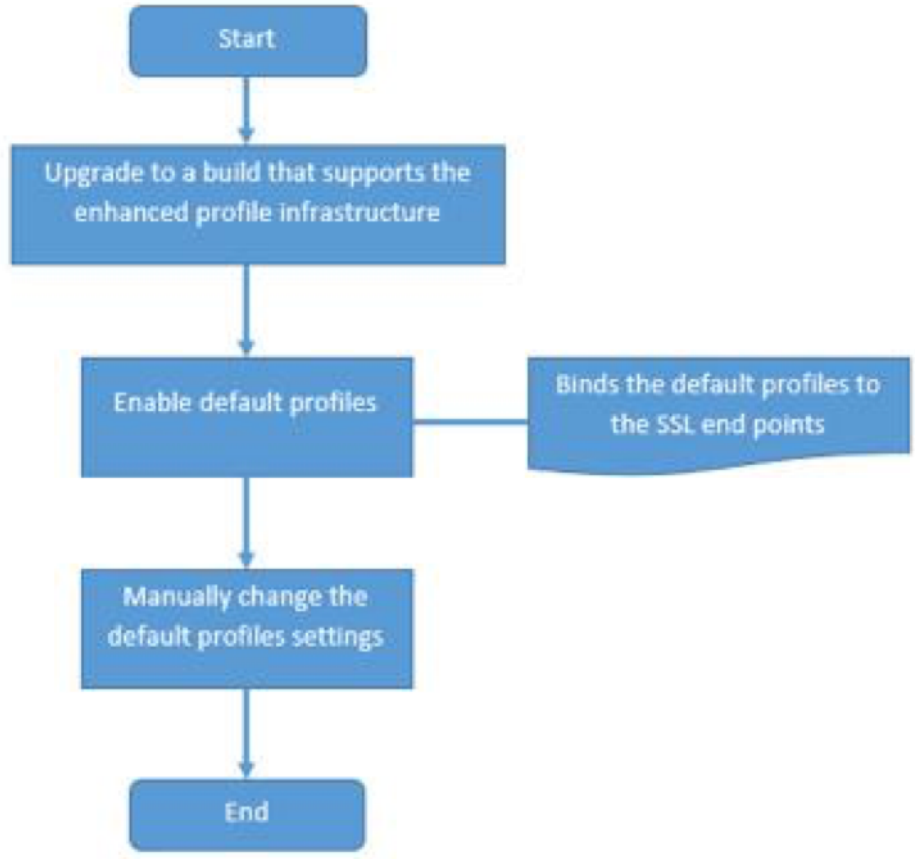


```
4
5 root@ns# cd /nsconfig
6
7 root@ns# cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber
     >
8 <!--NeedCopy-->
```

用例 **1**

启用默认配置文件后，它们将绑定到所有 SSL 端点。默认配置文件是可编辑的。如果您的部署使用大多数默认设置，并且只更改了少数参数，则可以编辑默认配置文件。这些更改会立即反映在所有端点上。

下面的流程图说明了您必须执行的步骤：



1. 有关升级软件的信息，请参阅升级系统软件。
2. 使用 Citrix ADC 命令行或 GUI 启用默认配置文件。
   - 在命令行中，键入：set ssl parameter -defaultProfile ENABLED
   - 如果您希望使用 GUI，请导航到流量管理 **> SSL >** 更改高级 **SSL** 设置，向下滚动，然后选择启用默认配置文件。

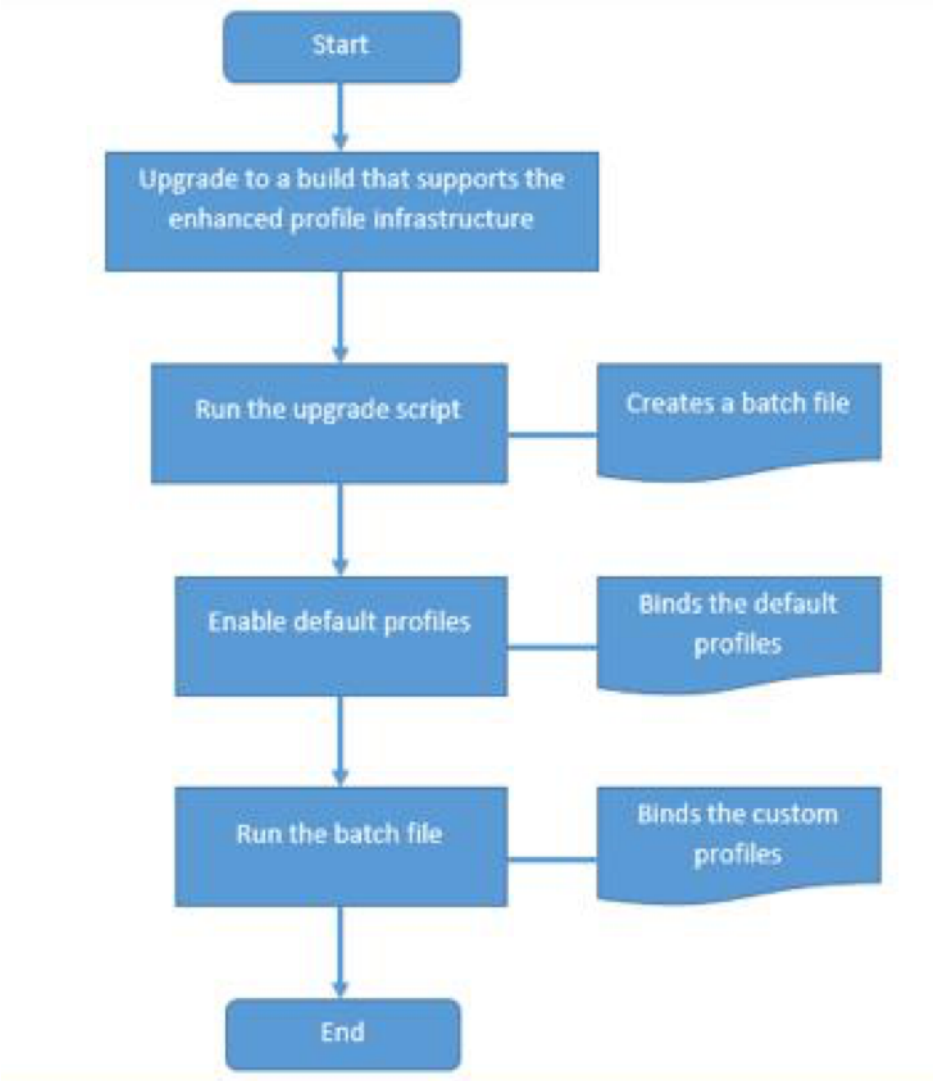
3.（可选）手动更改默认配置文件中的任何设置。

- 在命令行中，键入：`set ssl profile <name>`，后跟要修改的参数。
- 如果您希望使用 GUI，请导航到 “系统”>“配置文件”。在 SSL 配置文件中，选择一个配置文件，然后单击 编辑。

用例 **2**

如果您的部署使用了大多数 SSL 实体的特定设置，则可以运行脚本，为每个终点自动创建自定义配置文件并将其绑定到终点。使用本节中详细介绍的过程保留部署中所有 SSL 端点的 SSL 设置。升级软件后，下载并运行迁移脚本以捕获特定于 SSL 的更改。运行此脚本的输出是批处理文件。启用默认配置文件，然后应用批处理文件中的命令。有关升级后 SSL 配置迁移示例，请参阅附录。

下面的流程图说明了您必须执行的步骤：



1. 有关升级软件的信息，请参阅升级系统软件。

2. 下载并运行脚本以捕获特定于 SSL 的更改。除了其他迁移活动外，该脚本还会分析旧的 ns.conf 文件，并将任何特殊设置（默认设置除外）从 SSL 端点配置移动到自定义配置文件。您必须在升级后启用默认配置文件，才能应用配置更改。

   要下载脚本，请登录到 https://www.citrix.com/。在“下 载”选项卡上，选择 Citrix ADC，然后选择版本（例如，版本 12.0）。在版本中，在固件中，选择一个版本。SSL 默认配置文件脚本在其他组件中可用。

   > 注意：运行迁移脚本时，您可以选择自动生成配置文件名称，也可以以交互方式提示用户输入配置文件名称。迁移脚本会检查以下内容并相应地创建专业文件。
   >
   > - 具有默认设置和类似密码和密码组设置的端点：脚本创建一个配置文件。
   > - 具有默认设置、密码组不同或密码组不同优先级的端点：在每种情况下，脚本都会创建用户定义的密码组，将其绑定到配置文件，并将每个配置文件绑定到相应的端点。
   > - 具有默认设置和默认密码的端点：默认配置文件绑定到终点。

```
要运行脚本，请在命令提示符处键入：
```

```
./default_profile_script /nsconfig/ns.conf -b > <output file name
    >`
<!--NeedCopy-->
```

```
必须从存储脚本的文件夹中运行此命令。
```

3. 使用 Citrix ADC 命令行或 GUI 启用默认配置文件。

   - 在命令行中，键入：`set ssl parameter -defaultProfile ENABLED`
   - 如果您希望使用 GUI，请导航到流量管理 **> SSL >** 更改高级 **SSL** 设置，向下滚动，然后选择启用默认配置文件。

自定义 **SSL** 配置文件

除了默认的 SSL 配置文件，客户可以为特定用例创建自定义的前端和后端 SSL 配置文件。在某些情况下，不同的应用程序需要不同的密码和 SSL 参数。在这些情况下，客户可以创建新的配置文件并将其绑定到终端。

可在系统中创建的自定义配置文件的数量没有上限。

有关如何启用 SSL 配置文件等信息，请访问 SSL 配置文件 文档。

SSL 前端配置文件

前端 SSL 配置文件与 SSL 类型的虚拟服务器和内部服务相关。前端配置文件适用于负载平衡虚拟服务器、内容交换虚拟服务器、AAA-TM 虚拟服务器和网关 VPN 虚拟服务器类别中的所有 SSL 类型虚拟服务器。

以下类型的虚拟服务器支持前端配置文件 — SSL、SSL_TCP、SIP_SSL、SSL_FIX 和 SSL_DIAMETER。

所有内部服务都支持前端配置文件。

SSL 后端配置文件

后端配置文件与 SSL 类型的服务、服务组和安全监视器相关。以下类型的服务和服务组支持后端配置文件 — SSL、SSL_TCP、SIP_SSL、SSL_FIX、SSL_DIAMETER。

某些监视器可以配置为通过安全连接检查后端服务器的运行状况。SSL 配置文件可以绑定到此类监视器以配置 SSL 参数和密码。这些监视器包括 HTTP、HTTP-ECV、HTTP-INLINE、TCP 和 TCP-ECV。

Citrix ADC 和 Amazon Web Services 验证的参考设计

May 20, 2020

概述 Citrix Networking VPX

Citrix ADC 是一款一体化应用程序交付 Controller，可使应用程序运行速度提高五倍，降低应用程序拥有成本，优化用户体验，并通过使用以下方式确保应用程序始终可用：

- 高级第 4-7 层服务负载平衡和流量管理
- 经验证的应用程序加速，例如 HTTP 压缩和缓存
- 集成的应用程序防火墙可确保应用程序安全
- 服务器卸载以显著降低成本并整合服务器

作为服务和应用交付领域无可争议的领导者，Citrix ADC 部署在全球数千个网络中，以优化、保护和控制所有企业和云服务的交付。Citrix ADC 直接部署在 Web 和数据库服务器前，将高速负载平衡和内容交换、HTTP 压缩、内容缓存、SSL 加速、应用程序流可见性和强大的应用程序防火墙集成到一个易于使用的集成平台中。通过端到端监视，将网络数据转换为可操作的商业智能，大大简化了会议 SLA。Citrix ADC 允许使用简单的声明性策略引擎来定义和管理策略，无需编程专业知识。

概述 Amazon Web Services 中的 Citrix ADC

自版本 10.5–61.11 起，即可支持 Amazon Web Services (AWS) 中的 Citrix Networking VPX。Citrix Networking VPX 在 AWS 应用商店中作为 Amazon Machine Image (AMI) 提供。通过 AWS 上的 Citrix Networking VPX，客

户能够利用 AWS 云计算功能，并使用 Citrix ADC 负载平衡和流量管理功能满足业务需求。AWS 上的 Citrix ADC 支持物理 Citrix ADC 设备的所有流量管理功能。在 AWS 中运行的 Citrix ADC 实例可以作为独立实例或高可用性对进行部署。

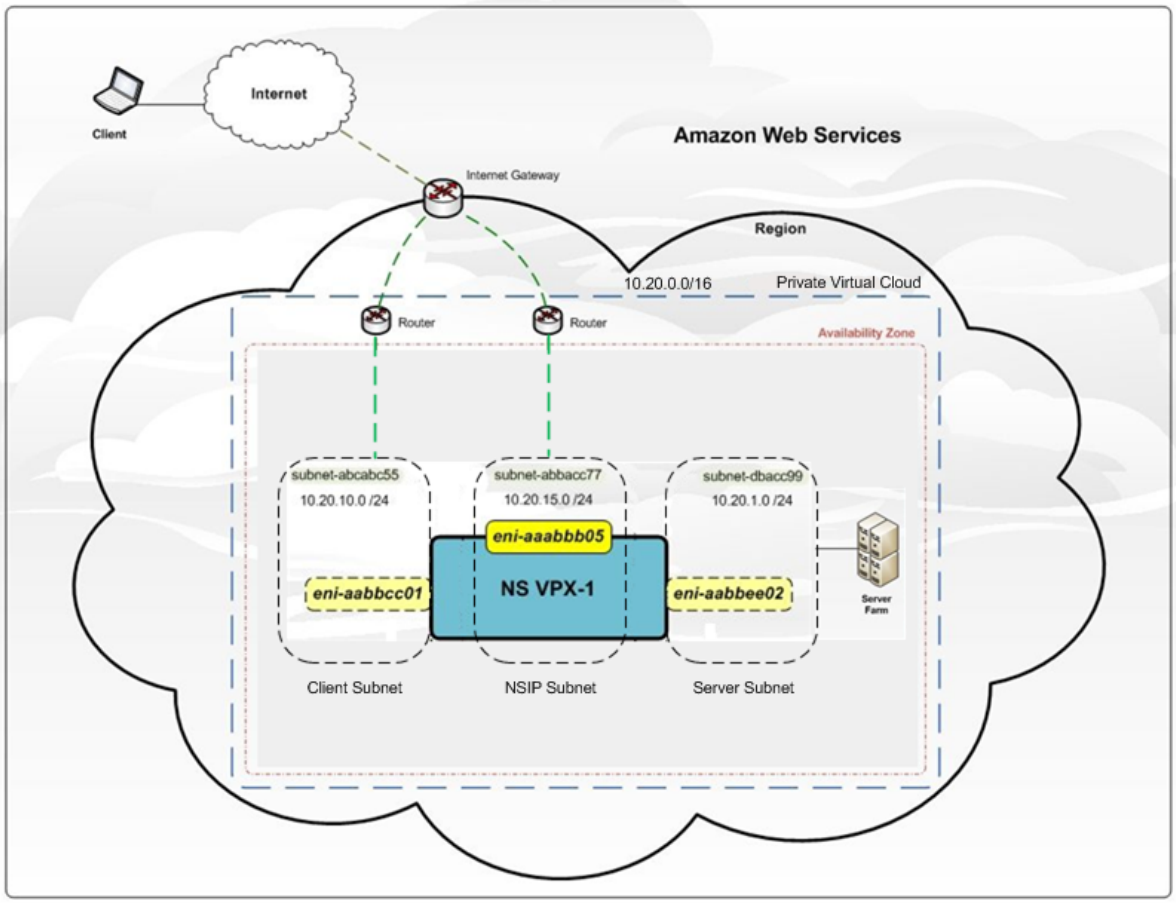
Citrix Networking VPX AMI 打包为在 AWS VPC 中启动的 EC2 实例。VPX AMI 实例最低需要 2 个虚拟 CPU 和 2 GB 内存。从 AWS VPC 内启动的 EC2 实例还可以提供多个接口，每个接口有多个 IP 地址，以及 VPX 配置所需的公共和专用 IP 地址。目前，在 AWS 上，VPX 只能在 VPC 内启动，因为每个 VPX 实例至少需要三个 IP 地址。（虽然 AWS 上的 VPX 可以通过一个或两个弹性网络接口来实现，但 Citrix 建议对 AWS 上的标准 VPX 安装使用三个网络接口）。AWS 目前只对在 AWS VPC 中运行的实例提供多 IP 功能。VPC 中的 VPX 实例可用于对 EC2 实例中运行的服务器实现负载平衡。

Amazon VPC 允许您创建和控制虚拟网络环境，包括您自己的 IP 地址范围、子网、路由表和网络网关。

> 注意：
>
> 默认情况下，每个 AWS 帐户的每个 AWS 区域最多可以创建 5 个 VPC 实例。您可以通过提交 Amazon 的申请表 请求更高的 VPC 限制。

在 AWS VPC 中启动了 Citrix Networking VPX 的 EC2 实例（AMI 映像）。

下图显示了典型的 AWS VPX 部署。



下图显示了具有 Citrix Networking VPX 部署的 AWS VPC 的简单拓扑。AWS VPC 包含：

1. 用于路由 VPC 内外部流量的单个 Internet 网关。
2. Internet 网关和 Internet 之间的网络连接。
3. 三个子网，分别用于管理、客户端和服务器。
4. Internet 网关与两个子网（管理和客户端）之间的网络连接。
5. 在 VPC 中部署的单个 Citrix Networking VPX。VPX 实例有三个弹性网络接口 (ENI)，分别连接到每个子网。

局限性与用法指南

- VPX 不支持群集功能。
- 要使高管局按预期工作，请将专用的 NATing 设备关联到管理接口或将 EIP 与 NSIP 关联。有关 NAT 的更多信息，请参阅 AWS 文档中的 NAT 实例。
- 应使用属于两个不同子网的 ENI 将数据流量与管理流量隔离。
- 管理 ENI 上只能使用 NSIP 地址。
- 如果使用 NAT 实例来实现安全性，而不是将 EIP 分配给 NSIP，需要更改恰当的 VPC 级别路由。有关更改 VPC 级路由的说明，请参阅 AWS 文档中的 场景 2：具有公用子网和专用子网的 VPC。
- 可将 VPX 实例从一个 EC2 实例类型移至另一个实例类型（例如，从 m3.large 移至 m3.xlarge）。
- 对于 AWS 上的 VPX 的存储方案，Citrix 建议选择 EBS，因为它具有持久性，并且即使从实例断开连接，仍然可用。
- 不支持将 ENI 动态添加到 VPX。您必须重新启动 VPX 实例才能应用更新。Citrix 建议您停止独立或高可用性实例，连接新的 ENI，然后重新启动实例。
- 您可以将多个 IP 地址分配给一个 ENI。每个 ENI 的最大 IP 地址数由 EC2 实例类型决定，请参阅 ENI 和 IP 地址的 EC2 支持。
- Citrix 建议避免在 Citrix Networking VPX 接口上使用启用和禁用接口命令。

由于 AWS 的限制，以下功能不受支持：

第 3 层限制：

- 动态路由
- IPV6

第 2 层限制：

- Gratuitous ARP(GARP)
- L2 模式
- 已标记的 VLAN
- 虚拟 MAC (VMAC)

支持的 **EC2** 实例

可以在以下任一 EC2 实例类型上启动 Citrix ADC AMI：

- m4.large
- m4.xlarge
- m4.2xlarge
- m4.4xlarge
- m4.10xlarge
- m3.large
- m3.xlarge
- m3.2xlarge

有关详细信息，请参阅Amazon EC2 实例。

ENI 支持

下表列出了 EC2 实例类型及相应的受支持的 ENI 数和每个 ENI 的专用 IP 地址数。

Instance Name（实例名称）	ENI 数量	每个 ENI 的专用 IP 地址
m4.large	2	10
m4.xlarge	4	15
m4.2xlarge	4	15
m4.4xlarge	8	30
m4.10xlarge	8	30
m3.large	3	10
m3.xlarge	4	15
m3.2xlarge	4	30

用例

与需要作为单独虚拟设备部署每项服务的替代解决方案相比，AWS 上的 Citrix ADC 将 L4 负载平衡、L7 流量管理、服务器卸载、应用程序加速、应用程序安全性和其他基本应用程序交付功能集成在一个 VPX 中实例，可通过 AWS 市场方便地访问。此外，所有内容都由单个策略框架进行管理，并使用用于管理本地 Citrix ADC 部署的功能强大的工具集进行管理。最终结果是，AWS 上的 Citrix ADC 支持了几个引人注目的使用案例，这些案例不仅支持当今企业的迫切需求，还支持从传统计算基础设施向企业云数据中心的持续演变。

Web 和 **Virtual Apps** 以及桌面应用程序的生产交付

积极采用 AWS 作为基础设施即服务 (IaaS) 产品，用于生产交付应用程序的企业现在可以使用世界上最大的网站和云服务提供商所使用的相同云网络平台前端应用程序。可以利用广泛的卸载、加速和安全功能来提高性能并降低成本。

XenDesktop 7.5 和 XenApp 7.5 已被重新设计为云就绪型解决方案，可将任何 Windows 应用程序或桌面交付到通过任何网络和任何设备提供的云服务中。通过今天部署这个扩展的应用和桌面交付平台，您可以利用任何虚拟基础体系结构或云管理平台。这使您能够充分利用云计算的自动化和调配功能。

混合云设计

遵循混合云战略的企业 IT 组织通过选择哪些应用程序和哪些使用方案最适合其私有云，哪些使用方案最适合公有云，从而使他们能够灵活、增长和转型以满足现代工作场所的需求，从而获得两方面的优势。

借助 AWS 上的 Citrix ADC，跨企业数据中心并扩展到 AWS 的混合云可以受益于同一云网络平台。Citrix ADC 大大简化了应用程序和工作负载在私有数据中心和 AWS 之间来回转换的过程。AWS 上的 Citrix ADC 可充分利用全套功能，包括使用 DataStream 的智能数据库负载平衡、使用 AppFlow® 实现前所未有的应用程序可视性，以及使用操作分析进行实时监视和响应。

业务连续性

希望将 AWS 用作灾难恢复和业务连续性计划的一部分的企业可以依靠在本地和 AWS 内部运行的 Citrix ADC 全球服务器负载平衡来持续监控企业数据中心和 AWS 环境的可用性和性能，从而确保用户始终发送到最佳位置。

在 Citrix ADC 设备上配置 GSLB 并启用衡量指标交换协议 (MEP) 时，这些设备将使用 DNS 基础结构将客户端连接到最符合您设置条件的数据中心。条件可以指定负载最少的数据中心、最近的数据中心、对来自客户端位置的请求响应最快的数据中心、这些指标的组合以及 SNMP 指标。设备会跟踪每个数据中心的位置、性能、负载和可用性，并使用这些因素选择要向其发送客户端请求的数据中心。GSLB 配置由配置中的每个设备上的一组 GSLB 实体组成。这些实体包括 GSLB 站点、GSLB 服务、GSLB 虚拟服务器、负载平衡和/或内容交换服务器以及 ADNS 服务。

开发和测试

企业在本地运行生产交付，但使用 AWS 进行开发和测试现在可以将 Citrix ADC 包括在其 AWS 测试环境中，这样可以更好地模拟其测试环境中的生产实施，从而加快生产速度。

在每个使用案例中，网络体系结构师还可以利用 Citrix CloudBridge（配置为独立实例或 Citrix ADC 铂金版实例的功能）来保护和优化一个或多个企业数据中心与 AWS 云之间的连接，从而加速数据传输/同步并最大限度地降低网络成本。

AWS 网络体系结构 — ENI 和 EIP

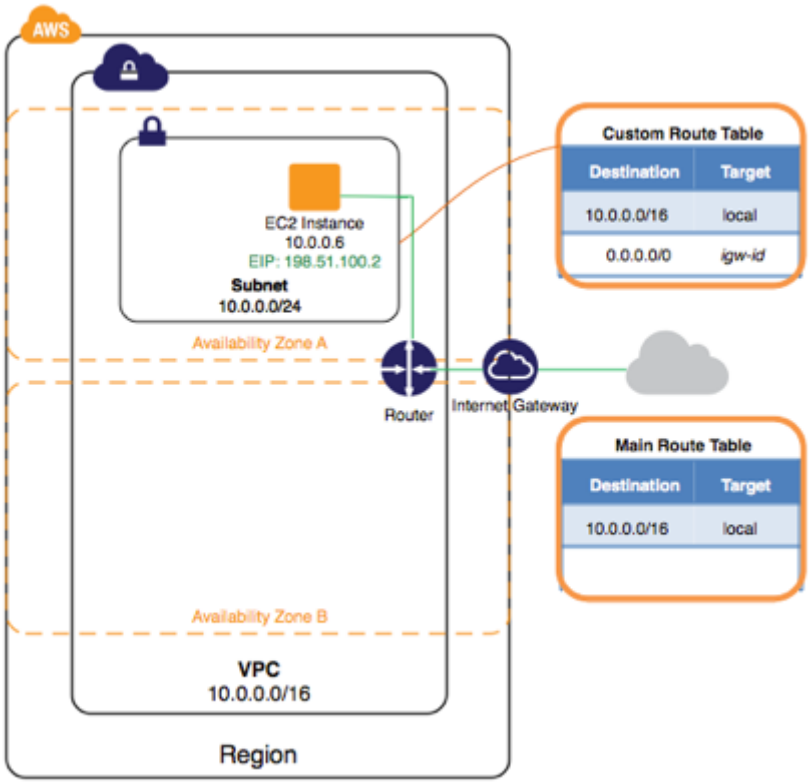
启动到 VPC 中的 Citrix ADC 实例最多可以具有八个弹性网络接口 (ENI)。反过来，可以为每个 ENI 分配一个或多个私有 IP 地址，其中每个地址都可以选择映射到可公开路由的弹性 IP 地址。

在这种情况下，网络接口和 IP 地址“弹性”的原因是能够以编程方式将它们重新映射到其他实例 — 这一功能可以从实例或可用区故障中恢复，而无需等待硬件更换或 DNS 更改完全传播到所有您的客户。

其他要说明的详细信息包括以下内容：

- 一个实例可以在不同的子网中具有不同的 ENI（但不能在不同的可用区域中）。

- 每个 ENI 必须至少分配一个 IP 地址，并且必须分配给安全组（见下文）。
- 每个子网的地址 1–4（即 10.x.x.1-4）均保留供 Amazon 使用。
- Citrix ADC 只能识别专用 IP 地址。分配的任何弹性 IP 都不会显示在 Citrix ADC CLI 或任何相关管理工具中。



EC2 与 VPC

AWS 包含多种不同的服务，例如 Amazon Simple Storage Services (S3)、Amazon Elastic Compute Cloud (EC2) 和 Amazon Virtual Private Cloud (VPC)。在这种情况下，后两者之间的区别很重要。特别是，对于 EC2，虚拟机实例仅限于单个网络接口和单个 IP 地址。此外，网络功能和控制极少。这就排除了对 Citrix ADC 使用 EC2（至少需要三个 IP 地址），这也是 Citrix ADC 实例只能在 AWS VPC 内启动的原因。

VPC 不仅支持具有多个接口以及多个私有和公有 IP 地址的虚拟机，还允许您使用自己的 IP 地址范围、子网、路由表和网络网关创建和控制隔离的虚拟网络环境。

区域和可用区

在 AWS 云中，区域是指特定的地理位置，如美国东部。在每个区域中，至少有两个可用区，每个可用区都可以被视为独立的云数据中心，该数据中心经过设计，可以避免其他可用区的故障，并提供低成本、低延迟的网络连接到同一区域。

通过在单独的可用区中实施实例，您可以保护您的应用程序免受影响单个位置的故障影响。

网络体系结构师在此级别需要注意的限制和依赖关系包括以下内容：

- 尽管虚拟私有云可以跨越多个可用区，但不能跨越多个区域。

- VPC 中的单个子网不能跨越多个可用区。
- 所有进出 VPC 的流量必须通过相应的默认互联网 Gateway 路由

在 **AWS** 上配置 **VPX**

在本练习中，您将创建 VPC 和子网，并在您的子网中启动面向公共的实例。您的实例将能够与 Internet 通信，并且您可以使用 SSH（如果是 Linux 实例）或远程桌面（如果是 Windows 实例）从本地计算机访问您的实例。在现实环境中，您可以使用此方案创建面向公众的 Web 服务器；例如，托管博客。

> 注意：
>
> 本练习旨在帮助您快速设置自己的非默认 VPC。如果您已经拥有默认 VPC 并且希望开始在其中启动实例（而不是创建或配置新 VPC），请参阅 在默认 VPC 中启动 EC2 实例。

要完成本练习，您将执行以下操作：

- 创建具有单个公有子网的非默认 VPC。子网使您能够根据您的安全和运营需求对实例进行分组。公有子网是可以通过互联网 Gateway 访问 Internet 的子网。
- 为您的实例创建一个仅允许通过特定端口的流量的安全组。
- 在您的子网中启动 Amazon EC2 实例。
- 将弹性 IP 地址与您的实例关联。这允许您的实例访问 Internet。

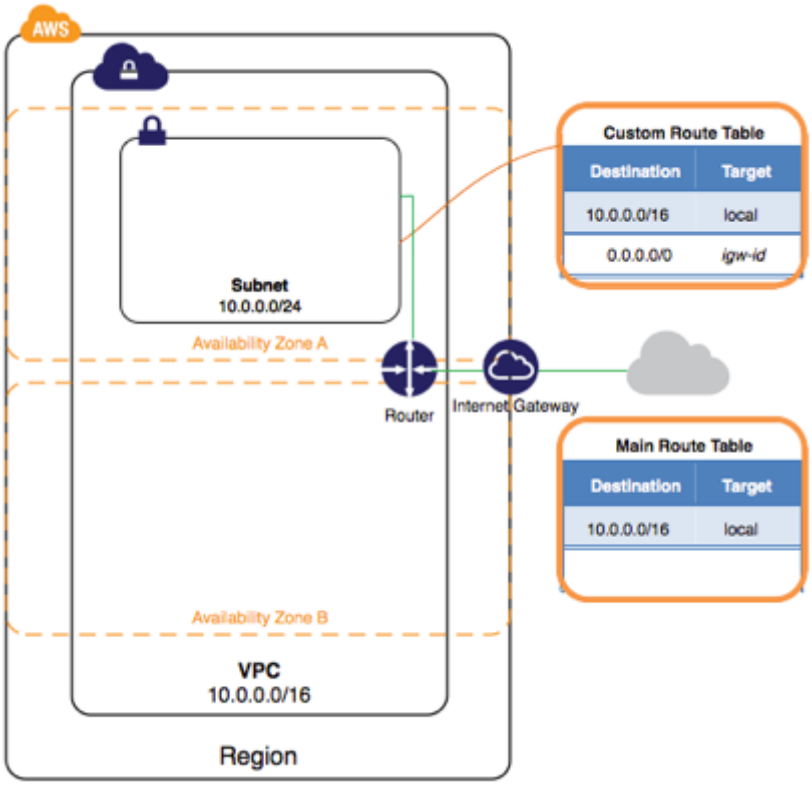
您必须先注册 AWS，然后才能首次使用 Amazon VPC。当您注册时，您的 AWS 帐户将自动注册 AWS 中的所有服务，包括 Amazon VPC。如果您尚未创建 AWS 帐户，请转到 http://aws.amazon.com/cn/，然后选择创建免费帐户。

步骤 **1**：创建 **VPC**

在此步骤中，您将使用 Amazon VPC 控制台中的 Amazon VPC 向导创建 VPC。向导将为您执行以下步骤：

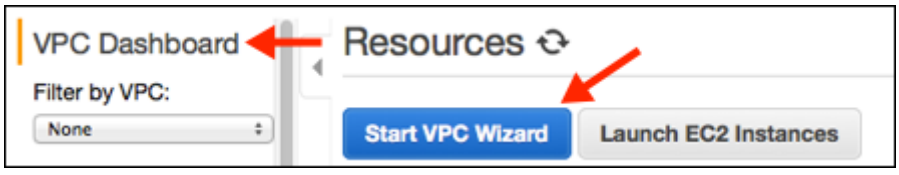
- 创建具有 /16 CIDR 块的 VPC（具有 65,536 个私有 IP 地址的网络）。有关 CIDR 表示法和 VPC 大小的更多信息，请参阅您的 VPC。
- 将互联网 Gateway 连接到 VPC。有关 Internet 网关的详细信息，请参阅 互联网网关。
- 在 VPC 中创建大小 /24 子网（256 个私有 IP 地址范围）。
- 创建自定义路由表，并将其与您的子网相关联，以便流量可以在子网和 Internet Gateway 之间流动。有关路由表的详细信息，请参阅 路由表。

下图显示了您完成此步骤后 VPC 的体系结构。



使用 **Amazon VPC** 向导创建 **VPC**

1. 打开 Amazon VPC 控制台，网址为 https://console.aws.amazon.com/vpc/。
2. 在导航栏的右上角，记下要在其中创建 VPC 的区域。确保在本练习的剩余时间内继续在同一区域工作，因为您无法从其他区域启动实例到 VPC 中。有关区域的详细信息，请参阅 区域和可用区。
3. 在导航窗格中，选择 **VPC** 控制板，然后选择启动 **VPC** 向导。



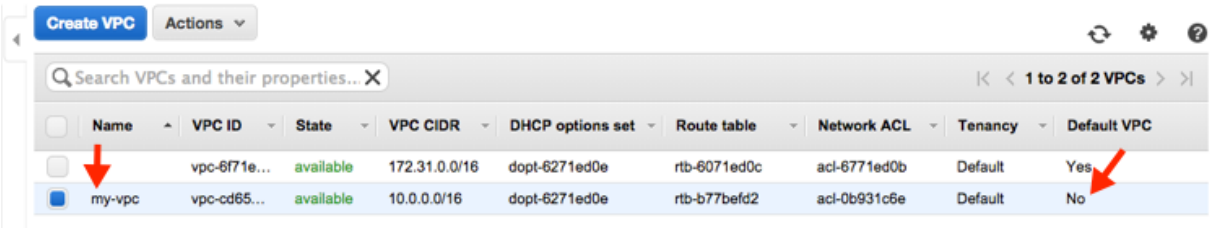
> 注意：
>
> 请勿在导航窗格中选择您的 VPC；您无法从此页面访问 VPC 向导。

4. 选择第一个选项，即 具有单个公有子网的 **VPC**，然后选择 “选择”。
5. 在配置页面上，在 VPC 名称字段中输入 VPC 的名称，例如 my-VPC，然后在子网名称字段中输入子网名称。这有助于您在创建 VPC 和子网后在 Amazon VPC 控制台中识别它们。在本练习中，您可以将其余配置设置保留在页面上，然后选择创建 **VPC**。

   （可选）如果您愿意，可以按如下方式修改配置设置，然后选择 “创建 **VPC**”。

- IP CIDR 块显示您将用于 VPC 的 IP 地址范围 (10.0.0.0/16)，公有子网字段显示您将用于子网的 IP 地址范围 (10.0.0.0/24)。如果您不想使用默认 CIDR 范围，可以指定自己的范围。有关详细信息，请参阅VPC 和子网大小调整。
- 使用可用区列表，您可以选择要在其中创建子网的可用区。您可以离开 “无首选项”，让 AWS 为您选择可用区。有关详细信息，请参阅区域和可用区。
- 在 “将 S3 端点添加到您的子网” 部分，您可以选择一个子网，在其中创建 VPC 端点到同一区域的 Amazon S3。有关详细信息，请参阅VPC 端点。
- `Enable DNS hostnames` 选项，设置为 “是” 时，可确保在 VPC 中启动的实例接收 DNS 主机名。有关详细信息，请参阅将 DNS 与您的 VPC 结合使用。
- 使用硬件租赁选项，您可以选择在 VPC 中启动的实例是在共享硬件还是专用硬件上运行。选择专用租户会产生额外费用。有关硬件租赁的详细信息，请参阅 专用实例。

6. 状态窗口显示正在进行的工作。工作完成后，选择 “确定” 以关闭状态窗口。
7. `Your VPCs page` 显示您的默认 VPC 和您刚刚创建的 VPC。您创建的 VPC 是非默认 VPC，因此默认 **VPC** 列显示 “否”。



查看有关您的 **VPC** 的信息

创建 VPC 后，您可以查看有关子网、Internet Gateway 和路由表的信息。您创建的 VPC 有两个路由表 — 一个默认情况下所有 VPC 都具有的主路由表，另一个由向导创建的自定义路由表。自定义路由表与您的子网相关联，这意味着该表中的路由决定子网流量的流动方式。如果您将新子网添加到 VPC，则默认情况下会使用主路由表。

查看有关您的 VPC 的信息

1. 打开 Amazon VPC 控制台，网址为 https://console.aws.amazon.com/vpc/。
2. 在导航窗格中，选择您的 **VPC**。记下您创建的 VPC 的名称和 ID（查看 “名称” 和 “VPC ID” 列）。您将使用此信息来标识与您的 VPC 关联的组件。
3. 在导航窗格中，选择子网。控制台显示创建 VPC 时创建的子网。您可以在 “名称” 列中按子网的名称标识子网，也可以使用在上一步中获得的 VPC 信息并查看 VPC 列。
4. 在导航窗格中，选择 **Internet** 网关。您可以通过查看 VPC 列查找附加到 VPC 的互联网 Gateway，该列显示 VPC 的 ID 和名称（如果适用）。
5. 在导航窗格中，选择 “路由表”。有两个与 VPC 关联的路由表。选择自定义路由表（主列显示否），然后选择 “路由” 选项卡以在详细信息窗格中显示路由信息：

- 表中的第一行是本地路由，它允许 VPC 中的实例进行通信。默认情况下，该路由存在于每个路由表中，您无法将其删除。
- 第二行显示 Amazon VPC 向导添加的路由，该路由旨在启用发往 VPC 外部 IP 地址 (0.0.0.0/0) 的流量从子网流向互联网 Gateway。

6. 选择主路由表。主路由表具有本地路由，但没有其他路由。

步骤 **2**：创建安全组 **12**

安全组充当虚拟防火墙来控制其关联实例的流量。要使用安全组，您可以添加入站规则来控制实例的传入流量，并添加出站规则来控制来自实例的传出流量。要将安全组与实例关联，请在启动实例时指定安全组。如果您在安全组中添加和删除规则，我们会自动将这些更改应用于与安全组关联的实例。

您的 VPC 附带默认安全组。启动期间未与其他安全组关联的任何实例都与默认安全组关联。在本练习中，您将创建一个新的安全组 WebServerSG，并在您将实例启动到 VPC 时指定此安全组。

主题

- 创建您的 WebServerSG 安全组
- WebServerSG 安全组的规则

创建您的 WebServerSG 安全组

您可以使用 Amazon VPC 控制台创建安全组。

WebServerSG 安全组的规则

下表介绍了 WebServerSG 安全组的入站和出站规则。您将自行添加入站规则。出站规则是允许所有出站通信到任何位置的默认规则，您无需自行添加此规则。

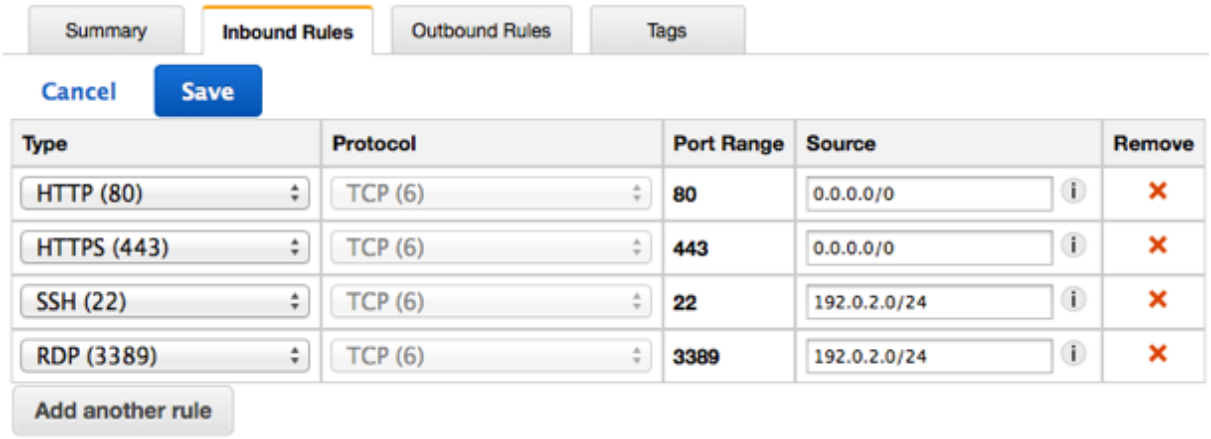
入站			
源 IP	协议	端口范围	备注
0.0.0.0/0	TCP	80	允许从任何位置访问入站 HTTP。
0.0.0.0/0	TCP	443	允许从任何位置访问入站 HTTPS。
家庭网络的公有 IP 地址范围	TCP	22	允许从您的家庭网络对 Linux/UNIX 实例的入站 SSH 访问。
家庭网络的公有 IP 地址范围	TCP	3389	允许从您的家庭网络到 Windows 实例的入站 RDP 访问。

出站			
目标 IP	协议	端口范围	备注
0.0.0.0/0	全部	全部	允许所有出站通信的默认出站规则。

创建 WebServerSG 安全组并添加规则

1. 打开 Amazon VPC 控制台，网址为 https://aws.amazon.com/console/。
2. 在导航窗格中，选择“安全组”。
3. 选择“创建安全组”。
4. 在组名称字段中，输入 WebServerSG 作为安全组的名称，并提供描述。您可以选择使用“名称”标记字段为安全组创建一个标记，其中键为“名称”和您指定的值。
5. 从 VPC 菜单中选择 **VPC** 的 **ID**，然后选择是、创建。
6. 选择您刚刚创建的 **WebServerSG** 安全组（您可以在“组名称”列中查看其名称）。
7. 在“入站规则”选项卡上，选择“编辑”并添加入站流量的规则，如下所示，然后选择“完成后 保存”：
   - 从“类型”列表中选择 **HTTP** ，然后在“源”字段中输入 **0.0.0.0/0** 。
   - 选择“添加其他规则”，然后从“类型”列表中选择 **HTTPS**，然后在“源”字段中输入 **0.0.0.0/0**。
   - 选择“添加其他规则”。如果要启动 Linux 实例，请从“类型”列表中选择 **SSH** ，或者如果要启动 Windows 实例，请从“类型”列表中选择 **RDP** 。在“源”字段中输入网络的公有 IP 地址范围。如果您不知道此地址范围，则可以使用 0.0.0.0/0 进行本练习。

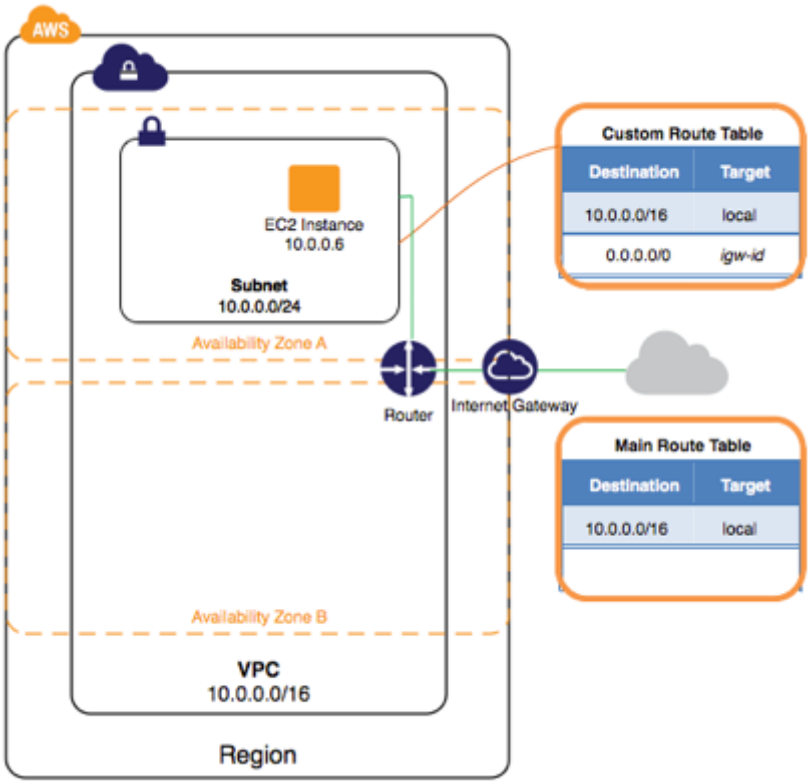
> 小心：
>
> 如果使用 0.0.0.0/0，则启用所有 IP 地址以使用 SSH 或 RDP 访问您的实例。这对于短期练习来说是可以接受的，但对于生产环境来说是不安全的。在生产中，您只授权特定的 IP 地址或地址范围以访问您的实例。



步骤 **3**：在您的 **VPC 14** 中启动实例

在 VPC 中启动 EC2 实例时，您必须指定要在其中启动实例的子网。在这种情况下，您将在您创建的 VPC 的公有子网中启动实例。您将使用 Amazon EC2 控制台中的 Amazon EC2 启动向导来启动您的实例。

下图显示了您完成此步骤后 VPC 的体系结构。



在 VPC 中启动 EC2 实例

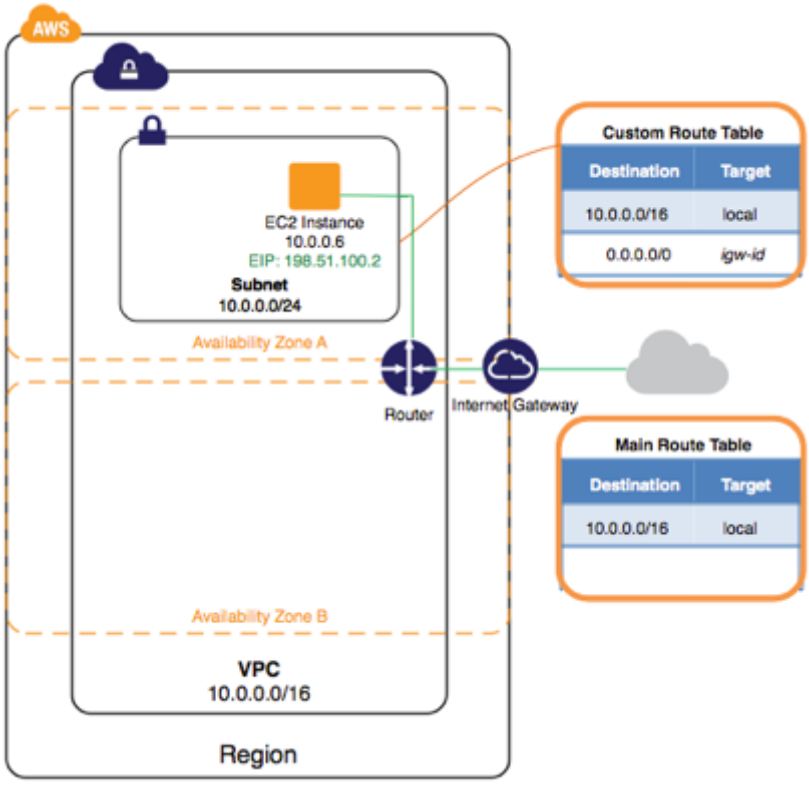
1. 打开 Amazon EC2 控制台。
2. 在导航栏中的右上角，确保您选择了创建 VPC 和安全组的相同区域。
3. 从控制板中选择启动实例。
4. 在向导的第一页上，选择要使用的 AMI。在本次练习中，我们建议您选择 **Amazon Linux AMI** 或 **Windows AMI** 。
5. 在“选 择实例类型”页面上，您可以选择要启动的实例的硬件配置和大小。默认情况下，向导会根据您选择的 AMI 选择第一个可用的实例类型。您可以保留默认选择，然后选择“下一步：配置实例详细信息”。
6. 在“配置实例详细信息”页面上，从“网络”列表中选择您创建的 **VPC**，并从子网列表中选择子网。保留其余的默认设置，然后浏览向导的下一页，直到您进入“标签实例”页面。
7. 在“标记实例”页面上，您可以使用名称标记标记实例；例如，名称 = MyWebServer。这可帮助您在启动后在 Amazon EC2 控制台中识别您的实例。选择“下一步：完成后配置安全组”。
8. 在“配置安全组”页面上，向导会自动定义启动向导 x 安全组，以允许您连接到您的实例。相反，选择“选择现有安全组”选项，选择之前创建的 WebServerSG 组，然后选择“查看和启动”。
9. 在“查看实例启动”页面上，检查实例的详细信息，然后选择“启动”。

10. 在“选 择现有密钥对或 创建新密钥对”对话框中，您可以选择现有密钥对，也可以创建新密钥对。如果您创建新的密钥对，请确保您下载该文件并将其存储在安全位置。启动实例后，您需要私钥的内容才能连接到实例。要启动实例，请选中确认复选框，然后选择启动实例。
11. 在确认页面上，选择查看实例以在实例页面上查看您的实例。选择您的实例，然后在“描述”选项卡中查看其详细信息。私有 IP 字段显示从子网中的 IP 地址范围中分配给您的实例的私有 IP 地址。

步骤 **4**：为您的实例分配弹性 **IP** 地址

在上一步中，您将实例启动到公有子网中，该子网具有通往 Internet Gateway 的路由。但是，您的子网中的实例还需要一个公有 IP 地址才能与 Internet 通信。默认情况下，不会为非默认 VPC 中的实例分配公有 IP 地址。在此步骤中，您将为您的帐户分配弹性 IP 地址，然后将其与您的实例关联。有关弹性 IP 地址的更多信息，请参阅弹性 IP 地址。

下图显示了您完成此步骤后 VPC 的体系结构。



分配和分配弹性 IP 地址

1. 打开 Amazon VPC 控制台，网址为 https://console.aws.amazon.com/vpc/。
2. 在导航窗格中，选择弹性 **IP**。
3. 选择“分配新地址”，然后选择“是，分配”。

> 注意：
>
> 如果您的帐户支持 EC2-Classic，请首先从网络平台列表中选择 **EC2-VPC**。

4. 从列表中选择弹性 **IP** 地址，选择操作，然后选择关联地址。
5. 在对话框中，从与列表关联中选择实例，然后从实例列表中选择您的实例。完成后，选择是，关联。

您的实例现在可以从互联网访问。您可以使用 SSH 或远程桌面从家庭网络通过其弹性 IP 地址连接到您的实例。有关如何连接到 Linux 实例的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的连接到您的 Linux 实例。有关如何连接到 Windows 实例的更多信息，请参阅《适用于 Windows 实例的 Amazon EC2 用户指南》中的使用 RDP 连接到您的 Windows 实例。

这就完成了练习；您可以选择继续在 VPC 中使用您的实例，或者如果您不需要实例，则可以终止实例并释放其弹性 IP 地址，以避免为实例产生费用。您还可以删除您的 VPC — 请注意，在本练习中创建的 VPC 和 VPC 组件（如子网和路由表）不需要为您付费。

为 **Citrix Virtual Apps and Desktops** 配置 **Unified Gateway**

导航到您的 Citrix ADC 的管理控制台。

使用 nsroot 和 AWS 在构建过程中分配的实例 ID 登录到 Citrix ADC。

安装 SSL 证书：

1. 导航到流量管理 **— SSL**。右键单击并启用此功能。
2. 导入 SSL 证书密钥对。

安装 SSL 证书：

1. 展开 Citrix Gateway 并选择虚拟服务器。
2. 单击添加。

   输入在 Citrix ADC 构建过程中分配的公有子网中的 Gateway 和 IP 地址的名称。

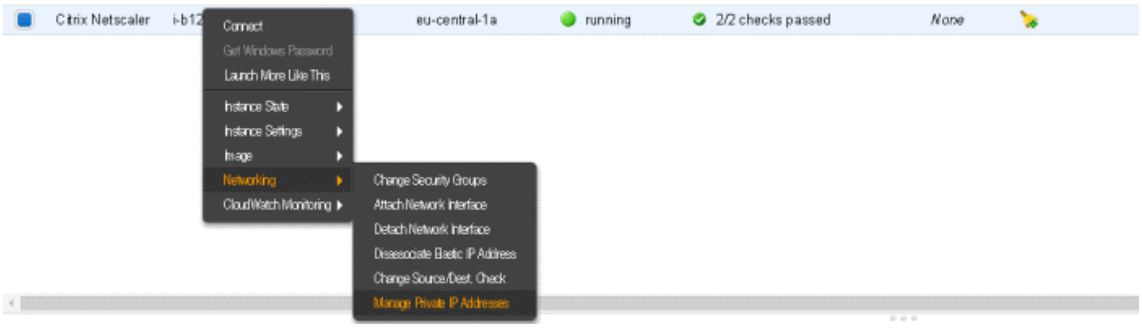
   > 注意：
   >
   > 在稍后分配弹性 IP 地址时，请根据需要记下此 IP 地址。

3. 单 击 **“**确定”，然后单击“无服务器证书”，然后选择您之前导入的证书。单击 **Bind**（绑定）。
4. 单击确定和完成，在此阶段，您应将 Citrix Gateway 显示为“向上”状态。

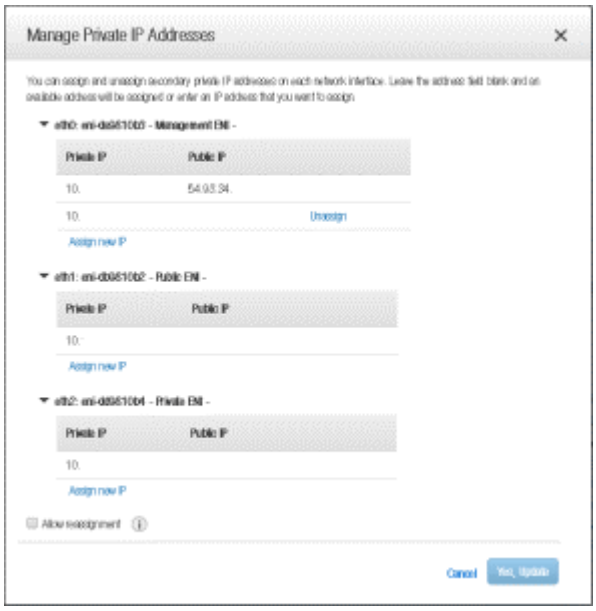
要配置 Unified Gateway，请参阅 https://support.citrix.com/article/CTX205485。

提供对 Unified Gateway 实例的外部访问：

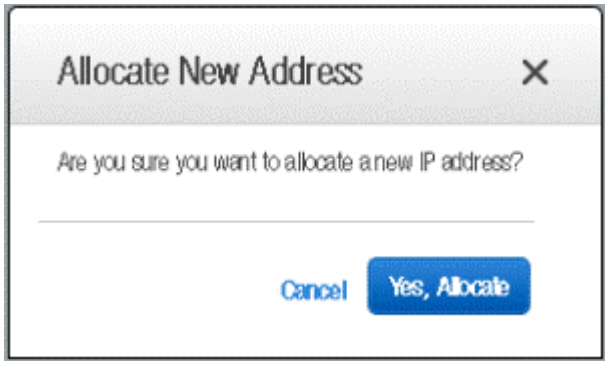
1. 登录 AWS 门户网站，然后导航到您的实例。
2. 右键单击 Citrix ADC，选择“网络”，然后选择“管理私有 **IP** 地址”。



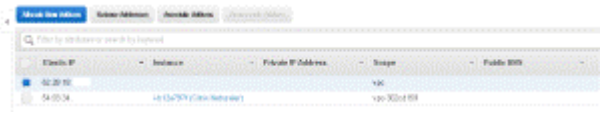
3. 在要运行 Citrix ADC 网关的接口上单击“分配新 **IP**”。
4. 分配 IP 地址，请确保您使用分配给 Citrix ADC 网关的地址相同。



5. 单击 是更新。这将在 AWS 级别为实例分配新的 IP 地址。您现在可以为此私有 IP 分配新的弹性 IP。
6. 导航到“网络和安全”以及“弹性 IP”。
7. 出现提示时，单击“分配新地址”— 选择“是”以获取新的 IP 地址。



8. 从列表中选择地址，然后选择关联地址。



9. 从实例列表中选择先前构建的 **Citrix ADC** 实例。选择此选项后，您将能够选择静态分配给实例的 IP 地址（与 Citrix Gateway 相同的地址），然后选择关联。

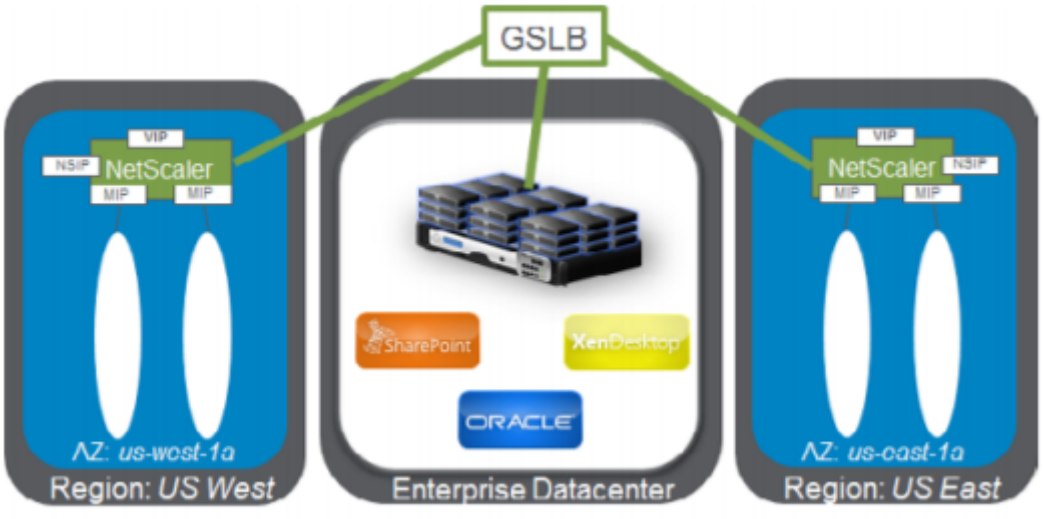
10. 将您的 DNS 名称记录指向 Amazon 为您分配的弹性 IP 地址。
11. 登录到您的 Citrix Gateway。

面向 **StoreFront** 的高可用性负载平衡

请参阅 Citrix 配置步骤。

在两个 **AWS** 位置配置 **GSLB**

在 AWS 上为 Citrix ADC 设置 GSLB 主要包括配置 Citrix ADC，以便将流量负载平衡到位于 Citrix ADC 所属 VPC 外的服务器，例如在不同可用区域中的另一个 VPC 或本地数据中心内等。



具有云负载平衡器的基于域名的服务 **(GSLB DBS)**

GSLB 和 **DBS** 概述

对云负载平衡器使用 DBS（基于域的服务）的 Citrix ADC GSLB 支持允许使用云负载平衡器解决方案自动发现动态云服务。此配置允许 Citrix ADC 在主动-主动环境中实现全局服务器负载平衡基于域名的服务 (GSLB DBS)。DBS 允许通过 DNS 发现扩展 AWS 和 Microsoft Azure 环境中的后端资源。

本部分内容介绍了 AWS 中的 Citrix ADC 与 Azure Auto Scaling 环境之间的集成。文档的最后一部分详细介绍了设置跨两个特定于 AWS 区域的不同可用区 (AZ) 的 HA 对 Citrix ADC 的能力。

必备条件

Citrix ADC GSLB 服务组的先决条件包括正常运行的 AWS/Microsoft Azure 环境，具有配置安全组、Linux Web 服务器、AWS 内的 Citrix ADC、弹性 IP 和弹性负载平衡器的知识和能力。

GSLB DBS 服务集成要求为 AWS ELB 和 Microsoft Azure ALB 负载平衡器实例使用 Citrix ADC 版本 12.0.57。

Citrix ADC GSLB 服务组功能增强

GSLB 服务组实体：Citrix ADC 版本 12.0.57

引入了 GSLB 服务组，它支持使用 BDS 动态发现自动缩放。

DBS 功能组件（基于域的服务）必须绑定到 GSLB 服务组

示例:

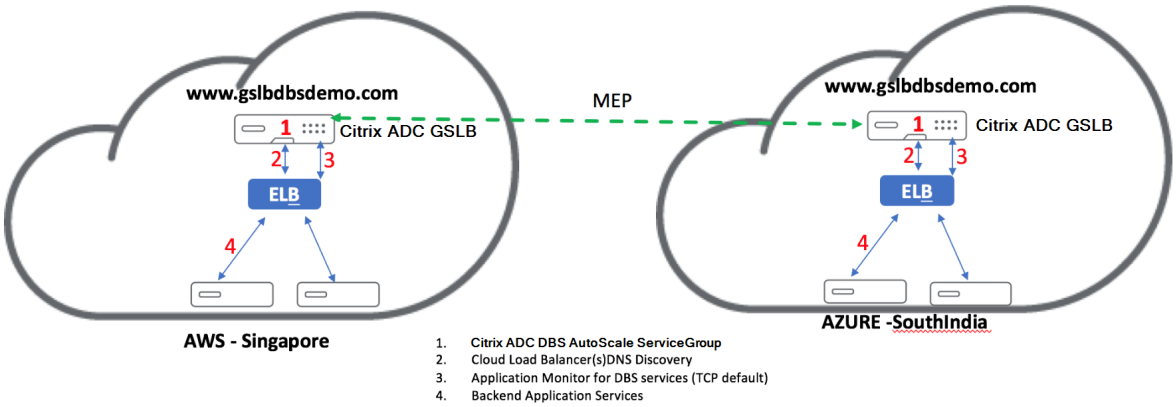
```
1 > add server sydney_server LB-Sydney-xxxxxxxxxx.ap-southeast-2.elb.
     amazonaws.com
2 > add gslb serviceGroup sydney_sg HTTP -autoScale DNS -siteName sydney
3 > bind gslb serviceGroup sydney_sg sydney_server 80
4 <!--NeedCopy-->
```

基于域名的服务 **— AWS ELB**

GLSB DBS 利用弹性负载平衡器的 FQDN 动态更新 GSLB 服务组，以包括在 AWS 中创建和删除的后端服务器。AWS 中的后端服务器或实例可配置为根据网络需求或 CPU 利用率进行扩展。要配置此功能，我们将 Citrix ADC 指向弹性负载平衡器，以动态路由到 AWS 中的不同服务器，而无需每次在 AWS 中创建和删除实例时手动更新 Citrix ADC。用于 GSLB 服务组的 Citrix ADC DBS 功能使用 DNS 感知服务发现来确定 AutoScaler 组中标识的 DBS 命名空间的成员服务资源。

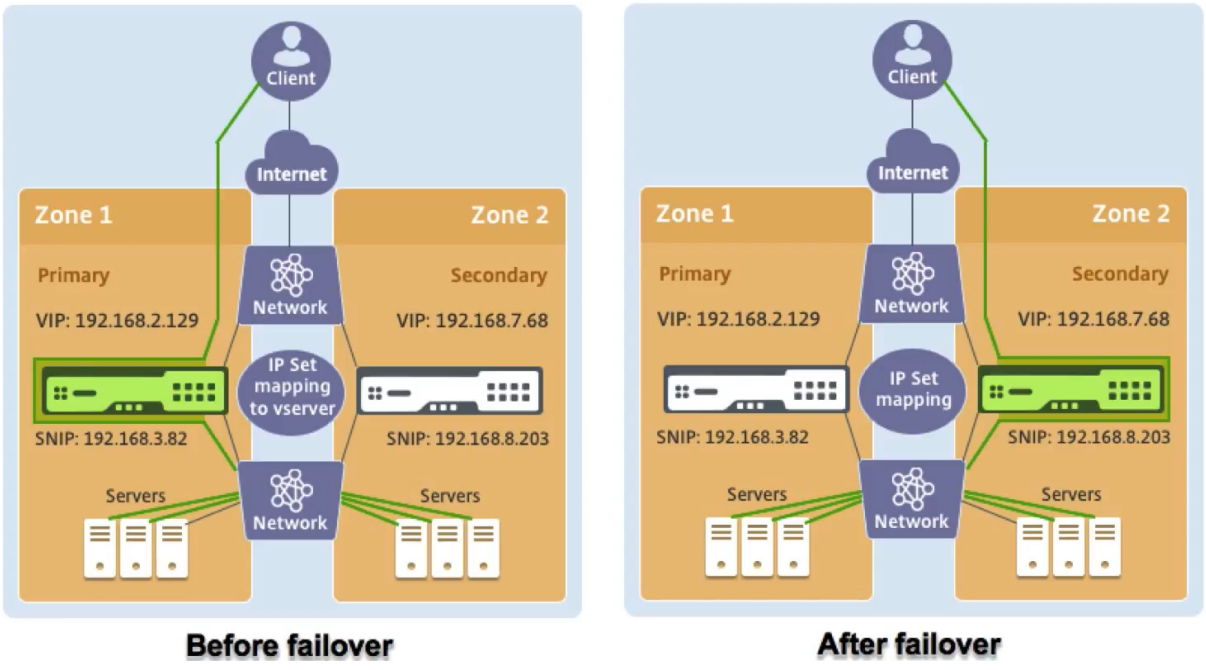
图表：

Citrix ADC GSLB DBA 使用云负载平衡器 AutoScale 组件



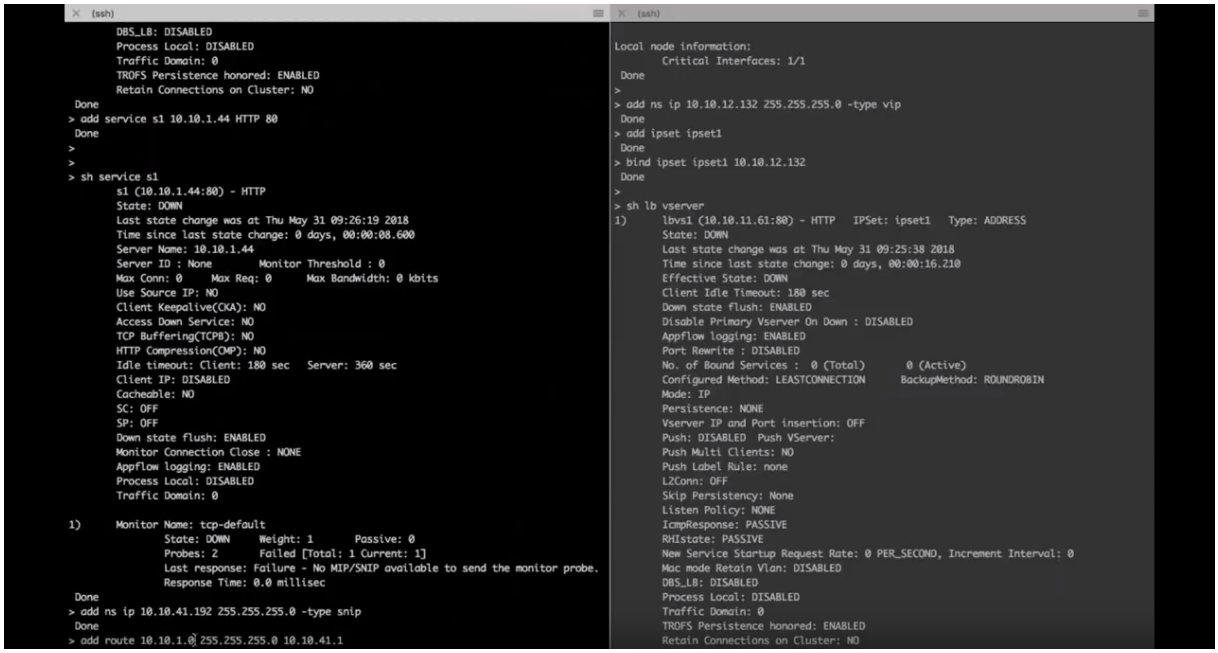
在 **AWS** 中跨多个可用区使用 **Citrix ADC HA**

在 AWS 中跨不同可用区部署 Citrix ADC 是针对 Citrix ADC 12.1 发布的一项新功能。这是通过将 Citrix ADC 连接到弹性网络 IP 地址 (ENI) 来完成的。

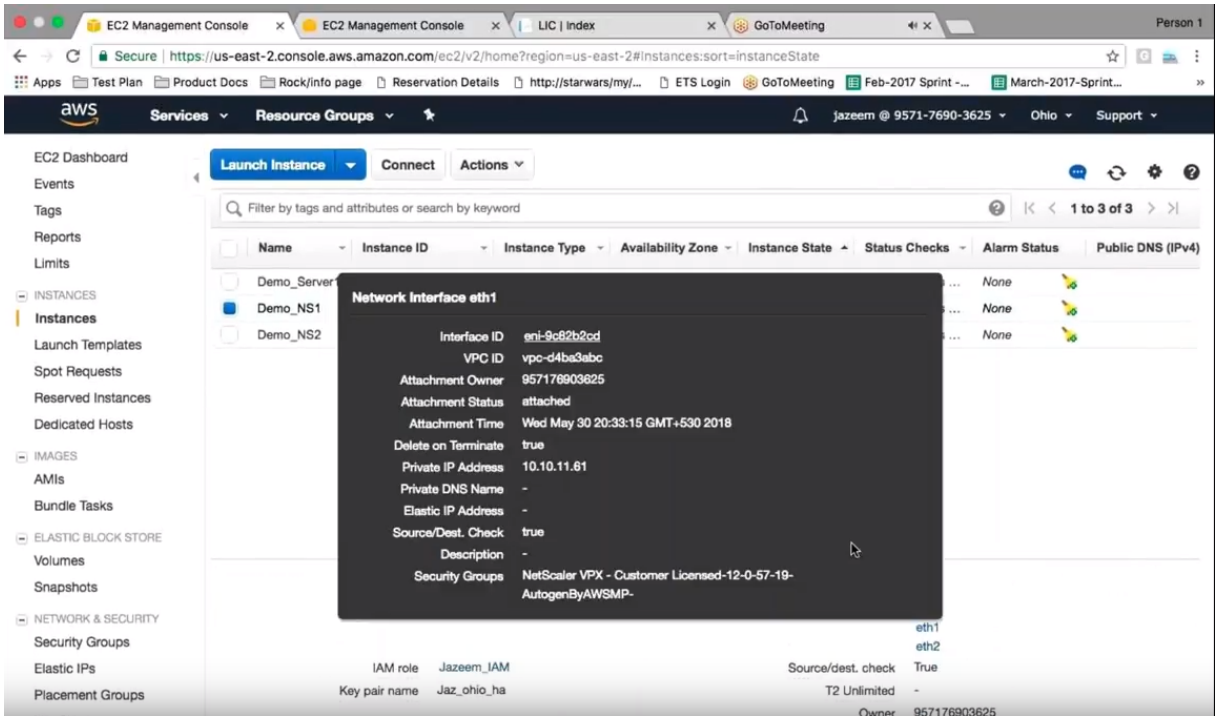


该解决方案的工作方式与其他解决方案略有不同，因为它要求您在 VPX 上设置 HA 和独立的网络配置。此解决方案为虚拟服务器使用 IP 集功能的新功能来维护故障切换。

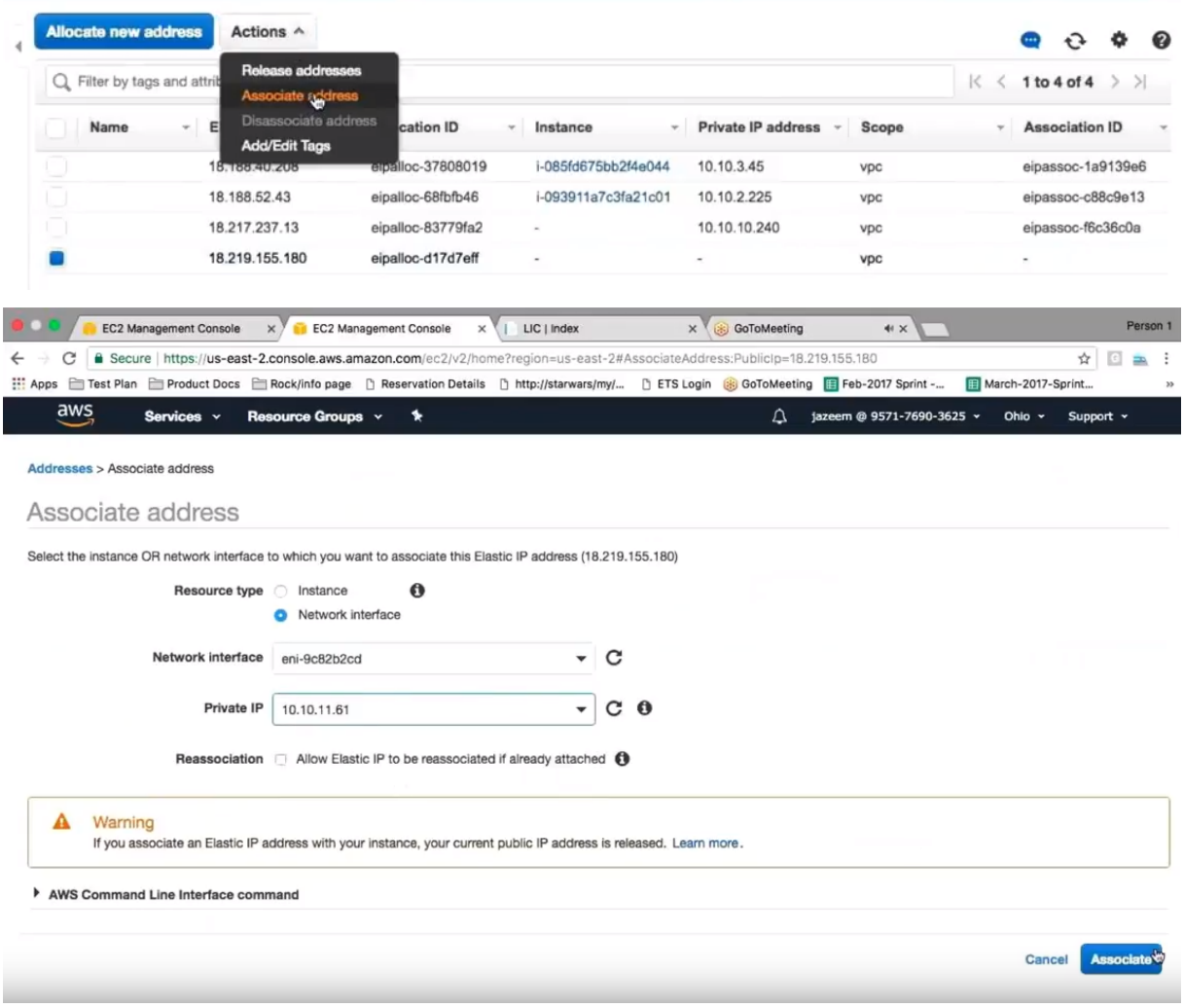
要开始使用，您必须登录到 Citrix ADC，并定义或支持服务器端网络地址、客户端地址以及到两者的路由。



在 AWS 控制台中，使用弹性 IP 设置了第一个 VPX。

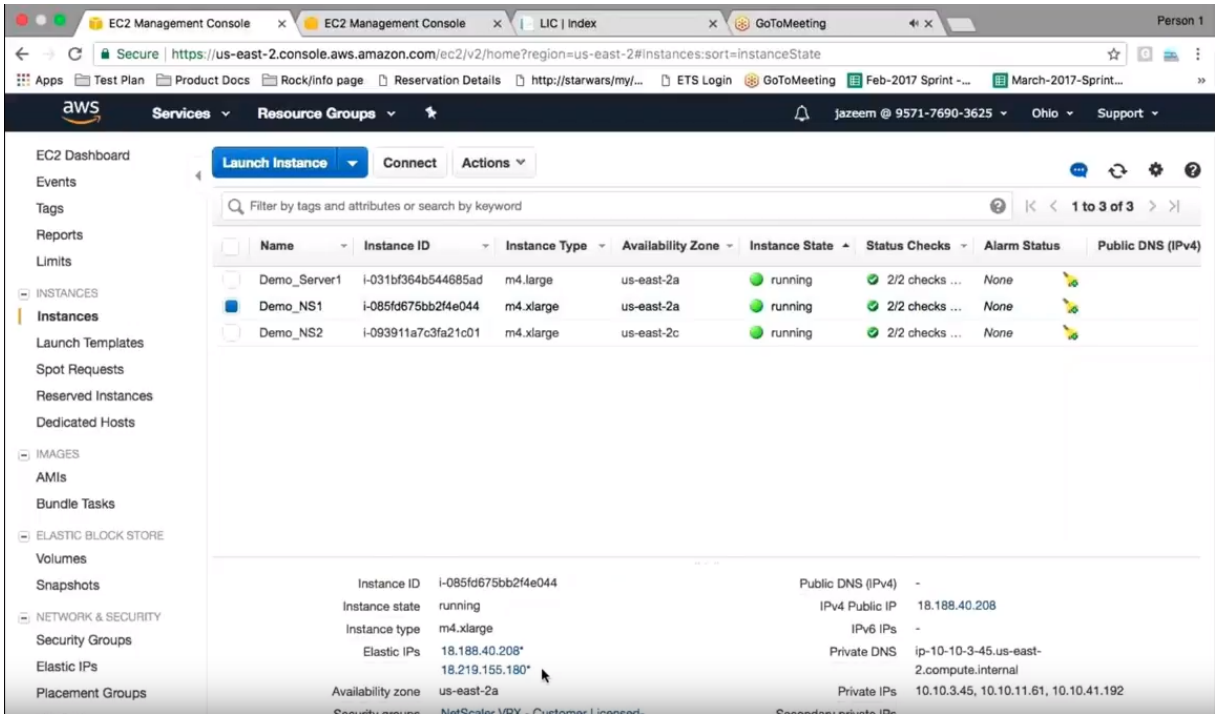


进入弹性接口，使解决方案工作的第一件事是将弹性 IP 关联到该接口上的现有私有地址。

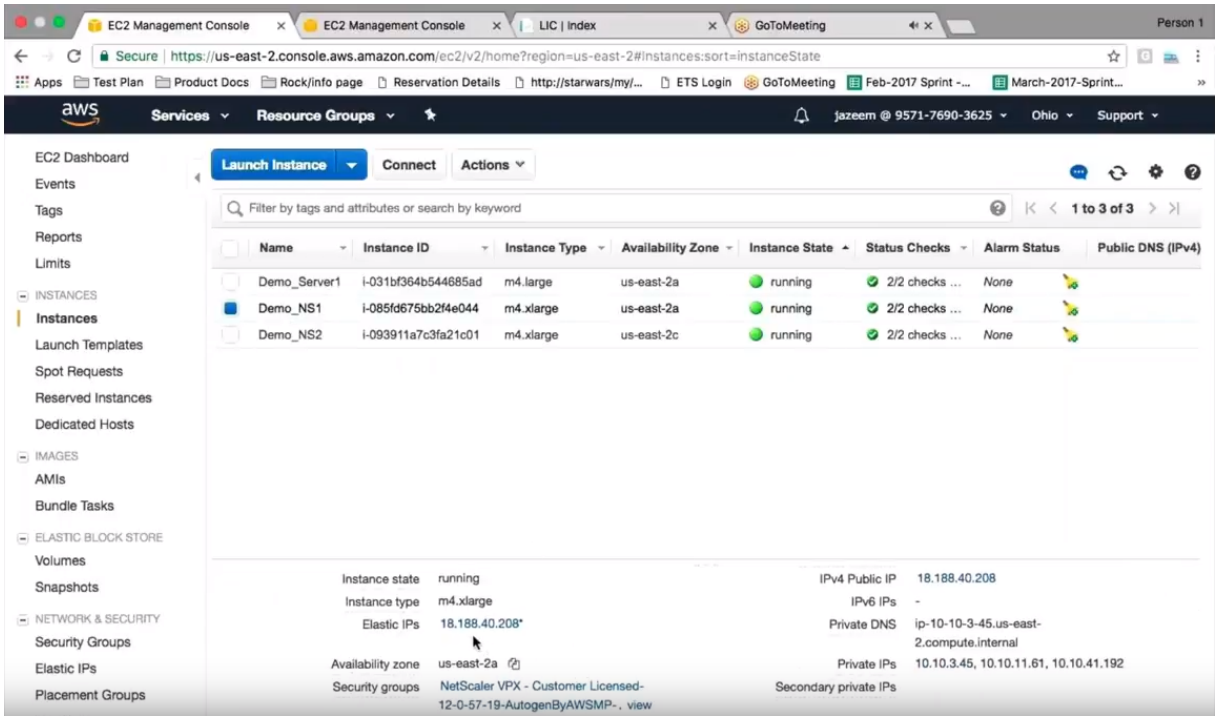


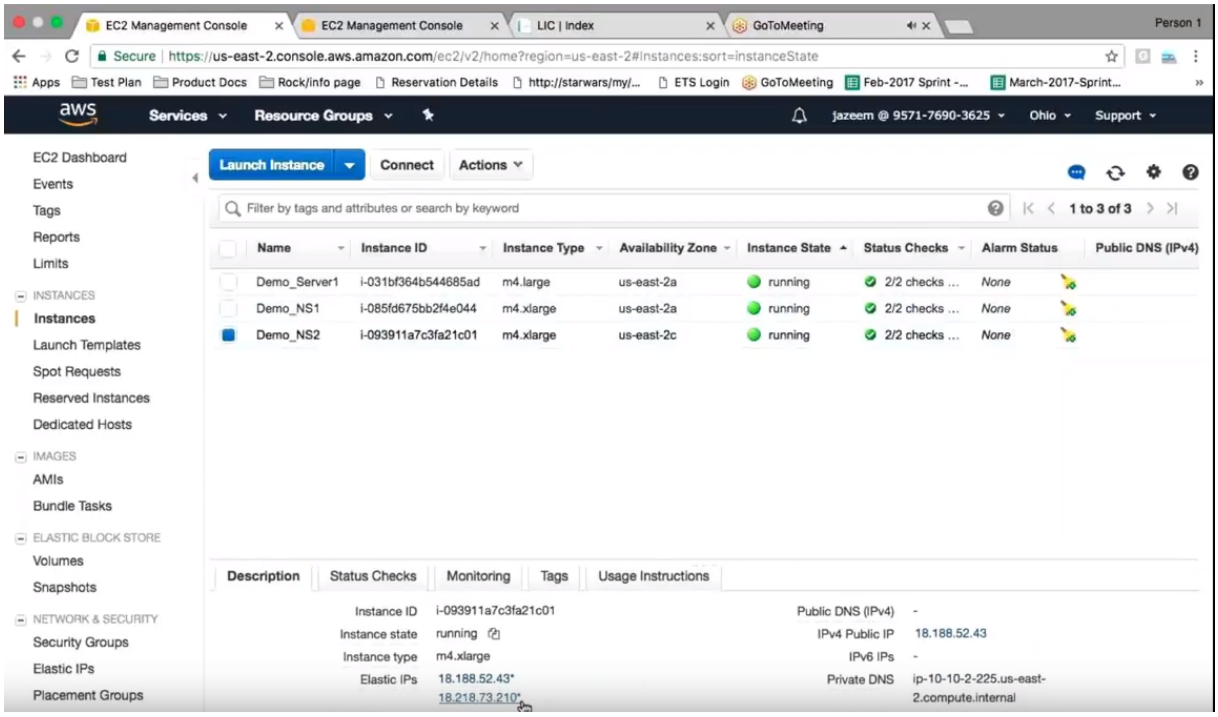
建立该关联后，您就可以继续执行故障转移。

在底部，VPX 上现在应该有第二个弹性 IP。

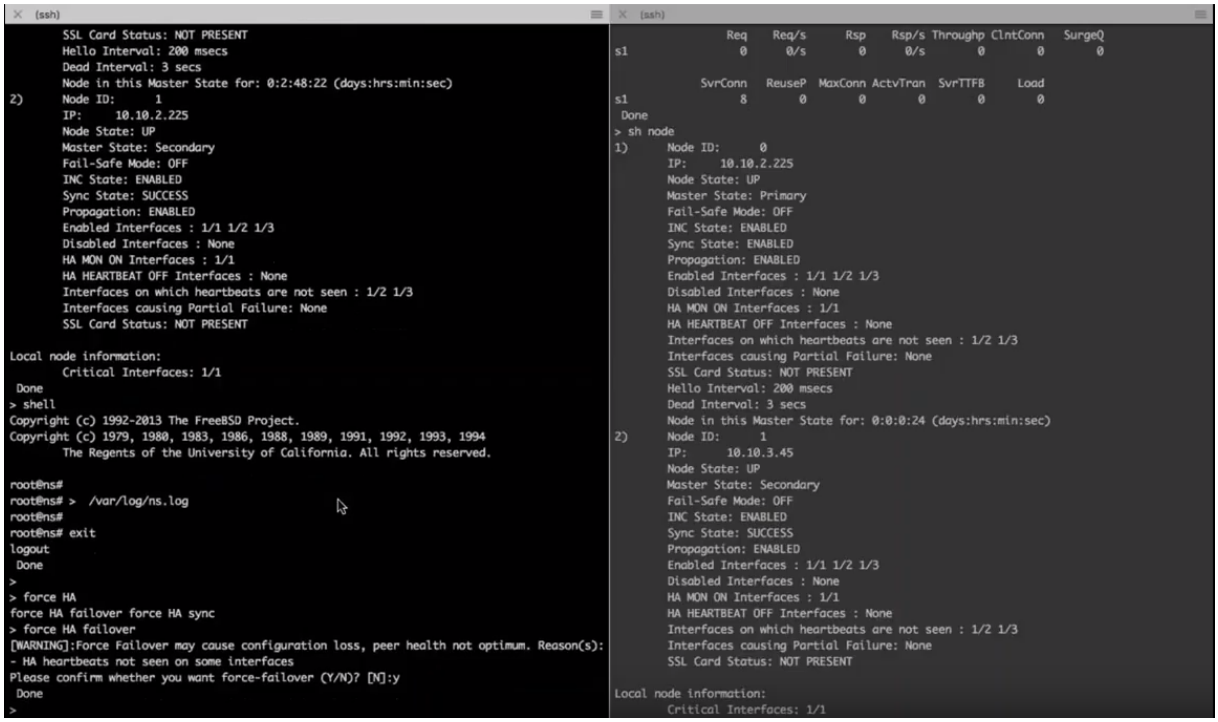


因此，请转到 VPX 启动故障转移，然后返回 AWS 控制台。这次查看属于第一个 Citrix ADC 的弹性 IP 时，请注意新的弹性 IP 不在那里，因为它现在已移动到第二个 Citrix ADC。





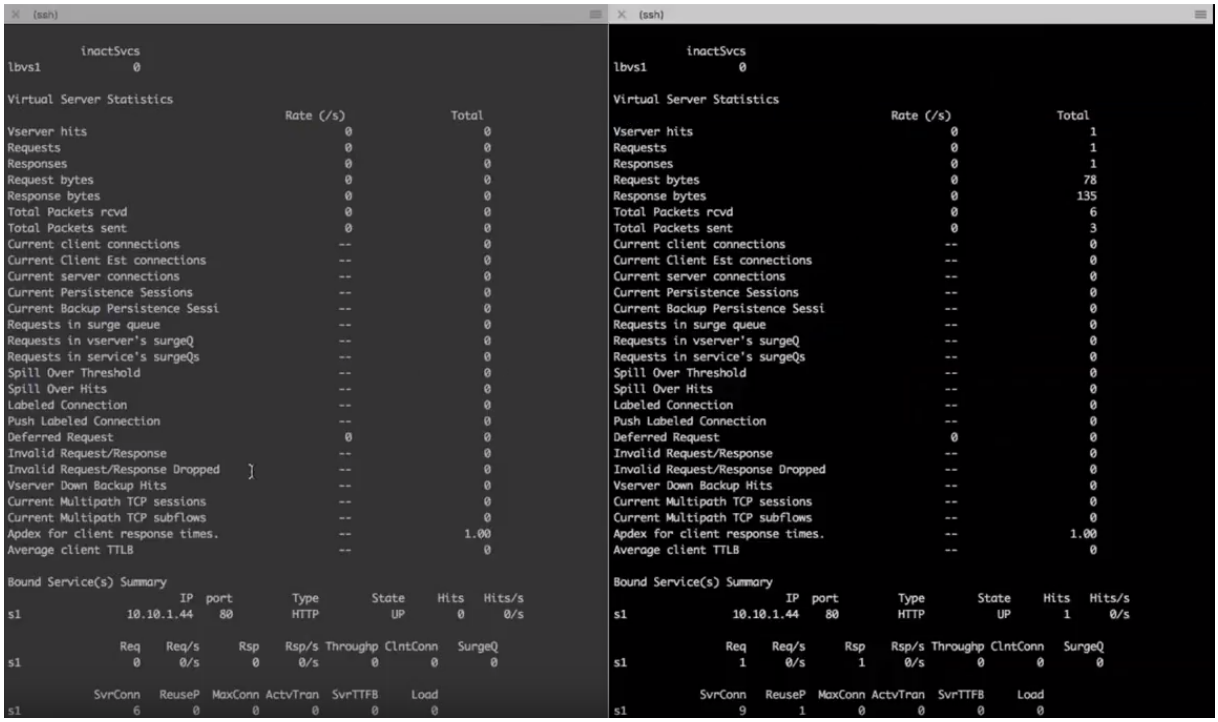
要验证这一点，请在第一个和第二个 Citrix ADC 上输入一个 **show node** 命令，以查看第二个 Citrix ADC 现在与处于待机状态之前一样处于主状态。



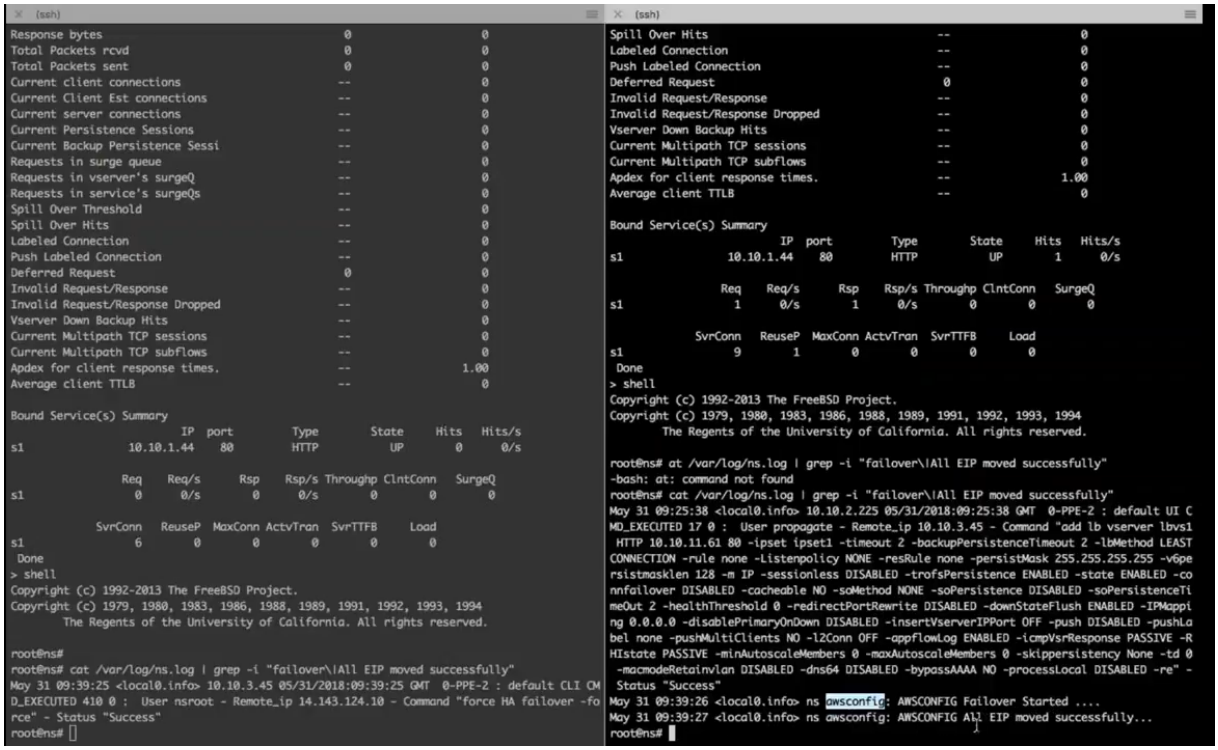
现在，您可以查看实时流量。


```
                    </div>
                    </div>
                </div>
        </div>
</form>

<script language="JavaScript" type="text/javascript">
//Don't allow this page to be embedded inside a frame
if(self != top)
{
    document.getElementsByTagName("body")[0].style.display = "none";
    top.location = self.location;
}
else
{

    $('form[name="form1"] #username').focus();
}

function input_hints() {
    var inputs = document.getElementsByTagName("input");
    for (var i = 0; i < inputs.length; i++) {
            // test to see if the hint span exists first
            if (inputs[i].parentNode.getElementsByTagName("span")[0]) {
                    // the span exists!  on focus, show the hint
                    inputs[i].onfocus = function() {
                            this.parentNode.getElementsByTagName("span")[0].className = "ns_active ns_active_color";
                    };
                            // when the cursor moves away from the field, hide the hint
                    inputs[i].onblur = function() {
^C
Jazeems-MacBook-Pro:~ jazeem$ curl -I http://18.218.73.210/
HTTP/1.1 200 OK
Date: Thu, 31 May 2018 09:38:23 GMT
Server: Apache
X-Frame-Options: DENY
Content-Type: text/html; charset=UTF-8

Jazeems-MacBook-Pro:~ jazeem$ curl -I http://18.218.73.210/
HTTP/1.1 200 OK
Date: Thu, 31 May 2018 09:40:03 GMT
Server: Apache
X-Frame-Options: DENY
Content-Type: text/html; charset=UTF-8
```

您可以在故障转移后向 VIP 发送请求。如果您在 Citrix ADC 上首次处于活动状态的 LB 虚拟服务器上执行统计信息，请注意没有任何请求单击该处。如果您在以前的备用（现为活动的 Citrix ADC）上运行相同的命令，则可以看到有一个虚拟服务器单击那里。显示高可用性过渡后，流量传入新的 Citrix ADC。



现在，如果您想进行一些调试或查看当前状态是什么，您可以将其放到 shell 并查找记录，以显示 HA 故障切换发生的时间，以及 AWS 配置或 API 调用何时将所有 EIP 从主 Citrix ADC 转移到辅助设备。



配置 **AWS** 组件

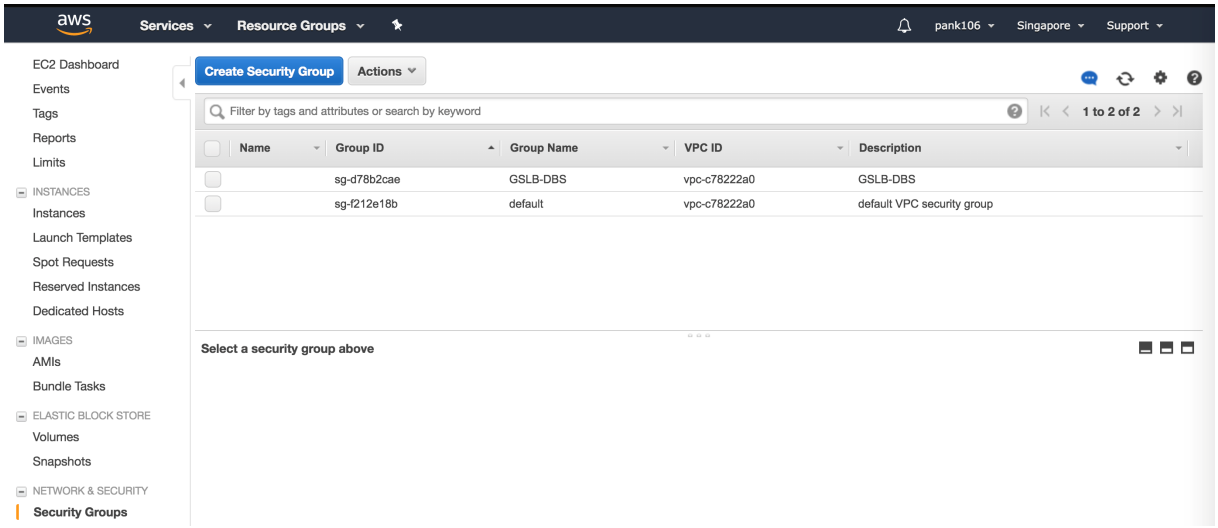
安全组

> 注意：
>
> 建议为 ELB、Citrix ADC GSLB 实例和 Linux 实例创建不同的安全组，因为这些实体所需的规则集各不相同。此示例具有一个合并的安全组配置，以便简洁起见。

请参阅 VPC 的安全组 以确保虚拟防火墙的正确配置。

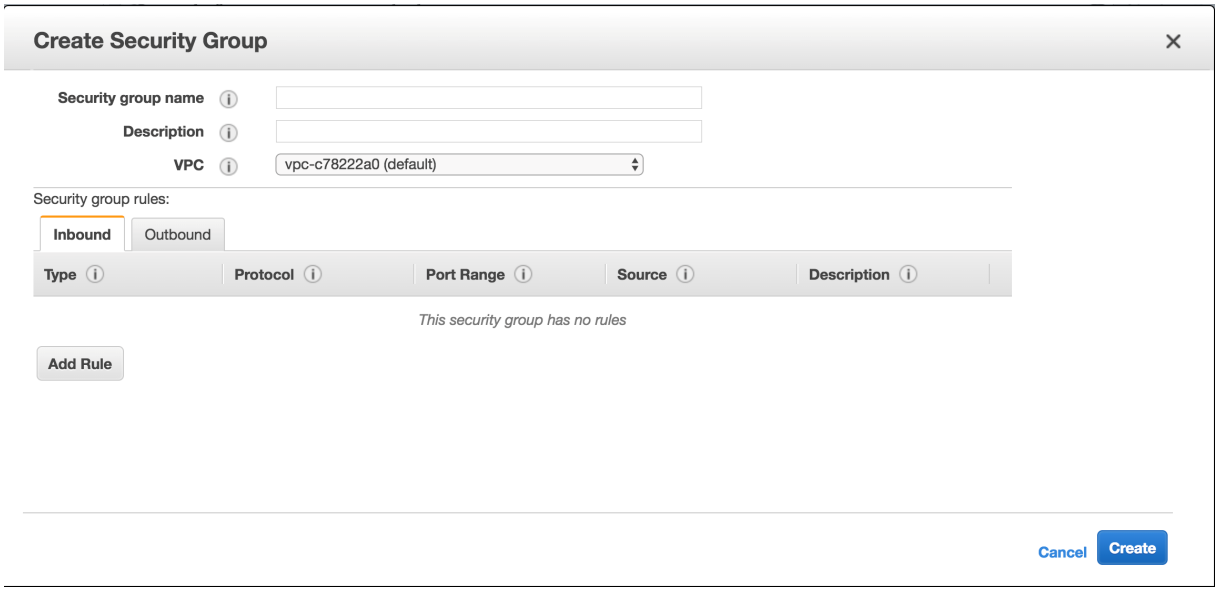
步骤 1:

登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 内导航到 “网络和安全**”>“**安全组”。



步骤 2:

单击创建安全组并提供名称和说明。此安全组包括 Citrix ADC 和 Linux 后端 Web 服务器。



步骤 3：

从下面的屏幕截图中添加入站端口规则。

> 注意：
>
> 建议对源 IP 访问进行粒度强化。

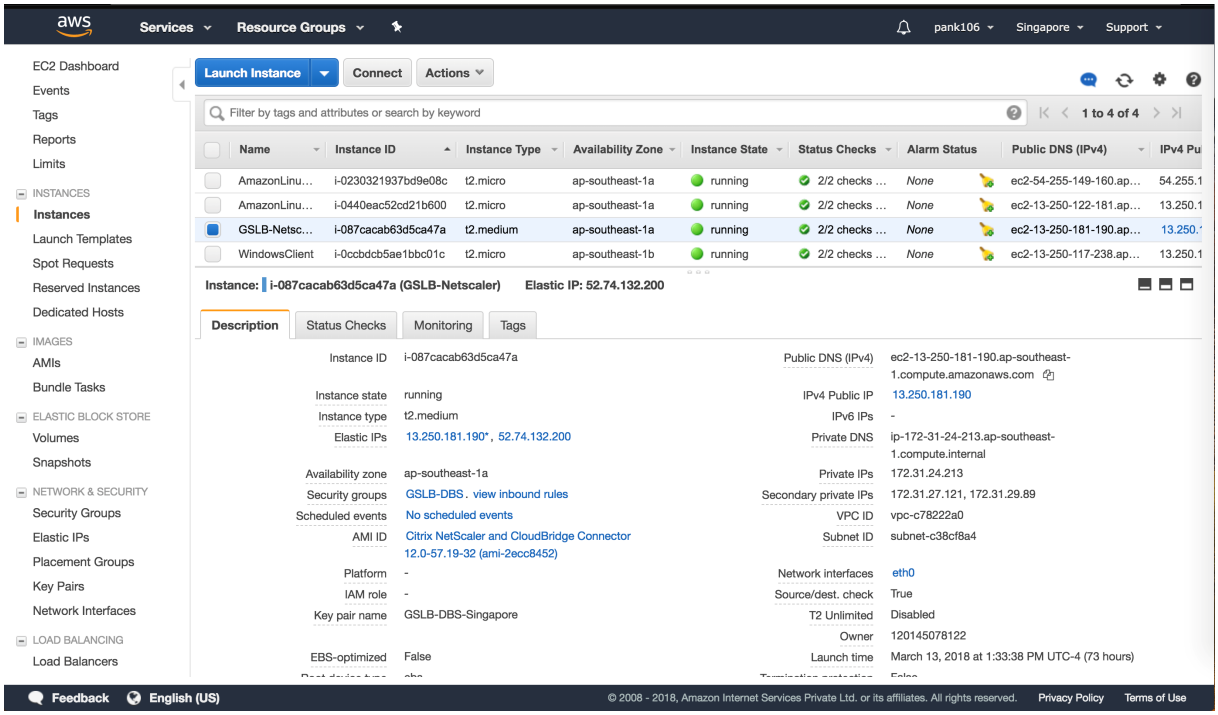
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/security-group-rules-reference.html#sg-rules-web-server

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	
DNS (UDP)	UDP	53	0.0.0.0/0	
DNS (UDP)	UDP	53	::/0	
Custom TCP Rule	TCP	3389	0.0.0.0/0	
Custom TCP Rule	TCP	3389	::/0	
All ICMP - IPv4	All	N/A	0.0.0.0/0	
All ICMP - IPv4	All	N/A	::/0	
Custom TCP Rule	TCP	5985	0.0.0.0/0	
Custom TCP Rule	TCP	5985	::/0	
Custom TCP Rule	TCP	3008 - 3011	0.0.0.0/0	
Custom TCP Rule	TCP	3008 - 3011	::/0	

Amazon Linux Backend Web Services

步骤 4：

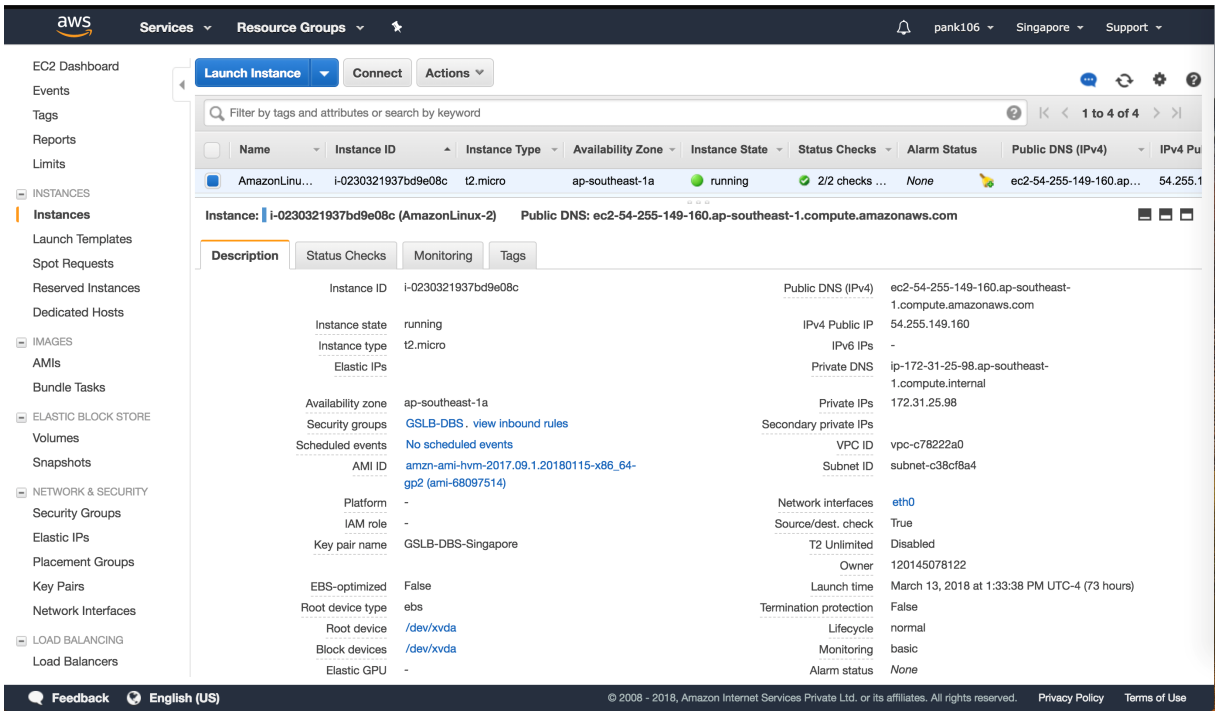
登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 内导航到实例。



步骤 5：

使用以下详细信息单击启动实例配置 **Amazon Linux** 实例。

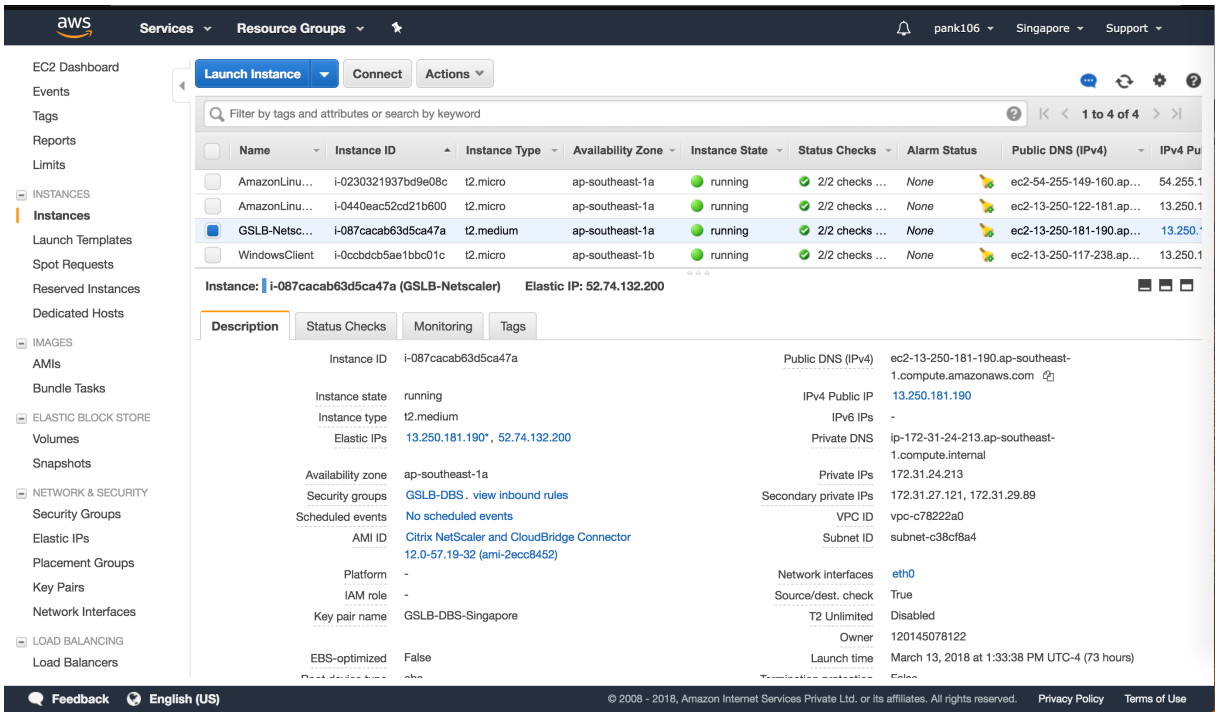
填写有关在此实例上设置 Web 服务器或后端服务的详细信息。



Citrix ADC 配置

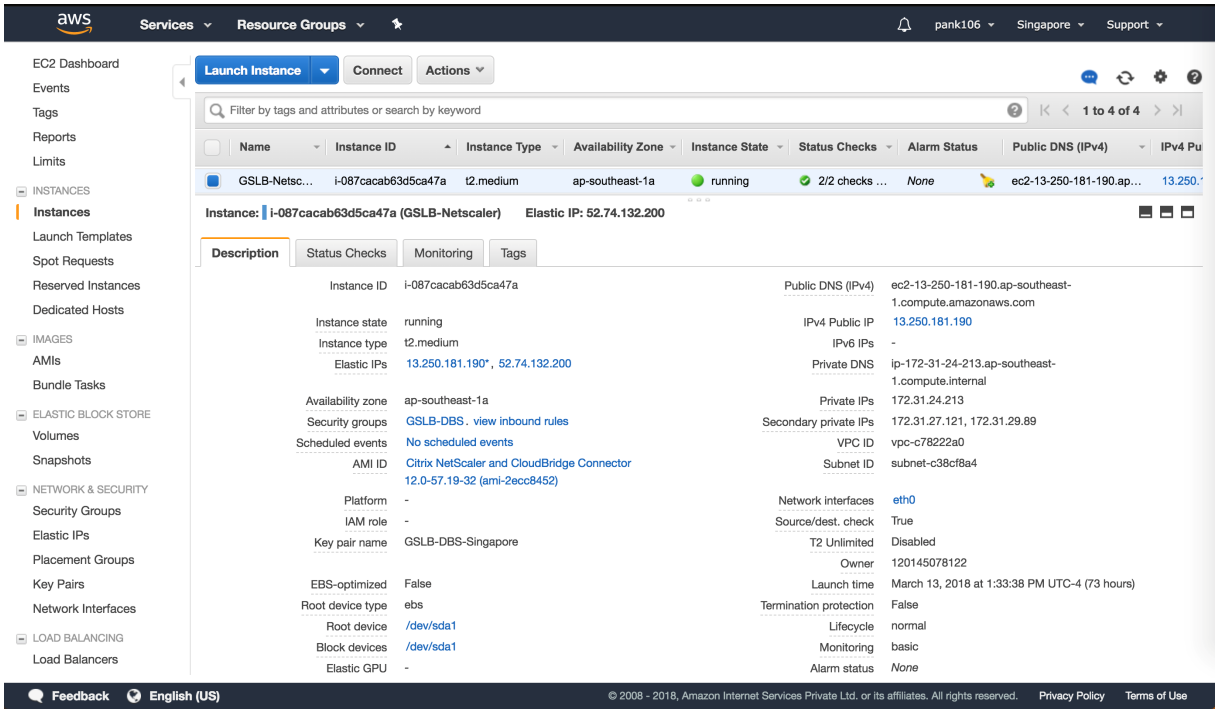
步骤 6：

登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 内导航到实例。



步骤 7：

使用以下详细信息单击启动实例配置 **Amazon AMI** 实例。



弹性 **IP** 配置

> 注意：
>
> 如果需要，Citrix ADC 还可以通过不具有 NSIP 的公共 IP 来降低成本，使其能够使用单个弹性 IP 运行。相反，将弹性 IP 附加到 SNIP，可以覆盖管理访问框，以及 GSLB 站点 IP 和 ADNS IP。

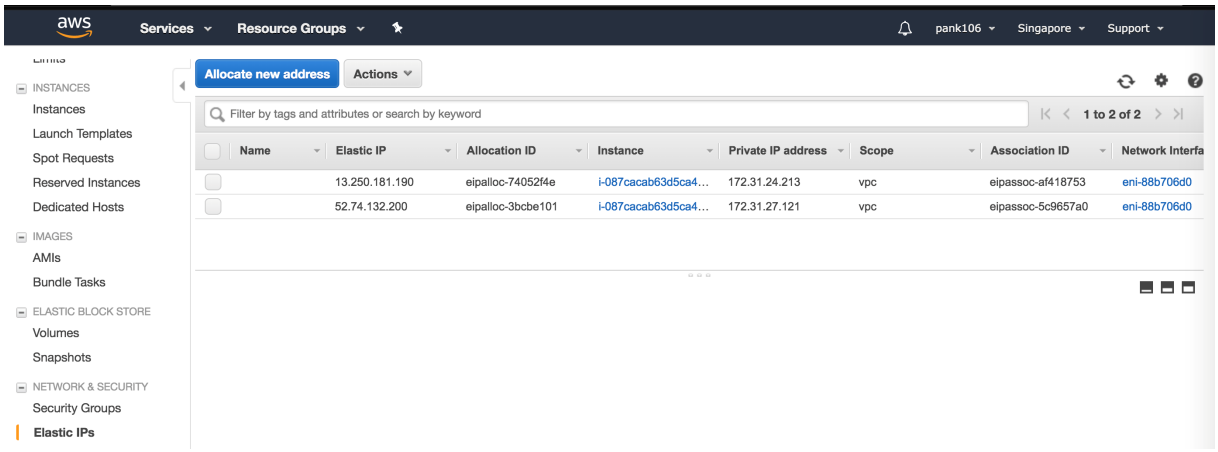
步骤 8：

登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 中导航到网络和安全，然后配置弹性 **IP**。

单击“分配新地址”以创建新的弹性 IP 地址。

将弹性 IP 配置为指向 AWS 中正在运行的 Citrix ADC 实例。

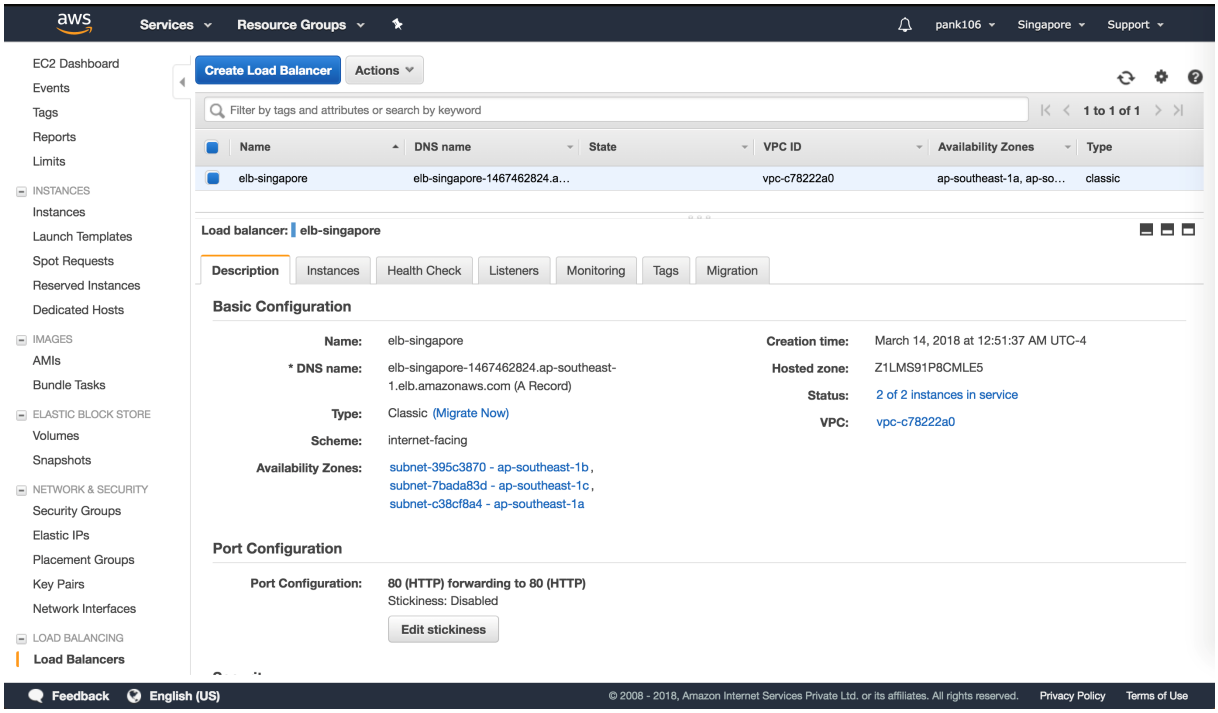
配置第二个弹性 IP，然后再次将其指向正在运行的 Citrix ADC 实例。



弹性负载平衡器

步骤 9：

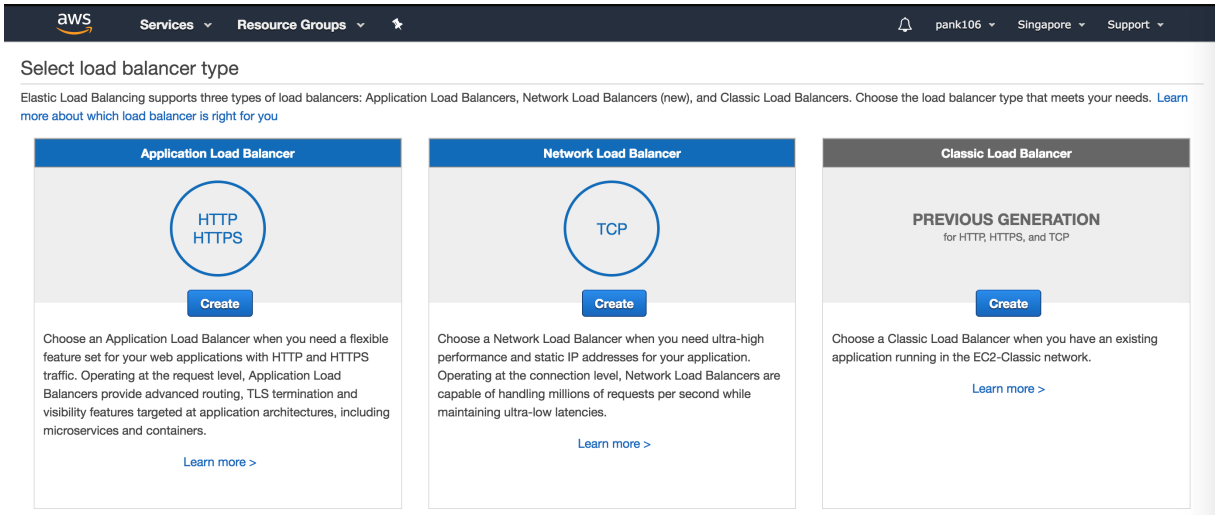
登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 中导航到负载平衡，然后导航到负载平衡器。



步骤 10：

单击 创建负载平衡器以配置传统负载平衡器

您的弹性负载平衡器允许您对后端 Amazon Linux 实例进行负载均衡，同时还能够根据需求对其他实例进行负载均衡。



配置全局服务器负载平衡基于域名的服务

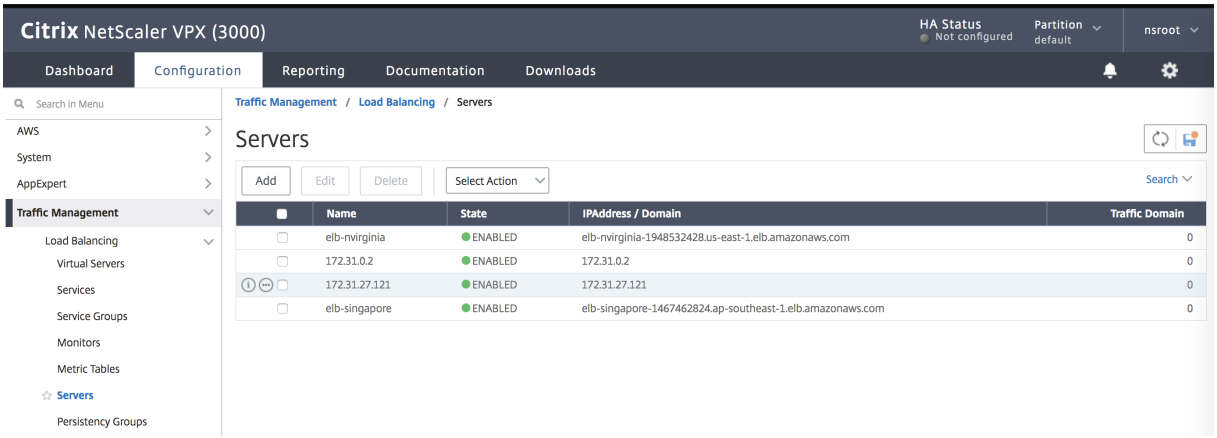
流量管理配置

> 注意：
>
> 需要使用名称服务器或 DNS 虚拟服务器配置 Citrix ADC，通过该服务器可以解析 DBS 服务组的 ELB/ALB 域。

https://developer-docs.citrix.com/projects/netscaler-command-reference/en/12.0/dns/dns-nameserver/dns-nameserver/

步骤 1:

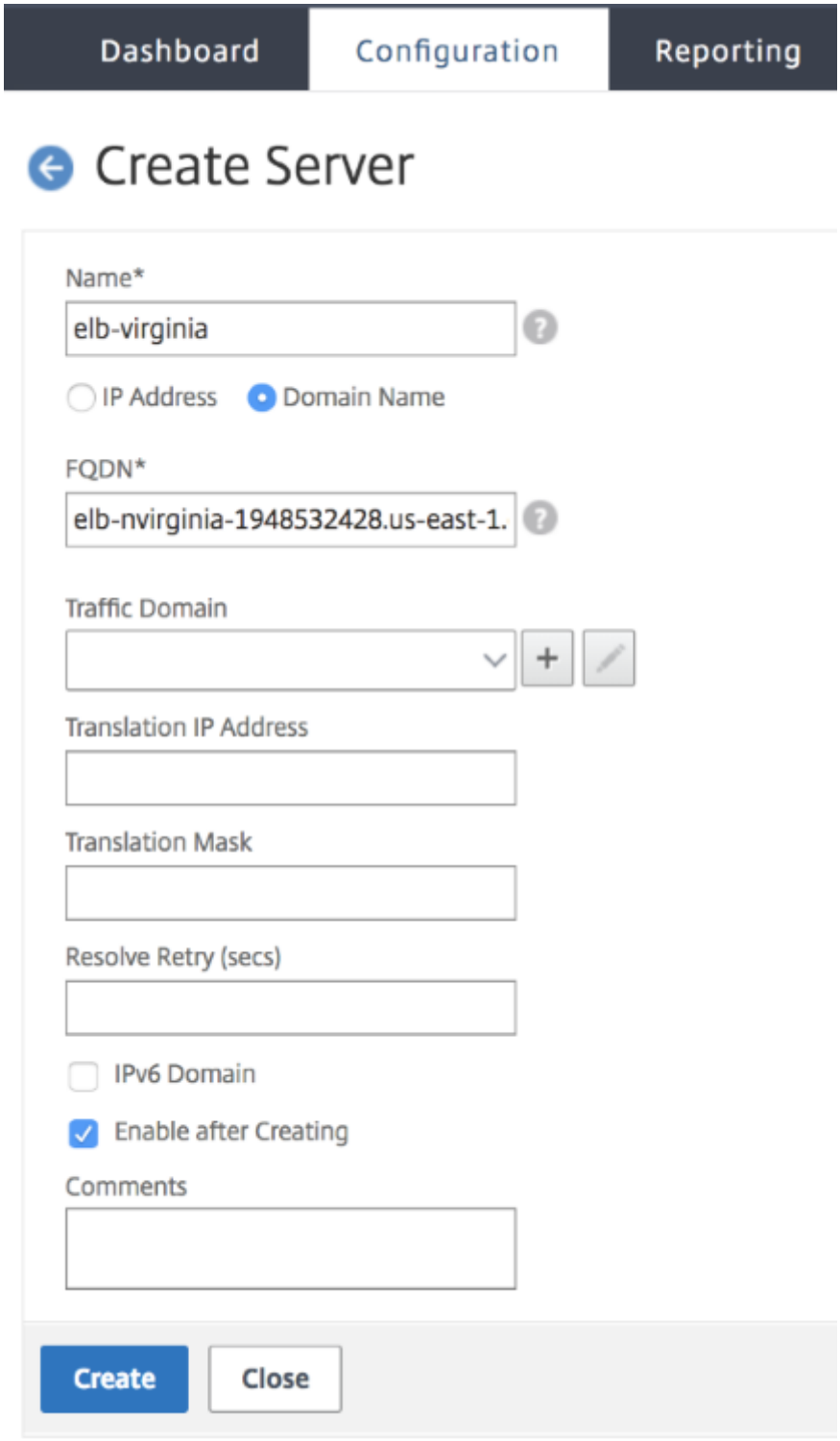
导航到流量管理 **>** 负载平衡 **>** 服务器。



步骤 2:

单击“添加”创建服务器，并为弹性负载平衡器 (ELB) 提供与 AWS 中的 A 记录（域名）相对应的名称和 FQDN。

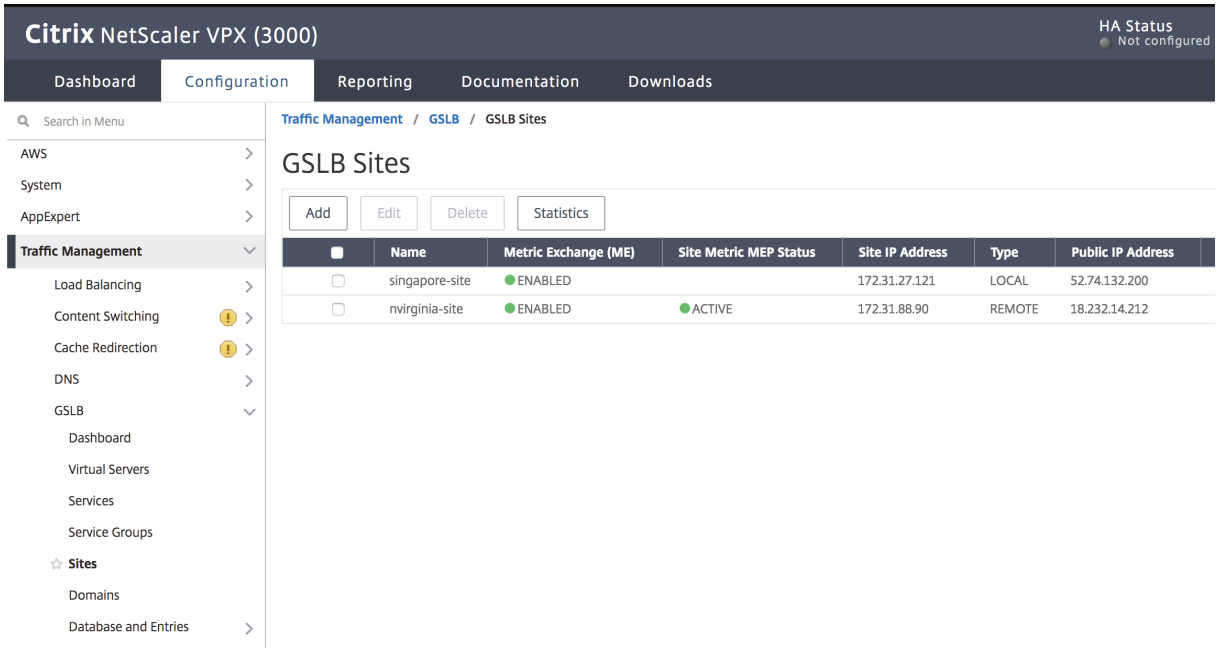
重复步骤 2，从 AWS 中的第二个资源位置添加第二个 ELB。



GSLB 配置

步骤 1:

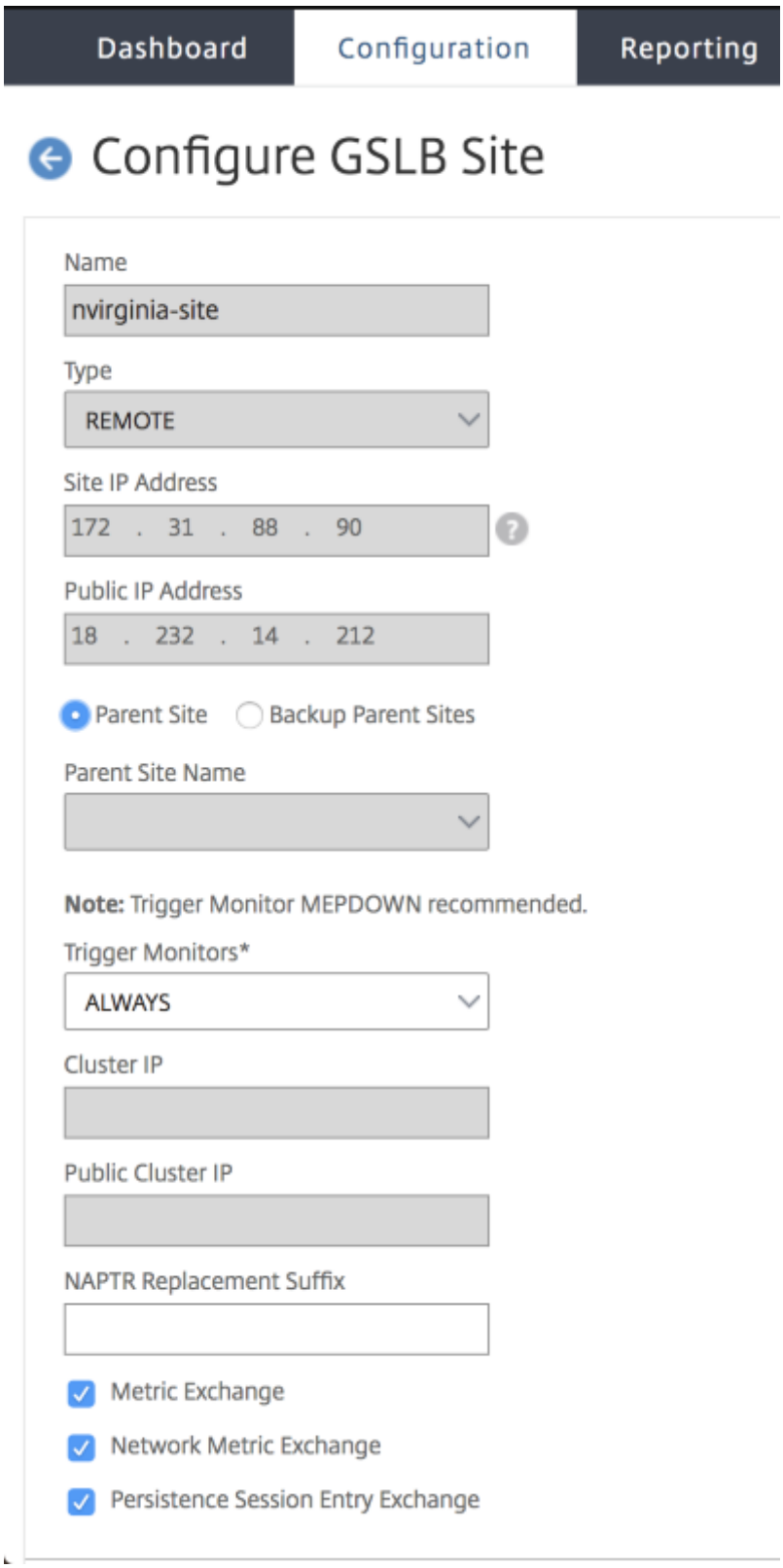
导航到流量管理 **> GSLB >** 站点。



步骤 3：

单击添加按钮以配置 GSLB 站点。

为站点命名。类型配置为“远程”或“本地”，具体取决于要在其上配置站点的 Citrix ADC。站点 IP 地址是 GSLB 站点的 IP 地址。GSLB 站点使用此 IP 地址与其他 GSLB 站点进行通信。在使用某个特定 IP 托管在外部防火墙或 NAT 设备上的云服务时，需要公有 IP 地址。该站点应配置为父站点。确保触发器监视器设置为“始终”，并确保勾选“衡量指标交换”、“网络衡量指标交换”和“持久性会话条目交换”底部的三个复选框。

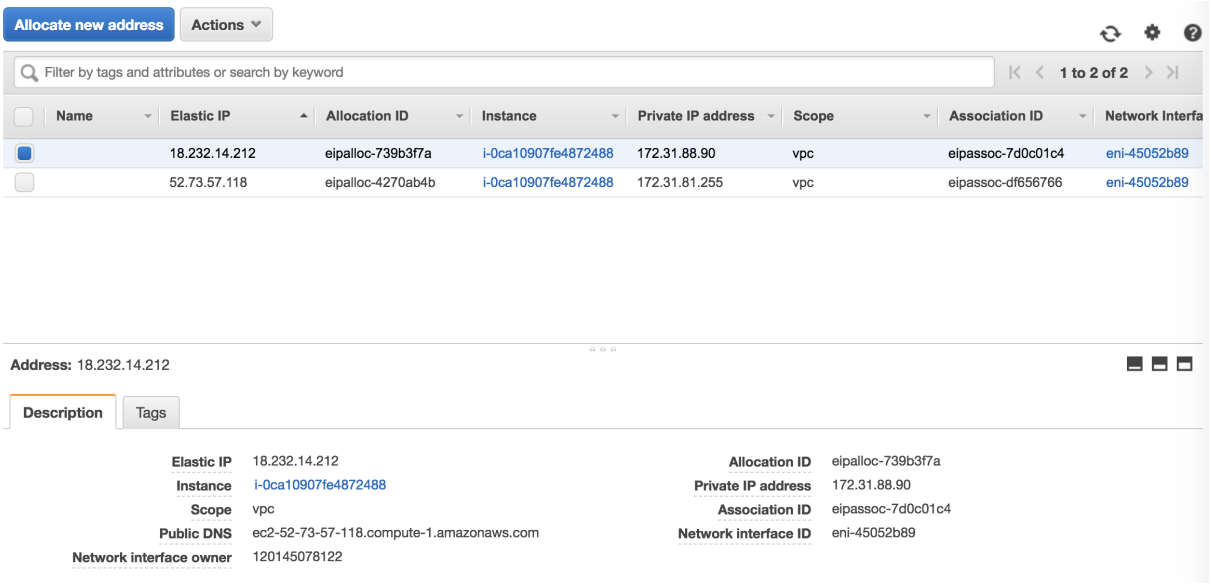


建议将触发器监视器设置设置为 MEPDOWN。有关详细信息，请参阅配置 GSLB 服务组。

步骤 4：

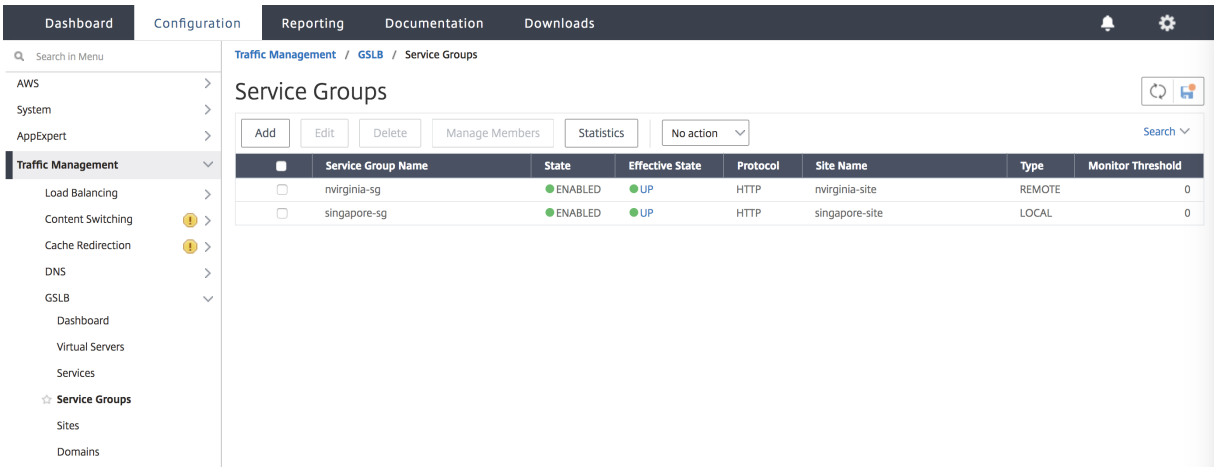
以下是我们 AWS 配置的屏幕截图，显示了您可以在哪里找到站点 IP 地址和公有 IP 地址。它们位于“网络和安全**”>“**弹性 **IP”** 下。

单击“创建”，重复步骤 3 和步骤 4，为 Azure 中的其他资源位置配置 GSLB 站点（这可以在同一 Citrix ADC 上进行配置）



步骤 5：

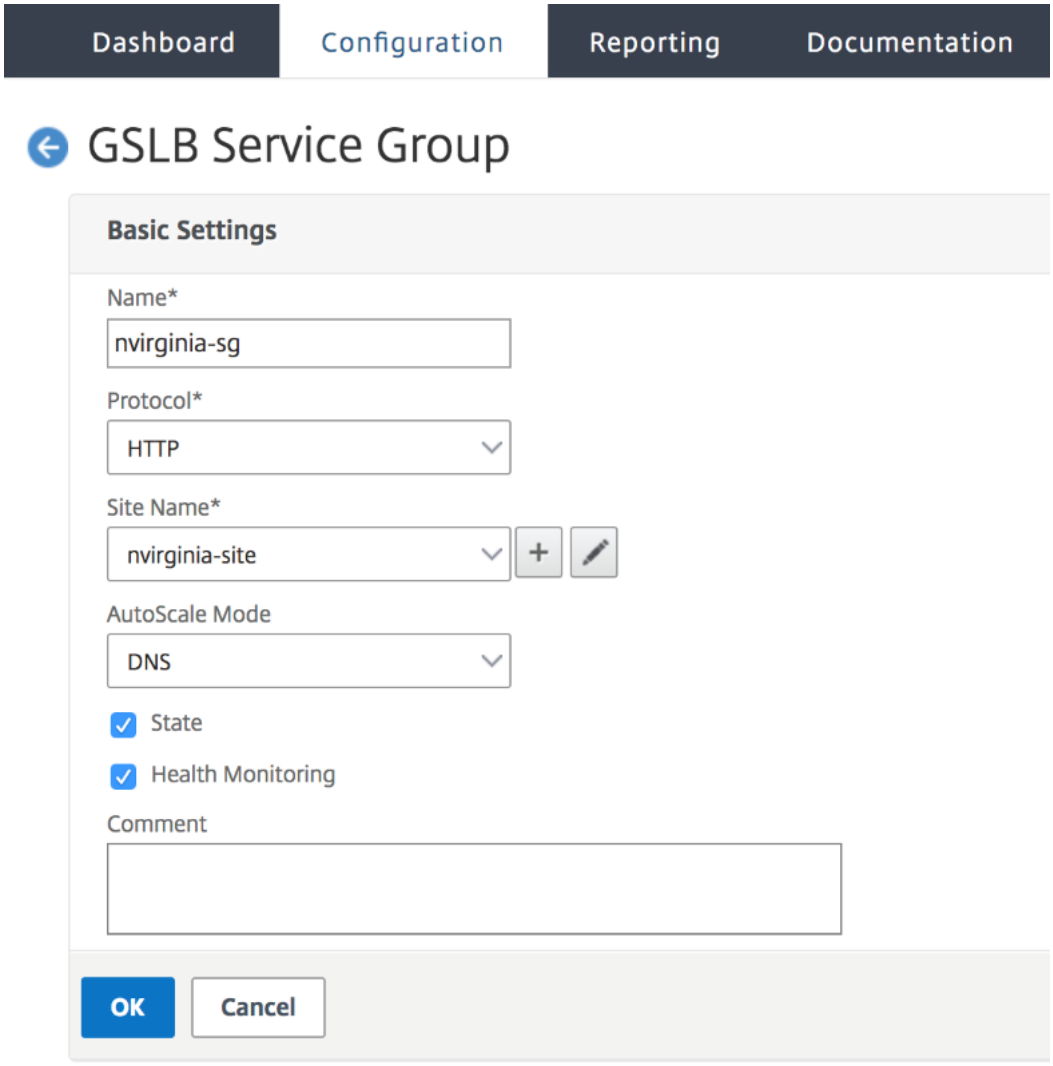
导航到流量管理 **> GSLB >** 服务组



步骤 6：

单击“添加”以添加新的服务组。命名服务组，使用 HTTP 协议，然后在“站点名称”下选择在前面的步骤中创建的相应站点。请务必将自动缩放模式配置为 DNS，并勾选“状态”和“运行状况监视”对应的复选框。

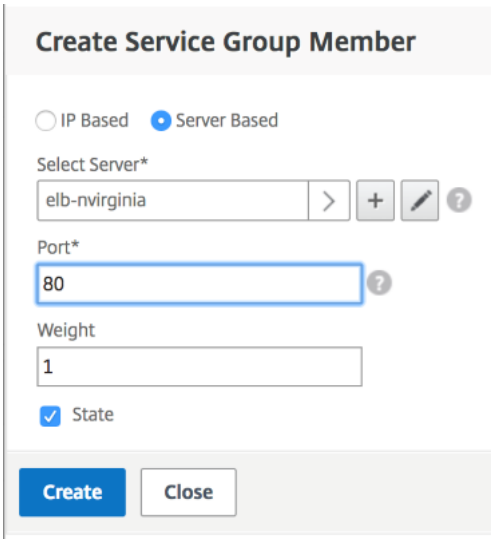
单 击 **“**确定”以创建服务组。



步骤 7：

单击“服务组成员”并选择“基于服务器”。选择在运行指南开始部分配置的各自弹性负载平衡服务。将流量配置为通过端口 80。

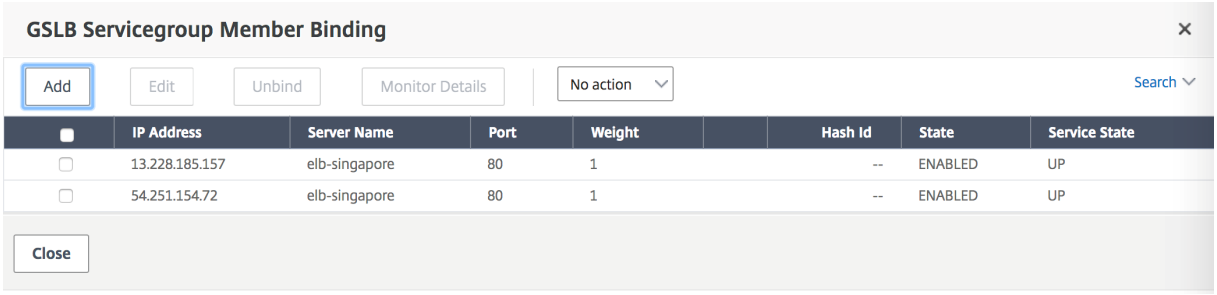
单击创建。



步骤 8：

服务组成员绑定应填充从弹性负载平衡器接收的两个实例。

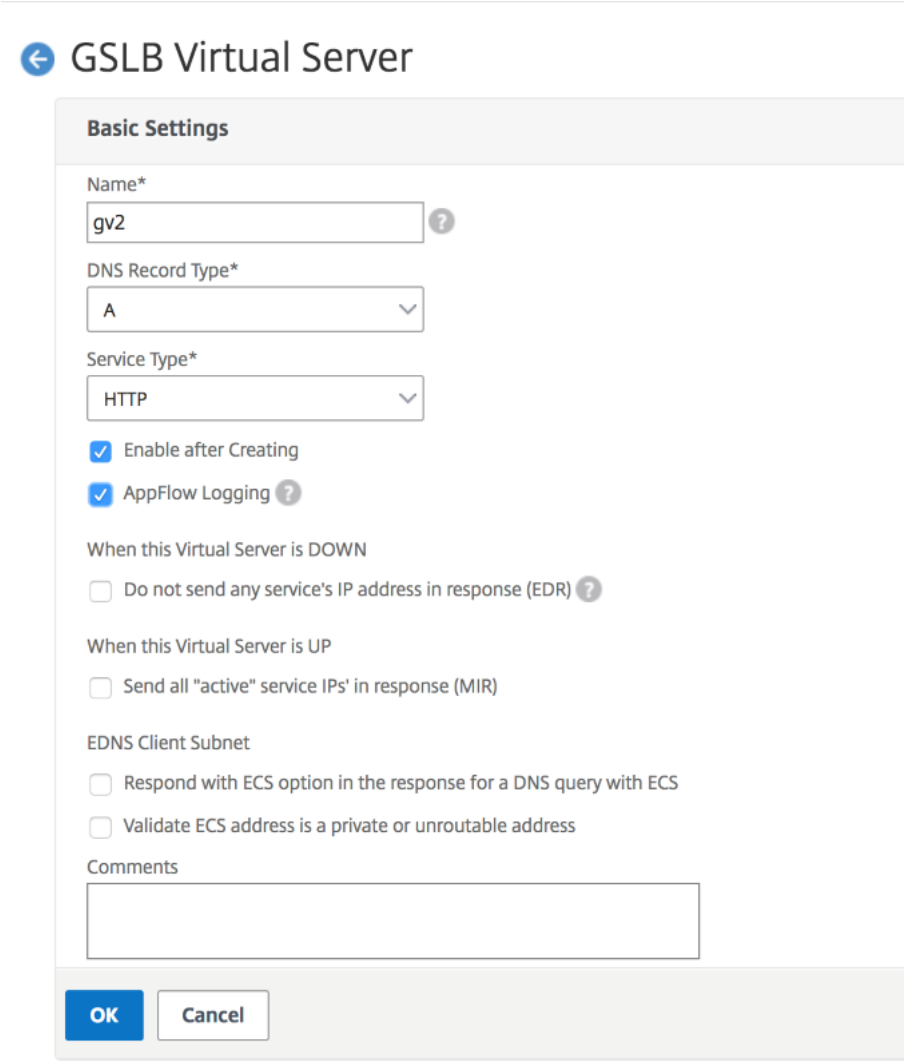
重复步骤，为 AWS 中的第二个资源位置配置服务组。（这可以从同一位置完成）。



步骤 9：

导航到流量管理 **> GSLB >** 虚拟服务器。

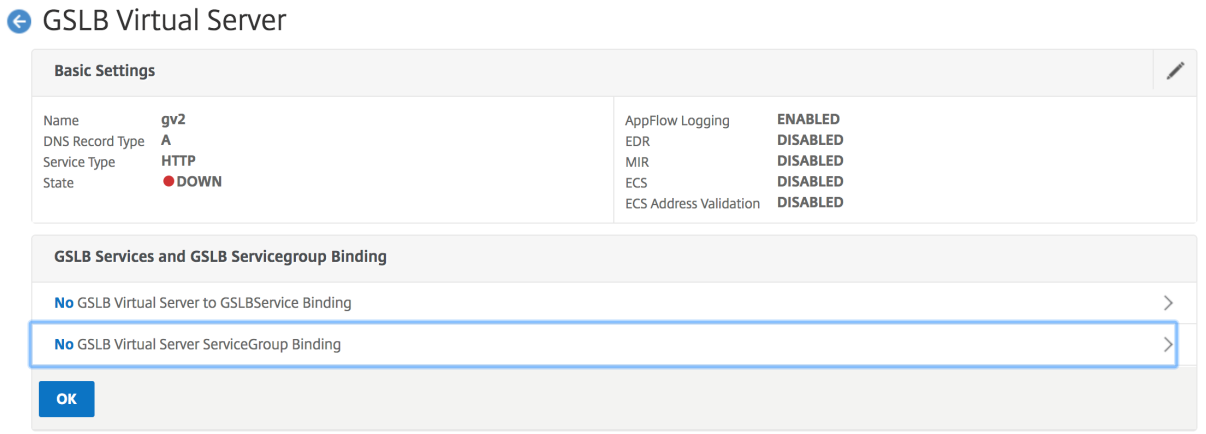
单击“添加”以创建虚拟服务器。命名服务器，将 DNS 记录类型设置为 A，服务类型设置为 HTTP，并选中“创建后启用”和“AppFlow 日志记录”复选框。单击确定以创建 GSLB 虚拟服务器。(Citrix ADC GUI)



步骤 10：

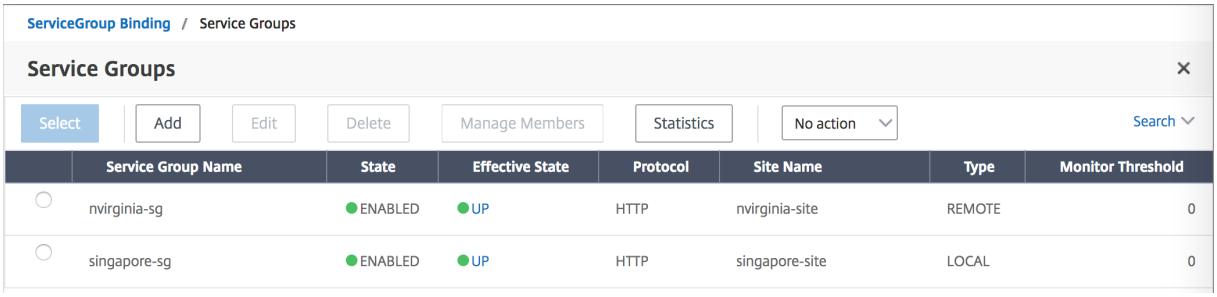
创建 GSLB 虚拟服务器时，单击无 **GSLB** 虚拟服务器服务组绑定。

单击“添加”以创建虚拟服务器。命名服务器，将 DNS 记录类型设置为 A，服务类型设置为 HTTP，并选中“创建后启用”和“AppFlow 日志记录”复选框。单击确定以创建 GSLB 虚拟服务器。(Citrix ADC GUI)



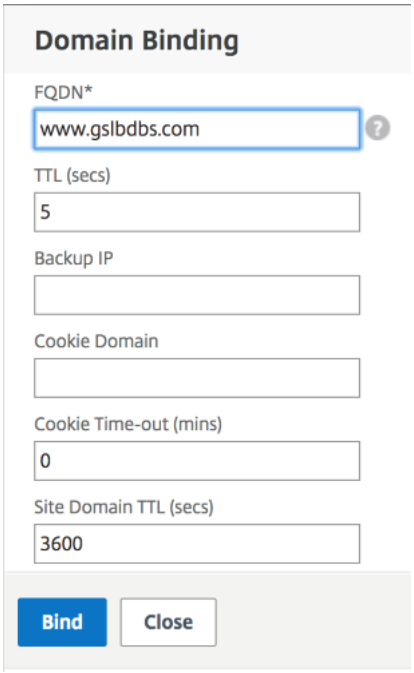
步骤 11：

在“服务组绑定”下，使用“选择服务组名称”选择并添加在上述步骤中创建的服务组。



步骤 12：

接下来，通过单击无 **GSLB** 虚拟服务器域绑定配置 GSLB 虚拟服务器域绑定。配置 FQDN 和绑定，其余设置可保留为默认值。

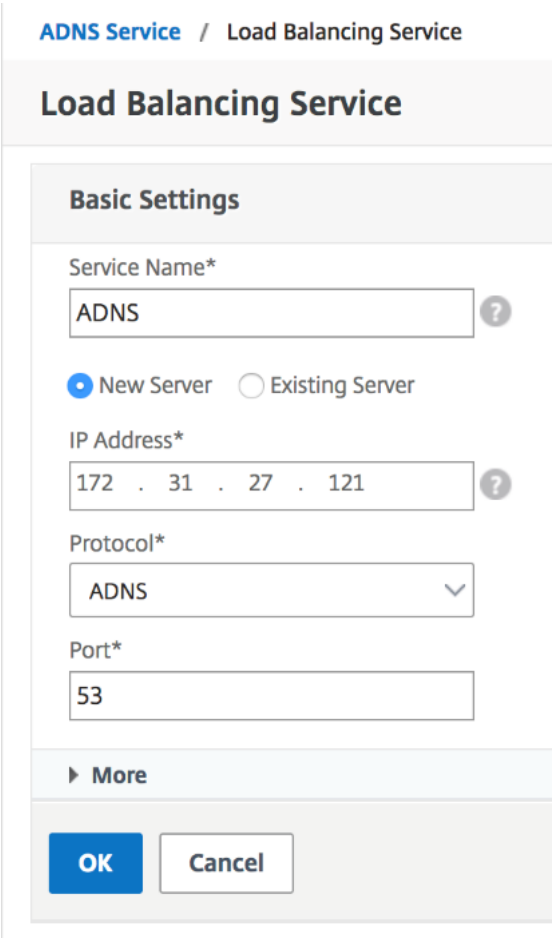


步骤 13：

通过单击“无服务”来配置 **ADNS** 服务。添加服务名称，单击“新建服务器”，然后输入 ADNS 服务器的 IP 地址。

此外，如果您的 ADNS 已配置，您可以选择“现有服务器”，然后从菜单中选择您的 ADNS。确保协议是 ADNS，并且流量通过端口 53。

将方法配置为 LEASTCONNECTION，并将备份方法配置为 ROUNDROBIN



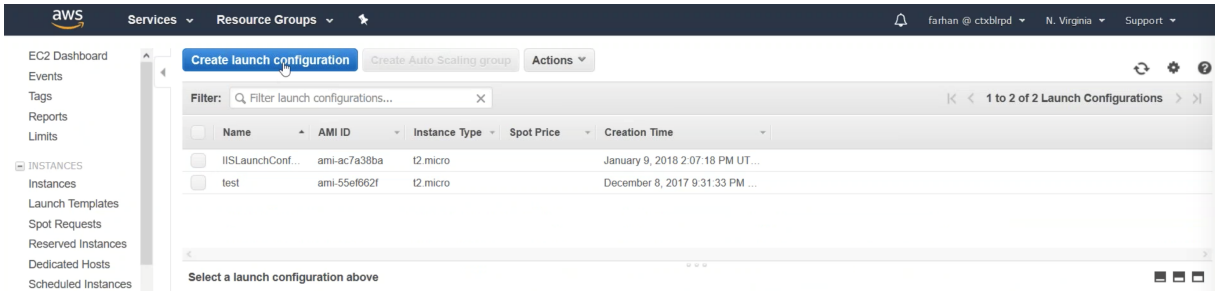
使用 **AWS** 的 **Citrix ADC** 后端 **Auto Scaling**

AWS 包括一项名为 Auto Scaling 的功能，该功能可根据管理员设置的规则在 AWS 中运行的其他实例进行旋转。这些规则由 CPU 利用率定义，围绕按需创建和删除实例。Citrix ADC 直接与 AWS Auto Scaling 解决方案集成，使 Citrix ADC 了解所有可用的后端服务器可以平衡负载。此功能的局限性在于它目前仅在 AWS 中的一个可用区内运行。

配置 **AWS** 组件

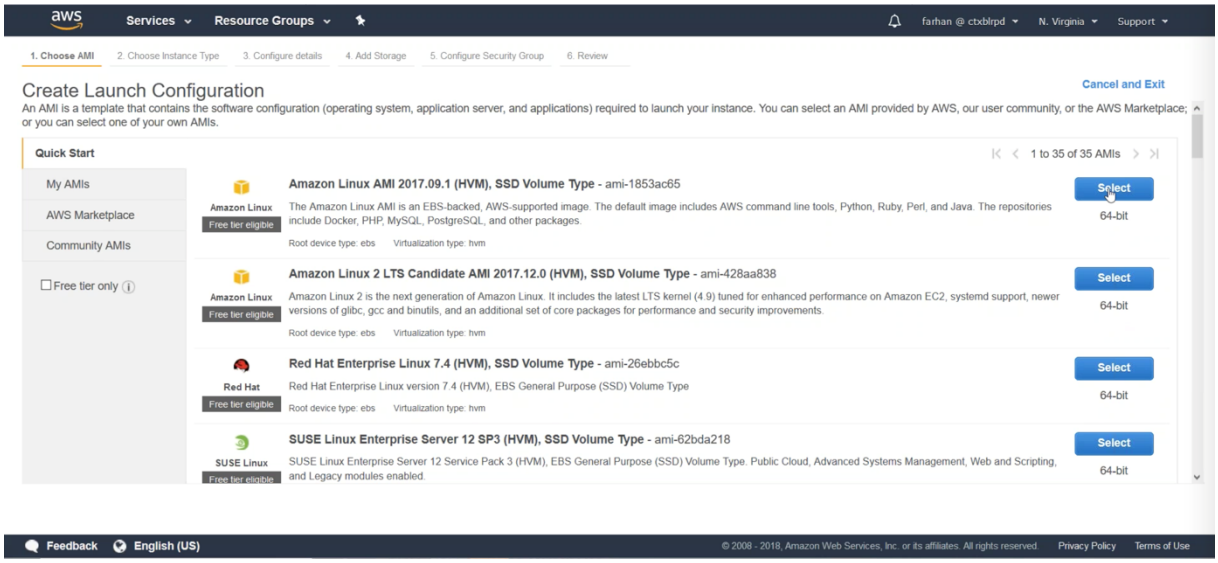
步骤 1:

登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 中导航到 **AUTO SCALING >** 启动配置。单击 创建启动配置。



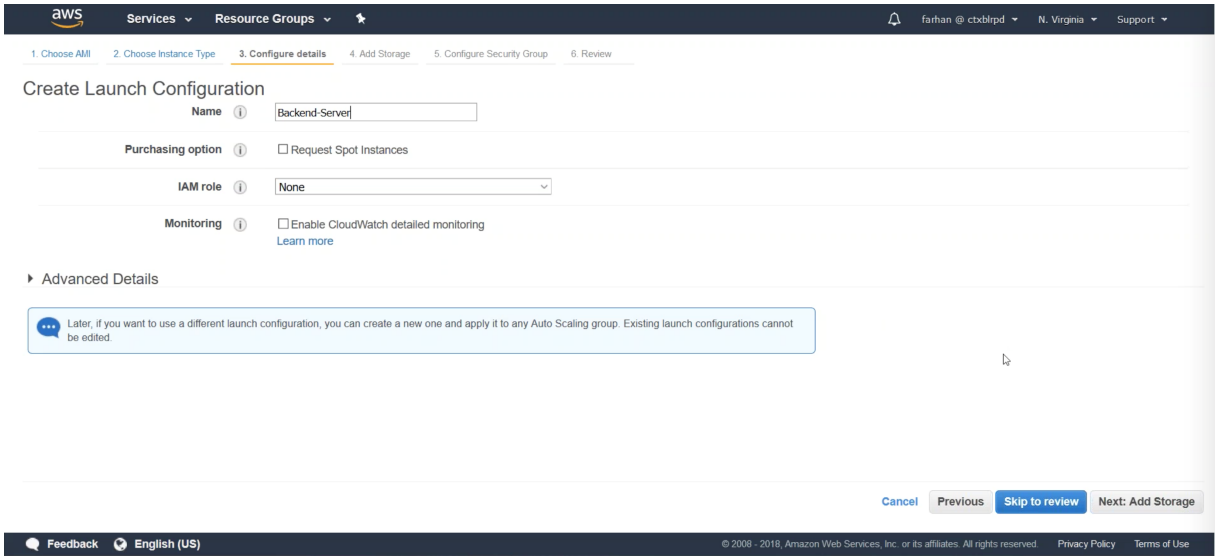
步骤 2:

在此步骤中，您可以选择您选择的服务器类型。可以在此处配置要自动缩放的 VM。在此示例中，我们必须选择 **Amazon Linux AMI**。



步骤 3:

通过从后端资源的潜在差异中选择您需要的实例类型。为运行指南的剩余部分命名您的实例。实例的名称称为后端服务器。为实例配置存储并将其添加到安全组，或者创建包含本运行指南中创建的所有 AWS 组件的新安全组。



步骤 4:

安全组的附加说明。对于本运行指南，以下打开的端口：

Type	Protocol	Port Range	Source
All traffic	All	All	0.0.0.0/0
SSH	TCP	22	185.25.64.249/32
SSH	TCP	22	125.16.224.135/32

Citrix ADC Back End Auto Scaling 组和策略

在 AWS 中配置 Citrix ADC 前端 Auto Scaling:

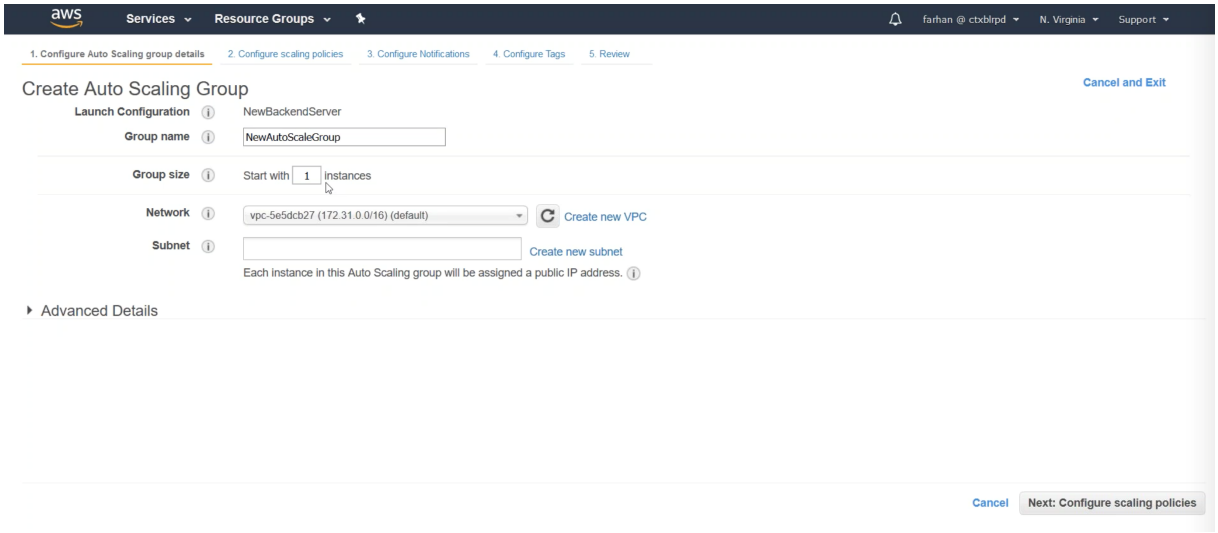
步骤 1:

登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 中导航到 **AUTO SCALING > Auto Scaling** 组。

单击单选按钮以根据现有启动配置创建 Auto Scaling 组。请务必选择我们在实验指南的前一步中创建的后台服务器。

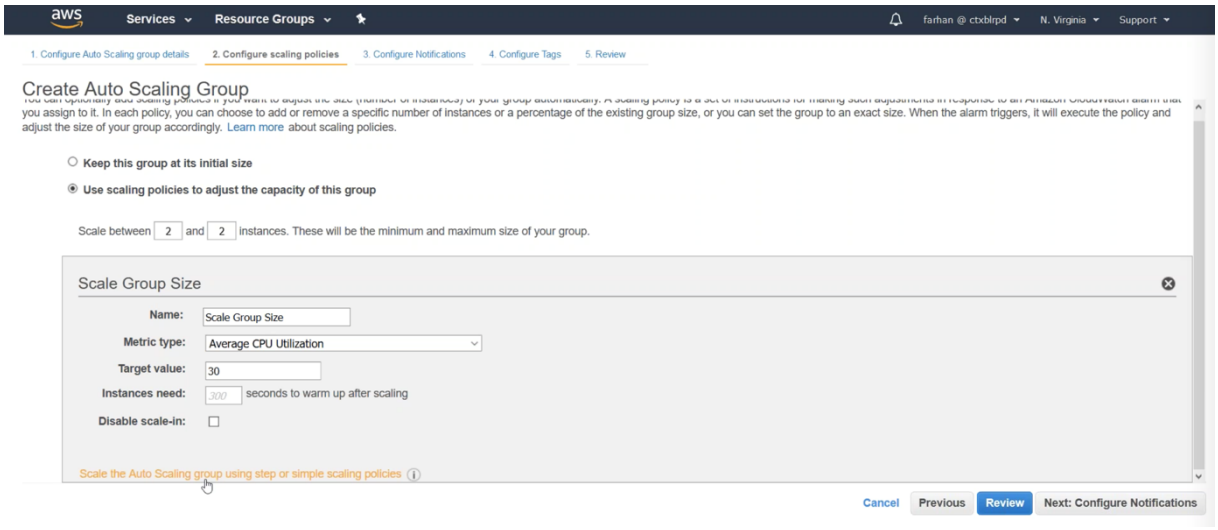
在创建 **Auto Scaling** 组下添加组名称，选择初始组大小，选择网络和子网，
然后单击下一步。

> 注意:
>
> 子网必须可以从 Citrix ADC 的子网 IP (SNIP) 访问。



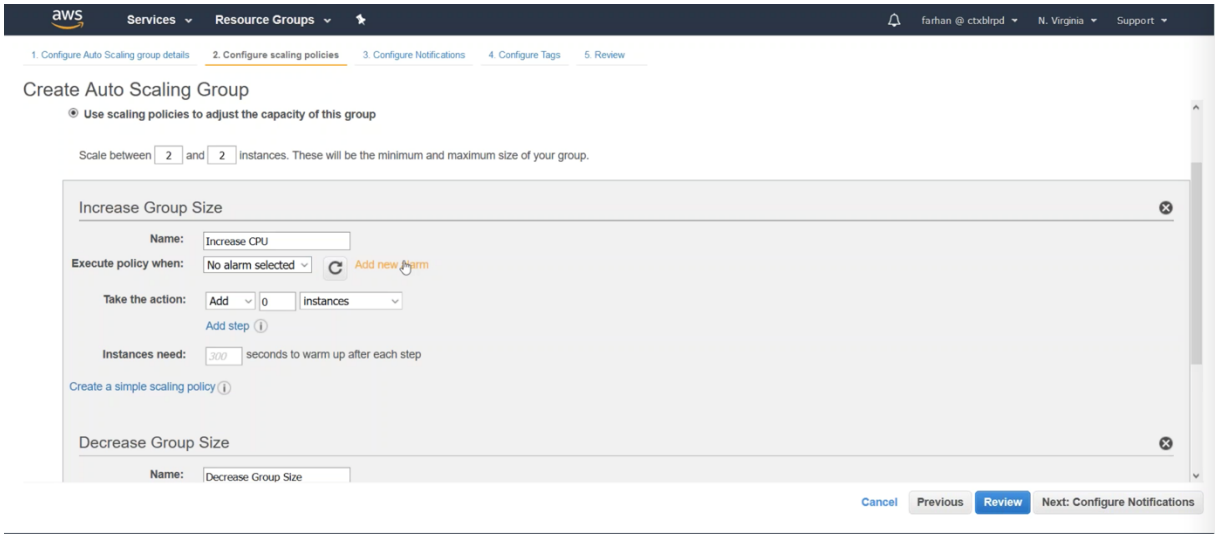
步骤 2:

在创建 **Auto Scaling** 组配置页面上，配置您的缩放策略。您可以通过单击使用扩展策略调整此组容量的单选按钮来完成此操作。下一步，单击使用步骤或简单缩放策略的缩放 **Auto Scaling** 组。



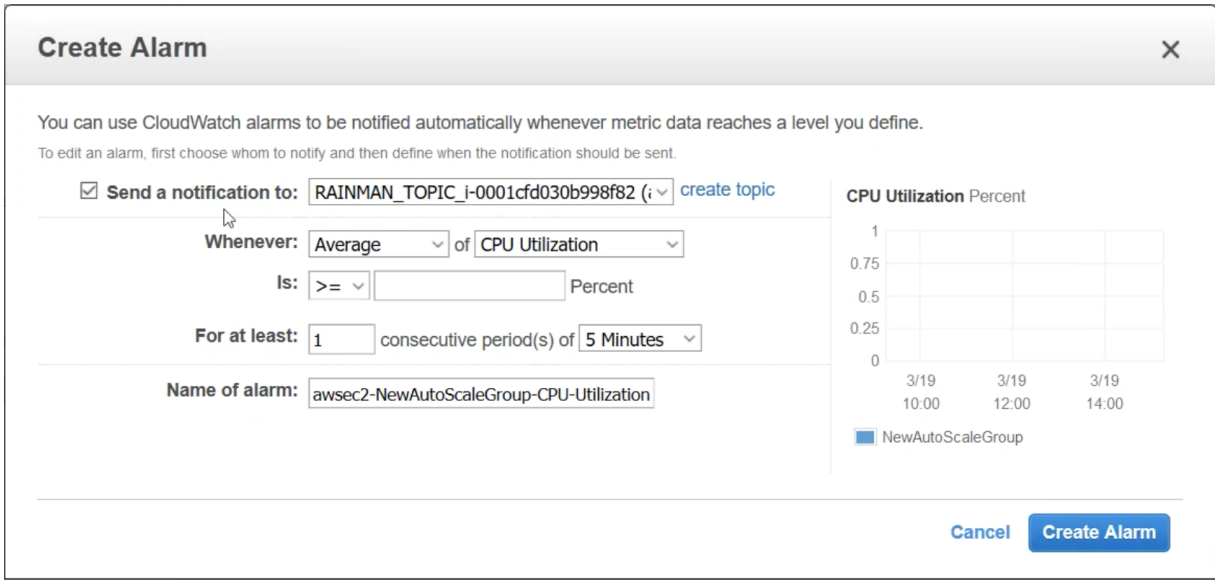
步骤 3:

选择“添加新警报”。



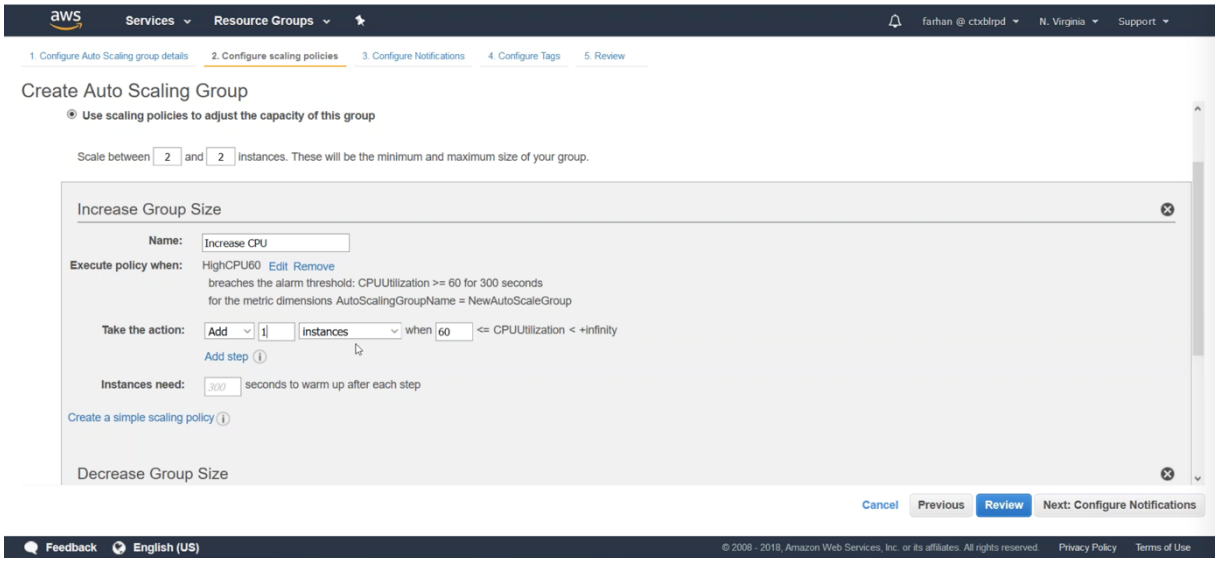
步骤 4：

在创建警报时，配置为向 Citrix ADC 发送通知。配置警报，使 CPU 平均利用率为 >= 70，连续至少一个 5 分钟的时间段。应用策略。



步骤 5：

在 Auto Scaling 组中配置以便在触发策略时添加一个实例。



步骤 6：

配置相同的警报和策略，但这次是在 CPU 平均小于 30 时间 5 分钟时删除后端服务器。在触发减少策略时，将缩小组大小设置为“删除 1 实例”。

> 注意：
>
> 对于删除服务器，我们通知 Citrix ADC 不向标记为删除的后端服务器发送任何流量。

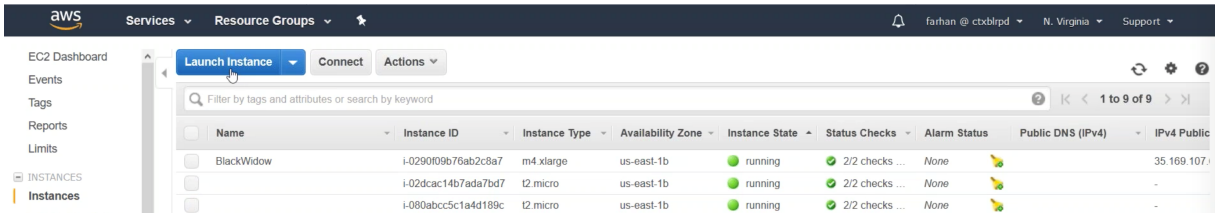
单击“配置通知”和“配置标签”以查看和创建 Auto Scaling 组。

> 注意：
>
> 可以将最小和最大变量配置为设置在 Auto Scaling 组内创建和运行的最少和最高实例数。目前，AWS 支持仅使用一个网络接口启动额外的实例。

在 **AWS** 中创建 **Citrix ADC**

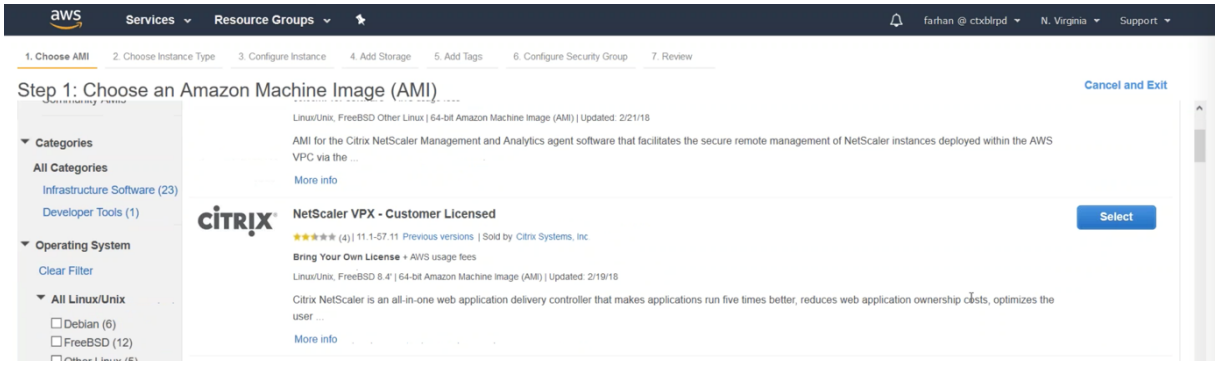
步骤 1:

登录您的 **AWS** 资源组并导航至 **EC2**。在 EC2 中导航到“实例**”>“**实例**”**。



步骤 2:

导航至左侧的 AWS 市场，然后搜索 Citrix ADC。选择 **Citrix Networking VPX — 客户许可。确保您的版本号为 12.0.51.x，以使用 Auto Scaling。您可以选择早期版本以选择支持 Auto Scaling 的 Citrix ADC 版本。



步骤 3：

导航至左侧的 AWS 市场，然后搜索 Citrix ADC。选择 **Citrix Networking VPX — 客户许可**。确保您的版本号为 12.0.51.x，以使用 Auto Scaling。您可以选择早期版本以选择支持 Auto Scaling 的 Citrix ADC 版本。

选择实例类型，例如通用型 m4.xlarge 4vCPU 和 16 GB RAM。单击下一步。

步骤 4：

在配置实例详细信息选项卡上，选择子网（最终必须为 NSIP、SNIP 和 VIP/网关配置三个子网）。此外，您还必须添加 IAM 角色。单击以创建新的 IAM 角色。添加在以下步骤中找到的 IAM 角色。创建此角色后，您需要将其添加到 Citrix ADC 上的云配置文件中。

步骤 5：

云配置文件的配置如下：

默认情况下，CloudFormation 模板会创建和附加以下 IAM 角色

```
”Version”: ”2012-10-17”,
”Statement”: [
   {

       ”Action”: [
       ”ec2:DescribeInstances”,
       ”ec2:DescribeNetworkInterfaces”,
       ”ec2:DetachNetworkInterface”,
       ”ec2:AttachNetworkInterface”,
       ”ec2:StartInstances”,
       ”ec2:StopInstances”,
       ”ec2:RebootInstances”,
       ”autoscaling:\*”,
       ”sns:\*”,
       ”sqs:\*”
        “iam: SimulatePrincipalPolicy”
        “iam: GetRole”
```



```
18          ],
19          "Resource": "\*",
20          "Effect": "Allow"
21       }
22
23   ]
24   }
25
26 <!--NeedCopy-->
```

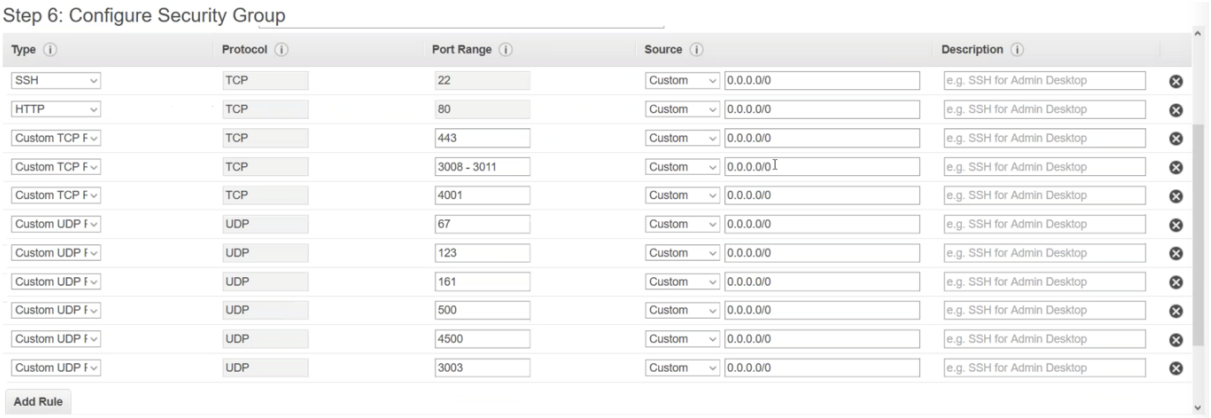
步骤 6：

单击添加存储选项。在添加标签选项卡上，将密钥值设置为名称，并将值设置为 Citrix ADC-Autoscale 以标记这些 EC2 资源。

步骤 7：

在“配置安全组”选项卡上，创建具有以下端口要求的新安全组：

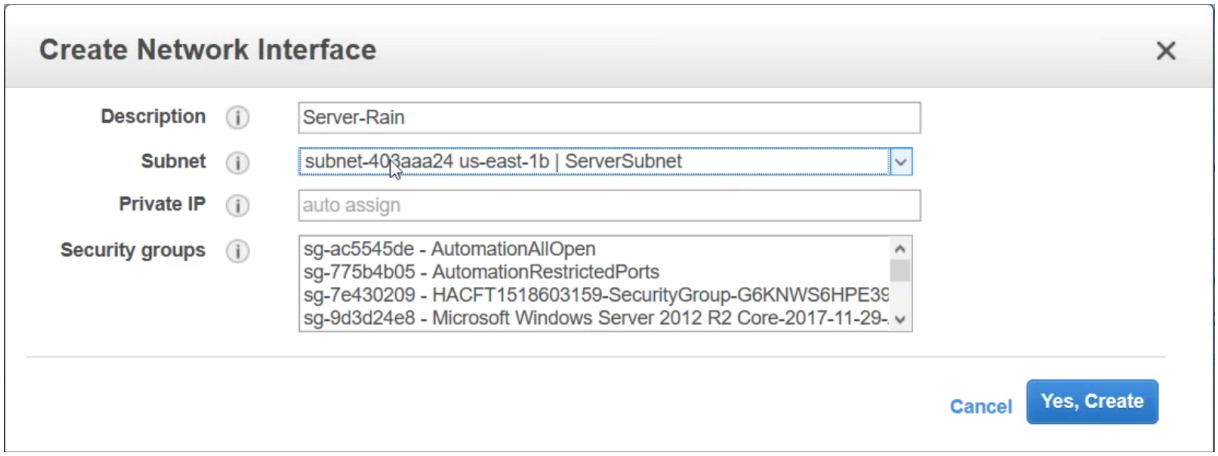
查看并启动实例。



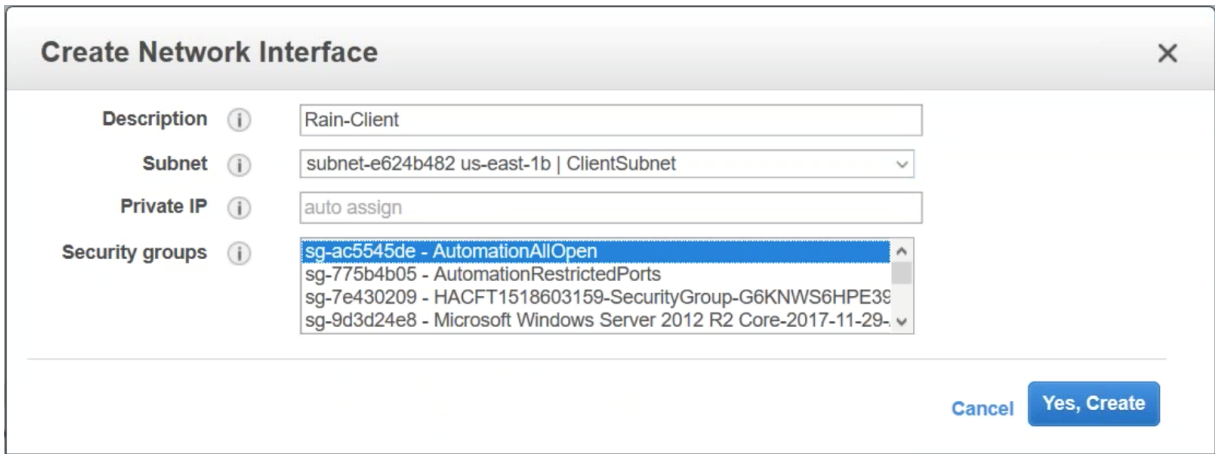
步骤 8：

导航到“网络和安全**”>“**网络接口”，然后单击“创建网络接口”。

添加描述，然后选择子网。此子网用于您的 SNIP，因此应将其放置在内部网络的子网中。此外，选择上一步中填入的安全组。单击 **Yes, Create**（是，创建）。



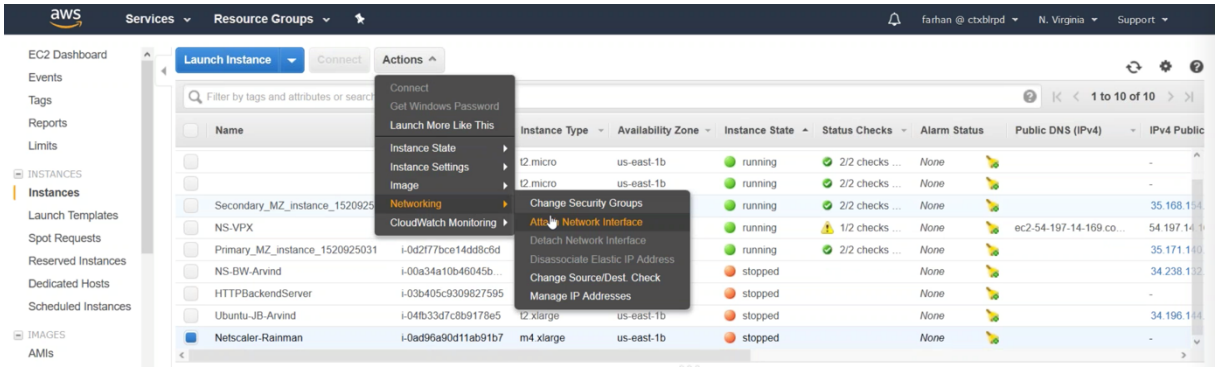
添加其他网络接口。这是网关/LB VIP 的面向公用的子网。创建描述并选择上面配置的安全组。



步骤 9：

导航回实例并选择您的 Citrix ADC。要将网络接口添加到 Citrix ADC，必须停止实例。在操作列表中，选择实例状态，然后单击停止。

再次单击“操作”按钮，然后向下导航到“网络”和“附加网络接口”。



NSIP 接口已连接到 VM，下一个要添加的接口应该是 LB-VIP，然后添加 SNIP 的服务器/内部接口。连接网络接口后，实例就可以启动。

配置新的弹性 IP 并将其与您的 NSIP 接口关联。

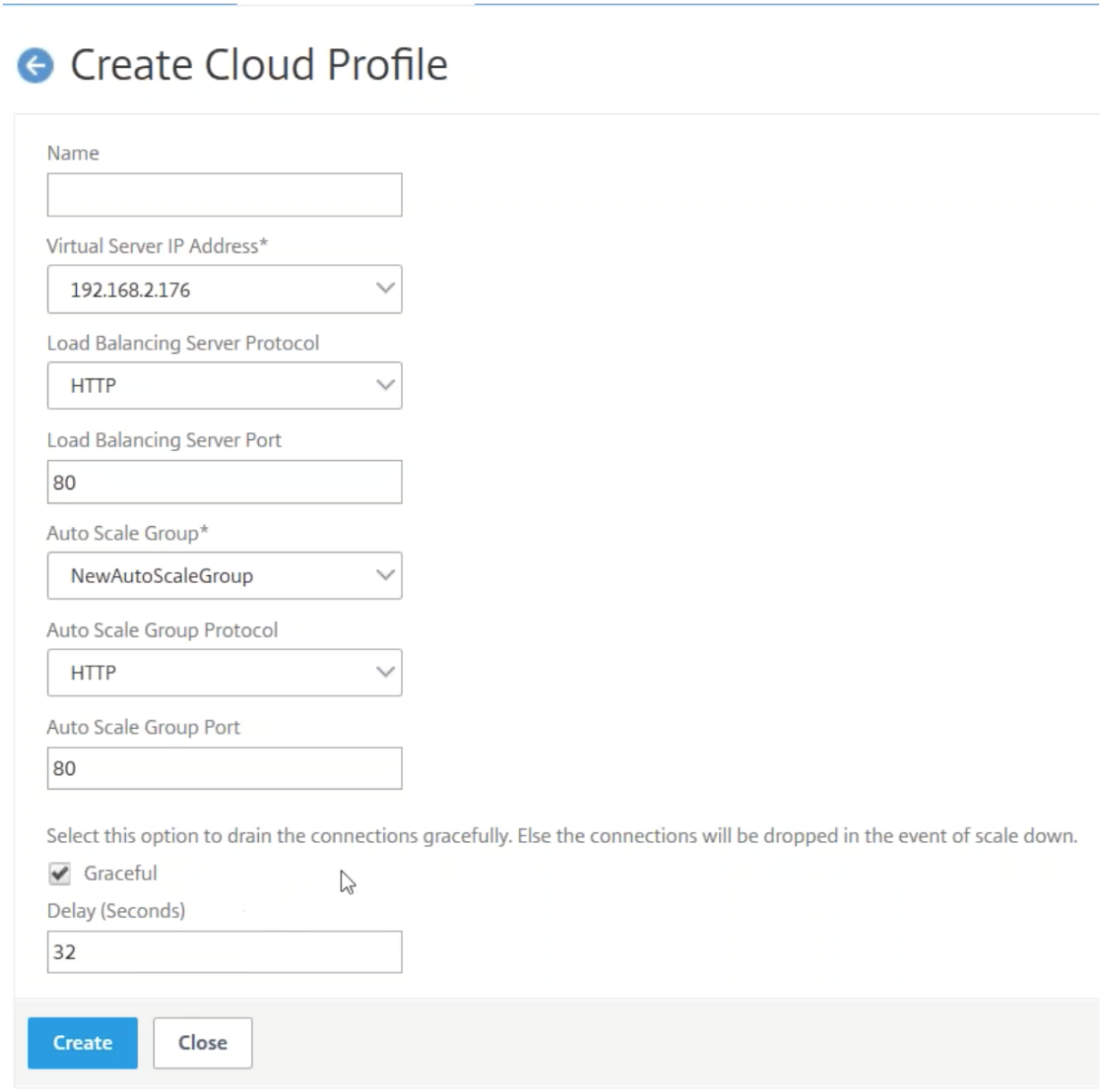
配置 **Citrix ADC** 以与 **AWS Auto Scaling** 集成

步骤 1:

导航到本实验指南上一步中与 NSIP 关联的弹性 IP，以访问 Citrix ADC 管理控制台。

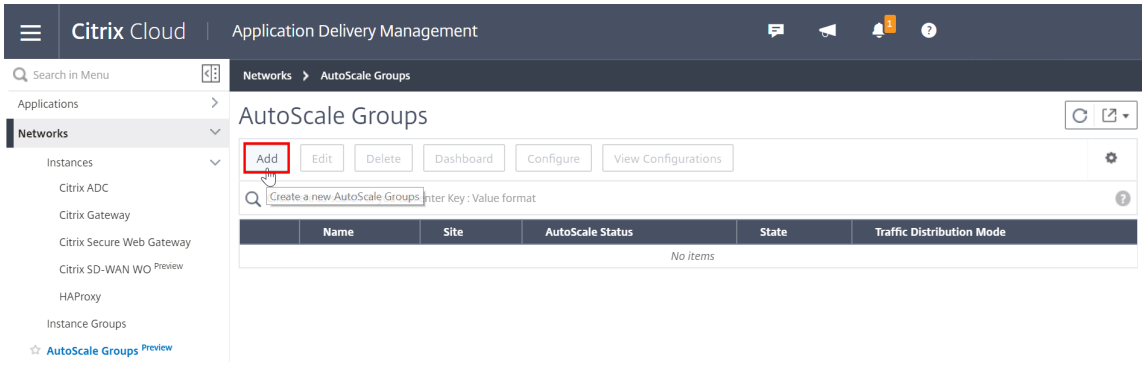
配置 Citrix ADC 的第一步是附加云配置文件。单击 **AWS**，然后单击云配置文件。下一步单击添加以创建云配置文件。

提供云配置文件的名称。虚拟服务器 IP 地址应填充 Citrix ADC 上的内部 IP 并与其关联。自动缩放组是您在本实验指南前面的步骤中创建的组。选择“优雅”，这将允许删除后端实例的超时，从而允许在宽限期内完成任何数据包传输，并且不会终止会话。可以调整宽限期的时间延迟。

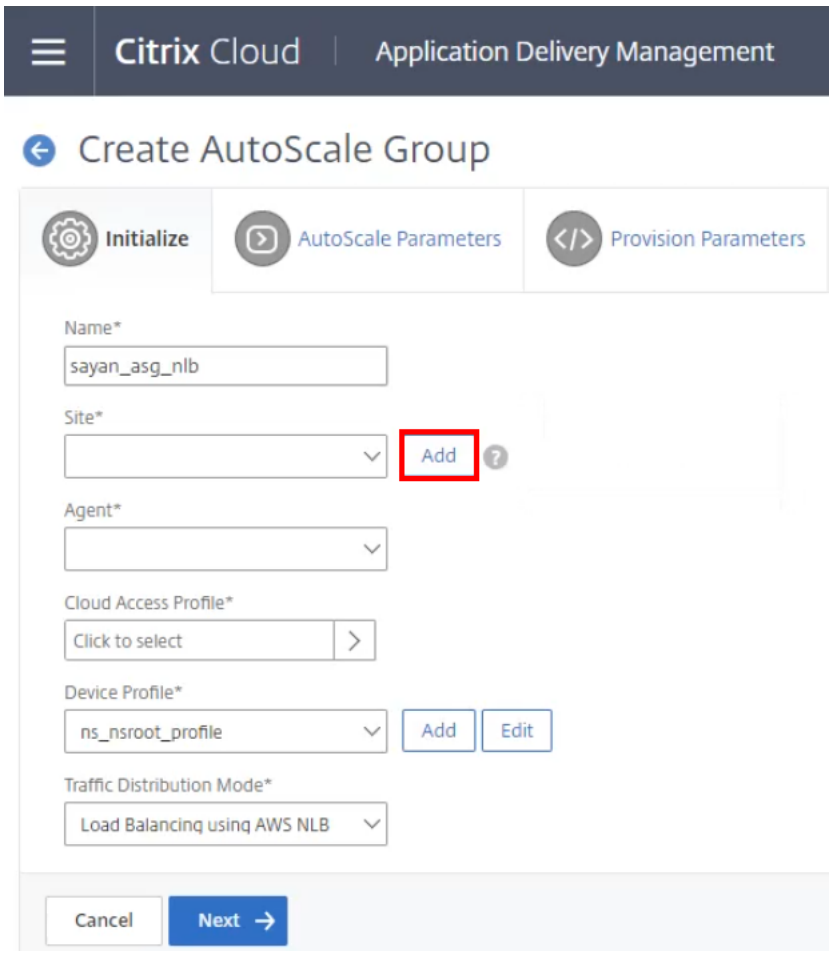


在 **AWS** 中配置 **Citrix ADC** 前端 **Auto Scaling**

1. 要创建 Auto Scaling 组，请登录到 Citrix ADM。
2. 导航到网络 **> AutoScale** 组，然后单击添加以创建组名称。

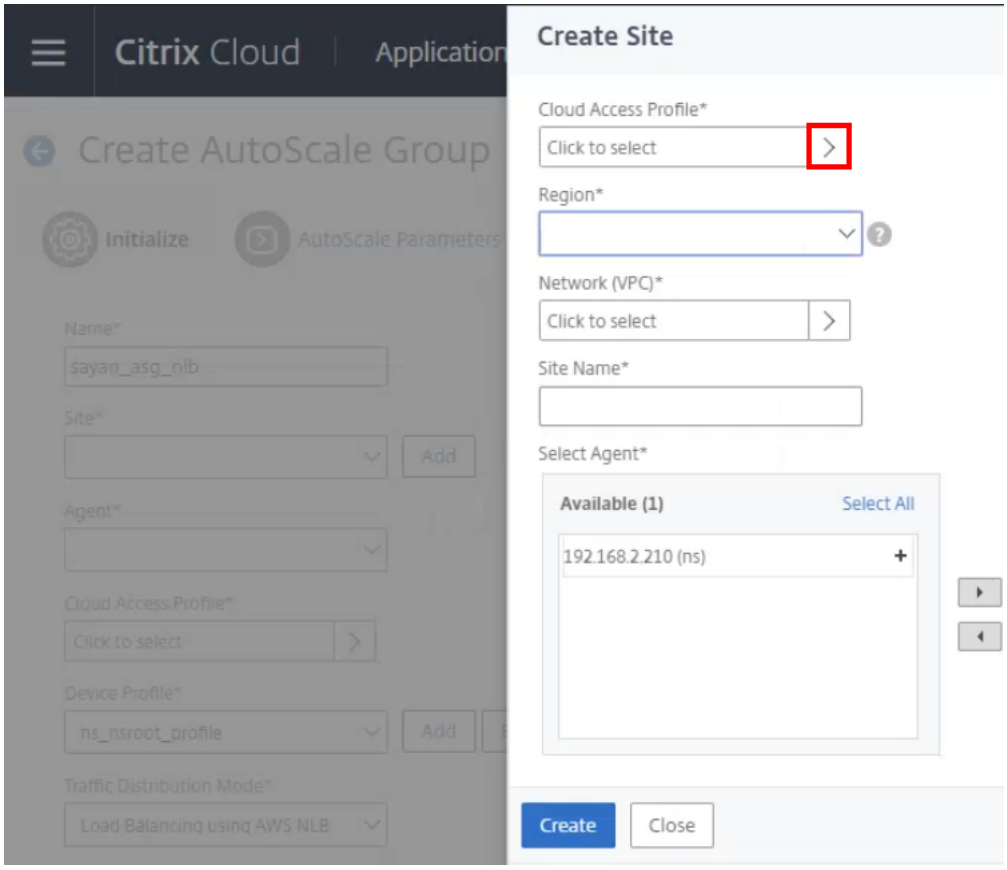


3. 在“站点”设置中，单击“添加”。

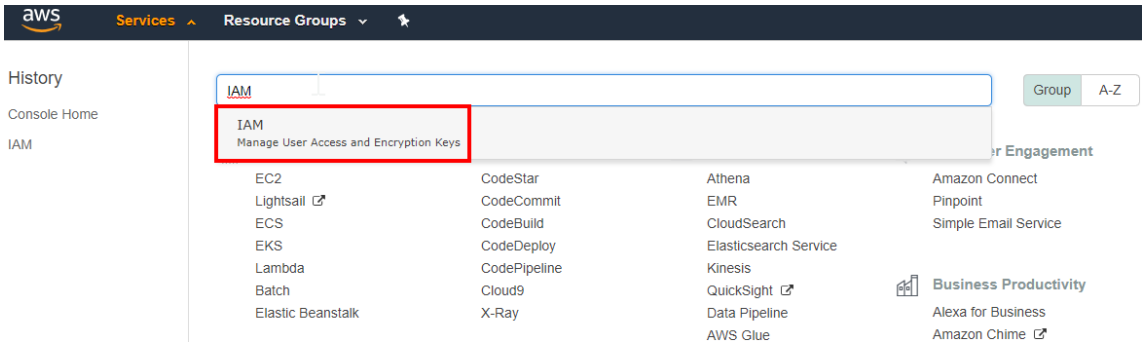


创建云访问配置文件

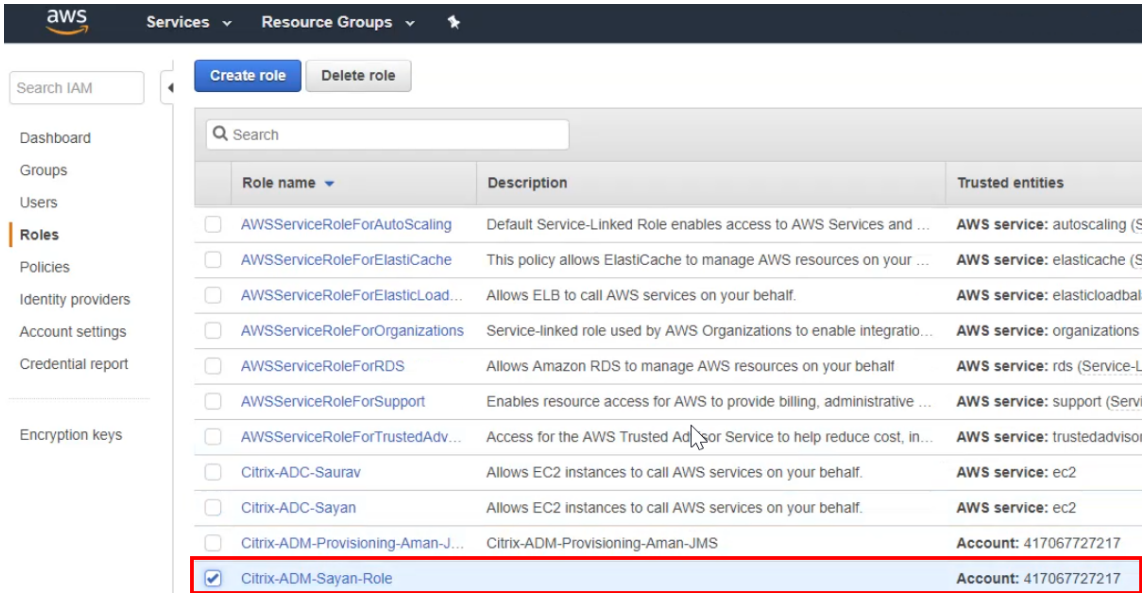
1. 创建站点时，在云访问配置文件中添加 **AWS**。



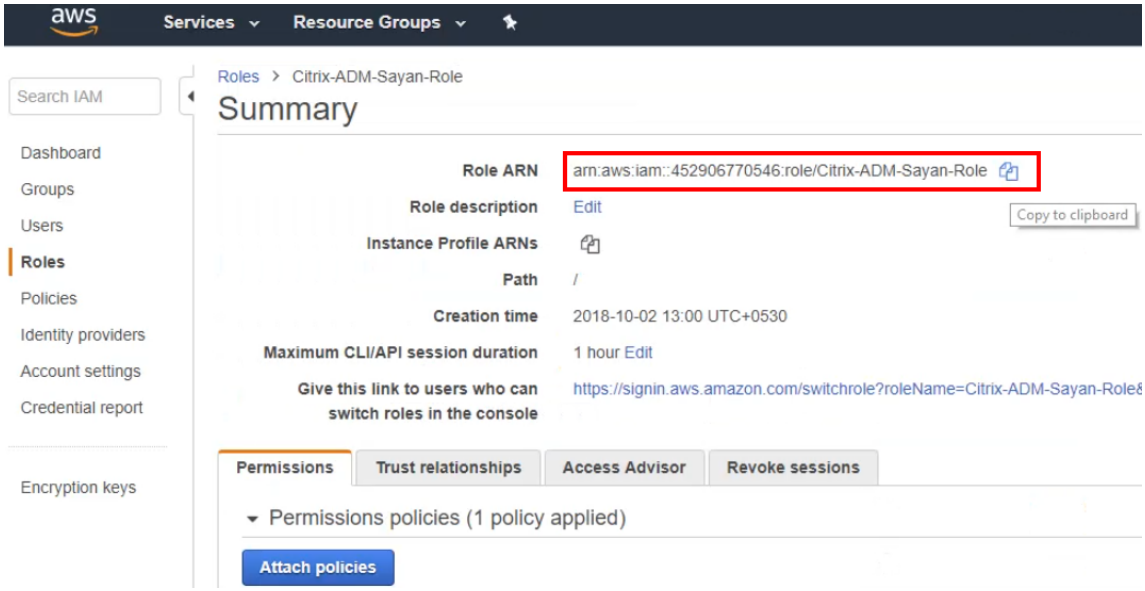
2. 命名配置文件并登录 AWS 门户。搜索身份和访问管理 **(IAM)** 服务以管理用户访问和加密密钥。



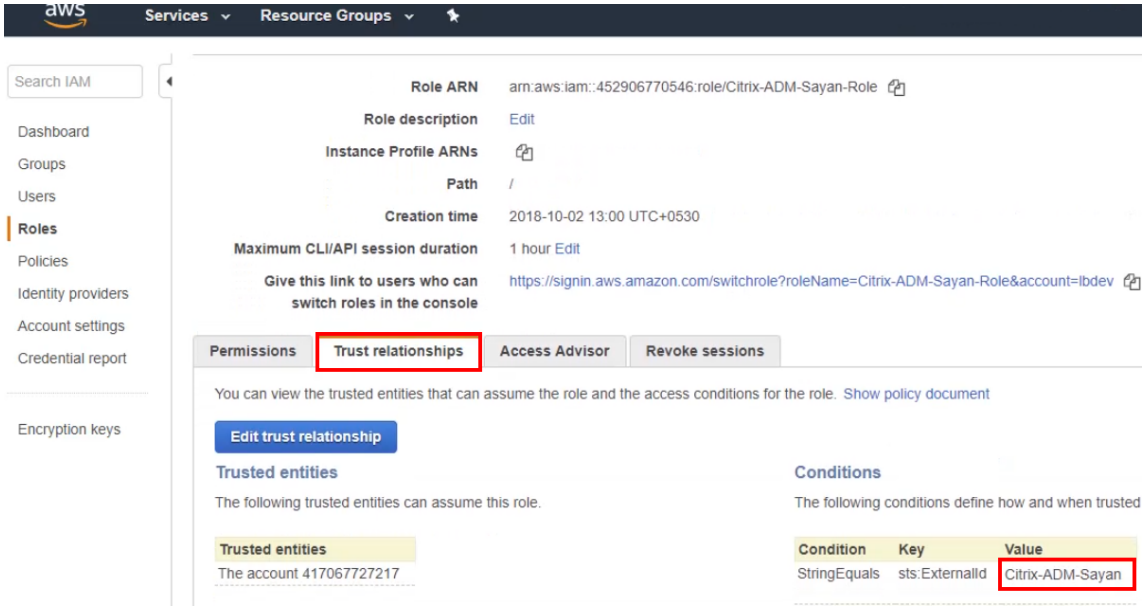
3. 在 **IAM** 控制面板中，选择左侧面板上的角色，然后搜索相应的 Citrix ADM 角色。



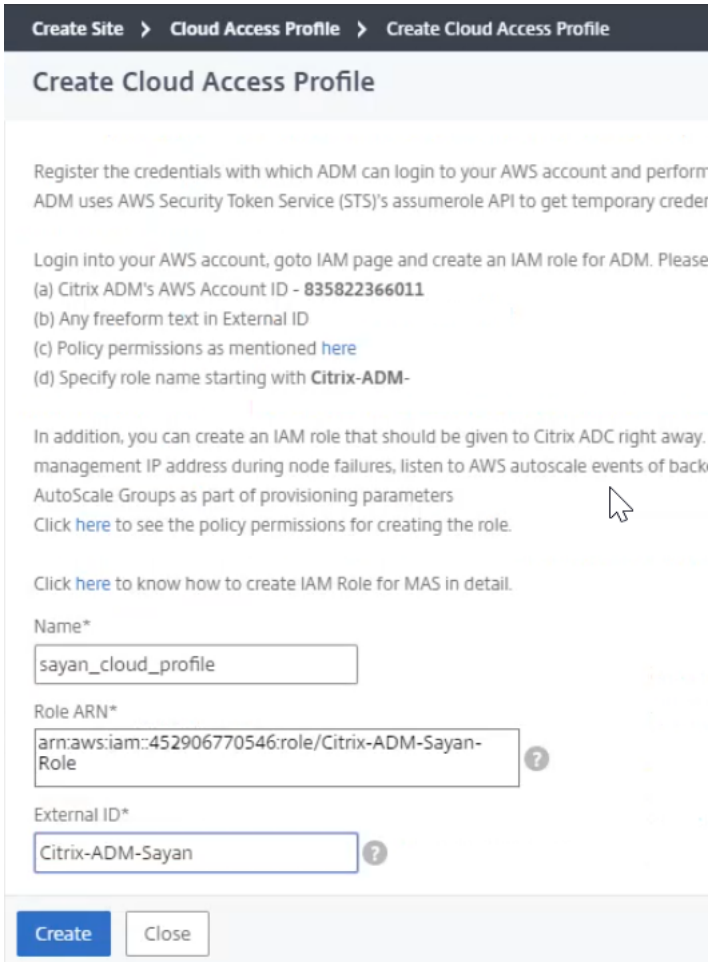
4. 将角色 **ARN** 复制到剪贴板。



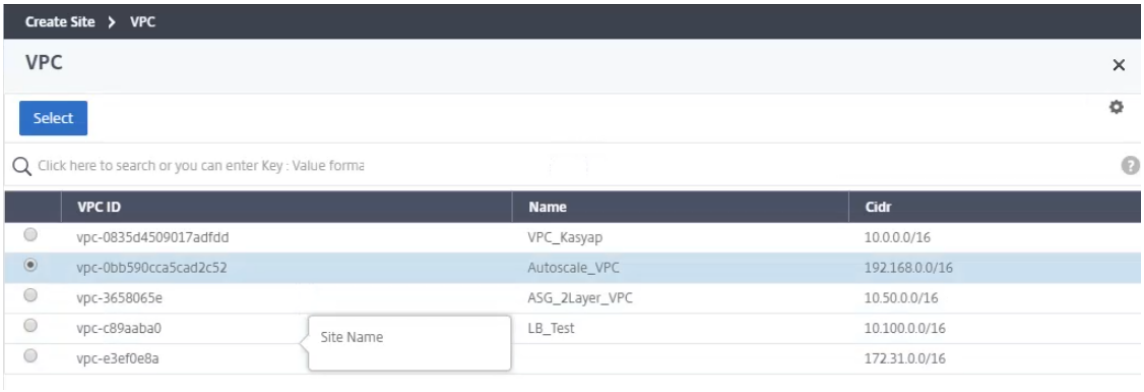
5. 复制名称后，返回 Citrix ADM 控制台并将名称粘贴到角色 **ARN** 文本字段中。
6. 要获取外部 ID，请返回 AWS 角色控制板，导航至“信任关系”选项卡，然后从“条件”复制值。



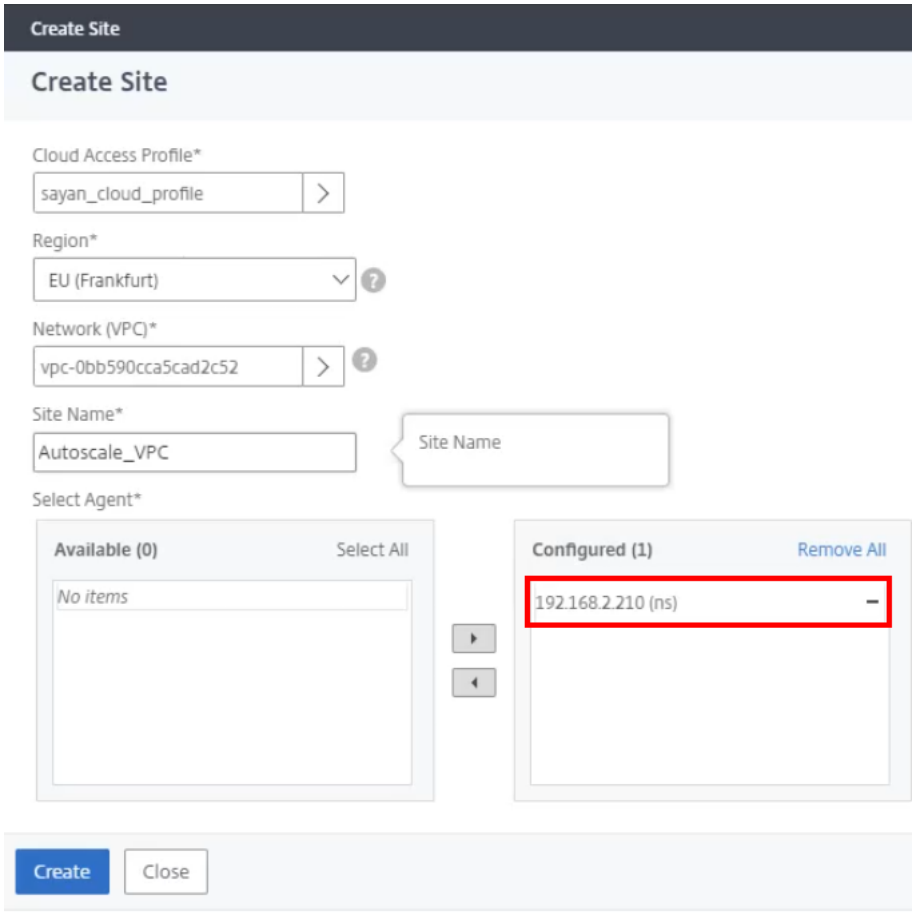
7. 在 Citrix ADM 控制台中，将值粘贴到“外部 **ID**”字段中，然后单击“创建”。



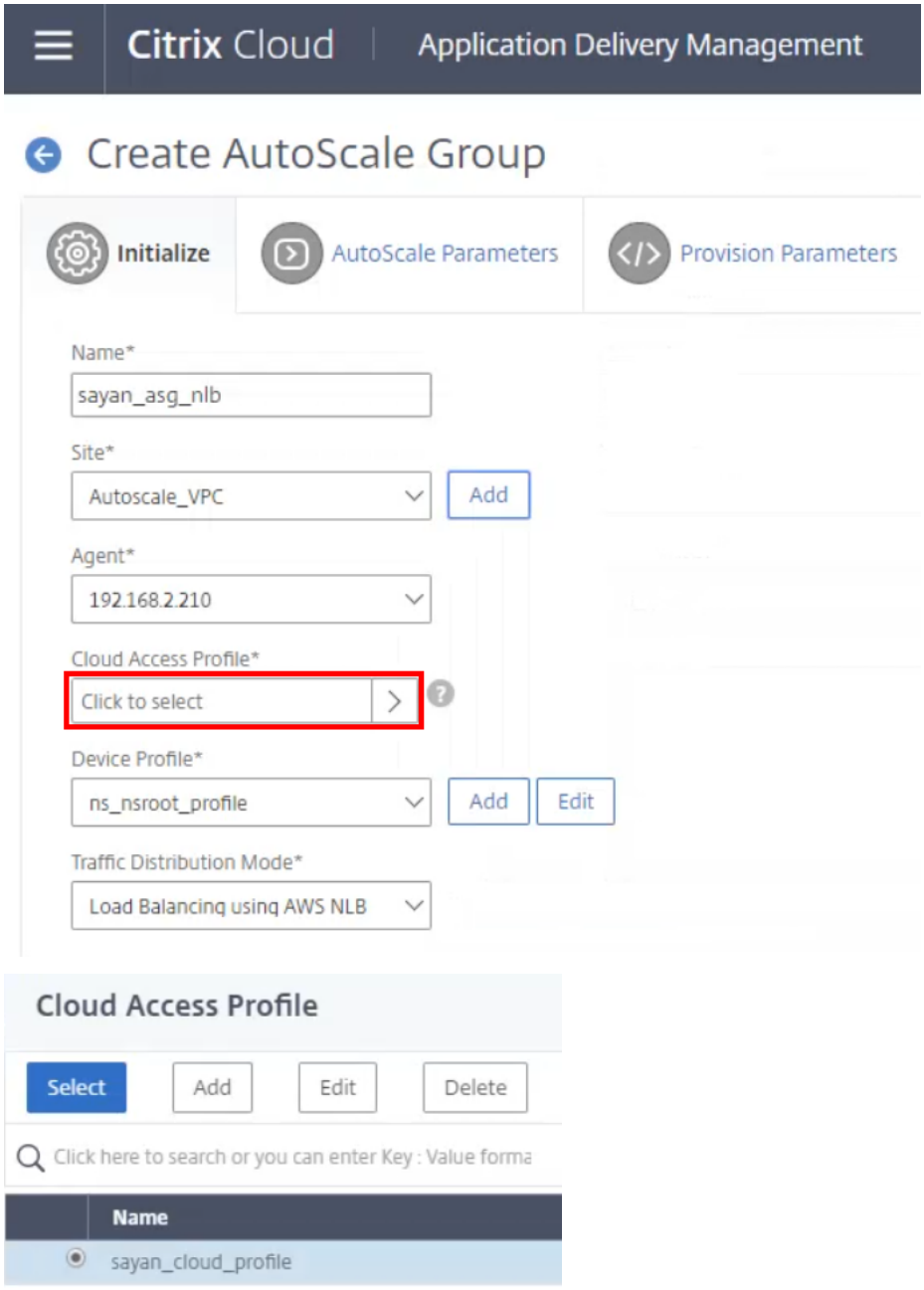
8. 选择区域，然后选择适当的 VPC 网络。



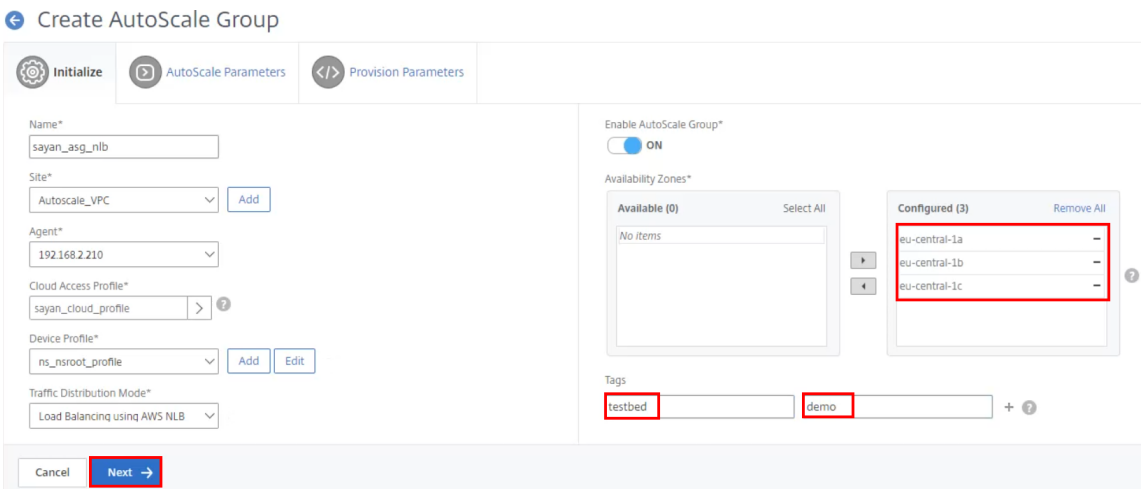
9. 将代理从“可用”移至“已配置”。



10. 选择相应的云访问配置文件。

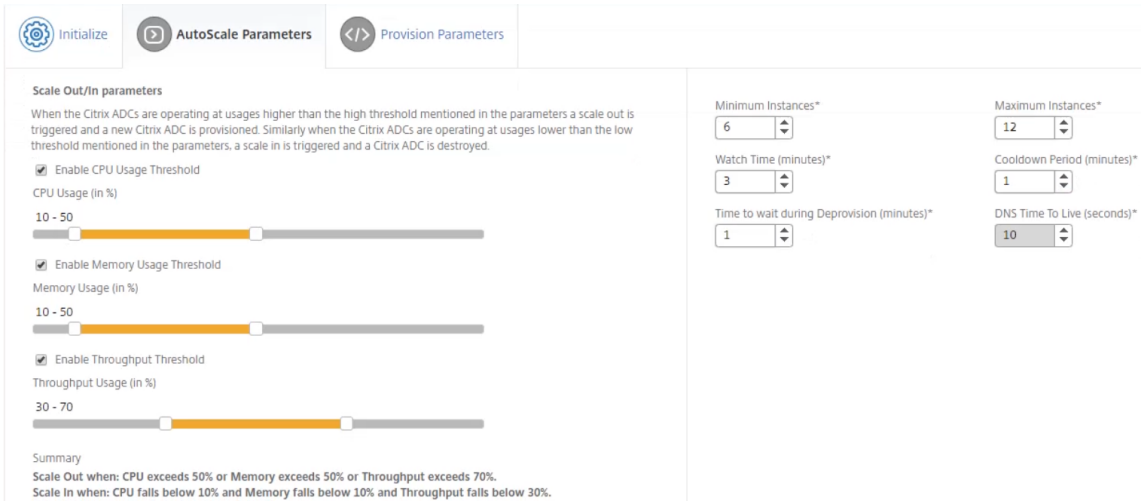


11. 加载后，将可用区域从可用移至已配置，并将相应的标签添加到 **AutoScale** 组。选择下一步以开始设置 **AutoScale** 参数。

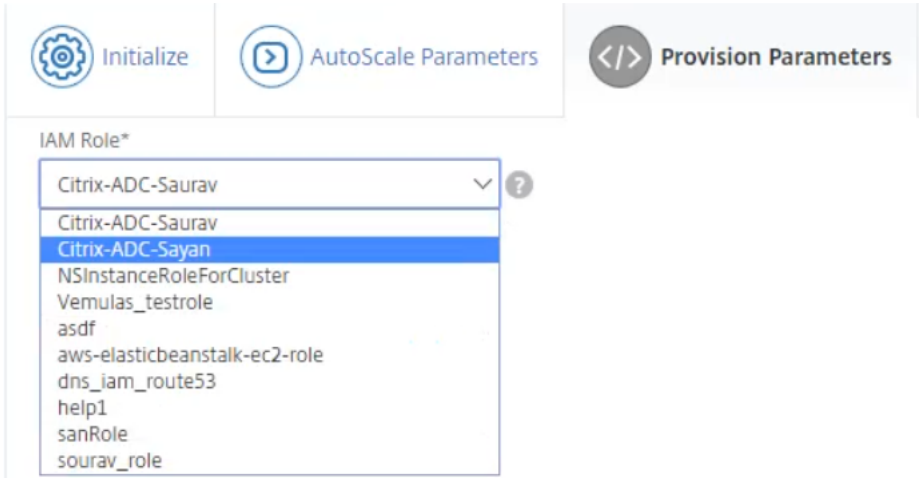


设置 **AutoScale** 参数

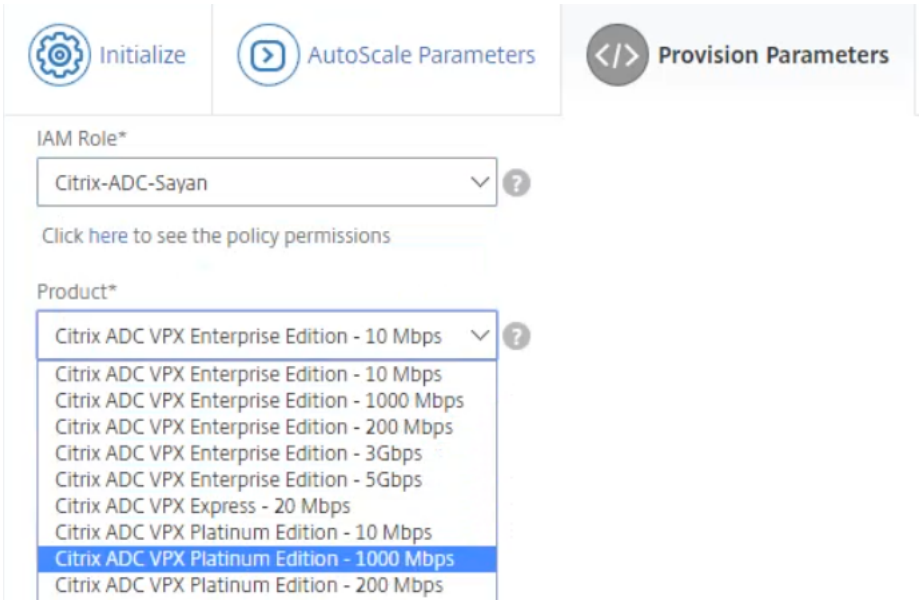
1. 设置 AutoScale 参数时，将阈值和参数调整为所需的设置。然后单击“下一步”开始配置 置参数设置。



2. 在“设 置参数”部分，从“**IAM** 角色”字段中选择角色。

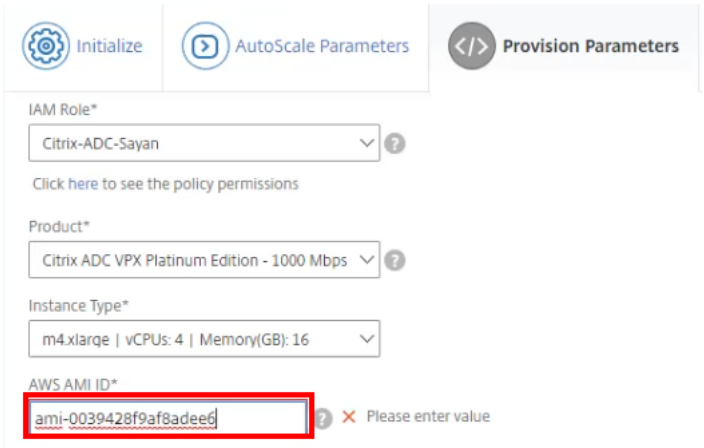


3. 选择适当的 Citrix ADC 产品和版本。

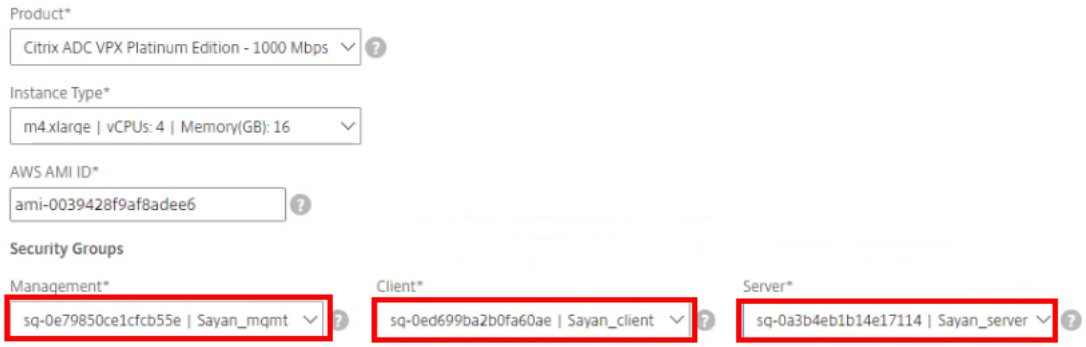


4. 从 AWS 中的特定实例收集 Amazon Machine Image (AMI) ID。在 **AWS AMI ID** 字段中输入该 ID。

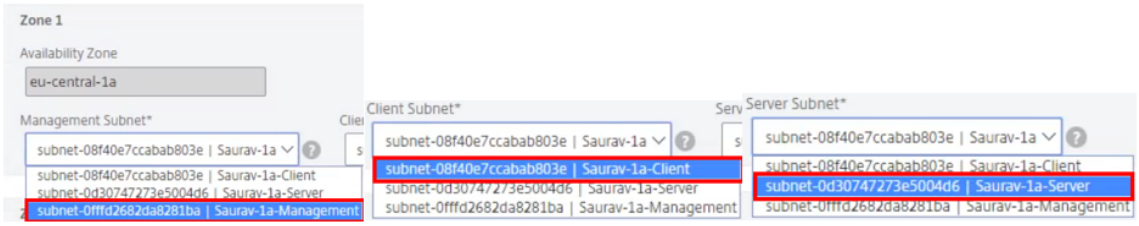
```
AMI ID:
--------------
ami-0039428f9af8adee6
```



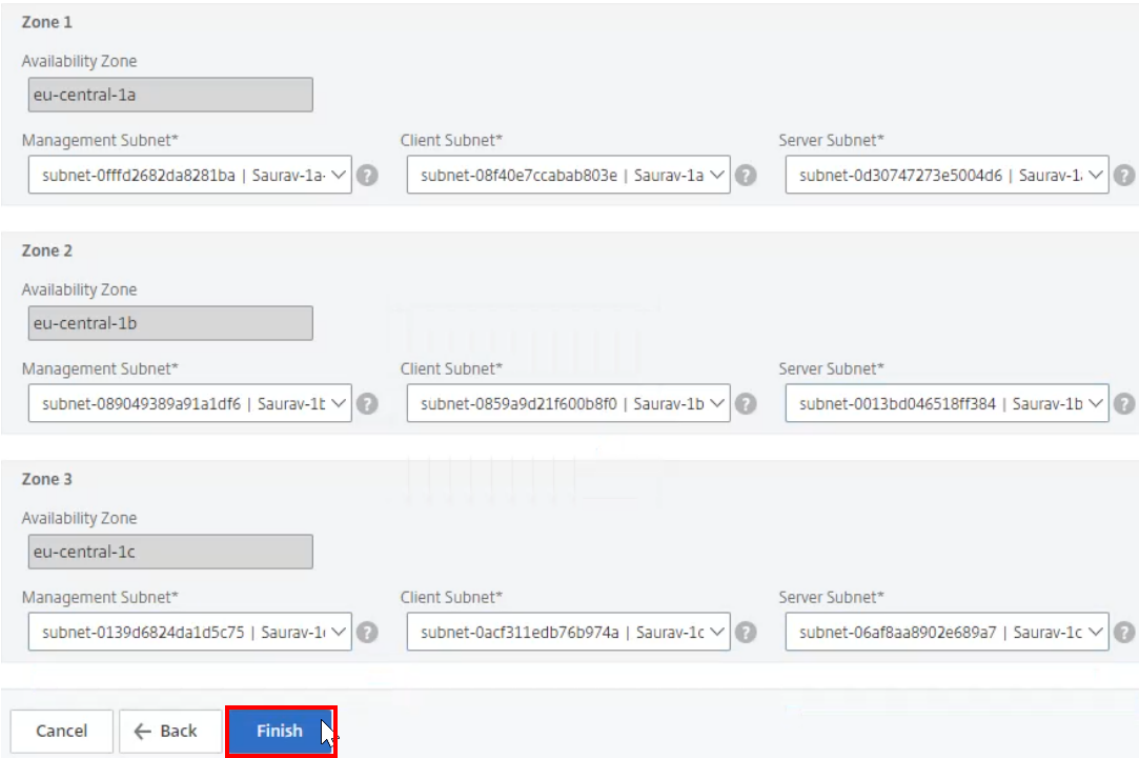
5. 添加 **AMI ID** 后，使用相应的组更新安全组。

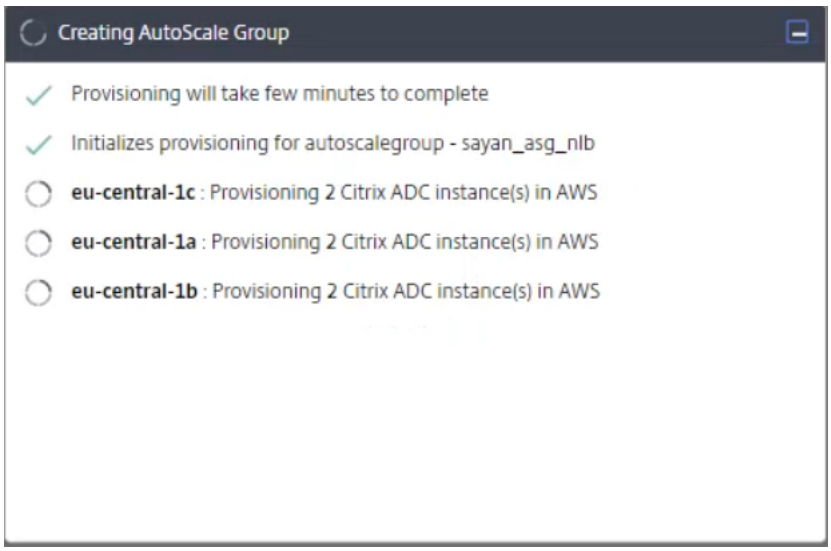


6. 要启动区域 1、2 和 3 的配置，请分配相应的管理、客户端和服务器子网。



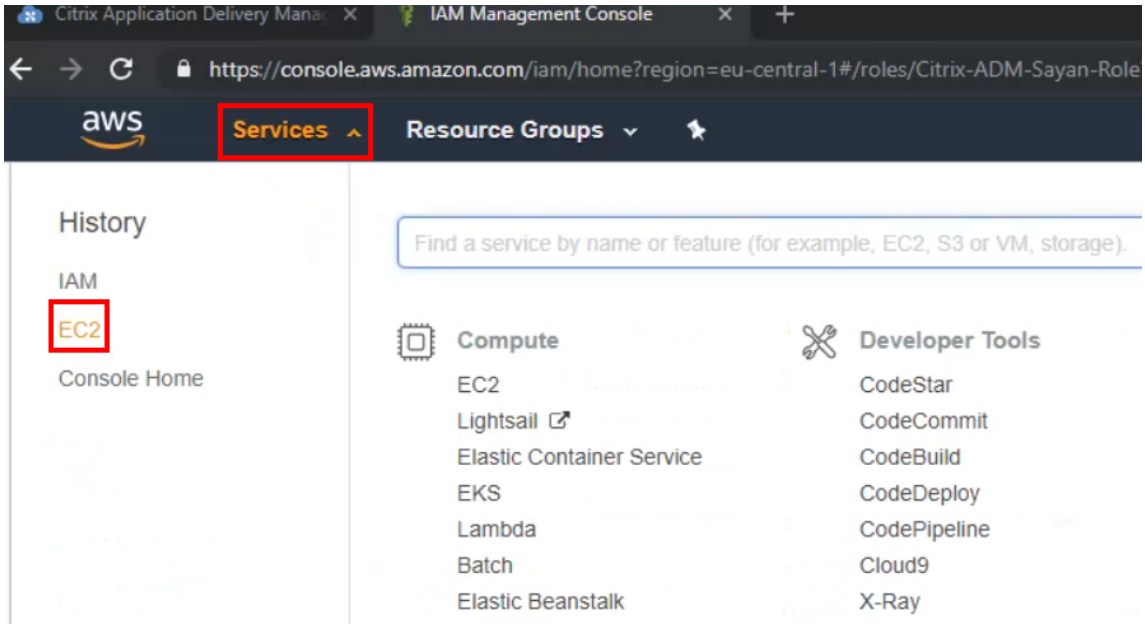
7. 单击完成以创建此 Auto Scaling 组的配置。创建过程可能需要 10-20 分钟。



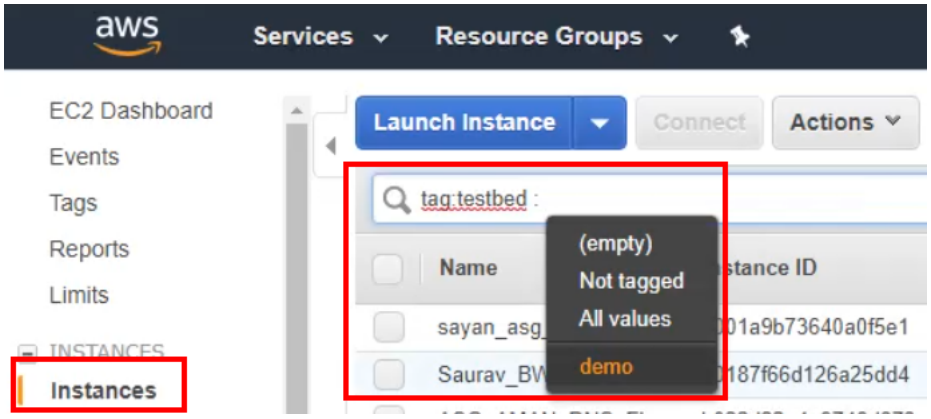


在 **AWS** 中初始化实例

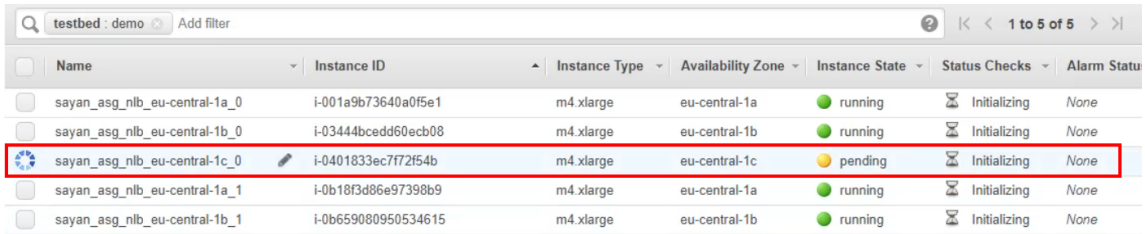
1. 创建 Auto Scaling 组时，打开您的 AWS 控制台并导航到服务选项卡。选择 **Amazon Elastic Compute Cloud (EC2)** 服务。



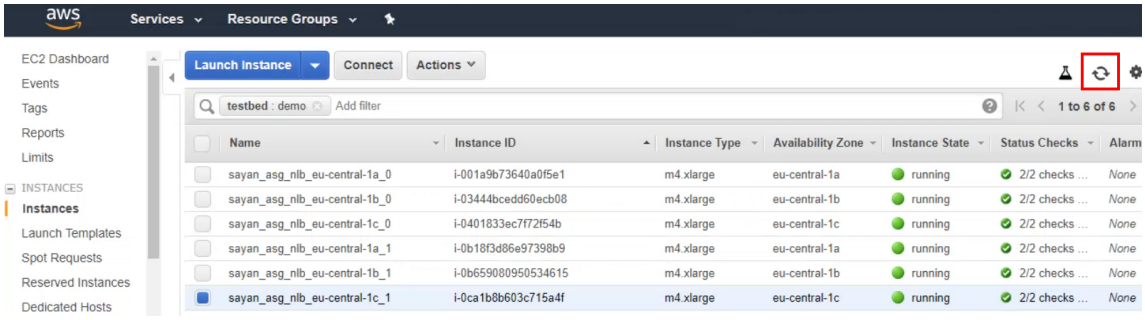
2. 在 EC2 控制板上，选择实例选项卡，然后使用 **AutoScale** 组部分中设置的标签进行筛选。



3. 筛选后，您可以看到仍在初始化的挂起实例。

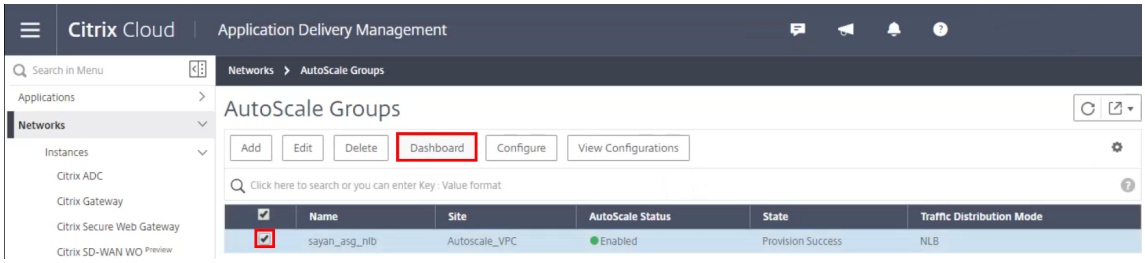


4. 这些实例在创建完成后应该完成初始化。

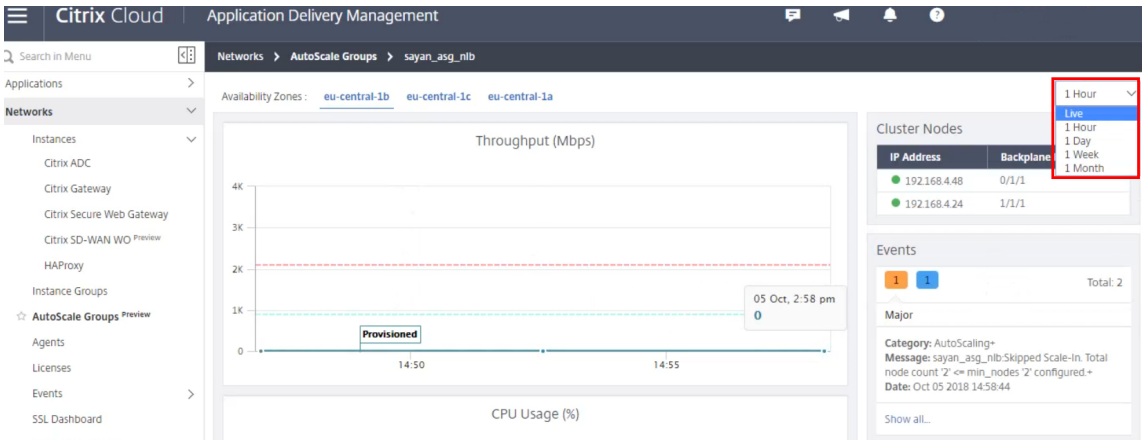


监视 **Auto Scaling** 组事件

1. 创建 Auto Scaling 组后，选择您的组并转到 **AutoScale** 组控制板。

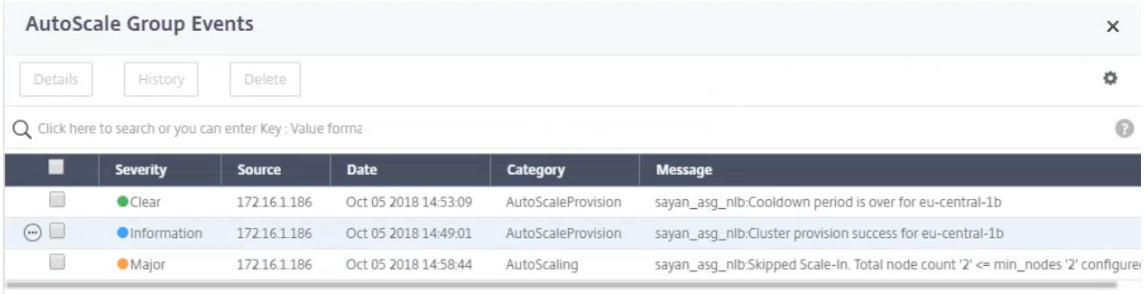


2. 筛选出特定时间段以监视 Auto Scaling 组。要获得实时洞察，请将监视周期更改为 **Live**。



3. 单击图表中显示的以下数据点可查看任何组事件。

4. 查看特定实时事件时，可以监视相应的 Auto Scaling 组的特定事件。



使用 **Citrix ADM** 服务置备 **Citrix ADC VPX** 实例

Citrix ADM 服务是一种基于云的解决方案，它能够监视 Citrix ADC 实例并了解应用程序的运行状况、性能和安全性。此外，通过利用 Provisioning 工具在公有云（如 AWS）中自动创建实例，它还简化了对多个位置的 ADC 实例的管理，无论它们位于本地还是云中。

必备条件

使用 Citrix ADM 服务在 AWS 上配置 Citrix ADC 实例需要执行一些步骤，这些步骤将在先决条件文档中进行总结。有关详细信息，请参阅在 AWS 上配置 Citrix ADC VPX 实例。

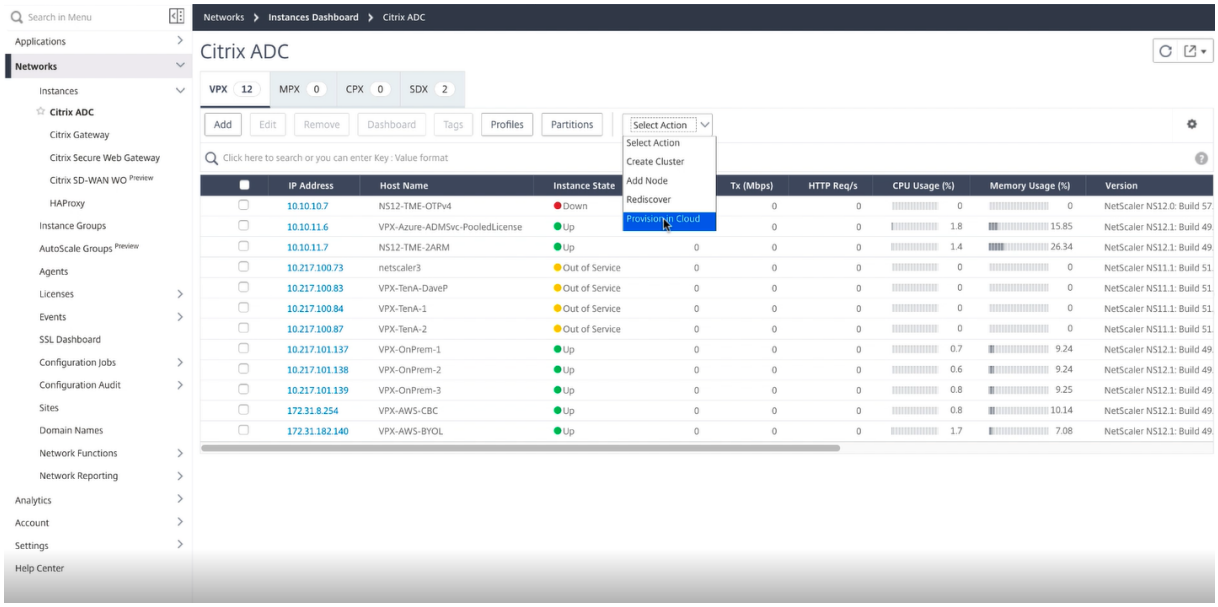
这些步骤包括在 Citrix ADM 中配置 Citrix ADC VPX 实例之前，在 AWS 上执行以下任务：

- 创建子网
- 创建安全组
- 创建 IAM 角色并定义策略

IAM 角色必须配置为允许 Citrix ADM 服务访问 AWS 帐户的权限。设置完所有内容后，您可以利用 Citrix ADM 服务在 AWS 上配置 VPX 实例。

使用 **Citrix ADM** 服务置备 **Citrix ADC VPX** 实例

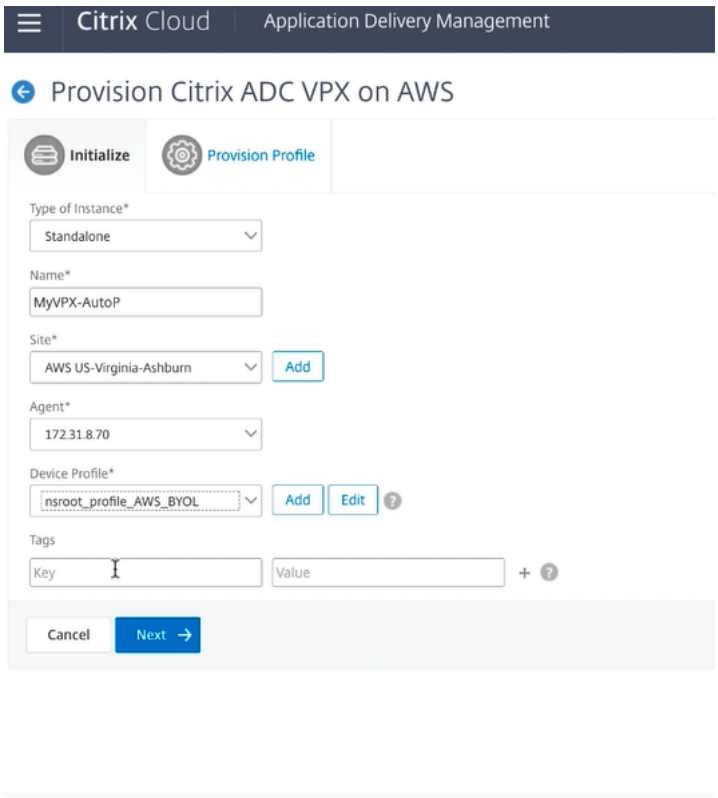
登录到 Citrix Cloud ADM Service，然后导航到网络 **>** 实例 **> Citrix ADC**。然后在“选择操作”选项卡下，单击“在云中设置”。



这将提示您定义有关要置备的实例的信息。

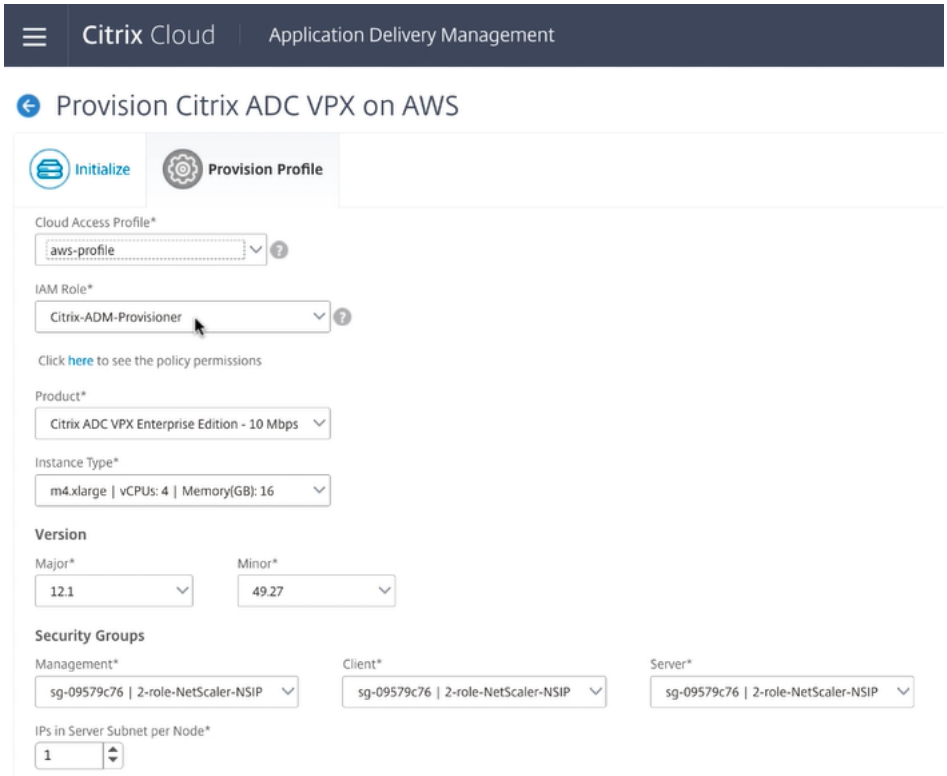
具体而言，您必须定义以下内容：

- 实例类型：此处选择独立实例。
- 名称：您希望实例在预配置时采用的名称。
- 站点：站点定义要在哪个区域或区域执行部署。
- 代理：代理指定站点中可用的 ADM 代理。在执行自动置备之前，必须先设置此选项。在开始本练习之前，您需要同时创建一个站点和属于该站点的代理。
- 设备配置文件：具有“nsroot”作为用户名和所需密码的设备配置文件。Citrix ADM 设置了 Citrix ADC 后，ADC 的 nsroot 用户口令将设置为配置文件中提到的密码。此外，每当 Citrix ADM 需要登录到实例时，都会使用此配置文件。
- 标签：实例或实例组的可选标签。



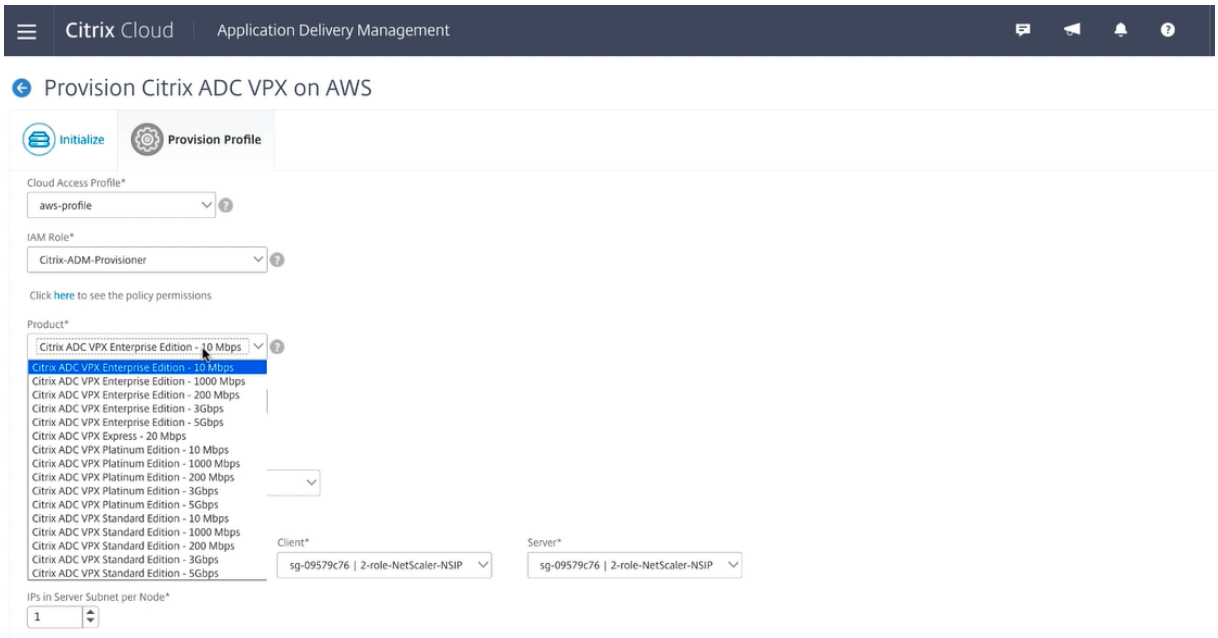
然后为您的 AWS 帐户选择云访问配置文件。这是 Citrix ADM 用于登录 AWS 帐户以获取实体和执行预 Provisioning 和取消 Provisioning 等操作的配置文件。使用该配置文件，Citrix ADM 服务将使用与您的帐户相关的对象填充其余字段。

在这种情况下，Citrix ADM 服务使用预定义的 IAM 角色来配置 VPX 实例，但您可以创建其他角色。



然后，您必须根据所需吞吐量选择要部署的 VPX 实例的产品版本。

> 注意：
>
> VPX Express 可供您部署无需许可证的 VPX 实例。



版本

确定要运行的软件版本，选择主要版本和次要版本。

安全组

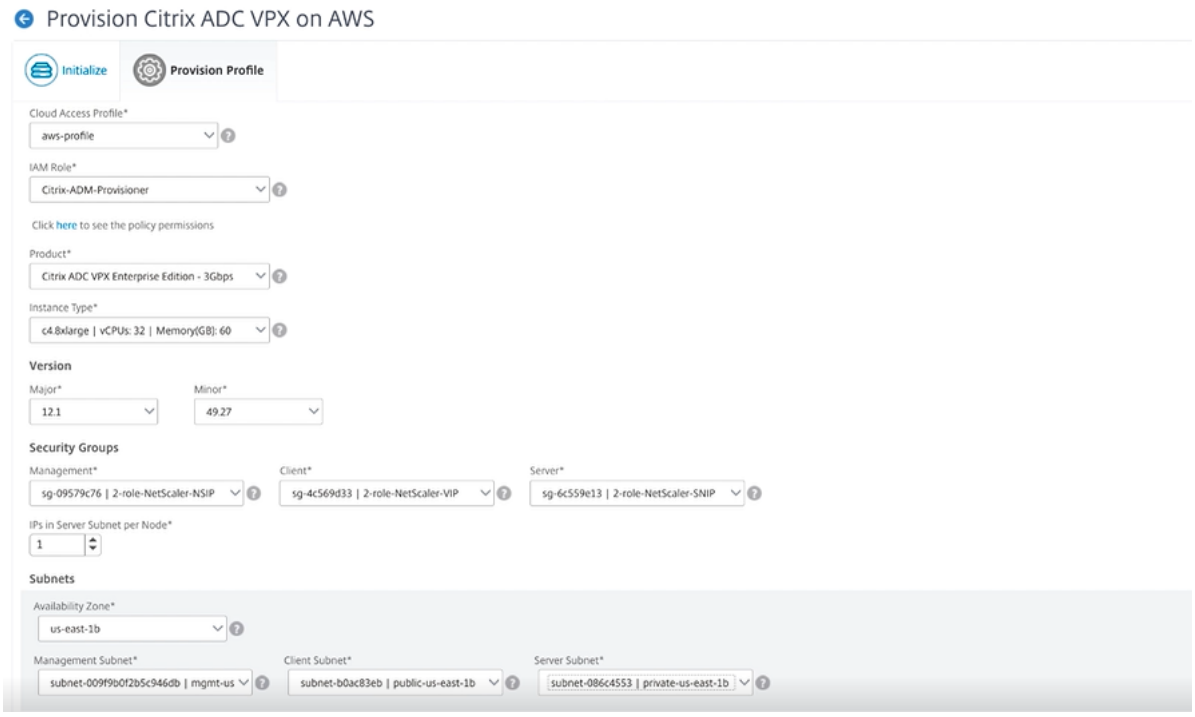
安全组应具有访问不同虚拟私有云 (VPC) 的预定义权限。由于每个实例都需要三个网络接口或 vNIC，因此您需要将三个不同的安全组应用到您所部署的服务，包括：

- 一个用于远程管理（角色 NSIP）
- 一个用于客户端访问（角色 VIP）
- 一个用于服务器端通信（角色 SNIP）

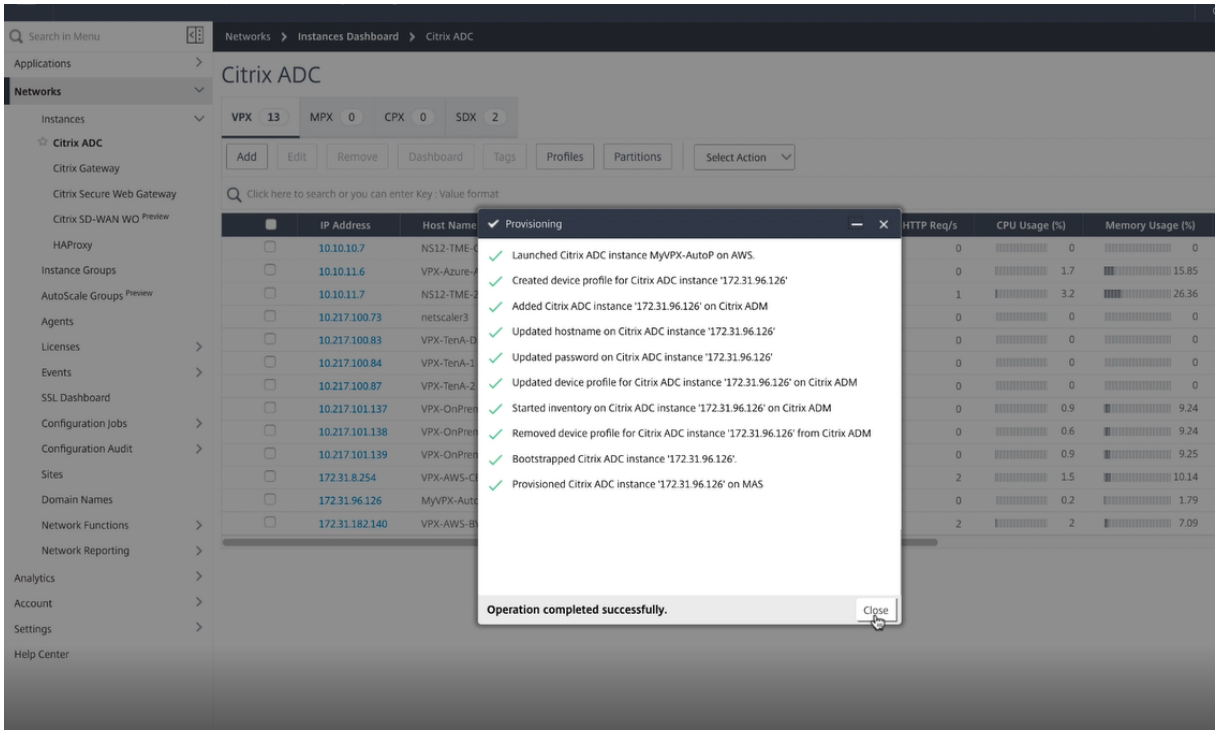
此外，您应该选择此解决方案的可扩展性所需的必要数量的 IP。

最后，您必须选择希望部署位于哪个可用区域，并为每个子网定义重合的 VPC 子网信息：

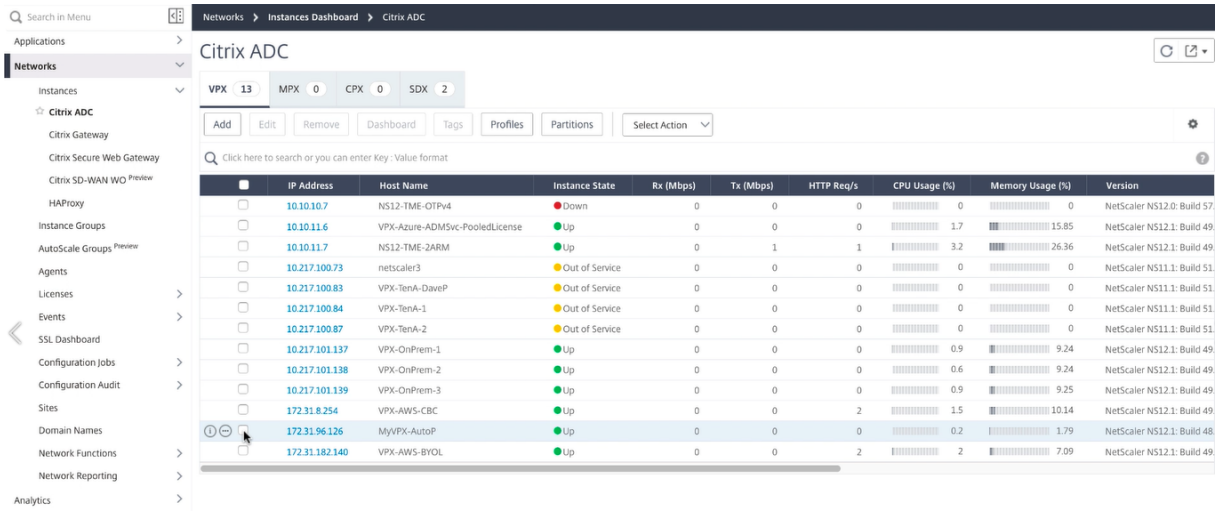
- 一个用于管理接口 (NSIP)
- 一个供客户访问 (VIP)
- 一个用于 SNIP 访问后端服务器 (SNIP)



单击完成后，将开始部署。部署成功完成后，您会收到一条通知，告知您的 VPX 已部署。

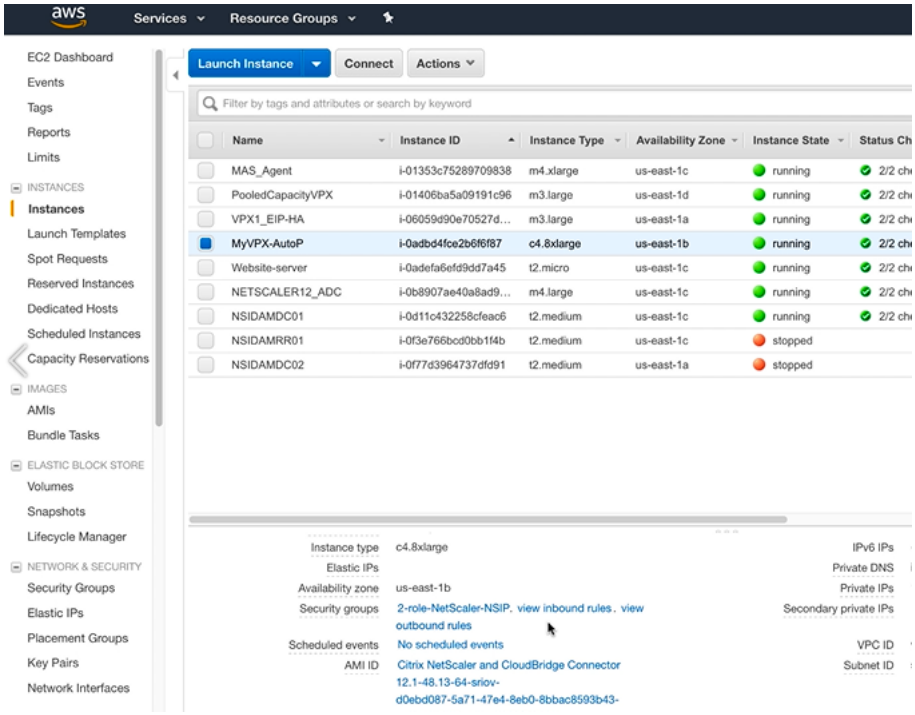


部署完成后，您可以在 Citrix ADM 中查看 Citrix ADC VPX 实例，以实现所有管理和部署目的。



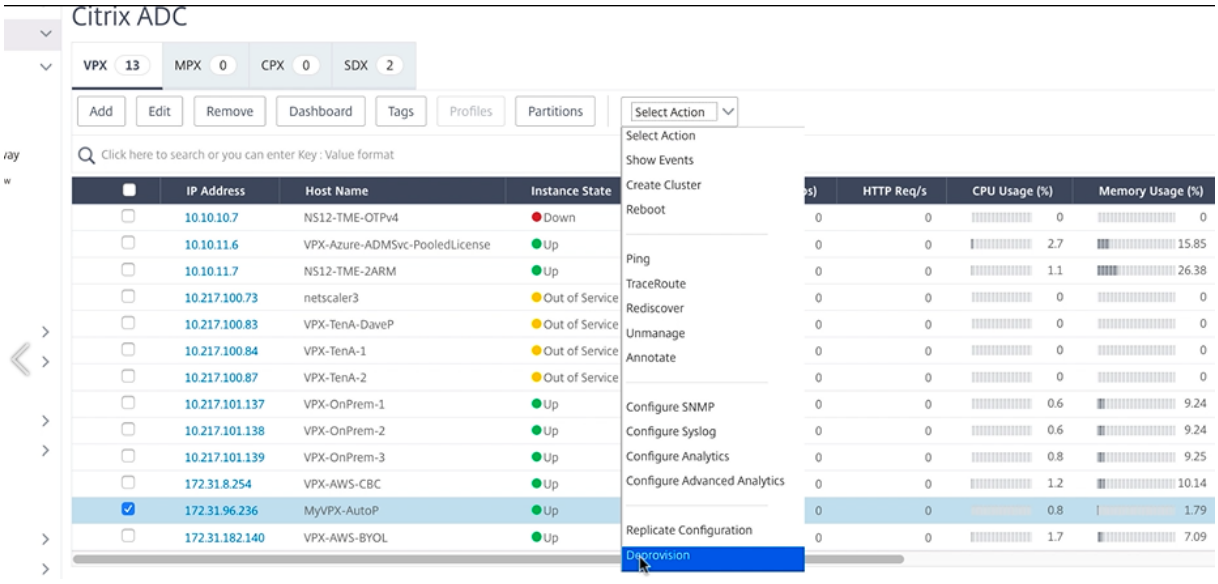
然后，您可以导航到 EC2 控制台，查看使用我们在 Citrix ADM 设置中建立的名称创建的新实例。它全部同步以便在 Citrix ADM 中进行管理，并准备将您的应用程序部署到 Citrix ADC。

AWS 部署

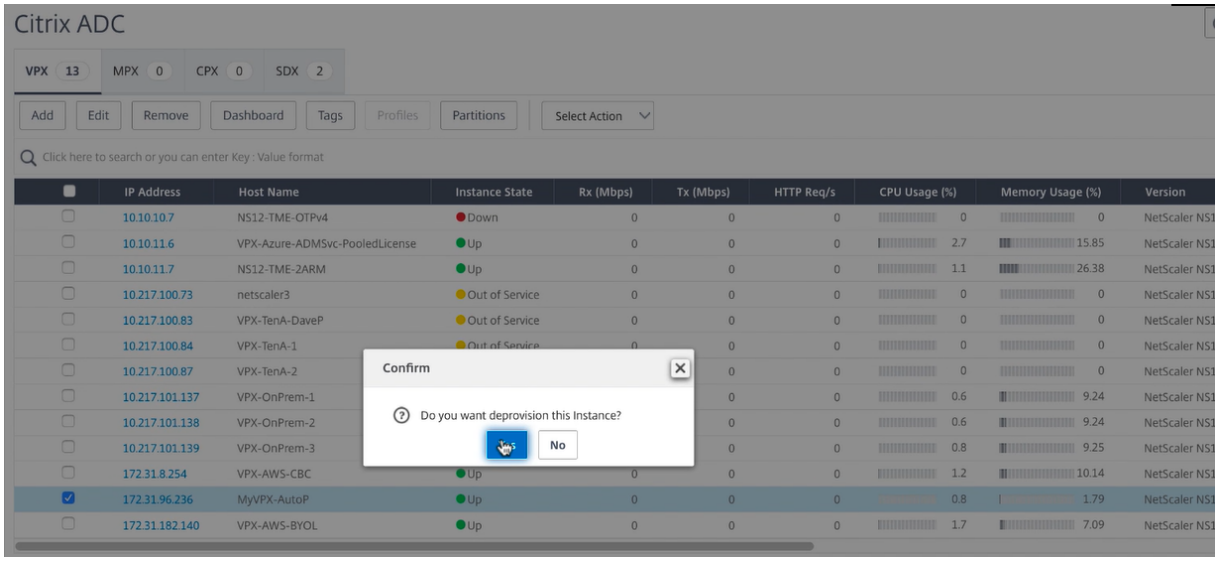


要取消配置这些实例，请导航回 Citrix Cloud ADM Service，然后转到网络 **>** 实例 **> Citrix ADC**。在“选择操作”选项卡下，单击“取 消设置”。

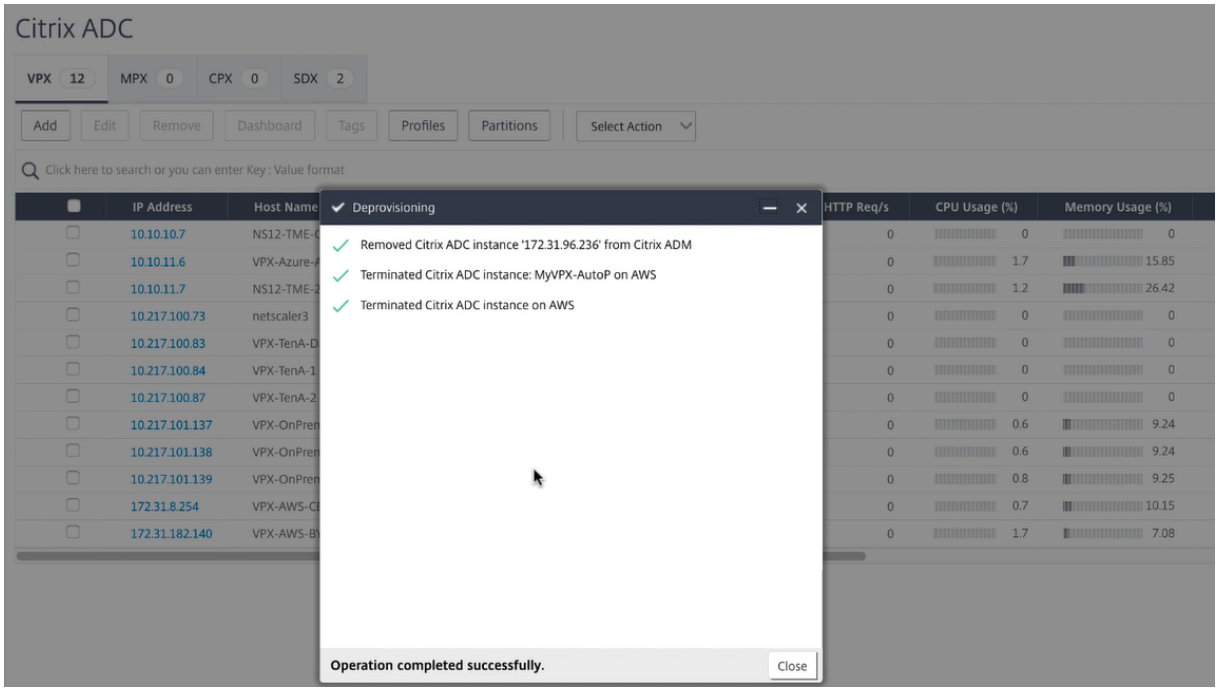
AWS 取消配置

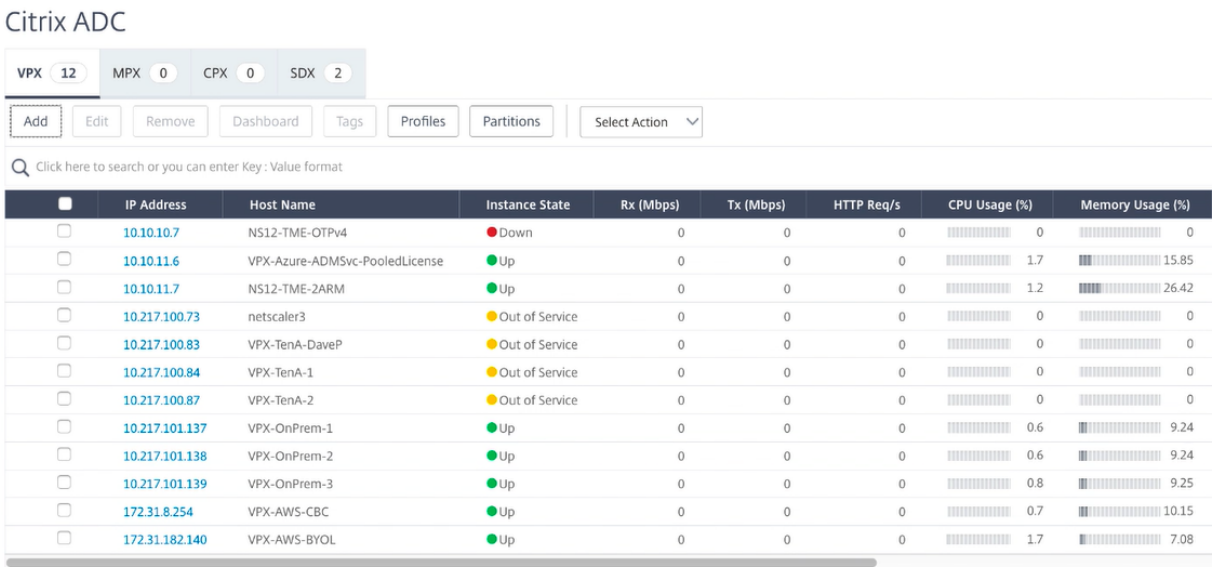


系统将提示您确认您的操作。要继续，请选择“是”，然后冲销所有 Provisioning。



收到 VPX 实例已取消置备的确认信息后，您将不再在 Citrix ADM 控制台中看到该设备。





更多信息

- 实施指南：AWS 中的 Citrix XenDesktop
- 如何为通用企业应用程序配置 Unified Gateway
- 通过在 Amazon Web Services (AWS) 上运行 Citrix 解决方案来加快您的业务速度
- 适用于 AWS 的云网络和桌面虚拟化解决方案
- 在 AWS 中跨多个可用区使用 Citrix ADC HA
- NetScaler VPX 与 AWS AutoScale 集成

Citrix ADC 管理分区验证的参考设计

May 20, 2020

功能概览

Citrix ADC 管理分区在单个 Citrix ADC 实例中启用软件级别的多租户。每个分区都有自己的控制平面和网络平面。

管理分区的主要优势是：

1. 控制平面 — 隔离的配置和管理
2. 数据平面 — 在分区边界内严格控制的关键分区数据和文件
3. 网络平面 — 使用自己的网络配置隔离流量。同一 Citrix ADC 上的两个分区看不到相同的流量通过每个分区

本文档详细介绍了管理分区启用的典型使用案例，以及在客户环境中使用管理分区的指南。

管理分区使用案例

管理分区的企业用例

Citrix ADC 管理员可以将 Citrix ADC 分区为多个 ADC，并将分区分配给不同的应用程序管理员，例如 Microsoft SharePoint 和 Microsoft Lync。每个应用程序管理员/所有者都可以进行自己的配置更改。

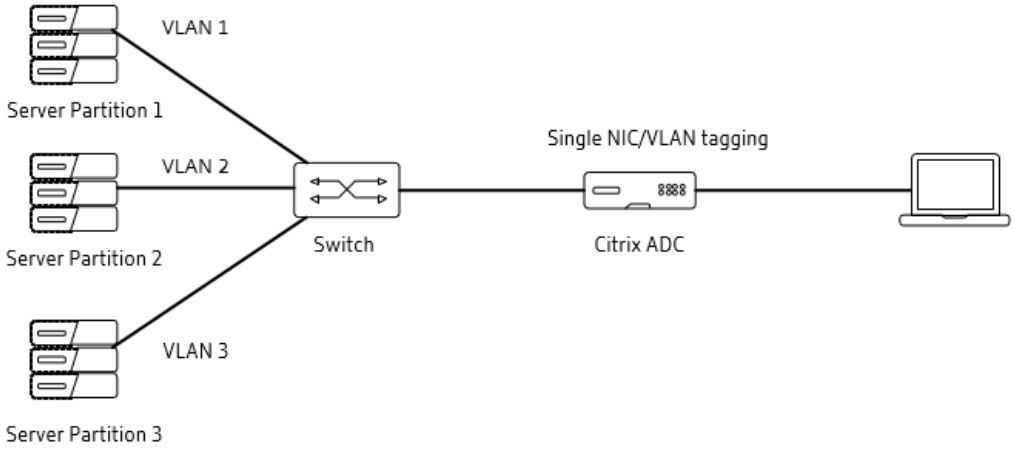
IP 重叠： IP 重叠的主要好处是，相同的 IP 范围可以在不同的管理分区中使用，而不会发生任何 IP 冲突。对于后端服务器，您可以使用相同的专用 IP 地址集。在 IP 重叠方案中，VLAN 无法共享。

虚拟路由：路由配置对于每个分区都是唯一的，每个分区所有者都可以配置自己的路由协议。

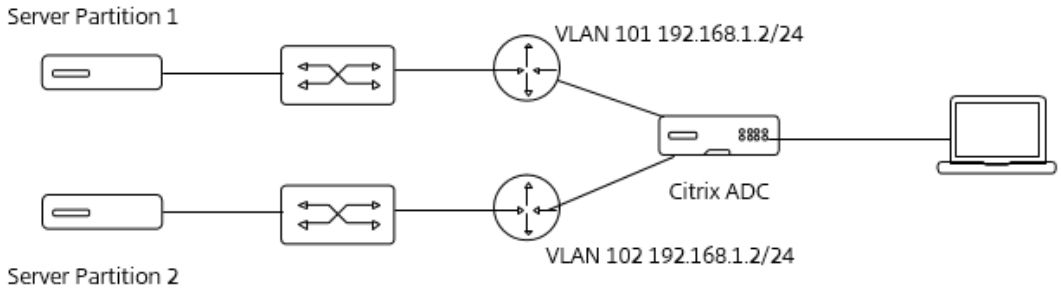
名称空间隔离：实体名称在不同分区中是唯一的，因此您可以在不同的管理分区中使用相同的名称。

参考图：

单个网卡 — 多个 VLAN



IP 重叠：



管理分区的服务提供商用例

服务提供商可以对 Citrix ADC 进行分区，并根据其带宽要求和并发连接数将其分配给各个客户端。

服务提供商可以使用 NITRO API 开发编排工具，从各客户端获取有关其带宽要求和并发连接的输入，创建分区并将其分配给客户端。

以下是一组有助于服务提供商的隔离：

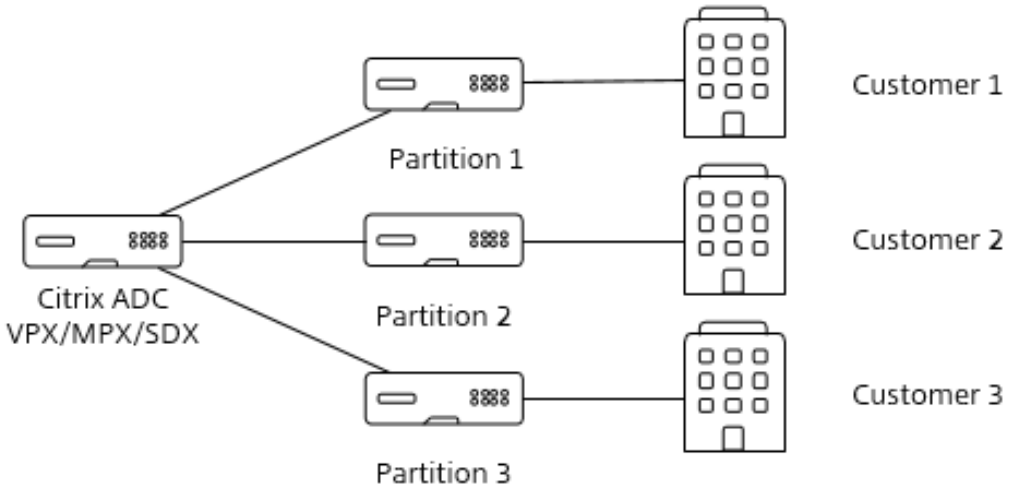
文件系统：为每个分区分配文件系统的一部分，并且存储在该分区空间中的文件对其他分区不可见。SSL 证书/密钥存储在该分区中，其他分区所有者不可见，从而使每个分区都安全。

共享 **VLAN**：在具有多租户部署的典型服务提供商中，最终客户可能没有用于传入流量的独立 VLAN。当无法使用专用 VLAN 时，共享 VLAN 功能共享 VLAN。

VLAN 标记：单个接口可以跨多个管理分区共享，并通过使用标记的 VLAN 隔离。对于未标记的 VLAN，请使用共享 VLAN。

故障排除和调试：管理员可以独立查看每个分区的流量统计信息，并通过分区 ID 进行筛选来分离日志。跟踪函数确保分区独立性，因为从一个分区触发的跟踪永远不会看到来自另一个分区的数据包。

参考图



实施管理分区的指导原则

管理分区支持共享资源，包括带宽、内存和并发连接，并在网络、数据和管理层面提供隔离。

资源分区

ADC 管理员需要以下详细信息来配置管理分区：

1. 连接 —（TCP 连接数）
2. 内存
3. 带宽要求

连接数和带宽要求取决于应用程序和相应分区处理的流量。ADC 管理员与应用程序管理员协商后将获得分区的连接/带宽。

内存分配准则

分配给默认分区的内存量至少应为可用内存总量的 50%，原因如下：

1. 在未来为客户提供灵活性，以便在达到限制时增加其他分区的内存。
2. 所有分区的集成缓存内存从默认分区中获取。

PE 可消耗的总内存为 4 GB。所以总共 2 GB 可以分配给除管理分区以外的所有分区。

分配给管理分区的内存有两种用途：

1. 存储静态对象（配置、SSL 密钥）
2. 动态对象 — 具体取决于启用的功能列表以及为动态对象分配的内存连接数量不同

ADC 管理员使用应用程序所有者的连接和带宽要求以及以下准则来计算内存估计值。

为配置分配静态内存的指导原则

表 1 列出了常用配置和所需内存。

表 **1**

配置类型	每个数据包引擎分配的内存（以 KB 为单位）
添加 SNIP	255
添加 IPv4 服务器	0.384
添加服务	5.253
添加带有服务的虚拟服务器	11.157
将 vlan 绑定到分区	0.116
向分区添加路由	0.564
添加 acl	0.5
添加监视器	4.34

配置类型	每个数据包引擎分配的内存（以 KB 为单位）
添加服务组	4.625
将服务器绑定到服务组	5.817
添加 cs 操作	4.532
add cs policy	2.548
添加 cs 虚拟服务器	11.589
将 cs 策略绑定到 cs 虚拟服务器	7.348

这些配置在 PE 之间进行复制，因此上述要求需要乘以 PE 的数量。

动态存储器指南

表 **2**

功能	内存要求
连接（仅当 Citrix ADC 版本为 12.0 及更高版本时适用）	每 1000 个连接有 2.4 MB
持续会话	每 1000 个会话中有 600 KB
GSLB 永久会话	每 1000 个会话 6 MB
SSL	6 MB 用于 SSL 卸载中的 1000 个 SSL 连接/会话，9 MB 用于端到端 SSL 中的 1000 个 SSL 连接/会话
AAA — 取决于用户数量	用户数量 * 2 KB
重写 — 获取重写策略将解析的最大长度	连接数 * 最大长度
响应程序 — 获取响应程序策略将解析的最大长度	连接数 * 最大长度
TCP 缓存	20% 的连接 * 已配置 TCP 缓冲区大小

动态内存 = 从上表中每一行计算出的内存总和。

将 10-20% 的缓冲区添加到计算的总内存中。

没有提供某些功能（如 AppQoE）的内存要求，因为从分区内存消耗的内存对于这些功能来说可以忽略不计，并且 10-20 %的缓冲区足以处理它们。

总内存 = 静态内存 * PE 数 + 动态内存

假设我们得出结论，所需的内存是 1 GB，数据包引擎的数量是 4。然后，对于该特定分区，所需的内存量由以下公式派

生：

管理分区内存配置 =（所需内存量/数据包引擎数）

管理员分区内存 = 1GB/4 = 250 MB

达到资源限制时的行为

1. 连接 — 新连接将被删除
2. 带宽 — 新流量将被丢弃
3. 内存 — 新流量将被丢弃

您可以配置 SNMP 警报，如果特定分区的资源耗尽，则会触发这些警报。SNMP 陷阱列表在附加资源部分中给出。

网络平面

VLAN：配置不同的 VLAN 并将其分配给管理分区，以维护网络级隔离。

路由：每个分区的路由配置是唯一的。

ADC 管理员与网络管理员协商（通过应用程序管理员的输入），根据网络拓扑定义 VLAN 和与路由相关的配置。

L3 参数：可以是特定于分区的。一些 L3 参数是丢弃 DF 数据包、ICMP 错误阈值、覆盖等，并且输入应来自网络或 ADC 管理员。

控制平面：用户体验

管理分区在不同级别提供隔离，允许用户安全地管理隔离的 ADC 实例。

不同的隔离级别包括：

1. UI 页面 — 配置，仅显示分区的统计信息
2. 诊断 — 跟踪隔离。跟踪不会捕获其他分区的流量
3. SNMP 警报-在分区级别配置
4. 日志级隔离

可以使用以下方法配置 UI 级别隔离：

1. 在相应的分区中，为一个 SNIP 启用 Mgmt. 访问权限，并使用该 SNIP 访问 GUI。这将提供 UI 级别的隔离和仅对该分区的可见性。

表 **3**

日志类型	特定于分区
博文	是

日志类型	特定于分区
技术支持捆绑包	是
审计日志	否
/var/log	否

企业用例的管理分区

本节介绍企业客户用例，其中包含四个使用管理分区的应用程序。

客户要求

- 需要托管 4 个应用程序
- 每个应用程序都有自己的管理员和一组不同的 ADC 要求。下表列出了应用程序及其独特要求。

表 **4**

应用程序	特征	要求/特征
SharePoint	共享文件、音频、文件等	缓存、压缩、身份验证、SSL 卸载、SSL 配置文件
数据库	自定义 SQL 规则，身份验证，在读取和写入之间拆分以获得更好的性能	内容切换，针对 SQL 相关关键字的策略基础
企业网站	公共访问-容易受到攻击，应用程序防火墙	DDoS、AppQOE、AppFW、SSL 配置文件
Outlook	与广告、SSO 集成，在 HTTP 中更好的性能	身份验证 SSO，SSL 卸载

从上述要求表中可以清楚地看出，每个应用程序都需要一组不同的配置来实现 Citrix ADC 的全部优势。建议对 Citrix ADC 进行分区，并将这些分区分配给各自的应用程序所有者。

带宽和连接估计

Outlook 和 **SharePoint**

SharePoint、Exchange 和 Lync 等企业应用程序的带宽取决于：

1. 并发用户数
2. 使用类型
    a) 交换 — 消息的平均大小和数量
    b) SharePoint — 文件类型、读取与写入的比率

应用程序管理员使用上述两个因素计算带宽要求，并向 Citrix ADC 管理员提供有关配置管理分区的信息。Microsoft technet 和 MSDN 博客 中提供了有关如何计算带宽的广泛准则。

示例：

Outlook 2010 的带宽：用户类型（轻型、中型、重型等）。对于中等用户，发送 10 封电子邮件，接收 40 封电子邮件，平均邮件大小 50 KB = 2.15 Kbps。对于 1000 个用户，所需的带宽为 2150 Kbps。

用于 SharePoint 的带宽：用户数量 = 1000。假设 20% 的用户在任何时间点处于活动状态，平均页面加载大小为 100 KB，并且在 1 小时内访问大约 10 个页面：

= 100 KB * 200 * 10/小时 = 200000 KB/小时 = 200000*8（8 位/字节）/3600（秒数）

= 444 Kbps

每秒连接数 = 活动用户数 * 10

MSSQL

根据查询速率和响应大小，得出带宽和连接。

企业网站

带宽要求：平均页面大小* 随时最大用户数 * 2

连接：最大用户数 x 每个用户的连接数

示例：

带宽：4 KB*10002* = 48000 Kbps

最大用户数 = 1000，每个用户的连接数 = 10。连接 = 1 万

如果大多数用户来自 HTTP/1.1，则每个用户的连接数将为 2–3，但如果混合更倾向于 HTTP/1.0，则连接数将为 10–15。根据流量/客户端组合的不同，每个用户的连接数的乘法系数为 3—15。

要配置的内存取决于：

1. 各自管理分区中的配置列表 — 静态内存。有关详细信息，请参阅表 1。
2. 动态内存 — 连接数和连接类型（HTTP 与 SSL）— 详情请参阅表 2。
3. 数据包引擎的数量。内存 =（静态内存 + 动态内存）/（数据包引擎的数量）

ADC 管理员的步骤

1. 收集每个应用程序的带宽和连接
2. 分别为 SharePoint、数据库和 Outlook 创建三个分区。使用上一步的带宽和连接，并将其分配给相应的分区。如果客户需要 AppFW，则企业网站可以托管在默认分区上，因为 AppFW 仅在默认分区上受支持。
3. 为每个分区创建用户并共享凭据。
4. 启用集成缓存并设置缓存内存。缓存内存从默认分区中配置的缓存内存中获取。有关分配的详细信息，请参阅 IC 的附录部分。
    a) 咨询 ADC 管理员后分配缓存内存。尝试分配系统中总缓存内存的 30— 40%。如果分配的总数为 10 GB，则为 SharePoint 分区中的缓存分配大约 3-4 GB。
    b) 应用程序所有者应首先监视缓存统计信息，以检查优势级别。
    c) 检查缓存对象命中率，如果大量缓存对象的命中率高，则增加该特定分区的 IC 内存大小。
5. 启用压缩
    a) SharePoint 将发布不同类型的文件（Excel、PowerPoint、Word），而相同的文件，如果压缩并交付给客户端，则会降低带宽使用率。

数据库用户

1. 配置 CS、VIP 和后端服务器。
2. 使用内容切换拆分读/写请求并重定向到相应的服务器集。

企业网站

1. 配置 VIP 和后端服务器。
2. 启用集成缓存。
    a) 企业网站位于默认分区中，因此来自其他分区的未使用缓存内存可用于企业网站。因此，假设 SharePoint 和 Outlook 每个消耗 35%，那么总消耗将为 70%，剩余的 30% 留给默认分区（企业网站）。如果总缓存内存为 10 GB，则默认分区将具有 3 GB 的缓存内存。
    b) 应用程序所有者应首先监视缓存统计信息，以检查优势级别。
    c) 检查缓存对象命中率，如果大量缓存对象的命中率高，则增加该特定分区的 IC 内存大小。
3. 启用前端优化。
4. 启用 AppFW。

服务提供商管理分区使用案例

服务提供商托管 Microsoft 应用程序，并将 IIS、SharePoint 和 MSSQL 应用程序作为服务提供。他们的客户通常具有以下要求：

客户要求

- 客户 1：访问数据库服务器，其读/写拆分为 90:10，最终客户希望配置自定义 SQL 相关筛选器
- 客户 2：通过 SSL 访问 Web 应用程序，最终客户希望控制其 SSL 证书
- 客户 3：从服务提供商访问托管的 SharePoint

服务提供商为其客户托管一个门户，以便：

1. 选择要托管的应用程序
2. 带宽要求

服务提供商为其客户托管一个门户，以便：

1. 选择要托管的应用程序
2. 带宽要求
3. 连接

根据选择，服务提供商可以使用 NITRO API 配置与后端特定应用程序相关的配置相应的分区。

根据客户选择的应用程序，选择相应的选项。

1. 使用 SSL 的 Web 应用程序
    a) 要绑定到 VIP 的 SSL 证书选项
    b) HTTP 到 HTTPS 重定向
    c) SSL 配置文件相关参数
2. SQL
    a) 客户想要配置的 SQL 相关过滤器
3. SharePoint
    a) 缓存内存限制和规则
    b) 压缩策略

服务提供商遵循两个选项之一来实现管理分区创建后的确切要求。

配置选项 **1**：

服务提供商收集客户的请求，并在相应的分区上执行这些请求。

配置选项 **2**：

使用 NITRO API 自动执行管理分区。输入可以从前端门户收集，并在后端执行 NITRO API 来配置分区。

功能注意事项

功能支持：大多数功能都支持管理分区，只有少数功能不支持。有关确切的列表，请参阅 Citrix Docs 并签入特定的软件版本。它将包含一个列出可支持性列表的表格。

配置限制。管理分区在以下项目中不受支持：

1. 群集

2. MPX-FIPS 装置

结论

管理分区的主要优势是在软件级别实现 ADC 的分离，并为每个分区所有者提供安全、隔离的用户体验。

其他资源

故障排除工具

管理分区中的常见问题：

ESX 上 **VPX** 上的管理分区：

- 配置自定义 MAC 地址时，无法访问非默认分区。
- 解决方案：需要在 ESX 上启用混杂模式，才能使非默认分区工作。

配置失败：

- 配置可能无法引发错误输入文件不存在。
- 需要使用相对路径，而不是绝对路径。

VLAN 配置：

- 管理分区 VLAN 支持标记的 VLAN，因此，在标记 VLAN 时，Citrix ADC 接口所连接的交换机应配置相应的 VLAN。对于未标记的 VLAN，请使用共享 VLAN 配置

集成缓存内存分配

要在分区 Citrix ADC 上配置集成缓存 (IC)，在默认分区上定义 IC 内存后，超级用户可以在每个管理分区上配置 IC 内存，使分配给所有管理分区的 IC 内存总量不超过默认分区上定义的 IC 内存。未为管理分区配置的内存仍可用于默认分区。

例如，如果具有两个管理分区的 Citrix ADC 装置具有 10 GB 的 IC 内存分配给默认分区，并且两个管理分区的 IC 内存分配如下：

- 分区 1：4 GB
- 分区 2：3 GB

然后，默认分区有 10-（4 + 3）= 3 GB 可供使用的 IC 内存。

> 注意：
>
> 如果管理分区使用了所有 IC 内存，则没有可用于默认分区的 IC 内存。

用于检查内存使用情况的命令

- 分区内的 Sat 系统内存将显示分区的聚合系统级内存分配，stat 分区名称将显示分区内使用的内存百分比。

```
>add partition p1
Done
>switch partition p1
Done
p1> stat system memory
done

Citrix ADC Memory Information:
Maximum Memory Available (MB): 50
Memory Currently Available (MB): 50
Memory Allocated (MB) 7
Memory Allocated (%) 14.95
InUse Memory  (MB) 7
InUse Memory (%) 14.95
Free Memory (MB) 42

>stat partition p1

Partition(s) Summary
   MinBW MaxBW MaxConn MaxMem

p1 10240 10240  1024  10

Partition Stats:

                       Rates (/s)   Total
Current Bandwidth         --          0
Current Connections       --          0
Memory Usage (%)          --          14
Total Packet Drops        0           7
Total Drops (KB)          0           0
Total Connection Drops    0           0
<!--NeedCopy-->
```

- 配置内存：由于每个配置都在每个数据包引擎中复制，因此内存会在每个数据包引擎中分配。例如，如果 “add lb vserver” 命令在 peach Packet Engine 中大约需要 10 KB，并且我们在 5 – Packet Engine 系统中创建了 10 MB 分区，则将总共消耗 50 KB 的分区内存。
- 通过在 Citrix ADC 外壳上应用配置并运行以下命令，可以测量规范配置所需内存的精确值：


```
root@ns# nsconmsg -s nsppeid=0 -s nspartid=1 -g mem_cur_usedsize -d
   current
Displaying performance information
Citrix ADC V20 Performance Data
Citrix ADC NS11.0: Build 65.572.nc, Date: Apr 7 2016, 10:32:51

reltime:mili second between two records Thu Feb 23 13:45:18 2017
 Index rtime totalcount-val delta rate/sec symbol-name&device-no
    0  22681     1597631     8965  5333     mem_cur_usedsize
       partition_ctx(p1) (PART-1)
<!--NeedCopy-->
```

在此实验中，在 **PPE-0** 中为分区 **ID 1** 使用大约 **9 KB** 的内存。在 **Citrix ADC** 上配置的每个分区都有一个唯一的 **ID**。

以下命令允许测量给定分区的完整系统（包括所有数据包引擎）的内存估计。

```
root@ns# nsconmsg -s nspartid=1 -g mem_cur_used -d current
Displaying performance information
Citrix ADC V20 Performance Data
Citrix ADC NS11.0: Build 65.572.nc, Date: Apr 7 2016, 10:32:51

reltime:mili second between two records Thu Feb 23 13:44:27 2017
Index rtime totalcount-val delta rate/sec symbol-name&device-no
 0   7000    7881865       6403    5333    mem_cur_usedsize
     partition_ctx(p1) (PART-1)
<!--NeedCopy-->
```

Citrix ADC 12.0 中引入的 **SNMP** 陷阱的列表

陷阱名称	说明
partitionCONNLimitExceeded	分区的连接限制已耗尽，新连接正在丢弃
partitionCONNLimitNormal	分区现在可以接受新连接
partitionBWLimitExceeded	分区的 BW 限制已耗尽，数据包被丢弃
paritionBWThresholdReached	当前 BW 使用量大于等于 80%
partitionCONNThresholdReached	当前活动连接计数 >= 80%
paritionCONNThresholdNormal	当前活动连接计数小于 60%
partitionMEMThresholdReached	PE 的当前内存使用率 >= 80%

陷阱名称	说明
partitionMEMThresholdNormal	PE 的当前内存使用率 <= 60%
partitionMEMLimitExceeded	PE 的当前内存使用率 >= 95%

其他参考资料

交换客户端网络带宽计算器测试版

运行 Microsoft Online Services 需要多少带宽

Citrix Gateway SaaS 和 O365 云 Citrix 验证的参考设计

May 20, 2020

概述

软件即服务 (SaaS) 是一种软件分发模式，可作为基于 Web 的服务远程交付软件。常用的 SaaS 应用程序，包括 Microsoft Office 365 订阅。

现在可以使用 Citrix Gateway 服务使用 Citrix Workspace 访问 SaaS 应用程序。Citrix Gateway 服务与 Citrix Workspace 相结合，可为已配置的 SaaS 应用程序、已配置的虚拟应用程序或任何其他工作区资源提供统一的用户体验。

使用 Citrix Gateway 服务交付的 SaaS 应用程序可为您提供简单、安全、可靠且可扩展的解决方案来管理应用程序。在云上交付的 SaaS 应用具有以下优势：

配置简单 - 易于操作、更新和使用。
单点登录 — 通过单点登录轻松登录。
适用于不同应用的标准模板 — 基于模板的热门应用配置。

Citrix Gateway SaaS 应用程序

在应用程序详细信息部分中，填写以下内容：

- 位置 = 我的公司网络外部
- 名称 = Office 365 * URL = `https://login.microsoftonline.com/login.srf`
- 相关域：*.login.microsoftonline.com

- 说明 = (默认值)

在单点登录部分中，按如下所示进行填写：

- 断言 URL = `https://login.microsoftonline.com/login.srf`
- 受众 = urn:federation:MicrosoftOnline
- 名称 ID 格式 = 静态
- 名称 ID = Active Directory GUID
- 高级属性：
  - 属性名称：`IDPEmail`
  - 属性格式：未指定
  - 属性值：`Email`

O365 SaaS 应用程序联合到 Citrix Gateway

在 Microsoft 云上配置 FEDERATED 模式的 PowerShell 命令：

- PS> `connect-msolservice`

> 注意：应使用 Microsoft 云帐户连接到 `msolservice`。
>
> 例如：*admin.user@onmicrosoft.com*

- PS> `Install-Module AzureAD -Force`
- PS> `Import-Module AzureAD -Force`
- PS> `Install-Module MSOnline -Force`
- PS> `Import-module MSOnline -Force`

配置 Citrix Gateway 客户订阅所独有的联合身份验证设置：

- PS> `$dom = "ad-domain.com"`

> 注意：
> *ad-domain.com* 命名空间为用户身份验证域

- `PS> $fedBrandName = "CitrixNS(TME)"`
- `PS> $url = "https://customerID.cloud.com/cgi/tmlogout"`
- `PS> $uri = "https://citrix.com/customerID"`
- `PS> $ecpUrl = "https://customerID.cloud.com/saml/login"`

> 注意：
> *`customerID` 为 Citrix Workspace URL*

从 Citrix Gateway 提供 SAML IdP 证书：

- `PS> $cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2("c:\cert\saml_idp.crt")`
- `PS> $certData = [system.convert]::tobase64string($cert.rawdata)`

执行 PS 字符串以完成到 Citrix Gateway 的 `msol` 联合：

- `PS> Set-MsolDomainAuthentication -DomainName $dom -federationBrandName $fedBrandName -Authentication Federated -PassiveLogOnUri $uri -SigningCertificate $certData -IssuerUri $uri -ActiveLogOnUri $ecpUrl -LogOffUri $url -PreferredAuthenticationProtocol SAMLP`

验证域联盟和设置是否已完成：

- `PS> Get-MsolDomainFederationSettings`

```
DomainName: customerID.com

ActiveLogOnUri https://customerID.cloud.com/saml/login

FederationBrandName customerID(TME)

IssuerUri https://citrix.com/customerID

LogOffUri https://customerID.cloud.com/cgi/tmlogout

PassiveLogOnUri https://citrix.comcustomerID

SigningCertificate MIIG3zCCBMegAwIBAgIJAMUTG1zqJgUZMA0GCSqGSIb3DQEBCwUAMIGcMQswCQYD
```

Office 365 套件应用程序

- Outlook `https://outlook.office365.com/`
- OneDrive for Business `https://customerid.sharepoint.com/`
- Word `https://office.live.com/start/Word`
- Excel `https://office.live.com/start/Excel`
- PowerPoint `https://office.live.com/start/PowerPoint`
- OneNote `https://www.onenote.com/`
- SharePoint `https://customerid.sharepoint.com/`
- Teams `https://teams.microsoft.com/`
- Yammer `https://www.yammer.com/office365`

- Dynamics 365 https://customerid.dynamics.com/
- Flow https://flow.microsoft.com/

引用链接

Azure PowerShell 模块参考

Azure PowerShell 命令参考

在 Microsoft Azure 中部署 Office 365 目录同步

具有访问控制的 **Citrix Gateway** 服务 **SSO Citrix** 验证的参考设计

May 20, 2020

Citrix Gateway 服务

Citrix Gateway 服务是一项 Citrix 产品，提供身份验证、单点登录，并实现快速、安全地交付 Citrix VDI 和 SaaS 应用程序。

Citrix Gateway 服务还向 SaaS 和 Web 应用程序提供 SSO。软件即服务 (SaaS) SSO 功能是 Citrix Cloud 中基于云的完全托管服务，可为公共托管 SaaS 应用程序和企业托管 Web 应用程序提供远程访问和单点登录。

现在可以使用用户 Workspace 订阅中的 Citrix Gateway 服务访问 SaaS 应用程序。Citrix Gateway 服务提供对公有托管外部 SaaS 应用程序提供商中运行的第三方 SaaS 应用程序的身份验证访问。

Citrix Gateway 服务与 Citrix Workspace 相结合，可为已配置的 SaaS 应用程序、已配置的虚拟应用程序或任何其他工作区资源提供统一的用户体验。

使用 NetScaler Gateway Service 交付的 SaaS 应用程序可为您提供简单、安全、可靠且可扩展的解决方案来管理应用程序。在云上交付的 SaaS 应用具有以下优势：

- 配置简单 - 易于操作、更新和使用。
- 单点登录 — 通过单点登录轻松登录。
- 适用于不同应用的标准模板 — 基于模板的热门应用配置。

Citrix Gateway 服务功能

- 简单性：使用基于云的产品降低 NetScaler 部署和管理复杂性
- 始终保持最新状态：使用始终最新的产品简化 Citrix Gateway 的管理

- 安全性和高可用性：提高 XenApp 和 XenDesktop 服务的安全性和可用性
- 速度：提供更快、更轻松的部署和管理 Citrix Gateway 的方法
- 便利性：网关服务一起打包和销售，以简化 IT 最常见的使用案例的解决

Citrix Gateway 服务通过安全的远程访问
降低成本、简化管理并改善用户体验。

访问控制服务概述

使用访问控制服务，管理员可以提供一致的体验，将单点登录、远程访问和内容检查集成到单个解决方案中，以实现端到端访问控制。IT 管理员可以通过简化的单点登录体验来管理对已批准的 SaaS 应用程序的访问。通过访问控制服务，管理员还可以通过筛选对特定网站和网站类别的访问来保护组织的网络和最终用户设备免遭恶意软件和数据泄漏。管理员可以强制执行增强的访问安全策略，以安全访问 SaaS 应用程序。经过身份验证后，员工可以从任何设备访问所有关键业务应用程序，无论他们是在办公场所、家中还是出行。

管理员可以监视用户活动，例如

- 恶意、危险或未知网站访问
- 消耗的带宽
- 有风险的下载和上传行为。

使用围绕访问的网站和网站类别的 Analytics，管理员可以采取纠正措施来保护企业网络。同时，该服务还为最终用户提供对其所有托管应用程序的无缝安全访问。

管理员还可以限制操作，例如受限打印、下载和剪贴板访问（复制粘贴）。

下图是访问控制服务的可视化描述。

具有访问控制功能的网关服务

您可以使用访问控制服务完成的一些关键任务如下所示：

- 通过单点登录访问发布 SaaS 应用程序。
- 为 SaaS 应用设置增强的安全策略。（例如，水印、复制粘贴限制和阻止下载。）
- 为要阻止的网站类别和网站定义访问策略。
- 定义要重定向到 Secure Browser 服务的 Web 站点类别和 Web 站点的访问策略。
- 了解 SaaS 应用环境中的用户和网站活动，并将其与定义的策略相关联。
- 进行策略更改以允许或阻止网站访问，并在安全的浏览器服务会话中启用访问。

Citrix Gateway Service SaaS 发布步骤

支持软件即服务应用

通过四个简单的步骤入门

1. 注册 Citrix Cloud
2. 申请 NetScaler Gateway Service 试用版
3. 已设置 NetScaler Gateway Service
4. 访问 NetScaler Gateway Service 用户界面

在此处开始使用 Citrix Cloud

在此处开始使用 Citrix Workspace

Citrix Gateway Service SaaS 应用程序配置

在此示例中，我们将介绍使用 SalesForce.com SaaS 应用程序
配置 Citrix Gateway 服务所需的配置步骤。

配置最终用户对已配置的 **SaaS**、**Web** 和虚拟应用程序的访问权限

配置 Workspace 以安全地提供从任何设备对应用程序的访问权限。转到 Workspace 配置
从库管理和添加 SaaS 应用程序转到库 | 添加 SaaS 应用程序

要从 **Citrix Gateway** 服务应用程序目录添加 **SaaS** 应用程序，请完成以下步骤

通过以下网址访问 **Citrix** 订阅实例

Citrix Cloud 帐户登录 并提供您的组织登录凭据。

1. 从 Citrix Cloud 管理门户启动 **Citrix Gateway** 服务磁贴。
2. 启动 “入门” 链接以配置 SSO SaaS 应用程序。
3. 从 “应用程序目录” 列表中选择 SaaS 应用程序模板。

   在此示例中，我们将 SSO 的 Salesforce 配置为 Workspace SaaS 应用程序。
4. 完成所需的 SaaS 应用程序特定参数：

   > 注意：
   >
   > 在此示例中，我们选择 “在我的公司网络之外”，因为这是由第三方应用程序订阅托管的 SaaS 应用程序。
5. 管理 Workspace 的 SaaS 应用程序订阅者。

6. 从用户域将用户分配到 SaaS 应用程序。

> 注意：
>
> 您可以通过以下凭据对 Workspace 进行身份验证：
>
> - Windows Active Directory
> - Azure Active Directory

SaaS 应用程序配置的访问控制

Citrix 访问控制 (CAC)，它基于 Gateway 服务中包含的 SSO 和多因素身份验证 (MFA) 功能构建，为 SaaS 和 Web 应用程序的访问和使用提供更精细的策略控制。CAC 结合基于用户行为分析及其风险分数的高级分析，增强了向企业最终用户提供安全数字工作区的整体安全态势。

访问控制增强的安全性设置

- 启用增强安全性：在 Citrix 嵌入式浏览器中启动和监视 Web 或 SaaS 应用程序，并将未知流量路由到访问控制。
- 限制剪贴板访问：禁用应用程序和系统剪贴板之间的剪切/复制/粘贴操作
- 限制打印：禁用从应用程序浏览器中打印功能。
- 限制导航：禁用下一个/后退应用程序浏览器按钮。
- 限制下载：禁用用户从应用程序内下载的功能。
- 显示水印：在用户屏幕上显示水印，显示用户计算机的用户名和 IP 地址。

内容访问设置的访问控制

配置 Web 筛选以允许/阻止最终用户访问，并将其重定向到 Citrix Secure Browser 服务。

- 选择配置内容访问
- 选择编辑
- 启用筛选网站列表
  - 添加/删除阻止或允许的网站
  - 添加/删除阻止或允许的网站分类

启动具有访问控制的 Citrix Workspace 应用程序

Workspace SaaS 应用程序位于以下适用于美国-美洲托管订阅的 FQDN：

Citrix Cloud 帐户登录

可以通过 Workspace 应用程序访问您的工作区体验，该应用程序有三种风格：

- 桌面 (Windows/Mac)

- 移动设备 (iOS/Android)
- Web (HTML5)

1. 使用 Web 浏览器，连接到 Workspace URL。
2. 在 Workspace 中选择 SaaS 应用程序磁贴。
3. 应用程序在浏览器选项卡中无缝启动，并使用本机 SSO。

引用链接

具有 **Azure** 负载平衡器前端 **IP** 验证的参考设计的 **Citrix ADC** 高可用性

March 2, 2021

概述

利用 Azure 负载平衡器 (ALB) 作为前端 (FE) 负载平衡器，在 Microsoft Azure 中实施 Citrix ADC 高可用性部署。

您可以在 Azure 上的主动-被动高可用性（高可用性）设置中部署具有多个 NIC 的一对 Citrix ADC 虚拟设备。每个 NIC 可以包含多个 IP 地址。

在 **Azure** 服务管理中以高可用性模式配置 **Citrix ADC VPX**

主被模式提供故障转移功能。在此模式下，VPX 实例将同步其配置状态。当主实例失败时，辅助实例将接管工作。

有关 Citrix ADC 设备中的高可用性的信息，请参阅 高可用性

在 Microsoft Azure 部署中，使用 Azure 负载均衡器实现了两个 Citrix ADC 虚拟机的高可用性配置，该负载均衡器将客户端流量分配到两个 Citrix ADC 实例上的虚拟服务器上。有两种类型的 Azure 负载平衡器可用于高可用性：

Azure 外部负载均衡器：如果客户端流量来自 Internet，则必须在 Internet 和 Citrix ADC VPX 实例之间部署外部负载均衡器以分配客户端流量。

Azure 内部负载平衡器：如果客户端流量源自云服务，或者由云服务中的网关或防火墙转发，则必须部署内部负载平衡器来分配客户端流量。

要在 Azure 上实现高可用性，必须将两个 Citrix ADC 虚拟机添加为负载平衡集并配置端点。

Citrix ADC 主动-被动部署假设

- 高可用性独立网络配置 (INC) 配置
- Azure 负载平衡器 (ALB) 在直接服务器返回 (DSR) 模式下

- 所有流量都通过主节点。
- 在主节点发生故障前，辅助节点一直处于备用模式。

> 注意：
>
> 要在 Azure 云上执行 Citrix ADC 高可用性部署，您需要一个可在两个 Citrix ADC 高可用性节点之间移动的浮动公共 IP (PIP)。Azure 负载平衡器 (ALB) 提供浮动 PIP，在发生故障转移时，该浮动 PIP 会自动移动到第二个节点。
>
> 浮动 *IP* 设置是在 *ALB* 负载平衡规则中配置的，如 ***ALB*** 配置部分下的步骤 *4* 中所定义的。

Citrix ADC IPSET 功能概述

IP 集是一组 IP 地址，它们在 Citrix ADC 设备上配置为子网 IP 地址 (SNIP) 或虚拟 IP 地址 (VIP)。IP 集通过有意义的名称进行标识，这些名称有助于确定其中所含 IP 地址的用途。要创建 IP 集，请添加 IP 集并将 Citrix ADC 拥有的 IP 地址绑定到该 IP 集。SNIP 地址和 VIP 地址可以存在于同一个 IP 集中。

Azure 负载平衡器概述

使用 Citrix ADC 12.1 高可用性 Azure Resource Manager (ARM) 模板部署 Citrix ADC 高可用性实例。

此模板将指导 Citrix ADC 高可用性主动-被动模式的部署。预配置为包含组件和设置，以提供无缝的高可用性体验。有关拓扑的详细信息，请参阅 高可用性。

成功部署后，将在 HA-INC 模式下预配置一对 Citrix ADC 装置。Citrix ADC VPX 高可用性模板支持 Citrix ADC 的不同 SKU，例如 BYOL 和每小时许可证，例如 VPX 10、VPX 200、VPX 1000 和 VPX 3000。

> 注意：
>
> 适用于 Citrix ADC 的 ARM 模板包含特定的 Azure 负载平衡变量作为资源。

部署的配置先决条件

- Azure 负载平衡器配置
- Citrix ADC 配置

Azure 负载平衡器配置

1. 为将通过 Azure 负载平衡器提供的每个 Citrix ADC 服务添加前端 IP 地址。
2. 为每个应用程序添加 alb 后端池。
3. 为每个应用程序添加 alb 运行状况探测。
4. 添加 alb 负载平衡规则。

5. 向网络安全组 (NSG) 添加一个或多个入站安全规则

Citrix NetScaler 配置

NetScaler 需要添加 IPSETS 以将 Citrix ADC 资源映射到 Azure 前端 IP 配置。

> 注意：对于需要来自 ALB 的前端公有 IP 的每个 VIP，
>
> 重复以下步骤。

1. 将 Azure 前端公有 IP 地址添加到 Citrix ADC

   ```
   add ns ip 23.99.xx.xx 255.255.255.255 -type vip (Azure Frontend Ip)
   ```

2. 为 Azure 前端 IP 在 Citrix ADC 上创建并绑定 IPSET

   ```
   add ipset net_1
   bind ipset net_1 23.99.xx.xx
   ```

3. 使用 IPSET 更新 Citrix ADC VIP

   ```
   set lb vserver net_1 -ipset net_1
   ```

验证端口与 ALB 的连接

使用类似 `https://ping.eu/port-chk/` 或类似的工具来验证 ALB 和 Citrix ADC 服务是否可用。

```
IP address or host name:
23.99.xx.xx
Port number:”80, 443. etc”
23.99.xx.xx:80 port is open
<!--NeedCopy-->
```

故障排除

- `nstcpdump` - 验证 Citrix ADC 前端 IP 配置
- `nstrace` - 验证 ALB 运行状况探测

XenDesktop 7 的数据库大小调整工具

May 20, 2020

当前，XenDesktop 7 的数据库调整大小取决于能否解释和理解数据库大小知识库文章 CTX139508。如果您知道您在列出的环境中存在变体，这并不起作用。为了提供帮助，我们的主要软件工程师 Chris Gilbert 创建了一个简单的工具，可帮助生成定制尺寸信息。

为什么是工具?

许多人都要求使用 Excel 文件或简单的公式来计算数据库大小，但根据复杂程度和所涉及的因素，这些方法并不是最佳的。

该工具隐藏了计算的复杂性，并允许 XenDesktop 7.5 和 7.6 之间的差异。它消耗和显示的数据是相当原始的，但欢迎提供改进的反馈。

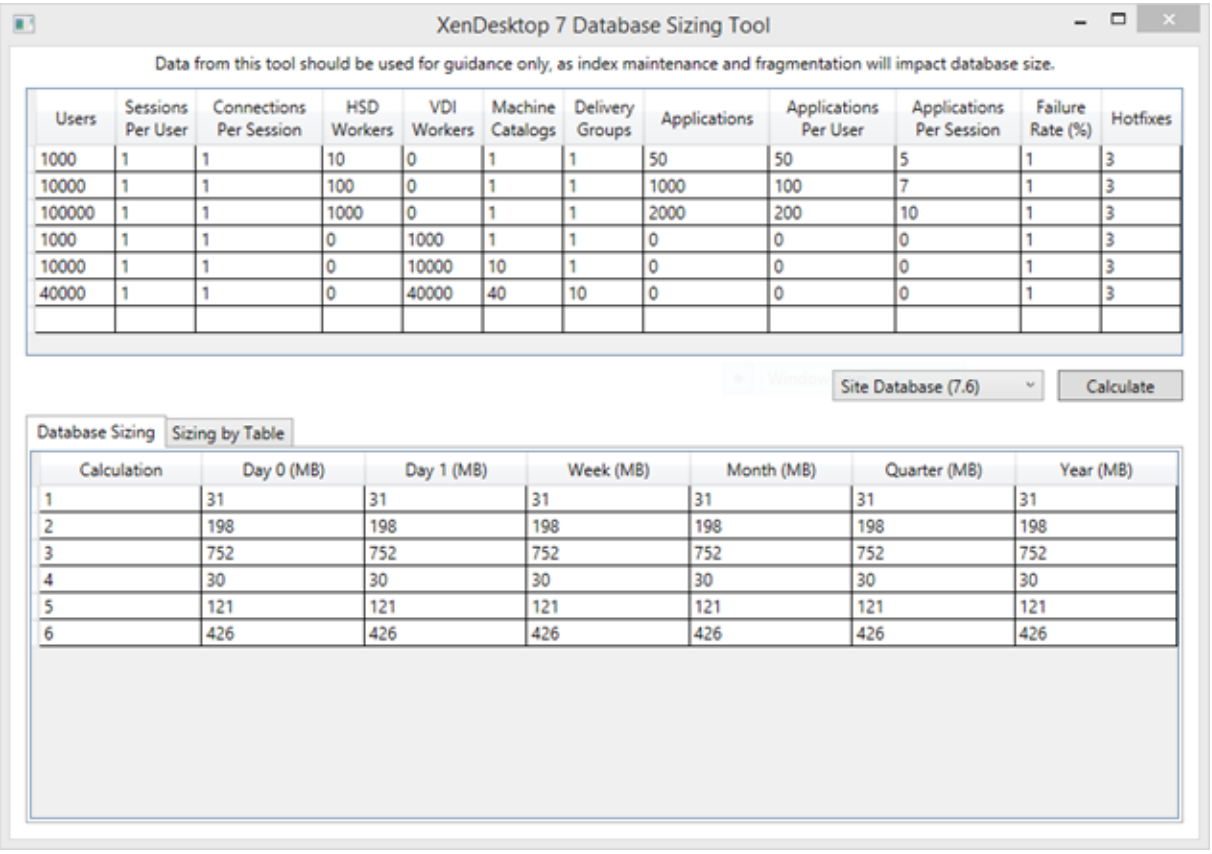
下载信息

该工具是一个压缩的 MSI 文件，所以它很容易安装和卸载。该工具具有的唯一依赖关系是.NET 4.0；它不需要 XenDesktop 的任何部分。

从 CTX209080 下载该工具。

如何使用该工具

启动该工具时，您将看到一个窗口，其顶部有一个部分，允许输入有关预期环境的参数。默认设置为各种大小的 VDI 和 HSD，应类似于以下屏幕截图：



Users	Sessions Per User	Connections Per Session	HSD Workers	VDI Workers	Machine Catalogs	Delivery Groups	Applications	Applications Per User	Applications Per Session	Failure Rate (%)	Hotfixes
1000	1	1	10	0	1	1	50	50	5	1	3
10000	1	1	100	0	1	1	1000	100	7	1	3
100000	1	1	1000	0	1	1	2000	200	10	1	3
1000	1	1	0	1000	1	1	0	0	0	1	3
10000	1	1	0	10000	10	1	0	0	0	1	3
40000	1	1	0	40000	40	10	0	0	0	1	3

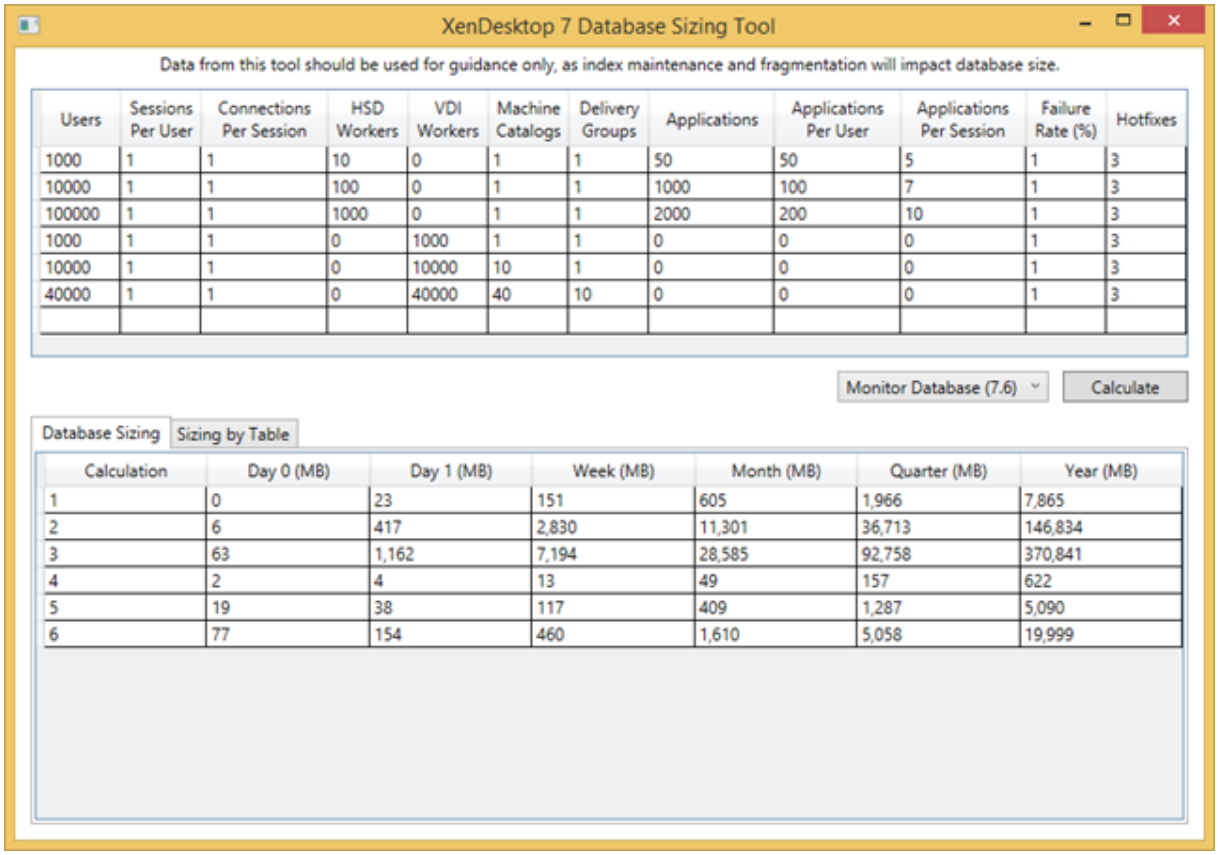
Calculation	Day 0 (MB)	Day 1 (MB)	Week (MB)	Month (MB)	Quarter (MB)	Year (MB)
1	31	31	31	31	31	31
2	198	198	198	198	198	198
3	752	752	752	752	752	752
4	30	30	30	30	30	30
5	121	121	121	121	121	121
6	426	426	426	426	426	426

您可以更新其中一行，或者只是开始在空白底行中键入数字，它会添加更多行。

如果您然后选择数据库类型和 XenDesktop 版本，然后单击“计算”，程序将运行数学并生成上述屏幕截图底部显示的大小指导。

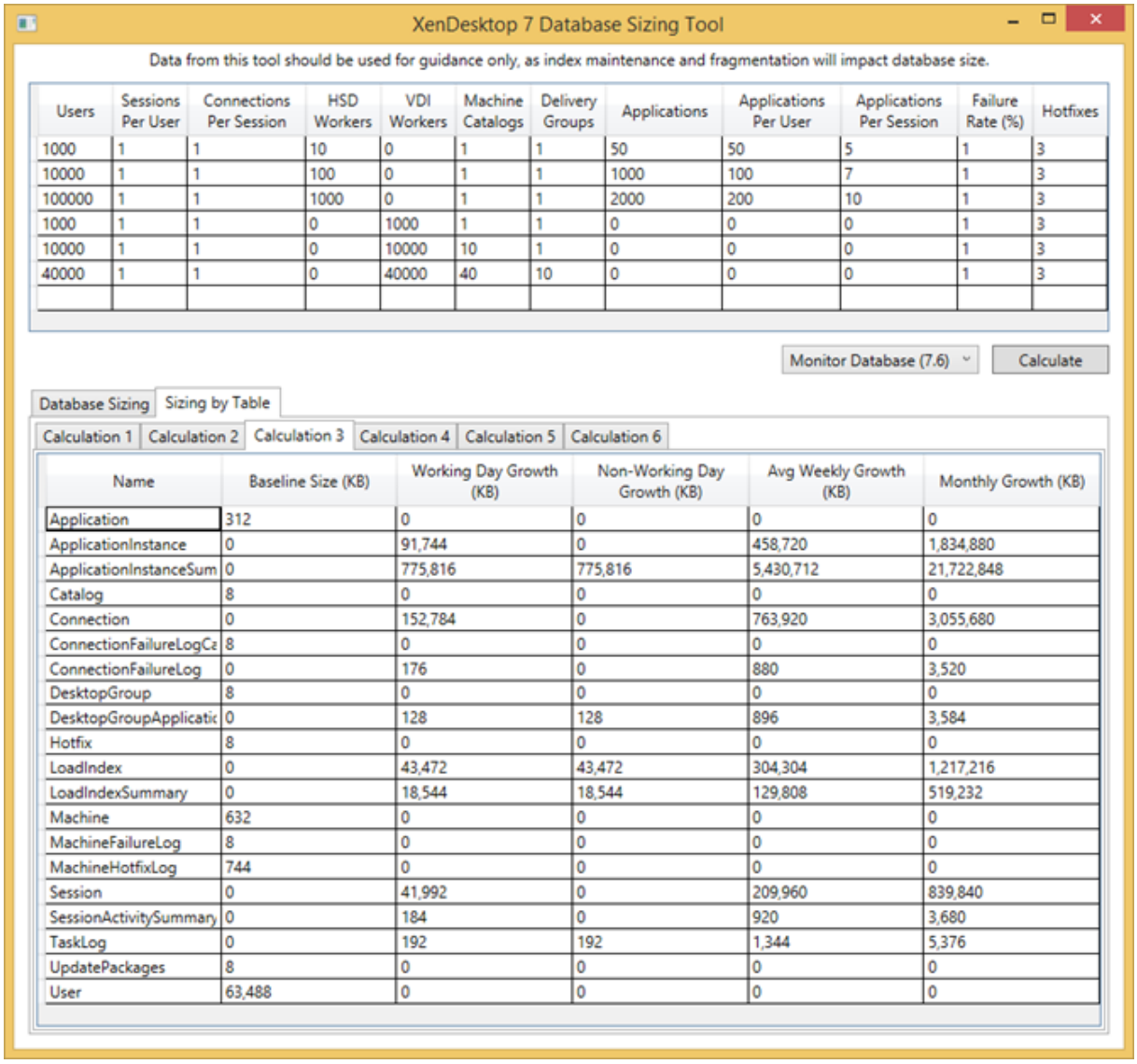
生成的数据包括在顶部输入的每个行的一行。这些列将指示不同时间点的近似大小。对于站点数据库，大小往往达到最大大小并保持在那里，因为它不会累积数据。对于监视，数据库将随着时间的推移变得越来越大，具体取决于配置的监视梳理设置。请注意，这也取决于许可（例如，只有白金客户可以将整理间隔配置为超过七天）。

因此，对于监视，数据如下所示：



表格详细信息

有关哪些表实际占用空间的详细信息，请单击“按表调整大小”选项卡，其中有一个用于每个计算的选项卡：



Users	Sessions Per User	Connections Per Session	HSD Workers	VDI Workers	Machine Catalogs	Delivery Groups	Applications	Applications Per User	Applications Per Session	Failure Rate (%)	Hotfixes
1000	1	1	10	0	1	1	50	50	5	1	3
10000	1	1	100	0	1	1	1000	100	7	1	3
100000	1	1	1000	0	1	1	2000	200	10	1	3
1000	1	1	0	1000	1	1	0	0	0	1	3
10000	1	1	0	10000	10	1	0	0	0	1	3
40000	1	1	0	40000	40	10	0	0	0	1	3

Name	Baseline Size (KB)	Working Day Growth (KB)	Non-Working Day Growth (KB)	Avg Weekly Growth (KB)	Monthly Growth (KB)
Application	312	0	0	0	0
ApplicationInstance	0	91,744	0	458,720	1,834,880
ApplicationInstanceSum	0	775,816	775,816	5,430,712	21,722,848
Catalog	8	0	0	0	0
Connection	0	152,784	0	763,920	3,055,680
ConnectionFailureLogCa	8	0	0	0	0
ConnectionFailureLog	0	176	0	880	3,520
DesktopGroup	8	0	0	0	0
DesktopGroupApplicatic	0	128	128	896	3,584
Hotfix	8	0	0	0	0
LoadIndex	0	43,472	43,472	304,304	1,217,216
LoadIndexSummary	0	18,544	18,544	129,808	519,232
Machine	632	0	0	0	0
MachineFailureLog	8	0	0	0	0
MachineHotfixLog	744	0	0	0	0
Session	0	41,992	0	209,960	839,840
SessionActivitySummary	0	184	0	920	3,680
TaskLog	0	192	192	1,344	5,376
UpdatePackages	8	0	0	0	0
User	63,488	0	0	0	0

此更详细的视图显示了哪些表可能变得较大，因此您可以调整监视整理以保持较小的某些区域。细分涵盖基线大小（例如固定大小）。通常，基于用户或计算机的表每天都有增长（历史负载平衡信息），并且仅在工作日（连接和会话）增长。然后将这些数据并入每周增长列（假定一周 7 天中的五个工作日），然后纳入每月列。

将数据导出到 **Excel** 中

要将任何表格导出到 Excel 中，只需在表格中单击，选择并复制所有内容，然后将其粘贴到 Excel 中。

这篇文章是从 *Chris Gilbert* 撰写的博客文章修改而来的。你可以找到原始帖子，阅读评论，并在这里发布反馈：*https://www.citrix.com/blogs/2014/11/20/database-sizing-tool-for-xendesktop-7/*。

实施和配置

March 2, 2021

网络连接

Citrix ADC 和 OpenShift 4 解决方案简介

在 SC2S 中创建 VPX Amazon Machine Image (AMI)

Citrix Gateway 和 Microsoft Azure 多重身份验证

面向 Azure DNS 专用区域的 Citrix ADC 部署指南

工作区

利用本地主机缓存进行无中断数据库升级

借助 RDP 通过 AWS 中的 Linux 堡垒主机连接到 Citrix 体系结构

SQL Server 和 Citrix 数据库

Citrix 联合身份验证服务可扩展性（PDF 下载）

针对 XenApp 和 XenDesktop 的组策略管理模板更新

XenApp 和 XenDesktop 7.11 至当前：延迟和 SQL 阻止查询改进

XenDesktop 7.6 的数据库大小指南

使用 SQL Server 高可用性解决方案时更新数据库连接字符串

本地主机缓存大小调整和扩展

XenApp 和 XenDesktop 7.9 中的 Citrix 通用打印服务器负载平衡

Citrix Endpoint Management

部署

有关最新完整的 XenMobile Server 文档，请参阅 XenMobile Server。

Citrix ADC 和 OpenShift 4 解决方案简介

May 20, 2020

开放式移位解决和开放式移位问题简介

Red Hat OpenShift 4 是 Kubernetes 平台，可为本地云、混合云和多云部署提供企业级基础。

OpenShift Container Platform 为 Kubernetes 提供企业就绪的增强功能，其中包括：

混合云部署

您可以将 OpenShift 容器平台群集部署到各种公有云平台或数据中心中。

集成 **Red Hat** 技术

OpenShift Container Platform 中的主要组件来自 Red Hat Enterprise Linux 和相关的 Red Hat 技术。OpenShift Container Platform 受益于针对 Red Hat 企业级质量软件的密集测试和认证计划。

开源开发模型

开发完成，源代码可从公共软件存储库获得。这种开放式合作促进了快速创新和发展。

有关更详细的参考，请参阅 OpenShift Container Platform 体系结构。

外部负载均衡器的 **OpenShift4** 要求

外部负载均衡器使 Kubernetes 节点能够与群集之外的子网进行通信。这对于 OpenShift 部署的可操作性至关重要，因为容器和群集需要了解传入流量才能正确扩展和缩小不同容器，并且显然需要将正确的传入流量定向到相应容器。OpenShift 需要 Citrix 提供的外部负载均衡器才能有效运行。

这意味着我们可以使用我们的技术，特别是我们的容器化应用程序交付控制器 (CPX) 和我们的 Citrix Ingress Controller (CIC) 来支持全面运行和优化的 OpenShift 部署，以及包括 VPX 在内的自动化外部 Citrix ADC 外形规格。MPX 和 BLX。

Citrix ADC 和 **OpenShift** 集成的优势

生产级入口

Citrix ADC 经验证可以大规模工作，为互联网最大的 Web 属性和数千家企业提供高级负载平衡、TLS 终止、L3-L7 协议优化和冗余解决方案等功能。

灵活性

Citrix ADC 支持体系结构灵活性 — Citrix 为群集内外的每个环境提供了一系列完整的 ADC 外形规格。

可见性和故障排除

带服务图表的 Citrix ADM 提供了有关应用程序运行状况和性能的可操作见解，并针对任何问题提供主动故障排除。

有关更详细的参考，请参阅 使用 Citrix 和 Red Hat OpenShift 实现基于微服务的应用程序和交付。

开放转变的实施

如果您当前是 OpenShift“4.x” 客户，则您知道您的部署有网络拓扑要求。在以下部分中，您可以找到开始 Citrix 和 OpenShift 部署所需的配置先决条件。

OpenShift4 要求为每个服务同时提供负载平衡服务和相应的 DNS 映射，如以下部分所述。

外部负载平衡服务先决条件

在安装 OpenShift Container Platform 之前，必须预配两个 4 层负载平衡服务。第一个是 API 服务器所需的，第二个是为应用程序提供入口所必需的。

此外，某些端口需要可以访问，以满足网络拓扑要求。

1. 首先，您必须为引导和控制平面打开端口 6443 (Kubernetes API Server) 和 22623 (Machine Configuration Server)。确保在控制平面初始化后从负载均衡器中删除引导计算机。您还必须在路由器容器、计算机和工作程序上打开端口 443（HTTPS 流量）和 80（HTTP 流量）。

   有关更详细的参考，请访问 OpenShift4 Container Platform 文档 上的网络拓扑要求。

2. 外部 DNS 映射先决条件：

群集节点：

```
1     master1.openshift4.example.com +short    10.217.101.X
2     master2.openshift4.example.com +short    10.217.101.X
3     master3.openshift4.example.com +short    10.217.101.X
4     worker1.openshift4.example.com +short    10.217.101.X
5     worker2.openshift4.example.com +short    10.217.101.X
6     bootstrap.openshift4.example.com +short  10.217.101.X
7    <!--NeedCopy-->
```

ETCD 节点：

```
1     etcd-0.openshift4.example.com +short     10.217.101.X
2     etcd-1.openshift4.example.com +short     10.217.101.X
3     etcd-2.openshift4.example.com +short     10.217.101.X
4    <!--NeedCopy-->
```


API 端点：

```
    api.openshift4.example.com +short         10.217.101.X
    api-int.openshift4.example.com +short     10.217.101.X
<!--NeedCopy-->
```

通配符 DNS 条目：

```
    *.apps.openshift4.example.com +short      10.217.101.X
<!--NeedCopy-->
```

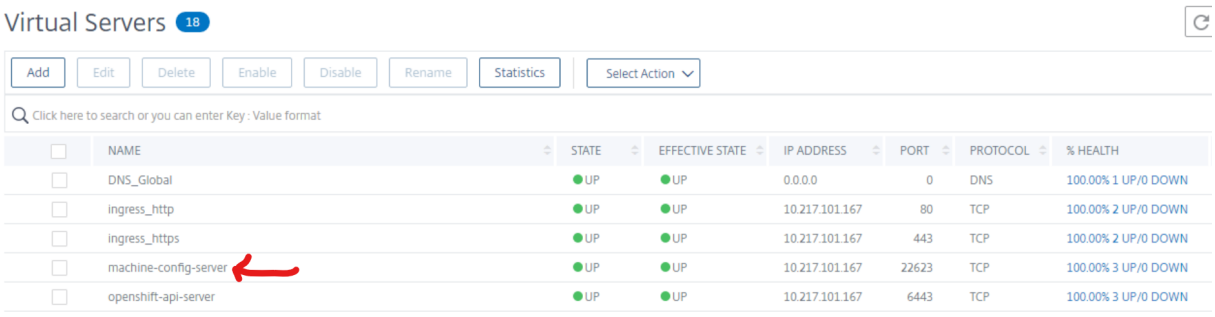
SRV 记录：

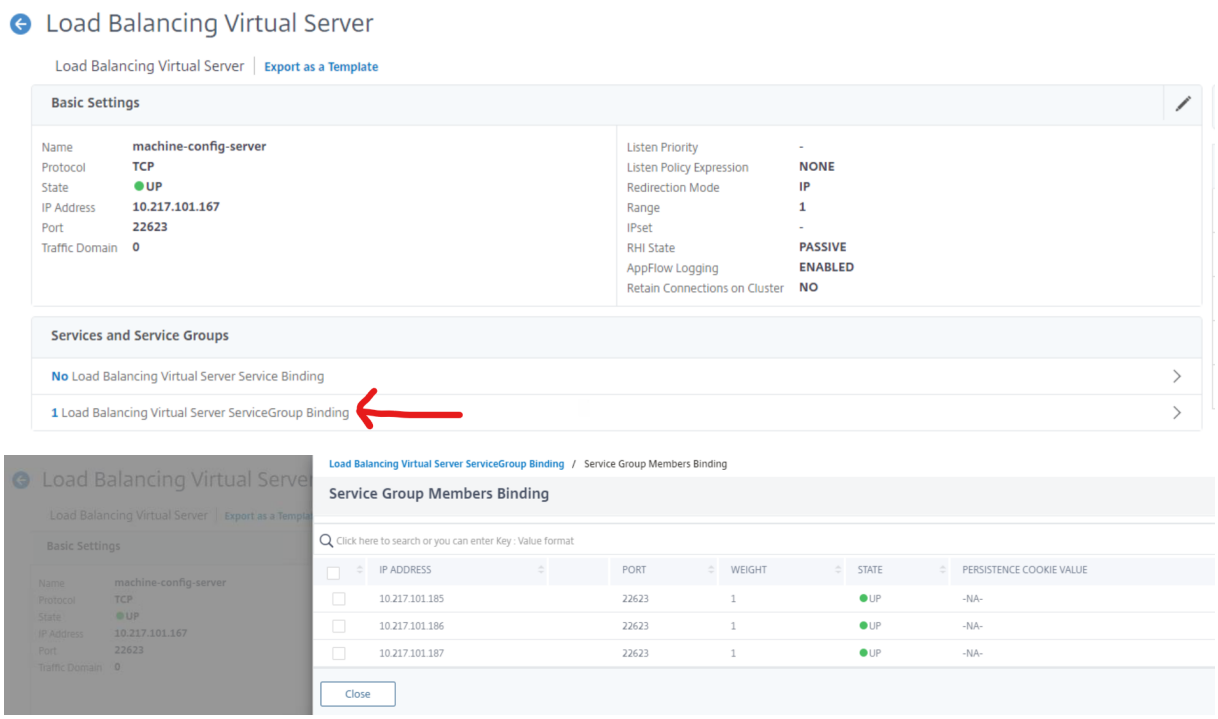
```
    $ dig _etcd-server-ssl._tcp.openshift4.example.com SRV +short

    0 10 2380 etcd-0.openshift4.example.com
    0 10 2380 etcd-1.openshift4.example.com
    0 10 2380 etcd-2.openshift4.example.com
<!--NeedCopy-->
```

Citrix ADC 的实施

Citrix ADC 配置概述

我们希望确保正确的虚拟 IP 对应于正确的服务组成员。如下所示，我们已将 `machine-config-server` 配置为指向一个服务组，该服务组拥有三个具有唯一 IP 地址的成员（10.217.101.185、10.217.101.186、10.217.101.187）。

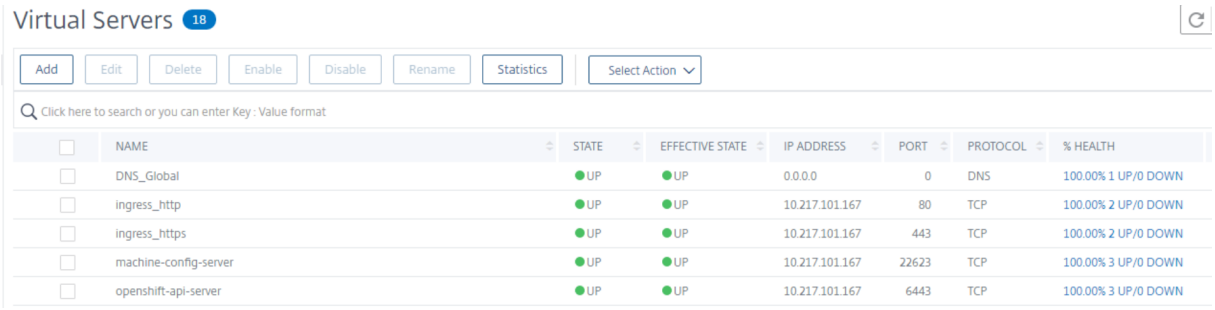




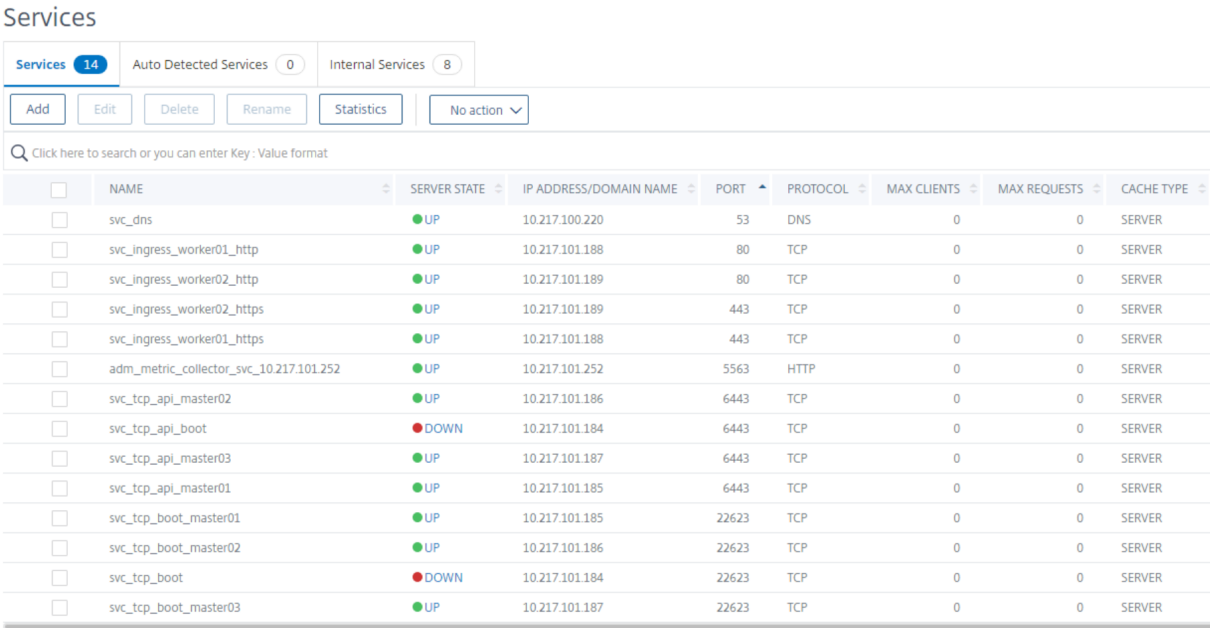
ADC 中的虚拟服务器和服务

使用以下图像作为参考，确保您的配置在正确的端口上运行了适当的虚拟服务器和服务。

虚拟服务器：



服务：



摘要

Citrix ADC 可以无缝集成到任何 OpenShift4 群集中，并根据 OpenShift4 安装要求为群集节点组件提供集成的外部负载平衡服务以实现高可用性和 DNS 支持。此外，可以使用 Citrix CPX 和 Citrix Ingress Controller 将 Citrix ADC 集成到 OpenShift4 集成到 OpenShift4 群集中，以便与所有容器化部署的 OpenShift4 集成。

要了解有关 Citrix Cloud Native 解决方案的更多信息，请访问 Citrix ADC 平台。

Citrix Gateway 和 Microsoft Azure 多重身份验证

March 4, 2021

Citrix Gateway 可通过任何设备和任何浏览器向用户展示所有托管、SaaS、Web、企业和移动应用程序。它使用 nFactor 身份验证根据内部部署 Microsoft AD 对用户进行身份验证，并利用 Microsoft AD FS 进行 Azure 多重身份验证 (MFA)。

概述

Citrix Gateway

Citrix Gateway 为用户提供一个访问点和单点登录 (SSO)，访问部署在数据中心和云中的业务应用程序和数据。它以 SaaS 的形式交付到各种设备（笔记本电脑、台式机、瘦客户端、平板电脑和智能手机）。Citrix Gateway 提供整合，有助于减少远程访问基础结构的占用空间，降低成本，并提供易于管理和更好的最终用户体验。Citrix Gateway 帮助 IT 过渡到混合云和 SaaS 环境。

- 联合身份验证和单点登录

Citrix Gateway 提供联合身份并支持 SAML 2.0、OAuth 和 OpenID，以便在所有应用程序中实现单点登录，无论这些应用程序是 Web、VDI、企业应用程序还是 SaaS 应用程序。

- 本地用户目录

Citrix Gateway 为 SaaS 应用程序（例如 Office 365 和 Salesforce）提供 SSO，并且将用户目录保留在本地。它可以作为 IdP 或代理来实现 Microsoft Active Directory 联合身份验证服务 (AD FS)。

- 多重 **(nFactor)** 身份验证

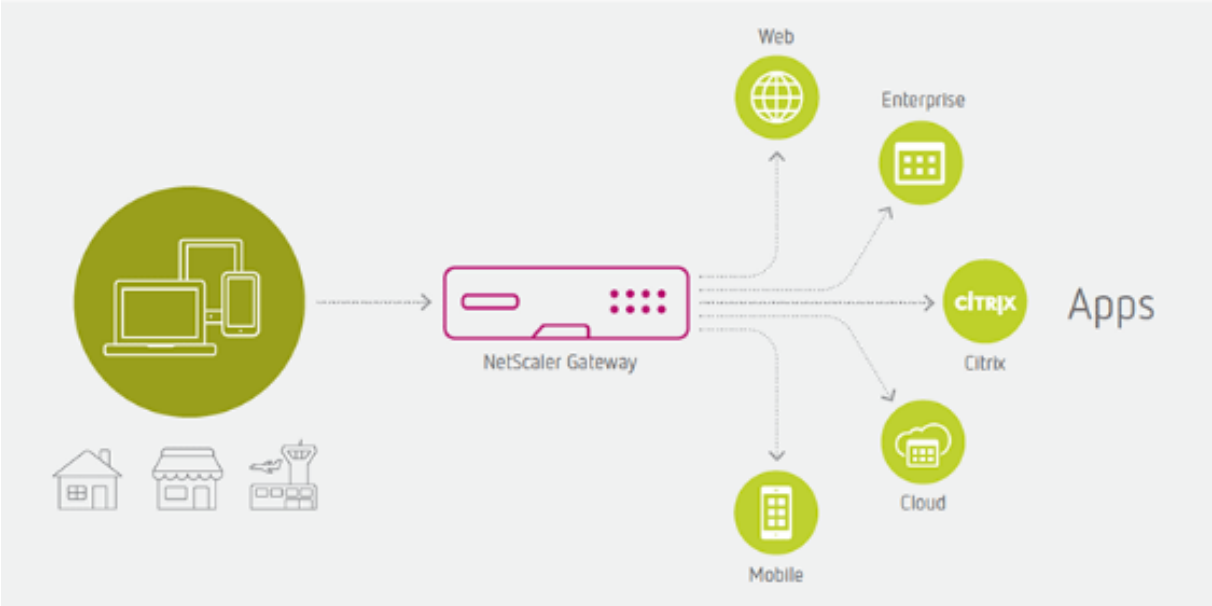
Citrix Gateway 提供了 nFactor 身份验证机制，并允许对访问网络的用户、访问内容以及访问网络的方式和时间进行精细控制。它支持所有身份验证机制，如 RADIUS、TACACS、NTLM、Diameter、SAML 2.0、OAuth 2.0 和 OpenID
2.0。

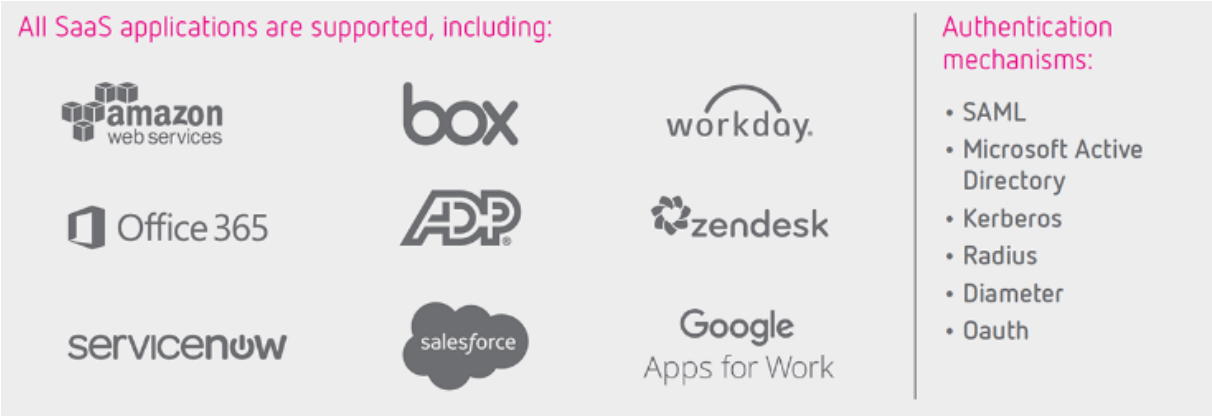
- 上下文访问控制策略

Citrix Gateway 允许根据最终用户设备、用户、用户位置和其他数据的状态对业务应用程序进行精细的访问控制。IT 管理员可以创建、管理和强制执行策略，以便在应用程序环境中安全地访问数据。这些策略可以用于 VDI、Web、移动、企业和 SaaS 应用程序。

- 可视性和监视

Citrix Application Delivery Management 包括 Gateway Insight，它为通过 Citrix Gateway 访问的所有应用程序提供端到端用户体验的可见性。它为应用程序支持团队提供信息，以解决与身份验证失败有关的问题，包括 EPA 检查失败和单点登录失败。





Microsoft Azure MFA

在日益复杂的情况下，人们正在连接到组织资源。人们通常在多个平台上使用智能手机、平板电脑、PC 和笔记本电脑，从公司网络上和公共网络上的组织拥有、个人和公共设备进行连接。在这个始终连接、多设备和多平台的世界中，用户帐户的安全性比以往任何时候都更加重要。跨设备、网络和平台使用的密码，无论其复杂性如何，都不足以确保用户帐户的安全性，尤其是当用户倾向于跨账户重复使用密码时。复杂的网络钓鱼和其他社交工程攻击可能导致用户名和密码在黑暗网络上发布和销售。

两步核查进程的安全在于其分层办法。妥协多个身份验证因素对攻击者来说是一项重大挑战即使攻击者设法了解用户的密码，在没有其他身份验证方法的情况下，它也是无用的。其工作原理是要求使用以下两种或更多种身份验证方法：

- 您知道的内容（通常为密码）
- 您拥有的对象（不容易复制的可信设备，例如手机）
- 您是什么（生物识别）

Azure 多重身份验证有助于保护数据和应用程序的访问。它使用第二种身份验证形式提供了额外的安全层。组织可以使用条件访问来使解决方案满足其特定需求。

Microsoft Azure MFA 部署方法

利用 Azure MFA 作为身份验证的第二个因素有不同的方法。下面将简要介绍这些方法及其优缺点。

Azure MFA 服务器

Microsoft Azure 多重身份验证服务器是原始方法，将被弃用。不应该考虑进行任何新的实施，因为

- Microsoft 对此方法没有进一步的投资。
- 没有与基于云的 SSPR 和 Azure MFA 集成。
- 没有从 MFA 服务器到基于 MFA 云的解决方案的无缝迁移工具。

Azure MFA 网络策略服务器扩展

Azure MFA 的网络策略服务器 (NPS) 扩展是支持的解决方案，它使用 NPS 适配器连接到基于云的 Azure MFA。它可以用作本地 RADIUS 服务器。

- NPS 适配器 (RADIUS) 将提供 MFA 规则或开/关内部/外部的网络位置。
- 它与与 SAML 集成方法类似的 Azure AD 条件访问策略不兼容。条件访问策略具有更丰富、更好的用户体验。
- 用户必须在 MFA 中注册，然后才能使用 NPS 适配器。与 Azure MFA 基于云的访问和条件访问不同，如果用户未注册，则 NPS 扩展将无法对用户进行身份验证，从而生成对帮助台的更多调用。
- 当 NPS 适配器调用 MFA 时，它会触及用户注册的默认选项。没有向用户提供需要 MFA 并且 MFA 即将推出的任何直观通知。在封闭过程中，用户没有用户界面可以更改 MFA 方法。如果用户没有默认设备，则会失败。用户必须返回自助服务门户并重置默认选项，然后尝试重新连接。

Microsoft AD FS 和 **Azure MFA**

如果贵组织与 Azure AD 联合，但密码哈希未与 Azure AD 同步，则可以将本地 AD 用于轻型目录访问协议 (LDAP)，并将 Azure MFA 作为 AD FS 中继方访问策略的一部分启用 Azure MFA。从 Windows Server 2016 开始，您现在可以配置 Azure MFA 进行主身份验证。

- Azure MFA 适配器内置到 Windows Server 2016，并且不需要额外的安装。
- Azure MFA 适配器直接与 Azure AD 集成，不需要本地 Azure MFA 服务器。
- 如果用户没有注册 MFA，则会在下次登录时引导他们完成该过程。它可以确保减少对服务台的呼叫，并为用户提供更好的流程。
- 用户会收到一个需要 MFA 并且 MFA 即将推出的直观通知。用户可以在 UI 中的封闭过程中更改网关选项。

Azure AD 和 **Azure MFA**

如果贵组织正在将密码哈希同步到 Azure AD 中，则可以通过条件访问策略利用 Azure MFA 来挑战用户进行第二因素身份验证。

- 此方法不需要在本地安装任何额外的安装。
- 如果用户没有注册 MFA，则在下次登录时将引导他们完成该过程。它可以确保减少对服务台的呼叫，并为用户提供更好的流程。
- 用户会收到一个需要 MFA 并且 MFA 即将推出的直观通知。用户可以在 UI 中的封闭过程中更改网关选项。

Azure AD 直通身份验证和 **Azure MFA**

Azure AD 直通身份验证 (PTA) 允许用户使用相同的密码登录本地和基于云的应用程序。当用户使用 Azure AD 登录时，此功能将直接针对本地 Active Directory 验证用户的密码。Azure AD PTA 是 Azure AD 密码哈希同步的替代方案，它为组织提供云身份验证的同样好处。

- Azure AD PTA 要求在本地安装轻型代理。
- Azure AD PTA 通过与 Azure AD 条件访问策略（包括 Azure MFA）无缝协作来保护用户帐户。

- 用户可以在云中完成自助服务密码管理任务。
- 本地密码永远不会以任何形式存储在云中。
- 代理程序只能从您的网络内进行出站连接。因此，不需要在外围网络（也称为 DMZ）中安装代理。

目前的情况

具有以下特征的环境需要将 Azure MFA 作为身份验证的第二个因素：

- 配置了具有 Azure AD 同步的本地 AD。
- Azure AD 密码哈希同步已禁用。
- 需要访问 O365 应用程序。
- 需要访问本地 Citrix Virtual Apps and Desktops。
- 需要使用现代身份验证方法（SAML、OAuth）访问应用程序。
- 需要使用旧版身份验证方法访问应用程序。

设计要点

下面是所提出的解决方案的设计要点：

- 在单个门户中安全地访问托管、SaaS、企业和 Web 应用程序是必需的。
- 在身份验证过程中，用户只需要输入一次凭据。
- 必须为所有托管、SaaS、企业和 Web 应用程序提供单点登录。

提议的解决方案

概述

提议的解决方案基于以下组成部分：

- 本地 Citrix Gateway
- 本地 Microsoft AD
- 本地 Microsoft AD FS
- 作为 AD FS 代理的本地 Citrix ADC
- Microsoft Azure MFA

Citrix Gateway 正在利用身份验证、授权和审核功能（Citrix ADC AAA）和 nFactor 身份验证机制使用 LDAP 策略对用户进行身份验证，并利用 AD FS Relay 方上的访问策略触发 Azure MFA 验证过程。Azure MFA 验证用户后，AD FS 将生成 SAML 断言（SAML 响应）并将用户重定向回 Citrix Gateway。此时，用户将进行身份验证，Citrix Gateway 将显示用户有权使用的所有应用程序。

该解决方案需要两个公共 DNS 记录和两个公共 IP 地址：

说明	值
Citrix Gateway FQDN	access.ctxdemos.com
Citrix 身份验证、授权和审核 FQDN	aaa.ctxdemos.com

该解决方案使用一个公共 SSL 证书：

说明	值
公用名	access.ctxdemos.com
使用者备用名称	sts.ctxdemos.com
使用者备用名称	aaa.ctxdemos.com

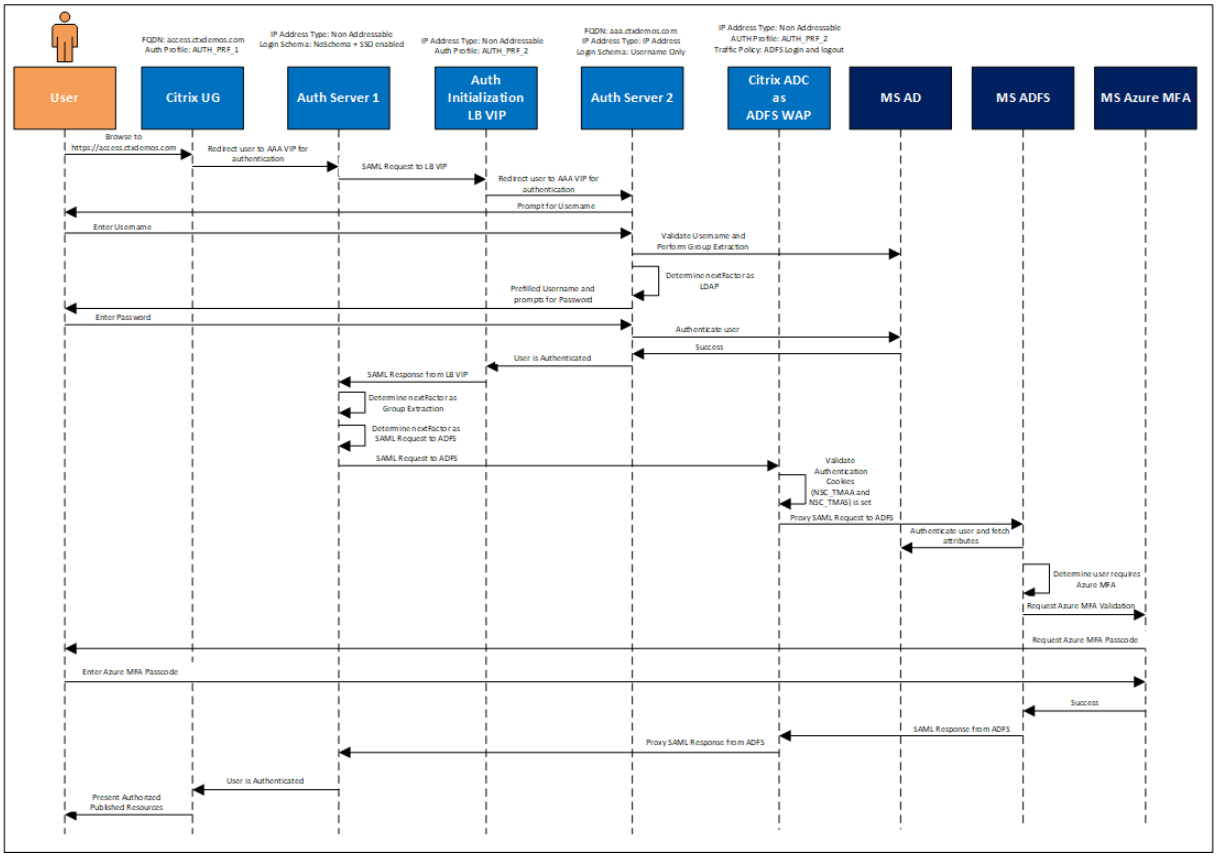
该解决方案还使用由内部 Microsoft 证书颁发机构服务颁发的通配符 SSL 证书：

说明	值
公用名	*.ctxdemos.com

身份验证流程

序列图

下面的顺序图显示了解决方案的身份验证流程：



身份验证步骤

身份验证步骤是：

1. 用户导航到 https://access.ctxdemos.com。
2. Citrix Gateway 将用户重定向到第一个 Citrix ADC AAA VIP（不可寻址）。
3. 首先 Citrix ADC AAA VIP 使用无体系结构登录，该登录配置为单点登录。然后它开始处理高级身份验证策略。
4. 第一个身份验证策略是 SAML SP 到不可寻址的 LB VIP，以生成身份验证 Cookie。
5. 帮助器 LB VIP 配置为使用第二个 Citrix ADC AAA VIP（可寻址）进行身份验证。因此，它会将用户重定向到第二个身份验证、授权和审核 VIP。
6. 第二个 Citrix ADC AAA VIP 使用`Username Only`登录体系结构，提示用户输入用户名。然后它开始处理高级身份验证策略。
7. 第一个身份验证策略是组提取，用于查询本地 AD 中的用户名，并验证用户是否属于 AzureMFACAUsers 安全组。验证结果成功后，它将开始处理下一个身份验证因素，即 LDAP 策略。
8. LDAP 策略使用`UsernameAndPassword`登录体系结构和预填充的用户名字段，并提示用户输入 AD 密码。
9. 当第二个 Citrix ADC AAA VIP 上的身份验证成功完成时，它将返回到帮助程序 LB VIP，该辅助程序将为第一个身份验证、授权和审核 VIP 生成 SAML 响应。

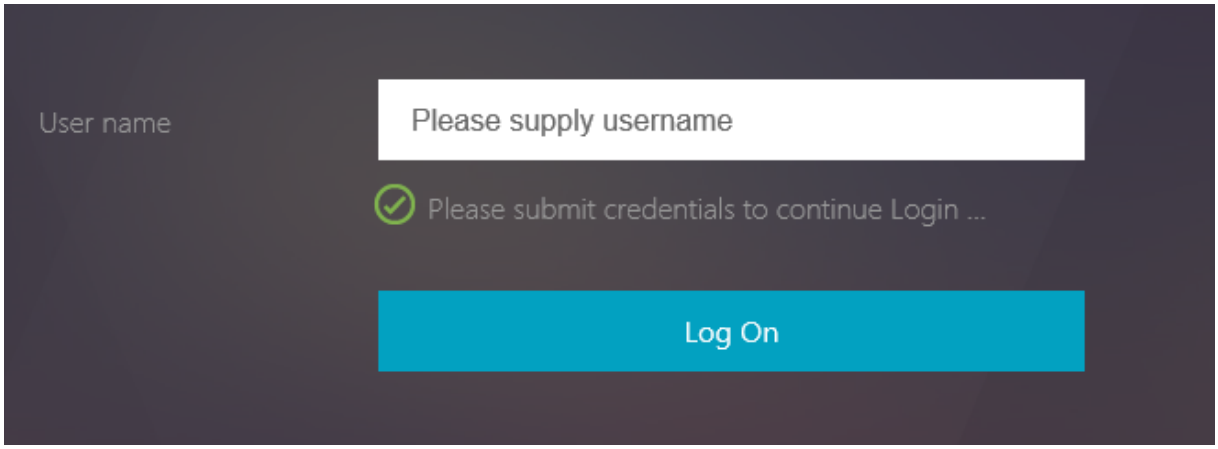
10. 第一个 Citrix ADC AAA VIP 开始处理下一个因素，即组提取，以确保从 AD 中提取用户的组并存储在身份验证、授权和审计变量中，以便稍后在此过程中使用。
11. 第一个 Citrix ADC AAA VIP 开始处理下一个因素，这是 Citrix ADC 上的 SAML SP 到 AD FS 代理 VIP。

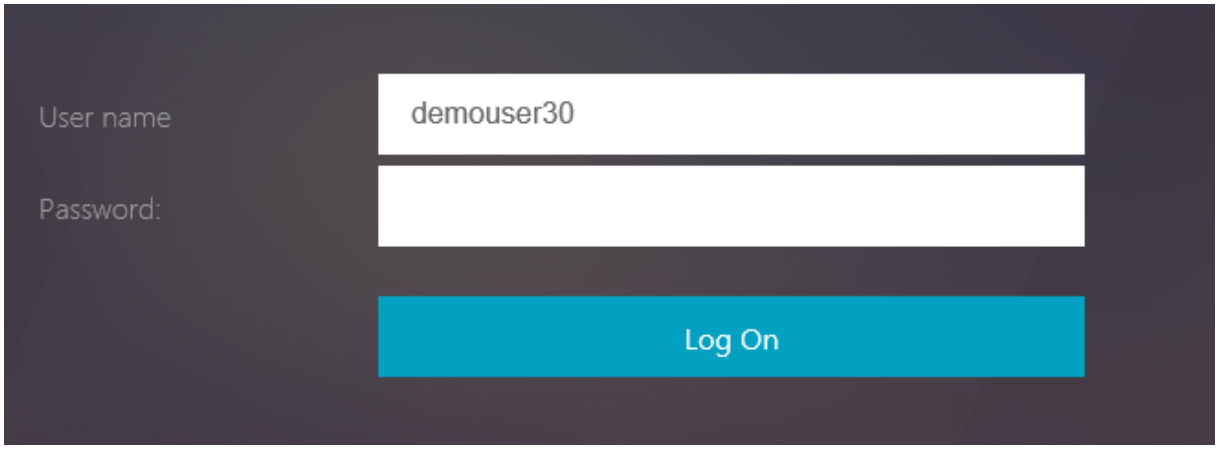
    > 注意：
    >
    > Citrix ADC 与 AD FS 场联合。详细步骤将在后面的章节中进行说明。

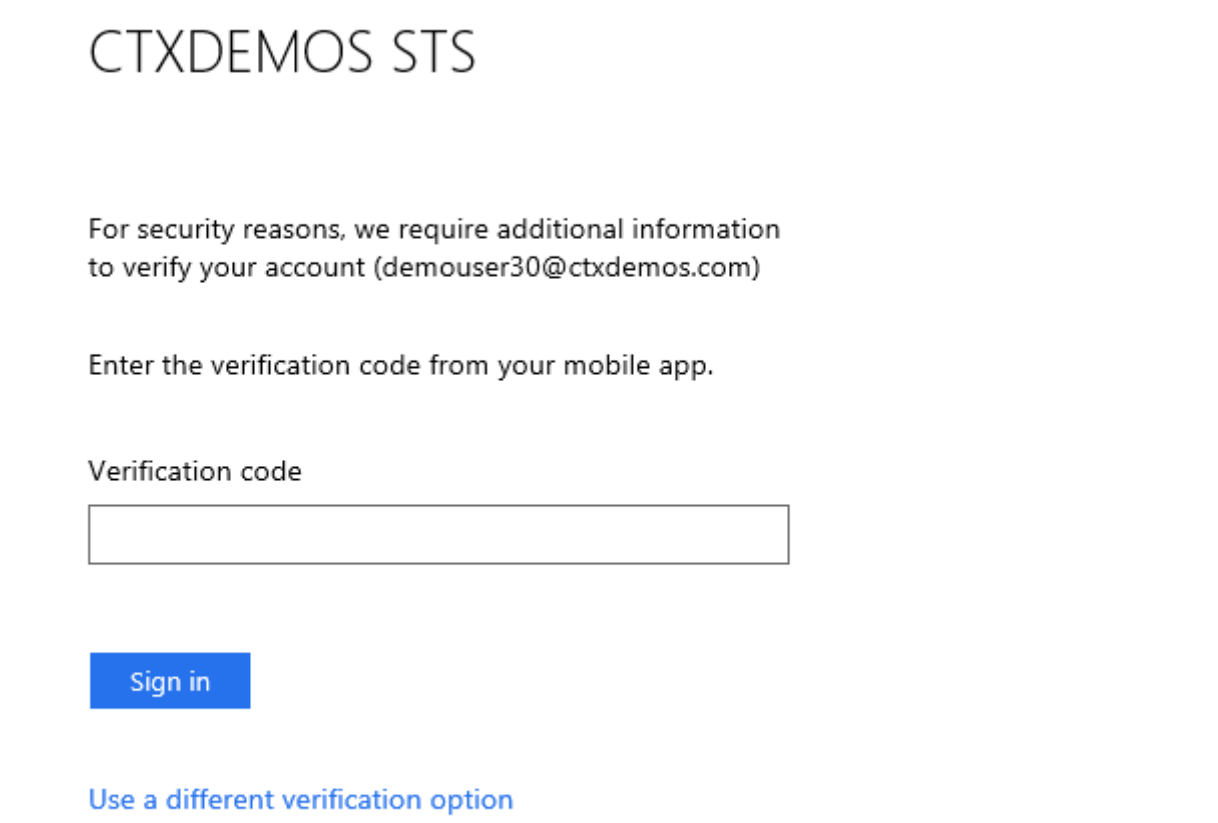
12. AD FS 代理 VIP 验证身份验证 Cookie（NSC_TMAA 和 NSC_TMAS）是否已设置。然后它将 SAML 请求发送到后端 AD FS 服务器（后端 AD FS 服务器应在内部 Citrix ADC 上进行负载平衡，以实现服务的高可用性和弹性）。
13. AD FS 服务器处理 SAML 请求。由于中继方上的访问策略设置为“允许所有用户并要求 MFA 进行身份验证”，因此它会触发 Azure MFA 身份验证过程。
14. Azure MFA 处理用户名。如果已注册，则会使用配置的方法向用户提出质询。否则，它会提示用户注册并设置主身份验证方法和辅助身份验证方法。
15. 一旦成功完成 Azure MFA 身份验证过程，AD FS 将为 Citrix Gateway（第一个 Citrix ADC AAA VIP）生成一个 SAML 响应。
16. 首先 Citrix ADC AAA VIP 会收到 SAML 响应，并确认用户的身份验证过程已完成。
17. Citrix Gateway 向 Citrix StoreFront 发送身份验证信息，该信息会枚举授权用户使用的所有应用程序和桌面。此外，它会处理用户的组成员资格以在 Citrix Gateway 上显示已发布的书签。

身份验证屏幕

上面提到的大多数步骤对用户来说都是无缝的，因为这些步骤是在 Citrix ADC 上的各个 VIP 之间内部进行的。用户体验如下所示：







实现

Microsoft AD FS

证书要求

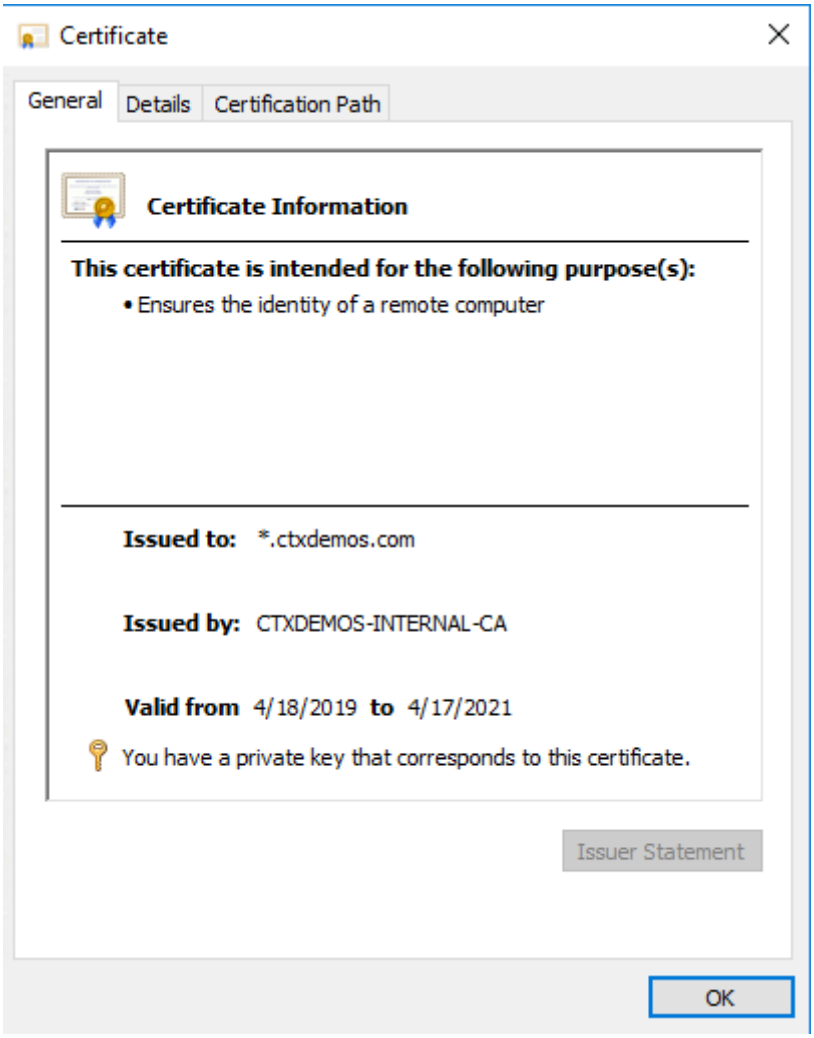
联合身份验证服务器需要下表中的证书：

证书类型	说明	在部署之前需要知道什么
安全套接字层 (SSL) 证书	这是标准的安全套接字层 (SSL) 证书，用于保护联合身份验证服务器与客户端之间的通信。	此证书必须绑定到联合身份验证服务器或联合身份验证服务器代理的 Internet 信息服务 (IIS) 中的默认网站。对于联合身份验证服务器代理，在成功运行联合身份验证服务器代理配置向导之前，必须在 IIS 中配置绑定。建议：由于此证书必须受 AD FS 客户端的信任，因此请使用由公共（第三方）证书颁发机构 (CA) 颁发的服务器身份验证证书。例如，威瑞信。提示：此证书的使用者名称用于表示您部署的每个 AD FS 实例的联合身份验证服务名称。出于这个原因，您可能需要考虑在 CA 颁发的任何新证书上选择一个主题名称，该名称最能代表合作伙伴的公司或组织的名称。
服务通信证书	此证书启用 WCF 消息安全性，以保护联合身份验证服务器之间的通信。	默认情况下，SSL 证书用作服务通信证书。这可以使用 AD FS 管理控制台进行更改。
令牌签名证书	这是标准 X509 证书，用于对联合身份验证服务器颁发的所有令牌安全地进行签名。	令牌签名证书必须包含私钥，并且该证书应链接到联合身份验证服务中的受信任根目录。默认情况下，AD FS 会创建自签名证书。但是，您可以稍后使用 AD FS Management 管理单元将其更改为 CA 颁发的证书，具体取决于组织的需求。
令牌解密证书	这是标准 SSL 证书，用于解密由合作伙伴联合身份验证服务器加密的任何传入令牌。它还发布在联合身份验证元数据中。	默认情况下，AD FS 会创建自签名证书。但是，您可以稍后使用 AD FS Management 管理单元将其更改为 CA 颁发的证书，具体取决于组织的需求。

演示环境配置

证书类型	演示环境配置
安全套接字层 (SSL) 证书	由 AD FS 服务器上的内部颁发 CA 颁发的内部证书。 Citrix ADC 上的公共可信证书。
服务通信证书	由 AHS 内部颁发证书颁发机构颁发的内部证书。
令牌签名证书	由 AD FS 服务自动生成。
令牌解密证书	由 AD FS 服务自动生成。

在演示环境中，将注册通配符证书并安装在服务器上。

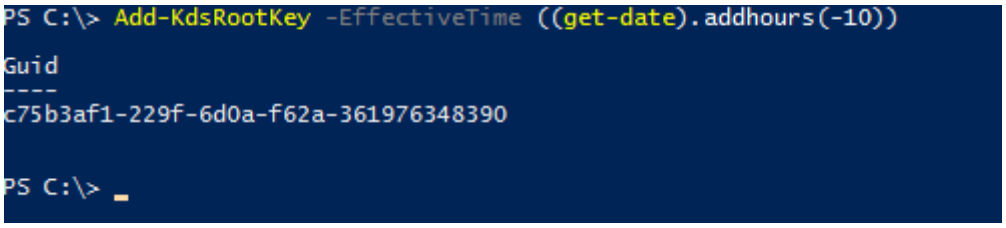


服务帐户要求

您可以创建服务帐户或利用组托管服务帐户 (gMSA)。要使用 gMSA，您需要创建密钥分发服务根密钥。因此，请启动 PowerShell 并运行以下命令：


```
Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10))
<!--NeedCopy-->
```

此命令创建存储在 Active Directory 中的密钥分发服务根密钥，并允许您创建一个组托管服务帐户 (gMSA) 作为以后创建的 AD FS 服务帐户。使用域管理员权限运行此命令。



DNS 记录要求

在内部和外部，您的 AD FS 联合身份验证服务名称都需要 DNS A 记录。在演示环境中，内部 DNS 记录指向 AD FS 服务器 IP，外部 DNS 记录指向 Citrix Gateway 公有 IP。

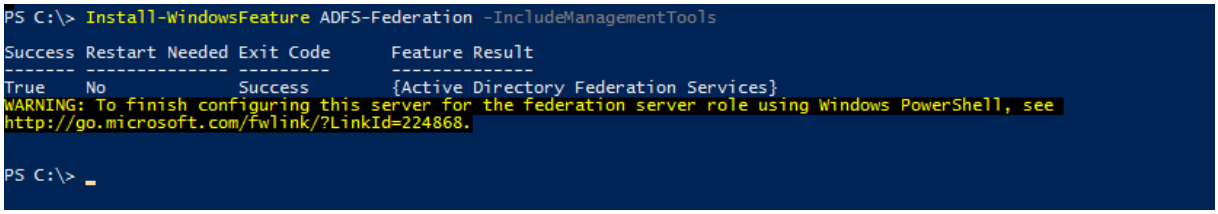
记录名	范围	类型	IP 地址
sts.ctxdemox.com	内部	A	22.22.22.6
sts.ctxdemox.com	外部	A	40.85.225.175

添加 AD FS 角色并配置 AD FS 场

添加 AD FS 角色

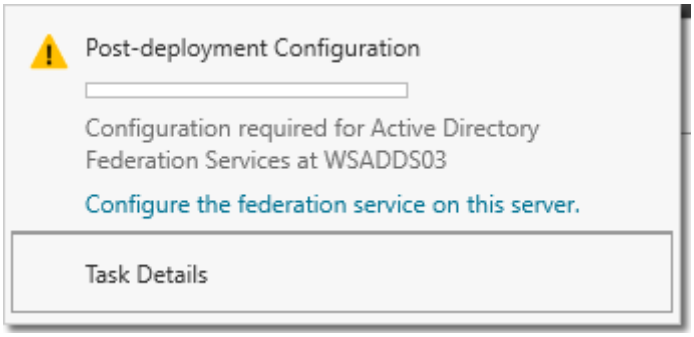
要将 AD FS 角色添加到 Windows Server 2016，请启动 PowerShell 并运行以下命令：

```
Install-WindowsFeature AD FS-Federation -IncludeManagementTools
<!--NeedCopy-->
```

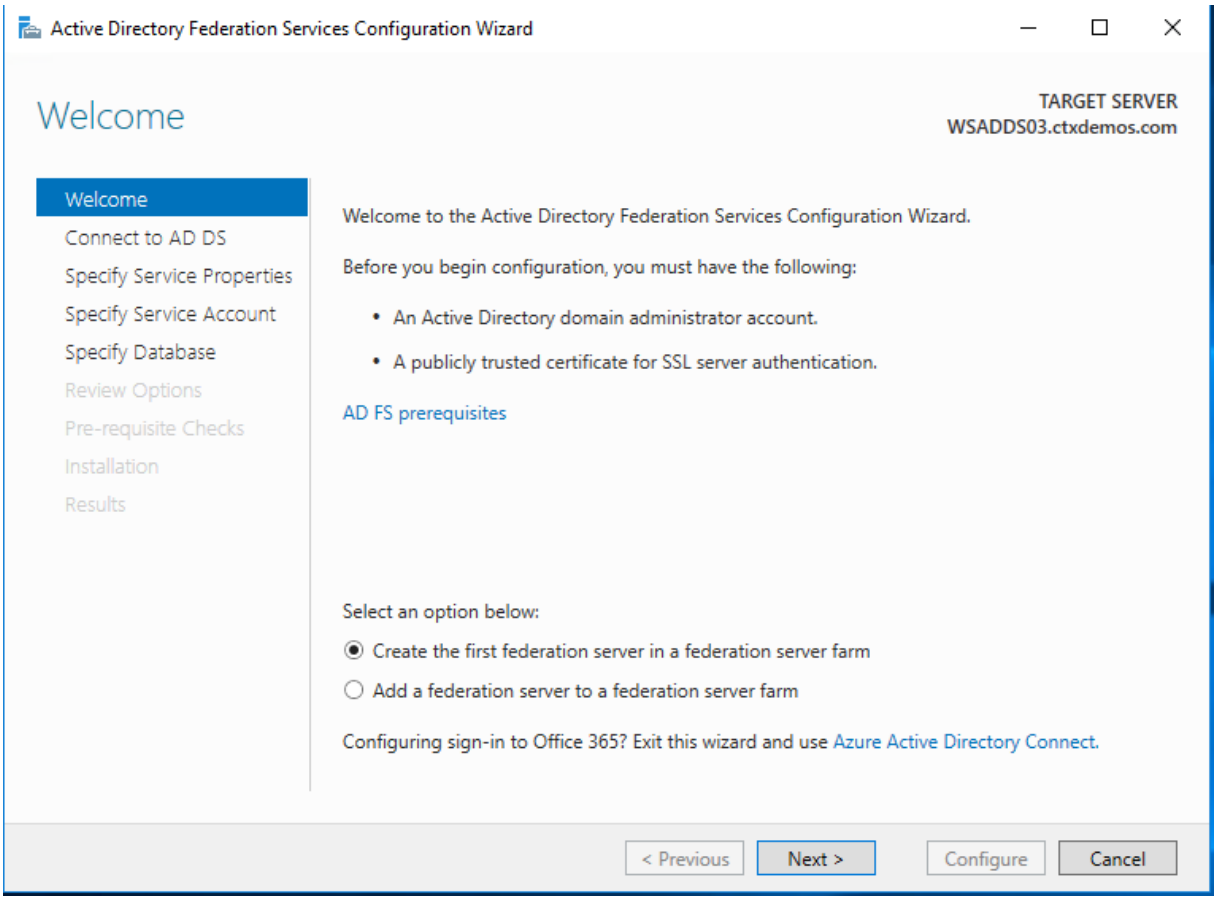


配置 **AD FS** 场

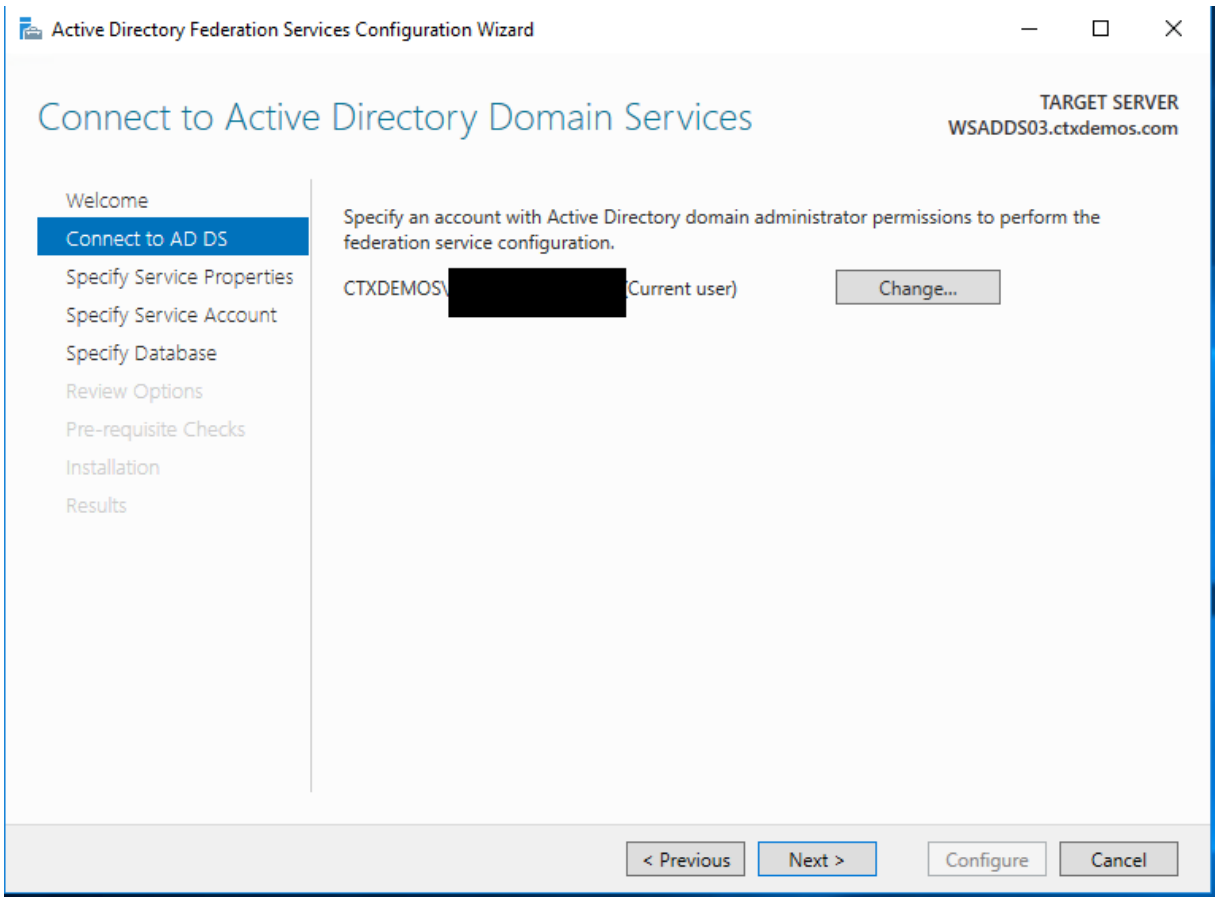
现在，您可以从服务器管理器开始 AD FS 部署后配置。单击“配置此服务器上的联合身份验证服务”。



在“欢迎”页上，选择“在 联合服务器场中创建第一个联合服务器”，然后单击“下一步”。

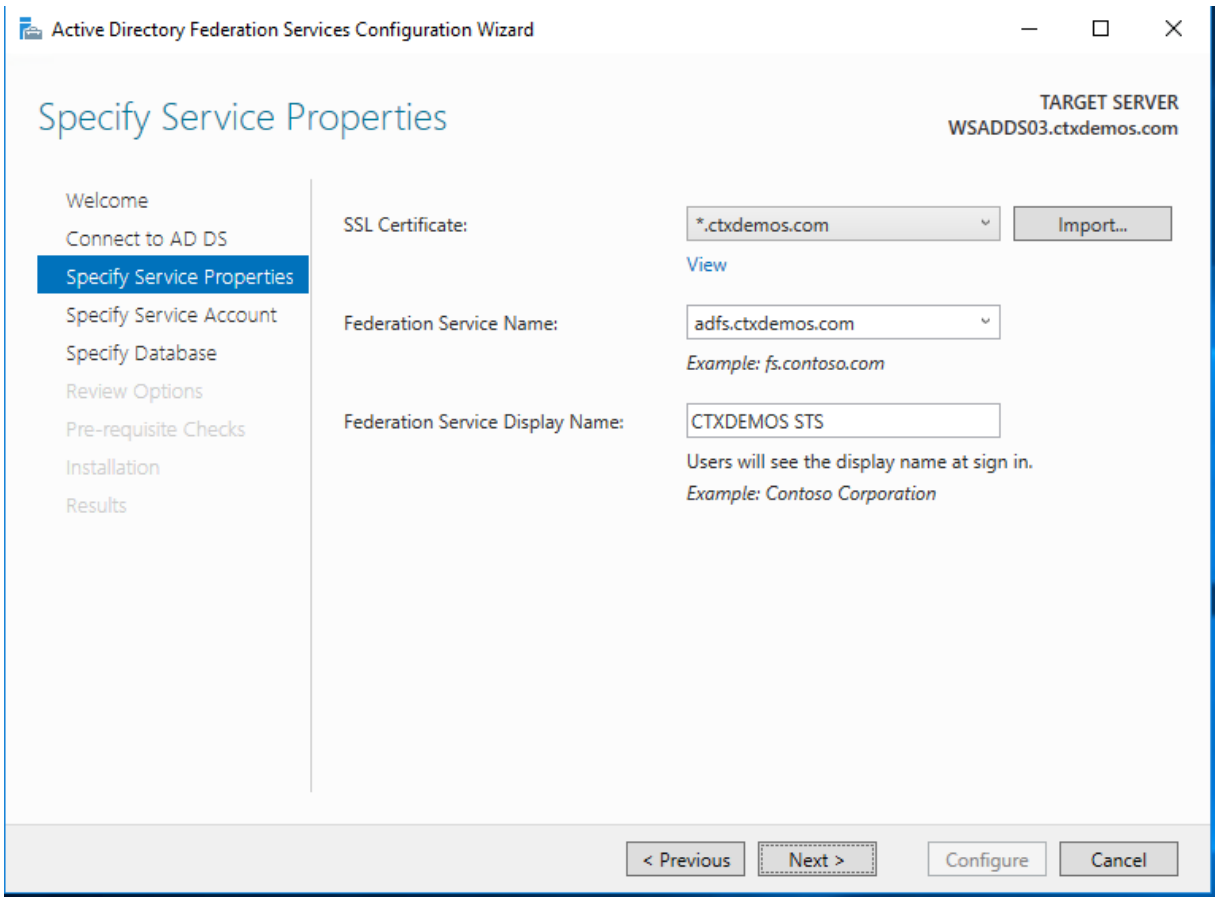


在连接到 **Active Directory** 域服务页面上，确保指定了域管理员帐户，然后单击下一步。

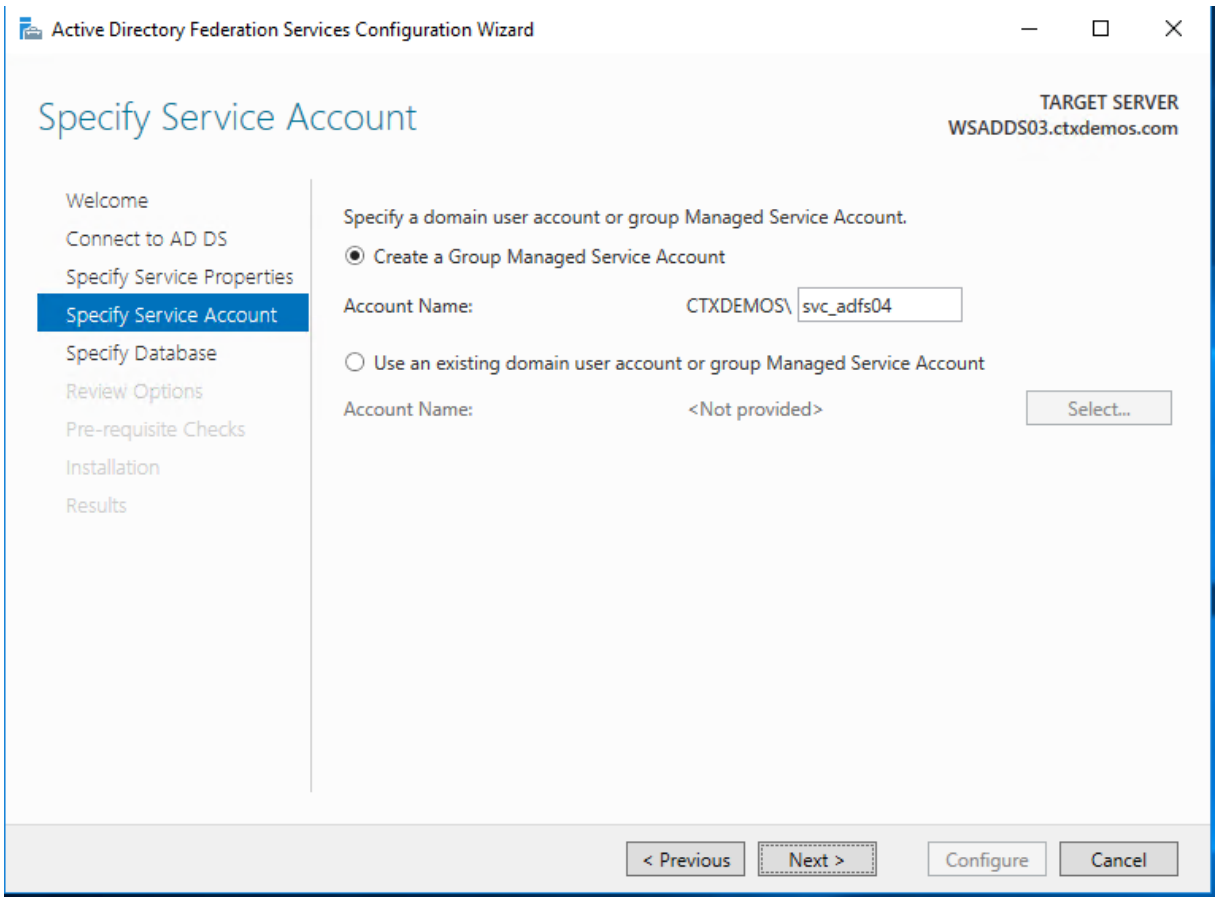


在“指定服务属性”页上，完成以下步骤，然后单击“下一步”：

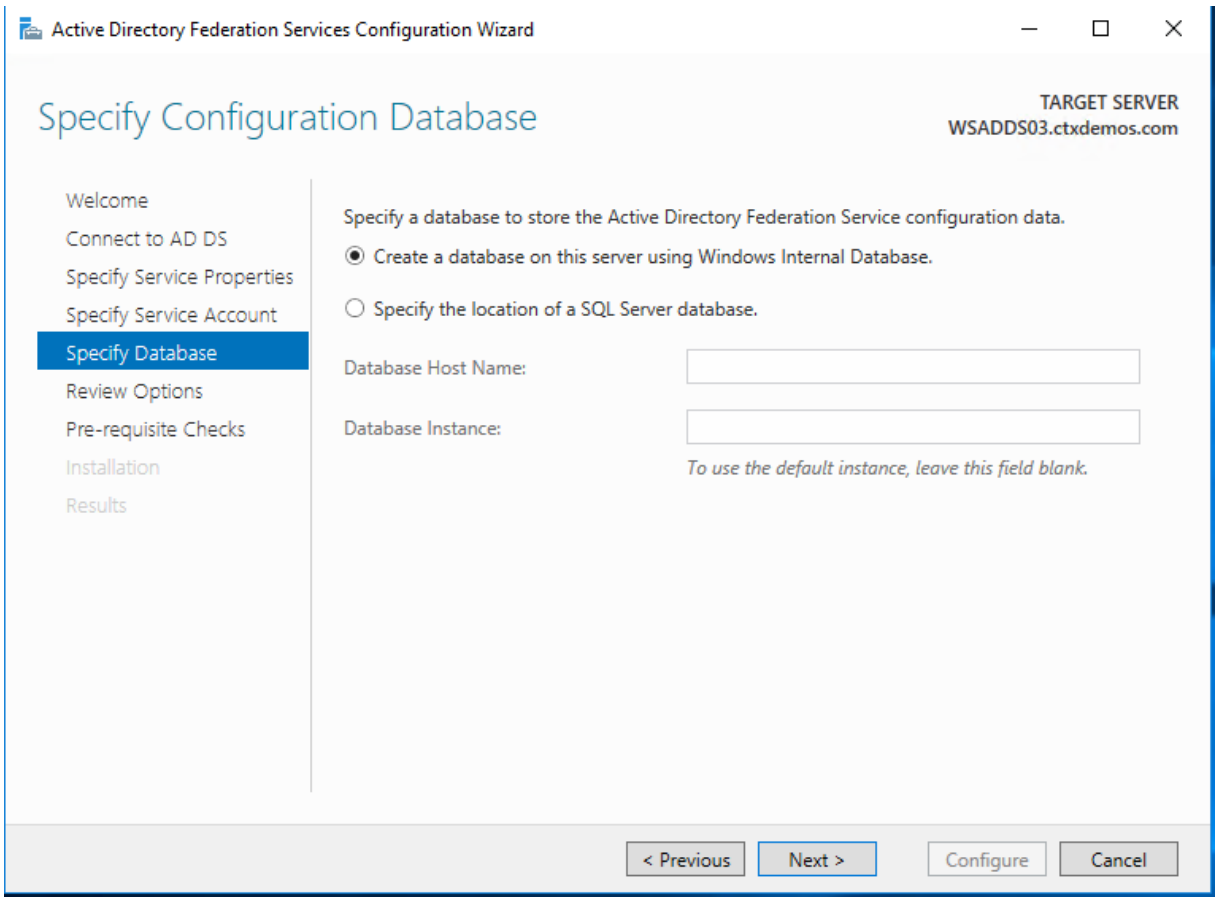
- 选择在前面的步骤中安装在服务器上的证书。
- 联合身份验证服务名称将根据证书的主题名称自动填充。
- 输入联合身份验证服务的显示名称。例如，**CTXDEMOS STS**。



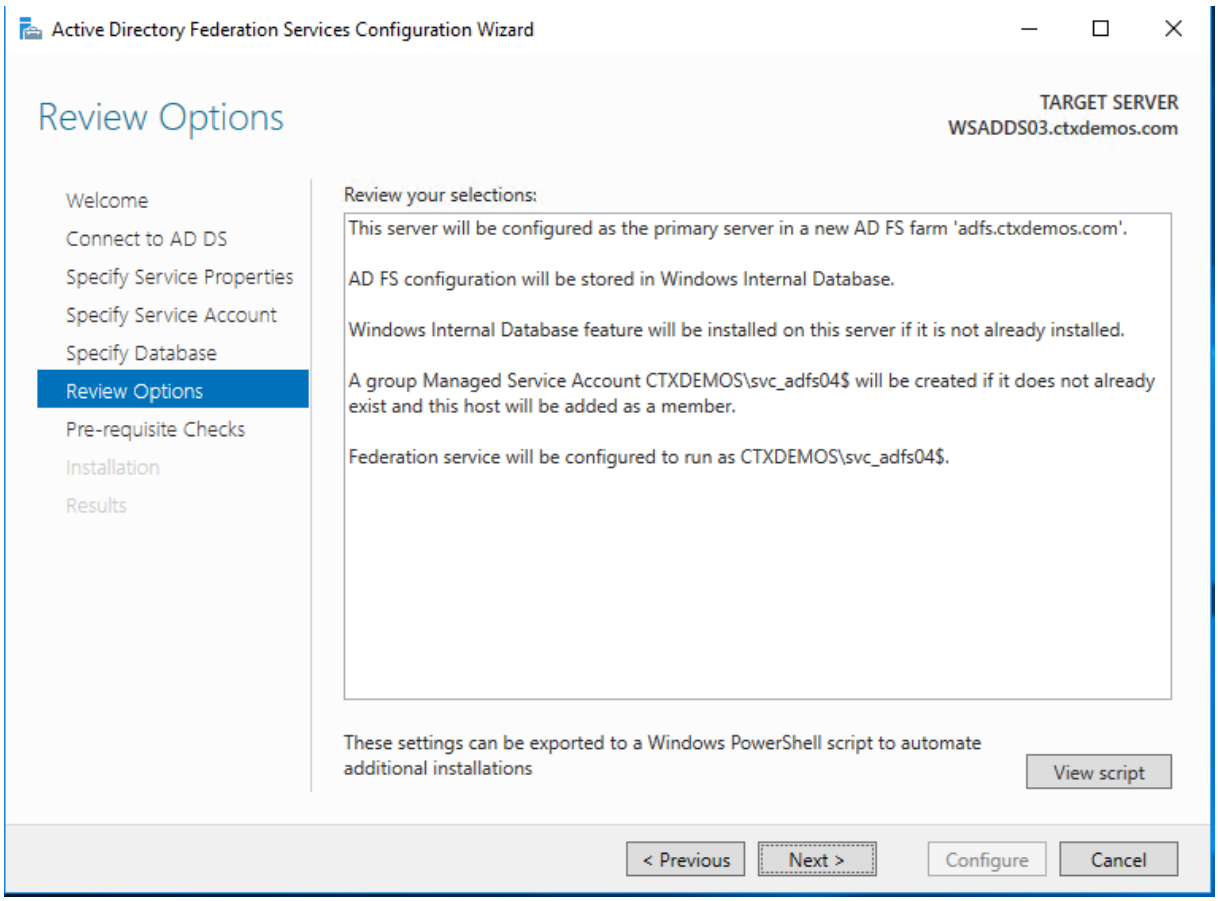
在“指定服务帐户”页上，选择“创建组托管服务帐户”，然后为此帐户输入唯一名称。Windows Server 2012 以后支持组托管服务帐户，并附带严格、复杂的密码，每 30 天自动更改一次。单击下一步。



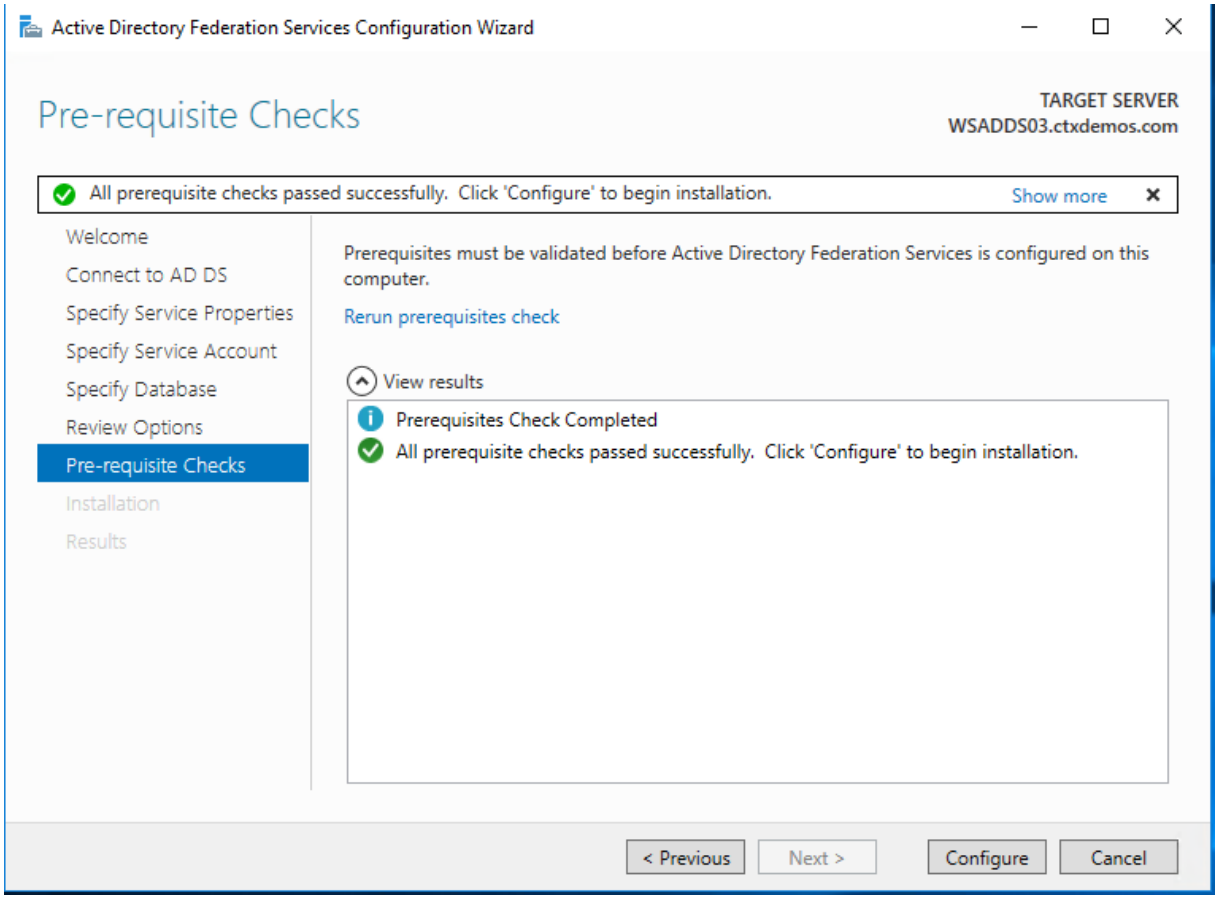
在“指定配置数据库”页上，选择指定 SQL Server 数据库的位置。单击下一步。



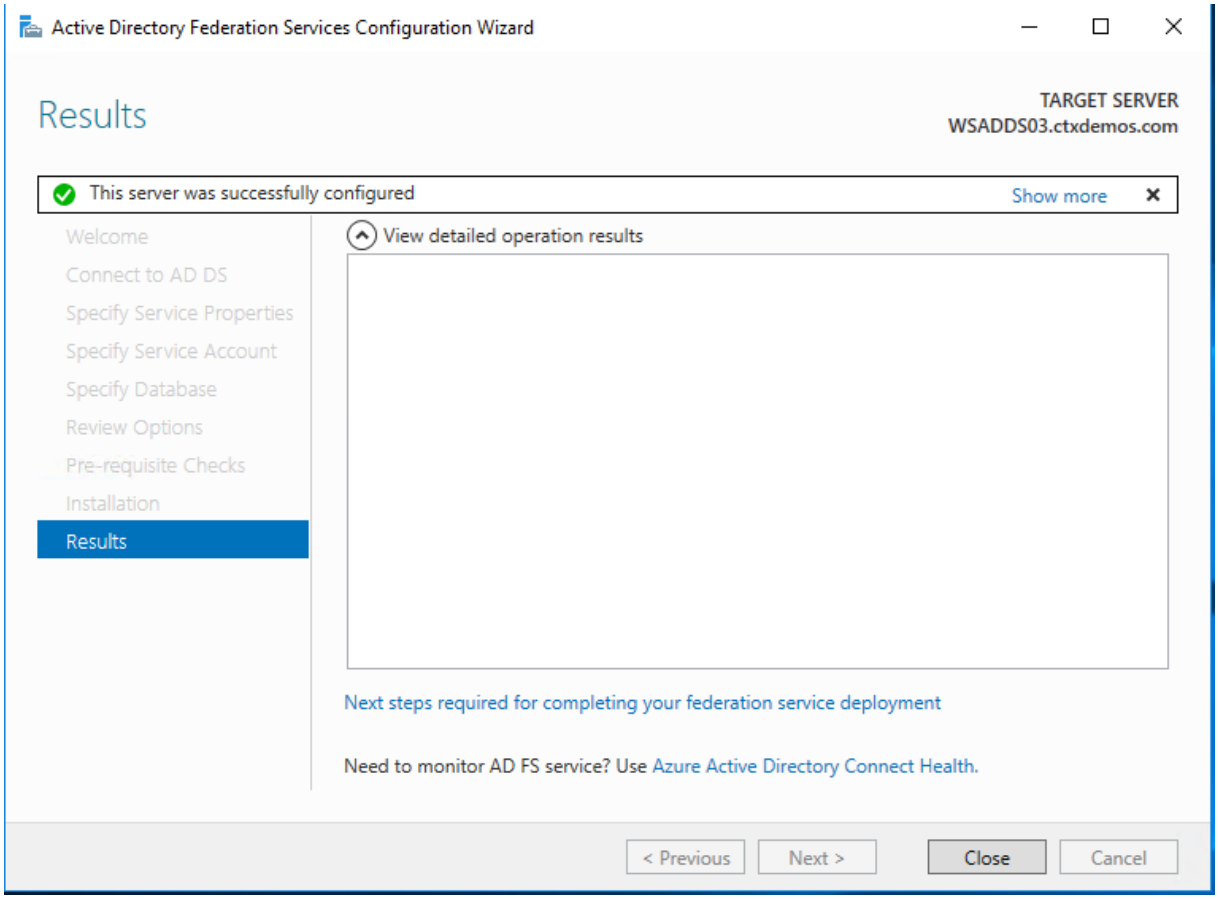
在“查看选项”页上，验证您的配置选择，然后单击“下一步”。



在“先决条件检查”页上，验证所有先决条件检查是否已成功完成，然后单击“配置”。



在“结 果”页面上，确保安装成功。单击关闭退出向导。



> 注意：
>
> 要完成以下步骤，需要您的 Azure 租户 ID。
>
> 您可以按照 Microsoft 文档文章 获取 AzureID 租户详细信息 中的步骤获取 Azure 租户 ID。
>
> Microsoft 文档还提供了有关 配置 AD FS 2016 和 Azure MFA 中的 Azure MFA 客户端 GUID 的信息。

配置 **AD FS** 场-自动

可以运行以下 PowerShell 脚本：

```
#
# Windows PowerShell script for AD FS Deployment
#
Import-Module ADFS
Install-AdfsFarm `
-CertificateThumbprint:"BD02F30D90A96EEE4A5934F2EA979E7A052584AE" `
-FederationServiceDisplayName:"CTXDEMOS STS" `
-FederationServiceName:"adfs.ctxdemos.com" `
-GroupServiceAccountIdentifier:"C
```



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

使用 **Azure MFA** 配置 **AD FS**

配置 **AD FS** 服务器

在每个 AD FS 服务器上，启动 PowerShell 并运行以下命令：

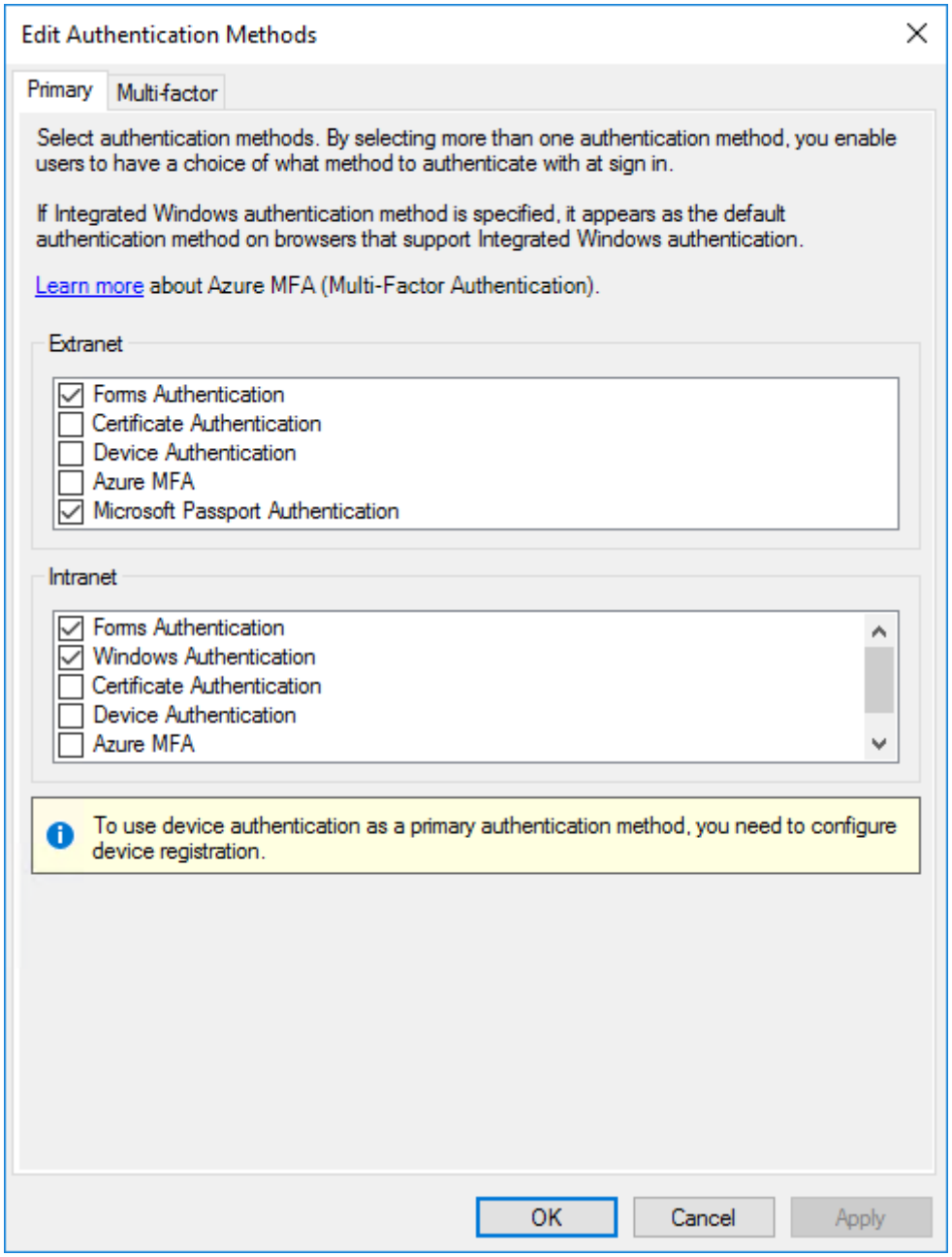
```
# Install Windows PowerShell MSOnline Module
Install-Module MSOnline

# Import Windows PowerShell MSOnline Module
Import-Module MSOnline

# Get the Azure Global Administrator credential
$credential = Get-Credential

# Sign in to your Azure Active Directory environment
Connect-MsolService -Credential $credential

# Set a variable for the Azure Tenant name
$azureTenantID = "ctxdemos.onmicrosoft.com"

# Set a variable for the Azure MFA Client GUID
$azureMFAClientGUID = "981f26a1-7f43-403b-a875-f8b09b8cd720"

# Generate a certificate for the Azure MFA on AD FS server
$azureMFACertificate = New-AdfsAzureMfaTenantCertificate -TenantId
    $azureTenantID

# Add the new credentials to the Azure MFA Client Service Principal
New-MsolServicePrincipalCredential -AppPrincipalId $azureMFAClientGUID
    -Type asymmetric -Usage verify -
Value $azureMFACertificate
<!--NeedCopy-->
```

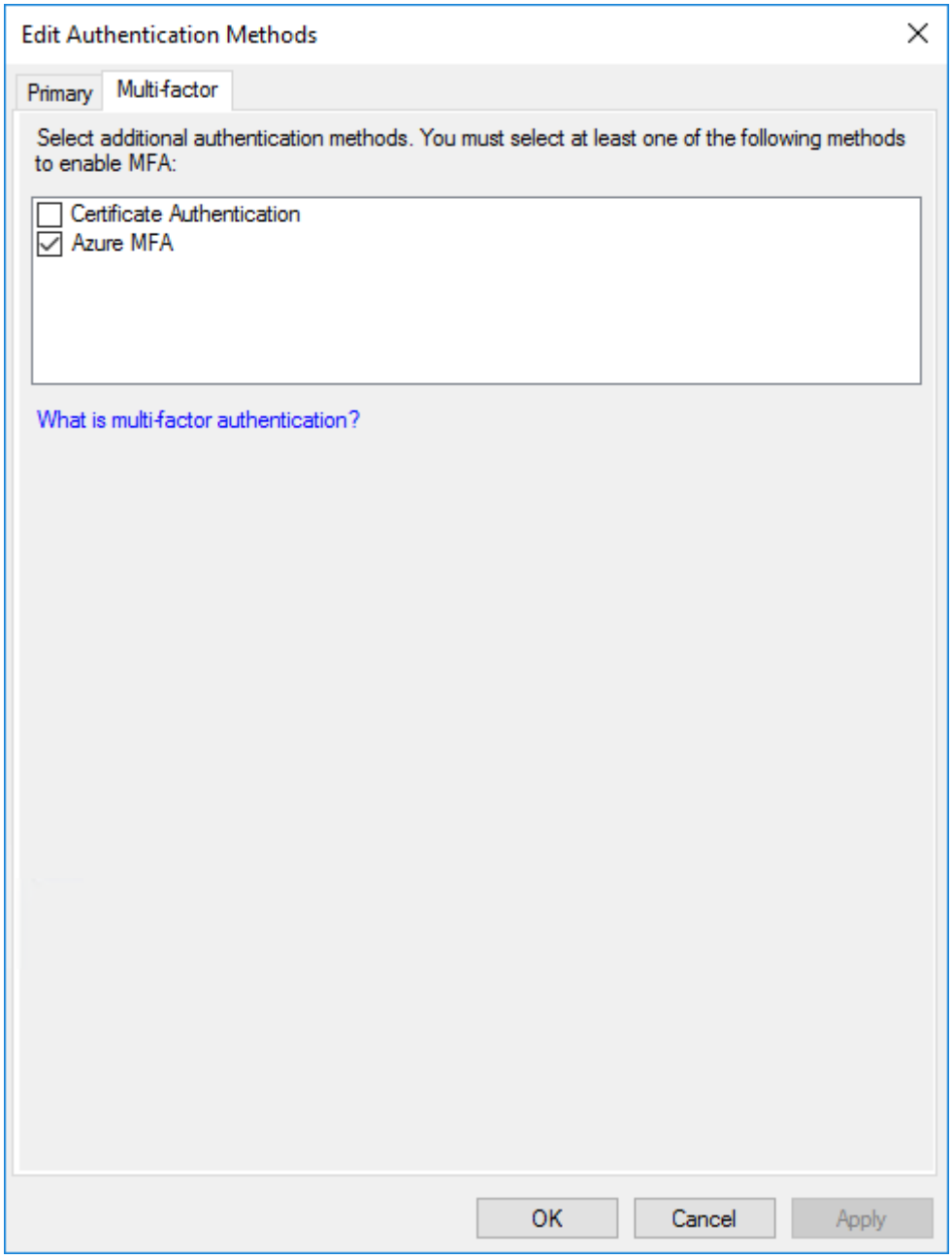
配置 **AD FS** 场

仅在其中一个 AD FS 服务器上，运行以下命令：


```
Set-AdfsAzureMfaTenant -TenantId $azureTenantID -ClientId
    $azureMFAClientGUID
<!--NeedCopy-->
```

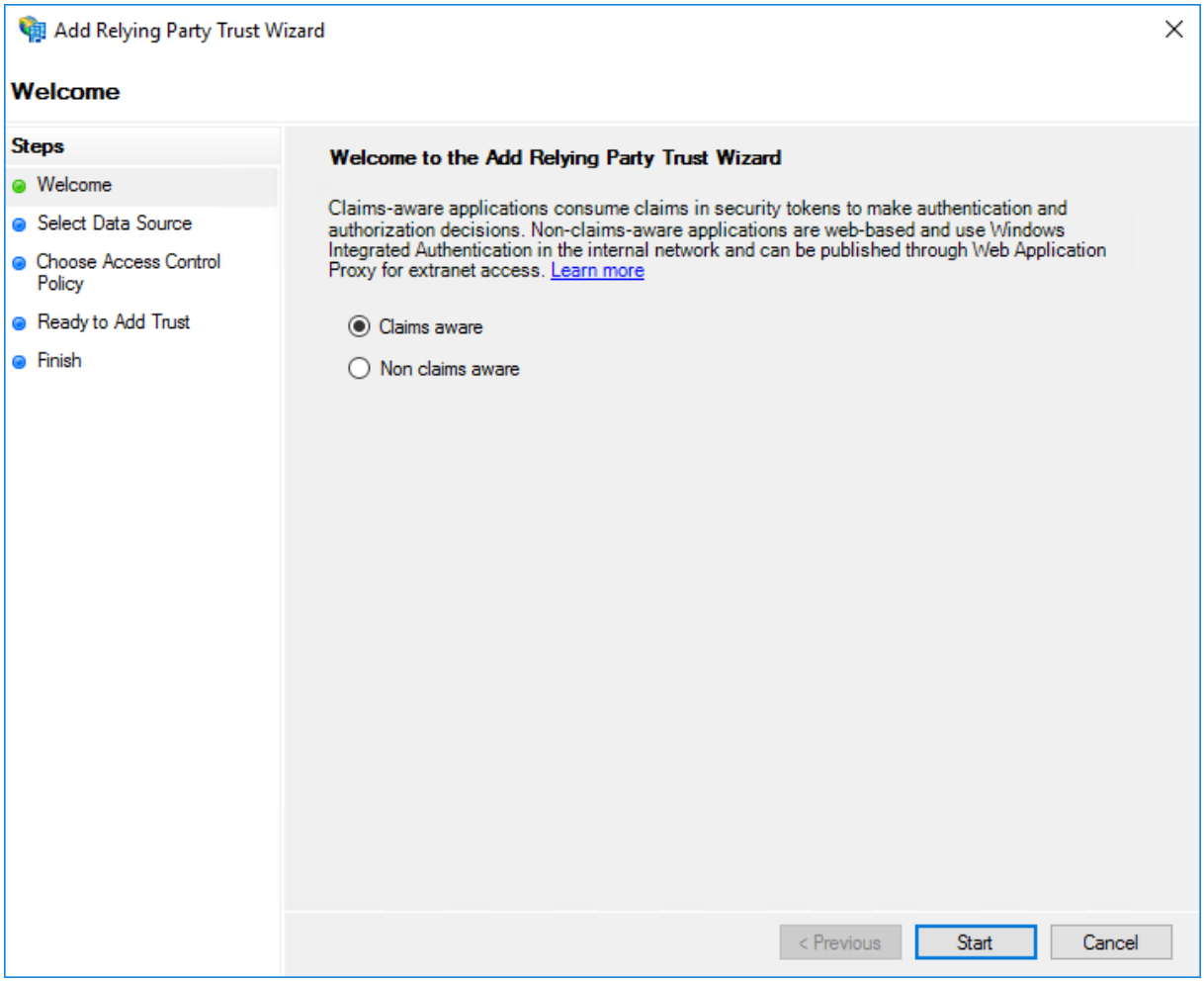
在每台服务器上重新启动 AD FS 服务。然后您会看到 Azure MFA 可用作内部网和外联网使用的主要身份验证方法和多重身份验证方法。



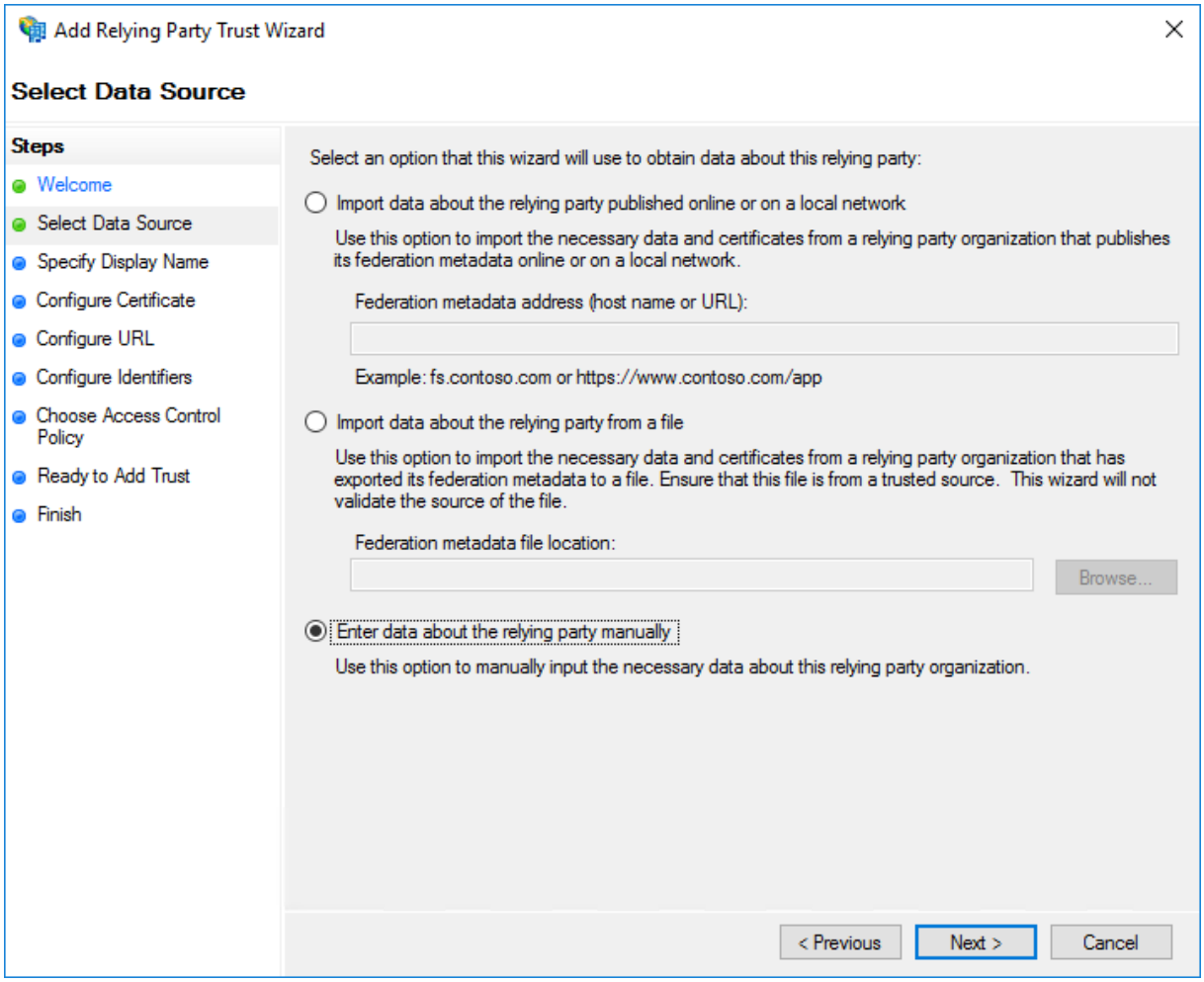


使用 **Citrix ADC** 配置 **AD FS**

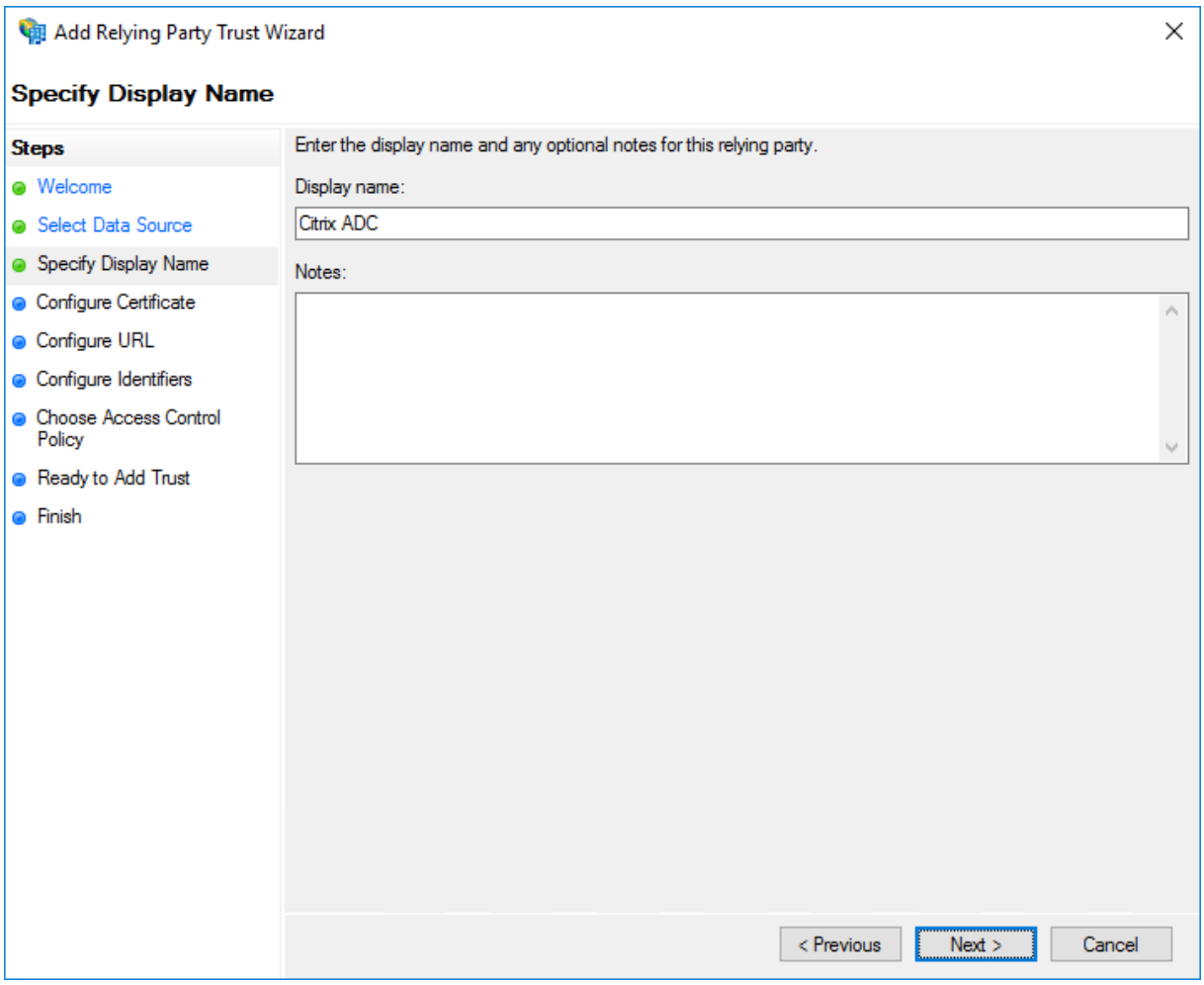
您需要在 AD FS 和 Citrix ADC 之间创建联合身份验证信任。在 AD FS 管理控制台中，导航到信赖方信任，然后选择添加信赖方信任。



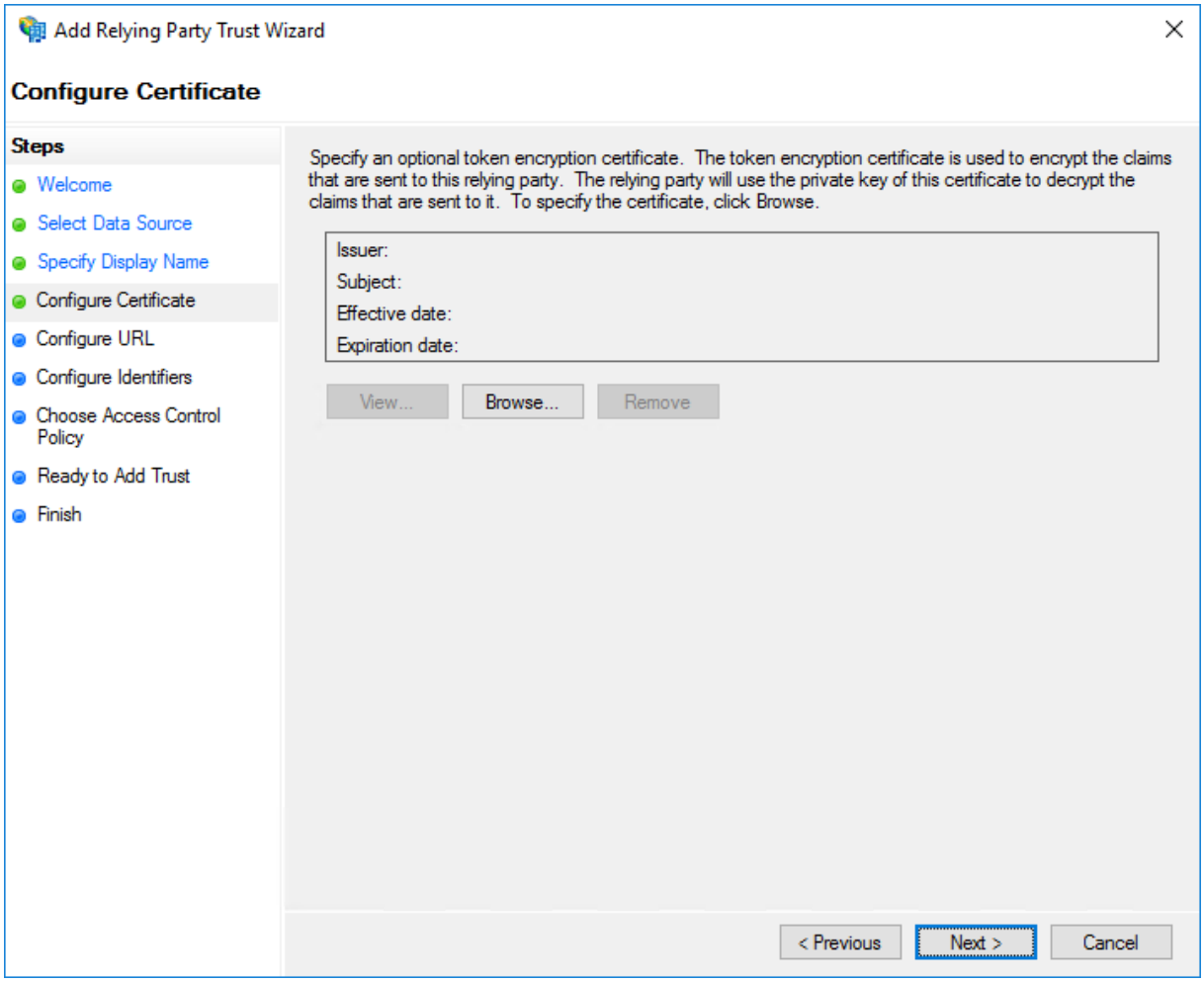
选择手动输入有关信赖方的数据，然后单击下一步。



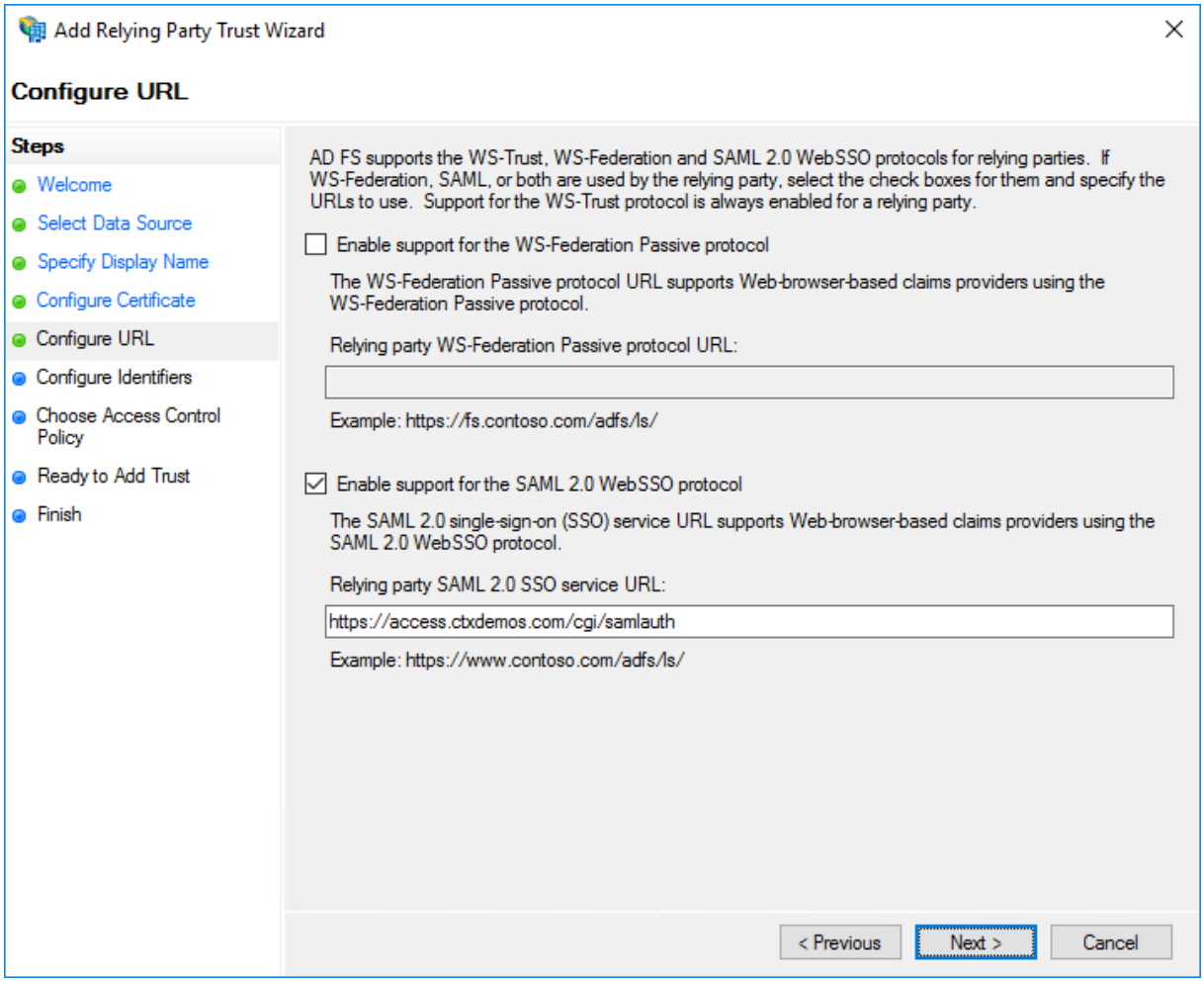
输入描述性显示名称和可选备注。单击下一步。



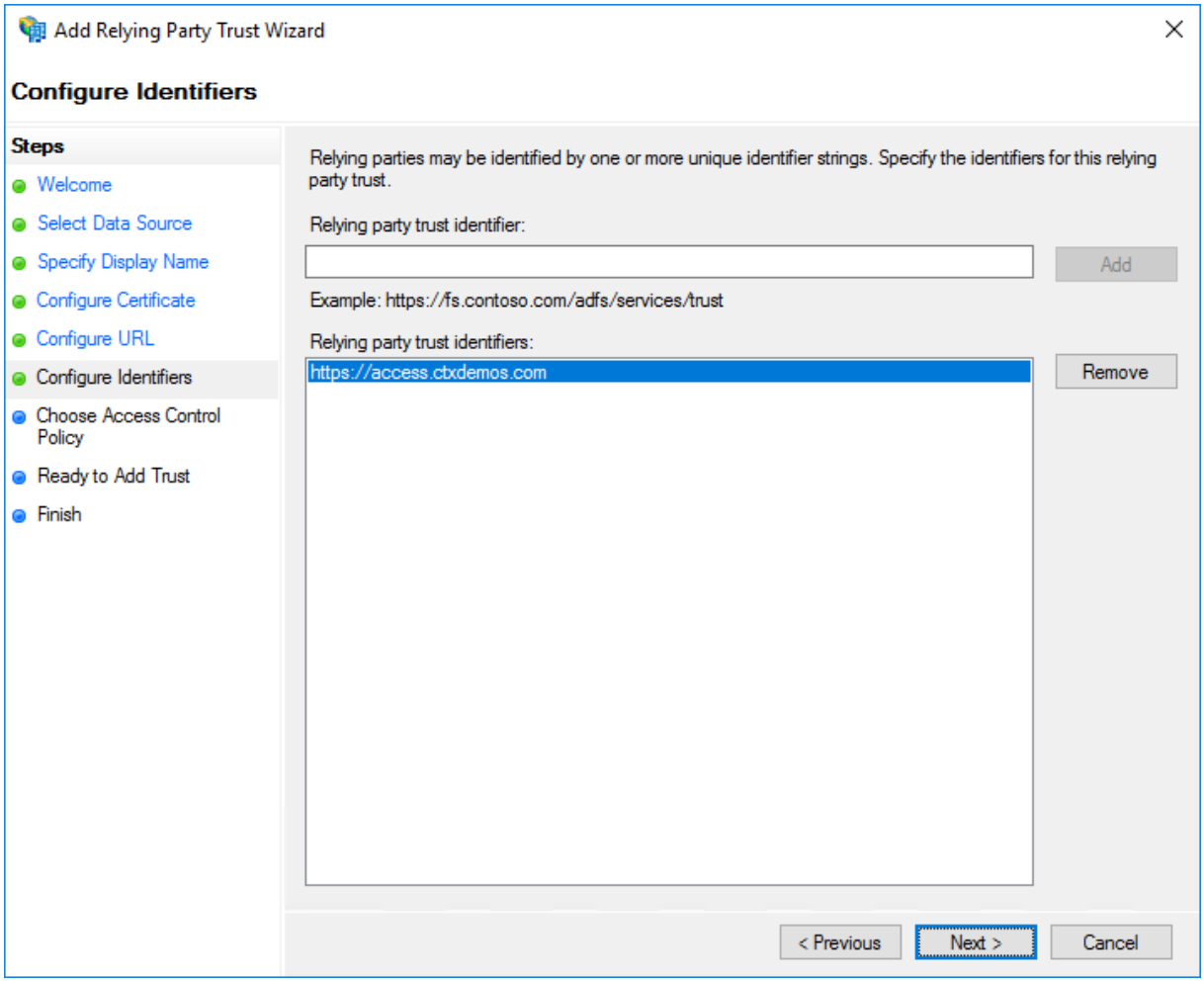
单击下一步。



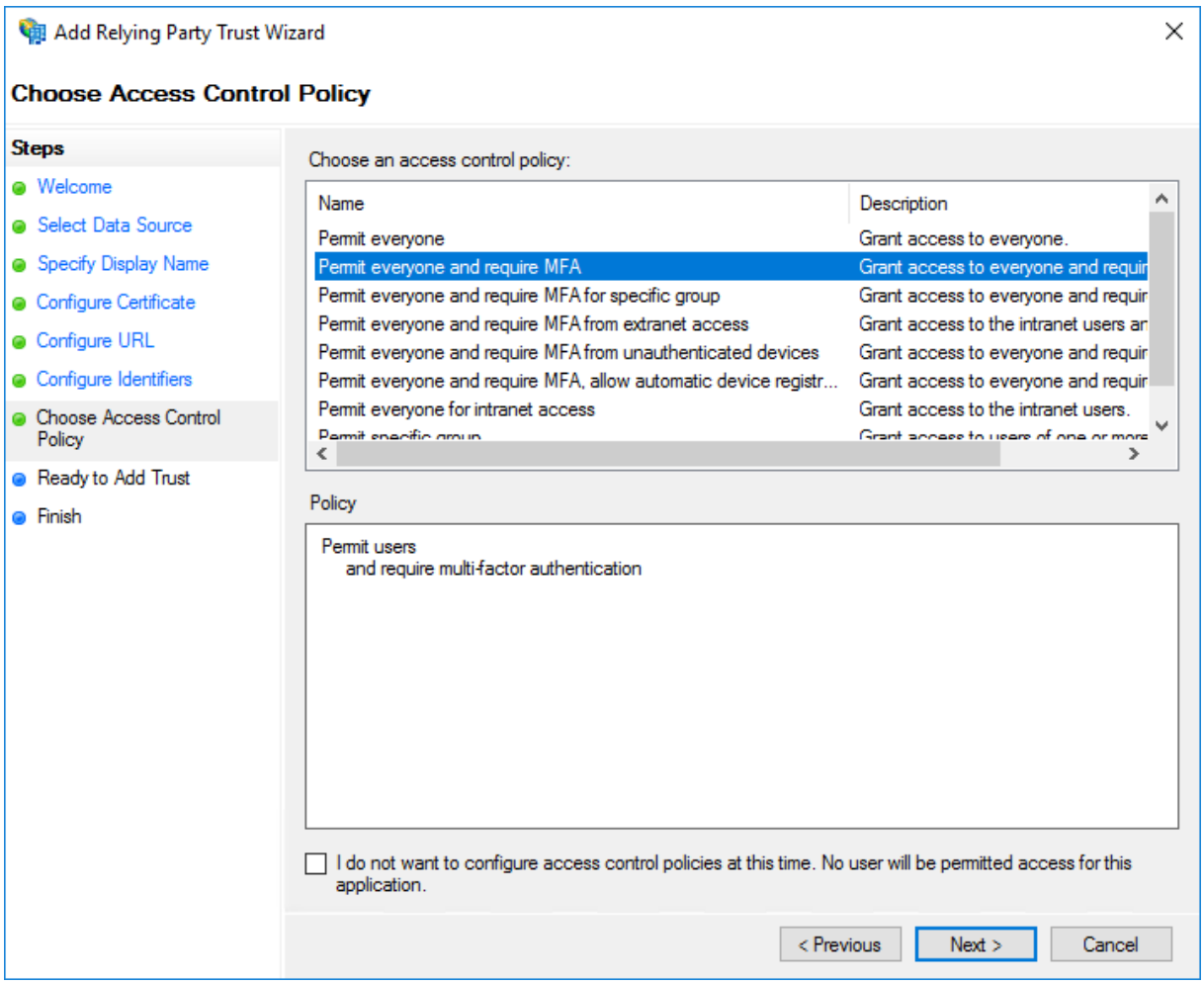
选择启用对 **SAML 2.0 WebSSO** 协议的支持，然后输入 `https://CitrixGatewayFQDN/cgi/samlauth`。在演示环境中，为 `https://access.ctxdemos.com/cgi/samlauth`。单击下一步。



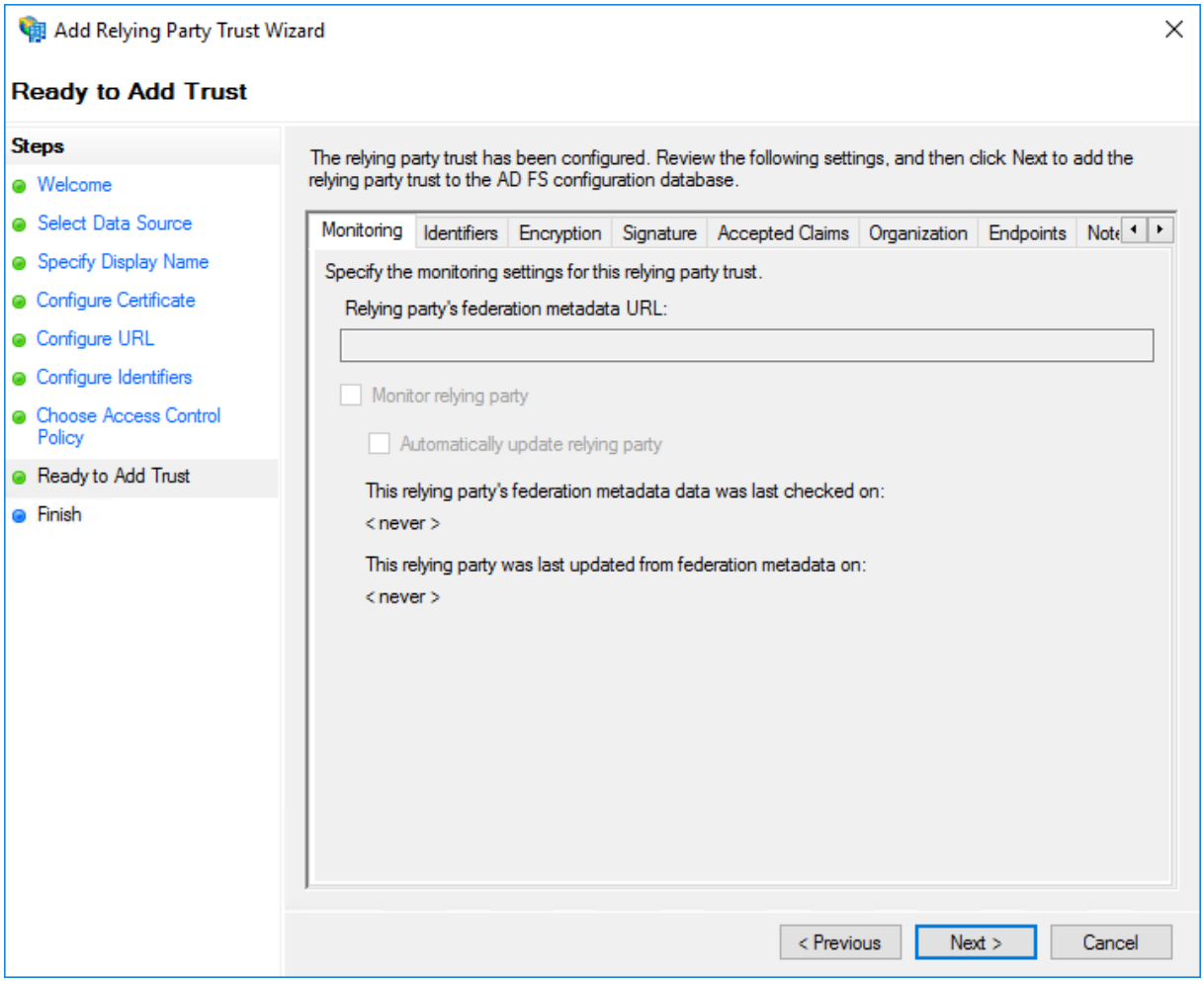
输入信赖方信任的唯一标识符字符串。在演示环境中，为 `https://access.ctxdemos.com`。此标识符将用作 Citrix ADC SAML 配置文件中的发布者 URL。单击下一步。



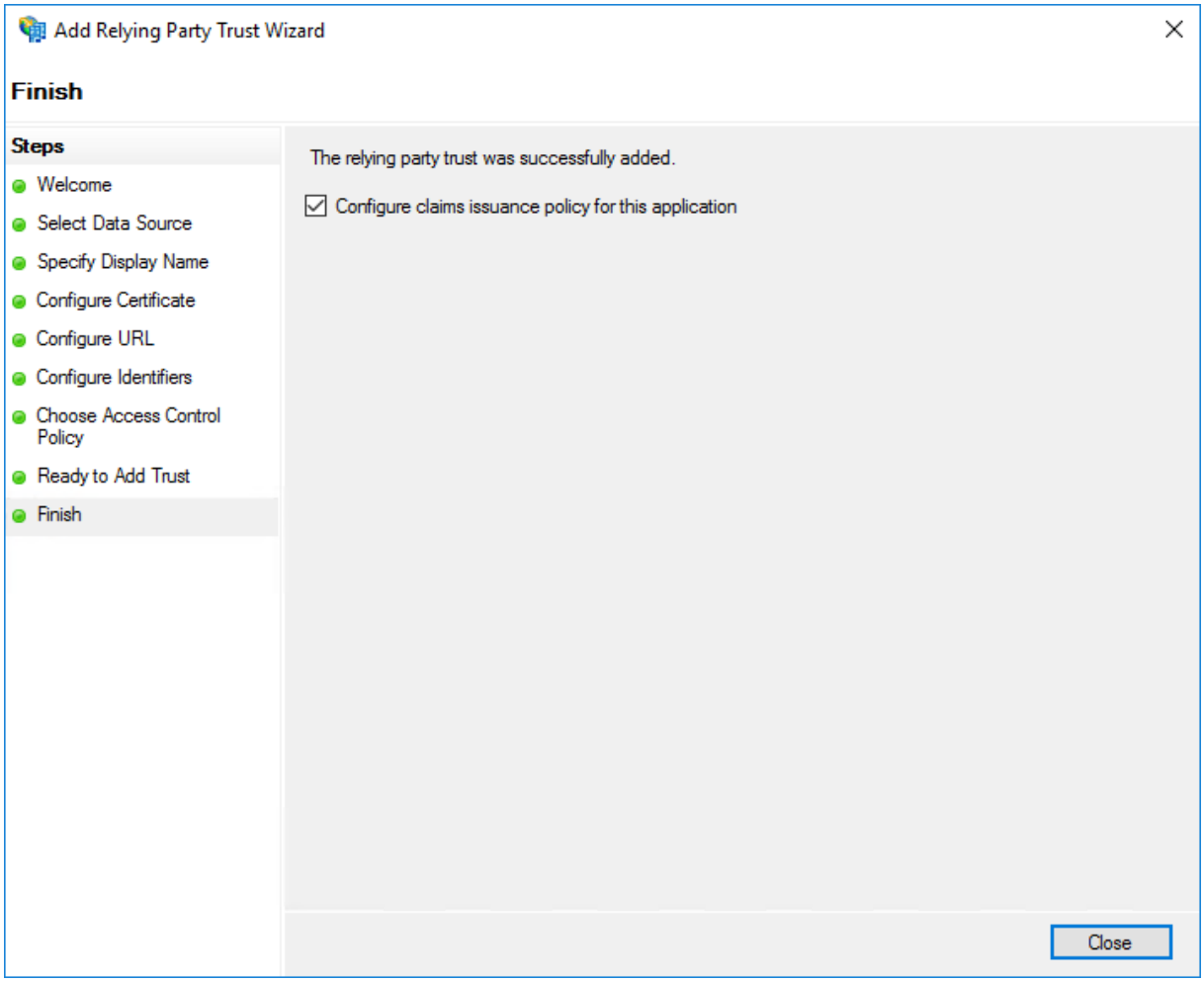
在“选择访问控制策略”页面上，选择“允许所有人并要求 **MFA**”。单击下一步。



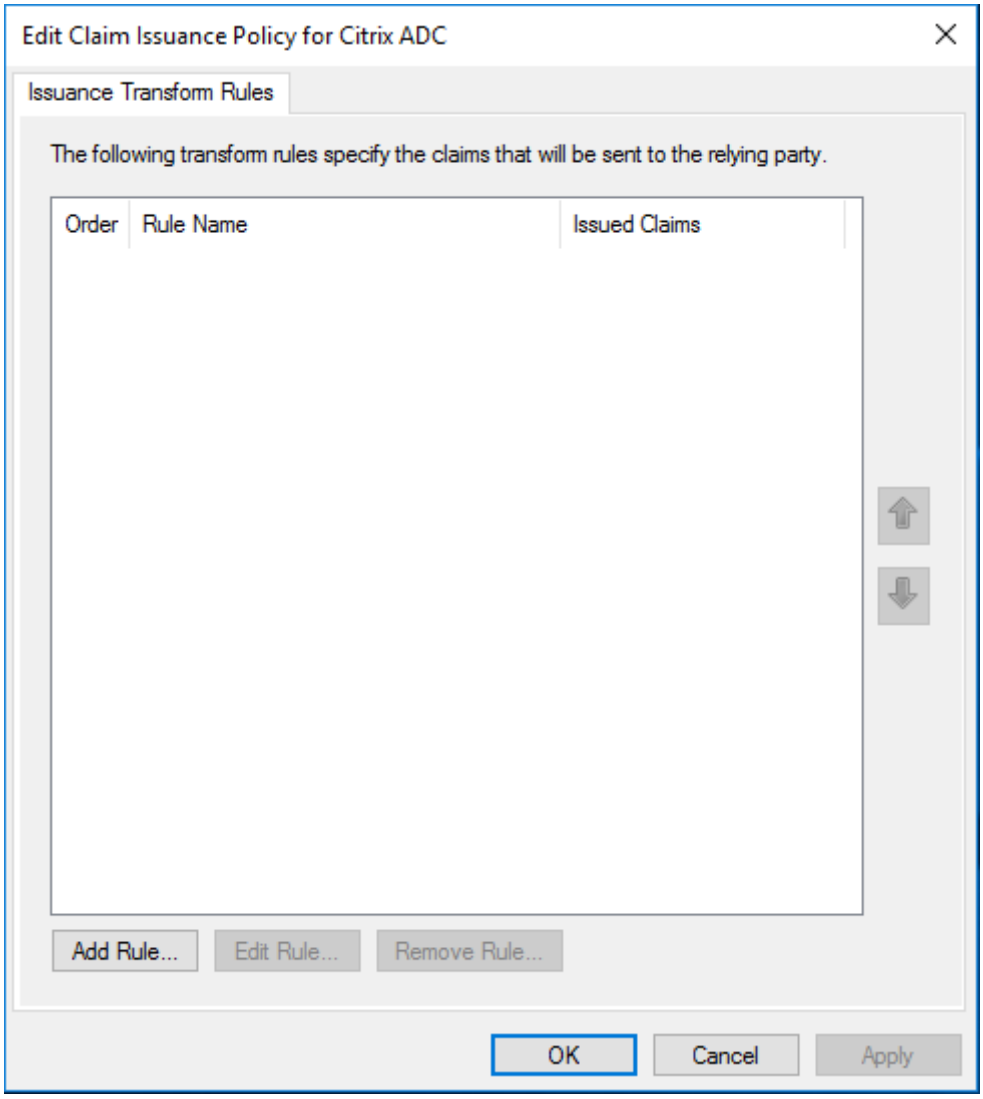
单击下一步。



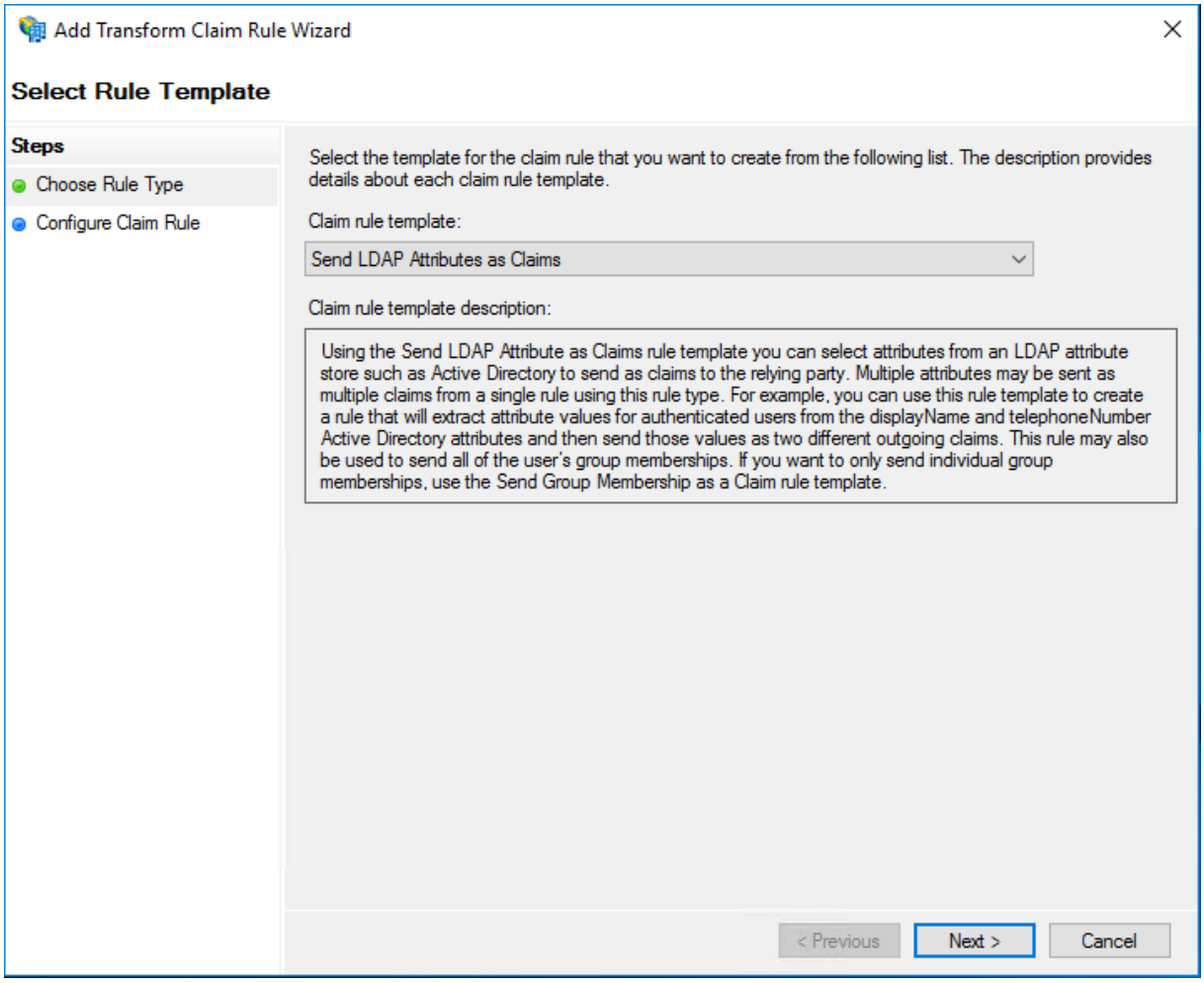
在“完成”页面上，选择“为此应用程序配置声明发布策略”。单击 **Close**（关闭）。



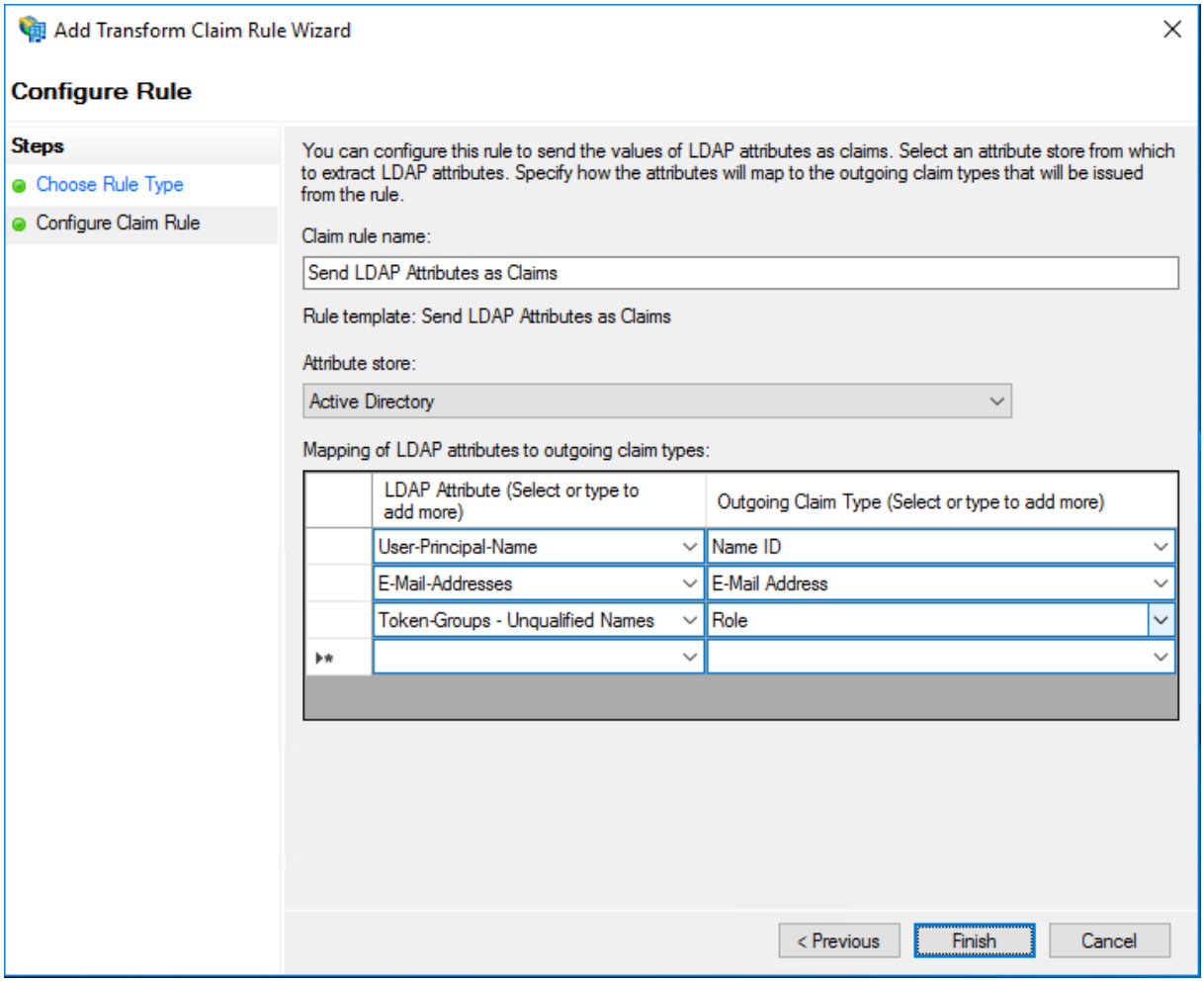
在“发行转换规则”页上，单击“添加规则”。



单击下一步。

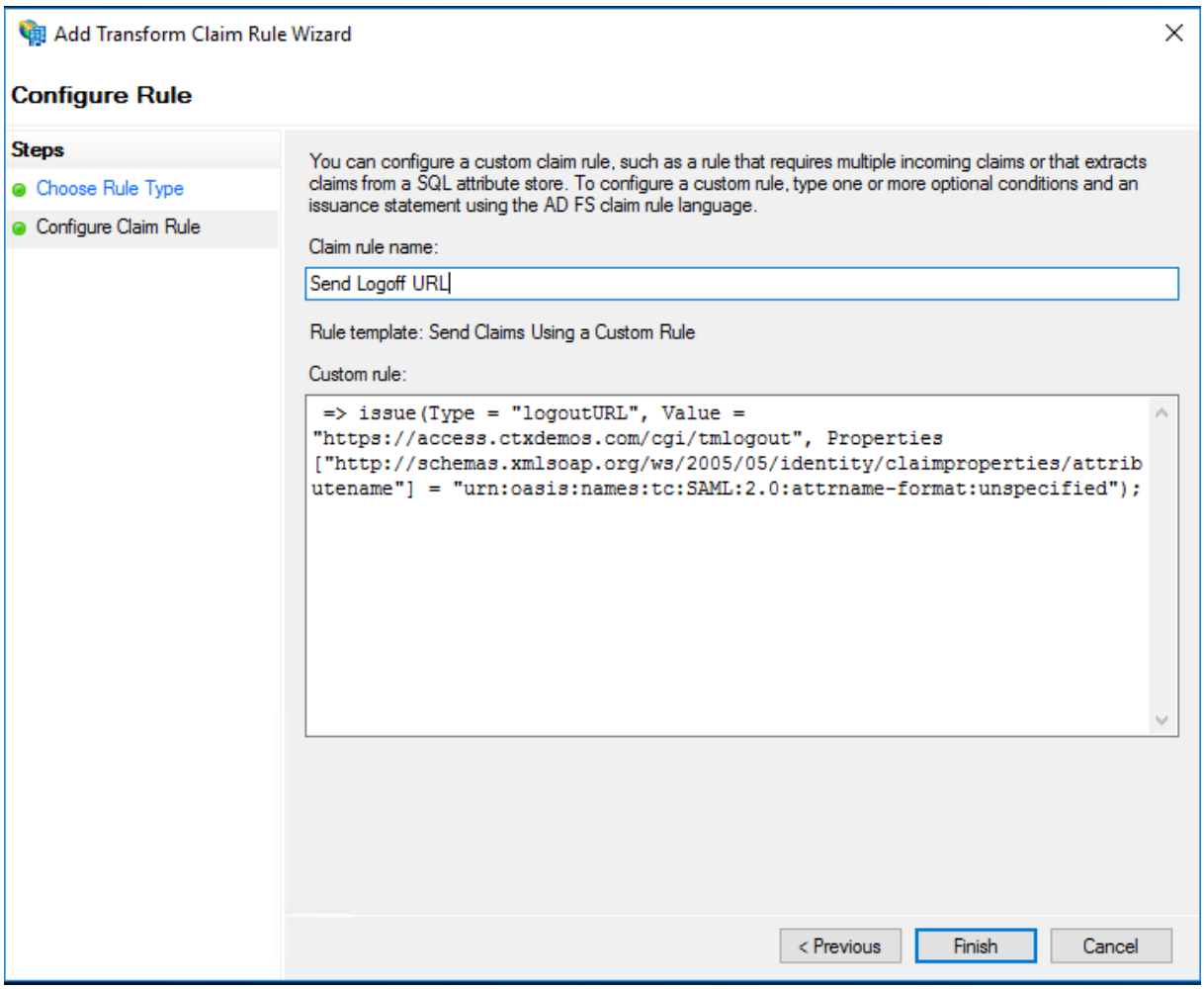


在声明规则名称字段中输入描述性名称。在属性存储下，选择 **Active Directory**。然后选择以下选项：**LDAP** 属性和传出声明类型。



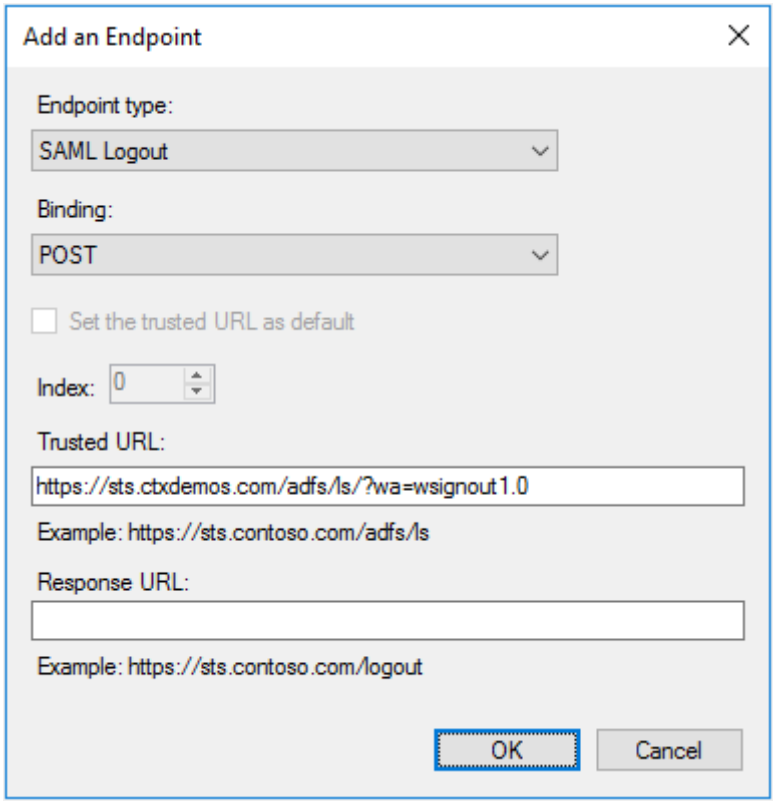
创建新规则并使用使用自定义规则作为声明规则模板发送声明。输入声明规则名称的描述性名称，并为自定义规则输入以下字符串：

```
1 => issue(Type = "logoutURL", Value = "https://access.ctxdemos.com/cgi/
    tmlogout", Properties["http://schemas.xmlsoap.org/ws/2005/05/
    identity/claimproperties/attributename"] = "urn:oasis:names:tc:SAML
    :2.0:attrname-format:unspecified");
2 <!--NeedCopy-->
```

创建“索赔发放策略”后，单击“确定”。

右键单击“信 赖方信任**”>“Citrix ADC**”，然后选择“属性”。通过单击添加 **SAML** 进行注销，选择端点并添加端点。从端点类型列表中，选择 **SAML** 注销。对于“绑定”，选择“**POST**”，对于“受信任的 **URL**”，输入 `https://sts.ctxdemos.com/adfs/ls/?wa=wsignout1.0`。注销 Citrix ADC 时，这将充当注销 URL。单击 **OK**（确定）。



右键单击“信 赖方信任**”>“Citrix ADC**”，然后选择“属性”。选择加密并添加安装在 Citrix Gateway 上的公有 SSL 证书。此证书将用于解密来自 Citrix ADC 的传入 SML 请求。在“签名”选项卡上重复相同的操作。此证书将用于检查传入的 SAML 请求的签名。单击 **OK**（确定）。

启用 **IdP** 启动的登录页面

您可以启用 AD FS IdP 启动的登录页面。您将使用 IdP 启动的登录向未注册的 MFA 用户显示自定义错误页面。要启用，请运行以下命令：

```
1 Set-AdfsProperties -EnableIdPInitiatedSignonPage $true
2 <!--NeedCopy-->
```

测试 **AD FS** 场

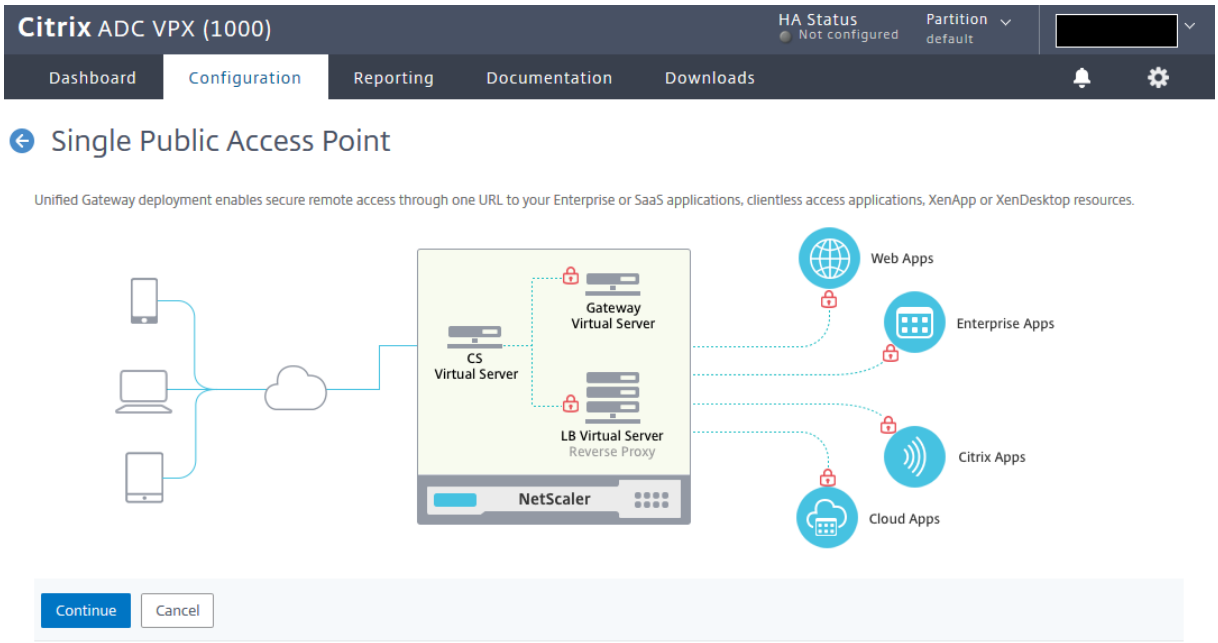
打开 Web 浏览器并导航至：

- https://sts.ctxdemos.com/FederationMetadata/2007-06/FederationMetadata.xml
- https://sts.ctxdemos.com/adfs/fs/federationserverservice.asmx
- https://sts.ctxdemos.com/adfs/ls/idpinitatedsignon.aspx

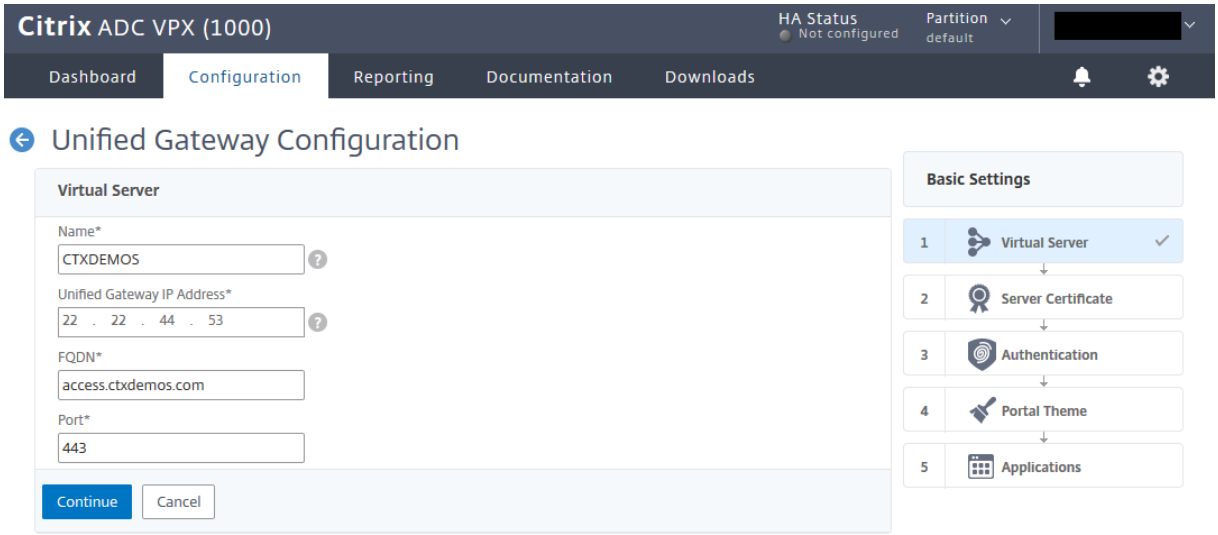
Citrix ADC 和 Citrix Gateway

配置 Citrix Gateway

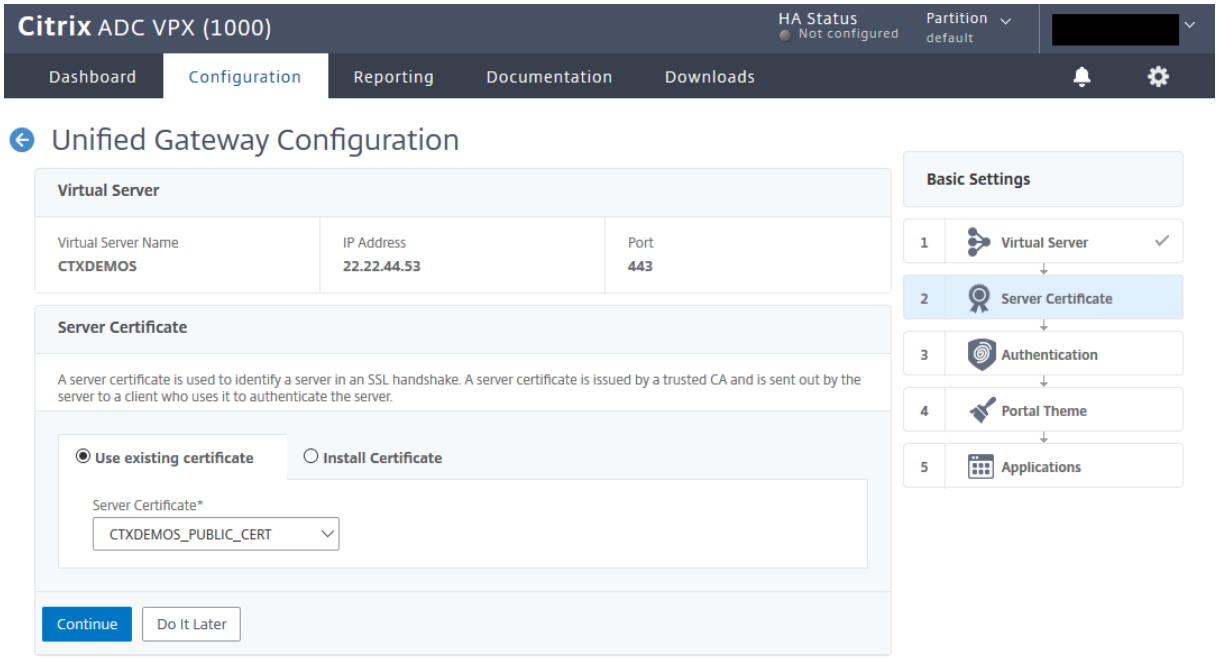
您可以通过向导配置 Citrix Gateway。登录到 Citrix ADC 管理 GUI，导航到 **Unified Gateway**，然后单击创建新网关。然后单击继续。



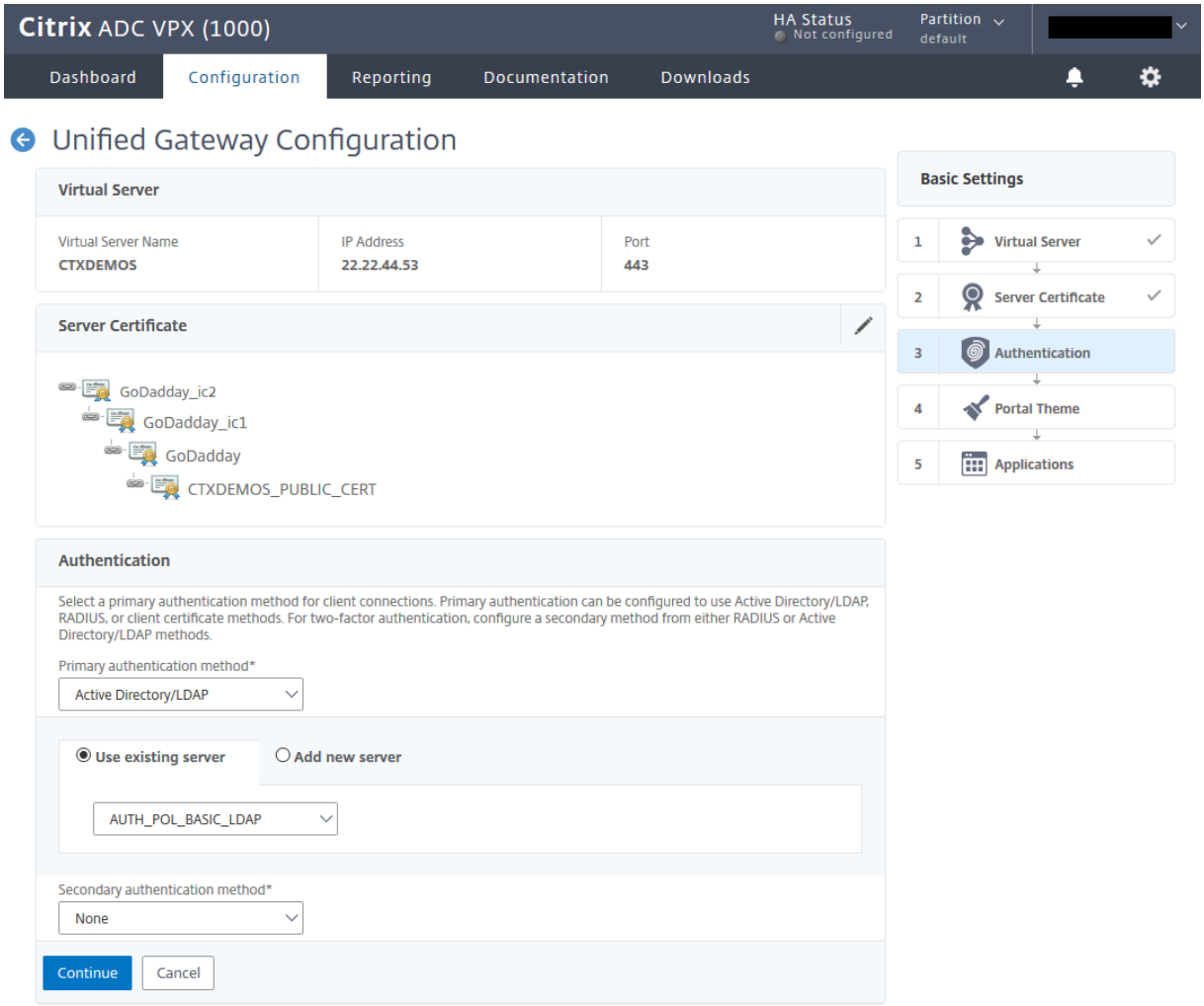
输入 **Unified Gateway** 的名称、IP 和 FQDN，然后单击继续。



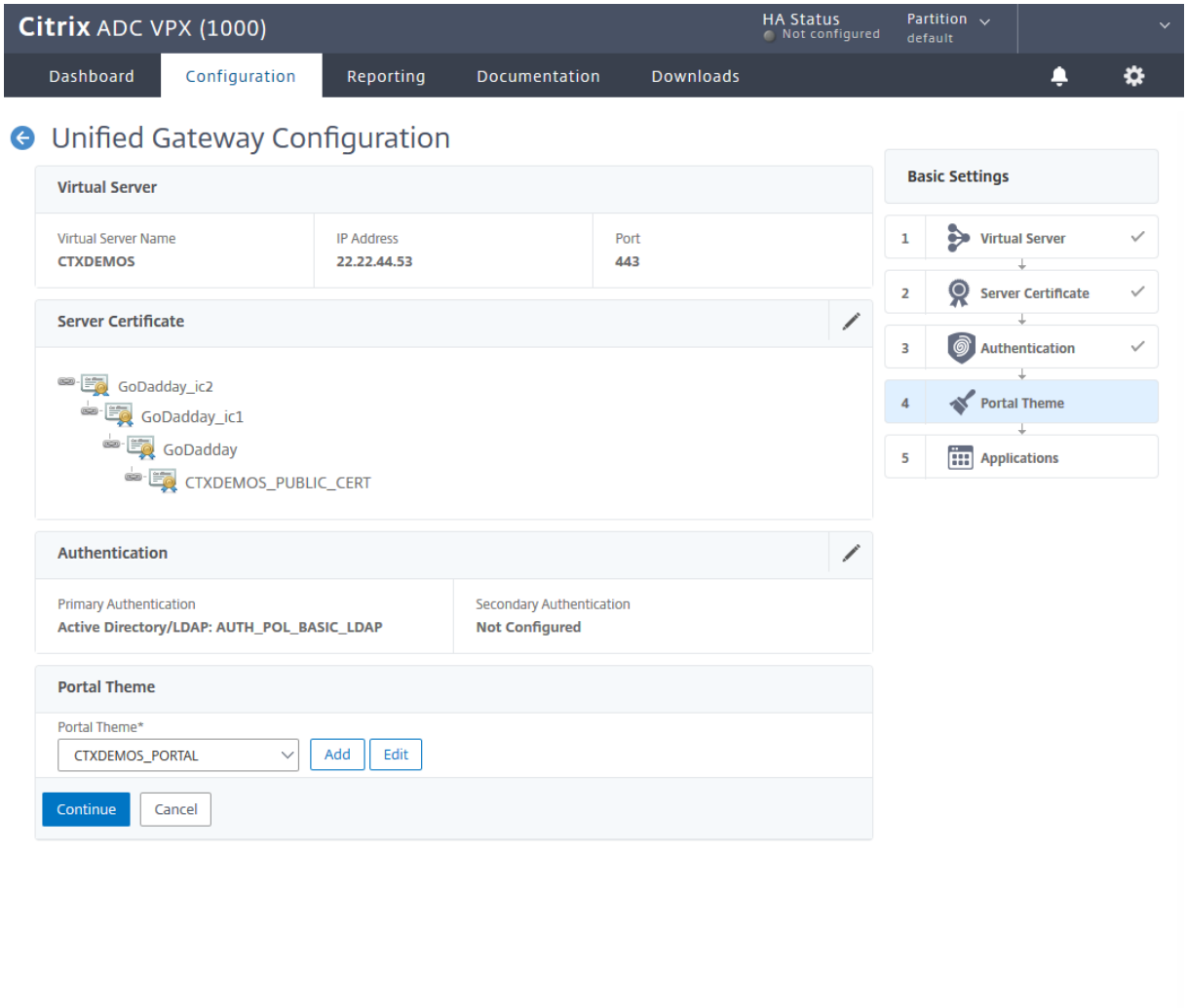
选择公用 SSL 证书，然后单击继续。



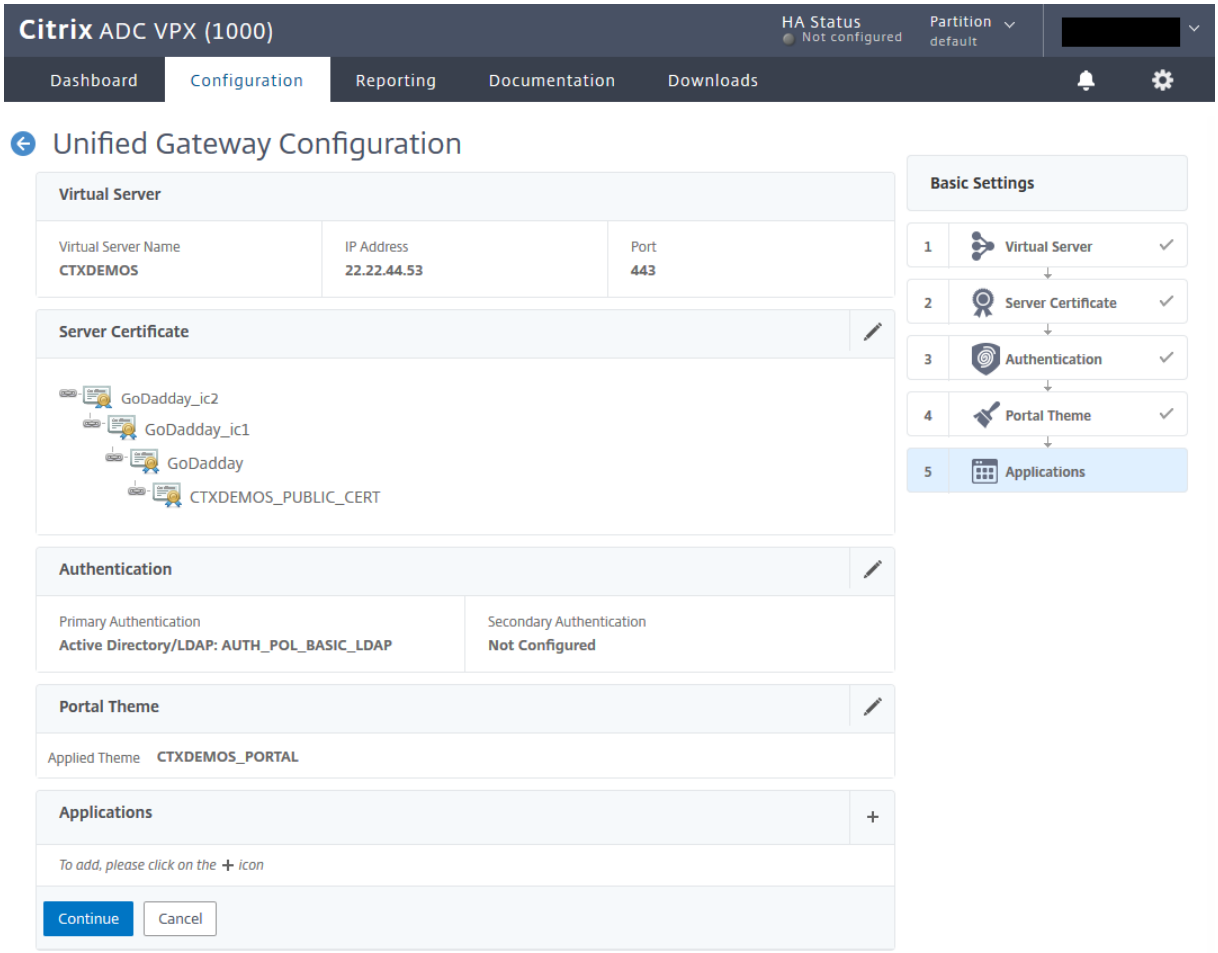
创建基本的 LDAP 策略并将其绑定到 **Unified Gateway**。单击继续。



创建基于 RfWebUI 的门户主题并将其绑定到 **Unified Gateway**。单击继续。

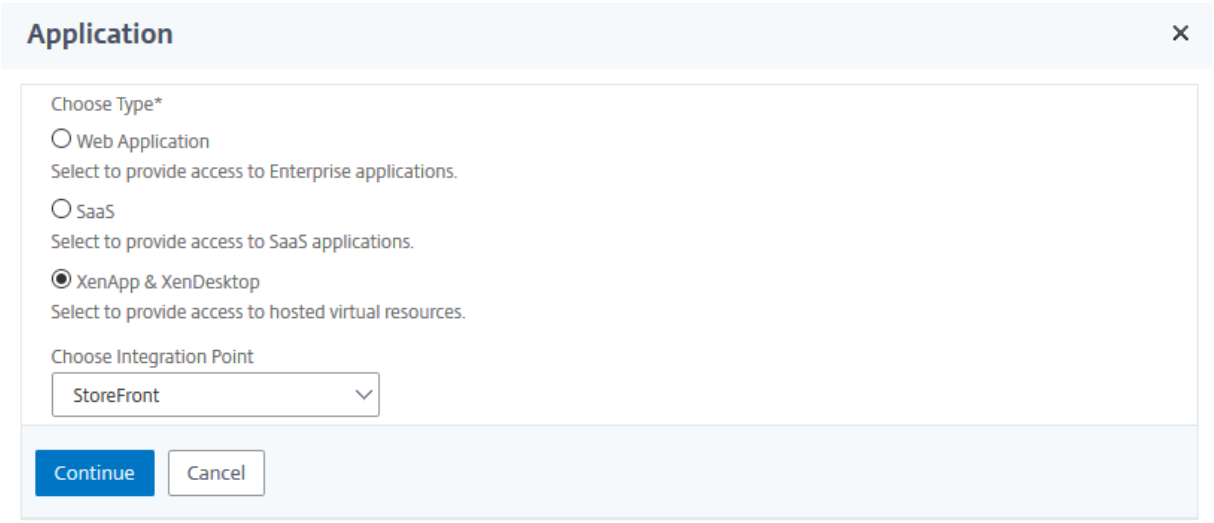


选择应用程序前面的加号 (+) 以将 Citrix Gateway 与 StoreFront 集成。

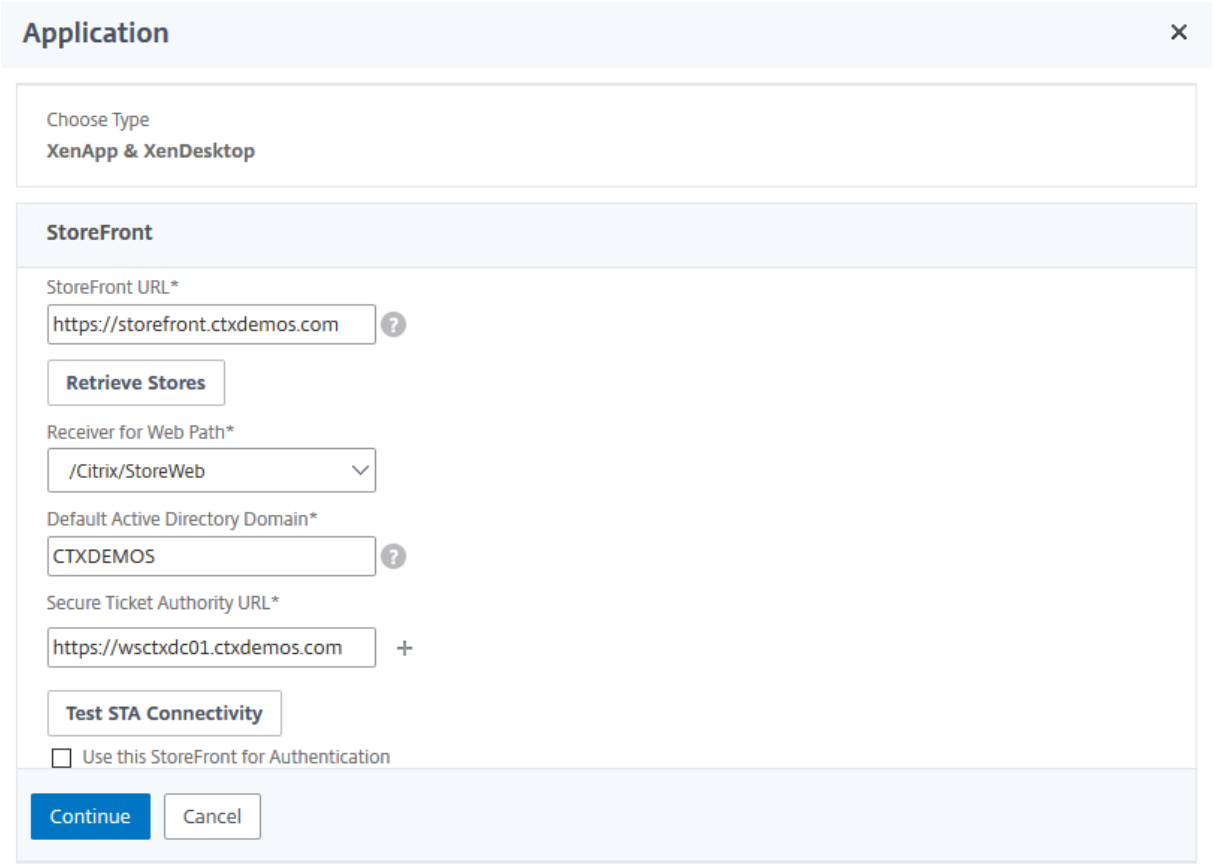


将 **Citrix StoreFront** 集成到 **Citrix Gateway**

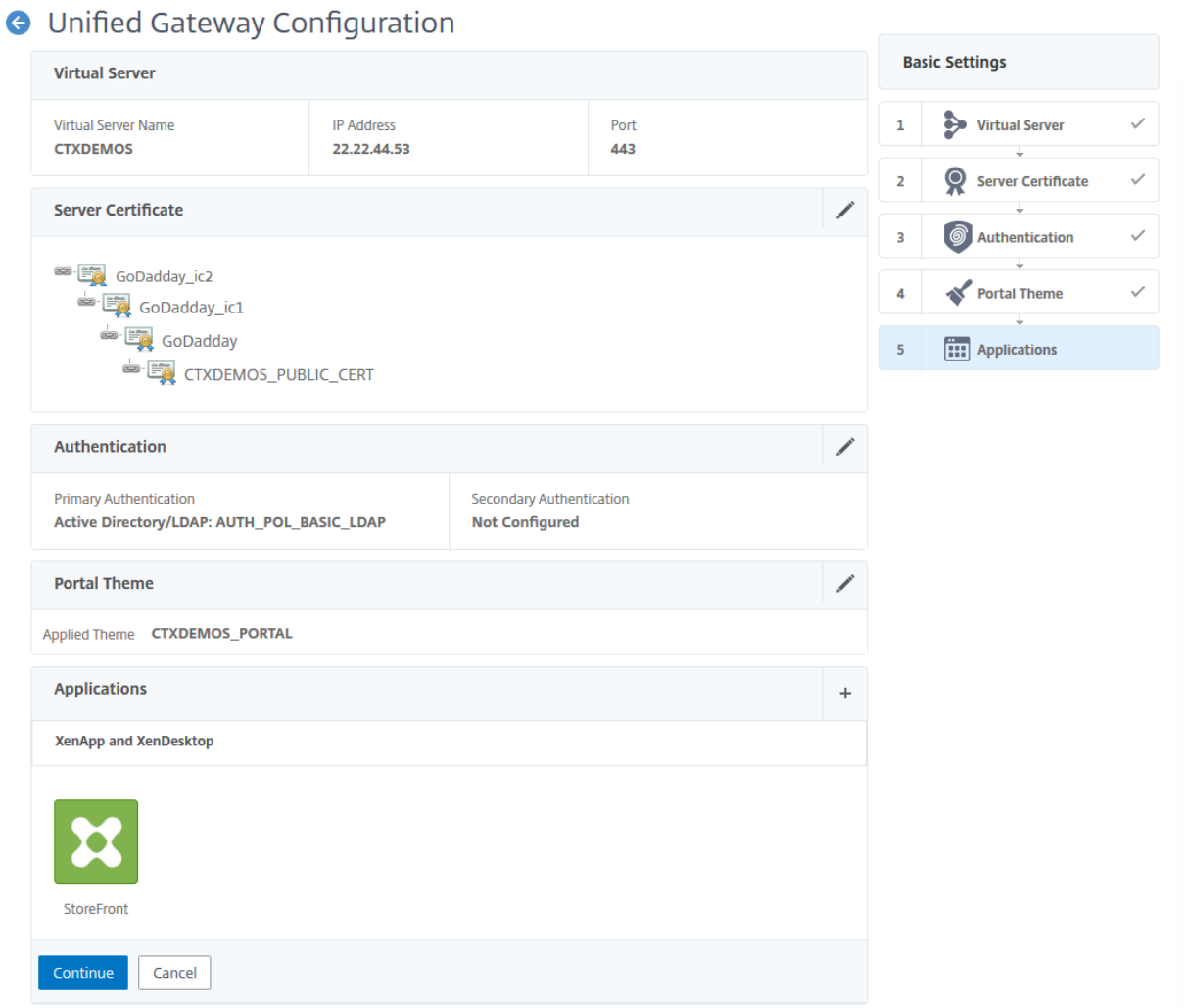
在“应用程序”页面上，选择 **XenApp** 和 **XenDesktop**，然后从选择集成点列表中选择 **StoreFront**。单击继续。



输入 StoreFront URL，然后单击检索应用商店。然后输入默认 **Active Directory** 域和 **Secure Ticket Authority URL** 设置。单击测试 **STA** 连接，然后单击继续。



单击 完成，然后单击 继续。



配置 **Citrix Gateway** 并与 **StoreFront** 集成 **– CLI**

```
# Create Session Policy and Action for Citrix Receiver
add vpn sessionAction AC_OS_22.22.44.50 -transparentInterception OFF -
    defaultAuthorizationAction ALLOW -SSO ON -icaProxy ON -wihome "https
    ://access.ctxdemos.com/Citrix/ExternalWeb" -ClientChoices OFF -
    ntDomain CTXDEMOS -clientlessVpnMode OFF -storefronturl "https://
    access.ctxdemos.com"
add vpn sessionPolicy PL_OS_22.22.44.50 "HTTP.REQ.HEADER("User-Agent").
    CONTAINS("CitrixReceiver") && HTTP.REQ.HEADER("User-Agent").CONTAINS
    ("CitrixVPN").NOT && HTTP.REQ.HEADER("User-Agent").CONTAINS("
    NSGiOSplugin").NOT" AC_OS_22.22.44.50

# Create Session Policy and Action for Citrix Web Client
```



```
6  add vpn sessionAction AC_WB_22.22.44.50 -transparentInterception ON -
       defaultAuthorizationAction ALLOW -forceCleanup cookie -SSO ON -
       ssoCredential PRIMARY -icaProxy OFF -wihome "https://storefront.
       ctxdemos.com/Citrix/ExternalWeb" -wiPortalMode COMPACT -
       ClientChoices OFF -ntDomain CTXDEMOS -clientlessVpnMode ON -
       clientlessPersistentCookie ALLOW
7  add vpn sessionPolicy PL_WB_22.22.44.50 "HTTP.REQ.HEADER("User-Agent").
       CONTAINS("CitrixReceiver").NOT" AC_WB_22.22.44.50
8
9  # Create Session Policy and Action for Citrix Gateway Client
10 add vpn sessionAction UG_VPN_SAct_22.22.44.50 -transparentInterception
       ON -defaultAuthorizationAction ALLOW -SSO ON -ClientChoices ON -
       clientlessVpnMode ON
11 add vpn sessionPolicy UG_VPN_SPol_22.22.44.50 true UG_VPN_SAct_22
       .22.44.50
12
13 # Create Responder Policy and Action for Gateway Logout
14 add responder action RESACT_GATEWAY_LOGOFF_REDIRECT redirect ""https://
       " + HTTP.REQ.HOSTNAME.HTTP_URL_SAFE" -responseStatusCode 302
15 add responder policy RESPOL_GATEWAY_LOGOFF_REDIRECT "HTTP.REQ.URL.
       CONTAINS("/cgi/logout")" RESACT_GATEWAY_LOGOFF_REDIRECT
16
17 # Create Citrix Gateway vServer
18 add vpn vserver UGVS_VPN_UGCTXDEMOS SSL 0.0.0.0 -loginOnce ON -
       Listenpolicy NONE -vserverFqdn access.ctxdemos.com
19 set ssl vserver UGVS_VPN_UGCTXDEMOS -ssl3 DISABLED -tls1 DISABLED -
       tls11 DISABLED -tls13 ENABLED -ocspStapling ENABLED -HSTS ENABLED -
       maxage 157680000 -IncludeSubdomains YES
20 bind ssl vserver UGVS_VPN_UGCTXDEMOS -certkeyName CTXDEMOS_PUBLIC_CERT
21 bind ssl vserver UGVS_VPN_UGCTXDEMOS -cipherName
       CTXDEMOS_FRONTEND_APLUS
22 bind vpn vserver UGVS_VPN_UGCTXDEMOS -portaltheme CTXDEMOS_PORTAL
23 bind vpn vserver UGVS_VPN_UGCTXDEMOS -staServer "https://wsctxdc01.
       ctxdemos.com"
24 bind vpn vserver UGVS_VPN_UGCTXDEMOS -policy
       RESPOL_GATEWAY_LOGOFF_REDIRECT -priority 100 -gotoPriorityExpression
        END -type REQUEST
25 bind vpn vserver UGVS_VPN_UGCTXDEMOS -policy PL_OS_22.22.44.50 -
       priority 100 -gotoPriorityExpression NEXT -type REQUEST
26 bind vpn vserver UGVS_VPN_UGCTXDEMOS -policy PL_WB_22.22.44.50 -
       priority 110 -gotoPriorityExpression NEXT -type REQUEST
27 bind vpn vserver UGVS_VPN_UGCTXDEMOS -policy UG_VPN_SPol_22.22.44.50 -
       priority 58000 -gotoPriorityExpression NEXT -type REQUEST
28
29 # Create Content Switching Policy and Action for Citrix Gateway
```



```
add cs action CSACT_UGCTXDEMOS -targetVserver UGVS_VPN_UGCTXDEMOS
add cs policy CSPOL_UGCTXDEMOS -rule "is_vpn_url  ||  HTTP.REQ.URL.PATH.SET_TEXT_MODE(IGNORECASE).STARTSWITH("/Citrix/External")" -action CSACT_UGCTXDEMOS

# Create Content Switching vServer for Citrix Gateway
add cs vserver CSVS_UGCTXDEMOS SSL 22.22.44.50 443 -cltTimeout 180
set ssl vserver CSVS_UGCTXDEMOS -ssl3 DISABLED -tls1 DISABLED -tls11 DISABLED -tls13 ENABLED -ocspStapling ENABLED -HSTS ENABLED -maxage 157680000 -IncludeSubdomains YES
bind ssl vserver CSVS_UGCTXDEMOS -certkeyName CTXDEMOS_PUBLIC_CERT
bind ssl vserver CSVS_UGCTXDEMOS -cipherName CTXDEMOS_FRONTEND_APLUS
bind cs vserver CSVS_UGCTXDEMOS -policyName CSPOL_UGCTXDEMOS -priority 63000

# Create Responder Policy and Action for HTTP to HTTPS Redirection
add responder action RESACT_HTTP_TO_HTTPS redirect ""https://" + HTTP.REQ.HOSTNAME.HTTP_URL_SAFE + HTTP.REQ.URL.PATH_AND_QUERY.HTTP_URL_SAFE" -responseStatusCode 301
add responder policy RESPOL_HTTP_TO_HTTPS HTTP.REQ.IS_VALID RESACT_HTTP_TO_HTTPS

# Create Always On Server and Service
add server LBSRV_ALWAYS_UP 127.0.0.1
add service LBSVC_ALWAYS_UP LBSRV_ALWAYS_UP HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip ENABLED cip-header -usip YES -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

# Create Always On vServer for Citrix Gateway
add lb vserver CSVS_UGCTXDEMOS_REDIRECT_HTTP_TO_HTTPS HTTP 22.22.44.50 80 -persistenceType NONE -cltTimeout 180
bind lb vserver CSVS_UGCTXDEMOS_REDIRECT_HTTP_TO_HTTPS LBSVC_ALWAYS_UP
bind lb vserver CSVS_UGCTXDEMOS_REDIRECT_HTTP_TO_HTTPS -policyName RESPOL_HTTP_TO_HTTPS -priority 100 -gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

配置第一台身份验证服务器

```
# Create Initialization SAML SP Policy and Action and Bind it to Citrix ADC AAA Authentication vServer
```



```
add authentication samlAction AUTH_ACT_SAML_SP_VPN_TO_LB -samlIdPCertName CTXDEMOS_PUBLIC_CERT -samlSigningCertName CTXDEMOS_PUBLIC_CERT -samlRedirectUrl "https://access.ctxdemos.com/samltolb" -signatureAlg RSA-SHA256 -digestMethod SHA256 -samlBinding REDIRECT -groupNameField Groups
add authentication Policy AUTH_POL_SAMP_SP_VPN_TO_LB -rule TRUE -action AUTH_ACT_SAML_SP_VPN_TO_LB

# Create Authentication Policy and Action for SAML SP to ADFS
add authentication samlAction AUTH_ACT_SAML_SP_ADFS -samlIdPCertName CTXDEMOS_ADFS_TOKEN_SIGNING -samlSigningCertName CTXDEMOS_PUBLIC_CERT -samlRedirectUrl "https://sts.ctxdemos.com/adfs/ls/" -samlUserField "Name ID" -samlRejectUnsignedAssertion OFF -samlIssuerName "https://access.ctxdemos.com" -Attribute1 "E-Mail Address" -signatureAlg RSA-SHA256 -digestMethod SHA256 -logoutURL "https://sts.ctxdemos.com/adfs/ls/wa=wsignout1.0" -forceAuthn ON
add authentication Policy AUTH_POL_SAML_SP_ADFS -rule TRUE -action AUTH_ACT_SAML_SP_ADFS

# Create Authentication Policy Label for for SAML SP to ADFS
add authentication policylabel AUTH_POLLBL_ADFS_AZUREMFA -loginSchema LSCHEMA_INT
bind authentication policylabel AUTH_POLLBL_ADFS_AZUREMFA -policyName AUTH_POL_SAML_SP_ADFS -priority 100 -gotoPriorityExpression NEXT

# Create Authentication Policy and Action for Group Extraction
add authentication ldapAction AUTH_ACT_LDAP_GROUP_EXTRACTION_AZUREMFACA -serverIP 22.22.22.61 -serverPort 636 -ldapBase "DC=ctxdemos,DC=com" -ldapBindDn "CN=svc_ctxadc01,OU=Services,OU=Accounts,DC=ctxdemos,DC=com" -ldapBindDnPassword 0 c4fe86d56a865ef514a15affd1429f3e079ce1089731d4a407772d21036f3c8 -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -searchFilter "memberOf:1.2.840.113556.1.4.1941:=CN=AzureMFACAUsers,OU=Groups,OU=Authorizations,DC=ctxdemos,DC=com" -groupAttrName memberOf -subAttributeName cn -secType SSL -authentication DISABLED -nestedGroupExtraction ON -maxNestingLevel 5 -groupNameIdentifier sAMAccountName -groupSearchAttribute memberOf -groupSearchSubAttribute CN -Attribute1 mail -Attribute2 objectGUID
add authentication Policy AUTH_POL_LDAP_GROUP_EXTRACTION_AZURAMFACA -rule TRUE -action AUTH_ACT_LDAP_GROUP_EXTRACTION_AZUREMFACA

# Create Authentication Policy Label for Group Extraction
add authentication policylabel AUTH_POLLBL_LDAP_GROUP_EXTRACTION_AZURAMFACA -loginSchema LSCHEMA_INT
```

```
bind authentication policylabel
    AUTH_POLLBL_LDAP_GROUP_EXTRACTION_AZURAMFACA -policyName
    AUTH_POL_LDAP_GROUP_EXTRACTION_AZURAMFACA -priority 100 -
    gotoPriorityExpression NEXT -nextFactor AUTH_POLLBL_ADFS_AZUREMFA


# Create Login Schema Policy and Profile for First Citrix ADC AAA
    Authentication vServer
add authentication loginSchema LSCHEMA_PRF_NOSCHEMA -
    authenticationSchema noschema -SSOCredentials YES
add authentication loginSchemaPolicy LSCHEMA_POL_NOSCHEMA -rule TRUE -
    action LSCHEMA_PRF_NOSCHEMA

# Create First Citrix ADC AAA Authentication vServer
add authentication vserver AAAVS_CTXDEMOS_COM_FOR_VPN SSL 0.0.0.0
set ssl vserver AAAVS_CTXDEMOS_COM_FOR_VPN -ssl3 DISABLED -tls1
    DISABLED -tls11 DISABLED -tls13 ENABLED -ocspStapling ENABLED -HSTS
    ENABLED -maxage 157680000 -IncludeSubdomains YES
bind ssl vserver AAAVS_CTXDEMOS_COM_FOR_VPN -certkeyName
    CTXDEMOS_PUBLIC_CERT
bind ssl vserver AAAVS_CTXDEMOS_COM_FOR_VPN -cipherName
    CTXDEMOS_FRONTEND_APLUS
bind authentication vserver AAAVS_CTXDEMOS_COM_FOR_VPN -policy
    LSCHEMA_POL_NOSCHEMA -priority 100 -gotoPriorityExpression END
bind authentication vserver AAAVS_CTXDEMOS_COM_FOR_VPN -policy
    AUTH_POL_SAMP_SP_VPN_TO_LB -priority 100 -nextFactor
    AUTH_POLLBL_LDAP_GROUP_EXTRACTION_AZURAMFACA -gotoPriorityExpression
     NEXT

# Create First Citrix ADC AAA Authentication Profile
add authentication authnProfile AAA_AUTH_PRF_VPN -authnVsName
    AAAVS_CTXDEMOS_COM_FOR_VPN -AuthenticationHost aaa.ctxdemos.com

# Set Authentication Profile on Gateway vServer
set vpn vserver UGVS_VPN_UGCTXDEMOS -authnProfile AAA_AUTH_PRF_VPN
<!--NeedCopy-->
```

配置第二台身份验证服务器

```
# Create Authentication Policy and Action for LDAP
add authentication ldapAction AUTH_ACT_LDAP -serverIP 22.22.22.61 -
    serverPort 636 -authTimeout 60 -ldapBase ”DC=ctxdemos,DC=com” -
```



```
    ldapBindDn "CN=svc_ctxadc01,OU=Services,OU=Accounts,DC=ctxdemos,DC=
    com" -ldapBindDnPassword 273881819
    af883e70c33d83c0546eac84e81d6eeba904f2d65bbebf2819c025a -encrypted -
    encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName
     memberOf -subAttributeName cn -secType SSL -passwdChange ENABLED -
    nestedGroupExtraction ON -maxNestingLevel 5 -groupNameIdentifier
    sAMAccountName -groupSearchAttribute memberOf -
    groupSearchSubAttribute CN -Attribute1 userprincipalname -Attribute2
     mail -Attribute3 userParameters
3 add authentication Policy AUTH_POL_LDAP_USER_NAME_PASSWORD -rule TRUE -
    action AUTH_ACT_LDAP
4
5 # Create Login Schema Policy and Profile for Second Citrix ADC AAA
    Authentication vServer - Username (Pre-filled ) and Password
6 add authentication loginSchema LSCHEMA_USER_NAME_PASSWORD -
    authenticationSchema "/nsconfig/loginschema/CTXDEMOS_USER_NAME_PASS.
    xml" -SSOCredentials YES
7 add authentication loginSchemaPolicy LSCHEMA_POL_USER_NAME_PASSWORD -
    rule TRUE -action LSCHEMA_USER_NAME_PASSWORD
8
9 # Create Authentication Policy Label for LDAP Username and Password
10 add authentication policylabel AUTH_POLLBL_LDAP_USER_NAME_PASSWORD -
    loginSchema LSCHEMA_USER_NAME_PASSWORD
11 bind authentication policylabel AUTH_POLLBL_LDAP_USER_NAME_PASSWORD -
    policyName AUTH_POL_LDAP_USER_NAME_PASSWORD -priority 110 -
    gotoPriorityExpression NEXT
12
13 # Create Login Schema Policy and Profile for Second Citrix ADC AAA
    Authentication vServer - Username Only
14 add authentication loginSchema LSCHEMA_USER_NAME_ONLY -
    authenticationSchema "/nsconfig/loginschema/CTXDEMOS_USER_NAME_ONLY.
    xml"
15 add authentication loginSchemaPolicy LSCHEMA_POL_NOPASSWORD -rule TRUE
    -action LSCHEMA_USER_NAME_ONLY
16
17 # Create Citrix ADC AAA Session Policy and Profile
18 add tm sessionAction AAA_SESSION_PRF_CTXDEMOS -SSO ON -ssoDomain
    CTXDEMOS -persistentCookie ON -persistentCookieValidity 30
19 add tm sessionPolicy AAA_SESSION_POL_CTXDEMOS TRUE
    AAA_SESSION_PRF_CTXDEMOS
20
21 # Create Second Citrix ADC AAA Authentication vServer
22 add authentication vserver AAAVS_CTXDEMOS_COM SSL 22.22.44.51 443
23 set ssl vserver AAAVS_CTXDEMOS_COM -ssl3 DISABLED -tls1 DISABLED -tls11
     DISABLED -tls13 ENABLED -ocspStapling ENABLED -HSTS ENABLED -maxage
```



```
      157680000 -IncludeSubdomains YES
bind ssl vserver AAAVS_CTXDEMOS_COM -certkeyName CTXDEMOS_PUBLIC_CERT
bind ssl vserver AAAVS_CTXDEMOS_COM -cipherName CTXDEMOS_FRONTEND_APLUS
bind authentication vserver AAAVS_CTXDEMOS_COM -portaltheme
    CTXDEMOS_PORTAL
bind authentication vserver AAAVS_CTXDEMOS_COM -policy
    AAA_SESSION_POL_CTXDEMOS -priority 100 -gotoPriorityExpression NEXT
bind authentication vserver AAAVS_CTXDEMOS_COM -policy
    LSCHEMA_POL_NOPASSWORD -priority 110 -gotoPriorityExpression END
bind authentication vserver AAAVS_CTXDEMOS_COM -policy
    AUTH_POL_LDAP_GROUP_EXTRACTION_AZURAMFACA -priority 140 -nextFactor
    AUTH_POLLBL_LDAP_USER_NAME_PASSWORD -gotoPriorityExpression NEXT

# Create Second Citrix ADC AAA Authentication Profile
add authentication authnProfile AAA_AUTH_PRF -authnVsName
    AAAVS_CTXDEMOS_COM -AuthenticationHost aaa.ctxdemos.com
<!--NeedCopy-->
```

将 **Citrix ADC** 配置为 **AD FS WAP**

在 Citrix ADC CLI 中运行以下命令以将 Citrix ADC 配置为 AD FS Web 应用程序代理 (WAP)：

```
# Pattern Set - ADFS Proxy Hostname
add policy patset PATSET_ADFS_HOSTNAME
bind policy patset PATSET_ADFS_HOSTNAME sts.ctxdemos.com -index 1 -
    charset ASCII
# Policy Expression - ADFS Proxy Hostname
add policy expression is_ADFS_HOSTNAME "HTTP.REQ.HEADER("Host").
    TO_LOWER.CONTAINS_ANY("PATSET_ADFS_HOSTNAME")"

# Pattern Set - ADFS Proxy Path for NoAuth
add policy patset PATSET_ADFS_PATH_NOAUTH
bind policy patset PATSET_ADFS_PATH_NOAUTH "/adfs/services/trust" -
    index 1 -charset ASCII
bind policy patset PATSET_ADFS_PATH_NOAUTH "/federationmetadata
    /2007-06/federationmetadata.xml" -index 2 -charset ASCII
bind policy patset PATSET_ADFS_PATH_NOAUTH "/adfs/fs/
    federationserverservice.asmx" -index 3 -charset ASCII
bind policy patset PATSET_ADFS_PATH_NOAUTH "/adfs/ls/FormsSignIn.aspx"
    -index 4 -charset ASCII
bind policy patset PATSET_ADFS_PATH_NOAUTH "/adfs/services/trust/2005/
    usernamemixed" -index 5 -charset ASCII
```



```
14 bind policy patset PATSET_ADFS_PATH_NOAUTH "/adfs/services/trust/mex" -
       index 6 -charset ASCII
15
16 # Policy Expression - ADFS Proxy Path for NoAuth
17 add policy expression is_ADFS_PROXY_NOAUTH "HTTP.REQ.URL.PATH.TO_LOWER.
       CONTAINS_ANY("PATSET_ADFS_PATH_NOAUTH")"
18
19 # Pattern Set - ADFS Proxy Path for Passive Client
20 add policy patset PATSET_ADFS_PATH_ACTIVE_PASSIVE
21 bind policy patset PATSET_ADFS_PATH_ACTIVE_PASSIVE "/adfs" -index 1 -
       charset ASCII
22 bind policy patset PATSET_ADFS_PATH_ACTIVE_PASSIVE "/cgi/selfauth" -
       index 2 -charset ASCII
23
24 # Policy Expression - ADFS Proxy Path for Passive Client
25 add policy expression is_ADFS_PROXY_ACTIVE_PASSIVE "(HTTP.REQ.HEADER("
       Host").TO_LOWER.CONTAINS_ANY("PATSET_ADFS_HOSTNAME") && HTTP.REQ.URL
       .PATH.TO_LOWER.STARTSWITH_ANY("PATSET_ADFS_PATH_ACTIVE_PASSIVE"))"
26
27 # Rewrite Policies for ADFS PIP
28 add rewrite action RWACT_X_MS_Proxy insert_http_header X-MS-Proxy ""
       NETSCALER""
29 add rewrite policy RWPOL_X_MS_Proxy true RWACT_X_MS_Proxy
30
31 add rewrite action RWACT_X_MS_Forwarded_Client_IP insert_http_header X-
       MS-Forwarded-Client-IP CLIENT.IP.SRC
32 add rewrite policy RWPOL_X_MS_Forwarded_Client_IP true
       RWACT_X_MS_Forwarded_Client_IP
33
34 add rewrite action RWACT_X_MS_Endpoint_Absolute_Path insert_http_header
        X-MS-Endpoint-Absolute-Path HTTP.REQ.URL
35 add rewrite policy RWPOL_X_MS_Endpoint_Absolute_Path true
       RWACT_X_MS_Endpoint_Absolute_Path
36
37 add rewrite action RWACT_X_MS_Target_Role insert_http_header X-MS-
       Target-Role ""PrimaryComputer""
38 add rewrite policy RWPOL_X_MS_Target_Role true RWACT_X_MS_Target_Role
39
40 add rewrite action RWACT_X_MS_ADFS_Proxy_Client_IP insert_http_header X
       -MS-ADFS-Proxy-Client-IP CLIENT.IP.SRC
41 add rewrite policy RWPOL_X_MS_ADFS_Proxy_Client_IP true
       RWACT_X_MS_ADFS_Proxy_Client_IP
42
43 add rewrite action RWACT_X_MS_Client_User_Agent insert_http_header X-MS
       -Client-User-Agent "HTTP.REQ.HEADER("User-Agent")"
```



```
add rewrite policy RWPOL_X_MS_Client_User_Agent true
    RWACT_X_MS_Client_User_Agent

add rewrite action RWACT_ADFS_PROXYMEX replace HTTP.REQ.URL.
    PATH_AND_QUERY ""/adfs/services/trust/proxymex" + HTTP.REQ.URL.
    SET_TEXT_MODE(IGNORECASE).PATH_AND_QUERY.STRIP_START_CHARS("/adfs/
    services/trust/mex").HTTP_URL_SAFE"
add rewrite policy RWPOL_ADFS_PROXYMEX "is_ADFS_HOSTNAME && HTTP.REQ.
    URL.TO_LOWER.STARTSWITH("/adfs/services/trust/mex")"
    RWACT_ADFS_PROXYMEX

add rewrite policy RWPOL_ADFS_PROXY_HEADERS-NOACT TRUE NOREWRITE

add rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS http_req
bind rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS RWPOL_X_MS_Proxy
    100 NEXT
bind rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS
    RWPOL_X_MS_Forwarded_Client_IP 110 NEXT
bind rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS
    RWPOL_X_MS_Endpoint_Absolute_Path 120 NEXT
bind rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS
    RWPOL_X_MS_Target_Role 130 NEXT
bind rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS
    RWPOL_X_MS_ADFS_Proxy_Client_IP 140 NEXT
bind rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS
    RWPOL_X_MS_Client_User_Agent 150 NEXT
bind rewrite policylabel RWPOLLBL_ADFS_PROXY_HEADERS
    RWPOL_ADFS_PROXYMEX 160 NEXT

# Create ADFS Server and Service Group
add server LBSRV_ADFS wsadfs01.ctxdemos.com
add serviceGroup LBSVCGRP_ADFS_443 SSL -maxClient 0 -maxReq 0 -cip
    ENABLED X-MS-Forwarded-Client-IP -usip NO -useproxyport YES -sp ON -
    cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES
bind ssl serviceGroup LBSVCGRP_ADFS_443 -cipherName CTXDEMO_BACKEND
set ssl serviceGroup LBSVCGRP_ADFS_443 -ssl3 DISABLED -tls1 DISABLED -
    tls11 DISABLED
bind serviceGroup LBSVCGRP_ADFS_443 LBSRV_ADFS 443

# Create ADFS Proxy NoAuth Load Balancing vServer
add lb vserver LBVS_ADFS_PROXY_NOAUTH SSL 0.0.0.0 0 -persistenceType
    NONE -cltTimeout 180
set ssl vserver LBVS_ADFS_PROXY_NOAUTH -ssl3 DISABLED -tls1 DISABLED -
    tls11 DISABLED -tls13 ENABLED -ocspStapling ENABLED -HSTS ENABLED -
    maxage 157680000 -IncludeSubdomains YES
```

```
bind ssl vserver LBVS_ADFS_PROXY_NOAUTH -certkeyName CTXDEMOS-PUBLIC
bind ssl vserver LBVS_ADFS_PROXY_NOAUTH -cipherName CTXDEMO_BACKEND
bind lb vserver LBVS_ADFS_PROXY_NOAUTH LBSVCGRP_ADFS_443
bind lb vserver LBVS_ADFS_PROXY_NOAUTH -policyName
    RWPOL_ADFS_PROXY_HEADERS-NOACT -priority 100 -gotoPriorityExpression
     NEXT -type REQUEST -invoke policylabel RWPOLLBL_ADFS_PROXY_HEADERS

# Create ADFS Proxy NoAuth Content Switching Policy and Action
add cs action CSACT_ADFS_PROXY_NOAUTH -targetLBVserver
    LBVS_ADFS_PROXY_NOAUTH
add cs policy CSPOL_ADFS_PROXY_NOAUTH -rule is_ADFS_PROXY_NOAUTH -
    action CSACT_ADFS_PROXY_NOAUTH

# Create ADFS Proxy Active-Passive Load Balancing vServer
add lb vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE SSL 0.0.0.0 0 -
    persistenceType NONE -cltTimeout 180 -Authentication ON -
    authnProfile AAA_AUTH_PRF
set ssl vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE -ssl3 DISABLED -tls1
    DISABLED -tls11 DISABLED -tls13 ENABLED -ocspStapling ENABLED -HSTS
    ENABLED -maxage 157680000 -IncludeSubdomains YES
bind lb vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE LBSVCGRP_ADFS_443
bind ssl vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE -certkeyName CTXDEMOS-
    PUBLIC
bind ssl vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE -cipherName
    CTXDEMO_FRONTEND_APLUS
bind lb vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE -policyName
    RWPOL_ADFS_PROXY_HEADERS-NOACT -priority 100 -gotoPriorityExpression
     NEXT -type REQUEST -invoke policylabel RWPOLLBL_ADFS_PROXY_HEADERS

# Create ADFS Proxy Active-Passive Content Switching Policy and Action
add cs action CSACT_ADFS_PROXY_ACTIVE_PASSIVE -targetLBVserver
    LBVS_ADFS_PROXY_ACTIVE_PASSIVE
add cs policy CSPOL_ADFS_PROXY_ACTIVE_PASSIVE -rule
    is_ADFS_PROXY_ACTIVE_PASSIVE -action CSACT_ADFS_PROXY_ACTIVE_PASSIVE

# Bind Content Switching Policies to Citrix Gateway Content Switching
    vServer
bind cs vserver CSVS_UGCTXDEMOS -policyName CSPOL_ADFS_PROXY_NOAUTH -
    priority 100
bind cs vserver CSVS_UGCTXDEMOS -policyName
    CSPOL_ADFS_PROXY_ACTIVE_PASSIVE -priority 300

# Create Citrix ADC AAA Traffic Policies and Bind them to ADFS Proxy
    Active-Passive Load Balancing vServer
add tm formSSOAction AAATM_SSOPRF_ADFS_LOGIN -actionURL "/adfs/ls" -
```



```
    userField UserName -passwdField Password -ssoSuccessRule true -
    nameValuePair AuthMethod=FormsAuthentication -responsesize 15000 -
    submitMethod POST
add tm trafficAction AAATM_PRF_ADFS_LOGIN -appTimeout 1 -SSO ON -
    formSSOAction AAATM_SSOPRF_ADFS_LOGIN -persistentCookie OFF -
    InitiateLogout OFF -kcdAccount NONE -userExpression "HTTP.REQ.USER.
    ATTRIBUTE(3)" -passwdExpression "HTTP.REQ.USER.ATTRIBUTE(2)"
add tm trafficPolicy AAATM_POL_ADFS_LOGIN "HTTP.REQ.URL.TO_LOWER.
    STARTSWITH("/adfs/ls")" AAATM_PRF_ADFS_LOGIN
add tm trafficAction AAATM_PRF_ADFS_LOGOUT -appTimeout 1 -
    persistentCookie OFF -InitiateLogout ON -kcdAccount NONE
add tm trafficPolicy AAATM_POL_ADFS_LOGOUT "HTTP.REQ.URL.TO_LOWER.
    STARTSWITH("/adfs/ls") && HTTP.REQ.URL.QUERY.VALUE("wa").EQ("
    wsignout1.0")" AAATM_PRF_ADFS_LOGOUT
bind lb vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE -policyName
    AAATM_POL_ADFS_LOGIN -priority 100 -gotoPriorityExpression END -type
     REQUEST
bind lb vserver LBVS_ADFS_PROXY_ACTIVE_PASSIVE -policyName
    AAATM_POL_ADFS_LOGOUT -priority 110 -gotoPriorityExpression END -
    type REQUEST
<!--NeedCopy-->
```

配置初始身份验证流程

```
# Pattern Set - Gateway and AAA Hostname
add policy patset PATSET_GATEWAY_HOSTHEADER
bind policy patset PATSET_GATEWAY_HOSTHEADER access.ctxdemos.com -index
     1 -charset ASCII
bind policy patset PATSET_GATEWAY_HOSTHEADER aaa.ctxdemos.com -index 2
    -charset ASCII
# Policy Expression - Gateway and AAA Hostname
add policy expression is_GATEWAY_HOSTNAME "HTTP.REQ.HEADER("Host").
    TO_LOWER.CONTAINS_ANY("PATSET_GATEWAY_HOSTHEADER")"

# Create Initialization Load Balancing vServer
add lb vserver LBVS_SAML_SP_INITIALIZATION SSL 0.0.0.0 0 -
    persistenceType NONE -cltTimeout 180 -Authentication ON -
    authnProfile AAA_AUTH_PRF
set ssl vserver LBVS_SAML_SP_INITIALIZATION -ssl3 DISABLED -tls1
    DISABLED -tls11 DISABLED -tls13 ENABLED -ocspStapling ENABLED -HSTS
    ENABLED -maxage 157680000 -IncludeSubdomains YES
bind lb vserver LBVS_SAML_SP_INITIALIZATION LBSVC_ALWAYS_UP
```



```
bind ssl vserver LBVS_SAML_SP_INITIALIZATION -certkeyName
    CTXDEMOS_PUBLIC_CERT
bind ssl vserver LBVS_SAML_SP_INITIALIZATION -cipherName
    CTXDEMOS_FRONTEND_APLUS

# Create Initialization Content Switching Policy and Action
add cs action CSACT_SAML_SP_INITIALIZATION -targetLBVserver
    LBVS_SAML_SP_INITIALIZATION
add cs policy CSPOL_SAML_SP_INITIALIZATION -rule "is_GATEWAY_HOSTNAME
    && HTTP.REQ.URL.PATH.TO_LOWER.STARTSWITH("/samltolb")" -action
    CSACT_SAML_SP_INITIALIZATION

# Bind Content Switching Policies to Citrix Gateway Content Switching
    vServer
bind cs vserver CSVS_UGCTXDEMOS -policyName
    CSPOL_SAML_SP_INITIALIZATION -priority 500

# Create Initialization Citrix ADC AAA Traffic Policy and Action and
    Bind it to Load Balancing vServer
add tm samlSSOProfile AAATM_SAMLSSOPRF_VPN_TO_LB -samlSigningCertName
    CTXDEMOS_PUBLIC_CERT -assertionConsumerServiceURL "https://access.
    ctxdemos.com/cgi/samlauth" -relaystateRule "HTTP.REQ.URL.QUERY.VALUE
    ("RelayState")" -signatureAlg RSA-SHA256 -digestMethod SHA256 -
    Attribute1 Password -Attribute1Expr AAA.USER.PASSWD -Attribute2
    Groups -Attribute2Expr AAA.USER.GROUPS -encryptAssertion ON -
    samlSPCertName CTXDEMOS_PUBLIC_CERT
add tm trafficAction AAATM_PRF_VPN_TO_LB -SSO ON -persistentCookie OFF
    -InitiateLogout OFF -kcdAccount NONE -samlSSOProfile
    AAATM_SAMLSSOPRF_VPN_TO_LB
add tm trafficPolicy AAATM_POL_VPN_TO_LB "HTTP.REQ.URL.STARTSWITH("/
    samltolb")" AAATM_PRF_VPN_TO_LB
bind lb vserver LBVS_SAML_SP_INITIALIZATION -policyName
    AAATM_POL_VPN_TO_LB -priority 100 -gotoPriorityExpression END -type
    REQUEST
<!--NeedCopy-->
```

密码组

```
# Create Cipher Group for Backend vServers
add ssl cipher CTXDEMOS_BACKEND
bind ssl cipher CTXDEMOS_BACKEND -cipherName TLS1.3-AES256-GCM-SHA384 -
    cipherPriority 1
```



```
bind ssl cipher CTXDEMOS_BACKEND -cipherName TLS1.3-CHACHA20-POLY1305-SHA256 -cipherPriority 2
bind ssl cipher CTXDEMOS_BACKEND -cipherName TLS1.3-AES128-GCM-SHA256 -cipherPriority 3
bind ssl cipher CTXDEMOS_BACKEND -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 -cipherPriority 4
bind ssl cipher CTXDEMOS_BACKEND -cipherName TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 -cipherPriority 5
bind ssl cipher CTXDEMOS_BACKEND -cipherName TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384 -cipherPriority 6
bind ssl cipher CTXDEMOS_BACKEND -cipherName TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256 -cipherPriority 7

# Create Cipher Group for Frondend vServers
add ssl cipher CTXDEMOS_FRONTEND
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.3-AES256-GCM-SHA384 -cipherPriority 1
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.3-CHACHA20-POLY1305-SHA256 -cipherPriority 2
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.3-AES128-GCM-SHA256 -cipherPriority 3
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256 -cipherPriority 4
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384 -cipherPriority 5
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-ECDSA-AES128-SHA256 -cipherPriority 6
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-ECDSA-AES256-SHA384 -cipherPriority 7
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1-ECDHE-ECDSA-AES128-SHA -cipherPriority 8
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1-ECDHE-ECDSA-AES256-SHA -cipherPriority 9
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 -cipherPriority 10
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 -cipherPriority 11
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-RSA-AES-128-SHA256 -cipherPriority 12
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-ECDHE-RSA-AES-256-SHA384 -cipherPriority 13
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1-ECDHE-RSA-AES128-SHA -cipherPriority 15
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1-ECDHE-RSA-AES256-SHA -cipherPriority 16
```



```
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-DHE-RSA-AES128-GCM-SHA256 -cipherPriority 17
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1.2-DHE-RSA-AES256-GCM-SHA384 -cipherPriority 18
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1-DHE-RSA-AES-128-CBC-SHA -cipherPriority 19
bind ssl cipher CTXDEMOS_FRONTEND -cipherName TLS1-DHE-RSA-AES-256-CBC-SHA -cipherPriority 20

# Create Cipher Group for Frondend vServers - A+
add ssl cipher CTXDEMOS_FRONTEND_APLUS
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.3-AES256-GCM-SHA384 -cipherPriority 1
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.3-CHACHA20-POLY1305-SHA256 -cipherPriority 2
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.3-AES128-GCM-SHA256 -cipherPriority 3
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384 -cipherPriority 4
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256 -cipherPriority 5
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305 -cipherPriority 6
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-ECDSA-AES256-SHA384 -cipherPriority 7
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-ECDSA-AES128-SHA256 -cipherPriority 8
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 -cipherPriority 9
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 -cipherPriority 13
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-RSA-CHACHA20-POLY1305 -cipherPriority 14
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-RSA-AES-256-SHA384 -cipherPriority 15
bind ssl cipher CTXDEMOS_FRONTEND_APLUS -cipherName TLS1.2-ECDHE-RSA-AES-128-SHA256 -cipherPriority 16
<!--NeedCopy-->
```

登录架构 **XML** 文件

CTXDEMOS_USER_NAME_PASS.XML


```
<?xml version=”1.0” encoding=”utf-8”?>
<AuthenticateResponse xmlns=”http://citrix.com/authentication/response/1”>
    <Status>success</Status>
    <Result>more-info</Result>
    <StateContext/>
    <AuthenticationRequirements>
        <PostBack>/nf/auth/doAuthentication.do</PostBack>
        <CancelPostBack>/Citrix/Authentication/ExplicitForms/CancelAuthenticate</CancelPostBack>
        <CancelButtonText>Cancel</CancelButtonText>
        <Requirements>
            <Requirement>
                <Credential>
                    <ID>login</ID>
                    <SaveID>ExplicitForms-Username</SaveID>
                    <Type>username</Type>
                </Credential>
                <Label>
                    <Text>User name</Text>
                    <Type>plain</Type>
                </Label>
                <Input>
                    <AssistiveText>Please supply username</AssistiveText>
                    <Text>
                        <Secret>false</Secret>
                        <ReadOnly>false</ReadOnly>
                        <InitialValue>${
AAA.USER.NAME }
</InitialValue>
                        <Constraint>.+</Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
                    <ID>passwd</ID>
                    <SaveID>ExplicitForms-Password</SaveID>
                    <Type>password</Type>
                </Credential>
                <Label>
                    <Text>Password:</Text>
                    <Type>plain</Type>
```



```
                </Label>
                <Input>
                    <Text>
                        <Secret>true</Secret>
                        <ReadOnly>false</ReadOnly>
                        <InitialValue/>
                        <Constraint>.+</Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
                    <ID>saveCredentials</ID>
                    <Type>savecredentials</Type>
                </Credential>
                <Label>
                    <Text>Remember my password</Text>
                    <Type>plain</Type>
                </Label>
                <Input>
                    <CheckBox>
                        <InitialValue>false</InitialValue>
                    </CheckBox>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
                    <ID>loginBtn</ID>
                    <Type>none</Type>
                </Credential>
                <Label>
                    <Type>none</Type>
                </Label>
                <Input>
                    <Button>Log On</Button>
                </Input>
            </Requirement>
        </Requirements>
    </AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

CTXDEMOS_USER_NAME_ONLY.XML


```
<?xml version=”1.0” encoding=”utf-8”?>
<AuthenticateResponse xmlns=”http://citrix.com/authentication/response/1”>
    <Status>success</Status>
    <Result>more-info</Result>
    <StateContext/>
    <AuthenticationRequirements>
        <PostBack>/nf/auth/doAuthentication.do</PostBack>
        <CancelPostBack>/Citrix/Authentication/ExplicitForms/CancelAuthenticate</CancelPostBack>
        <CancelButtonText>Cancel</CancelButtonText>
        <Requirements>
            <Requirement>
                <Credential>
                    <ID>login</ID>
                    <SaveID>ExplicitForms-Username</SaveID>
                    <Type>username</Type>
                </Credential>
                <Label>
                    <Text>User name</Text>
                    <Type>plain</Type>
                </Label>
                <Input>
                    <AssistiveText>Please supply username</AssistiveText>
                    <Text>
                        <Secret>false</Secret>
                        <ReadOnly>false</ReadOnly>
                        <InitialValue/>
                        <Constraint>.+</Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
                    <Type>none</Type>
                </Credential>
                <Label>
                    <Text> Please submit credentials to continue Login ...</Text>
                    <Type>confirmation</Type>
                </Label>
                <Input/>
            </Requirement>
```



```
            <Requirement>
                <Credential>
                    <ID>saveCredentials</ID>
                    <Type>savecredentials</Type>
                </Credential>
                <Label>
                    <Text>Remember my password</Text>
                    <Type>plain</Type>
                </Label>
                <Input>
                    <CheckBox>
                        <InitialValue>false</InitialValue>
                    </CheckBox>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
                    <ID>loginBtn</ID>
                    <Type>none</Type>
                </Credential>
                <Label>
                    <Type>none</Type>
                </Label>
                <Input>
                    <Button>Log On</Button>
                </Input>
            </Requirement>
        </Requirements>
    </AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

引用

在 Azure 云中的 Server 2016、Citrix FAS 和 Azure MFA 上使用 AD FS 4.0 对 NetScaler 进行身份验证。(2018)。从 https://www.jgspiers.com/authentication-to-netscaler-using-ad-fs-4-0-server-2016-citrix-fas-azure-mfa-azure-cloud/ 中检索

将 Azure MFA 配置为使用 AD FS 的身份验证提供程序。(2019)。从 https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/operations/configure-ad-fs-and-azure-mfa 中检索

部署联合身份验证服务器场。(2017)。从 https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/deploying-a-federation-server-farm 中检索

联合身份验证服务 ADFS 部署。(2018)。从 https://docs.citrix.com/zh-cn/citrix-virtual-apps-desktops/secure/federated-authentication-service/fas-architectures/fas-adfs.html 中检索

将 NetScaler 部署为 Active Directory 联合身份验证服务代理的指南。(n.d.)。从 https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/guide-to-deploying-netscaler-as-an-active-directory-federation-services-proxy.pdf 中检索

工作原理：Azure 多重身份验证。(2018)。从 https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-mfa-howitworks 中检索

规划基于云的 Azure 多重身份验证部署。(2019)。从 https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-getstarted 中检索

Tijl Van den Broeck。(Dec 7, 2017)。在 Windows Server 2012 R2 上使用 NetScaler 的 ADFS v3。从 https://www.citrix.com/blogs/2015/05/29/adfs-v3-on-windows-server-2012-r2-with-netscaler/ 中检索

使用 Citrix Gateway 过渡到混合云和 SaaS。(n.d.)。从 https://www.citrix.com/products/citrix-gateway/resources/netscaler-unified-gateway.html 中检索

用户使用 Azure Active Directory 直通身份验证登录。(2018)。从 https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-pta 中检索

由首席网络销售工程师 Saman Salehian 撰写。

利用本地主机缓存进行无中断数据库升级

May 20, 2020

本地主机缓存 (LHC) 功能允许在发生中断时，XenApp 或 XenDesktop 站点中的连接代理操作能够继续。以下过程演示了如何在没有辅助区域时使用 LHC 执行站点的无中断升级。如果 Citrix 希望扩展本指南，以包含具有多个区域的环境的过程，请返回以获取将来的更新。

在继续之前，建议先查看本地主机缓存功能及其要求和限制：https://docs.citrix.com/zh-cn/xenapp-and-xendesktop/7-15-ltsr/manage-deployment/local-host-cache.html

还有一个关于本地主机缓存大小调整和扩展的高级概念指南，可以在这里找到：https://docs.citrix.com/zh-cn/advanced-concepts/implementation-guides/local-host-cache-sizing-scaling.html

免责声明：在实时生产环境中实施这些步骤之前，在测试环境中执行这些步骤，以确保您熟悉该流程，并为可能出现的任何特定环境的问题或问题做好准备。还建议使用最新的 LTSR 累积更新 (CU)，因为有几个与 LHC 相关的修补程序可以使您的环境受益。

概述

1. 为此过程配置环境。

2. 确定选定的主要经纪商。
3. 强制停机以触发本地主机缓存功能。
4. 允许 VDA 向选定的辅助代理重新注册。
5. 在未选择的辅助代理上执行产品升级。
6. 执行强制性站点升级，包括数据库升级。
7. 对任何剩余的未选次级经纪商执行产品升级。
8. 退出中断和本地主机缓存模式。
9. 允许 VDA 向新升级的 Delivery Controller 重新注册。
10. 在最后剩余的 Delivery Controller（以前选出的二级代理）上执行产品升级。
11. 将环境返回到默认配置。

过程

1. 检查以确定是否使用以下 PowerShell 命令启用了本地主机缓存。

   ```
   Get-BrokerSite
   ```

   查找 `LocalHostCacheEnabled : True`

   ```
   PS C:\Program Files\Citrix\Desktop Studio> Get-BrokerSite


   BaseOU                                        :
   BrokerServiceGroupUid                         :
   ColorDepth                                    : TwentyFourBit
   ConfigLastChangeTime                          :
   ConfigurationServiceGroupUid                  :
   ConnectionLeasingEnabled                      : False
   DefaultMinimumFunctionalLevel                 : L7_9
   DesktopGroupIconUid                           : 1
   DnsResolutionEnabled                          : False
   IsSecondaryBroker                             : False
   LicenseEdition                                : PLT
   LicenseGraceSessionsRemaining                 :
   LicenseModel                                  : Concurrent
   LicenseServerName                             :
   LicenseServerPort                             : 27000
   LicensedSessionsActive                        : 0
   LicensingBurnIn                               :
   LicensingBurnInDate                           :
   LicensingGraceHoursLeft                       :
   LicensingGracePeriodActive                    : False
   LicensingOutOfBoxGracePeriodActive            : False
   LocalHostCacheEnabled                         : True
   MetadataMap                                   : {}
   Name                                          :
   PeakConcurrentLicenseUsers                    : 0
   ReuseMachinesWithoutShutdownInOutageAllowed   : True
   SecureIcaRequired                             : False
   TotalUniqueLicenseUsers                       : 0
   TrustManagedAnonymousXmlServiceRequests       : False
   TrustRequestsSentToTheXmlServicePort          : False
   ```

   如果为 false，请启用本地主机缓存。

   ```
   Set-BrokerSite -LocalHostCacheEnabled $true -ConnectionLeasingEnabled
   $false
   ```

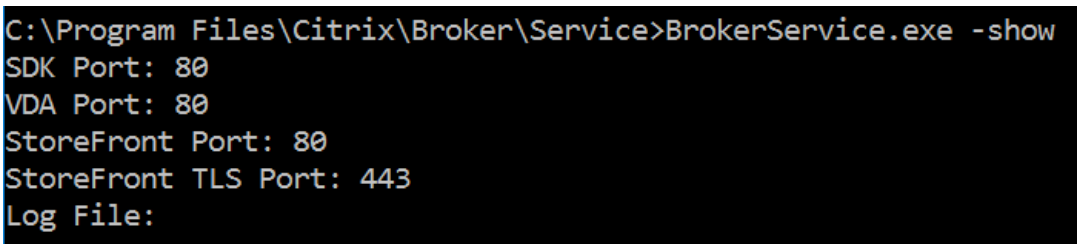

此 cmdlet 还禁用连接租用功能。请勿启用本地主机缓存和连接租用。

2. 默认情况下，启用了 ShutdownDesktopsAfterUse 属性的池交付组中的电源管理的桌面 VDA 会在发生中断时置于维护模式。要覆盖默认行为，必须在站点范围内并针对受影响的每个交付组启用它。运行以下 PowerShell cmdlet。

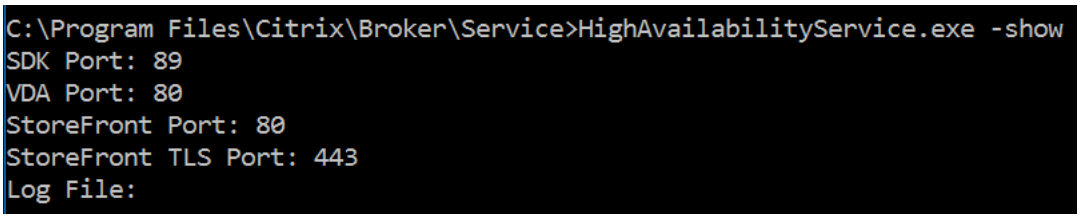
```
Set-BrokerSite -ReuseMachinesWithoutShutdownInOutageAllowed $true

Set-BrokerDesktopGroup -Name ”<Delivery Group Name>”- ReuseMachinesWithoutShutdow
 $true
```

3. 如果 Broker Service 已配置为使用自定义 VDA、StoreFront 或 StoreFront TLS 端口，请执行以下步骤以确保高可用性 (HA) 服务也配置了正确的自定义端口。

   - 通过发出以下命令验证每个 Broker 上的当前 Broker Service 端口设置：%programfiles%\Citrix\Broker\Service\BrokerService.exe -show

   - 通过发出以下命令验证每个 Broker 上的当前 HA 服务端口设置：%programfiles%\Citrix\Broker\Service\HighAvailabilityService.exe -show

   - 如果为 HA 服务列出的 VDA、StoreFront 或 StoreFront TLS 端口与 Broker Service 不匹配，请使用下面列出的相应命令行交换机将 HA 服务端口设置设置为相应匹配。

```
1 %programfiles%\Citrix\Broker\Service\HighAvailabilityService.exe -
    VdaPort <port>
2 %programfiles%\Citrix\Broker\Service\HighAvailabilityService.exe -
    StoreFrontPort <port>
3 %programfiles%\Citrix\Broker\Service\HighAvailabilityService.exe -
    StoreFrontTlsPort <port>
4 <!--NeedCopy-->
```



```
C:\Program Files\Citrix\Broker\Service>HighAvailabilityService.exe -VdaPort 80
Stopping service: CitrixHighAvailabilityService
Starting service: CitrixHighAvailabilityService
Command completed successfully
```

> 注意：
> Broker Service 和 HA 服务之间的 SDK 端口预计不同。
>
> 更改 Broker Service 的 StoreFront 端口时，高可用性服务的 StoreFront 端口将自动更新以匹配。但是，接收自动更新的服务仍然需要手动重新启动才能开始使用新端口。

4. 在中断期间，选定的辅助代理将处理所有连接。中断开始时，辅助 Broker 没有当前 VDA 注册数据，但当 VDA 与其通信时，就会立即触发重新注册过程。在该过程中，辅助 Broker 还获取有关该 VDA 的当前会话信息。要将 VDA 的重新注册速度从默认的 5 分钟间隔加快到 1 分钟间隔，需要将此设置应用于站点中的所有 Controller。

   ```
   New-ItemProperty -Path HKLM:\SOFTWARE\Citrix\DesktopServer -Name
   HeartbeatPeriodMs -PropertyType DWORD -Value 60000
   ```

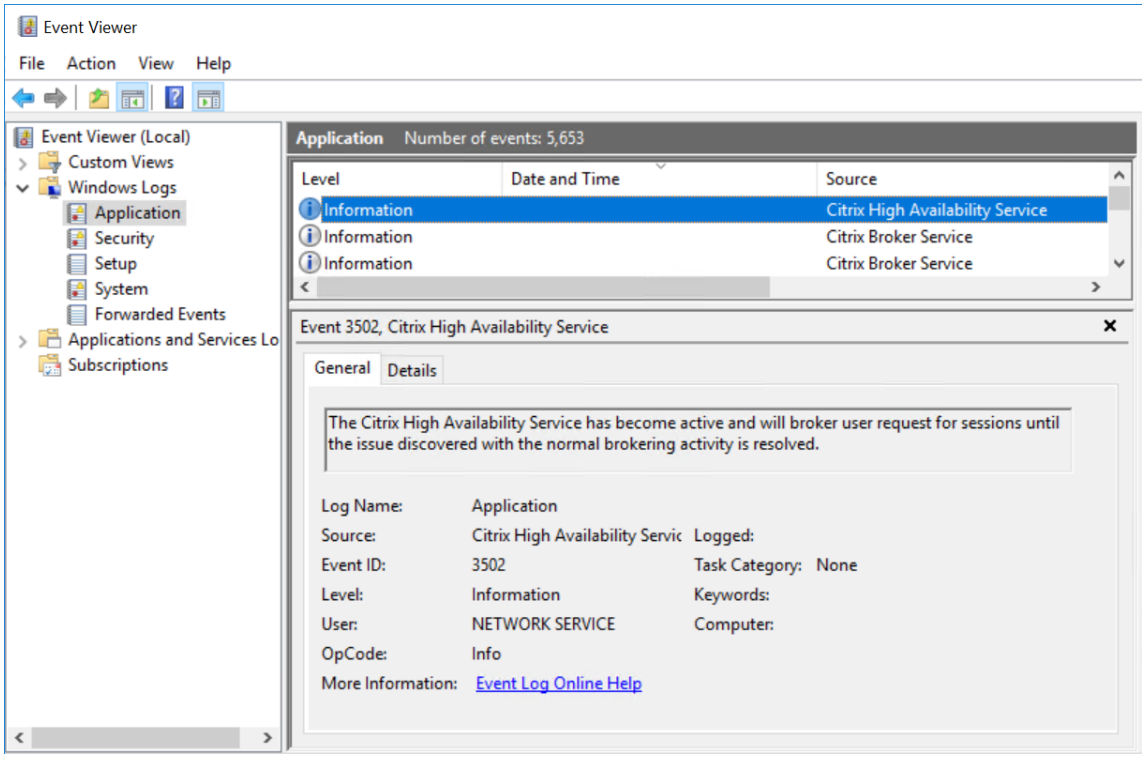
5. 要监视 VDA 重新注册，请启动 Citrix Studio，然后单击配置 **>** 控制器节点，并查看向主代理注册的 VDA 数量。保持 Citrix Studio 处于打开状态，以查看 VDA 在停机期间向选定的辅助代理重新注册时计数降至零。请注意，您不能使用 Citrix Studio 查看在辅助代理中注册的 VDA 计数。

6. 要强制中断并进入 LHC 模式，请编辑每个 Delivery Controller 的注册表。

   ```
   New-ItemProperty -Path HKLM:\SOFTWARE\Citrix\DesktopServer\LHC -Name
   OutageModeForced -PropertyType DWORD -Value 1
   ```

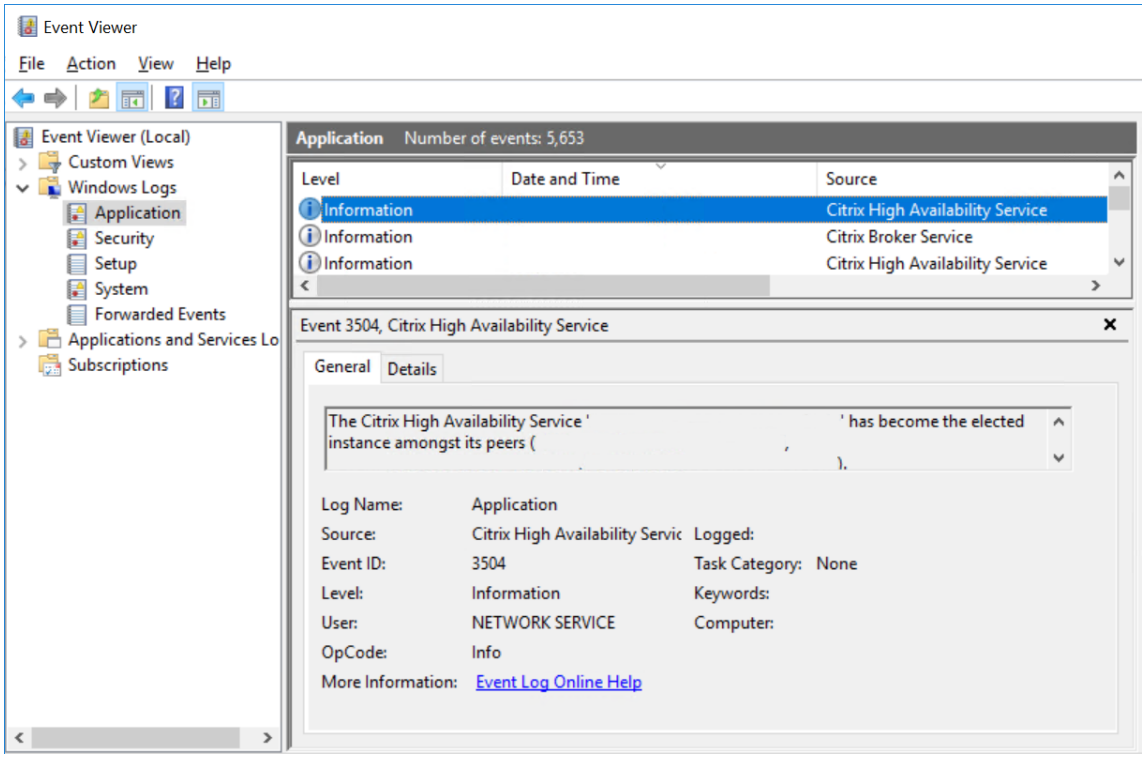
7. 要确定是否已触发中断并且每个主代理都进入 LHC 模式，请转到每个控制器上事件日志的应用程序节点，然后从 Citrix 高可用性服务中查找以下事件。

   3502：Citrix 高可用性服务已变为活动状态，并将代理用户请求会话，直到通过正常代理活动发现的问题得到解决。

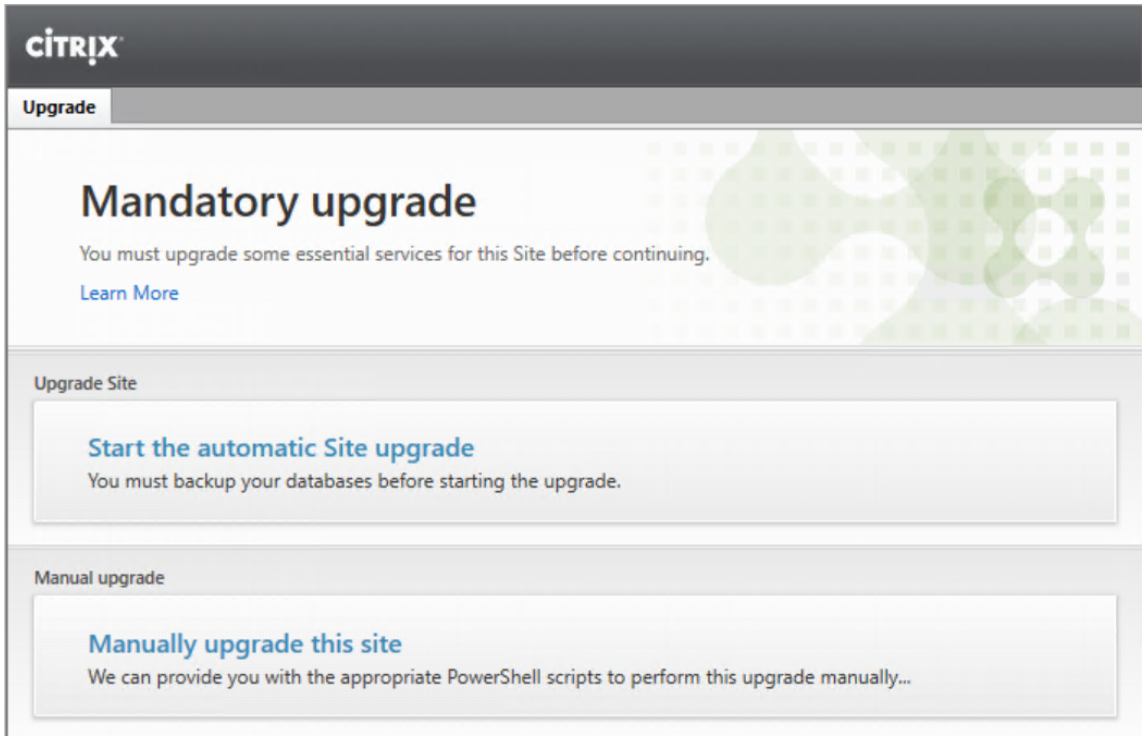


8. 通过刷新 Citrix Studio 中的“控制器”节点，确认所有 VDA 已在选定的辅助代理中重新注册。当主代理显示零注册 VDA 时，所有 VDA 都可能重新注册。
9. 二级代理商使用他们正在运行的计算机的 FQDN 的字母顺序列表来确定（选择）在发生中断时，哪个辅助代理将负责区域中的代理操作。要确认已选择哪个辅助代理，请从 Windows 事件应用程序日志中的 Citrix 高可用性服务中查找以下事件。

   3504：Citrix 高可用性服务“选定的控制器的 FQDN”已成为其对等方（对等控制器 *FQDN* 列表）中的选择实例。



10. 选择一个未选择的对等控制器，然后在未选择的控制器上执行产品升级。
11. 从新升级的控制器启动 Citrix Studio 并执行强制性站点升级，包括数据库升级。



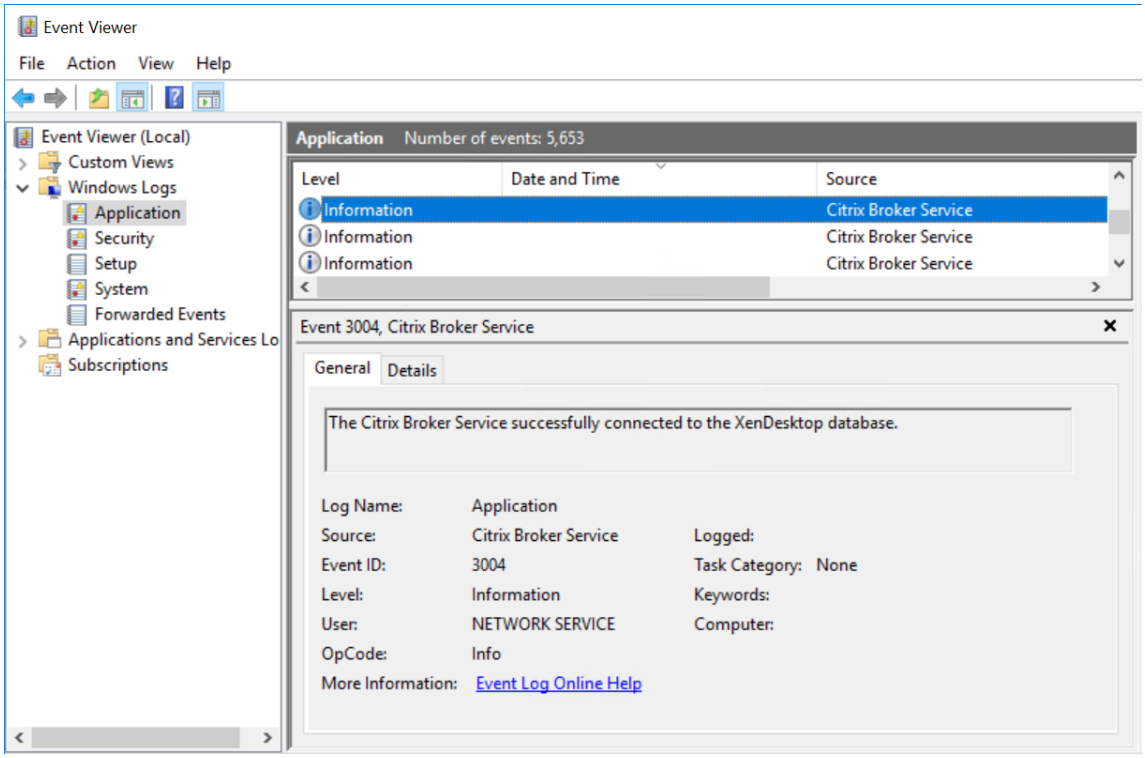
12. 对其余未选择的对等控制器执行产品升级。请确保不要中断仍在管理环境中所有新连接和活动连接的选定 Controller。

13. 升级所有未选择的控制器后，现在是时候让站点摆脱停机并退出 LHC 模式。要删除强制停机触发器，请编辑每个控制器的注册表。如果需要，也可以删除密钥。

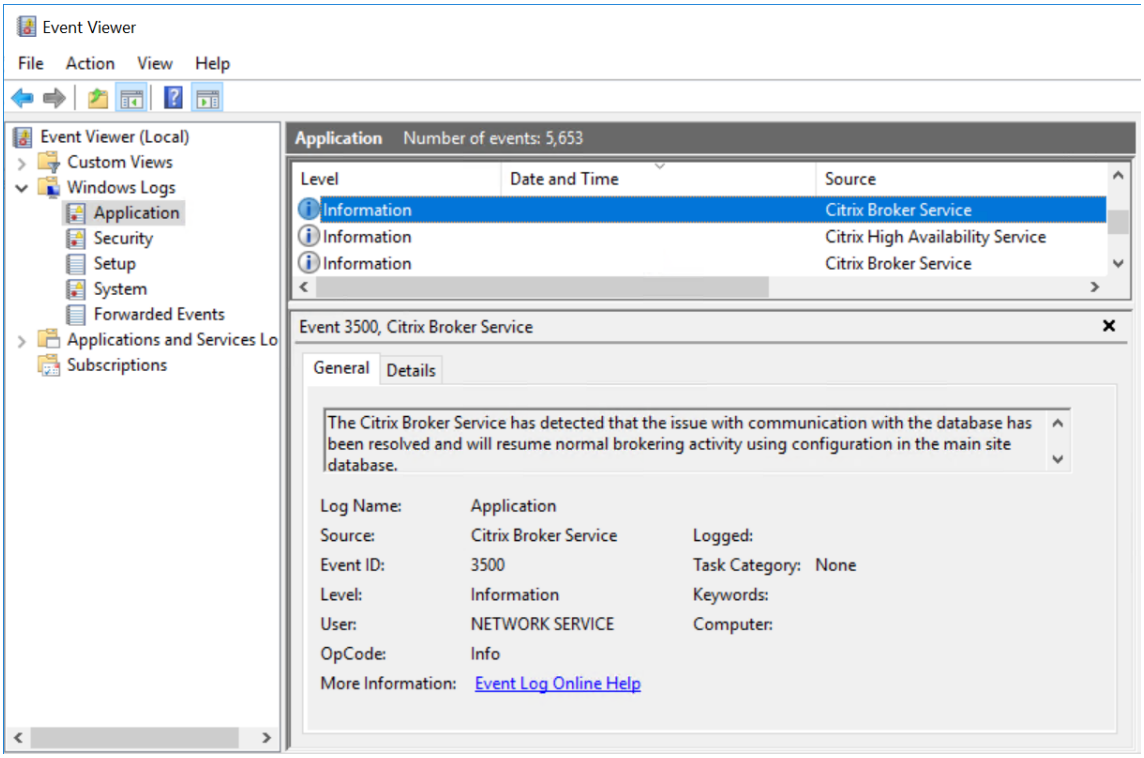
```
Set-ItemProperty -Path HKLM:\SOFTWARE\Citrix\DesktopServer\LHC -Name OutageModeForced -Value 0
```

14. 要确认站点是否处于停机模式之外，请在每个控制器上从应用程序事件日志中的 Citrix Broker Service 中查找以下事件。

    3004：Citrix Broker Service 成功连接到 XenDesktop 数据库。



    3500：Citrix Broker Service 检测到与数据库的通信问题已解决，并将使用主站点数据库中的配置恢复正常的代理活动。



15. 从 Citrix Studio 刷新“控制器”节点，以观看 VDA 向升级的控制器重新注册。确认所有 VDA 已成功重新注册。
16. 在停机期间担任选择的辅助代理的最后一个剩余控制器上执行产品升级。
17. 通过修改每个控制器上的注册表，将 VDA 注册间隔设置回默认值 5 分钟（如果首选，也可以删除该项）。

    ```
    Set-ItemProperty -Path HKLM:\SOFTWARE\Citrix\DesktopServer -Name
    HeartbeatPeriodMs -PropertyType DWORD -Value 300000
    ```

18. 如果要更改回电源管理的交付组的默认行为，请使用以下 cmdlet。

```
1 Set-BrokerSite  -ReuseMachinesWithoutShutdownInOutageAllowed
     $false
2 Set-BrokerDesktopGroup -Name "<Delievery Group Name>" -
     ReuseMachinesWithoutShutdownInOutage $false
3 <!--NeedCopy-->
```

现在应该完成使用本地主机缓存的无中断升级。

供稿人：Roman Siryk, Sr. 产品开发经理和高级吴锦涛质量工程师

借助 **RDP** 通过 **AWS** 中的 **Linux** 堡垒主机连接到 **Citrix** 体系结构

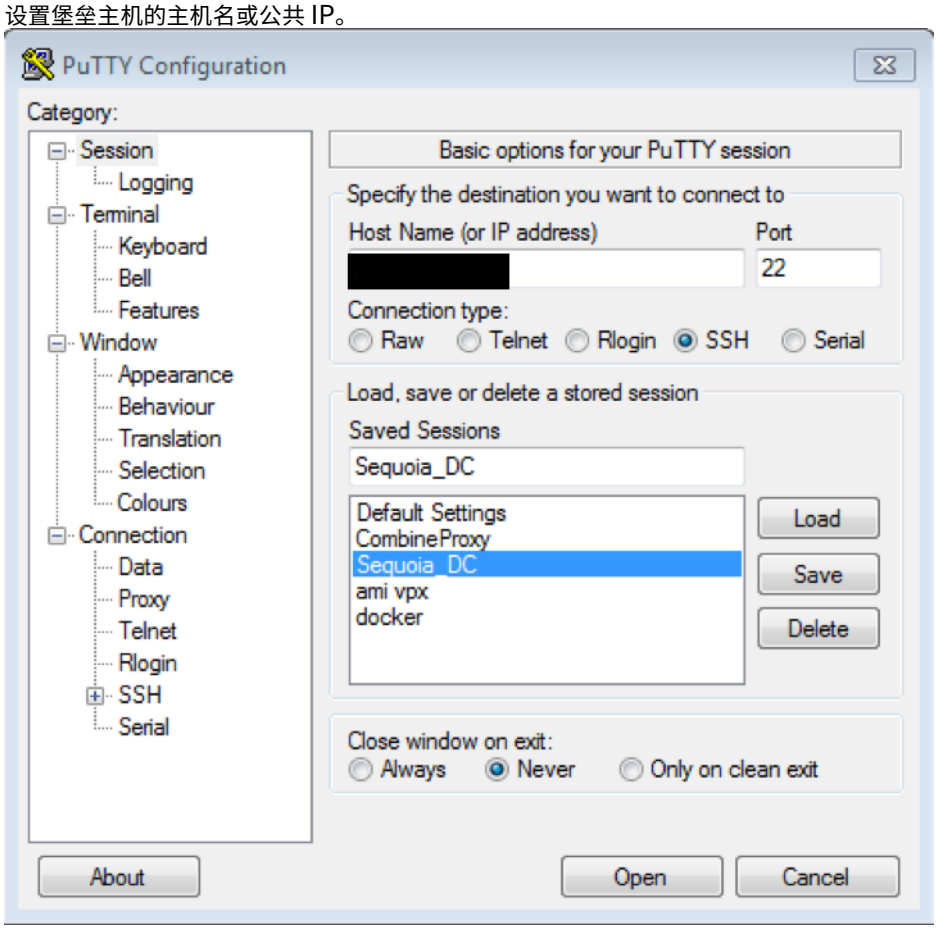
May 20, 2020

在 AWS 中设置 Citrix Virtual Apps and Desktops 环境时，请务必牢记安全注意事项。堡垒主机通常用于增强外部网络和内部网络之间的安全性和隔离，通常是承载代理服务器的剥离式 Linux 实例。对于 AWS 中的 Citrix 实现，管理员可能有权访问堡垒主机，但无法直接网络访问 Citrix 基础结构。由于 Citrix 基础架构由基于 Windows 的实例组成，并且包括基于 GUI 的元安装程序，因此通过基于 Linux 的堡垒主机进行连接会成为一个问题。

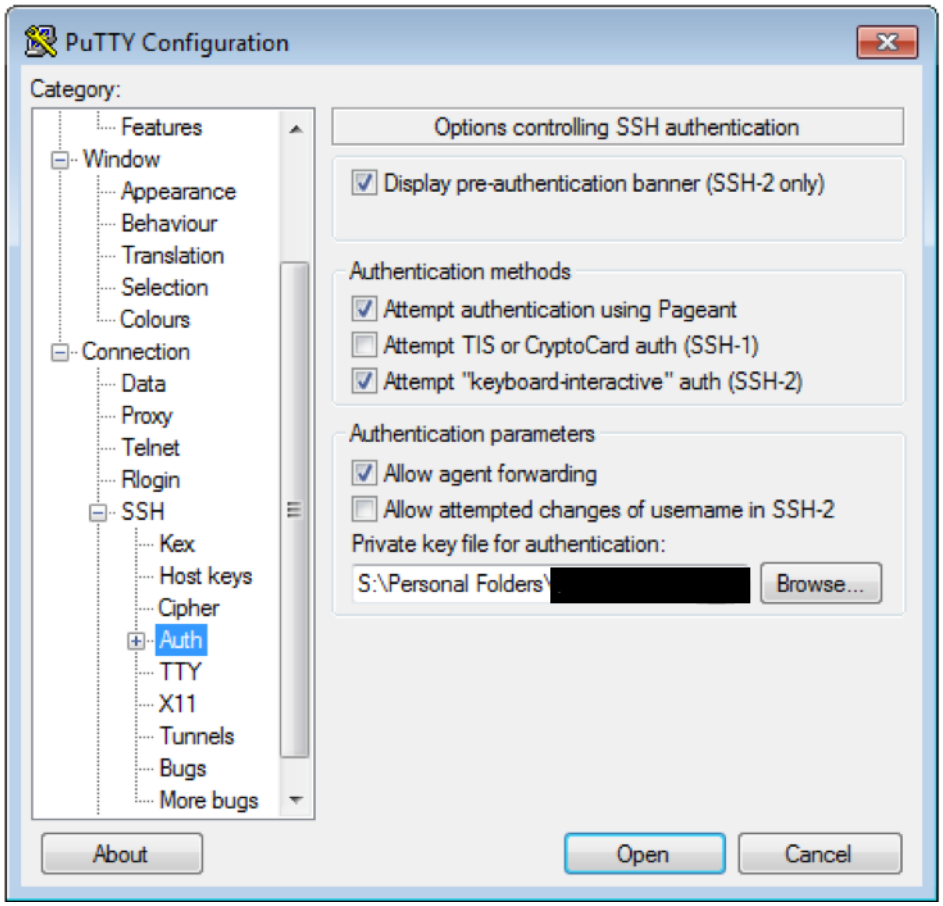
通过堡垒主机连接到 AWS 中的 Linux 实例与将 PuTTYing 连接到堡垒和将 SSHing 连接到所需实例一样简单。通过堡垒主机创建到 Windows 实例的 RDP 会话是可以使用端口转发。端口转发是目标 IP 和端口号的重新映射。它使受保护网络上的服务在 Gateway 的另一侧（如路由器）可用。在这种情况下，通过在首选 SSH/ 隧道实用程序中创建隧道，使用端口转发将本地端口映射到所需实例上的 RDP 端口。

例如，在 PuTTY 控制台中，创建 SSH 会话。输入堡垒主机的公有 IP，在“身份验证”部分提供私有密钥，然后创建隧道。隧道的源端口应该是未使用的本地端口，如本地主机 5000 及以上。IP 地址是附加 RDP 端口 (3389) 的目标主机（您试图访问的 Windows 实例）的 IP。请务必保存您的配置。连接到堡垒主机，然后登录。然后，启动本地端口的 RDP 会话。

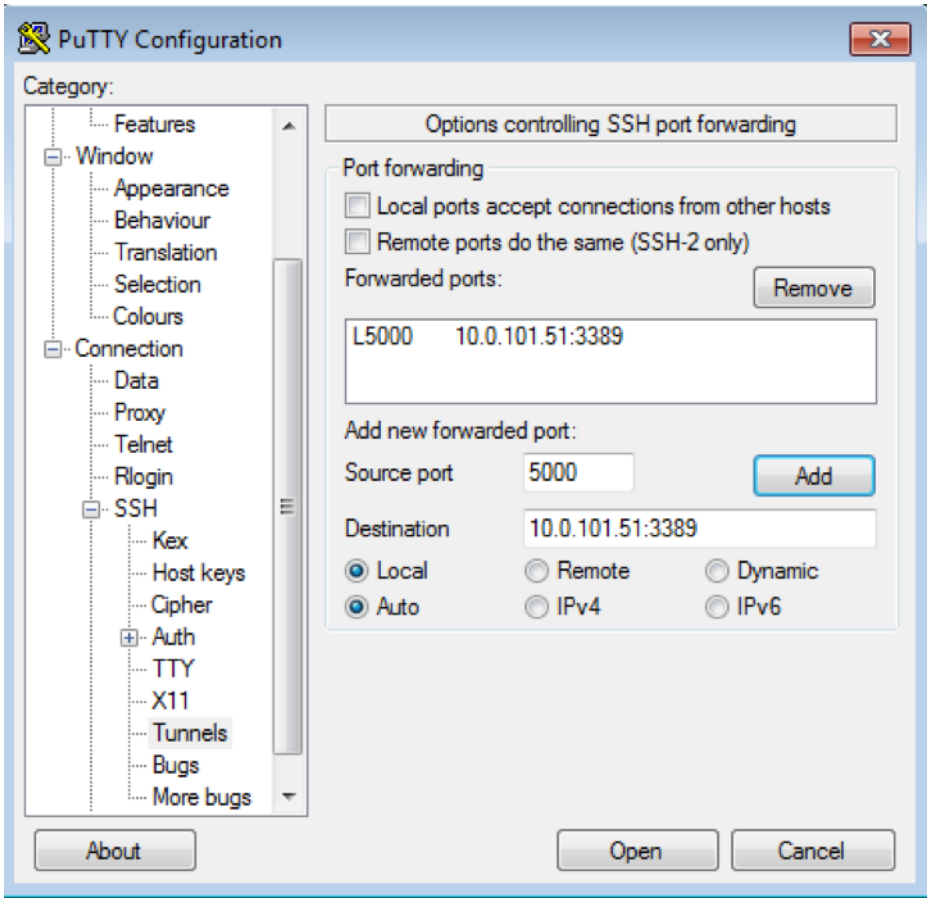
设置堡垒主机的主机名或公共 IP。



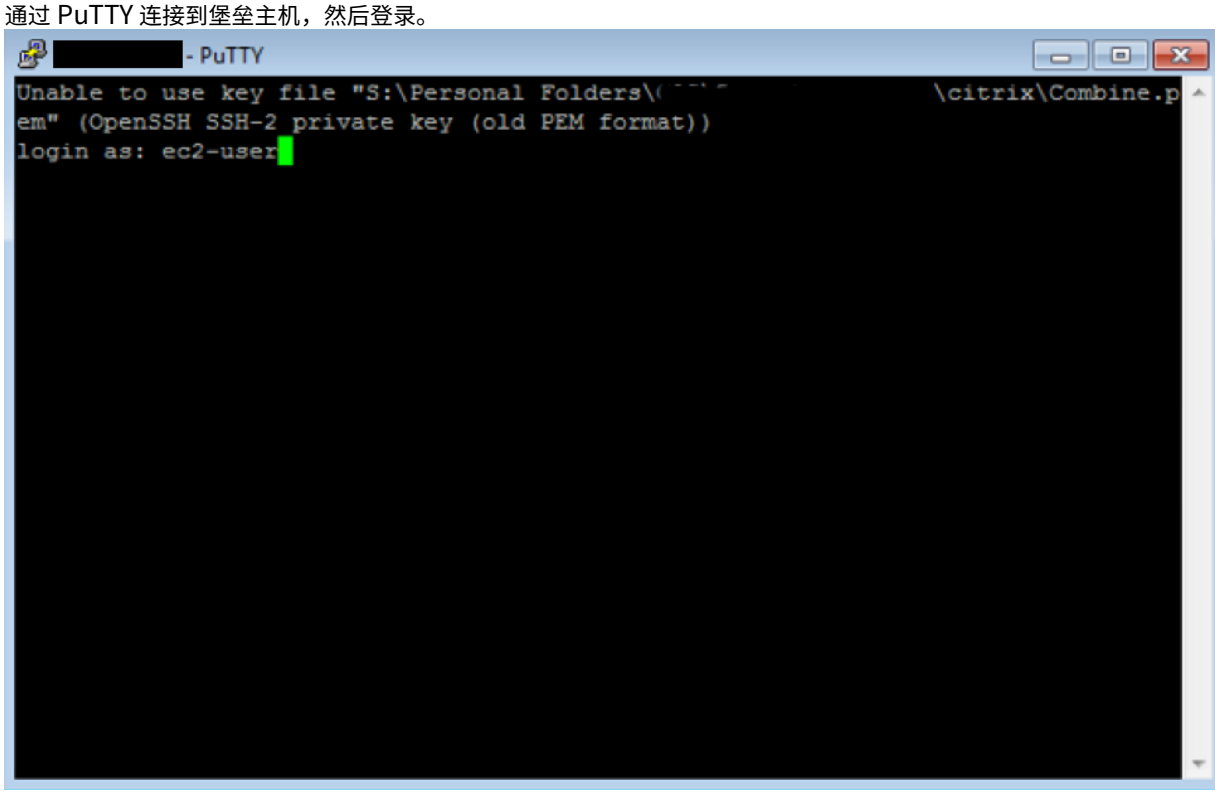
在 **SSH > 身份验证 ** 中，以.ppk 格式设置私钥文件。



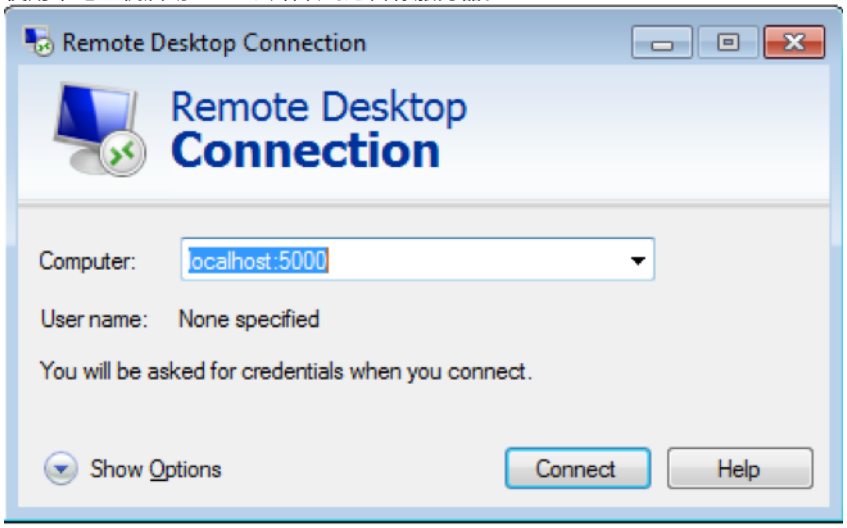
在 **SSH >** 隧道 ** 中，添加新的转发端口。源端口应是任意未使用的端口，目标端口应是堡垒主机后面目标服务器的 **IP**，并附加 **RDP** 端口。在源端口字段中，单击添加 ** 连接新的转发端口。



通过 PuTTY 连接到堡垒主机，然后登录。



使用本地主机启动 RDP 会话以到达目标服务器。



供稿人：Jill Fetscher，Citrix 首席顾问

面向 **Azure DNS** 专用区域的 **Citrix ADC** 部署指南

May 20, 2020

简介

Citrix ADC 以前称为 NetScaler，是应用程序交付控制器 (ADC) 空间中的世界级产品，具有负载平衡、管理全局流量、压缩和安全应用程序的能力。

Azure DNS 是 Microsoft 基础结构上的服务，用于托管 DNS 域和提供名称解析。

Azure DNS 私有区域是一项专注于解析专用网络中域名的服务。使用私有区域，客户可以使用自己的自定义域名，而不是现在可用的 Azure 提供的名称。

Azure DNS 概述

域名系统或 DNS 负责将服务名转换（或解析）为其 IP 地址。作为 DNS 域的托管服务，Azure DNS 通过使用 Microsoft Azure 基础结构提供名称解析。除了支持面向互联网的 DNS 域外，Azure DNS 现在还支持私有 DNS 域。

Azure DNS 提供可靠、安全的 DNS 服务，用于管理和解析虚拟网络中的域名，而无需自定义 DNS 解决方案。通过使用私有 DNS 区域，您可以使用自己的自定义域名，而不是现在可用的 Azure 提供的名称。使用自定义域名可帮助您定制虚拟网络体系结构，以最适合您组织的需求。它为虚拟网络内和虚拟网络之间的虚拟机 (VM) 提供名称解析。此外，客户还可以使用水平分割视图配置区域名称，从而允许私有和公有 DNS 区域共享名称。

适用于 **Azure DNS** 专用区域的 **Citrix GSLB?**

在当今世界中，企业希望将其工作负载从本地迁移到 Azure 云。向云的过渡使他们能够利用上市时间、资本开支/价格、易于部署和安全性。Azure DNS 专用区域服务为将部分工作负载过渡到 Azure 云的企业提供了独特的主张。这些企业在使用专用区域服务时，可以创建其私有 DNS 名称（在本地部署中使用了多年）。这种混合模型的内部网应用程序服务器位于本地，而 Azure 云通过安全 VPN 隧道连接，面临的一个挑战是用户如何能够无缝访问这些内部网应用程序。Citrix ADC 通过其全局负载平衡功能解决了这一独特的使用案例，该功能将应用程序流量路由到本地或 Azure 云上最佳的分布式工作负载/服务器，并提供应用程序服务器运行状况状态。

用例

本地网络和不同的 Azure VNET 中的用户应能够连接到内部网络中最佳服务器以访问所需内容。这确保了应用程序始终可用，优化的成本和用户体验是良好的。Azure 专用流量管理 (PTM) 是此处的主要要求。Azure PTM 确保用户的 DNS 查询解析为应用程序服务器的适当私有 IP 地址。

用例解决方案

Citrix ADC 包括全局服务器负载平衡 (GSLB) 功能，可帮助满足 Azure PTM 要求。GSLB 就像一个 DNS 服务器，它获取 DNS 请求并将该 DNS 请求解析为适当的 IP 地址，以提供：

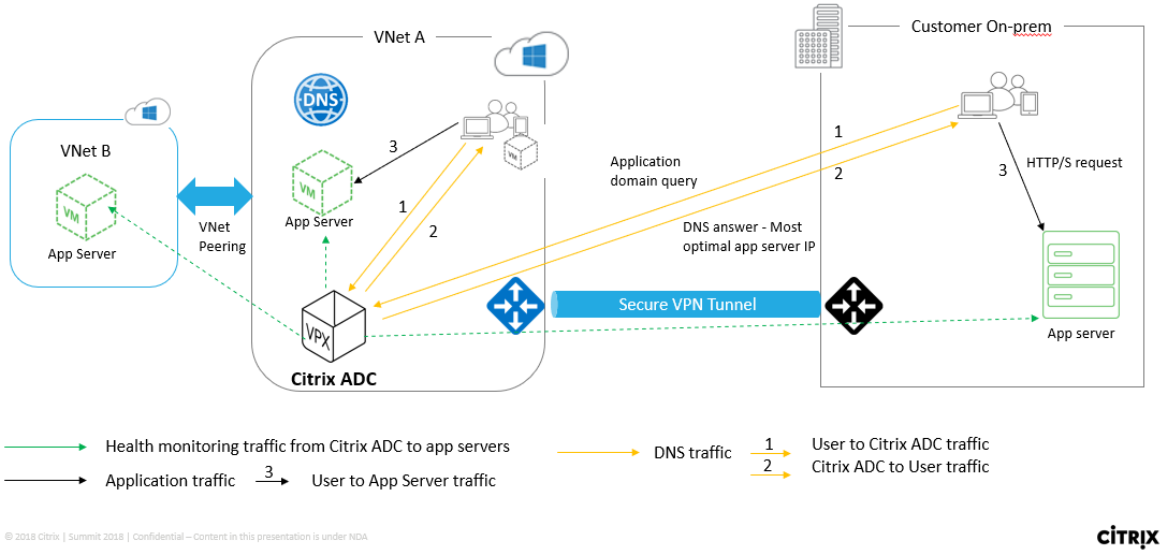
- 基于 DNS 的无缝故障切换
- 从内部部署到云的分阶段迁移
- A/B 测试新功能

在支持的许多负载平衡方法中，以下方法在此解决方案中非常有用：

1. 轮询机制
2. 静态接近（基于位置的服务器选择）：它可以通过两种方式进行部署
    a) Citrix ADC 上基于 EDNS 客户端子网 (ECS) 的 GSLB
    b) 为每个虚拟网络部署 DNS 转发器

拓扑

- 适用于 Azure 私有 DNS 区域的 Citrix ADC GSLB 部署在逻辑上如图 1 所示。



- 用户可以在 Azure 私有 DNS 区域中基于 Citrix ADC GSLB 负载平衡方法访问任何应用程序服务器
- 内部部署和 Azure 虚拟网络之间的所有流量仅通过安全 VPN 隧道
- 应用程序流量、DNS 流量和监视流量显示在前面的拓扑中。
- 根据所需的冗余，Citrix ADC 和 DNS 转发器可以部署在虚拟网络和数据中心中。为简单起见，此处仅显示一个 Citrix ADC，但我们建议至少使用一组适用于 Azure 区域的 Citrix ADC 和 DNS 转发器。
- 所有用户 DNS 查询首先转到具有将查询转发到相应 DNS 服务器定义的规则的 DNS 转发器。

为 **Azure DNS** 专用区域配置 **Citrix ADC**

经过测试的产品和版本

产品	版本
Azure	云订阅
Citrix ADC VPX	BYOL（自备许可证）

> 注意：部署已经过测试，并且与 Citrix ADC 12.0 版及更高版本保持不变。

先决条件和配置说明

以下是针对本指南测试的一般先决条件和配置，请在配置 Citrix ADC 之前交叉检查：

- 具有有效订阅的 Microsoft Azure 门户帐户
- 确保本地云和 Azure 云之间的连接（安全 VPN 隧道）。若要在 Azure 中设置安全 VPN 隧道，请参阅 分步：在 Azure 和本地之间配置站点到站点 VPN 网关

解决方案描述

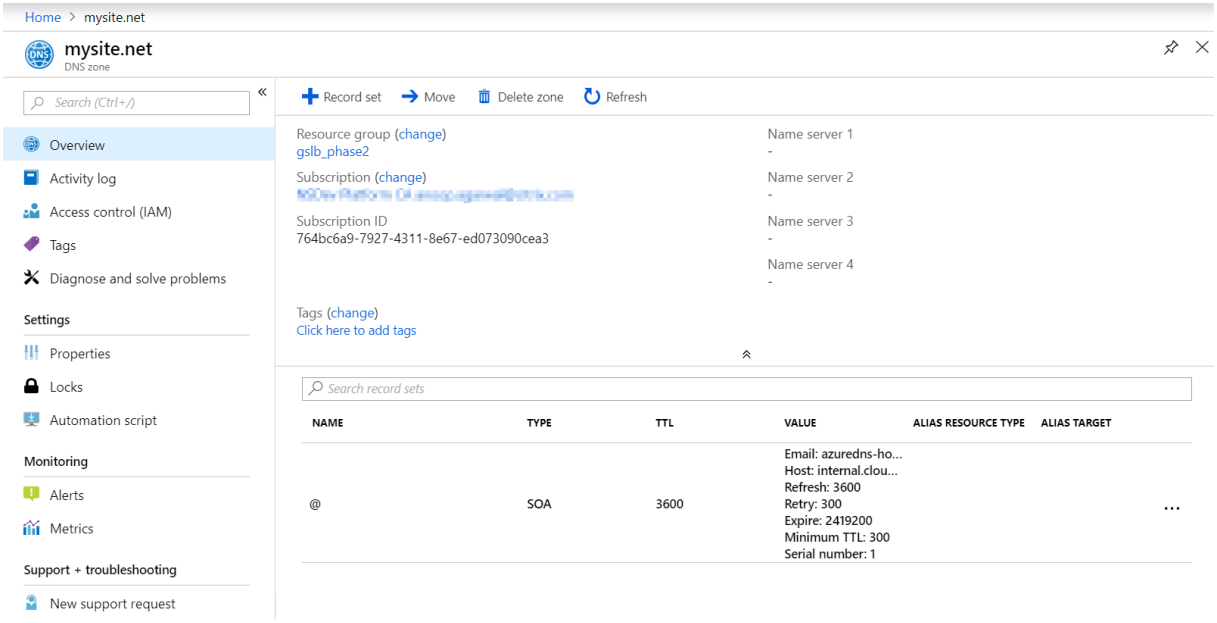
假设客户希望托管一个应用程序 Azure DNS 专用区域（rrr.ptm.mysite.net），该应用程序在 HTTPS 上运行，并且基于轮询 GSLB 负载平衡方法在 Azure 和本地部署，具有内联网访问权限。为了实现此部署，通过使用 Citrix ADC 启用 Azure 专用 DNS 区域的 GSLB 由两部分组成 — 配置 Azure、本地和 Citrix ADC 设备。

第 **1** 部分：配置 **Azure**，本地安装

如拓扑中所示，设置 Azure 虚拟网络（在本例中为 VNet A，VNet B）和本地设置。
步骤 1：创建具有域名 (mysite.net) 的 Azure 私有 DNS 区域
步骤 2：在 Azure 区域的集线器和辐条模型中创建两个虚拟网络（VNet A、VNet B）
步骤 3：在 VNet 中部署应用程序服务器、DNS 转发器、Windows 10 专业客户端和 Citrix ADC
步骤 4：部署应用程序服务器并部署 DNS 转发器（如果任何客户端位于 VNet B 中）
步骤 5：在本地部署应用程序服务器、DNS 转发器和 Windows 10 专业客户端

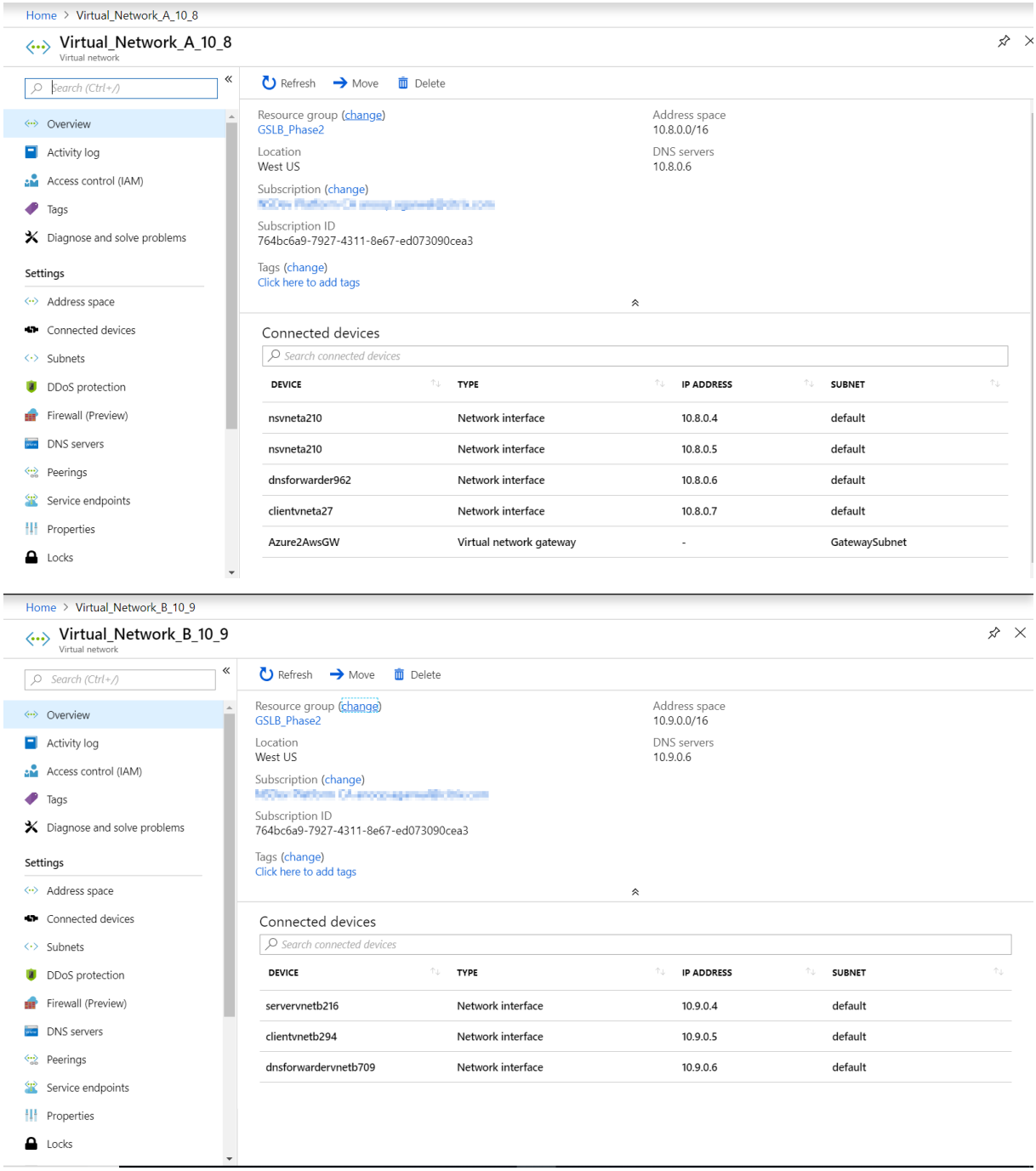
Azure 私有 **DNS** 区域

登录 Azure 门户并选择或创建控制板。现在，单击创建资源并搜索 **DNS** 区域以创建一个资源（本例中为 mysite.net），如下图所示。



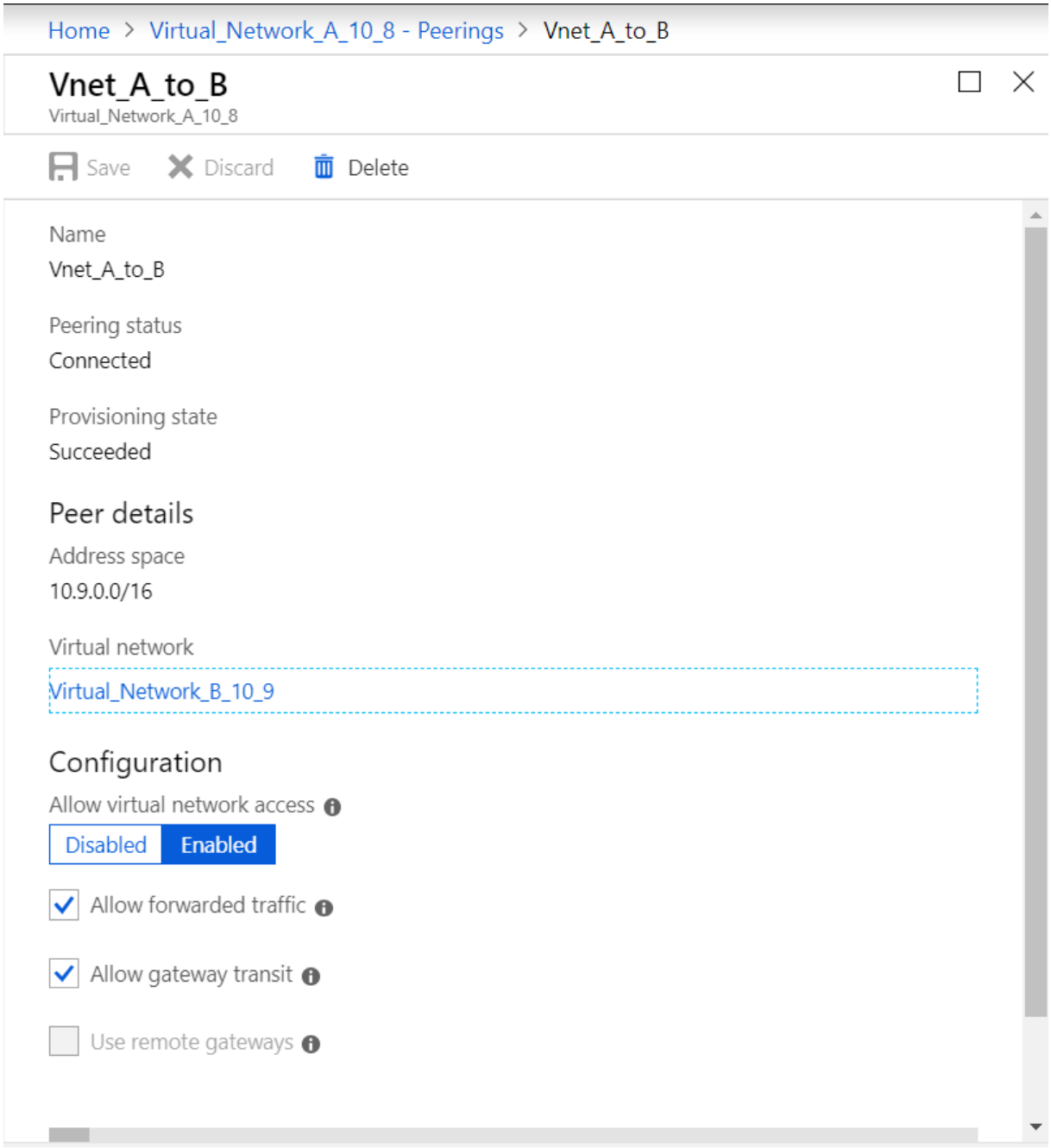
在集线器和辐条模型中的 **Azure** 虚拟网络（**VNet A**，**VNet B**）

选择同一控制板，然后单击 创建资源并搜索虚拟网络以创建两个虚拟网络，即同一区域中的 VNet A、VNet B，然后对等它们形成集线器和辐条模型，如下图所示。有关如何设置集线器和分支拓扑的信息，请参阅在 Azure 中实施集线器辐射网络拓扑。

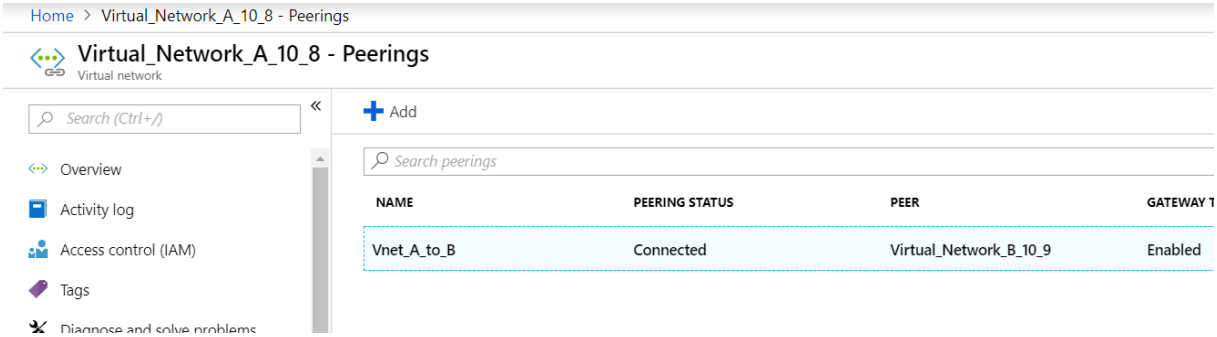


VNet A 到 **VNet B** 对等网络

要对等 VNet A 和 VNet B，请从 VNet A 和对等 VNet B 的设置菜单中单击对等，启用“允许转发流量”和“允许网关传输”，如下图所示。

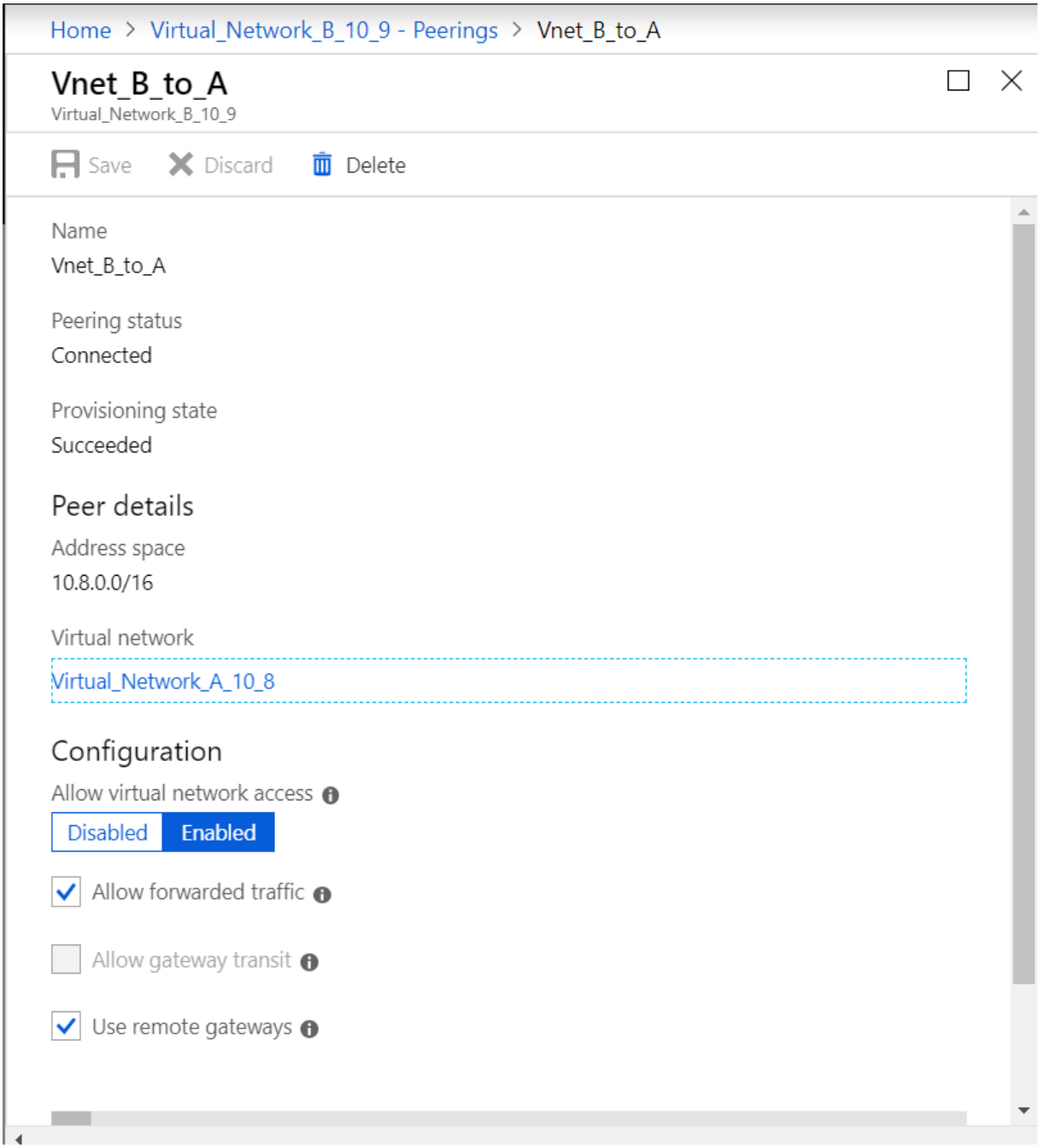


成功对等后，您会看到如下图所示：

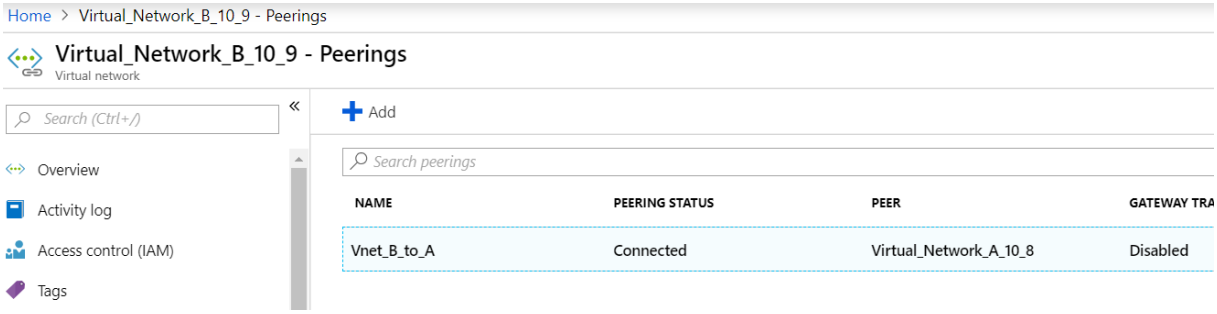


VNet B 到 **VNet A** 对等网络

要将 VNet B 和 VNet A 对等，请在 VNet B 和 VNet A 的设置菜单中单击对等，启用“允许转发通信”并使用远程网关，如下图所示。



成功对等后，您会看到如下图所示：



在 **VNet A** 中部署应用程序服务器、**DNS** 转发器、**Windows 10 Pro** 客户端、**Citrix ADC**

我们简要讨论有关 VNet A 上的应用服务器、DNS 转发器、Windows 10 专业客户端和 Citrix ADC 的信息。选择相同的控制板，单击创建资源，搜索相应的实例并从 VNet A 子网分配 IP

应用程序服务器

应用程序服务器只不过是 Web 服务器（HTTP 服务器），其中 Ubuntu 服务器 16.04 作为 Azure 或本地虚拟机上的实例部署并运行 CLI 命令：sudo apt 安装 apache2 使其成为 Web 服务器

Windows 10 Pro 客户端

将 Windows 10 专业实例作为客户端计算机在 VNet A 和本地也启动。

Citrix ADC

Citrix ADC 通过运行状况检查和来自 Citrix MAS 的分析对 Azure DNA 专用区域进行补充。根据您的要求从 Azure 应用商店启动 Citrix ADC，在本文中，我们已使用 Citrix ADC (BYOL) 进行此部署。有关如何在 Microsoft Azure 上部署 Citrix ADC 的详细步骤，请参阅下面的 URL。部署后，使用 Citrix ADC IP 配置 Citrix ADC GSLB。请参阅在 Microsoft Azure 上部署一个 NetScaler VPX 实例

DNS 转发器

它用于转发绑定到 Citrix ADC GSLB (ADNS IP) 的托管域的客户端请求。启动 Ubuntu 服务器 16.04 作为 Linux 实例（Ubuntu 服务器 16.04），并参考下文 URL 了解如何将其设置为 DNS 转发器。

> 注意：对于轮询 GSLB 负载平衡方法，Azure 区域一个 DNS 转发器就足够了，但对于静态邻近，我们需要每个虚拟网络一个 DNS 转发器。从 https://github.com/Azure/azure-quickstart-templates/tree/master/301-dns-forwarder 下载快速入门模板

部署转发器后，将虚拟网络 A 的 DNS 服务器设置从默认值更改为使用 VNet A DNS 转发器 IP 进行自定义，如下图所示，然后修改 VNet A DNS 转发器中的 `named.conf.options` 文件以将域 (mysite.net) 和子域 (ptm.mysite.net) 的转发规则添加到 Citrix ADC GSLB 的 ADNS IP。现在，重新启动 DNS 转发器以反映在文件 `named.conf.options` 中所做的更改。

VNet A DNS 转发器设置

```
1     zone "mysite.net" {
2
3               type forward;
4     forwarders {
```



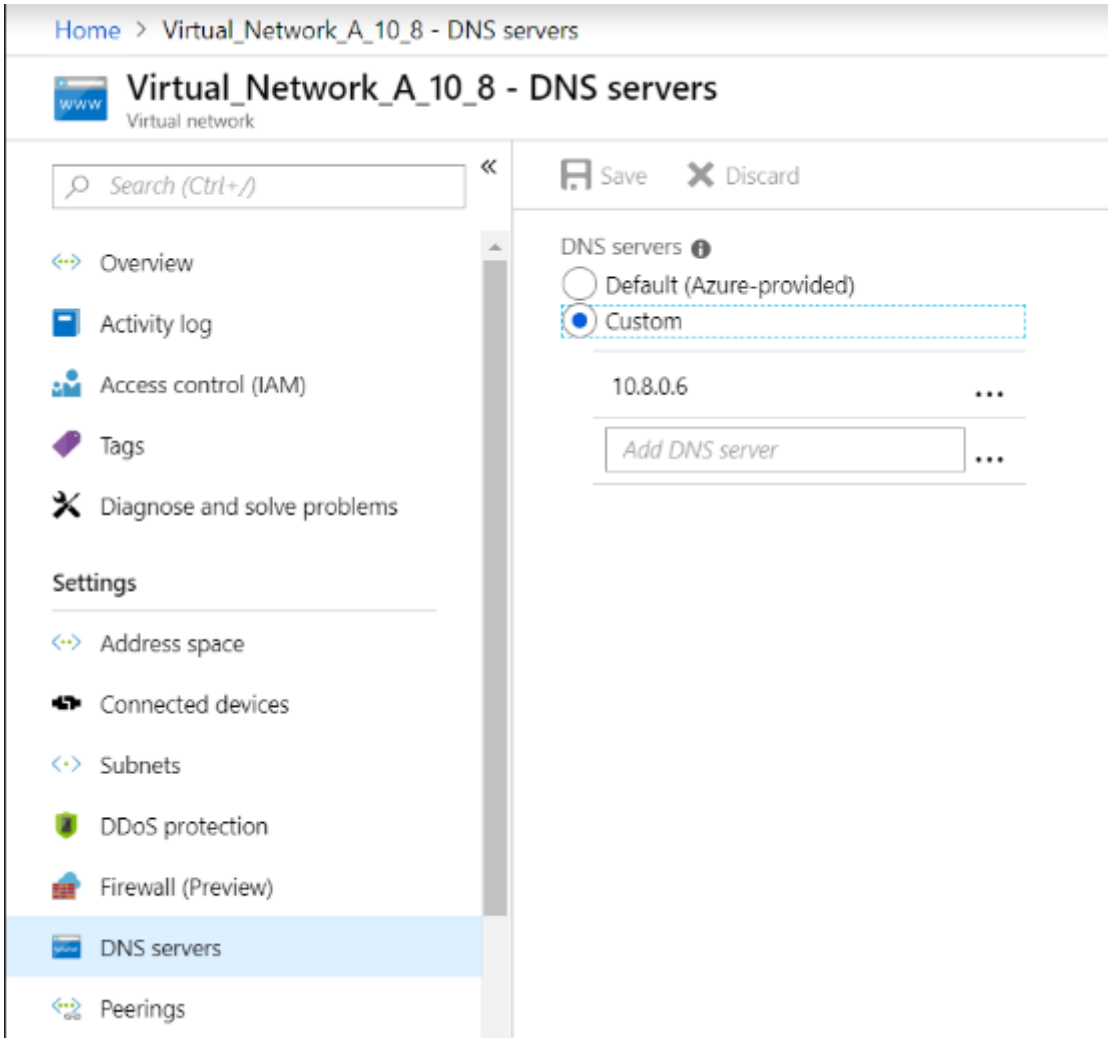
```
5    168.63.129.16;  }
6   ;
7       }
8   ;
9      zone ”ptm.mysite.net” {
10
11          type forward;
12          forwarders {
13    10.8.0.5;  }
14  ;
15      }
16  ;
17 <!--NeedCopy-->
```

> 注意：对于域（“mysite.net”）区域 IP 地址，请使用 Azure 区域的 DNS IP。对于子域（“ptm.mysite.net”）区域 IP 地址，请使用您的 GSLB 实例的所有 ADNS IP 地址。

如果任何客户端在 **VNet B** 中，则部署应用程序服务器并部署 **DNS** 转发器

现在，对于虚拟网络 B，选择相同的控制板，单击 创建资源，然后搜索相应的实例，然后从 VNet B 子网分配一个 IP。如果存在类似于 VNet A 的静态邻近 GSLB 负载平衡，则启动应用程序服务器和 DNS 转发器。
编辑 `named.conf.options` 中的 VNet B DNS 转发器设置，如下所示：

```
1      VNet B DNS Forwarder Settings:
2      zone ”ptm.mysite.net” {
3
4          type forward;
5          forwarders {
6    10.8.0.5;  }
7   ;
8       }
9   ;
10 <!--NeedCopy-->
```

在本地部署应用程序服务器、**DNS** 转发器和 **Windows 10** 专业客户端

现在，对于本地，在裸机上启动虚拟机，并带来类似于 VNet A 的应用程序服务器、DNS 转发器和 Windows 10 专业客户端。

编辑 `named.conf.options` 中的本地 DNS 转发器设置，如下例所示。

本地 **DNS** 转发器设置

```
    zone ”mysite.net” {

                type forward;
                forwarders {
 10.8.0.6;  }
;
```



```
7       }
8   ;
9      zone ”ptm.mysite.net” {
10
11          type forward;
12          forwarders {
13    10.8.0.5;  }
14   ;
15       }
16   ;
17  <!--NeedCopy-->
```

在本文中，mysite.net 我们已经给予 VNet A 的 DNS 转发器 IP 而不是 Azure 私有 DNS 区域服务器 IP，因为它是一个特殊的 IP 无法从内部部署. 因此，在内部部署的 DNS 转发器设置中需要进行此更改。

第 **2** 部分：配置 **Citrix ADC**

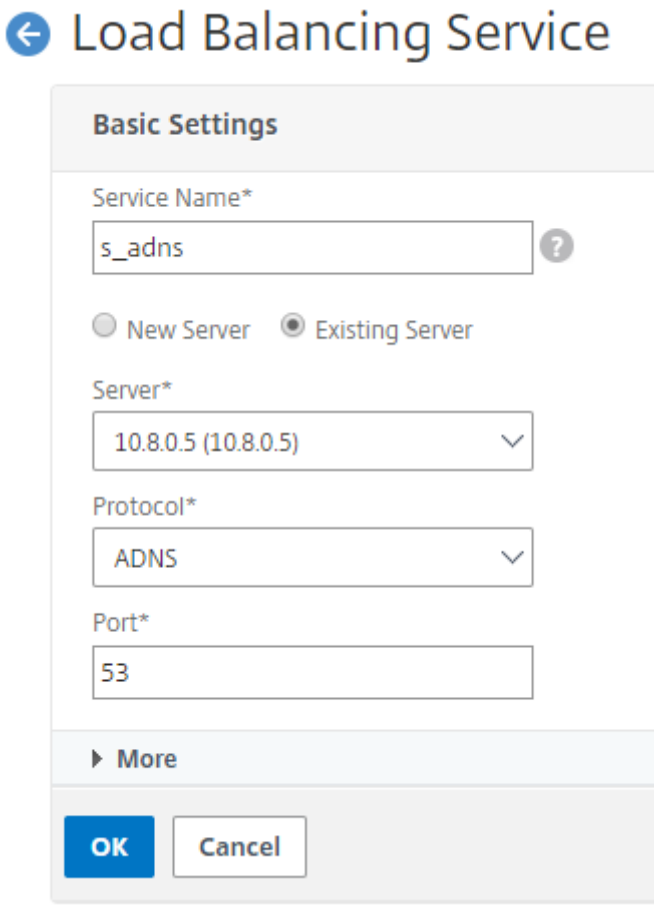
如拓扑中所示，在 Azure 虚拟网络（本例中为 VNet A）上部署 Citrix ADC，并通过 Citrix ADC GUI 访问它。

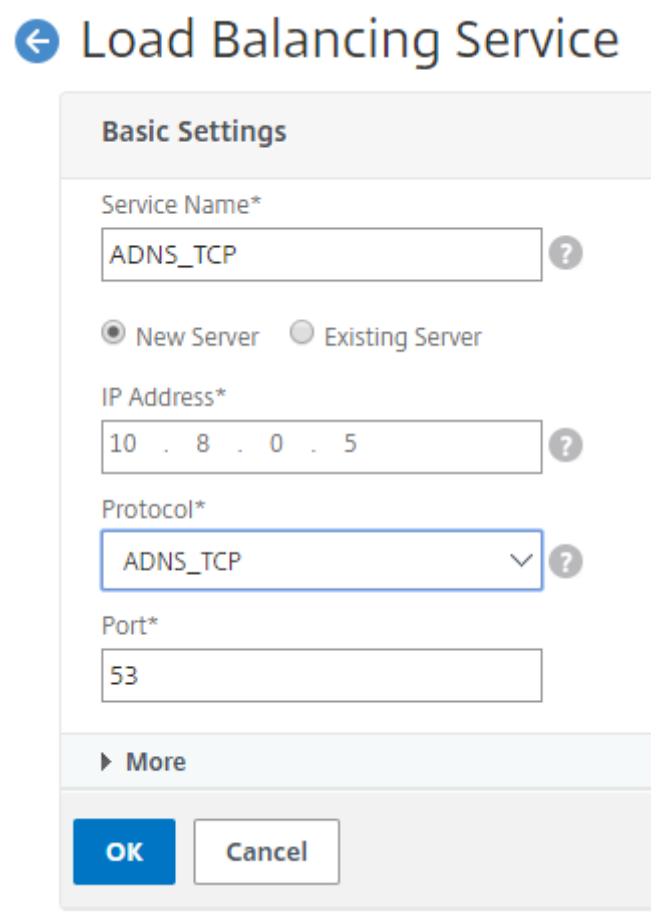
配置 **Citrix ADC GSLB**

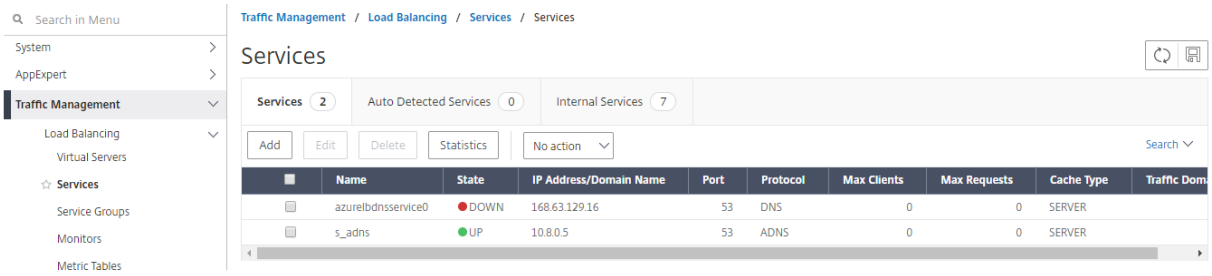
步骤 1：创建 ADNS 服务
步骤 2：创建站点 — 本地和远程
步骤 3：为本地虚拟服务器创建服务
步骤 4：为 GSLB 服务创建虚拟服务器

添加 **ADNS** 服务

登录到 Citrix ADC GUI。在“配置”选项卡上，导航到“流量管理”>“负载平衡”>“服务”。添加服务。建议在 TCP 和 UDP 中配置 ADNS 服务，如下所示：

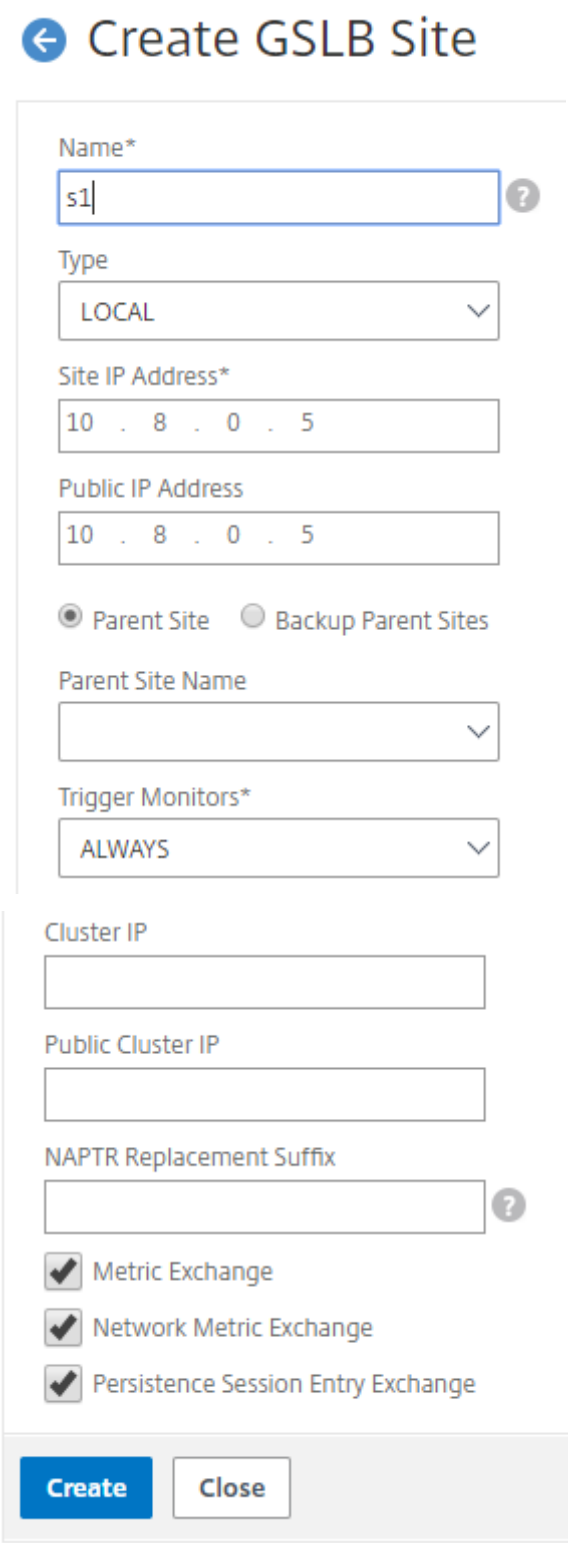


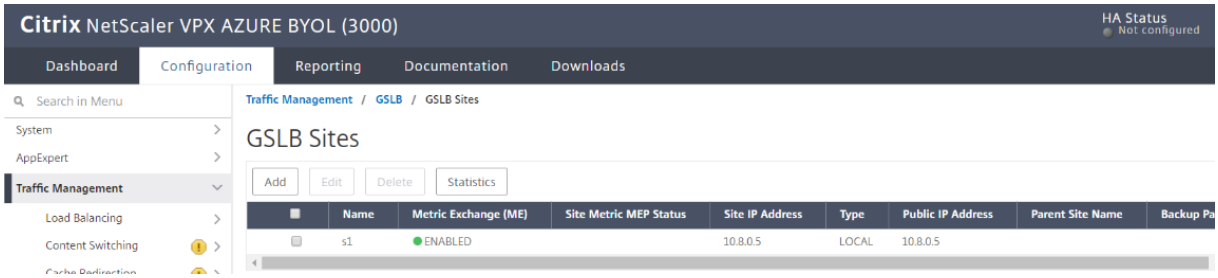




添加 **GSLB** 站点

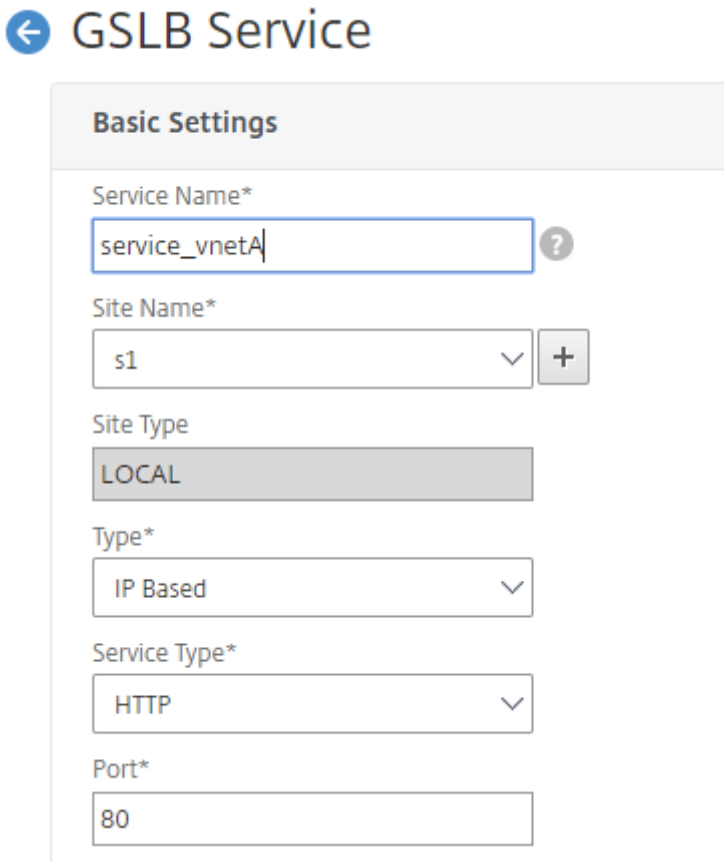
添加将在其中配置 GSLB 的本地和远程站点。在配置选项卡上，导航到流量管理 **> GSLB > GSLB** 站点。如此处所示添加站点，然后对其他站点重复相同的步骤。

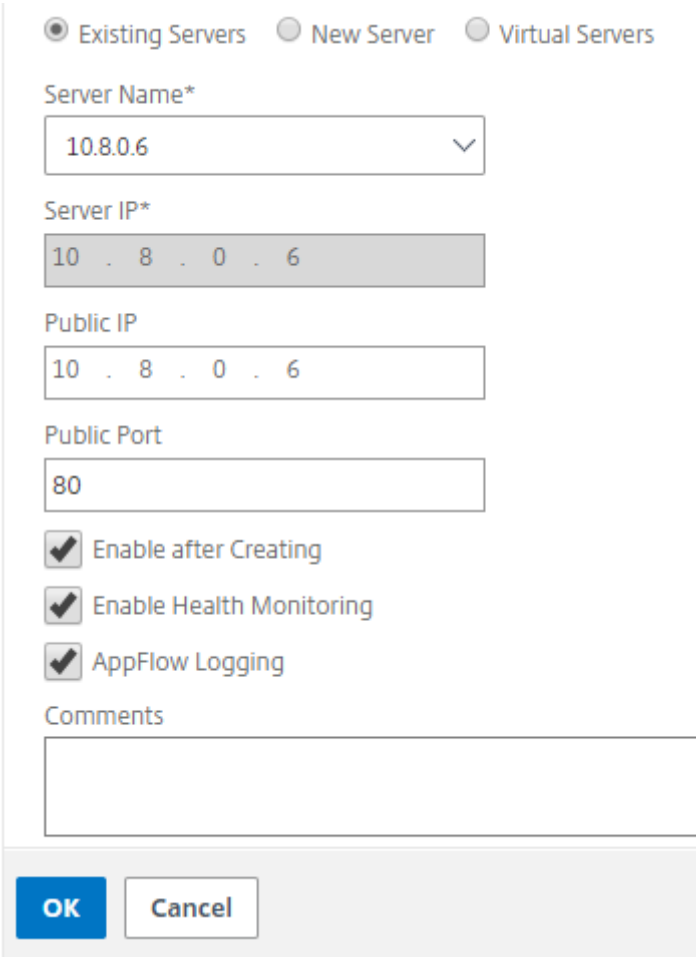




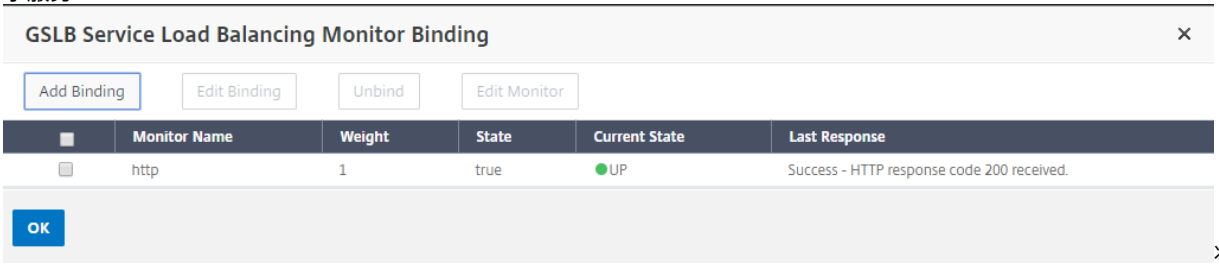
添加 **GSLB** 服务

为本地和远程虚拟服务器添加 GSLB 服务，以平衡应用服务器。在配置选项卡上，导航到流量管理 **> GSLB > GSLB** 服务。添加服务，如以下示例所示。绑定 HTTP 监视器以检查服务器状态。



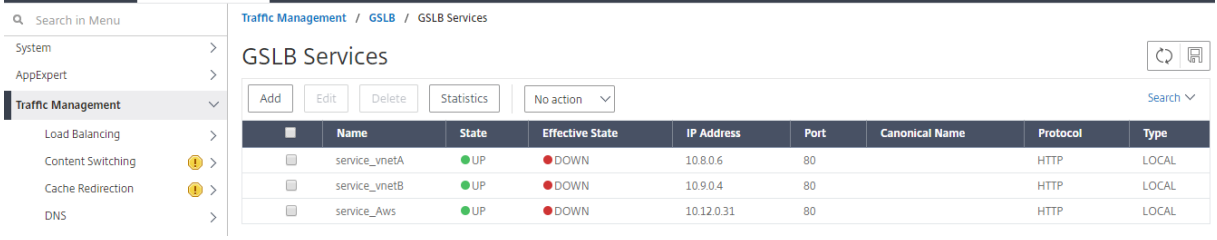


创建服务后，转到 GSLB 服务内的高级设置选项卡，并添加监视器选项卡以将 GSLB 服务与 HTTP 监视器绑定，以显示服务



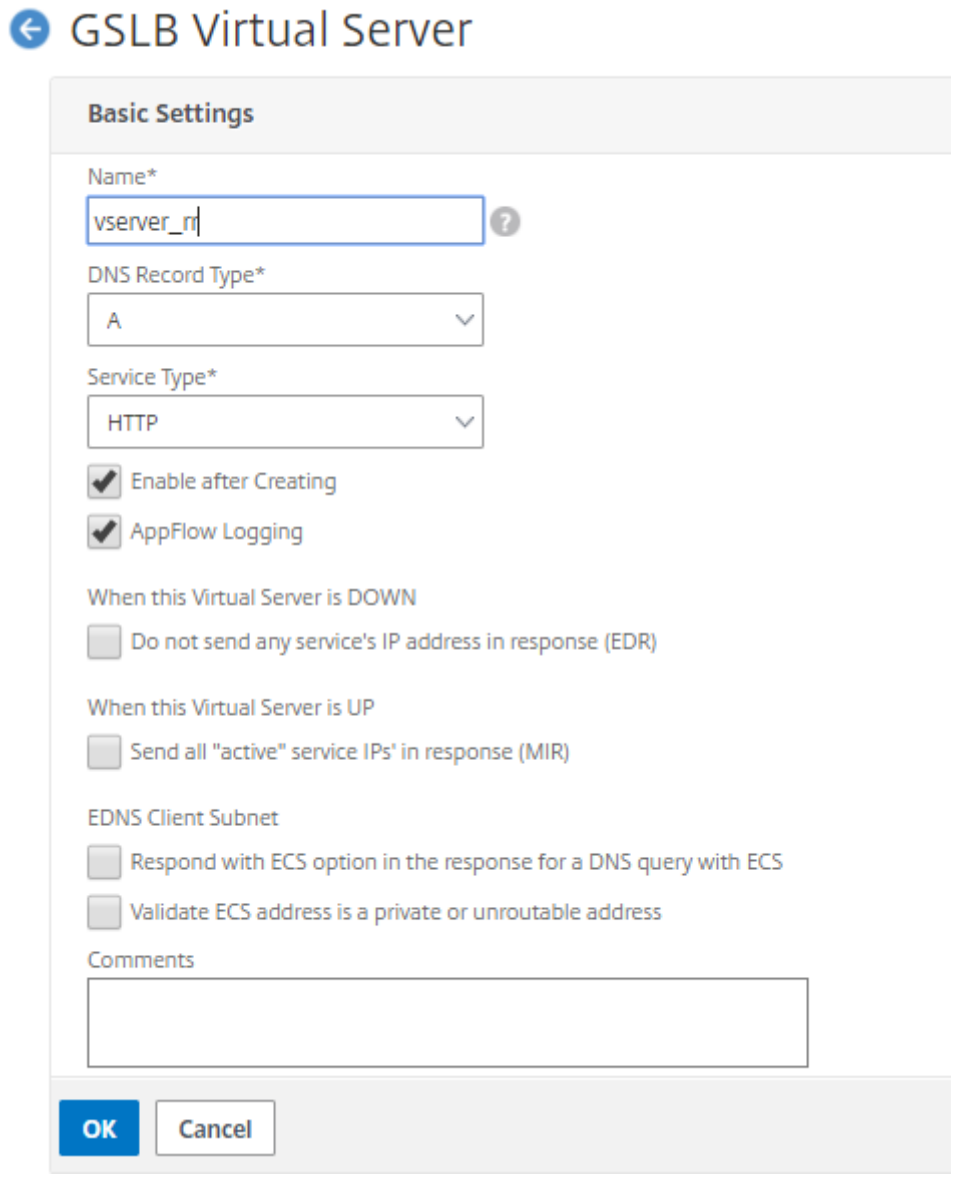
状态

使用 HTTP 监视器绑定后，服务状态将为 UP，如下所示：

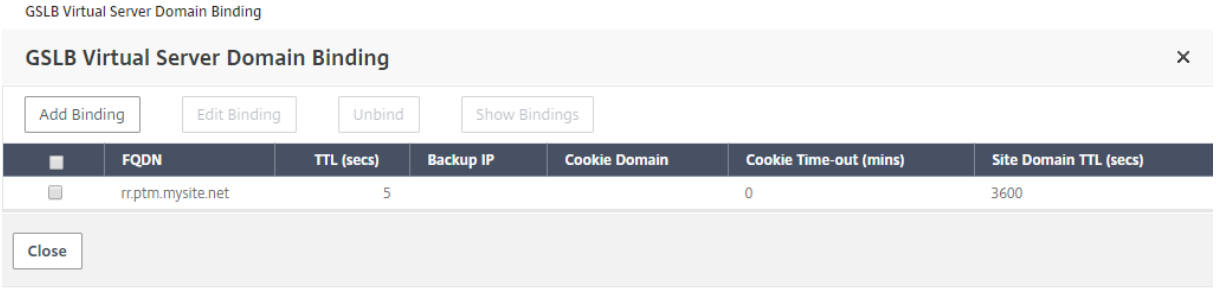


添加 **GSLB** 虚拟服务器

添加 GSLB 虚拟服务器，通过该服务器可以访问应用服务器的别名 GSLB 服务。在配置选项卡上，导航到流量管理 **> GSLB > GSLB** 虚拟服务器。添加虚拟服务器，如以下示例所示。将 GSLB 服务和域名绑定到它。

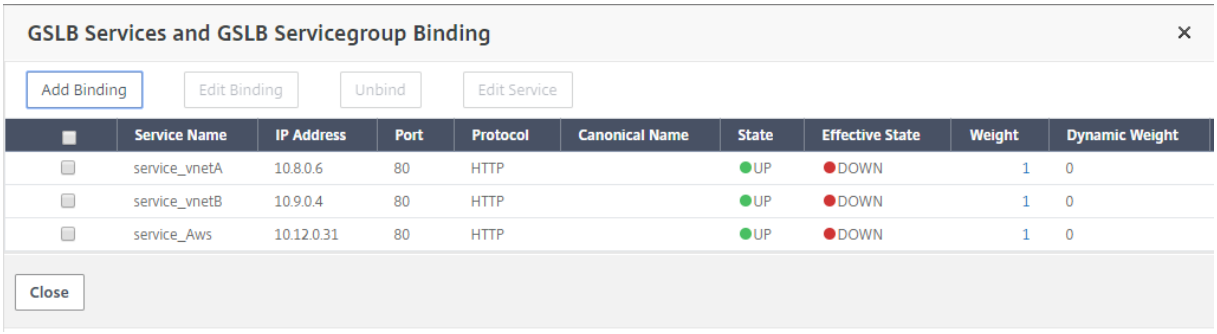


创建 GSLB 虚拟服务器并选择适当的负载平衡方法（在本例中为轮询）后，绑定 GSLB 服务和域以完成步骤

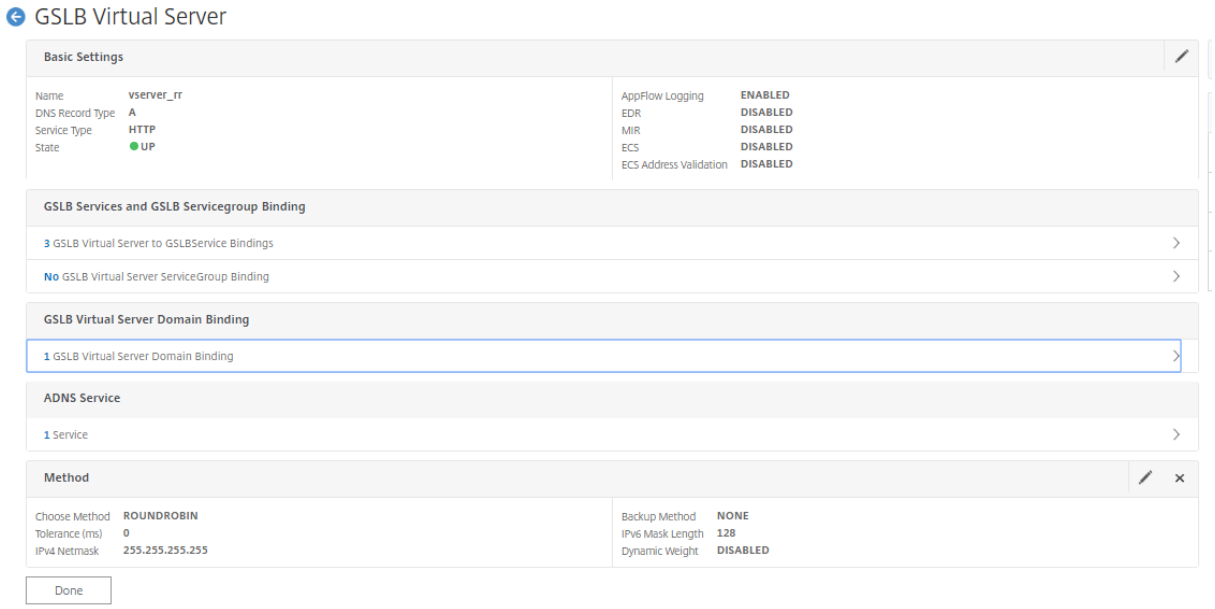


转到虚拟服务器内的高级设置选项卡并添加域选项卡以绑定域

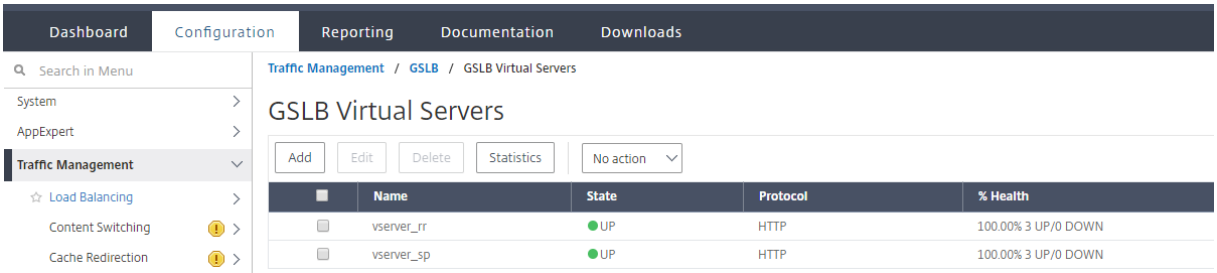
转到 高级 > 服务，然后单击箭头以绑定 GSLB 服务并将所有三个服务（VNet A、VNet B、本地）绑定到虚拟服务器



将 GSLB 服务和域绑定到虚拟服务器后，如下所示：



检查 GSLB 虚拟服务器是否启动且 100% 正常运行。当监视器显示服务器正常运行时，表示站点处于同步状态，并且后端服务可用。



现在要测试部署，请从云客户端计算机或本地客户端计算机访问域 URL `rr.ptm.mysite.net`。假设从云 Windows 客户端计算机访问它，即使本地应用程序服务器也可以在私有 DNS 区域中访问，而无需任何第三方或自定义 DNS 解决方案。

结论

Citrix ADC 是领先的应用程序交付解决方案，最适合为 Azure DNS 专用区域提供负载平衡和 GSLB 功能。通过订阅 Azure DNS 专用区域，企业可以依靠 Citrix ADC 全球服务器负载平衡 (GSLB) 的能力和智能来跨位于多个地理位置的工作负载和跨数据中心（通过安全 VPN 隧道连接）分配内部网流量。这种协作可确保企业无缝访问他们希望迁移到 Azure 公有云的部分工作负载。

Citrix 联合身份验证服务登录证据概述

May 20, 2020

简介

联合身份验证服务 (FAS) 是与 Active Directory 证书颁发机构 (CA) 集成的 Citrix 组件，允许用户在 Citrix 环境中无缝进行身份验证。有关 FAS 体系结构和部署的信息，请参阅 联合身份验证服务文档。

您可以部署 FAS 以允许用户单次登录 VDA（或已发布的应用程序），而无需使用密码或智能卡。FAS 登录证据功能提供由 Citrix Gateway 和 StoreFront 传递给 FAS 的登录证据。FAS 可以验证证据，以确保证据是由受信任的身份提供商 (IdP) 颁发的。

本文介绍如何配置 FAS 登录证据功能。

概述

FAS 信托基金

FAS 基础结构涉及 Citrix Gateway (NSG)、StoreFront (SF) 和 FAS 之间的“信任链”；每个箭头从信任组件指向受信任组件：

每个组件之间信任的关键数据是访问系统的用户的用户主体名称 (UPN)。UPN 通过链接流动（与箭头相反的方向）。当 UPN 流经系统时，也可能会转换为不同的 UPN，但这与本主题并不直接相关。

身份提供程序 (IdP) 是用户进行身份验证的地方。IdP 通常是第三方网站，如 Okta 或 Azure。用户通过提供一组凭据（如密码或更复杂的东西）在 IdP 进行身份验证。由于组件之间存在信任链，沿链的组件接受 UPN 是真实的。

通过以下方式建立信任，在上图中标记为 1、2、3：

(1) Citrix Gateway 或 StoreFront 使用涉及签名声明的协议（例如，声明用户的 UPN）信任 IdP。信任方可以验证 IdP 提出的声明，因为它配置了用于检查签名是否有效的证书。有两种主要协议用于验证：SAML（安全断言标记语言）和 OpenID Connect。登录证据功能目前仅支持 SAML。

(2) 此信任是通过使用受信任的 Citrix Gateway 的详细信息配置 StoreFront 来建立的。这些组件之间的协议“CitrixAgBasic”允许 StoreFront 确认它是由受信任的 Citrix Gateway 调用的。

(3) 此信任是通过 Kerberos 建立的。FAS 配置了受信任的 StoreFront 服务器列表。Kerberos 用于检查调用 StoreFront 服务器的标识是否在此列表中。

安全性

安全身份验证依赖于正确建立的信任链。通过验证 IdP 提供的证据来加强信任链，IdP 是安全身份验证的信任根源。这一点很重要，因为通过信任链提供给 FAS 的用户凭据包括用户名（UPN），但不包括 FAS 可以自行验证的密码（如密码）。因此，密码的暴露仅限于 IdP。大多数联合身份验证系统都是这样运行的，包括 FAS。

登录证据

FAS 登录证据功能在 FAS 部署中提供额外的安全保证。它允许您定义允许或拒绝访问 FAS 的规则。

登录证据（或只是“证据”）是 IdP 在用户进行身份验证时创建的一段数据。这些数据与 UPN 一起流入整个系统。在 VDA 启动时，FAS 可以在允许启动之前检查证据是否有效。

目前，只有支持 SAML 的 IdP 才受支持。证据是 SAML 响应，它是一个 XML 文档，其中包含一组由 IdP 签署的声明。（IdP 是身份验证的信任根）。

FAS 插件

FAS 没有任何内置功能来检查登录证据是否有效。相反，您需要使用 FAS 断言 SDK 编写自己的 FAS 插件。您的插件负责检查提供的 UPN 和证据（SAML 响应）。

配置登录证据收集

步骤 **1-**创建部署

像往常一样使用 Citrix Gateway、StoreFront 和 FAS 创建部署。将 Citrix Gateway 或 StoreFront 配置为对 IdP 使用 SAML 身份验证。

> 重要：
>
> 如果您使用的是 Citrix Gateway，则在使用 Citrix Gateway 的详细信息配置 StoreFront 时，必须配置回调 **URL**，因为登录证据是通过回调传输的：

Citrix 建议您将 登录类型配置为“智能卡”，以帮助本机客户端执行 SAML 身份验证。

检查您的部署是否正常工作。换句话说，检查您是否可以登录并启动 VDA 会话，而无需在 VDA 上提示您输入凭据。

步骤 **2-**安装示例 **FAS** 断言插件

FAS 断言 SDK 包含一个示例插件，您可以将其用作自己插件的基础。

> 注意：
>
> Citrix 强烈建议您先安装示例插件，而不进行任何更改。

有关安装插件的说明，请参阅随 FAS 断言 SDK 提供的 *Readme.txt* 文件。

步骤 **3-**检查 **FAS** 断言插件是否正常工作

安装插件后，额外的事件将写入 FAS 服务器事件日志的 **Windows** 日志/应用程序部分。有关日志记录和跟踪的说明，请参阅 FAS 断言 SDK。

步骤 **4 -** 在 **Citrix Gateway** 上启用证据收集

如果使用 Citrix Gateway 进行身份验证，则必须启用证据收集功能，以便将证据从 Citrix Gateway 传输到 StoreFront。为此，请使用 Citrix ADC 管理控制台为 Gateway 服务器启用“存储 SAML 响应”选项，请参阅 SAML 身份验证。

步骤 **5 -** 在 **StoreFront** 上启用证据收集

> 注意：
>
> 如果启用登录证据，则必须在 FAS 服务器上部署 FAS 断言插件模块。

默认情况下，StoreFront 不会向 FAS 发送证据（即使配置了 SAML 身份验证）。若要在 StoreFront 中启用登录证据，请使用以下 PowerShell 为与名为“应用商店”的应用商店关联的身份验证服务启用该证据。

```
Get-Module ”Citrix.StoreFront.* -ListAvailable | Import-Module

$StoreName = ”Store” $StoreVirtualPath = ”/Citrix/” + $StoreName $store = Get-STFStoreService -VirtualPath $StoreVirtualPath $auth = Get-STFAuthenticationService -StoreService $store

$auth.AuthenticationOptions.CollectFasEvidence = $true

$auth.Save()
<!--NeedCopy-->
```

步骤 **6-**修改示例 **FAS** 断言插件

示例插件中的骨架代码接受任何证据。更新示例中的代码以检查提供的登录证据（SAML 响应）是否有效。

您有责任确保所提供的证据得到核对。请注意：

- 检查 SAML 声明是否具有加密有效的签名

- 检查 SAML 声明是否使用 IdP 的证书签名
- 检查 SAML 索赔中的 UPN 是否与提交的 UPN 相对应
- 检查索赔是否在可接受的时间跨度内发出（什么是“可接受”由你决定）

StoreFront 身份验证 SDK

您可以使用 StoreFront 身份验证软件开发工具包执行证据数据的高级自定义。有关详细信息，请参阅 SDK 中或 https://developer-docs.citrix.com/ 中提供的文档“自定义联合登录服务示例 1811.pdf”。

相关信息

- Citrix 联合身份验证服务文档
- StoreFront PowerShell SDK 文档
- Citrix 联合身份验证服务断言 SDK 来自 https://www.citrix.com/downloads/citrix-virtual-apps-and-desktops/

XenApp 和 XenDesktop 7.6 到当前版本的 HDX 策略模板

May 20, 2020

XenApp 和 XenDesktop 包含可简化对用户的部署的 HDX 策略模板。本文档提供了使用这些模板创建策略时的设计注意事项。我们还提供了规划指导，帮助您确定给定使用案例的正确设置。

本文档的目标受众是熟悉 HDX 概念、策略模板和产品早期版本的高级 Citrix 管理员。

本文档不会取代关于XenApp 和 XenDesktop 策略的综合产品文档。

XenApp 和 XenDesktop 支持随产品和自定义模板一起提供的内置模板 Citrix 支持站点。本文档重点介绍内置模板。

安全和控制模板

在风险容差较低的环境中使用此模板，以最大限度地减少在 XenApp 和 XenDesktop 中默认启用的功能。此模板包含禁用用户设备访问

- 印刷
- 剪贴板
- 外围设备
- 驱动器映射
- 端口重定向
- Flash 加速

应用安全和控制模板可能会占用更多带宽并降低每台服务器的用户密度。

高服务器可扩展性模板

应用此模板可节约服务器资源。此模板可以平衡用户体验和服务器可扩展性。它可以提供良好的用户体验，同时增加单个服务器上可以托管的用户数。在其他设置中，此模板启用 Thinwire 兼容模式（不使用视频编解码器），并阻止服务器端视频渲染。

您可以使用此模板提供每台服务器的最大用户密度。此设计适用于运行新版操作系统（例如 Windows 8、Windows 10 和 Windows Server 2012 R2）的 VDA。Citrix 还为 Windows 7 和 Windows 服务器 2008 R2 提供了一个单独的模板，后缀为 “-传统操作系统”。

此模板的功能

禁用使用视频编解码器压缩图形。仅此更改就可以提高每个服务器的用户密度，同时对服务器渲染图形进行交易。大多数用户应用程序不会受到此更改的影响，同时减少服务器渲染的多媒体播放体验。为了进一步提高密度，此模板中的设置会阻止服务器在默认 Windows 应用程序上渲染多媒体播放，同时尽可能允许 Citrix 重定向技术。

如何使用此模板

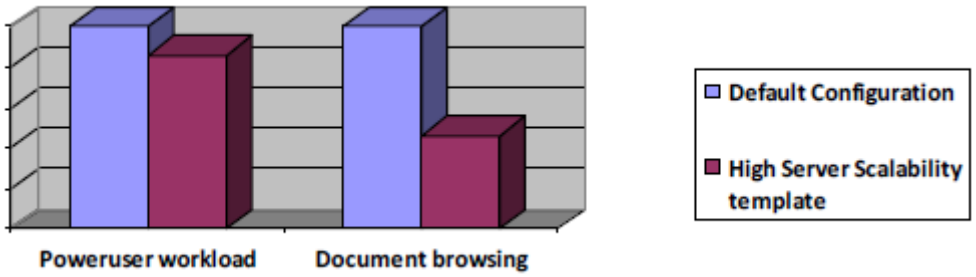
最佳做法是将从此模板创建的策略应用于服务器中的所有用户。您可以通过应用筛选到所需用户的更高优先级策略来设置异常（例如不同的打印设置，甚至高级图形模式，如 Framehawk 或 DCR）。

使用此模板时的注意事项

- 简单图形任务工作（办公套件等）没有客户端先决条件与使用此模板创建的策略一起使用。
- 为了适应用户影响最小的多媒体播放重定向，请使用安装了最新版本的 Receiver 的 Windows 或 Linux 客户端设备，该版本将提供最佳多媒体重定向支持。iOS 和 Mac 将获得有限的多媒体播放选项。此外，确保用户将 Internet 资源管理器和 Windows Media Player 作为默认应用程序。
- 为了适应用户影响最小的多媒体播放重定向，请使用安装了最新版本的 Receiver 的 Windows 或 Linux 客户端设备，该版本将提供最佳多媒体重定向支持。iOS 和 Mac 将获得有限的多媒体播放选项。此外，确保用户将 Internet 资源管理器和 Windows Media Player 作为默认应用程序。
- 如前所述，服务器渲染的多媒体播放可能不是最佳的，也可能根据所使用的应用程序和媒体类型而被阻止。
- 该模板禁用某些个性化和图形效果，如桌面壁纸和菜单动画。虽然操作系统中可能存在更多优化，但这些优化可能会导致不希望的可扩展性影响，并且只有在测试并与此模板中的设置进行比较后才应用。其中包括 “拖动时显示内容”，传统上在远程访问场景中禁用，但在模板上启用。由于此模板中使用的 Thinwire 兼容模式的行为，禁用此选项不会有助于模板的整体性能。
- 产品添加（例如 Lync Optimization Pack 和 Citrix 通用打印机服务器）可增强用户体验，并有可能提高用户密度。
- 除了每台服务器的用户密度增加之外，此模板的使用还可能减少简单 GUI 的每个会话所需的带宽（如 Microsoft Office）。
- 打印配置为仅映射默认客户端打印机（防止每个会话自动映射多个客户端打印机），并使用 Citrix 通用打印机驱动程序。实施这两种设置可以减少会话建立和断开连接过程中的处理。

- 在某些情况下，使用桌面组合重定向 (DCR) 可以帮助提高每个服务器的用户密度。在此模板中不推荐使用此图形模式，因为它仅与 Windows 7、8 和 8.1 VDA 和 Windows 或 Mac Receiver 兼容。

使用高服务器可扩展性模板时的 **CPU** 节省



VDA：单个会话 1920x1080，Windows 10 32 位，2 GB RAM 2vCPU@3.2GHz

免责声明：这是一个示例比较。实际节省取决于特定用户工作流程。

在对每个模板进行说明和注意事项之后，将提供一个指示性比较，以便使用模板可视化资源消耗节省的情况。这些不是作为性能基准，因为它们基于使用 LoginVSI 4.1“超级用户”工作负载的单个会话进行的简单测试。请使用组织中典型的工作负载进行测试，以确定与该环境相关的系统可扩展性。

高服务器可扩展性 — 旧式操作系统

此高服务器可扩展性模板仅适用于运行 Server 2008 R2 或 Windows 7 及更早版本的 VDA。此模板依靠对这些操作系统较为有效的旧图形模式。

此模板用于在具有 Windows 7 和 Windows Server 2008 R2 操作系统的 VDA 上每台服务器的最大用户密度。它使用针对这些操作系统优化的旧版 Thinwire 图形模式，并提供的结果类似于 XenApp 6.5 和 XenDesktop 5.6 中的结果。

如何使用此模板

旧图形模式是计算机策略，应该应用于服务器上的所有会话。通过应用筛选到所需用户的更高优先级策略，可以实现图形以外的设置（如不同的打印设置）的例外。

此模板的功能

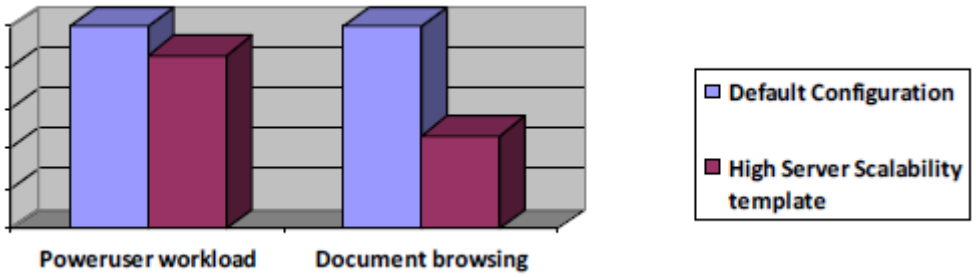
除了“旧图形模式”和“拖动时显示内容”之外的所有其他设置都与高服务器可扩展性模板相同。所有相同的注意事项均适用，除非以下注意事项具体说明。

有关 Thinwire 传统模式的深入描述，请参阅 Citrix 产品文档 站点。

使用此模板时的注意事项

- 禁用 VDA 软件光栅化器（7.x 功能）时，管理员可以期待与 XenApp 6.5 和 XenDesktop 5.6 类似的用户体验和可扩展性。有关详细信息，请参阅 扩大 XenDesktop 中的 3D 应用程序显示兼容性 — 将黑名单列入黑名单，一种临时解决方法。
- 如果适用，使用 DCR 还可以提高每台服务器的用户密度。

使用高服务器可扩展性时节省 **CPU —** 旧式操作系统模板



VDA：单个会话 1920x1080，Windows 7 32 位，2 GB RAM 2vCPU@3.2GHz

免责声明：图形是一个示例比较。实际节省取决于特定用户工作流程。

针对广域网模板进行了优化

此模板适用于使用共享 WAN 连接的分支机构的任务工作人员，或者具有低带宽连接的远程位置访问具有图形简单用户界面且视频内容很少的应用程序（适用于 Thinwire 兼容模式）。此模板通过降低视频播放体验和某些服务器可扩展性实现最佳带宽效率。

此模板旨在改善用户使用低带宽连接访问具有图形简单用户界面的应用程序时的用户体验。模板设计适用于运行现代操作系统（如 Windows 10 和 Windows Server 2012 R2）的 VDA，则为 Windows 7 和 Windows Server 2008 R2 提供了带有 “- Legacy OS” 后缀的单独模板。

此模板的功能

此模板禁用视频编解码器压缩图形。此更改非常有效地降低 Office 应用程序的带宽要求，但它可能会降低服务器渲染的视频质量，并且如果接口是高度图形化的（如 CAD 应用程序），则可能会降低交互性。

该模板允许对 Windows 媒体层和闪存进行所有（默认启用）Citrix 多媒体重定向，并在需要时对正在使用的 WAN 链接进行动态优化 Windows 媒体。

如果用户持续查看多媒体，请勿使用此模板。在这种情况下，使用默认设置或通过使用视频编解码器启用压缩来自定义此模板。

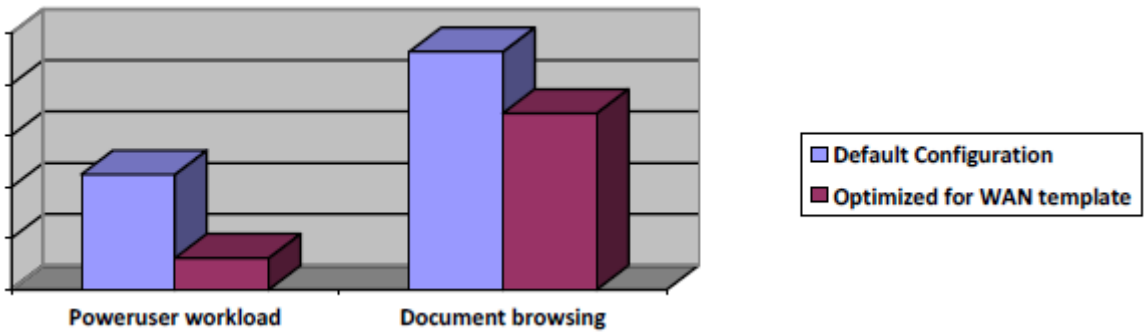
如何使用此模板

您可以将从此模板创建的策略应用于通过描述的 WAN 链接或按用户（具有高策略优先级）为用户提供服务的交付组。用户连接确定策略并使用可用的用户设置策略筛选器，如客户端 IP 地址、NetScaler Gateway 访问条件等。

使用此模板时的注意事项

- 简单图形任务工作（办公套件等）没有客户端先决条件与使用此模板创建的策略一起使用。
- 使用此模板时产生的不良结果可能表明用户持续查看多媒体内容。如果接受此用户行为，请参阅上述建议。
- 服务器渲染的多媒体播放可能不是最佳的，因为帧速率限制在最大目标 16。
- 具有渐变颜色（如天际线）的复杂图形图像不会逐渐变化，而是按步骤变化，因为模板“简单图形的颜色深度”将显示中的颜色子采样设置为每像素 16 位，以减少带宽要求。
- 该模板禁用某些个性化和图形效果，如桌面墙纸和菜单动画。虽然操作系统中可能存在更多优化，但这些优化可能会导致不希望的可扩展性影响，并且只有在测试并与此模板中的设置进行比较后才应用。其中包括“拖动时显示内容”，传统上在远程访问场景中禁用，但在模板上启用。由于此模板中使用的 Thinwire 兼容模式的行为，禁用此选项不会有助于模板的整体性能。
- Lync Optimization Pack 和 Citrix 通用打印机服务器等产品添加增强了用户体验，并有可能提高用户密度。
- 除了针对 WAN 的优化之外，如果会话主要具有简单的 GUI（如 Microsoft Office），则此模板的使用也可能会增加每台服务器的用户密度。
- 打印配置为仅映射默认客户端打印机（防止每个会话自动映射多个客户端打印机），并使用 Citrix 通用打印机驱动程序。模板实现这两种设置，以减少打印所需的带宽。
- 为所有打印机启用通用打印机驱动程序。无论打印机如何，它都可以保证低带宽要求。虽然某些特定于打印机的驱动程序和使用打印服务器可能会产生比通用打印机驱动程序更好的结果，但我们无法将其启用用于一般用途，因为它们需要额外的配置和或测试。
- 禁用与打印服务器的直接连接，并且连接到客户端设备的网络打印机将使用 Citrix 通用打印机驱动程序横向 WAN 并从客户端中断打印作业。执行这些操作是为了管理打印带宽（在 ICA 会话中优化），因为我们无法预测特定于打印机的驱动程序在 WAN 链路上的行为。
- 不建议在受约束的带宽链路中使用桌面组合重定向 (DCR)，在此模板中已禁用。

使用针对广域网优化模板时的带宽节省



VDA：单个会话 1920x1080，Windows 10 32 位，2 GB RAM 2vCPU@3.2GHz

免责声明：这是一个示例比较。实际节省取决于特定用户工作流程。

针对广域网优化 — 旧式操作系统模板

此针对 WAN 优化模板仅适用于运行 Windows Server 2008 R2 或 Windows 7 及更早版本的 VDA。此模板依靠对这些操作系统较为有效的旧图形模式。

此模板旨在改善连接到旧版操作系统时的用户体验，例如 Windows 7 或 Windows Server 2008 R2，具有低带宽连接的 VDA 通过图形简单的用户界面访问应用程序。它使用传统的 Thinwire 图形模式，该模式提供的结果类似于 XenApp 6.5 和 XenDesktop 5.6 中的结果。

如何使用此模板

旧图形模式是计算机策略，适用于服务器上的所有会话。您可以通过配置筛选给所需用户的更高优先级策略，将图形以外的设置（如不同的打印设置）设置为例外。

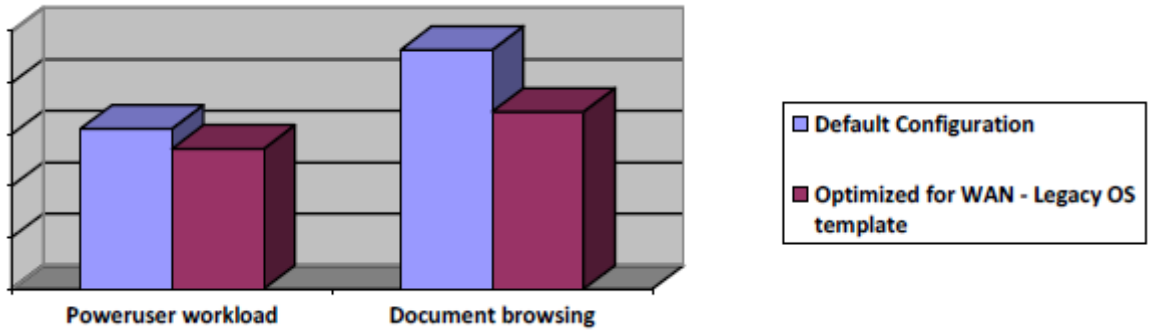
此模板的功能

除了“旧图形模式”、“拖动时显示内容”和“额外颜色压缩”之外的所有其他设置都与“针对 WAN 优化”模板相同。除非下面指定，否则该模板中的所有注意事项均适用。

有关 Thinwire 传统模式的深入描述，请参阅 Citrix 产品文档网站上的 图形策略设置。

使用此模板时的注意事项

- 管理员可以期待与 XenApp 6.5 和 XenDesktop 5.6 类似的用户体验和带宽效率。



VDA：单个会话 1920x1080，Windows 7 32 位，2 GB RAM 2vCPU@3.2GHz

免责声明：这是一个示例比较。实际节省取决于特定用户工作流程。

超高清晰度用户体验模板

此模板强制实施尽可能实现最佳用户体验的默认设置。在按照优先级顺序处理多个策略的场景中使用此模板。

产品配置为提供高清晰度用户体验。仔细查看此模板会强制执行默认值，但“高视觉质量”和“最佳质量打印”除外，后者将这些值设置为高于默认值。

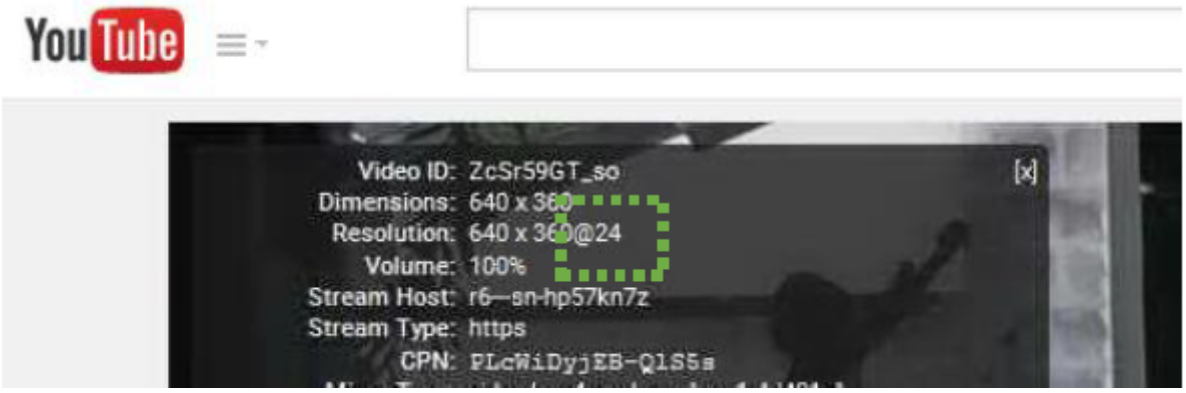
何时使用此模板

使用此模板可以通过使用创建的策略来确保最大的用户体验，该策略优先于具有特定筛选器的其他策略（如 VIP 用户）。

如何使用此模板：使用筛选器创建和应用策略，以选择所需的用户或方案，并且优先级高于从模板创建的其他策略。您可以将此策略与常规用户群结合使用，例如服务器可扩展性策略或 WAN。

使用此模板时的注意事项

- 需要更新后的客户端硬件和 Citrix Receiver。例如，CPU 速度高于 2.0 GHz，支持 h.264 和 Receiver for Windows 4.x/Mac 11.8/Linux 13.0。如果用户设备不符合要求，则使用此模板创建的策略可能会产生不利影响。
- 如内置策略说明中所述，应用此模板可能会占用更多带宽并降低每台服务器的用户密度。
- 考虑使用 VDA GPU 解决方案。高端用户应用程序，无论是图形或具有极端处理需求，都可以利用 GPU 的处理能力，从而实现更好的性能，并在某些情况下更好的服务器可扩展性。有关详细信息，请参阅 Citrix 生产文档中的 HDX 3D Pro。
- 与默认配置一样，如果 Receiver 支持并已启用，则此模板将允许使用视频编解码器 (h.264) 压缩屏幕图形。视觉质量会根据需要由视频编解码器自动调整，并且不需要“视觉质量”设置。
- 高视觉质量设置仅适用于使用此模板而不使用视频编解码器 (h.264) 进行压缩（以增加每台服务器的用户密度），从而导致出现称为 Thinwire 兼容模式的图形模式。管理员可以通过将较高优先级的策略与压缩的视频编解码器“未使用”或客户端没有或已禁用视频编解码器支持（请参阅客户端兼容性矩阵）来实现这一目标。在这种情况下，只有当 VDA 和客户端设备通过不受限制的高带宽 LAN 网络连接时，才能获得非常高的用户体验。
- 此模板还使用 30 fps 的目标帧速率设置，适用于需要高清体验的大多数用例；默认的 30 fps 应视为初始基准，可以调整以满足最终用户的高清体验（默认情况下，YouTube 流以 24 到 30 fps 的速率）。具有高端图形和支持 GPU 的应用程序的用户可以请求高达 60 fps 的更高值）。



- 此模板允许 Windows Media Player 和 Flash 多媒体重定向到 Citrix Receiver for Windows 和 Citrix Receiver for Linux，仅允许 Windows Media Player 定向到 IOS 设备。使用客户端中最新的 Citrix Receiver 来利用最近的改进。
- 该模板配置打印以获得最佳用户体验，包括最佳打印输出质量和所有可用的配置选项。如果用户在其设备上安装了大量打印机，并且制造商打印机驱动程序未针对远程打印进行优化，这可能导致极高的带宽使用率、降低每台服务器的用户密度以及 VDA 可能出现的互操作性问题。

附录

以下是模板中的详细策略设置。

模板	超高清晰度用户体验	高服务器可扩展性	高服务器可扩展性 — 旧式操作系统	WAN 优化	针对广域网优化 — 旧式操作系统
Bandwidth（带宽）					
总会话带宽限制				0 注意：只是为了公开设置	
图形					
旧图形模式	已禁用		已启用		已启用
桌面组合重定向				已禁用	已禁用
使用视频编解码器进行压缩	在可用时使用	请勿使用	不适用	请勿使用	不适用
目标帧速率	30	16	12	16	16
目标最低帧速率	10	8	8	8	8
视觉质量	高	中	不适用	低	不适用
简单图形的首选颜色深度	24bpp		不适用	16bpp	不适用
有损压缩级别	不适用	不适用		不适用	高
允许的最大颜色深度	不适用	不适用		不适用	16bpp
额外颜色压缩	已禁用	已禁用	已禁用	已禁用	已启用
桌面 UI					
桌面墙纸	允许	禁止	禁止	禁止	禁止
拖动时查看窗口内容	允许	允许	禁止	允许	禁止

模板	超高清晰度用户体验	高服务器可扩展性	高服务器可扩展性 — 旧式操作系统	WAN 优化	针对广域网优化 — 旧式操作系统
动态窗口预览	已启用			已禁用	
菜单动画	允许	禁止	禁止		
多媒体					
优化通过 WAN 进行的 Windows Media 多媒体重定向		禁止	允许		
限制视频质量			最大值 480p		
Windows 介质回退防护	未配置	仅在客户端上播放所有内容			
Flash 视频回退预防	未配置	仅限小型内容			
闪存视频回退预防错误 *.swf		应用默认示例（请参阅策略）			
多媒体会议	允许	禁止			
音频					
音频质量	高-高清晰度音频	针对语音进行中等优化	低-适用于低速连接		
打印					
自动创建客户端打印机	自动创建所有客户端打印机	仅自动创建客户端的默认打印机	仅自动创建客户端的默认打印机		
直接连接打印服务器	已启用		已禁用		
通用打印机驱动程序使用 (UPD)	仅当请求的驱动程序不可用时才使用通用打印	仅使用通用打印	仅使用通用打印		
通用打印的打印质量限制	无限制		中等分辨率 (600 DPI)		

模板	超高清晰度用户体验	高服务器可扩展性	高服务器可扩展性 — 旧式操作系统	WAN 优化	针对广域网优化 — 旧式操作系统
通用打印优化默认值	ImageCompress = BestQuality；其他设置 = 默认	ImageCompress = Standard-Quality；其他设置 = 默认	ImageCompress = ReducedQual-ity；其他设置 = 默认		
文件重定向					
使用异步写入	已禁用		已启用		

> 注意：
>
> - 粗体设置等于默认值。
> - 分配给默认值的设置是为了确保使用堆叠策略时所需的结果。
> - 划线框标记不适用于每个模板中的图形模式的设置（在该行中）。
> - 空框表示在该列中没有针对模板的特定建议的设置。
> - 该表不列出所有策略设置，仅列出内置模板中使用的策略设置。有关完整的策略列表，请参阅 Citrix 产品文档 网站。

图形传输注意事项

使用视频编解码器进行压缩

在 XenApp/XenDesktop 7.6 FP3 中，HDX 为提供简单图形（例如，基础 Office 应用程序）提供了增强的支持。我们这样做是为了让我们的客户不断延长已部署的网络链路、客户端设备和锁定接收器的使用寿命。收件人

实现这一目标，我们现在允许管理员控制何时使用视频编解码器对 Thinwire 图形进行编码。

默认情况下，产品仍然启用了视频编解码器使用情况（如果可用），适用于大多数常规使用场景。

当高服务器可扩展性至关重要，并且正在使用“针对 *WAN* 优化”（受约束的带宽链路）时，Citrix 建议禁用视频编解码器的使用。当不使用视频编解码器时，会议可以专注于以最低 CPU 和带宽要求优化文本和简单图形的交付（这是大多数业务应用程序的基础）。

如果带宽受限，服务器渲染的视频性能将不会达到最佳状态。有关详细信息，请参阅 Citrix 产品文档中的 Thinwire 兼容模式。

允许使用 **GDI** 的旧版操作系统（**Windows 7** 和 **Windows Server 2008 R2**）

Citrix 建议启用旧图形模式。旧版 Thinwire 模式的设计适用于旧版 Windows 操作系统的体系结构，对于许多用例来说，它仍然是这些操作系统最优化的图形模式。

拖动时显示窗口内容

与以前版本提供的建议相反，Thinwire 兼容性模式在允许拖动时显示窗口内容时执行最佳操作。

禁用旧式图形模式和更好的可扩展性方案的策略是视频编解码器 (h.264) 压缩是必需的，但需要最大的用户密度.

目标帧速率和目标最小帧速率

对于针对简单图形使用的“高服务器可扩展性和针对 WAN（受约束带宽链接）优化模板，Citrix 建议将目标帧速率设置降低到 16 或 12，并将目标最小值降低到 8。这些设置有助于实现这些模板的所需目标。

我们选择目标帧速率为 16，因为这是人眼检测运动的绝对最低值。

VDA 和客户端不断协商适当的帧速率，以便在会话中交付。通常，帧速率保持在屏幕上显示更改所需的最低值。当检测到高动作时，从视频播放到窗口拖动到滚动的任何内容，会话将尝试提供最高达目标帧速率的每个屏幕更改。在受限带宽连接中，可能无法保持目标帧速率，VDA 会自动平衡增加屏幕图形压缩和降低帧速率，直到达到目标最小帧速率，然后增加压缩直到预设（不可配置）值，最后根据需要进一步降低帧速率。将其视为自动自适应显示（先前的 Citrix 产品中使用的一种技术）。

> 注意：将视觉质量设置为低不会影响高对比度文本（例如黑色与白色），并且仍以高品质传递。

颜色深度

与传统模式不同，XenDesktop 和 XenApp 7.x 超级编解码器无法控制颜色深度，并且每像素接收 24 位。在 FP3 中，作为增强 Thinwire 而不使用视频编解码器进行压缩的一部分，我们添加了以 16 位每像素发送会话图形的选项（由简单图形设置的首选颜色深度控制）。此选项可减少简单图形所需的带宽，而且只有在使用颜色渐变时才会明显显示出来。在这种情况下，设备使用的服务器 CPU 消耗可能略高一些。

对于传统图形模式，在用于可扩展性和 WAN 的旧式操作系统模板中，设置 允许的最大颜色深度也限制为 16 或 12。请注意，使用旧图形模式时，可以根据其他条件请求或提供较低的颜色深度。

关于 **DCR** 的一个词 **

桌面组合重定向 (DCR) 是 Citrix 在 XenDesktop 5.5 中引入的显示虚拟通道。虽然此技术具有许多优势，但它目前适用于 Windows 7、8 和 8.1 VDA，减少了 Citrix Receiver 的支持，主要是客户采用率低。此外，低带宽 WAN 链路不建议使用 DCR。因此，从 FP3 开始，此虚拟通道默认处于禁用状态，并且针对广域网优化模板将主动禁用该虚拟通道。

在 **HDX** 图形模式下进一步阅读

有关如何检查正在使用的图形模式的信息，请参阅 Citrix 支持知识中心中的 如何确定 HDX 显示模式。

多媒体考虑因素

在所有模板中，默认情况下，在产品上，允许重定向多媒体播放。以下是模板中使用的其他设置：

优化通过 **WAN** 进行的 **Windows Media** 多媒体重定向

默认情况下允许对媒体内容进行即时转码以便在 WAN 上高效传输是进程密集型的，因此在“高服务器可扩展性”模板中禁止使用。请注意，如果服务器可以使用 NVIDIA GPU，并且启用了以下设置：使用 **GPU** 优化通过广域网的 **Windows** 媒体多媒体重定向，则可以将处理负载卸载到 GPU。

限制视频质量

此设置仅适用于 Windows 媒体多媒体重定向的优化。在使用它的“针对 WANE 优化”模板中选择了一个相当于小型嵌入式视频播放器的值作为初始建议。否则，未配置此设置。

多媒体会议

从客户端重定向网络摄像头（在 VDA 中运行的统一通信或会议应用程序中使用）将增加所需的服务器资源。此功能在“高服务器可扩展性”模板中被禁用。

有关多媒体和 Flash 重定向策略设置的详细信息，请参阅 Citrix 生产文档站点中的以下主题：

- 多媒体策略设置
- Flash 重定向策略设置

音频注意事项

该产品附带默认的高品质音频（约 128 Kbps）。如果从此模板创建的策略优先于其他策略，则此值也适用于“非常高”定义模板。

下表中列出的值分别用于音频方向（输出和输入）。

模板	超高清体验和默认设置	高服务器可扩展性	WAN 优化
音频质量设置	高	中	低
使用的预期带宽	128 Kbps	60 Kbps	44 Kbps

打印注意事项

XenApp 和 XenDesktop 包括一个通用打印机驱动程序，可用于大多数客户端连接的打印机。默认情况下，仅当 VDA 找不到特定于打印机的驱动程序时，VDA 才使用此驱动程序。此外，默认情况下，所有客户端连接的打印机都会默认映射到会话中。

新的内置模板使用以下不同于默认值的打印设置：

仅自动创建客户端的默认打印机

高服务器可扩展性和针对广域网进行优化 — 创建一台打印机而不是可能的多台打印机可确保在这两种情况下节省成本。

用户可以在会话中更改默认的客户端打印机，并且即使在双跳场景中，映射的打印机也会更新。

仅使用通用打印机 [驱动程序]

高服务器可扩展性 — 防止 VDA 通过每个连接搜索打印机驱动程序，从而节省了由于预打印机型号驱动程序而导致的磁盘 I/O 操作和服务器负载。

针对 WAN 进行优化 — 通用打印机驱动程序可以保证低带宽要求，而不考虑打印机。虽然某些特定于打印机的驱动程序和使用打印服务器（需要额外的设置）可能会产生比通用打印机驱动程序更好的结果，但我们不建议使用它，因为它们需要额外的配置和测试。

直接连接到打印服务器

默认情况下启用此设置允许直接从会话访问客户端中配置的网络打印机。启用此设置可能会改进用户可用的选项（具体取决于打印机），并节省客户端设备上的流量。

对于 WAN 模板优化，此设置被禁用，因为网络打印机可能与用户并置（他必须检索打印输出?），因此我们可以确保使用 Citrix 通用打印机驱动程序打印时，打印作业大小会很小。此外，打印作业使用的带宽将遵守会话带宽限制。

SQL Server 和 Citrix 数据库

May 20, 2020

Microsoft SQL Server 是任何 Citrix Virtual Apps and Desktops 部署的重要组成部分。规划和了解 Citrix SQL 交互非常有益于您和您的组织维护健康且性能良好的 Citrix 环境。缺乏 SQL Server 高可用性和充足的计算资源会对 Citrix 基础结构的用户体验和正常运行时间产生负面影响。

数据库摘要

在 Citrix Virtual Apps and Desktops 部署期间，需要/创建的 3 个数据库：

站点：（也称为站点配置）存储正在运行的站点配置，以及与代理相关的动态数据，如当前会话状态、连接、加载和 VDA 状态信息。

配置日志记录：（也称为日志记录）存储有关站点配置更改和管理活动的信息。启用“配置日志记录”功能时使用此数据库（默认值 = 启用）。

监视：存储 Director 使用的数据，如会话和连接信息。

在早期版本的 Citrix Virtual Apps and Desktops 中（例如 XenApp 和 XenDesktop 7.6），在初始站点配置期间（通过 Studio 或在 SQL Server 上运行脚本），Citrix Virtual Apps and Desktops 所需的数据库已创建为一个数据库。安装完成后，管理员可以将其拆分为不同的数据库，以提高性能或遵守备份/安全准则。

使用较新版本的 Citrix Virtual Apps and Desktops，您可以在初始站点配置期间创建数据库，也可以通过 Studio 或在 SQL Server 上运行脚本。您的数据库会自动拆分为三个独立的数据库。

对于具有大型监视数据库的环境，理想的配置是在站点配置和配置日志记录数据库不同的服务器上托管监视数据库。它会记录更多的数据，更频繁地发生更改，并且数据不会像其他数据库那样重要。有关详细信息，请参阅 数据库大小调整指南 中的 VDI 手册 或第 97 页。

长期服务版本 (LTSR) 的每个累积更新 (CU) 包含 SQL 数据库体系结构的修补程序。例如，请参阅 CTX230536。为了最好地保护您的环境免受意外问题的影响，请确保您有一个定期将环境升级到最新的 CU 的过程。此外，请确保适当的 SQL Server 和数据库监视到位，以捕获故障事件和具有高资源利用率和可用空间的问题。

Citrix 与 SQL 交互

Citrix Virtual Apps and Desktops 代理将数据库用作消息总线，用于代理通信、存储配置、监视和审核数据。这些数据库不断使用，并可能消耗 SQL Server 上的大量计算资源。

例如，资源枚举（标识并向用户显示的资源）、资源启动和会话启动阶段需要 Citrix Delivery Controller 与 SQL Server 交互。

枚举：通过 Citrix ADC 和 StoreFront 成功进行身份验证后，Delivery Controller 会联系 Citrix 站点数据库，以根据 AD 凭据检查哪些应用程序可供用户使用。识别资源后，会从数据库中提取其他信息，例如应用程序名称、桌面、图标。

启动：当用户选择要启动的应用程序或桌面时，StoreFront 会向 Delivery Controller 发出启动请求。然后，Delivery Controller 与 SQL Server 上的站点数据库联系，以选择要将用户发送到的 VDA。

会话初始化：会话启动后，VDA 与 Delivery Controller 联系，将会话信息写入站点数据库。

数据库建议

要确保 SQL Server 中断对 Citrix Virtual Apps and Desktops 基础结构的影响最小，客户可以从以下 Citrix 支持的高可用性选项中进行选择：

- AlwaysOn 可用性组
- AlwaysOn 故障转移群集
- Basic 可用性组
- Hypervisor 高可用性 *

> 注意：
>
> 虽然 Citrix 支持 Hypervisor 高可用性，但不建议在托管 EHR 应用的环境中使用它，因为正常运行时间至关重要。

Citrix 和 Epic 建议对所有三个数据库使用相同的高可用性方法，即使在建立最终用户会话时不需要配置日志记录和监视数据库可用性。例如，如果计划使用 SQL 始终在线可用性组作为 HA 策略，请将其用于所有三个数据库对象。

我们还建议您对 Citrix 数据库本身（尤其是站点数据库）进行完整的每日备份。保留期根据组织要求而有所不同，但通常需要维护七天的完整备份和至少一个月的每周备份。事务日志备份计划应基于组织的标准和事务日志相对于必须分配的可用存储量的增长率的组合。确保监视 SQL Server 上的可用存储。

使 Citrix 数据库的恢复模型与您正在采用的高可用性方法的要求保持一致。

根据 Microsoft 的建议，客户应将维护计划设置为每晚和每周运行以维护数据库索引。维护计划可能只是在一周内夜间重新组织索引，并在周末重建索引。

此建议可避免在日常操作中重建任何大型索引造成的任何性能影响，尤其是对于大型监视数据库。

Microsoft 建议重新构建索引，如果索引的碎片大于 30%，如果小于 30%，则重新组织索引。请参阅数据库维护部分。

本地主机缓存

要考虑数据库不可用的情况，Citrix 将本地主机缓存 (LHC) 功能添加到 Citrix Virtual Apps and Desktops 7.x 平台（7.12 及更高版本，包括 XenApp 和 XenDesktop 7.15 LTSR）中。启用此选项允许发布应用程序的用户在 Delivery Controller 与 Citrix 站点配置数据库之间的通信中断时进行连接。如果 SQL 是在高可用性体系结构（例如“始终打开”、“镜像”或“群集”）中配置的，则当发生完全 SQL 中断或网络连接中断时，此功能可提供额外的容错能力。

这不应被视为 SQL 高可用性的替代方法，因为站点管理功能在 SQL 中断期间不可用，故障转移过程也不是瞬时的。在 SQL 中断的情况下，代理功能会丢失，直到它转换为 LHC 并且 VDA 重新注册为止。当恢复 SQL 连接性/可用性时，转换回正常操作模式时，也会遇到此方案。

本地主机缓存会在每个 Delivery Controller 上的本地 SQL Express LocalDB 数据库中保留静态站点数据的副本，并在数据库中断期间依赖此数据来持续支持 VDA 注册和会话代理请求。

本地主机缓存设计注意事项

由于 Epic 社区成员部署的大小不同，建议您与 Citrix 密切合作，以确定使用 LHC 所需的其他资源。

- 可扩展性注意事项
  - XenApp 和 XenDesktop 7.15 中记录的 LHC 最大限制为单区域中的 10000 个 VDA 和多区域部署中的 40000 个 VDA。在 Citrix Virtual Apps 环境中，LHC 和区域可扩展性取决于登录速率和用户计数。因此，在您的环境中观察到的实际可扩展性可能低于已发布的最大值。对于此体系结构，如果您的预期会话计数超过 10000 和/或您的登录速率大于每秒 10 个用户，我们建议您考虑其他区域。
- Delivery Controller 大小：当 LHC 处于活动状态时，每个区域所选的主 Delivery Controller (DC) 处理所有 VDA 注册、枚举、启动和更新。
  - RAM：本地主机缓存服务可以消耗 2 GB 以上的 RAM，具体取决于停机的持续时间和用户在中断期间启动的次数。
  - CPU：由于选定的 DC 上的额外 CPU 负载，应考虑额外的内核来补偿。

Controller 的 CPU 配置，尤其是可用于 SQL Server Express LocalDB 的核心数，直接影响本地主机缓存性能，甚至比内存分配还要严重。仅在数据库不可访问且 High Availability Service 处于活动状态时，在中断期间观察此 CPU 开销。

虽然 LocalDB 可以使用多个核心（最多 4 个），但它仅限于单个插槽。添加更多套接字，例如，具有 4 个插槽，每个插槽都有 1 个内核，不会提高性能。相反，Citrix 建议结合使用多个套接字和多个核心。在 Citrix 测试中，2x3（2 个套接字，3 个核心）配置提供的性能优于 4x1 和 6x1 配置。

- 存储：在本地主机缓存模式下，假定平均每秒 10 次登录，存储使用率大约每 2 至 3 分钟增加 1 MB。存储消耗量相对于登录速率而言会增加。有关详细信息，请参阅 本地主机缓存 一文。

数据库中断的影响

如果数据库完全中断，几乎所有的关键 Delivery Controller 功能都会受到影响，这突出说明了设计和实施推荐的 SQL HA 策略的重要性。下表列出了这些效果：

组件	数据库中断的影响
站点配置数据库	用户无法连接或重新连接到虚拟桌面。注意：本地主机缓存 (LHC) 允许具有托管共享桌面、托管 Windows 和浏览器应用程序以及个人桌面的用户重新连接到其应用程序和桌面，即使站点数据库不可用也是如此。在 LHC 模式下，不会收集监视数据，并且无法对站点进行配置更改。
监视数据库	Director 不显示任何历史数据，Studio 无法启动。传入用户请求和现有用户会话的代理不受影响。
配置日志记录数据库	如果在 Citrix Virtual Apps and Desktops 日志记录首选项中启用了数据库断开连接时允许更改，则配置日志记录数据库的中断不会产生任何影响（不包括未记录配置更改）。否则，管理员无法对 Citrix Virtual Apps and Desktops 进行任何更改。

SQL 大小建议

SQL Server 的大小必须正确，以确保环境的性能和稳定性。由于每个 Citrix 产品都以不同的方式使用 SQL Server，并且每个客户都有不同的使用模式，因此无法提供通用的包罗万象的大小调整建议。相反，下面提供了每个产品 SQL Server 的大小调整建议，并且在部署期间应仔细监视性能以验证大小假设。

对于仅承载与 Citrix 相关数据库的 SQL 环境，应为 SQL Server 配置至少 4 vCPU 和 8 GB RAM，最多可供 10000 名用户使用。对于较大的部署或登录速率较高的部署，我们建议至少使用 8 vCPU 和 16 GB RAM。有关 Citrix Virtual Apps and Desktops 7.x 部署的 SQL 数据库大小调整概念的详细信息，请参阅Citrix XenDesktop 7.x 数据库大小

调整。本文还包括有关工作负载特征的信息，如预计的事务日志增长率。

请记住，监视数据库的大小因数据保留设置而异。向产品中添加捕获精细的 VDA 性能数据功能后，XenApp 和 XenDesktop 7.15 LTSR 具有比 7.6 LTSR 更多的选项。有关配置这些设置的详细信息，请参阅 监视策略设置 并将其纳入数据库大小计算中。

CU 更新和修复

Citrix 每年发布几次 CU，用于 Citrix Virtual Apps and Desktops LTSR。这些 CU 仅包含安全更新和错误修复，没有引入新功能。Citrix 建议运行最新的 CU，因为它们可以修复产品中发现的问题。其中一些修复程序与 SQL 相关。它们解决了 Citrix 或我们的客户发现的锁定、死锁、存储过程等问题。例如，XenApp 7.6 CU 中有许多与 SQL 相关的修补程序，直到 CU5。建议查看每个 CU 的“固定问题”部分，并在页面中搜索 **SQL** 。

- The connection between the Delivery Controller and the SQL Server might be lost intermittently due to a deadlock in the SQL database. [#LC8477]

> 注意：
>
> LC8477 是在 7.6 CU5 和 7.17 中发布的

其他参考资料

- XenApp 和 XenDesktop 7.15 LTSR
- VDI 手册
- Citrix 数据库大小调整
- Citrix 本地主机缓存

供稿人：*enry Vernov*，首席系统工程师。

针对 **XenApp** 和 **XenDesktop** 的组策略管理模板更新

May 20, 2020

模板是从预定义的起点创建策略的源。您可以导入或导出它们。内置 Citrix 模板已针对特定环境或网络条件优化，可以用作：

- 用于创建自己的策略和模板以便在站点之间共享的源。
- 一个参考，用于更轻松地比较部署之间的结果，因为您可以引用结果，例如“... 使用 Citrix 模板 x 或 y 时...”
- 通过导入或导出模板与 Citrix 支持或可信第三方传递策略的方法。

关于这篇文章

本文包含的链接可用于下载其他模板和更新到 XenApp 和 XenDesktop 组策略管理包内置的 Citrix 模板。

虽然您仍然可以在 Citrix Studio 和 GPO 编辑器中找到 Citrix 提供的模板（安装组策略管理包后），但我们希望能够快速为您提供超出安装程序包中包含的方案的更新和模板。

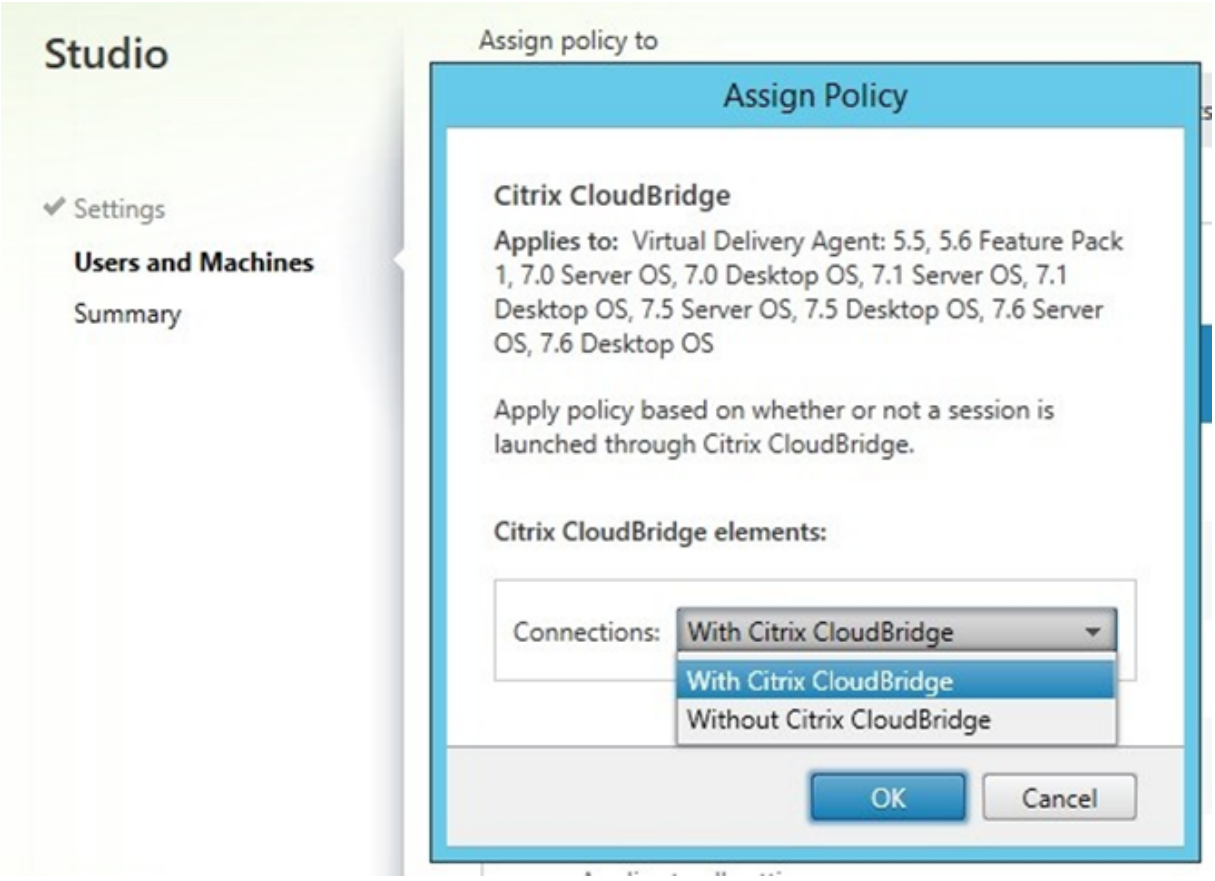
除非另有指定，否则本文中的模板应与最新版本的 XenApp 和 XenDesktop 一起使用。虽然它们的名称和描述是英语，但您可以导入它们并以其他语言使用它们。

模板

CloudBridge WAN

此模板是对内置 Citrix 模板的更新，旨在最大限度地发挥 CloudBridge 广域网加速的优势。它配置允许 XenApp 和 XenDesktop 在客户端和 VDA 之间路径中最大限度地发挥 CloudBridge 广域网加速设备的优势的设置。

可以使用“使用 Citrix CloudBridge”策略筛选器将此模板专门分配给具有 CloudBridge 的会话，从而允许对 CloudBridge 和其他连接使用不同的策略设置。



- 下载模板
  注意：导入后，它将显示为自定义模板。

Citrix Receiver for Chrome

此模板显示了一些可用的策略设置，这些策略设置对 Chromebook 端点的会话具有独占效果或特殊效果。

- 下载模板

Citrix Receiver for HTML5

此模板包括使用 Citrix Receiver for HTML5 的会话的适用设置。在此版本的 Citrix Receiver 中，您需要启用 WebSocket，因为在默认安装 XenApp 和 XenDesktop 时，它将被禁用。

- 下载模板

改善不可靠网络中的用户体验

此模板启用了基于 Framehawk UDP 的图形协议，并公开了一些对这些会话具有独占或特殊影响的可用策略设置。

- 下载模板

直通（双跃点）应用程序会话

在具有托管/池 /RDS 托管虚拟桌面的部署中，通常使用到 XenApp 的直通会话为用户提供不在桌面映像中的应用程序。此模板适用于向桌面提供应用程序的 XenApp 部署。

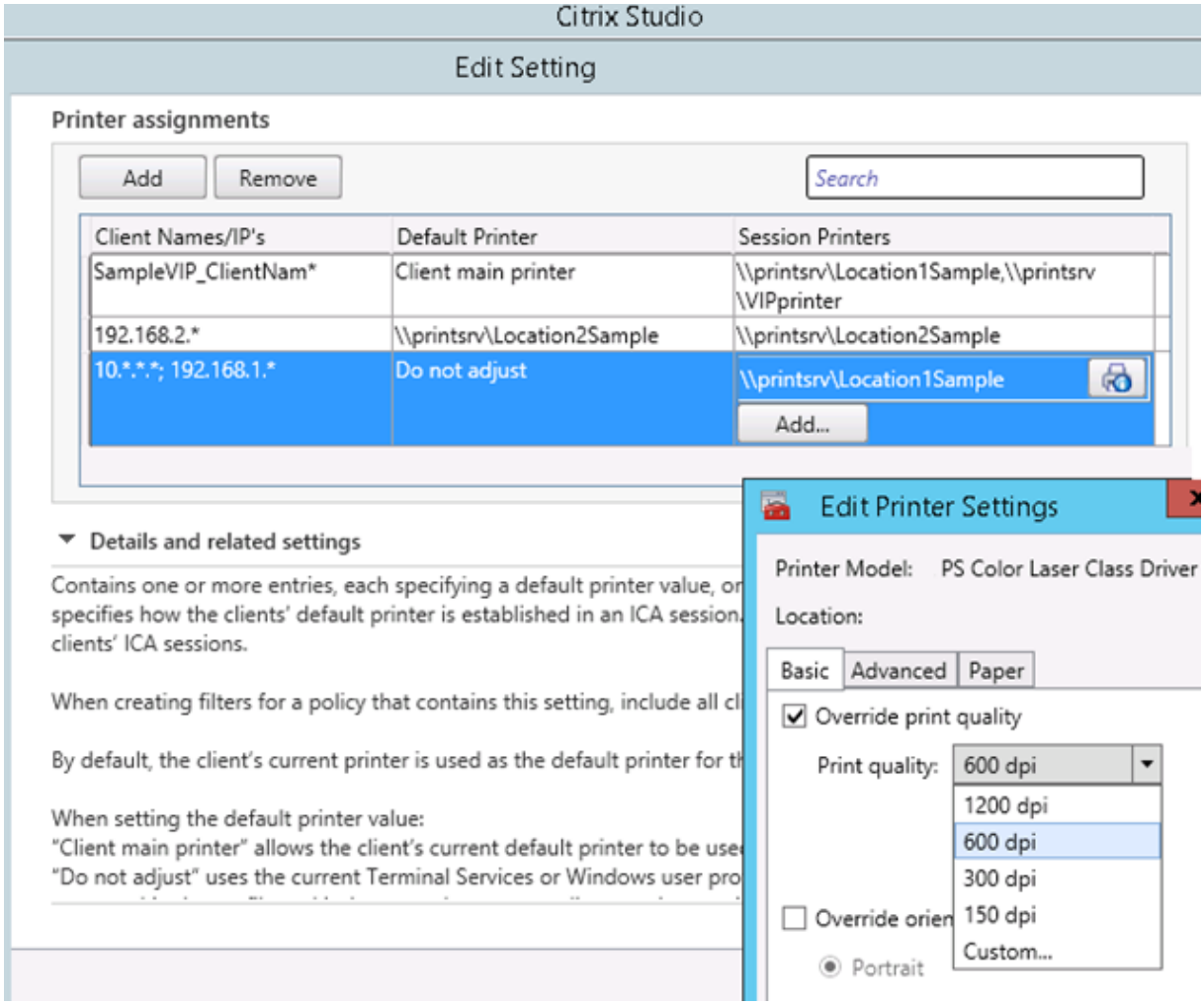
它应该降低应用程序会话服务器中的 CPU 使用率，同时允许桌面会话优化传输到客户端端点设备。您可以通过禁用用于图形压缩的视频编解码器并将压缩图形从应用程序传送到桌面来实现这一目标。

重要提示：使用此模板可增加应用程序和桌面之间的流量。

- 下载模板

Citrix 通用打印服务器

这是一个示例模板，旨在用作配置 Citrix 通用打印服务器部署的起点。生成的策略允许使用特定于打印机的驱动程序（如果先前由 VDA 管理员安装）；否则，使用 Citrix 通用打印机驱动程序。它还列出了示例打印机分配（会话打印机和默认打印机设置的新替代方案），这些分配应替换为适用于您的部署的正确分配。此模板需要对打印机进行域访问才能正确配置。



- 下载模板

以前版本的模板

以下链接包含 7.6FP3 之前版本的模板。这些模板在导入时显示为自定义模板。

- 下载模板

免责声明

上述示例和下载的策略模板（示例代码）按原样提供给您，没有任何类型的声明、保证或条件。您可以自行承担使用、修改和分发它们的风险。CITRIX 不承诺任何形式的明示、默示、书面、口头或法定的保证，包括但不仅限于适销性、特定目的适用性、所有权和非侵权性。在不限制上述内容的通用性的情况下，您承认并同意：(a) 示例代码可能出现错误、设计缺陷或其他问题，可能导致数据丢失或财产损坏；(b) 示例代码可能无法充分发挥作用；(c) Citrix 可能在恕不另行通知的情况下或对您负有责任，停止提供当前版本和/或示例代码的任何未来版本。在任何情况下都不应使用本代码来支持极度危险的行为，包括但不仅限于维持生命或爆破行为。在违反合同或任何其他责任理论的情况下，CITRIX 或其附属公

司或代理人均不对因使用示例代码而产生的任何损害承担责任，包括但不限于直接、特殊、附带、惩罚性、后果性或其他损害，即使被告知这种损害的可能性。尽管代码中的版权属于 Citrix，但代码的任何分发都应仅包含您自己的标准版权归属，而不是 Citrix 的版权归属。您同意保护 Citrix 并使 Citrix 免于承担因您使用、修改或分发本代码导致的任何和所有索赔。

* 本文最初出现在 Citrix 支持知识中心中。

XenApp 和 XenDesktop 7.11 至当前：延迟和 SQL 阻止查询改进

May 20, 2020

文章 XenApp 和 XenDesktop 7.7：区域、延迟和代理性能 提供了有关延迟代理的性能信息。本文介绍了来自 XenApp 和 XenDesktop 7.11 的延迟代理的改进。它还介绍了在 VDA 注册时防止死锁的改进。

通过延迟改进代理

在 XenApp 和 XenDesktop 7.11 中，我们重新讨论了核心代理 SQL 代码，该代码确定哪个 VDA 负载最少，然后向该 VDA 发送启动请求。我们决定从“完美”负载平衡算法切换到“足够好”负载平衡算法。

在 XenApp 和 XenDesktop 7.11 之前，代码会查找负载最少的 VDA，并会锁定/阻止其他启动请求，直到该 VDA 变为可用。这阻止了所有其他经纪请求。

在 XenApp 和 XenDesktop 7.11 中，代码会查找当前未锁定的负载最少的工作程序。这意味着，虽然我们可能无法获得负载最少的工作人员（也许我们只得到第二或第三个最少的工作人员），但我们可以在不锁定所有其他启动请求的情况下执行此操作。如果我们找不到解锁的工作人员，我们坐下来等待锁。如果有足够的 VDA，很少发现所有这些都同时锁定，但是当发生这种情况时，行为与以前的算法相同。

在某些情况下，管理员可能会注意到负载平衡方面的略有差异，但需要密切注意，以便注意我们不使用负载最少的 VDA。

核心代理代码中还有其他位置，SQL 阻塞问题已得到改进。Citrix 建议大型站点使用 7.13 或 7.6 CU3 代理来实现所有当前已知的改进。

绩效结果

下表向上一篇文章中的数据添加了两个数据点，以显示随着延迟的改进而产生的代理：

产品版本	**7.7**	**7.11+**	**7.7**	**7.7**	**7.11+**
延迟时间（毫秒）	90	90	250	250	250
并发请求	48	48	36	48	48

产品版本	**7.7**	**7.11+**	**7.7**	**7.7**	**7.11+**
平均响应时间（秒）	12.9	3.7	26.7	不适用	7.6
每秒钟的代理请求	3.7	12.6	1.3	不适用	6.3
错误 (%)	0	0	4.6	42.8	0
启动 1 万个用户的时间	44 分 55 秒	13 分 10 秒	2 小时 3 分	不适用	26 分 27 秒

正如您所看到的，在 250 毫秒的延迟时间内，XenApp 和 XenDesktop 7.11 现在的性能优于 7.7 代码，以 90 毫秒的速度。因此，我们不会花时间测试大量的数据点，而是测试之前失败的数据点。您可以看到，如果使用 7.11 或更高版本，用户将体验更快的资源代理，即使代理和 SQL Server 之间的延迟。

使用 LTSR 7.6 CU3 控制器的客户也可以从同样的改进中获益。虽然我们不希望 LTSR 7.6 CU3 以延迟的方式部署，但这些更改仍然可以提高性能，即使没有延迟 — 我们知道，一些客户确实拥有 LTSR 7.6 CU3，而且有些延迟。

注册风暴序列化

不幸的是，我们知道有锁的一个领域是 VDA 注册。锁定的原因是在注册工作人员时避免死锁。我们现在对死锁的原因有了更好的了解，这是由于没有在多个注册线程上以一致的顺序锁定工作人员的会话。我们现在按会话 ID 执行会话锁定，这会停止 VDA 注册死锁。

我们已经在内部测试了这种行为变化，并发现它有助于解决重新注册规模测试中的一些问题。但是，由于某些客户的环境非常复杂，我们没有完全删除此锁，以便有时间进行更多的测试。相反，我们为使用 XenApp 和 XenDesktop 7.12 或更高版本的客户提供了有关此锁的使用情况的调整。此可调整位于 XenApp 和 XenDesktop 7.12 数据库的 CHB_config. 站点表中：

```
1     select SerializeMultiSessionAudits,
          SerializeMultiSessionDeregistrations from chb_config.Site
2
3     SerializeMultiSessionAudits SerializeMultiSessionDeregistrations
4
5     --------------------------- ------------------------------------
6
7     1 1
8  <!--NeedCopy-->
```

您可以将这些标志设置为 0 以删除锁的使用：


```
    update chb_config.Site set SerializeMultiSessionAudits=0,
       SerializeMultiSessionDeregistrations=0

    select SerializeMultiSessionAudits,
       SerializeMultiSessionDeregistrations from chb_config.Site

    (1 row(s) affected)

    SerializeMultiSessionAudits SerializeMultiSessionDeregistrations

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

    0 0

<!--NeedCopy-->
```

由于 XenApp 和 XenDesktop 7.15，默认为禁用此锁定。同时升级到 XenApp 和 XenDesktop 7.15 或更高版本也会禁用锁定。为需要重新启用锁定的客户提供可调节功能。

这篇文章是从 Chris Gilbert 撰写的博客文章修改而来的。要阅读原始博客并查看评论，请转到 *https://www.citrix.com/blogs/2017/03/06/latency-and-sql-blocking-query-improvements/*。

XenApp 和 XenDesktop 版本 7.6 到当前版本的数据库调整指南

May 20, 2020

免责声明

本文档包含指向由 Citrix 以外的其他方控制的网站的链接。Citrix 不对这些第三方网站的内容或使用负责，也不认可或承担任何责任。Citrix 仅为了方便您提供这些链接，并且包含任何链接并不意味着 Citrix 对链接网站的认可。您应该自己采取防范措施，以确保选择使用的内容未携带病毒或者其他有害特性。

概述

典型的 XenDesktop 7 部署由三个数据库组成，如下所示：

- 站点配置数据库
  存储 XenDesktop 部署的当前配置和状态
- 监视数据库
  存储历史数据，以便在 Director 中显示

- 配置日志记录数据库
  跟踪对 XenDesktop 部署所做的配置更改

默认情况下，配置日志记录和监视数据库（辅助数据库）与站点配置数据库位于同一服务器上。最初，这三个数据库同名。Citrix 建议您在创建站点后更改辅助数据库的位置。

典型的部署还使用 SQL Server 提供的临时数据库 TempDB。

每个数据库都有不同的用途，并以不同的速度增长。

本文档提供了有关每个数据库的信息，并突出说明了调整数据库以支持 XenDesktop 7 时需要考虑的主要注意事项。

注意：提供的所有数字均为估计数。预计部署之间的差异会有所不同。

本文档还说明了托管共享桌面 (HSD) 和虚拟桌面基础架构 (VDI) 之间的大小差异。混合环境需要将这两种桌面类型的估计值结合起来，以生成总体数据库大小的估计值。

XenDesktop 7.6 的文档更改

本文档已扩展至涵盖 7.6 XenDesktop。这是为了允许更新 7.6 中添加的功能的大小变化。影响数据库大小的三个新功能是：

- 连接租赁 — 压缩的租赁文件存储在站点数据库中
- 应用程序使用情况监视 — 在环境中使用的所有应用程序的详细信息存储在监视器数据库中
- 修补程序清单监视-应用于环境中的控制器、VDA 和 VDA 映像的 Citrix 修补程序的详细信息

下面更新了有关表大小的信息。每秒事务和事务日志增长在 7.6 到 7.5 中类似，因此没有对这些部分进行更新。

高级注意事项

站点数据库

站点数据库包含系统运行的配置信息。

它的使用特点是：

- 当用户登录生成要跟踪的会话和连接信息时，会在高峰时段达到最大大小。
- 如果没有活动会话且 VDA 全部关闭并取消注册，则达到最小大小。
- 48 小时后达到峰值大小，因为数据库存储的持久性信息很少。
  这是由于在站点数据库中维护 48 小时的小型连接日志。
- 数据库的基线大小随着站点的配置信息的增加而增加。
  也就是说，越来越多的工作人员和用户占用更多的数据库空间。
- 登录过程中每秒发生高水平的事务，因为每个用户登录都需要执行多个单独的事务，并根据并发启动速率进行扩展。
- VDA 心跳事务的低级背景噪声。每个 VDA 每 5 分钟提供一次检测信号，此更新将触发数据库上的事务。

故障的影响

站点数据库中断使系统无法进行管理和监视。维护现有连接。在 XenDesktop 7.6 连接租赁中，允许建立新的连接和重新连接。在以前的版本中，不可能进行新的连接和重新连接。

监视数据库

监视数据库包含有关站点的历史信息。Director 使用此信息显示历史信息。

它的使用特点是：

- 最大大小由配置的保留期控制，如下所示：
  - 对于非白金卡客户，默认值为 7 天，最长期限为 7 天。
  - 对于白金卡客户，默认值为 90 天，没有最长期限。
- 峰值大小可能需要一些时间才能达到，因为系统必须达到配置的保留期。
- 由于监视服务的更新批处理性质，每秒发生的事务级别较低。很少见到每秒通过 20 个交易每秒标记。
- 由监视服务定期合并调用引起的一些后台事务。
- 执行隔夜处理以删除配置的保留期之外的数据。

故障的影响

监视数据库中断会阻止为站点收集数据，这意味着数据在 Director 中不可见。

配置日志记录数据库

配置日志记录数据库包含对站点所有配置更改的历史日志。此信息用于生成报告或显示在 Studio 中。

它的使用特点是：

- 最大大小难以预测，因为它取决于有多少配置活动。
- 从 Director 中的任何操作（例如，会话重置）都会记录到此数据库，因此在管理员使用 Director 时，可能会出现一些缓慢的增长。
- 当未进行任何配置更改时，数据库上发生的最小事务。
- 更新过程中的事务速率较低，因为在可能的情况下对更新进行批处理。
- 手动删除数据。配置日志记录数据库中的数据不受任何保留策略的约束，除非管理员手动执行此操作，否则不会删除。

故障的影响

配置日志记录数据库中断的影响取决于站点配置，如下所示：

- 如果站点不允许在配置日志记录数据库不可用时进行更改，则无法重新配置 XenDesktop 部署。
- 如果站点允许在配置日志记录数据库不可用时进行更改，则可能会对 XenDesktop 部署进行未跟踪的配置更改。

临时数据库

临时数据库是由 SQL Server 提供的系统范围的数据库。它用作读提交快照隔离的版本存储。XenDesktop 7 使用此 SQL Server 功能来减少 XenDesktop 数据库中的锁争用。

版本存储的大小取决于活动事务的数量。然而，一般来说，它不超过几 MB。

TempDB 的性能确实影响 XenDesktop 代理的性能，因为生成新数据的任何事务都需要 TempDB 空间。但是，XenDesktop 往往具有短暂的事务，这有助于保持版本存储大小较小。

当查询生成大型中间结果集时，也会使用临时数据库。

MSDN 中可以找到有关调整和配置 TempDB 的指南：

http://technet.microsoft.com/en-us/library/ms175527(v=sql.105).aspx

争用的主要区域以要使用的文件数为中心。较旧版本的 SQL Server，如 SQL Server 2000，需要更多的文件比较新版本。有关要使用的文件数的详细信息，请参阅：

http://www.sqlskills.com/blogs/paul/a-sql-server-dba-myth-a-day-1230-tempdb-should-alwayshave-one-data-file-per-processor-core/

读提交的快照隔离

Citrix 建议所有 XenDesktop 7 数据库都使用读提交的快照隔离。有关详细信息，请参阅如何在 XenDesktop 中启用读取已提交的快照。

调整数据库的大小

数据库大小取决于许多关键因素，包括在工作日内创建的会话数和连接数。

会话是指可能断开连接并重新连接到的任何桌面或应用程序运行一段时间。

连接是用户连接到会话的任何时间。断开连接会关闭连接，但不会关闭会话。当用户重新连接时，这将创建到现有会话的新连接。

站点数据库

站点数据库的最大大小取决于 VDA 和活动会话数，如下所示：

用户	应用程序	类型	预期峰值大小 7.5 (MB)	预期峰值大小 7.6 (MB)
1000	50	HSD	30	31
10000	100	HSD	60	198
100000	200	HSD	330	752

用户	应用程序	类型	预期峰值大小 7.5 (MB)	预期峰值大小 7.6 (MB)
1000	不适用	VDI	30	30
10000	不适用	VDI	115	121
40000	不适用	VDI	390	426

每个已发布的应用程序将 110 KB 添加到数据库以存储每个唯一图标。

> 注意：
>
> 7.6 的大小增加是由于连接租约作为 Controller 之间复制的一部分存储在数据库中。

监视数据库

在这三个数据库中，预计监测数据库将随着时间的推移增加到最大数据库。

它的大小取决于许多因素，包括以下因素：

- 用户数
- 会话数
- 连接数
- VDI 或 HSD 工作人员
- 已配置保留期

下面是对多个数据点处数据库大小的估计值。此数据是基于扩展测试 XenDesktop 时看到的数据的估计值。这些估计数据被认为是现实的。

但是，维护其数据库的客户可能会发现他们的数据库小于估计值。

HSD 用户基于每台 HSD 服务器 100 个用户。

最长保留期限

保留的最大数据量由许可证控制，如下所示：

- 非白金客户最多可以保留 1 周（7 天）的数据。
- 白金客户可以保留无限数据；默认时间为 3 个月（90 天）。

可以使用设置 *Set-MonitorConfiguration* cmdlet 调整保留期。

当数据早于配置的保留期后，它将从数据库中删除。

XenDesktop 7.5 监视数据库大小

估计每个用户有 1 个连接和 1 个会话，每个工作周为期 5 天

用户	类型	1 周 (MB)	1 个月 (MB)	3 个月 (MB)	1 年 (MB)
1000	HSD	151	70	230	900
10000	HSD	2,830	600	1950	7700
100000	HSD	1500	5900	19000	76000
1000	VDI	15	55	170	670
10000	VDI	120	440	1400	5500
40000	VDI	464	1700	5400	21500

每个用户有 2 个连接和 1 个会话，每个用户每周工作 5 天

用户	类型	1 周 (MB)	1 个月 (MB)	3 个月 (MB)	1 年 (MB)
1000	HSD	30	100	330	1300
10000	HSD	240	925	3000	12000
100000	HSD	2400	9200	30000	119000
1000	VDI	25	85	280	1100
10000	VDI	200	750	2500	9800
40000	VDI	800	3000	9700	38600

请注意，由于负载平衡信息的记录，HSDs 会随着时间的推移生成更多数据，但最初的大小与 VDI 桌面类似。

XenDesktop 7.6 监视数据库大小

7.5 的主要变化是：

- 修补程序信息
  下面的数据基于每个工作程序（VDI 或 HSD）3 修补程序
- 应用程序使用历史记录
  这主要与 HSD 系统相关。

估计每个用户有 1 个连接和 1 个会话，每个工作周为期 5 天

用户	类型	1 周 (MB)	1 个月 (MB)	3 个月 (MB)	1 年 (MB)
1000	HSD	151	605	1,966	7,865
10000	HSD	2,830	11,301	36,712	146,834

用户	类型	1 周 (MB)	1 个月 (MB)	3 个月 (MB)	1 年 (MB)
100000	HSD	7,194	28,585	92,758	370,841
1000	VDI	13	49	157	622
10000	VDI	117	409	1,287	5,090
40000	VDI	460	1,610	5,058	19,999

每个用户有 2 个连接和 1 个会话，每个用户每周工作 5 天

用户	类型	1 周 (MB)	1 个月 (MB)	3 个月 (MB)	1 年 (MB)
1000	HSD	159	635	2,063	8,251
10000	HSD	2,904	11,599	37,684	150,718
100000	HSD	7,940	31,572	102,465	409,672
1000	VDI	21	79	253	1,008
10000	VDI	191	708	2,258	8,974
40000	VDI	759	2,805	8,941	35,532

配置日志记录数据库

为配置日志记录数据库大小提供指导要困难得多，因为它根据 Director 的日常活动和配置站点的大小而有很大的差异。

会记录对会话或用户有影响的活动，例如，会话注销和重置。被动活动（例如列出用户的会话）不是。

用于部署桌面的机制还会影响所记录数据的大小。

在不使用 MCS 的 HSD 环境中，数据库大小往往介于 30 MB 和 40 MB 之间。

对于 MCS 环境，由于记录了所有 VM 构建数据，数据库大小可能很容易超过 200 MB。

未针对 7.6 版本对配置日志记录数据库做出重大更改。

登录 **10** 万个 **HSD** 会话期间的数据库活动

在可扩展性测试（模拟 10 万个 HSD 会话登录）期间，事务日志增长在两个登录速率下测量，如下所示：

- 超过 1 小时的 10 万用户登录
- 超过 2 小时的 10 万用户登录

选择这些比率是为了提供示例数据点。

环境包括：

- 2 个 Delivery Controller
- 43 个 HSD VDA 工作人员
- 3 个 SQL Server，配置了数据库，保存在一个始终处于可用性组中

本文档末尾提供了服务器配置的详细信息。

事务日志增长

使用性能监视器计数器 SqlServer:Databases — 日志文件使用大小 (KB) 来监视所有数据库的事务日志增长。

站点数据库

当系统处于空闲状态时，事务日志每小时增长 3.5 MB。这是 VDA 和 Broker Service 检测信号的组合。

测试	登录总数增长 (MB)	注销总量增长 (MB)
超过 1 小时 10 万	1,900	1,150
超过 2 小时 10 万	1,900	1,150

在测量的时间段内，对数增长是线性的。此数据表明，每个用户登录时，事务日志增长 20 KB。每个用户注销事务日志增长 12 KB。

因此，每天增长为 32 KB 每用户登录/注销周期。

监视数据库

当系统处于空闲状态时，事务日志每小时增长 30.5 MB。这是整合存储过程和 HSD VDA 负载索引更新的组合。

测试	登录总数增长 (MB)	注销总量增长 (MB)
超过 1 小时 10 万	670	190
两小时内 10 万人	650	220

在测量的时间段内，对数增长是线性的。此数据表明，每个用户登录事务日志增长 7 KB。每个用户注销事务日志增长 2 KB。

因此，每天增长为每个用户登录/注销周期 9 KB。

每秒事务数

使用以下性能监视计数器监视所有数据库的事务日志增长：

- SqlServer:Databases – 事务/秒
- SqlServer:Databases - 写入事务/秒

站点数据库

当系统处于空闲状态时，每秒有 5 个事务处理，其中 1 个写事务处理/秒维护 VDA 和代理检测信号。

> 注意：这些数字是从给定时间段得出的估计数。确切负载取决于每秒并发启动次数。

测试	每秒登录事务	每秒登录写入事务	每秒注销事务数	每秒注销写入事务
超过 1 小时 10 万	870	310	250	100
两小时内 10 万人	475	170	140	60

监视数据库

当系统处于空闲状态时，合并存储过程每分钟运行一次，并生成事务。然而，交易水平很小。通常，每个合并存储过程有 2—3 个事务和 1 个写入事务，并运行 3 个合并存储过程。在活动期间，随着开展更多的工作，开销会增加。

> 注意：这些数字是从给定时间段得出的估计数。

测试	每秒登录事务	每秒登录写入事务	每秒注销事务数	每秒注销写入事务
超过 1 小时 10 万	4	2	4	2
两小时内 10 万人	4	2	3.5	2

CPU 使用率

用于此测试的所有 SQL Server 都是启用了超线程的双六核服务器。本文档末尾提供了确切的硬件规格。

众所周知，这些服务器对于正在运行的负载而言过大。这使我们能够确定硬件的限制和最大限度。预计 SQL CPU 负载实际上可能已由具有单个四核而不是双十六核系统的 SQL Server 处理。

在测试期间，系统的 CPU 使用性能监视器计数器
处理器进行监视 — %处理器时间 —_ 总计。

主复制副本

而空闲 CPU 以 0-2% 的可用 CPU 运行。整合存储过程每分钟都会导致系统 CPU 的大约 1 到 8-10% 的峰值。预计这将根据正在处理的数据量进行扩展。

在 1 小时内登录 100000 个用户期间，CPU 上升至 7%，随着环境中存在的会话和用户增加，线性上升至 11%。请注意，整合存储过程峰值增加了 7% 的 CPU 总量，导致峰值达到 18% 的 CPU。

在注销期间，CPU 运行率为 3.5%，整合存储过程占 7% 的额外 CPU。总体而言，这表明需要低于 20% 的双十六进制核才能维持登录和注销速率。

注意： Windows Server 2012 调度程序偏向于仅在需要时使用超线程，也就是说，在系统达到 50% 负载之前，它只运行每个核心一个线程，因此 24 个超线程上的 20% 负载正在 4.8 内核上运行。

鉴于工作负载，相信这是一个沉重的压力测试，并且单个四核 SQL Server 将足够用于 XenDesktop 部署。

辅助副本

发现辅助副本在登录期间配置 2% CPU，注销期间配置 1.5%。这是预期的，因为在大多数情况下，副本将从主副本存储在其磁盘上的数据，并且事务只涉及同步副本，因为在辅助副本确认之前，主体副本不会提交事务。

根据关于 HA 硬件与主副本匹配的建议，此负载将由类似指定的服务器非常容易地处理。

临时数据库使用情况

TempDB 用于多种用途，包括版本存储、大型查询集的空间和其他临时表使用。

TempDB 大小

在这个 SQL 配置中，TempDB 被配置为有 8 个数据库文件，每个文件的大小固定为 5 GB。这允许更好地并发使用 TempDB，但也提供了足够的空间，并且不会触发任何自动增长事件。根据捕获的数据，该部署的规模过大。但是，有足够的磁盘空间可用。

它还保持在 TempDB 数据库文件数量的一般指导范围内，在可用的 CPU 数量的一半之间，但不超过 8，不知道是否有实际争用。

请注意，仅使用一个 TempDB 日志文件，因为 SQL Server 不能从多个日志文件中受益。

版本存储

TempDB 包含与 XenDesktop 数据库使用的读取提交的快照隔离相关的行版本的版本存储。

使用情况可以通过以下性能计数器来衡量：

- SQLServer:Transactions — 版本存储大小 (KB)
- SQLServer: 事务 - 版本清理速率（KB/秒）
- SQLServer: 事务 - 版本生成速率（KB/秒）

在超过 1 小时的 100000 次登录期间，版本存储大小保持在 10 MB 到 30 MB 的范围内，在创建并清理版本时具有锯齿效果。在注销期间，范围为 10 MB 到 21 MB。空闲时，版本存储大小介于 1 MB 到 4 MB 之间。

在登录期间，版本生成速率在 250-500 KB 范围内；注销期间为 150 — 400 KB/秒，空闲时为 0 - 250 KB/秒。

版本清理每分钟运行一次，登录时达到 2500 KB/秒，注销时达到 1750 KB/秒，空闲期间达到 400 KB/秒。

磁盘 I/O/O

在登录测试期间，使用以下性能计数器测量磁盘 I/O：

- PhysicalDisk — 磁盘读取字节/秒
- PhysicalDisk — 磁盘写入字节/秒
- PhysicalDisk — 磁盘读数/秒
- PhysicalDisk — 磁盘写入/秒

读取 I/O 被发现是最小的，因为 SQL Server 能够将所有数据保存在内存中，从而导致系统上的读取活动很少。

由于数据库和存储系统的布局，卷被拆分，其中一个卷保存所有数据文件，另一个卷保存所有事务日志文件。

数据显示的模式很难放置到表中。一般来说，事务日志的写入字节/秒为 1 小时测试 800 KB/秒，2 小时测试 400 KB/秒。每分钟一次，当合并存储过程运行时，事务日志显示峰值为 30 MB/秒。

对合并存储过程的分析表明，有时统计信息会使查询计划变得不理想，并且临时表会溢出到 TempDB 中。此触发器会向 TempDB 的事务日志写入。

此数据传输转换为 1 小时测试的稳定状态为 300 次写入输入/输出操作 (IOPS)，2 小时测试为 200 次写入 IOPS。整合存储过程的峰值会在运行时添加另一个 2—300 写 IOPS。请注意，在大型环境中，合并存储过程运行时间不到一秒钟。

当每个数据库进行检查点时，数据将从内存中表同步到数据卷上的数据文件。

有关 SQL 检查点的详细信息，请参见 http://technet.microsoft.com/enus/。

这些检查点是非常短的活动时间，通常不到 1 秒。

在登录过程中，检查点消耗了 6-7 MB/秒和 500 写入 IOPS。注销期间，检查点消耗了 7 MB/秒，介于 200—700 IOPS 之间。这些数字有所不同，因为站点和监控数据库具有不同数量的要检查点的数据量。

数据库维护

在大型部署中维护数据库非常重要。如果数据库未得到正确维护，则可能会因数据库空间不足而导致数据库中断，例如，如果事务日志设置为自动增长并填充磁盘，或事务日志是固定大小并已满。

事务日志维护

使用 SQL Server 高可用性功能（例如“始终处于可用性组”或“数据库镜像”）时，XenDesktop 数据库以完全事务日志记录模式运行。

通过在完全事务日志记录模式下运行，事务日志将继续增长，直到执行数据库或事务日志备份。

如果事务日志文件不受监视，因为默认情况下，SQL Server 将日志文件配置为自动增长，这可能会导致问题。这会导致 2 个问题：

1. 事务日志文件可能会占用大量磁盘空间。
2. 每次事务日志增长时，它都会停止所有事务，直到日志空间为零。

Citrix 建议定期备份日志文件。这可以通过计划作业或维护计划来完成。

或者，使用 SQL Server 代理来监视日志使用的大小何时超过阈值并运行备份作业。

在规模测试中，使用了 4 GB 的固定大小的日志，并设置了警报，以便在日志文件达到 80% 时将日志备份到另一个文件。这会阻止日志增长和占用所有磁盘空间，并停止日志将磁盘空间归零并停止数据库。

示例作业将运行一个脚本，如：

```
BACKUP LOG [CitrixXenDesktop-SiteDB] TO DISK = N'D:\LogBackup\
    CitrixXenDesktopSiteDB.bak' WITH NOFORMAT, NOINIT, COMPRESSION, NAME
     = N'Site-Transaction Log Backup', SKIP, NOREWIND, NOUNLOAD
```

用于警报的 SQL 性能计数器为：

SQLServer:Databases - 使用的百分比日志 - CitrixXenDesktopSiteDB

对 3 个数据库中的每个数据库重复此操作。

发现日志文件的备份对正在运行的 XenDesktop 环境的影响最小，代理时间略有增加，但不是我们认为重要的东西。

有关配置作业的详细信息，请参阅：http://msdn.microsoft.com/en-us/library/ms187880.aspx

有关配置警报的更多详细信息，请参阅：http://msdn.microsoft.com/en-us/library/ms191508.aspx

索引维护

随着数据库中输入的数据越来越多，某些索引开始变得越来越不完整，也就是说，存储在每个 SQL 页中的记录越来越少。SQL 页面是 8 KB。这会导致数据库增加其内存和磁盘上的存储需求。通过维护索引，可以增加页面完整性，从而减少数据库的内存需求。

Citrix 建议客户安装维护计划每晚和每周运行以维护索引。维护计划可能只是在一周内夜间重新组织索引，并在周末重建索引。

此建议可避免在日常操作中重建任何大型索引造成的任何性能影响，尤其是对于大型监视数据库。

Microsoft 建议重新构建索引，如果索引的碎片大于 30%，如果小于 30%，则重新组织索引。有关详细信息，请参阅 Microsoft TechNet 库中的 重新组织和重建索引。

重新组织索引后，统计信息也应更新。随着数据库的增长，这一点尤其重要；否则某些统计信息可能较差，SQL 可能会生成次优的 SQL 查询计划。

就节省的空间而言，下面的 Microsoft 脚本针对 1.2 GB 的监视数据库运行。它改进了页面填充并释放了 300 MB 的空间。

第三方脚本

Microsoft

Microsoft 建议使用以下脚本更新其 WSUS SQL 数据库的索引：

http://gallery.technet.microsoft.com/scriptcenter/6f8cde49-5c52-4abd-9820-f1d270ddea61

通过更改 “USE SUSDB”，也可以对 XenDesktop 数据库运行此脚本。此脚本遵循 Microsoft 的最佳实践，重新构建超过 30% 碎片的索引，并重新组织低于 30% 的索引。然后更新数据库的统计信息。

Ola Hallengren

更高级的脚本也可从以下网址获得：

http://ola.hallengren.com/

这些脚本在 SQL Server 社区中得到了很好的认可。具体来说，索引脚本可从以下位置获得：

http://ola.hallengren.com/sql-server-index-and-statistics-maintenance.html

这些脚本可用于更好地控制重新组织或重建索引的级别。

测试服务器配置

SQL Server 配置

SQL 可用性组由 3 个指定完全相同的戴尔 R720XD 服务器组成。

系统规格：

- 2 个半核 Intel Xeon CPU E5-2630，运行频率为 2.30 GHz，并且启用了超线程
- 64 GB ECC RAM
- PERC H710P 迷你带 1 GB 电池支持的高速缓存
- 26 300 GB 10k RPM SAS 驱动器

磁盘被拆分为以下卷：

- 系统卷
  - 包含操作系统和页面文件
  - 2 个磁盘作为 RAID 1 镜像
  - 总容量 278 GB
- 数据库卷
  - 包含 SQL Server 实例和数据库数据文件
  - 16 个磁盘作为 RAID 10 镜像条带
  - 总容量 2231 GB
- 日志卷
  - 包含数据库日志文件
  - 8 个磁盘作为 RAID 10 镜像条带

  - 总容量 1115 GB
- 软件：
  - Windows Server 2012 R2 Standard Edition，在测试时具有当前 Windows 更新（2014 年 8 月）
  - 具有累积更新 1 的 SQL Server Enterprise 2012 SP2
- 配置更改
  - SQL Server 配置为使用最多 61440 MB
  - 已在所有 SQL 实例上启用数据库包含
  - SQL Server 代理服务配置为自动启动
- 可用性组设置：
  - 所有服务器都放置在 Windows 故障转移群集中
  - 在群集中配置了“始终打开可用性”组
  - 辅助副本配置为“同步提交”，要求事务完成之前在两个副本上提交事务
  - 已为可用性组配置并启用只读副本路由

Delivery Controller 和 **HSD** 测试服务器

Delivery Controller 和 HSD 测试服务器使用 HP BL460c G1 刀片在相同的硬件配置上运行。2 台服务器用于 Delivery Controller，43 台服务器提供了模拟的 HSD 工作负载。

注意：虽然这些服务器相对较旧，但 HSD 服务器上的工作负载较低，因为会话模拟主要侧重于将负载放在 Delivery Controller 上，而不是 HSD 服务器上。

系统规格：

- 2 个四核 Intel Xeon L5320，运行频率为 1.86 GHz，不具备超线程功能
- 16 GB ECC RAM
- HP 智能阵列 E200I Raid 卡（无电池支持缓存）
- 一个 36 GB 或 72 GB SAS 硬盘

软件：

- Windows Server 2012 R2 Standard Edition，在测试时具有当前 Windows 更新（2014 年 8 月）
- Citrix XenDesktop 7.6

分析带溢出的 **PVS RAM** 缓存

May 20, 2020

本文提供了有关准确确定 RAM 高速缓存大小的信息，当使用 具有溢出到磁盘的功能 *RAM* 高速缓存大小。

具有溢出到磁盘的 RAM 缓存是一项 PVS 功能，其中虚拟磁盘写入首先写入 Windows 非页面缓冲池 RAM。一旦用户指定的 RAM 缓存大小达到其指定的大小，PVS 会将 RAM 缓存内容刷新到磁盘，以便为新数据创建空间。RAM 高速缓

存大小会根据工作负载模式和其他变化而波动。PoolMon 是一个工具，通过查找池标签 *VhdR* 来拍摄当前 RAM 缓存使用大小快照。

有关此 PVS 功能的其他信息，请参阅有关的博客 使用带溢出的 RAM 缓存。

> 重要
>
> 本文中介绍的工具适用于掌握 Provisioning Services 高级知识的管理员。此信息可用于帮助调试与性能相关的问题，这些问题超越了使用常用的工具和进程，包括进程监视器 (ProcMon)。通过这些信息，您将更好地了解 PVS 驱动程序的工作原理。

内存池监视器

PoolMon (poolmon.exe) 是指内存池监视器。它用于显示操作系统收集的数据（来自系统分页和非分页内核池的内存分配，以及用于终端服务会话的内存池）。此数据按池分配标记分组。

通过 非分页缓冲池内存，您可以使用 PoolMon 工具来验证 *VhdR* 表示的 **pooltag** 是否存在。VhdR 用于 RAM 缓存分配；此标记与池标记 *VhdL* 一起在创建脚本以帮助分析与非页面缓冲池内存中的 RAM 缓存相关联的数据时非常有用。

> 提示
>
> 开发人员和测试人员通常使用 PoolMon 来检测创建驱动程序、更改驱动程序代码或对驱动程序进行压力测试时的内存泄漏。PoolMon 还可用于测试过程的每个阶段，以验证驱动程序的内存分配模式和可用操作，包括用于确定驱动程序在任何给定时间使用的池内存量。有关使用内存池监视器的详细信息，请参阅 Microsoft Developers Network 站点。

使用 **Windows Performance Analyzer**

Windows Performance Analyzer (WPA) 是一个工具，允许您创建与 Windows Performance Recorder (WPR) 记录的事件（特别是 Windows 的事件跟踪）相关的图形和数据表。在调试与 PVS 驱动程序、存储堆栈和写入 VHDX 磁盘时出现的性能相关问题时，使用 WPA 帮助识别性能瓶颈。使用这些工具，您可以运行评估并打开任何事件跟踪日志文件进行分析。有关Windows Performance Analyzer的详细信息，请参阅 Microsoft 开发人员网络站点。

> 注意
>
> WPA 和 WPR 包含在 Windows 评估和部署工具包 (Windows ADK) 中；有关此部署工具包的详细信息，请参阅 Microsoft Web 站点。请访问 Microsoft Web 站点，获取最新版本的 Windows Performance Analyzer。

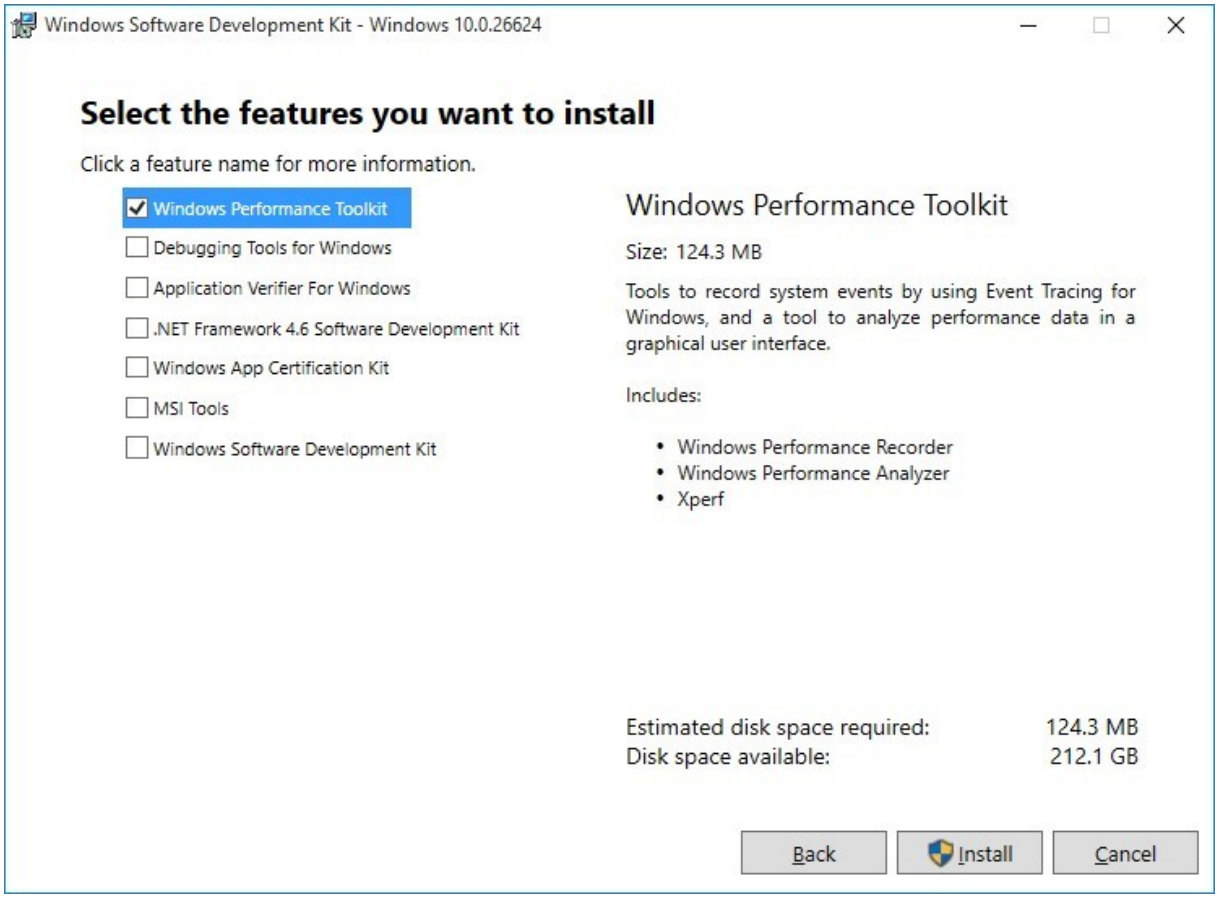
Windows Performance Analyzer 与 **Provisioning Services** 的工作原理

PVS 生成由 Windows 事件跟踪 (ETW) 机制捕获的事件。此功能提供了一种跟踪和记录由用户模式应用程序和内核模式驱动程序引发的事件的方法。ETW 在 Windows 操作系统中实现，为开发人员提供了一套事件跟踪功能的简单方法。有关详细信息，请参阅 Microsoft Developers Network。

安装 **Windows Performance Analyzer**

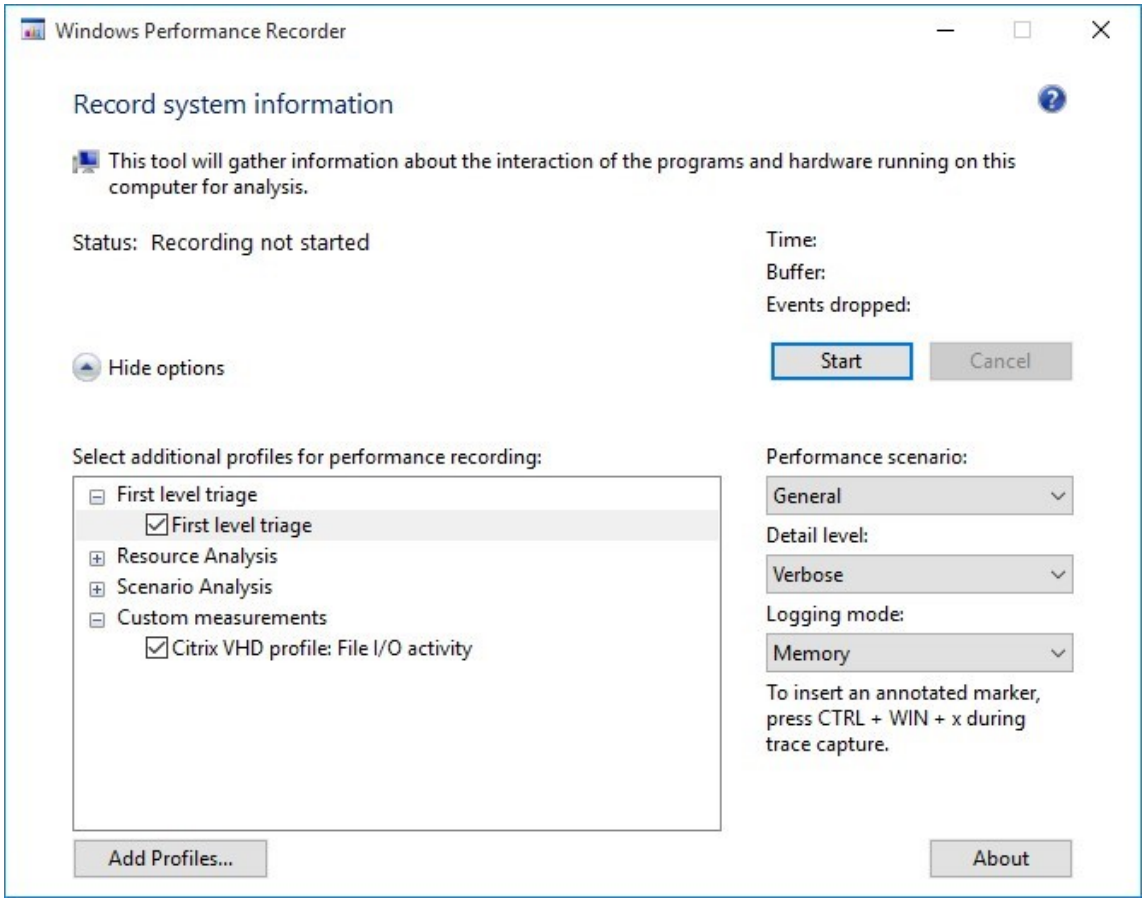
需要在主映像上安装 WPA。

WPA 是面向 Windows 10 操作系统的 最新 SDK 的一部分。您可以选择性地安装包含 WPA 和 WPR 的性能工具包：



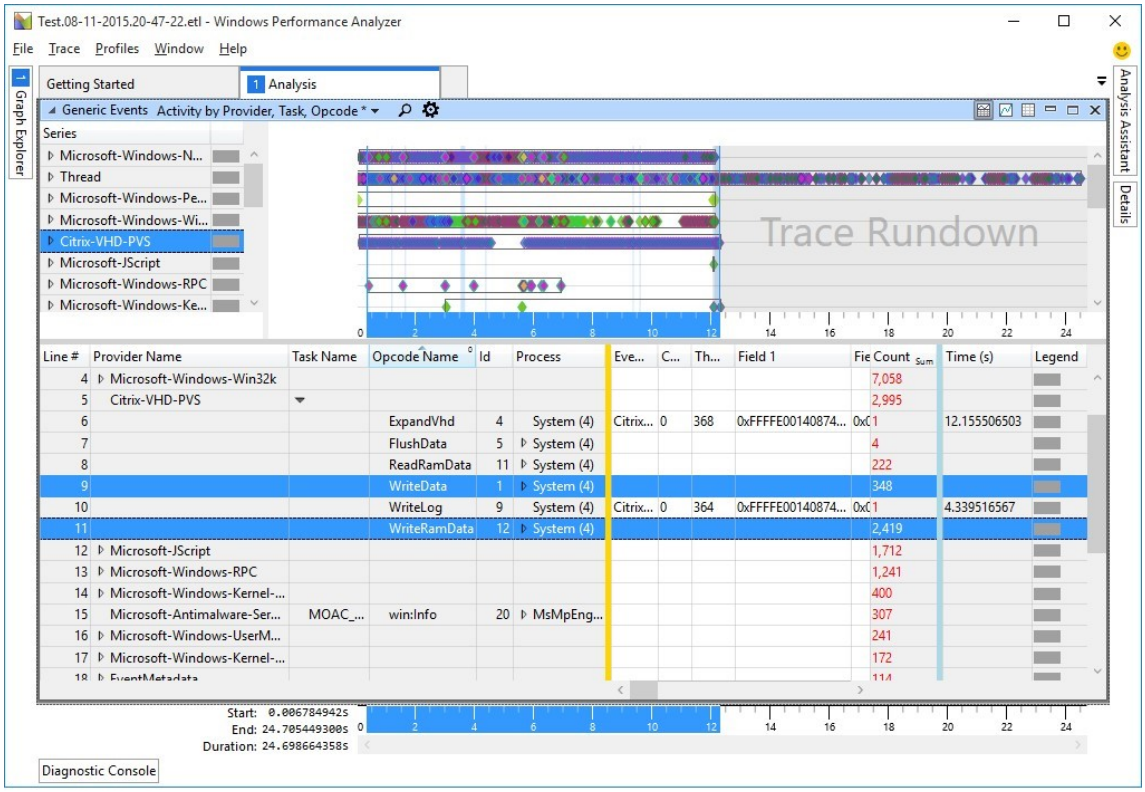
安装 WPA 和 WPR 后，使用 WPR 模拟 PVS 磁盘和文件 I/O 活动。创建此流量后，使用 WPA 分析数据。要执行这些操作，请执行以下操作：

1. 在目标设备上启动 WPR，然后单击添加配置文件。
2. 在“添加配置文件”屏幕中，浏览至特定于 PVS 的模板或配置文件。这允许您接收 PVS 事件提供程序生成的事件。导入配置文件后，返回 WPR 屏幕并选择要分析的任何其他选项，然后单击“开 始”按钮：



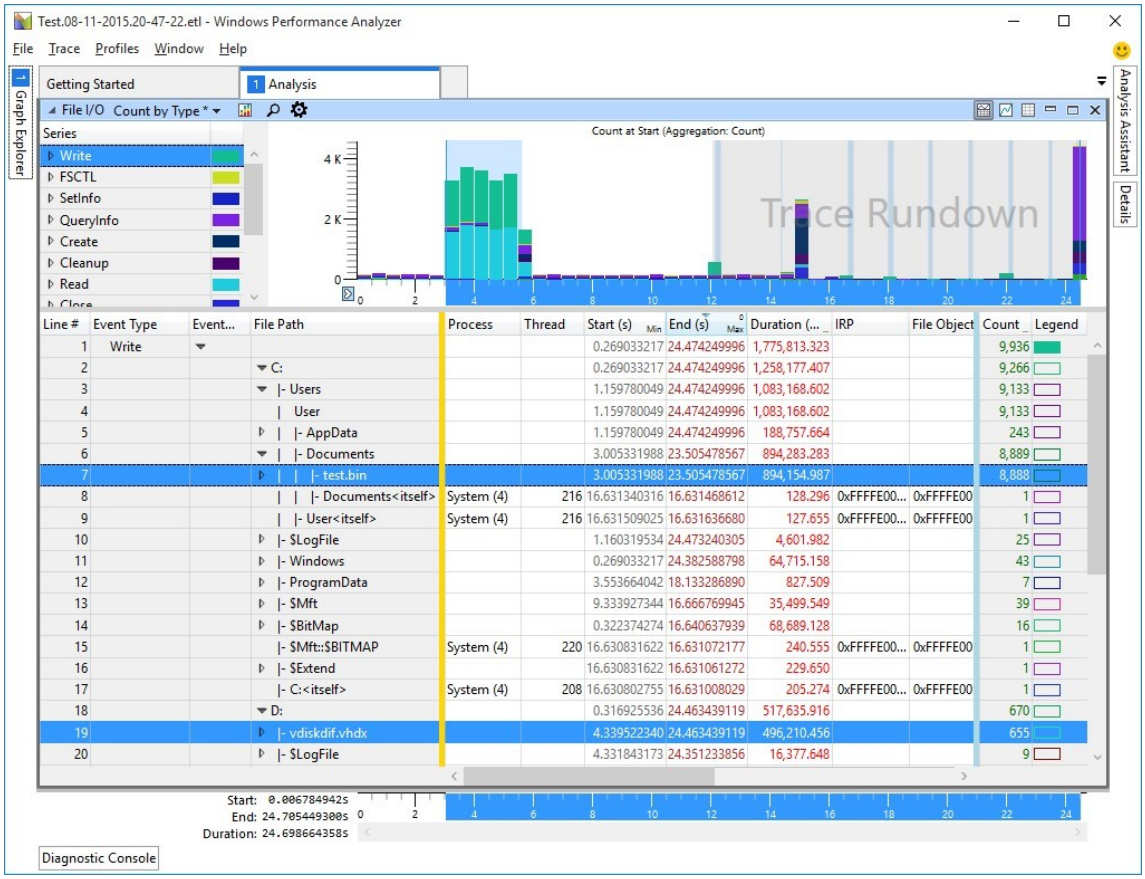
添加选项并单击“开始”后，您可以模拟 PVS 活动。在此示例中，将创建一个具有小内存缓冲区 (128 MB) 的新写缓存。更大的文件 (279 MB) 被复制到 C:\Users\User\Documents\test.bin，以强制 PVS 驱动程序将一些数据写入非分页池，以验证发生故障转移时会发生什么，从而开始写入本地磁盘（例如，D:\vdiskdif.vhdx）。复制文件并强制缓冲区超出容量后，您可以停止 WPR 中的捕获进程并使用 WPA 打开结果。

3. 使用 WPA 打开图形资源管理器，展开系统活动并选择一般事件。使用下面的屏幕作为示例，查看 WriteData 和 WriteRamData 部分中的内容。此信息显示写入到 C:\vDisk（2419 个文件）的确切文件数，包括 D: 驱动器上的 VHDX 文件（348 个文件）：



> 提示
>
> 写数据小于显示的值，因为它缓存在 RAM 中，并且尚未刷新到磁盘。

4. 返回到图形资源管理器屏幕，然后展开文件 **IO** 和按类型计数。下图说明了 IO（文件计数）的减少以及写入到 C:\Users\User\Documents\test.bin 和位于 D:\vdiskdif.vhdx 的溢出写入缓存文件之间所需的时间。使用此数据，您可以查看潜在的性能瓶颈，并有效地排除 PVS 筛选器驱动程序的问题：



5. 查看文件计数和写入之间（日志文件和溢出写缓存之间）的持续时间后，您可以在调试过程中进一步了解使用磁盘偏移的数据最初写入位置（以及最终写入位置）。在 Windows Performance Analyzer 中，打开图形资源管理器，展开系统活动，然后选择通用事件。修改列视图，使 WPA 工具能够显示不同存储层中的数据转换。为了进一步调试，请返回到 PVS 环境并将 RAM 缓存缓冲区设置为 0 MB，然后重新运行记录器 (WPR) 和分析器 (WPA) 工具。下图说明了溢出到磁盘的方式：

使用 **Citrix Secure Browser** 延长旧版 **Web** 应用程序的寿命

May 20, 2020

在网络应用程序和框架的世界中，必须接受多样性。不同类型的用户、组和公司需要访问正确的工具、应用程序和权限，才能连接到支持 Web 的业务应用程序。在大多数情况下，有一些法规遵从性因素决定了如何访问这些应用程序。需要支持较旧的子系统和较旧的浏览器框架的企业面临着一项艰巨的任务，即为业务关键型应用提供充分的访问权限并满足合规性要求。以下文档介绍了如何在创建更新和迁移策略时利用 Citrix Secure Browser 延长旧版 Web 应用程序和浏览器的访问权限以及使用寿命。

该解决方案要求发布兼容的浏览器，允许外部或内部用户访问，无论用户如何连接或用于连接内部站点的浏览器如何。此解决方案利用 XenDesktop 服务器操作系统 VDA、StoreFront、NetScaler Gateway 和 XenApp Secure Browser。当本机浏览器满足 IT 管理员设置的所有要求时，用户将其重定向到使用本机浏览器；如果策略检测到不兼容的浏览器或终端，则会将用户重定向到远程容器化已发布的浏览器会话。无论用户如何连接到环境，每个资源只需知道一个 URL（这样可以降低培训和支持成本）。

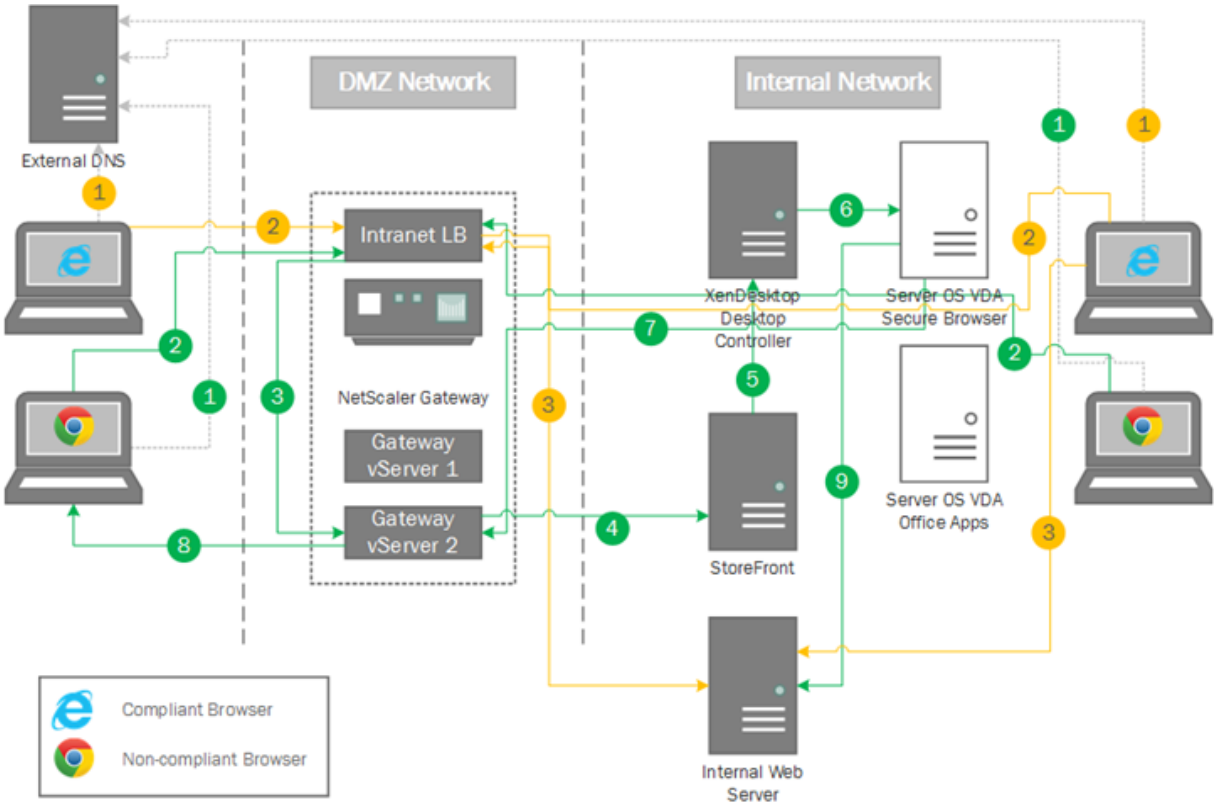
体系结构

以下部分说明用户如何访问内部站点，无论用户是从内部网络还是外部网络连接。在这种情况下，一种浏览器类型 (Internet Explorer) 是兼容的浏览器，另一种 (Google Chrome) 是不兼容的。由每家公司决定如何以及哪些浏览器映射到合规性策略。

对于此解决方案，我们假设 NetScaler Gateway 配置为对已发布的应用程序进行外部访问，这在图 1 中表示为网关虚拟服务器 1。第二个虚拟服务器（网关虚拟服务器 2）将用户重定向到启动 Secure Browser 的 HTML5 Receiver 会话。

用例

需要维护当前浏览器不再支持的旧式 Web 应用程序。在这种情况下，IT 仍然需要维护一个专为 Internet Explorer 8 设计的网站，供应商不再发布增强功能来支持新浏览器或其他浏览器。要解决此问题，IT 管理员发布了 Secure Browser，以允许满足浏览器要求的用户访问站点。下图解释了工作流中针对内部和外部用户的每个连接。



连接工作流

1. 每个用户都输入从外部 DNS 服务器解析的站点 URL，在我们的示例中为 `https://train.qckr.net`
2. 浏览器连接到 NetScaler Gateway 负载平衡器并确定合规性要求。
3. 当浏览器不符合要求时，内部和外部用户都会重定向到 NetScaler Gateway 虚拟服务器。当浏览器符合要求时，NetScaler Gateway 将通过外部用户的负载均衡器代理到内部站点的连接，并将本地浏览器重定向到内部用户的站点。
4. 虚拟服务器自动启动 StoreFront 枚举的会话。
5. StoreFront 联系 XenDesktop 控制器以获取会话信息和路由。
6. 会话通过 Secure Browser 桌面组启动；在这种情况下，会话是具有已发布兼容浏览器的服务器操作系统 VDA。
7. 会话通过 NetScaler Gateway 设备上的 ICA 代理进行连接。
8. Citrix Receiver for HTML5 可在本机浏览器中建立用户的会话。
9. 内部站点通过与 Citrix Receiver for HTML5 的 Secure Browser 会话显示。

设置和配置

本部分内容介绍如何使用 NetScaler Gateway 远程连接实现当前 XenDesktop 环境的解决方案。

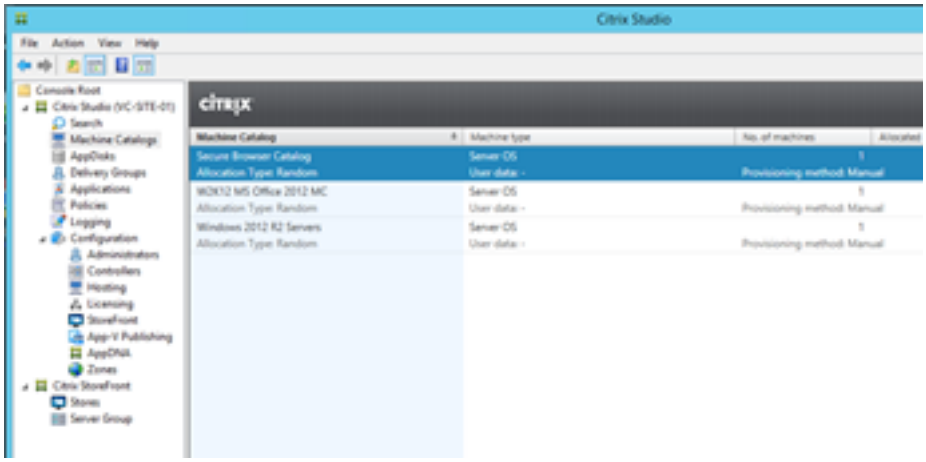
解决方案要求

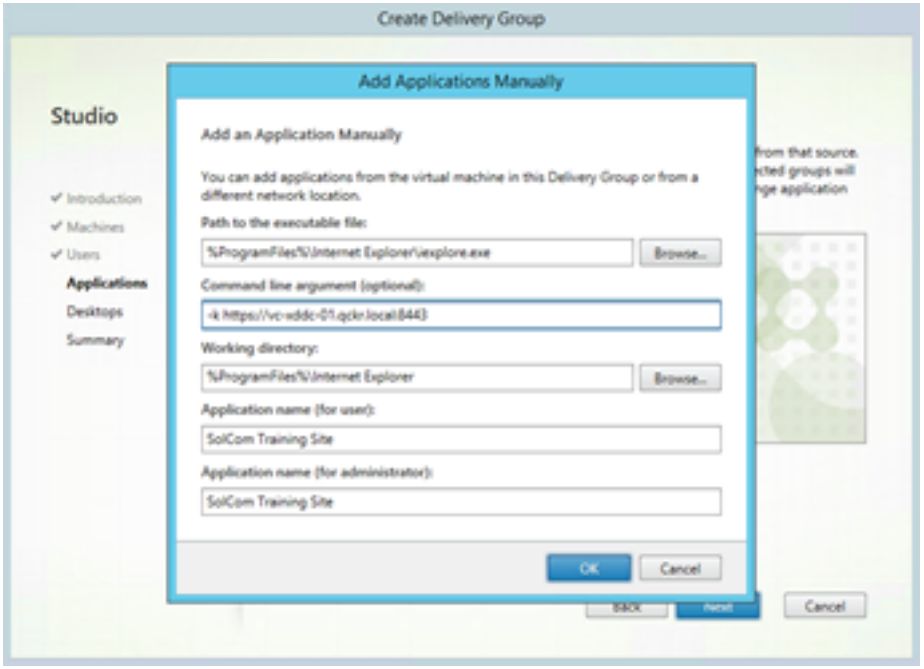
安装程序需要安装和配置以下组件：

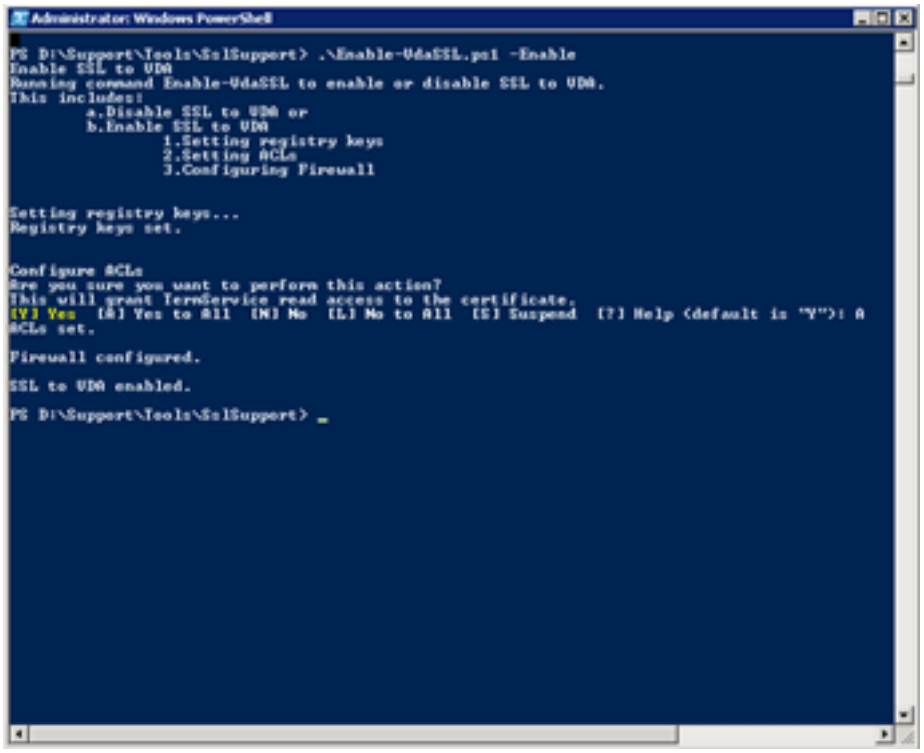
- XenDesktop 桌面控制器服务器
- 具有配置为外部访问的应用商店的 Citrix StoreFront 服务器
- 具有 XenDesktop 虚拟服务器的 NetScaler Gateway
- 使用已安装的浏览器作为 Secure Browser 的服务器操作系统 VDA
- 指向新 NetScaler 负载平衡器的外部 DNS 地址
- 指向新的 NetScaler Gateway 虚拟服务器的外部 DNS 地址

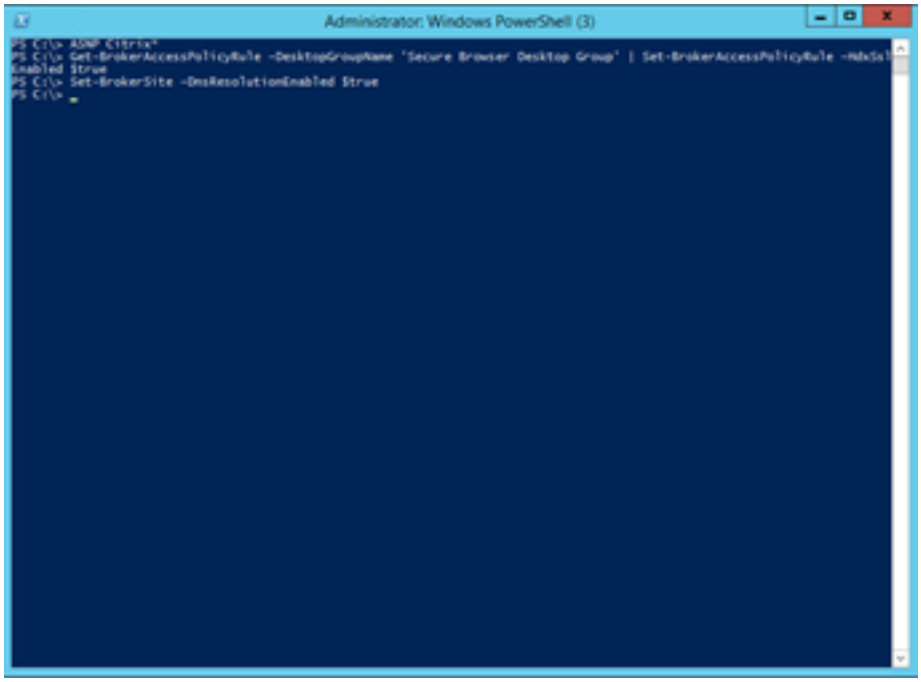
配置

XenDesktop 台式机控制器









将服务器操作系统 VDA 添加到名为 **Secure Browser** 目录的新计算机目录。

为 **Secure Browser** 目录创建交付组并发布 Internet Explorer。在命令行参数中，键入 *-k <URL of Internal Site>*。-k 参数是在自助服务亭模式下打开 Internet 资源管理器。在此示例中，我们正在发布 Internet 浏览器 8 并使用 URL 的内部站点。

您可以将交付组分配给特定用户和组。如果使用案例不需要添加桌面访问权限，则无需添加桌面访问权限。

在服务器操作系统 VDA 上，安装服务器或客户端身份验证证书，以便在控制器和 VDA 通信上启用 SSL。

装载 XenDesktop 7.6 或更高版本安装介质。打开一个 PowerShell 命令窗口，然后运行 *%MediaDrive%:\Support\Tools\SslSupport\Enable-VdaSSL.ps1 –Enable*

重新启动服务器操作系统 VDA 实例。

在 XenDesktop 控制器上，打开 PowerShell 命令窗口并运行命令 *ASNP Citrix**。

运行以下三个命令，以启用代理与 VDA 的安全通信：

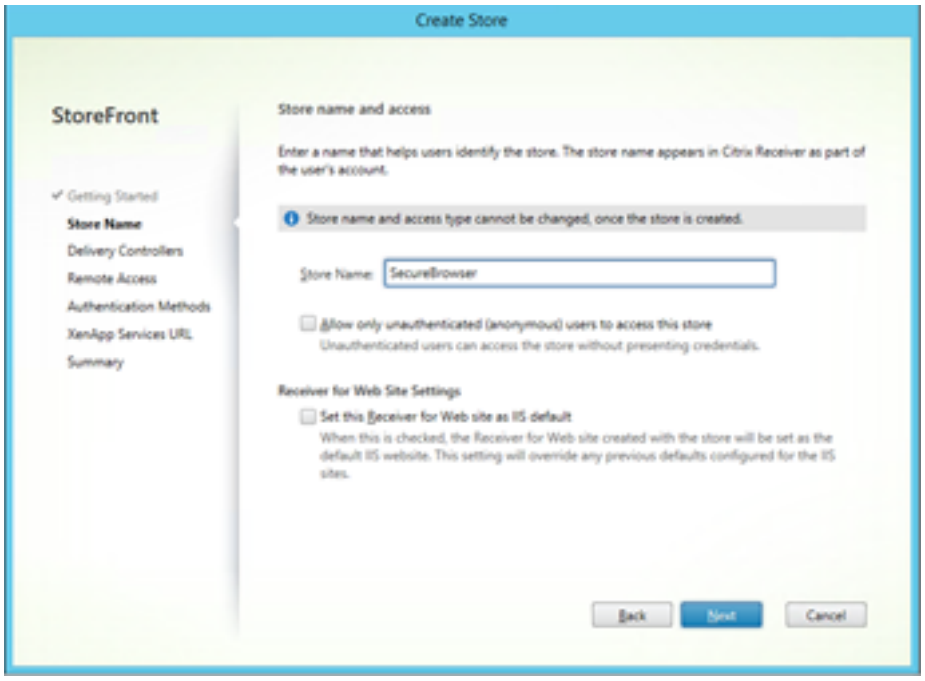
```
1 Get-BrokerAccessPolicyRule  -DesktopGroupName 'Secure Browser Desktop
     Group' | Set-BrokerAccessPolicyRule  -HdxSslEnabled $true*
2 <!--NeedCopy-->
```

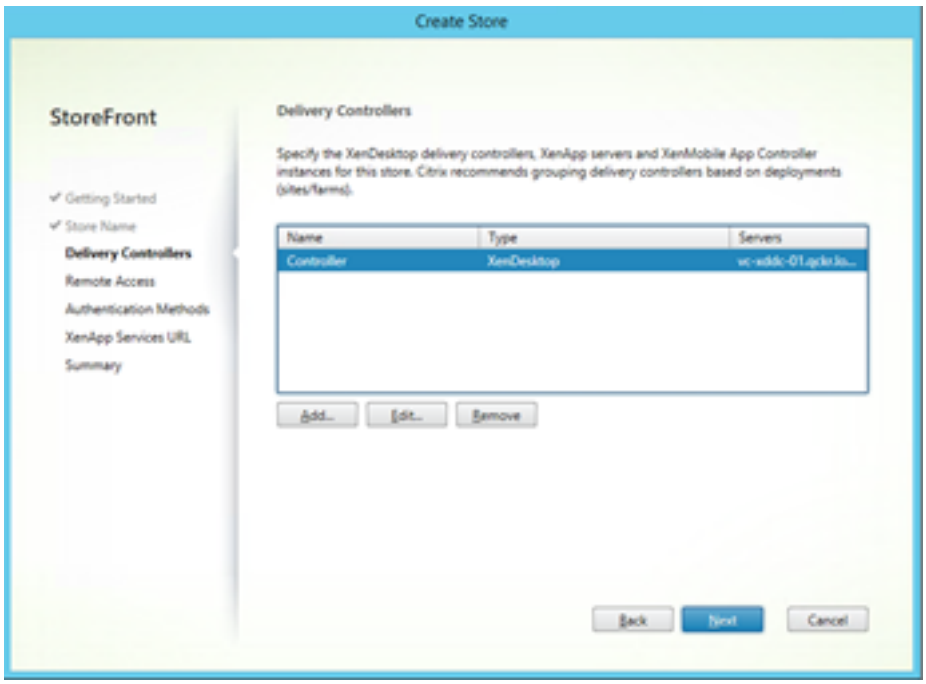
```
1 Set-BrokerSite  -DnsResolutionEnabled $true
2 <!--NeedCopy-->
```

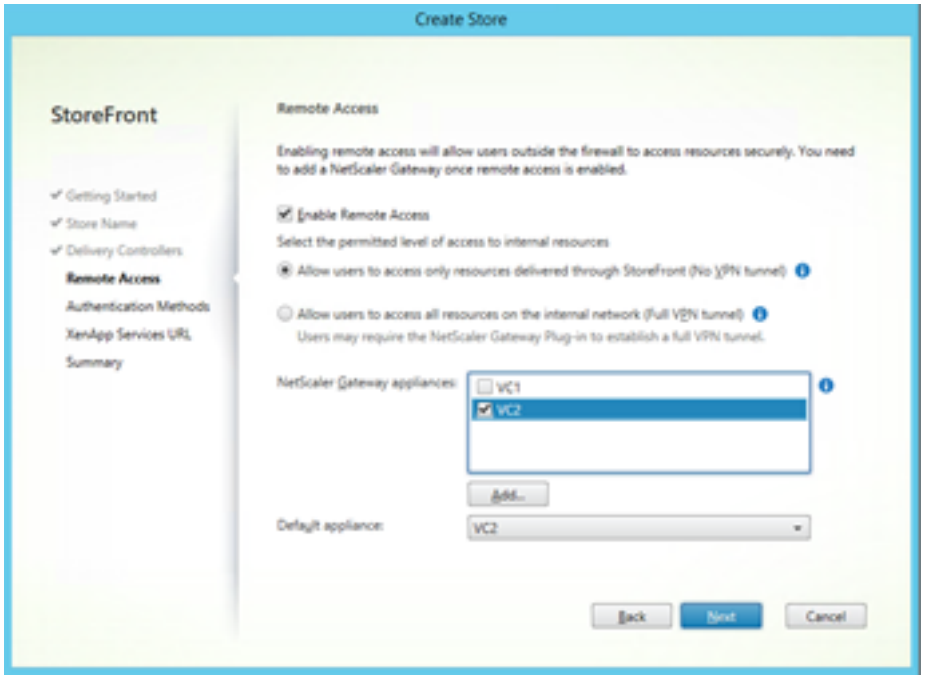


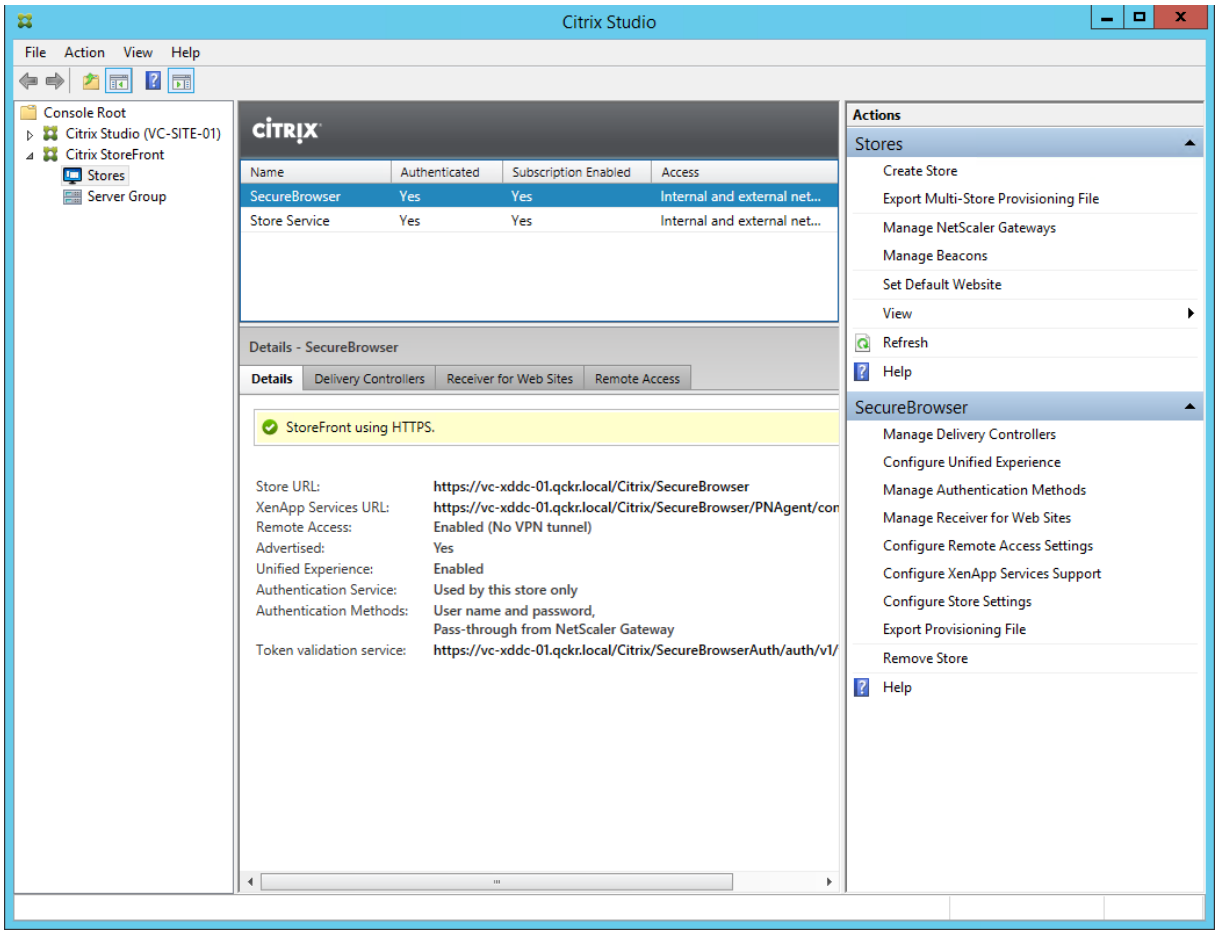
```
Set-BrokerSite -TrustRequestsSentToTheXmlServicePort $true*
<!--NeedCopy-->
```

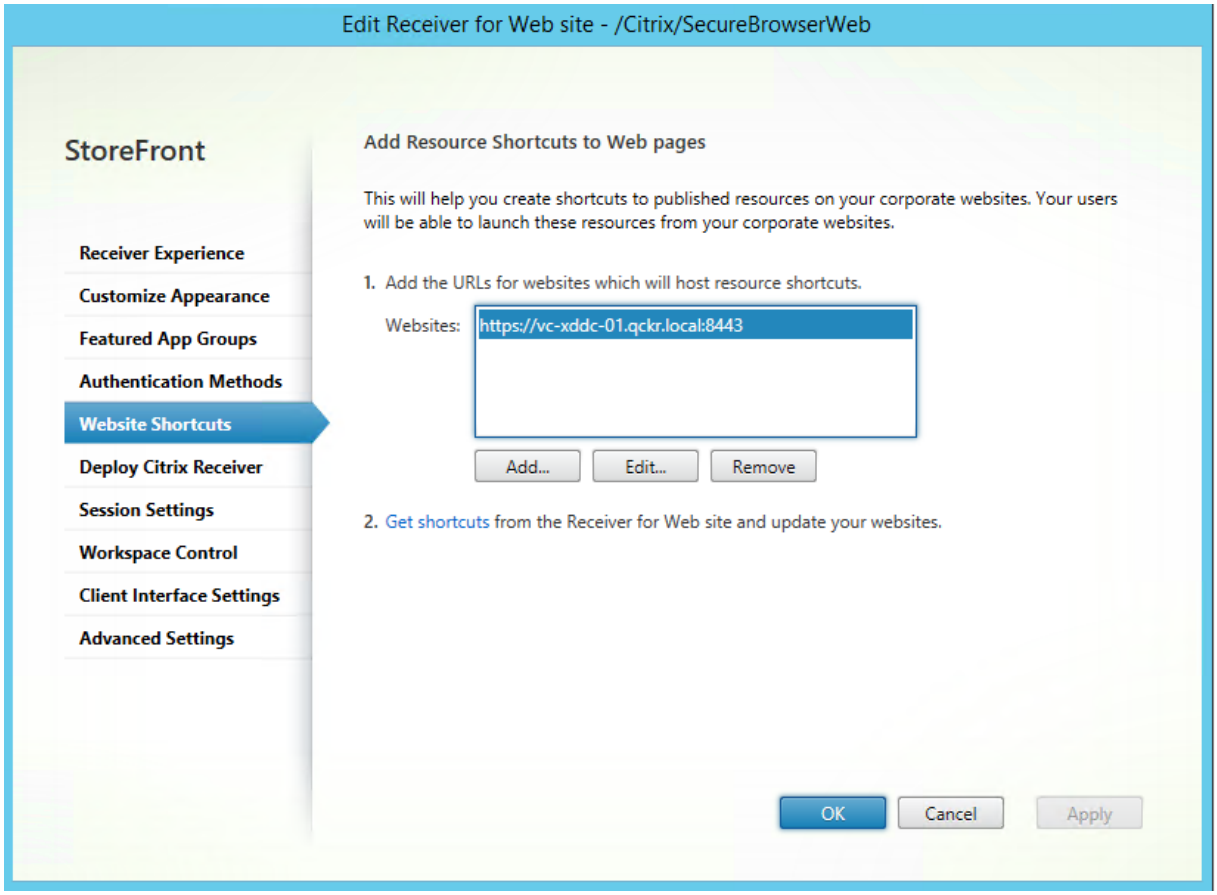
StoreFront

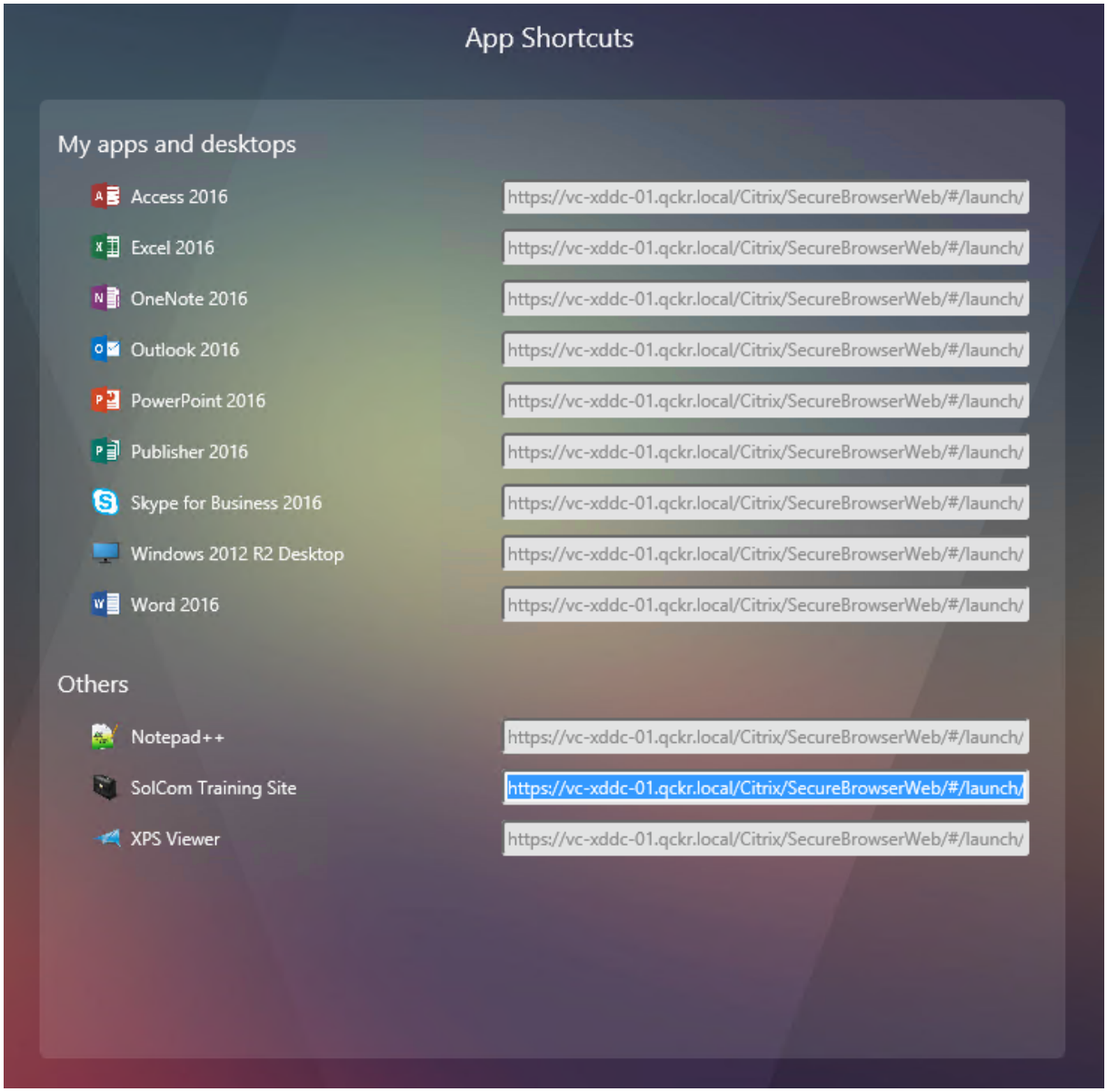


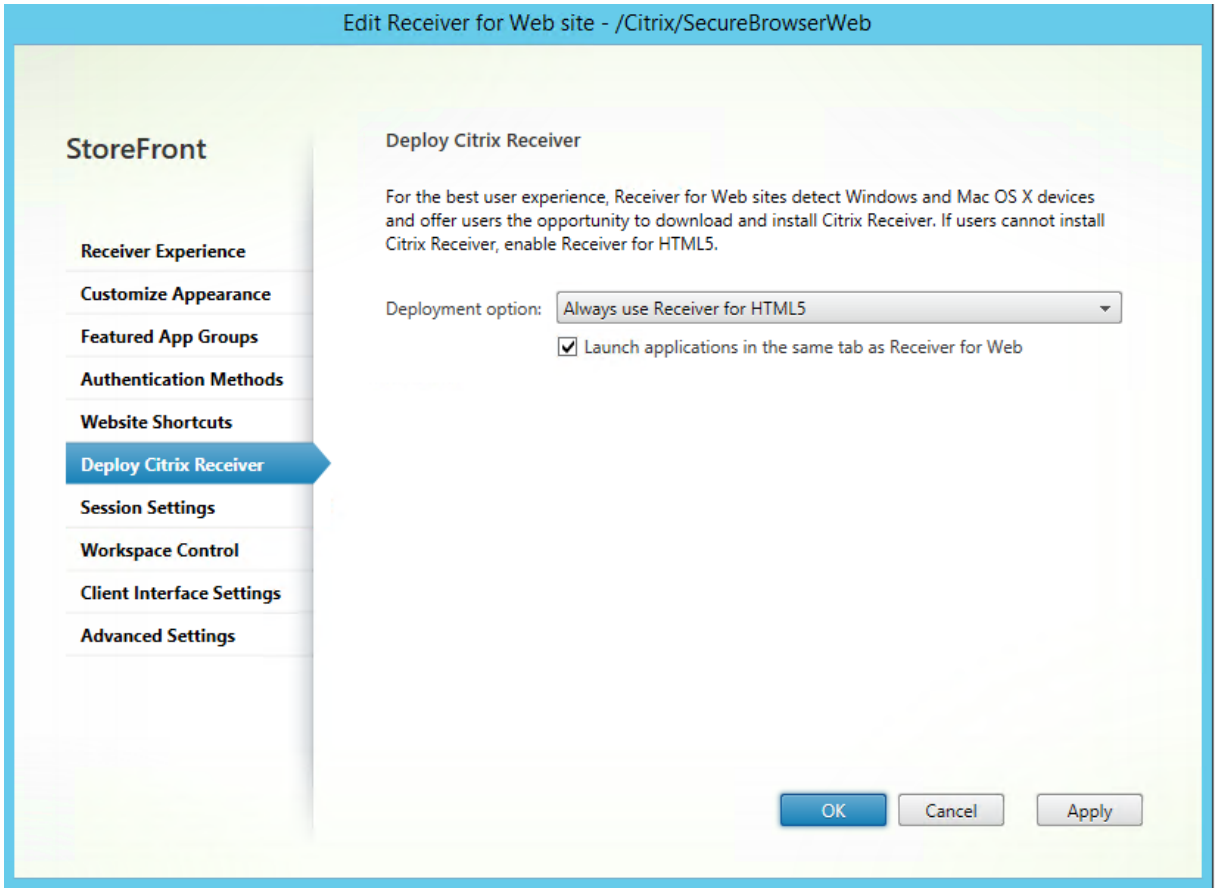


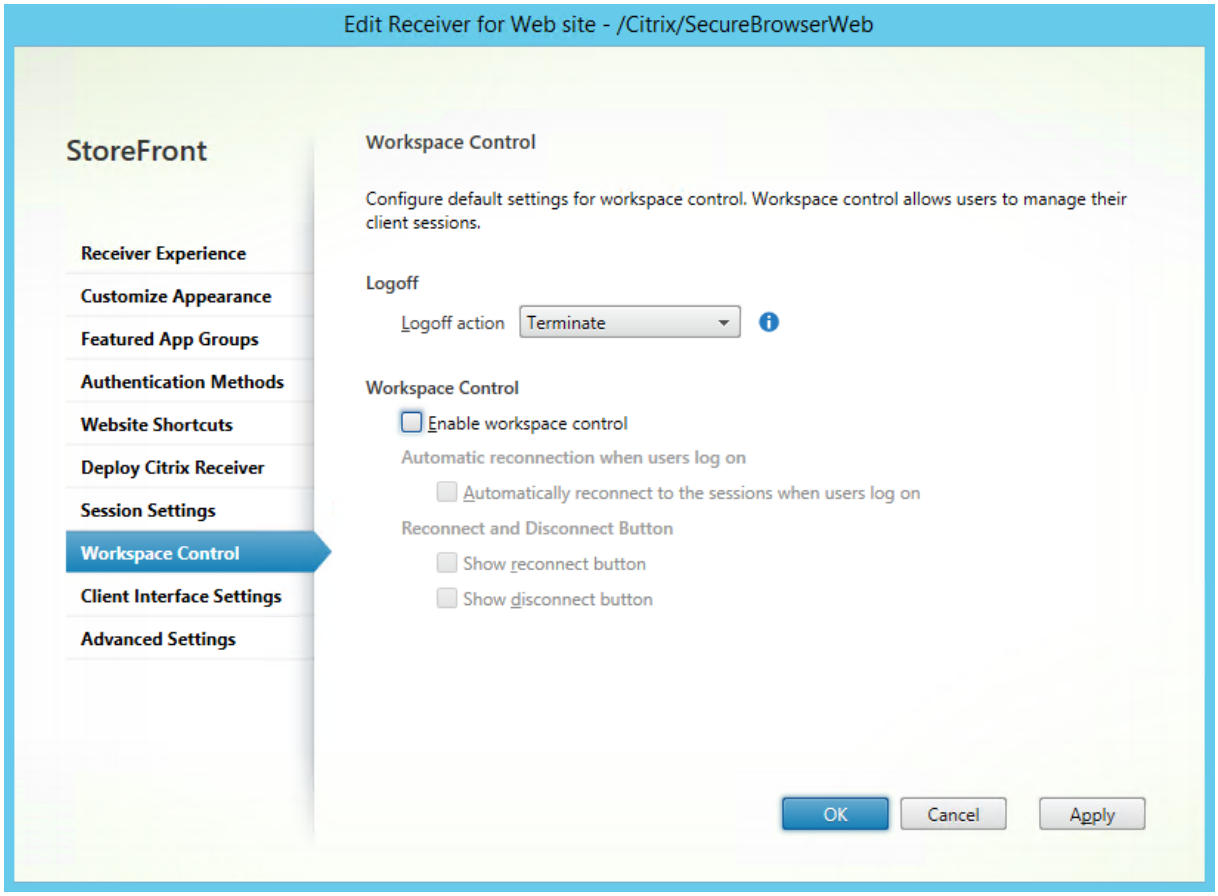


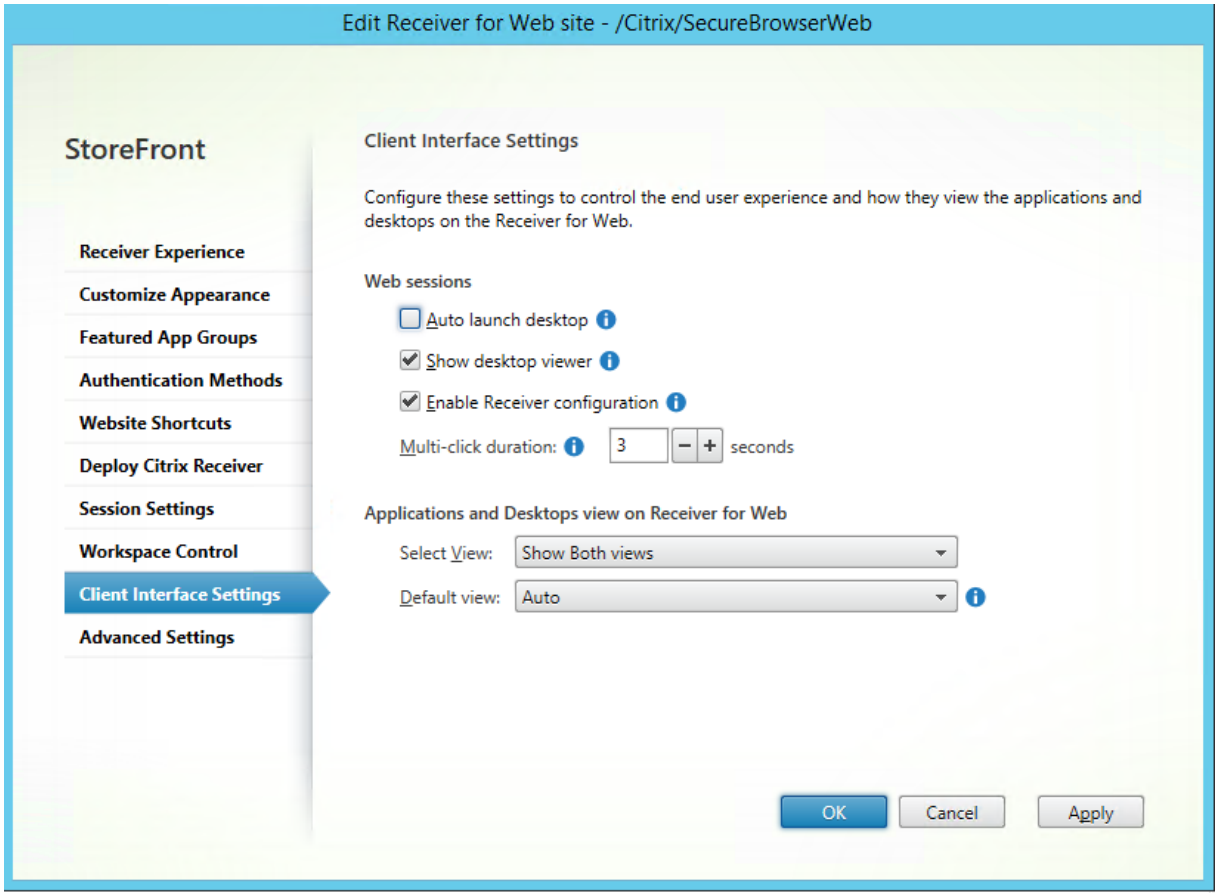


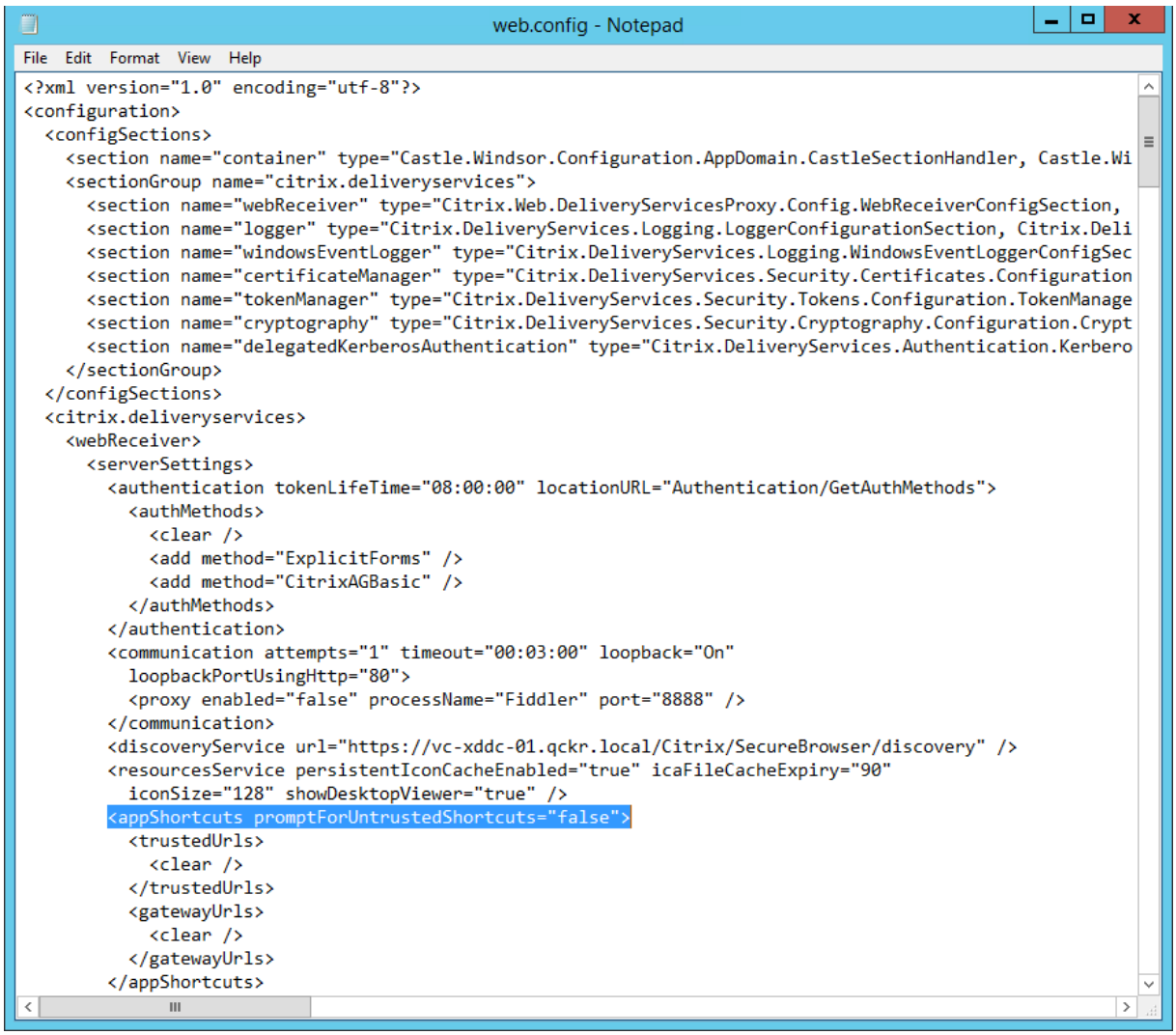












创建名为 **SecureBrowser** 的新应用商店，然后选择“仅允许未经身份验证的用户访问此应用商店”。由于所有用户都将令牌从 NetScaler Gateway 传递到控制器，因此流量将经过身份验证。

添加 XenDesktop 控制器。

启用远程访问并添加将在以下步骤中配置的第二个 NetScaler Gateway。对于此配置，您不需要在 StoreFront/NetScaler Gateway 配置中使用回调或 **VIP** 地址。

使用向导默认值完成创建应用商店。

创建应用商店后，单击管理 **Receiver for Web** 站点。

在“管理 **Receiver for Web** 站点”页面上，单击“配置”，转到“网站快捷方式”，添加内部网站 URL，然后单击“获取快捷方式”链接。

以有权访问已发布的 **Secure Browser** 应用程序的普通用户身份登录。

复制 Secure Browser 应用程序的 URL，并将其保存在文本文件中，以便稍后在 NetScaler Gateway 配置中使用。

返回到编辑 **Receiver for Web** 属性，单击“部署 Citrix Receiver”，然后选择始终使用 **Receiver for HTML5**。** 选择选项在与 Receiver for Web 相同的选项卡中启动应用程序 **。

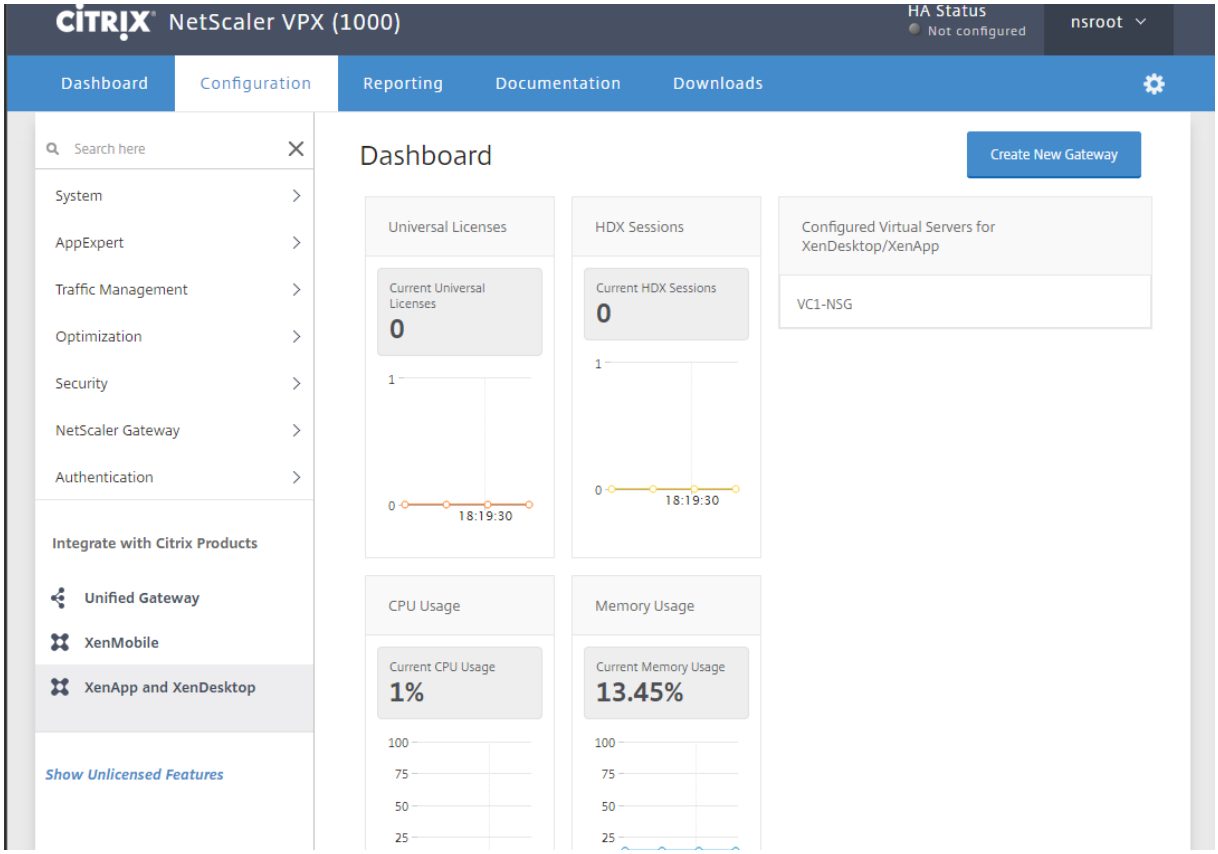
单击 **Workspace** 控制，在注销操作中，选择终止。清除选项启用 **Workspace** 控制。

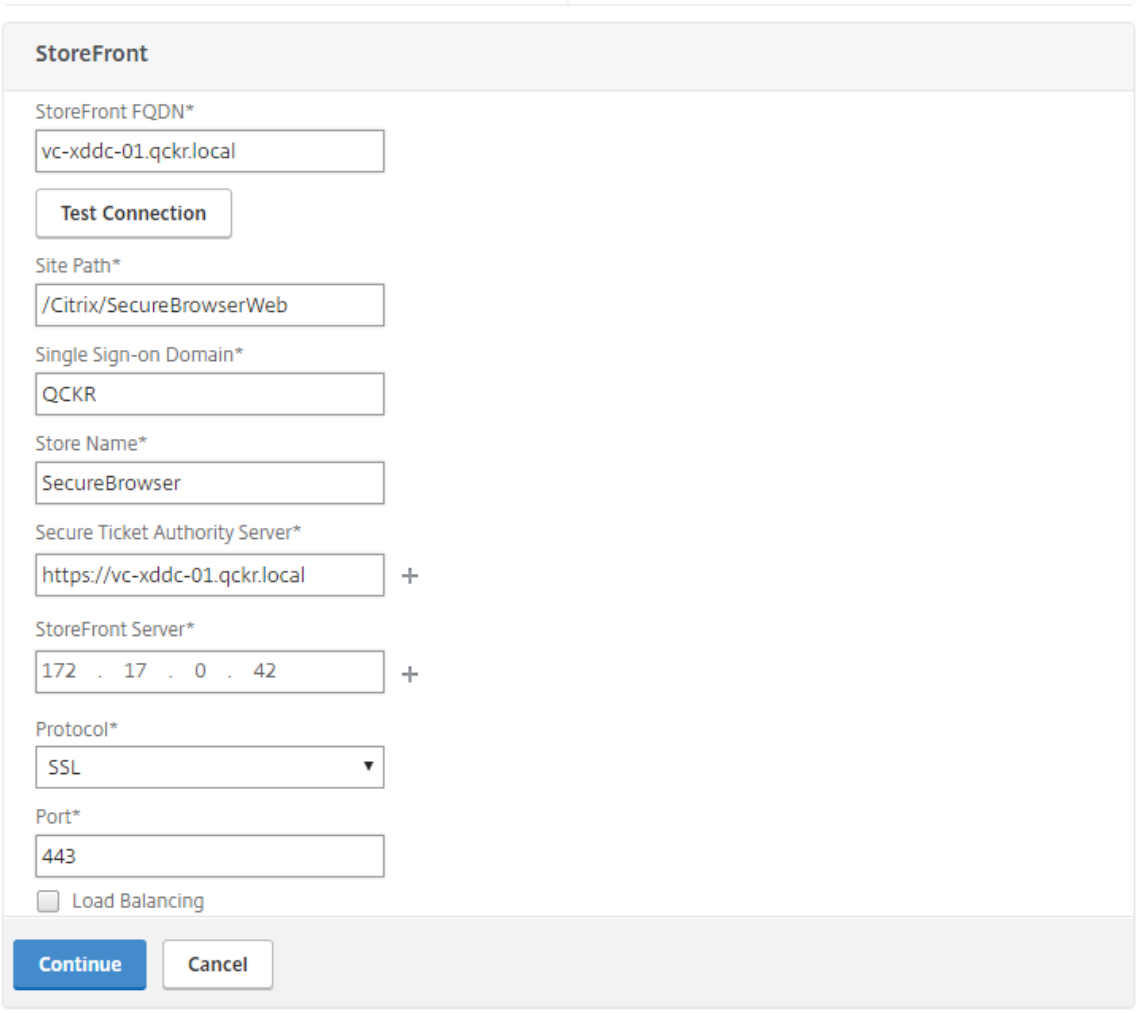
单击“客户端界面设置”，清除“自动启动桌面”选项，然后单 击“确定”以保存设置。

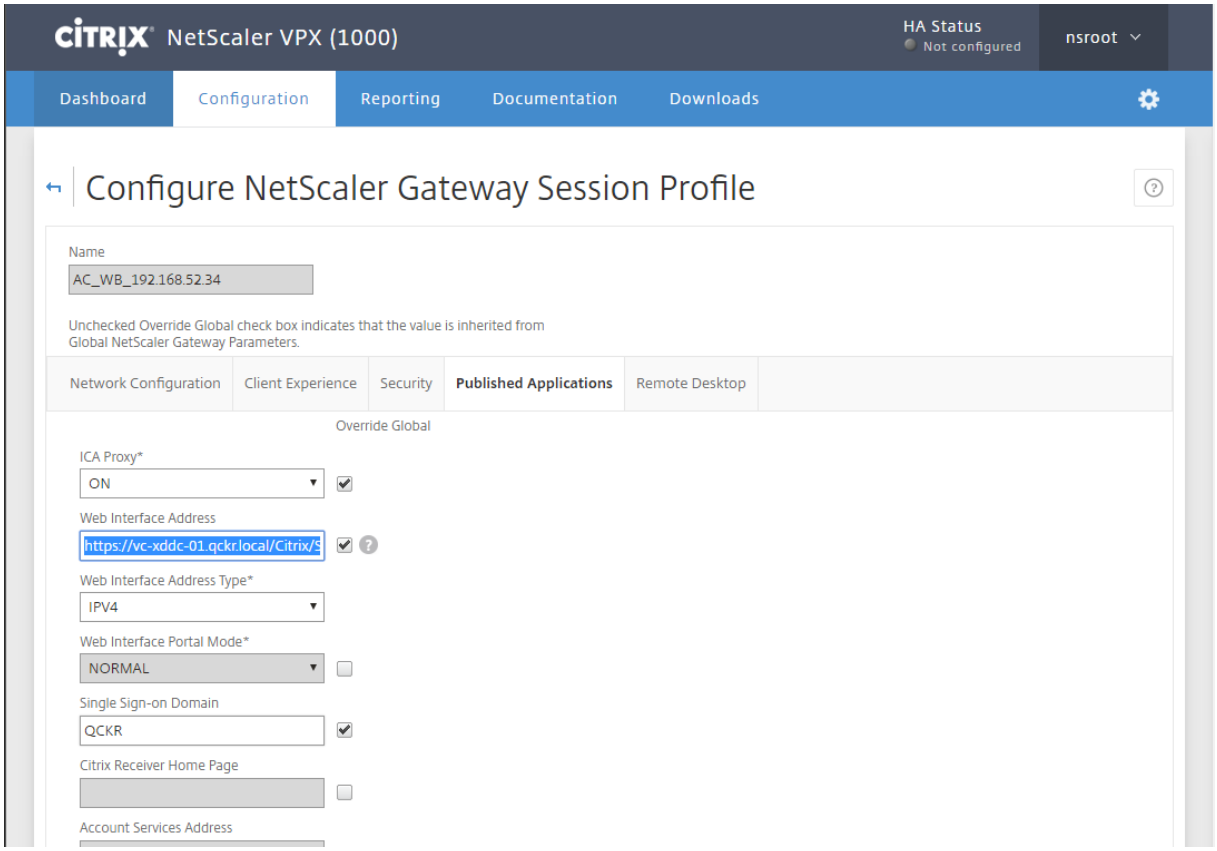
在文本编辑器中，打开文件 C:\inetpub\wwwroot\Citrix\SecureBrowserWeb\web.config。

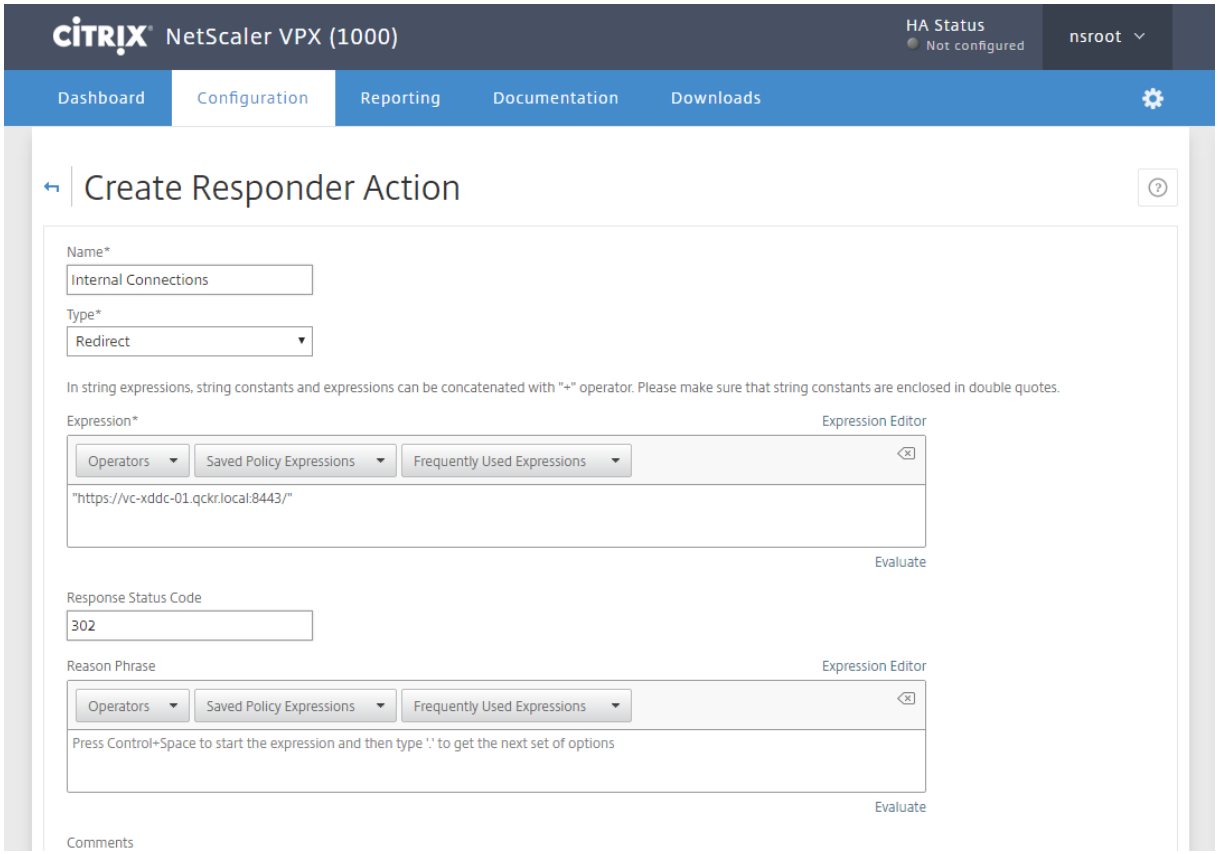
查找设置 <appShortcuts promptForUntrustedShortcuts="true">，将其设置为 false 并保存更改。** 禁用此设置可防止 StoreFront 询问用户是否要启动应用程序。

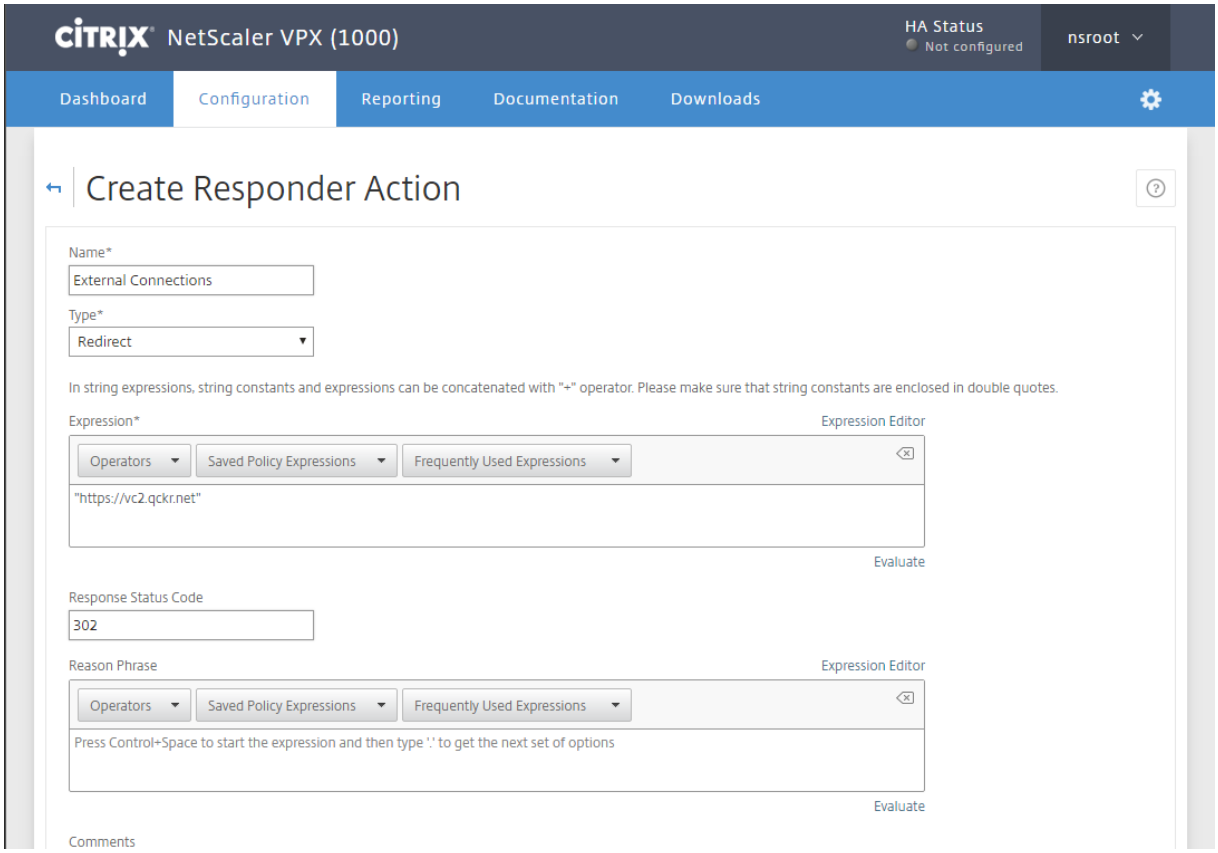
NetScaler Gateway

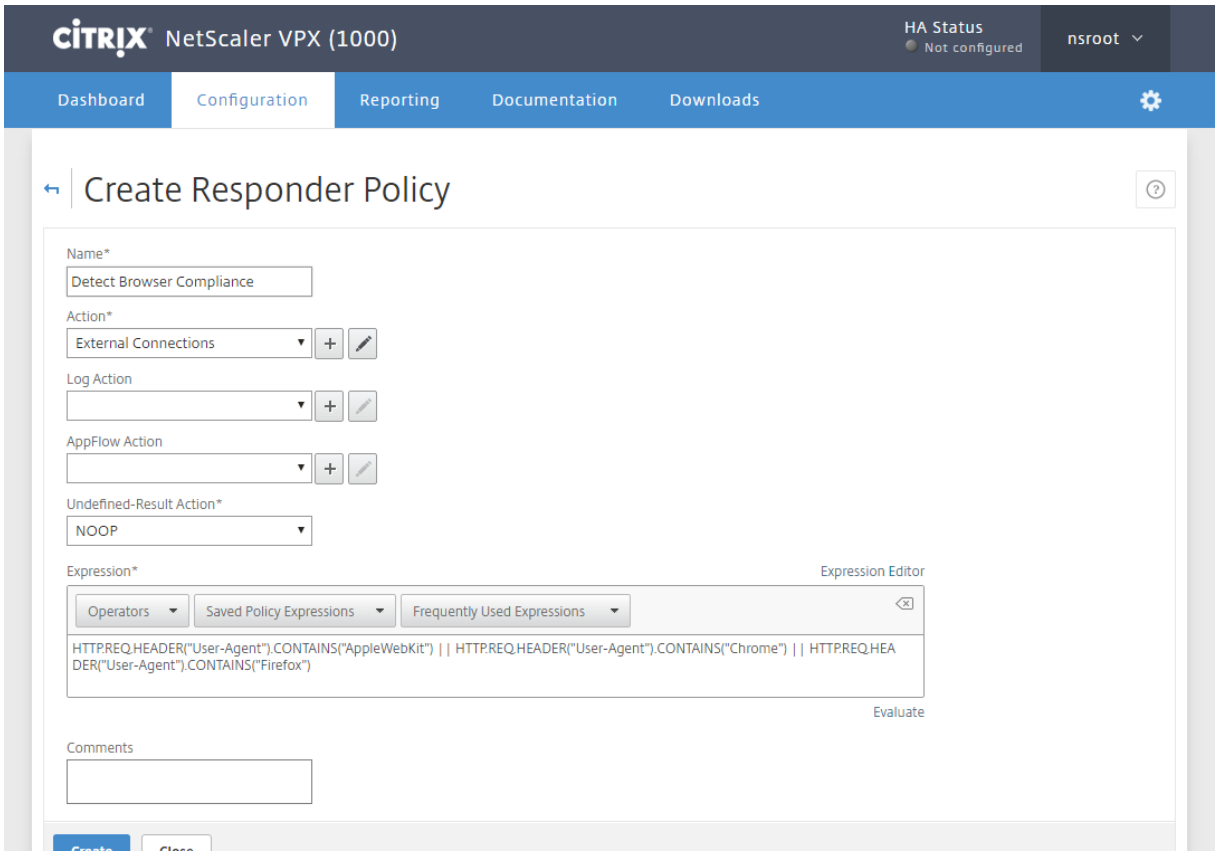


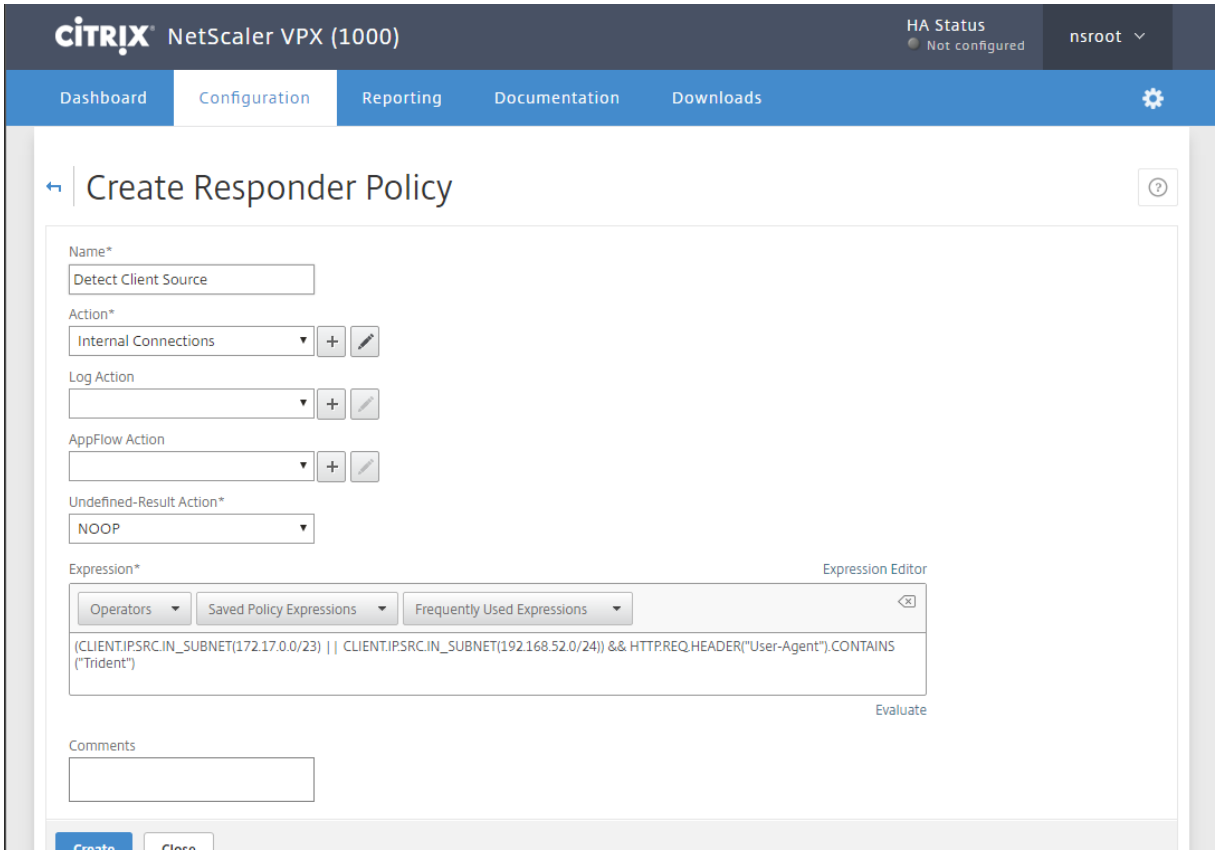


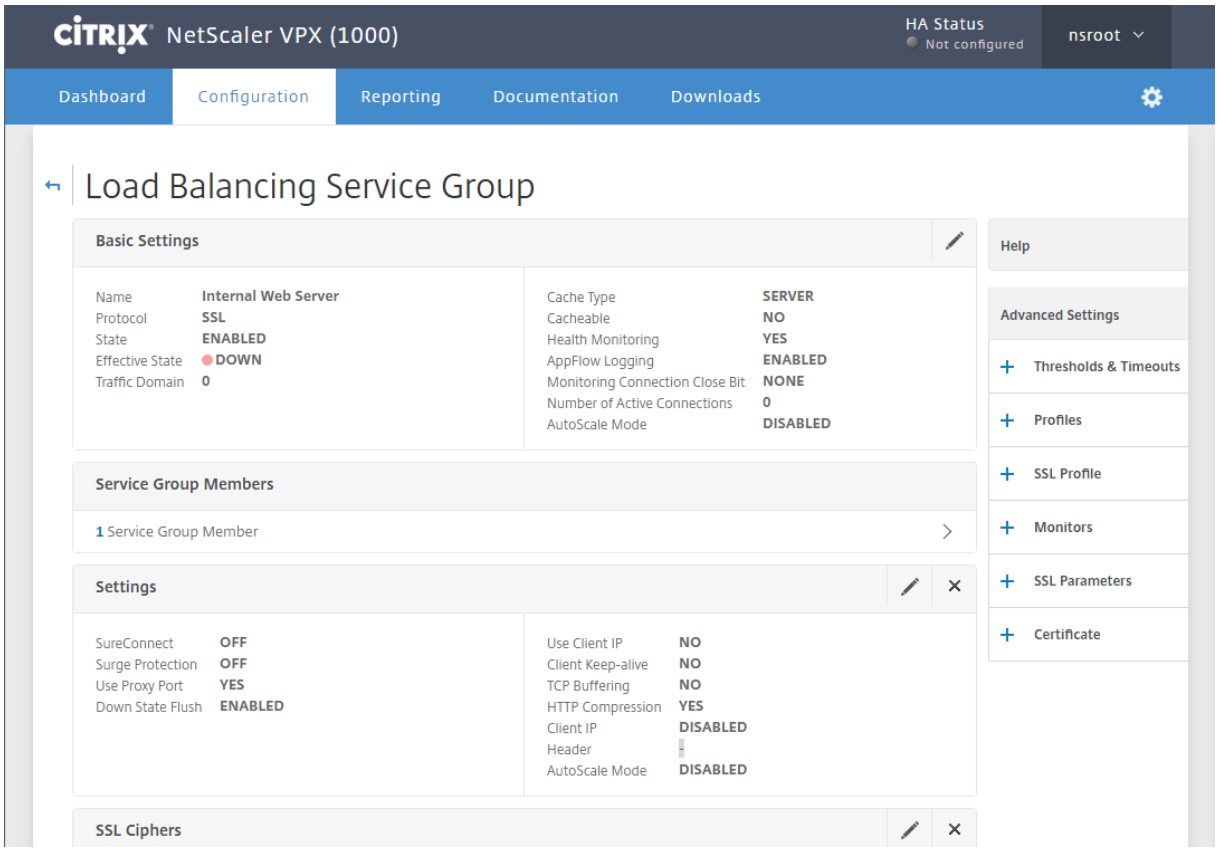


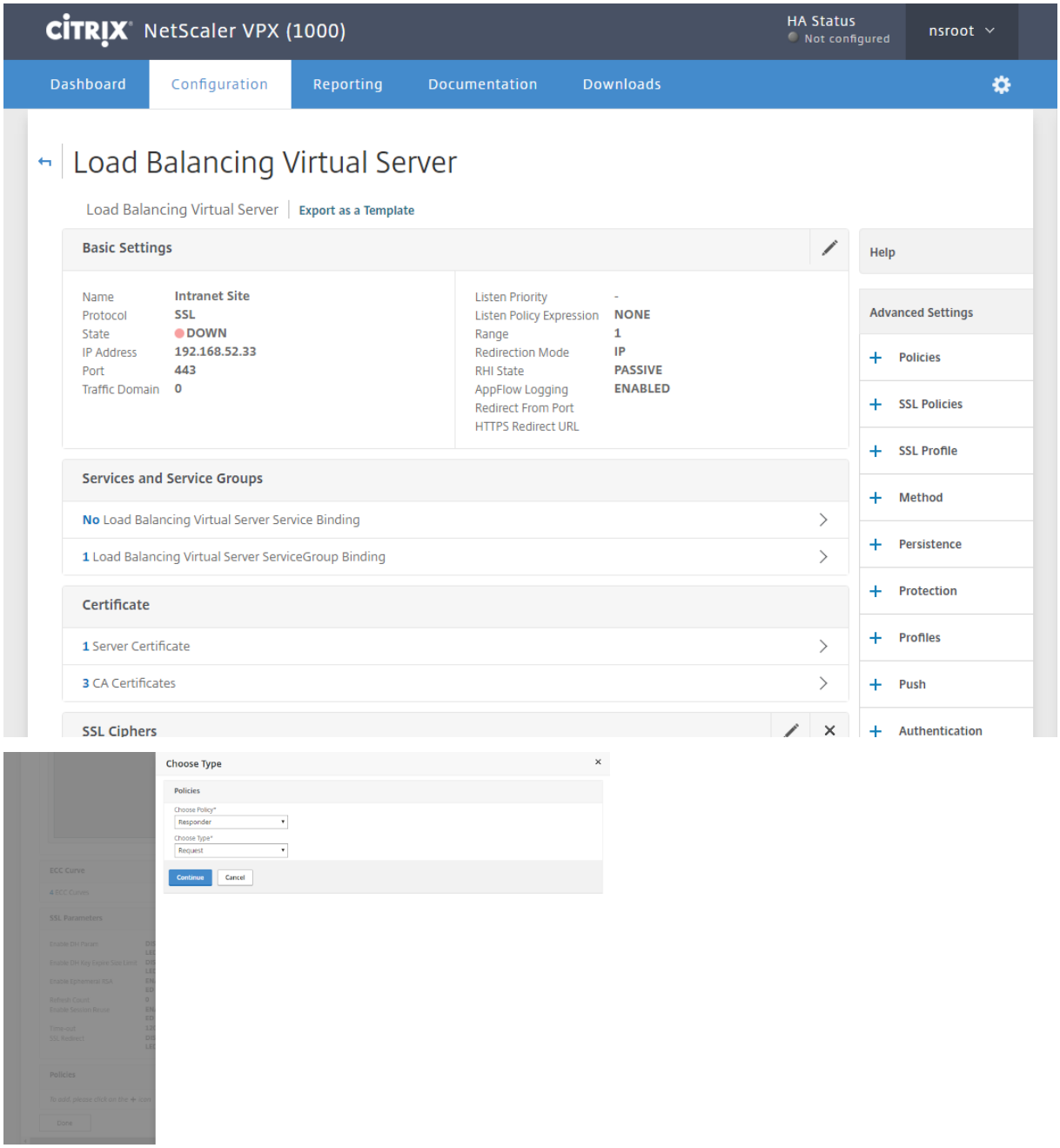


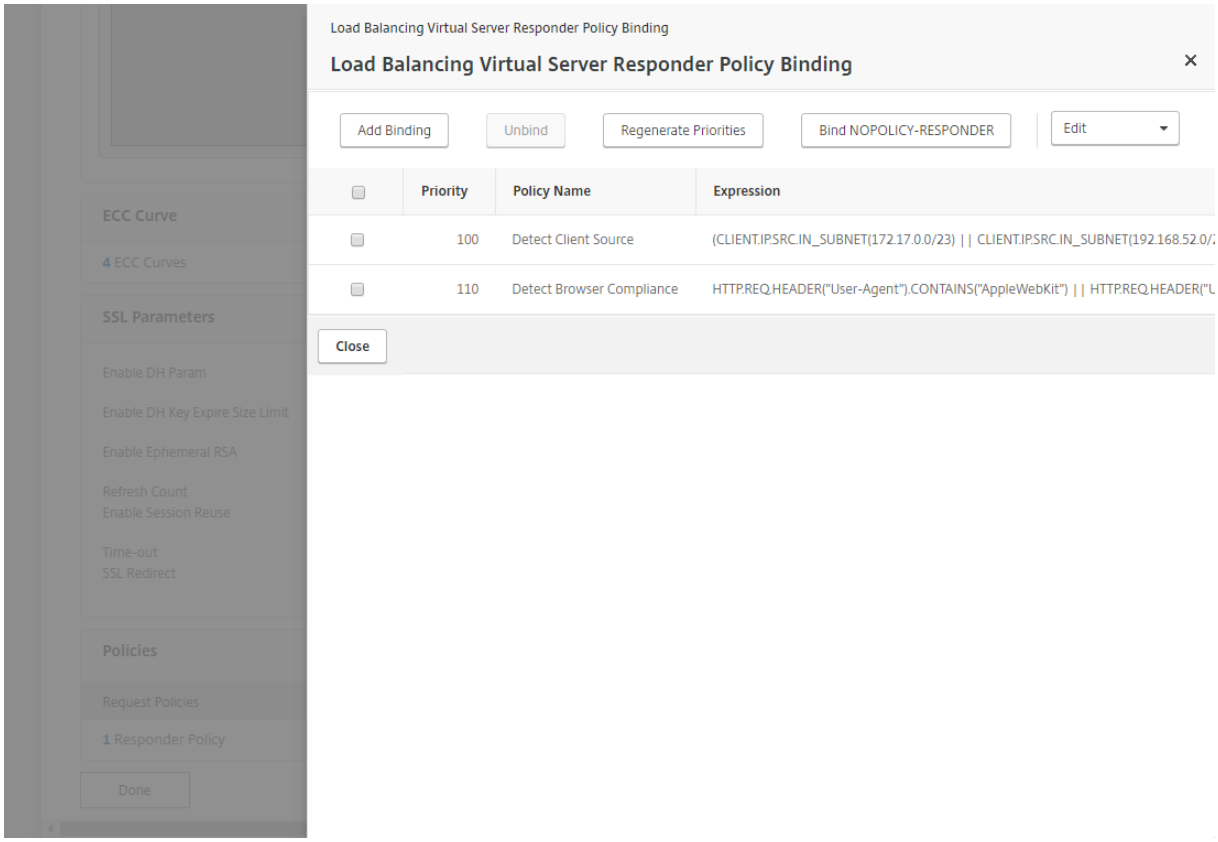












在 NetScaler Gateway GUI 的导航窗格中，单击 **XenApp** 和 **XenDesktop**，然后在控制板上单击创建新网关。

在 StoreFront 属性中，将站点路径设置为 /Citrix/SecureBrowserWeb，并将应用商店名称设置为 *SecureBrowser* 作为 StoreFront 服务器中的新应用商店。

继续向导并保存新的虚拟服务器。

在 **NetScaler Gateway** 节点上，展开策略并转到会话。

选择“操作”选项卡，编辑第二个虚拟服务器的新创建操作，然后编辑 **AC_WB_** 策略操作。

在“已发布的应用程序”选项卡上，粘贴以前保存的“Web 界面地址”字段中的应用程序快捷方式 **URL**，然后单 击“确定”。

在导航窗格中，单击 **AppExpert** 节点，展开响应者部分，然后单击操作。

添加一个新的 操作，将其命名为“内部连接”，并将类型设置为“重定向”。

在“表 达式”字段中，添加要用引号连接的内部站点的 URL，例如`https://mysite.acme.com`

单击创建保存操作。

添加新操作，将其命名为“外部连接”，并将类型设置为“重定向”。

在表达式字段中，添加由引号引起的第二个 NetScaler Gateway 虚拟服务器的 URL，例如 `https://gateway.acme.com`

单击创建保存操作。

转到“响应程序策略”节点。

添加新策略，将其命名为“检测浏览器符合性”，在“操作”下拉列表中，选择您之前创建的“外部连接”操作。

将未定义的结果操作设置为 **NOOP**。

在表达式字段中，添加以下文本：

HTTP.REQ.HEADEI Agent”).CONTAINS	HTTP.REQ.HEADEI Agent”).CONTAINS	HTTP.REQ.HEADER(“User- Agent”).CONTAINS(“Firefox”)

上面的表达式检测不兼容的浏览器，或者在这种用例中不是 Internet Explorer。

单击创建保存更改。

添加新策略，将其命名为“检测客户端源”，将之前创建的“操作”设置为“内部连接”操作。

将未定义的结果操作设置为 **NOOP**。

在表达式字段中，添加以下文本：

(CLIENT.IP.SRC.IN_SUBNET(172.	CLIENT.IP.SRC.IN_SUBNET(192.168.52.0/24)) && HTTP.REQ.HEADER(“User- Agent”).CONTAINS(“Trident”)

替换或添加上述每个子网以匹配您的内部网络环境。在这种情况下，用户代理与配置的 Internet Explorer 版本匹配，并且客户端正在从内部网络连接。

单击创建保存更改。

在导航窗格中，展开流量管理 **>** 负载平衡，然后选择服务器。添加用于托管内部站点的服务器。

在导航窗格中，单击负载均衡下的服务组，添加新的服务组，将协议设置为 **SSL**，并将上一步中创建的服务器绑定到服务组成员列表。

单击完成。

在导航窗格中，单击负载平衡节点中的虚拟服务器，单击添加并命名服务器 *Intranet* 站点。

将 协议设置为 **SSL** ，然后键入负载平衡器的 IP 地址。

绑定在上一步中创建的服务组内部 **Web** 服务器，并配置用于外部访问的证书。将内部根 CA 证书绑定到 CA 证书，以便负载平衡器可以将 SSL 卸载到内部 Web 服务器。

在详细信息窗格的高级设置中，单击 **+** 策略。单击加号 (+) 以绑定新策略。

选择“选择策略**”** 的响应程序，然后单击“继续”。选择“检测客户端源”并将优先级设置为 100。

单击 **Bind**（绑定）。

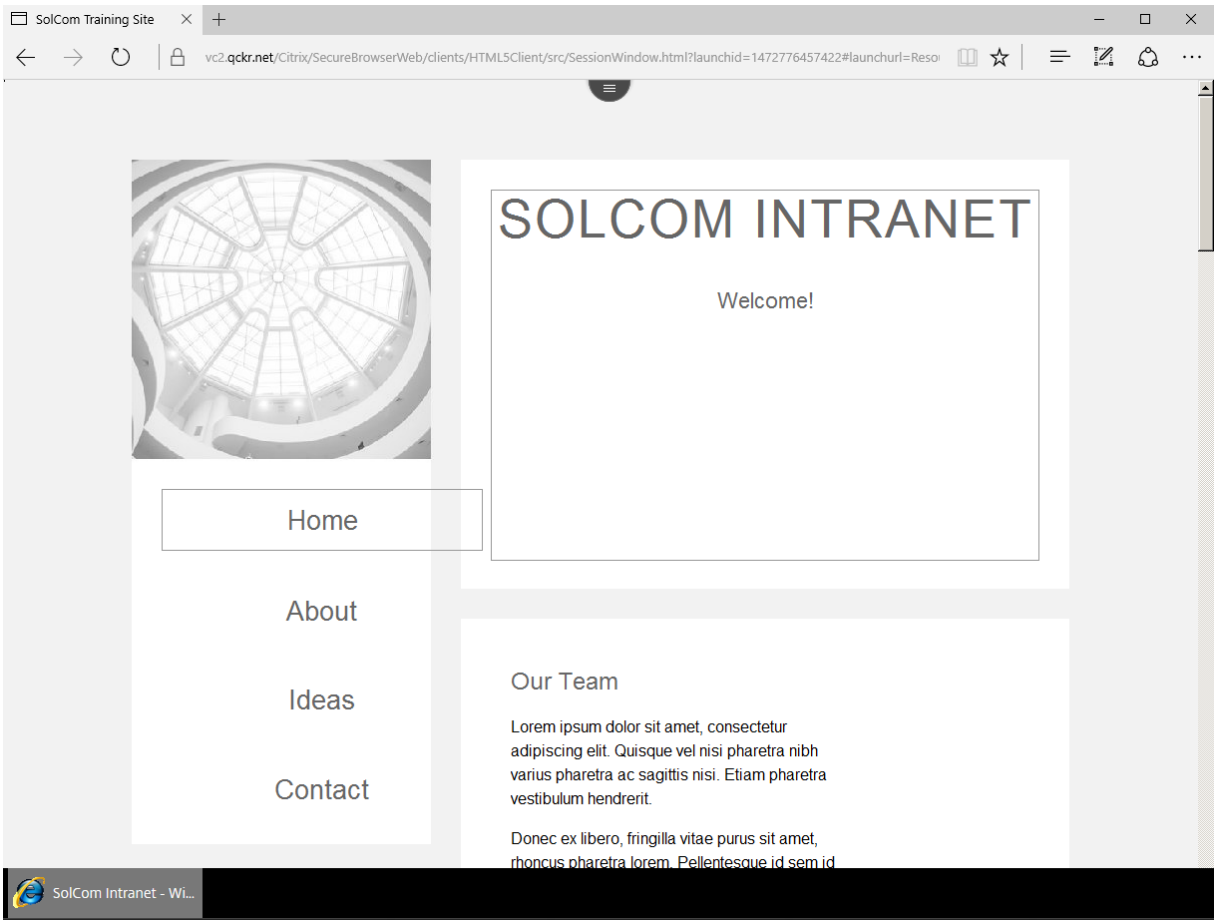
单击响应程序策略部分，单击添加绑定，选择检测浏览器符合性并将优先级设置为 **110**。单击 **Bind**（绑定）。

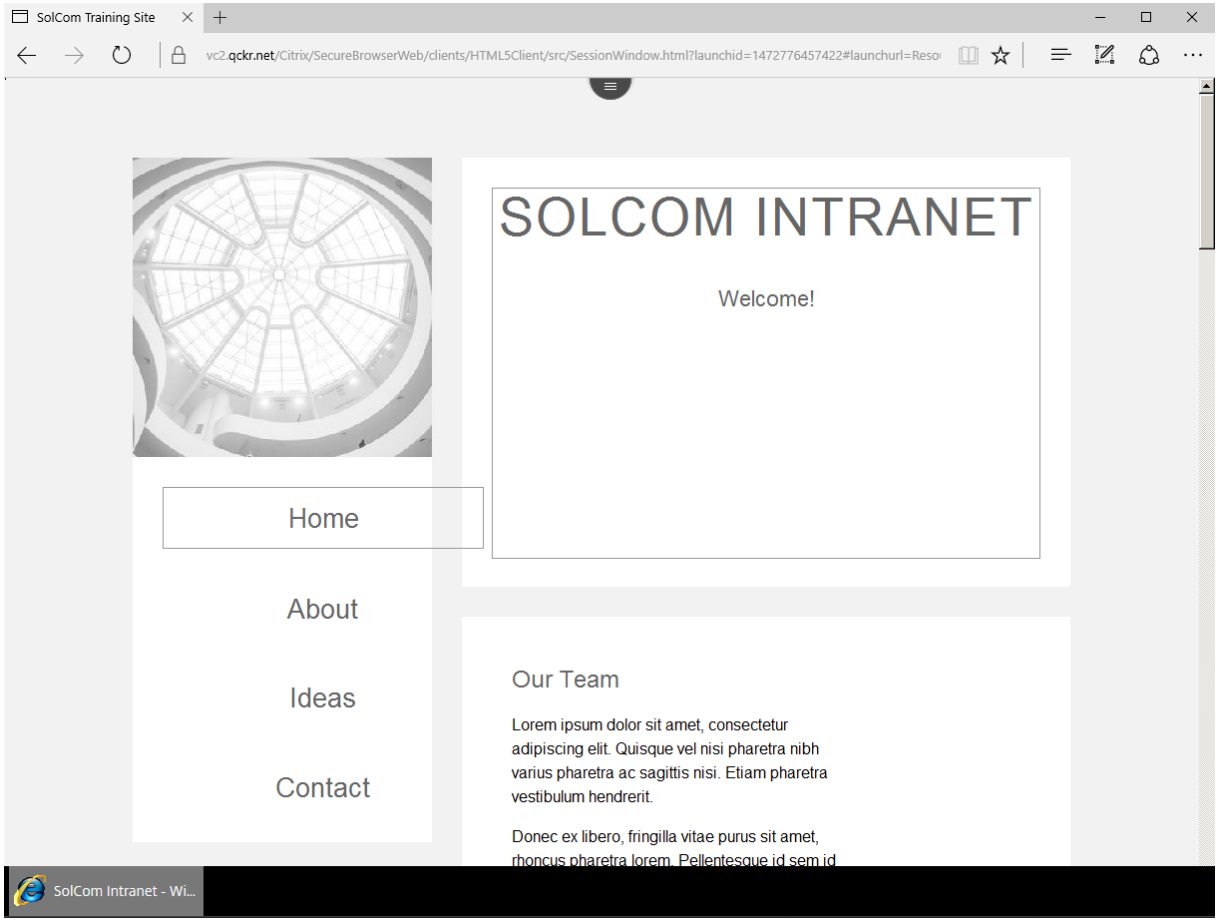
单击“关闭”，然后单击“完成”。

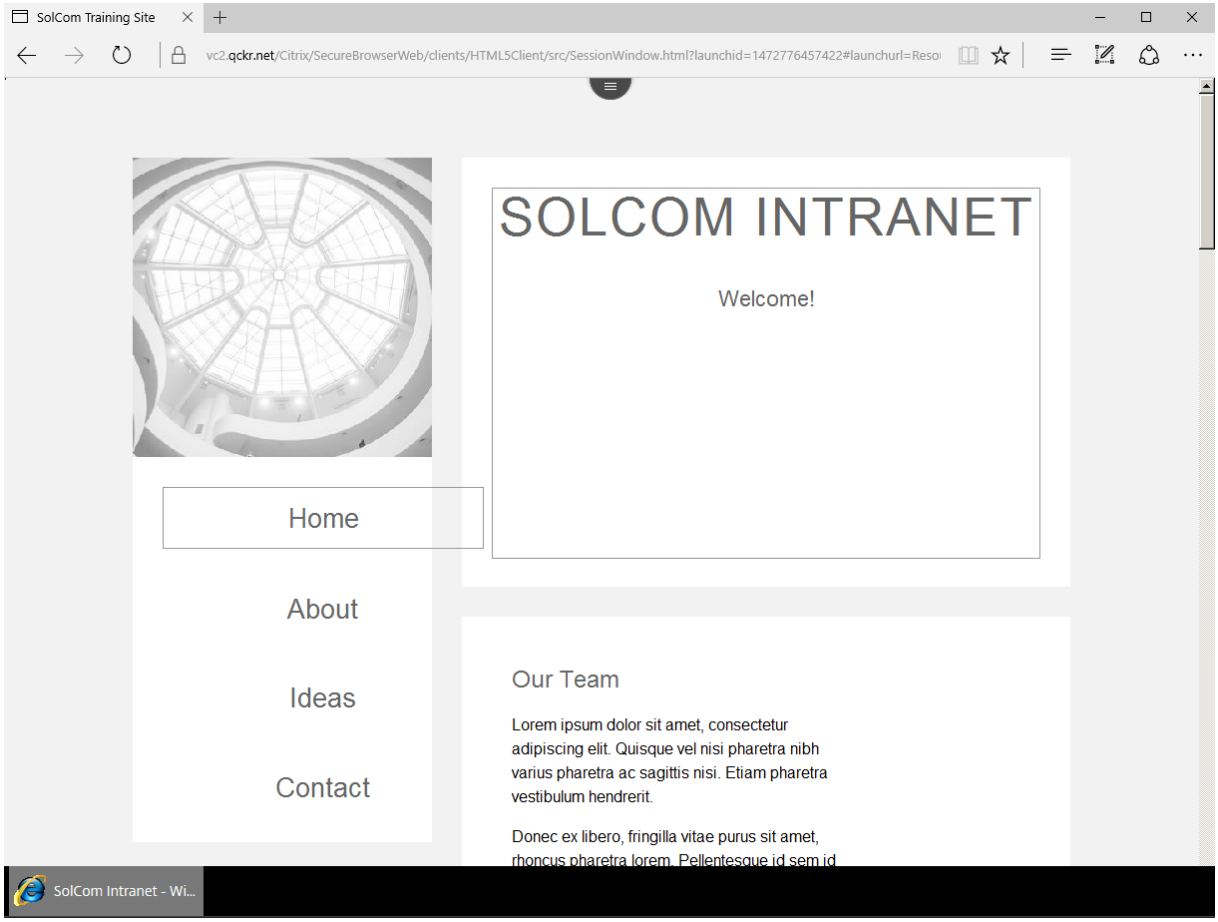
保存 NetScaler Gateway 配置。

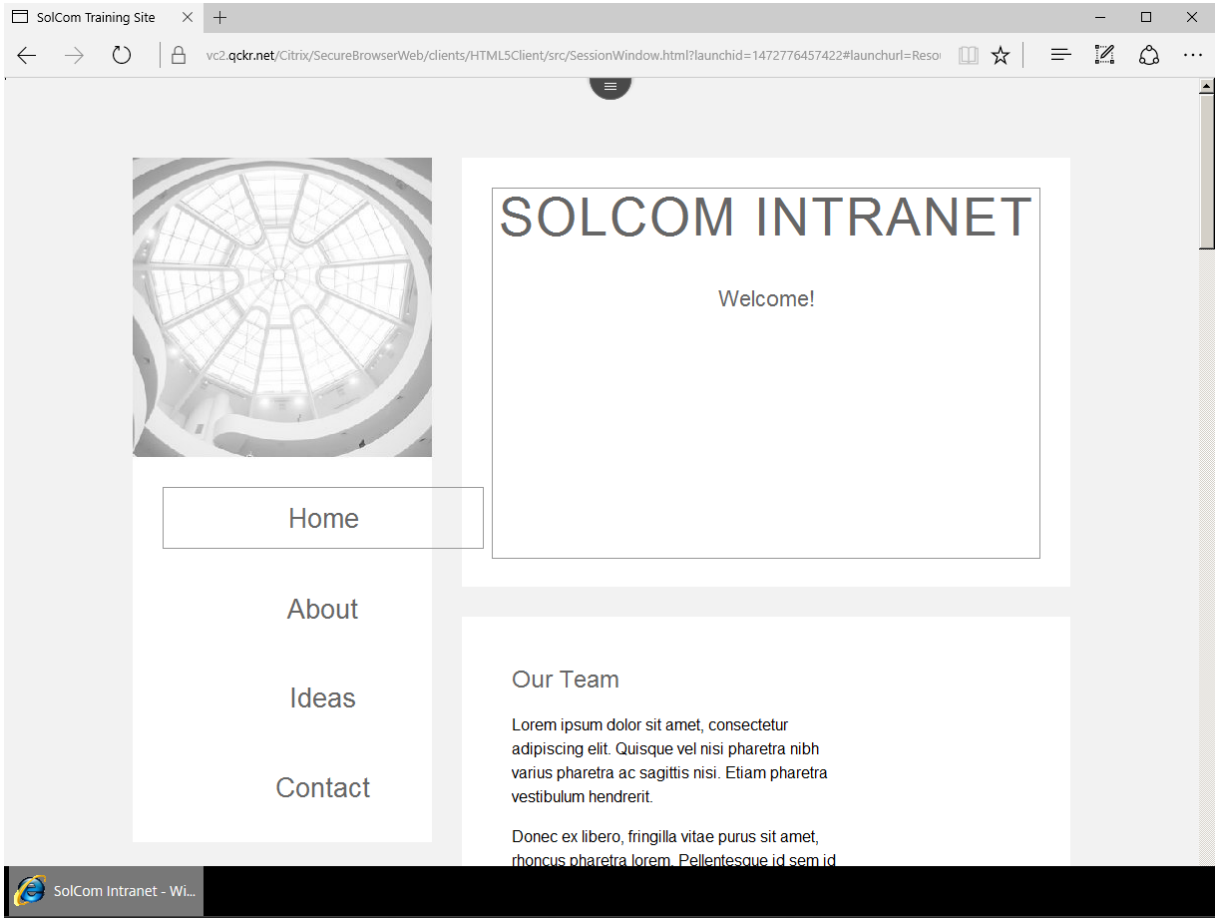
用例结果和期望

本节将回顾每个用户如何与上述配置连接的使用案例和预期结果。在以下所有用例中，用户打开本地安装的浏览器，并键入培训站点的外部 URL。









具有不兼容浏览器的外部用户

预期结果：用户在浏览器选项卡中启动 Citrix Receiver 会话，该选项卡使用已发布的 Secure Browser 呈现站点。

具有兼容浏览器的外部用户

预期结果：NetScaler Gateway 代理本地浏览器和内部网站之间的流量。

具有不合规浏览器的内部用户

预期结果：用户在浏览器选项卡中启动 Citrix Receiver 会话，该选项卡使用已发布的 Secure Browser 呈现站点。

具有兼容浏览器的内部用户

预期结果：用户会话重定向到内部站点；NetScaler Gateway 不代理连接，因为客户端是从内部网络进行连接。

已知限制

- 传递给 NetScaler Gateway 虚拟服务器的动态 URL 不支持将 Citrix Receiver for HTML5 用于 Secure Browser。

  - 要将启动 URL 传递给虚拟服务器，请在会话配置文件中禁用 ICA 代理。ICA 代理是 Citrix Receiver for HTML5 的一个要求。
- Citrix Receiver for HTML5 不支持内容重定向。
  - 管理员可以在 StoreFront 中为 Web 站点配置 Citrix Receiver。
- 具有多个独立站点的环境，为每个站点创建不同的 NetScaler Gateway 会话策略，并将它们绑定到虚拟服务器，或创建一个可以托管内部站点的 URL 的内部启动门户。

引用

- 如何使用 SSL 保护 XenApp 和 XenDesktop 7.6 中的 ICA 连接

本地主机缓存大小调整和扩展

May 20, 2020

在 XenApp 和 XenDesktop 7.12 中引入了本地主机缓存，以替代 XenDesktop 7.6 中提供的连接租用功能。本地主机缓存涵盖的方案比连接租赁更多，但需要不同的设计注意事项。

有关连接租赁功能设计注意事项的详细信息，请访问 https://www.citrix.com/blogs/2014/11/11/xendesktop-7-6-connection-leasing-design-considerations/。

- 本地主机缓存支持比连接租赁更多的用例。
- 运行时，本地主机缓存需要比连接租用更多的资源（CPU 和内存）。
- 在停机模式下，每个区域只有一个代理处理 VDA 注册和代理会话。
- 选举过程决定哪个经纪商在停机期间将处于活动状态，但不考虑经纪商资源。
- 如果一个区域中的任何一个代理在正常操作期间无法处理所有登录，则在停机模式下无法正常工作。
- 中断模式期间没有可用的站点管理。
- 高可用性 SQL Server 仍然是推荐的设计。
- 对于间歇性数据库连接方案，最好是隔离 SQL Server 并使站点处于中断模式，直到所有基本问题都得到解决。
- 每个区域的限制为 5000 VDA（未强制执行）。
- 没有 14 天的限制。
- 在缺省配置中断模式下，不支持池桌面。

体系结构

XenDesktop 7.6 中引入了连接租赁，以允许在站点数据库中断期间继续访问资源。它不支持 VDI 池桌面，默认情况下，用户必须在过去 14 天内连接到资源。另一个限制是，它将尝试将用户连接到其在正常操作期间连接到的最后一个桌面或应用程序主机。如果不可用，则不会中断连接。

这两种技术（连接租赁和本地主机缓存）的体系结构非常不同，它们需要不同的资源来运行。连接租赁创建单独的 XML 租赁文件，这些文件可能需要几 GB 的磁盘空间，具体取决于站点中的资源数。本地主机缓存使用本地 SQL Server 数

据库，并且磁盘空间使用效率更高，但与连接租用相比，所需的内存和 CPU 要多得多。两者都具有同步阶段，其中主站点数据库的详细信息同步到代理（Controller）。由于在文件系统中创建的单个文件数量庞大，连接租赁初始同步可能会导致相当大的 IOPS。尽管本地主机缓存使用仍需要 IOPS 的 SQL 数据库，但它具有 SQL 优化这些写入的优势。

在具有多个代理商的连接租赁设置中，每个代理商都有一个 XML 租赁副本，并且能够在中断期间代理连接，从而帮助平衡负载。但是，使用本地主机缓存，选择一个代理来代理所有连接并处理 VDA 注册。站点中的所有 VDA 都会向这个单一代理重新注册，因此，与正常运行的多代理站点相比，该代理对资源的需求将更高，特别是在具有大量 VDA 的站点中。

本地主机缓存使用 Microsoft LocalDB，该数据库在任务管理器中显示为 sqlserver.exe 进程。已将其配置为使用最多 1 GB 的内存进行数据库缓冲池缓存。但是，由于 SQL 引擎自身和其他较小的缓存需要内存，该过程将超出此范围。通常，中断模式下的中断时间越长，访问的资源越多，LocalDB 内存使用量就越多。但是，当站点数据库连接恢复时，sqlserver.exe 将保留到此内存，而不会立即将其返回到主池。

CPU 插座和内核在中断模式下的影响

在以前版本的 XenApp 和 XenDesktop 中，管理员不一定关心代理（控制器）计算机的 CPU 配置：物理机或虚拟机中的套接字和内核数的布局。

本地主机缓存使用名为 LocalDB 的 SQL Server 的运行时版本，该版本具有特定许可，将其限制为四个内核中较小者或单个套接字。如果物理机或虚拟机已配置多个套接字，则此操作可能会对性能产生显著影响。具有 4 个套接字和每个套接字一个内核的经纪机将限制 LocalDB 使用单个内核，而配置为 1 套接字 4 核心计算机的相同 VM 意味着 LocalDB 可以访问所有 4 个内核（尽管它们与其他进程共享）。在中断模式下，LocalDB 将运行与正常操作期间相同的代理和 SQL 代码。许多 SQL 查询可能占用 CPU 密集型，并直接影响中断模式期间的代理性能。

其他因素包括站点配置本身：

- 已发布的应用程序数量
- 正在中介的用户数
- 用户尝试启动会话的速率
- Active Directory 性能

随着总代理 CPU 利用率接近 100%，代理响应时间将增加，登录将需要更长的时间来处理，并且某些登录尝试可能会失败。

具有多个经纪商的网站

在站点中断模式下，只有一个代理处理注册和登录请求。在多代理站点中，选举过程会发生，以提名在中断期间处于活动状态的代理。然而，这一选举进程并没有考虑到经纪人可以利用的物质资源。这意味着，在经纪商拥有不同资源量的站点中，选定的代理商不一定是 CPU 或 RAM 方面最强大的代理商，这可能导致停机模式下的性能不佳。重要的是，每个代理必须满足本地主机缓存的附加要求，以防它被选中。

与站点数据库同步

CitrixConfigSync 服务处理从站点数据库导入到经纪商的本地副本中的数据。它监视站点数据库是否有对站点配置的更改，并在发生更改时触发新的导入。导入开始之前创建当前本地数据库的副本。站点中的资源（如 VDA）数量越大，导入所花费的时间就越长，但对于具有 5000 个 VDA 的站点，导入时间应少于 10 分钟。

数据库位置

本地数据库存储在：

C:\Windows\ServiceProfiles\NetworkService\HaDatabaseName.mdf

为了确保可靠性，CitrixConfigSync 服务在开始新的站点数据库同步之前对先前成功的同步数据库导入进行备份。如果由于任何原因同步未能成功，则会使用备份，直到同步成功完成。不应手动复制数据库。

本地主机缓存与连接租赁的比较

	连接租用	本地主机缓存
磁盘空间	推荐 2 GB	取决于站点配置。对于拥有 125 K 用户的 50 个 RDS 主机，使用 300 MB。
RAM	100 MB	3 GB，~ 1 GB 用于 SQL Server，2 GB 用于高可用性服务和 CitrixConfigSync 服务。
同步配置的时间	取决于 IOPS；40000 个 VDA：约 26 分钟	5,000 个 VDA：约 7 分钟
停机期间激活的时间	150 秒，30 秒默认 SQL 超时 + 120 秒等待时间（可配置）	取决于 VDA 的数量以及与代理的上次注册同步。在停机模式下，只有一个代理可用于 VDA 注册，因此对于大量 VDA，可能需要几分钟时间才能注册所有 VDA。
恢复正常操作的时间	租赁停用 120 秒，然后 VDA 必须向经纪商重新注册	如上所述，VDA 需要从辅助代理中取消注册，然后在主代理中重新注册。
支持的 VDA 数量	50000	5000。一个站点的数量可以超过这个数量，但同步站点数据库所需的时间将随着 VDA 的数量而增加。具有大量 VDA 的单个代理的性能可能会导致某些连接在中断期间无法中断。

	连接租用	本地主机缓存
停机期间的站点管理	否	否

配置 **LocalDB** 内存限制

LocalDB 进程被限制为 1 GB 的 RAM 用于缓存，通常不需要更改。如果需要，可以更改此值；为了使更改生效，站点必须处于正常操作模式（而非中断模式），并且必须强制站点数据库同步。

要将内存减少到 768 MB：

步骤 **1.** 编辑文件 C:\Program Files\Citrix\Broker\Service\HighAvailabilityService.exe.config。

查找 <appSettings> 部分并添加一个条目：

```
<add key="MaxServerMemoryInMB" value="768" />
```

步骤 **2.** 要强制数据库同步，请停止高可用性服务。如果 sqlserver.exe 正在运行，请使用任务管理器或 PowerShell 将其停止。

步骤 **3.** 现在对站点进行微不足道的更改，例如，通过添加 "." 来更改交付组的描述，然后再次将其删除。然后，启动高可用性服务; 这应该启动 sqlserver.exe 并应该进行同步。

过多地减少内存会影响性能，因此不建议使用。但是，将其增加到 2 GB 不会显著提高性能，而 CPU 资源比 RAM 更多的瓶颈。

启用或禁用本地主机缓存

可以禁用连接租赁和本地主机缓存，但一次只能激活其中一个。

```
Set-BrokerSite –ConnectionLeasingEnabled $False

Set-BrokerSite –LocalHostCacheEnabled $True
```

限制

必须已分配台式机，然后才能在停机模式下使用。未分配的桌面将不可用于代理。这可能导致桌面不可用，并在所有分配同步之前发生停机时报告 "处于维护模式"，尽管用户实际分配了桌面。

在缺省配置中断模式下，不支持池桌面。有一个解决方法，但它可能对安全和性能产生潜在影响。如果将包含池桌面的交付组配置为在注销时不重新启动，则该组中的任何已打开电源的池桌面都将在停机模式下可用。但是，用户注销后，桌面将不会处于干净状态，因为桌面没有重新启动。这在任何情况下都可能是一个安全问题。如果该桌面的下一个用户是该桌面的本地管理员，则可以访问以前用户的数据。虽然对于标准（非管理员）用户来说，这种风险不是一个问题，但请记住，应用程序可能行为不正确，并随着时间的推移导致性能问题。重要提示：管理员应仔细考虑在停机模式下使用未重新启动的池桌面时使用此替代方法的潜在影响。

与连接租赁一样，在停机期间不能进行站点更改；数据库实际上是主站点数据库的快照，每次发生新同步时都会丢弃。

压力条件下 **6** 和 **8 vCPU** 代理的性能比较

一个站点配置了 50 个 RDS 工作程序和 5075 个 VDI VDA。每个 RDS 工作人员都能够支持 2500 个模拟用户。设置了每秒 20 个用户的启动率。向站点中添加了不同数量的已发布应用程序。

对于 RDS 工作人员，为 VDI 5075 启动了 100000 个用户。

测试是在正常和本地主机缓存操作（中断）模式下执行的。

在 6 vCPU 系统中，CPU 余量非常小，而且少数例外情况 0 <10) occurred when the published app count was >。

Windows 更新被禁用的组策略，因为在多次运行期间，tiworker.exe 进程（Windows 安装程序模块）被发现使用几乎整个核心很长一段时间。这导致了大量启动失败，因此测试被重新运行。代理处理器相当旧，但 tiworker 进程将消耗一个较新的核心，从而影响测试。

Hypervisor 配置

- XenServer 7.0
- AMD Opteron 8431 2.4 GHz – 4x6 个核心
- 128 GB RAM
- 已启用读缓存
- 配备基于 SSD 的存储的 Windows Storage Server 2012R2

Broker 配置

- 带 3 个核心的 2 个插槽，带 4 个核心的 2 个插槽
- 10 GB RAM
- Windows Server 2016
- 每种 VDA 类型的单个交付组
- 用户可见的所有应用程序
- XenDesktop 7.12

StoreFront 配置

- 6 个 StoreFront 服务器
- Windows Server 2016
- 4 vCPU
- 10 GB RAM

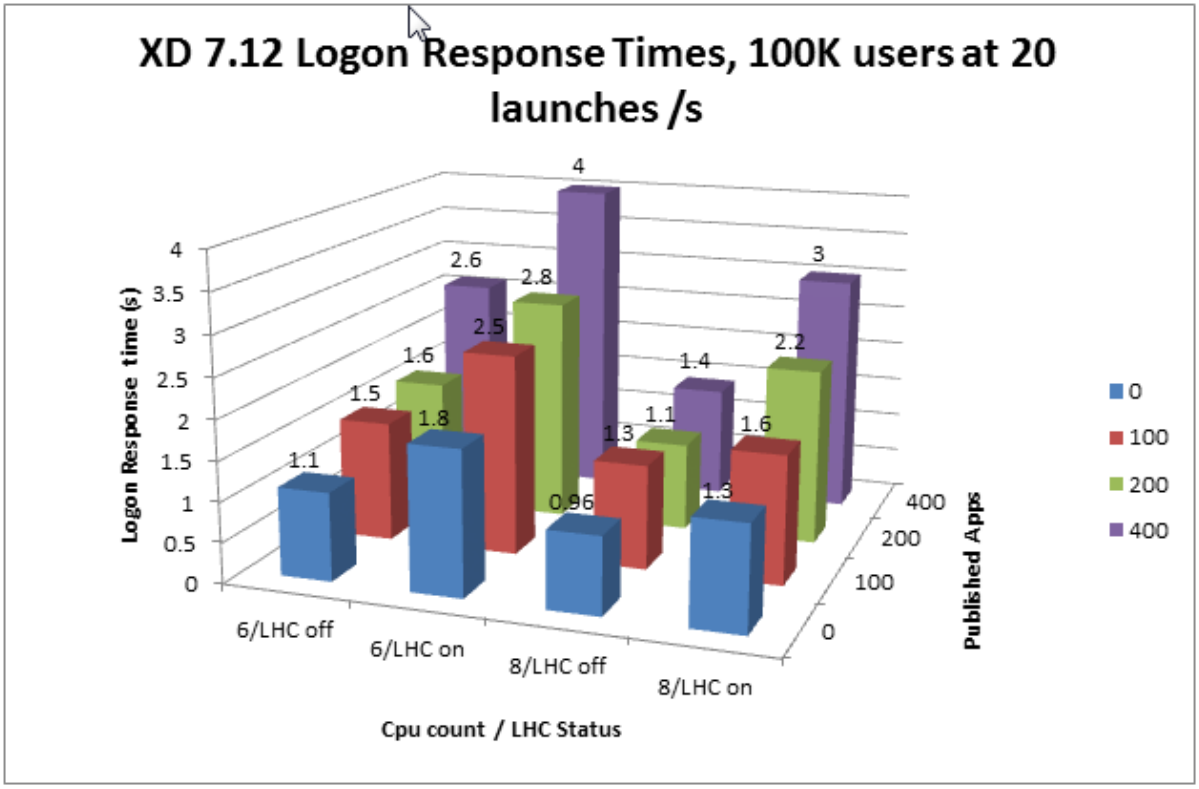
NetScaler 配置

- VPX 8000 版本 11.063.16

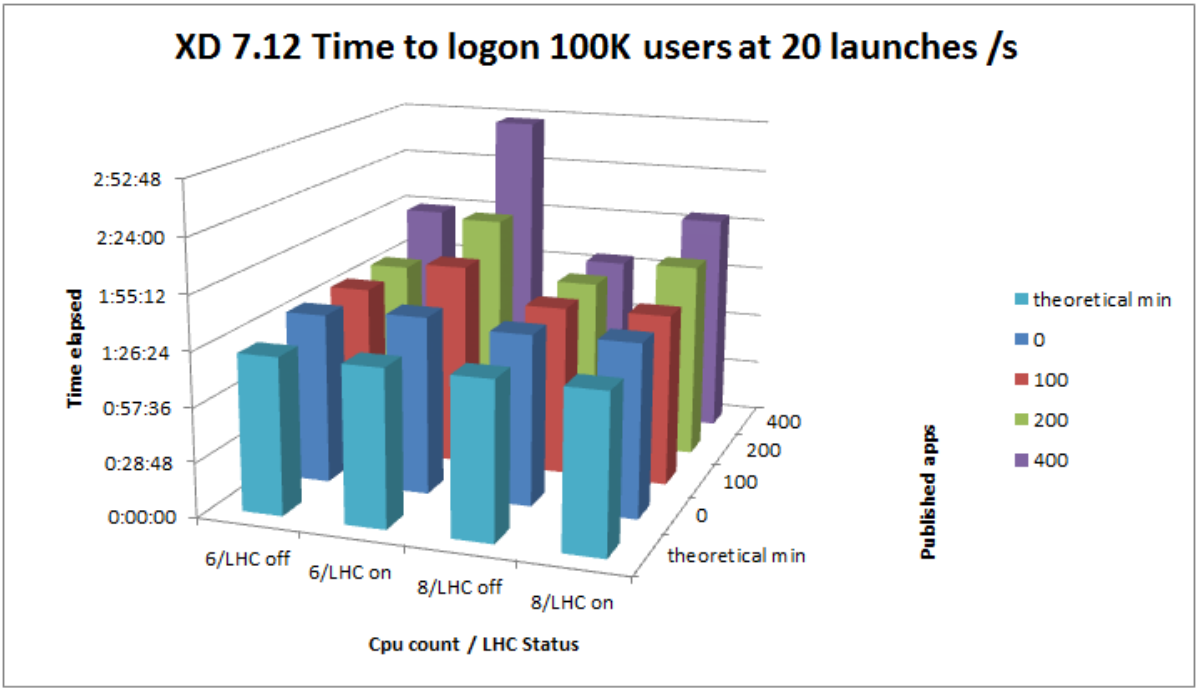
SQL Server 配置

- Intel E5-2630 2.3GHz 2x12 个核心
- SQL Server 2012
- 64 GB RAM

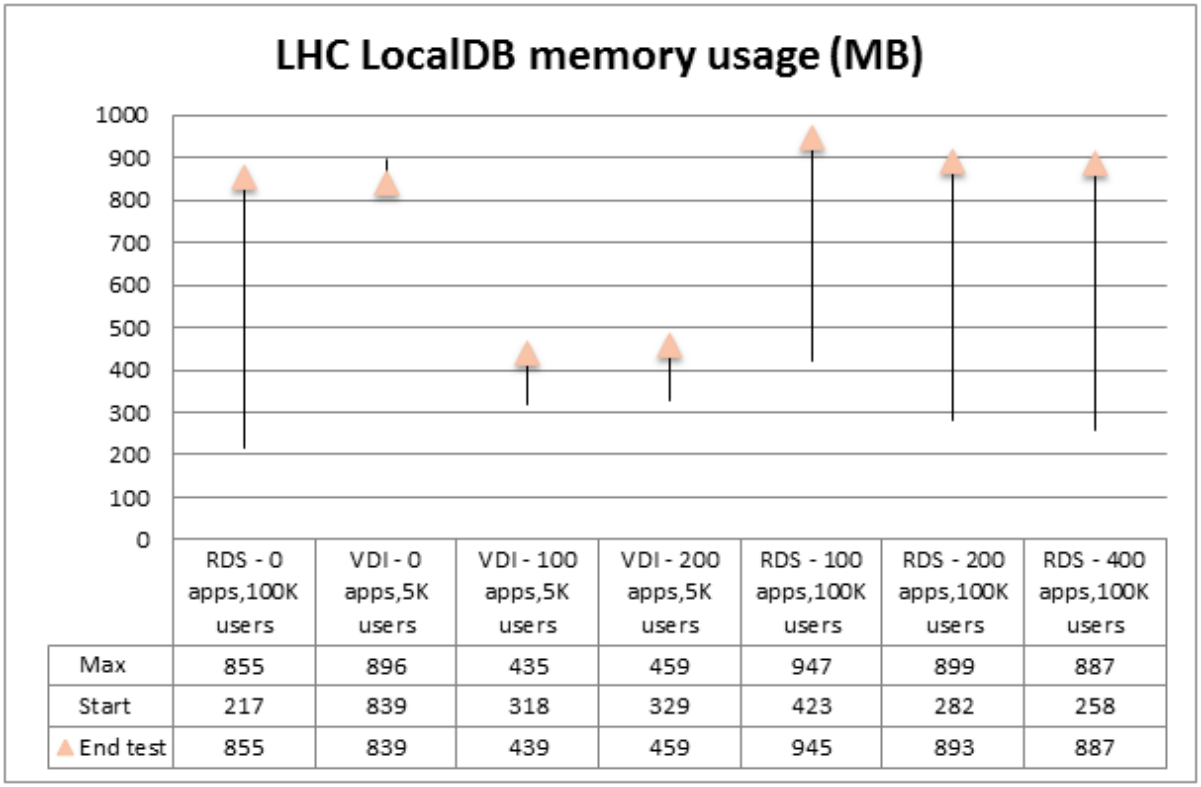
注意：只有站点数据库处于脱机状态；监视和配置数据库在测试期间仍可访问。这意味着监视器服务在测试期间占用了一些 CPU。



通常，随着应用程序数量的增加，登录响应时间会增加，停机响应时间会比正常操作要差。增加 vCPU 数量可缩短响应时间。请记住，测试中使用的 CPU 相当旧，而且更现代的 CPU 通常会提供更好的性能。

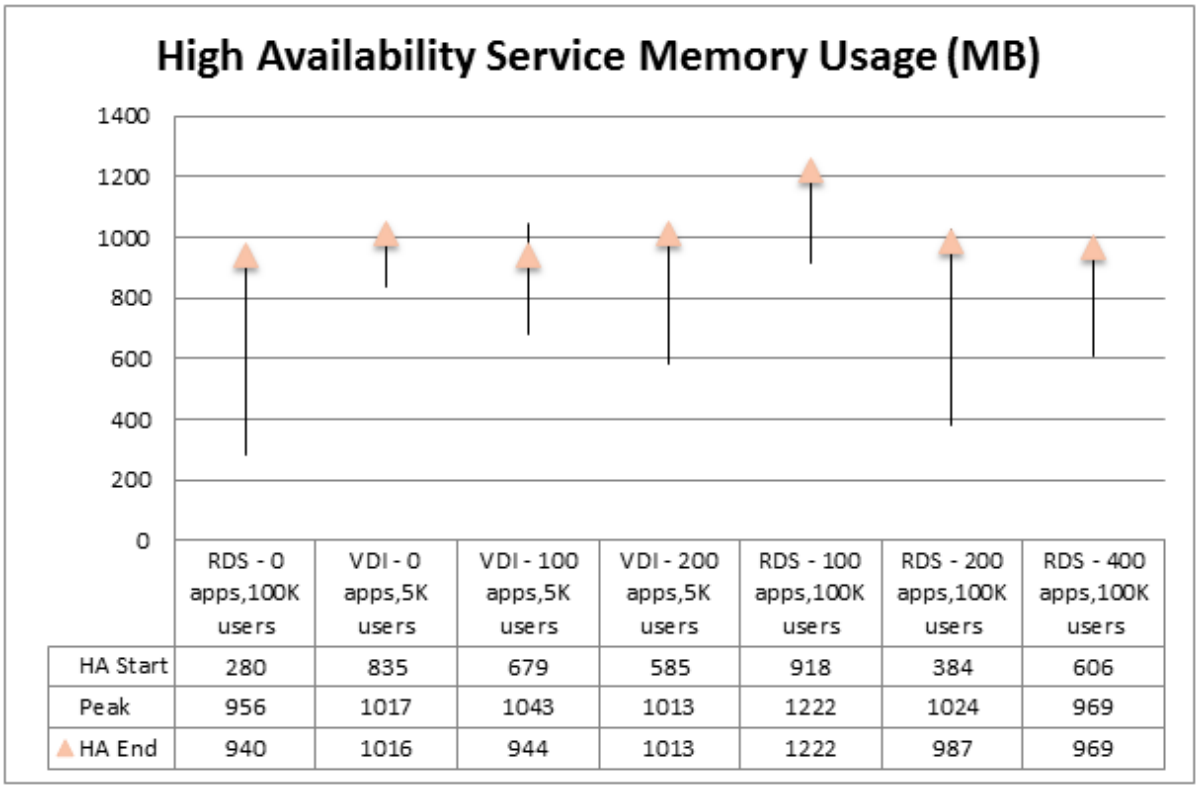


“理论上最小”行是，如果环境能够每秒处理 20 次启动，则 100,000 名用户登录所需的绝对最短时间，给予 1 小时 23 分 20 秒。在这些测试中，0 个应用程序行在 6 个 vCPU 案例中管理 1:30:57，在 8 个 vCPU 案例中管理 1:30:48。Active Directory 域的性能将对用户进行身份验证的速度产生一定的影响。

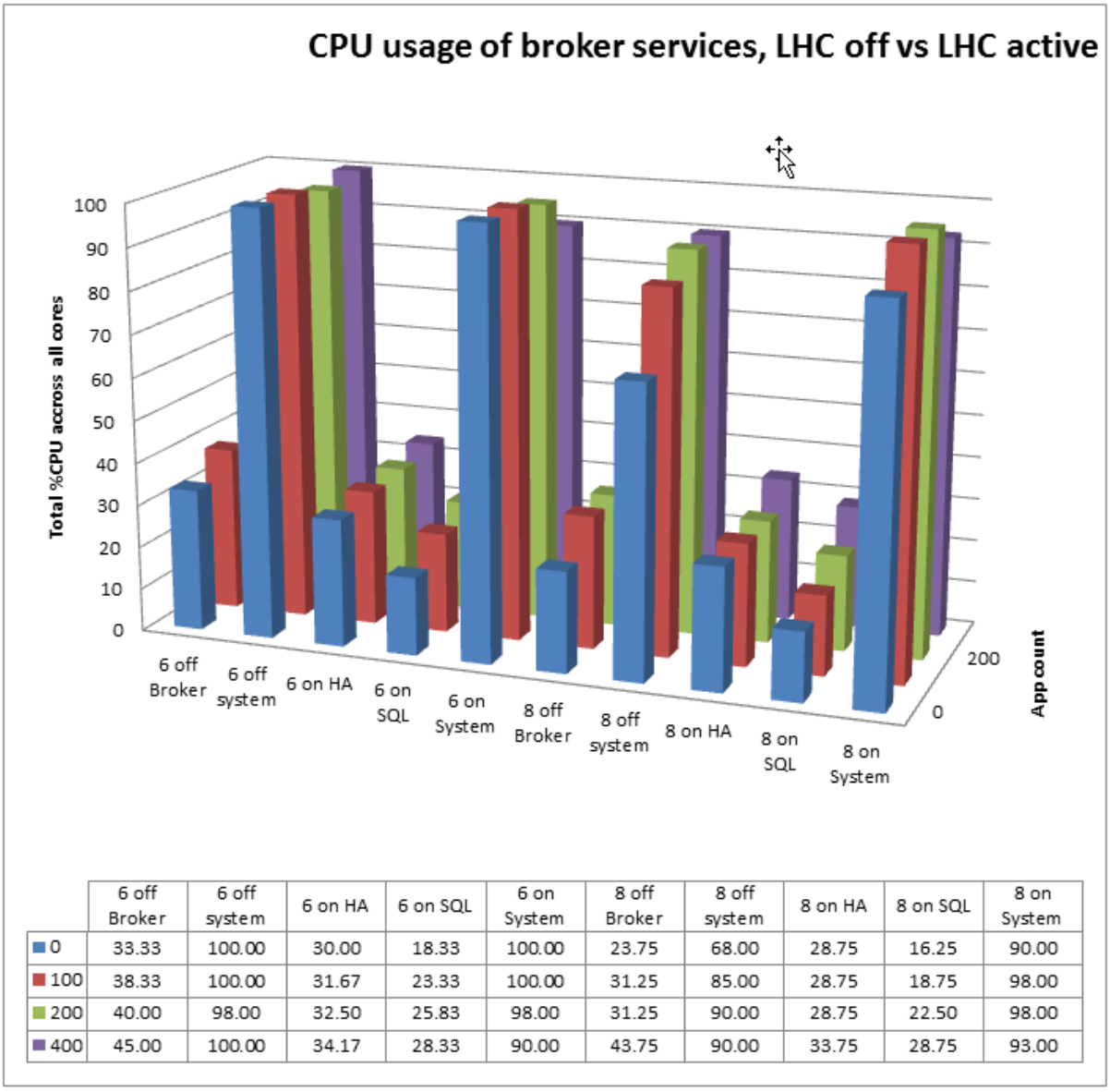


	RDS - 0 apps,100K users	VDI - 0 apps,5K users	VDI - 100 apps,5K users	VDI - 200 apps,5K users	RDS - 100 apps,100K users	RDS - 200 apps,100K users	RDS - 400 apps,100K users
Max	855	896	435	459	947	899	887
Start	217	839	318	329	423	282	258
End test	855	839	439	459	945	893	887

在正常情况下，LocalDB 内存使用量将在操作期间增加，并挂在内存上，除非系统有压力；处理的用户越多，使用的内存越大，最高限为 1 GB。在第一次 VDI 测试中，LocalDB 未释放以前的 RDS 运行的内存。对于后续的 VDI 测试，在测试之前重新启动代理，使用的内存较少。



	RDS - 0 apps,100K users	VDI - 0 apps,5K users	VDI - 100 apps,5K users	VDI - 200 apps,5K users	RDS - 100 apps,100K users	RDS - 200 apps,100K users	RDS - 400 apps,100K users
HA Start	280	835	679	585	918	384	606
Peak	956	1017	1043	1013	1222	1024	969
HA End	940	1016	944	1013	1222	987	969



	6 off Broker	6 off system	6 on HA	6 on SQL	6 on System	8 off Broker	8 off system	8 on HA	8 on SQL	8 on System
0	33.33	100.00	30.00	18.33	100.00	23.75	68.00	28.75	16.25	90.00
100	38.33	100.00	31.67	23.33	100.00	31.25	85.00	28.75	18.75	98.00
200	40.00	98.00	32.50	25.83	98.00	31.25	90.00	28.75	22.50	98.00
400	45.00	100.00	34.17	28.33	90.00	43.75	90.00	33.75	28.75	93.00

注意：值已经归一化为总系统 CPU，因此 33.33% 是六个内核中的两个，在 8 vCPU 设置中，30% 是 2-1/2 内核。

术语 “off” 指的是正常的站点操作；“打开” 表示本地主机缓存处于活动状态。

每秒 20 个用户的请求启动速率对经纪商来说非常苛刻，即使在正常运行下也是如此。在正常操作下，6 vCPU 系统没有额外空间；代理 CPU 随着已发布应用程序数量的增加而增加，从而导致响应时间更慢。当本地主机缓存处于活动状态时，辅助代理 (HA) 服务必须与 LocalDB (SQL) 竞争 CPU 资源，从而提供比正常操作更糟糕的响应时间。

在 8 vCPU 系统中，有一些裕量，LocalDB 将扩大，但在此环境中，尽管仍有少量的 CPU 可用，但在 2-1/2 内核达到顶峰。

在正常操作期间，除非正在发生同步，否则 LocalDB 不会消耗 CPU。

在停机期间，主代理几乎不会使用 CPU。

数据库大小

对于 5075 VDI 配置，LocalDB 约为 40 MB，对于 100000 个 RDS，这在 100-300 MB 之间有所不同，具体取决于应用程序和登录的数量。由于在开始新导入之前获取数据库的副本，因此为 LocalDB 留出 1 GB 的空间。

摘要

在站点数据库中断期间，本地主机缓存支持比连接租赁更广泛的资源和条件，但运行时需要更多的 CPU 和内存。

在多个经纪商站点中，任何经纪商都可能被选为中断代理商，因此所有经纪商都必须有足够的资源来应对中断模式。没有对经纪人资源进行评估，因此在一个经纪人较少和更强大的经纪人的站点中，有可能在停机期间选择最弱的经纪人。

核心和插座的布局必须被视为经纪商设计的一部分。

已发布的应用程序数将影响登录响应时间和最大登录吞吐量。

CPU 资源不足的代理商可能会导致启动失败。

与连接租赁相比，额外的两个内核和 2 GB 的 RAM 是测试本地主机缓存中断模式性能的良好起点。

1 GB 的磁盘空间将足以供 LocalDB 数据库使用。

过载的代理程序将导致连接失败。

这篇文章是由 *Joe Deller* 撰写的。

XenApp 和 **XenDesktop 7.9** 中的 **Citrix** 通用打印服务器负载平衡

May 20, 2020

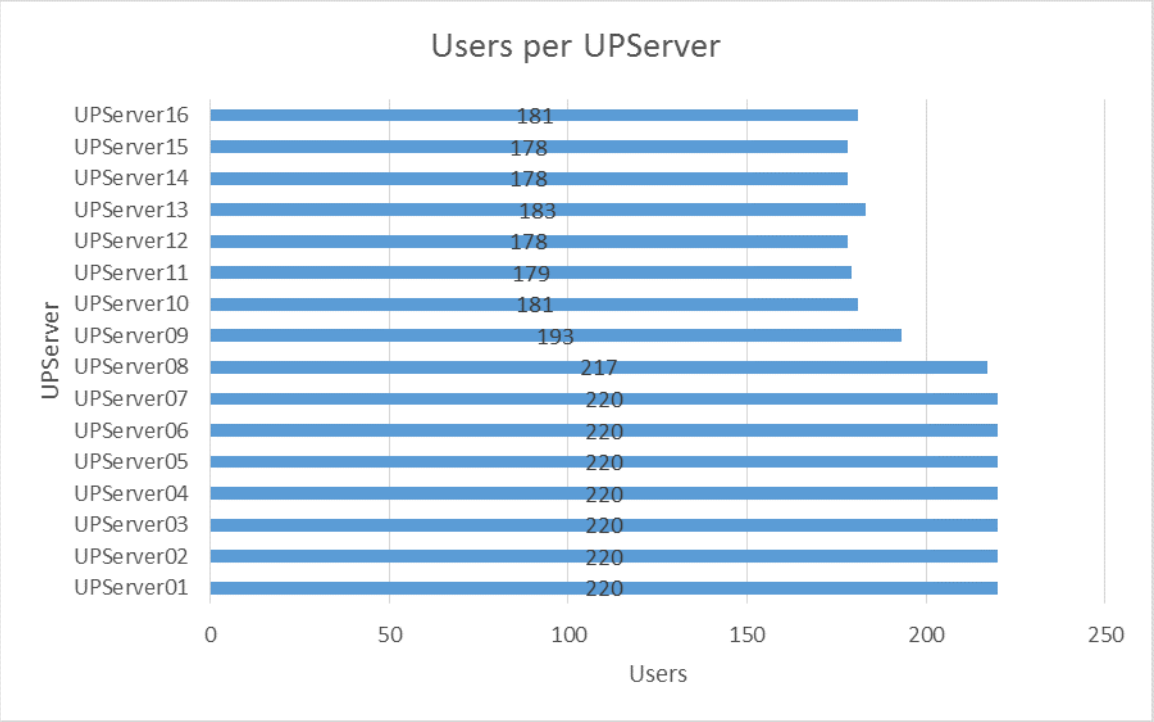
我们如何测试

所有测试场景都是以现实世界的方式进行的，最终端印刷是唯一的模拟组件。测试系统包括 XenServer VM、用于基础结构（DDC、StoreFront 和通用打印服务器）的 Windows 2012R2 和 XenApp VDA 系统、适用于 XenDesktop VDA 系统的 Windows 10 以及用于通过 Citrix Receiver 启动的 ICA 的 Windows 8.1。每个 ICA 启动器连接到 10 台打印机，ICA 启动器之间没有打印机重叠，并使用自定义脚本，以便为每个 XenApp VDA 会话提供一个随机打印机供使用。自定义脚本还使用 AutoIt 来控制每个会话内部的打印，以执行相同的打印操作。最后，我们使用了一个内部开发的工具来协调 ICA 会话的启动并为测试收集绩效数据。

通用打印服务器负载平衡大小

与环境中的所有其他组件一样，大小调整对于通用打印服务器负载均衡以最佳方式执行至关重要。由于打印大型文档是主观的，具体取决于您的需求，本文主要侧重于打印速度和使用我们认为是一项普通工作。

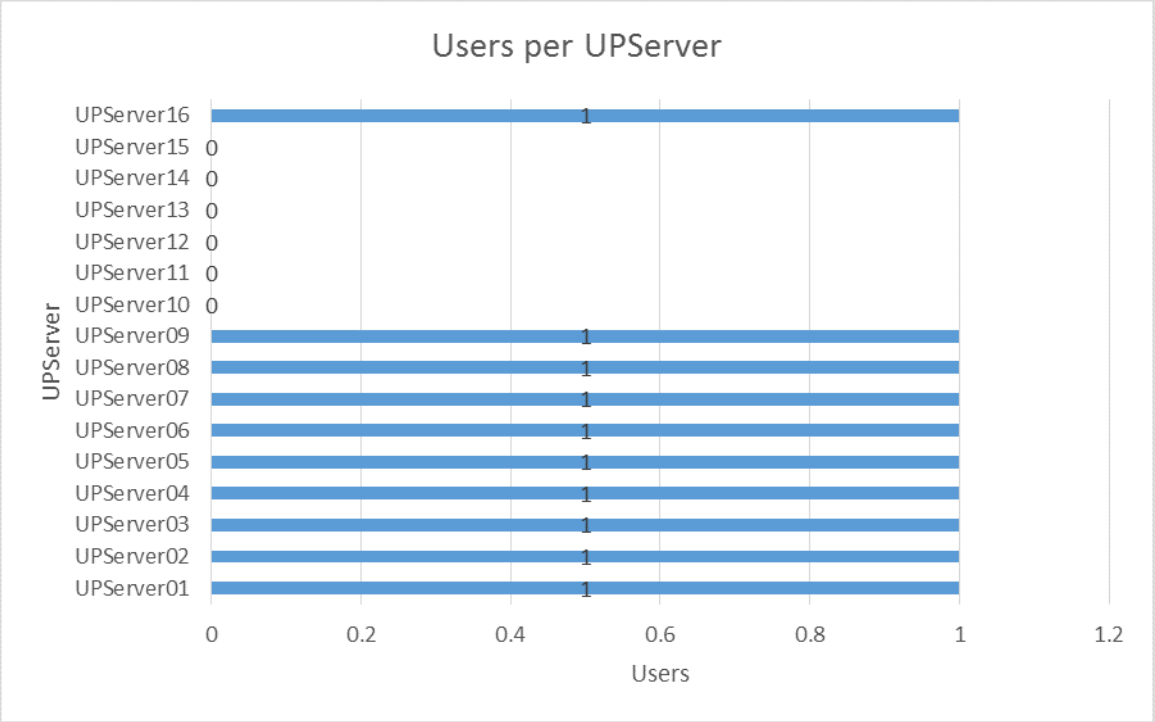
通过内部测试，发现您不希望在负载平衡中设置太多或太少的通用打印服务器实例。虽然这是事实，当您谈论的打印机分发时，您可以使用额外的通用打印服务器实例增加打印效果，但这并不一定是这种情况。特别是，配置了太多通用打印服务器实例的情况。在这种情况下，用户会明确倾斜到第一个可用的通用打印服务器实例。



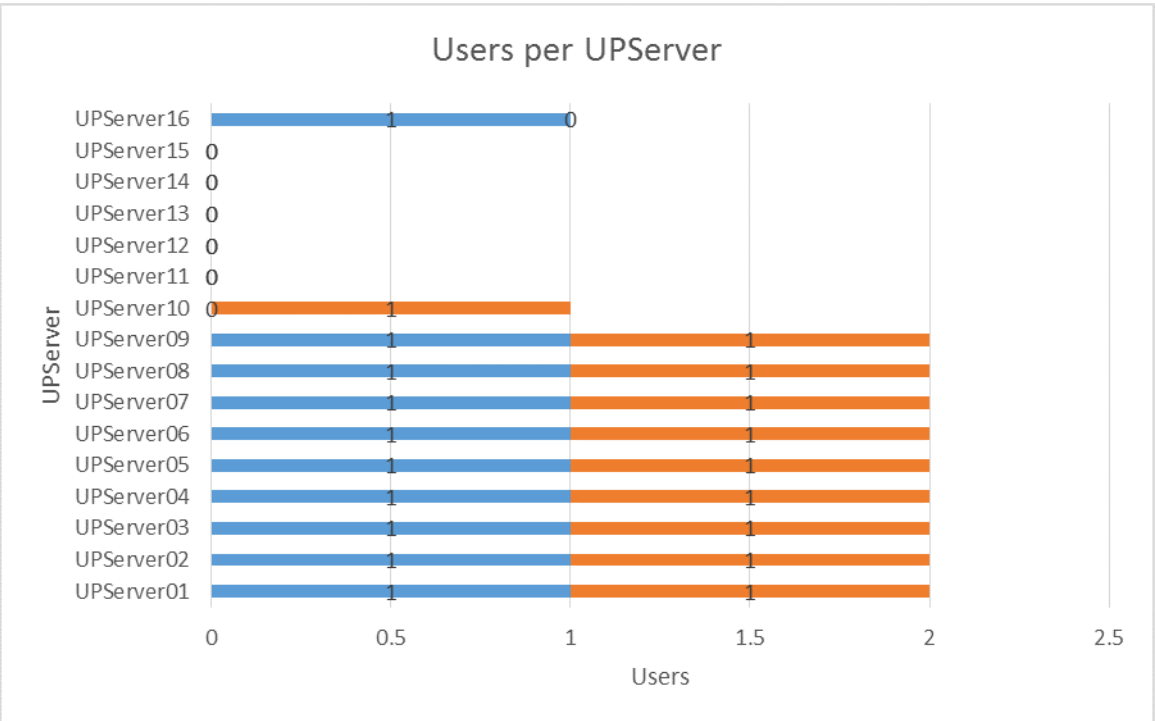
该图形引用了利用 16 个通用打印服务器实例的 48 台 XenApp 服务器上的 3,500 个用户测试方案。如上所示，前 8 个通用打印服务器实例占据了大多数连接，并且连接分布不均匀。这种情况也假定打印率较低，我们将在稍后讨论。

为了理解为什么会出现这种情况，我们需要查看负载平衡机制如何选择通用打印服务器实例。假设我们有 16 个通用打印服务器实例（编号为 1-16）进行负载平衡，一个 XenApp 服务器最多有 10 个用户（稍微夸大，但你会明白为什么）。还假设负载平衡的通用打印服务器实例在负载平衡策略中按数字顺序排列。

当用户登录并创建会话（在 XenApp 中）时，用户接收的通用打印服务器实例是随机的。它可以是任何可用的通用打印服务器实例，并且没有给任何服务器提供首选项。在此示例中，假设他们的连接接收到服务器 16。在该第一个用户登录并完成其会话创建后，另一个用户登录。此用户将利用策略列表中的第一个通用打印服务器实例，在本例中为服务器 1。另一个用户登录，现在正在使用服务器 2。继续这一趋势将提供下面看到的服务器负载。当所有 10 个用户都登录时，通用打印服务器实例 1-9 和 16 将具有连接，而其余服务器则没有连接。



我们现在在组合中添加了一个额外的 XenApp 服务器，最多有 10 个用户。此服务器遵循与上一示例相同的过程，但第一个用户随机接收服务器 4 而不是服务器 16。在这种情况下，每次后续登录都会发生相同的过程，只是它跳过服务器 4，因为当前已建立连接。在这种情况下，服务器 1-10 将具有连接。您可以在下图中以橙色显示其他用户，并观察其平衡方式。



继续增加服务器数量，可以明确观察到发生的偏斜。实际上，您应该拥有一定数量的 XenApp 主机，这些主机与通用打

印服务器实例相同或多个。最好将用户会话计数保留为计划使用的通用打印服务器实例的倍数，以便进行更加优化的加载。从这个角度来看上述情况，使用 2 个 XenApp 服务器和 16 个通用打印服务器实例，我们发现我们的用户负载应至少为每个 XenApp 服务器 16 个用户。另一种查看同样问题的方法是，如果我们每台 XenApp 服务器只支持 10 个用户，则不需要超过 10 个通用打印服务器实例。这将提供更平衡的负载，并将更有效地利用现有资源。

以上是一个过于简单的看待处理严格的用户连接平衡的设置。更复杂的设置，每台服务器有更多的用户可能是看到的配置。随着用户数量的增加，前面提到的打印率在通用打印服务器大小调整方面起着更大的作用。我们建议使用以下公式来确定您的环境所需的通用打印服务器实例。这将使您能够根据所需的打印速率确定负载平衡中所需的通用打印服务器实例的数量。为了帮助简化尺寸，此公式可用于为更理想的设置提供指导，以提供所需的打印率。一般来说，您将最感兴趣的是解析 N 以确定您自己的打印服务器计数。

```
V\*P/N \< J
```

地点：

V = 使用 LB 的 VDA 数量

P = 每个 VDA 每分钟活动网络打印作业的平均数

N = 负载平衡打印服务器的数量

J = 通用打印服务器上每分钟的最大作业数

通过查看 VDA 上的现有通用打印客户端 perfmon 计数器，可在 7.8 和更新的 VDA 上观察到 P，更具体地说，通过监视已启用通用打印服务器策略且网络的 VDA 在正常工作日内为网络打印机创建每分钟计数器的作业的平均值打印机映射到会话。

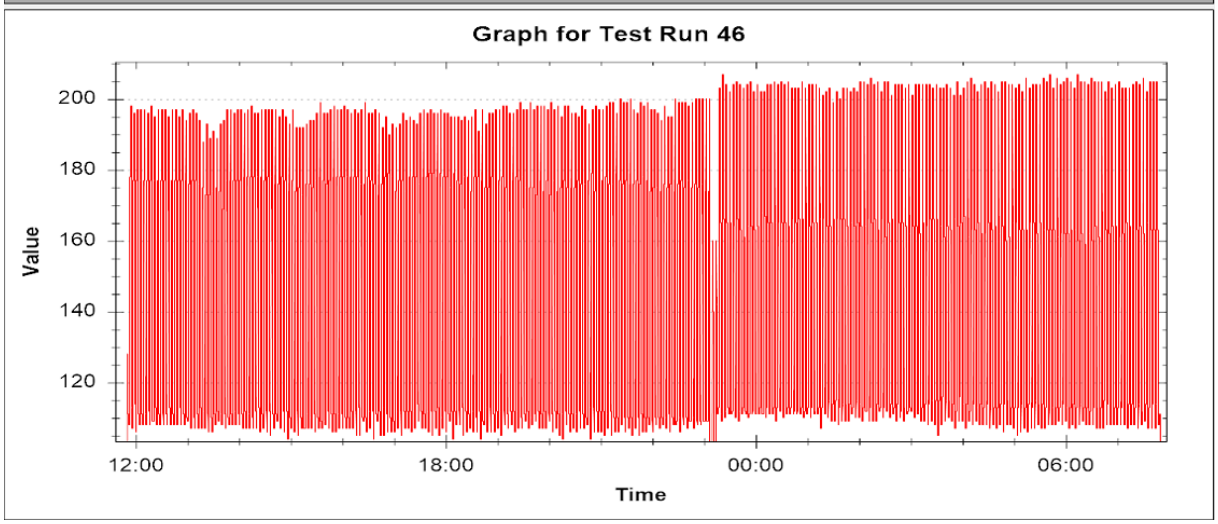
J 应该是介于 50 到 100 之间的数字，具体取决于打印服务器的硬件性能和要打印的文档的大小。

上述公式是广义的，并且在很大程度上取决于您的环境的要求。在实施通用打印服务器负载平衡之前，必须充分了解环境的打印要求。

每分钟 **100** 个作业 **(JPM)** 测试

新的通用打印服务器负载平衡所带来的最大改进之一是每分钟并发打印作业的增加。打印速率阈值为每分钟 100 个作业，现在允许在单个通用打印服务器实例上实现更高的密度。分布在多个负载平衡的通用打印服务器实例中，使得这种增加倍数更具影响力。以下是每分钟创建的任务计数器（公式中引用的计数器）的性能输出，该计数器在 18 小时以上的测试周期内平均值为每分钟约 100 个作业。

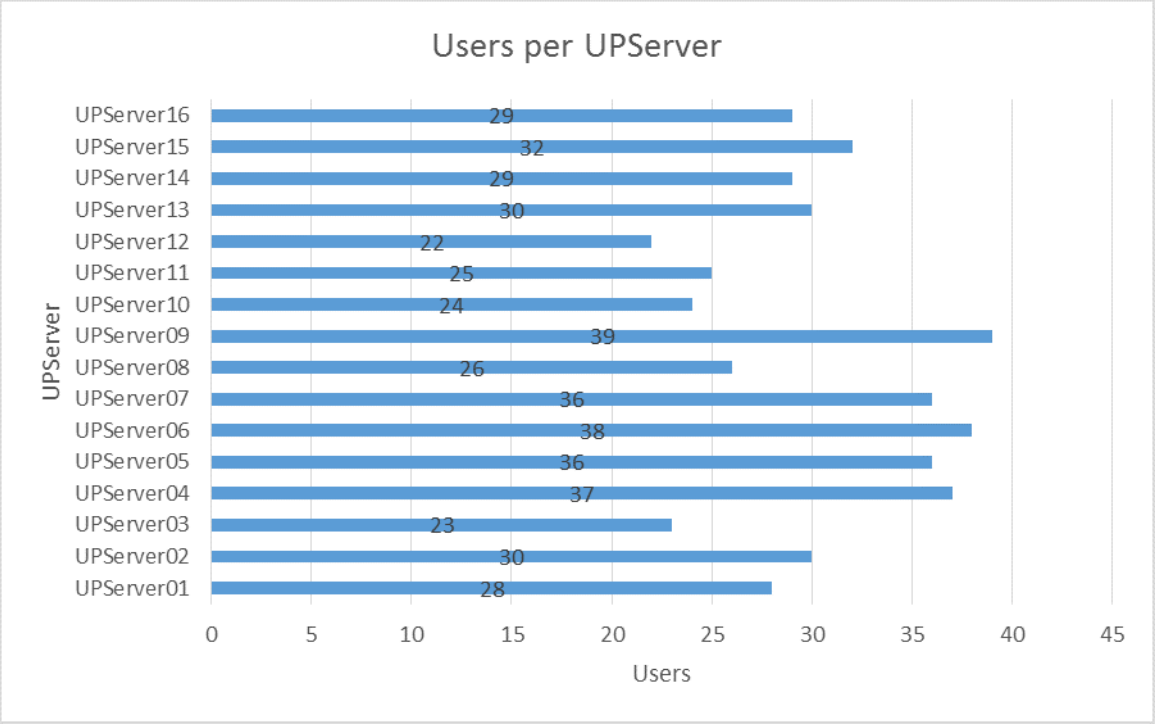
Counter Name	Min	Max	Average
R182UPS-PS003:Universal Print Server -- Jobs Created per Minute -- _UPS Total	0	207	99.2186762778506



VDI 随机化

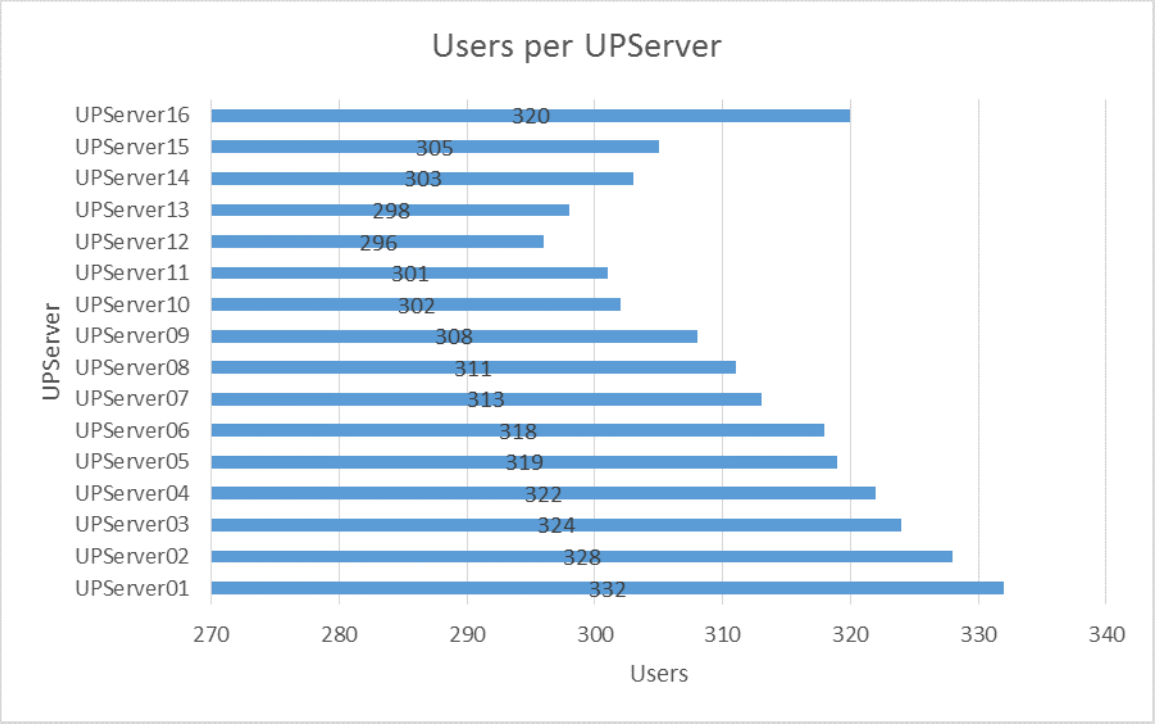
VDI 环境中使用的负载平衡方法仅是随机化函数的实现。如前所述，负载平衡过程仅在 VDA 上进行，随机化第一个连接，然后在通用打印服务器实例列表中进行负载平衡。由于 VDI 实现每个 VDA 都有一个用户，因此随机化函数是唯一适用的。

为确保随机化起作用，使用 16 个通用打印服务器实例执行了 500 用户 XenDesktop 测试。这适用于每个通用打印服务器实例大约 31 个会话（在一个完美的负载平衡的世界中），从而明确确定随机化的有效性。以下是通过此测试创建的打印机连接。可以很容易地观察到这些连接被随机分配给通用打印服务器实例。



5000 用户测试

XenApp 是通用打印服务器负载平衡的最大好处，因为 XenApp 服务器本身的密度很高。为了确定通用打印服务器负载平衡在更大的用户计数环境中扩展的程度，决定了 5000 个用户测试运行足够大，以验证负载平衡。

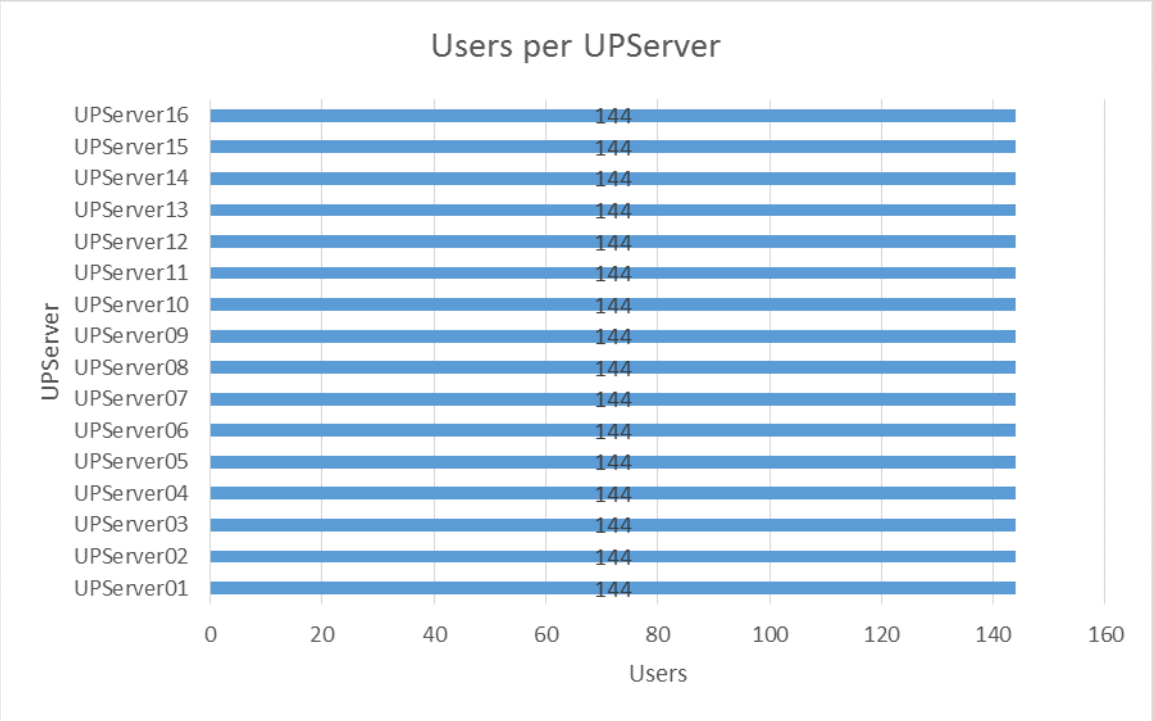


以上是 5000 个用户测试的结果，该测试使用了 48 台 XenApp 服务器和 16 个通用打印服务器实例。每台 XenApp 服务器大约有 100 个用户，或者每台 XenApp 服务器每个通用打印服务器实例约有 6.5 个用户。结果确实显示了前面确定的偏斜，因为这不是一个最佳设计的测试。最终，这是为了演示负载平衡功能。

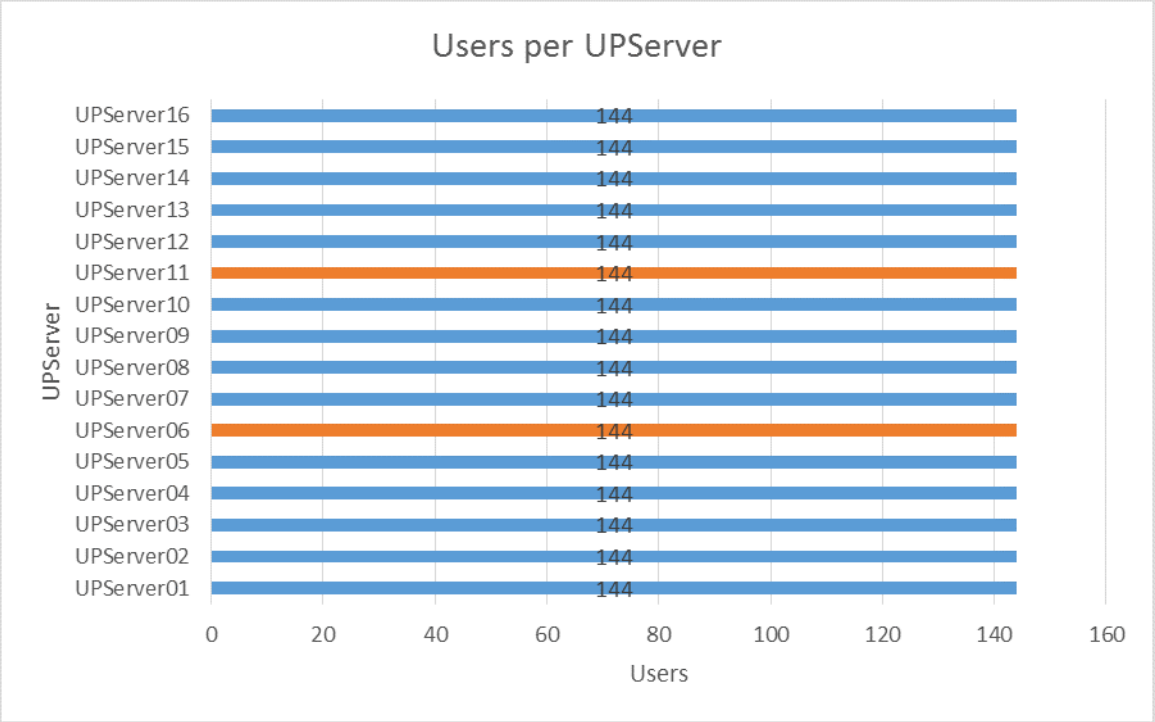
通用打印服务器负载平衡故障转移

默认情况下，通用打印服务器实例在 180 秒最小值时不会报告为失败，并且可能需要长达 360 秒才能被视为失败。此超时非常重要，因为这会导致故障转移不会发生通用打印服务器实例故障的瞬间。通用打印服务器实例将有时间尝试在故障转移发生之前进行恢复。如果需要更快速的故障切换，则需要根据您的环境需求进行修改。这些修改可以通过 Citrix 策略进行，也可以通过两个注册表项进行。

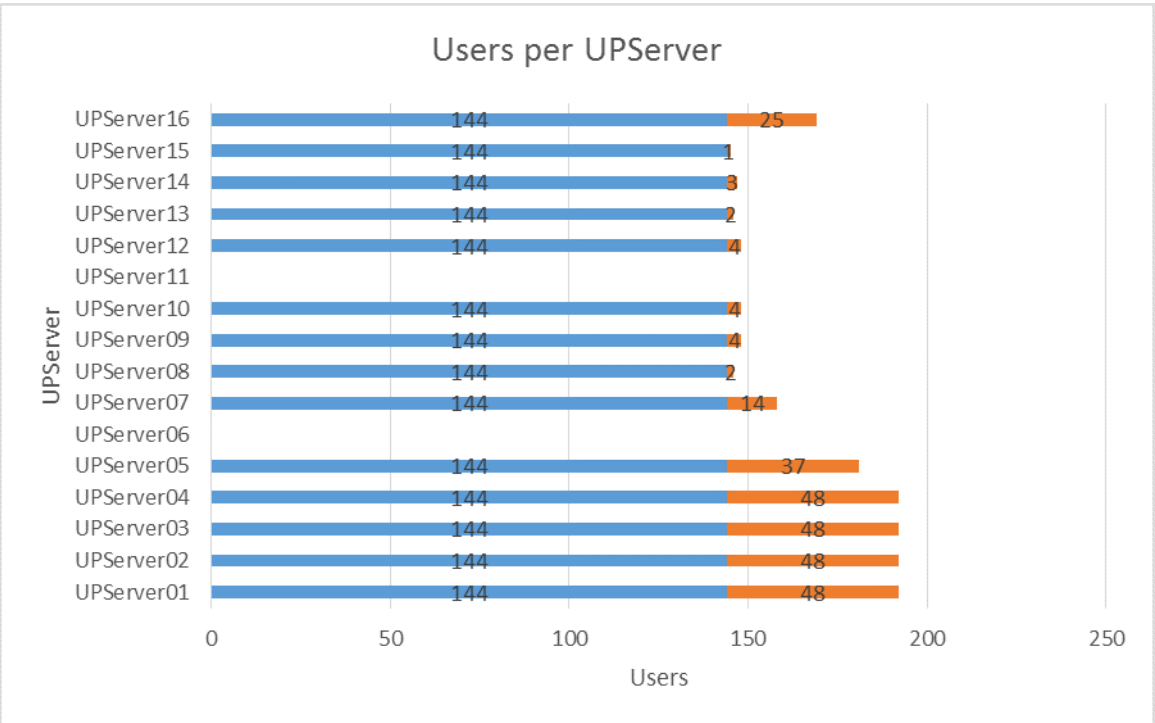
以下是实施故障转移的演示。使用了跨 16 个通用打印服务器实例的 48 台 XenApp 服务器的 2,304 个用户来说明初始负载平衡和后续故障转移。选择了上述值，以便在每个 XenApp 服务器的每个通用打印服务器实例中可以使用 3 个用户（理想情况下）。



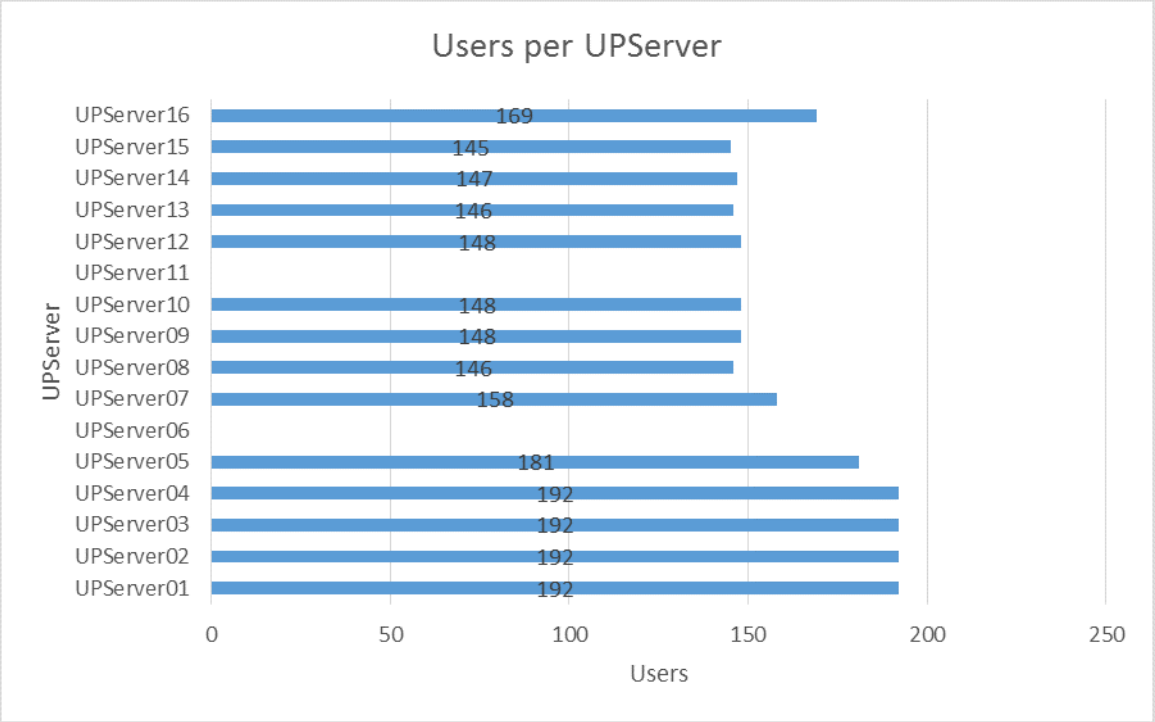
所有通用打印服务器实例均匀加载，并且所有用户都已登录。通用打印服务器实例 UPServer06 和 UPServer11 在虚拟机管理程序级别受到强制关机（由于 PING 用于确定通用打印服务器实例的可用性状态），因此它们将被视为完全失败。下面，受影响的服务器连接以橙色突出显示。接下来，橙色的失败服务器连接将重新分配到仍然启动的剩余通用打印服务器实例。



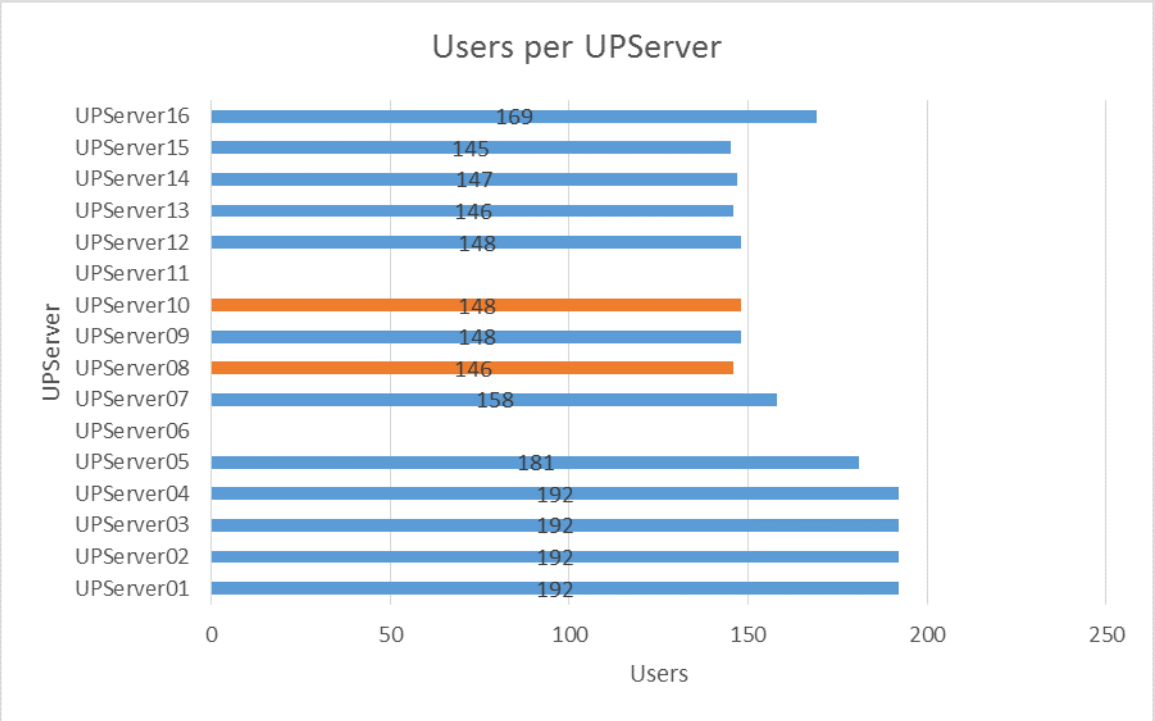
然后，将与仍然可用的现有服务器建立新连接。下面可以看到以前失败的连接如何重新分配到现有服务器。



如前所述，现有连接不会动态负载平衡。负载平衡仅在用户会话登录时进行。因此，当出现故障的通用打印服务器实例再次可用时，不会发生重新平衡或故障恢复。这可以在下面看到，其中已失败的通用打印服务器实例将恢复联机状态，但不采用任何现有连接。



为了说明这些通用打印服务器实例再次可用并将接受连接，有必要登录其他用户或强制执行更多通用打印服务器实例的故障。下面，通用打印服务器实例 UPServer08 和 UPServer10 遭受强制故障，其各自的连接以橙色高亮显示。

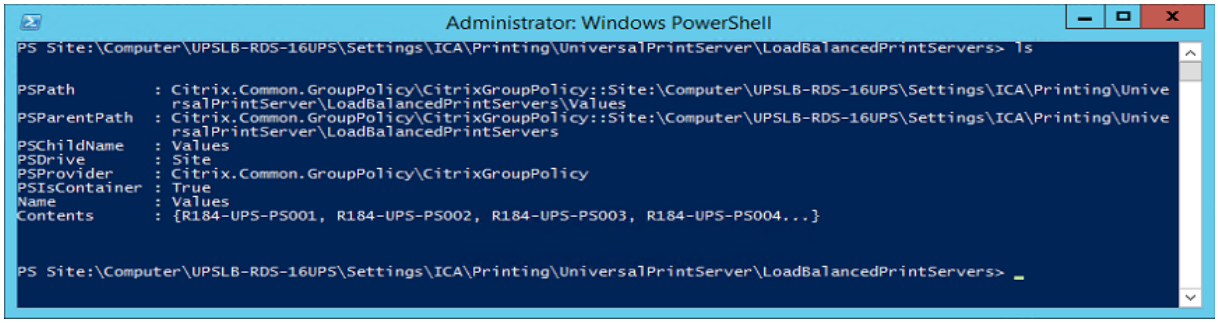


出现故障后，可以相应地看到连接迁移到其他服务器。在这种情况下，它们将被迁移回以前发生故障的两个服务器，现在可以重新连接。由于这些服务器的负载最小（无负载），它们将占用大部分连接。

添加多个打印服务器

可以通过两种方式将多个通用打印服务器实例添加到负载平衡策略中：通过 Citrix 策略 GUI 或通过 PowerShell cmdlet。Citrix 策略 GUI 的使用不言而喻。以下是利用 PowerShell 以更加快速的方式将多个通用打印服务器实例添加到负载平衡策略中的一种方法。

1. `Add-PSSnapin Citrix.Common.GroupPolicy`
2. `New-PSDrive -PSProvider CitrixGroupPolicy -Name Site -Root \ -Controller localhost`
3. CD site:\Computer
4. CD 到要编辑的策略（包含您的 UPSLB 策略的策略名称）
5. CD Settings\ICA\Printing\UniversalPrintServer\LoadBalancedPrintServers
6. 使用 New-Item 将新打印机添加到列表中



虽然使用这种方法有一些警告。首先，请确保您输入的信息正确。您可能需要手动向策略添加几台打印机，并通过 PowerShell 显示屏查看它们的显示方式，以确保正确添加它们。其次，由于您正在绕过策略 UI，因此没有对执行的打印服务器进行验证。

通用打印服务器计数器

正如大小调整部分中提到的，有新的性能计数器可用于确定有关当前打印条件的信息。通用打印服务器实例和 XenApp/XenDesktop 系统上都存在唯一计数器。

与整个通用打印服务器实例相关的计数器将位于相应的通用打印服务器实例上，例如前面提到的每分钟创建的作业计数器（这些计数器仅适用于该通用打印服务器实例，并且不在多个实例）。与通用打印服务器负载平衡相关的计数器存在于每个 XenApp/XenDesktop 系统（负载平衡发生在单个 VDA 级别），例如“当前连接”计数器。

UPClient（VDA 组件）特定计数器可以配置为捕获特定通用打印服务器实例、所有通用打印服务器实例的数据，或作为 VDA 上所有通用打印服务器实例的总和。这些计数器可以直接通过性能查看，也可以通过 PowerShell 编写脚本。以下计数器将位于性能的 Citrix 打印负载平衡部分下方。这些计数器可以进一步选择，方法是为 VDA 选择总计 (_loadbalance cers_ 总数)，或者选择该特定 VDA 上可用的单个通用打印服务器实例（通用打印服务器实例名称）。

活动打印机连接计数器：`Performance\\Citrix Printing Load Balancer (SELECTION)\\Active Printer Connections`

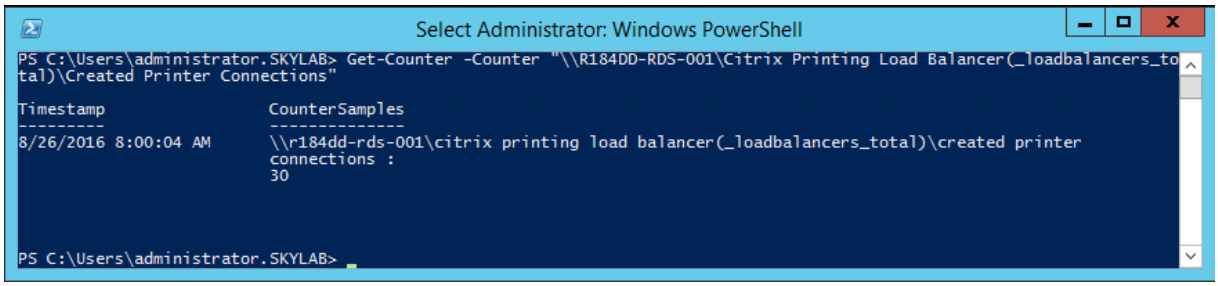
已创建打印机连接计数器：`Performance\\Citrix Printing Load Balancer (SELECTION)\\Created Printer Connections`

已删除打印机连接计数器： `Performance\\Citrix Printing Load Balancer (SELECTION)\\Deleted Printer Connections`

由于这些是标准的性能计数器，因此 PowerShell 内置的获取计数器 cmdlet 可以按如下方式使用来从特定 VDA 获取信息。

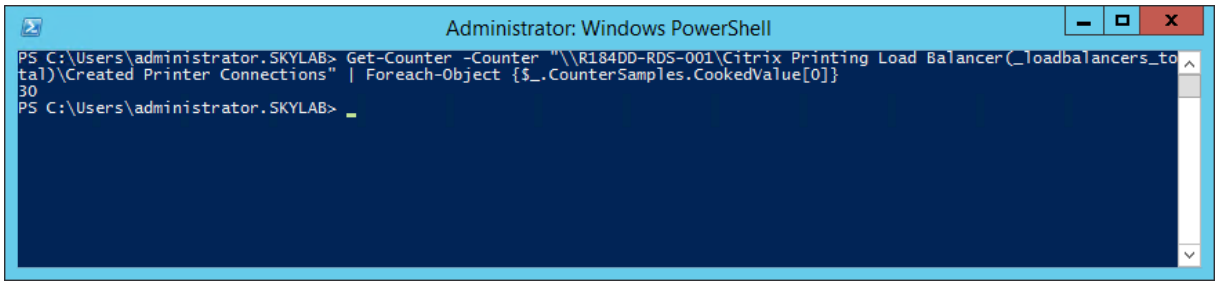
```
Get-Counter -Counter \\\\\VDAName\\Citrix Printing Load Balancer(SELECTION)\\COUNTER
```

上述命令将从所需的 VDAName（VDA 的名称、FQDN 或 IP 地址）中检索所需的计数器（活动/创建/删除的打印机连接）信息（通用打印服务器实例名称或 _loadbalancers_total）。这将提供计数器对象的完整列表。



如果您只关心实际值，则需要将此命令传送到另一个命令中以检索熟值（或只是连接值）。为了实现这一点，我们将添加这个命令到前一个命令的末尾：

```
| Foreach-Object { $_.CounterSamples.CookedValue[0] }
```



测试环境

测试环境由运行 XenServer 6.2 和 6.5 的三组独立硬件池组成。存在一个 XenServer 6.2 池，其中包含 Citrix 基础结构组件（DDC、StoreFront、ICA Launcher、指标集合）和两个 6.5 池，其中包含通用打印服务器实例和测试 RDS VDA。测试使用了两个单独的集中式存储库（每个池版本一个），并且所有测试虚拟机都位于此处。所有使用的软件都是测试时最新的软件。所有测试都使用相同的驱动程序和驱动程序版本进行，以确保结果一致。测试了其他驱动程序；个别结果会因所使用的驱动程序而异。

XenServer 6.2 物理服务器 **(x10)**

- 2 个 Intel Xeon E5620 @ 2.40 GHz（4 核超线程）– 16 个 CPU
- 64 GB 内存
- NFS 存储

XenServer 6.5 物理服务器 **(x25)**

- 2 个 Intel Xeon E5-2640 @ 2.50 GHz（6 核超线程）– 24 个 CPU
- 256 GB 内存
- NFS 存储

通用打印服务器 **VM**

- 16 个 vCPU（16 个插槽 x 1 个核心）
- 16 GB vRAM
- 75 GB 存储空间
- Windows Server 2012 R2

RDS 虚拟机

- 16 个 vCPU（16 个插槽 x 1 个核心）
- 16 GB vRAM
- 75 GB 存储空间
- Windows Server 2012 R2

ICA 启动器虚拟机

- 2 个 vCPU（2 个插槽 x 1 个核心）
- 4 GB vRAM
- 60 GB 存储空间
- Windows 8.1 x64 Enterprise

Citrix 通用打印服务器策略

ICA\Printing

- 通用驱动程序首选项 — XPS;EMF;PCL5c;PCL4;PS
- 通用打印驱动程序使用 — 仅使用通用打印
- 启用通用打印服务器 — 启用时没有回退到 Windows 的本机远程打印
- 等待创建打印机 — 已启用
- 用于负载平衡的通用打印服务器 — 打印服务器列表

通用打印服务器和 RDS/VDA VM 保存在同一硬件池物理服务器中，以确保在一致的硬件配置上执行测试。DDC 和 StoreFront 服务器不包括在上述服务器中，因为它们对通用打印服务器负载平衡没有影响，DDC 的策略传播除外。域中使用了最小策略，XenDesktop/XenApp 站点是默认安装，默认策略除了上述通用打印服务器和负载平衡策略外。

使用 SQL Server 高可用性解决方案时更新数据库连接字符串

May 20, 2020

Citrix 提供了多个 PowerShell 脚本，用于在使用 SQL Server 高可用性数据库解决方案（如始终打开和镜像）时更新 XenApp 和 XenDesktop 数据库连接字符串。

使用 XenApp 和 XenDesktop PowerShell API 的脚本包括：

- **DBConnectionStringFuncs.ps1**：执行实际工作的核心脚本。此脚本包含其他脚本使用的常用函数。
- **Change_XD_Failover_Partner_v1.ps1**：更新（添加、更改或删除）故障转移合作伙伴。此脚本提示输入每个数据库的故障转移伙伴位置 (FQDN)。（提供空白故障切换伙伴将删除故障切换伙伴。您也可以使用 ClearPartner 选项删除合作伙伴。）不要将故障转移伙伴设置为与主体数据库服务器相同的位置。
- **Change_XD_To_ConnectionString.ps1**：使用提供的连接字符串更新到数据库的连接字符串。此脚本可确保某些 Citrix 服务已启动并运行，然后在站点中的所有 Controller 上按正确的顺序更新这些服务。用引号括起每个数据库的连接字符串信息。
- **Change_XD_To_MultiSubnetFailover.ps1**：切换的添加和删除MultiSubnetFailover=true。如果您使用始终在可用性组中，Microsoft 建议连接字符串包括 MultiSubnetFailover=true。此选项可在发生高可用性事件时加快恢复速度，建议用于单子网和多子网环境。运行此脚本一次以添加选项。如果必须删除该选项，请使用 `Change_XD_To_ConnectionString.ps1` 再次运行脚本并提供不带设置的字符串。
- **`Change_XD_To_Null.ps1`**：重置本地主机上的所有连接字符串，因为出现了问题。通过将连接字符串重置为 null，此脚本将控制器置于“初始”状态。如果在运行此脚本后运行 Studio，系统将询问您是要创建站点还是要加入现有站点。如果出现问题并且需要重置，这很有用。重置后，您可以再次尝试使用 Change_XD_To_ConnectionString.ps1 设置连接字符串。

您也可以手动更新数据库连接字符串；请参阅 手动更新字符串。要下载 PowerShell 脚本，请参阅 过程 部分。

要求和注意事项

- 您必须是完整站点管理员才能运行脚本。
- 在一个控制器的 **PowerShell** 窗口中运行这些脚本。PowerShell v3 是必需的。
- 必须安装 XenApp 和 XenDesktop 核心组件，并且站点必须启动并运行。
- 在运行脚本之前，请禁用强制配置日志记录。
- .NET 4.5 及更高版本支持 MultiSubnetFailover 选项。但是，Studio 在 Windows 7 或 2008 R2 计算机上使用的 MMC 包含早期的.NET 版本，因此当您在 **Studio** 导航窗格中选择配置时，您可能会看到错误“不支持关键字: `multisubnetfailover`”。在这种情况下，按如下方式修补或更新：
- 对于.NET 3.5 SP1，使用修补程序 http://support.microsoft.com/kb/2654347 进行修补。
- 对于.NET 4.0，建议更新至 4.0.2. 4.0.3：http://support.microsoft.com/kb/2600211。

然后，使用 `Change_XD_To_MultiSubnetFailover.ps1` 脚本使用此选项更新数据库连接字符串。

过程

1. 从 Citrix ShareFile 中下载包含脚本的 zip 文件。
2. 解压文件。
3. 确保 DBConnectionStringFuncs.ps1 与您运行的脚本位于同一文件夹中，因为您运行的脚本使用 DBConnectionStringFuncs.ps1 中的函数。
4. 在控制器上运行脚本。

如果要将站点设置为多子网故障转移，则只需运行 Change_XD_To_MultiSubnetFailover.ps1 脚本。（请记住：确保 DBConnectionStringFuncs.ps1 脚本位于同一文件夹中。）

> 提示：
>
> - 更新连接字符串时，通常会看到一条消息，指示“`Server=SQLxxx\\CITRIX\\...`”正在更改为 `Data Source=SQLxxx\\CITRIX\\...`。术语“服务器”和“数据源”是同义词。
> - 如果要操作连接字符串，请查看 `Change\\_XD\\_` 脚本如何使用 `DBConnectionStringFuncs.ps1` 中的函数。

手动更新字符串

要手动更新字符串，请运行 XenApp 和 XenDesktop PowerShell 命令行管理程序的命令。

步骤 **1.** 将 SQL 数据库移动到不同的 SQL Server，并分配正确的权限

1. 备份原始 SQL Server 上的数据库，并在新的 SQL Server 上还原它们。
2. 在 **SQL Management Studio > 安全 > 登录**中，添加 Delivery Controller 计算机帐户。例如，CORP\DDC01$。
3. 添加 SQL 登录时，在“用户映射”页面上，单击三个 Citrix 数据库：站点数据库、监视数据库和日志记录数据库。
4. 对于三个 Citrix 数据库中的每个数据库，将 Delivery Controller 计算机帐户添加到各种数据库角色。与日志记录和监视数据库相比，站点数据库具有更多的角色。

```
Site database - ADIdentitySchema_ROLE

Site database - Analytics_ROLE                # for 7.8 and newer
Site database - AppLibrarySchema_ROLE         # for 7.8 and newer
Site database - chr_Broker
Site database - chr_Controller
Site database - ConfigLoggingSiteSchema_ROLE
Site database - ConfigurationSchema_ROLE
Site database - DAS_ROLE
Site database - DesktopUpdateManagerSchema_ROLE
Site database - EnvTestServiceSchema_ROLE
Site database - HostingUnitServiceSchema_ROLE
Site database - Monitor_ROLE
```



```
Site database - OrchestrationSchema_ROLE        # for 7.11 and newer
Site database - public
Site database - StorefrontSchema_ROLE           # for 7.8 and newer
Site database - TrustSchema_ROLE                # for 7.11 and newer
Monitoring database - Monitor_ROLE
Monitoring database - public
Logging database - ConfigLoggingSchema_ROLE
Logging database - public
<!--NeedCopy-->
```

步骤 **2.** 检索现有数据库连接（可选）

运行以下命令查看现有数据库连接字符串：

```
## Load the Citrix snap-ins
asnp Citrix.*

## Get the current Delivery Controller database connections
Get-ConfigDBConnection
Get-AcctDBConnection
Get-AnalyticsDBConnection               #  for 7.6 and newer
Get-AppLibDBConnection                  #  for 7.8 and newer
Get-OrchDBConnection                    #  for 7.11 and newer
Get-TrustDBConnection                   #  for 7.11 and newer
Get-HypDBConnection
Get-ProvDBConnection
Get-BrokerDBConnection
Get-EnvTestDBConnection
Get-SfDBConnection
Get-MonitorDBConnection
Get-MonitorDBConnection -DataStore Monitor
Get-LogDBConnection
Get-LogDBConnection -DataStore Logging
Get-AdminDBConnection
<!--NeedCopy-->
```

步骤 **3.** 删除现有数据库连接

在 **Delivery Controller** 上，以管理员身份打开 PowerShell 并运行以下命令。此过程将清除现有的数据库连接。

```
## Note the state of the log site
Get-LogSite

```



```
## Load the Citrix snap-ins
asnp Citrix.*

## Disable configuration logging for the XD site:
Set-LogSite -State Disabled

## Clear the current Delivery Controller database connections

## Note: AdminDBConnection must be the last command

Set-ConfigDBConnection -DBConnection $null -Force
Set-AcctDBConnection -DBConnection $null -Force
Set-AnalyticsDBConnection -DBConnection $null -Force             # for 7.6 and newer
Set-AppLibDBConnection -DBConnection $null -Force                # for 7.8 and newer
Set-OrchDBConnection -DBConnection $null -Force                  # for 7.11 and newer
Set-TrustDBConnection -DBConnection $null -Force                 # for 7.11 and newer
Set-HypDBConnection -DBConnection $null -Force
Set-ProvDBConnection -DBConnection $null -Force
Set-BrokerDBConnection -DBConnection $null
Set-EnvTestDBConnection -DBConnection $null -Force
Set-SfDBConnection -DBConnection $null -Force
Set-MonitorDBConnection -DataStore Monitor -DBConnection $null -Force
Set-MonitorDBConnection -DBConnection $null -Force
Set-LogDBConnection -DataStore Logging -DBConnection $null -Force
Set-LogDBConnection -DBConnection $null -Force
Set-AdminDBConnection -DBConnection $null -Force
<!--NeedCopy-->
```

如果您看到错误消息，则必须重新启动所有 Citrix 服务。

```
Get-Service Citrix* | Stop-Service -Force
Get-Service Citrix* | Start-Service
<!--NeedCopy-->
```

重新启动 Citrix 服务后，如果仍看到错误，则必须重新启动服务器。重新运行原始命令集以确认现有连接已正确删除。

以下 cmdlet 必须返回空输出：


```
## Load the Citrix snap-ins
asnp Citrix.*

## Get the current Delivery Controller database connections
Get-ConfigDBConnection
Get-AcctDBConnection
Get-AnalyticsDBConnection               #  for 7.6 and newer
Get-AppLibDBConnection                  #  for 7.8 and newer
Get-OrchDBConnection                    #  for 7.11 and newer
Get-TrustDBConnection                   #  for 7.11 and newer
Get-HypDBConnection
Get-ProvDBConnection
Get-BrokerDBConnection
Get-EnvTestDBConnection
Get-SfDBConnection
Get-MonitorDBConnection
Get-LogDBConnection
Get-AdminDBConnection
<!--NeedCopy-->
```

步骤 **4.** 指定新的数据库连接字符串

调整变量以匹配所需的连接字符串。

- 对于独立 SQL Server 连接字符串：`Server=SQLServerName; Initial Catalog=DBName; Integrated Security=True`
- 对于数据库镜像连接字符串：`Server=PrimarySQLServerName; Initial Catalog=DBName; Integrated Security=True; Failover Partner=SecondSQLServer`
- 对于始终保持高可用性：`Server=ListenerName; Initial Catalog=XDdb; Integrated Security=True; MultiSubnetFailover=True`

运行以下命令来设置新的连接字符串。

```
$ServerName = "<dbserver>"
$SiteDBName = "<SiteDbName>"
$LogDBName = "<LoggingDbName>"
$MonitorDBName = "<MonitorDbName>"
$csSite = "Server=$ServerName;Initial Catalog=$SiteDBName;Integrated
    Security=True"
$csLogging = "Server=$ServerName;Initial Catalog=$LogDBName;Integrated
    Security=True"
$csMonitoring = "Server=$ServerName;Initial Catalog=$MonitorDBName;
    Integrated Security=True"

```



```
Set-AdminDBConnection -DBConnection $csSite
Set-ConfigDBConnection -DBConnection $csSite
Set-AcctDBConnection -DBConnection $csSite
Set-AnalyticsDBConnection -DBConnection $csSite              # for 7.6
    and newer
Set-HypDBConnection -DBConnection $csSite
Set-ProvDBConnection -DBConnection $csSite
Set-AppLibDBConnection  -DBConnection $csSite                 # for 7.8
     and newer
Set-OrchDBConnection  -DBConnection $csSite                   # for
    7.11 and newer
Set-TrustDBConnection  -DBConnection $csSite                  # for
    7.11 and newer
Set-BrokerDBConnection -DBConnection $csSite
Set-EnvTestDBConnection -DBConnection $csSite
Set-SfDBConnection -DBConnection $csSite
Set-LogDBConnection -DBConnection $csSite
Set-LogDBConnection -DataStore Logging $null -force
Set-LogDBConnection -DataStore Logging -DBConnection $csLogging
Set-MonitorDBConnection -DBConnection $csSite
Set-MonitorDBConnection -DataStore Monitor -DBConnection $null -force
Set-MonitorDBConnection -DataStore Monitor -DBConnection $csMonitoring

## If necessary, enable configuration logging for the XD site:
Set-LogSite -State Enabled
<!--NeedCopy-->
```

> 注意：
>
> 验证前面的所有Set-<service>DBConnection 命令都返回了“确定”的结果。如果这些命令的结果不是“** 确定”，则可能需要启用日志记录或跟踪以确定连接失败的原因。
>
> Set-LogDBConnection -DBConnection $null 和 Set-MonitorDBConnection -DBConnection $null 返回 **DBUnconfigured** 而非 **OK**。

步骤 **5.** 测试新的数据库连接字符串

1. 运行以下命令以验证与数据库的连接。

```
## Load the Citrix snap-ins
asnp citrix.*

$ServerName = "<dbserver>"
$SiteDBName = "<SiteDbName>"
$LogDBName = "<LoggingDbName>"
```



```
$MonitorDBName = "<MonitorDbName>"
$csSite = "Server=$ServerName;Initial Catalog=$SiteDBName;
    Integrated Security=True"
$csLogging = "Server=$ServerName;Initial Catalog=$LogDBName;
    Integrated Security=True"
$csMonitoring = "Server=$ServerName;Initial Catalog=$MonitorDBName
    ;Integrated Security=True"

Test-AcctDBConnection -DBConnection $csSite
Test-AdminDBConnection -DBConnection $csSite
Test-AnalyticsDBConnection -DBConnection $csSite   # for 7.6 and
    newer
Test-AppLibDBConnection -DBConnection $csSite      # for 7.8 and
    newer
Test-BrokerDBConnection -DBConnection $csSite
Test-ConfigDBConnection -DBConnection $csSite
Test-EnvTestDBConnection -DBConnection $csSite
Test-HypDBConnection -DBConnection $csSite
Test-LogDBConnection -DBConnection $csSite
Test-LogDBConnection -DataStore Logging -DBConnection $csLogging
Test-MonitorDBConnection -DBConnection $csSite
Test-MonitorDBConnection -Datastore Monitor -DBConnection
    $csMonitoring
Test-OrchDBConnection -DBConnection $csSite       # for 7.11 and
    newer
Test-ProvDBConnection -DBConnection $csSite
Test-SfDBConnection -DBConnection $csSite
Test-TrustDBConnection -DBConnection $csSite      # for 7.11 and
    newer
<!--NeedCopy-->
```

2. 重新启动 Citrix Studio。

更多信息

- 如何配置独立 SQL Server、数据库镜像和始终高可用性

基于 **Active Directory OU** 的控制器发现

May 20, 2020

此 Delivery Controller 发现方法主要支持向后兼容，并且仅适用于 Windows 桌面操作系统的 Virtual Delivery Agent (VDA)，不适用于 Windows 服务器操作系统的 VDA。

有关可以配置使 VDA 能够向控制器注册的其他方法的信息（包括推荐的方法），请参阅向控制器注册 VDA。

基于 Active Directory 的发现要求站点中的所有计算机都是域的成员，并且控制器使用的域与桌面使用的域之间具有相互信任关系。如果您使用此方法，则必须在每个桌面注册表中配置 OU 的 GUID。

要执行基于 OU 的 Controller 发现，请在 Controller 上运行 **Set-ADControllerDiscovery.ps1** PowerShell 脚本（每个 Controller 都在文件夹 $Env:ProgramFiles\Citrix\Broker\Service\Setup Scripts 中包含此脚本）。要运行脚本，必须对父 OU 具有 CreateChild 权限，以及完全的管理权限。

如果希望桌面通过 Active Directory 发现站点中的控制器，在创建站点时，必须在 Active Directory 中创建相应的组织单位 (OU)。可以在包含您计算机的林的任何域中创建该 OU。最佳做法是，将站点中的控制器也包含在该 OU 内，但这不是强制或必需的条件。具有适当权限的域管理员可以创建 OU 作为一个空容器，然后通过该 OU 将管理权限委派给 Citrix 管理员。

该脚本会创建多个关键的对象。仅创建和使用标准 Active Directory 对象。扩展架构是没有必要的。

- 一个控制器安全组。站点中所有控制器的计算机帐户必须是此安全组中的成员。只有当控制器是此安全组的成员时，站点中的桌面才会接受来自这些控制器的数据。

  确保所有 Controller 在运行 VDA 的所有虚拟桌面上都具有“从网络访问此计算机”权限。为此，您可以向控制器安全组授予此权限。如果控制器没有此权限，VDA 将不会注册。

- 一个服务连接点 (SCP) 对象，包含有关站点的信息（例如站点名称）。如果使用 Active Directory 用户和计算机管理工具检查站点 OU，则可能需要在“视图”菜单中启用“高级功能”才能查看 SCP 对象。

- 一个名为 RegistrationServices 的容器，该容器在站点 OU 中创建。容器中针对站点内的每个控制器包含一个 SCP 对象。每次启动控制器时，都会验证其 SCP 内容并根据需要进行更新。

如果多个管理员可能在初始安装后添加和删除 Controller，他们需要在 RegistrationServices 容器上创建和删除子项的权限，以及 Controller 安全组上的写入属性。这些权限自动授予运行 Set-ADControllerDiscovery.ps1 脚本的管理员。域管理员或原始安装管理员可以授予这些权限，Citrix 建议您设置安全组来执行此任务。

如果使用站点 OU：

- 只有在安装或卸载此软件，或者控制器启动并需要更新其 SCP 中的信息时（例如，由于重命名了控制器或更改了通信端口），信息才会写入 Active Directory。默认情况下，Set-ADControllerDiscovery.ps1 脚本对站点 OU 中的对象设置适当的权限，授予每个控制器对其 SCP 的写入访问权限。站点 OU 中对象的内容用于在桌面与控制器之间建立信任关系。请确保：
  - 只有经过授权的管理员才可以使用安全组的访问控制列表 (ACL) 在控制器安全组中添加或删除计算机。
  - 只有经过授权的管理员和各自的控制器才可以更改控制器 SCP 中的信息。
- 如果您的部署使用复制，请注意潜在延迟。有关详细信息，请参阅 Microsoft 文档。如果要在域控制器位于多个 Active Directory 站点的域中创建站点 OU，这一点尤其重要。如果您在最初创建站点 OU 时、安装或卸载控制器时或者更改控制器名称或通信端口时，对 Active Directory 进行了更改，则在信息被复制到适当的域控制器之前，这些更改可能对桌面不可见，具体取决于桌面、控制器和域控制器的位置。此类复制延迟的表现包括桌面无法与控制器建立通信，因此不可用于用户连接。

- 此软件在 Active Directory 中使用多个标准的计算机对象属性来管理桌面。根据您的部署，存储于桌面的 Active Directory 记录中的计算机对象完全限定的域名，可以包含在返回给用户用以建立连接的连接设置中。请确保此信息与 DNS 环境中的信息一致。

要使用基于 OU 的 Controller 发现将 Controller 移动到另一个站点，请按照上述说明移动 Controller。从旧站点（步骤 2）中删除控制器后，运行 PowerShell 脚本 **Set-ADControllerDiscovery –sync**。该脚本会将 OU 与控制器的当前设置进行同步。加入现有站点（步骤 3）后，在新站点中的所有控制器上运行相同的脚本。

基于 **OU** 的发现所需的权限

要创建站点，运行脚本的 Citrix 管理员必须具有在整个站点 OU 中创建对象（SCP、容器和安全组）的权限。

如果站点 OU 不存在，则管理员也必须具有创建该站点的权限。Citrix 建议 AD 域管理员预先创建该 OU，并将权限委派给 Citrix 站点管理员。此外，脚本还可以创建站点 OU。为此，管理员需要新 OU 的父 OU 上的“创建 OU”权限。但是，如上所述，Citrix 不建议这样做。

稍后，要从站点中添加或删除 Controller，Citrix 管理员必须具有从安全组中添加/删除计算机以及创建/删除 SCP 的权限。

在正常操作过程中，Controller 和 VDA 需要对 OU 及子 OU 中的所有对象具有读取权限。VDA 将 OU 作为其自己的计算机标识进行访问；该计算机标识至少需要该 OU 的读取权限，从而能够发现 Controller。Controller 也需要具有在容器中自己的 SCP 对象上设置属性的权限。

通过将 Citrix 管理员完整权限授予子 OU，可准许所有这些操作。但是，如果您的部署有更严格的安全要求（例如限制谁可以使用脚本执行哪些操作），则可以使用“控制委派”向导来设置特定权限。以下示例过程授予了创建站点的权限。

1. 创建一个 OU 以包含子对象（服务连接点 (SCP)、容器和安全组）。
2. 选择 OU，然后右键单击并选择委派控制。
3. 在“控制委派”向导中，指定可为该 OU 委派控制的域用户。
4. 在“要委派的任务”页上，选择“创建要委派的自定义任务”。
5. 在 **Active Directory** 对象类型页面上，接受默认的此文件夹、此文件夹中的现有对象以及在此文件夹中创建新对象。
6. 在“权限”页上，选中“写入和创建所有子对象”复选框。
7. 完成向导以确认权限。

Locations
Corporate Headquarters | 851 Cypress Creek Road Fort Lauderdale, FL 33309, United States
Silicon Valley | 4988 Great America Parkway Santa Clara, CA 95054, United States